

Description

DATA PROCESSING APPARATUS AND DATA PROCESSING METHOD

Technical Field

[0001]

This invention relates to a data processing apparatus and a data processing method, and particularly to a data processing apparatus and a data processing method which make it possible to improve the tolerance to burst errors or erasure of, for example, an LDPC (Low Density Parity Check) code.

Background Art

[0002]

The LDPC code has a high error correction capacity and, in recent years, begins to be adopted widely in transmission systems including satellite digital broadcasting systems such as, for example, the DVB-S.2 system used in Europe (refer to, for example, Non-Patent Document 1). Further, it is investigated to adopt the LDPC code also in terrestrial digital broadcasting of the next generation.

[0003]

It is being found by recent research that a

performance proximate to the Shannon limit is provided by the LDPC code as the code length is increased similarly to a turbo code and so forth. Further, since the LDPC code has a property that the minimum distance increases in proportion to the code length, it has a characteristic that it has a superior block error probability characteristic. Also it is advantageous that a so-called error floor phenomenon which is observed in a decoding characteristic of the turbo code and so forth little occurs.

[0004]

In the following, such an LDPC code as described above is described particularly. It is to be noted that the LDPC code is a linear code, and although it is not necessarily be a two-dimensional code, the following description is given under the assumption that it is a two-dimensional code.

[0005]

The LDPC code has the most significant characteristic in that a parity check matrix which defines the LDPC code is a sparse matrix. Here, the sparse matrix is a matrix in which the number of those elements whose value is "1" is very small (matrix in which almost all elements are 0).

[0006]

FIG. 1 shows an example of a parity check matrix H of an LDPC code.

[0007]

In the parity check matrix H of FIG. 1, the weight of each column (column weight) (number of "1") (weight) is "3" and the weight of each row (row weight) is "6."

[0008]

In encoding by LDPC codes (LDPC encoding), for example, a generator matrix G is produced based on a parity check matrix H and this generator matrix G is multiplied by two-dimensional information bits to produce a codeword (LDPC code).

[0009]

In particular, an encoding apparatus which carries out LDPC encoding first calculates a generator matrix G which satisfies an expression $GH^T = 0$ together with a transposed matrix H^T of a parity check matrix H . Here, if the generator matrix G is a $K \times N$ matrix, then the encoding apparatus multiplies the generator matrix G by a bit string (vector u) of K information bits to produce a codeword c ($= uG$) of N bits. The codeword (LDPC code) produced by the encoding apparatus is received by the reception side through a predetermined communication path.

[0010]

Decoding of the LDPC code can be carried out using an algorithm proposed as probabilistic decoding (Probabilistic Decoding) by the Gallager, that is, a message passing algorithm by belief propagation on a so-called Tanner graph including a variable node (also called message node) and a check node. In the following description, each of the variable node and the check node is suitably referred to simply as node.

[0011]

FIG. 2 illustrates a procedure of decoding of an LDPC code.

[0012]

It is to be noted that, in the following description, a real number value where the "0" likelihood in the value of the n th code bit of an LDPC code (one codeword) received by the reception side is represented in a log likelihood ratio is suitably referred to as reception value u_{0i} . Further, a message outputted from a check node is represented by u_j and a message outputted from a variable node is represented by v_i .

[0013]

First, in decoding of an LDPC code, as seen in FIG. 2, an LDPC code is received and a message (check node

message) u_j is initialized to "0" and besides a variable k which assumes an integer as a counter of repeated processes is initialized to "0" at step S11, whereafter the processing advances to step S12. At step S12, mathematical operation represented by an expression (1) (variable node mathematical operation) is carried out based on the reception value u_{oi} obtained by the reception of the LDPC code to determine a message (variable node message) v_i . Further, mathematical operation represented by an expression (2) (check node mathematical operation) is carried out based on the message v_i to determine the message u_j .

[0014]

[Expression 1]

$$v_i = u_{oi} + \sum_{j=1}^{d_v-1} u_j \quad \dots (1)$$

[0015]

[Expression 2]

$$\tanh\left(\frac{u_j}{2}\right) = \prod_{i=1}^{d_c-1} \tanh\left(\frac{v_i}{2}\right) \quad \dots (2)$$

[0016]

Here, d_v and d_c in the expression (1) and the expression (2) are parameters which can be selected arbitrarily and represent the number of "1s" in a

vertical direction (column) and a horizontal direction (row) of the parity check matrix H . For example, in the case of a $(3, 6)$ code, $d_v = 3$ and $d_c = 6$.

[0017]

It is to be noted that, in the variable node mathematical operation of the expression (1) and the check node mathematical operation of the expression (2), the range of the mathematical operation is 1 to $d_v - 1$ or 1 to $d_c - 1$ because a message inputted from an edge (line interconnecting a variable node and a check node) from which a message is to be outputted is not made an object of the mathematical operation. Meanwhile, the check node mathematical operation of the expression (2) is carried out by producing in advance a table of a function $R(v_1, v_2)$ represented by an expression (3) defined by one output with respect to two inputs v_1 and v_2 and using the table successively (recursively) as represented by an expression (4).

[0018]

[Expression 3]

$$x = 2 \tanh^{-1} \{ \tanh(v_1/2) \tanh(v_2/2) \} = R(v_1, v_2) \quad \dots (3)$$

[0019]

[Expression 4]

$$u_i = R(v_1, R(v_2, R(v_3, \dots R(v_{d_c-2}, v_{d_c-1})))) \quad \dots (4)$$

[0020]

At step S12, the variable k is incremented by "1" further, and the processing advances to step S13. At step S13, it is decided whether or not the variable k is higher than a predetermined repeated decoding time number C. If it is decided at step S13 that the variable k is not higher than C, then the processing returns to step S12, and similar processing is repeated thereafter.

[0021]

On the other hand, if it is decided at step S13 that the variable k is higher than C, then the processing advances to step S14, at which a message v_i as a decoding result to be outputted finally by carrying out mathematical operation represented by an expression (5) is determined and outputted, thereby ending the decoding process of the LDPC code.

[0022]

[Expression 5]

$$v_i = u_{oi} + \sum_{j=1}^{d_v} u_j \quad \dots (5)$$

[0023]

Here, the mathematical operation of the expression (5) is carried out, different from the variable node mathematical operation of the expression (1), using

messages u_j from all edges connecting to the variable node.

[0024]

FIG. 3 illustrates an example of the parity check matrix H of a $(3, 6)$ LDPC code (encoding rate: $1/2$, code length: 12).

[0025]

In the parity check matrix H of FIG. 3, the weight of a column is 3 and the weight of a row is 6 similarly as in FIG. 1.

[0026]

FIG. 4 shows a Tanner graph of the parity check matrix H of FIG. 3.

[0027]

Here, in FIG. 4, a check node is represented by "+," and a variable node is represented by "=". A check node and a variable node correspond to a row and a column of the parity check matrix H , respectively. A connection between a check node and a variable node is an edge and corresponds to "1" of an element of the parity check matrix.

[0028]

In particular, where the element in the j th row of the i th column of the parity check matrix is 1, the i th variable node (node of "=") from above and the j th check

node (node of "+") from above are connected by an edge. The edge represents that a code bit corresponding to the variable node has a constraint condition corresponding to the check node.

[0029]

In the sum product algorithm (Sum Product Algorithm) which is a decoding method for LDPC codes, variable node mathematical operation and check node mathematical cooperation are carried out repetitively.

[0030]

FIG. 5 illustrates the variable node mathematical operation carried out with regard to a variable node.

[0031]

With regard to the variable node, a message v_i corresponding to an edge to be calculated is determined by variable node mathematical operation of the expression (1) which uses messages u_1 and u_2 from the remaining edges connecting to the variable node and the reception value u_{0i} . Also a message corresponding to any other edge is determined similarly.

[0032]

FIG. 6 illustrates the check node mathematical operation carried out at a check node.

[0033]

Here, the check node mathematical operation of the expression (2) can be carried out by rewriting the expression (2) into an expression (6) using the relationship of an expression $a \times b = \exp\{\ln(|a|) + \ln(|b|)\} \times \text{sign}(a) \times \text{sign}(b)$. It is to be noted that $\text{sign}(x)$ is 1 where $x \geq 0$ but is -1 where $x < 0$.

[0034]

[Expression 6]

$$\begin{aligned}
 u_j &= 2 \tanh^{-1} \left(\prod_{i=1}^{d_c-1} \tanh \left(\frac{v_i}{2} \right) \right) \\
 &= 2 \tanh^{-1} \left[\exp \left\{ \sum_{i=1}^{d_c-1} \ln \left(\tanh \left(\frac{v_i}{2} \right) \right) \right\} \times \prod_{i=1}^{d_c-1} \text{sign} \left(\tanh \left(\frac{v_i}{2} \right) \right) \right] \\
 &= 2 \tanh^{-1} \left[\exp \left\{ - \left(\sum_{i=1}^{d_c-1} - \ln \left(\tanh \left(\frac{|v_i|}{2} \right) \right) \right) \right\} \times \prod_{i=1}^{d_c-1} \text{sign}(v_i) \right] \quad \dots (6)
 \end{aligned}$$

[0035]

Further, if, where $x \geq 0$, a function $\phi(x)$ is defined as an expression $\phi(x) = \ln(\tanh(x/2))$, then since an expression $\phi^{-1}(x) = 2 \tanh^{-1}(e^x)$ is satisfied, the expression (6) can be transformed into an expression (7).

[0036]

[Expression 7]

$$u_j = \phi^{-1} \left(\sum_{i=1}^{d_c-1} \phi(|v_i|) \right) \times \prod_{i=1}^{d_c-1} \text{sign}(v_i) \quad \dots (7)$$

[0037]

At the check node, the check node mathematical operation of the expression (2) is carried out in accordance with the expression (7).

[0038]

In particular, at the check node, the message u_j corresponding to the edge to be calculated is determined by check node mathematical operation of the expression (7) using messages v_1, v_2, v_3, v_4 and v_5 from the remaining edges connecting to the check node. Also a message corresponding to any other edge is determined in a similar manner.

[0039]

It is to be noted that the function $\phi(x)$ of the expression (7) can be represented also as $\phi(x) = \ln((e^x + 1)/(e^x - 1))$, and where $x > 0$, $\phi(x) = \phi^{-1}(x)$. When the functions $\phi(x)$ and $\phi^{-1}(x)$ are incorporated in hardware, while they are sometimes incorporated using an LUT (Look Up Table), such LUTs become the same LUT.

[0040]

Non-Patent Document 1: DVB-S.2: ETSI EN 302 307
V1.1.2 (2006-06)

Disclosure of Invention

Technical Problem

[0041]

Although it has been known that LDPC codes exhibit a very high capacity in an AWGN (Additive White Gaussian Noise) communication path, in recent years it has become clear that even in the other communication paths they have a higher error correction capacity than conventional convolution codes or connection codes of convolution codes and RS (Reed Solomon) codes.

[0042]

In short, if a code having a good performance in an AWGN communication path is selected, then the code frequently has a better performance than the other codes also in other communication paths.

[0043]

Incidentally, it has been proposed that, for example, where LDPC codes are applied to terrestrial digital broadcasting, LDPC codes perceived in the DVB-S.2 standards and a modulation system prescribed in the DVB-T standards are combined and an interleaver (bitinterleaver) for interleaving code bits of an LDPC code is provided between the LDPC encoding and modulation to improve the performance of LDPC codes in an AWGN communication path.

[0044]

However, in a communication path supposed with

regard to ground waves, burst errors or erasure sometimes occurs. For example, in an OFDM (Orthogonal Frequency Division Multiplexing) system, in a multi-path environment wherein the D/U (Desired to Undesired Ratio) is 0 dB (power of Undesired = echo is equal to the power of Desired = main path), the power of a particular symbol becomes zero (erasure) in response to a delay of an echo (paths other than the main path).

[0045]

Further, also in a flutter (communication path in which an echo whose delay is zero and to which a Doppler (doppler) frequency is applied is added), where the D/U is 0 dB, a case wherein the power of an entire OFDM symbol at a specific point of time is reduced to zero by the Doppler frequency occurs.

