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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 May 2015 (14.05.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/069092 A1

(51) **International Patent Classification**:
H04N 21/234 (2011.01) *H04J 11/00* (2006.01)
H04N 21/236 (2011.01)

(21) **International Application Number**:
PCT/KR2014/010818

(22) **International Filing Date**:
11 November 2014 (11.11.2014)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:

61/902,300	11 November 2013 (11.11.2013)	US
61/902,302	11 November 2013 (11.11.2013)	US
61/902,303	11 November 2013 (11.11.2013)	US
61/902,304	11 November 2013 (11.11.2013)	US

(71) **Applicant**: **LG ELECTRONICS INC.** [KR/KR]; 128, Yeoui-daero, Yeongdeungpo-gu, Seoul 150-721 (KR).

(72) **Inventors**: **BAEK, Jongseob**; 19, Yangjae-daero 11gil, Seocho-gu, Seoul 137-893 (KR). **SHIN, Jongwoong**; 19, Yangjae-daero 11gil, Seocho-gu, Seoul 137-893 (KR). **KIM, Jaehyung**; 19, Yangjae-daero 11gil, Seocho-gu, Seoul 137-893 (KR). **KO, Woosuk**; 19, Yangjae-daero 11-gil, Seocho-gu, Seoul 137-893 (KR). **HONG, Sungryong**; 19, Yangjae-daero 11gil, Seocho-gu, Seoul 137-893 (KR).

(74) **Agents**: **KIM, Yong In** et al.; KBK & Associates, 7th Floor, 82, Olympic-ro, Songpa-ku, Seoul 138-861 (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, KR, 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.

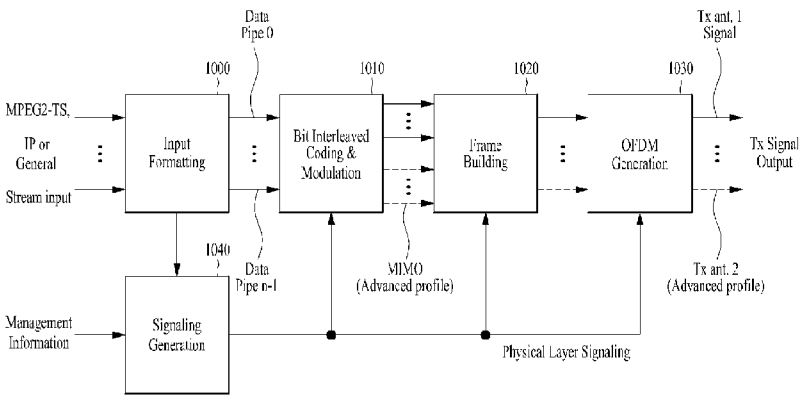
(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, 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))*

WO 2015/069092 A1

(54) **Title**: APPARATUS FOR TRANSMITTING BROADCAST SIGNALS, APPARATUS FOR RECEIVING BROADCAST SIGNALS, METHOD FOR TRANSMITTING BROADCAST SIGNALS AND METHOD FOR RECEIVING BROADCAST SIGNALS



(57) **Abstract**: The present invention provides an apparatus for transmitting broadcast signals. The apparauts includes an encoder for encoding service data, a mapper for mapping the encoded service data into a plurality of OFDM symbols to build at least one signal frame, a modulator for modulating data in the built at least one signal frame by an OFDM (Orthogonal Frequency Division Multi - plex) scheme and a transmitter for transmitting the broadcast signals having the modulated data.

Description

Title of Invention: APPARATUS FOR TRANSMITTING BROADCAST SIGNALS, APPARATUS FOR RECEIVING BROADCAST SIGNALS, METHOD FOR TRANSMITTING BROADCAST SIGNALS AND METHOD FOR RECEIVING BROADCAST SIGNALS

Technical Field

[1] The present invention relates to an apparatus for transmitting broadcast signals, an apparatus for receiving broadcast signals and methods for transmitting and receiving broadcast signals.

Background Art

[2] As analog broadcast signal transmission comes to an end, various technologies for transmitting/receiving digital broadcast signals are being developed. A digital broadcast signal may include a larger amount of video/audio data than an analog broadcast signal and further include various types of additional data in addition to the video/audio data.

[3] That is, a digital broadcast system can provide HD (high definition) images, multi-channel audio and various additional services. However, data transmission efficiency for transmission of large amounts of data, robustness of transmission/reception networks and network flexibility in consideration of mobile reception equipment need to be improved for digital broadcast.

Disclosure of Invention

Technical Problem

[4] An object of the present invention is to provide an apparatus and method for transmitting broadcast signals to multiplex data of a broadcast transmission/reception system providing two or more different broadcast services in a time domain and transmit the multiplexed data through the same RF signal bandwidth and an apparatus and method for receiving broadcast signals corresponding thereto.

[5] Another object of the present invention is to provide an apparatus for transmitting broadcast signals, an apparatus for receiving broadcast signals and methods for transmitting and receiving broadcast signals to classify data corresponding to services by components, transmit data corresponding to each component as a data pipe, receive and process the data

[6] Still another object of the present invention is to provide an apparatus for transmitting broadcast signals, an apparatus for receiving broadcast signals and

methods for transmitting and receiving broadcast signals to signal signaling information necessary to provide broadcast signals.

Solution to Problem

[7] To achieve the object and other advantages and in accordance with the purpose of the invention, as embodied and broadly described herein, the present invention provides a method for transmitting broadcast signals. The method of transmitting broadcast signals includes encoding service data, building at least one signal frame including the encoded service data, wherein the at least one signal frame includes a plurality of OFDM symbols, modulating data in the built at least one signal frame by an OFDM (Orthogonal Frequency Division Multiplex) scheme and transmitting the broadcast signals having the modulated data.

Advantageous Effects of Invention

[8] The present invention can process data according to service characteristics to control QoS (Quality of Services) for each service or service component, thereby providing various broadcast services.

[9] The present invention can achieve transmission flexibility by transmitting various broadcast services through the same RF signal bandwidth.

[10] The present invention can improve data transmission efficiency and increase robustness of transmission/reception of broadcast signals using a MIMO system.

[11] According to the present invention, it is possible to provide broadcast signal transmission and reception methods and apparatus capable of receiving digital broadcast signals without error even with mobile reception equipment or in an indoor environment.

Brief Description of Drawings

[12] The accompanying drawings, which are included to provide a further understanding of the invention and are incorporated in and constitute a part of this application, illustrate embodiment(s) of the invention and together with the description serve to explain the principle of the invention. In the drawings:

[13] FIG. 1 illustrates a structure of an apparatus for transmitting broadcast signals for future broadcast services according to an embodiment of the present invention.

[14] FIG. 2 illustrates an input formatting block according to one embodiment of the present invention.

[15] FIG. 3 illustrates an input formatting block according to another embodiment of the present invention.

[16] FIG. 4 illustrates an input formatting block according to another embodiment of the present invention.

[17] FIG. 5 illustrates a BICM block according to an embodiment of the present

invention.

[18] FIG. 6 illustrates a BICM block according to another embodiment of the present invention.

[19] FIG. 7 illustrates a frame building block according to one embodiment of the present invention.

[20] FIG. 8 illustrates an OFMD generation block according to an embodiment of the present invention.

[21] FIG. 9 illustrates a structure of an apparatus for receiving broadcast signals for future broadcast services according to an embodiment of the present invention.

[22] FIG. 10 illustrates a frame structure according to an embodiment of the present invention.

[23] FIG. 11 illustrates a signaling hierarchy structure of the frame according to an embodiment of the present invention.

[24] FIG. 12 illustrates preamble signaling data according to an embodiment of the present invention.

[25] FIG. 13 illustrates PLS1 data according to an embodiment of the present invention.

[26] FIG. 14 illustrates PLS2 data according to an embodiment of the present invention.

[27] FIG. 15 illustrates PLS2 data according to another embodiment of the present invention.

[28] FIG. 16 illustrates a logical structure of a frame according to an embodiment of the present invention.

[29] FIG. 17 illustrates PLS mapping according to an embodiment of the present invention.

[30] FIG. 18 illustrates EAC mapping according to an embodiment of the present invention.

[31] FIG. 19 illustrates FIC mapping according to an embodiment of the present invention.

[32] FIG. 20 illustrates a type of DP according to an embodiment of the present invention.

[33] FIG. 21 illustrates DP mapping according to an embodiment of the present invention.

[34] FIG. 22 illustrates an FEC structure according to an embodiment of the present invention.

[35] FIG. 23 illustrates a bit interleaving according to an embodiment of the present invention.

[36] FIG. 24 illustrates a cell-word demultiplexing according to an embodiment of the present invention.

[37] FIG. 25 illustrates a time interleaving according to an embodiment of the present invention.

[38] FIG. 26 illustrates the basic operation of a twisted row-column block interleaver

according to an embodiment of the present invention.

[39] FIG. 27 illustrates an operation of a twisted row-column block interleaver according to another embodiment of the present invention.

[40] FIG. 28 illustrates a diagonal-wise reading pattern of a twisted row-column block interleaver according to an embodiment of the present invention.

[41] FIG. 29 illustrates interlaved XFECBLOCKs from each interleaving array according to an embodiment of the present invention.

[42] FIG. 30 is a view illustrating an operation of a frequency interleaver 7020 according to an embodiment of the present invention.

[43] FIG. 31 illustrates a basic switch model for MUX and DEMUX procedures according to an embodiment of the present invention.

[44] FIG. 32 is a view illustrating a concept of frequency interleaving applied to a single super-frame according to an embodiment of the present invention.

[45] FIG. 33 is a view illustrating logical operation mechanism of frequency interleaving applied to a single super-frame according to an embodiment of the present invention.

[46] FIG. 34 illustrates math figures of logical operation mechanism of frequency interleaving applied to a single super-frame according to an embodiment of the present invention.

[47] FIG. 35 illustrates an operation of a memory bank according to an embodiment of the present invention.

[48] FIG. 36 illustrates a frequency deinterleaving procedure according to an embodiment of the present invention.

[49] FIG. 37 is a view illustrates concept of frequency interleaving applied to a single signal frame according to an embodiment of the present invention.

[50] FIG. 38 is a view illustrating logical operation mechanism of frequency interleaving applied to a single signal frame according to an embodiment of the present invention.

[51] FIG. 39 illustrates math figures of logical operation mechanism of frequency interleaving applied to a single signal frame according to an embodiment of the present invention.

[52] FIG. 40 is a view illustrating single-memory deinterleaving for input sequential OFDM symbols.

[53] FIG. 41 is a view illustrating an output signal of a time interleaver according to an embodiment of the present invention.

[54] FIG. 42 is a view of a 4K FFT mode random seed generator according to an embodiment of the present invention.

[55] FIG. 43 illustrates math figures representing an operation of a 4K FFT mode random seed generator according to an embodiment of the present invention.

[56] FIG. 44 is a view illustrating a 4K FFT mode random symbol-offset generator

according to an embodiment of the present invention.

[57] FIG. 45 illustrates math figures showing operations of a random symbol-offset generator and a random Symbol-offset generator for 4K FFT mode including a 0 bits-spreader and a 12 bits-PN generator according to an embodiment of the present invention.

[58] FIG. 46 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 4K FFT mode including a 1 bits-spreader and an 11 bits-PN generator according to an embodiment of the present invention.

[59] FIG. 47 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 4K FFT mode including a 2 bits-spreader and a 10 bits-PN generator according to an embodiment of the present invention.

[60] FIG. 48 is a view illustrating logical composition of a 4K FFT mode random seed generator according to an embodiment of the present invention.

[61] FIG. 49 is a view illustrating an output signal of a time interleaver according to another embodiment of the present invention.

[62] FIG. 50 is a view illustrating a 4K FFT mode quasi-random interleaving-seed generator according to an embodiment of the present invention.

[63] FIG. 51 is math figures representing operations of 4K FFT mode bit shuffling and 4K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[64] FIG. 52 is a view illustrating logical composition of a 4K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[65] FIG. 53 is a view of an 8K FFT mode random seed generator according to an embodiment of the present invention.

[66] FIG. 54 illustrates math figures representing an operation of an 8K FFT mode random seed generator according to an embodiment of the present invention.

[67] FIG. 55 is a view illustrating an 8K FFT mode random symbol-offset generator according to an embodiment of the present invention.

[68] FIG. 56 illustrates math figures showing operations of a random symbol-offset generator and a random Symbol-offset generator for 8K FFT mode including a 0 bits-spreader and a 13 bits-PN generator according to an embodiment of the present invention.

[69] FIG. 57 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 8K FFT mode including a 1 bits-spreader and an 12 bits-PN generator according to an embodiment of the present invention.

[70] FIG. 58 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 8K FFT mode including a 2 bits-spreader and an 11 bits-PN generator according to an embodiment of the present invention.

[71] FIG. 59 is a view illustrating logical composition of an 8K FFT mode random seed generator according to an embodiment of the present invention.

[72] FIG. 60 is a view illustrating an 8K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[73] FIG. 61 is math figures representing operations of 8K FFT mode bit shuffling and 8K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[74] FIG. 62 is a view illustrating logical composition of an 8K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[75] FIG. 63 is a view of a 16K FFT mode random seed generator according to an embodiment of the present invention.

[76] FIG. 64 illustrates math figures representing an operation of a 16K FFT mode random seed generator according to an embodiment of the present invention.

[77] FIG. 65 is a view illustrating a 16K FFT mode random symbol-offset generator according to an embodiment of the present invention.

[78] FIG. 66 illustrates math figures showing operations of a random symbol-offset generator and a random Symbol-offset generator for 16K FFT mode including a 0 bits-spreader and a 14 bits-PN generator according to an embodiment of the present invention.

[79] FIG. 67 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 16K FFT mode including a 1 bits-spreader and a 13 bits-PN generator according to an embodiment of the present invention.

[80] FIG. 68 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 16K FFT mode including a 2 bits-spreader and a 12 bits-PN generator according to an embodiment of the present invention.

[81] FIG. 69 is a view illustrating logical composition of a 16K FFT mode random seed generator according to an embodiment of the present invention.

[82] FIG. 70 is a view illustrating a 16K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[83] FIG. 71 is math figures representing operations of 16K FFT mode bit shuffling and 16K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[84] FIG. 72 is a view illustrating logical composition of a 16K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[85] FIG. 73 is a view of a 32K FFT mode random seed generator according to an embodiment of the present invention.

[86] FIG. 74 illustrates math figures representing an operation of a 32K FFT mode random seed generator according to an embodiment of the present invention.

[87] FIG. 75 is a view illustrating a 32K FFT mode random symbol-offset generator according to an embodiment of the present invention.

[88] FIG. 76 illustrates math figures showing operations of a random symbol-offset generator and a random Symbol-offset generator for 32K FFT mode including a 0 bits-spreader and a 15 bits-PN generator according to an embodiment of the present invention.

[89] FIG. 77 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 32K FFT mode including a 1 bits-spreader and a 14 bits-PN generator according to an embodiment of the present invention.

[90] FIG. 78 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 32K FFT mode including a 2 bits-spreader and a 13 bits-PN generator according to an embodiment of the present invention.

[91] FIG. 79 is a view illustrating logical composition of a 32K FFT mode random seed generator according to an embodiment of the present invention.

[92] FIG. 80 is a view illustrating a 32K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[93] FIG. 81 is math figures representing operations of 32K FFT mode bit shuffling and 32K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[94] FIG. 82 is a view illustrating logical composition of a 32K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[95] FIG. 83 is a change procedure for an interleaving seed in each memory bank according to another embodiment of the present invention.

[96] FIG. 84 is a view of a 4K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[97] FIG. 85 illustrates math figures representing an operation of a 4K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[98] FIG. 86 is a view illustrating logical composition of a 4K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[99] FIG. 87 is a view illustrating a 4K FFT mode random interleaving-sequence

generator according to another embodiment of the present invention.

[100] FIG. 88 is math figures representing operations of 4K FFT mode bit shuffling and 4K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[101] FIG. 89 is a view of an8K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[102] FIG. 90 illustrates math figures representing an operation of an8K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[103] FIG. 91 is a view illustrating logical composition of an 8K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[104] FIG. 92 is a view illustrating an 8K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[105] FIG. 93 is math figures representing operations of 8K FFT mode bit shuffling and 8K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[106] FIG. 94 is a view of a 16K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[107] FIG. 95 illustrates math figures representing an operation of a 16K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[108] FIG. 96 is a view illustrating logical composition of a 16K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[109] FIG. 97 is a view illustrating a 16K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[110] FIG. 98 is math figures representing operations of 16K FFT mode bit shuffling and 16K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[111] FIG. 99 is a view of a 32K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[112] FIG. 100 illustrates math figures representing an operation of a 32K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[113] FIG. 101 is a view illustrating logical composition of a 32K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[114] FIG. 102 is a view illustrating a 32KFFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[115] FIG. 103 is math figures representing operations of 32K FFT mode bit shuffling and 32K FFT mode random interleaving-sequence generator according to another em-

bodiment of the present invention.

[116] FIG. 104 is a flowchart illustrating a method for transmitting broadcast signals according to an embodiment of the present invention.

[117] FIG. 105 is a flowchart illustrating a method for receiving broadcast signals according to an embodiment of the present invention.

Best Mode for Carrying out the Invention

[118] Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings. The detailed description, which will be given below with reference to the accompanying drawings, is intended to explain exemplary embodiments of the present invention, rather than to show the only embodiments that can be implemented according to the present invention. The following detailed description includes specific details in order to provide a thorough understanding of the present invention. However, it will be apparent to those skilled in the art that the present invention may be practiced without such specific details.

[119] Although most terms used in the present invention have been selected from general ones widely used in the art, some terms have been arbitrarily selected by the applicant and their meanings are explained in detail in the following description as needed. Thus, the present invention should be understood based upon the intended meanings of the terms rather than their simple names or meanings.

[120] The present invention provides apparatuses and methods for transmitting and receiving broadcast signals for future broadcast services. Future broadcast services according to an embodiment of the present invention include a terrestrial broadcast service, a mobile broadcast service, a UHDTV service, etc.

[121] The apparatuses and methods for transmitting according to an embodiment of the present invention may be categorized into a base profile for the terrestrial broadcast service, a handheld profile for the mobile broadcast service and an advanced profile for the UHDTV service. In this case, the base profile can be used as a profile for both the terrestrial broadcast service and the mobile broadcast service. That is, the base profile can be used to define a concept of a profile which includes the mobile profile. This can be changed according to intention of the designer.

[122] The present invention may process broadcast signals for the future broadcast services through non-MIMO (Multiple Input Multiple Output) or MIMO according to one embodiment. A non-MIMO scheme according to an embodiment of the present invention may include a MISO (Multiple Input Single Output) scheme, a SISO (Single Input Single Output) scheme, etc.

[123] While MISO or MIMO uses two antennas in the following for convenience of de-

scription, the present invention is applicable to systems using two or more antennas.

[124] The present invention may defines three physical layer (PL) profiles (base, handheld and advanced profiles) each optimized to minimize receiver complexity while attaining the performance required for a particular use case. The physical layer (PHY) profiles are subsets of all configurations that a corresponding receiver should implement.

[125] The three PHY profiles share most of the functional blocks but differ slightly in specific blocks and/or parameters. Additional PHY profiles can be defined in the future. For the system evolution, future profiles can also be multiplexed with the existing profiles in a single RF channel through a future extension frame (FEF). The details of each PHY profile are described below.

[126] 1. Base profile

[127] The base profile represents a main use case for fixed receiving devices that are usually connected to a roof-top antenna. The base profile also includes portable devices that could be transported to a place but belong to a relatively stationary reception category. Use of the base profile could be extended to handheld devices or even vehicular by some improved implementations, but those use cases are not expected for the base profile receiver operation.

[128] Target SNR range of reception is from approximately 10 to 20dB, which includes the 15dB SNR reception capability of the existing broadcast system (e.g. ATSC A/53). The receiver complexity and power consumption is not as critical as in the battery-operated handheld devices, which will use the handheld profile. Key system parameters for the base profile are listed in below table 1.

[129] Table 1

[Table 1]

LDPC codeword length	16K, 64K bits
Constellation size	4~10 bpcu (bits per channel use)
Time de-interleaving memory size	≤ 2^{19} data cells
Pilot patterns	Pilot pattern for fixed reception
FFT size	16K, 32K points

[130] 2. Handheld profile

[131] The handheld profile is designed for use in handheld and vehicular devices that operate with battery power. The devices can be moving with pedestrian or vehicle speed. The power consumption as well as the receiver complexity is very important for the implementation of the devices of the handheld profile. The target SNR range of the handheld profile is approximately 0 to 10dB, but can be configured to reach below 0dB when intended for deeper indoor reception.

[132] In addition to low SNR capability, resilience to the Doppler Effect caused by receiver mobility is the most important performance attribute of the handheld profile. Key

system parameters for the handheld profile are listed in the below table 2.

[133] Table 2

[Table 2]

LDPC codeword length	16K bits
Constellation size	2~8 bpcu
Time de-interleaving memory size	≤ 2^{18} data cells
Pilot patterns	Pilot patterns for mobile and indoor reception
FFT size	8K, 16K points

[134] 3. Advanced profile

[135] The advanced profile provides highest channel capacity at the cost of more implementation complexity. This profile requires using MIMO transmission and reception, and UHDTV service is a target use case for which this profile is specifically designed. The increased capacity can also be used to allow an increased number of services in a given bandwidth, e.g., multiple SDTV or HDTV services.

[136] The target SNR range of the advanced profile is approximately 20 to 30dB. MIMO transmission may initially use existing elliptically-polarized transmission equipment, with extension to full-power cross-polarized transmission in the future. Key system parameters for the advanced profile are listed in below table 3.

[137] Table 3

[Table 3]

LDPC codeword length	16K, 64K bits
Constellation size	8~12 bpcu
Time de-interleaving memory size	≤ 2^{19} data cells
Pilot patterns	Pilot pattern for fixed reception
FFT size	16K, 32K points

[138] In this case, the base profile can be used as a profile for both the terrestrial broadcast service and the mobile broadcast service. That is, the base profile can be used to define a concept of a profile which includes the mobile profile. Also, the advanced profile can be divided advanced profile for a base profile with MIMO and advanced profile for a handheld profile with MIMO. Moreover, the three profiles can be changed according to intention of the designer.

[139] The following terms and definitions may apply to the present invention. The following terms and definitions can be changed according to design.

[140] auxiliary stream: sequence of cells carrying data of as yet undefined modulation and coding, which may be used for future extensions or as required by broadcasters or network operators

[141] base data pipe: data pipe that carries service signaling data

[142] baseband frame (or BBFRAME): set of *K*bch bits which form the input to one FEC

encoding process (BCH and LDPC encoding)

[143] cell: modulation value that is carried by one carrier of the OFDM transmission

[144] coded block: LDPC-encoded block of PLS1 data or one of the LDPC-encoded blocks of PLS2 data

[145] data pipe: logical channel in the physical layer that carries service data or related metadata, which may carry one or multiple service(s) or service component(s).

[146] data pipe unit: a basic unit for allocating data cells to a DP in a frame.

[147] data symbol: OFDM symbol in a frame which is not a preamble symbol (the frame signaling symbol and frame edge symbol is included in the data symbol)

[148] DP_ID: this 8bit field identifies uniquely a DP within the system identified by the SYSTEM_ID

[149] dummy cell: cell carrying a pseudorandom value used to fill the remaining capacity not used for PLS signaling, DPs or auxiliary streams

[150] emergency alert channel: part of a frame that carries EAS information data

[151] frame: physical layer time slot that starts with a preamble and ends with a frame edge symbol

[152] frame repetition unit: a set of frames belonging to same or different physical layer profile including a FEF, which is repeated eight times in a super-frame

[153] fast information channel: a logical channel in a frame that carries the mapping information between a service and the corresponding base DP

[154] FECBLOCK: set of LDPC-encoded bits of a DP data

[155] FFT size: nominal FFT size used for a particular mode, equal to the active symbol period T_s expressed in cycles of the elementary period T

[156] frame signaling symbol: OFDM symbol with higher pilot density used at the start of a frame in certain combinations of FFT size, guard interval and scattered pilot pattern, which carries a part of the PLS data

[157] frame edge symbol: OFDM symbol with higher pilot density used at the end of a frame in certain combinations of FFT size, guard interval and scattered pilot pattern

[158] frame-group: the set of all the frames having the same PHY profile type in a super-frame.

[159] future extension frame: physical layer time slot within the super-frame that could be used for future extension, which starts with a preamble

[160] Futurecast UTB system: proposed physical layer broadcasting system, of which the input is one or more MPEG2-TS or IP or general stream(s) and of which the output is an RF signal

[161] input stream: A stream of data for an ensemble of services delivered to the end users by the system.

[162] normal data symbol: data symbol excluding the frame signaling symbol and the

frame edge symbol

[163] PHY profile: subset of all configurations that a corresponding receiver should implement

[164] PLS: physical layer signaling data consisting of PLS1 and PLS2

[165] PLS1: a first set of PLS data carried in the FSS symbols having a fixed size, coding and modulation, which carries basic information about the system as well as the parameters needed to decode the PLS2

[166] NOTE: PLS1 data remains constant for the duration of a frame-group.

[167] PLS2: a second set of PLS data transmitted in the FSS symbol, which carries more detailed PLS data about the system and the DPs

[168] PLS2 dynamic data: PLS2 data that may dynamically change frame-by-frame

[169] PLS2 static data: PLS2 data that remains static for the duration of a frame-group

[170] preamble signaling data: signaling data carried by the preamble symbol and used to identify the basic mode of the system

[171] preamble symbol: fixed-length pilot symbol that carries basic PLS data and is located in the beginning of a frame

[172] NOTE: The preamble symbol is mainly used for fast initial band scan to detect the system signal, its timing, frequency offset, and FFTsize.

[173] reserved for future use: not defined by the present document but may be defined in future

[174] superframe: set of eight frame repetition units

[175] time interleaving block (TI block): set of cells within which time interleaving is carried out, corresponding to one use of the time interleaver memory

[176] TI group: unit over which dynamic capacity allocation for a particular DP is carried out, made up of an integer, dynamically varying number of XFECBLOCKs.

[177] NOTE: The TI group may be mapped directly to one frame or may be mapped to multiple frames. It may contain one or more TI blocks.

[178] Type 1 DP: DP of a frame where all DPs are mapped into the frame in TDM fashion

[179] Type 2 DP: DP of a frame where all DPs are mapped into the frame in FDM fashion

[180] XFECBLOCK: set of *N*cells cells carrying all the bits of one LDPC FECBLOCK

[181] FIG. 1 illustrates a structure of an apparatus for transmitting broadcast signals for future broadcast services according to an embodiment of the present invention.

[182] The apparatus for transmitting broadcast signals for future broadcast services according to an embodiment of the present invention can include an input formatting block 1000, a BICM (Bit interleaved coding & modulation) block 1010, a frame structure block 1020, an OFDM (Orthogonal Frequency Division Multiplexing) generation block 1030 and a signaling generation block 1040. A description will be given of the operation of each module of the apparatus for transmitting broadcast

signals.

[183] IP stream/packets and MPEG2-TS are the main input formats, other stream types are handled as General Streams. In addition to these data inputs, Management Information is input to control the scheduling and allocation of the corresponding bandwidth for each input stream. One or multiple TS stream(s), IP stream(s) and/or General Stream(s) inputs are simultaneously allowed.

[184] The input formatting block 1000 can demultiplex each input stream into one or multiple data pipe(s), to each of which an independent coding and modulation is applied. The data pipe (DP) is the basic unit for robustness control, thereby affecting quality-of-service (QoS). One or multiple service(s) or service component(s) can be carried by a single DP. Details of operations of the input formatting block 1000 will be described later.

[185] The data pipe is a logical channel in the physical layer that carries service data or related metadata, which may carry one or multiple service(s) or service component(s).

[186] Also, the data pipe unit: a basic unit for allocating data cells to a DP in a frame.

[187] In the BICM block 1010, parity data is added for error correction and the encoded bit streams are mapped to complex-value constellation symbols. The symbols are interleaved across a specific interleaving depth that is used for the corresponding DP. For the advanced profile, MIMO encoding is performed in the BICM block 1010 and the additional data path is added at the output for MIMO transmission. Details of operations of the BICM block 1010 will be described later.

[188] The Frame Building block 1020 can map the data cells of the input DPs into the OFDM symbols within a frame. After mapping, the frequency interleaving is used for frequency-domain diversity, especially to combat frequency-selective fading channels. Details of operations of the Frame Building block 1020 will be described later.

[189] After inserting a preamble at the beginning of each frame, the OFDM Generation block 1030 can apply conventional OFDM modulation having a cyclic prefix as guard interval. For antenna space diversity, a distributed MISO scheme is applied across the transmitters. In addition, a Peak-to-Average Power Reduction (PAPR) scheme is performed in the time domain. For flexible network planning, this proposal provides a set of various FFT sizes, guard interval lengths and corresponding pilot patterns. Details of operations of the OFDM Generation block 1030 will be described later.

[190] The Signaling Generation block 1040 can create physical layer signaling information used for the operation of each functional block. This signaling information is also transmitted so that the services of interest are properly recovered at the receiver side. Details of operations of the Signaling Generation block 1040 will be described later.

[191] FIGS. 2, 3 and 4 illustrate the input formatting block 1000 according to embodiments of the present invention. A description will be given of each figure.

[192] FIG. 2 illustrates an input formatting block according to one embodiment of the present invention. FIG. 2 shows an input formatting module when the input signal is a single input stream.

[193] The input formatting block illustrated in FIG. 2 corresponds to an embodiment of the input formatting block 1000 described with reference to FIG. 1.

[194] The input to the physical layer may be composed of one or multiple data streams. Each data stream is carried by one DP. The mode adaptation modules slice the incoming data stream into data fields of the baseband frame (BBF). The system supports three types of input data streams: MPEG2-TS, Internet protocol (IP) and Generic stream (GS). MPEG2-TS is characterized by fixed length (188 byte) packets with the first byte being a sync-byte (0x47). An IP stream is composed of variable length IP datagram packets, as signaled within IP packet headers. The system supports both IPv4 and IPv6 for the IP stream. GS may be composed of variable length packets or constant length packets, signaled within encapsulation packet headers.

[195] (a) shows a mode adaptation block 2000 and a stream adaptation 2010 for signal DP and (b) shows a PLS generation block 2020 and a PLS scrambler 2030 for generating and processing PLS data. A description will be given of the operation of each block.

[196] The Input Stream Splitter splits the input TS, IP, GS streams into multiple service or service component (audio, video, etc.) streams. The mode adaptation module 2010 is comprised of a CRC Encoder, BB (baseband) Frame Slicer, and BB Frame Header Insertion block.

[197] The CRC Encoder provides three kinds of CRC encoding for error detection at the user packet (UP) level, i.e., CRC-8, CRC-16, and CRC-32. The computed CRC bytes are appended after the UP. CRC-8 is used for TS stream and CRC-32 for IP stream. If the GS stream doesn't provide the CRC encoding, the proposed CRC encoding should be applied.

[198] BB Frame Slicer maps the input into an internal logical-bit format. The first received bit is defined to be the MSB. The BB Frame Slicer allocates a number of input bits equal to the available data field capacity. To allocate a number of input bits equal to the BBF payload, the UP packet stream is sliced to fit the data field of BBF.

[199] BB Frame Header Insertion block can insert fixed length BBF header of 2 bytes is inserted in front of the BB Frame. The BBF header is composed of STUFFI (1 bit), SYNCD (13 bits), and RFU (2 bits). In addition to the fixed 2-Byte BBF header, BBF can have an extension field (1 or 3 bytes) at the end of the 2-byte BBF header.

[200] The stream adaptation 2010 is comprised of stuffing insertion block and BB scrambler.

[201] The stuffing insertion block can insert stuffing field into a payload of a BB frame. If the input data to the stream adaptation is sufficient to fill a BB-Frame, STUFFI is set to

'0' and the BBF has no stuffing field. Otherwise STUFFI is set to '1' and the stuffing field is inserted immediately after the BBF header. The stuffing field comprises two bytes of the stuffing field header and a variable size of stuffing data.

[202] The BB scrambler scrambles complete BBF for energy dispersal. The scrambling sequence is synchronous with the BBF. The scrambling sequence is generated by the feed-back shift register.

[203] The PLS generation block 2020 can generate physical layer signaling (PLS) data. The PLS provides the receiver with a means to access physical layer DPs. The PLS data consists of PLS1 data and PLS2 data.

[204] The PLS1 data is a first set of PLS data carried in the FSS symbols in the frame having a fixed size, coding and modulation, which carries basic information about the system as well as the parameters needed to decode the PLS2 data. The PLS1 data provides basic transmission parameters including parameters required to enable the reception and decoding of the PLS2 data. Also, the PLS1 data remains constant for the duration of a frame-group.

[205] The PLS2 data is a second set of PLS data transmitted in the FSS symbol, which carries more detailed PLS data about the system and the DPs. The PLS2 contains parameters that provide sufficient information for the receiver to decode the desired DP. The PLS2 signaling further consists of two types of parameters, PLS2 Static data (PLS2-STAT data) and PLS2 dynamic data (PLS2-DYN data). The PLS2 Static data is PLS2 data that remains static for the duration of a frame-group and the PLS2 dynamic data is PLS2 data that may dynamically change frame-by-frame.

[206] Details of the PLS data will be described later.

[207] The PLS scrambler 2030 can scramble the generated PLS data for energy dispersal.

[208] The above-described blocks may be omitted or replaced by blocks having similar or identical functions.

[209] FIG. 3 illustrates an input formatting block according to another embodiment of the present invention.

[210] The input formatting block illustrated in FIG. 3 corresponds to an embodiment of the input formatting block 1000 described with reference to FIG. 1.

[211] FIG. 3 shows a mode adaptation block of the input formatting block when the input signal corresponds to multiple input streams.

[212] The mode adaptation block of the input formatting block for processing the multiple input streams can independently process the multiple input streams.

[213] Referring to FIG. 3, the mode adaptation block for respectively processing the multiple input streams can include an input stream splitter 3000, an input stream synchronizer 3010, a compensating delay block 3020, a null packet deletion block 3030, a head compression block 3040, a CRC encoder 3050, a BB frame slicer 3060 and a BB

header insertion block 3070. Description will be given of each block of the mode adaptation block.

[214] Operations of the CRC encoder 3050, BB frame slicer 3060 and BB header insertion block 3070 correspond to those of the CRC encoder, BB frame slicer and BB header insertion block described with reference to FIG. 2 and thus description thereof is omitted.

[215] The input stream splitter 3000 can split the input TS, IP, GS streams into multiple service or service component (audio, video, etc.) streams.

[216] The input stream synchronizer 3010 may be referred as ISSY. The ISSY can provide suitable means to guarantee Constant Bit Rate (CBR) and constant end-to-end transmission delay for any input data format. The ISSY is always used for the case of multiple DPs carrying TS, and optionally used for multiple DPs carrying GS streams.

[217] The compensating delay block 3020 can delay the split TS packet stream following the insertion of ISSY information to allow a TS packet recombining mechanism without requiring additional memory in the receiver.

[218] The null packet deletion block 3030, is used only for the TS input stream case. Some TS input streams or split TS streams may have a large number of null-packets present in order to accommodate VBR (variable bit-rate) services in a CBR TS stream. In this case, in order to avoid unnecessary transmission overhead, null-packets can be identified and not transmitted. In the receiver, removed null-packets can be re-inserted in the exact place where they were originally by reference to a deleted null-packet (DNP) counter that is inserted in the transmission, thus guaranteeing constant bit-rate and avoiding the need for time-stamp (PCR) updating.

[219] The head compression block 3040 can provide packet header compression to increase transmission efficiency for TS or IP input streams. Because the receiver can have a priori information on certain parts of the header, this known information can be deleted in the transmitter.

[220] For Transport Stream, the receiver has a-priori information about the sync-byte configuration (0x47) and the packet length (188 Byte). If the input TS stream carries content that has only one PID, i.e., for only one service component (video, audio, etc.) or service sub-component (SVC base layer, SVC enhancement layer, MVC base view or MVC dependent views), TS packet header compression can be applied (optionally) to the Transport Stream. IP packet header compression is used optionally if the input steam is an IP stream.

[221] The above-described blocks may be omitted or replaced by blocks having similar or identical functions.

[222] FIG. 4 illustrates an input formatting block according to another embodiment of the present invention.

[223] The input formatting block illustrated in FIG. 4 corresponds to an embodiment of the input formatting block 1000 described with reference to FIG. 1.

[224] FIG. 4 illustrates a stream adaptation block of the input formatting module when the input signal corresponds to multiple input streams.

[225] Referring to FIG. 4, the mode adaptation block for respectively processing the multiple input streams can include a scheduler 4000, an 1-Frame delay block 4010, a stuffing insertion block 4020, an in-band signaling 4030, a BB Frame scrambler 4040, a PLS generation block 4050 and a PLS scrambler 4060. Description will be given of each block of the stream adaptation block.

[226] Operations of the stuffing insertion block 4020, the BB Frame scrambler 4040, the PLS generation block 4050 and the PLS scrambler 4060 correspond to those of the stuffing insertion block, BB scrambler, PLS generation block and the PLS scrambler described with reference to FIG. 2 and thus description thereof is omitted.

[227] The scheduler 4000 can determine the overall cell allocation across the entire frame from the amount of FECBLOCKs of each DP. Including the allocation for PLS, EAC and FIC, the scheduler generate the values of PLS2-DYN data, which is transmitted as in-band signaling or PLS cell in FSS of the frame. Details of FECBLOCK, EAC and FIC will be described later.

[228] The 1-Frame delay block 4010 can delay the input data by one transmission frame such that scheduling information about the next frame can be transmitted through the current frame for in-band signaling information to be inserted into the DPs.

[229] The in-band signaling 4030 can insert un-delayed part of the PLS2 data into a DP of a frame.

[230] The above-described blocks may be omitted or replaced by blocks having similar or identical functions.

[231] FIG. 5 illustrates a BICM block according to an embodiment of the present invention.

[232] The BICM block illustrated in FIG. 5 corresponds to an embodiment of the BICM block 1010 described with reference to FIG. 1.

[233] As described above, the apparatus for transmitting broadcast signals for future broadcast services according to an embodiment of the present invention can provide a terrestrial broadcast service, mobile broadcast service, UHDTV service, etc.

[234] Since QoS (quality of service) depends on characteristics of a service provided by the apparatus for transmitting broadcast signals for future broadcast services according to an embodiment of the present invention, data corresponding to respective services needs to be processed through different schemes. Accordingly, the a BICM block according to an embodiment of the present invention can independently process DPs input thereto by independently applying SISO, MISO and MIMO schemes to the data

pipes respectively corresponding to data paths. Consequently, the apparatus for transmitting broadcast signals for future broadcast services according to an embodiment of the present invention can control QoS for each service or service component transmitted through each DP.

[235] (a) shows the BICM block shared by the base profile and the handheld profile and (b) shows the BICM block of the advanced profile.

[236] The BICM block shared by the base profile and the handheld profile and the BICM block of the advanced profile can include plural processing blocks for processing each DP.

[237] A description will be given of each processing block of the BICM block for the base profile and the handheld profile and the BICM block for the advanced profile.

[238] A processing block 5000 of the BICM block for the base profile and the handheld profile can include a Data FEC encoder 5010, a bit interleaver 5020, a constellation mapper 5030, an SSD (Signal Space Diversity) encoding block 5040 and a time interleaver 5050.

[239] The Data FEC encoder 5010 can perform the FEC encoding on the input BBF to generate FECBLOCK procedure using outer coding (BCH), and inner coding (LDPC). The outer coding (BCH) is optional coding method. Details of operations of the Data FEC encoder 5010 will be described later.

[240] The bit interleaver 5020 can interleave outputs of the Data FEC encoder 5010 to achieve optimized performance with combination of the LDPC codes and modulation scheme while providing an efficiently implementable structure. Details of operations of the bit interleaver 5020 will be described later.

[241] The constellation mapper 5030 can modulate each cell word from the bit interleaver 5020 in the base and the handheld profiles, or cell word from the Cell-word demultiplexer 5010-1 in the advanced profile using either QPSK, QAM-16, non-uniform QAM (NUQ-64, NUQ-256, NUQ-1024) or non-uniform constellation (NUC-16, NUC-64, NUC-256, NUC-1024) to give a power-normalized constellation point, e_l. This constellation mapping is applied only for DPs. Observe that QAM-16 and NUQs are square shaped, while NUCs have arbitrary shape. When each constellation is rotated by any multiple of 90 degrees, the rotated constellation exactly overlaps with its original one. This "rotation-sense" symmetric property makes the capacities and the average powers of the real and imaginary components equal to each other. Both NUQs and NUCs are defined specifically for each code rate and the particular one used is signaled by the parameter DP_MOD filed in PLS2 data.

[242] The SSD encoding block 5040 can precode cells in two (2D), three (3D), and four (4D) dimensions to increase the reception robustness under difficult fading conditions.

[243] The time interleaver 5050 can operates at the DP level. The parameters of time in-

terleaving (TI) may be set differently for each DP. Details of operations of the time interleaver 5050 will be described later.

[244] A processing block 5000-1 of the BICM block for the advanced profile can include the Data FEC encoder, bit interleaver, constellation mapper, and time interleaver. However, the processing block 5000-1 is distinguished from the processing block 5000 further includes a cell-word demultiplexer 5010-1 and a MIMO encoding block 5020-1.

[245] Also, the operations of the Data FEC encoder, bit interleaver, constellation mapper, and time interleaver in the processing block 5000-1 correspond to those of the Data FEC encoder 5010, bit interleaver 5020, constellation mapper 5030, and time interleaver 5050 described and thus description thereof is omitted.

[246] The cell-word demultiplexer 5010-1 is used for the DP of the advanced profile to divide the single cell-word stream into dual cell-word streams for MIMO processing. Details of operations of the cell-word demultiplexer 5010-1 will be described later.

[247] The MIMO encoding block 5020-1 can processing the output of the cell-word demultiplexer 5010-1 using MIMO encoding scheme. The MIMO encoding scheme was optimized for broadcasting signal transmission. The MIMO technology is a promising way to get a capacity increase but it depends on channel characteristics. Especially for broadcasting, the strong LOS component of the channel or a difference in the received signal power between two antennas caused by different signal propagation characteristics makes it difficult to get capacity gain from MIMO. The proposed MIMO encoding scheme overcomes this problem using a rotation-based pre-coding and phase randomization of one of the MIMO output signals.

[248] MIMO encoding is intended for a 2x2 MIMO system requiring at least two antennas at both the transmitter and the receiver. Two MIMO encoding modes are defined in this proposal; full-rate spatial multiplexing (FR-SM) and full-rate full-diversity spatial multiplexing (FRFD-SM). The FR-SM encoding provides capacity increase with relatively small complexity increase at the receiver side while the FRFD-SM encoding provides capacity increase and additional diversity gain with a great complexity increase at the receiver side. The proposed MIMO encoding scheme has no restriction on the antenna polarity configuration.

[249] MIMO processing is required for the advanced profile frame, which means all DPs in the advanced profile frame are processed by the MIMO encoder. MIMO processing is applied at DP level. Pairs of the Constellation Mapper outputs NUQ ($e_{1,i}$ and $e_{2,i}$) are fed to the input of the MIMO Encoder. Paired MIMO Encoder output (g1,i and g2,i) is transmitted by the same carrier k and OFDM symbol l of their respective TX antennas.

[250] The above-described blocks may be omitted or replaced by blocks having similar or identical functions.

[251] FIG. 6 illustrates a BICM block according to another embodiment of the present invention.

[252] The BICM block illustrated in FIG. 6 corresponds to an embodiment of the BICM block 1010 described with reference to FIG. 1.

[253] FIG. 6 illustrates a BICM block for protection of physical layer signaling (PLS), emergency alert channel (EAC) and fast information channel (FIC). EAC is a part of a frame that carries EAS information data and FIC is a logical channel in a frame that carries the mapping information between a service and the corresponding base DP. Details of the EAC and FIC will be described later.

[254] Referring to FIG. 6, the BICM block for protection of PLS, EAC and FIC can include a PLS FEC encoder 6000, a bit interleaver 6010, a constellation mapper 6020 and time interleaver 6030.

[255] Also, the PLS FEC encoder 6000 can include a scrambler, BCH encoding/zero insertion block, LDPC encoding block and LDPC parity punturing block. Description will be given of each block of the BICM block.

[256] The PLS FEC encoder 6000 can encode the scrambled PLS 1/2 data, EAC and FIC section.

[257] The scrambler can scramble PLS1 data and PLS2 data before BCH encoding and shortened and punctured LDPC encoding.

[258] The BCH encoding/zero insertion block can perform outer encoding on the scrambled PLS 1/2 data using the shortened BCH code for PLS protection and insert zero bits after the BCH encoding. For PLS1 data only, the output bits of the zero insertion may be permutted before LDPC encoding.

[259] The LDPC encoding block can encode the output of the BCH encoding/zero insertion block using LDPC code. To generate a complete coded block, C_{ldpc}, parity bits, P_{ldpc} are encoded systematically from each zero-inserted PLS information block, I_{ldpc} and appended after it.

[260] MathFigure 1

[Math.1]

$$\mathbf{C}_{ldpc}=[\mathbf{I}_{ldpc}\ \mathbf{P}_{ldpc}]=[i_0,i_1,\ldots,i_{K_{ldpc}-1},\ p_0,p_1,\ldots,p_{N_{ldpc}-K_{ldpc}-1}]$$

[261] The LDPC code parameters for PLS1 and PLS2 are as following table 4.

[262] Table 4

[Table 4]

Signaling Type	K_{sig}	K_{bch}	N_{bch_parity}	K_{ldpc} (= N_{bch})	N_{ldpc}	N_{ldpc_parity}	code rate	Q_{ldpc}
PLS1	342	1020	60	1080	4320	3240	1/4	36
PLS2	<1021							
	>1020	2100		2160	7200	5040	3/10	56

[263] The LDPC parity punturing block can perform puncturing on the PLS1 data and PLS 2 data.

[264] When shortening is applied to the PLS1 data protection, some LDPC parity bits are punctured after LDPC encoding. Also, for the PLS2 data protection, the LDPC parity bits of PLS2 are punctured after LDPC encoding. These punctured bits are not transmitted.

[265] The bit interleaver 6010 can interleave the each shortened and punctured PLS1 data and PLS2 data.

[266] The constellation mapper 6020 can map the bit ineterlaeved PLS1 data and PLS2 data onto constellations.

[267] The time interleaver 6030 can interleave the mapped PLS1 data and PLS2 data.

[268] The above-described blocks may be omitted or replaced by blocks having similar or identical functions.

[269] FIG. 7 illustrates a frame building block according to one embodiment of the present invention.

[270] The frame building block illustrated in FIG. 7 corresponds to an embodiment of the frame building block 1020 described with reference to FIG. 1.

[271] Referring to FIG. 7, the frame building block can include a delay compensation block 7000, a cell mapper 7010 and a frequency interleaver 7020. Description will be given of each block of the frame building block.

[272] The delay compensation block 7000 can adjust the timing between the data pipes and the corresponding PLS data to ensure that they are co-timed at the transmitter end. The PLS data is delayed by the same amount as data pipes are by addressing the delays of data pipes caused by the Input Formatting block and BICM block. The delay of the BICM block is mainly due to the time interleaver. In-band signaling data carries information of the next TI group so that they are carried one frame ahead of the DPs to be signaled. The Delay Compensating block delays in-band signaling data accordingly.

[273] The cell mapper 7010 can map PLS, EAC, FIC, DPs, auxiliary streams and dummy cells into the active carriers of the OFDM symbols in the frame. The basic function of the cell mapper 7010 is to map data cells produced by the TIs for each of the DPs, PLS cells, and EAC/FIC cells, if any, into arrays of active OFDM cells corresponding to each of the OFDM symbols within a frame. Service signaling data (such as PSI(program specific information)/SI) can be separately gathered and sent by a data pipe. The Cell Mapper operates according to the dynamic information produced by the scheduler and the configuration of the frame structure. Details of the frame will be described later.

[274] The frequency interleaver 7020 can randomly interleave data cells received from the cell mapper 7010 to provide frequency diversity. Also, the frequency interleaver 7020

can operate on very OFDM symbol pair comprised of two sequential OFDM symbols using a different interleaving-seed order to get maximum interleaving gain in a single frame.

[275] The above-described blocks may be omitted or replaced by blocks having similar or identical functions.

[276] FIG. 8 illustrates an OFMD generation block according to an embodiment of the present invention.

[277] The OFMD generation block illustrated in FIG. 8 corresponds to an embodiment of the OFMD generation block 1030 described with reference to FIG. 1.

[278] The OFDM generation block modulates the OFDM carriers by the cells produced by the Frame Building block, inserts the pilots, and produces the time domain signal for transmission. Also, this block subsequently inserts guard intervals, and applies PAPR (Peak-to-Average Power Radio) reduction processing to produce the final RF signal.

[279] Referring to FIG. 8, the frame building block can include a pilot and reserved tone insertion block 8000, a 2D-eSFN encoding block 8010, an IFFT (Inverse Fast Fourier Transform) block 8020, a PAPR reduction block 8030, a guard interval insertion block 8040, a preamble insertion block 8050, other system insertion block 8060 and a DAC block 8070. Description will be given of each block of the frame building block.

[280] The pilot and reserved tone insertion block 8000 can insert pilots and the reserved tone.

[281] Various cells within the OFDM symbol are modulated with reference information, known as pilots, which have transmitted values known a priori in the receiver. The information of pilot cells is made up of scattered pilots, continual pilots, edge pilots, FSS (frame signaling symbol) pilots and FES (frame edge symbol) pilots. Each pilot is transmitted at a particular boosted power level according to pilot type and pilot pattern. The value of the pilot information is derived from a reference sequence, which is a series of values, one for each transmitted carrier on any given symbol. The pilots can be used for frame synchronization, frequency synchronization, time synchronization, channel estimation, and transmission mode identification, and also can be used to follow the phase noise.

[282] Reference information, taken from the reference sequence, is transmitted in scattered pilot cells in every symbol except the preamble, FSS and FES of the frame. Continual pilots are inserted in every symbol of the frame. The number and location of continual pilots depends on both the FFT size and the scattered pilot pattern. The edge carriers are edge pilots in every symbol except for the preamble symbol. They are inserted in order to allow frequency interpolation up to the edge of the spectrum. FSS pilots are inserted in FSS(s) and FES pilots are inserted in FES. They are inserted in order to allow time interpolation up to the edge of the frame.

[283] The system according to an embodiment of the present invention supports the SFN network, where distributed MISO scheme is optionally used to support very robust transmission mode. The 2D-eSFN is a distributed MISO scheme that uses multiple TX antennas, each of which is located in the different transmitter site in the SFN network.

[284] The 2D-eSFN encoding block 8010 can process a 2D-eSFN processing to distorts the phase of the signals transmitted from multiple transmitters, in order to create both time and frequency diversity in the SFN configuration. Hence, burst errors due to low flat fading or deep-fading for a long time can be mitigated.

[285] The IFFT block 8020 can modulate the output from the 2D-eSFN encoding block 8010 using OFDM modulation scheme. Any cell in the data symbols which has not been designated as a pilot (or as a reserved tone) carries one of the data cells from the frequency interleaver. The cells are mapped to OFDM carriers.

[286] The PAPR reduction block 8030 can perform a PAPR reduction on input signal using various PAPR reduction algorithm in the time domain.

[287] The guard interval insertion block 8040 can insert guard intervals and the preamble insertion block 8050 can insert preamble in front of the signal. Details of a structure of the preamble will be described later. The other system insertion block 8060 can multiplex signals of a plurality of broadcast transmission/reception systems in the time domain such that data of two or more different broadcast transmission/reception systems providing broadcast services can be simultaneously transmitted in the same RF signal bandwidth. In this case, the two or more different broadcast transmission/ reception systems refer to systems providing different broadcast services. The different broadcast services may refer to a terrestrial broadcast service, mobile broadcast service, etc. Data related to respective broadcast services can be transmitted through different frames.

[288] The DAC block 8070 can convert an input digital signal into an analog signal and output the analog signal. The signal output from the DAC block 7800 can be transmitted through multiple output antennas according to the physical layer profiles. A Tx antenna according to an embodiment of the present invention can have vertical or horizontal polarity.

[289] The above-described blocks may be omitted or replaced by blocks having similar or identical functions according to design.

[290] FIG. 9 illustrates a structure of an apparatus for receiving broadcast signals for future broadcast services according to an embodiment of the present invention.

[291] The apparatus for receiving broadcast signals for future broadcast services according to an embodiment of the present invention can correspond to the apparatus for transmitting broadcast signals for future broadcast services, described with reference to FIG. 1.

[292] The apparatus for receiving broadcast signals for future broadcast services according to an embodiment of the present invention can include a synchronization & demodulation module 9000, a frame parsing module 9010, a demapping & decoding module 9020, an output processor 9030 and a signaling decoding module 9040. A description will be given of operation of each module of the apparatus for receiving broadcast signals.

[293] The synchronization & demodulation module 9000 can receive input signals through m Rx antennas, perform signal detection and synchronization with respect to a system corresponding to the apparatus for receiving broadcast signals and carry out demodulation corresponding to a reverse procedure of the procedure performed by the apparatus for transmitting broadcast signals.

[294] The frame parsing module 9100 can parse input signal frames and extract data through which a service selected by a user is transmitted. If the apparatus for transmitting broadcast signals performs interleaving, the frame parsing module 9100 can carry out deinterleaving corresponding to a reverse procedure of interleaving. In this case, the positions of a signal and data that need to be extracted can be obtained by decoding data output from the signaling decoding module 9400 to restore scheduling information generated by the apparatus for transmitting broadcast signals.

[295] The demapping & decoding module 9200 can convert the input signals into bit domain data and then deinterleave the same as necessary. The demapping & decoding module 9200 can perform demapping for mapping applied for transmission efficiency and correct an error generated on a transmission channel through decoding. In this case, the demapping & decoding module 9200 can obtain transmission parameters necessary for demapping and decoding by decoding the data output from the signaling decoding module 9400.

[296] The output processor 9300 can perform reverse procedures of various compression/ signal processing procedures which are applied by the apparatus for transmitting broadcast signals to improve transmission efficiency. In this case, the output processor 9300 can acquire necessary control information from data output from the signaling decoding module 9400. The output of the output processor 8300 corresponds to a signal input to the apparatus for transmitting broadcast signals and may be MPEG-TSs, IP streams (v4 or v6) and generic streams.

[297] The signaling decoding module 9400 can obtain PLS information from the signal demodulated by the synchronization & demodulation module 9000. As described above, the frame parsing module 9100, demapping & decoding module 9200 and output processor 9300 can execute functions thereof using the data output from the signaling decoding module 9400.

[298] FIG. 10 illustrates a frame structure according to an embodiment of the present

invention.

[299] FIG. 10 shows an example configuration of the frame types and FRUs in a super-frame. (a) shows a super frame according to an embodiment of the present invention, (b) shows FRU (Frame Repetition Unit) according to an embodiment of the present invention, (c) shows frames of variable PHY profiles in the FRU and (d) shows a structure of a frame.

[300] A super-frame may be composed of eight FRUs. The FRU is a basic multiplexing unit for TDM of the frames, and is repeated eight times in a super-frame.

[301] Each frame in the FRU belongs to one of the PHY profiles, (base, handheld, advanced) or FEF. The maximum allowed number of the frames in the FRU is four and a given PHY profile can appear any number of times from zero times to four times in the FRU (e.g., base, base, handheld, advanced). PHY profile definitions can be extended using reserved values of the PHY_PROFILE in the preamble, if required.

[302] The FEF part is inserted at the end of the FRU, if included. When the FEF is included in the FRU, the minimum number of FEFs is 8 in a super-frame. It is not recommended that FEF parts be adjacent to each other.

[303] One frame is further divided into a number of OFDM symbols and a preamble. As shown in (d), the frame comprises a preamble, one or more frame signaling symbols (FSS), normal data symbols and a frame edge symbol (FES).

[304] The preamble is a special symbol that enables fast Futurecast UTB system signal detection and provides a set of basic transmission parameters for efficient transmission and reception of the signal. The detailed description of the preamble will be will be described later.

[305] The main purpose of the FSS(s) is to carry the PLS data. For fast synchronization and channel estimation, and hence fast decoding of PLS data, the FSS has more dense pilot pattern than the normal data symbol. The FES has exactly the same pilots as the FSS, which enables frequency-only interpolation within the FES and temporal interpolation, without extrapolation, for symbols immediately preceding the FES.

[306] FIG. 11 illustrates a signaling hierarchy structure of the frame according to an embodiment of the present invention.

[307] FIG. 11 illustrates the signaling hierarchy structure, which is split into three main parts: the preamble signaling data 11000, the PLS1 data 11010 and the PLS2 data 11020. The purpose of the preamble, which is carried by the preamble symbol in every frame, is to indicate the transmission type and basic transmission parameters of that frame. The PLS1 enables the receiver to access and decode the PLS2 data, which contains the parameters to access the DP of interest. The PLS2 is carried in every frame and split into two main parts: PLS2-STAT data and PLS2-DYN data. The static and dynamic portion of PLS2 data is followed by padding, if necessary.

[308] FIG. 12 illustrates preamble signaling data according to an embodiment of the present invention.

[309] Preamble signaling data carries 21 bits of information that are needed to enable the receiver to access PLS data and trace DPs within the frame structure. Details of the preamble signaling data are as follows:

[310] PHY_PROFILE: This 3-bit field indicates the PHY profile type of the current frame. The mapping of different PHY profile types is given in below table 5.

[311] Table 5

[Table 5]

Value	PHY profile
000	Base profile
001	Handheld profile
010	Advanced profiled
011~110	Reserved
111	FEF

[312] FFT_SIZE: This 2 bit field indicates the FFT size of the current frame within a frame-group, as described in below table 6.

[313] Table 6

[Table 6]

Value	FFT size
00	8K FFT
01	16K FFT
10	32K FFT
11	Reserved

[314] GI_FRACTION: This 3 bit field indicates the guard interval fraction value in the current super-frame, as described in below table 7.

[315] Table 7

[Table 7]

Value	GI_FRACTION
000	1/5
001	1/10
010	1/20
011	1/40
100	1/80
101	1/160
110~111	Reserved

[316] EAC_FLAG: This 1 bit field indicates whether the EAC is provided in the current frame. If this field is set to '1', emergency alert service (EAS) is provided in the current frame. If this field set to '0', EAS is not carried in the current frame. This field can be switched dynamically within a super-frame.

[317] PILOT_MODE: This 1-bit field indicates whether the pilot mode is mobile mode or fixed mode for the current frame in the current frame-group. If this field is set to '0', mobile pilot mode is used. If the field is set to '1', the fixed pilot mode is used.

[318] PAPR_FLAG: This 1-bit field indicates whether PAPR reduction is used for the current frame in the current frame-group. If this field is set to value '1', tone reservation is used for PAPR reduction. If this field is set to '0', PAPR reduction is not used.

[319] FRU_CONFIGURE: This 3-bit field indicates the PHY profile type configurations of the frame repetition units (FRU) that are present in the current super-frame. All profile types conveyed in the current super-frame are identified in this field in all preambles in the current super-frame. The 3-bit field has a different definition for each profile, as show in below table 8.

[320] Table 8

[Table 8]

	Current PHY_PROFILE = '000' (base)	Current PHY_PROFILE = '001' (handheld)	Current PHY_PROFILE = '010' (advanced)	Current PHY_PROFILE = '111' (FEF)
FRU_CONFIGURE = 000	Only base profile present	Only handheld profile present	Only advanced profile present	Only FEF present
FRU_CONFIGURE = 1XX	Handheld profile present	Base profile present	Base profile present	Base profile present
FRU_CONFIGURE = X1X	Advanced profile present	Advanced profile present	Handheld profile present	Handheld profile present
FRU_CONFIGURE = XX1	FEF present	FEF present	FEF present	Advanced profile present

[321] RESERVED: This 7-bit field is reserved for future use.

[322] FIG. 13 illustrates PLS1 data according to an embodiment of the present invention.

[323] PLS1 data provides basic transmission parameters including parameters required to enable the reception and decoding of the PLS2. As above mentioned, the PLS1 data remain unchanged for the entire duration of one frame-group. The detailed definition of the signaling fields of the PLS1 data are as follows:

[324] PREAMBLE_DATA: This 20-bit field is a copy of the preamble signaling data

excluding the EAC_FLAG.

[325] NUM_FRAME_FRU: This 2-bit field indicates the number of the frames per FRU.

[326] PAYLOAD_TYPE: This 3-bit field indicates the format of the payload data carried in the frame-group. PAYLOAD_TYPE is signaled as shown in table 9.

[327] Table 9

[Table 9]

value	Payload type
1XX	TS stream is transmitted
X1X	IP stream is transmitted
XX1	GS stream is transmitted

[328] NUM_FSS: This 2-bit field indicates the number of FSS symbols in the current frame.

[329] SYSTEM_VERSION: This 8-bit field indicates the version of the transmitted signal format. The SYSTEM_VERSION is divided into two 4-bit fields, which are a major version and a minor version.

[330] Major version: The MSB four bits of SYSTEM_VERSION field indicate major version information. A change in the major version field indicates a non-backward-compatible change. The default value is '0000'. For the version described in this standard, the value is set to '0000'.

[331] Minor version: The LSB four bits of SYSTEM_VERSION field indicate minor version information. A change in the minor version field is backward-compatible.

[332] CELL_ID: This is a 16-bit field which uniquely identifies a geographic cell in an ATSC network. An ATSC cell coverage area may consist of one or more frequencies, depending on the number of frequencies used per Futurecast UTB system. If the value of the CELL_ID is not known or unspecified, this field is set to '0'.

[333] NETWORK_ID: This is a 16-bit field which uniquely identifies the current ATSC network.

[334] SYSTEM_ID: This 16-bit field uniquely identifies the Futurecast UTB system within the ATSC network. The Futurecast UTB system is the terrestrial broadcast system whose input is one or more input streams (TS, IP, GS) and whose output is an RF signal. The Futurecast UTB system carries one or more PHY profiles and FEF, if any. The same Futurecast UTB system may carry different input streams and use different RF frequencies in different geographical areas, allowing local service insertion. The frame structure and scheduling is controlled in one place and is identical for all transmissions within a Futurecast UTB system. One or more Futurecast UTB systems may have the same SYSTEM_ID meaning that they all have the same physical layer structure and configuration.

[335] The following loop consists of FRU_PHY_PROFILE, FRU_FRAME_LENGTH, FRU_GI_FRACTION, and RESERVED which are used to indicate the FRU configuration and the length of each frame type. The loop size is fixed so that four PHY profiles (including a FEF) are signaled within the FRU. If NUM_FRAME_FRU is less than 4, the unused fields are filled with zeros.

[336] FRU_PHY_PROFILE: This 3-bit field indicates the PHY profile type of the $(i+1)^{th}$ (i is the loop index) frame of the associated FRU. This field uses the same signaling format as shown in the table 8.

[337] FRU_FRAME_LENGTH: This 2-bit field indicates the length of the $(i+1)^{th}$ frame of the associated FRU. Using FRU_FRAME_LENGTH together with FRU_GI_FRACTION, the exact value of the frame duration can be obtained.

[338] FRU_GI_FRACTION: This 3-bit field indicates the guard interval fraction value of the $(i+1)^{th}$ frame of the associated FRU. FRU_GI_FRACTION is signaled according to the table 7.

[339] RESERVED: This 4-bit field is reserved for future use.

[340] The following fields provide parameters for decoding the PLS2 data.

[341] PLS2_FEC_TYPE: This 2-bit field indicates the FEC type used by the PLS2 protection. The FEC type is signaled according to table 10. The details of the LDPC codes will be described later.

[342] Table 10

[Table 10]

Content	PLS2 FEC type
00	4K-1/4 and 7K-3/10 LDPC codes
01 ~ 11	Reserved

[343] PLS2_MOD: This 3-bit field indicates the modulation type used by the PLS2. The modulation type is signaled according to table 11.

[344] Table 11

[Table 11]

Value	PLS2_MODE
000	BPSK
001	QPSK
010	QAM-16
011	NUQ-64
100~111	Reserved

[345] PLS2_SIZE_CELL: This 15-bit field indicates $C_{total\_partial\_block}$, the size (specified as the number of QAM cells) of the collection of *full* coded blocks for PLS2 that is

carried in the current frame-group. This value is constant during the entire duration of the current frame-group.

[346] PLS2_STAT_SIZE_BIT: This 14-bit field indicates the size, in bits, of the PLS2-STAT for the current frame-group. This value is constant during the entire duration of the current frame-group.

[347] PLS2_DYN_SIZE_BIT: This 14-bit field indicates the size, in bits, of the PLS2-DYN for the current frame-group. This value is constant during the entire duration of the current frame-group.

[348] PLS2_REP_FLAG: This 1-bit flag indicates whether the PLS2 repetition mode is used in the current frame-group. When this field is set to value '1', the PLS2 repetition mode is activated. When this field is set to value '0', the PLS2 repetition mode is de-activated.

[349] PLS2_REP_SIZE_CELL: This 15-bit field indicates $C_{total\_partial\_block}$, the size (specified as the number of QAM cells) of the collection of *partial* coded blocks for PLS2 carried in every frame of the current frame-group, when PLS2 repetition is used. If repetition is not used, the value of this field is equal to 0. This value is constant during the entire duration of the current frame-group.

[350] PLS2_NEXT_FEC_TYPE: This 2-bit field indicates the FEC type used for PLS2 that is carried in every frame of the next frame-group. The FEC type is signaled according to the table 10.

[351] PLS2_NEXT_MOD: This 3-bit field indicates the modulation type used for PLS2 that is carried in every frame of the next frame-group. The modulation type is signaled according to the table 11.

[352] PLS2_NEXT_REP_FLAG: This 1-bit flag indicates whether the PLS2 repetition mode is used in the next frame-group. When this field is set to value '1', the PLS2 repetition mode is activated. When this field is set to value '0', the PLS2 repetition mode is deactivated.

[353] PLS2_NEXT_REP_SIZE_CELL: This 15-bit field indicates $C_{total\_full\_block}$, The size (specified as the number of QAM cells) of the collection of full coded blocks for PLS2 that is carried in every frame of the next frame-group, when PLS2 repetition is used. If repetition is not used in the next frame-group, the value of this field is equal to 0. This value is constant during the entire duration of the current frame-group.

[354] PLS2_NEXT_REP_STAT_SIZE_BIT: This 14-bit field indicates the size, in bits, of the PLS2-STAT for the next frame-group. This value is constant in the current frame-group.

[355] PLS2_NEXT_REP_DYN_SIZE_BIT: This 14-bit field indicates the size, in bits, of the PLS2-DYN for the next frame-group. This value is constant in the current frame-group.

[356] PLS2_AP_MODE: This 2-bit field indicates whether additional parity is provided for PLS2 in the current frame-group. This value is constant during the entire duration of the current frame-group. The below table 12 gives the values of this field. When this field is set to '00', additional parity is not used for the PLS2 in the current frame-group.

[357] Table 12

[Table 12]

Value	PLS2-AP mode
00	AP is not provided
01	AP1 mode
10~11	Reserved

[358] PLS2_AP_SIZE_CELL: This 15-bit field indicates the size (specified as the number of QAM cells) of the additional parity bits of the PLS2. This value is constant during the entire duration of the current frame-group.

[359] PLS2_NEXT_AP_MODE: This 2-bit field indicates whether additional parity is provided for PLS2 signaling in every frame of next frame-group. This value is constant during the entire duration of the current frame-group. The table 12 defines the values of this field

[360] PLS2_NEXT_AP_SIZE_CELL: This 15-bit field indicates the size (specified as the number of QAM cells) of the additional parity bits of the PLS2 in every frame of the next frame-group. This value is constant during the entire duration of the current frame-group.

[361] RESERVED: This 32-bit field is reserved for future use.

[362] CRC_32: A 32-bit error detection code, which is applied to the entire PLS1 signaling.

[363] FIG. 14 illustrates PLS2 data according to an embodiment of the present invention.

[364] FIG. 14 illustrates PLS2-STAT data of the PLS2 data. The PLS2-STAT data are the same within a frame-group, while the PLS2-DYN data provide information that is specific for the current frame.

[365] The details of fields of the PLS2-STAT data are as follows:

[366] FIC_FLAG: This 1-bit field indicates whether the FIC is used in the current frame-group. If this field is set to '1', the FIC is provided in the current frame. If this field set to '0', the FIC is not carried in the current frame. This value is constant during the entire duration of the current frame-group.