[0046]

Furthermore, burst errors sometimes arise from a situation of a wiring line from an antenna to a reception apparatus or instability of the power supply.

[0047]

Conventionally, also in a communication path in which such burst errors or erasure occurs as described above, an error correction code of a high performance is used in an AWGN communication path.

[0048]

Meanwhile, in decoding of LDPC codes, since variable node mathematical operation of the expression (1) wherein addition of (reception values u_{0i} of) code bits of an LDPC code as seen in FIG. 5 is carried out in a column of the parity check matrix H and hence a variable node corresponding to a code bit of the LDPC code, if an error occurs with the code bit used for the variable node mathematical operation, then the accuracy of a message to be determined drops.

[0049]

Then, since, in decoding of the LDPC code, the message determined at the variable node connecting to the check node is used to carry out check node mathematical operation of the expression (7) at the check node, if the number of check nodes where (code bits of the LDPC code corresponding to) a plurality of variable nodes connected thereto exhibit an error (including erasure) at the same time becomes great, then the performance of the decoding deteriorates.

[0050]

For example, if two or more of the variable nodes connected to the check node suffer from erasure at the same time, then the check node returns a message that the

probability that the value may be 0 and the probability that the value may be 1 are equal to each other to all variable nodes. In this instance, those check nodes to which the message of the equal probabilities does not contribute to one cycle of decoding processing (one set of variable node mathematical operation and check node mathematical operation), and as a result, an increased number of times of repetition of decoding processing are required. Consequently, the performance of the decoding deteriorates. Further, the power consumption of a reception apparatus which carries out decoding of the LDPC code increases.

[0051]

Accordingly, at present, a technique for improving the tolerance to burst errors or erasure while the performance in an AWGN communication path is maintained is demanded.

[0052]

Here, it has been proposed that the performance of LDPC codes in an AWGN communication path is improved by providing an interleaver for interleaving code bits of an LDPC code between LDPC encoding and modulation as described hereinabove, and if the interleaver can carry out interleave of lowering the probability that a

plurality of ones of (code bits of LDPC codes corresponding to) variable nodes connected to a check node may exhibit an error, then the performance of decoding can be improved.

[0053]

The present invention has been made in view of such a situation as described above and makes it possible to improve the tolerance to errors of code bits of an LDPC code such as burst errors or erasure.

Technical Solution

[0054]

A data processing apparatus of one aspect of the present invention is a data processing apparatus which interleaves data, including parity interleave means for carrying out parity interleave of interleaving parity bits of an LDPC code obtained by carrying out LDPC encoding in accordance with a parity check matrix in which a parity matrix which is a portion corresponding to parity bits of an LDPC (Low Density Parity Check) code has a staircase structure individually to positions of other ones of the parity bits.

[0055]

A data processing method of the one aspect of the present invention is a data processing method for a data

processing apparatus which interleaves data, including a step carried out by the data processing apparatus of carrying out parity interleave of interleaving parity bits of an LDPC code obtained by carrying out LDPC encoding in accordance with a parity check matrix in which a parity matrix which is a portion corresponding to parity bits of an LDPC (Low Density Parity Check) code has a staircase structure individually to positions of other ones of the parity bits.

[0056]

In such one aspect of the present invention as described above, the parity interleave of interleaving parity bits of an LDPC code obtained by carrying out LDPC encoding in accordance with a parity check matrix in which a parity matrix which is a portion corresponding to parity bits of an LDPC code has a staircase structure individually to positions of other ones of the parity bits is carried out.

[0057]

It is to be noted that the data processing apparatus may be an independent apparatus or may be an internal block which composes one apparatus.

Advantageous Effect

[0058]

According to the one aspect of the present invention, the tolerance of code bits of an LDPC code to errors can be improved.

Brief Description of Drawings

[0059]

[FIG. 1] FIG. 1 is a view illustrating a parity check matrix H of an LDPC code.

[FIG. 2] FIG. 2 is a flow chart illustrating a decoding procedure of an LDPC code.

[FIG. 3] FIG. 3 is a view illustrating an example of a parity error matrix of an LDPC code.

[FIG. 4] FIG. 4 is a view showing a Tanner graph of a parity check matrix.

[FIG. 5] FIG. 5 is a view showing a variable node.

[FIG. 6] FIG. 6 is a view showing a check node.

[FIG. 7] FIG. 7 is a view showing an example of a configuration of an embodiment of a transmission system to which the present invention is applied.

[FIG. 8] FIG. 8 is a block diagram showing an example of a configuration of a transmission apparatus 11.

[FIG. 9] FIG. 9 is a view illustrating a parity check matrix.

[FIG. 10] FIG. 10 is a view illustrating a parity matrix.

[FIG. 11] FIG. 11 is a view illustrating a parity check

matrix of an LDPC code and column weights prescribed in the DVB-S.2 standard.

[FIG. 12] FIG. 12 is a view illustrating a signal point arrangement of 16QAM.

[FIG. 13] FIG. 13 is a view illustrating a signal point arrangement of 64QAM.

[FIG. 14] FIG. 14 is a view illustrating a signal point arrangement of 64QAM.

[FIG. 15] FIG. 15 is a view illustrating a signal point arrangement of 64QAM.

[FIG. 16] FIG. 16 is a view illustrating processing of a demultiplexer 25.

[FIG. 17] FIG. 17 is a view illustrating processing of the demultiplexer 25.

[FIG. 18] FIG. 18 is a view showing a Tanner graph regarding decoding of an LDPC code.

[FIG. 19] FIG. 19 is a view showing a parity matrix H_T having a staircase structure and a Tanner graph corresponding to the parity matrix H_T .

[FIG. 20] FIG. 20 is a view showing the parity matrix H_T of a parity check matrix H corresponding to the LDPC code after parity interleaving.

[FIG. 21] FIG. 21 is a view illustrating a conversion parity check matrix.

[FIG. 22] FIG. 22 is a view illustrating processing of a column twist interleaver 24.

[FIG. 23] FIG. 23 is a view illustrating column numbers of a memory 31 necessary for the column twist interleaving and addresses of writing starting positions.

[FIG. 24] FIG. 24 is a view illustrating column numbers of the memory 31 necessary for the column twist interleaving and addresses of writing starting positions.

[FIG. 25] FIG. 25 is a flow chart illustrating a transmission process.

[FIG. 26] FIG. 26 is a view showing a model of a communication path adopted in a simulation.

[FIG. 27] FIG. 27 is a view illustrating a relationship between an error rate obtained by the simulation and a Doppler frequency f_d of a flutter.

[FIG. 28] FIG. 28 is a view illustrating a relationship between an error rate obtained by the simulation and a Doppler frequency f_d of a flutter.

[FIG. 29] FIG. 29 is a block diagram showing an example of a configuration of a reception apparatus 12.

[FIG. 30] FIG. 30 is a flow chart illustrating a reception process.

[FIG. 31] FIG. 31 is a view illustrating an example of a parity check matrix of an LDPC code.

[FIG. 32] FIG. 32 is a view illustrating a matrix (conversion parity check matrix) obtained by applying row replacement and column replacement to a parity check matrix.

[FIG. 33] FIG. 33 is a view illustrating a conversion parity check matrix divided into a unit of 5×5 bits.

[FIG. 34] FIG. 34 is a block diagram showing an example of a configuration of a decoding apparatus in which node mathematical operation is carried out collectively for P nodes.

[FIG. 35] FIG. 35 is a block diagram showing an example of a configuration of a LDPC decoding section 56.

[FIG. 36] FIG. 36 is a block diagram showing an example of a configuration of an embodiment of a computer to which the present invention is applied.

[FIG. 37] FIG. 37 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $2/3$ and a code length of 16,200.

[FIG. 38] FIG. 38 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $2/3$ and a code length of 64,800.

[FIG. 39] FIG. 39 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $2/3$ and the code length of 64,800.

[FIG. 40] FIG. 40 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $2/3$ and the code length of 64,800.

[FIG. 41] FIG. 41 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $3/4$ and a code length of 16,200.

[FIG. 42] FIG. 42 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $3/4$ and a code length of 64,800.

[FIG. 43] FIG. 43 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $3/4$ and the code length of 64,800.

[FIG. 44] FIG. 44 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $3/4$ and the code length of 64,800.

[FIG. 45] FIG. 45 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $3/4$ and the code length of 64,800.

[FIG. 46] FIG. 46 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $4/5$ and a code length of 16,200.

[FIG. 47] FIG. 47 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $4/5$ and a code length of 64,800.

[FIG. 48] FIG. 48 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $4/5$ and the code length of 64,800.

[FIG. 49] FIG. 49 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $4/5$ and the code length of 64,800.

[FIG. 50] FIG. 50 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $4/5$ and the code length of 64,800.

[FIG. 51] FIG. 51 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $5/6$ and a code length of 16,200.

[FIG. 52] FIG. 52 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $5/6$ and a code length of 64,800.

[FIG. 53] FIG. 53 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $5/6$ and the code length of 64,800.

[FIG. 54] FIG. 54 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $5/6$ and the code length of 64,800.

[FIG. 55] FIG. 55 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $5/6$ and the code length of 64,800.

[FIG. 56] FIG. 56 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $8/9$ and a code length of 16,200.

[FIG. 57] FIG. 57 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $8/9$ and the code length of 64,800.

[FIG. 58] FIG. 58 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $8/9$ and the code length of 64,800.

[FIG. 59] FIG. 59 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $8/9$ and the code length of 64,800.

[FIG. 60] FIG. 60 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $8/9$ and the code length of 64,800.

[FIG. 61] FIG. 61 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $9/10$ and a code length of 64,800.

[FIG. 62] FIG. 62 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $9/10$ and the code length of 64,800.

[FIG. 63] FIG. 63 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $9/10$ and the code length of 64,800.

[FIG. 64] FIG. 64 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $9/10$ and the code length of 64,800.

[FIG. 65] FIG. 65 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $1/4$ and a code length of 64,800.

[FIG. 66] FIG. 66 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $1/4$ and the code length of 64,800.

[FIG. 67] FIG. 67 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $1/3$ and a code length of 64,800.

[FIG. 68] FIG. 68 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $1/3$ and the code length of 64,800.

[FIG. 69] FIG. 69 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $2/5$ and a code length of 64,800.

[FIG. 70] FIG. 70 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $2/5$ and the code length of 64,800.

[FIG. 71] FIG. 71 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $1/2$ and a code length of 64,800.

[FIG. 72] FIG. 72 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $1/2$ and the code length of 64,800.

[FIG. 73] FIG. 73 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $1/2$ and the code length of 64,800.

[FIG. 74] FIG. 74 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $3/5$ and a code length of 64,800.

[FIG. 75] FIG. 75 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $3/5$ and the code length of 64,800.

[FIG. 76] FIG. 76 is a view illustrating the example of the parity check matrix initial value table of the encoding rate of $3/5$ and the code length of 64,800.

[FIG. 77] FIG. 77 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $1/4$ and a code length of 16,200.

[FIG. 78] FIG. 78 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $1/3$ and a code length of 16,200.

[FIG. 79] FIG. 79 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $2/5$ and a code length of 16,200.

[FIG. 80] FIG. 80 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $1/2$ and a code length of 16,200.

[FIG. 81] FIG. 81 is a view illustrating an example of a parity check matrix initial value table of an encoding rate of $3/5$ and a code length of 16,200.

[FIG. 82] FIG. 82 is a view illustrating another example of the parity check matrix initial value table of the encoding rate of $3/5$ and the code length of 16,200.

[FIG. 83] FIG. 83 is a view illustrating a method of determining a parity check matrix H from a parity check matrix initial table.

[FIG. 84] FIG. 84 is a view illustrating an example of replacement of code bits.

[FIG. 85] FIG. 85 is a view illustrating another example of replacement of code bits.

[FIG. 86] FIG. 86 is a view illustrating a further example of replacement of code bits.

[FIG. 87] FIG. 87 is a view illustrating a still further example of replacement of code bits.

[FIG. 88] FIG. 88 is a view illustrating a simulation result of the BER.

[FIG. 89] FIG. 89 is a view illustrating another simulation result of the BER.

[FIG. 90] FIG. 90 is a view illustrating a further simulation result of the BER.

[FIG. 91] FIG. 91 is a view illustrating a still simulation result of the BER.

[FIG. 92] FIG. 92 is a view illustrating an example of replacement of code bits.

[FIG. 93] FIG. 93 is a view illustrating another example of replacement of code bits.

[FIG. 94] FIG. 94 is a view illustrating a further example of replacement of code bits.

[FIG. 95] FIG. 95 is a view illustrating a still further example of replacement of code bits.

[FIG. 96] FIG. 96 is a view illustrating a yet further example of replacement of code bits.

[FIG. 97] FIG. 97 is a view illustrating a yet further example of replacement of code bits.

[FIG. 98] FIG. 98 is a view illustrating a yet further example of replacement of code bits.

[FIG. 99] FIG. 99 is a view illustrating a yet further example of replacement of code bits.

[FIG. 100] FIG. 100 is a view illustrating a yet further example of replacement of code bits.

[FIG. 101] FIG. 101 is a view illustrating a yet further example of replacement of code bits.

[FIG. 102] FIG. 102 is a view illustrating a yet further example of replacement of code bits.

[FIG. 103] FIG. 103 is a view illustrating a yet further example of replacement of code bits.

[FIG. 104] FIG. 104 is a view illustrating processing of a multiplexer 54 which composes a deinterleaver 53.

[FIG. 105] FIG. 105 is a view illustrating processing of a column twist deinterleaver 55.

[FIG. 106] FIG. 106 is a block diagram showing another example of a configuration of the reception apparatus 12.

[FIG. 107] FIG. 107 is a block diagram showing a first example of a configuration of a reception system which can be applied to the reception apparatus 12.

[FIG. 108] FIG. 108 is a block diagram showing a second example of the configuration of the reception system which can be applied to the reception apparatus 12.

[FIG. 109] FIG. 109 is a block diagram showing a third example of the configuration of the reception system which can be applied to the reception apparatus 12.

Explanation of Reference Symbols

[0060]

11 Transmission apparatus, 12 Reception apparatus,
21 LDPC encoding section, 22 Bit interleaver, 23 Parity
interleaver, 24 Column twist interleaver, 25

Demultiplexer, 26 Mapping section, 27 Orthogonal
 modulation section, 31 Memory, 32 Replacement section, 51
 Orthogonal demodulation section, 52 Demapping section, 53
 Deinterleaver, 54 Multiplexer, 55 Column twist
 interleaver, 56 LDPC decoding section, 300 Edge data
 storage memory, 301 Selector, 302 Check node calculation
 section, 303 Cyclic shift circuit, 304 Edge data storage
 memory, 305 Selector, 306 Reception data memory, 307
 Variable node calculation section, 308 Cyclic shift
 circuit, 309 Decoded word calculation section, 310
 Reception data re-arrangement section, 311 Decoded data
 re-arrangement section, 401 Bus, 402 CPU, 403 ROM, 404
 RAM, 405 Hard disk, 406 Outputting section, 407 Inputting
 section, 408 Communication section, 409 Drive, 410
 Input/output interface, 411 Removable recording medium,
 1001 Reverse replacement section, 1002 Memory, 1011
 Parity deinterleaver, 1021 LDPC decoding section, 1101
 Acquisition section, 1101 Transmission line decoding
 processing section, 1103 Information source decoding
 processing section, 1111 Outputting section, 1121
 Recording section

Best Mode for Carrying Out the Invention

[0061]

FIG. 7 shows an example of a configuration of an embodiment of a transmission system to which the present invention is applied (the term system signifies a logical aggregate of a plurality of apparatus irrespective of whether or not the individual component apparatus are included in the same housing).

[0062]

Referring to FIG. 7, the transmission system includes a transmission apparatus 11 and a reception apparatus 12.

[0063]

The transmission apparatus 11 is, for example, an apparatus which carries out transmission of a television broadcasting program and transmits object data which are an object of transmission such as image data, sound data and so forth as a television broadcasting program, for example, through a satellite channel or ground waves.

[0064]

The reception apparatus 12 is, for example, a tuner or a television receiver for receiving a television broadcasting program, and receives object data transmitted thereto from the transmission apparatus 11.

[0065]

FIG. 8 shows an example of a configuration of the

transmission apparatus 11 of FIG. 7.

[0066]

Referring to FIG. 8, the transmission apparatus 11 includes an LDPC encoding section 21, a bit interleaver 22, a mapping section 26 and an orthogonal modulation section 27.

[0067]

To the LDPC encoding section 21, object data are supplied.

[0068]

The LDPC encoding section 21 carries out LDPC encoding of the object data supplied thereto in accordance with a parity check matrix in which a parity matrix which is a portion corresponding to parity bits of an LDPC code has a staircase structure and outputs an LDPC code wherein the object data are information bits.

[0069]

In particular, the LDPC encoding section 21 carries out LDPC encoding of encoding the object data into an LDPC code prescribed, for example, in the DVB-S.2 standards and outputs an LDPC code obtained as a result of the LDPC encoding.

[0070]

Here, the LDPC code prescribed in the DVB-S.2

standard is an IRA (Irregular Repeat Accumulate) code, and the parity matrix in the parity check matrix of the LDPC code has a staircase structure. The parity matrix and the staircase structure are hereinafter described. Further, the IRA code is described, for example, in "Irregular Repeat-Accumulate Codes," H. Jin., A. Khandekar, and R. J. McEliece, in Proceedings of 2nd International Symposium on Turbo codes and Related Topics, pp.1-8, Sept. 2000.

[0071]

The LDPC code outputted from the LDPC encoding section 21 is supplied to the bit interleaver 22.

[0072]

The bit interleaver 22 is a data processing apparatus for interleaving data and includes a parity interleaver 23, a column twist interleaver 24 and a demultiplexer (DEMUX) 25.

[0073]

The parity interleaver 23 carries out parity interleave of interleaving parity bits of the LDPC code from the LDPC encoding section 21 to positions of other parity bits and supplies the LDPC code after the parity interleave to the column twist interleaver 24.

[0074]

The column twist interleaver 24 carries out column twist interleave for the LDPC code from the parity interleaver 23 and supplies the LDPC code after the column twist interleave to the demultiplexer 25.

[0075]

In particular, the LDPC code is transmitted after two or more code bits thereof are mapped to one symbol of orthogonal modulation by the mapping section 26 hereinafter described.

[0076]

The column twist interleaver 24 carries out, for example, such column twist interleave as hereinafter described as a re-arranging process of re-arranging code bits of the LDPC code from the parity interleaver 23 such that a plurality of code bits of the LDPC code corresponding to the value 1 included in one arbitrary row of the parity check matrix used in the LDPC encoding section 21 are not mapped to one symbol.

[0077]

The demultiplexer 25 carries out a replacing process of replacing the positions of two or more code bits of the LDPC code from the column twist interleaver 24 which are mapped to a symbol to obtain an LDPC code whose tolerance to AWGN is reinforced and supplies the

LDPC code to the mapping section 26.

[0078]

The mapping section 26 maps two or more code bits of the LDPC code from the demultiplexer 25 to signal points determined by a modulation method of orthogonal modulation (multi-value modulation) carried out by the orthogonal modulation section 27.

[0079]

In particular, the mapping section 26 symbolizes the LDPC code from the demultiplexer 25 into a symbol (symbol value), represented by a signal point determined by the modulation system, on an IQ plane (IQ constellation) defined by an I axis representative of an I component which is in phase with a carrier and a Q axis representative of a Q component which is orthogonal to the carrier wave.

[0080]

Here, as the modulation method of orthogonal modulation carried out by the orthogonal modulation section 27, modulation methods including, for example, a modulation method defined in the DVB-T standards, that is, for example, QPSK (Quadrature Phase Shift Keying), 16QAM (Quadrature Amplitude Modulation), 64QAM, 256QAM, 1024QAM, 4086QAM and so forth are available. What modulation

method should be used for orthogonal modulation to be carried out by the orthogonal modulation section 27 is set in advance, for example, in accordance with an operation of the transmission apparatus 11 by an operator. It is to be noted that the orthogonal modulation section 27 can carry out some other orthogonal modulation such as, for example, 4PAM (Pulse Amplitude Modulation).

[0081]

The symbol obtained by the mapping section 26 is supplied to the orthogonal modulation section 27.

[0082]

The orthogonal modulation section 27 carries out orthogonal modulation of a carrier in accordance with the symbol from the mapping section 26 and transmits a modulation signal obtained by the orthogonal modulation.

[0083]

Now, FIG. 9 illustrates a parity check matrix H used in LDPC encoding by the LDPC encoding section 21 of FIG. 8.

[0084]

The parity check matrix H has an LDGM (Low-Density Generation Matrix) structure and can be represented by an expression $H = [H_A | H_T]$ from an information matrix H_A of a portion corresponding to information bits and a parity

matrix H_T corresponding to parity bits from among code bits of the LDPC code (matrix in which elements of the information matrix H_A are elements on the left side and elements of the parity matrix H_T are elements on the right side).

[0085]

Here, the bit number of information bits and the bit number of parity bits from among code bits of one LDPC code (one codeword) are referred to as information length K and parity length M , and the bit number of code bits of one LDPC code is referred to as code length N ($= K + M$).

[0086]

The information length K and the parity length M regarding an LDPC code of a certain code length N depend upon the encoding rate. Meanwhile, the parity check matrix H is a matrix whose rows $\times$ columns are $M \times N$. Then, the information matrix H_A is an $M \times K$ matrix and the parity matrix H_T is an $M \times M$ matrix.

[0087]

FIG. 10 illustrates the parity matrix H_T of the parity check matrix H of an LDPC code prescribed in the DVB-S.2 standard.

[0088]

The parity matrix H_T of the parity check matrix H of the LDPC code prescribed in the DVB-S.2 standard has a staircase structure wherein elements of the value 1 are arranged like a staircase as seen in FIG. 10. The row weight of the parity matrix H_T is 1 with regard to the first row but is 2 with regard to all of the remaining rows. Meanwhile, the column weight is 1 with regard to the last column but is 2 with regard to all of the remaining columns.

[0089]

As described above, the LDPC code of the parity check matrix H wherein the parity matrix H_T has a staircase structure can be produced readily using the parity check matrix H .

[0090]

In particular, an LDPC code (one codeword) is represented by a row vector c and a column vector obtained by transposing the row vector is represented by c^T . Further, a portion of information bits from within the row vector c which is an LDPC code is represented by an row vector A and a portion of parity bits is represented by a row vector T .

[0091]

Here, in this instance, the row vector c can be

presented by an expression $c = [A|T]$ from the row vector A as information bits and the row vector T as parity bits (row vector wherein the elements of the row vector A are elements on the left side and the elements of the row vector T are elements on the right side).

[0092]

It is necessary for the parity check matrix H and the row vector $c = [A|T]$ as the LDPC code to satisfy an expression $Hc^T = 0$, and where the parity matrix H_T of the parity check matrix $H = [H_A|H_T]$ has such a staircase structure as shown in FIG. 10, the row vector T as parity bits which forms the row vector $c = [A|T]$ which satisfies the expression $Hc^T = 0$ can be determined sequentially by successively setting the elements in the rows beginning with the elements in the first row of the column vector Hc^T in the expression $Hc^T = 0$ to zero.

[0093]

FIG. 11 illustrates the parity check matrix H of an LDPC code and column weights defined in the DVB-S.2 standard.

[0094]

In particular, A of FIG. 11 illustrates the parity check matrix H of an LDPC code defined in the DVB-S.2 standard.

[0095]

With regard to KX columns from the first column of the parity check matrix H, the column weight is X; with regard to succeeding K3 columns, the column weight is 3; with regard to succeeding M-1 rows, the column weight is 2; and with regard to the last one column, the column weight is 1.

[0096]

Here, $KX + K3 + M - 1 + 1$ is equal to the code length N.

[0097]

In the DVB-S.2 standard, the column numbers KX, K3 and M (parity length) as well as the column weight X are prescribed in such a manner as seen in B of FIG. 11.