[367] AUX_FLAG: This 1-bit field indicates whether the auxiliary stream(s) is used in the current frame-group. If this field is set to '1', the auxiliary stream is provided in the current frame. If this field set to '0', the auxiliary stream is not carried in the current frame. This value is constant during the entire duration of current frame-group.

[368] NUM_DP: This 6-bit field indicates the number of DPs carried within the current

frame. The value of this field ranges from 1 to 64, and the number of DPs is NUM_DP+1.

[369] DP_ID: This 6-bit field identifies uniquely a DP within a PHY profile.

[370] DP_TYPE: This 3-bit field indicates the type of the DP. This is signaled according to the below table 13.

[371] Table 13

[Table 13]

Value	DP Type
000	DP Type 1
001	DP Type 2
010~111	reserved

[372] DP_GROUP_ID: This 8-bit field identifies the DP group with which the current DP is associated. This can be used by a receiver to access the DPs of the service components associated with a particular service, which will have the same DP_GROUP_ID.

[373] BASE_DP_ID: This 6-bit field indicates the DP carrying service signaling data (such as PSI/SI) used in the Management layer. The DP indicated by BASE_DP_ID may be either a normal DP carrying the service signaling data along with the service data or a dedicated DP carrying only the service signaling data

[374] DP_FEC_TYPE: This 2-bit field indicates the FEC type used by the associated DP. The FEC type is signaled according to the below table 14.

[375] Table 14

[Table 14]

Value	FEC_TYPE
00	16K LDPC
01	64K LDPC
10 ~ 11	Reserved

[376] DP_COD: This 4-bit field indicates the code rate used by the associated DP. The code rate is signaled according to the below table 15.

[377] Table 15

[Table 15]

Value	Code rate
0000	5/15
0001	6/15
0010	7/15
0011	8/15
0100	9/15
0101	10/15
0110	11/15
0111	12/15
1000	13/15
1001 ~ 1111	Reserved

[378] DP_MOD: This 4-bit field indicates the modulation used by the associated DP. The modulation is signaled according to the below table 16.

[379] Table 16

[Table 16]

Value	Modulation
0000	QPSK
0001	QAM-16
0010	NUQ-64
0011	NUQ-256
0100	NUQ-1024
0101	NUC-16
0110	NUC-64
0111	NUC-256
1000	NUC-1024
1001~1111	reserved

[380] DP_SSD_FLAG: This 1-bit field indicates whether the SSD mode is used in the associated DP. If this field is set to value '1', SSD is used. If this field is set to value '0', SSD is not used.

[381] The following field appears only if PHY_PROFILE is equal to '010', which indicates the advanced profile:

[382] DP_MIMO: This 3-bit field indicates which type of MIMO encoding process is applied to the associated DP. The type of MIMO encoding process is signaled according to the table 17.

[383] Table 17

[Table 17]

Value	MIMO encoding
000	FR-SM
001	FRFD-SM
010~111	reserved

[384] DP_TI_TYPE: This 1-bit field indicates the type of time-interleaving. A value of '0' indicates that one TI group corresponds to one frame and contains one or more TI-blocks. A value of '1' indicates that one TI group is carried in more than one frame and contains only one TI-block.

[385] DP_TI_LENGTH: The use of this 2-bit field (the allowed values are only 1, 2, 4, 8) is determined by the values set within the DP_TI_TYPE field as follows:

[386] If the DP_TI_TYPE is set to the value '1', this field indicates P_I, the number of the frames to which each TI group is mapped, and there is one TI-block per TI group (N_{TI} =1). The allowed P_I values with 2-bit field are defined in the below table 18.

[387] If the DP_TI_TYPE is set to the value '0', this field indicates the number of TI-blocks N_{TI} per TI group, and there is one TI group per frame (P_I=1). The allowed P_I values with 2-bit field are defined in the below table 18.

[388] Table 18

[Table 18]

2-bit field	P_I	N_{TI}
00	1	1
01	2	2
10	4	3
11	8	4

[389] DP_FRAME_INTERVAL: This 2-bit field indicates the frame interval (I_{JUMP}) within the frame-group for the associated DP and the allowed values are 1, 2, 4, 8 (the corresponding 2-bit field is '00', '01', '10', or '11', respectively). For DPs that do not appear every frame of the frame-group, the value of this field is equal to the interval between successive frames. For example, if a DP appears on the frames 1, 5, 9, 13, etc., this field is set to '4'. For DPs that appear in every frame, this field is set to '1'.

[390] DP_TI_BYPASS: This 1-bit field determines the availability of time interleaver. If time interleaving is not used for a DP, it is set to '1'. Whereas if time interleaving is used it is set to '0'.

[391] DP_FIRST_FRAME_IDX: This 5-bit field indicates the index of the first frame of the super-frame in which the current DP occurs. The value of DP_FIRST_FRAME_IDX ranges from 0 to 31

[392] DP_NUM_BLOCK_MAX: This 10-bit field indicates the maximum value of

DP_NUM_BLOCKS for this DP. The value of this field has the same range as DP_NUM_BLOCKS.

[393] DP_PAYLOAD_TYPE: This 2-bit field indicates the type of the payload data carried by the given DP. DP_PAYLOAD_TYPE is signaled according to the below table 19.

[394] Table 19

[Table 19]

Value	Payload Type
00	TS.
01	IP
10	GS
11	reserved

[395] DP_INBAND_MODE: This 2-bit field indicates whether the current DP carries in-band signaling information. The in-band signaling type is signaled according to the below table 20.

[396] Table 20

[Table 20]

Value	In-band mode
00	In-band signaling is not carried.
01	INBAND-PLS is carried only
10	INBAND-ISSY is carried only
11	INBAND-PLS and INBAND-ISSY are carried

[397] DP_PROTOCOL_TYPE: This 2-bit field indicates the protocol type of the payload carried by the given DP. It is signaled according to the below table 21 when input payload types are selected.

[398] Table 21

[Table 21]

Value	If DP_PAYLOAD_TYPE Is TS	If DP_PAYLOAD_TYPE Is IP	If DP_PAYLOAD_TYPE Is GS
00	MPEG2-TS	IPv4	(Note)
01	Reserved	IPv6	Reserved
10	Reserved	Reserved	Reserved
11	Reserved	Reserved	Reserved

[399] DP_CRC_MODE: This 2-bit field indicates whether CRC encoding is used in the Input Formatting block. The CRC mode is signaled according to the below table 22.

[400] Table 22

[Table 22]

Value	CRC mode
00	Not used
01	CRC-8
10	CRC-16
11	CRC-32

[401] DNP_MODE: This 2-bit field indicates the null-packet deletion mode used by the associated DP when DP_PAYLOAD_TYPE is set to TS ('00'). DNP_MODE is signaled according to the below table 23. If DP_PAYLOAD_TYPE is not TS ('00'), DNP_MODE is set to the value '00'.

[402] Table 23

[Table 23]

Value	Null-packet deletion mode
00	Not used
01	DNP-NORMAL
10	DNP-OFFSET
11	reserved

[403] ISSY_MODE: This 2-bit field indicates the ISSY mode used by the associated DP when DP_PAYLOAD_TYPE is set to TS ('00'). The ISSY_MODE is signaled according to the below table 24 If DP_PAYLOAD_TYPE is not TS ('00'), ISSY_MODE is set to the value '00'.

[404] Table 24

[Table 24]

Value	ISSY mode
00	Not used
01	ISSY-UP
10	ISSY-BBF
11	reserved

[405] HC_MODE_TS: This 2-bit field indicates the TS header compression mode used by the associated DP when DP_PAYLOAD_TYPE is set to TS ('00'). The HC_MODE_TS is signaled according to the below table 25.

[406] Table 25

[Table 25]

Value	Header compression mode
00	HC_MODE_TS 1
01	HC_MODE_TS 2
10	HC_MODE_TS 3
11	HC_MODE_TS 4

[407] HC_MODE_IP: This 2-bit field indicates the IP header compression mode when DP_PAYLOAD_TYPE is set to IP ('01'). The HC_MODE_IP is signaled according to the below table 26.

[408] Table 26

[Table 26]

Value	Header compression mode
00	No compression
01	HC_MODE_IP 1
10~11	reserved

[409] PID : This 13-bit field indicates the PID number for TS header compression when DP_PAYLOAD_TYPE is set to TS ('00') and HC_MODE_TS is set to '01' or '10'.

[410] RESERVED: This 8-bit field is reserved for future use.

[411] The following field appears only if FIC_FLAG is equal to '1':

[412] FIC_VERSION: This 8-bit field indicates the version number of the FIC.

[413] FIC_LENGTH_BYTE: This 13-bit field indicates the length, in bytes, of the FIC.

[414] RESERVED: This 8-bit field is reserved for future use.

[415] The following field appears only if AUX_FLAG is equal to '1':

[416] NUM_AUX: This 4-bit field indicates the number of auxiliary streams. Zero means no auxiliary streams are used.

[417] AUX_CONFIG_RFU: This 8-bit field is reserved for future use.

[418] AUX_STREAM_TYPE: This 4-bit is reserved for future use for indicating the type of the current auxiliary stream.

[419] AUX_PRIVATE_CONFIG: This 28-bit field is reserved for future use for signaling auxiliary streams.

[420] FIG. 15 illustrates PLS2 data according to another embodiment of the present invention.

[421] FIG. 15 illustrates PLS2-DYN data of the PLS2 data. The values of the PLS2-DYN data may change during the duration of one frame-group, while the size of fields remains constant.

[422] The details of fields of the PLS2-DYN data are as follows:

[423] FRAME_INDEX: This 5-bit field indicates the frame index of the current frame within the super-frame. The index of the first frame of the super-frame is set to '0'.

[424] PLS_CHANGE_COUNTER: This 4-bit field indicates the number of super-frames ahead where the configuration will change. The next super-frame with changes in the configuration is indicated by the value signaled within this field. If this field is set to the value '0000', it means that no scheduled change is foreseen: e.g., value '1' indicates that there is a change in the next super-frame.

[425] FIC_CHANGE_COUNTER: This 4-bit field indicates the number of super-frames ahead where the configuration (i.e., the contents of the FIC) will change. The next super-frame with changes in the configuration is indicated by the value signaled within this field. If this field is set to the value '0000', it means that no scheduled change is foreseen: e.g. value '0001' indicates that there is a change in the next super-frame..

[426] RESERVED: This 16-bit field is reserved for future use.

[427] The following fields appear in the loop over NUM_DP, which describe the parameters associated with the DP carried in the current frame.

[428] DP_ID: This 6-bit field indicates uniquely the DP within a PHY profile.

[429] DP_START: This 15-bit (or 13-bit) field indicates the start position of the first of the DPs using the DPU addressing scheme. The DP_START field has differing length according to the PHY profile and FFT size as shown in the below table 27.

[430] Table 27

[Table 27]

PHY profile	DP_START field size	
	64K	16K
Base	13 bit	15 bit
Handheld	-	13 bit
Advanced	13 bit	15 bit

[431] DP_NUM_BLOCK: This 10-bit field indicates the number of FEC blocks in the current TI group for the current DP. The value of DP_NUM_BLOCK ranges from 0 to 1023

[432] RESERVED: This 8-bit field is reserved for future use.

[433] The following fields indicate the FIC parameters associated with the EAC.

[434] EAC_FLAG: This 1-bit field indicates the existence of the EAC in the current frame. This bit is the same value as the EAC_FLAG in the preamble.

[435] EAS_WAKE_UP_VERSION_NUM: This 8-bit field indicates the version number of a wake-up indication.

[436] If the EAC_FLAG field is equal to '1', the following 12 bits are allocated for EAC_LENGTH_BYTE field. If the EAC_FLAG field is equal to '0', the following 12

bits are allocated for EAC_COUNTER.

[437] EAC_LENGTH_BYTE: This 12-bit field indicates the length, in byte, of the EAC. .

[438] EAC_COUNTER: This 12-bit field indicates the number of the frames before the frame where the EAC arrives.

[439] The following field appears only if the AUX_FLAG field is equal to '1':

[440] AUX_PRIVATE_DYN: This 48-bit field is reserved for future use for signaling auxiliary streams. The meaning of this field depends on the value of AUX_STREAM_TYPE in the configurable PLS2-STAT.

[441] CRC_32: A 32-bit error detection code, which is applied to the entire PLS2.

[442] FIG. 16 illustrates a logical structure of a frame according to an embodiment of the present invention.

[443] As above mentioned, the PLS, EAC, FIC, DPs, auxiliary streams and dummy cells are mapped into the active carriers of the OFDM symbols in the frame. The PLS1 and PLS2 are first mapped into one or more FSS(s). After that, EAC cells, if any, are mapped immediately following the PLS field, followed next by FIC cells, if any. The DPs are mapped next after the PLS or EAC, FIC, if any. Type 1 DPs follows first, and Type 2 DPs next. The details of a type of the DP will be described later. In some case, DPs may carry some special data for EAS or service signaling data. The auxiliary stream or streams, if any, follow the DPs, which in turn are followed by dummy cells. Mapping them all together in the above mentioned order, i.e. PLS, EAC, FIC, DPs, auxiliary streams and dummy data cells exactly fill the cell capacity in the frame.

[444] FIG. 17 illustrates PLS mapping according to an embodiment of the present invention.

[445] PLS cells are mapped to the active carriers of FSS(s). Depending on the number of cells occupied by PLS, one or more symbols are designated as FSS(s), and the number of FSS(s) N_{FSS} is signaled by NUM_FSS in PLS1. The FSS is a special symbol for carrying PLS cells. Since robustness and latency are critical issues in the PLS, the FSS(s) has higher density of pilots allowing fast synchronization and frequency-only interpolation within the FSS.

[446] PLS cells are mapped to active carriers of the N_{FSS} FSS(s) in a top-down manner as shown in an example in FIG. 17. The PLS1 cells are mapped first from the first cell of the first FSS in an increasing order of the cell index. The PLS2 cells follow immediately after the last cell of the PLS1 and mapping continues downward until the last cell index of the first FSS. If the total number of required PLS cells exceeds the number of active carriers of one FSS, mapping proceeds to the next FSS and continues in exactly the same manner as the first FSS.

[447] After PLS mapping is completed, DPs are carried next. If EAC, FIC or both are present in the current frame, they are placed between PLS and "normal" DPs.

[448] FIG. 18 illustrates EAC mapping according to an embodiment of the present invention.

[449] EAC is a dedicated channel for carrying EAS messages and links to the DPs for EAS. EAS support is provided but EAC itself may or may not be present in every frame. EAC, if any, is mapped immediately after the PLS2 cells. EAC is not preceded by any of the FIC, DPs, auxiliary streams or dummy cells other than the PLS cells. The procedure of mapping the EAC cells is exactly the same as that of the PLS.

[450] The EAC cells are mapped from the next cell of the PLS2 in increasing order of the cell index as shown in the example in FIG. 18. Depending on the EAS message size, EAC cells may occupy a few symbols, as shown in FIG. 18.

[451] EAC cells follow immediately after the last cell of the PLS2, and mapping continues downward until the last cell index of the last FSS. If the total number of required EAC cells exceeds the number of remaining active carriers of the last FSS mapping proceeds to the next symbol and continues in exactly the same manner as FSS(s). The next symbol for mapping in this case is the normal data symbol, which has more active carriers than a FSS.

[452] After EAC mapping is completed, the FIC is carried next, if any exists. If FIC is not transmitted (as signaled in the PLS2 field), DPs follow immediately after the last cell of the EAC.

[453] FIG. 19 illustrates FIC mapping according to an embodiment of the present invention.

[454] (a) shows an example mapping of FIC cell without EAC and (b) shows an example mapping of FIC cell with EAC.

[455] FIC is a dedicated channel for carrying cross-layer information to enable fast service acquisition and channel scanning. This information primarily includes channel binding information between DPs and the services of each broadcaster. For fast scan, a receiver can decode FIC and obtain information such as broadcaster ID, number of services, and BASE_DP_ID. For fast service acquisition, in addition to FIC, base DP can be decoded using BASE_DP_ID. Other than the content it carries, a base DP is encoded and mapped to a frame in exactly the same way as a normal DP. Therefore, no additional description is required for a base DP. The FIC data is generated and consumed in the Management Layer. The content of FIC data is as described in the Management Layer specification.

[456] The FIC data is optional and the use of FIC is signaled by the FIC_FLAG parameter in the static part of the PLS2. If FIC is used, FIC_FLAG is set to '1' and the signaling field for FIC is defined in the static part of PLS2. Signaled in this field are FIC_VERSION, and FIC_LENGTH_BYTE. FIC uses the same modulation, coding and time interleaving parameters as PLS2. FIC shares the same signaling parameters

such as PLS2_MOD and PLS2_FEC. FIC data, if any, is mapped immediately after PLS2 or EAC if any. FIC is not preceded by any normal DPs, auxiliary streams or dummy cells. The method of mapping FIC cells is exactly the same as that of EAC which is again the same as PLS.

[457] Without EAC after PLS, FIC cells are mapped from the next cell of the PLS2 in an increasing order of the cell index as shown in an example in (a). Depending on the FIC data size, FIC cells may be mapped over a few symbols, as shown in (b).

[458] FIC cells follow immediately after the last cell of the PLS2, and mapping continues downward until the last cell index of the last FSS. If the total number of required FIC cells exceeds the number of remaining active carriers of the last FSS, mapping proceeds to the next symbol and continues in exactly the same manner as FSS(s). The next symbol for mapping in this case is the normal data symbol which has more active carriers than a FSS.

[459] If EAS messages are transmitted in the current frame, EAC precedes FIC, and FIC cells are mapped from the next cell of the EAC in an increasing order of the cell index as shown in (b).

[460] After FIC mapping is completed, one or more DPs are mapped, followed by auxiliary streams, if any, and dummy cells.

[461] FIG. 20 illustrates a type of DP according to an embodiment of the present invention.

[462] (a) shows type 1 DP and (b) shows type 2 DP.

[463] After the preceding channels, i.e., PLS, EAC and FIC, are mapped, cells of the DPs are mapped. A DP is categorized into one of two types according to mapping method:

[464] Type 1 DP: DP is mapped by TDM

[465] Type 2 DP: DP is mapped by FDM

[466] The type of DP is indicated by DP_TYPE field in the static part of PLS2. FIG. 20 illustrates the mapping orders of Type 1 DPs and Type 2 DPs. Type 1 DPs are first mapped in the increasing order of cell index, and then after reaching the last cell index, the symbol index is increased by one. Within the next symbol, the DP continues to be mapped in the increasing order of cell index starting from $p = 0$. With a number of DPs mapped together in one frame, each of the Type 1 DPs are grouped in time, similar to TDM multiplexing of DPs.

[467] Type 2 DPs are first mapped in the increasing order of symbol index, and then after reaching the last OFDM symbol of the frame, the cell index increases by one and the symbol index rolls back to the first available symbol and then increases from that symbol index. After mapping a number of DPs together in one frame, each of the Type 2 DPs are grouped in frequency together, similar to FDM multiplexing of DPs.

[468] Type 1 DPs and Type 2 DPs can coexist in a frame if needed with one restriction; Type 1 DPs always precede Type 2 DPs. The total number of OFDM cells carrying

Type 1 and Type 2 DPs cannot exceed the total number of OFDM cells available for transmission of DPs:

[469] MathFigure 2

[Math.2]

$$D_{DP1} + D_{DP2} \leq D_{DP}$$

[470] where *DDP1* is the number of OFDM cells occupied by Type 1 DPs, *DDP2* is the number of cells occupied by Type 2 DPs. Since PLS, EAC, FIC are all mapped in the same way as Type 1 DP, they all follow "Type 1 mapping rule". Hence, overall, Type 1 mapping always precedes Type 2 mapping.

[471] FIG. 21 illustrates DP mapping according to an embodiment of the present invention.

[472] (a) shows an addressing of OFDM cells for mapping type 1 DPs and (b) shows an an addressing of OFDM cells for mapping for type 2 DPs.

[473] Addressing of OFDM cells for mapping Type 1 DPs (0, ..., *DDP1*1) is defined for the active data cells of Type 1 DPs. The addressing scheme defines the order in which the cells from the TIs for each of the Type 1 DPs are allocated to the active data cells. It is also used to signal the locations of the DPs in the dynamic part of the PLS2.

[474] Without EAC and FIC, address 0 refers to the cell immediately following the last cell carrying PLS in the last FSS. If EAC is transmitted and FIC is not in the corresponding frame, address 0 refers to the cell immediately following the last cell carrying EAC. If FIC is transmitted in the corresponding frame, address 0 refers to the cell immediately following the last cell carrying FIC. Address 0 for Type 1 DPs can be calculated considering two different cases as shown in (a). In the example in (a), PLS, EAC and FIC are assumed to be all transmitted. Extension to the cases where either or both of EAC and FIC are omitted is straightforward. If there are remaining cells in the FSS after mapping all the cells up to FIC as shown on the left side of (a).

[475] Addressing of OFDM cells for mapping Type 2 DPs (0, ..., *DDP2*1) is defined for the active data cells of Type 2 DPs. The addressing scheme defines the order in which the cells from the TIs for each of the Type 2 DPs are allocated to the active data cells. It is also used to signal the locations of the DPs in the dynamic part of the PLS2.

[476] Three slightly different cases are possible as shown in (b). For the first case shown on the left side of (b), cells in the last FSS are available for Type 2 DP mapping. For the second case shown in the middle, FIC occupies cells of a normal symbol, but the number of FIC cells on that symbol is not larger than C_{FSS}. The third case, shown on the right side in (b), is the same as the second case except that the number of FIC cells mapped on that symbol exceeds C_{FSS}.

[477] The extension to the case where Type 1 DP(s) precede Type 2 DP(s) is straightforward since PLS, EAC and FIC follow the same "Type 1 mapping rule" as

the Type 1 DP(s).

[478] A data pipe unit (DPU) is a basic unit for allocating data cells to a DP in a frame.

[479] A DPU is defined as a signaling unit for locating DPs in a frame. A Cell Mapper 7010 may map the cells produced by the TIs for each of the DPs. A Time interleaver 5050 outputs a series of TI-blocks and each TI-block comprises a variable number of XFECBLOCKs which is in turn composed of a set of cells. The number of cells in an XFECBLOCK, N_{cells}, is dependent on the FECBLOCK size, N_{ldpc}, and the number of transmitted bits per constellation symbol. A DPU is defined as the greatest common divisor of all possible values of the number of cells in a XFECBLOCK, N_{cells}, supported in a given PHY profile. The length of a DPU in cells is defined as L_{DPU}. Since each PHY profile supports different combinations of FECBLOCK size and a different number of bits per constellation symbol, L_{DPU} is defined on a PHY profile basis.

[480] FIG. 22 illustrates an FEC structure according to an embodiment of the present invention.

[481] FIG. 22 illustrates an FEC structure according to an embodiment of the present invention before bit interleaving. As above mentioned, Data FEC encoder may perform the FEC encoding on the input BBF to generate FECBLOCK procedure using outer coding (BCH), and inner coding (LDPC). The illustrated FEC structure corresponds to the FECBLOCK. Also, the FECBLOCK and the FEC structure have same value corresponding to a length of LDPC codeword.

[482] The BCH encoding is applied to each BBF (K_{bch} bits), and then LDPC encoding is applied to BCH-encoded BBF (K_{ldpc} bits = N_{bch} bits) as illustrated in FIG. 22.

[483] The value of N_{ldpc} is either 64800 bits (long FECBLOCK) or 16200 bits (short FECBLOCK).

[484] The below table 28 and table 29 show FEC encoding parameters for a long FECBLOCK and a short FECBLOCK, respectively.

[485] Table 28

[Table 28]

LDPC Rate	N_{ldpc}	K_{ldpc}	K_{bch}	BCH error correction capability	N_{bch}-K_{bch}
5/15	64800	21600	21408	12	192
6/15		25920	25728		
7/15		30240	30048		
8/15		34560	34368		
9/15		38880	38688		
10/15		43200	43008		
11/15		47520	47328		
12/15		51840	51648		
13/15		56160	55968		

[486] Table 29

[Table 29]

LDPC Rate	N_{ldpc}	K_{ldpc}	K_{bch}	BCH error correction capability	N_{bch}-K_{bch}
5/15	16200	5400	5232	12	168
6/15		6480	6312		
7/15		7560	7392		
8/15		8640	8472		
9/15		9720	9552		
10/15		10800	10632		
11/15		11880	11712		
12/15		12960	12792		
13/15		14040	13872		

[487] The details of operations of the BCH encoding and LDPC encoding are as follows:

[488] A 12-error correcting BCH code is used for outer encoding of the BBF. The BCH generator polynomial for short FECBLOCK and long FECBLOCK are obtained by multiplying together all polynomials.

[489] LDPC code is used to encode the output of the outer BCH encoding. To generate a completed B_{ldpc} (FECBLOCK), P_{ldpc} (parity bits) is encoded systematically from each I_{ldpc} (BCH-encoded BBF), and appended to I_{ldpc}. The completed B_{ldpc} (FECBLOCK) are expressed as follow Math figure.

[490] MathFigure 3

[Math.3]

$$B_{ldpc} = [\mathbf{I}_{ldpc} \ \mathbf{P}_{ldpc}] = [i_0, i_1, \ldots, i_{K_{ldpc}-1}, \ p_0, p_1, \ldots, p_{N_{ldpc}-K_{ldpc}-1}]$$

[491] The parameters for long FECBLOCK and short FECBLOCK are given in the above table 28 and 29, respectively.

[492] The detailed procedure to calculate N_{ldpc} - K_{ldpc} parity bits for long FECBLOCK, is as follows:

[493] 1) Initialize the parity bits,

[494] MathFigure 4

[Math.4]

$$p_0 = p_1 = p_2 = \ldots = p_{N_{ldpc}-K_{idpc}-1} = 0$$

[495] 2) Accumulate the first information bit - i_0, at parity bit addresses specified in the first row of an addresses of parity check matrix. The details of addresses of parity check matrix will be described later. For example, for rate 13/15:

[496] MathFigure 5

[Math.5]

$$p_{983} = p_{983} \oplus i_0 \qquad p_{2815} = p_{2815} \oplus i_0$$

$$p_{4837} = p_{4837} \oplus i_0 \qquad p_{4989} = p_{4989} \oplus i_0$$

$$p_{6138} = p_{6138} \oplus i_0 \qquad p_{6458} = p_{6458} \oplus i_0$$

$$p_{6921} = p_{6921} \oplus i_0 \qquad p_{6974} = p_{6974} \oplus i_0$$

$$p_{7572} = p_{7572} \oplus i_0 \qquad p_{8260} = p_{8260} \oplus i_0$$

$$p_{8496} = p_{8496} \oplus i_0$$

[497] 3) For the next 359 information bits, i_s, s=1, 2, …, 359 accumulate i_s at parity bit addresses using following Math figure.

[498] MathFigure 6

[Math.6]

$$\{x + (s \mod 360) \times Q_{ldpc}\} \bmod (N_{ldpc} - K_{ldpc})$$

[499] where x denotes the address of the parity bit accumulator corresponding to the first bit i_0, and Q_{ldpc} is a code rate dependent constant specified in the addresses of parity check matrix. Continuing with the example, Q_{ldpc} = 24 for rate 13/15, so for information bit i_1, the following operations are performed:

[500] MathFigure 7

[Math.7]

$$p_{1007} = p_{1007} \oplus i_1 \qquad p_{2839} = p_{2839} \oplus i_1$$

$$p_{4861} = p_{4861} \oplus i_1 \qquad p_{5013} = p_{5013} \oplus i_1$$

$$p_{6162} = p_{6162} \oplus i_1 \qquad p_{6482} = p_{6482} \oplus i_1$$

$$p_{6945} = p_{6945} \oplus i_1 \qquad p_{6998} = p_{6998} \oplus i_1$$

$$p_{7596} = p_{7596} \oplus i_1 \qquad p_{8284} = p_{8284} \oplus i_1$$

$$p_{8520} = p_{8520} \oplus i_1$$

[501] 4) For the 361st information bit i_{360}, the addresses of the parity bit accumulators are given in the second row of the addresses of parity check matrix. In a similar manner the addresses of the parity bit accumulators for the following 359 information bits i_s, s= 361, 362, …, 719 are obtained using the Math figure 6, where x denotes the address of the parity bit accumulator corresponding to the information bit i_{360}, i.e., the entries in the second row of the addresses of parity check matrix.

[502] 5) In a similar manner, for every group of 360 new information bits, a new row from addresses of parity check matrixes used to find the addresses of the parity bit accumulators.

[503] After all of the information bits are exhausted, the final parity bits are obtained as follows:

[504] 6) Sequentially perform the following operations starting with i=1

[505] MathFigure 8

[Math.8]

$$p_i = p_i \oplus p_{i-1}, \qquad i = 1,2,\ldots, N_{ldpc} - K_{ldpc} - 1$$

[506] where final content of p_i, i=0,1,...N_{ldpc} - K_{ldpc} - 1 is equal to the parity bit p_i.

[507] Table 30

[Table 30]

Code Rate	Q_{ldpc}
5/15	120
6/15	108
7/15	96
8/15	84
9/15	72
10/15	60
11/15	48
12/15	36
13/15	24

[508] This LDPC encoding procedure for a short FECBLOCK is in accordance with t LDPC encoding procedure for the long FECBLOCK, except replacing the table 30 with table 31, and replacing the addresses of parity check matrix for the long FECBLOCK with the addresses of parity check matrix for the short FECBLOCK.

[509] Table 31

[Table 31]

Code Rate	Q_{ldpc}
5/15	30
6/15	27
7/15	24
8/15	21
9/15	18
10/15	15
11/15	12
12/15	9
13/15	6

[510] FIG. 23 illustrates a bit interleaving according to an embodiment of the present invention.

[511] The outputs of the LDPC encoder are bit-interleaved, which consists of parity interleaving followed by Quasi-Cyclic Block (QCB) interleaving and inner-group interleaving.

[512] (a) shows Quasi-Cyclic Block (QCB) interleaving and (b) shows inner-group interleaving.

[513] The FECBLOCK may be parity interleaved. At the output of the parity interleaving,

the LDPC codeword consists of 180 adjacent QC blocks in a long FECBLOCK and 45 adjacent QC blocks in a short FECBLOCK. Each QC block in either a long or short FECBLOCK consists of 360 bits. The parity interleaved LDPC codeword is interleaved by QCB interleaving. The unit of QCB interleaving is a QC block. The QC blocks at the output of parity interleaving are permutated by QCB interleaving as illustrated in FIG. 23, where N_{cells} =64800/η_{mod} or 16200/η_{mod} according to the FECBLOCK length. The QCB interleaving pattern is unique to each combination of modulation type and LDPC code rate.

[514] After QCB interleaving, inner-group interleaving is performed according to modulation type and order (η_{mod}) which is defined in the below table 32. The number of QC blocks for one inner-group, N_{QCB_IG}, is also defined.

[515] Table 32

[Table 32]

Modulation type	η_{mod}	N_{QCB_IG}
QAM-16	4	2
NUC-16	4	4
NUQ-64	6	3
NUC-64	6	6
NUQ-256	8	4
NUC-256	8	8
NUQ-1024	10	5
NUC-1024	10	10

[516] The inner-group interleaving process is performed with N_{QCB_IG} QC blocks of the QCB interleaving output. Inner-group interleaving has a process of writing and reading the bits of the inner-group using 360 columns and N_{QCB_IG} rows. In the write operation, the bits from the QCB interleaving output are written row-wise. The read operation is performed column-wise to read out m bits from each row, where m is equal to 1 for NUC and 2 for NUQ.

[517] FIG. 24 illustrates a cell-word demultiplexing according to an embodiment of the present invention.

[518] (a) shows a cell-word demultiplexing for 8 and 12 bpcu MIMO and (b) shows a cell-word demultiplexing for 10 bpcu MIMO.

[519] Each cell word ($c_{0,l}$, $c_{1,l}$, …, $c_{nmod-1,l}$) of the bit interleaving output is demultiplexed into ($d_{1,0,m}$, $d_{1,1,m}$…, $d_{1,nmod-1,m}$) and ($d_{2,0,m}$, $d_{2,1,m}$…, $d_{2,nmod-1,m}$) as shown in (a), which describes the cell-word demultiplexing process for one XFECBLOCK.

[520] For the 10 bpcu MIMO case using different types of NUQ for MIMO encoding, the

Bit Interleaver for NUQ-1024 is re-used. Each cell word ($c_{0,l}$, $c_{1,l}$, ..., $c_{9,l}$) of the Bit Interleaver output is demultiplexed into ($d_{1,0,m}$, $d_{1,1,m}$..., $d_{1,3,m}$) and ($d_{2,0,m}$, $d_{2,1,m}$..., $d_{2,5,m}$), as shown in (b).

[521] FIG. 25 illustrates a time interleaving according to an embodiment of the present invention.

[522] (a) to (c) show examples of TI mode.

[523] The time interleaver operates at the DP level. The parameters of time interleaving (TI) may be set differently for each DP.

[524] The following parameters, which appear in part of the PLS2-STAT data, configure the TI:

[525] DP_TI_TYPE (allowed values: 0 or 1): Represents the TI mode; '0' indicates the mode with multiple TI blocks (more than one TI block) per TI group. In this case, one TI group is directly mapped to one frame (no inter-frame interleaving). '1' indicates the mode with only one TI block per TI group. In this case, the TI block may be spread over more than one frame (inter-frame interleaving).

[526] DP_TI_LENGTH: If DP_TI_TYPE = '0', this parameter is the number of TI blocks N_{TI} per TI group. For DP_TI_TYPE = '1', this parameter is the number of frames P_I spread from one TI group.

[527] DP_NUM_BLOCK_MAX (allowed values: 0 to 1023): Represents the maximum number of XFECBLOCKs per TI group.

[528] DP_FRAME_INTERVAL (allowed values: 1, 2, 4, 8): Represents the number of the frames I_{JUMP} between two successive frames carrying the same DP of a given PHY profile.

[529] DP_TI_BYPASS (allowed values: 0 or 1): If time interleaving is not used for a DP, this parameter is set to '1'. It is set to '0' if time interleaving is used.

[530] Additionally, the parameter DP_NUM_BLOCK from the PLS2-DYN data is used to represent the number of XFECBLOCKs carried by one TI group of the DP.

[531] When time interleaving is not used for a DP, the following TI group, time interleaving operation, and TI mode are not considered. However, the Delay Compensation block for the dynamic configuration information from the scheduler will still be required. In each DP, the XFECBLOCKs received from the SSD/MIMO encoding are grouped into TI groups. That is, each TI group is a set of an integer number of XFECBLOCKs and will contain a dynamically variable number of XFECBLOCKs. The number of XFECBLOCKs in the TI group of index n is denoted by $N_{xBLOCK\_Group}(n)$ and is signaled as DP_NUM_BLOCK in the PLS2-DYN data. Note that $N_{xBLOCK\_Group}(n)$ may vary from the minimum value of 0 to the maximum value $N_{xBLOCK\_Group\_MAX}$ (corresponding to DP_NUM_BLOCK_MAX) of which the largest value is 1023.

[532] Each TI group is either mapped directly onto one frame or spread over P_I frames.

Each TI group is also divided into more than one TI blocks(N_{TI}), where each TI block corresponds to one usage of time interleaver memory. The TI blocks within the TI group may contain slightly different numbers of XFECBLOCKs. If the TI group is divided into multiple TI blocks, it is directly mapped to only one frame. There are three options for time interleaving (except the extra option of skipping the time interleaving) as shown in the below table 33.

[533] Table 33

[Table 33]

Modes	Descriptions
Option-1	Each TI group contains one TI block and is mapped directly to one frame as shown in (a). This option is signaled in the PLS2-STAT by DP_TI_TYPE='0' and DP_TI_LENGTH ='1'(N_{TI}=1).
Option-2	Each TI group contains one TI block and is mapped to more than one frame. (b) shows an example, where one TI group is mapped to two frames, i.e., DP_TI_LENGTH ='2' (P_I=2) and DP_FRAME_INTERVAL (I_{JUMP} = 2). This provides greater time diversity for low data-rate services. This option is signaled in the PLS2-STAT by DP_TI_TYPE ='1'.
Option-3	Each TI group is divided into multiple TI blocks and is mapped directly to one frame as shown in (c). Each TI block may use full TI memory, so as to provide the maximum bit-rate for a DP. This option is signaled in the PLS2-STAT signaling by DP_TI_TYPE='0' and DP_TI_LENGTH = N_{TI}, while P_I=1.

[534] In each DP, the TI memory stores the input XFECBLOCKs (output XFECBLOCKs from the SSD/MIMO encoding block). Assume that input XFECBLOCKs are defined as

[535] $$(d_{n,s,0,0},d_{n,s,0,1},\ldots,d_{n,s,0,N_{cells}-1},d_{n,s,1,0},\ldots,d_{n,s,1,N_{cells}-1},\ldots,d_{n,s,N_{xBLOCK\_TI}(n,s)-1,0},\ldots,d_{n,s,N_{xBLOCK\_TI}(n,s)-1,N_{cells}-1}),$$

[536] where $d_{n,s,r,q}$ is the qth cell of the rth XFECBLOCK in the sth TI block of the nth TI group and represents the outputs of SSD and MIMO encodings as follows.

[537] $$d_{n,s,r,q}=\begin{cases} f_{n,s,r,q} & \text{, the output of SSD encoding} \\ g_{n,s,r,q} & \text{, the output of MIMO encoding} \end{cases}$$

[538] In addition, assume that output XFECBLOCKs from the time interleaver are defined as

[539] $$(h_{n,s,0},h_{n,s,1},\ldots,h_{n,s,i},\ldots,h_{n,s,N_{xBLOCK\_TI}(n,s)\times N_{cells}-1}),$$

[540] where $h_{n,s,i}$ is the ith output cell (for $i=0,\ldots,N_{xBLOCK\_TI}(n,s)\times N_{cells}-1$) in the sth TI block of the nth TI group.

[541] Typically, the time interleaver will also act as a buffer for DP data prior to the process of frame building. This is achieved by means of two memory banks for each

DP. The first TI-block is written to the first bank. The second TI-block is written to the second bank while the first bank is being read from and so on.

[542] The TI is a twisted row-column block interleaver. For the sth TI block of the nth TI group, the number of rows N_r of a TI memory is equal to the number of cells N_{cells}, i.e., $N_r = N_{cells}$ while the number of columns N_c is equal to the number $N_{xBLOCK\_TI}(n,s)$.

[543] FIG. 26 illustrates the basic operation of a twisted row-column block interleaver according to an embodiment of the present invention.

[544] shows a writing operation in the time interleaver and (b) shows a reading operation in the time interleaver The first XFECBLOCK is written column-wise into the first column of the TI memory, and the second XFECBLOCK is written into the next column, and so on as shown in (a). Then, in the interleaving array, cells are read out diagonal-wise. During diagonal-wise reading from the first row (rightwards along the row beginning with the left-most column) to the last row, N_r cells are read out as shown in (b). In detail, assuming $z_{n,s,i}(i = 0,\ldots,N_r N_c)$ as the TI memory cell position to be read sequentially, the reading process in such an interleaving array is performed by calculating the row index $R_{n,s,i}$, the column index $C_{n,s,i}$, and the associated twisting parameter $T_{n,s,i}$ as follows expression.

[545] MathFigure 9

[Math.9]

$$GENERATE(R_{n,s,i}, C_{n,s,i}) =$$
$$\{$$
$$R_{n,s,i} = \mathrm{mod}(i, N_r),$$
$$T_{n,s,i} = \mathrm{mod}(S_{shift} \times R_{n,s,i}, N_c),$$
$$C_{n,s,i} = \mathrm{mod}(T_{n,s,i} + \left\lfloor \frac{i}{N_r} \right\rfloor, N_c)$$
$$\}$$

[546] where

S_{shift}

is a common shift value for the diagonal-wise reading process regardless of

$N_{xBLOCK\_TI}(n,s)$

, and it is determined by

$N_{xBLOCK\_TI\_MAX}$

given in the PLS2-STAT as follows expression.

[547] MathFigure 10

[Math.10]

$$for \begin{cases} N'_{xBLOCK\_TI\_MAX} = N_{xBLOCK\_TI\_MAX} + 1, & if\ N_{xBLOCK\_TI\_MAX} \bmod 2 = 0 \\ N'_{xBLOCK\_TI\_MAX} = N_{xBLOCK\_TI\_MAX}, & if\ N_{xBLOCK\_TI\_MAX} \bmod 2 = 1 \end{cases},$$

$$S_{shift} = \frac{N'_{xBLOCK\_TI\_MAX} - 1}{2}$$

[548] As a result, the cell positions to be read are calculated by a coordinate as $z_{n,s,i} = N_r C_{n,s,i} + R_{n,s,i}$.

[549] FIG. 27 illustrates an operation of a twisted row-column block interleaver according to another embodiment of the present invention.

[550] More specifically, FIG. 27 illustrates the interleaving array in the TI memory for each TI group, including virtual XFECBLOCKs when $N_{xBLOCK\_TI}(0,0) = 3$, $N_{xBLOCK\_TI}(1,0) = 6$, $N_{xBLOCK\_TI}(2,0) = 5$.

[551] The variable number $N_{xBLOCK\_TI}(n,s) = N_r$ will be less than or equal to $N'_{xBLOCK\_TI\_MAX}$. Thus, in order to achieve a single-memory deinterleaving at the receiver side, regardless of $N_{xBLOCK\_TI}(n,s)$, the interleaving array for use in a twisted row-column block interleaver is set to the size of $N_r \times N_c = N_{cells} \times N'_{xBLOCK\_TI\_MAX}$ by inserting the virtual XFECBLOCKs into the TI memory and the reading process is accomplished as follow expression.

[552] MathFigure 11

[Math.11]

$$p = 0;$$
$$for\ i = 0; i < N_{cells} N'_{xBLOCK\_TI\_MAX}; i = i + 1$$
$$\{GENERATE(R_{n,s,i}, C_{n,s,i});$$
$$V_i = N_r C_{n,s,j} + R_{n,s,j}$$
$$if\ V_i < N_{cells} N_{xBLOCK\_TI}(n,s)$$
$$\{$$
$$Z_{n,s,p} = V_i;\ p = p + 1;$$
$$\}$$
$$\}$$

[553] The number of TI groups is set to 3. The option of time interleaver is signaled in the PLS2-STAT data by DP_TI_TYPE='0', DP_FRAME_INTERVAL='1', and DP_TI_LENGTH='1', i.e.,N_{TI}=1, I_{JUMP}=1, and P_I=1. The number of XFECBLOCKs, each of which has N_{cells} = 30 cells, per TI group is signaled in the PLS2-DYN data by $N_{xBLOCK\_TI}(0,0)$=3, $N_{xBLOCK\_TI}(1,0)$=6, and $N_{xBLOCK\_TI}(2,0)$=5, respectively. The maximum number of XFECBLOCK is signaled in the PLS2-STAT data by $N_{xBLOCK\_Group\_MAX}$, which leads to

$$\lfloor N_{xBLOCK\_Group\_MAX} / N_{TI} \rfloor = N_{xBLOCK\_TI\_MAX} = 6$$

.

[554] FIG. 28 illustrates a diagonal-wise reading pattern of a twisted row-column block interleaver according to an embodiment of the present invention.

[555] More specifically FIG. 28 shows a diagonal-wise reading pattern from each interleaving array with parameters of

$$N'_{xBLOCK\_TI\_MAX} = 7$$

and S_{shift}=(7-1)/2=3. Note that in the reading process shown as pseudocode above, if

$$V_i \geq N_{cells} N_{xBLOCK\_TI}(n,s)$$

, the value of V_i is skipped and the next calculated value of V_i is used.

[556] FIG. 29 illustrates interlaved XFECBLOCKs from each interleaving array according to an embodiment of the present invention.

[557] FIG. 29 illustrates the interleaved XFECBLOCKs from each interleaving array with parameters of

$$N'_{xBLOCK\_TI\_MAX} = 7$$

and S_{shift}=3.

[558] Hereinafter, a frequency interleaving procedure according to an embodiment of the present invention will be described.

[559] The purpose of the frequency interleaver 7020 in the present invention, which operates on a single OFDM symbol, is to provide frequency diversity by randomly interleaving data cells received from the cell mapper 7010. In order to get maximum in-

terleaving gain in a single signal frame (or frame), a different interleaving-seed is used for every OFDM symbol pair comprised of two sequential OFDM symbols.

[560] The frequency interleaver 7020 may interleave cells in a transport block as a unit of a signal frame to acquire additional diversity gain. According to an embodiment of the present invention, the frequency interleaver 7020 may apply different interleaving seeds to at least one OFDM sysmbol or apply different interleaving seeds to a frame including a plurality of OFDM symbols.

[561] In the present invention, the aforementioned frequency interleaving method may be referred to as random frequency interleaving (random FI).

[562] In addition, according to an embodiment of the present invention, the random FI may be applied to a super-frame structure including a plurality of signal frames with a plurality of OFDM symbols.

[563] As described above, a broadcast signal transmitting apparatus or a frequency interleaver 7020 therein according to an embodiment of the present invention may apply different interleaving seeds (or interleaving patterns) for at least one OFDM symbol, that is, for each OFDM symbol or each of pair-wise OFDM symbols (or each OFDM symbol pair) and perform the random FI, thereby acquiring frequency diversity. In addition, the frequency interleaver 7020 according to an embodiment of the present invention may apply different interleaving seed for each respective signal frame and perform the random FI, thereby acquiring additional frequency diversity.

[564] Accordingly, a broadcast transmitting apparatus or a frequency interleaver 7020 according to an embodiment of the present invention may have a ping-pong frequency interleaver 7020 structure that perform frequency interleaving in units of one pair of consecutive OFDM symbols (pair-wise OFDM symbol) using two memory banks. Hereinafter, an interleaving operation of the frequency interleaver 7020 according to an embodiment of the present invention may be referred to as pair-wise symbol FI (or pair-wise FI) or ping-pong FI (ping-pong interleaving). The aforementioned interleaving operation corresponds to an embodiment of the random FI, which can be changed according to a designer's intention.

[565] Even-indexed pair-wise OFDM symbols and odd pair-wise OFDM symbols may be intermittently interleaved via different FI memory banks. In addition, the frequency interleaver 7020 according to an embodiment of the present invention may simultaneously perform reading and writing operations on one pair of consecutive OFDM symbols input to each memory bank using an arbitrary interleaving seed. A detailed operation will be described below.

[566] In addition, according to an embodiment of the present invention, as a logical frequency interleaving operation for logically and effectively interleaving all OFDM symbols in a super-frame, an interleaving seed is basically changed in units of one pair

of OFDM symbols.

[567] In this case, according to an embodiment of the present invention, the interleaving seed may be generated by an arbitrary random generator or a random generator formed by a combination of various random generators. In addition, according to an embodiment of the present invention, various interleaving seeds may be generated by cyclic-shifting one main interleaving seed in order to effectively change an interleaving seed. That is different interleaving seed to be used every OFDM symbol pair can be generated by cyclic-shifting one interleaving seed (main interleaving seed). In this case, a cyclic-shifting rule may be hierarchically defined in consideration of OFDM symbol and signal frame units. Therefore, the symbol offset according to the present invention may be referred as a cyclic shifting value. This can be changed according to a designer's intention, which will be described in detail.

[568] A broadcast signal receiving apparatus according to an embodiment of the present invention may perform an inverse procedure of the aforementioned random frequency interleaving. In this case, the broadcast signal receiving apparatus or a frequency deinterleaver thereof according to an embodiment of the present invention may not use a ping-pong structure using a double-memory and may perform deinterleaving on consecutive input OFDM symbols via a single-memory. Accordingly, memory use efficiency can be enhanced. In addition, reading and writing operations are still required, which is called as a single-memory deinterleaving operation. Such a deinterleaving scheme is very efficient in a memory-use aspect.

[569] FIG. 30 is a view illustrating an operation of a frequency interleaver 7020 according to an embodiment of the present invention.

[570] FIG. 30 illustrates the basic operation of the frequency interleaver 7020 using two memory banks at the transmitter, which enables a single-memory deinterleaving at the receiver.

[571] As described above, the frequency interleaver 7020 according to an embodiment of the present invention may perform a ping-pong interleaving operation.

[572] Typically, ping-pong interleaving operation is accomplished by means of two memory banks. In the proposed FI operation, two memory banks are for each pair-wise OFDM symbol.

[573] The maximum memory ROM (Read Only Memory) size for interleaving is approximately two times to a maximum FFT size. At a transmit side, the ROM size increase is rather less critical, compared to a receiver side.

[574] As described above, odd pair-wise OFDM symbols and odd pair-wise OFDM symbols may be intermittently interleaved via different FI memory-banks. That is, the second (odd-indexed) pair-wise OFDM symbol is interleaved in the second bank, while the first (even-indexed) pair-wise OFDM symbol is interleaved in the first bank

and so on. For each pair-wise OFDM symbol, a single interleaving seed is used. Based on the interleaving seed and reading-writing (or writing-reading) operation, two OFDM symbols are sequentially interleaved.

[575] Reading-writing operations according to an embodiment of the present invention are simultaneously accomplished without a collision. Writing-reading operations according to an embodiment of the present invention are simultaneously accomplished without a collision.

[576] FIG. 30 illustrates an operation of the aforementioned frequency interleaver 7020. As illustrated in FIG. 30, the frequency interleaver 7020 may include a demux 16000, two memory banks, a memory bank-A 16100 and a memory bank-B 16200, and a demux 16300.

[577] First, the frequency interleaver 7020 according to an embodiment of the present invention may perform a demultiplexing processing to the input sequential OFDM symbols for the pair-wise OFDM symbol FI. Then the frequency interleaver 7020 according to an embodiment of the present invention performs a reading-writing FI operation in each memory bank A and B with a single interleaving seed. As shown in FIG. 30, two memory banks are used for each OFDM symbol pair. Operationally, the first (even-indexed) OFDM symbol pair is interleaved in memory bank-A, while the second (odd-indexed) OFDM symbol pair is interleaved in memory bank-B and so on, alternating between A and B.

[578] Then the frequency interleaver 7020 according to an embodiment of the present invention may perform a multiplexing processing to ping-pong FI outputs for sequential OFDM symbol transmission.

[579] FIG. 31 illustrates a basic switch model for MUX and DEMUX procedures according to an embodiment of the present invention.

[580] FIG. 31 illustrates simple operations the DEMUX and MUX applied input and output of memory-bank-A/-B in the aforementioned ping-pong FI structure.

[581] The DEMUX and MUX may control the input sequential OFDM symbols to be interleaved, and the output OFDM symbol pair to be transmitted, respectively. Different interleaving seeds are used for every OFDM symbol pair.

[582] Hereinafter, reading-writing operations of frequency interleaving according to an embodiment of the present invention will be described.

[583] A frequency interleaver 7020 according to an embodiment of the present invention may select or use a single interleaving see and use the interleaving seed in writing and reading operations for the first and second OFDM symbols, respectively. That is, the frequency interleaver 7020 according to an embodiment of the present invention may use the one selected arbitrary interleaving seed in an operation of writing a first OFDM symbol of a pair-wise OFDM symbol, and use a second OFDM symbol in a reading

operation, thereby achieving effective interleaving. Virtually, it seems like that two different interleaving seeds are applied to two OFDM symbols, respectively.

[584] Details of the reading-writing operation according to an embodiment of the present invention are as follows:

[585] For the first OFDM symbol, the frequency interleaver 7020 according to an embodiment of the present invention may perform random writing into memory (according to an interleaving seed) and perform then linear reading. For the second OFDM symbol, the frequency interleaver 7020 according to an embodiment of the present invention may perform linear writing into memory, (affected by the linear reading operation for the first OFDM symbol), simultaneously. Also, the frequency interleaver 7020 according to an embodiment of the present invention may perform then random reading (according to an interleaving seed).

[586] As described above, the broadcast signal receiving apparatus according to an embodiment of the present invention may continuously transmit a plurality of frames on the time axis. In the present invention, a set of signal frames transmitted for a predetermined period of time may be referred to as a super-frame. Accordingly, one super-frame may include N signal frames and each signal frame may include a plurality of OFDM symbols.

[587] FIG. 32 is a view illustrating a concept of frequency interleaving applied to a single super-frame according to an embodiment of the present invention.

[588] A frequency interleaver 7020 according to an embodiment of the present invention may change interleaving seed every pair-wise OFDM symbol in a single signal frame (symbol index reset) and change interleaving seed to be used in a single signal frame by every frame (frame index reset). Consequently, the frequency interleaver 7020 according to an embodiment of the present invention may change interleaving seed in a super-frame (super-frame index reset).

[589] Accordingly, the frequency interleaver 7020 according to an embodiment of the present may logically and effectively interleave all OFDM symbols in a super-frame.

[590] FIG. 33 is a view illustrating logical operation mechanism of frequency interleaving applied to a single super-frame according to an embodiment of the present invention.

[591] FIG. 33 illustrates logical operation mechanism of a frequency interleaver 7020 and related parameter thereof, for effectively changing interleaving seeds to be used the one super-frame described with reference to FIG. 32.

[592] As described above, in the present invention, various interleaving seeds may be effectively generated by cyclic-shifting one main interleaving seed by as much as an arbitrary offset. As illustrated in FIG. 33, according to an embodiment of the present invention, the aforementioned offset may be differently generated for each frame and each of pair-wise OFDM symbol to generate different interleaving seeds. Hereinafter,

the logical operation mechanism will be described.

[593] As illustrated in a lower block of FIG. 33, a frequency interleaver 7020 according to an embodiment of the present invention may randomly generate a frame offset for each signal frame using an input frame index. The frame offset according to an embodiment of the present invention may be generated by a frame offset generator included in a frequency interleaver 7020. In this case, when super-frame index is reset, a frame offset applied to each frame is generated for each signal frame in each super-frame identified according to a super-frame index.

[594] As illustrated in a middle block of FIG. 33, a frequency interleaver 7020 according to an embodiment of the present invention may randomly generate a symbol offset to be applied to each OFDM symbol included in each signal frame using an input symbol index. The symbol offset according to an embodiment of the present invention may be generated by a symbol offset generator included in a frequency interleaver 7020. In this case, when a frame index is reset, a symbol offset for each symbol is generated for symbols in each signal frame identified according to a frame index. In addition, the frequency interleaver 7020 according to an embodiment of the present invention may generate various interleaving seeds by cyclic-shifting a main interleaving seed on each OFDM symbol by as much as a symbol offset.

[595] Then, as illustrated in an upper block of FIG. 33, a frequency interleaver 7020 according to an embodiment of the present invention may perform random FI on cells included in each OFDM symbol using an input cell index. A random FI parameter according to an embodiment of the present invention may be generated by a random FI generator included in the frequency interleaver 7020.

[596] FIG. 34 illustrates math figures of logical operation mechanism of frequency interleaving applied to a single super-frame according to an embodiment of the present invention.

[597] In detail, FIG. 34 illustrates a correlation of the aforementioned frame offset parameter, symbol offset, parameter, and random FI applied to a cell included in each OFDM. As illustrated in FIG. 34, an offset to be used in an OFDM symbol may be generated through a hierarchical structure of the aforementioned frame offset generator and the aforementioned symbol offset generator. In this case, the frame offset generator and the symbol offset generator may be designed using an arbitrary random generator.

[598] FIG. 35 illustrates an operation of a memory bank according to an embodiment of the present invention.

[599] As described above, two memory banks according to an embodiment of the present invention may apply an arbitrary interleaving seed generated via the aforementioned procedure to each pair-wise OFDM symbol. In addition, each memory bank may

change interleaving seed every pair-wise OFDM symbol.

[600] FIG. 36 illustrates a frequency deinterleaving procedure according to an embodiment of the present invention.

[601] A broadcast signal receiving apparatus according to an embodiment of the present invention may perform an inverse procedure of the aforementioned frequency interleaving procedure. FIG. 36 illustrates single-memory deinterleaving (FDI) for input sequential OFDM symbols.

[602] Basically, frequency deinterleaving operation follows to the inverse processing of frequency interleaving operation. For a single-memory use, no further processing is required.

[603] When pair-wise OFDM symbols illustrated in a left portion of FIG. 36 are input, the broadcast signal receiving apparatus according to an embodiment of the present invention may perform the aforementioned reading and writing operation using a single memory, as illustrated in a right portion of FIG. 36. In this case, the broadcast signal receiving apparatus according to an embodiment of the present invention may generate a memory-index and perform frequency deinterleaving (reading and writing) corresponding to an inverse procedure of frequency interleaving (writing and reading) performed by a broadcast signal transmitting apparatus. The benefit is inherently caused by the proposed pair-wise ping-pong interleaving architecture.

[604] The following mathematical formulae show the aforementioned reading-writing operation.

[605] MathFigure 12

[Math.12]

$$for\ j = 0, 1, \ldots, N_{sym}\ and\ k = 0, 1, \ldots, N_{data}$$

$$F_j(C_j(k)) = X_j(k)$$

$$where\ C_j(k)\ is\ a\ random\ seed\ generated\ by\ a\ random\ generator,$$

$$in\ the\ ith\ pair-wise\ OFDM\ symbol$$

$$F_j = [F_j(0),\ F_j(1), \ldots,\ F_j(N_{data}-2),\ F_j(N_{data}-1)],$$

$$where\ N_{data}\ is\ the\ number\ of\ data\ cells$$

$$X_j = [X_j(0),\ X_j(1),\ \ldots,\ X_j(N_{data}-2),\ X_j(N_{data}-1)]$$

[606] MathFigure 13

[Math.13]

$$for\ j = 0, 1, \ldots, N_{sym}\ and\ k = 0, 1, \ldots, N_{data}$$

$$F_j(k) = X_j(C_j(k))$$

$$where\ C_j(k)\ is\ the\ same\ random\ seed\ used\ for\ the\ first\ symbol$$

$$F_j = [F_j(0), F_j(1), \ldots, F_j(N_{data} - 2), F_j(N_{data} - 1)],$$

$$where\ N_{data}\ is\ the\ number\ of\ data\ cells$$

$$X_j = [X_j(0), X_j(1), \ldots, X_j(N_{data} - 2), X_j(N_{data} - 1)]$$

[607] The above math figure 12 is for the first OFDM symbol, i.e., (j mod 2) = 0 of the ith pair-wise OFDM symbol. The above math figure 13 is for the second OFDM symbol, i.e., (j mod 2) = 1 of the ith pair-wise OFDM symbol. Fj denotes an interleaved vector of the jth OFDM symbol (vector) and Xj denotes an input vector of the jth OFDM symbol (vector). As shown in the math figures, the reading-writing operation according to an embodiment of the present invention may be performed by applying one random seed generated by an arbitrary random generator to a pair-wise OFDM symbol.

[608] FIG. 37 is a view illustrates concept of frequency interleaving applied to a single signal frame according to an embodiment of the present invention.

[609] As described above, a frequency interleaver 7020 according to an embodiment of the present invention may change interleaving seed every pair-wise OFDM symbol in a single frame. Details thereof will be described below.

[610] FIG. 38 is a view illustrating logical operation mechanism of frequency interleaving applied to a single signal frame according to an embodiment of the present invention.

[611] FIG. 38 illustrates logical operation mechanism of a frequency interleaver 7020 and related parameter thereof, for effectively changing interleaving seeds to be used the one single signal frame described with reference to FIG. 37.

[612] As described above, in the present invention, various interleaving seed can be effectively generated by cyclic-shifting one main interleaving seed by as much as an arbitrary symbol offset. As illustrated in FIG. 38, according to an embodiment of the present invention, the aforementioned symbol offset may be differently generated for each pair-wise OFDM symbol to generate different interleaving seeds. In this case, the symbol offset may be differently generated for each pair-wise OFDM symbol using an arbitrary random symbol offset generator.

[613] Hereinafter, the logical operation mechanism will be described.

[614] As illustrated in a lower block of FIG. 38, a frequency interleaver 7020 according to

an embodiment of the present invention may randomly generate a symbol offset to be applied to each OFDM symbol included in each signal frame using an input symbol index. The symbol offset (or a random symbol offset) according to an embodiment of the present invention may be generated by an arbitrary random generator (or a symbol offset generator) included in a frequency interleaver 7020. In this case, when a frame index is reset, the symbol offset for each symbol is generated for symbols in each signal frame identified according to a frame index. In addition, the frequency interleaver 7020 according to an embodiment of the present invention may generate various interleaving seeds by cyclic-shifting a main interleaving seed for each OFDM symbol by as much as the generated symbol offset.

[615] Then, as illustrated in an upper block of FIG. 38, a frequency interleaver 7020 according to an embodiment of the present invention may perform random FI on cells included in each OFDM symbol using an input cell index. A random FI parameter according to an embodiment of the present invention may be generated by a random FI generator included in a frequency interleaver 7020.

[616] FIG. 39 illustrates math figures of logical operation mechanism of frequency interleaving applied to a single signal frame according to an embodiment of the present invention.

[617] FIG. 39 illustrates a correlation of the aforementioned symbol offset parameter and a parameter of random FI applied to a cell included in each OFDM. As illustrated in FIG. 39, an offset to be used in each OFDM symbol may be generated through a hierarchical structure of the aforementioned symbol offset generator. In this ca se, the symbol offset generator may be designed using an arbitrary random generator.

[618] The following math figure shows a change procedure of interleaving seed in each of the aforementioned memory banks.

[619] MathFigure 14

[Math.14]

$$\text{for } j = 0, 1, \ldots, N_{sym} \text{ and for } k = 0, 1, \ldots, N_{data},$$

$$F_j(C_j(k)) = X_j(k)$$

$$\text{where } C_j(k) = (T(k) + S_{\lfloor j/2 \rfloor}) \bmod N_{data}$$

$T(k)$ *is a main interleaving seed generated by a random generator, used in the main FI*

$S_{\lfloor j/2 \rfloor}$ *is a random symbol offset generated by a random generator, used in the jth pair – wise OFDM symbol*

[620] MathFigure 15

[Math.15]

$$for\ \ j = 0, 1, \ldots, N_{sym}\ \ and\ \ k = 0, 1, \ldots, N_{data}$$

$$F_j(k) = X_j(C_j(k))$$

$$where\ C_j(k)\ is\ the\ same\ random\ seed\ used\ for\ the\ first\ symbol$$

[621] The above math figure 14 is for the first OFDM symbol, i.e., (j mod 2) = 0 of the ith pair-wise OFDM symbol and the above math figure 15 is for the second OFDM symbol, i.e., (j mod 2) = 1 of the ith pair-wise OFDM symbol.

[622] FIG. 40 is a view illustrating single-memory deinterleaving for input sequential OFDM symbols.

[623] FIG. 40 is a view illustrating concept of a broadcast signal receiving apparatus or a frequency deinterleaver thereof, for applying interleaving seed used in a broadcast signal transmitting apparatus (or a frequency interleaver 7020) to each pair-wise OFDM symbol to perform deinterleaving.

[624] As described above, the broadcast signal receiving apparatus according to an embodiment of the present invention may perform an inverse procedure of the aforementioned frequency interleaving procedure using a single memory. FIG. 40 illustrates an operation of the broadcast signal receiving apparatus for processing single-memory deinterleaving (FDI) for input sequential OFDM symbols.

[625] The broadcast signal receiving apparatus according to an embodiment of the present invention may perform an inverse procedure of the aforementioned operation of a frequency interleaver 7020. Thus, deinterleaving seeds correspond to the aforementioned interleaving seed.

[626] As described above, an OFDM generation block 1030 may perform FFT transformation on input data. According to an embodiment of the present invention, an FFT size may be 4K, 8K, 16K, 32K, or the like, and an FFT mode indicating the FFT size may be defined. The aforementioned FFT mode may be signaled via a preamble (or a preamble signal, a preamble symbol) in a signal frame or signal via PLS-pre or PLS-prost. The FFT size may be changed according to a designer's intention.

[627] A frequency interleaver 7020 or an interleaving seed generator included therein according to an embodiment of the present invention may perform an operation according to the aforementioned FFT mode. In addition, an interleaving seed generator according to an embodiment of the present invention may include a random seed

generator or a quasi-random interleaving seed generator. The quasi-random interleaving seed generator may be an embodiment of the random seed generator. The random seed generator and the quasi-random interleaving seed generator may be referred as an interleaving address generator and it may be changed by the designer's intention. Also, both of the random seed generator and the quasi-random interleaving seed generator may include a first generator and a second generator. The first generator is for generating a main interleaving seed generator and the second generator is for generating a symbol offset. The name of the first generator and the second generator can be changed according to the designer's indention. Hereinafter, an operation of the interleaving seed generator according to each FFT mode is divided into an operation of the random seed generator and an operation of the quasi-random interleaving seed generator and will be described.

[628] Hereinafter, the random seed generator for a 4K FFT mode will be described.

[629] As described above, the random seed generator according to an embodiment of the present invention may apply different interleaving seeds to respective OFDM symbols to acquire frequency diversity. Logical composition of the random seed generator may include a random main-seed generator (or a random main interleaving-seed generator) (Cj(K)) for interleaving cells in a single OFDM symbol and a random symbol-offset generator ($S_{\lfloor j/2 \rfloor}$) for changing a symbol offset.

[630] The random main-seed generator may generate the aforementioned random FI parameter. That is, the random main-seed generator may generate seed for interleaving cells in a single OFDM symbol.

[631] The random main-seed generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 4K FFT mode, the random main-seed generator may include a 1 bit spreader and an 11 bit-randomizer. The random main-seed generator or the randomizer according to an embodiment of the present invention may be referred as a main-PRBS(Pseudo Random Bit Stream) generator which is defined based on the 11-bit binary word sequence (or binary sequence).

[632] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset of each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may

include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for the respective FFT modes. According to an embodiment of the present invention, in the case of 4K FFT mode, a (12-k) bit-randomizer may be used. The (X-k) bits-randomizer according to an embodiment of the present invention may be referred as a sub-PRBS generator which is defined based on (12-k) bit binary word sequence (or binary sequence).

[633] The aforementioned spreader and randomizer may be used to achieve spreading and random effects during generation of the interleaving seed.

[634] FIG. 41 is a view illustrating an output signal of a time interleaver according to an embodiment of the present invention.

[635] As above described, the time interleaver according to an embodiment of the present invention may perform a column-wise writing operation and a row-wise reading operation on one FEC block, as illustrated in a left portion of FIG. 41. A right block of FIG. 41 indicates an output signal of the time interleaver and the output signal is input to a frequency interleaver 7020 according to an embodiment of the present invention.

[636] Thus, one FEC block is periodically spread in each FI block. Accordingly, in order to increase the robustness of a channel with strong periodic properties, the aforementioned random interleaving seed generator may be used.

[637] FIG. 42 is a view of a 4K FFT mode random seed generator according to an embodiment of the present invention.

[638] The 4K FFT mode random seed generator according to an embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a memory-index check, a random symbol-offset generator, and a modulo operator. As described above, the random main-seed generator may include a spreader and a randomizer. Hereinafter, an operation of each block will be described.

[639] The (cell) spreader may be operated using an upper portion of n-bit of total 12-bit and may function as a multiplexer based on a look-up table. In the case of 4K FFT mode, the (cell) spreader may be a 1-bit multiplexer (or toggling).

[640] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, in the case of 4K FFT mode, the randomizer may be a PN generator that considers 11-bit. This can be changed according to a designer's intention. Also the spreader and the randomizer are operated through multiplexer and PN generator, respectively.

[641] The memory-index check may not use seed when a memory-index generated by the spreader and the randomizer is greater than Ndata and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata. The Ndata according to the embodiment of the present invention is equal to the number of the data cells.

[642] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting main interleaving-seed generated by the main-interleaving seed generator for each pair-wise OFDM symbol. A detailed operation will be described below.

[643] The modulo operator may be operated when a result value, obtained by adding a symbol-offset output by the random symbol-offset generator for each pair-wise OFDM symbol to the memory-index output by the memory-index check, exceeds Ndata. Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[644] FIG. 43 illustrates math figures representing an operation of a 4K FFT mode random seed generator according to an embodiment of the present invention.

[645] The math figures illustrated in an upper portion of FIG. 43 show initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 11th primitive polynomial and the initial value may be changed by arbitrary values.

[646] The math figures illustrated in a lower portion of FIG. 43 show procedures of calculating and outputting main-interleaving seed for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset may be applied to each pair-wise OFDM in the same way.

[647] FIG. 44 is a view illustrating a 4K FFT mode random symbol-offset generator according to an embodiment of the present invention.

[648] As above described, the random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer.

[649] Hereinafter, each block will be described.

[650] The k bits-spreader may be operated through a 2k multiplexer and may be optimally designed to maximize inter-symbol spreading properties (or to minimize correlation properties).

[651] The randomizer may be operated through a N bits-PN generator and designed to provide randomness.