[0098]

In particular, B of FIG. 11 illustrates the column numbers KX, K3 and M as well as the column weight X regarding different encoding rates of LDPC codes prescribed in the DVB-S.2 standard.

[0099]

In the DVB-S.2 standard, LDPC codes of the code lengths N of 64,800 bits and 16,200 bits are prescribed.

[0100]

And as seen in B of FIG. 11, for the LDPC code

whose code length N is 64,800 bits, 11 encoding rates (nominal rates) $1/4$, $1/3$, $2/5$, $1/2$, $3/5$, $2/3$, $3/4$, $4/5$, $5/6$, $8/9$ and $9/10$ are prescribed, and for the LDPC code whose code length N is 16,200 bits, 10 encoding rates $1/4$, $1/3$, $2/5$, $1/2$, $3/5$, $2/3$, $3/4$, $4/5$, $5/6$ and $8/9$ are prescribed.

[0101]

Regarding LDPC codes, it is known that code bits corresponding to a column of the parity check matrix H which has a higher column weight exhibits a lower error rate.

[0102]

The parity check matrix H prescribed in the DVB-S.2 standard and illustrated in FIG. 11 has a tendency that a column nearer to the head side (left side) has a higher column weight. Accordingly, the LDPC code corresponding to the parity check matrix H has a tendency that a code bit nearer to the head is higher in tolerance to an error (has a higher tolerance to an error) and a code bit nearer to the tail is lower in tolerance to an error.

[0103]

FIG. 12 illustrates an arrangement of (signal points corresponding to) 16 symbols on the IQ plane where 16QAM is carried out by the orthogonal modulation section

27 of FIG. 8.

[0104]

In particular, A of FIG. 12 illustrates symbols of 16QAM.

[0105]

In 16QAM, one symbol represents 4 bits, and 16 ($= 2^4$) symbols exist. Then, the 16 symbols are disposed such that they form a square shape of 4×4 symbols in the I direction $\times$ Q direction centered at the origin of the IQ plane.

[0106]

Now, if 4 bits represented by one symbol of 16QAM are represented by y_0 , y_1 , y_2 and y_3 in order from the most significant bit, then where the modulation system is the 16QAM, 4 code bits of the LDPC code are mapped to a symbol of the 4 bits y_0 to y_3 which coincide with the 4 bits by the mapping section 26 (FIG. 8).

[0107]

B of FIG. 12 indicates bit boundaries regarding the 4 bits y_0 to y_3 represented by the symbol of the 16QAM.

[0108]

Here, a bit boundary regarding a bit y_i (in FIG. 12, $i = 0, 1, 2, 3$) signifies a boundary between a symbol whose bit y_i is 0 and another symbol whose bit y_i is 1.

[0109]

As seen in B of FIG. 12, as regards the most significant bit y_0 from among the 4 bits y_0 to y_3 represented by the symbol of 16QAM, only one location of the Q axis on the IQ plane makes a bit boundary, and as regards the second bit y_1 (second from the most significant bit), only one location of the I axis on the IQ plane makes a bit boundary.

[0110]

Further, as regards the third bit y_3 , each of two locations between the first and second columns and between the third and fourth columns from the left of the 4×4 symbols makes a boundary.

[0111]

Furthermore, as regards the fourth bit y_3 , each of two locations between the first and second rows and between the third and fourth rows of the 4×4 symbols makes a boundary.

[0112]

The bit y_1 represented by a symbol is less likely to become erroneous as the number of symbols spaced away from a bit boundary increases but is more likely to become erroneous as the number of symbols positioned nearer to a bit boundary increases.

[0113]

If a bit which is less likely to become erroneous (is tolerant to an error) is referred to as "strong bit" but a bit which is more likely to become erroneous (is less tolerant to an error) is referred to as "weak bit," then as regards the 4 bits y_0 to y_3 represented by symbols of 16QAM, the most significant bit y_0 and the second bit y_1 are strong bits and the third bit y_2 and the fourth bit y_3 are weak bits.

[0114]

FIGS. 13 to 15 illustrate arrangements of (signal points corresponding to) 64 symbols on the IQ plane where 64QAM is carried out by the orthogonal modulation section 27 of FIG. 8.

[0115]

In 64QAM, one symbol represents 6 bits, and 64 ($= 2^6$) symbols exist. Then, the 64 symbols are arranged such that they make a square of 8×8 symbols in the I direction $\times$ Q direction centered at the origin of the IQ plane.

[0116]

Now, if it is assumed that the 6 bits represented by one symbol of 64QAM are represented as y_0 , y_1 , y_2 , y_3 , y_4 and y_5 in order from the most significant bit, then

where the modulation system is 64QAM, 6 code bits of the LDPC code are mapped to a symbol of 6 bits y_0 to y_5 coincident to the 6 bits.

[0117]

Here, FIG. 13 indicates bit boundaries regarding the most significant bit y_0 and the second bit y_1 from among the 6 bits y_0 to y_5 represented by symbols of 64QAM; FIG. 14 indicates bit boundaries regarding the third bit y_2 and the fourth bit y_3 ; and FIG. 15 indicates bit boundaries regarding the fifth bit y_4 and the sixth bit y_5 .

[0118]

As seen in FIG. 13, the number of bit boundaries with regard to each of the most significant bit y_0 and the second bit y_1 is one. Meanwhile, as seen in FIG. 14, the number of bit boundaries with regard to each of the third bit y_2 and the fourth bit y_3 is two, and as seen in FIG. 15, the number of bit boundaries with regard to each of the fifth bit y_4 and the sixth bit y_5 is four.

[0119]

Accordingly, among the 6 bits y_0 to y_5 represented by symbols of 64QAM, the most significant bit y_0 and the second bit y_1 are the strongest bits, and the third bit y_2 and the fourth bit y_3 are the second strongest bits. Then, the fifth bit y_4 and the sixth bit y_5 are the weakest bits.

[0120]

From FIG. 12 and further from FIGS. 13 to 15, it can be seen that, as regards bits of symbols of orthogonal modulation, there is a tendency that a high-order bit is a strong bit and a low-order bit is a weak bit.

[0121]

Here, as described hereinabove with reference to FIG. 11, an LDPC code outputted from the LDPC encoding section 21 (FIG. 8) includes code bits which are tolerant to errors and code bits which are less tolerant to errors.

[0122]

Meanwhile, as described hereinabove with reference to FIGS. 12 to 15, bits of symbols of orthogonal modulation carried out by the orthogonal modulation section 27 include strong bits and weak bits.

[0123]

Accordingly, if mapping of allocating a code bit of the LDPC code which is low in tolerance to an error to a weak bit of a symbol of orthogonal modulation is carried, then the tolerance to an error drops as a whole.

[0124]

Therefore, an interleaver has been proposed which interleaves code bits of an LDPC code such that mapping

of allocating code bits of the LDPC code which are low in tolerance to an error to strong bits of a symbol of orthogonal modulation is carried out.

[0125]

The demultiplexer 25 of FIG. 8 carries out processing of the interleaver.

[0126]

FIG. 16 is a view illustrating processing of the demultiplexer 25 of FIG. 8.

[0127]

In particular, A of FIG. 16 shows an example of a functional configuration of the demultiplexer 25.

[0128]

The demultiplexer 25 includes a memory 31 and a replacement section 32.

[0129]

To the memory 31, an LDPC code is supplied.

[0130]

The memory 31 has a storage capacity for storing mb bits in the (horizontal) direction of a row and storing $N/(mb)$ bits in the (vertical) direction of a column. The memory 31 writes code bits of the LDPC code supplied thereto into the column direction and reads out the code bits in the row direction and then supplies the read out

code bits to the replacement section 32.

[0131]

Here, m represents the bit number of code bits of an LDPC code to be mapped to one symbol, and b is a predetermined positive integer and is a multiple to be used for multiplying m by the integer. Meanwhile, N (= information length K + parity length M) represents the code length of the LDPC code as described hereinabove.

[0132]

A of FIG. 16 shows an example of a configuration of the demultiplexer 25 where the modulation system is 64QAM, and accordingly, the bit number m of code bits of the LDPC code to be mapped to one symbol is 6 bits.

[0133]

Further, in A of FIG. 16, the multiple b is 1, and accordingly, the memory 31 has a storage capacity of $N/(6 \times 1) \times (6 \times 1)$ bits in the column direction $\times$ row direction.

[0134]

Here, a storage region of the memory 31 which extends in the column direction and includes one bit in the row direction is hereinafter referred to suitably as column. In A of FIG. 16, the memory 31 includes six (= 6×1) columns.

[0135]

The demultiplexer 25 carries out writing of the code bits of the LDPC code in a downward direction from above of a column which forms the memory 31 (in a column direction) beginning with a left side column toward a right side column.

[0136]

Then, if the writing of the code bits ends with the lowermost bit in the rightmost column, then the code bits are read out and supplied to the replacement section 32 in a unit of 6 bits (mb bits) in the row direction beginning with the first row of all of the columns which form the memory 31.

[0137]

The replacement section 32 carries out a replacement process of replacing the position of code bits of 6 bits from the memory 31 and outputs the 6 bits obtained by the replacement as 6 bits y_0, y_1, y_2, y_3, y_4 and y_5 representative of one symbol of 64QAM.

[0138]

In particular, if the 6 code bits read out in the row direction from the memory 31 are represented as b_0, b_1, b_2, b_3, b_4 and b_5 in order from the most significant bit, then from a relationship of the column weight described

hereinabove with reference to FIG. 11, the code bit positioned in the direction of the bit b_0 is a code bit high in tolerance to an error while the code bit in the direction of the bit b_5 is a code bit low in tolerance to an error.

[0139]

The replacement section 32 carries out a replacement process of replacing the position of the 6 code bits b_0 to b_5 from the memory 31 such that a code bit which is low in tolerance to an error from among the 6 code bits b_0 to b_5 from the memory 31 may be allocated to a bit which is high in tolerance from among the 6 bits y_0 to y_5 representative of one symbol of 64QAM.

[0140]

Here, for a replacement method for replacing the 6 code bits b_0 to b_5 from the memory 31 so as to be allocated to the 6 bits y_0 to y_5 representative of one symbol of 64QAM, various systems have been proposed.

[0141]

B of FIG. 16 illustrates a first replacement method; C of FIG. 16 illustrates a second replacement method; and D of FIG. 16 illustrates a third replacement method.

[0142]

In B of FIG. 16 to D of FIG. 16 (similarly also in FIG. 17 hereinafter described), a line segment interconnecting the bits b_i and y_j signifies that the code bit b_i is allocated to the bit y_j of the symbol (is replaced into the position of the bit y_j).

[0143]

As the first replacement method of B of FIG. 16, it is proposed to adopt one of three kinds, and as the second replacement method of C of FIG. 16, it is proposed to adopt one of two kinds.

[0144]

As the third replacement method of D of FIG. 16, it is proposed to select and use six kinds in order.

[0145]

FIG. 17 illustrates an example of a configuration of the demultiplexer 25 in a case wherein the modulation method is 64QAM (accordingly, the bit number m of code bits of an LDPC code mapped to one symbol is 6 similarly as in FIG. 16) and the multiple b is 2, and a fourth replacement method.

[0146]

Where the multiple b is 2, the memory 31 has a storage capacity of $N/(6 \times 2) \times (6 \times 2)$ bits in the column direction $\times$ row direction and includes 12 ($= 6 \times$

2) columns.

[0147]

A of FIG. 17 illustrates a writing order of an LDPC code into the memory 31.

[0148]

The demultiplexer 25 carries out writing of code bits of an LDPC code in a downward direction from above of a column which forms the memory 31 (in the column direction) beginning with a left side column toward a right side column as described hereinabove with reference to FIG. 16.

[0149]

Then, if the writing of code bits ends with the lowermost bit in the rightmost column, then the code bits are read out and supplied to the replacement section 32 in a unit of 12 bits (mb bits) in the row direction beginning with the first row of all of the columns which form the memory 31.

[0150]

The replacement section 32 carries out a replacement process of replacing the position of 12 code bits from the memory 31 in accordance with the fourth replacement method and outputs the 12 bits obtained by the replacement as 12 bits representative of two symbols

(b symbols) of 64QAM, in particular, as 6 bits y_0, y_1, y_2, y_3, y_4 and y_5 representative of one symbol of 64QAM and 6 bits y_0, y_1, y_2, y_3, y_4 and y_5 representative of a next one symbol.