[652] The 4K FFT mode random symbol-offset generator may include a 0/1/2 bits-spreader and a 12/11/10 bits-random generator (or a PN generator). Details will be described below.

[653] FIG. 45 illustrates math figures showing operations of a random symbol-offset generator and a random Symbol-offset generator for 4K FFT mode including a 0 bits-spreader and a 12 bits-PN generator according to an embodiment of the present invention.

[654] (a) illustrates a random symbol-offset generator including a 0 bits-spreader and a 12 bits-PN generator. (b) illustrates an operation of a 4K FFT mode random Symbol-offset generator.

[655] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[656] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of the randomizer. In this case, the primitive polynomial may be 12th primitive polynomial and the initial value may be changed by arbitrary values.

[657] A math figure illustrated in a lower portion of (b) shows a procedure for calculating and outputting a symbol-offset for output signals of a spreader and a randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[658] FIG. 46 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 4K FFT mode including a 1 bits-spreader and an 11 bits-PN generator according to an embodiment of the present invention.

[659] (a) shows the random symbol-offset generator including a 1 bits-spreader and an 11 bits-PN generator. (b) shows a math figure representing an operation of a 4K FFT mode random symbol-offset generator.

[660] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[661] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 11th primitive polynomial and the initial value may be changed by arbitrary values.

[662] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting a symbol-offset for an output signal of the spreader and the randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[663] FIG. 47 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 4K FFT mode including a 2 bits-spreader and a 10 bits-PN generator according to an embodiment of the present invention.

[664] (a) shows the random Symbol-offset generator including a 2 bits-spreader and a 10 bits-PN generator. (b) shows a math figure representing an operation of a 4K FFT mode random symbol-offset generator.

[665] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[666] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be

10th primitive polynomial and the initial value may include arbitrary values.

[667] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting a symbol-offset for an output signal of the spreader and the randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[668] FIG. 48 is a view illustrating logical composition of a 4K FFT mode random seed generator according to an embodiment of the present invention.

[669] As described above, the 4K FFT mode random seed generator according to an embodiment of the present invention may include a random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[670] FIG. 48 illustrates the logical composition of a 4K FFT mode random seed generator formed by combining a random main interleaving-seed generator and a random symbol-offset generator. FIG. 48 illustrates an embodiment of the random main interleaving-seed generator including a 1 bit-spreader and an 11 bits-randomizer, and an embodiment of the random symbol-offset generator including a 2 bits-spreader and a 10 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[671] Hereinafter, a quasi-random interleaving seed generator for 4K FFT mode will be described.

[672] As described above, the quasi-random interleaving seed generator according to an embodiment of the present invention may apply different interleaving seeds to respective OFDM symbols to acquire frequency diversity. The logical composition of the quasi-random interleaving seed generator may include a main quasi-random seed generator ((or quasi-random main interleaving-seed generator) (Cj(K)) for interleaving cells in a single OFDM symbol and a random symbol-offset generator ($S_{\lfloor j/2 \rfloor}$) for changing a symbol offset.

[673] The main quasi-random seed generator may generate the aforementioned random FI parameter. That is, the main quasi-random seed generator may generate seed for interleaving cells in a single OFDM symbol.

[674] The main quasi-random seed generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 4K FFT mode, the main quasi-random seed generator may include a 3 bit spreader and a 9 bit-randomizer. The main quasi-random seed generator

or the randomizer according to an embodiment of the present invention may be referred as a main-PRBS generator which is defined based on the 11-bit binary word sequence (or binary sequence).

[675] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset for each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for respective FFT modes. According to an embodiment of the present invention, in the case of 4K FFT mode, a (12-k) bits-randomizer may be used. The (X-k) bits-randomizer according to an embodiment of the present invention may be referred as a sub-PRBS generator which is defined based on (12-k) bit binary word sequence (or binary sequence).

[676] The main roles of the spreader and the randomizer are as follows.

[677] Spreader: rendering a spreading effect to frequency interleaving (FI)

[678] Randomizer: rendering a random effect to FI

[679] FIG. 49 is a view illustrating an output signal of a time interleaver according to another embodiment of the present invention.

[680] The time interleaver according to an embodiment of the present invention may perform a column-wise writing operation and a row-wise reading operation on each FEC block with a size of 5, as illustrated in a left portion of FIG. 49. A right block of FIG. 49 indicates an output signal of the time interleaver and the output signal is input to a frequency interleaver 7020 according to an embodiment of the present invention.

[681] Thus, one FEC block has a length of 5 in each FI block and agglomerate in a burst form. Thus, in order to increase the robustness of a channel with strong burst error properties, interleaving seed having high spreading properties as well as high randomness is required. Accordingly, the aforementioned quasi-random interleaving seed generator may be used.

[682] FIG. 50 is a view illustrating a 4K FFT mode quasi-random interleaving-seed generator according to an embodiment of the present invention.

[683] The 4K FFT mode quasi-random interleaving-seed generator according to an embodiment of the present invention may include a spreader (3-bit toggling), a randomizer, a memory-index check, a random symbol-offset generator, and a modulo operator. As described above, the quasi-random main interleaving-seed generator may include a spreader and a randomizer. Hereinafter, an operation of each block will be described.

[684] The spreader may be operated through an n-bit multiplexer and may maximize (or minimize inter-cell correlation) inter-cell spreading. In the case of 4K FFT mode, the

spreader may use a look-up table that considers 3-bit.

[685] The randomizer may be operated as a (12-n) bits-PN generator and may provide randomness (or correlation properties). The randomizer according to an embodiment of the present invention may include bit shuffling. The bit shuffling optimizes spreading properties or random properties and is designed in consideration of Ndata. In the case of 4K FFT mode, the bit shuffling may use a 9-bit PN generator, which can be changed.

[686] The memory-index check may not use seed when a memory-index generated by the spreader and the randomizer is greater than Ndata and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata.

[687] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting main interleaving-seed generated by the main-interleaving seed generator for each pair-wise OFDM symbol. A detailed operation has been described with regard to the 4K FFT mode random main-seed generator and is not described again here.

[688] The modulo operator may be operated when a result value, obtained by adding a symbol-offset output by the random symbol-offset generator for each pair-wise OFDM symbol to the memory-index output by the memory-index check, exceeds Ndata. Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[689] FIG. 51 is math figures representing operations of 4K FFT mode bit shuffling and 4K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[690] (a) illustrates a math figure representing an operation of the 4K FFT mode bit shuffling and (b) illustrates a math figure representing an operation of the 4K FFT mode quasi-random interleaving seed generator.

[691] As illustrated in (a), the 4K FFT mode bit shuffling may mix bits of registers of a PN generator during calculation of a memory-index.

[692] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 9th primitive polynomial and the initial value may be changed by arbitrary values.

[693] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting main-interleaving seed for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset may be applied to each pair-wise OFDM symbol in the same way.

[694] FIG. 52 is a view illustrating logical composition of a 4K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[695] As described above, the 4K FFT mode quasi-random main interleaving-seed

generator according to an embodiment of the present invention may include a quasi-random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[696] FIG. 52 illustrates the logical composition of a 4K FFT mode quasi-random interleaving seed generator formed by combining a quasi-random main interleaving-seed generator and a random symbol-offset generator. FIG. 52 illustrates an embodiment of the quasi-random main interleaving-seed generator including a 3 bit-spreader and a 9 bits-randomizer and an embodiment of the random symbol-offset generator including a 2 bits-spreader and a 10 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[697] Hereinafter, the random seed generator for an 8K FFT mode will be described.

[698] As described above, the random seed generator according to an embodiment of the present invention may apply different interleaving seeds to respective OFDM symbols to acquire frequency diversity. Logical composition of the random seed generator may include a random main-seed generator (or a random main interleaving-seed generator) (Cj(K)) for interleaving cells in a single OFDM symbol and a random symbol-offset generator (

$$S_{\lfloor j/2 \rfloor}$$

for changing a symbol offset.

[699] The random main-seed generator may generate the aforementioned random FI parameter. That is, the random main-seed generator may generate seed for interleaving cells in a single OFDM symbol.

[700] The random main-seed generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 8K FFT mode, the random main-seed generator may include a 1 bit spreader and an 12 bit-randomizer. The random main-seed generator or the randomizer according to an embodiment of the present invention may be referred as a main-PRBS generator which is defined based on the 12-bit binary word sequence (or binary sequence).

[701] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset of each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for the respective FFT modes. According to an embodiment of the present invention, in the case of 8K FFT

mode, a (13-k) bit-randomizer may be used. The (X-k) bits-randomizer according to an embodiment of the present invention may be referred as a sub-PRBS generator which is defined based on (13-k) bit binary word sequence (or binary sequence).

[702] The aforementioned spreader and randomizer may be used to achieve spreading and random effects during generation of the interleaving seed.

[703] Details of the output signal of a time interleaver according to an embodiment of the present invention have been described above.

[704] FIG. 53 is a view of an 8K FFT mode random seed generator according to an embodiment of the present invention.

[705] The 8K FFT mode random seed generator according to an embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a memory-index check, a random symbol-offset generator, and a modulo operator. As described above, the random main-seed generator may include a spreader and a randomizer. Hereinafter, an operation of each block will be described.

[706] The (cell) spreader may be operated using an upper portion of n-bit of total 13-bit and may function as a multiplexer based on a look-up table. In the case of 8K FFT mode, the (cell) spreader may be a 1-bit multiplexer (or toggling).

[707] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, in the case of 8K FFT mode, the randomizer may be a PN generator that considers 12-bit. This can be changed according to a designer's intention. Also the spreader and the randomizer are operated through multiplexer and PN generator, respectively.

[708] The memory-index check may not use seed when a memory-index generated by the spreader and the randomizer is greater than Ndata and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata. The Ndata according to the embodiment of the present invention is equal to the number of the data cells.

[709] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting main interleaving-seed generated by the main-interleaving seed generator for each pair-wise OFDM symbol. A detailed operation will be described below.

[710] The modulo operator may be operated when a result value, obtained by adding a symbol-offset output by the random symbol-offset generator for each pair-wise OFDM symbol to the memory-index output by the memory-index check, exceeds Ndata. Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[711] FIG. 54 illustrates math figures representing an operation of an 8K FFT mode random seed generatoraccording to an embodiment of the present invention.

[712] The math figures illustrated in an upper portion of FIG. 54 show initial value setting

and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 12th primitive polynomial and the initial value may be changed by arbitrary values.

[713] The math figures illustrated in a lower portion of FIG. 54 show procedures of calculating and outputting main-interleaving seed for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset may be applied to each pair-wise OFDM in the same way.

[714] FIG. 55 is a view illustrating an 8K FFT mode random symbol-offset generator according to an embodiment of the present invention.

[715] As above described, the random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer.

[716] Hereinafter, each block will be described.

[717] The k bits-spreader may be operated through a 2k multiplexer and may be optimally designed to maximize inter-symbol spreading properties (or to minimize correlation properties).

[718] The randomizer may be operated through a N bits-PN generator and designed to provide randomness.

[719] The 8K FFT mode random symbol-offset generator may include a 0/1/2 bits-spreader and a 13/12/11 bits-random generator (or a PN generator). Details will be described below.

[720] FIG. 56 illustrates math figures showing operations of a random symbol-offset generator and a random Symbol-offset generator for 8K FFT mode including a 0 bits-spreader and a 13 bits-PN generator according to an embodiment of the present invention.

[721] (a) illustrates a random symbol-offset generator including a 0 bits-spreader and a 13 bits-PN generator. (b) illustrates an operation of an 8K FFT mode random Symbol-offset generator.

[722] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[723] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of the randomizer. In this case, the primitive polynomial may be 13th primitive polynomial and the initial value may be changed by arbitrary values.

[724] A math figure illustrated in a lower portion of (b) shows a procedure for calculating and outputting a symbol-offset for output signals of a spreader and a randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[725] FIG. 57 illustrates math figures illustrating operations of a random symbol-offset

generator and a random Symbol-offset generator for 8K FFT mode including a 1 bits-spreader and an 12 bits-PN generator according to an embodiment of the present invention.

[726] (a) shows the random symbol-offset generator including a 1 bits-spreader and a 12 bits-PN generator. (b) shows a math figure representing an operation of an 8K FFT mode random symbol-offset generator.

[727] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[728] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 12th primitive polynomial and the initial value may be changed by arbitrary values.

[729] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting a symbol-offset for an output signal of the spreader and the randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[730] FIG. 58 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 8K FFT mode including a 2 bits-spreader and an 11 bits-PN generator according to an embodiment of the present invention.

[731] (a) shows the random Symbol-offset generator including a 2 bits-spreader and an 11 bits-PN generator. (b) shows a math figure representing an operation of an 8K FFT mode random symbol-offset generator.

[732] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[733] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 11th primitive polynomial and the initial value may include arbitrary values.

[734] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting a symbol-offset for an output signal of the spreader and the randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[735] FIG. 59 is a view illustrating logical composition of an 8K FFT mode random seed generator according to an embodiment of the present invention.

[736] As described above, the 8K FFT random seed generator according to an embodiment of the present invention may include a random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[737] FIG. 59 illustrates the logical composition of an 8K FFT mode random seed generator formed by combining a random main interleaving-seed generator and a random symbol-offset generator. FIG. 59 illustrates an embodiment of the random main interleaving-seed generator including a 1 bit-spreader and a 12 bits-randomizer, and an embodiment of the random symbol-offset generator including a 2 bits-spreader and an 11 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[738] Hereinafter, a quasi-random interleaving seed generator for 8K FFT mode will be described.

[739] As described above, the quasi-random interleaving seed generator according to an embodiment of the present invention may apply different interleaving seeds to respective OFDM symbols to acquire frequency diversity. The logical composition of the quasi-random interleaving seed generator may include a main quasi-random seed generator (or quasi-random main interleaving-seed generator) (Cj(K)) for interleaving cells in a single OFDM symbol and a random symbol-offset generator ($S_{\lfloor j/2 \rfloor}$) for changing a symbol offset.

[740] The main quasi-random seed generator may generate the aforementioned random FI parameter. That is, the main quasi-random seed generator may generate seed for interleaving cells in a single OFDM symbol.

[741] The main quasi-random seed generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 8K FFT mode, the main quasi-random seed generator may include a 3 bit spreader and a 10 bit-randomizer. The quasi-random seed generator or the randomizer according to an embodiment of the present invention may be referred as a main-PRBS generator which is defined based on the 10-bit binary word sequence (or binary sequence).

[742] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset for each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for respective FFT modes. According to an embodiment of the present invention, in the case of 8K FFT mode, a (13-k) bits-randomizer may be used. The (X-k) bits-randomizer according to an embodiment of the present invention may be referred as a sub-PRBS generator

which is defined based on (13-k) bit binary word sequence (or binary sequence).

[743] The main roles of the spreader and the randomizer are as follows.

[744] Spreader: rendering a spreading effect to frequency interleaving (FI)

[745] Randomizer: rendering a random effect to FI

[746] Details of the output signal of a time interleaver according to an embodiment of the present invention have been described above.

[747] FIG. 60 is a view illustrating an 8K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[748] The 8K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention may include a spreader (3-bit toggling), a randomizer, a memory-index check, a random symbol-offset generator, and a modulo operator. As described above, the quasi-random main interleaving-seed generator may include a spreader and a randomizer. Hereinafter, an operation of each block will be described.

[749] The spreader may be operated through an n-bit multiplexer and may maximize (or minimize inter-cell correlation) inter-cell spreading. In the case of 8K FFT mode, the spreader may use a look-up table that considers 3-bit.

[750] The randomizer may be operated as a (13-n) bits-PN generator and may provide randomness (or correlation properties). The randomizer according to an embodiment of the present invention may include bit shuffling. The bit shuffling optimizes spreading properties or random properties and is designed in consideration of Ndata. In the case of 8K FFT mode, the bit shuffling may use a 10-bit PN generator, which can be changed.

[751] The memory-index check may not use seed when a memory-index generated by the spreader and the randomizer is greater than Ndata and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata.

[752] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting main interleaving-seed generated by the main-interleaving seed generator for each pair-wise OFDM symbol. A detailed operation has been described with regard to the 8K FFT mode random main-seed generator and is not described again here.

[753] The modulo operator may be operated when a result value, obtained by adding a symbol-offset output by the random symbol-offset generator for each pair-wise OFDM symbol to the memory-index output by the memory-index check, exceeds Ndata. Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[754] FIG. 61 is math figures representing operations of 8K FFT mode bit shuffling and 8K FFT mode quasi-random interleaving seed generator according to an embodiment of

the present invention.

[755] (a) illustrates a math figure representing an operation of the 8K FFT mode bit shuffling and (b) illustrates a math figure representing an operation of the 8K FFT mode quasi-random interleaving seed generator.

[756] As illustrated in (a), the 8K FFT mode bit shuffling may mix bits of registers of a PN generator during calculation of a memory-index.

[757] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 10th primitive polynomial and the initial value may be changed by arbitrary values.

[758] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting main-interleaving seed for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset may be applied to each pair-wise OFDM symbol in the same way.

[759] FIG. 62 is a view illustrating logical composition of an 8K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[760] As described above, the 8K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention may include a quasi-random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[761] FIG. 62 illustrates the logical composition of an 8K FFT mode quasi-random interleaving seed generator formed by combining a quasi-random main interleaving-seed generator and a random symbol-offset generator. FIG. 62 illustrates an embodiment of the quasi-random main interleaving-seed generator including a 3 bit-spreader and a 10 bits-randomizer and an embodiment of the random symbol-offset generator including a 2 bits-spreader and an 11 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[762] Hereinafter, the random seed generator for a 16K FFT mode will be described.

[763] As described above, the random seed generator according to an embodiment of the present invention may apply different interleaving seeds to respective OFDM symbols to acquire frequency diversity. Logical composition of the random seed generator may include a random main-seed generator (or a random main interleaving-seed generator) (Cj(K)) for interleaving cells in a single OFDM symbol and a random symbol-offset generator ($S_{\lfloor j/2 \rfloor}$) for changing a symbol offset.

[764] The random main-seed generator may generate the aforementioned random FI

parameter. That is, the random main-seed generator may generate seed for interleaving cells in a single OFDM symbol.

[765] The random main-seed generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 16K FFT mode, the random main-seed generator may include a 1 bit spreader and an 13 bit-randomizer. The random main-seed generator or the randomizer according to an embodiment of the present invention may be referred as a main-PRBS generator which is defined based on the 13-bit binary word sequence (or binary sequence).

[766] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset of each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for the respective FFT modes. According to an embodiment of the present invention, in the case of 16K FFT mode, a (14-k) bit-randomizer may be used. The (X-k) bits-randomizer according to an embodiment of the present invention may be referred as a sub-PRBS generator which is defined based on (14-k) bit binary word sequence (or binary sequence).

[767] The aforementioned spreader and randomizer may be used to achieve spreading and random effects during generation of the interleaving seed.

[768] Details of the output signal of a time interleaver according to an embodiment of the present invention have been described above.

[769] FIG. 63 is a view of a 16K FFT mode random seed generator according to an embodiment of the present invention.

[770] The 16K FFT mode random seed generator according to an embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a memory-index check, a random symbol-offset generator, and a modulo operator. As described above, the random main-seed generator may include a spreader and a randomizer. Hereinafter, an operation of each block will be described.

[771] The (cell) spreader may be operated using an upper portion of n-bit of total 14-bit and may function as a multiplexer based on a look-up table. In the case of 16K FFT mode, the (cell) spreader may be a 1-bit multiplexer (or toggling).

[772] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, in the case of 16K FFT mode, the randomizer may be a PN generator that considers 13-bit. This can be changed according to a designer's intention. Also the spreader and the randomizer are operated

through multiplexer and PN generator, respectively.

[773] The memory-index check may not use seed when a memory-index generated by the spreader and the randomizer is greater than Ndata and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata. The Ndata according to the embodiment of the present invention is equal to the number of the data cells.

[774] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting main interleaving-seed generated by the main-interleaving seed generator for each pair-wise OFDM symbol. A detailed operation will be described below.

[775] The modulo operator may be operated when a result value, obtained by adding a symbol-offset output by the random symbol-offset generator for each pair-wise OFDM symbol to the memory-index output by the memory-index check, exceeds Ndata. Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[776] FIG. 64 illustrates math figures representing an operation of a 16K FFT mode random seed generator according to an embodiment of the present invention.

[777] The math figures illustrated in an upper portion of FIG. 64 show initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 13th primitive polynomial and the initial value may be changed by arbitrary values.

[778] The math figures illustrated in a lower portion of FIG. 64 show procedures of calculating and outputting main-interleaving seed for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset may be applied to each pair-wise OFDM in the same way.

[779] FIG. 65 is a view illustrating a 16K FFT mode random symbol-offset generator according to an embodiment of the present invention.

[780] As above described, the random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer.

[781] Hereinafter, each block will be described.

[782] The k bits-spreader may be operated through a 2k multiplexer and may be optimally designed to maximize inter-symbol spreading properties (or to minimize correlation properties).

[783] The randomizer may be operated through a N bits-PN generator and designed to provide randomness.

[784] The 16K FFT mode random symbol-offset generator may include a 0/1/2 bits-spreader and a 14/13/12 bits-random generator (or a PN generator). Details will be described below.

[785] FIG. 66 illustrates math figures showing operations of a random symbol-offset

generator and a random Symbol-offset generator for 16K FFT mode including a 0 bits-spreader and a 14 bits-PN generator according to an embodiment of the present invention.

[786] (a) illustrates a random symbol-offset generator including a 0 bits-spreader and a 14 bits-PN generator. (b) illustrates an operation of a 16K FFT mode random Symbol-offset generator.

[787] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[788] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of the randomizer. In this case, the primitive polynomial may be 14th primitive polynomial and the initial value may be changed by arbitrary values.

[789] A math figure illustrated in a lower portion of (b) shows a procedure for calculating and outputting a symbol-offset for output signals of a spreader and a randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[790] FIG. 67 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 16K FFT mode including a 1 bits-spreader and a 13 bits-PN generator according to an embodiment of the present invention.

[791] (a) shows the random symbol-offset generator including a 1 bits-spreader and a 13 bits-PN generator. (b) shows a math figure representing an operation of a 16K FFT mode random symbol-offset generator.

[792] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[793] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 13th primitive polynomial and the initial value may be changed by arbitrary values.

[794] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting a symbol-offset for an output signal of the spreader and the randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[795] FIG. 68 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 16K FFT mode including a 2 bits-spreader and a 12 bits-PN generator according to an embodiment of the present invention.

[796] (a) shows the random Symbol-offset generator including a 2 bits-spreader and a 12

bits-PN generator. (b) shows a math figure representing an operation of a 16K FFT mode random symbol-offset generator.

[797] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[798] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 12th primitive polynomial and the initial value may include arbitrary values.

[799] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting a symbol-offset for an output signal of the spreader and the randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[800] FIG. 69 is a view illustrating logical composition of a 16K FFT mode random seed generator according to an embodiment of the present invention.

[801] As described above, the 16K FFT mode random seed generator according to an embodiment of the present invention may include a random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[802] FIG. 69 illustrates the logical composition of a 16K FFT mode random seed generator formed by combining a random main interleaving-seed generator and a random symbol-offset generator. FIG. 69 illustrates an embodiment of the random main interleaving-seed generator including a 1 bit-spreader and an 13 bits-randomizer, and an embodiment of the random symbol-offset generator including a 2 bits-spreader and a 12 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[803] Hereinafter, a quasi-random interleaving seed generator for 16K FFT mode will be described.

[804] As described above, the quasi-random interleaving seed generator according to an embodiment of the present invention may apply different interleaving seeds to respective OFDM symbols to acquire frequency diversity. The logical composition of the quasi-random interleaving seed generator may include a main quasi-random seed generator (or quasi-random main interleaving-seed generator) (Cj(K)) for interleaving cells in a single OFDM symbol and a random symbol-offset generator ($S_{\lfloor j/2 \rfloor}$) for changing a symbol offset.

[805] The main quasi-random seed generator may generate the aforementioned random FI parameter. That is, the main quasi-random seed generator may generate seed for in-

terleaving cells in a single OFDM symbol.

[806] The main quasi-random seed generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 16K FFT mode, the main quasi-random seed generator may include a 3 bit spreader and an 11 bit-randomizer. The main quasi-random seed generator or randomizer according to an embodiment of the present invention may be referred as a main-PRBS generator which is defined based on the 11-bit binary word sequence (or binary sequence).

[807] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset for each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for respective FFT modes. According to an embodiment of the present invention, in the case of 16K FFT mode, a (14-k) bits-randomizer may be used. The (X-k) bits-randomizer according to an embodiment of the present invention may be referred as a sub-PRBS generator which is defined based on (14-k) bit binary word sequence (or binary sequence).

[808] The main roles of the spreader and the randomizer are as follows.

[809] Spreader: rendering a spreading effect to frequency interleaving (FI)

[810] Randomizer: rendering a random effect to FI

[811] Details of the output signal of a time interleaver according to an embodiment of the present invention have been described above.

[812] FIG. 70 is a view illustrating a 16K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[813] The 16K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention may include a spreader (3-bit toggling), a randomizer, a memory-index check, a random symbol-offset generator, and a modulo operator. As described above, the random main-seed generator may include a spreader and a randomizer. Hereinafter, an operation of each block will be described.

[814] The spreader may be operated through an n-bit multiplexer and may maximize (or minimize inter-cell correlation) inter-cell spreading. In the case of 16K FFT mode, the spreader may use a look-up table that considers 3-bit.

[815] The randomizer may be operated as a (14-n) bits-PN generator and may provide randomness (or correlation properties). The randomizer according to an embodiment of the present invention may include bit shuffling. The bit shuffling optimizes spreading properties or random properties and is designed in consideration of Ndata. In the case

of 16K FFT mode, the bit shuffling may use an 11-bit PN generator, which can be changed.

[816] The memory-index check may not use seed when a memory-index generated by the spreader and the randomizer is greater than Ndata and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata.

[817] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting main interleaving-seed generated by the main-interleaving seed generator for each pair-wise OFDM symbol. A detailed operation has been described with regard to the 16K FFT mode random main-seed generator and is not described again here.

[818] The modulo operator may be operated when a result value, obtained by adding a symbol-offset output by the random symbol-offset generator for each pair-wise OFDM symbol to the memory-index output by the memory-index check, exceeds Ndata. Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[819] FIG. 71 is math figures representing operations of 16K FFT mode bit shuffling and 16K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[820] (a) illustrates a math figure representing an operation of the 16K FFT mode bit shuffling and (b) illustrates a math figure representing an operation of the 16K FFT mode quasi-random interleaving seed generator.

[821] As illustrated in (a), the 16K FFT mode bit shuffling may mix bits of registers of a PN generator during calculation of a memory-index.

[822] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 11th primitive polynomial and the initial value may be changed by arbitrary values.

[823] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting main-interleaving seed for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset may be applied to each pair-wise OFDM symbol in the same way.

[824] FIG. 72 is a view illustrating logical composition of a 16K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[825] As described above, the 16K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention may include a quasi-random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[826] FIG. 72 illustrates the logical composition of a 16K FFT mode quasi-random interleaving seed generator formed by combining a quasi-random main interleaving-seed

generator and a random symbol-offset generator. FIG. 72 illustrates an embodiment of the quasi-random main interleaving-seed generator including a 3 bit-spreader and an 11 bits-randomizer and an embodiment of the random symbol-offset generator including a 2 bits-spreader and a 12 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[827] Hereinafter, the random seed generator for a 32K FFT mode will be described.

[828] As described above, the random seed generator according to an embodiment of the present invention may apply different interleaving seeds to respective OFDM symbols to acquire frequency diversity. Logical composition of the random seed generator may include a random main-seed generator (or a random main interleaving-seed generator) (Cj(K)) for interleaving cells in a single OFDM symbol and a random symbol-offset generator ($S_{\lfloor j/2 \rfloor}$) for changing a symbol offset.

[829] The random main-seed generator may generate the aforementioned random FI parameter. That is, the random main-seed generator may generate seed for interleaving cells in a single OFDM symbol.

[830] The random main-seed generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 32K FFT mode, the random main-seed generator may include a 1 bit spreader and an 14 bit-randomizer. The random main-seed generator or the randomizer according to an embodiment of the present invention may be referred as a main-PRBS generator which is defined based on the 14-bit binary word sequence (or binary sequence).

[831] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset of each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for the respective FFT modes. According to an embodiment of the present invention, in the case of 32K FFT mode, a (15-k) bit-randomizer may be used. The (X-k) bits-randomizer according to an embodiment of the present invention may be referred as a sub-PRBS generator which is defined based on (15-k) bit binary word sequence (or binary sequence).

[832] The aforementioned spreader and randomizer may be used to achieve spreading and random effects during generation of the interleaving seed.

[833] Details of the output signal of a time interleaver according to an embodiment of the present invention have been described above.

[834] FIG. 73 is a view of a 32K FFT mode random seed generator according to an embodiment of the present invention.

[835] The 32K FFT mode random seed generator according to an embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a memory-index check, a random symbol-offset generator, and a modulo operator. As described above, the random main-seed generator may include a spreader and a randomizer. Hereinafter, an operation of each block will be described.

[836] The (cell) spreader may be operated using an upper portion of n-bit of total 15-bit and may function as a multiplexer based on a look-up table. In the case of 32K FFT mode, the (cell) spreader may be a 1-bit multiplexer (or toggling).

[837] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, in the case of 32K FFT mode, the randomizer may be a PN generator that considers 14-bit. This can be changed according to a designer's intention. Also the spreader and the randomizer are operated through multiplexer and PN generator, respectively.

[838] The memory-index check may not use seed when a memory-index generated by the spreader and the randomizer is greater than Ndata and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata. The Ndata according to the embodiment of the present invention is equal to the number of the data cells.

[839] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting main interleaving-seed generated by the main-interleaving seed generator for each pair-wise OFDM symbol. A detailed operation will be described below.

[840] The modulo operator may be operated when a result value, obtained by adding a symbol-offset output by the random symbol-offset generator for each pair-wise OFDM symbol to the memory-index output by the memory-index check, exceeds Ndata. Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[841] FIG. 74 illustrates math figures representing an operation of a 32K FFT mode random seed generator according to an embodiment of the present invention.

[842] The math figures illustrated in an upper portion of FIG. 74 show initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 14th primitive polynomial and the initial value may be changed by arbitrary values.

[843] The math figures illustrated in a lower portion of FIG. 74 show procedures of calculating and outputting main-interleaving seed for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset may be

applied to each pair-wise OFDM in the same way.

[844] FIG. 75 is a view illustrating a 32K FFT mode random symbol-offset generator according to an embodiment of the present invention.

[845] As above described, the random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer.

[846] Hereinafter, each block will be described.

[847] The k bits-spreader may be operated through a 2k multiplexer and may be optimally designed to maximize inter-symbol spreading properties (or to minimize correlation properties).

[848] The randomizer may be operated through a N bits-PN generator and designed to provide randomness.

[849] The 32K FFT mode random symbol-offset generator may include a 0/1/2 bits-spreader and a 15/14/13 bits-random generator (or a PN generator). Details will be described below.

[850] FIG. 76 illustrates math figures showing operations of a random symbol-offset generator and a random Symbol-offset generator for 32K FFT mode including a 0 bits-spreader and a 15 bits-PN generator according to an embodiment of the present invention.

[851] (a) illustrates a random symbol-offset generator including a 0 bits-spreader and a 15 bits-PN generator. (b) illustrates an operation of a 32K FFT mode random Symbol-offset generator.

[852] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[853] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of the randomizer. In this case, the primitive polynomial may be 12th primitive polynomial and the initial value may be changed by arbitrary values.

[854] A math figure illustrated in a lower portion of (b) shows a procedure for calculating and outputting a symbol-offset for output signals of a spreader and a randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[855] FIG. 77 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 32K FFT mode including a 1 bits-spreader and an 14 bits-PN generator according to an embodiment of the present invention.

[856] (a) shows the random symbol-offset generator including a 1 bits-spreader and an 14 bits-PN generator. (b) shows a math figure representing an operation of a 32K FFT

mode random symbol-offset generator.

[857] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[858] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 14th primitive polynomial and the initial value may be changed by arbitrary values.

[859] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting a symbol-offset for an output signal of the spreader and the randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[860] FIG. 78 illustrates math figures illustrating operations of a random symbol-offset generator and a random Symbol-offset generator for 32K FFT mode including a 2 bits-spreader and a 13 bits-PN generator according to an embodiment of the present invention.

[861] (a) shows the random Symbol-offset generator including a 2 bits-spreader and a 13 bits-PN generator. (b) shows a math figure representing an operation of a 32K FFT mode random symbol-offset generator.

[862] The random symbol-offset generator illustrated in (a) may be operated for each pair-wise OFDM symbol.

[863] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 13th primitive polynomial and the initial value may include arbitrary values.

[864] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting a symbol-offset for an output signal of the spreader and the randomizer. As illustrated in the math figure, the random symbol-offset generator may be operated for each pair-wise OFDM symbol. Accordingly, the length of an entire output offset may correspond to half of the length of an entire OFDM symbol.

[865] FIG. 79 is a view illustrating logical composition of a 32K FFT mode random seed generator according to an embodiment of the present invention.

[866] As described above, the 32K FFT mode random seed generator according to an embodiment of the present invention may include a random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[867] FIG. 79 illustrates the logical composition of a 32K FFT mode random seed generator formed by combining a random main interleaving-seed generator and a random symbol-offset generator. FIG. 79 illustrates an embodiment of the random main interleaving-seed generator including a 1 bit-spreader and an 14 bits-randomizer,

and an embodiment of the random symbol-offset generator including a 2 bits-spreader and a 13 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[868] Hereinafter, a quasi-random interleaving seed generator for 32K FFT mode will be described.

[869] As described above, the quasi-random interleaving seed generator according to an embodiment of the present invention may apply different interleaving seeds to respective OFDM symbols to acquire frequency diversity. The logical composition of the quasi-random interleaving seed generator may include a main quasi-random seed generator (or quasi-random main interleaving-seed generator) (Cj(K)) for interleaving cells in a single OFDM symbol and a random symbol-offset generator ( ) for changing a symbol offset.

[870] The main quasi-random seed generator may generate the aforementioned random FI parameter. That is, the main quasi-random seed generator may generate seed for interleaving cells in a single OFDM symbol.

[871] The main quasi-random seed generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 32K FFT mode, the main quasi-random seed generator may include a 3 bit spreader and an 12 bit-randomizer. The main quasi-random seed generator or the randomizer according to an embodiment of the present invention may be referred as a main-PRBS generator which is defined based on the 12-bit binary word sequence (or binary sequence).

[872] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset for each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for respective FFT modes. According to an embodiment of the present invention, in the case of 16K FFT mode, a (15-k) bits-randomizer may be used. The (X-k) bits-randomizer according to an embodiment of the present invention may be referred as a sub-PRBS generator which is defined based on (15-k) bit binary word sequence (or binary sequence).

[873] The main roles of the spreader and the randomizer are as follows.

[874] Spreader: rendering a spreading effect to frequency interleaving (FI)

[875] Randomizer: rendering a random effect to FI

[876] Details of the output signal of a time interleaver according to an embodiment of the present invention have been described above.

[877] FIG. 80 is a view illustrating a 32K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[878] The 32K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention may include a spreader (3-bit toggling), a randomizer, a memory-index check, a random symbol-offset generator, and a modulo operator. As described above, the quasi-random main-seed generator may include a spreader and a randomizer. Hereinafter, an operation of each block will be described.

[879] The spreader may be operated through an n-bit multiplexer and may maximize (or minimize inter-cell correlation) inter-cell spreading. In the case of 32K FFT mode, the spreader may use a look-up table that considers 3-bit.

[880] The randomizer may be operated as a (15-n) bits-PN generator and may provide randomness (or correlation properties). The randomizer according to an embodiment of the present invention may include bit shuffling. The bit shuffling optimizes spreading properties or random properties and is designed in consideration of Ndata. In the case of 32K FFT mode, the bit shuffling may use a 9-bit PN generator, which can be changed.

[881] The memory-index check may not use seed when a memory-index generated by the spreader and the randomizer is greater than Ndata and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata.

[882] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting main interleaving-seed generated by the main-interleaving seed generator for each pair-wise OFDM symbol. A detailed operation has been described with regard to the 32K FFT mode random main-seed generator and is not described again here.

[883] The modulo operator may be operated when a result value, obtained by adding a symbol-offset output by the random symbol-offset generator for each pair-wise OFDM symbol to the memory-index output by the memory-index check, exceeds Ndata. Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[884] FIG. 81 is math figures representing operations of 32K FFT mode bit shuffling and 32K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[885] (a) illustrates a math figure representing an operation of the 32K FFT mode bit shuffling and (b) illustrates a math figure representing an operation of the 32K FFT mode quasi-random interleaving seed generator.

[886] As illustrated in (a), the 32K FFT mode bit shuffling may mix bits of registers of a PN generator during calculation of a memory-index.

[887] A math figure illustrated in an upper portion of (b) shows initial value setting and

primitive polynomial of a randomizer. In this case, the primitive polynomial may be 12th primitive polynomial and the initial value may be changed by arbitrary values.

[888] A math figure illustrated in a lower portion of (b) shows a procedure of calculating and outputting main-interleaving seed for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset may be applied to each pair-wise OFDM symbol in the same way.

[889] FIG. 82 is a view illustrating logical composition of a 32K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention.

[890] As described above, the 32K FFT mode quasi-random interleaving seed generator according to an embodiment of the present invention may include a quasi-random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[891] FIG. 82 illustrates the logical composition of a 32K FFT mode quasi-random interleaving seed generator formed by combining a quasi-random main interleaving-seed generator and a random symbol-offset generator. FIG. 82 illustrates an embodiment of the quasi-random main interleaving-seed generator including a 3 bit-spreader and a 12 bits-randomizer and an embodiment of the random symbol-offset generator including a 2 bits-spreader and a 13 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[892] FIG. 83 is a change procedure for an interleaving seed in each memory bank according to another embodiment of the present invention.

[893] The block illustrated in an upper portion of FIG. 83 shows math figures for the first OFDM symbol, i.e., (j mod 2) = 0 of the ith OFDM symbol pair. The block illustrated in a lower portion of FIG. 83 shows math figures for the second OFDM symbol, i.e., (j mod 2) = 1 of the ith OFDM symbol pair.

[894] The word “a random generator” illustrated each portion of FIG. 83 may be a random interleaving-sequence generator described as follows. The random interleaving-sequence generator according to an embodiment of the present invention may be included in the frequency interleaver 7020.

[895] T(k) illustrated in an upper portion of FIG. 83 is a random sequence, it can be used as a same concept of a main random interleaving sequence or a single interleaving seed (or an interleaving seed). The Random sequence may be generated in a random interleaving-sequence generator or a random main-sequence generator which will be described later.

$$S_{\lfloor j/2 \rfloor}$$

is a symbol offset and be referred as a cyclic shifting value. The cyclic shifting value

can be generated based on sub PRBS sequence. The details will be described later.

[896] The interleaving process for the OFDM symbol pair in each memory bank-A/B is described as above, exploiting a single interleaving-seed. The available data cells (the output cells from the cell mapper 7010) to be interleaved in one OFDM symbol. The Ndata according to the embodiment of the present invention is equal to the number of the data cells. The maximum value of the Ndata can be referred as Nmax and Nmax is differently defined according to each FFT mode. For the OFDM symbol pair in each memory bank, the interleaved OFDM symbol pair is shown in FIG. 83.

[897] Hj(k) is the interleaving address for the interleaving seed generated by a random interleaving-sequence generator for each FFT mode. The composition of the random interleaving-sequence generator will be described later. As described above, the purpose of the frequency interleaver 7020, which operates on a single OFDM symbol, is to provide frequency diversity by randomly interleaving data cells. In order to get maximum interleaving gain in a single frame, a different interleaving-seed is used for every OFDM symbol pair comprised of two sequential OFDM symbols. As shown in FIG. 83, different interleaving seed can be generated based on the interleaving address generated by a random interleaving-sequence generator. Also, the different interleaving seed can be generated based on the cyclic shifting value as above mentioned. That means, the different interleaving address to be used every symbol pair may be generated by using the cyclic shifting value for every OFDM symbol pair.

[898] As described above, an OFDM generation block 1030 may perform FFT transformation on input data. Hereinafter, an operation of the frequency interleaver 7020 having the random interleaving-sequence generator according to another embodiment will be described. The random interleaving-sequence generator may be another embodiment of the interleaving seed generator which is described above. Therefore, the random interleaving-sequence generator may be referred as the random seed generator or the quasi-random interleaving seed generator or the interleaving address generator and it may be changed by the designer's intention. The random interleaving-sequence generator may include a first generator and a second generator. The first generator is for generating a main interleaving seed generator and the second generator is for generating a symbol offset. The name of the first generator and the second generator can be changed according to the designer's indention.

[899] As described above, an FFT size according to an embodiment of the present invention may be 4K, 8K, 16K, 32K, or the like, and it can be changed according to the designer's intention.

[900] Hereinafter, the random interleaving-sequence generator for a 4K FFT mode will be described. The random interleaving-sequence generator according to an embodiment of the present invention may be included in the frequency interleaver 7020 and is similar

to the random seed generator mentioned (mentioned above), the random interleaving-sequence generator has a different structure from the random seed generator.

[901] The random interleaving-sequence generator according to an embodiment of the present invention may apply different interleaving seeds to respective OFDM symbols to acquire frequency diversity. Logical composition of the random interleaving-sequence generator may include a random main-sequence generator (or a random main interleaving-sequence generator or a random main-interleaving seed generator) (Cj(K)) for interleaving cells in a single OFDM symbol and a random symbol-offset generator (

$S_{\lfloor j/2 \rfloor}$

) for changing a symbol offset(This parameter can be referred as a cyclic shifting value). The random main-sequence generator according to an embodiment of the present invention is similar to the random seed generator (mentioned above), the random main-sequence generator has a different structure from the random main-seed generator. Also, the random main-sequence generator or a randomizer in the random main-sequence generator may be referred as a main-PRBS generator and it may be changed according to the designer's attention.

[902] The random main-sequence generator may generate the aforementioned random FI parameter. That is, random main-sequence generator may generate seed for interleaving cells in a single OFDM symbol.

[903] The random main-sequence generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 4K FFT mode, the random main-sequence generator may include a 1 bit spreader and an 11 bit-randomizer. The random main-sequence generator or the randomizer according to an embodiment of the present invention may be referred as a main-PRBS generator which is defined based on the 11-bit binary word sequence (or binary sequence).

[904] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset of each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for the respective FFT modes. According to an embodiment of the present invention, in the case of 4K FFT mode, a (12-k) bit-randomizer may be used. The (X-k) bits-randomizer according to an embodiment of the present invention may be referred as a sub-PRBS generator which

is defined based on (12-k) bit binary word sequence (or binary sequence).

[905] The aforementioned spreader and randomizer may be used to achieve spreading and random effects during generation of the interleaving seed.

[906] In this embodiment, in generating of interleaving-value, PRBS operation order is modified to cope with the case of that the number of active carriers vary at start and last OFDM symbols within a single frame.

[907] FIG. 84 is a view of a 4K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[908] The 4K FFT mode random interleaving-sequence generator according to an embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a random symbol-offset generator, a modulo operator and a memory-index check. As described above, the random main-sequence generator may include a spreader and a randomizer.

[909] As shown in FIG. 84, the locations of the modulo operator and the memory-index check is changed as compared with the 4K FFT mode random main-seed generator as described above.

[910] The changed locations of the modulo operator and the memory-index check as shown in FIG. 84 is to increase a frequency deinterleaving performance of the frequency deinterleaver having single memory. As above described, a signal frame (or frame) according to the present invention may have normal data symbol (normal data symbol), frame edge symbol and frame signaling symbol and a length of the frame edge symbol and the frame signaling symbol may be shorter than the normal data symbol. For this reason, a frequency deinterleaving performance of the frequency deinterleaver having single memory can be decreased. In order to increase the frequency deinterleaving performance of the frequency deinterleaver with a single memory, the present invention may provide the changed locations of the modulo operator and the memory-index check.

[911] Hereinafter, an operation of each block will be described.

[912] The (cell) spreader may be operated using an upper portion of n-bit of total 12-bit and may function as a multiplexer based on a look-up table. In the case of 4K FFT mode, the (cell) spreader may be a 1-bit multiplexer (or toggling).

[913] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, in the case of 4K FFT mode, the randomizer may be a PN generator that considers 11-bit. This can be changed according to a designer's intention. Also the spreader and the randomizer are operated through multiplexer and PN generator, respectively.

[914] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting main interleaving-sequence generated by the main-interleaving sequence generator for

each pair-wise OFDM symbol. A detailed operation is the same as those describe above and thus are not described here.

[915] The modulo operator may be operated when input value exceeds Ndata or Nmax. The maximum value of the Ndata (Nmax) for 4K FFT mode may be 4096.

[916] The memory-index check may not use output from the modulo operator when a memory-index generated by the spreader and the randomizer is greater than Ndata or the maximum value of the Ndata (Nmax) and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata or the maximum value of the Ndata (Nmax).

[917] Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[918] FIG. 85 illustrates math figures representing an operation of a 4K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[919] The math figures illustrated in an upper portion of FIG. 85 show initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 11th primitive polynomial and the initial value may be changed by arbitrary values. That is, the math figures illustrated in an upper portion shows binary word sequences or binary bits used to define the main-PRBS generator which can generate main-PRBS sequence.

[920] The math figures illustrated in a lower portion of FIG. 85 show procedures of calculating and outputting the interleaving address for different interleaving sequence for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset (or a symbol offset or cyclic shifting value) is used to calculate the different interleaving sequence and the cyclic shifting value may be applied to each OFDM symbol pair in the same way.

[921] As above described, the random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer.

[922] The k bits-spreader may be operated through a 2k multiplexer and may be optimally designed to maximize inter-symbol spreading properties (or to minimize correlation properties).

[923] The randomizer may be operated through an N bits-PN generator (or N bits-sub-PRBS generator) and designed to provide randomness.

[924] The 4K FFT mode random symbol-offset generator may include a 0/1/2 bits-spreader and a 12/11/10 bits-random generator (or a PN generator). It can be changed according to the designer's intention.

[925] FIG. 86 is a view illustrating logical composition of a 4K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[926] As described above, the 4K FFT mode random interleaving-sequence generator according to an embodiment of the present invention may include a random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[927] FIG. 86 illustrates the logical composition of a 4K FFT mode random interleaving-sequence generator formed by combining a random main interleaving-seed generator and a random symbol-offset generator. FIG. 86 illustrates an embodiment of the random main interleaving-seed generator including a 1 bit-spreader and an 11 bits-randomizer, and an embodiment of the random symbol-offset generator including a 2 bits-spreader and a 10 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[928] Hereinafter, a random interleaving-sequence generator for 4K FFT mode according to another embodiment of the present invention will be described.

[929] the random interleaving-sequence generator for 4K FFT mode according to another embodiment of the present invention includes random main interleaving-sequence generator which have a randomizer including bit shuffling.

[930] FIG. 87 is a view illustrating a 4K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[931] The 4K FFT mode random interleaving-sequence generator according to another embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a random symbol-offset generator, a modulo operator and a memory-index check. As described above, the random main interleaving-sequence generator may include a spreader and a randomizer.

[932] Details thereof except bit shuffling have been described above and thus will be omitted here.

[933] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, the randomizer according to an embodiment of the present invention may include bit shuffling. The bit shuffling optimizes spreading properties or random properties and is designed in consideration of Ndata. In the case of 4K FFT mode, the bit shuffling may use a 11-bit PN generator, which can be changed.

[934] FIG. 88 is math figures representing operations of 4K FFT mode bit shuffling and 4K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[935] (a) illustrates a math figure representing an operation of the 4K FFT mode bit shuffling and (b) illustrates a math figure representing an operation of the 4K FFT random interleaving-sequence generator.

[936] The upper portion of (a) shows an operation of the 4K FFT mode bit shuffling and

the lower portion of (a) shows an embodiment of the 4K FFT mode bit shuffling for 11bits.

[937] As illustrated in (a), the 4K FFT mode bit shuffling may mix bits of registers of a PN generator during calculation of a memory-index.

[938] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 12th primitive polynomial and the initial value may be changed by arbitrary values. That is, the math figures illustrated in an upper portion shows binary word sequences or binary bits used to define the main-PRBS generator which can generate main-PRBS sequence.

[939] A math figure illustrated in a lower portion of (b) shows procedures of calculating and outputting the interleaving address for different interleaving sequence for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset (or a symbol offset or cyclic shifting value) is used to calculate the different interleaving sequence and the cyclic shifting value may be applied to each OFDM symbol pair in the same way.

[940] Hereinafter, the random interleaving-sequence generator for an 8K FFT mode will be described. The random interleaving-sequence generator according to an embodiment of the present invention may be included in the frequency interleaver 7020 and is similar to the random seed generator mentioned (mentioned above), the random interleaving-sequence generator has a different structure from the random seed generator.

[941] The random main-sequence generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 8K FFT mode, the random main-sequence generator may include a 1 bit spreader and an 12 bit-randomizer. The random main-sequence generator or the randomizer according to an embodiment of the present invention may be referred as a main-PRBS generator which is defined based on the 12-bit binary word sequence (or binary sequence).

[942] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset of each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for the respective FFT modes. According to an embodiment of the present invention, in the case of 8K FFT mode, a (13-k) bit-randomizer may be used. The (X-k) bits-randomizer according to an embodiment of the present invention may be referred as a sub-PRBS generator which

is defined based on (13-k) bit binary word sequence (or binary sequence).

[943] The aforementioned spreader and randomizer may be used to achieve spreading and random effects during generation of the interleaving seed.

[944] In this embodiment, in generating of interleaving-value, PRBS operation order is modified to cope with the case of that the number of active carriers vary at start and last OFDM symbols within a single frame.

[945] FIG. 89 is a view of an8K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[946] The 8K FFT mode random interleaving-sequence generator according to an embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a random symbol-offset generator, a modulo operator and a memory-index check. As described above, the random main-sequence generator may include a spreader and a randomizer.

[947] As shown in FIG. 89, the locations of the modulo operator and the memory-index check is changed as compared with the 8K FFT mode random main-seed generator as described above.

[948] The changed locations of the modulo operator and the memory-index check as shown in FIG. 89 is to increase a frequency deinterleaving performance of the frequency deinterleaver having single memory. As above described, a signal frame (or frame) according to the present invention may have normal data symbol (normal data symbol), frame edge symbol and frame signaling symbol and a length of the frame edge symbol and the frame signaling symbol may be shorter than the normal data symbol. For this reason, a frequency deinterleaving performance of the frequency deinterleaver having single memory can be decreased. In order to increase the frequency deinterleaving performance of the frequency deinterleaver with a single memory, the present invention may provide the changed locations of the modulo operator and the memory-index check.

[949] Hereinafter, an operation of each block will be described.

[950] The (cell) spreader may be operated using an upper portion of n-bit of total 13-bit and may function as a multiplexer based on a look-up table. In the case of 8K FFT mode, the (cell) spreader may be a 1-bit multiplexer (or toggling).

[951] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, in the case of 8K FFT mode, the randomizer may be a PN generator that considers 12-bit. This can be changed according to a designer's intention. Also the spreader and the randomizer are operated through multiplexer and PN generator, respectively.

[952] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting main interleaving-sequence generated by the main-interleaving sequence generator for

each pair-wise OFDM symbol. A detailed operation is the same as those describe above and thus are not described here.

[953] The modulo operator may be operated when input value exceeds Ndata or Nmax. The maximum value of the Ndata (Nmax) for 8K FFT mode may be 8192.

[954] The memory-index check may not use output from the modulo operator when a memory-index generated by the spreader and the randomizer is greater than Ndata or the maximum value of the Ndata (Nmax) and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata or the maximum value of the Ndata (Nmax).

[955] Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[956] FIG. 90 illustrates math figures representing an operation of an8K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[957] The math figures illustrated in an upper portion of FIG. 90 show initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 12th primitive polynomial and the initial value may be changed by arbitrary values. That is, the math figures illustrated in an upper portion shows binary word sequences or binary bits used to define the main-PRBS generator which can generate main-PRBS sequence.

[958] The math figures illustrated in a lower portion of FIG. 90 show procedures of calculating and outputting the interleaving address for different interleaving sequence for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset (or a symbol offset or cyclic shifting value) is used to calculate the different interleaving sequence and the cyclic shifting value may be applied to each OFDM symbol pair in the same way.

[959] As above described, the random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer.

[960] The k bits-spreader may be operated through a 2k multiplexer and may be optimally designed to maximize inter-symbol spreading properties (or to minimize correlation properties).

[961] The randomizer may be operated through a N bits-PN generator or N bits-sub-PRBS generator) and designed to provide randomness.

[962] The 8K FFT mode random symbol-offset generator may include a 0/1/2 bits-spreader and a 13/12/11 bits-random generator (or a PN generator). It can be changed according to the designer's intention.

[963] FIG. 91 is a view illustrating logical composition of an 8K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[964] As described above, the 8K FFT mode random interleaving-sequence generator according to an embodiment of the present invention may include a random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[965] FIG. 91 illustrates the logical composition of an 8K FFT mode random interleaving-sequence generator formed by combining a random main interleaving-seed generator and a random symbol-offset generator. FIG. 91 illustrates an embodiment of the random main interleaving-seed generator including a 1 bit-spreader and a 12 bits-randomizer, and an embodiment of the random symbol-offset generator including a 2 bits-spreader and an 11 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[966] Hereinafter, a random interleaving-sequence generator for 8K FFT mode according to another embodiment of the present invention will be described.

[967] The random interleaving-sequence generator for 8K FFT mode according to another embodiment of the present invention includes random main interleaving-sequence generator which have a randomizer including bit shuffling.

[968] FIG. 92 is a view illustrating an 8K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[969] The 8K FFT mode random interleaving-sequence generator according to another embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a random symbol-offset generator, a modulo operator and a memory-index check. As described above, the random main interleaving-sequence generator may include a spreader and a randomizer.

[970] Details thereof except bit shuffling have been described above and thus will be omitted here.

[971] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, the randomizer according to an embodiment of the present invention may include bit shuffling. The bit shuffling optimizes spreading properties or random properties and is designed in consideration of Ndata. In the case of 8K FFT mode, the bit shuffling may use a 12-bit PN generator, which can be changed.

[972] FIG. 93 is math figures representing operations of 8K FFT mode bit shuffling and 8K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[973] (a) illustrates a math figure representing an operation of the 8K FFT mode bit shuffling and (b) illustrates a math figure representing an operation of the 8K FFT random interleaving-sequence generator.

[974] The upper portion of (a) shows an operation of the 8K FFT mode bit shuffling and

the lower portion of (a) shows an embodiment of the 8K FFT mode bit shuffling for 12bits.

[975] As illustrated in (a), the 8K FFT mode bit shuffling may mix bits of registers of a PN generator during calculation of a memory-index.

[976] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 12th primitive polynomial and the initial value may be changed by arbitrary values. That is, the math figures illustrated in an upper portion shows binary word sequences or binary bits used to define the main-PRBS generator which can generate main-PRBS sequence.

[977] A math figure illustrated in a lower portion of (b) shows procedures of calculating and outputting the interleaving address for different interleaving sequence for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset (or a symbol offset or cyclic shifting value) is used to calculate the different interleaving sequence and the cyclic shifting value may be applied to each OFDM symbol pair in the same way.

[978] Hereinafter, the random interleaving-sequence generator for a 16K FFT mode will be described. The random interleaving-sequence generator according to an embodiment of the present invention may be included in the frequency interleaver 7020 and is similar to the random seed generator mentioned (mentioned above), the random interleaving-sequence generator has a different structure from the random seed generator.

[979] The random main-sequence generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 16K FFT mode, the random main-sequence generator may include a 1 bit spreader and an 13 bit-randomizer. The random main-sequence generator or the randomizer according to an embodiment of the present invention may be referred as a main-PRBS generator which is defined based on the 13-bit binary word sequence (or binary sequence).

[980] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset of each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for the respective FFT modes. According to an embodiment of the present invention, in the case of 16K FFT mode, a (14-k) bit-randomizer may be used. The (X-k) bits-randomizer according to an embodiment of the present invention may be referred as a sub-PRBS generator which

is defined based on (14-k) bit binary word sequence (or binary sequence).

[981] The aforementioned spreader and randomizer may be used to achieve spreading and random effects during generation of the interleaving seed.

[982] In this embodiment, in generating of interleaving-value, PRBS operation order is modified to cope with the case of that the number of active carriers vary at start and last OFDM symbols within a single frame.

[983] FIG. 94 is a view of a 16K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[984] The 16K FFT mode random interleaving-sequence generator according to an embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a random symbol-offset generator, a modulo operator and a memory-index check. As described above, the random main-sequence generator may include a spreader and a randomizer.

[985] As shown in FIG. 94, the locations of the modulo operator and the memory-index check is changed as compared with the 16K FFT mode random main-seed generator as described above.

[986] The changed locations of the modulo operator and the memory-index check as shown in FIG. 94 is to increase a frequency deinterleaving performance of the frequency deinterleaver having single memory. As above described, a signal frame (or frame) according to the present invention may have normal data symbol (normal data symbol), frame edge symbol and frame signaling symbol and a length of the frame edge symbol and the frame signaling symbol may be shorter than the normal data symbol. For this reason, a frequency deinterleaving performance of the frequency deinterleaver having single memory can be decreased. In order to increase the frequency deinterleaving performance of the frequency deinterleaver with a single memory, the present invention may provide the changed locations of the modulo operator and the memory-index check.

[987] Hereinafter, an operation of each block will be described.

[988] The (cell) spreader may be operated using an upper portion of n-bit of total 14-bit and may function as a multiplexer based on a look-up table. In the case of 16K FFT mode, the (cell) spreader may be a 1-bit multiplexer (or toggling).

[989] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, in the case of 16K FFT mode, the randomizer may be a PN generator that considers 13-bit. This can be changed according to a designer's intention. Also the spreader and the randomizer are operated through multiplexer and PN generator, respectively.

[990] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting main interleaving-sequence generated by the main-interleaving sequence generator for

each pair-wise OFDM symbol. A detailed operation is the same as those describe above and thus are not described here.

[991] The modulo operator may be operated when input value exceeds Ndata or Nmax. The maximum value of the Ndata (Nmax) for 16K FFT mode may be 16384.

[992] The memory-index check may not use output from the modulo operator when a memory-index generated by the spreader and the randomizer is greater than Ndata or the maximum value of the Ndata (Nmax) and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata or the maximum value of the Ndata (Nmax).

[993] Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[994] FIG. 95 illustrates math figures representing an operation of a 16K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[995] The math figures illustrated in an upper portion of FIG. 95 show initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 13th primitive polynomial and the initial value may be changed by arbitrary values. That is, the math figures illustrated in an upper portion shows binary word sequences or binary bits used to define the main-PRBS generator which can generate main-PRBS sequence.

[996] The math figures illustrated in a lower portion of FIG. 95 show procedures of calculating and outputting the interleaving address for different interleaving sequence for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset (or a symbol offset or cyclic shifting value) is used to calculate the different interleaving sequence and the cyclic shifting value may be applied to each OFDM symbol pair in the same way.

[997] As above described, the random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer.

[998] The k bits-spreader may be operated through a 2k multiplexer and may be optimally designed to maximize inter-symbol spreading properties (or to minimize correlation properties).

[999] The randomizer may be operated through an N bits-PN generator (or N bits-sub-PRBS generator) and designed to provide randomness.

[1000] The 16K FFT mode random symbol-offset generator may include a 0/1/2 bits-spreader and a 14/13/12 bits-random generator (or a PN generator). It can be changed according to the designer's intention.

[1001] FIG. 96 is a view illustrating logical composition of a 16K FFT mode random in-

terleaving-sequence generator according to an embodiment of the present invention.

[1002] As described above, the 16K FFT mode random interleaving-sequence generator according to an embodiment of the present invention may include a random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[1003] FIG. 96 illustrates the logical composition of a 16K FFT mode random interleaving-sequence generator formed by combining a random main interleaving-seed generator and a random symbol-offset generator. FIG. 96 illustrates an embodiment of the random main interleaving-seed generator including a 1 bit-spreader and a 13 bits-randomizer, and an embodiment of the random symbol-offset generator including a 2 bits-spreader and a 12 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[1004] Hereinafter, a random interleaving-sequence generator for 16K FFT mode according to another embodiment of the present invention will be described.

[1005] the random interleaving-sequence generator for 16K FFT mode according to another embodiment of the present invention includes random main interleaving-sequence generator which have a randomizer including bit shuffling.

[1006] FIG. 97 is a view illustrating a 16K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[1007] The 16K FFT mode random interleaving-sequence generator according to another embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a random symbol-offset generator, a modulo operator and a memory-index check. As described above, the random main interleaving-sequence generator may include a spreader and a randomizer.

[1008] Details thereof except bit shuffling have been described above and thus will be omitted here.

[1009] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, the randomizer according to an embodiment of the present invention may include bit shuffling. The bit shuffling optimizes spreading properties or random properties and is designed in consideration of Ndata. In the case of 16K FFT mode, the bit shuffling may use a 13-bit PN generator, which can be changed.

[1010] FIG. 98 is math figures representing operations of 16K FFT mode bit shuffling and 16K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[1011] (a) illustrates a math figure representing an operation of the 16K FFT mode bit shuffling and (b) illustrates a math figure representing an operation of the 16K FFT random interleaving-sequence generator.

[1012] The upper portion of (a) shows an operation of the 16K FFT mode bit shuffling and the lower portion of (a) shows an embodiment of the 16K FFT mode bit shuffling for 13bits.

[1013] As illustrated in (a), the 16K FFT mode bit shuffling may mix bits of registers of a PN generator during calculation of a memory-index.

[1014] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 13th primitive polynomial and the initial value may be changed by arbitrary values. That is, the math figures illustrated in an upper portion shows binary word sequences or binary bits used to define the main-PRBS generator which can generate main-PRBS sequence.

[1015] A math figure illustrated in a lower portion of (b) shows procedures of calculating and outputting the interleaving address for different interleaving sequence for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset (or a symbol offset or cyclic shifting value) is used to calculate the different interleaving sequence and the cyclic shifting value may be applied to each OFDM symbol pair in the same way.

[1016] Hereinafter, the random interleaving-sequence generator for a 32K FFT mode will be described. The random interleaving-sequence generator according to an embodiment of the present invention may be included in the frequency interleaver 7020 and is similar to the random seed generator mentioned (mentioned above), the random interleaving-sequence generator has a different structure from the random seed generator.

[1017] The random main-sequence generator according to an embodiment of the present invention may include a spreader and a randomizer and perform rendering a full randomness in frequency-domain. According to an embodiment of the present invention, in the case of 32K FFT mode, the random main-sequence generator may include a 1 bit spreader and a 14 bit-randomizer. The random main-sequence generator or the randomizer according to an embodiment of the present invention may be referred as a main-PRBS generator which is defined based on the 14-bit binary word sequence (or binary sequence).

[1018] The random symbol-offset generator according to an embodiment of the present invention may change a symbol offset of each OFDM symbol. That is, the random symbol-offset generator may generate the aforementioned symbol offset. The random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer and perform rendering a spreading as much as 2k cases, in time-domain. X may be differently set for the respective FFT modes. According to an embodiment of the present invention, in the case of 32K FFT mode, a (15-k) bit-randomizer may be used. The (X-k) bits-randomizer according to an

embodiment of the present invention may be referred as a sub-PRBS generator which is defined based on (15-k) bit binary word sequence (or binary sequence).

[1019] The aforementioned spreader and randomizer may be used to achieve spreading and random effects during generation of the interleaving seed.

[1020] In this embodiment, in generating of interleaving-value, PRBS operation order is modified to cope with the case of that the number of active carriers vary at start and last OFDM symbols within a single frame.

[1021] FIG. 99 is a view of a 32K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[1022] The 32K FFT mode random interleaving-sequence generator according to an embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a random symbol-offset generator, a modulo operator and a memory-index check. As described above, the random main-sequence generator may include a spreader and a randomizer.

[1023] As shown in FIG. 99, the locations of the modulo operator and the memory-index check is changed as compared with the 32K FFT mode random main-seed generator as described above.

[1024] The changed locations of the modulo operator and the memory-index check as shown in FIG. 99 is to increase a frequency deinterleaving performance of the frequency deinterleaver having single memory. As above described, a signal frame (or frame) according to the present invention may have normal data symbol (normal data symbol), frame edge symbol and frame signaling symbol and a length of the frame edge symbol and the frame signaling symbol may be shorter than the normal data symbol. For this reason, a frequency deinterleaving performance of the frequency deinterleaver having single memory can be decreased. In order to increase the frequency deinterleaving performance of the frequency deinterleaver with a single memory, the present invention may provide the changed locations of the modulo operator and the memory-index check.

[1025] Hereinafter, an operation of each block will be described.

[1026] The (cell) spreader may be operated using an upper portion of n-bit of total 15-bit and may function as a multiplexer based on a look-up table. In the case of 32K FFT mode, the (cell) spreader may be a 1-bit multiplexer (or toggling).

[1027] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, in the case of 32K FFT mode, the randomizer may be a PN generator that considers 14-bit. This can be changed according to a designer's intention. Also the spreader and the randomizer are operated through multiplexer and PN generator, respectively.

[1028] The random symbol-offset generator may generate a symbol-offset for cyclic-shifting

main interleaving-sequence generated by the main-interleaving sequence generator for each pair-wise OFDM symbol. A detailed operation is the same as those describe above and thus are not described here.

[1029] The modulo operator may be operated when input value exceeds Ndata or Nmax. The maximum value of the Ndata (Nmax) for 32K FFT mode may be 32768.

[1030] The memory-index check may not use output from the modulo operator when a memory-index generated by the spreader and the randomizer is greater than Ndata or the maximum value of the Ndata (Nmax) and may repeatedly operate the spreader and the randomizer to adjust the output memory-index such that the output memory-index does not exceed Ndata or the maximum value of the Ndata (Nmax).

[1031] Locations of the illustrated memory-index check and modulo operator can be changed according to a designer's intention.

[1032] FIG. 100 illustrates math figures representing an operation of a 32K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[1033] The math figures illustrated in an upper portion of FIG. 100 show initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 14th primitive polynomial and the initial value may be changed by arbitrary values. That is, the math figures illustrated in an upper portion shows binary word sequences or binary bits used to define the main-PRBS generator which can generate main-PRBS sequence.

[1034] The math figures illustrated in a lower portion of FIG. 100 show procedures of calculating and outputting the interleaving address for different interleaving sequence for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset (or a symbol offset or cyclic shifting value) is used to calculate the different interleaving sequence and the cyclic shifting value may be applied to each OFDM symbol pair in the same way.

[1035] As above described, the random symbol-offset generator according to an embodiment of the present invention may include k bits-spreader and (X-k) bits-randomizer.

[1036] The k bits-spreader may be operated through a 2k multiplexer and may be optimally designed to maximize inter-symbol spreading properties (or to minimize correlation properties).

[1037] The randomizer may be operated through a N bits-PN generator (or N bits-sub-PRBS generator) and designed to provide randomness.

[1038] The 32K FFT mode random symbol-offset generator may include a 0/1/2 bits-spreader and a 15/14/13 bits-random generator (or a PN generator). It can be changed according to the designer's intention.

[1039] FIG. 101 is a view illustrating logical composition of a 32K FFT mode random interleaving-sequence generator according to an embodiment of the present invention.

[1040] As described above, the 32K FFT mode random interleaving-sequence generator according to an embodiment of the present invention may include a random main interleaving-seed generator, a random symbol-offset generator, a memory index check, and a modulo operator.

[1041] FIG. 101 illustrates the logical composition of a 32KFFT mode random interleaving-sequence generator formed by combining a random main interleaving-seed generator and a random symbol-offset generator. FIG. 101 illustrates an embodiment of the random main interleaving-seed generator including a 1 bit-spreader and a 14 bits-randomizer, and an embodiment of the random symbol-offset generator including a 2 bits-spreader and an 13 bits-randomizer. Details thereof have been described above and thus will be omitted here.

[1042] Hereinafter, a random interleaving-sequence generator for 32K FFT mode according to another embodiment of the present invention will be described.

[1043] the random interleaving-sequence generator for 32K FFT mode according to another embodiment of the present invention includes random main interleaving-sequence generator which have a randomizer including bit shuffling.

[1044] FIG. 102 is a view illustrating a 32KFFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[1045] The 32K FFT mode random interleaving-sequence generator according to another embodiment of the present invention may include a spreader (1-bit toggling), a randomizer, a random symbol-offset generator, a modulo operator and a memory-index check. As described above, the random main interleaving-sequence generator may include a spreader and a randomizer.

[1046] Details thereof except bit shuffling have been described above and thus will be omitted here.

[1047] The randomizer may be operated via a PN generator and may provide full randomness during interleaving. As described above, the randomizer according to an embodiment of the present invention may include bit shuffling. The bit shuffling optimizes spreading properties or random properties and is designed in consideration of Ndata. In the case of 8K FFT mode, the bit shuffling may use a 14-bit PN generator, which can be changed.

[1048] FIG. 103 is math figures representing operations of 32K FFT mode bit shuffling and 32K FFT mode random interleaving-sequence generator according to another embodiment of the present invention.

[1049] (a) illustrates a math figure representing an operation of the 32K FFT mode bit shuffling and (b) illustrates a math figure representing an operation of the 32K FFT

random interleaving-sequence generator.

[1050] The upper portion of (a) shows an operation of the 32K FFT mode bit shuffling and the lower portion of (a) shows an embodiment of the 32K FFT mode bit shuffling for 14bits.

[1051] As illustrated in (a), the 32K FFT mode bit shuffling may mix bits of registers of a PN generator during calculation of a memory-index.

[1052] A math figure illustrated in an upper portion of (b) shows initial value setting and primitive polynomial of a randomizer. In this case, the primitive polynomial may be 14th primitive polynomial and the initial value may be changed by arbitrary values. That is, the math figures illustrated in an upper portion shows binary word sequences or binary bits used to define the main-PRBS generator which can generate main-PRBS sequence.

[1053] A math figure illustrated in a lower portion of (b) shows procedures of calculating and outputting the interleaving address for different interleaving sequence for an output signal of the spreader and the randomizer. As illustrated in the math figure, one random symbol-offset (or a symbol offset or cyclic shifting value) is used to calculate the different interleaving sequence and the cyclic shifting value may be applied to each OFDM symbol pair in the same way.

[1054] FIG. 104 is a flowchart illustrating a method for transmitting broadcast signals according to an embodiment of the present invention.

[1055] The apparatus for transmitting broadcast signals according to an embodiment of the present invention can encode service data (S104000). As described above, service data is transmitted through a data pipe which is a logical channel in the physical layer that carries service data or related metadata, which may carry one or multiple service(s) or service component(s). Data carried on a data pipe can be referred to as the DP data or the service data. The detailed process of step S104000 is as described in FIG. 1, FIG. 5-6 and FIG. 22.

[1056] The apparatus for transmitting broadcast signals according to an embodiment of the present invention can build at least one signal frame including the encoded service data (S104010). The detailed process of this step is as described in FIG. 7 and FIG. 10-21.

[1057] Then, the apparatus for transmitting broadcast signals according to an embodiment of the present invention can map the encoded service data into OFDM symbols of the signal frame and frequency interleave the mapped service data. As above described, the basic function of the cell mapper 7010 is to map data cells for each of the DPs, PLS data, if any, into arrays of active OFDM cells corresponding to each of the OFDM symbols within a signal frame. Then, the frequency interleaver 7020 may operate on a single OFDM symbol basis, provide frequency diversity by randomly interleaving the cells received from the cell mapper 7010. The purpose of the frequency interleaver

7020 in the present invention, which operates on a single OFDM symbol, is to provide frequency diversity by randomly interleaving data cells received from the cell mapper 7010. In order to get maximum interleaving gain in a single signal frame (or frame), a different interleaving-seed is used for every OFDM symbol pair comprised of two sequential OFDM symbols. The detailed process of the frequency interleaving is as described in FIG. 30 to 103.

[1058] Subsequently, the apparatus for transmitting broadcast signals according to an embodiment of the present invention can modulate data in the built at least one signal frame by an OFDM (Orthogonal Frequency Division Multiplexing) scheme (S104020). The detailed process of this step is as described in FIG. 1 or 8.

[1059] The apparatus for transmitting broadcast signals according to an embodiment of the present invention can transmit the broadcast signals including the modulated data (S104030). The detailed process of this step is as described in FIG. 1 or 8.

[1060] FIG. 105 is a flowchart illustrating a method for receiving broadcast signals according to an embodiment of the present invention.

[1061] The flowchart shown in FIG. 105 corresponds to a reverse process of the broadcast signal transmission method according to an embodiment of the present invention, described with reference to FIG. 104.

[1062] The apparatus for receiving broadcast signals according to an embodiment of the present invention can receive broadcast signals (S105000).

[1063] The apparatus for receiving broadcast signals according to an embodiment of the present invention can demodulate the received broadcast signals using an OFDM (Othogonal Frequency Division Multiplexing) scheme (S105010). Details are as described in FIG. 9.

[1064] The apparatus for receiving broadcast signals according to an embodiment of the present invention can parse at least one signal frame from the demodulated broadcast signals (S105020). Details are as described in FIG. 9. In this case, the apparatus for receiving broadcast signals according to an embodiment of the present invention can perform frequency de-interleaving corresponds to a reverse process of the frequency interleaving as shown in the above. The detailed process of the frequency interleaving is as described in FIG. 30 to 103.

[1065] Subsequently, the apparatus for receiving broadcast signals according to an embodiment of the present invention can decode data in the parsed at least one signal frame to output service data (S105030). Details are as described in FIG. 9.

[1066] As described above, service data is transmitted through a data pipe which is a logical channel in the physical layer that carries service data or related metadata, which may carry one or multiple service(s) or service component(s). Data carried on a data pipe can be referred to as the DP data or the service data.

[1067] It will be appreciated by those skilled in the art that various modifications and variations can be made in the present invention without departing from the spirit or scope of the inventions. Thus, it is intended that the present invention covers the modifications and variations of this invention provided they come within the scope of the appended claims and their equivalents.

[1068] Both apparatus and method inventions are mentioned in this specification and descriptions of both of the apparatus and method inventions may be complementarily applicable to each other.

Mode for the Invention

[1069] Various embodiments have been described in the best mode for carrying out the invention.

Industrial Applicability

[1070] The present invention is available in a series of broadcast signal provision fields.

[1071] It will be apparent to those skilled in the art that various modifications and variations can be made in the present invention without departing from the spirit or scope of the inventions. Thus, it is intended that the present invention covers the modifications and variations of this invention provided they come within the scope of the appended claims and their equivalents.

Claims

[Claim 1] A method for transmitting broadcast signals, the method comprising:
encoding service data;
building at least one signal frame including the encoded service data, wherein the at least one signal frame includes a plurality of OFDM symbols;
modulating data in the built at least one signal frame by an OFDM (Orthogonal Frequency Division Multiplex) scheme; and
transmitting the broadcast signals having the modulated data.

[Claim 2] The method of claim 1, wherein the method further includes:
frequency interleaving data in the at least one signal frame by using a different interleaving-seed which is used for every OFDM symbol pair comprised of two sequential OFDM symbols, wherein the different interleaving-seed is generated based on a cyclic shifting value, wherein an interleaving seed is variable based on an FFT size of the modulating.

[Claim 3] An apparatus for transmitting broadcast signals, the apparatus comprising:
an encoder for encoding service data;
a mapper for mapping the encoded service data into a plurality of OFDM symbols to build at least one signal frame;
a modulator for modulating data in the built at least one signal frame by an OFDM (Orthogonal Frequency Division Multiplex) scheme; and
a transmitter for transmitting the broadcast signals having the modulated data.

[Claim 4] The apparatus of claim 7, wherein a frequency interleaver for frequency interleaving data in the at least one signal frame by using a different interleaving-seed which is used for every OFDM symbol pair comprised of two sequential OFDM symbols, wherein the interleaving-seed is generated based on a cyclic shifting value and an FFT size of the modulating.

[Claim 5] A method for receiving broadcast signals, the method comprising:
receiving the broadcast signals;
demodulating the received broadcast signals by an OFDM (Orthogonal Frequency Division Multiplex) scheme;
parsing at least one signal frame from the demodulated broadcast signals; and
decoding data in the parsed at least one signal frame to output service

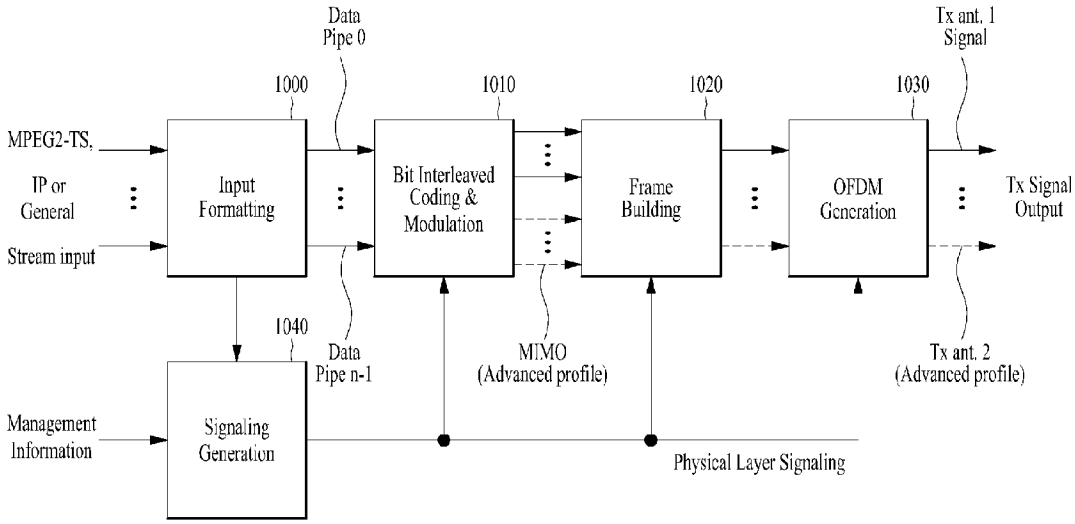
data.

[Claim 6] The method of claim 5, wherein the method includes:
frequency de-interleaving the demodulated broadcast signals, wherein the broadcast signals are frequency interleaved by using a different interleaving-seed which is used for every OFDM symbol pair comprised of two sequential OFDM symbols, wherein the different interleaving-seed is generated a cyclic shifting value and, wherein an interleaving seed is variable based on an FFT size of the demodulating.

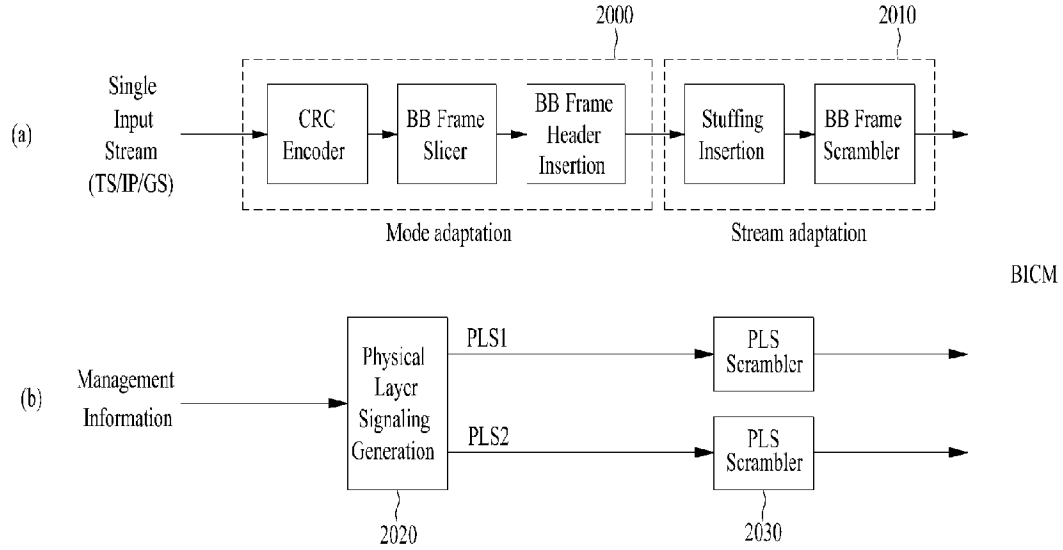
[Claim 7] An apparatus for receiving broadcast signals, the apparatus comprising:
a receiver for receiving the broadcast signals;
a demodulator for demodulating the received broadcast signals by an OFDM (Orthogonal Frequency Division Multiplex) scheme;
a demapper for demapping service data from at least one signal frame in the demodulated broadcast signals; and
a decoder for decoding the service data in the at least one signal frame.

[Claim 8] The apparatus of claim 7, the apparatus further include;
a frequency de-interleaver for frequency de-interleaving the demodulated broadcast signals wherein the broadcast signals are frequency interleaved by using a different interleaving-seed which is used for every OFDM symbol pair comprised of two sequential OFDM symbols, wherein the different interleaving-seed is based on a cyclic shifting value and, wherein an interleaving seed is variable based on an FFT size of the demodulating.

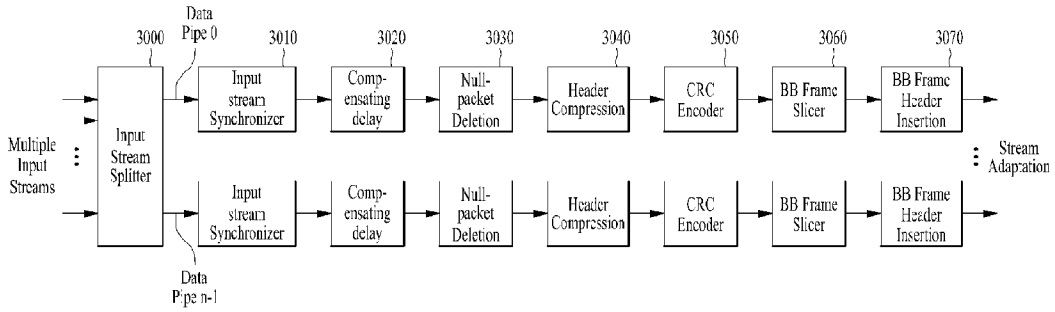
[Fig. 1]



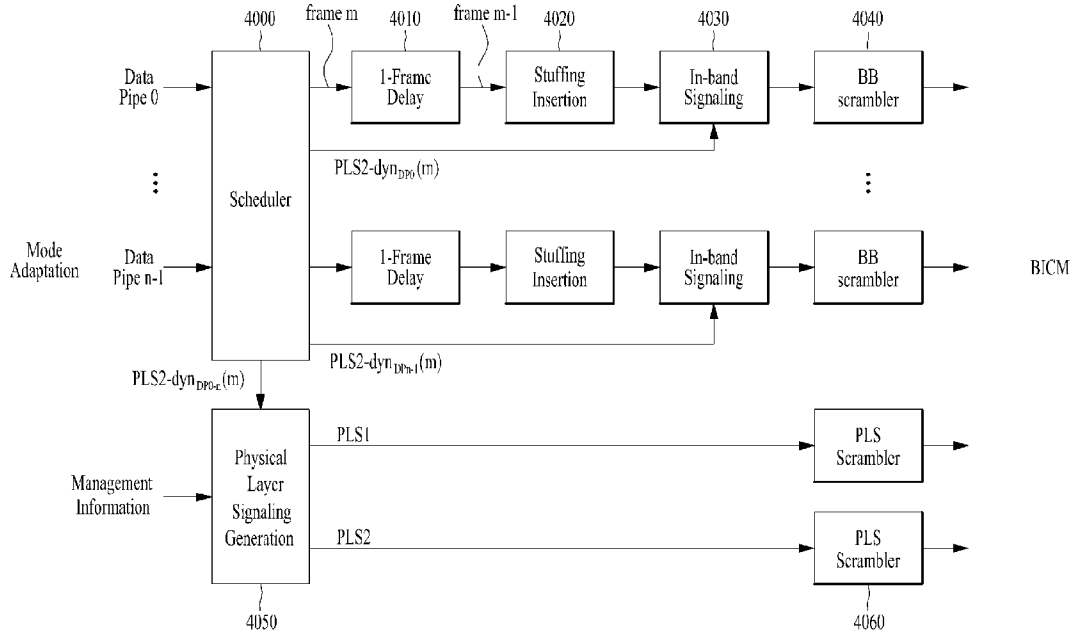
[Fig. 2]



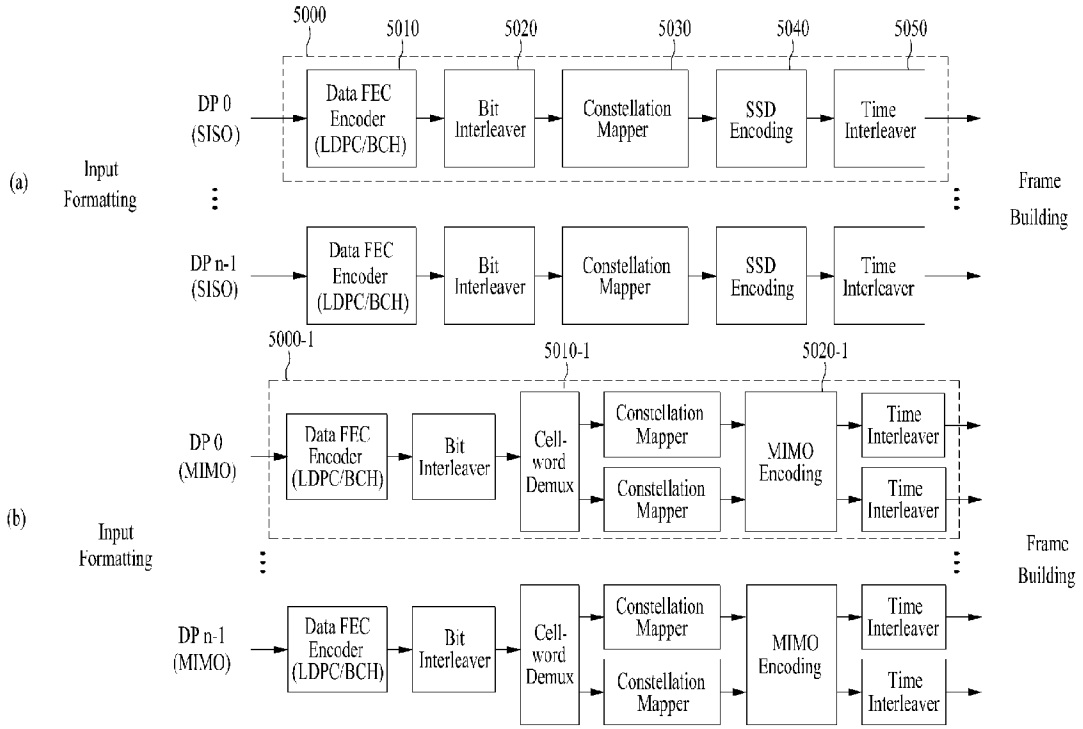
[Fig. 3]



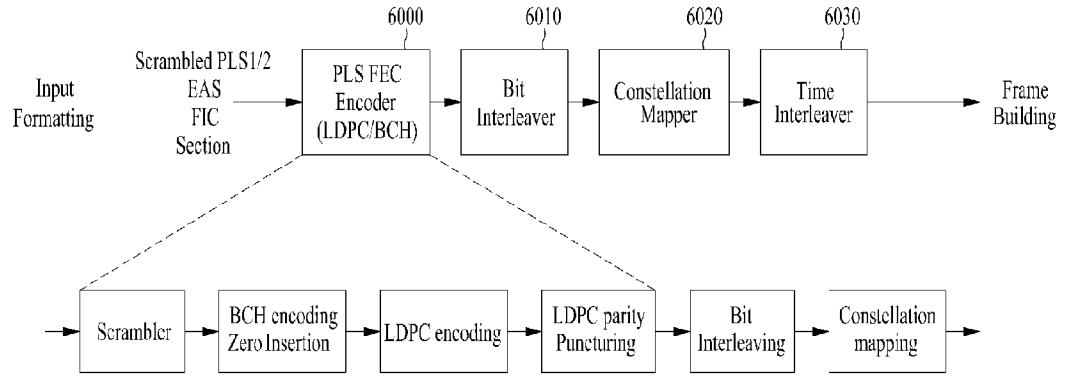
[Fig. 4]



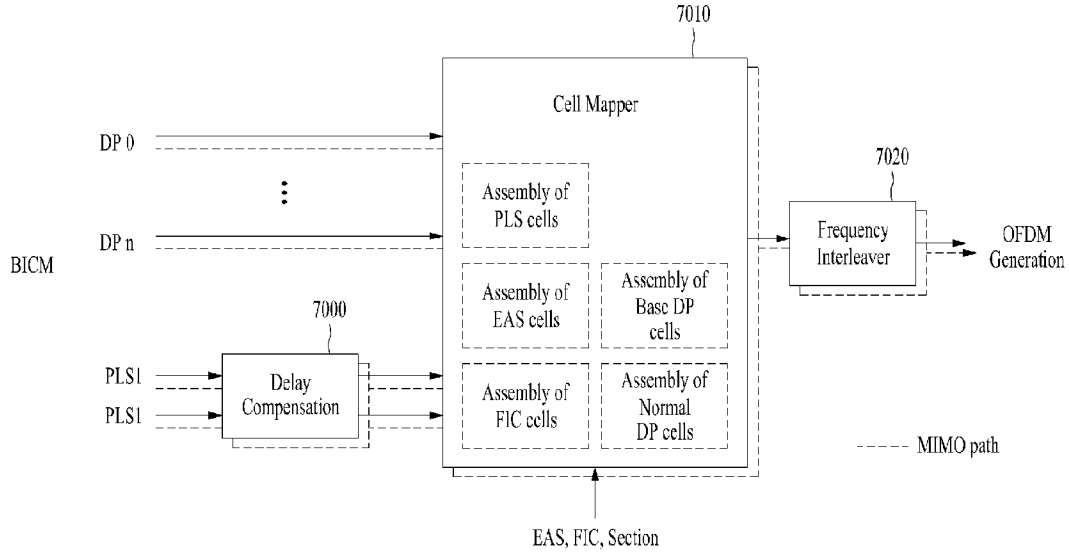
[Fig. 5]



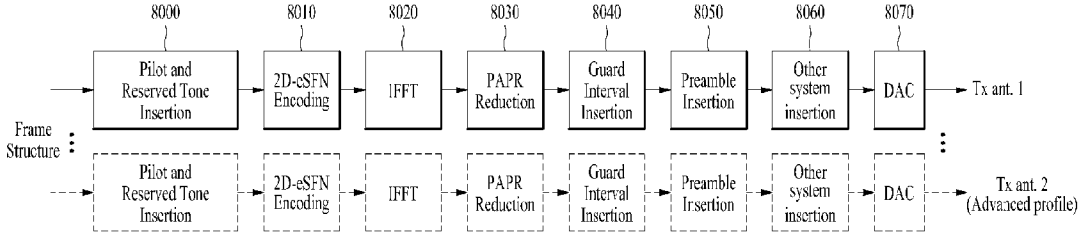
[Fig. 6]



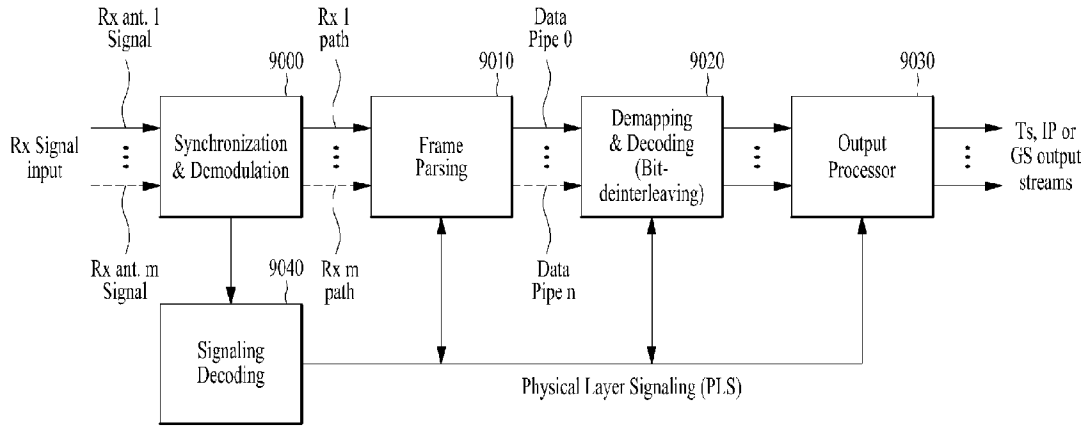
[Fig. 7]



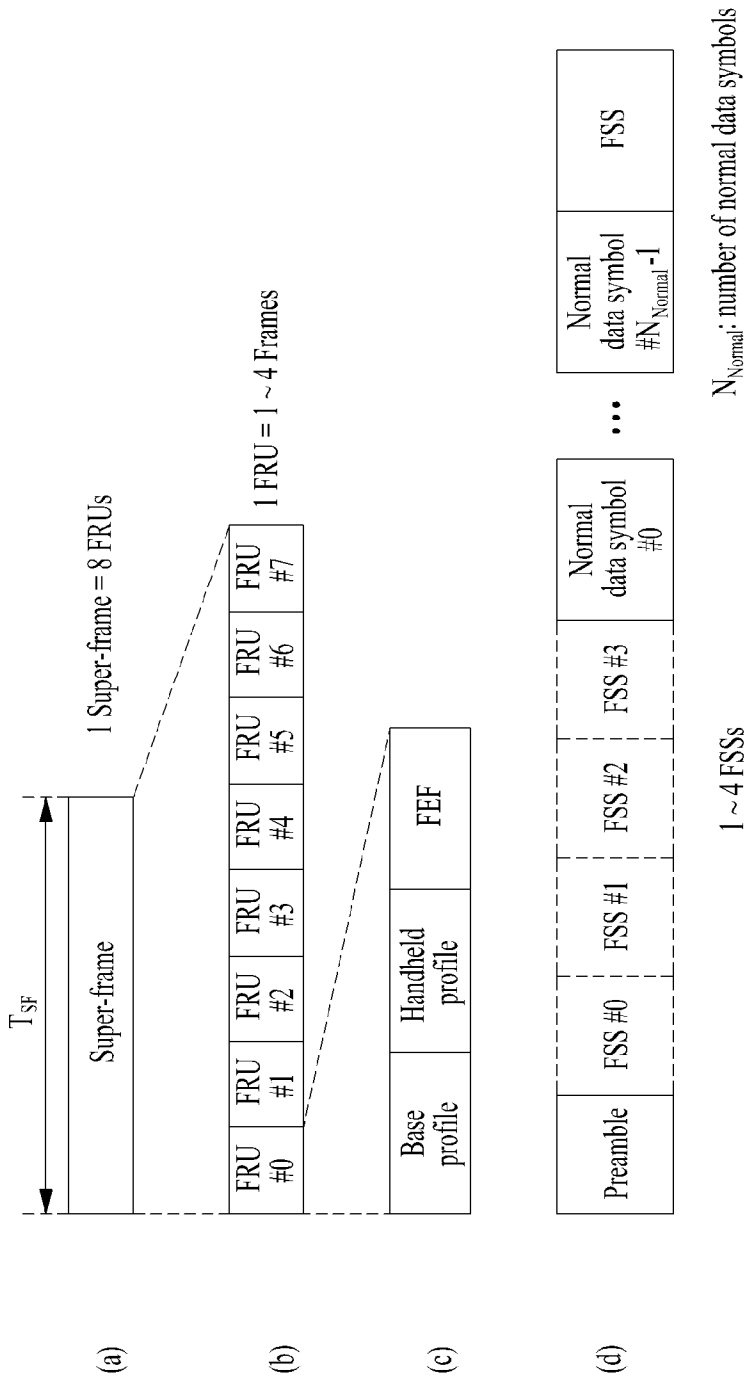
[Fig. 8]



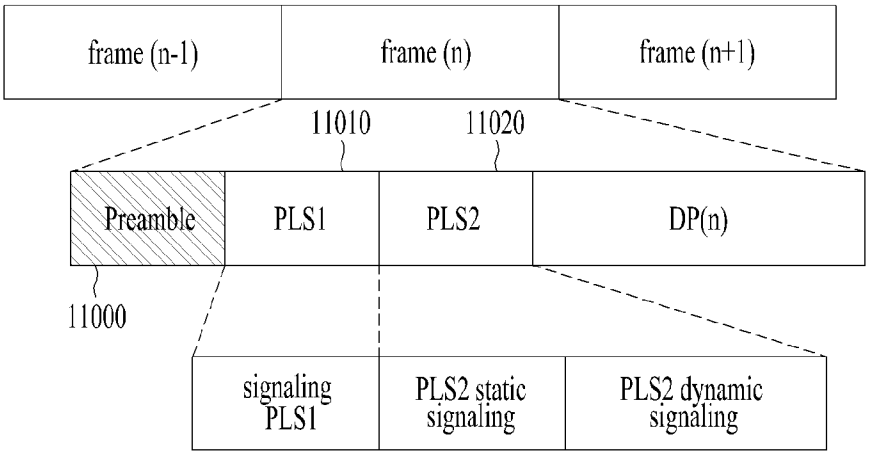
[Fig. 9]



[Fig. 10]


(a)
T_{SF}
Super-frame
1 Super-frame = 8 FRUs
(b)
FRU #0
FRU #1
FRU #2
FRU #3
FRU #4
FRU #5
FRU #6
FRU #7
1 FRU = 1 ~ 4 Frames
(c)
Base profile
Handheld profile
FEF
(d)
Preamble
FSS #0
FSS #1
FSS #2
FSS #3
Normal data symbol #0
Normal data symbol #N_{Normal}-1
FSS
1 ~ 4 FSSs
N_{Normal}: number of normal data symbols

[Fig. 11]



[Fig. 12]

Content	Bits
PHY_PROFILE	3
FFT_SIZE	2
GI_FRACTION	3
EAC_FLAG	1
PILOT_MODE	1
PAPR_FLAG	1
FRU_CONFIGURE	3
RESERVED	7

[Fig. 13]

Content	Bits
PREAMBLE_DATA	20
NUM_FRAME_FRU	2
PAYLOAD_TYPE	3
NUM_FSS	2
SYSTEM_VERSION	8
CELL_ID	16
NETWORK_ID	16
SYSTEM_ID	16
for i = 0:3	
FRU_PHY_PROFILE	3
FRU_FRAME_LENGTH	2
FRU_GI_FRACTION	3
RESERVED	4
end	
PLS2_FEC_TYPE	2
PLS2_MOD	3
PLS2_SIZE_CELL	15
PLS2_STAT_SIZE_BIT	14
PLS2_SYN_SIZE_BIT	14
PLS2_REP_FLAG	1
PLS2_REP_SIZE_CELL	15
PLS2_NEXT_FEC_TYPE	2
PLS2_NEXT_MODE	3
PLS2_NEXT_REP_FLAG	1
PLS2_NEXT_REP_SIZE_CELL	15
PLS2_NEXT_REP_STAT_SIZE_BIT	14
PLS2_NEXT_REP_DYN_SIZE_BIT	14
PLS2_AP_MODE	2
PLS2_AP_SIZE_CELL	15
PLS2_NEXT_AP_MODE	2
PLS2_NEXT_AP_SIZE_CELL	15
RESERVED	32
CRC 32	32

[Fig. 14]

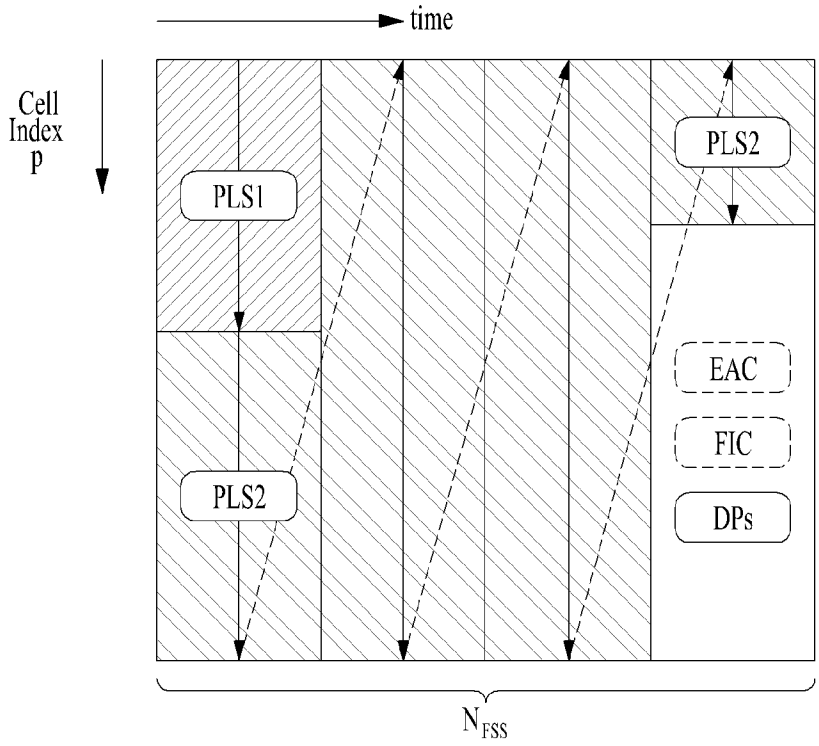
Content	Bits
FIC_FLAG	1
AUX_FLAG	1
NUM_DP	6
for i = 1: NUM_DP	
DP_ID	6
DP_TYPE	3
DP_GROUP_ID	8
BASE_DP_ID	6
DP_FEC_TYPE	2
DP_COD	4
DP_MOD	4
DP_SSD_FLAG	1
if PHY_PROFILE = '010'	
DP_MIMO	3
end	
DP_TI_TYPE	1
DP_TI_LENGTH	2
DP_TI_BYPASS	1
DP_FRAME_INTERVAL	2
DP_FIRST_FRAME_IDX	5
DP_NUM_BLOCK_MAX	10
DP_PAYLOAD_TYPE	2
DP_INBAND_MODE	2
DP_PROTOCOL_TYPE	2
DP_CRC_MODE	2
if DP_PAYLOAD_TYPE = = TS('00')	
DNP_MODE	2
ISSY_MODE	2
HC_MODE_TS	2
if HC_MODE_TS = = '01' or '10'	
PID	13
end	
if DP_PAYLOAD_TYPE == IP('01')	
HC_MODE_IP	2
end	
RESERVED	8
end	
if FIC_FLAG = = 1	
FIC_VERSION	8
FIC_LENGTH_BYTE	13
RESERVED	8
end	
if AUX_FLAG = = 1	
NUM_AUX	4
AUX_CONFIG_RFU	8
for - 1 : NUM_AUX	
AUX_STREAM_TYPE	4
AUX_PRIVATE_CONF	28
end	
end	

[Fig. 15]

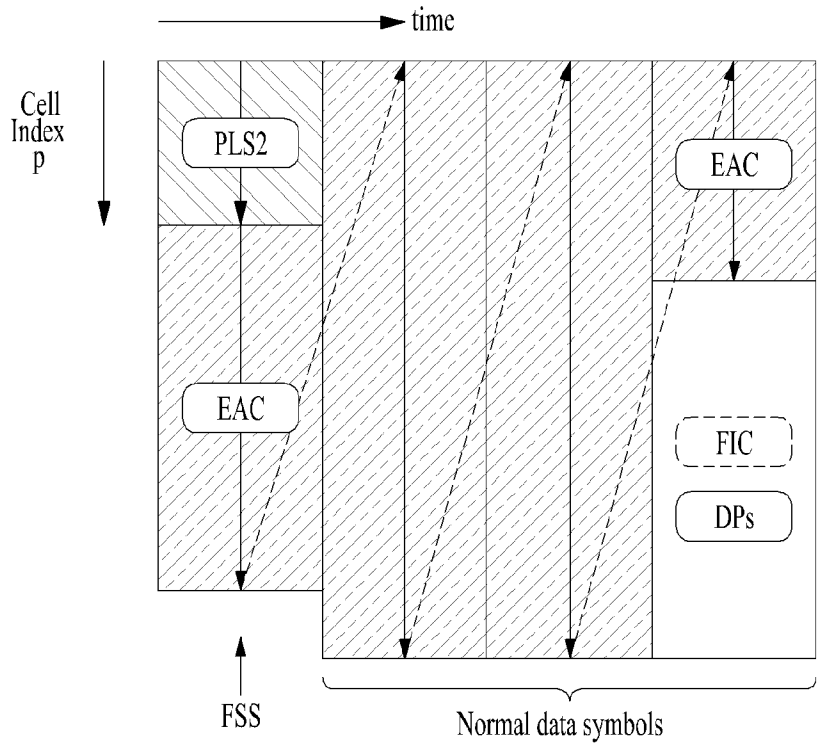
Content		Bit
FRAME_INDEX		5
PLS_CHANGE_COUNTER		4
FIC_CHANGE_COUNTER		4
RESERVED		16
for i = 1: NUM_DP		
	DP_ID DP_START	6 15 (or13)
	DP_NUM_BLOCK	10
end	RESERVED	8
EAC_FLAG		1
EAS_WAKE_UP_VERSION_NUM		8
if EAC_FLAG == 1		
	EAC_LENGTH_BYTE	12
else		
	EAC_COUNTER	12
end		
for i=1:NUM_AUX		
	AUX_PRIVATE_DYN	48
end		
CRC 32		32

[Fig. 16]

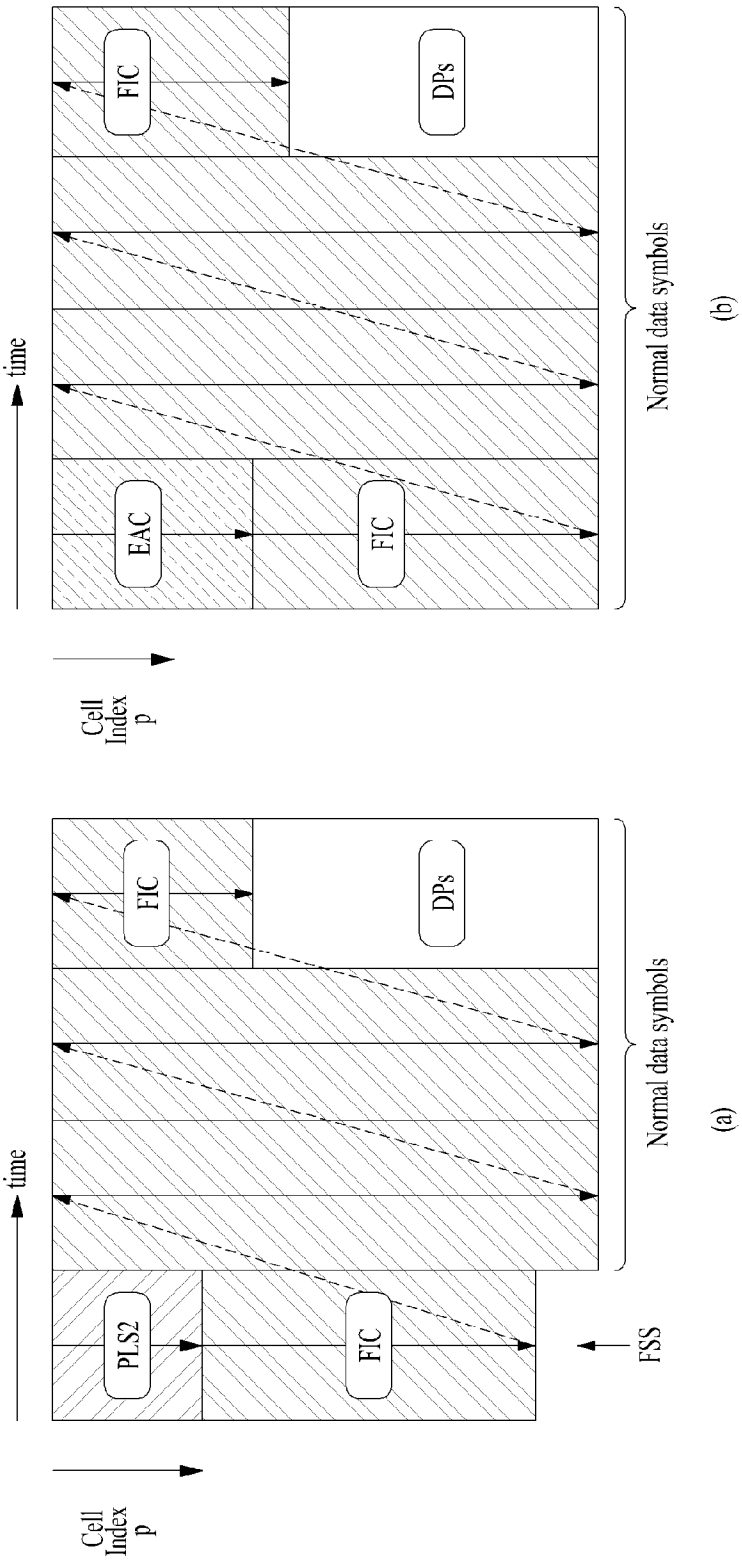
[Fig. 17]


time
Cell Index p
PLS1
PLS2
PLS2
EAC
FIC
DPs
N_{FSS}

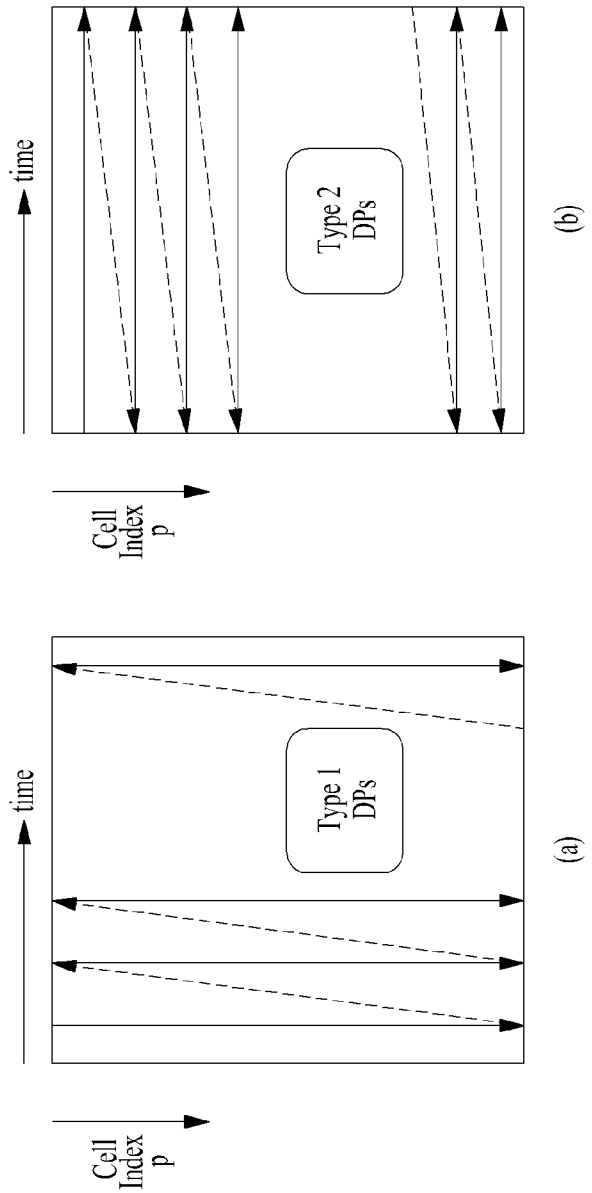
[Fig. 18]


time
Cell Index p
PLS2
EAC
EAC
FIC
DPs
FSS
Normal data symbols

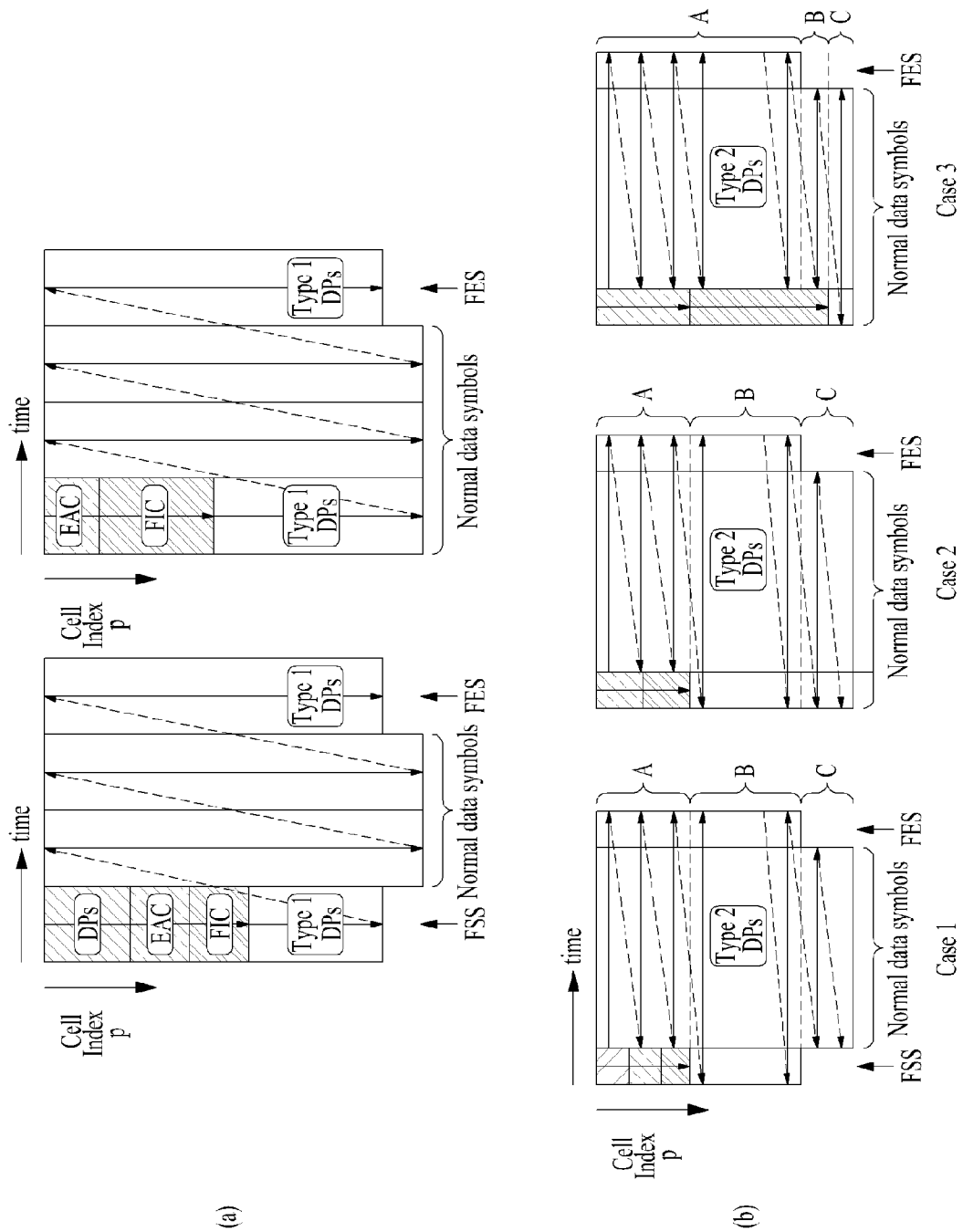
[Fig. 19]


time
Cell Index p
PLS2
FIC
FIC
DPs
FSS
Normal data symbols
(a)
time
Cell Index p
EAC
FIC
FIC
DPs
Normal data symbols
(b)

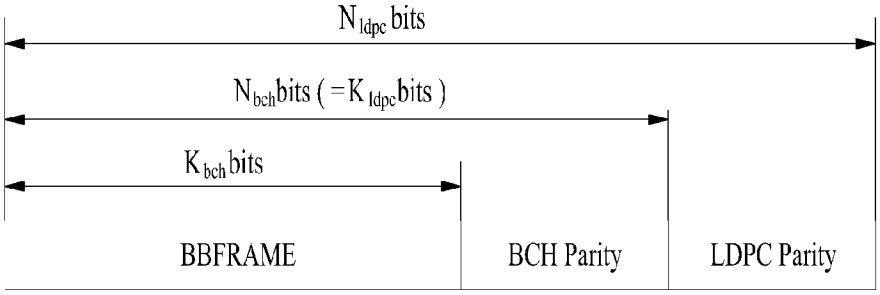
[Fig. 20]



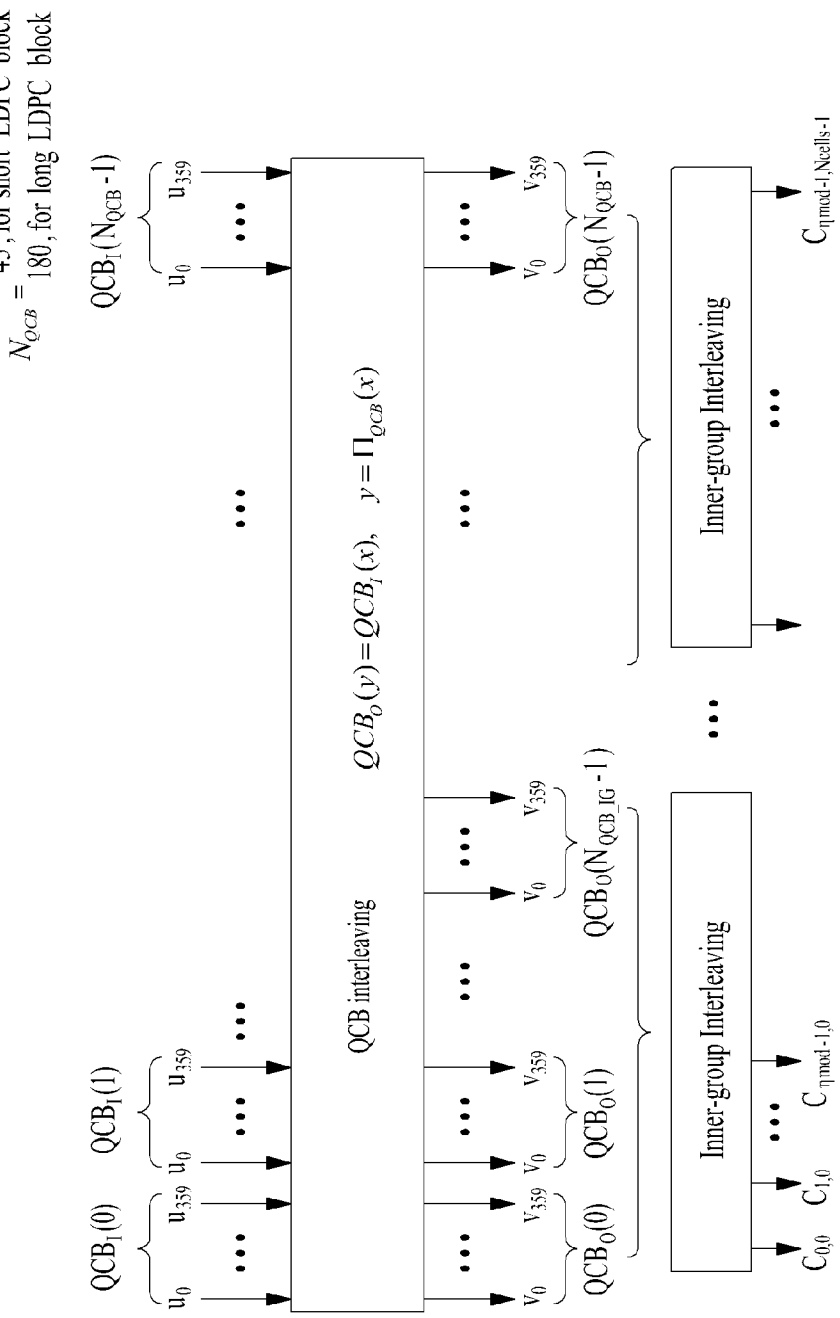
[Fig. 21]


time
Cell Index p
DPs
EAC
FIC
Type 1 DPs
Type 2 DPs
FSS
FES
Normal data symbols
A
B
C
Case 1
Case 2
Case 3
(a)
(b)

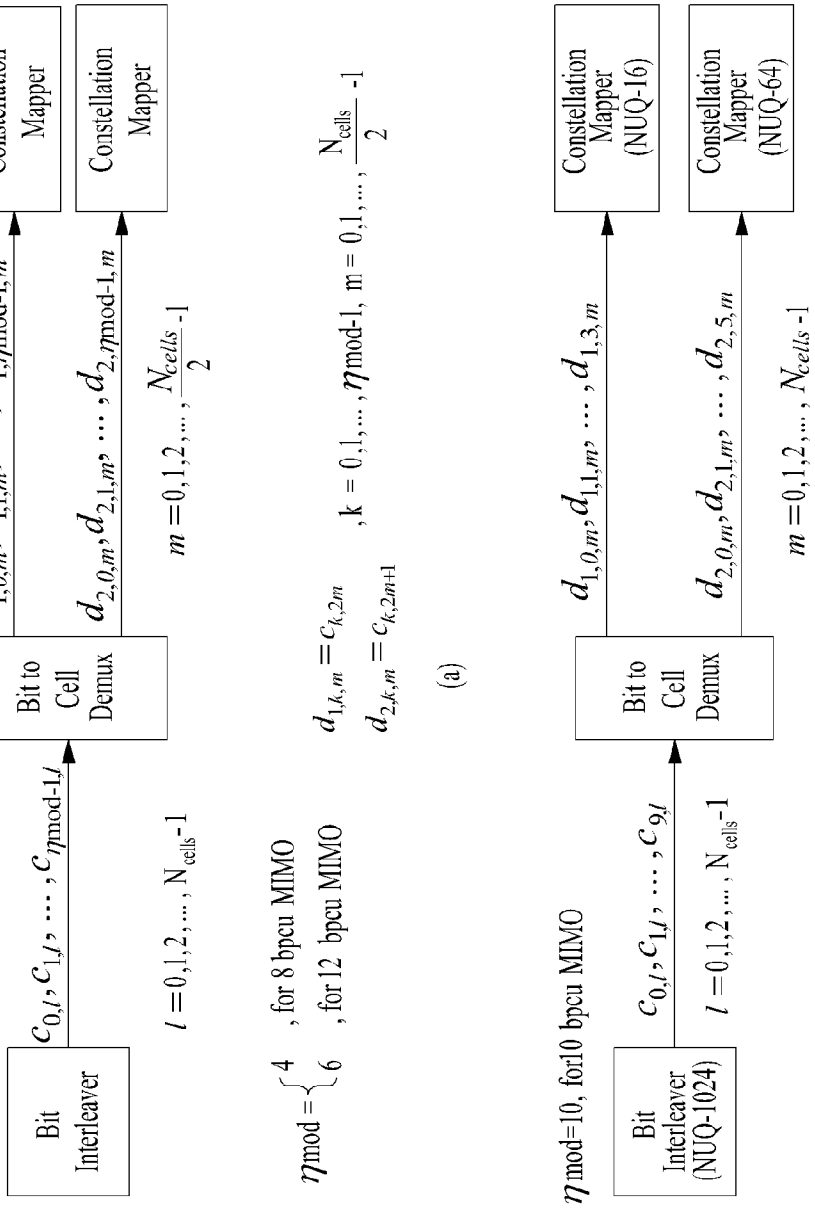
[Fig. 22]


N_{ldpc} bits
N_{bch} bits ($=K_{ldpc}$ bits)
K_{bch} bits
BBFRAME
BCH Parity
LDPC Parity

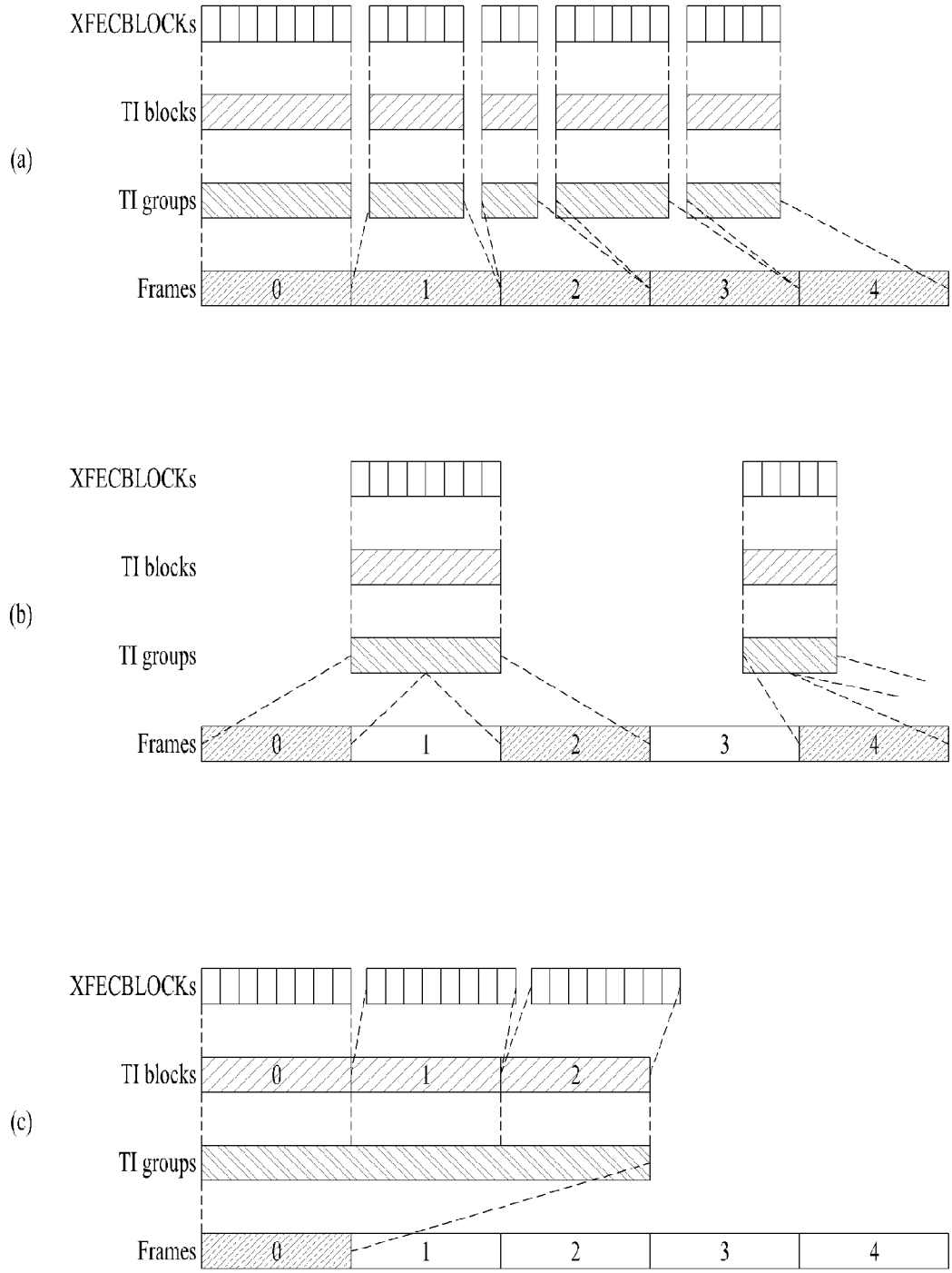
[Fig. 23]


N_{QCB} = 45, for short LDPC block
180, for long LDPC block
$QCB_I(N_{QCB}-1)$
$QCB_I(1)$
$QCB_I(0)$
u_0 u_{359}
QCB interleaving
$QCB_O(y)=QCB_I(x), \quad y=\Pi_{QCB}(x)$
v_0 v_{359}
$QCB_O(N_{QCB}-1)$
$QCB_O(1)$
$QCB_O(0)$
$QCB_O(N_{QCB\_IG}-1)$
Inner-group Interleaving
Inner-group Interleaving
$C_{0,0}$ $C_{1,0}$ $C_{\eta mod-1,0}$
$C_{\eta mod-1,Ncells-1}$

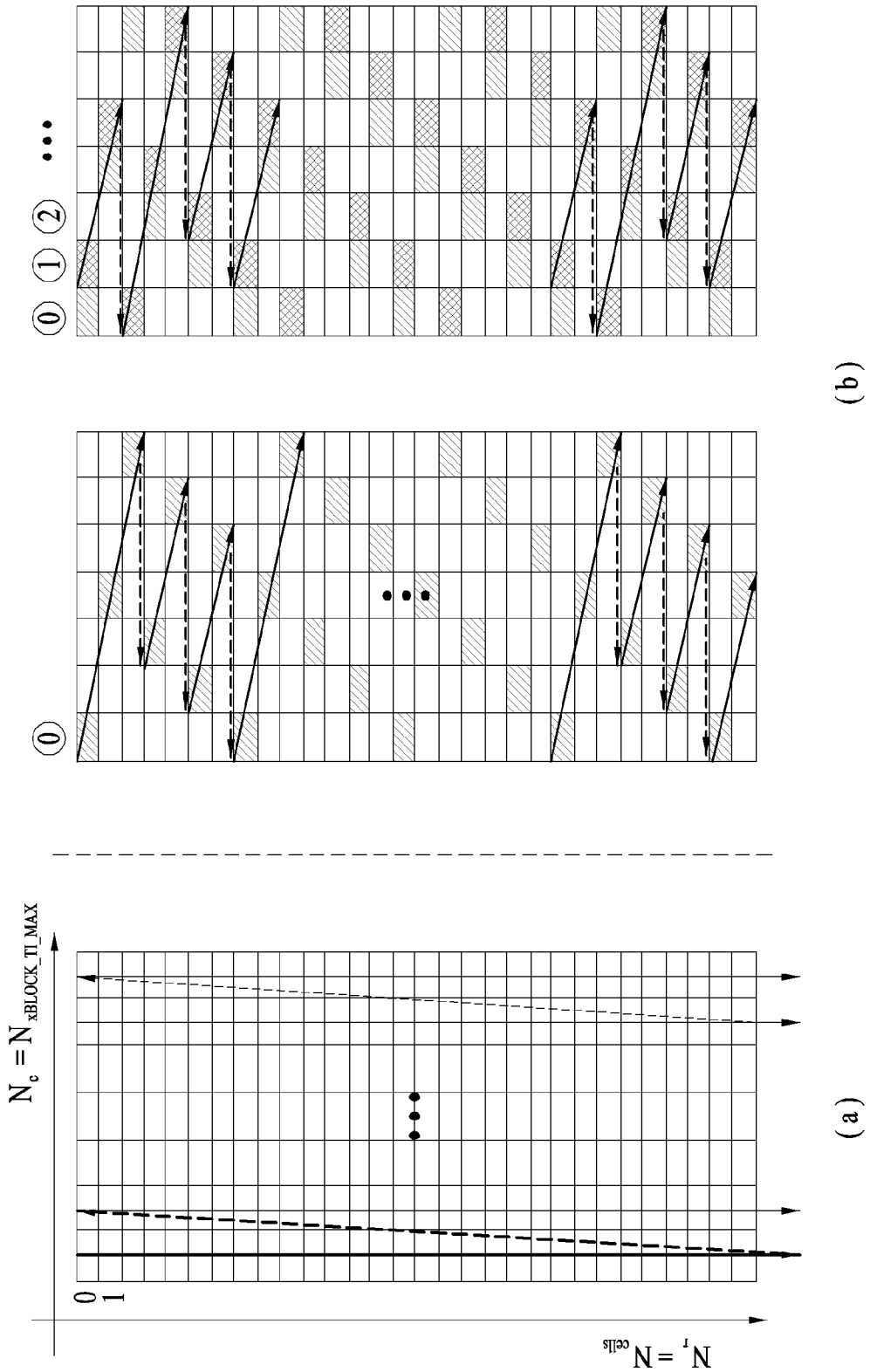
[Fig. 24]


Bit Interleaver
$c_{0,l}, c_{1,l}, \cdots, c_{\eta\text{mod-}1,l}$
$l = 0,1,2,\cdots, N_{cells}-1$
Bit to Cell Demux
$d_{1,0,m}, d_{1,1,m}, \cdots, d_{1,\eta\text{mod-}1,m}$
$d_{2,0,m}, d_{2,1,m}, \cdots, d_{2,\eta\text{mod-}1,m}$
$m = 0,1,2,\cdots, \frac{N_{cells}}{2}-1$
Constellation Mapper
Constellation Mapper
$\eta\text{mod} = \begin{cases} 4 & \text{, for 8 bpcu MIMO} \\ 6 & \text{, for 12 bpcu MIMO} \end{cases}$
$d_{1,k,m} = c_{k,2m}$
$d_{2,k,m} = c_{k,2m+1}$
$k = 0,1,\cdots \eta\text{mod-}1,\ m = 0,1,\cdots, \frac{N_{cells}}{2}-1$
(a)
$\eta\text{mod}=10$, for10 bpcu MIMO
Bit Interleaver (NUQ-1024)
$c_{0,l}, c_{1,l}, \cdots, c_{9,l}$
$l = 0,1,2,\cdots, N_{cells}-1$
Bit to Cell Demux
$d_{1,0,m}, d_{1,1,m}, \cdots, d_{1,3,m}$
$d_{2,0,m}, d_{2,1,m}, \cdots, d_{2,5,m}$
$m = 0,1,2,\cdots, N_{cells}-1$
Constellation Mapper (NUQ-16)
Constellation Mapper (NUQ-64)
$\{d_{1,0,m}, d_{1,1,m}, d_{1,2,m}, d_{1,3,m}\} = \{c_{0,m}, c_{1,m}, c_{4,m}, c_{5,m}\}$
$\{d_{2,0,m}, d_{2,1,m}, d_{2,2,m}, d_{2,3,m}, d_{2,4,m}, d_{2,5,m}\} = \{c_{2,m}, c_{3,m}, c_{6,m}, c_{7,m}, c_{8,m}, c_{9,m}\}$
$m = 0,1,\cdots, N_{cells}-1$
(b)

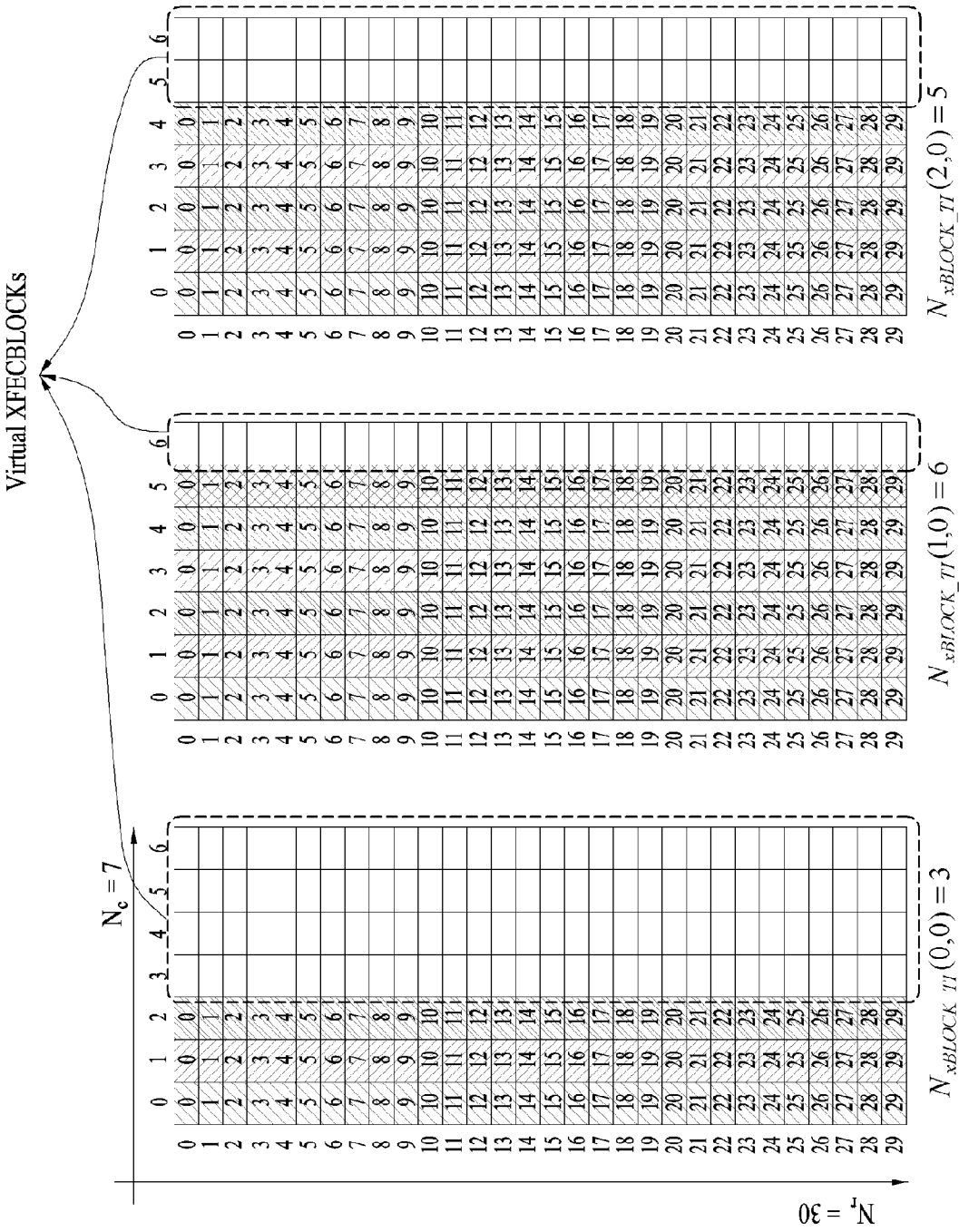
[Fig. 25]