[0151]

Here, B of FIG. 17 illustrates the fourth replacement method of the replacement process by the replacement section 32 of A of FIG. 17.

[0152]

It is to be noted that which replacement method is optimum, that is, which replacement method provides the best error rate in an AWGN communication path, differs depends upon the encoding rate of LDPC coded and so forth.

[0153]

Now, parity interleave by the parity interleaver 23 of FIG. 8 is described with reference to FIGS. 18 to 20.

[0154]

FIG. 18 shows (part of) a Tanner graph of the parity check matrix of the LDPC code.

[0155]

If a plurality of (code bits corresponding to) variable nodes connecting to a check node such as two variable nodes suffer from an error such as erasure at the same time as shown in FIG. 18, then the check node

returns a message of an equal probability representing that the probability that the value may be 0 and the probability that the value may be 1 are equal to each other to all variable nodes connecting to the check node. Therefore, if a plurality of variable nodes connecting to the same check node are placed into an erasure state or the like at the same time, then the performance in decoding is deteriorated.

[0156]

Incidentally, an LDPC code outputted from the LDPC encoding section 21 of FIG. 8 and prescribed in the DVB-S.2 standard is an IRA code, and the parity matrix H_T of the parity check matrix H has a staircase structure as shown in FIG. 10.

[0157]

FIG. 19 illustrates a parity matrix H_T having a staircase structure and a Tanner graph corresponding to the parity matrix H_T .

[0158]

In particular, A of FIG. 19 illustrates a parity matrix H_T having a staircase structure and B of FIG. 19 shows a Tanner graph corresponding to the parity matrix H_T of A of FIG. 19.

[0159]

Where the parity matrix H_T has a staircase structure, in the Tanner graph of the parity matrix H_T , variable nodes of the LDPC code which correspond to a column of an element of the parity matrix H_T having the value of 1 and whose message is determined using adjacent code bits (parity bits) are connected to the same check node.

[0160]

Accordingly, if the adjacent parity bits described above are placed into an error state by burst errors, erasure or the like, then since a check node connecting to a plurality of variable nodes corresponding to the plural parity bits which have become an error (variable nodes whose message are to be determined using parity bits) returns a message of an equal probability representing that the probability that the value may be 0 and the probability that the value is 1 may be equal to each other to the variable nodes connecting to the check node, the performance of the decoding deteriorates. Then, where the burst length (number of bits which are made an error by a burst) is great, the performance of the decoding further deteriorates.

[0161]

Therefore, in order to prevent the deterioration in

performance of decoding described above, the parity interleaver 23 (FIG. 8) carries out interleave of interleaving parity bits of the LDPC code from the LDPC encoding section 21 to positions of other parity bits.

[0162]

FIG. 20 illustrates a parity matrix H_T of a parity check matrix H corresponding to the LDPC code after the parity interleave carried out by the parity interleaver 23 of FIG. 8.

[0163]

Here, the information matrix H_A of the parity check matrix H corresponding to the LDPC code prescribed in the DVB-S.2 standard and outputted from the LDPC encoding section 21 has a cyclic structure.

[0164]

The cyclic structure signifies a structure wherein a certain column coincides with another column in a cyclically operated state and includes, for example, a structure wherein, for every P columns, the positions of the value 1 in the rows of the P columns coincide with positions to which the first one of the P columns is cyclically shifted in the column direction by a value which increases in proportion to a value q obtained by dividing the parity length M . In the following, the

number of P columns in a cyclic structure is hereinafter referred to suitably as a unit column number of the cyclic structure.

[0165]

As an LDPC code prescribed in the DVB-S.2 standard and outputted from the LDPC encoding section 21, two LDPC codes are available including those whose code length N is 64,800 bits and 16,200 bits as described hereinabove with reference to FIG. 11.

[0166]

Now, if attention is paid to the LDPC code whose code length N is 64,800 bits from the two different LDPC codes whose code length N is 64,800 bits and 16,200 bits, then eleven different encoding rates are available as the encoding rate of the LDPC code whose code length N is 64,800 bits as described hereinabove with reference to FIG. 11.

[0167]

With regard to LDPC codes whose code length N is 64,800 bits and which have the eleven different encoding rates, it is prescribed in the DVB-S.2 standard that the column number P of the cyclic structure is prescribed to 360 which is one of divisors of the parity length M except 1 and M.

[0168]

Further, with regard to LDPC codes whose code length N is 64,800 bits and which have the eleven different encoding rates, the parity length M has a value other than prime numbers and represented by an expression $M = q \times P = q \times 360$ using the value q which is different depending upon the encoding rate. Accordingly, also the value q is one of the divisors of the parity length M except 1 and M similarly to the column number P of the cyclic structure and is obtained by dividing the parity length M by the column number P of the cyclic structure (the product of P and q which are divisors of the parity length M is the parity length M).

[0169]

Where the information length is represented by K and an integer higher than 0 but lower than P is represented by x while an integer higher than 0 but lower than q is represented by y , the parity interleaver 23 interleaves, as parity interleave, the $K+qx+y+1$ th code bit from among parity bits which are $K+1$ th to $K+M$ th ($K + M = N$) bits of the LDPC code from the LDPC encoding section 21 to the position of the $K+Py+x+1$ th code bit.

[0170]

According to such parity interleave, since the

(parity bits corresponding to) variable nodes connecting to the same check node are spaced by a distance corresponding to the column number P of the cyclic structure, here, by 360 bits, where the burst length is smaller than 360 bits, such a situation that a plurality of variable nodes connecting to the same check node are rendered erroneous at the same time can be prevented. As a result, the tolerance to a burst error can be improved.

[0171]

It is to be noted that the LDPC code after the parity interleave by which the $K+qx+y+1$ th code bit is interleaved to the position of the $K+Py+x+1$ th code bit coincides with the LDPC code of a parity check matrix (hereinafter referred to also as conversion parity check matrix) obtained by column replacement of replacing the $K+qx+y+1$ th column of the original parity check matrix H into the $K+Py+x+1$ th column.

[0172]

Further, in the parity matrix of the conversion parity check matrix, a pseudo cyclic structure whose unit is P columns (in FIG. 20, 360 columns) appears as seen in FIG. 20.

[0173]

Here, the pseudo cyclic structure signifies a

structure which has a portion having a cyclic structure except part thereof. In a conversion parity check column obtained by applying column replacement corresponding to parity interleave to the parity check matrix of the LDPC code prescribed in the DVB-S.2 standard, a portion of 360 rows $\times$ 360 columns (shift matrix hereinafter described) at a right corner portion is short of one element of 1 (which has the value of 0). Therefore, the conversion parity check matrix does not have a (complete) cyclic structure but has a pseudo cyclic structure.

[0174]

It is to be noted that the conversion parity check matrix of FIG. 20 is a matrix to which also replacement of rows (row replacement) for configuring the conversion parity check matrix from a configuration matrix hereinafter described is applied to the original parity check matrix H in addition to the column replacement which corresponds to parity interleave.

[0175]

Now, column twist interleave as a re-arrangement process by the column twist interleaver 24 of FIG. 8 is described with reference to FIGS. 21 to 24.

[0176]

In the transmission apparatus 11 of FIG. 8, two or

more of the code bits of the LDPC code are transmitted as one symbol as described hereinabove in order to improve the utilization efficiency of frequencies. In particular, for example, where 2 bits of the code bits are used to form one symbol, for example, QPSK is used as the modulation method, but where 4 bits of the code bits are used to form one symbol, for example, 16QAM is used as the modulation method.

[0177]

Where two or more ones of the code bits are transmitted as one symbol in this manner, if erasure or the like occurs with a certain symbol, the all of the code bits of the symbol become an error (erasure).

[0178]

Accordingly, in order to lower the probability that a plurality of (code bits corresponding to) variable nodes connecting to the same check node may suffer from erasure at the same time to improve the performance in decoding, it is necessary to avoid the variable nodes corresponding to code bits of one symbol from connecting to the same check node.

[0179]

Meanwhile, in the parity check matrix H of an LDPC code prescribed in the DVB-S.2 standard and outputted

from the LDPC encoding section 21, the information matrix H_A has a cyclic structure and the parity matrix H_T has a staircase structure as described hereinabove. Then, in a conversion parity check matrix which is a parity check matrix of the LDPC code after parity interleave, a cyclic structure (accurately, a pseudo cyclic structure as described hereinabove) appears also in the parity matrix as described in FIG. 20.

[0180]

FIG. 21 shows a conversion parity check matrix.

[0181]

In particular, A of FIG. 21 illustrates a conversion parity check matrix of a parity check matrix H which has a code length N of 64,800 bits and an encoding rate (r) of $3/4$.

[0182]

In A of FIG. 21, the position of an element having the value of 1 in the conversion parity check matrix is indicated by a dot ($\cdot$).

[0183]

In B of FIG. 21, a process carried out by the demultiplexer 25 (FIG. 8) for the LDPC code of the conversion parity matrix of A of FIG. 21, that is, the LDPC code after the parity interleave.

[0184]

In B of FIG. 21, the code bits of the LDPC code after the parity interleave are written in the column direction in four columns which form the memory 31 of the demultiplexer 25 using 16QAM as the modulation method.

[0185]

The code bits written in the column direction in the four columns which form the memory 31 are read out in the row direction in a unit of 4 bits which make one symbol.

[0186]

In this instance, the 4 code bits B_0 , B_1 , B_2 and B_3 which make one symbol sometimes make code bits corresponding to 1 and included in one arbitrary row of the parity check matrix after the conversion of A of FIG. 21, and in this instance, variable nodes corresponding to the code bits B_0 , B_1 , B_2 and B_3 are connected to the same check node.

[0187]

Accordingly, where the 4 code bits B_0 , B_1 , B_2 and B_3 of one symbol become code bits corresponding to 1 and included in one arbitrary row, if erasure occurs with the symbol, then the same check node to which the variable nodes corresponding to the code bits B_0 , B_1 , B_2 and B_3 are

connected cannot determine an appropriate message. As a result, the performance in decoding deteriorates.

[0188]

Also with regard to the encoding rates other than the encoding rate of $3/4$, a plurality of code bits corresponding to a plurality of variable nodes connecting to the same check node sometimes make one symbol of 16QAM similarly.

[0189]

Therefore, the column twist interleaver 24 carries out column twist interleave wherein the code bits of the LDPC code after the parity interleave from the parity interleaver 23 are interleaved such that a plurality of code bits corresponding to 1 and included in one arbitrary row of the conversion parity check matrix are not mapped to one symbol.

[0190]

FIG. 22 is a view illustrating the column twist interleave.

[0191]

In particular, FIG. 22 illustrates the memory 31 (FIGS. 16 and 17) of the demultiplexer 25.

[0192]

The memory 31 has a storage capacity for storing mb

bits in the column (vertical) direction and stores $N/(mb)$ bits in the row (horizontal) direction and includes mb columns as described in FIG. 16. Then, the column twist interleaver 24 writes the code bits of the LDPC code in the column direction into the memory 31 and controls the writing starting position when the code bits are read out in the row direction to carry out column twist interleave.

[0193]

In particular, the column twist interleaver 24 suitably changes the writing starting position at which writing of code bits is to be started for each of a plurality of columns so that a plurality of code bits read out in the row direction and used to make one symbol may not become code bits corresponding to 1 and included in one arbitrary row of the conversion parity check matrix (re-arranges the code bits of the LDPC code such that a plurality of code bits corresponding to 1 and included in one arbitrary row of the parity check matrix may not be included in the same symbol).

[0194]

Here, FIG. 22 shows an example of a configuration of the memory 31 where the modulation method is 16QAM and besides the multiple b described hereinabove with reference to FIG. 16 is 1. Accordingly, the bit number m

of code bits of an LDPC code to be mapped to one symbol is 4 bits, and the memory 31 is formed from four (= mb) columns.

[0195]

The column twist interleaver 24 of FIG. 22 (instead of the demultiplexer 25 shown in FIG. 16) carries out writing of the code bits of the LDPC code in a downward direction (column direction) from above into the four columns which form the memory 31 beginning with a left side column towards a right side column.

[0196]

Then, when the writing of code bits ends to the rightmost column, the column twist interleaver 24 reads out the code bits in a unit of 4 bits (mb bits) in the row direction beginning with the first row of all columns which form the memory 31 and outputs the code bits as an LDPC code after the column twist interleave to the replacement section 32 (FIGS. 16 and 17) of the demultiplexer 25.

[0197]

However, if the address of the head (uppermost) position of each column is represented by 0 and the addresses of the positions in the column direction are represented by integers of an ascending order, then the

column twist interleaver 24 of FIG. 22 sets, for the leftmost column, the writing starting position to the position whose address is 0; sets, for the second column (from the left), the writing starting position to the position whose address is 2; sets, for the third column, the writing starting position to the position whose address is 4; and sets, for the fourth column, the writing starting position to the position whose address is 7.

[0198]

It is to be noted that, with regard to the columns for which the writing starting position is any other position than the position whose address is 0, after the code bits are written down to the lowermost position, the writing position returns to the top (position whose address is 0) and writing down to a position immediately preceding to the writing starting position is carried out. Thereafter, writing into the next (right) column is carried out.

[0199]

By carrying out such column twist interleave as described above, such a situation that a plurality of code bits corresponding to a plurality of variable nodes connecting to the same check node are made one symbol of

16QAM (included into the same symbol) with regard to LDPC codes of all encoding rates whose code length N is 64,800 as prescribed in the DVB-S.2 standard can be prevented, and as a result, the performance in decoding in a communication path which provides erasure can be improved.

[0200]

FIG. 23 illustrates the number of columns of the memory 31 necessary for column twist interleave and the address of the writing starting position for each modulation method with regard to LDPC codes of the eleven different encoding rates having the code length N of 64,800 as prescribed in the DVB-S.2 standard.

[0201]

As the replacement method for the replacement process of the demultiplexer 25 (FIG. 8), one of the first to third replacement methods of FIG. 16 is adopted, and besides, where QPSK is adopted as the modulation method, the bit number m of one symbol is 2 bits and the multiple b is 1.

[0202]

In this instance, according to FIG. 23, the memory 31 has two columns for storing $2 \times 1 (= mb)$ bits in the row direction and stores $64,800/(2 \times 1)$ bits in the column direction. Then, the writing starting position for

the first one of the two columns of the memory 31 is set to the position whose address is 0, and the writing starting position for the second column is set to the position whose address is 2.

[0203]

Further, where the fourth replacement method of FIG. 17 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides QPSK is adopted as the modulation method, the bit number m of one symbol is 2 bits and the multiple b is 2.

[0204]

In this instance, according to FIG. 23, the memory 31 has four columns for storing 2×2 bits in the row direction and stores $64,800/(2 \times 2)$ bits in the column direction. Then, the writing starting position for the first column from among the four columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 2; the writing starting position for the third column is set to the position whose address is 4; and the writing starting position for the fourth column is set to the position whose address is 7.

[0205]

Further, where one of the first to third replacement methods of FIG. 16 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 16QAM is adopted as the modulation method, the bit number m of one symbol is 4 bits and the multiple b is 1.

[0206]

In this instance, according to FIG. 23, the memory 31 has four columns for storing 4×1 bits in the row direction and stores $64,800 / (4 \times 1)$ bits in the column direction. Then, the writing starting position for the first column from among the four columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 2; the writing starting position for the third column is set to the position whose address is 4; and the writing starting position for the fourth column is set to the position whose address is 7.

[0207]

Further, where the fourth replacement method of FIG. 17 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 16QAM is adopted as the modulation method, the

bit number m of one symbol is 4 bits and the multiple b is 2.

[0208]

In this instance, according to FIG. 23, the memory 31 has eight columns for storing 4×2 bits in the row direction and stores $64,800/(4 \times 2)$ bits in the column direction. Then, the writing starting position for the first column from among the eight columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 0; the writing starting position for the third column is set to the position whose address is 2; the writing starting position for the fourth column is set to the position whose address is 4; the writing starting position for the fifth column is set to the position whose address is 4; the writing starting position for the sixth column is set to the position whose address is 5; the writing starting position for the seventh column is set to the position whose address is 7; and the writing starting position for the eighth column is set to the position whose address is 7.

[0209]

Further, where one of the first to third replacement methods of FIG. 16 is adopted as the

replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 64QAM is adopted as the modulation method, the bit number m of one symbol is 6 bits and the multiple b is 1.

[0210]

In this instance, according to FIG. 23, the memory 31 has six columns for storing 6×1 bits in the row direction and stores $64,800/(6 \times 1)$ bits in the column direction. Then, the writing starting position for the first column from among the six columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 2; the writing starting position for the third column is set to the position whose address is 5; the writing starting position for the fourth column is set to the position whose address is 9; the writing starting position for the fifth column is set to the position whose address is 10; and the writing starting position for the sixth column is set to the position whose address is 13.

[0211]

Further, where the fourth replacement method of FIG. 17 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and

besides 64QAM is adopted as the modulation method, the bit number m of one symbol is 6 bits and the multiple b is 2.

[0212]

In this instance, according to FIG. 23, the memory 31 has twelve columns for storing 6×2 bits in the row direction and stores $64,800/(6 \times 2)$ bits in the column direction. Then, the writing starting position for the first column from among the twelve columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 0; the writing starting position for the third column is set to the position whose address is 2; the writing starting position for the fourth column is set to the position whose address is 2; the writing starting position for the fifth column is set to the position whose address is 3; the writing starting position for the sixth column is set to the position whose address is 4; the writing starting position for the seventh column is set to the position whose address is 4; the writing starting position for the eighth column is set to the position whose address is 5; the writing starting position for the ninth column is set to the position whose address is 5; the writing starting

position for the tenth column is set to the position whose address is 7; the writing starting position for the eleventh column is set to the position whose address is 8; and the writing starting position for the twelfth column is set to the position whose address is 9.

[0213]

Further, where one of the first to third replacement methods of FIG. 16 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 256QAM is adopted as the modulation method, the bit number m of one symbol is 8 bits and the multiple b is 1.

[0214]

In this instance, according to FIG. 23, the memory 31 has eight columns for storing 8×1 bits in the row direction and stores $64,800/(8 \times 1)$ bits in the column direction. Then, the writing starting position for the first column from among the eight columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 0; the writing starting position for the third column is set to the position whose address is 2; the writing starting position for the fourth column is set to the position whose address is 4;

the writing starting position for the fifth column is set to the position whose address is 4; the writing starting position for the sixth column is set to the position whose address is 5; the writing starting position for the seventh column is set to the position whose address is 7; and the writing starting position for the eighth column is set to the position whose address is 7.

[0215]

Further, where the fourth replacement method of FIG. 17 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 256QAM is adopted as the modulation method, the bit number m of one symbol is 8 bits and the multiple b is 2.

[0216]

In this instance, according to FIG. 23, the memory 31 has 16 columns for storing 8×2 bits in the row direction and stores $64,800 / (8 \times 2)$ bits in the column direction. Then, the writing starting position for the first column from among the 16 columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 2; the writing starting position for the third column is set to the position

whose address is 2; the writing starting position for the fourth column is set to the position whose address is 2; the writing starting position for the fifth column is set to the position whose address is 2; the writing starting position for the sixth column is set to the position whose address is 3; the writing starting position for the seventh column is set to the position whose address is 7; the writing starting position for the eighth column is set to the position whose address is 15; the writing starting position for the ninth column is set to the position whose address is 16; the writing starting position for the tenth column is set to the position whose address is 20; the writing starting position for the eleventh column is set to the position whose address is 22; the writing starting position for the twelfth column is set to the position whose address is 22; the writing starting position for the thirteenth column is set to the position whose address is 27; the writing starting position for the fourteenth column is set to the position whose address is 27; the writing starting position for the fifteenth column is set to the position whose address is 28; and the writing starting position for the sixteenth column is set to the position whose address is 32.

[0217]

Further, where one of the first to third replacement methods of FIG. 16 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 1024QAM is adopted as the modulation method, the bit number m of one symbol is 10 bits and the multiple b is 1.

[0218]

In this instance, according to FIG. 23, the memory 31 has ten columns for storing 10×1 bits in the row direction and stores $64,800/(10 \times 1)$ bits in the column direction. Then, the writing starting position for the first column from among the ten columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 3; the writing starting position for the third column is set to the position whose address is 6; the writing starting position for the fourth column is set to the position whose address is 8; the writing starting position for the fifth column is set to the position whose address is 11; the writing starting position for the sixth column is set to the position whose address is 13; the writing starting position for the seventh column is set to the position whose address

is 15; the writing starting position for the eighth column is set to the position whose address is 17; the writing starting position for the ninth column is set to the position whose address is 18; and the writing starting position for the tenth column is set to the position whose address is 20.

[0219]

Further, where the fourth replacement method of FIG. 17 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 1024QAM is adopted as the modulation method, the bit number m of one symbol is 10 bits and the multiple b is 2.

[0220]

In this instance, according to FIG. 23, the memory 31 has 20 columns for storing 10×2 bits in the row direction and stores $64,800 / (10 \times 2)$ bits in the column direction. Then, the writing starting position for the first column from among the 20 columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 1; the writing starting position for the third column is set to the position whose address is 3; the writing starting position for the

fourth column is set to the position whose address is 4; the writing starting position for the fifth column is set to the position whose address is 5; the writing starting position for the sixth column is set to the position whose address is 6; the writing starting position for the seventh column is set to the position whose address is 6; the writing starting position for the eighth column is set to the position whose address is 9; the writing starting position for the ninth column is set to the position whose address is 13; the writing starting position for the tenth column is set to the position whose address is 14; the writing starting position for the eleventh column is set to the position whose address is 14; the writing starting position for the twelfth column is set to the position whose address is 16; the writing starting position for the thirteenth column is set to the position whose address is 21; the writing starting position for the fourteenth column is set to the position whose address is 21; the writing starting position for the fifteenth column is set to the position whose address is 23; the writing starting position for the sixteenth column is set to the position whose address is 25; the writing starting position for the seventeenth column is set to the position whose address is 25; the

writing starting position for the eighteenth column is set to the position whose address is 26; the writing starting position for the nineteenth column is set to the position whose address is 28; and the writing starting position for the twentieth column is set to the position whose address is 30.

[0221]

Further, where one of the first to third replacement methods of FIG. 16 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 4096QAM is adopted as the modulation method, the bit number m of one symbol is 12 bits and the multiple b is 1.

[0222]

In this instance, according to FIG. 23, the memory 31 has 12 columns for storing 12×1 bits in the row direction and stores $64,800/(12 \times 1)$ bits in the column direction. Then, the writing starting position for the first column from among the twelve columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 0; the writing starting position for the third column is set to the position whose address is 2; the writing starting position for the

fourth column is set to the position whose address is 2; the writing starting position for the fifth column is set to the position whose address is 3; the writing starting position for the sixth column is set to the position whose address is 4; the writing starting position for the seventh column is set to the position whose address is 4; the writing starting position for the eighth column is set to the position whose address is 5; the writing starting position for the ninth column is set to the position whose address is 5; the writing starting position for the tenth column is set to the position whose address is 7; the writing starting position for the eleventh column is set to the position whose address is 8; and the writing starting position for the twelfth column is set to the position whose address is 9.

[0223]

Further, where the fourth replacement method of FIG. 17 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 4096QAM is adopted as the modulation method, the bit number m of one symbol is 12 bits and the multiple b is 2.