(a)
XFECBLOCKs
TI blocks
TI groups
Frames
0
1
2
3
4
(b)
XFECBLOCKs
TI blocks
TI groups
Frames
0
1
2
3
4
(c)
XFECBLOCKs
TI blocks
TI groups
Frames
0
1
2
3
4

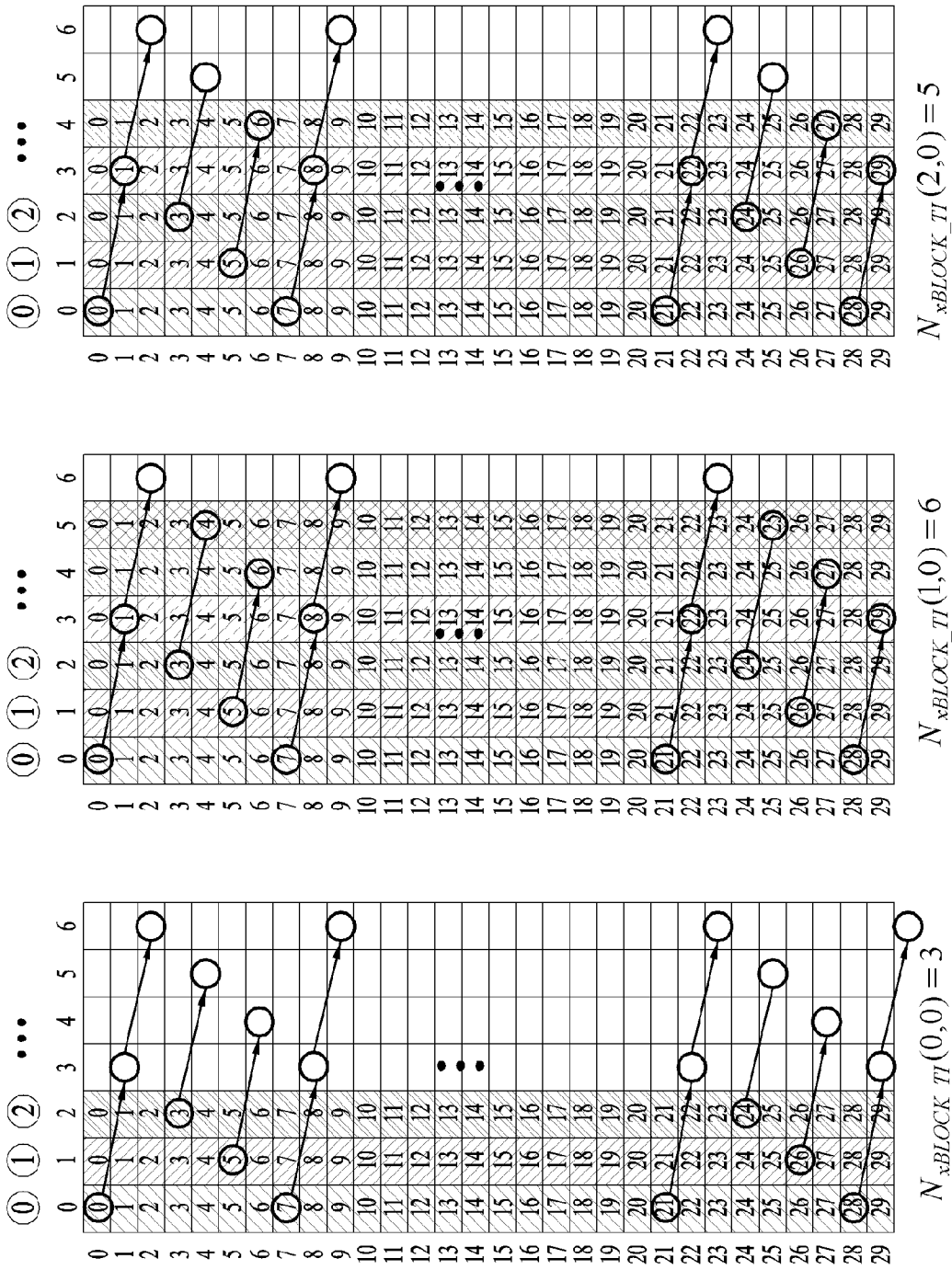
[Fig. 26]



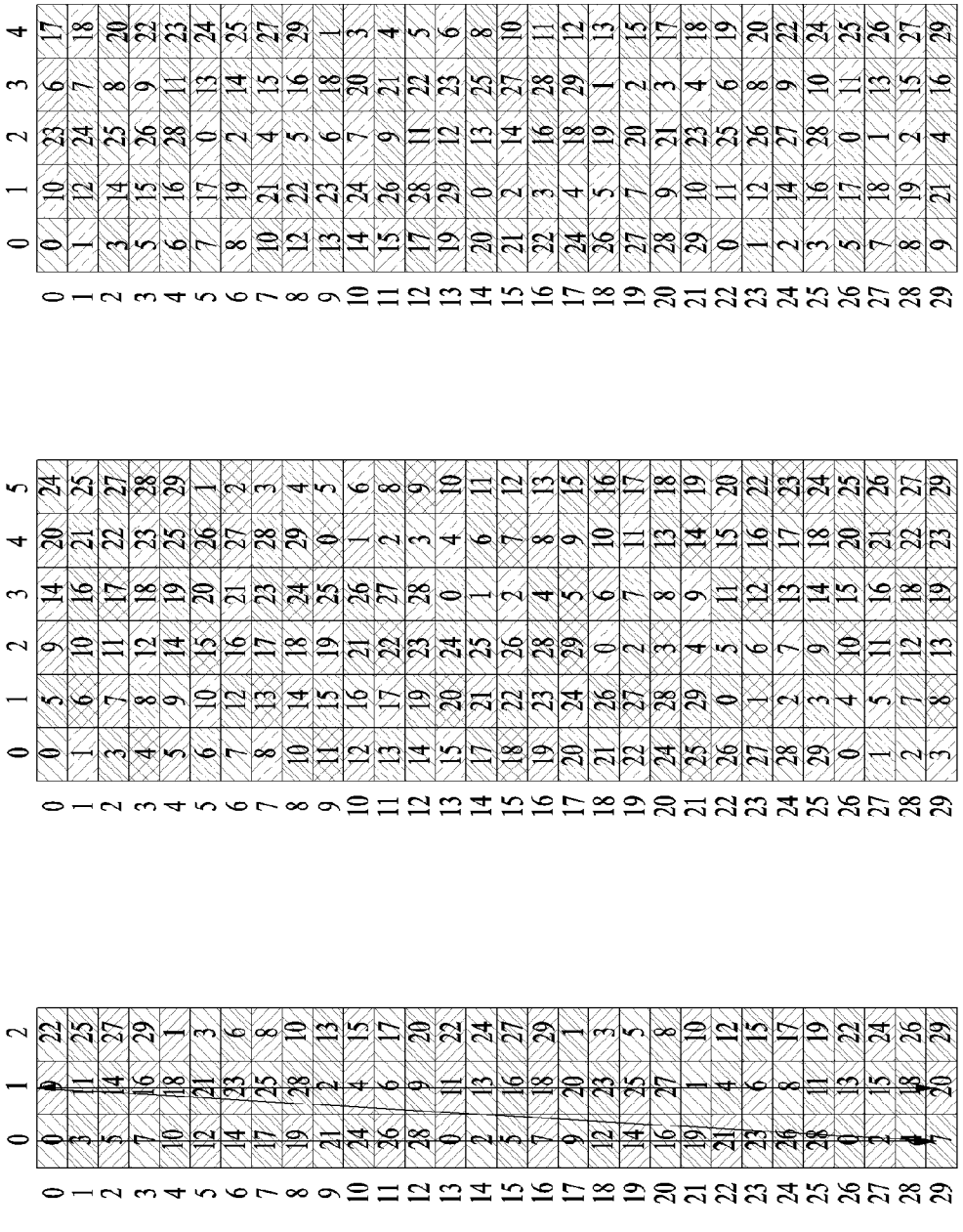
[Fig. 27]


Virtual XFECBLOCKs
$N_c = 7$
$N_r = 30$
$N_{xBLOCK\_TI}(0,0) = 3$
$N_{xBLOCK\_TI}(1,0) = 6$
$N_{xBLOCK\_TI}(2,0) = 5$

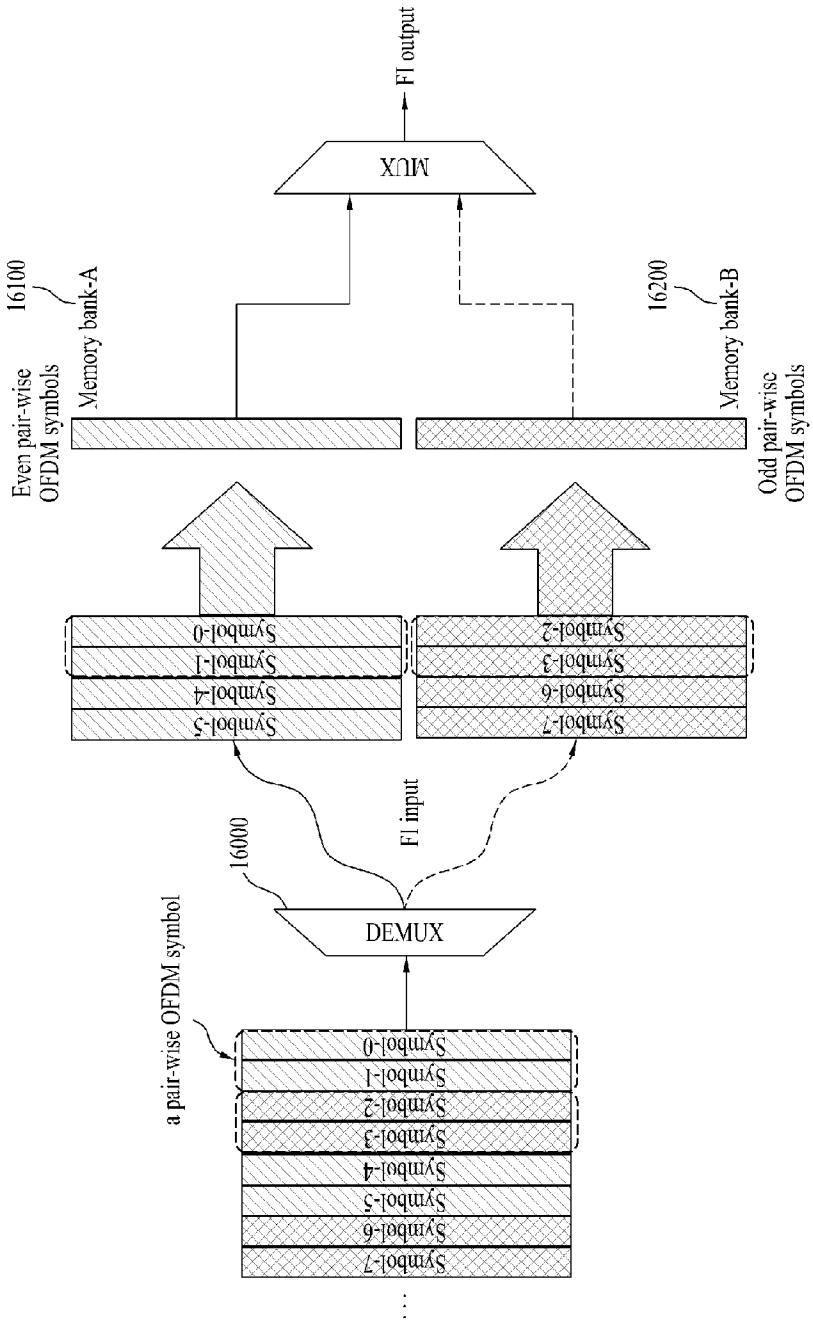
[Fig. 28]



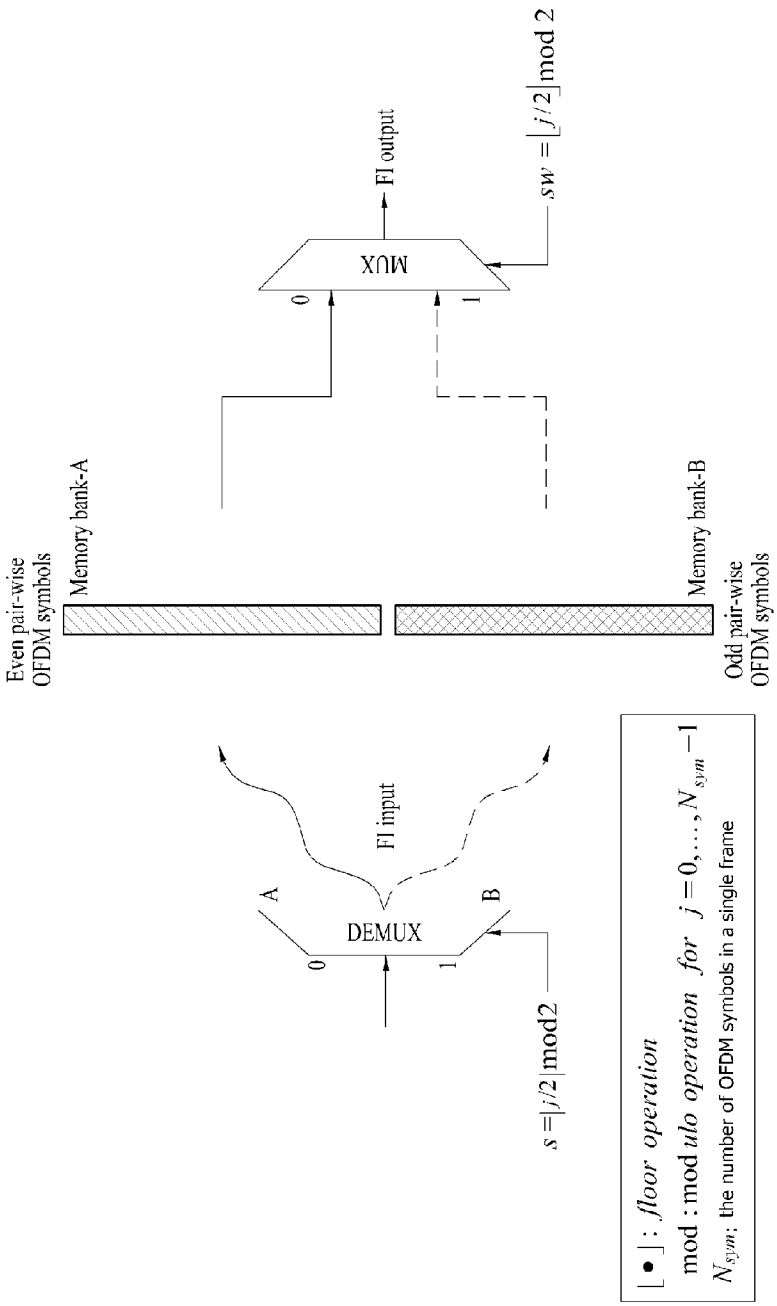
[Fig. 29]



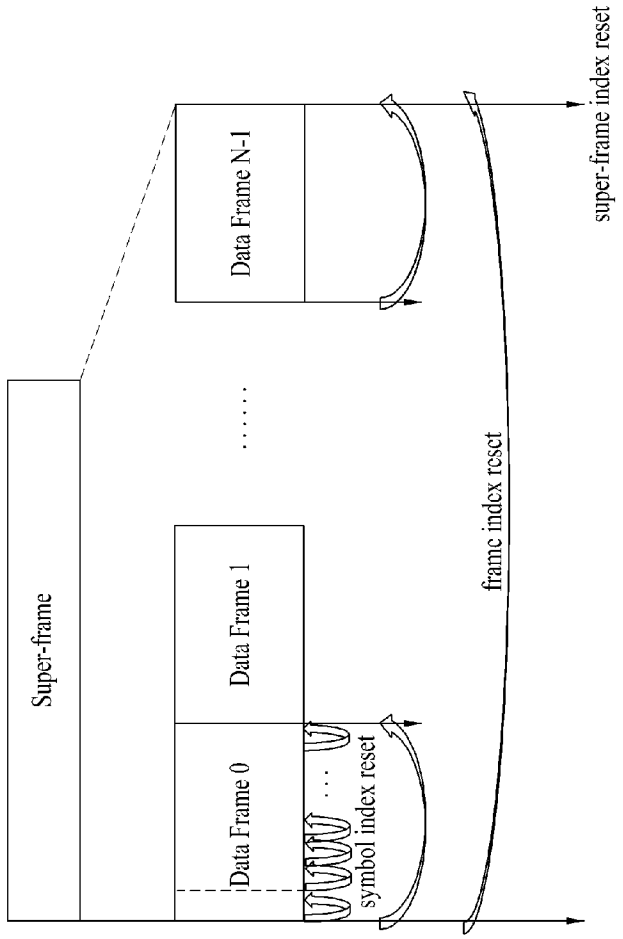
[Fig. 30]


FI output
MUX
16100
Memory bank-A
Even pair-wise OFDM symbols
16200
Memory bank-B
Odd pair-wise OFDM symbols
Symbol-0
Symbol-1
Symbol-4
Symbol-5
Symbol-2
Symbol-3
Symbol-6
Symbol-7
FI input
16000
DEMUX
a pair-wise OFDM symbol
Symbol-0
Symbol-1
Symbol-2
Symbol-3
Symbol-4
Symbol-5
Symbol-6
Symbol-7

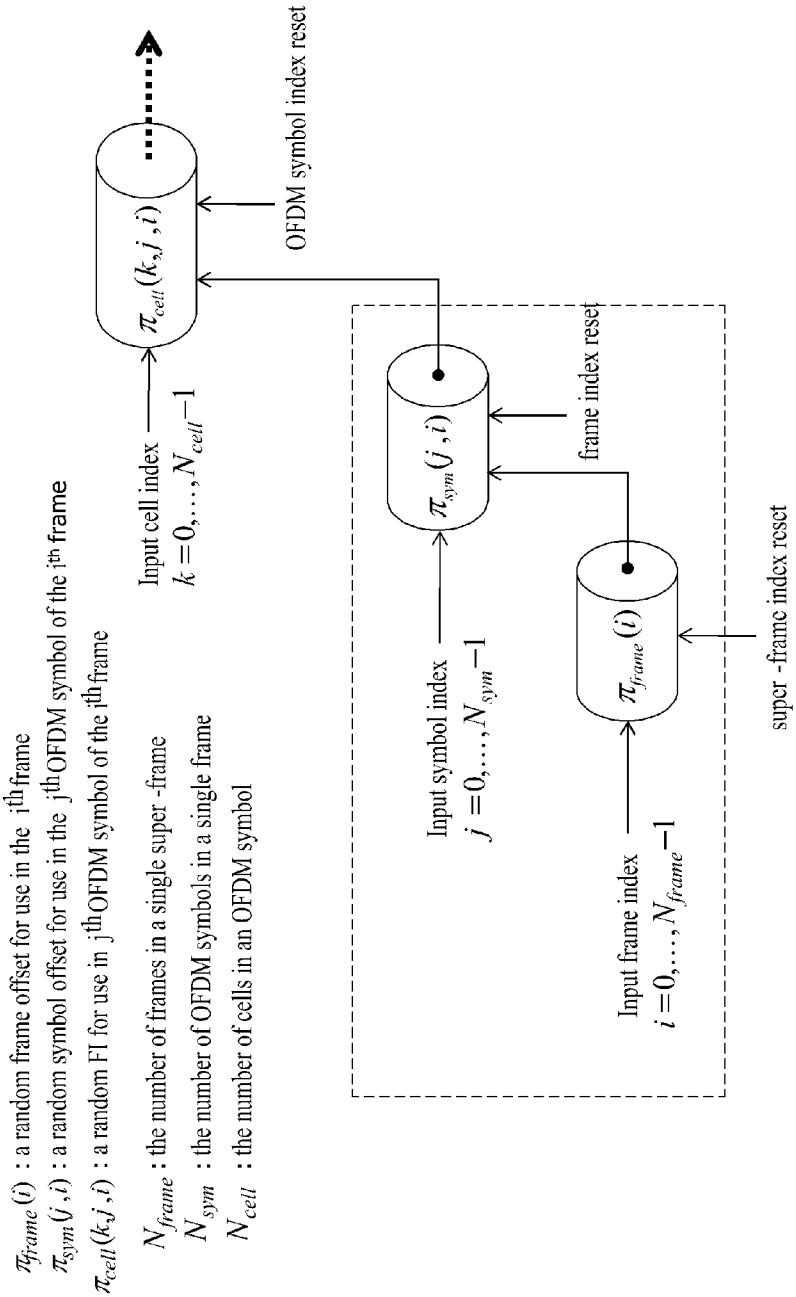
[Fig. 31]


Even pair-wise OFDM symbols
Memory bank-A
Memory bank-B
Odd pair-wise OFDM symbols
FI output
MUX
0
1
$sw = \lfloor j/2 \rfloor \bmod 2$
FI input
DEMUX
A
B
0
1
$s = \lfloor j/2 \rfloor \bmod 2$
$\lfloor \bullet \rfloor$: *floor operation*
mod : mod*ulo operation* *for* $j = 0, \ldots, N_{sym} - 1$
N_{sym}: the number of OFDM symbols in a single frame

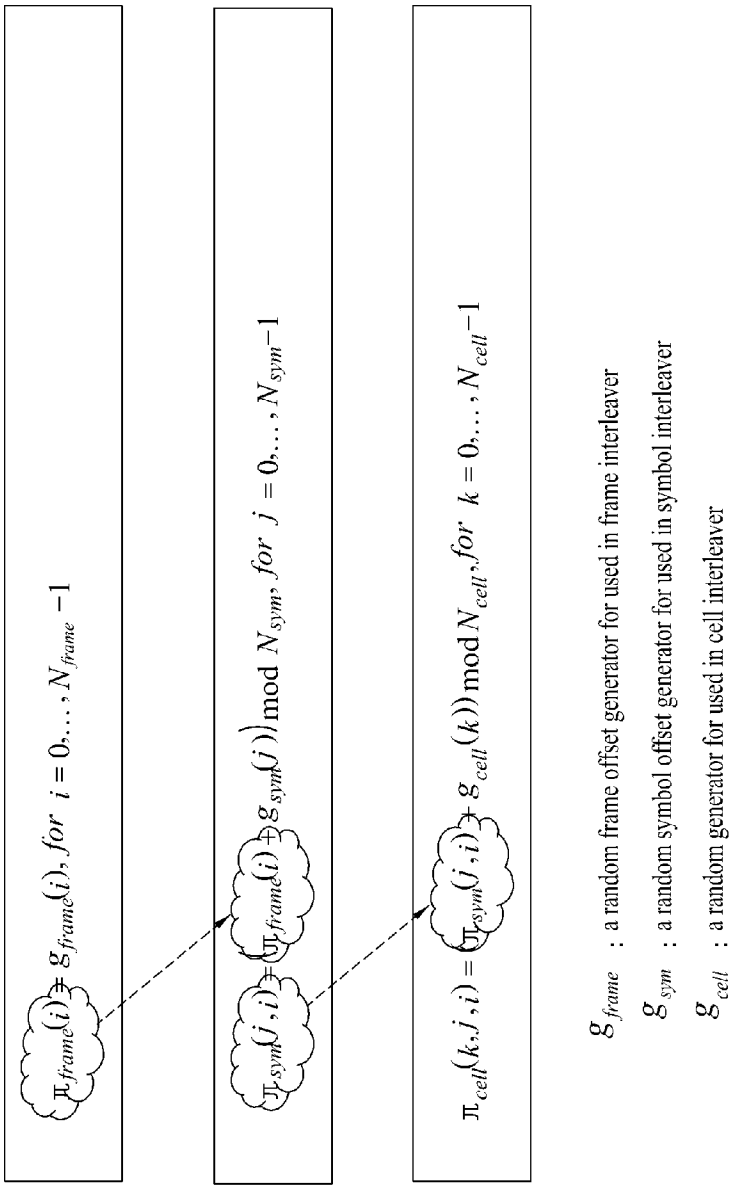
[Fig. 32]


Super-frame
Data Frame 0
Data Frame 1
Data Frame N-1
symbol index reset
framc indcx rcsct
super-frame index reset

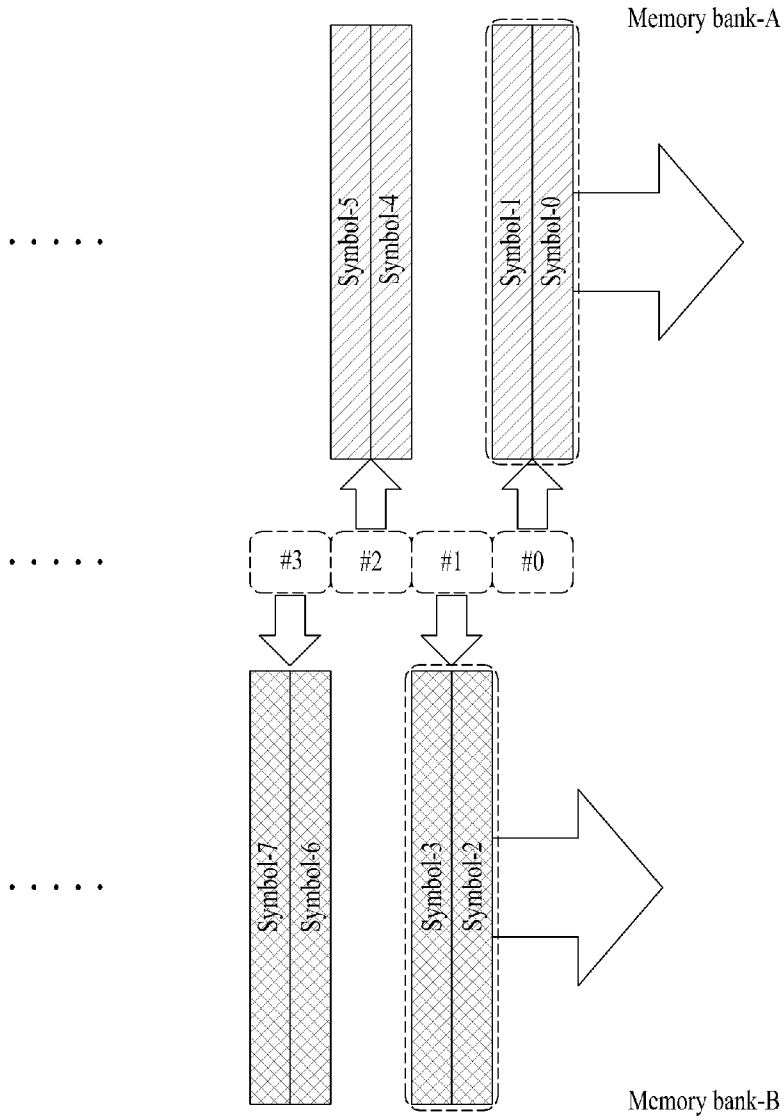
[Fig. 33]


$\pi_{frame}(i)$: a random frame offset for use in the i^{th} frame
$\pi_{sym}(j,i)$: a random symbol offset for use in the j^{th} OFDM symbol of the i^{th} frame
$\pi_{cell}(k,j,i)$: a random FI for use in j^{th} OFDM symbol of the i^{th} frame
N_{frame} : the number of frames in a single super -frame
N_{sym} : the number of OFDM symbols in a single frame
N_{cell} : the number of cells in an OFDM symbol
$\pi_{cell}(k,j,i)$
$\pi_{sym}(j,i)$
$\pi_{frame}(i)$
Input cell index
$k=0,\ldots,N_{cell}-1$
Input symbol index
$j=0,\ldots,N_{sym}-1$
Input frame index
$i=0,\ldots,N_{frame}-1$
OFDM symbol index reset
frame index reset
super -frame index reset

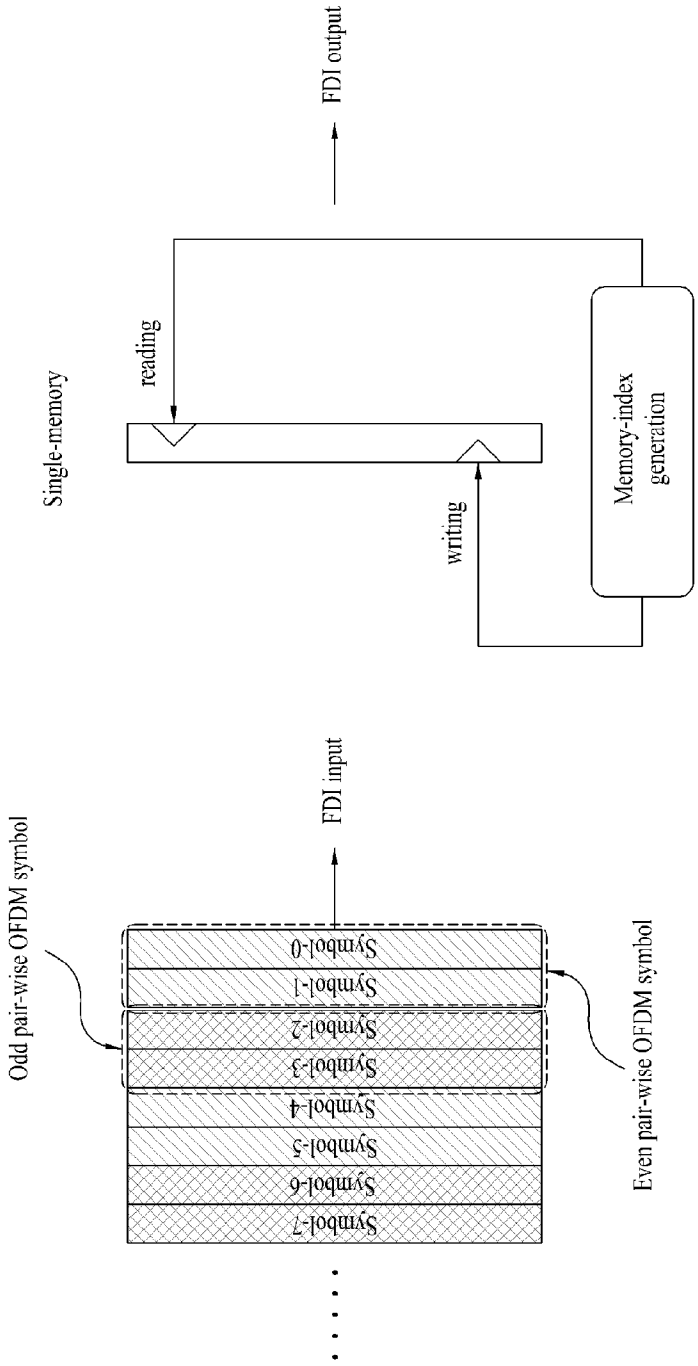
[Fig. 34]


$\pi_{frame}(i) = g_{frame}(i), for \ i = 0,\ldots,N_{frame}-1$
$\pi_{sym}(j,i) = \left(\pi_{frame}(i) + g_{sym}(j)\right) \bmod N_{sym}, for \ j = 0,\ldots,N_{sym}-1$
$\pi_{cell}(k,j,i) = \left(\pi_{sym}(j,i) + g_{cell}(k)\right) \bmod N_{cell}, for \ k = 0,\ldots,N_{cell}-1$
g_{frame} : a random frame offset generator for used in frame interleaver
g_{sym} : a random symbol offset generator for used in symbol interleaver
g_{cell} : a random generator for used in cell interleaver

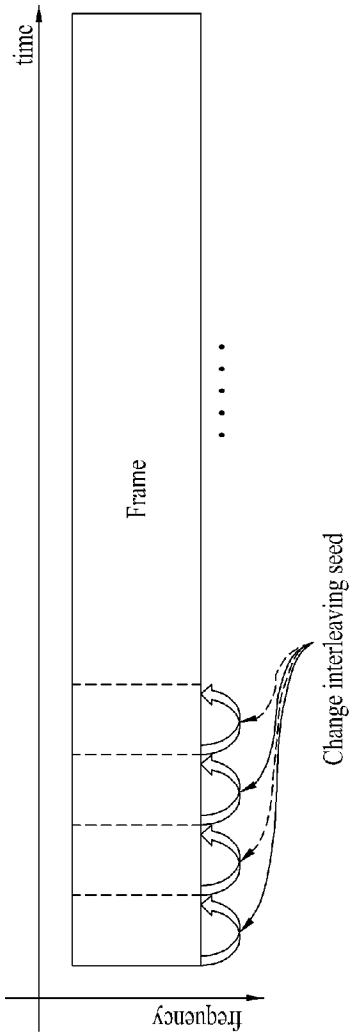
[Fig. 35]


Memory bank-A
Symbol-5
Symbol-4
Symbol-1
Symbol-0
#3
#2
#1
#0
Symbol-7
Symbol-6
Symbol-3
Symbol-2
Memory bank-B

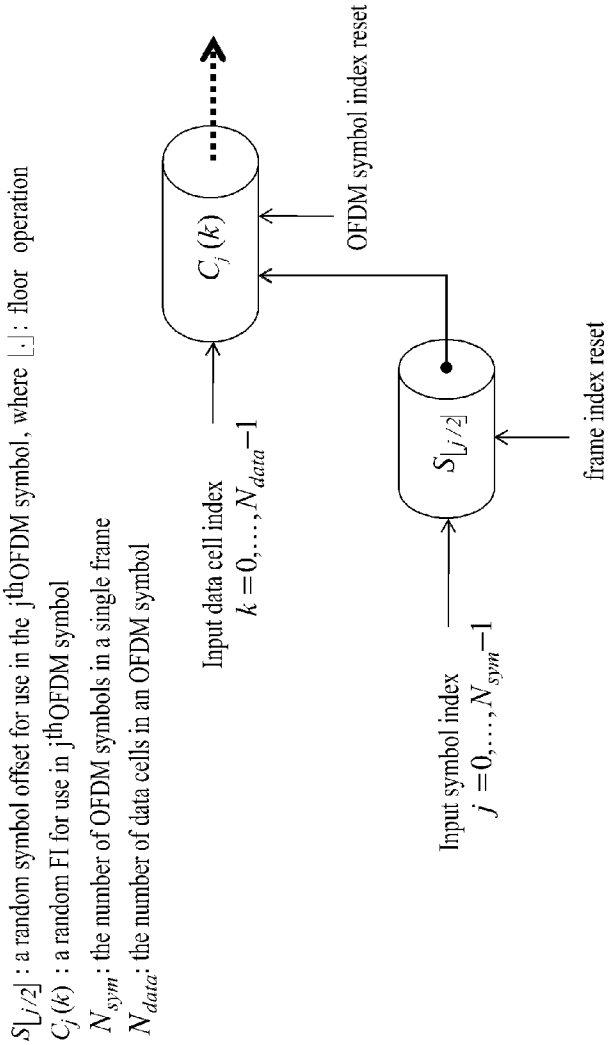
[Fig. 36]


FDI output
reading
Single-memory
writing
Memory-index generation
FDI input
Odd pair-wise OFDM symbol
Even pair-wise OFDM symbol
Symbol-0
Symbol-1
Symbol-2
Symbol-3
Symbol-4
Symbol-5
Symbol-6
Symbol-7

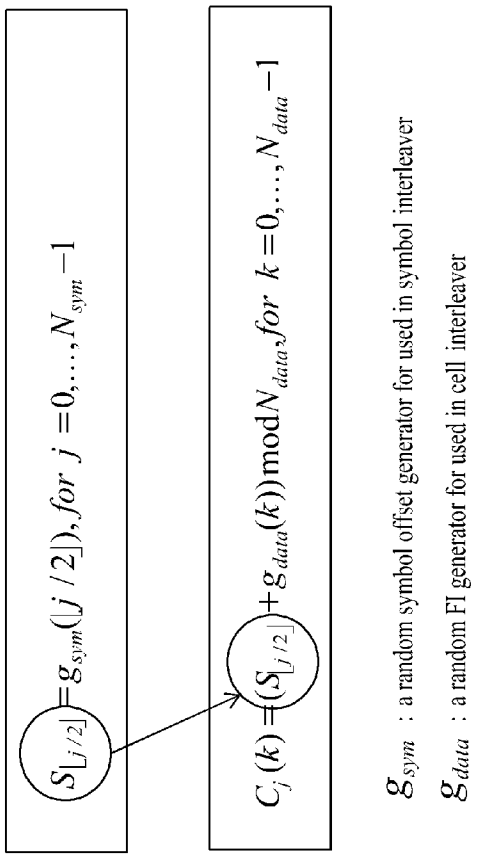
[Fig. 37]



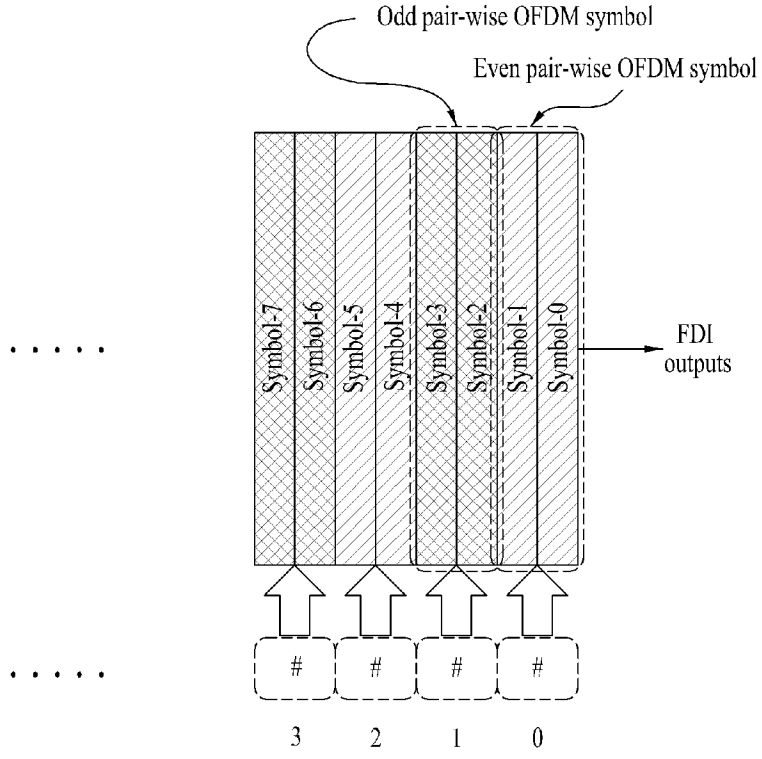
[Fig. 38]


$S_{\lfloor j/2 \rfloor}$: a random symbol offset for use in the j^thOFDM symbol, where $\lfloor \cdot \rfloor$: floor operation
$C_j(k)$: a random FI for use in j^thOFDM symbol
N_{sym}: the number of OFDM symbols in a single frame
N_{data}: the number of data cells in an OFDM symbol
$C_j(k)$
OFDM symbol index reset
Input data cell index
$k = 0, \ldots, N_{data} - 1$
$S_{\lfloor j/2 \rfloor}$
frame index reset
Input symbol index
$j = 0, \ldots, N_{sym} - 1$

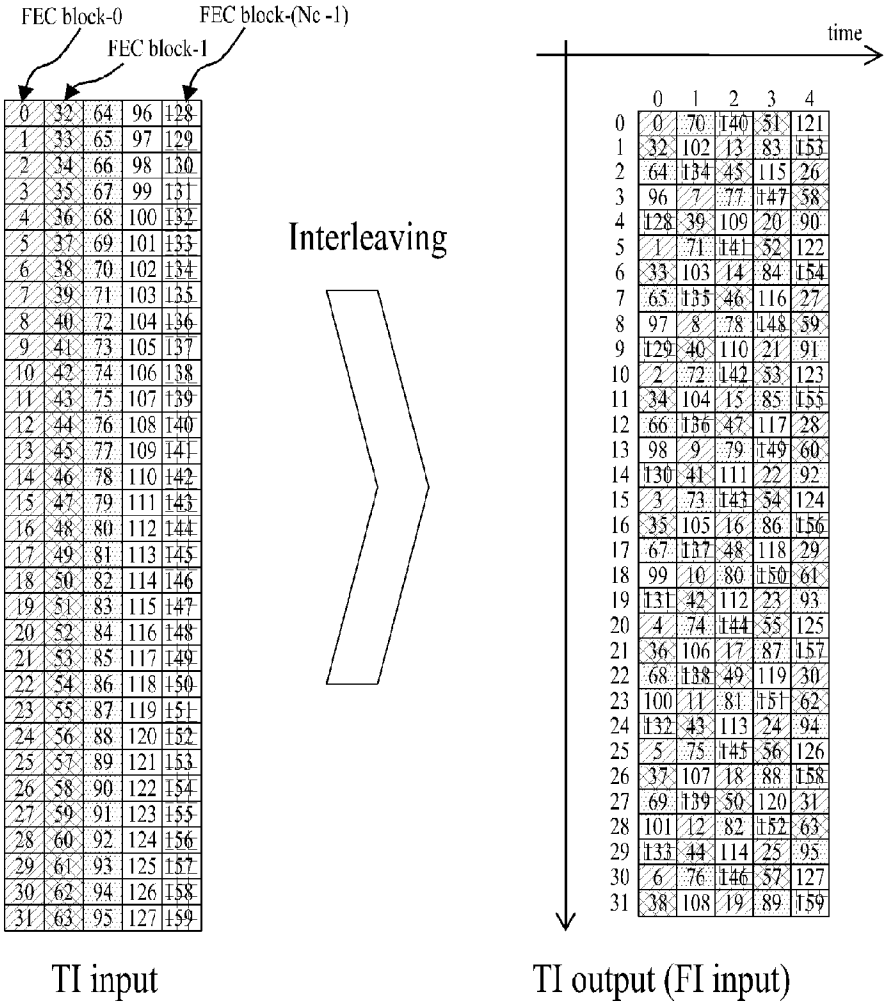
[Fig. 39]


$S_{\lfloor j/2 \rfloor} = g_{sym}(\lfloor j/2 \rfloor), for\ j = 0,\ldots,N_{sym}-1$
$C_j(k) = (S_{\lfloor j/2 \rfloor}) + g_{data}(k) \bmod N_{data}, for\ k = 0,\ldots,N_{data}-1$
g_{sym} : a random symbol offset generator for used in symbol interleaver
g_{data} : a random FI generator for used in cell interleaver

[Fig. 40]


Odd pair-wise OFDM symbol
Even pair-wise OFDM symbol
Symbol-7
Symbol-6
Symbol-5
Symbol-4
Symbol-3
Symbol-2
Symbol-1
Symbol-0
FDI outputs
#
#
#
#
3
2
1
0
Deinterleaving seeds corresponding to interleaving seeds

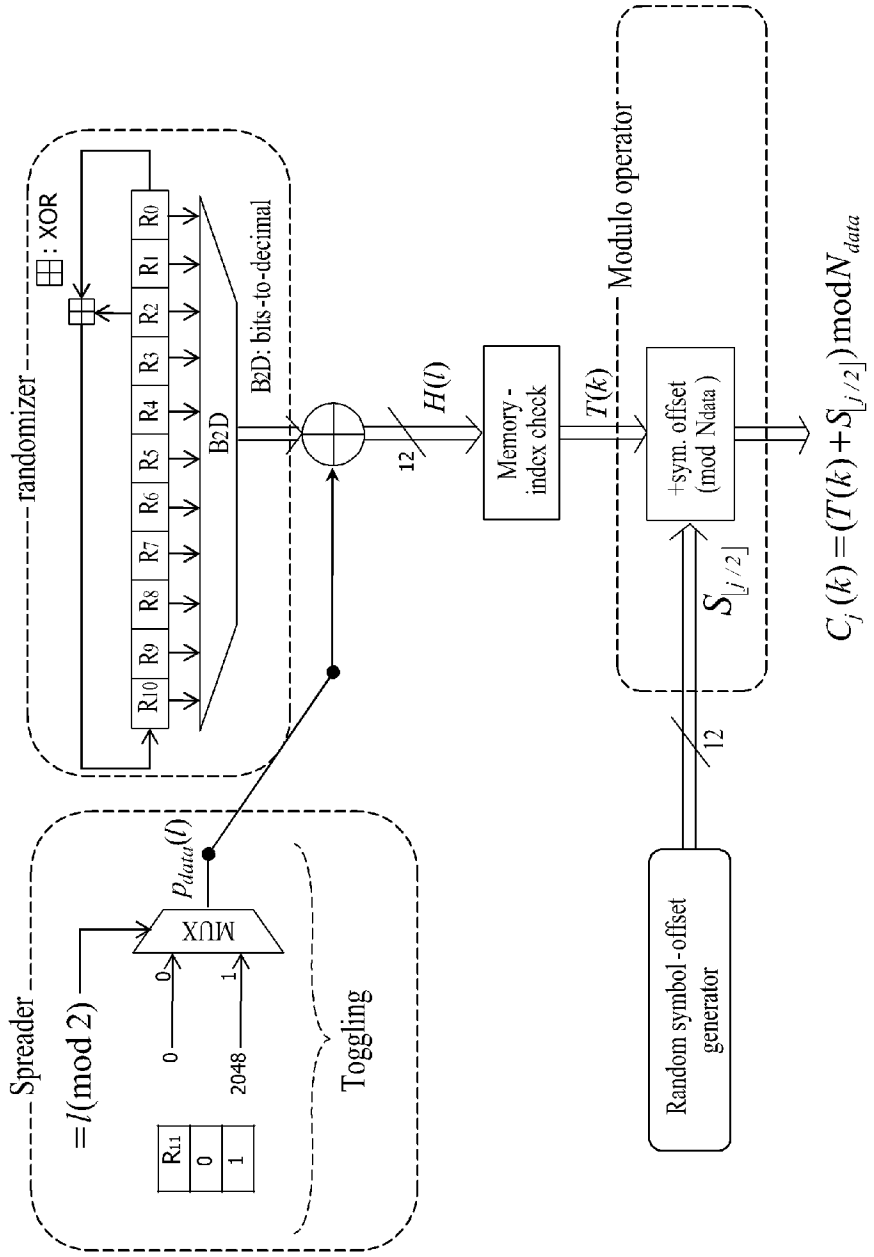
[Fig. 41]



0	32	64	96	128
1	33	65	97	129
2	34	66	98	130
3	35	67	99	131
4	36	68	100	132
5	37	69	101	133
6	38	70	102	134
7	39	71	103	135
8	40	72	104	136
9	41	73	105	137
10	42	74	106	138
11	43	75	107	139
12	44	76	108	140
13	45	77	109	141
14	46	78	110	142
15	47	79	111	143
16	48	80	112	144
17	49	81	113	145
18	50	82	114	146
19	51	83	115	147
20	52	84	116	148
21	53	85	117	149
22	54	86	118	150
23	55	87	119	151
24	56	88	120	152
25	57	89	121	153
26	58	90	122	154
27	59	91	123	155
28	60	92	124	156
29	61	93	125	157
30	62	94	126	158
31	63	95	127	159

	0	1	2	3	4
0	0	70	140	51	121
1	32	102	13	83	153
2	64	134	45	115	26
3	96	7	77	147	58
4	128	39	109	20	90
5	1	71	141	52	122
6	33	103	14	84	154
7	65	135	46	116	27
8	97	8	78	148	59
9	129	40	110	21	91
10	2	72	142	53	123
11	34	104	15	85	155
12	66	136	47	117	28
13	98	9	79	149	60
14	130	41	111	22	92
15	3	73	143	54	124
16	35	105	16	86	156
17	67	137	48	118	29
18	99	10	80	150	61
19	131	42	112	23	93
20	4	74	144	55	125
21	36	106	17	87	157
22	68	138	49	119	30
23	100	11	81	151	62
24	132	43	113	24	94
25	5	75	145	56	126
26	37	107	18	88	158
27	69	139	50	120	31
28	101	12	82	152	63
29	133	44	114	25	95
30	6	76	146	57	127
31	38	108	19	89	159

[Fig. 42]


randomizer
⊞: XOR
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
B2D
B2D: bits-to-decimal
12
H(l)
Memory - index check
T(k)
Modulo operator
+sym. offset (mod Ndata)
$S_{[j/2]}$
$C_j(k) = (T(k) + S_{[j/2]}) \bmod N_{data}$
12
Random symbol - offset generator
Spreader
$= l \pmod 2$
MUX
0
1
0
2048
R11
0
1
$p_{data}(l)$
Toggling

[Fig. 43]

· The associated 11th primitive polynomial, $f(x)=1+x^{9}+x^{11}$

· Operation of the PN generator for $0 \le l < N_{\max}(4096)$

$0 \le l < 2$

$R^{l}_{[10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0$

$l = 2$

$R^{2}_{[10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,1,0,1$

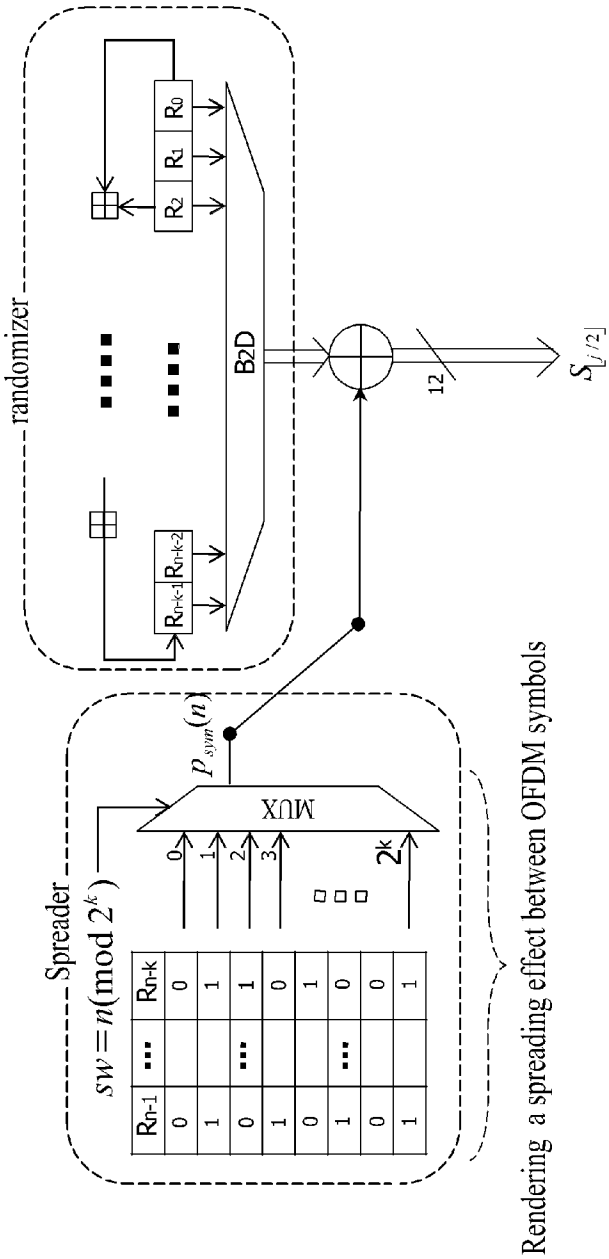
$2 < l < N_{\max}$

$R^{l}_{[9,8,7,6,5,4,3,2,1,0]} = R^{l-1}_{[10,9,8,7,6,5,4,3,2,1]}$

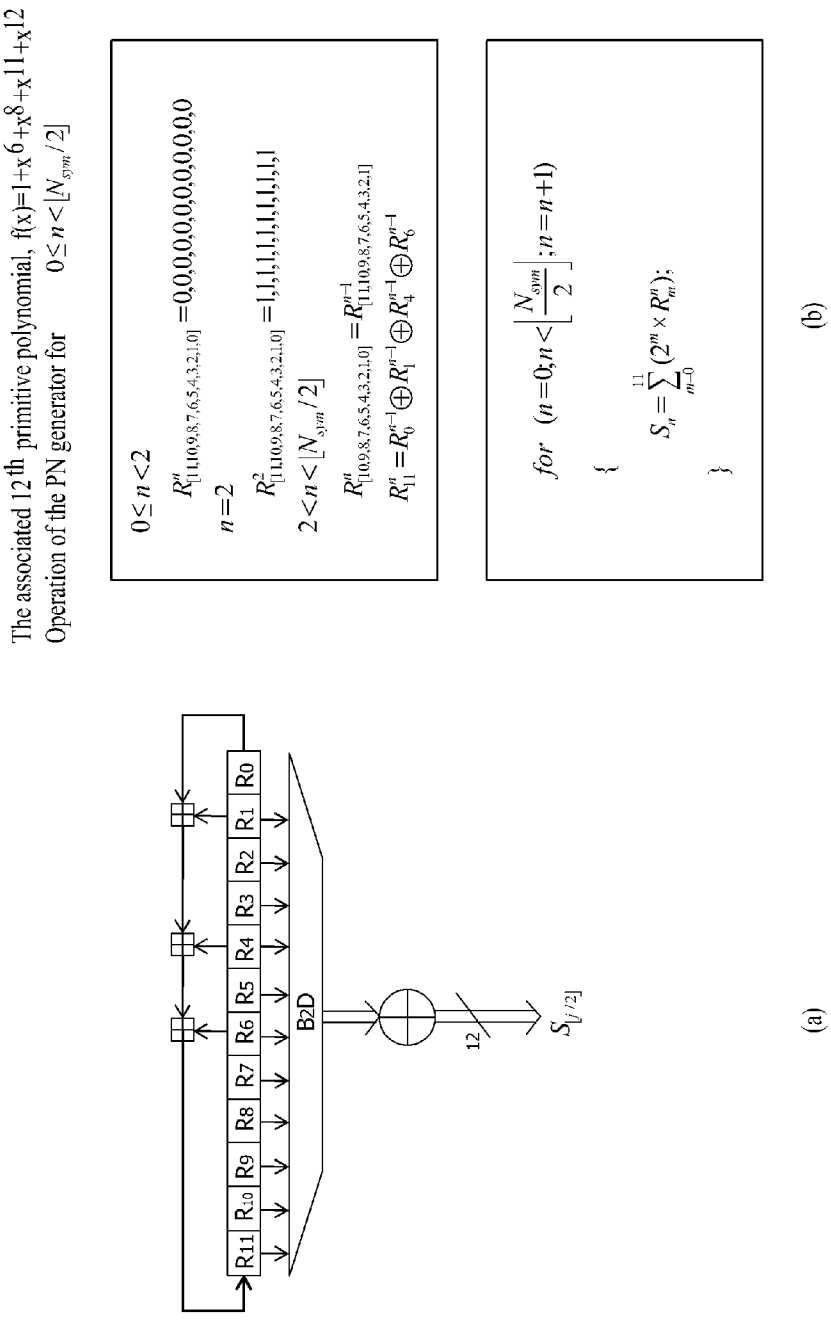
$R^{l}_{10} = R^{l-1}_{0} \oplus R^{l-1}_{2}$

```
k = 0;
for (l = 0; l < N_max; l = l + 1)
{
    H(l) = P_data(l) + Σ_{n=0}^{10} (2^n × R_n^l);
    if H(l) < N_data
    {
        T(k) = H(l);
        C_j(k) = (T(k) + S_⌊j/2⌋) mod N_data;
        k = k + 1;
    }
}
```

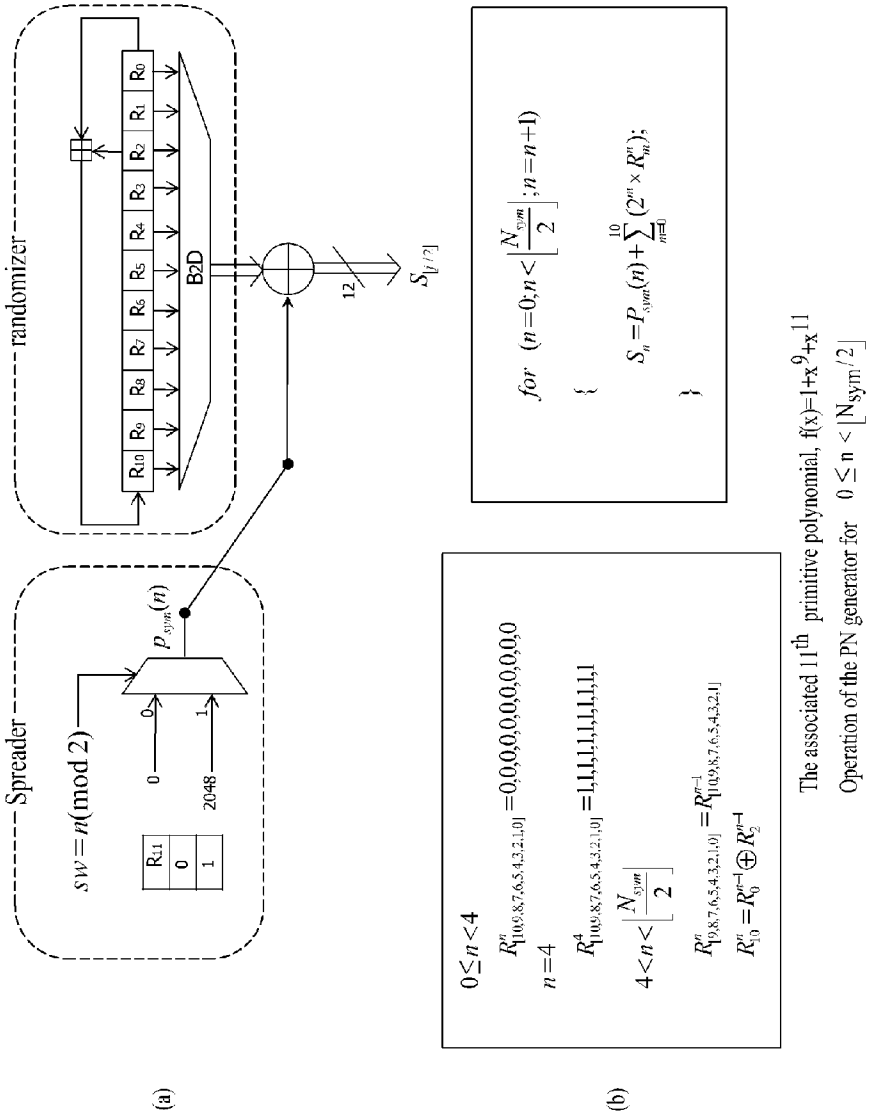
[Fig. 44]


randomizer
B2D
12
$S_{[j/2]}$
Spreader
$sw = n(\mathrm{mod}\ 2^k)$
$p_{sym}(n)$
MUX
Rendering a spreading effect between OFDM symbols

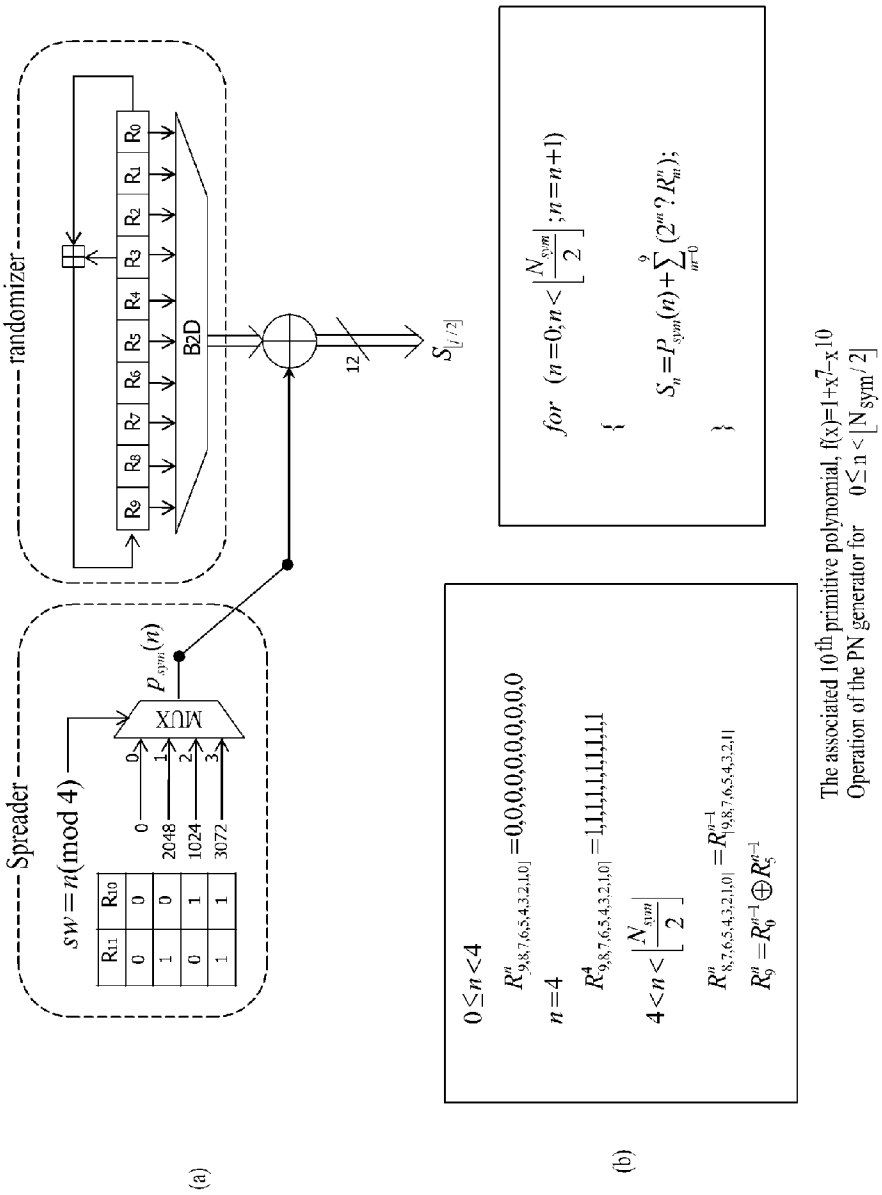
[Fig. 45]


The associated 12th primitive polynomial, f(x)=1+x6+x8+x11+x12
Operation of the PN generator for 0≤n<⌊Nsym/2⌋
0≤n<2
R0n[11,10,9,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,0,0,0
n=2
R02[11,10,9,8,7,6,5,4,3,2,1,0] =1,1,1,1,1,1,1,1,1,1,1,1
2<n<⌊Nsym/2⌋
R0n[10,9,8,7,6,5,4,3,2,1,0] =R0n-1[11,10,9,8,7,6,5,4,3,2,1]
R11n =R0n-1⊕R1n-1⊕R4n-1⊕R6n-1
for (n=0;n<⌊Nsym/2⌋;n=n+1)
{
Sn=∑m=011(2m×Rmn);
}
(b)
R11
R10
R9
R8
R7
R6
R5
R4
R3
R2
R1
R0
B2D
12
Sj/2
(a)

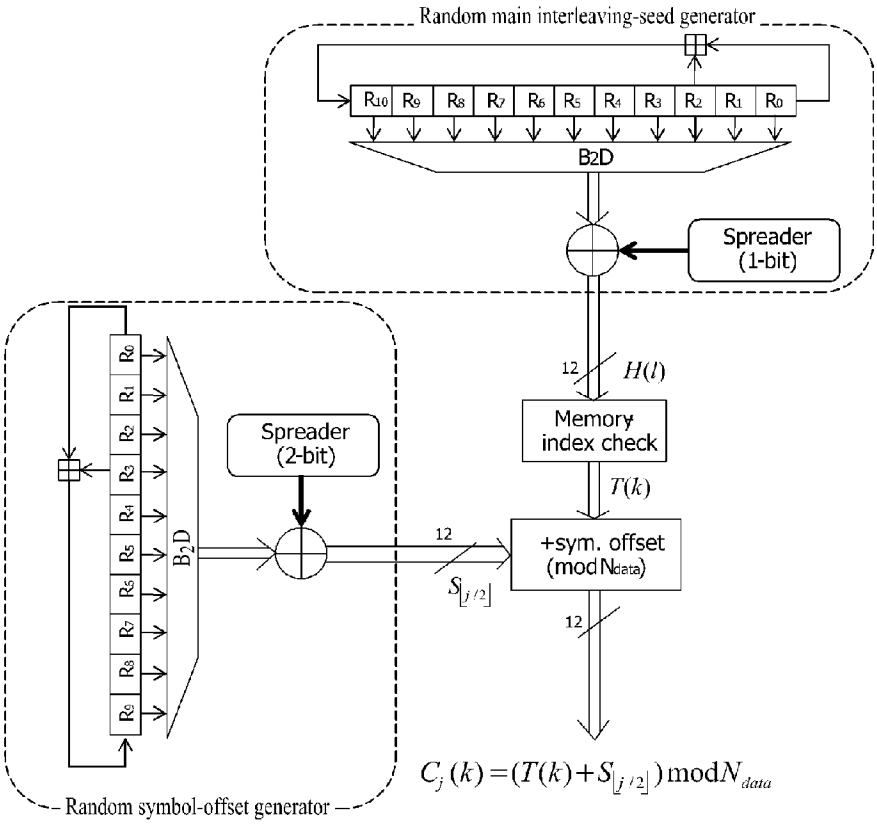
[Fig. 46]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
B2D
12
$S_{[j/2]}$
Spreader
$sw = n \pmod 2$
$p_{sym}(n)$
0
1
0
2048
R11
0
1
(a)
$0 \le n < 4$
$R^n_{[10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0$
$n = 4$
$R^4_{[10,9,8,7,6,5,4,3,2,1,0]} = 1,1,1,1,1,1,1,1,1,1,1$
$4 < n < \left\lfloor \frac{N_{sym}}{2} \right\rfloor$
$R^n_{[9,8,7,6,5,4,3,2,1,0]} = R^{n-1}_{[10,9,8,7,6,5,4,3,2,1]}$
$R^n_{10} = R^{n-1}_0 \oplus R^{n-1}_2$
The associated 11th primitive polynomial, $f(x) = 1 + x^9 + x^{11}$
Operation of the PN generator for $0 \le n < \lfloor N_{sym}/2 \rfloor$
$for \ (n = 0; n < \left\lfloor \frac{N_{sym}}{2} \right\rfloor; n = n + 1)$
{
$S_n = P_{sym}(n) + \sum_{n=1}^{10} (2^n \times R^n_n);$
}
(b)

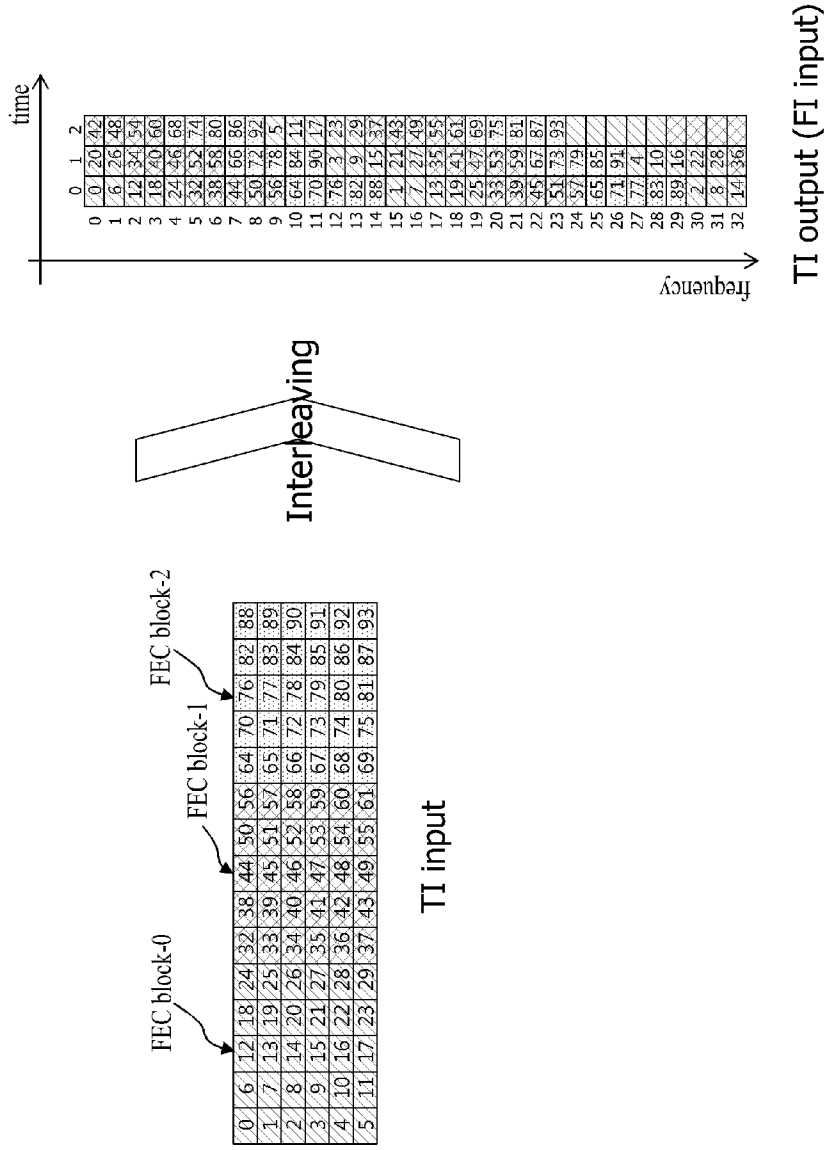
[Fig. 47]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
B2D
12
$S_{[/2]}$
Spreader
$sw = n \pmod 4$
$P_{sym}(n)$
MUX
0
2048
1024
3072
R11 R10
0 0
1 0
0 1
1 1
(a)
$0 \le n < 4$
$R^n_{9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0$
$n = 4$
$R^4_{9,8,7,6,5,4,3,2,1,0]} = 1,1,1,1,1,1,1,1,1,1$
$4 < n < \left\lfloor \frac{N_{sym}}{2} \right\rfloor$
$R^n_{8,7,6,5,4,3,2,1,0]} = R^{n-1}_{[9,8,7,6,5,4,3,2,1}$
$R^n_9 = R^{n-1}_0 \oplus R^{n-1}_5$
$for \quad (n=0; n < \left\lfloor \frac{N_{sym}}{2} \right\rfloor; n=n+1)$
{
$S_n = P_{sym}(n) + \sum_{m=0}^{9} (2^m ? R^n_m);$
}
(b)
The associated 10th primitive polynomial, f(x)=1+x7+x10
Operation of the PN generator for $0 \le n < \lfloor N_{sym}/2 \rfloor$

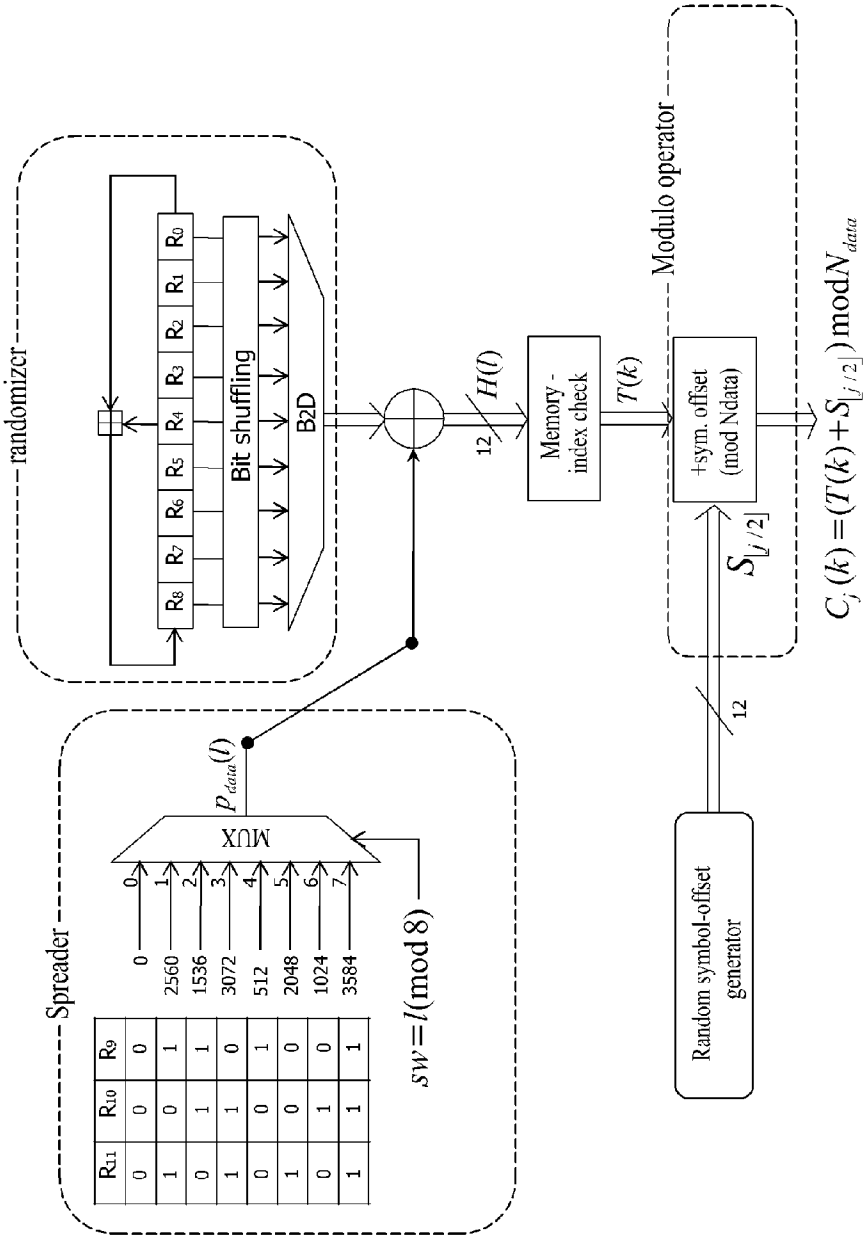
[Fig. 48]


Random main interleaving-seed generator
R10 R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
B2D
Spreader (1-bit)
12
H(l)
Memory index check
T(k)
+sym. offset (mod Ndata)
12
12
Spreader (2-bit)
B2D
R0 R1 R2 R3 R4 R5 R6 R7 R8 R9
S⌊j/2⌋
Random symbol-offset generator
$C_j(k) = (T(k) + S_{\lfloor j/2 \rfloor}) \bmod N_{data}$

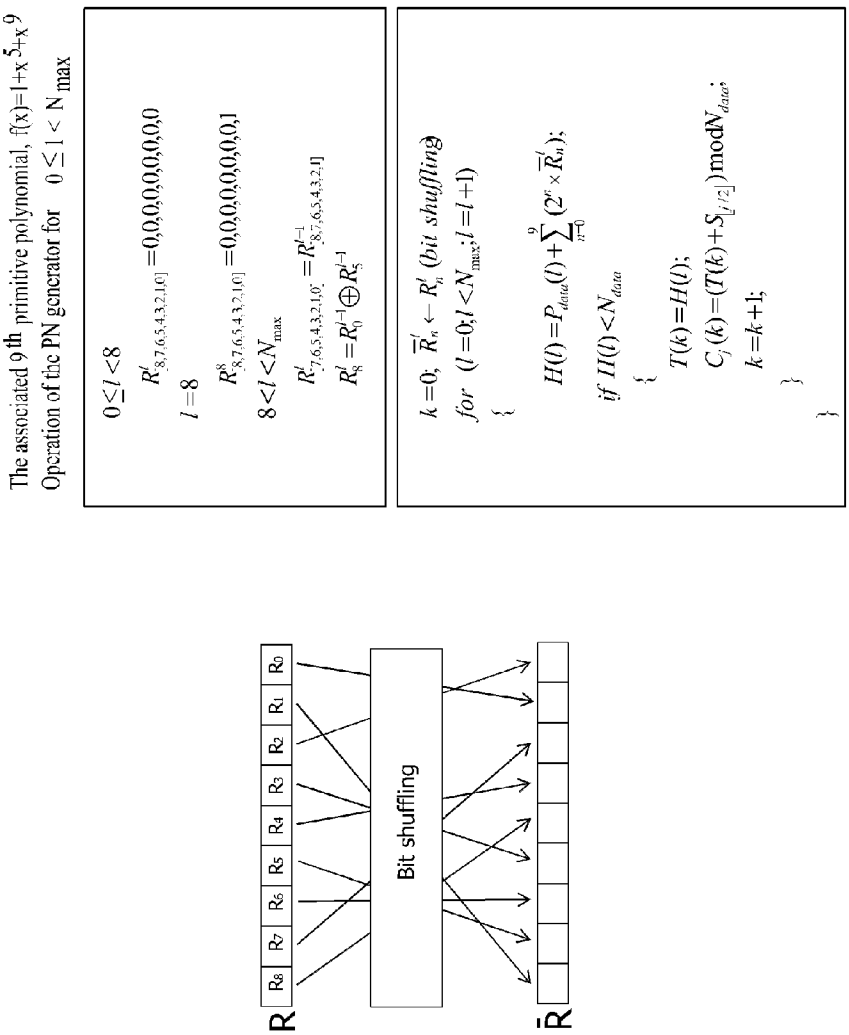
[Fig. 49]


time
frequency
TI output (FI input)
Interleaving
TI input
FEC block-0
FEC block-1
FEC block-2

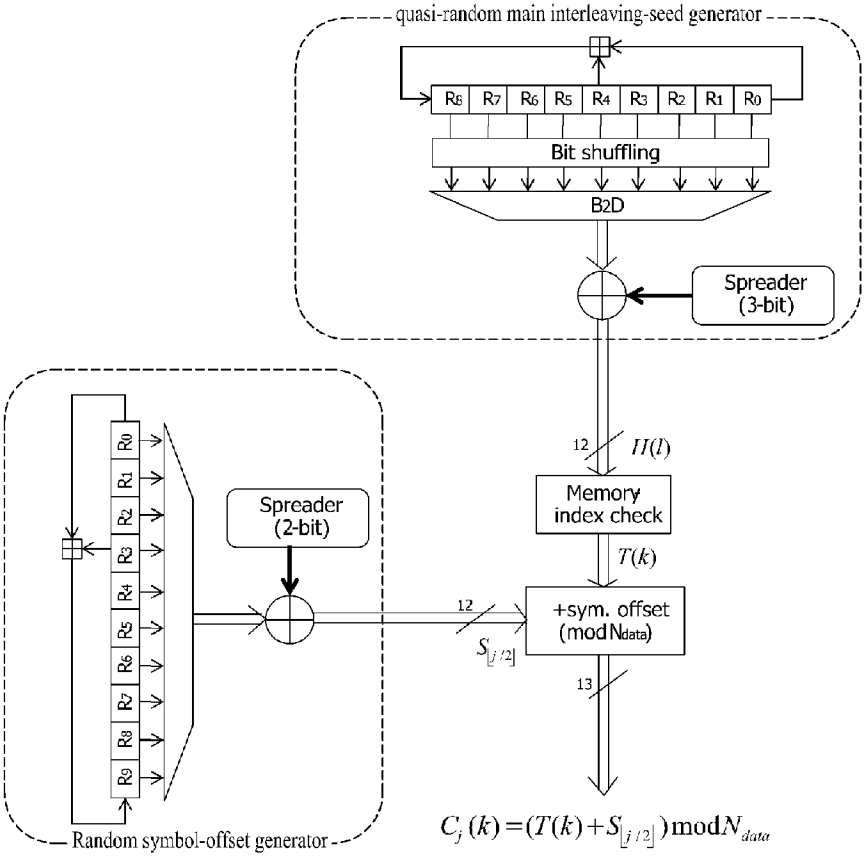
[Fig. 50]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
Bit shuffling
B2D
H(I)
12
Memory - index check
T(k)
Modulo operator
+sym. offset (mod Ndata)
$C_j(k) = (T(k) + S_{\lfloor j/2 \rfloor}) \bmod N_{data}$
$S_{\lfloor j/2 \rfloor}$
12
Random symbol-offset generator
Spreader
$P_{data}(I)$
MUX
0
1
2
3
4
5
6
7
0
2560
1536
3072
512
2048
1024
3584
$sw = I \pmod 8$
R11 R10 R9
0 0 0
1 0 1
0 1 1
1 1 0
0 0 1
1 0 0
0 1 0
1 1 1

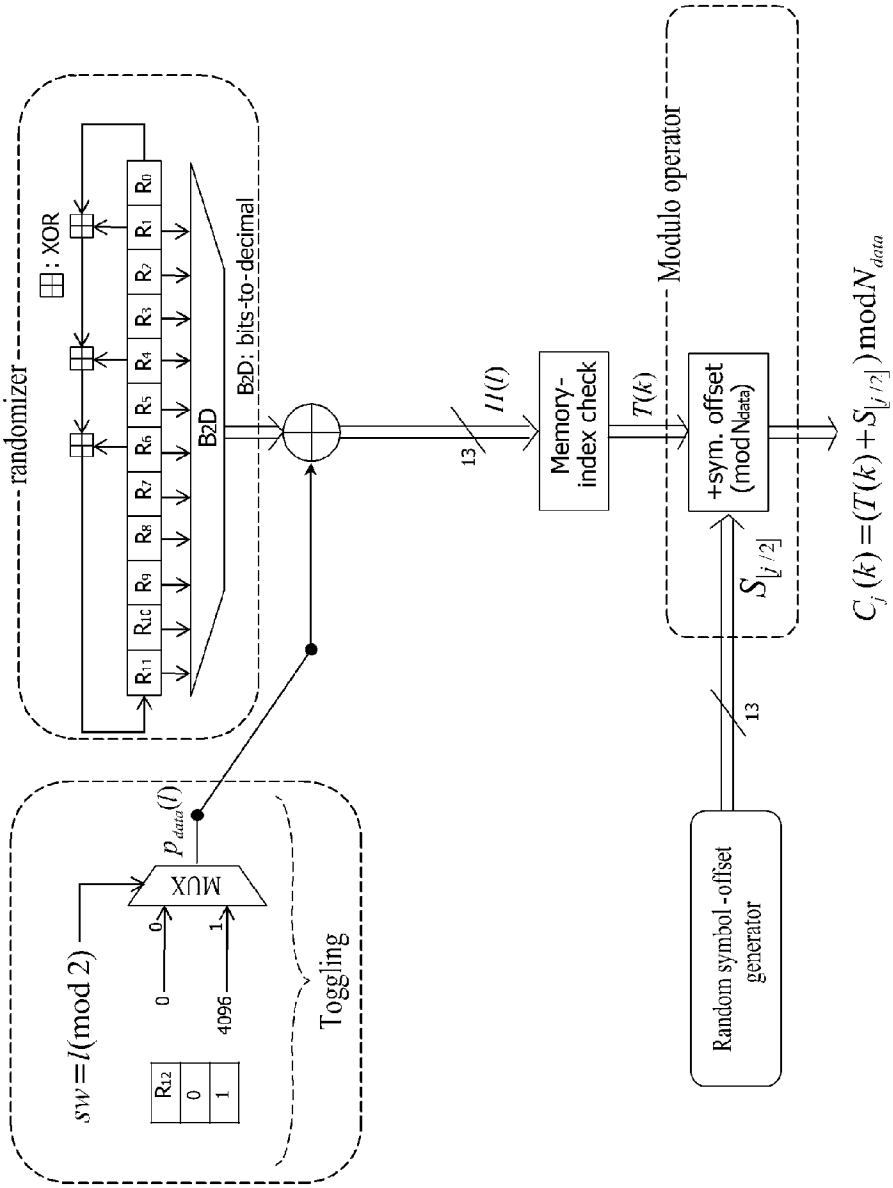
[Fig. 51]


The associated 9th primitive polynomial, f(x)=1+x^5+x^9
Operation of the PN generator for 0 ≤ l < N max
0≤l<8
$R^l_{8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0$
l=8
$R^8_{8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,1$
8<l<N_{max}
$R^l_{7,6,5,4,3,2,1,0,} = R^{l-1}_{8,7,6,5,4,3,2,1]}$
$R^l_8 = R^{l-1}_0 \oplus R^{l-1}_5$
k=0; $\bar{R}^l_n \leftarrow R^l_n$ (bit shuffling)
for (l=0;l<N_{max};l=l+1)
{
$H(l) = P_{data}(l) + \sum_{n=0}^{9} (2^n \times \bar{R}^l_n)$;
if H(l)<N_{data}
{
T(k)=H(l);
$C_j(k) = (T(k)+S_{\lfloor j/2 \rfloor}) \bmod N_{data}$;
k=k+1;
}
}
(b)
R
R8
R7
R6
R5
R4
R3
R2
R1
R0
Bit shuffling
$\bar{R}$
(a)

[Fig. 52]


quasi-random main interleaving-seed generator
R8 R7 R6 R5 R4 R3 R2 R1 R0
Bit shuffling
B2D
Spreader (3-bit)
12
Π(l)
Memory index check
T(k)
+sym. offset (mod Ndata)
12
S⌊j/2⌋
13
Spreader (2-bit)
R0 R1 R2 R3 R4 R5 R6 R7 R8 R9
Random symbol-offset generator
$C_j(k) = (T(k) + S_{\lfloor j/2 \rfloor}) \bmod N_{data}$

[Fig. 53]


randomizer
⊞: XOR
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
B2D
B2D: bits-to-decimal
13
I1(l)
Memory-index check
T(k)
Modulo operator
+sym. offset (mod Ndata)
$S_{\lfloor l/2 \rfloor}$
$C_f(k)=(T(k)+S_{\lfloor l/2 \rfloor}) \bmod N_{data}$
13
Random symbol-offset generator
$p_{data}(l)$
MUX
0
1
0
4096
$sw=l \pmod 2$
R12
0
1
Toggling

[Fig. 54]

The associated 12th primitive polynomial, $f(x)=1+x^6+x^8+x^{11}+x^{12}$
Operation of the PN generator for $0 \le l < N_{max}\,(8192)$

$0 \le l < 2$

$R^{l}_{[11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0,0$

$l = 2$

$R^{2}_{[11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,1,0,1,0,0,1,1$

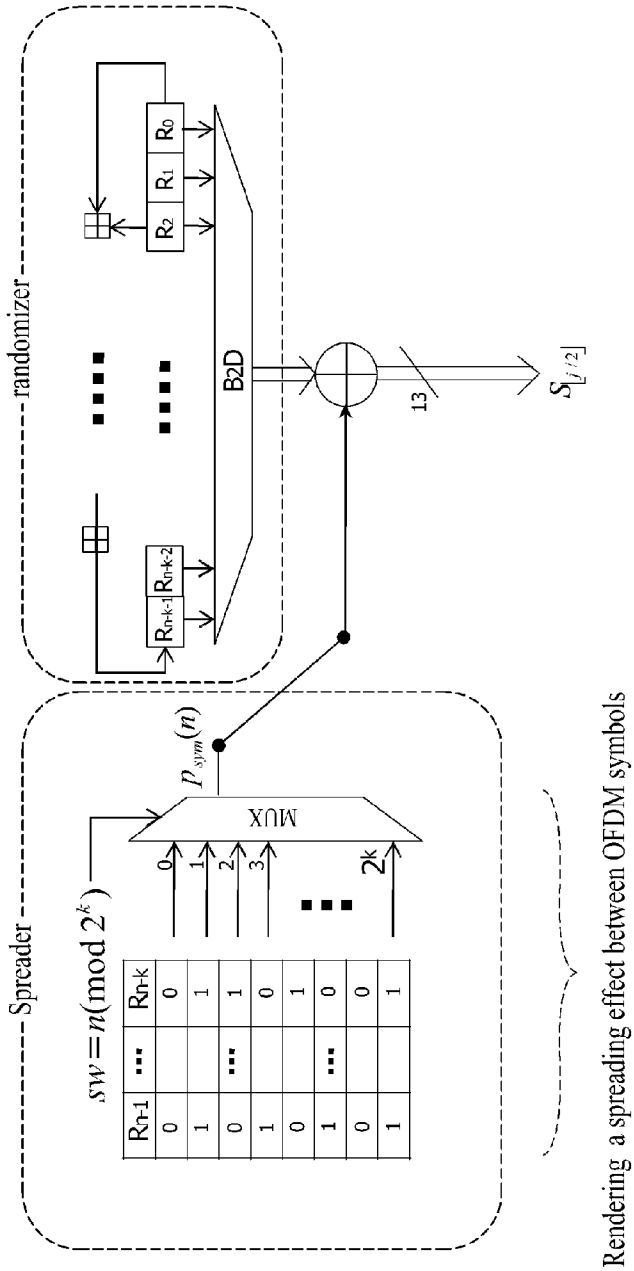
$2 < l < N_{max}$

$R^{l}_{[10,9,8,7,6,5,4,3,2,1,0]} = R^{l-1}_{[11,10,9,8,7,6,5,4,3,2,1]}$

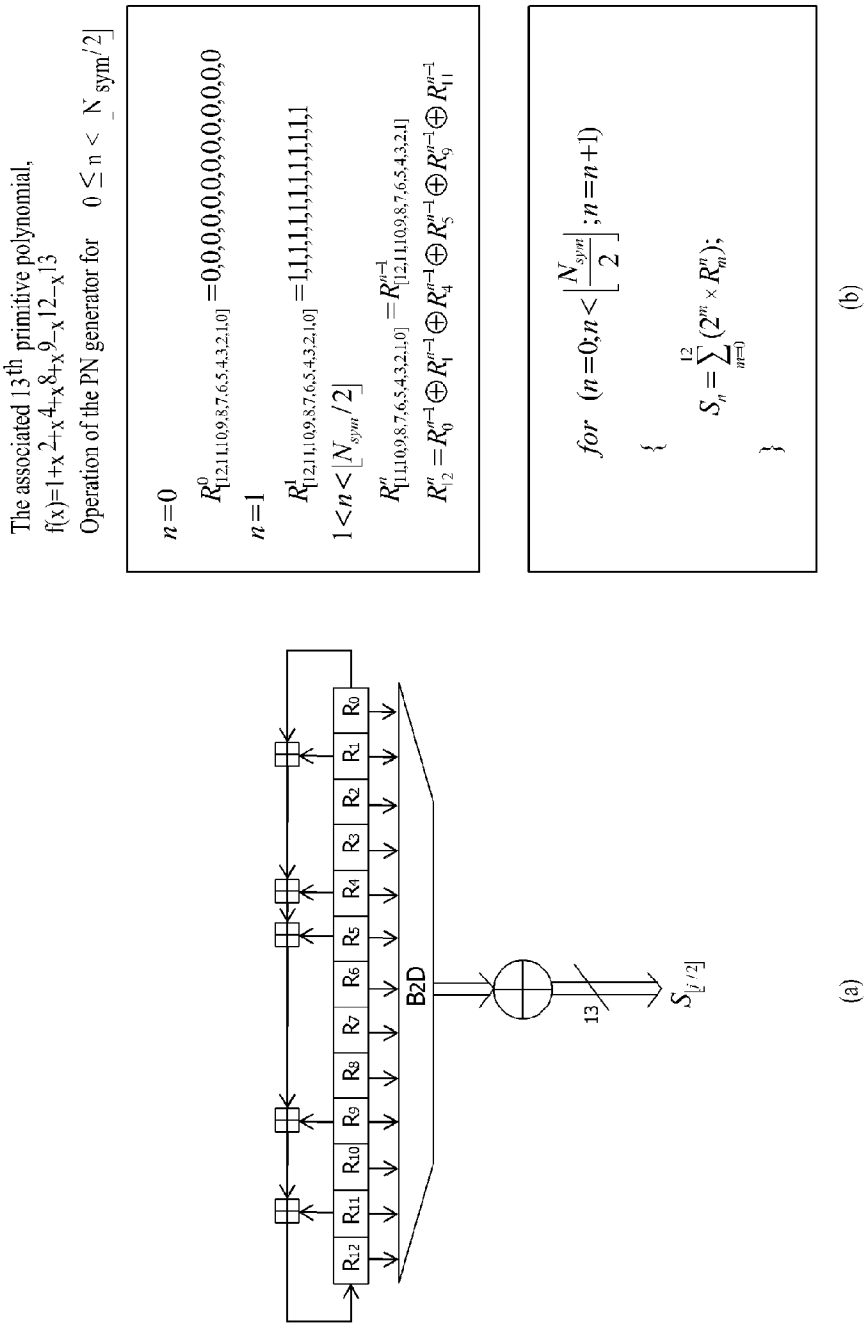
$R^{l}_{11} = R^{l-1}_{0} \oplus R^{l-1}_{1} \oplus R^{l-1}_{4} \oplus R^{l-1}_{6}$

```
k = 0;
for (l = 0; l < N_max; l = l + 1)
{
    H(l) = P_data(l) + Σ_{n=0}^{11} (2^n × R_n^l);
    if H(l) < N_data
    {
        T(k) = H(l);
        C_j(k) = (T(k) + S_⌊j/2⌋) mod N_data;
        k = k + 1;
    }
}
```

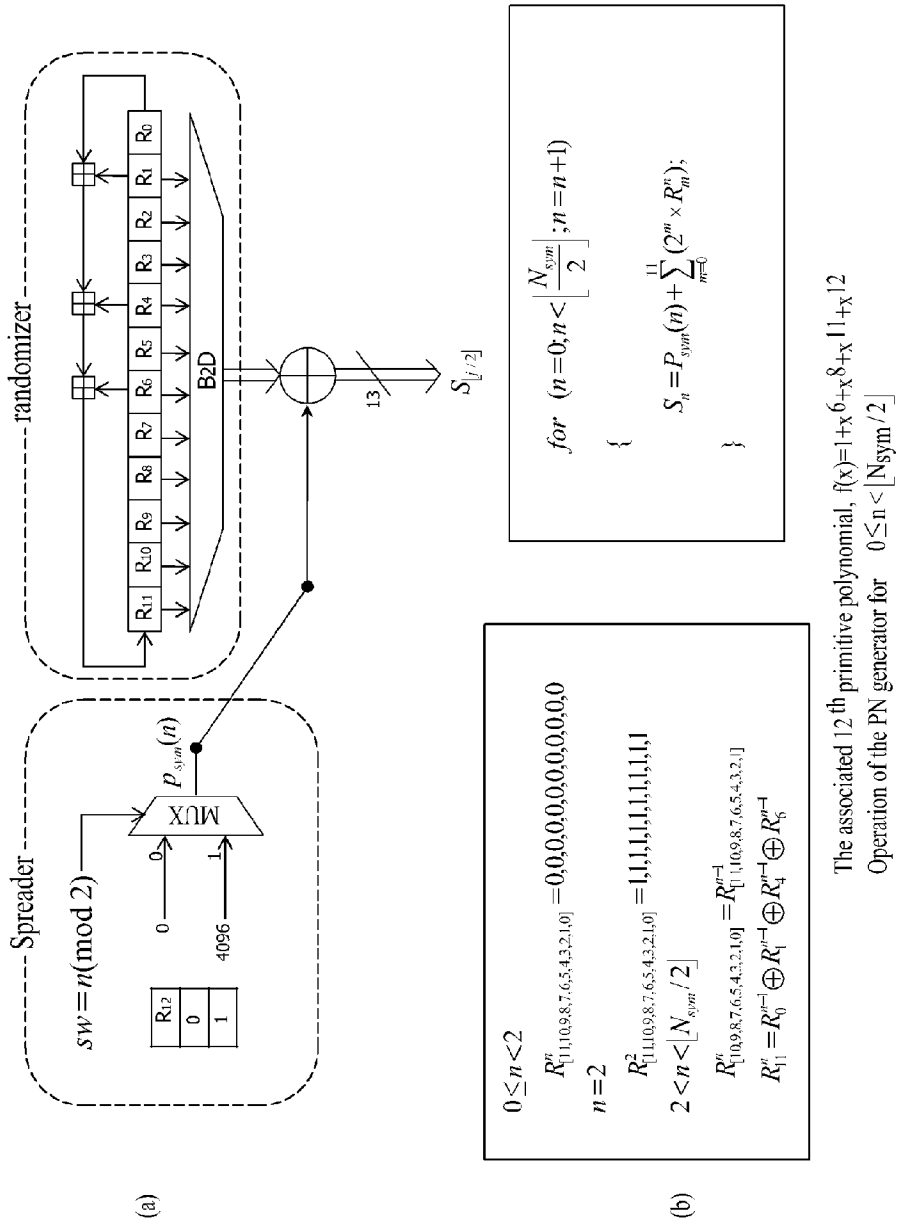
[Fig. 55]


randomizer
R0
R1
R2
Rn+k-1
Rn+k-2
B2D
13
S[j/2]
Spreader
$sw = n(\mathrm{mod}\ 2^k)$
Rn-1
Rn-k
MUX
$p_{sym}(n)$
0
1
2
3
2k
Rendering a spreading effect between OFDM symbols

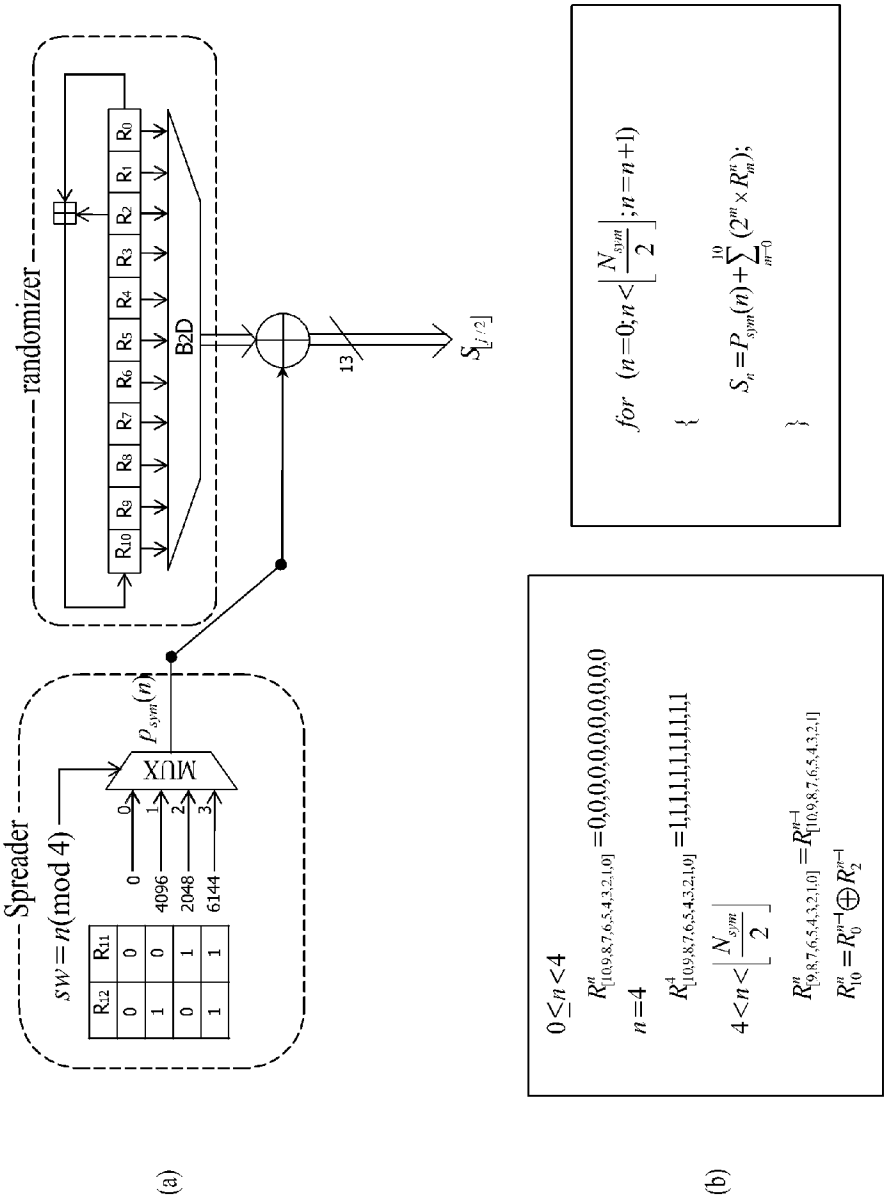
[Fig. 56]


The associated 13th primitive polynomial,
f(x)=1+x^2+x^4+x^8+x^9+x^12+x^13
Operation of the PN generator for 0 ≤ n < N_sym/2
n=0
R^0_[12,11,10,9,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,0,0,0,0
n=1
R^1_[12,11,10,9,8,7,6,5,4,3,2,1,0] =1,1,1,1,1,1,1,1,1,1,1,1,1
1<n<⌊N_sym/2⌋
R^n_[11,10,9,8,7,6,5,4,3,2,1,0] = R^{n-1}_[12,11,10,9,8,7,6,5,4,3,2,1]
R^n_12 = R^{n-1}_0 ⊕ R^{n-1}_1 ⊕ R^{n-1}_4 ⊕ R^{n-1}_5 ⊕ R^{n-1}_9 ⊕ R^{n-1}_11
for (n=0;n<⌊N_sym/2⌋;n=n+1)
{
S_n = Σ_{m=0}^{12} (2^m × R^n_m);
}
(b)
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
B2D
13
S_⌊j/2⌋
(a)

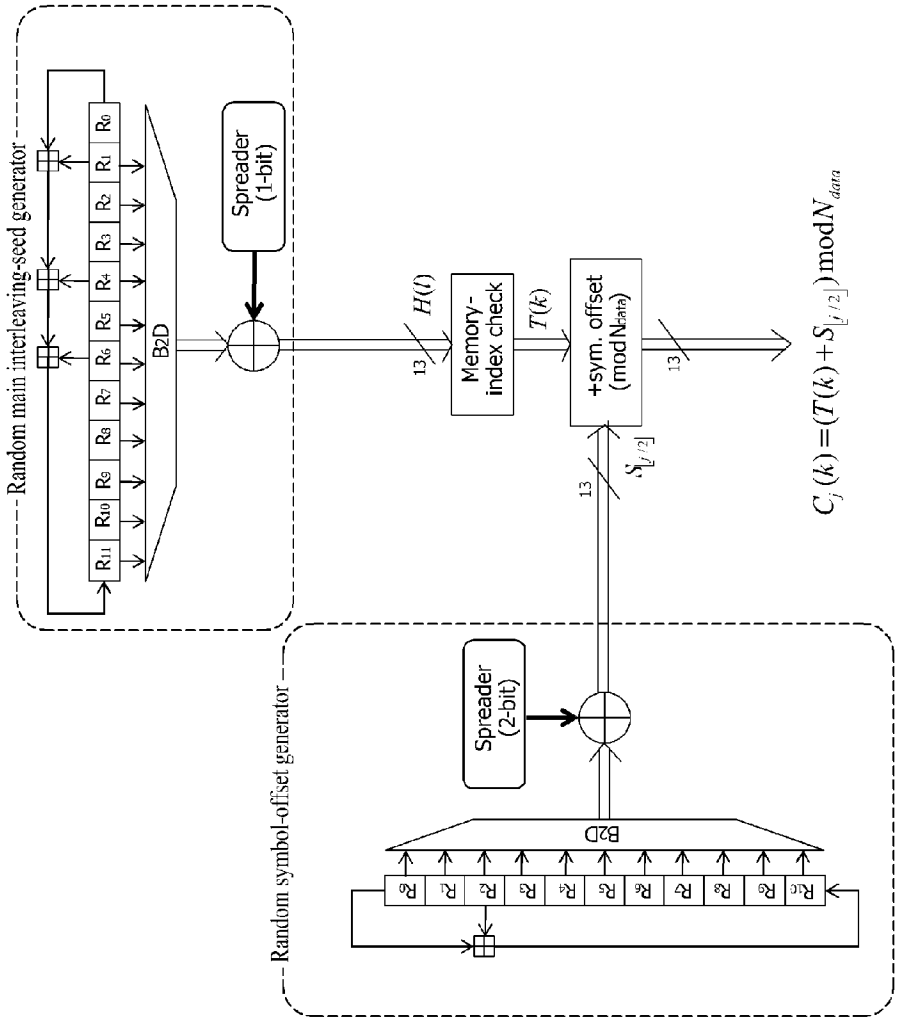
[Fig. 57]


randomizer
Spreader
$sw=n(\mathrm{mod}\ 2)$
MUX
$p_{sym}(n)$
0
4096
R12
B2D
13
$S_{[i/2]}$
(a)
$0\le n<2$
$R^n_{[11,10,9,8,7,6,5,4,3,2,1,0]}=0,0,0,0,0,0,0,0,0,0,0,0$
$n=2$
$R^2_{[11,10,9,8,7,6,5,4,3,2,1,0]}=1,1,1,1,1,1,1,1,1,1,1,1$
$2<n<\lfloor N_{sym}/2\rfloor$
$R^n_{[10,9,8,7,6,5,4,3,2,1,0]}=R^{n-1}_{[11,10,9,8,7,6,5,4,3,2,1]}$
$R^n_{11}=R^{n-1}_0\oplus R^{n-1}_1\oplus R^{n-1}_4\oplus R^{n-1}_5$
$for\ \ (n=0,n<\left\lfloor\frac{N_{sym}}{2}\right\rfloor;n=n+1)$
{
$S_n=P_{sym}(n)+\sum_{m=0}^{11}(2^m\times R^n_m);$
}
(b)
The associated 12 th primitive polynomial, $f(x)=1+x^6+x^8+x^{11}+x^{12}$
Operation of the PN generator for $0\le n<\lfloor \mathrm{Nsym}/2\rfloor$

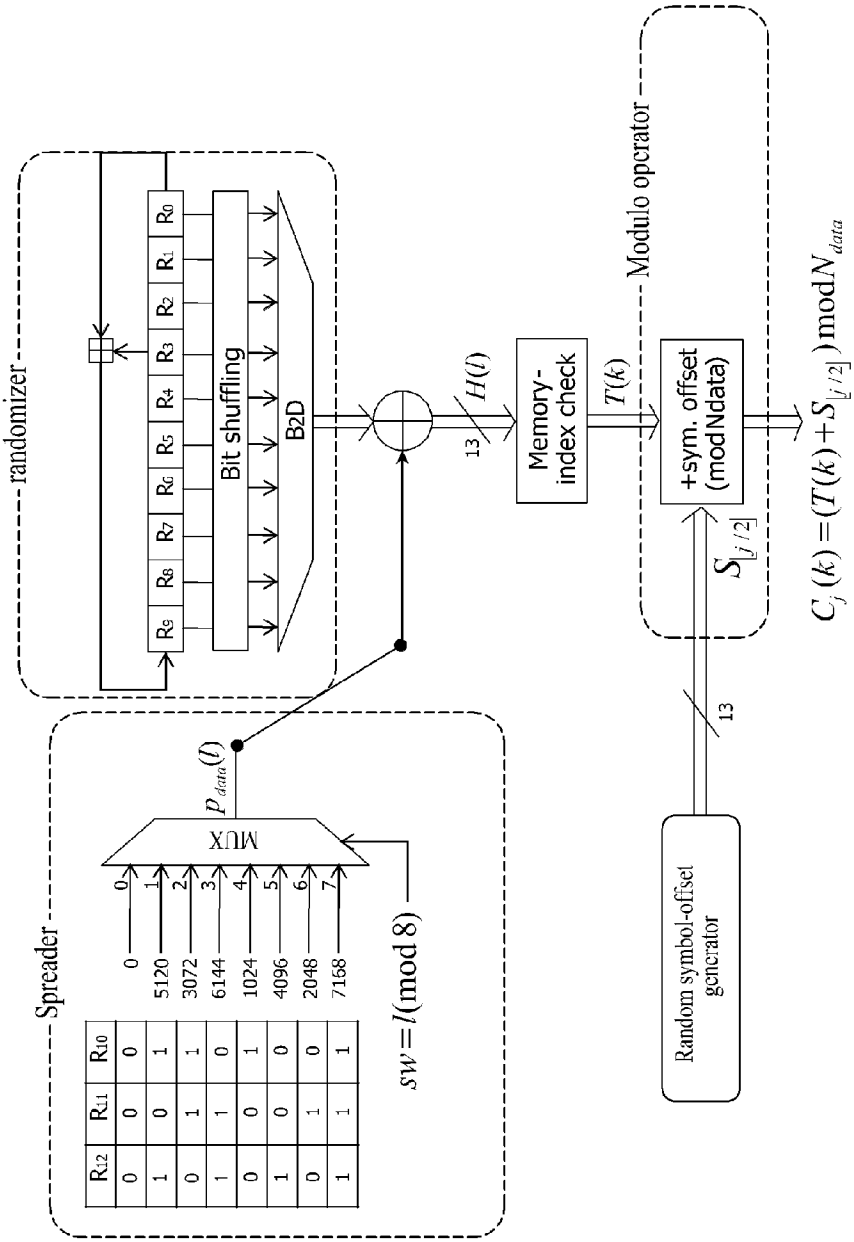
[Fig. 58]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
B2D
13
$S_{[i/2]}$
Spreader
$sw = n \pmod 4$
R12 R11
0 0
1 0
0 1
1 1
MUX
0
1
2
3
0
4096
2048
6144
$\rho_{sym}(n)$
(a)
$0 \le n < 4$
$R^n_{[10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0$
$n = 4$
$R^4_{[10,9,8,7,6,5,4,3,2,1,0]} = 1,1,1,1,1,1,1,1,1,1,1$
$4 < n < \left\lfloor \frac{N_{sym}}{2} \right\rfloor$
$R^n_{[9,8,7,6,5,4,3,2,1,0]} = R^{n-1}_{[10,9,8,7,6,5,4,3,2,1]}$
$R^n_{10} = R^{n-1}_0 \oplus R^{n-1}_2$
$for \quad (n=0; n < \left\lfloor \frac{N_{sym}}{2} \right\rfloor; n=n+1)$
{
$S_n = P_{sym}(n) + \sum_{m=0}^{10} (2^m \times R^n_m);$
}
(b)

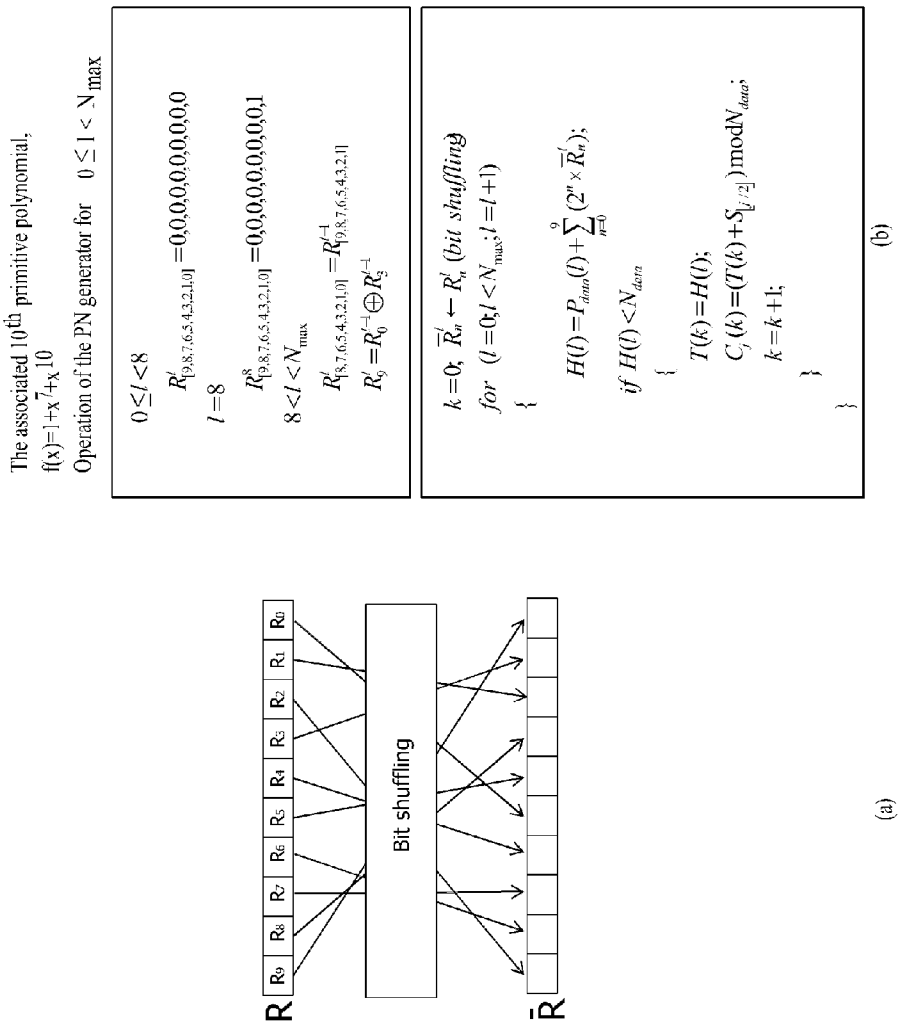
[Fig. 59]


Random main interleaving-seed generator
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
B2D
Spreader (1-bit)
13
H(l)
Memory-index check
T(k)
+sym. offset (mod Ndata)
13
13
S[l/2]
Random symbol-offset generator
Spreader (2-bit)
B2D
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
$C_j(k) = (T(k) + S_{[l/2]}) \bmod N_{data}$

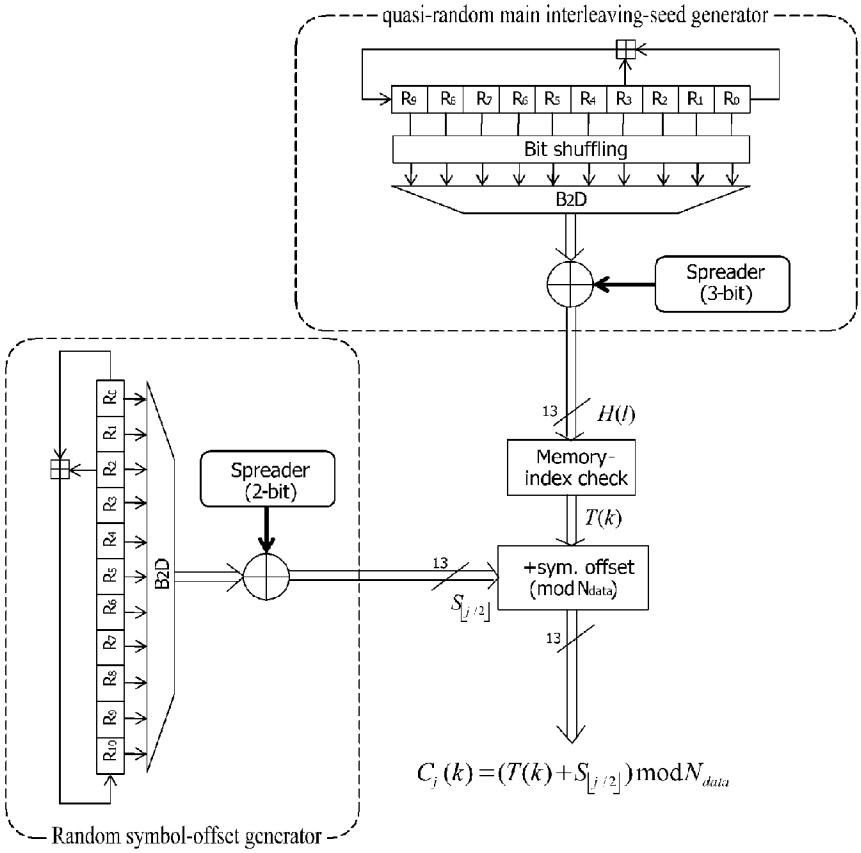
[Fig. 60]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
Bit shuffling
B2D
13
H(I)
Memory-index check
T(k)
Modulo oprator
+sym. offset (mod Ndata)
S[j/2]
13
Random symbol-offset gencrator
Spreader
MUX
P data (I)
sw=I(mod 8)
0
1
2
3
4
5
6
7
0
5120
3072
6144
1024
4096
2048
7168
R12 R11 R10
0 0 0
1 0 1
0 1 1
1 1 0
0 0 1
1 0 0
0 1 0
1 1 1

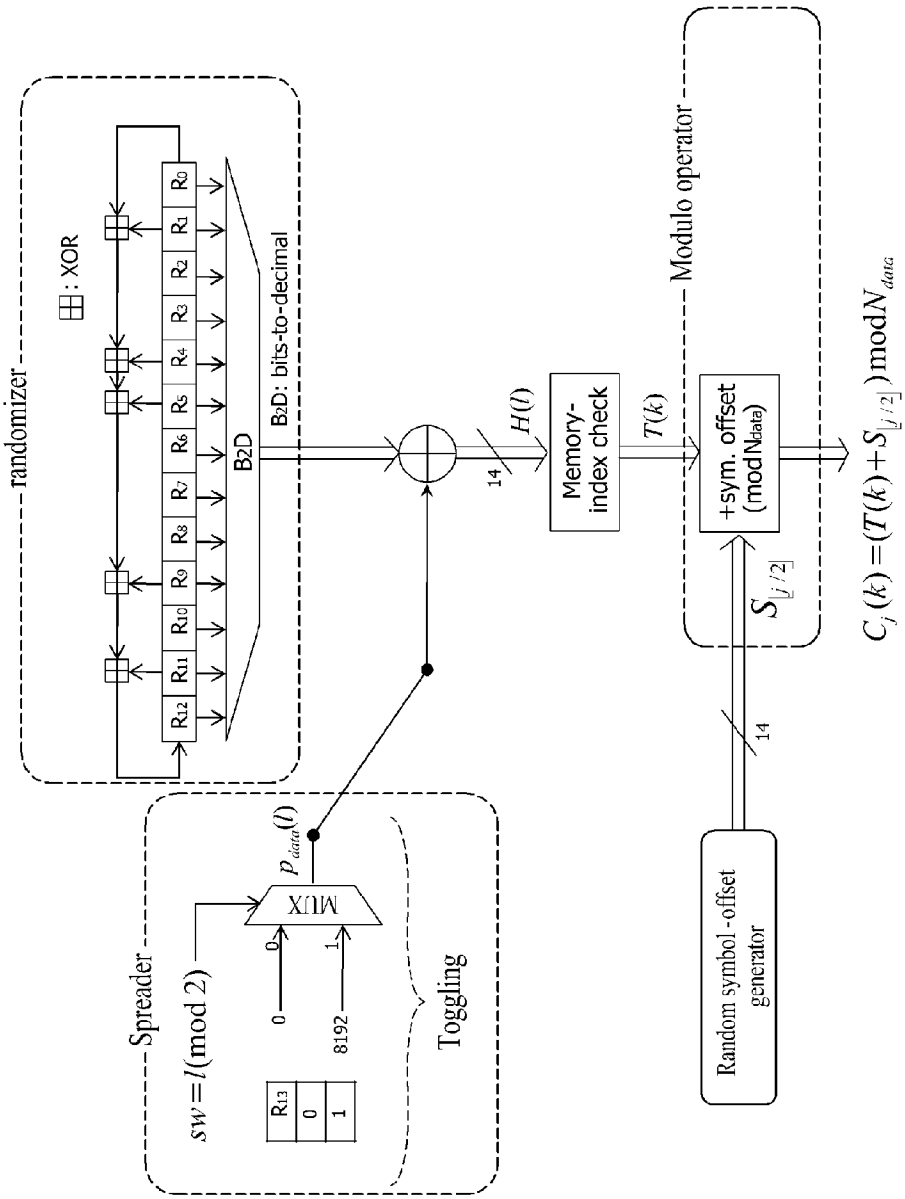
[Fig. 61]


The associated 10th primitive polynomial,
f(x)=1+x^7+x^10
Operation of the PN generator for 0 ≤ l < N max
0≤l<8
R^l_[9,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,0
l=8
R^8_[9,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,1
8<l<N_max
R^l_[8,7,6,5,4,3,2,1,0] =R^(l-1)_[9,8,7,6,5,4,3,2,1]
R^l_9 =R^(l-1)_0 ⊕ R^(l-1)_2
k=0; R̄^l_n ← R^l_n (bit shuffling)
for (l=0;l<N_max;l=l+1)
{
H(l)=P_data(l)+Σ_{n=0}^{9} (2^n × R̄^l_n);
if H(l)<N_data
{
T(k)=H(l);
C_j (k)=(T(k)+S_[j/2]) mod N_data;
k=k+1;
}
}
(b)
R
R9
R8
R7
R6
R5
R4
R3
R2
R1
R0
Bit shuffling
R̄
(a)

[Fig. 62]


quasi-random main interleaving-seed generator
R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
Bit shuffling
B2D
Spreader (3-bit)
13
H(l)
Memory-index check
T(k)
13
+sym. offset (mod Ndata)
S[j/2]
13
Spreader (2-bit)
B2D
R0 R1 R2 R3 R4 R5 R6 R7 R8 R9 R10
Random symbol-offset generator
$C_j(k) = (T(k) + S_{\lfloor j/2 \rfloor}) \bmod N_{data}$

[Fig. 63]


randomizer
⊞: XOR
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
B2D
B2D: bits-to-decimal
14
H(l)
Memory-index check
T(k)
Modulo operator
+sym. offset (mod Ndata)
$C_j(k)=(T(k)+S_{\lfloor j/2 \rfloor}) \bmod N_{data}$
$S_{\lfloor j/2 \rfloor}$
14
Random symbol-offset generator
Spreader
$sw=l \pmod 2$
$p_{data}(l)$
MUX
0
1
0
8192
R13
0
1
Toggling

[Fig. 64]

The associated 13th primitive polynomial, $f(x)=1+x^2+x^4+x^8+x^9+x^{12}+x^{13}$
Operation of the PN generator for $0 \le l < N_{max}(16384)$

$$0 \le l < 2$$

$$R^l_{[12,11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0,0,0$$

$$l = 2$$

$$R^2_{[12,11,10,9,8,7,6,5,4,3,2,1,0]} = 0,1,0,1,0,0,0,1,1,0,0,1,1$$

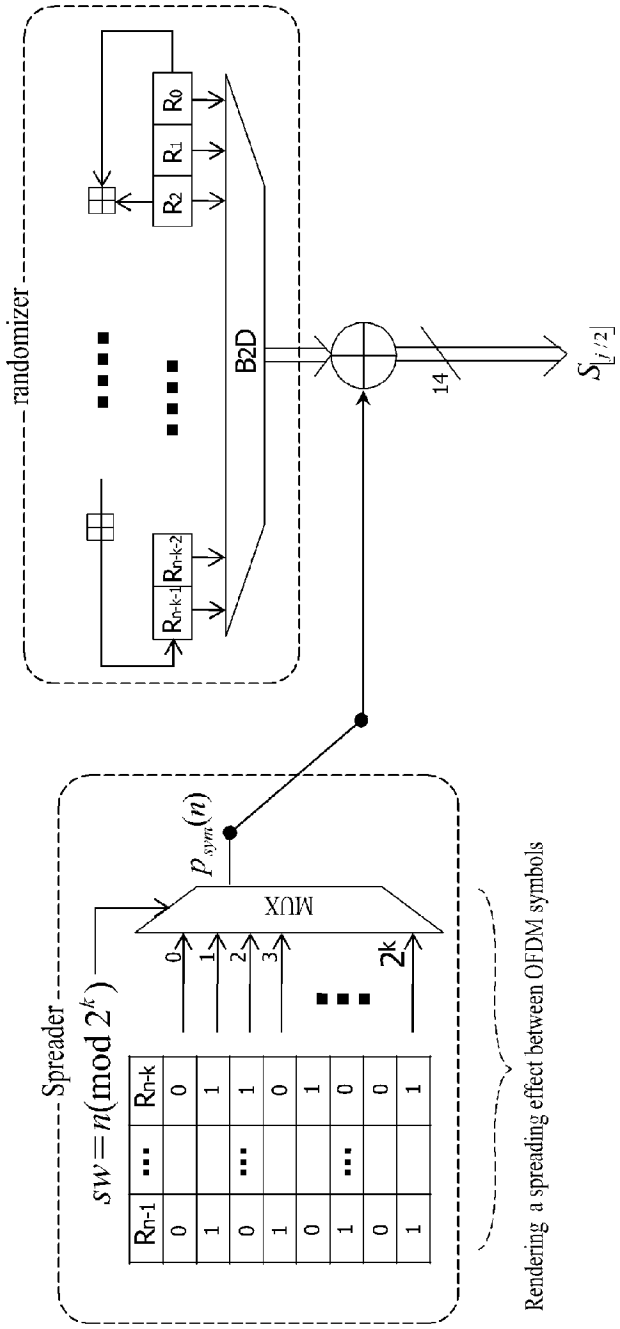
$$2 < l < N_{max}$$

$$R^l_{[11,10,9,8,7,6,5,4,3,2,1,0]} = R^{l-1}_{[12,11,10,9,8,7,6,5,4,3,2,1]}$$

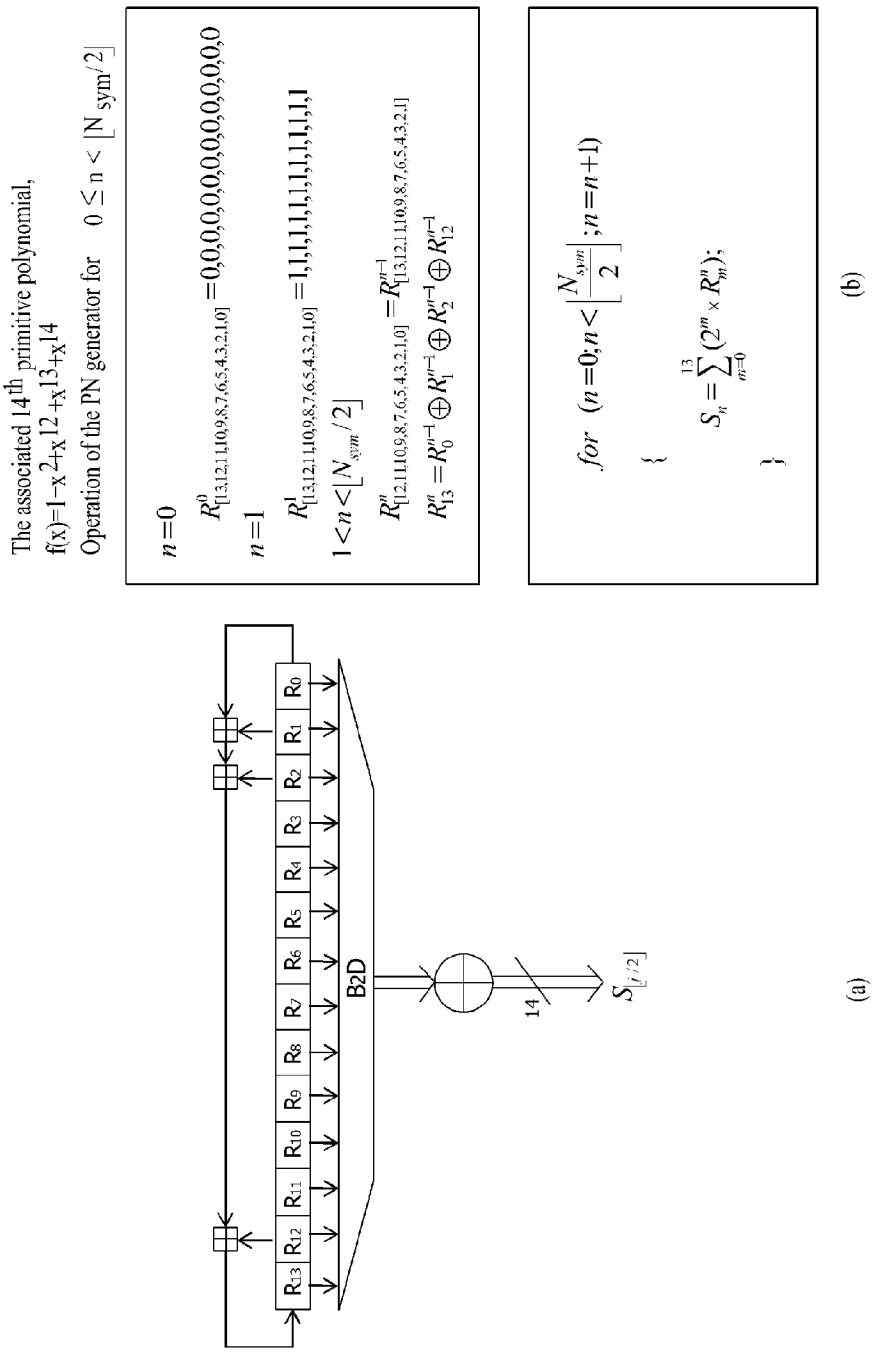
$$R^l_{12} = R^{l-1}_0 \oplus R^{l-1}_1 \oplus R^{l-1}_4 \oplus R^{l-1}_5 \oplus R^{l-1}_9 \oplus R^{l-1}_{11}$$

```
k = 0;
for (l = 0; l < N_max; l = l + 1)
{
    H(l) = P_data(l) + Σ_{n=0}^{12} (2^n × R_n^l);
    if H(l) < N_data
    {
        T(k) = H(l);
        C_j(k) = (T(k) + S_⌊j/2⌋) mod N_data;
        k = k + 1;
    }
}
```

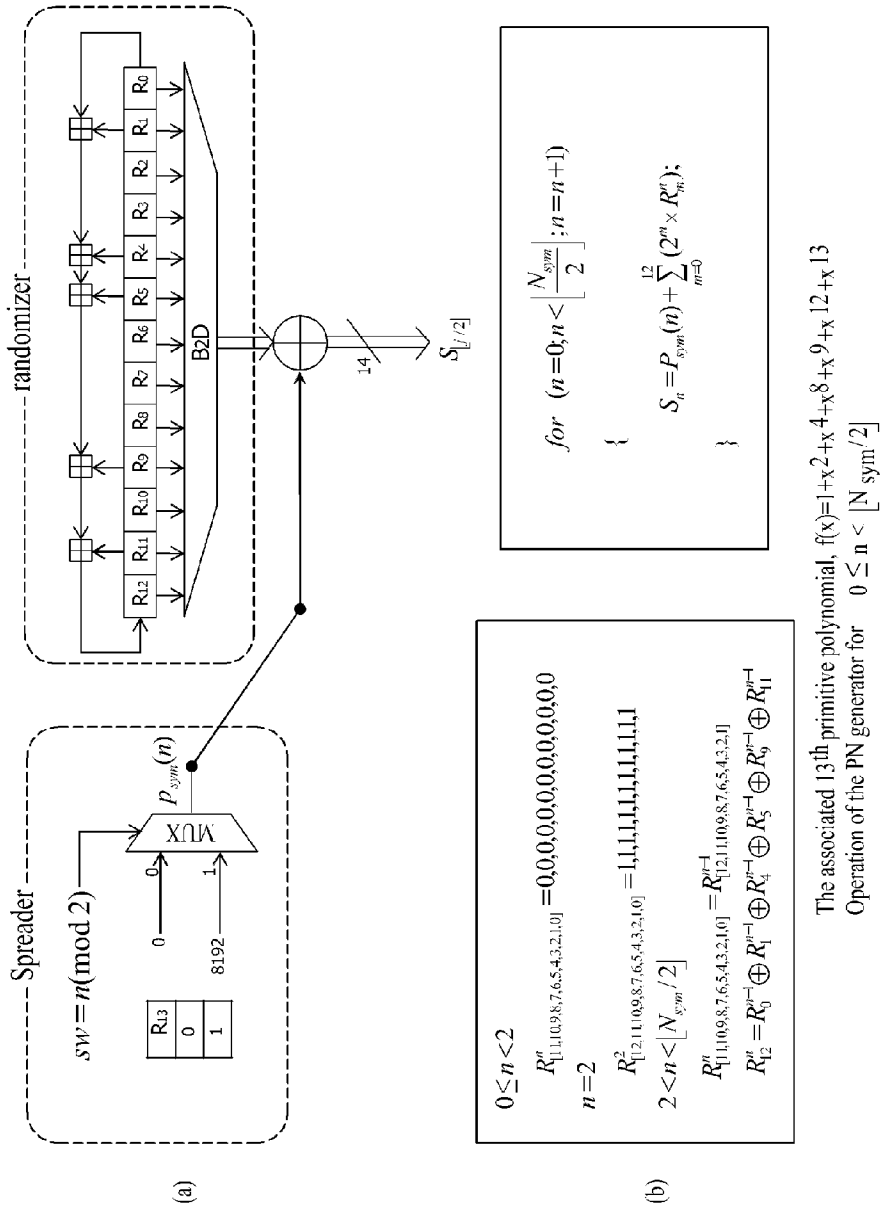
[Fig. 65]


randomizer
R0
R1
R2
Rn-k-1
Rn-k-2
B2D
14
$S_{[i/2]}$
Spreader
$sw = n \pmod{2^k}$
Rn-1
Rn-k
MUX
$p_{sym}(n)$
0
1
2
3
2k
Rendering a spreading effect between OFDM symbols

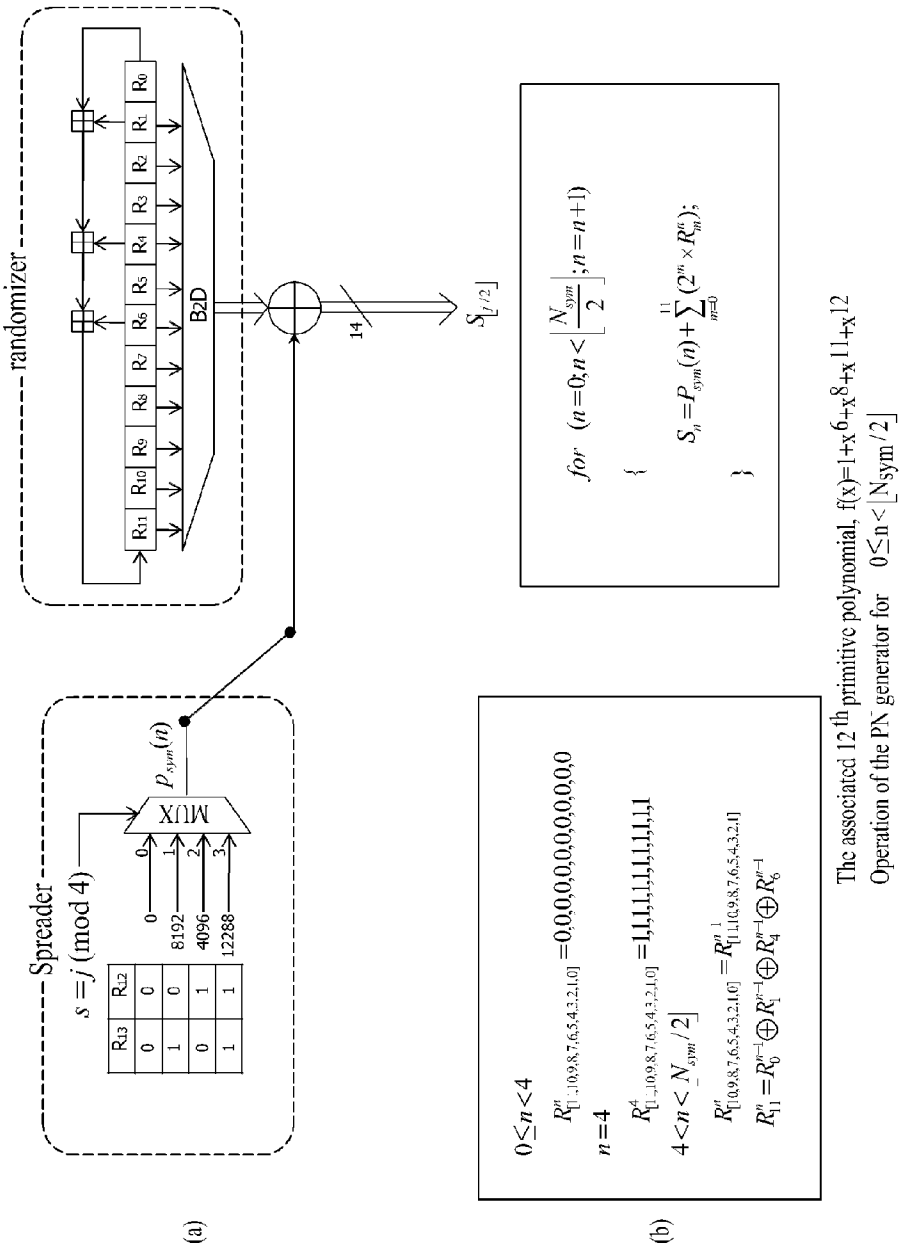
[Fig. 66]