[0224]

In this instance, according to FIG. 23, the memory

31 has 24 columns for storing 12×2 bits in the row direction and stores $64,800/(12 \times 2)$ bits in the column direction. Then, the writing starting position for the first column from among the 24 columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 5; the writing starting position for the third column is set to the position whose address is 8; the writing starting position for the fourth column is set to the position whose address is 8; the writing starting position for the fifth column is set to the position whose address is 8; the writing starting position for the sixth column is set to the position whose address is 8; the writing starting position for the seventh column is set to the position whose address is 10; the writing starting position for the eighth column is set to the position whose address is 10; the writing starting position for the ninth column is set to the position whose address is 10; the writing starting position for the tenth column is set to the position whose address is 12; the writing starting position for the eleventh column is set to the position whose address is 13; the writing starting position for the twelfth column is set to the position whose address is 16; the

writing starting position for the thirteenth column is set to the position whose address is 17; the writing starting position for the fourteenth column is set to the position whose address is 19; the writing starting position for the fifteenth column is set to the position whose address is 21; the writing starting position for the sixteenth column is set to the position whose address is 22; the writing starting position for the seventeenth column is set to the position whose address is 23; the writing starting position for the eighteenth column is set to the position whose address is 26; the writing starting position for the nineteenth column is set to the position whose address is 37; the writing starting position for the twentieth column is set to the position whose address is 39; the writing starting position for the 21st column is set to the position whose address is 40; the writing starting position for the 22nd column is set to the position whose address is 41; the writing starting position for the 23rd column is set to the position whose address is 41; and the writing starting position for the 24th column is set to the position whose address is 41.

[0225]

FIG. 24 indicates the number of columns of the

memory 31 necessary for column twist interleave and the address of the writing starting position for each modulation method with regard to the LDPC codes of the 10 different encoding rates having the code length N of 16,200 as prescribed in the DVB-S.2 standard.

[0226]

As the replacement method for the replacement process of the demultiplexer 25 (FIG. 8), one of the first to third replacement methods of FIG. 16 is adopted, and besides, where QPSK is adopted as the modulation method, the bit number m of one symbol is 2 bits and the multiple b is 1.

[0227]

In this instance, according to FIG. 24, the memory 31 has two columns for storing 2×1 bits in the row direction and stores $16,200/(2 \times 1)$ bits in the column direction. Then, the writing starting position for the first one of the two columns of the memory 31 is set to the position whose address is 0, and the writing starting position for the second column is set to the position whose address is 0.

[0228]

Further, where the fourth replacement method of FIG. 17 is adopted as the replacement method for the

replacement process of the demultiplexer 25 (FIG. 8) and besides QPSK is adopted as the modulation method, the bit number m of one symbol is 2 bits and the multiple b is 2.

[0229]

In this instance, according to FIG. 24, the memory 31 has four columns for storing 2×2 bits in the row direction and stores $16,200/(2 \times 2)$ bits in the column direction. Then, the writing starting position for the first column from among the four columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 2; the writing starting position for the third column is set to the position whose address is 3; and the writing starting position for the fourth column is set to the position whose address is 3.

[0230]

Further, where one of the first to third replacement methods of FIG. 16 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 16QAM is adopted as the modulation method, the bit number m of one symbol is 4 bits and the multiple b is 1.

[0231]

In this instance, according to FIG. 24, the memory 31 has four columns for storing 4×1 bits in the row direction and stores $16,200/(4 \times 1)$ bits in the column direction. Then, the writing starting position for the first column from among the four columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 2; the writing starting position for the third column is set to the position whose address is 3; and the writing starting position for the fourth column is set to the position whose address is 3.

[0232]

Further, where the fourth replacement method of FIG. 17 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 16QAM is adopted as the modulation method, the bit number m of one symbol is 4 bits and the multiple b is 2.

[0233]

In this instance, according to FIG. 24, the memory 31 has eight columns for storing 4×2 bits in the row direction and stores $16,200/(4 \times 2)$ bits in the column direction. Then, the writing starting position for the

first column from among the eight columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 0; the writing starting position for the third column is set to the position whose address is 0; the writing starting position for the fourth column is set to the position whose address is 1; the writing starting position for the fifth column is set to the position whose address is 7; the writing starting position for the sixth column is set to the position whose address is 20; the writing starting position for the seventh column is set to the position whose address is 20; and the writing starting position for the eighth column is set to the position whose address is 21.

[0234]

Further, where one of the first to third replacement methods of FIG. 16 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 64QAM is adopted as the modulation method, the bit number m of one symbol is 6 bits and the multiple b is 1.

[0235]

In this instance, according to FIG. 24, the memory 31 has six columns for storing 6×1 bits in the row

direction and stores $16,200/(6 \times 1)$ bits in the column direction. Then, the writing starting position for the first column from among the six columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 0; the writing starting position for the third column is set to the position whose address is 2; the writing starting position for the fourth column is set to the position whose address is 3; the writing starting position for the fifth column is set to the position whose address is 7; and the writing starting position for the sixth column is set to the position whose address is 7.

[0236]

Further, where the fourth replacement method of FIG. 17 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 64QAM is adopted as the modulation method, the bit number m of one symbol is 6 bits and the multiple b is 2.

[0237]

In this instance, according to FIG. 24, the memory 31 has twelve columns for storing 6×2 bits in the row direction and stores $16,200/(6 \times 2)$ bits in the column

direction. Then, the writing starting position for the first column from among the twelve columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 0; the writing starting position for the third column is set to the position whose address is 0; the writing starting position for the fourth column is set to the position whose address is 2; the writing starting position for the fifth column is set to the position whose address is 2; the writing starting position for the sixth column is set to the position whose address is 2; the writing starting position for the seventh column is set to the position whose address is 3; the writing starting position for the eighth column is set to the position whose address is 3; the writing starting position for the ninth column is set to the position whose address is 3; the writing starting position for the tenth column is set to the position whose address is 6; the writing starting position for the eleventh column is set to the position whose address is 7; and the writing starting position for the twelfth column is set to the position whose address is 7.

[0238]

Further, where one of the first to third

replacement methods of FIG. 16 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 256QAM is adopted as the modulation method, the bit number m of one symbol is 8 bits and the multiple b is 1.

[0239]

In this instance, according to FIG. 24, the memory 31 has eight columns for storing 8×1 bits in the row direction and stores $16,200/(8 \times 1)$ bits in the column direction. Then, the writing starting position for the first column from among the eight columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 0; the writing starting position for the third column is set to the position whose address is 0; the writing starting position for the fourth column is set to the position whose address is 1; the writing starting position for the fifth column is set to the position whose address is 7; the writing starting position for the sixth column is set to the position whose address is 20; the writing starting position for the seventh column is set to the position whose address is 20; and the writing starting position for the eighth column is set to the position whose address is 21.

[0240]

Further, where one of the first to third replacement methods of FIG. 16 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 1024QAM is adopted as the modulation method, the bit number m of one symbol is 10 bits and the multiple b is 1.

[0241]

In this instance, according to FIG. 24, the memory 31 has 10 columns for storing 10×1 bits in the row direction and stores $16,200/(10 \times 1)$ bits in the column direction. Then, the writing starting position for the first column from among the ten columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 1; the writing starting position for the third column is set to the position whose address is 2; the writing starting position for the fourth column is set to the position whose address is 2; the writing starting position for the fifth column is set to the position whose address is 3; the writing starting position for the sixth column is set to the position whose address is 3; the writing starting position for the seventh column is set to the position whose address is 4;

the writing starting position for the eighth column is set to the position whose address is 4; the writing starting position for the ninth column is set to the position whose address is 5; and the writing starting position for the tenth column is set to the position whose address is 7.

[0242]

Further, where the fourth replacement method of FIG. 17 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 1024QAM is adopted as the modulation method, the bit number m of one symbol is 10 bits and the multiple b is 2.

[0243]

In this instance, according to FIG. 24, the memory 31 has 20 columns for storing 10×2 bits in the row direction and stores $16,200/(10 \times 2)$ bits in the column direction. Then, the writing starting position for the first column from among the 20 columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 0; the writing starting position for the third column is set to the position whose address is 0; the writing starting position for the

fourth column is set to the position whose address is 2; the writing starting position for the fifth column is set to the position whose address is 2; the writing starting position for the sixth column is set to the position whose address is 2; the writing starting position for the seventh column is set to the position whose address is 2; the writing starting position for the eighth column is set to the position whose address is 2; the writing starting position for the ninth column is set to the position whose address is 5; the writing starting position for the tenth column is set to the position whose address is 5; the writing starting position for the eleventh column is set to the position whose address is 5; the writing starting position for the twelfth column is set to the position whose address is 5; the writing starting position for the thirteenth column is set to the position whose address is 5; the writing starting position for the fourteenth column is set to the position whose address is 7; the writing starting position for the fifteenth column is set to the position whose address is 7; the writing starting position for the sixteenth column is set to the position whose address is 7; the writing starting position for the seventeenth column is set to the position whose address is 7; the writing starting

position for the eighteenth column is set to the position whose address is 8; the writing starting position for the nineteenth column is set to the position whose address is 8; and the writing starting position for the twentieth column is set to the position whose address is 10.

[0244]

Further, where one of the first to third replacement methods of FIG. 16 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 4096QAM is adopted as the modulation method, the bit number m of one symbol is 12 bits and the multiple b is 1.

[0245]

In this instance, according to FIG. 24, the memory 31 has 12 columns for storing 12×1 bits in the row direction and stores $16,200/(12 \times 1)$ bits in the column direction. Then, the writing starting position for the first column from among the twelve columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 0; the writing starting position for the third column is set to the position whose address is 0; the writing starting position for the fourth column is set to the position whose address is 2;

the writing starting position for the fifth column is set to the position whose address is 2; the writing starting position for the sixth column is set to the position whose address is 2; the writing starting position for the seventh column is set to the position whose address is 3; the writing starting position for the eighth column is set to the position whose address is 3; the writing starting position for the ninth column is set to the position whose address is 3; the writing starting position for the tenth column is set to the position whose address is 6; the writing starting position for the eleventh column is set to the position whose address is 7; and the writing starting position for the twelfth column is set to the position whose address is 7.

[0246]

Further, where the fourth replacement method of FIG. 17 is adopted as the replacement method for the replacement process of the demultiplexer 25 (FIG. 8) and besides 4096QAM is adopted as the modulation method, the bit number m of one symbol is 12 bits and the multiple b is 2.

[0247]

In this instance, according to FIG. 24, the memory 31 has 24 columns for storing 12×2 bits in the row

direction and stores $16,200/(12 \times 2)$ bits in the column direction. Then, the writing starting position for the first column from among the 24 columns of the memory 31 is set to the position whose address is 0; the writing starting position for the second column is set to the position whose address is 0; the writing starting position for the third column is set to the position whose address is 0; the writing starting position for the fourth column is set to the position whose address is 0; the writing starting position for the fifth column is set to the position whose address is 0; the writing starting position for the sixth column is set to the position whose address is 0; the writing starting position for the seventh column is set to the position whose address is 0; the writing starting position for the eighth column is set to the position whose address is 1; the writing starting position for the ninth column is set to the position whose address is 1; the writing starting position for the tenth column is set to the position whose address is 1; the writing starting position for the eleventh column is set to the position whose address is 2; the writing starting position for the twelfth column is set to the position whose address is 2; the writing starting position for the thirteenth column is set to the

position whose address is 2; the writing starting position for the fourteenth column is set to the position whose address is 3; the writing starting position for the fifteenth column is set to the position whose address is 7; the writing starting position for the sixteenth column is set to the position whose address is 9; the writing starting position for the seventeenth column is set to the position whose address is 9; the writing starting position for the eighteenth column is set to the position whose address is 9; the writing starting position for the nineteenth column is set to the position whose address is 10; the writing starting position for the twentieth column is set to the position whose address is 10; the writing starting position for the 21st column is set to the position whose address is 10; the writing starting position for the 22nd column is set to the position whose address is 10; the writing starting position for the 23rd column is set to the position whose address is 10; and the writing starting position for the 24th column is set to the position whose address is 11.

[0248]

Now, a transmission process carried out by the transmission apparatus 11 of FIG. 8 is described with reference to a flow chart of FIG. 25.

[0249]

The LDPC encoding section 21 waits that object data are supplied thereto and, at step S101, encodes the object data into LDPC codes and supplies the LDPC codes to the bit interleaver 22. Thereafter, the processing advances to step S102.