The associated 14th primitive polynomial,
f(x)=1+x2+x12+x13+x14
Operation of the PN generator for 0 ≤ n < ⌊N sym/2⌋
n=0
R0[13,12,11,10,9,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,0,0,0,0,0
n=1
R1[13,12,11,10,9,8,7,6,5,4,3,2,1,0] =1,1,1,1,1,1,1,1,1,1,1,1,1,1
1<n<⌊Nsym/2⌋
Rn[12,11,10,9,8,7,6,5,4,3,2,1,0] =Rn-1[13,12,11,10,9,8,7,6,5,4,3,2,1]
R13n =R0n-1 ⊕ R1n-1 ⊕ R2n-1 ⊕ R12n-1
for (n=0;n<⌊Nsym/2⌋;n=n+1)
{
Sn=∑m=013(2m × Rmn);
}
(b)
R13 R12 R11 R10 R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
B2D
14
S⌊j/2⌋
(a)

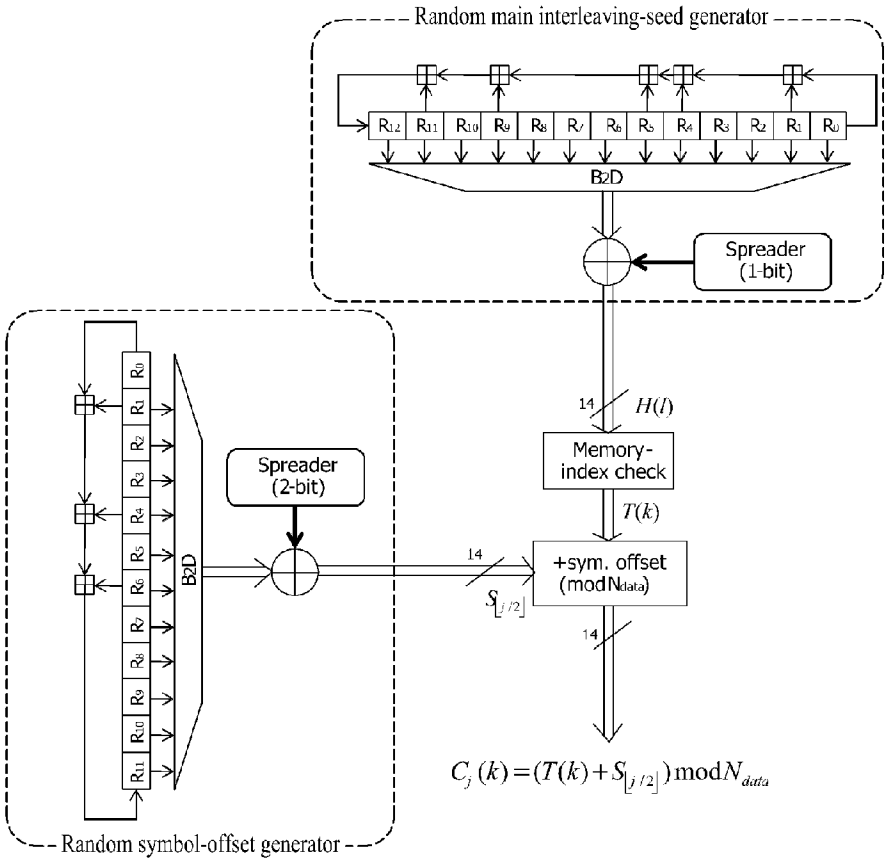
[Fig. 67]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
B2D
14
$S_{[j/2]}$
Spreader
$sw = n \pmod 2$
MUX
$p_{sym}(n)$
0
1
0
8192
R13
0
1
(a)
$0 \le n < 2$
$R^n_{[11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0,0,0$
$n = 2$
$R^2_{[12,11,10,9,8,7,6,5,4,3,2,1,0]} = 1,1,1,1,1,1,1,1,1,1,1,1,1$
$2 < n < \lfloor N_{sym}/2 \rfloor$
$R^n_{[11,10,9,8,7,6,5,4,3,2,1,0]} = R^{n-1}_{[12,11,10,9,8,7,6,5,4,3,2,1]}$
$R^n_{12} = R^{n-1}_0 \oplus R^{n-1}_1 \oplus R^{n-1}_4 \oplus R^{n-1}_5 \oplus R^{n-1}_9 \oplus R^{n-1}_{11}$
$for \quad (n=0; n < \left\lfloor \frac{N_{sym}}{2} \right\rfloor ; n=n+1)$
{
$S_n = P_{sym}(n) + \sum_{m=0}^{12} (2^m \times R^n_m);$
}
The associated 13th primitive polynomial, $f(x) = 1 + x^2 + x^4 + x^8 + x^9 + x^{12} + x^{13}$
Operation of the PN generator for $0 \le n < \lfloor N_{sym}/2 \rfloor$
(b)

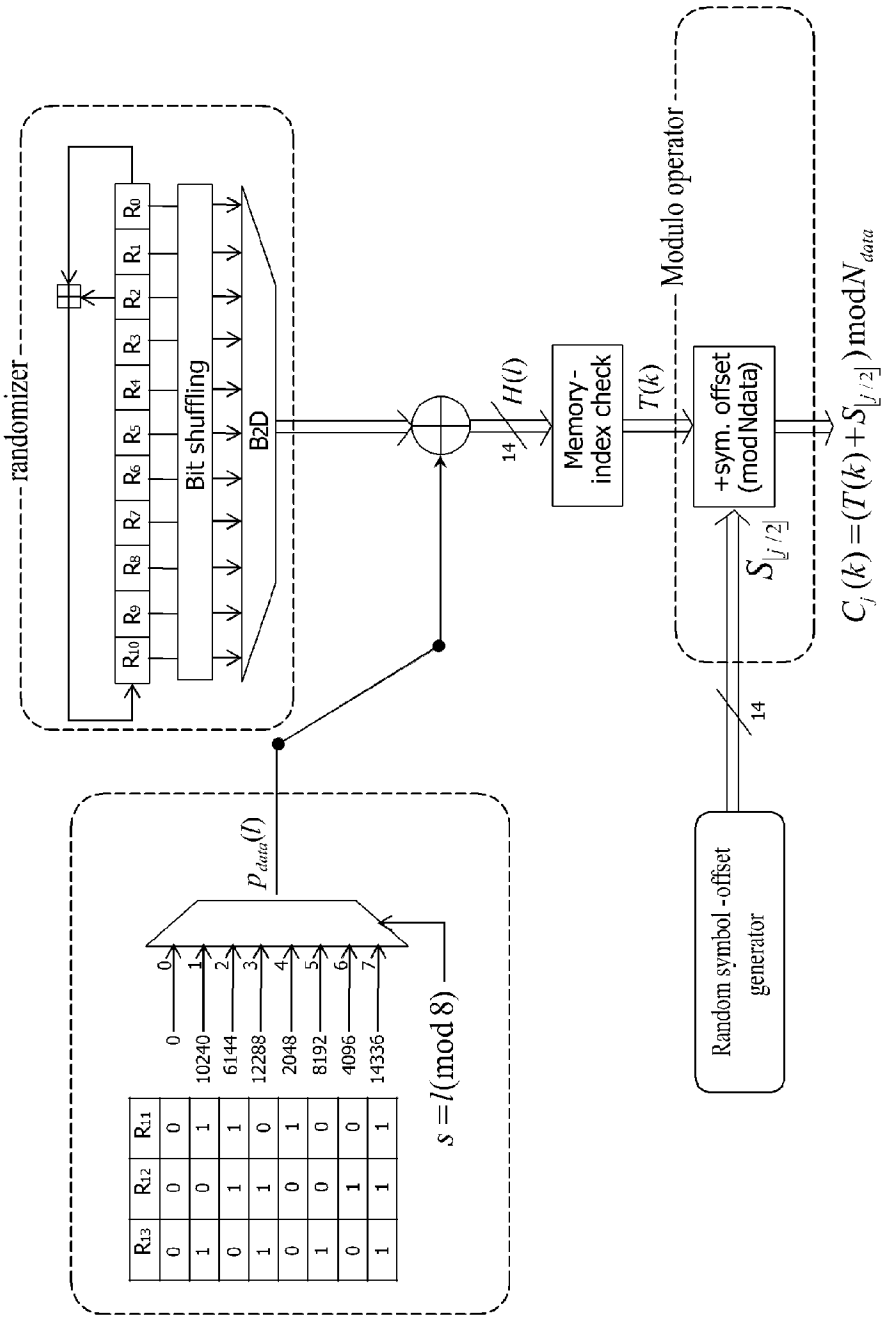
[Fig. 68]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
B2D
14
$S_{[j/2]}$
Spreader
$s = j \pmod 4$
R12
R13
0 0
1 0
0 1
1 1
MUX
$p_{sym}(n)$
0
8192
4096
12288
(a)
$0 \le n < 4$
$R^n_{[11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0,0$
$n = 4$
$R^4_{[11,10,9,8,7,6,5,4,3,2,1,0]} = 1,1,1,1,1,1,1,1,1,1,1,1$
$4 < n < N_{sym}/2$
$R^n_{[10,9,8,7,6,5,4,3,2,1,0]} = R^{n-1}_{[11,10,9,8,7,6,5,4,3,2,1]}$
$R^n_{11} = R^{n-1}_0 \oplus R^{n-1}_1 \oplus R^{n-1}_4 \oplus R^{n-1}_6$
(b)
for $(n=0; n < \left\lfloor \frac{N_{sym}}{2} \right\rfloor; n=n+1)$
{
$S_n = P_{sym}(n) + \sum_{m=0}^{11} (2^m \times R^n_m);$
}
The associated 12 th primitive polynomial, $f(x)=1+x^6+x^8+x^{11}+x^{12}$
Operation of the PN generator for $0 \le n < \lfloor Nsym/2 \rfloor$

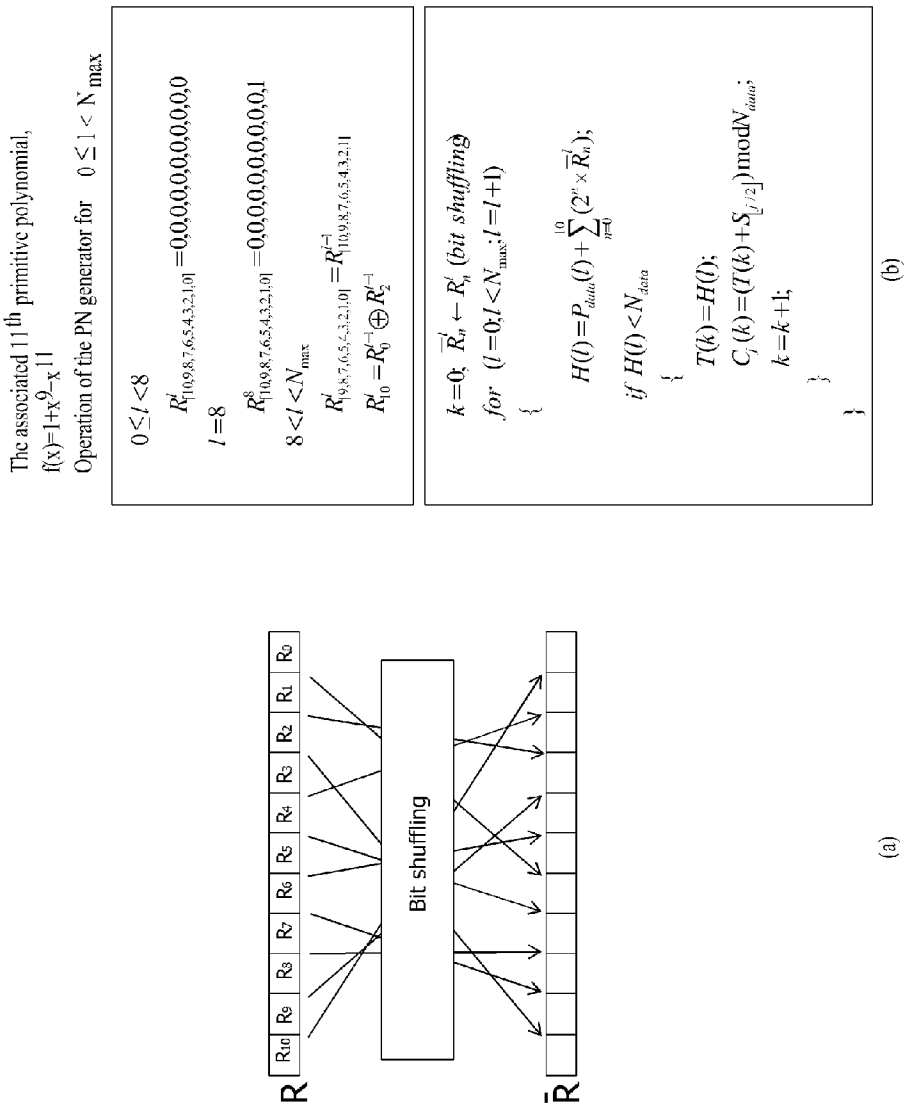
[Fig. 69]


Random main interleaving-seed generator
R12 R11 R10 R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
B2D
Spreader (1-bit)
14
H(l)
Memory-index check
T(k)
+sym. offset (modNdata)
14
14
Spreader (2-bit)
B2D
R0 R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11
Random symbol-offset generator
$C_j(k) = (T(k) + S_{\lfloor j/2 \rfloor}) \bmod N_{data}$

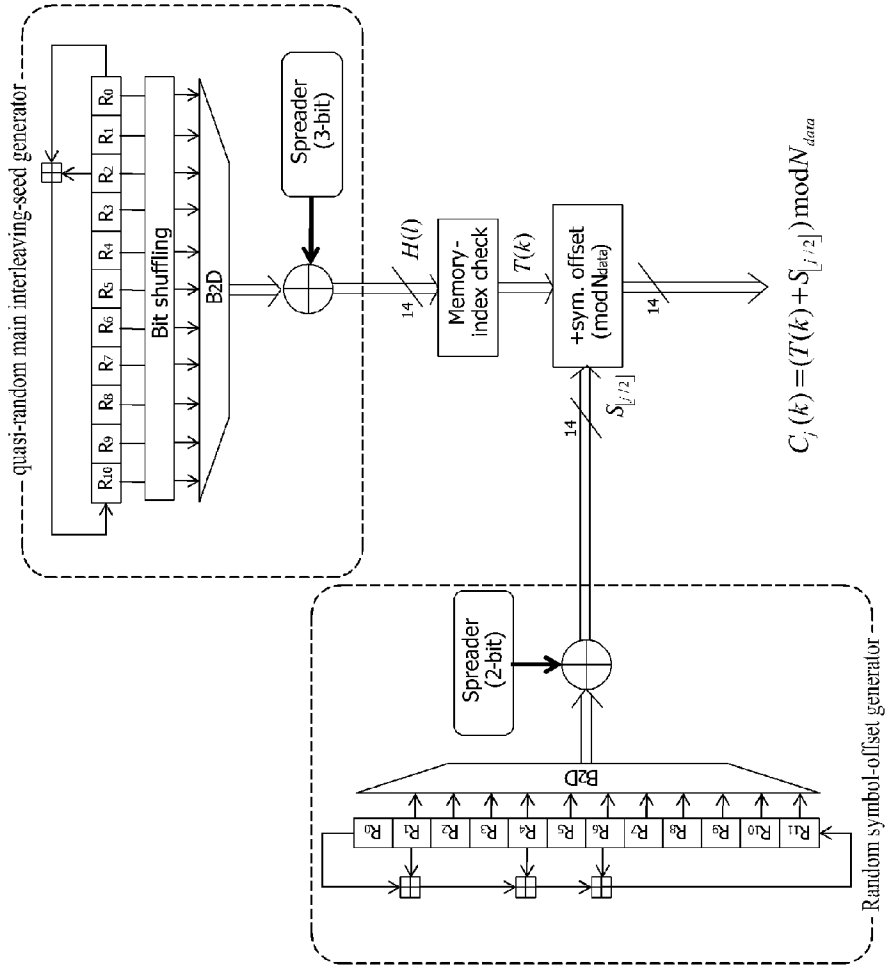
[Fig. 70]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
Bit shuffling
B2D
H(l)
14
Memory-index check
T(k)
Modulo operator
+sym. offset (modNdata)
$S_{\lfloor l/2 \rfloor}$
$C_j(k) = (T(k) + S_{\lfloor l/2 \rfloor}) \bmod N_{data}$
14
Random symbol -offset generator
$p_{data}(l)$
$s = l \pmod 8$
0
1
2
3
4
5
6
7
0
10240
6144
12288
2048
8192
4096
14336
R13 R12 R11
0 0 0
1 0 1
0 1 1
1 1 0
0 0 1
1 0 0
0 1 0
1 1 1

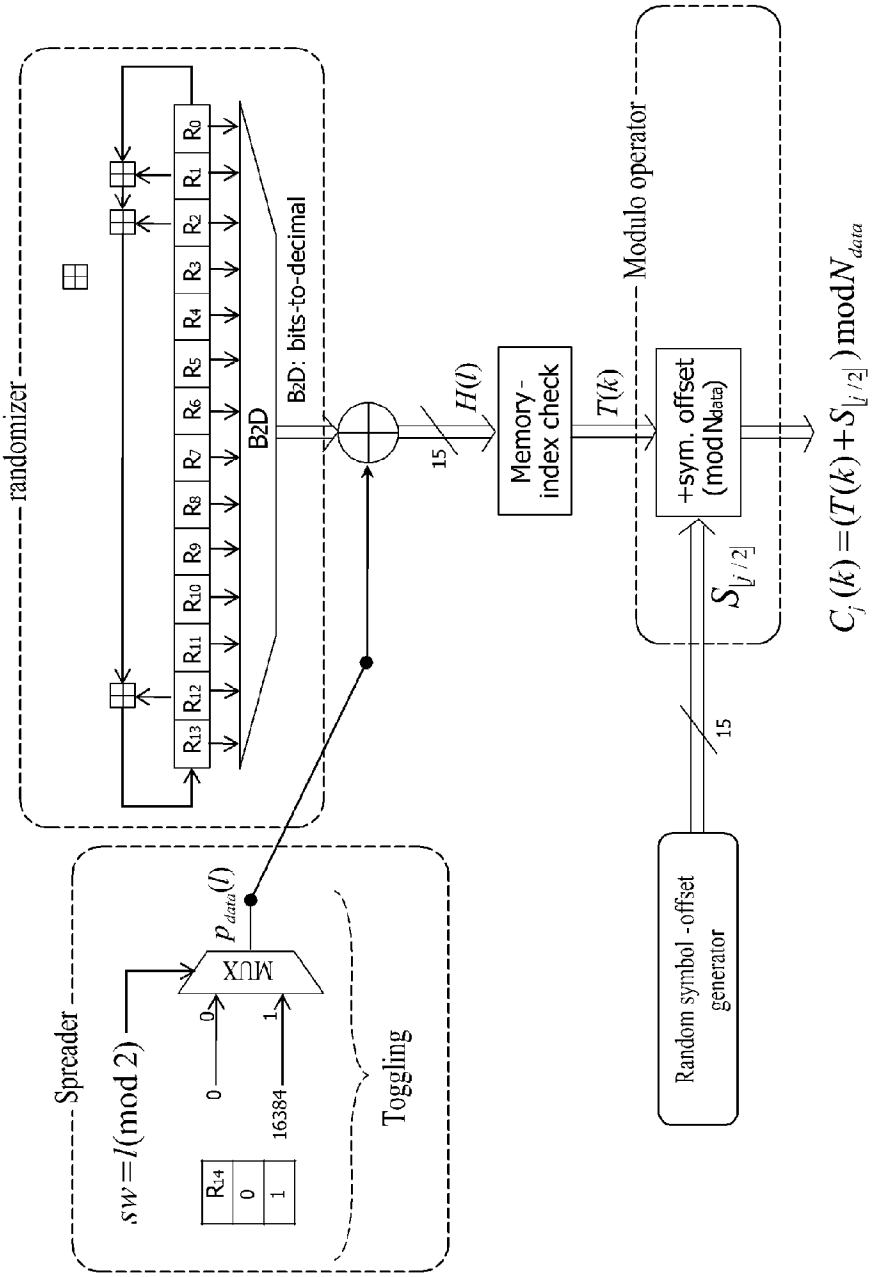
[Fig. 71]


The associated 11th primitive polynomial,
f(x)=1+x9+x11
Operation of the PN generator for 0 ≤ l < N max
0 ≤ l < 8
R'[10,9,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,0,0
l =8
R8[10,9,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,0,1
8 < l < Nmax
R'[9,8,7,6,5,4,3,2,1,0] =R'l-1[10,9,8,7,6,5,4,3,2,1]
R'10 =R'l-10 ⊕ R'l-12
k=0; R'n ← R'n (bit shuffling)
for (l=0;l<Nmax;l=l+1)
{
H(l)=Pdata(l)+∑n=010 (2n × R'n);
if H(l) <Ndata
{
T(k)=H(l);
Cj (k)=(T(k)+S⌊j/2⌋)modNdata;
k=k+1;
}
}
(b)
R
R10
R9
R8
R7
R6
R5
R4
R3
R2
R1
R0
Bit shuffling
R̄
(a)

[Fig. 72]


quasi-random main interleaving-seed generator
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
Bit shuffling
B2D
Spreader (3-bit)
H(l)
14
Memory-index check
T(k)
+sym. offset (modNdata)
14
$C_j(k)=(T(k)+S_{\lfloor j/2 \rfloor})\bmod N_{data}$
14
$S_{\lfloor j/2 \rfloor}$
Spreader (2-bit)
B2D
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
Random symbol-offset generator

[Fig. 73]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
B2D
B2D: bits-to-decimal
15
H(l)
Memory - index check
T(k)
Modulo operator
+sym. offset (mod N_data)
S_⌊j/2⌋
C_j(k) = (T(k) + S_⌊j/2⌋) mod N_data
15
Random symbol - offset generator
Spreader
sw = l (mod 2)
p_data(l)
MUX
0
1
0
16384
R14
0
1
Toggling

[Fig. 74]

The associated 14th polynomial, $f(x)=1+x^2+x^{12}+x^{13}+x^{14}$

Operation of the PN generator for $0 \le l < N_{max}(32768)$

$$0 \le l < 2$$

$$R^{l}_{[13,12,11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0,0,0,0$$

$$l = 2$$

$$R^{2}_{[13,12,11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,1,0,0,0,0,0,0,0,0,1,1,1$$

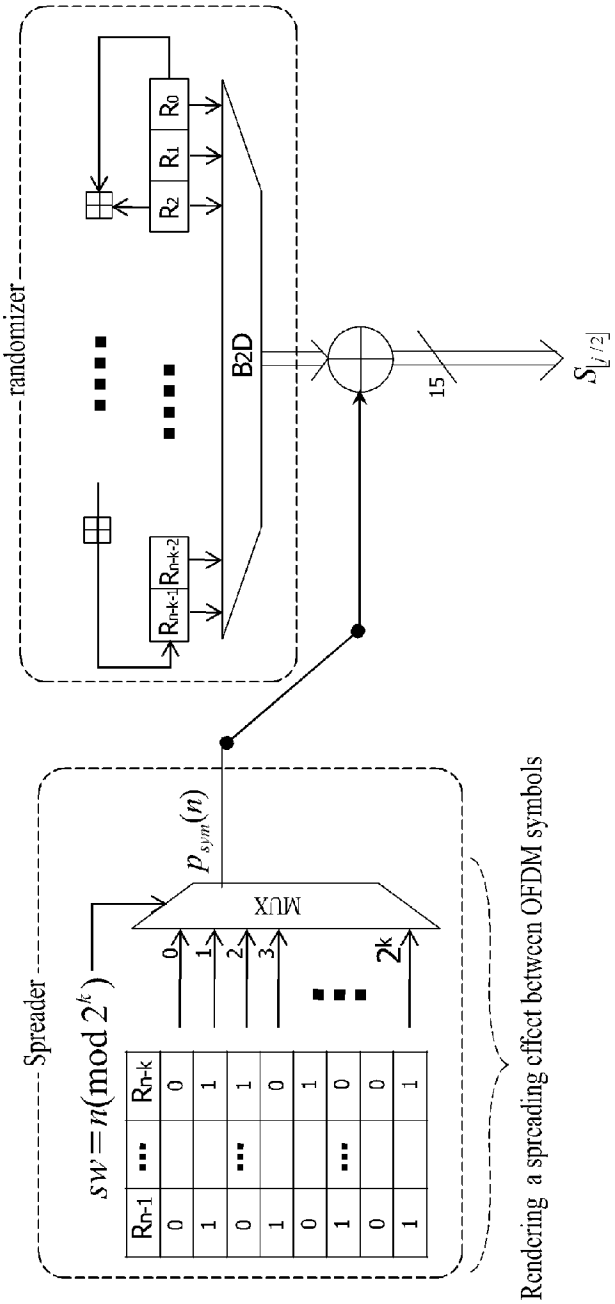
$$2 < l < N_{max}$$

$$R^{l}_{[11,10,9,8,7,6,5,4,3,2,1,0]} = R^{l-1}_{[13,12,11,10,9,8,7,6,5,4,3,2,1]}$$

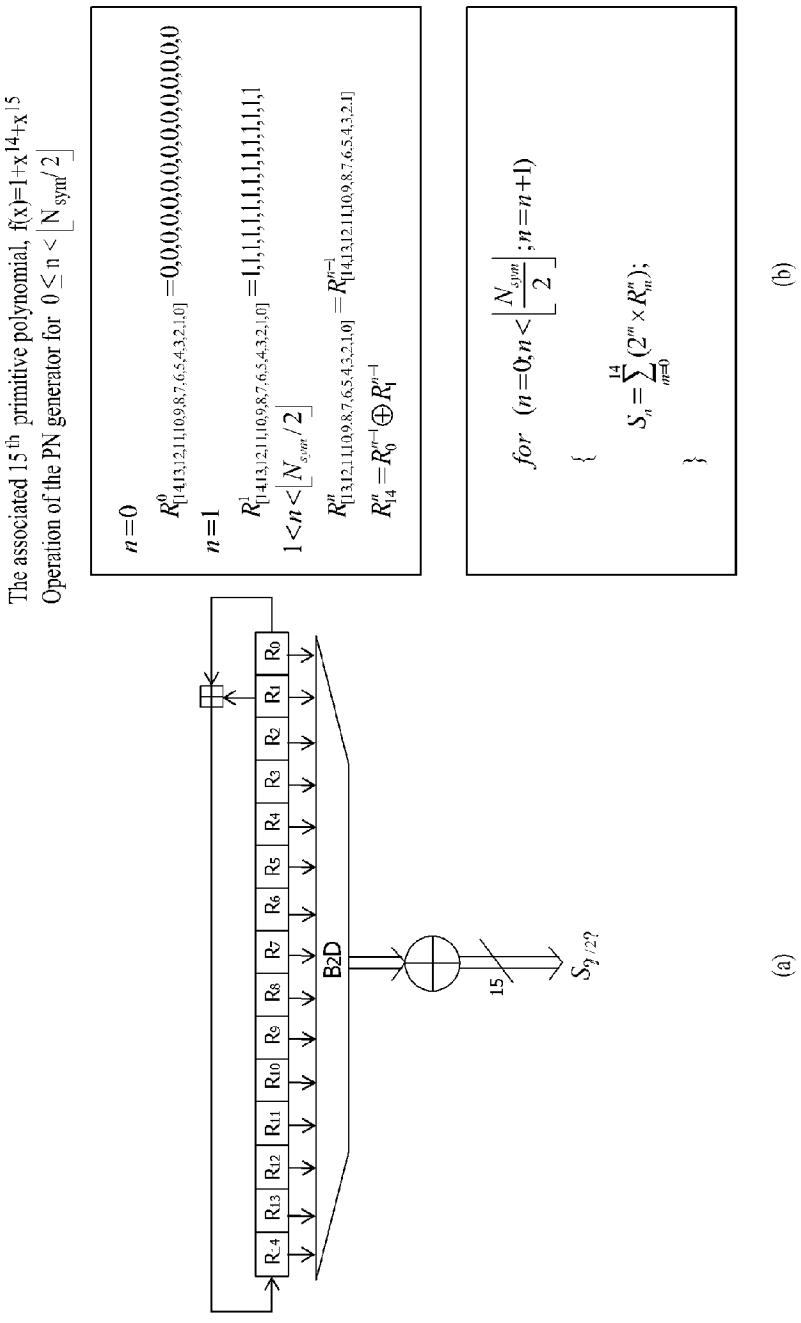
$$R^{l}_{13} = R^{l-1}_{0} \oplus R^{l-1}_{1} \oplus R^{l-1}_{2} \oplus R^{l-1}_{12}$$

```
k = 0;
for (l = 0; l < N_max; l = l + 1)
{
    H(l) = P_data(l) + Σ_{n=0}^{12} (2^n × R_n^l);
    if H(l) < N_data
    {
        T(k) = H(l);
        C_j(k) = (T(k) + S_⌊j/2⌋) mod N_data;
        k = k + 1;
    }
}
```

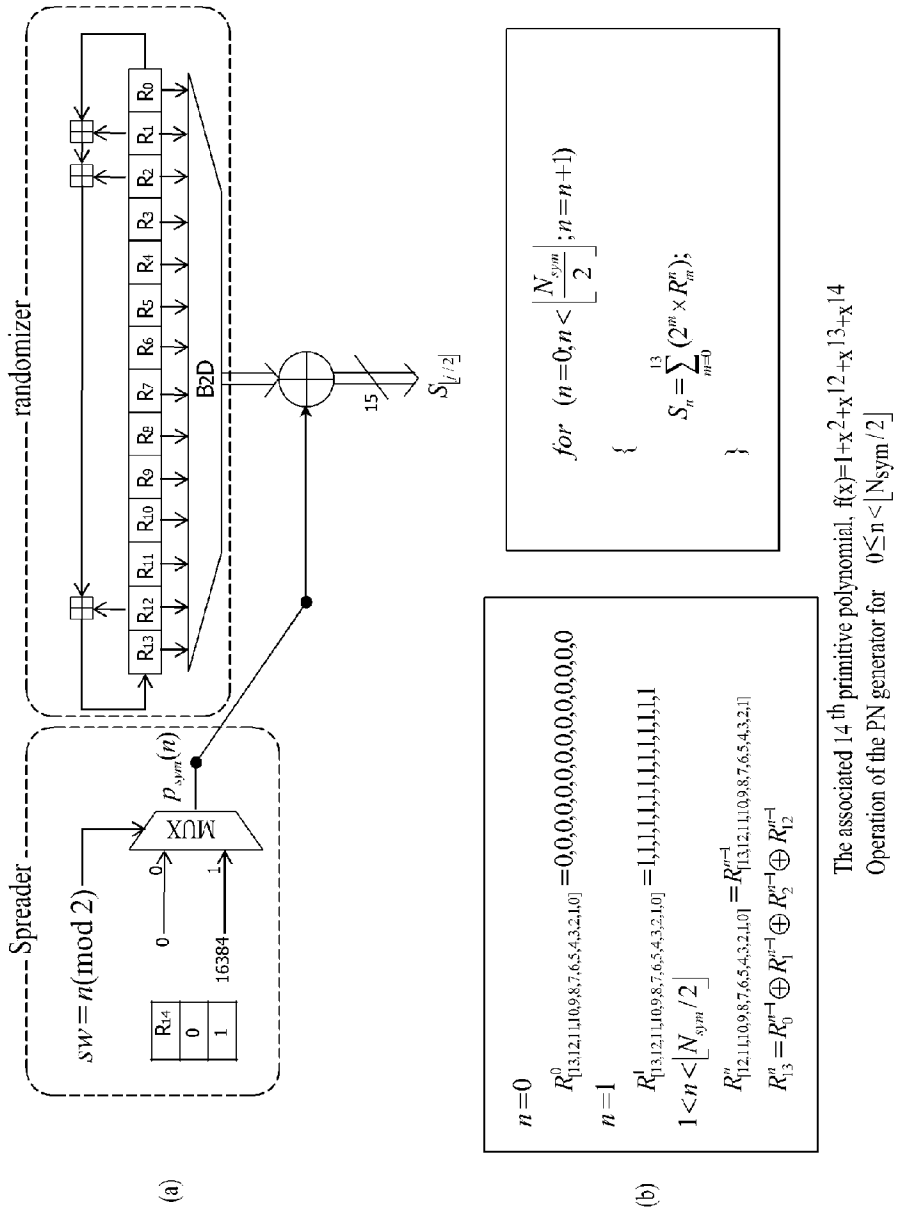
[Fig. 75]


randomizer
R0
R1
R2
Rn-k-1
Rn-k-2
B2D
15
$S_{[i/2]}$
Spreader
$sw = n(\mathrm{mod}\ 2^k)$
Rn-1
Rn-k
MUX
$p_{sym}(n)$
0
1
2
3
2k
Rendering a spreading effect between OFDM symbols

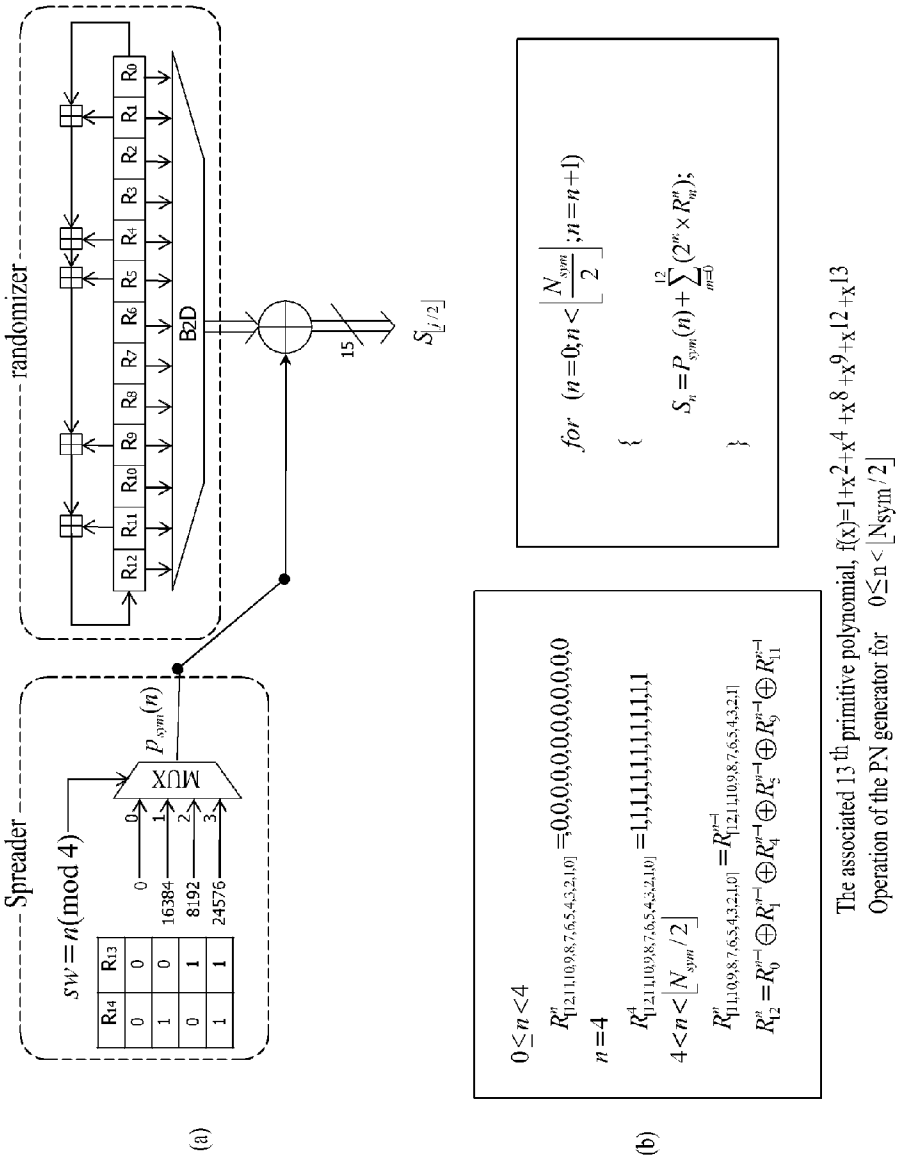
[Fig. 76]


The associated 15th primitive polynomial, f(x)=1+x14+x15
Operation of the PN generator for 0 ≤ n < ⌊Nsym/2⌋
n=0
R0[14,13,12,11,10,9,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
n=1
R1[14,13,12,11,10,9,8,7,6,5,4,3,2,1,0] =1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1<n<⌊Nsym/2⌋
Rn[13,12,11,10,9,8,7,6,5,4,3,2,1,0] =Rn-1[14,13,12,11,10,9,8,7,6,5,4,3,2,1]
R14n =R0n-1 ⊕ R1n-1
for (n=0;n<⌊Nsym/2⌋;n=n+1)
{
Sn=Σm=014(2m×Rmn);
}
(b)
R-4
R13
R12
R11
R10
R9
R8
R7
R6
R5
R4
R3
R2
R1
R0
B2D
15
S0/2?
(a)

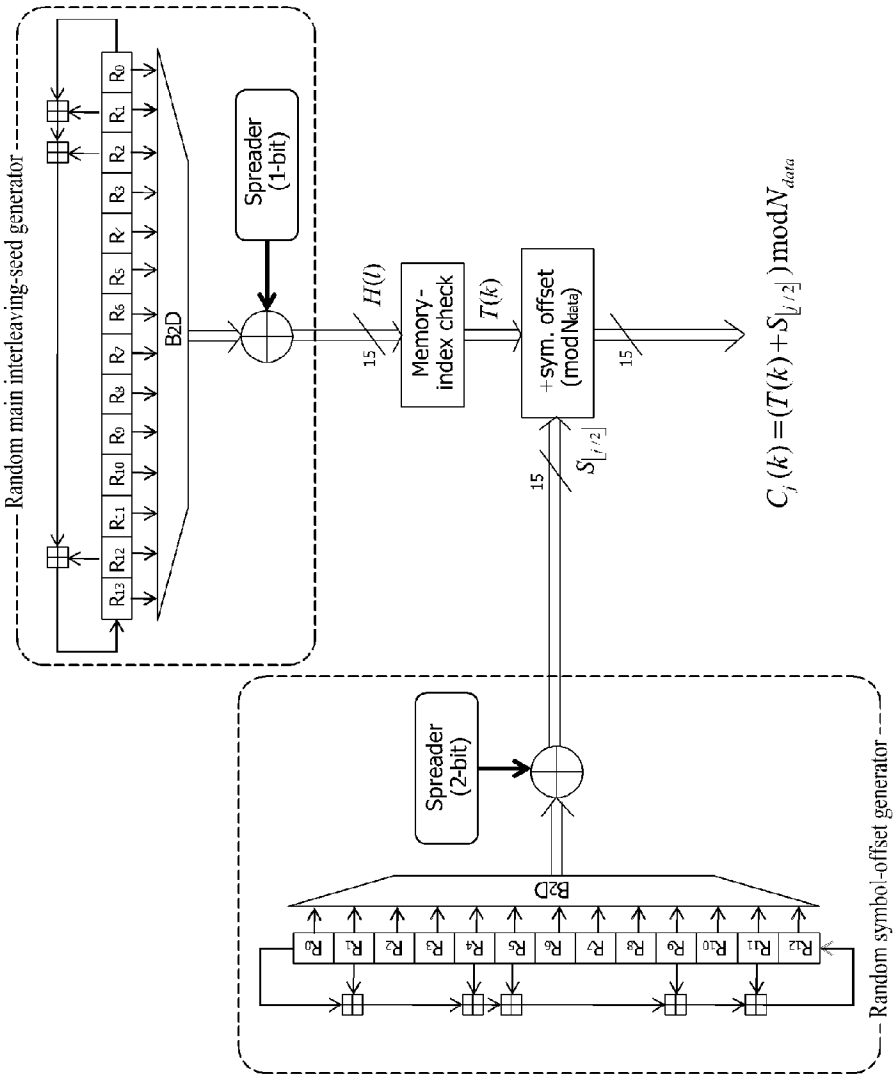
[Fig. 77]


randomizer
Spreader
$sw = n \pmod 2$
R14
0
1
0
16384
MUX
$p_{sym}(n)$
R13 R12 R11 R10 R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
B2D
15
$S_{\lfloor l/2 \rfloor}$
(a)
$n=0$
$R^0_{[13,12,11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0,0,0,0$
$n=1$
$R^1_{[13,12,11,10,9,8,7,6,5,4,3,2,1,0]} = 1,1,1,1,1,1,1,1,1,1,1,1,1,1$
$1 < n < \lfloor N_{sym}/2 \rfloor$
$R''_{[12,11,10,9,8,7,6,5,4,3,2,1,0]} = R^{n-1}_{[13,12,11,10,9,8,7,6,5,4,3,2,1]}$
$R^n_{13} = R^{n-1}_0 \oplus R^{n-1}_1 \oplus R^{n-1}_2 \oplus R^{n-1}_{12}$
The associated 14 th primitive polynomial, $f(x)=1+x^2+x^{12}+x^{13}+x^{14}$
Operation of the PN generator for $0 \le n < \lfloor N_{sym}/2 \rfloor$
for $(n=0; n < \lfloor \frac{N_{sym}}{2} \rfloor ; n=n+1)$
{
$S_n = \sum_{m=0}^{13} (2^m \times R^n_m);$
}
(b)

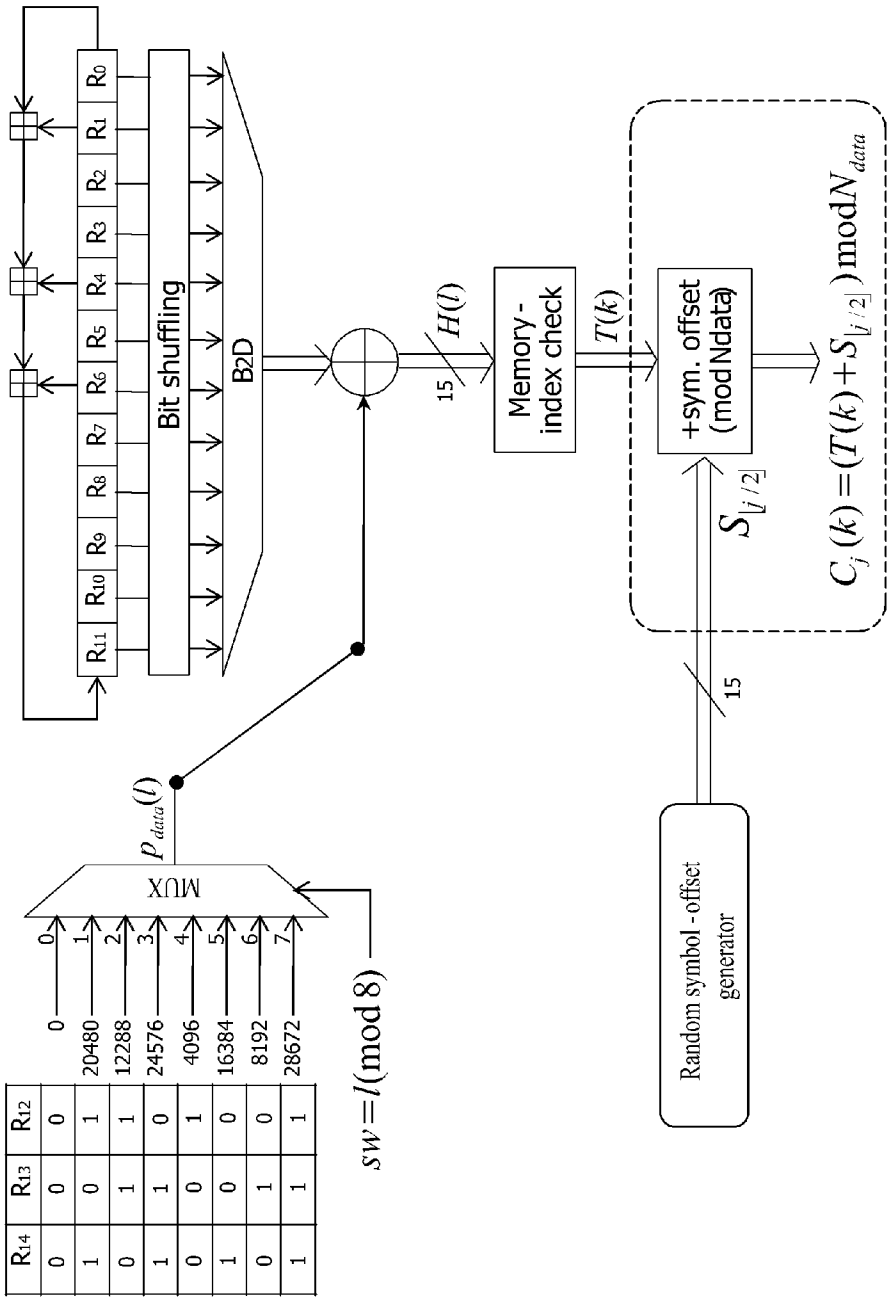
[Fig. 78]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
B2D
15
$S_{\lfloor i/2 \rfloor}$
Spreader
$sw = n \pmod 4$
R14
R13
0 0
1 0
0 1
1 1
MUX
$P_{sym}(n)$
0
16384
8192
24576
0
1
2
3
(a)
$0 \le n < 4$
$R^n_{[12,11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0,0,0$
$n = 4$
$R^4_{[12,11,10,9,8,7,6,5,4,3,2,1,0]} = 1,1,1,1,1,1,1,1,1,1,1,1,1$
$4 < n < \lfloor N_{sym} / 2 \rfloor$
$R^n_{[11,10,9,8,7,6,5,4,3,2,1,0]} = R^{n-1}_{[12,11,10,9,8,7,6,5,4,3,2,1]}$
$R^n_{12} = R^{n-1}_0 \oplus R^{n-1}_1 \oplus R^{n-1}_4 \oplus R^{n-1}_7 \oplus R^{n-1}_9 \oplus R^{n-1}_{11}$
The associated 13 th primitive polynomial, $f(x)=1+x^2+x^4+x^8+x^9+x^{12}+x^{13}$
Operation of the PN generator for $0 \le n < \lfloor Nsym/2 \rfloor$
$for \ (n=0; n < \left\lfloor \frac{N_{sym}}{2} \right\rfloor ; n=n+1)$
{
$S_n = P_{sym}(n) + \sum_{m=0}^{12} (2^m \times R^n_m);$
}
(b)

[Fig. 79]

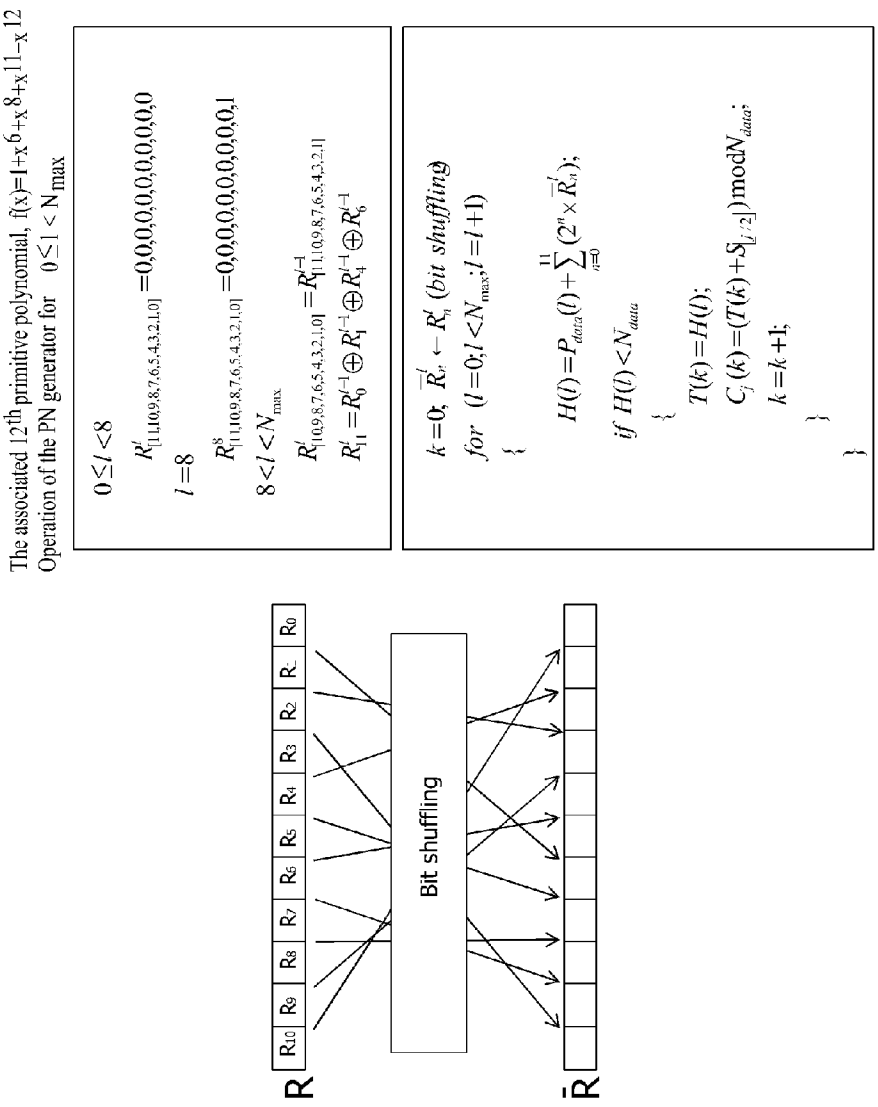

Random main interleaving-seed generator
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
B2D
Spreader (1-bit)
H(l)
15
Memory-index check
T(k)
+sym. offset (mod Ndata)
15
$C_j(k) = (T(k) + S_{\lfloor l/2 \rfloor}) \bmod N_{data}$
15
$S_{\lfloor l/2 \rfloor}$
Spreader (2-bit)
B2D
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
Random symbol-offset generator

[Fig. 80]

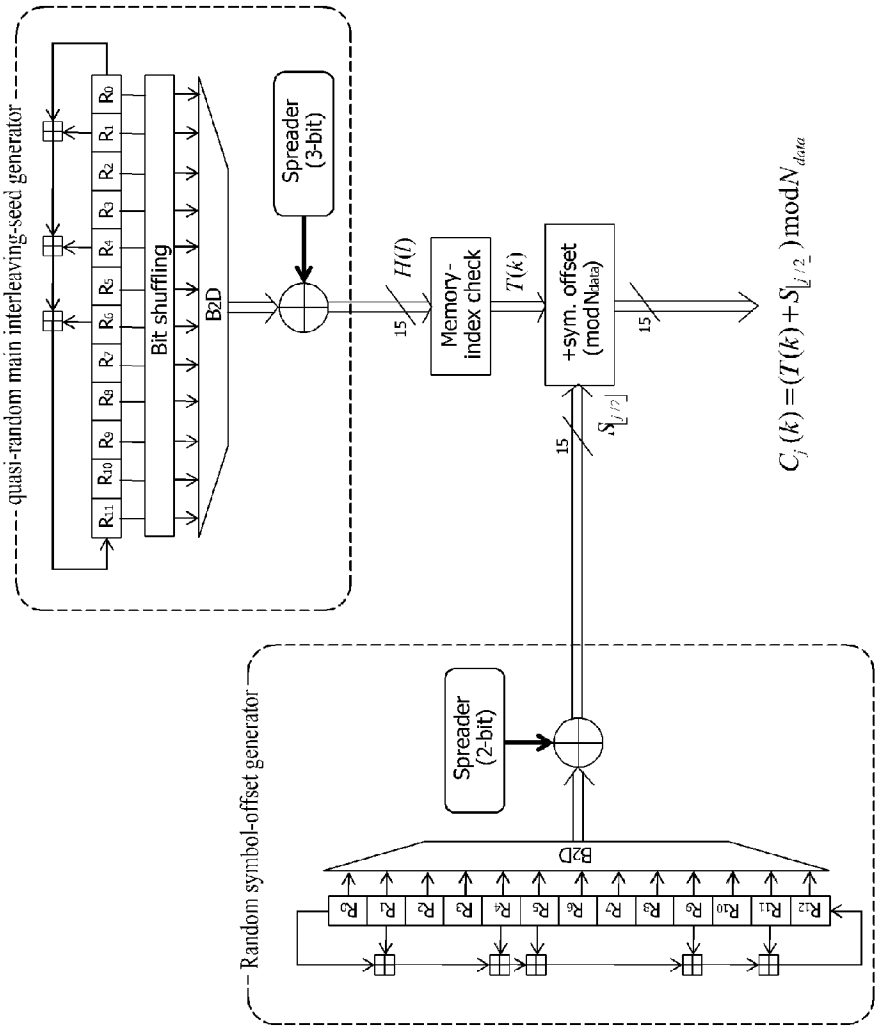


R14	R13	R12
0	0	0
1	0	1
0	1	1
1	1	0
0	0	1
1	0	0
0	1	0
1	1	1

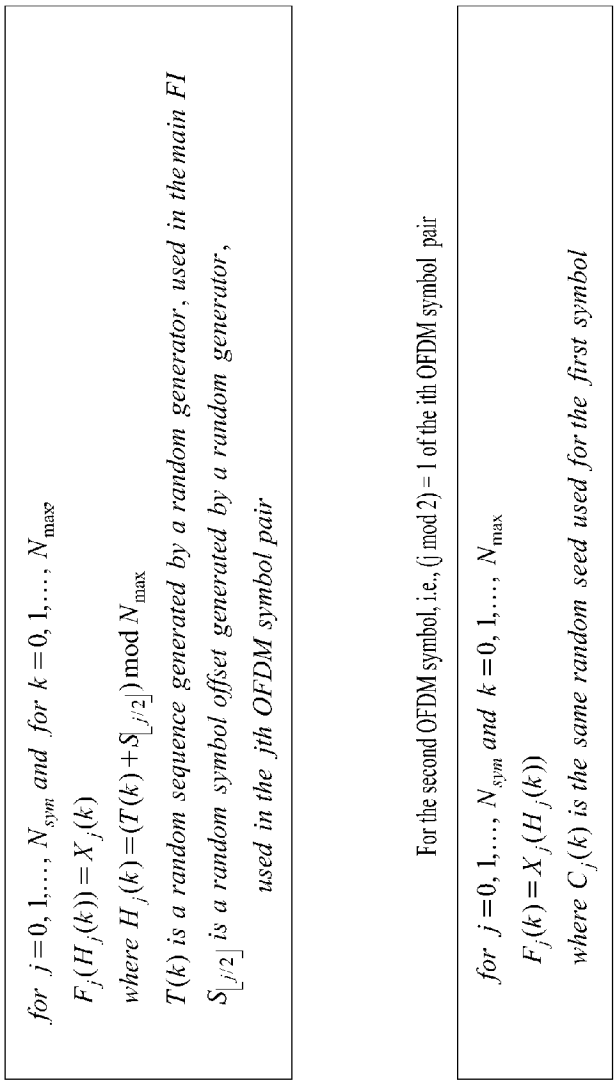
[Fig. 81]


The associated 12th primitive polynomial, f(x)=1+x6+x8+x11+x12
Operation of the PN generator for 0≤l < Nmax
0≤l <8
$R^l_{[11,10,9,8,7,6,5,4,3,2,1,0]}$ =0,0,0,0,0,0,0,0,0,0,0,0
l =8
$R^8_{[11,10,9,8,7,6,5,4,3,2,1,0]}$ =0,0,0,0,0,0,0,0,0,0,0,1
8<l <Nmax
$R^l_{[10,9,8,7,6,5,4,3,2,1,0]}$ =$R^{l-1}_{[11,10,9,8,7,6,5,4,3,2,1]}$
R^l_{11} =R^{l-1}_0 ⊕R^{l-1}_1 ⊕R^{l-1}_4 ⊕R^{l-1}_6
k =0; $\bar{R}_r^l \leftarrow R_i^l$ (bit shuffling)
for (l =0;l <Nmax;l =l +1)
{
$H(l)=P_{data}(l)+\sum_{n=0}^{11}(2^n \times \bar{R}_n^l)$;
if H(l) <Ndata
{
T(k)=H(l);
$C_j(k)=(T(k)+S_{\lfloor j/2 \rfloor})\bmod N_{data}$;
k =k +1;
}
}
(b)
R
R10
R9
R8
R7
R6
R5
R4
R3
R2
R1
R0
Bit shuffling
$\bar{R}$
(a)

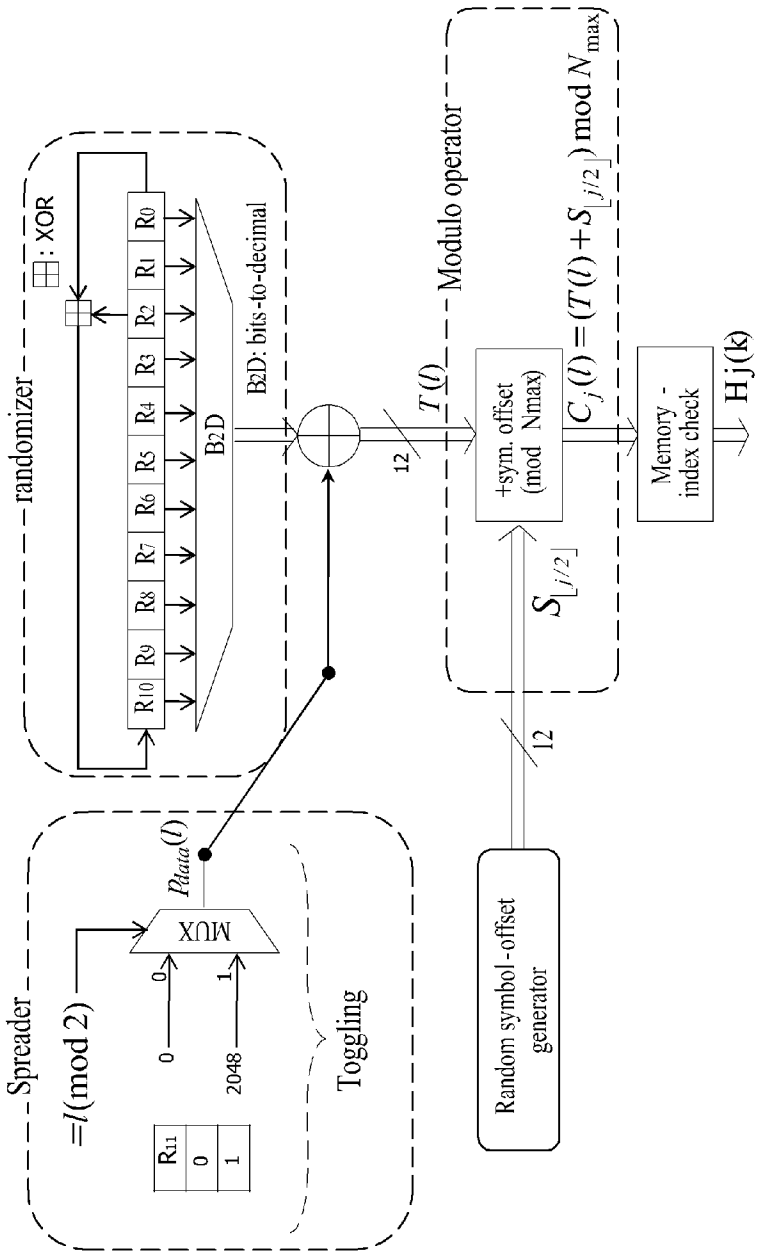
[Fig. 82]


quasi-random main interleaving-seed generator
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
Bit shuffling
B2D
Spreader (3-bit)
15
H(l)
Memory-index check
T(k)
+sym. offset (mod Ndata)
15
15
S⌊l/2⌋
Cj(k)=(T(k)+S⌊l/2⌋) mod Ndata
Random symbol-offset generator
Spreader (2-bit)
B2D
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11

[Fig. 83]


For the first OFDM symbol, i.e., (j mod 2) = 0 of the ith OFDM symbol pair
for $j = 0, 1, \ldots, N_{sym}$ and for $k = 0, 1, \ldots, N_{max}$
$F_j(H_j(k)) = X_j(k)$
where $H_j(k) = (T(k) + S_{\lfloor j/2 \rfloor}) \bmod N_{max}$
$T(k)$ is a random sequence generated by a random generator, used in the main FI
$S_{\lfloor j/2 \rfloor}$ is a random symbol offset generated by a random generator,
used in the jth OFDM symbol pair
For the second OFDM symbol, i.e., (j mod 2) = 1 of the ith OFDM symbol pair
for $j = 0, 1, \ldots, N_{sym}$ and $k = 0, 1, \ldots, N_{max}$
$F_j(k) = X_j(H_j(k))$
where $C_j(k)$ is the same random seed used for the first symbol

[Fig. 84]


randomizer
⊞: XOR
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
B2D
B2D: bits-to-decimal
12
T(l)
Modulo operator
+sym. offset (mod Nmax)
$C_j(l) = (T(l) + S_{\lfloor j/2 \rfloor}) \bmod N_{max}$
Memory - index check
Hj(k)
12
$S_{\lfloor j/2 \rfloor}$
Random symbol-offset generator
Spreader
$=l \pmod 2$
MUX
$P_{data}(l)$
0
2048
0
1
R11
0
1
Toggling

[Fig. 85]

· The associated 11th primitive polynomial, $f(x)=1+x^{9}+x^{11}$

· Operation of the PN generator for $0 \le l < N_{max}(4096)$

$0 \le l < 2$

$R^{l}_{[10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0$

$l = 2$

$R^{2}_{[10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,1,0,1$

$2 < l < N_{max}$

$R^{l}_{[9,8,7,6,5,4,3,2,1,0]} = R^{l-1}_{[10,9,8,7,6,5,4,3,2,1]}$

$R^{l}_{10} = R^{l-1}_{0} \oplus R^{l-1}_{2}$

$k = 0;$

$for \ (l = 0; l < N_{max}; l = l + 1)$

{

$T(l) = P_{data}(l) + \sum_{n=0}^{10} (2^{n} \times R^{l}_{n});$

$C_{j}(l) = (T(l) + S_{\lfloor j/2 \rfloor}) \bmod N_{max};$

$if \ C_{j}(l) < N_{data}$

{

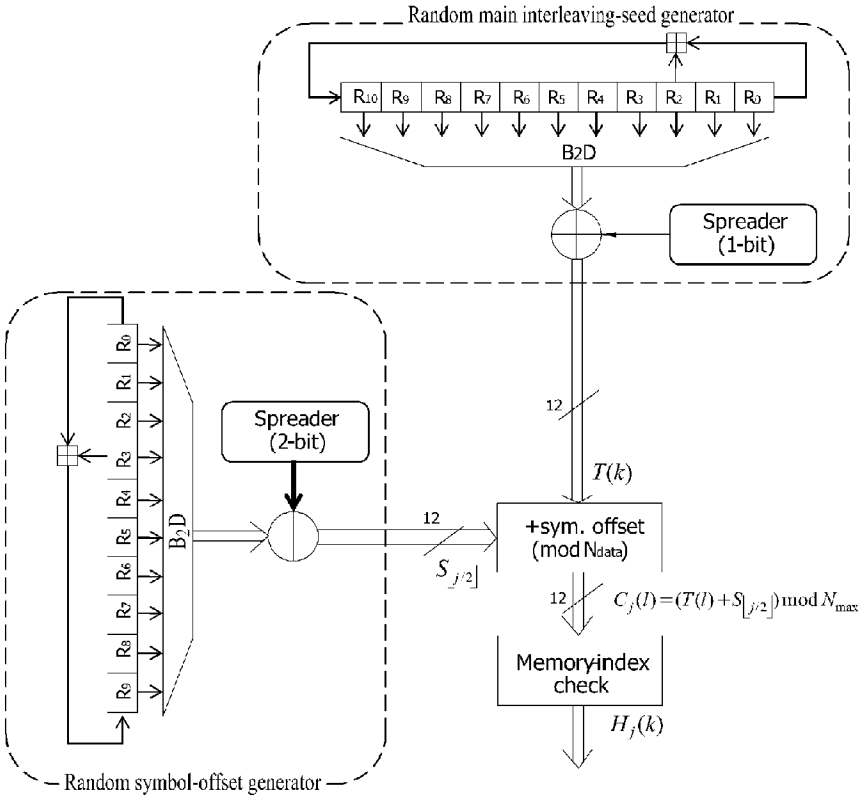
$H_{j}(k) = C_{j}(l);$

$k = k + 1;$

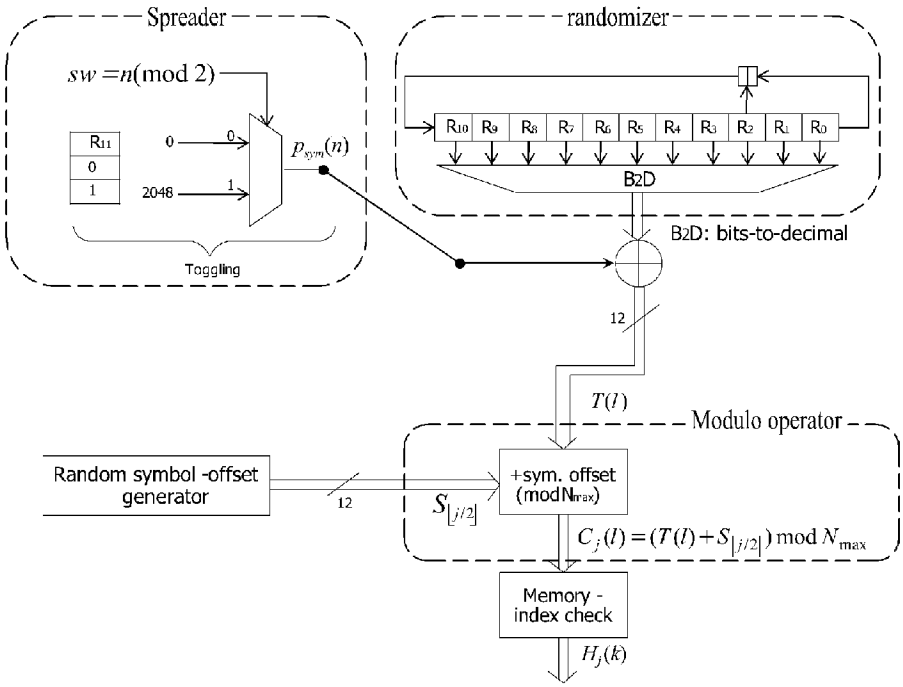
}

}

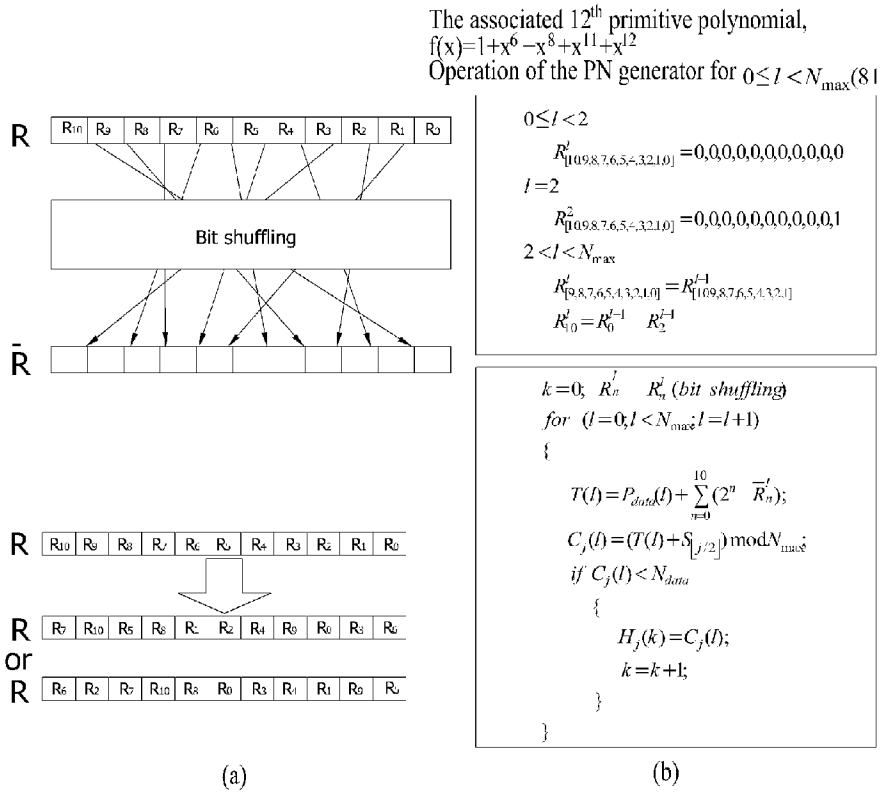
[Fig. 86]


Random main interleaving-seed generator
R10 R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
B2D
Spreader (1-bit)
12
T(k)
Spreader (2-bit)
12
+sym. offset (mod Ndata)
S⌊j/2⌋
12
Cj(l)=(T(l)+S⌊j/2⌋) mod Nmax
Memory-index check
Hj(k)
Random symbol-offset generator

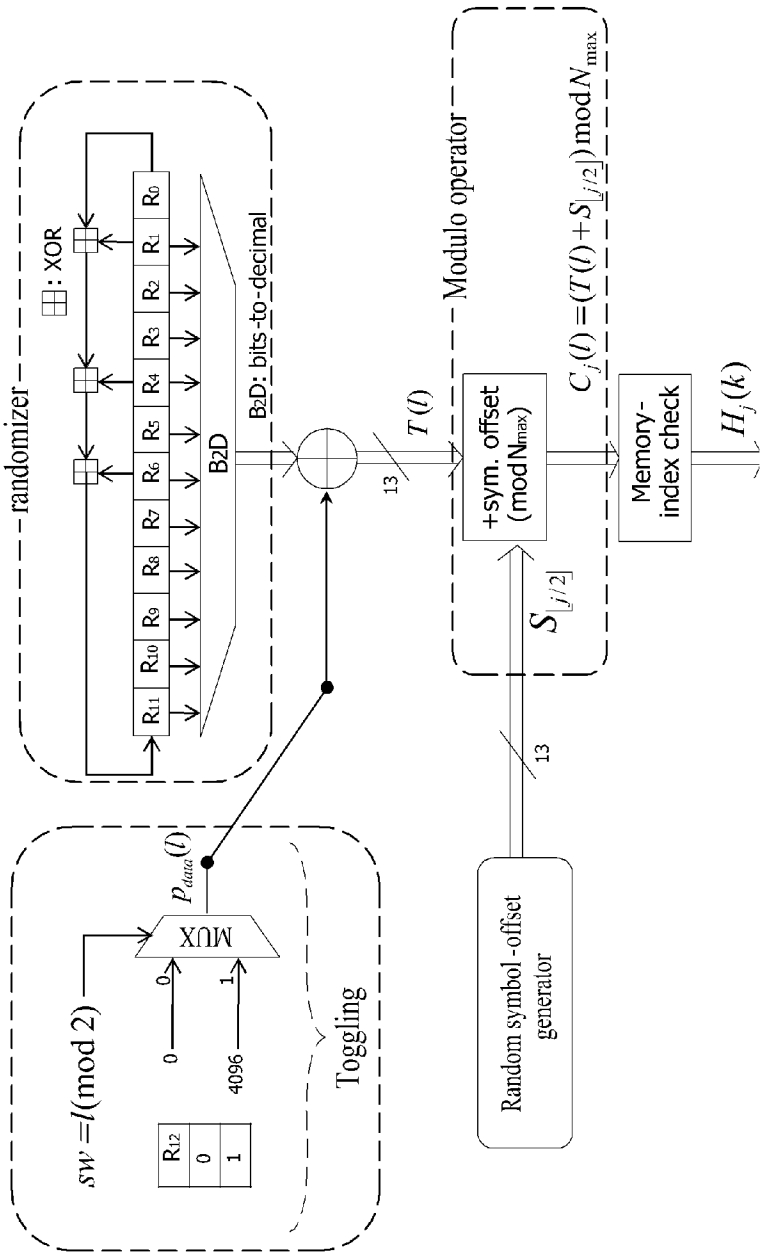
[Fig. 87]


Spreader
sw=n(mod 2)
R11
0
1
0
2048
0
1
psym(n)
Toggling
randomizer
R10 R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
B2D
B2D: bits-to-decimal
12
T(l)
Modulo operator
Random symbol -offset generator
12
S⌊j/2⌋
+sym. offset (mod Nmax)
Cj(l)=(T(l)+S⌊j/2⌋) mod Nmax
Memory - index check
Hj(k)

[Fig. 88]


The associated 12th primitive polynomial,
f(x)=1+x^6 −x^8+x^11+x^12
Operation of the PN generator for 0≤l<N_max(8192)
R
R10 R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
Bit shuffling
R̄
0≤l<2
R^l_[10,9,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,0,0
l=2
R^2_[10,9,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,0,1
2<l<N_max
R^l_[9,8,7,6,5,4,3,2,1,0] = R^{l-1}_[10,9,8,7,6,5,4,3,2,1]
R^l_10 = R^{l-1}_0 R^{l-1}_2
k=0; R^l_n R^l_n (bit shuffling)
for (l=0;l<N_max;l=l+1)
{
T(l)=P_data(l)+Σ_{n=0}^{10}(2^n R̄^l_n);
C_j(l)=(T(l)+S_⌊j/2⌋) mod N_max;
if C_j(l)<N_data
{
H_j(k)=C_j(l);
k=k+1;
}
}
R
R10 R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
R
R7 R10 R5 R8 R4 R2 R1 R0 R3 R6
or
R
R6 R2 R7 R10 R8 R0 R3 R9 R1 R5 R4
(a)
(b)

[Fig. 89]


randomizer
⊞: XOR
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
B2D
B2D: bits-to-decimal
13
T(l)
Modulo operator
+sym. offset (mod Nmax)
$C_j(l) = (T(l) + S_{\lfloor j/2 \rfloor}) \bmod N_{max}$
Memory-index check
$H_j(k)$
$S_{\lfloor j/2 \rfloor}$
13
Random symbol-offset generator
$p_{data}(l)$
MUX
0
1
0
4096
$sw = l \pmod 2$
R12
0
1
Toggling

[Fig. 90]

The associated 12th primitive polynomial, $f(x)=1+x^6+x^8+x^{11}+x^{12}$
Operation of the PN generator for $0 \le l < N_{max}$ (8192)

$0 \le l < 2$

$R^l_{[11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0,0$

$l = 2$

$R^2_{[11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,1,0,1,0,0,1,1$

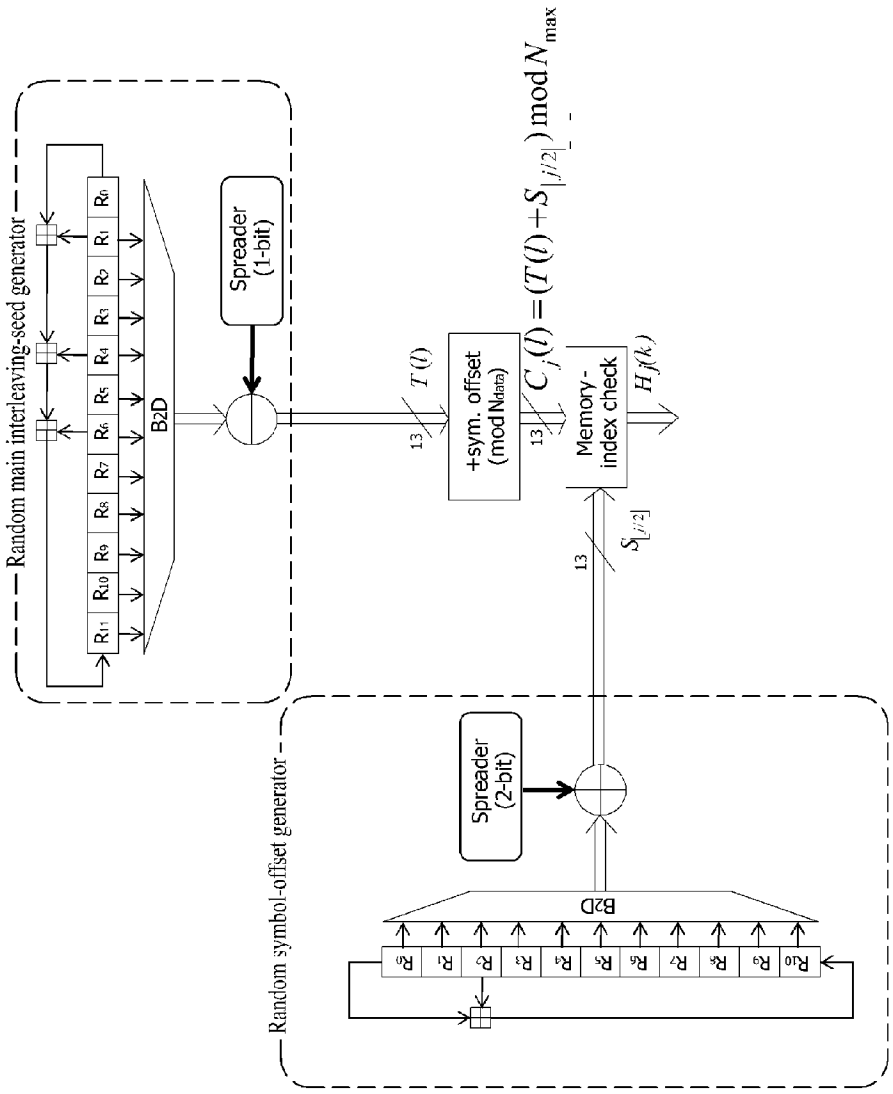
$2 < l < N_{max}$

$R^l_{[10,9,8,7,6,5,4,3,2,1,0]} = R^{l-1}_{[11,10,9,8,7,6,5,4,3,2,1]}$

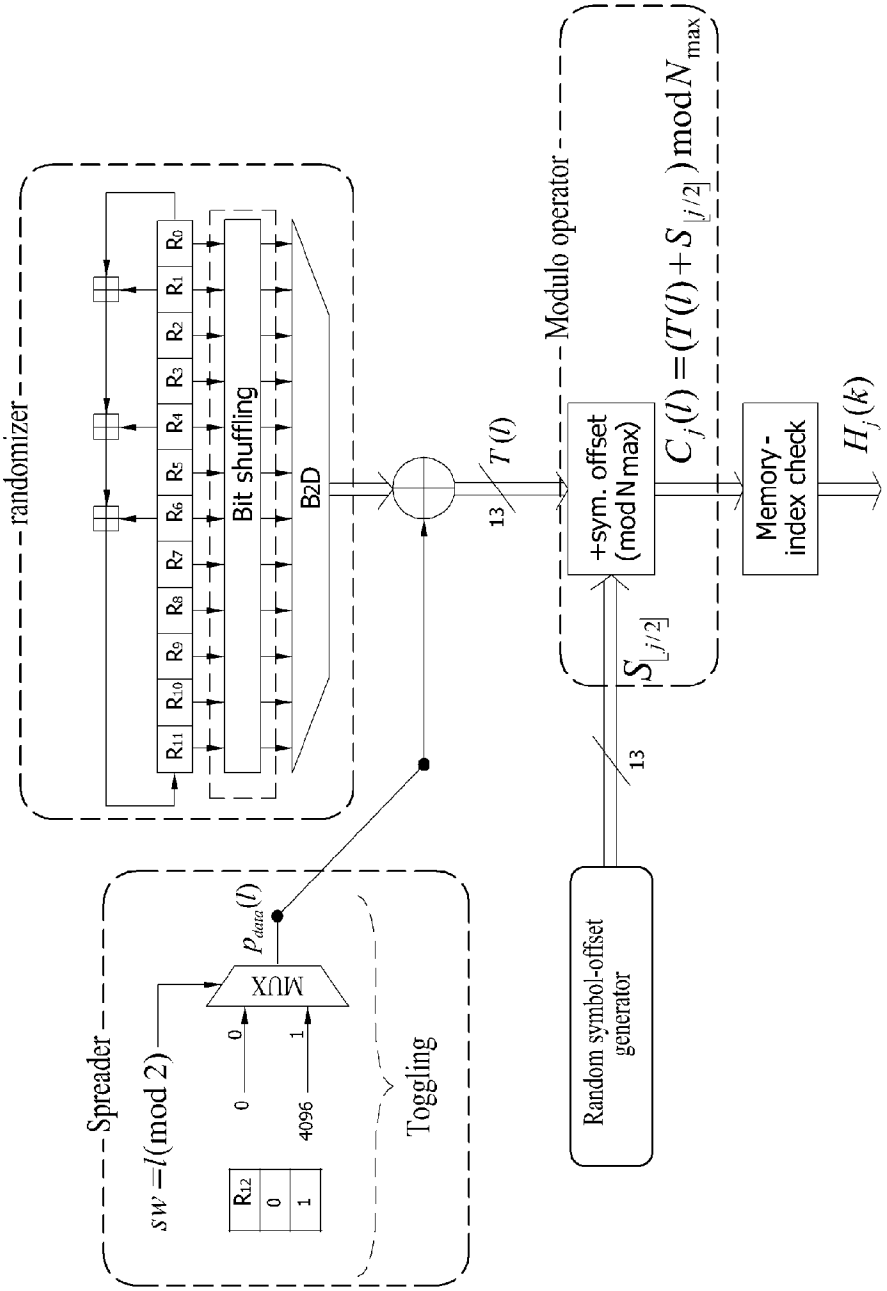
$R^l_{11} = R^{l-1}_0 \oplus R^{l-1}_1 \oplus R^{l-1}_4 \oplus R^{l-1}_6$

```
k = 0;
for (l = 0; l < Nmax; l = l + 1)
{
    T(l) = Pdata(l) + Σ_{n=0}^{11} (2^n × R_n^l);
    Cj(l) = (T(l) + S_{⌊j/2⌋}) mod Nmax;
    if Cj(l) < Ndata
    {
        Hj(k) = Cj(l);
        k = k + 1;
    }
}
```

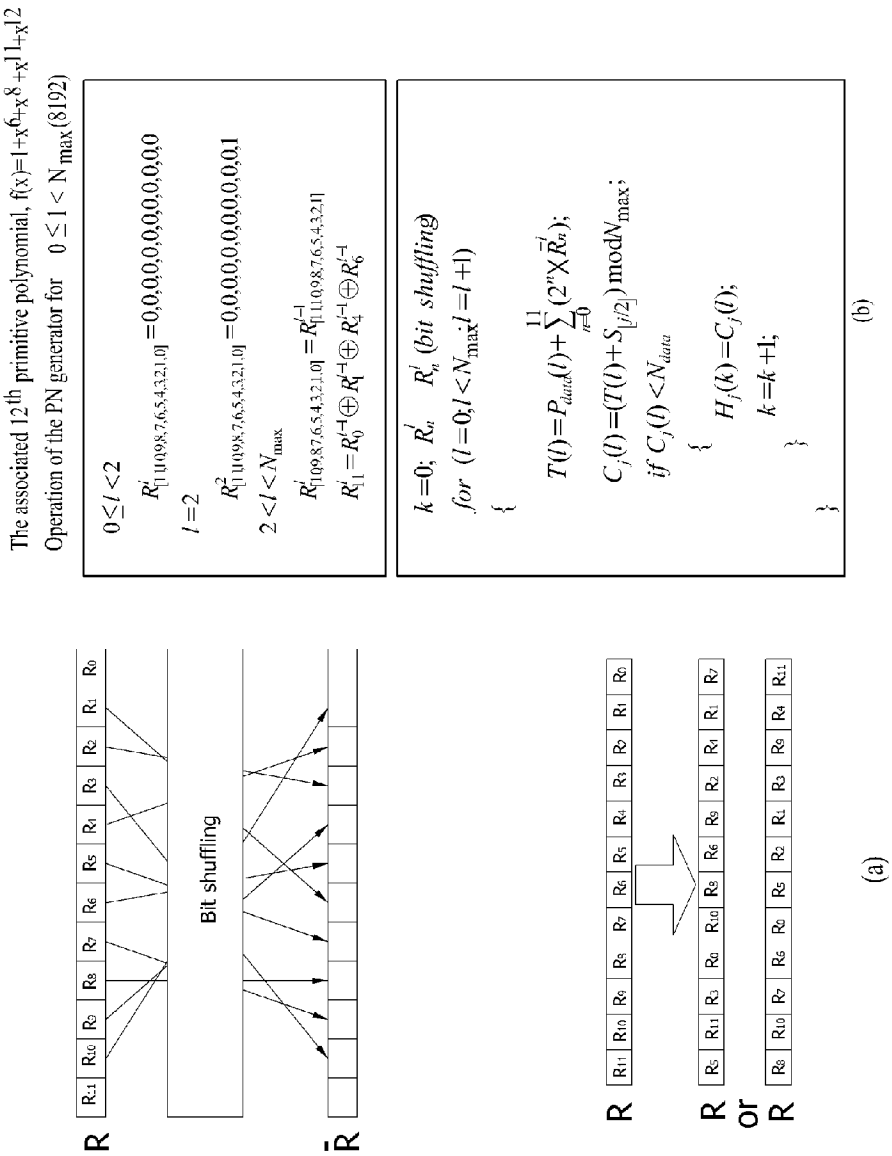
[Fig. 91]


Random main interleaving-seed generator
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
B2D
Spreader (1-bit)
T(l)
13
+sym. offset (mod Ndata)
13
Memory-index check
Hf(k)
13
Random symbol-offset generator
Spreader (2-bit)
B2D
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11

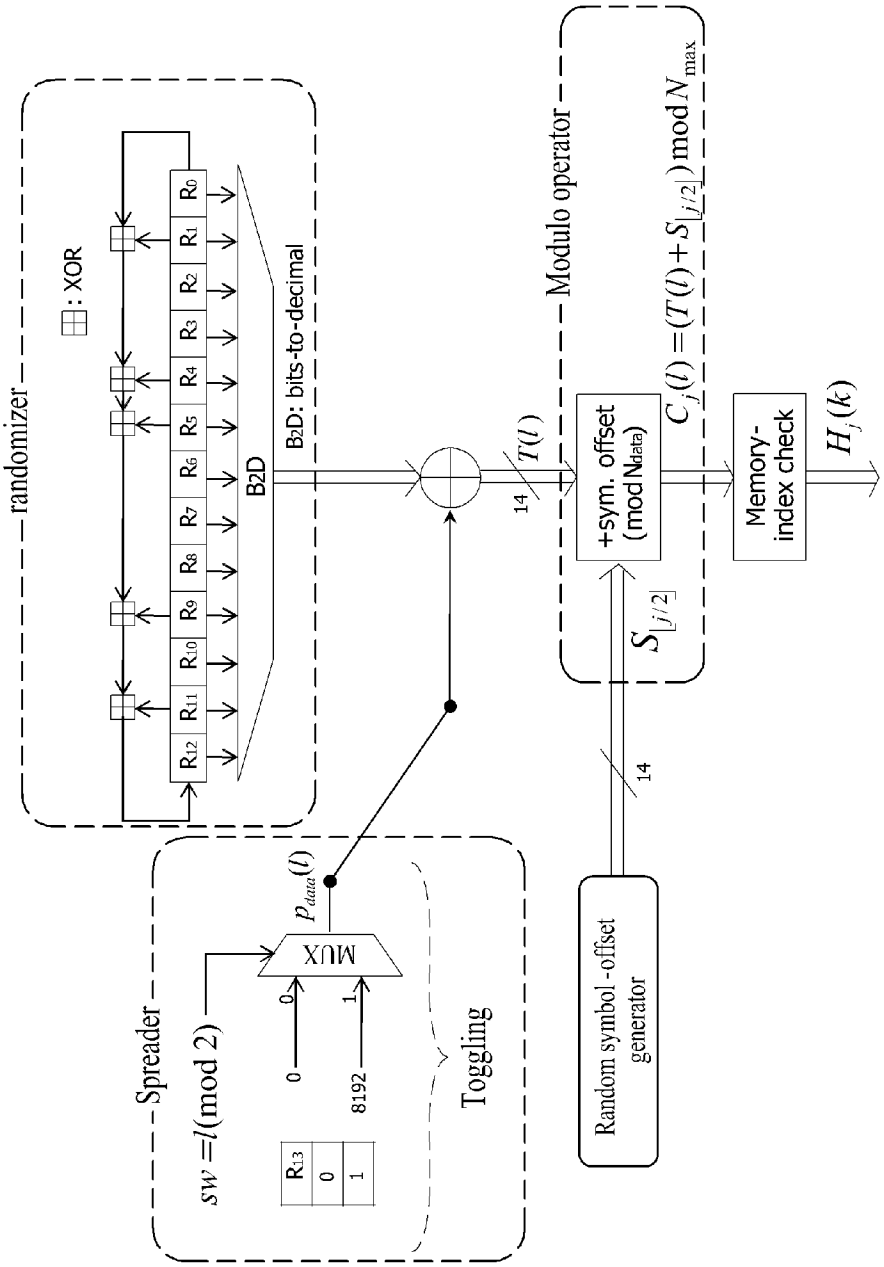
[Fig. 92]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
Bit shuffling
B2D
T(l)
13
Modulo operator
+sym. offset (mod Nmax)
$C_j(l) = (T(l) + S_{\lfloor j/2 \rfloor}) \bmod N_{max}$
Memory-index check
$H_j(k)$
$S_{\lfloor j/2 \rfloor}$
13
Random symbol-offset generator
Spreader
$sw = l \pmod 2$
$p_{data}(l)$
MUX
0
1
0
4096
R12
0
1
Toggling

[Fig. 93]


The associated 12th primitive polynomial, f(x)=1+x6+x8 +x11+x12
Operation of the PN generator for 0 ≤ l < Nmax (8192)
0≤l <2
R'[11109,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,0,0,0
l =2
R2[11109,8,7,6,5,4,3,2,1,0] =0,0,0,0,0,0,0,0,0,0,0,1
2 <l <Nmax
R'[109,87,6,5,4,3,2,1,0] = R'-1[11109,8,7,6,5,4,3,2,1]
R'11 = R'-10⊕ R'-14⊕ R'-12⊕ R'-16⊕
k =0; R'n R'n (bit shuffling)
for (l =0;l <Nmax;l =l +1)
{
T(l)=Pdata(l)+ ∑n=0 11 (2n×R'n);
Cj(l)=(T(l)+S[j/2]) mod Nmax;
if Cj(l) <Ndata
{
Hj(k)=Cj(l);
k =k +1;
}
}
R
Bit shuffling
R
(a)
R
R
or
R
(b)

[Fig. 94]


randomizer
⊞: XOR
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
B2D
B2D: bits-to-decimal
14
T(l)
Modulo operator
+sym. offset (mod Ndata)
$C_j(l) = (T(l) + S_{\lfloor j/2 \rfloor}) \bmod N_{max}$
Memory-index check
$H_j(k)$
$S_{\lfloor j/2 \rfloor}$
14
Random symbol -offset generator
Spreader
$sw = l \pmod 2$
$p_{data}(l)$
MUX
0
1
0
8192
R13
0
1
Toggling

[Fig. 95]

The associated 13th primitive polynomial, $f(x)=1+x^2+x^4+x^8+x^9+x^{12}+x^{13}$
Operation of the PN generator for $0 \le l < N_{max}$ (16384)

$0 \le l < 2$

$$R^l_{[12,11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0,0,0$$

$l = 2$

$$R^2_{[12,11,10,9,8,7,6,5,4,3,2,1,0]} = 0,1,0,1,0,0,0,1,1,0,0,1,1$$

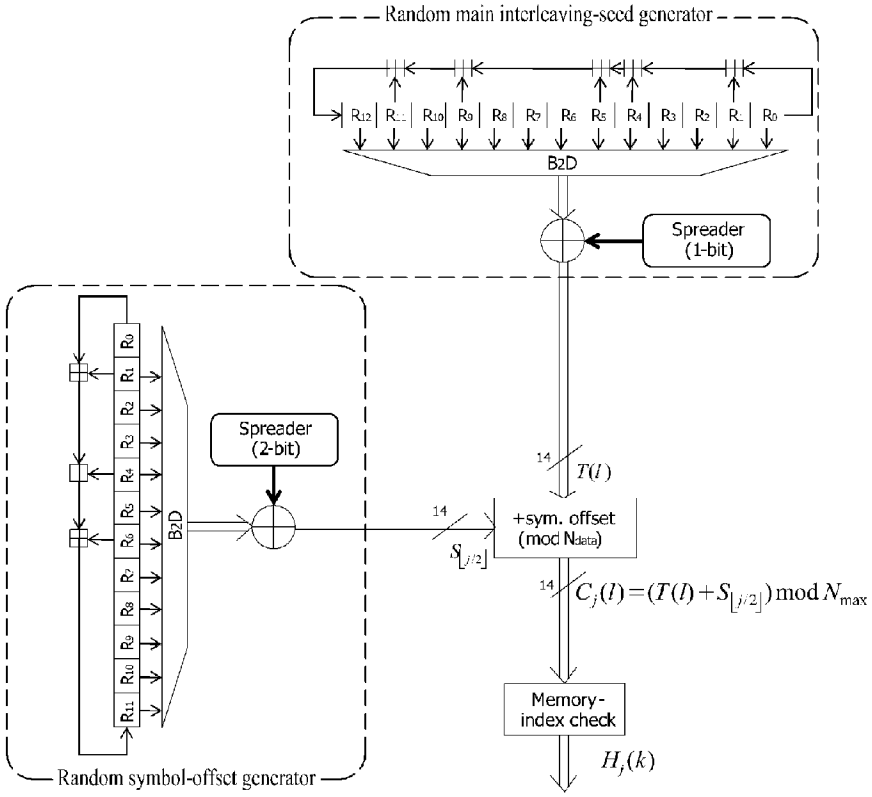
$2 < l < N_{max}$

$$R^l_{[11,10,9,8,7,6,5,4,3,2,1,0]} = R^{l-1}_{[12,11,10,9,8,7,6,5,4,3,2,1]}$$

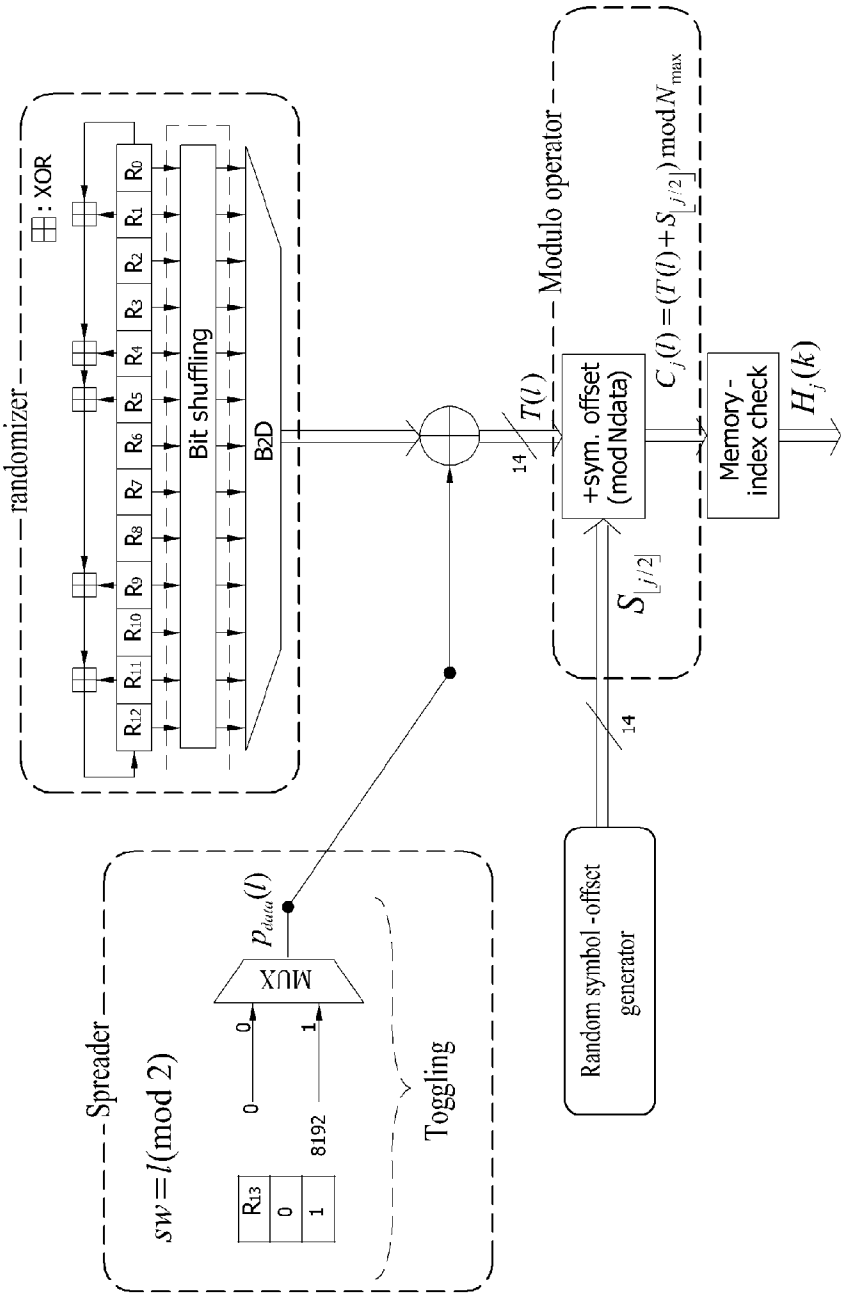
$$R^l_{12} = R^{l-1}_0 \oplus R^{l-1}_1 \oplus R^{l-1}_4 \oplus R^{l-1}_5 \oplus R^{l-1}_9 \oplus R^{l-1}_{11}$$

```
k = 0;
for (l = 0; l < N_max; l = l + 1)
{
    T(l) = P_data(l) + Σ_{n=0}^{12} (2^n × R_n^l);
    C_j(l) = (T(l) + S_⌊j/2⌋) mod N_max;
    if C_j(l) < N_data
    {
        H_j(k) = C_j(l);
        k = k + 1;
    }
}
```

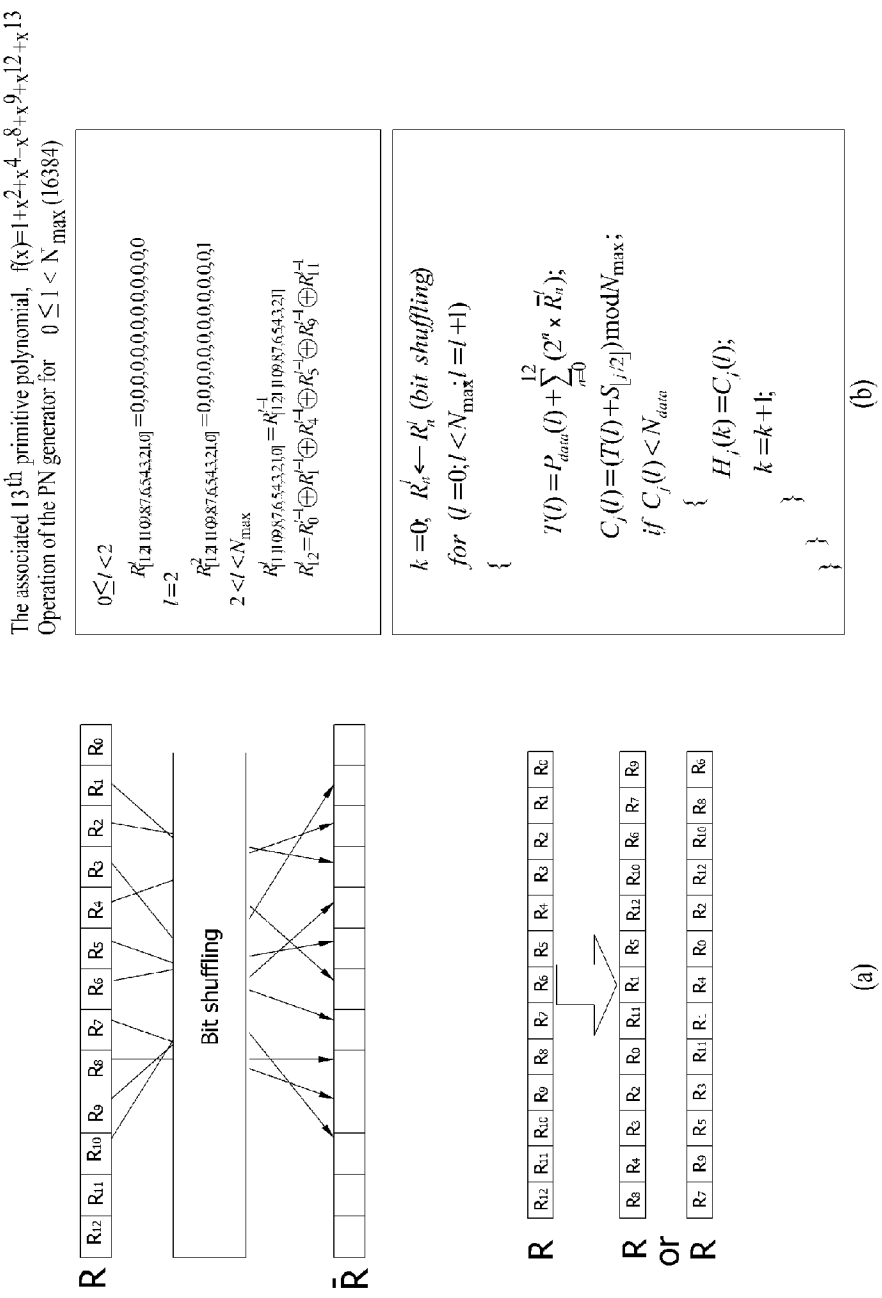
[Fig. 96]


Random main interleaving-seed generator
B2D
Spreader (1-bit)
Spreader (2-bit)
B2D
Random symbol-offset generator
14
T(l)
+sym. offset (mod Ndata)
S⌊j/2⌋
14
Cj(l)=(T(l)+S⌊j/2⌋) mod Nmax
Memory-index check
Hj(k)

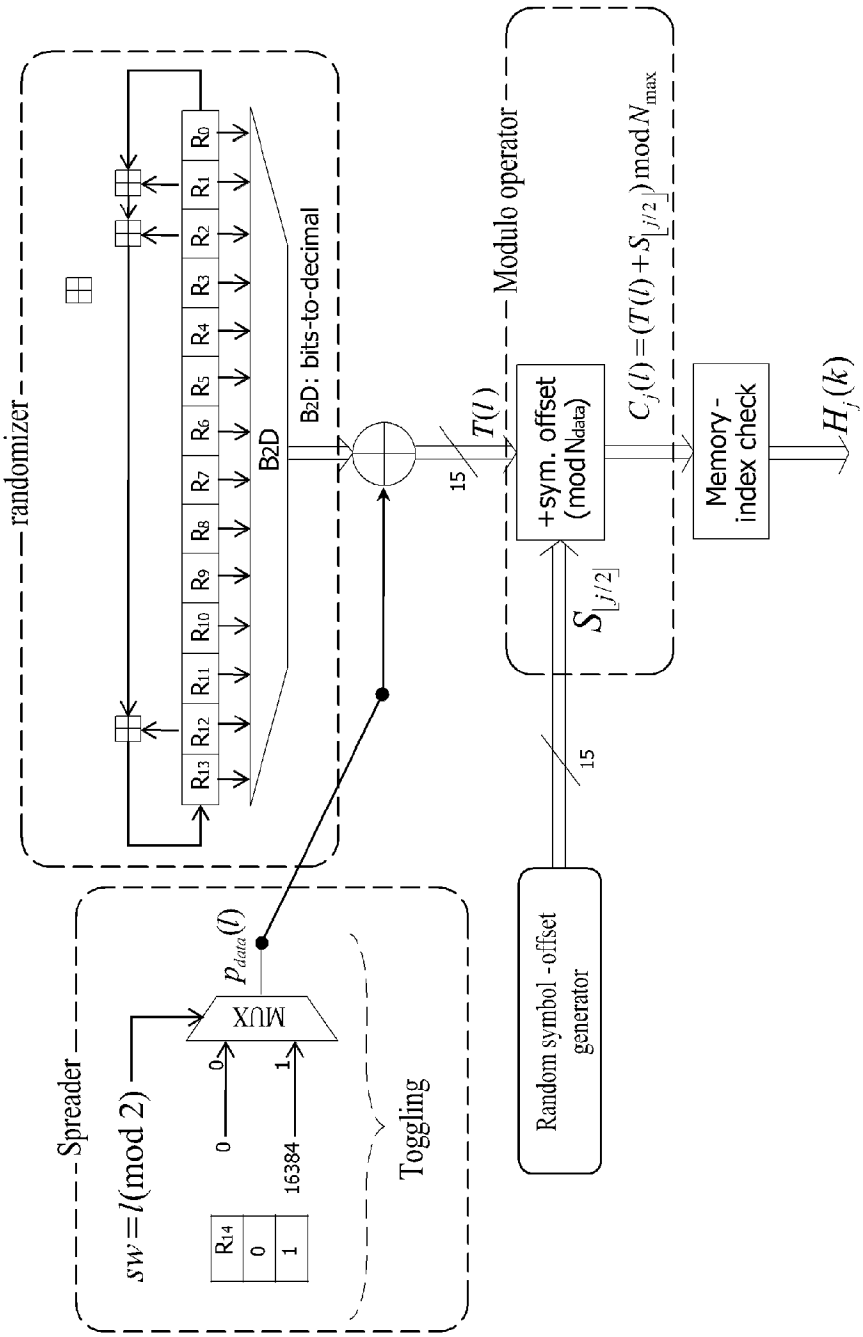
[Fig. 97]


randomizer
⊞: XOR
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
Bit shuffling
B2D
14
T(l)
Modulo operator
+sym. offset (mod Ndata)
$C_j(l) = (T(l) + S_{\lfloor j/2 \rfloor}) \bmod N_{max}$
Memory-index check
$H_j(k)$
$S_{\lfloor j/2 \rfloor}$
14
Random symbol -offset generator
Spreader
$sw = l \pmod 2$
$p_{data}(l)$
MUX
0
1
0
8192
R13
0
1
Toggling

[Fig. 98]


The associated 13th primitive polynomial, f(x)=1+x2+x4+x8+x9+x12+x13
Operation of the PN generator for 0 ≤ l < N max (16384)
0≤l<2
l=2
2<l<Nmax
k =0; (bit shuffling)
for (l =0;l <Nmax;l =l +1)
{
}
if
{
k =k +1;
}
(b)
Bit shuffling
R
R
R
or
R
(a)

[Fig. 99]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
B2D
B2D: bits-to-decimal
15
T(l)
Modulo operator
+sym. offset (mod Ndata)
$C_j(l)=(T(l)+S_{\lfloor j/2 \rfloor}) \bmod N_{max}$
Memory - index check
$H_j(k)$
$S_{\lfloor j/2 \rfloor}$
15
Random symbol - offset generator
Spreader
$sw=l \pmod 2$
MUX
$p_{data}(l)$
0
1
0
16384
R14
0
1
Toggling

[Fig. 100]

The associated 14th polynomial, $f(x)=1+x^2+x^{12}+x^{13}+x^{14}$

Operation of the PN generator for $0 \le l < N_{max}$ (32768)

$0 \le l < 2$

$R^l_{[13,12,11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,0,0,0,0,0,0,0,0,0,0,0,0$

$l = 2$

$R^2_{[13,12,11,10,9,8,7,6,5,4,3,2,1,0]} = 0,0,1,0,0,0,0,0,0,0,0,1,1,1$

$2 < l < N_{max}$

$R^l_{[11,10,9,8,7,6,5,4,3,2,1,0]} = R^{l-1}_{[13,12,11,10,9,8,7,6,5,4,3,2,1]}$

$R^l_{13} = R^{l-1}_0 \oplus R^{l-1}_1 \oplus R^{l-1}_2 \oplus R^{l-1}_{12}$

$k = 0;$

$for\ \ (l = 0; l < N_{max}; l = l+1)$

{

$T(l) = P_{data}(l) + \sum_{n=0}^{13} (2^n \times R^l_n);$

$C_j(l) = (T(l) + S_{\lfloor j/2 \rfloor}) \bmod N_{max};$

$if\ \ C_j(l) < N_{data}$

{

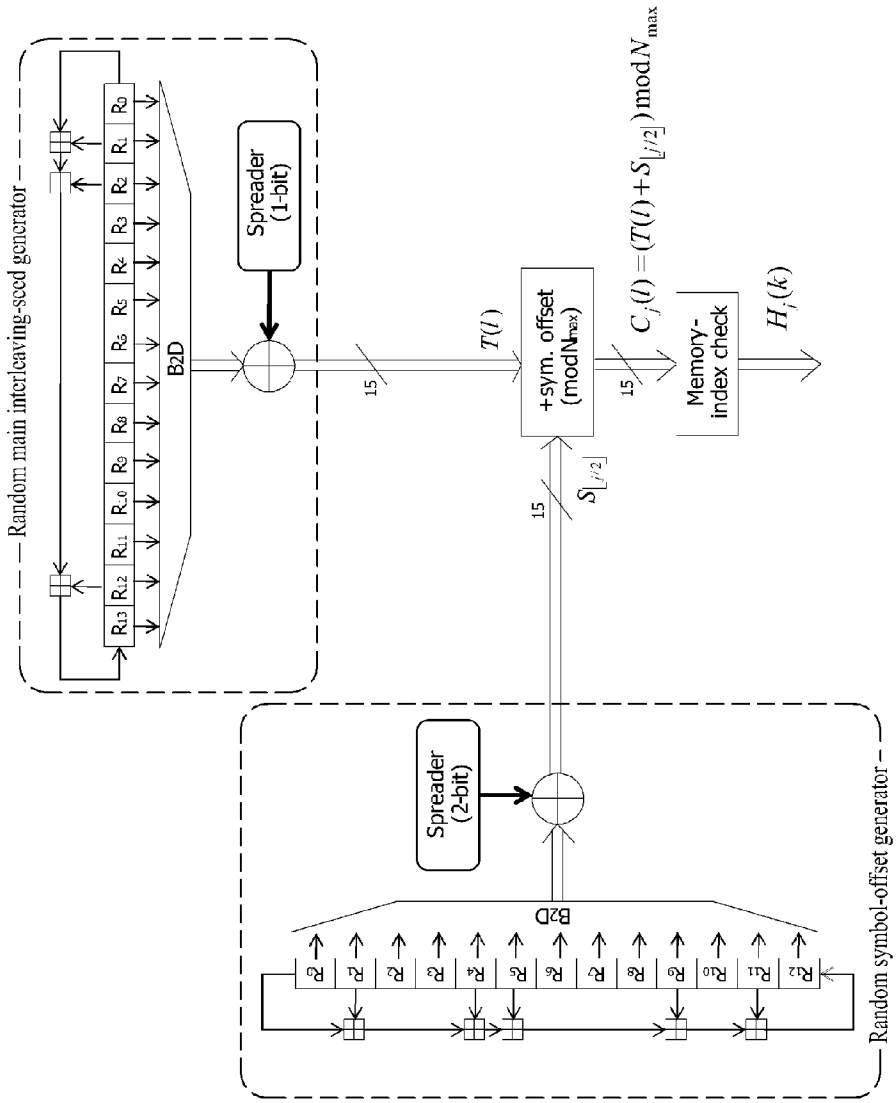
$H_j(k) = C_j(l);$

$k = k+1;$

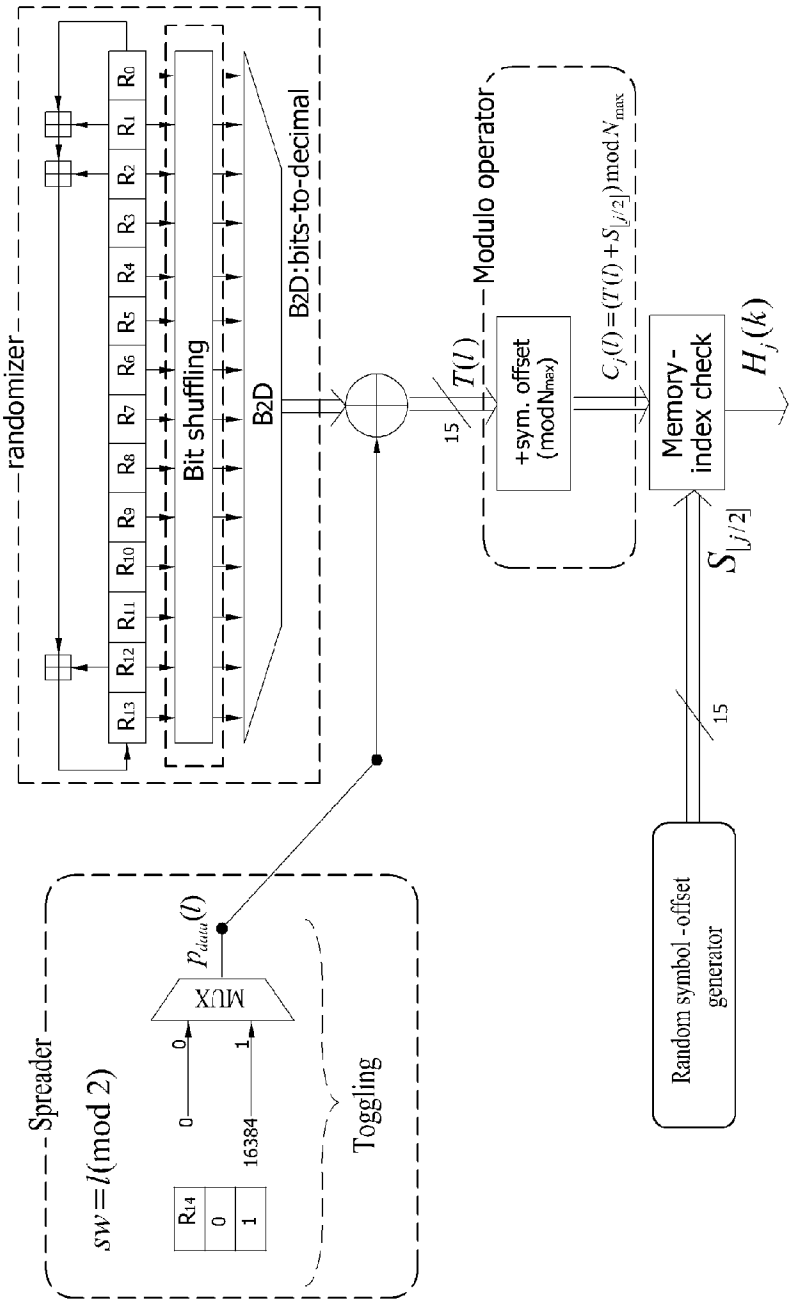
}

}

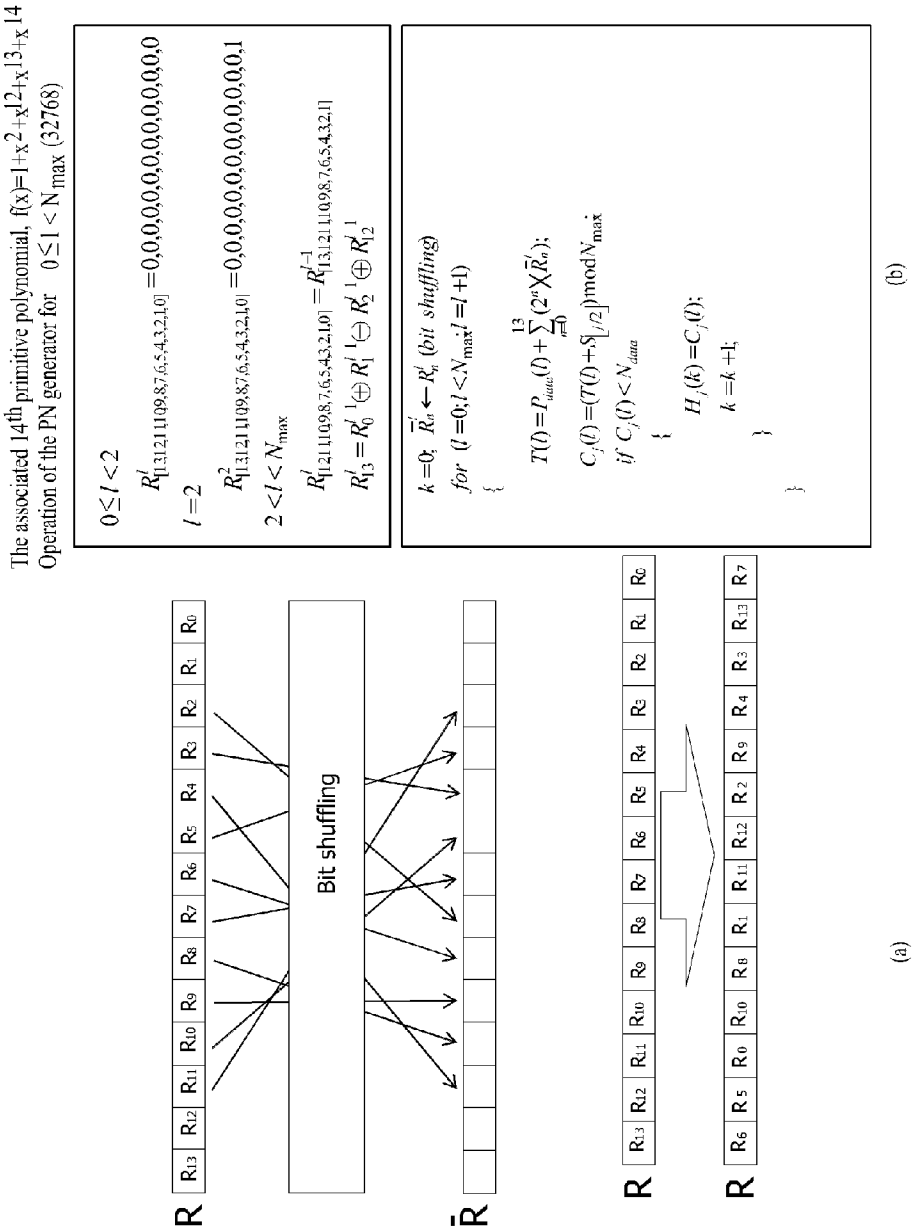
[Fig. 101]


Random main interleaving-seed generator
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
B2D
Spreader (1-bit)
15
T(l)
+sym. offset (modNmax)
15
Memory-index check
Hi(k)
15
Spreader (2-bit)
B2D
Random symbol-offset generator

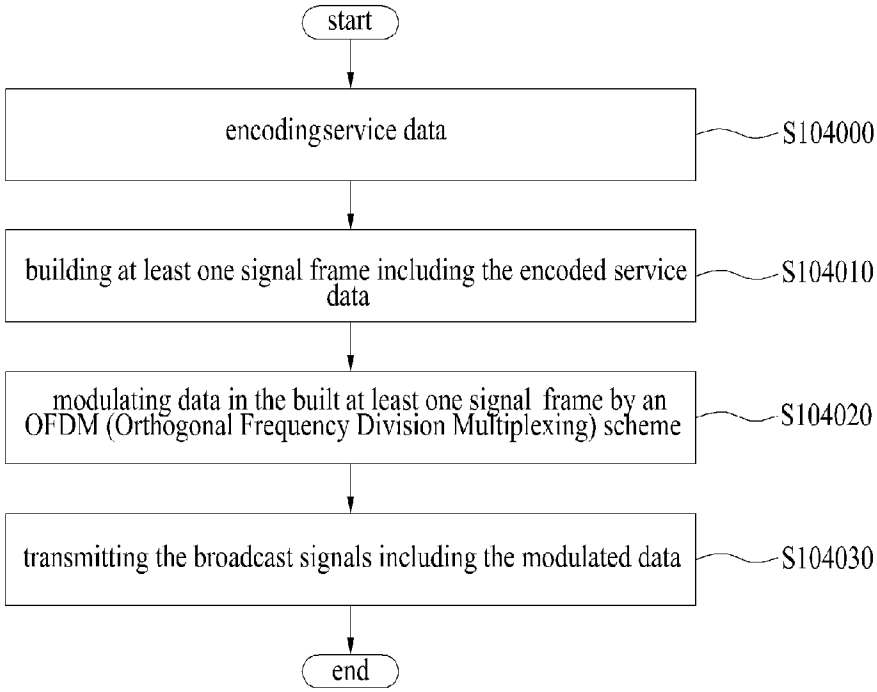
[Fig. 102]


randomizer
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
Bit shuffling
B2D
B2D:bits-to-decimal
Modulo operator
15
T(l)
+sym. offset
(mod Nmax)
Cj(l)=(T(l)+S⌊j/2⌋) mod Nmax
Memory-
index check
Hj(k)
S⌊j/2⌋
15
Random symbol - offset
generator
Spreader
sw=l(mod 2)
pdata(l)
MUX
0
1
0
16384
R14
0
1
Toggling

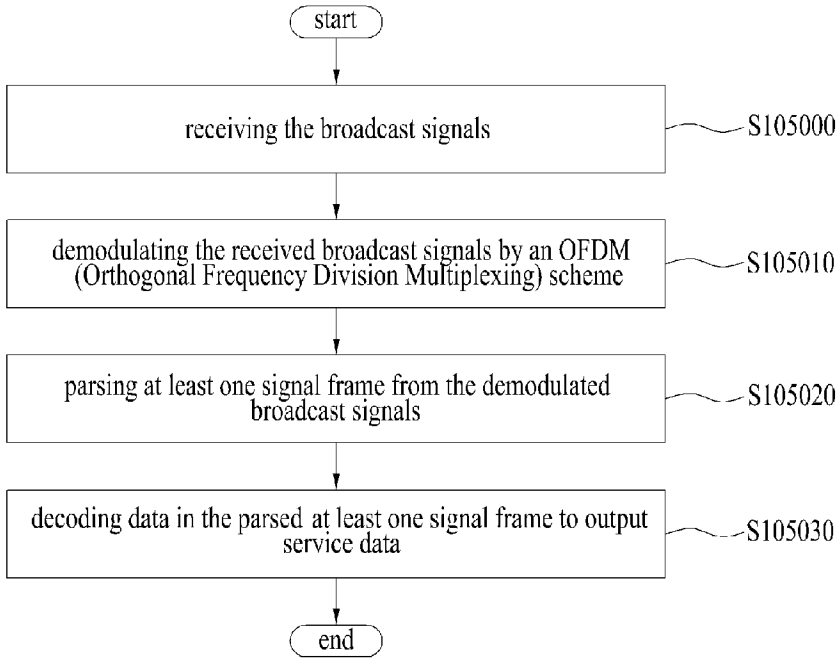
[Fig. 103]


The associated 14th primitive polynomial, f(x)=1+x2+x12+x13+x14
Operation of the PN generator for 0≤l<Nmax (32768)
0≤l<2
l=2
2<l<Nmax
k=0; R̄n ← Rn (bit shuffling)
for (l=0;l<Nmax;l=l+1)
T(l)=Pinit(l)+Σ(2^n X R̄n);
Cj(l)=(T(l)+S[j/2])mod Nmax
if Cj(l)<Ndata
Hj(k)=Cj(l);
k=k+1;
Bit shuffling
R
R̄
(a)
(b)

[Fig. 104]



[Fig. 105]



INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2014/010818

A. CLASSIFICATION OF SUBJECT MATTER

H04N 21/234(2011.01)i, H04N 21/236(2011.01)i, H04J 11/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N 21/234; H04L 27/00; H04N 7/015; H04N 7/173; H03M 13/05; H03M 13/00; H04L 27/28; H03M 13/03; H04N 21/236; H04J 11/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: broadcast signal, encoding, building, frame, OFDM (Orthogonal Frequency Division Multiplex) symbols, FFT (Fast Fourier Transform) size

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012-0189079 A1 (MATTHEW PAUL ATHOL TAYLOR et al.) 26 July 2012 See paragraphs [0078], [0141]-[0144], [0223-[0224], [0314]-[0346], [0431]; claims 1, 8-9; and figures 1, 24, 29-30, 37.	1,3,7
Y		5
A		2,4,6,8
Y	KR 10-2013-0122756 A (LG ELECTRONICS INC.) 08 November 2013 See paragraphs [0023]-[0043]; claim 4; and figures 1-3.	5
A	US 2009-0259913 A1 (SEHO MYUNG et al.) 15 October 2009 See paragraphs [0013]-[0021], [0044]-[0067]; and figure 4.	1-8
A	US 2011-0051825 A1 (TAO TAO et al.) 03 March 2011 See paragraphs [0013]-[0035], [0059]-[0062]; and figure 2.	1-8
A	JP 2011-199859 A (MASPRO DENKOH CORP.) 06 October 2011 See paragraphs [0042]-[0075]; and figures 1-3.	1-8

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 February 2015 (11.02.2015)	**11 February 2015 (11.02.2015)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea
Facsimile No. ++82 42 472 3473

Authorized officer
LEE, Jin Ick
Telephone No. +82-42-481-5770

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2014/010818

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0189079 A1	26/07/2012	AU 2008-229928 A1	14/05/2009
		AU 2008-230048 A1	14/05/2009
		AU 2008-230049 A1	14/05/2009
		AU 2008-237590 A1	14/05/2009
		AU 2008-237593 A1	14/05/2009
		AU 2008-330660 A1	04/06/2009
		AU 2008-330661 A1	04/06/2009
		AU 2008-330661 A2	04/06/2009
		AU 2008-330666 A1	04/06/2009
		AU 2008-330672 A1	04/06/2009
		AU 2008-330672 A2	04/06/2009
		AU 2008-330716 A1	04/06/2009
		AU 2008-330716 A2	04/06/2009
		AU 2008-330816 A1	04/06/2009
		AU 2008-330816 A2	04/06/2009
		CN 101425992 A	06/05/2009
		CN 101425992 B	14/08/2013
		CN 101425993 A	06/05/2009
		CN 101425993 B	28/08/2013
		CN 101425994 A	06/05/2009
		CN 101425994 B	14/08/2013
		CN 101425995 A	06/05/2009
		CN 101425995 B	11/06/2014
		CN 101425996 A	06/05/2009
		CN 101425996 B	11/06/2014
		CN 101425997 A	06/05/2009
		CN 101425997 B	11/06/2014
		CN 101425998 A	06/05/2009
		CN 101425998 B	01/05/2013
		CN 101510865 A	19/08/2009
		CN 101510865 B	11/12/2013
		CN 101843023 A	22/09/2010
		CN 101843023 B	17/04/2013
		CN 101874350 A	27/10/2010
		CN 101874351 A	27/10/2010
		CN 101874352 A	27/10/2010
		CN 101874352 B	17/04/2013
		CN 101911504 A	08/12/2010
		CN 101911504 B	11/06/2014
		CN 101911505 A	08/12/2010
		CN 101911506 A	08/12/2010
		CN 101911506 B	02/04/2014
		CN 101911507 A	08/12/2010
		CN 101911507 B	16/10/2013
		CN 103401830 A	20/11/2013
		CN 103560993 A	05/02/2014
		DK 2056464 T3	18/02/2013
		DK 2056510 T3	21/05/2013
		DK 2056549 T3	04/02/2013

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		DK 2056550 T3	17/06/2013
		EP 1463255 A1	29/09/2004
		EP 1463256 A1	29/09/2004
		EP 1463256 B1	10/05/2006
		EP 1662739 A1	31/05/2006
		EP 1662739 B1	24/09/2008
		EP 1662739 B9	05/08/2009
		EP 1662740 A1	31/05/2006
		EP 1662740 B1	30/04/2008
		EP 1931097 A1	11/06/2008
		EP 1931097 B1	07/10/2009
		EP 2056463 A2	06/05/2009
		EP 2056463 A3	26/08/2009
		EP 2056463 B1	04/12/2013
		EP 2056464 A2	06/05/2009
		EP 2056464 A3	16/09/2009
		EP 2056464 B1	12/12/2012
		EP 2056466 A1	06/05/2009
		EP 2056466 B1	29/02/2012
		EP 2056467 A1	06/05/2009
		EP 2056467 B1	29/02/2012
		EP 2056468 A2	06/05/2009
		EP 2056468 A3	26/08/2009
		EP 2056468 B1	03/07/2013
		EP 2056469 A1	06/05/2009
		EP 2056469 B1	21/03/2012
		EP 2056470 A1	06/05/2009
		EP 2056470 B1	21/03/2012
		EP 2056470 B9	01/08/2012
		EP 2056471 A1	06/05/2009
		EP 2056471 B1	16/09/2009
		EP 2056472 A1	06/05/2009
		EP 2056472 B1	09/12/2009
		EP 2056473 A1	06/05/2009
		EP 2056474 A1	06/05/2009
		EP 2056475 A2	06/05/2009
		EP 2056475 A3	13/05/2009
		EP 2056476 A2	06/05/2009
		EP 2056476 A3	13/05/2009
		EP 2056477 A1	06/05/2009
		EP 2056478 A1	06/05/2009
		EP 2056510 A2	06/05/2009
		EP 2056510 A3	16/09/2009
		EP 2056510 B1	03/04/2013
		EP 2056549 A2	06/05/2009
		EP 2056549 A3	16/09/2009
		EP 2056549 B1	12/12/2012
		EP 2056550 A2	06/05/2009
		EP 2056550 A3	16/09/2009
		EP 2056550 B1	24/04/2013

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2014/010818

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		EP 2117195 A1	11/11/2009
		EP 2117195 B1	04/08/2010
		EP 2204002 A1	07/07/2010
		EP 2204002 B1	01/05/2013
		EP 2214318 A1	04/08/2010
		EP 2214319 A1	04/08/2010
		EP 2214320 A1	04/08/2010
		EP 2214321 A1	04/08/2010
		EP 2216907 A1	11/08/2010
		EP 2216908 A1	11/08/2010
		EP 2237427 A2	06/10/2010
		EP 2237429 A1	06/10/2010
		EP 2237430 A2	06/10/2010
		EP 2237431 A2	06/10/2010
		EP 2237432 A2	06/10/2010
		EP 2247055 A1	03/11/2010
		EP 2247055 B1	14/03/2012
		EP 2333963 A2	15/06/2011
		EP 2333963 A3	28/03/2012
		EP 2333963 B1	03/04/2013
		EP 2333964 A2	15/06/2011
		EP 2333964 A3	08/08/2012
		EP 2333964 B1	09/04/2014
		EP 2333965 A2	15/06/2011
		EP 2333965 A3	30/05/2012
		EP 2333965 B1	04/12/2013
		EP 2333966 A2	15/06/2011
		EP 2333966 A3	08/08/2012
		EP 2333966 B1	04/12/2013
		EP 2333967 A2	15/06/2011
		EP 2333967 A3	28/03/2012
		EP 2333967 B1	19/02/2014
		EP 2341629 A2	06/07/2011
		EP 2341629 A3	30/05/2012
		EP 2341629 B1	03/07/2013
		EP 2403147 A2	04/01/2012
		EP 2403147 A3	08/08/2012
		EP 2405584 A2	11/01/2012
		EP 2405584 A3	08/08/2012
		EP 2421158 A2	22/02/2012
		EP 2421158 A3	03/10/2012
		EP 2421158 B1	04/12/2013
		EP 2421159 A2	22/02/2012
		EP 2421159 A3	15/08/2012
		EP 2421160 A2	22/02/2012
		EP 2421160 A3	03/10/2012
		EP 2421160 B1	16/10/2013
		EP 2421161 A2	22/02/2012
		EP 2421161 A3	28/03/2012
		EP 2421161 B1	04/09/2013

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2014/010818

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		EP 2421162 A2	22/02/2012
		EP 2421162 A3	28/03/2012
		EP 2421162 B1	31/07/2013
		EP 2421163 A2	22/02/2012
		EP 2421163 A3	29/02/2012
		EP 2421163 B1	13/03/2013
		EP 2421164 A2	22/02/2012
		EP 2421164 A3	28/03/2012
		EP 2421164 B1	21/08/2013
		EP 2421165 A2	22/02/2012
		EP 2421165 A3	28/03/2012
		EP 2421165 B1	10/07/2013
		EP 2421166 A2	22/02/2012
		EP 2421166 A3	29/02/2012
		EP 2421166 B1	13/03/2013
		EP 2421167 A2	22/02/2012
		EP 2421167 A3	03/10/2012
		EP 2421167 B1	18/12/2013
		EP 2421168 A2	22/02/2012
		EP 2421168 A3	03/10/2012
		EP 2421168 B1	04/12/2013
		EP 2421169 A2	22/02/2012
		EP 2421169 A3	15/08/2012
		EP 2426823 A2	07/03/2012
		EP 2426823 A3	30/05/2012
		EP 2426823 B1	31/07/2013
		EP 2456077 A2	23/05/2012
		EP 2456077 A3	30/05/2012
		EP 2456077 B1	31/07/2013
		EP 2469715 A2	27/06/2012
		EP 2469715 A3	22/10/2014
		EP 2469785 A2	27/06/2012
		EP 2469785 A3	22/10/2014
		EP 2509270 A2	10/10/2012
		EP 2509270 A3	29/10/2014
		ES 2398851 T3	22/03/2013
		ES 2399157 T3	26/03/2013
		ES 2407505 T3	12/06/2013
		ES 2416356 T3	31/07/2013
		GB 2454193 A	06/05/2009
		GB 2454194 A	06/05/2009
		GB 2454195 A	06/05/2009
		GB 2454196 A	06/05/2009
		GB 2454267 A	06/05/2009
		GB 2454307 A	06/05/2009
		GB 2454308 A	06/05/2009
		GB 2454311 A	06/05/2009
		GB 2454312 A	06/05/2009
		GB 2454316 A	06/05/2009
		GB 2454317 A	06/05/2009

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2014/010818

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		GB 2454318 A	06/05/2009
		GB 2454319 A	06/05/2009
		GB 2454321 A	06/05/2009
		GB 2454322 A	06/05/2009
		GB 2454323 A	06/05/2009
		GB 2454324 A	06/05/2009
		GB 2454722 A	20/05/2009
		GB 2455071 A	03/06/2009
		GB 2462039 A	27/01/2010
		GB 2462040 A	27/01/2010
		GB 2462041 A	27/01/2010
		GB 2462042 A	27/01/2010
		GB 2462749 A	24/02/2010
		GB 2462750 A	24/02/2010
		JP 05048629 B2	17/10/2012
		JP 05248983 B2	31/07/2013
		JP 05252552 B2	31/07/2013
		JP 05252553 B2	31/07/2013
		JP 05253092 B2	31/07/2013
		JP 05253093 B2	31/07/2013
		JP 05253094 B2	31/07/2013
		JP 05273054 B2	28/08/2013
		JP 05273055 B2	28/08/2013
		JP 05288212 B2	11/09/2013
		JP 05288213 B2	11/09/2013
		JP 05359881 B2	04/12/2013
		JP 05359882 B2	04/12/2013
		JP 05371374 B2	18/12/2013
		JP 2009-112008 A	21/05/2009
		JP 2009-112009 A	21/05/2009
		JP 2009-112010 A	21/05/2009
		JP 2009-112011 A	21/05/2009
		JP 2009-112012 A	21/05/2009
		JP 2009-112013 A	21/05/2009
		JP 2009-153109 A	09/07/2009
		JP 2009-239886 A	15/10/2009
		WO 2009-069513 A1	04/06/2009
		WO 2009-069580 A1	04/06/2009
		WO 2009-069617 A1	04/06/2009
		WO 2009-069618 A1	04/06/2009
		WO 2009-069623 A1	04/06/2009
KR 10-2013-0122756 A	08/11/2013	KR 10-2013-0121881 A	06/11/2013
		US 2013-0235952 A1	12/09/2013
		US 2013-0243116 A1	19/09/2013
		WO 2012-067362 A2	24/05/2012
		WO 2012-067362 A3	12/07/2012
		WO 2012-070837 A2	31/05/2012
		WO 2012-070837 A3	13/09/2012

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/KR2014/010818

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009-0259913 A1	15/10/2009	AU 2009-220323 A1	11/09/2009
		CN 102017558 A	13/04/2011
		CN 102017558 B	09/07/2014
		CN 103297193 A	11/09/2013
		CN 103354481 A	16/10/2013
		EP 2099150 A2	09/09/2009
		EP 2099150 A3	16/12/2009
		EP 2099150 B1	18/04/2012
		EP 2101430 A2	16/09/2009
		EP 2101430 A3	16/12/2009
		EP 2101430 B1	15/01/2014
		EP 2101431 A2	16/09/2009
		EP 2101431 A3	16/12/2009
		EP 2101431 B1	12/02/2014
		EP 2388941 A2	23/11/2011
		EP 2388941 A3	12/12/2012
		EP 2469746 A1	27/06/2012
		JP 05441270 B2	12/03/2014
		JP 2011-514094 A	28/04/2011
		JP 2014-060758 A	03/04/2014
		KR 10-2009-0094738 A	08/09/2009
		KR 10-2009-0094744 A	08/09/2009
		KR 10-2014-0036346 A	25/03/2014
		KR 10-2014-0036347 A	25/03/2014
		TW 201004196 A	16/01/2010
		TW I376907 B	11/11/2012
		US 8352844 B2	08/01/2013
		WO 2009-110739 A2	11/09/2009
		WO 2009-110739 A3	29/10/2009
US 2011-0051825 A1	03/03/2011	CN 100502380 C	17/06/2009
		CN 1960357 A	09/05/2007
		US 8428160 B2	23/04/2013
		WO 2008-049282 A1	02/05/2008
JP 2011-199859 A	06/10/2011	WO 2011-105569 A1	01/09/2011