[0250]

At step S102, the bit interleaver 22 carries out bit interleave for the LDPC codes from the LDPC encoding section 21 and supplies the LDPC codes after the interleave to the mapping section 26. Thereafter, the processing advances to step S103.

[0251]

In particular, at step S102, the parity interleaver 23 in the bit interleaver 22 carries out parity interleave for the LDPC codes from the LDPC encoding section 21 and supplies the LDPC codes after the parity interleave to the column twist interleaver 24.

[0252]

The column twist interleaver 24 carries out column twist interleave for the LDPC codes from the parity interleaver 23 and then the demultiplexer 25 carries out a replacement process for the LDPC codes after the column twist interleave by the column twist interleaver 24. Then,

the LDPC codes after the replacement process are supplied from the demultiplexer 25 to the mapping section 26.

[0253]

At step S103, the mapping section 26 maps the m code bits of the LDPC codes from the demultiplexer 25 to symbols represented by signal points defined by the modulation method of orthogonal modulation carried out by the orthogonal modulation section 27 and supplies the mapped code bits to the orthogonal modulation section 27. Then, the processing advances to step S104.

[0254]

At step S104, the orthogonal modulation section 27 carries out orthogonal modulation of a carrier in accordance with the symbols from the mapping section 26. Then, the processing advances to step S105, at which the modulation signal obtained as a result of the orthogonal modulation is transmitted, whereafter the processing is ended.

[0255]

It is to be noted that the transmission process of FIG. 25 is carried out repetitively.

[0256]

By carrying out the parity interleave and the column twist interleave as described above, the tolerance

to erasure or burst errors where a plurality of code bits of an LDPC codes are transmitted as one symbol can be improved.

[0257]

Here, while, in FIG. 8, the parity interleaver 23 which is a block for carrying out parity interleave and the column twist interleaver 24 which is a block for carrying out column twist interleave are configured separately from each other for the convenience of description, the parity interleaver 23 and the column twist interleaver 24 may otherwise be configured integrally with each other.

[0258]

In particular, both of the parity interleave and the column twist interleave can be carried out by writing and reading out of code bits into and from a memory and can be represented by a matrix for converting addresses (write addresses) into which writing of code bits is to be carried out into addresses (readout addresses) from which reading out of code bits is to be carried out.

[0259]

Accordingly, if a matrix obtained by multiplying a matrix representative of the parity interleave and a matrix representative of the column twist interleave is

determined in advance, then if the matrix is used to convert code bits, then a result when parity interleave is carried out and then LDPC codes after the parity interleave are column twist interleaved can be obtained.

[0260]

Further, in addition to the parity interleaver 23 and the column twist interleaver 24, also the demultiplexer 25 may be configured integrally.

[0261]

In particular, also the replacement process carried out by the demultiplexer 25 can be represented by a matrix for converting a write address of the memory 31 for storing an LDPC code into a read address.

[0262]

Accordingly, if a matrix obtained by multiplication of a matrix representative of the parity interleave, another matrix representative of the column twist interleave and a further matrix representative of the replacement process is determined in advance, then the parity interleave, column twist interleave and replacement process can be carried out collectively by the determined matrix.

[0263]

It is to be noted that it is possible to carry out

only one of the parity interleave and the column twist interleave.

[0264]

Now, a simulation carried out with regard to the transmission apparatus 11 of FIG. 8 for measuring the error rate (bit error rate) is described with reference to FIGS. 26 to 28.

[0265]

The simulation was carried out adopting a communication path which has a flutter whose D/U is 0 dB.

[0266]

FIG. 26 shows a model of the communication path adopted in the simulation.

[0267]

In particular, A of FIG. 26 shows a model of the flutter adopted in the simulation.

[0268]

Meanwhile, B of FIG. 26 shows a model of a communication path which has the flutter represented by the model of A of FIG. 26.

[0269]

It is to be noted that, in B of FIG. 26, H represents the model of the flutter of A of FIG. 26. Further, in B of FIG. 26, N represents ICI (Inter Carrier

Interference), and in the simulation, an expected value $E[N^2]$ of the power was approximated by AWGN.

[0270]

FIGS. 27 and 28 illustrate relationships between the error rate obtained by the simulation and the Doppler frequency f_d of the flutter.

[0271]

It is to be noted that FIG. 27 illustrates a relationship between the error rate and the Doppler frequency f_d where the modulation method is 16QAM and the encoding rate (r) is $(3/4)$ and besides the replacement method is the first replacement method. Meanwhile, FIG. 28 illustrates the relationship between the error rate and the Doppler frequency f_d where the modulation method is 64QAM and the encoding rate (r) is $(5/6)$ and besides the replacement method is the first replacement method.

[0272]

Further, in FIGS. 27 and 28, a thick line curve indicates the relationship between the error rate and the Doppler frequency f_d where all of the parity interleave, column twist interleave and replacement process were carried out, and a thin line curve indicates the relationship between the error rate and the Doppler frequency f_d where only the replacement process from among

the parity interleave, column twist interleave and replacement process was carried out.

[0273]

In both of FIGS. 27 and 28, it can be recognized that the error rate improves (decreases) where all of the parity interleave, column twist interleave and replacement process are carried out rather than where only the replacement process is carried out.

[0274]

FIG. 29 is a block diagram showing an example of a configuration of the reception apparatus 12 of FIG. 7.

[0275]

Referring to FIG. 29, the reception apparatus 12 is a data processing apparatus for receiving a modulation signal from the transmission apparatus 11 (FIG. 7) and includes an orthogonal demodulation section 51, a demapping section 52, a deinterleaver 53 and an LDPC decoding section 56.

[0276]

The orthogonal demodulation section 51 receives a modulation signal from the transmission apparatus 11 and carries out orthogonal demodulation, and then supplies symbols obtained as a result of the orthogonal demodulation (values on the I and Q axes) to the

demapping section 52.

[0277]

The demapping section 52 carries out demapping of converting the symbols from the orthogonal demodulation section 51 to code bits of an LDPC code and supplies the code bits to the deinterleaver 53.

[0278]

The deinterleaver 53 includes a multiplexer (MUX) 54 and a column twist deinterleaver 55 and carries out deinterleave of the code bits of the LDPC code from the demapping section 52.

[0279]

In particular, the multiplexer 54 carries out a reverse replacement process (reverse process to the replacement process) corresponding to the replacement process carried out by the demultiplexer 25 of FIG. 8 for the LDPC codes from the demapping section 52, that is, a reverse replacement process of returning the positions of the code bits replaced by the replacement process to the original positions. Then, the multiplexer 54 supplies an LDPC code obtained as a result of the reverse replacement process to the column twist deinterleaver 55.

[0280]

The column twist deinterleaver 55 carries out

column twist deinterleave (reverse process to the column twist interleave) corresponding to the column twist interleave as the re-arrangement process carried out by the column twist interleaver 24 of FIG. 8, that is, for example, column twist deinterleave as a reverse re-arrangement process of returning the arrangement of the code bits of the LDPC code having an arrangement changed by the column twist interleave as the re-arrangement process to the original arrangement, for the LDPC code from the multiplexer 54.

[0281]

In particular, the column twist deinterleaver 55 carries out column twist deinterleave by writing the code bits of the LDPC code into and reading out the written code bits from the memory for deinterleave, the memory being configured similarly to the memory 31 shown in FIG. 22 and so forth.

[0282]

It is to be noted that, in the column twist deinterleaver 55, writing of the code bits is carried out in the row direction of the memory for deinterleave using read addresses upon reading out the codes from the memory 31 as write addresses. Meanwhile, readout of the code bits is carried out in the column direction of the memory

for deinterleave using the write addresses upon writing of the code bits into the memory 31 as read addresses.

[0283]

The LDPC codes obtained as a result of the column twist interleave are supplied from the column twist deinterleaver 55 to the LDPC decoding section 56.

[0284]

Here, while the LDPC code supplied from the demapping section 52 to the deinterleaver 53 has been obtained by the parity interleave, column twist interleave and replacement process carried out in this order therefor, the deinterleaver 53 carries out only a reverse replacement process corresponding to the replacement process and column twist deinterleave corresponding to the column twist interleave. Accordingly, parity deinterleave corresponding to the parity interleave (process reverse to the parity interleave), that is, the parity deinterleave returning the arrangement of the code bits of the LDPC codes, whose arrangement has been varied by the parity interleave, to the original arrangement, is not carried out.

[0285]

Accordingly, the LDPC code for which the reverse replacement process and the column twist deinterleave

have been carried out but the parity deinterleave has not been carried out is supplied from the (column twist deinterleaver 55 of the) deinterleaver 53 to the LDPC decoding section 56.

[0286]

The LDPC decoding section 56 carries out LDPC decoding of the LDPC code from the deinterleaver 53 using a conversion parity check matrix, obtained by carrying out at least column replacement corresponding to the parity interleave for the parity check matrix H used for the LDPC encoding by the LDPC encoding section 21 of FIG. 8, and outputs data obtained as a result of the LDPC decoding as a decoding result of the object data.

[0287]

FIG. 30 is a flow chart illustrating a reception process carried out by the reception apparatus 12 of FIG. 29.

[0288]

The orthogonal demodulation section 51 receives a modulation signal from the transmission apparatus 11 at step S111. Then, the processing advances to step S112, at which the orthogonal demodulation section 51 carries out orthogonal demodulation of the modulation signal. The orthogonal demodulation section 51 supplies symbols

obtained as a result of the orthogonal demodulation to the demapping section 52, whereafter the processing advances from step S112 to step S113.

[0289]

At step S113, the demapping section 52 carries out demapping of converting the symbols from the orthogonal demodulation section 51 into code bits of an LDPC code and supplies the code bits to the deinterleaver 53, whereafter the processing advances to step S114.

[0290]

At step S114, the deinterleaver 53 carries out deinterleave of the code bits of the LDPC code from the demapping section 52, whereafter the processing advances to step S115.

[0291]

In particular, at step S114, the multiplexer 54 in the deinterleaver 53 carries out a reverse replacement process for the LDPC code from the demapping section 52 and supplies LDPC code obtained as a result of the reverse replacement process to the column twist deinterleaver 55.

[0292]

The column twist deinterleaver 55 carries out column twist deinterleave for the LDPC code from the

multiplexer 54 and supplies an LDPC code obtained as a result of the column twist deinterleave to the LDPC decoding section 56.

[0293]

At step S115, the LDPC decoding section 56 carries out LDPC decoding of the LDPC code from the column twist deinterleaver 55 using a conversion parity check matrix obtained by carrying out at least column replacement corresponding to the parity interleave for the parity check matrix H used for the LDPC encoding by the LDPC encoding section 21 of FIG. 8, and outputs data obtained by the LDPC decoding as a decoding result of the object data. Thereafter, the processing is ended.

[0294]

It is to be noted that the reception process of FIG. 30 is carried out repetitively.

[0295]

Also in FIG. 29, the multiplexer 54 for carrying out the reverse replacement process and the column twist deinterleaver 55 for carrying out the column twist deinterleave are configured separately from each other for the convenience of description similarly as in the case of FIG. 8. However, the multiplexer 54 and the column twist deinterleaver 55 can be configured

integrally with each other.

[0296]

Further, where the transmission apparatus 11 of FIG. 8 does not carry out the column twist interleave, there is no necessity to provide the column twist deinterleaver 55 in the reception apparatus 12 of FIG. 29.

[0297]

Now, the LDPC decoding carried out by the LDPC decoding section 56 of FIG. 29 is further described.

[0298]

The LDPC decoding section 56 of FIG. 29 carries out LDPC decoding of an LDPC code, for which the reverse replacement process and the column twist deinterleave have been carried out but the parity deinterleave has not been carried out, from the column twist deinterleaver 55 as described above using a conversion parity check matrix obtained by carrying out at least column replacement corresponding to the parity interleave for the parity check matrix H used for the LDPC encoding by the LDPC encoding section 21 of FIG. 8.

[0299]

Here, LDPC decoding which can suppress the operation frequency within a sufficiently implementable range while suppressing the circuit scale by carrying out

the LDPC decoding using the conversion parity check matrix has been proposed formerly (refer to, for example, Japanese Patent Laid-Open No. 2004-343170).

[0300]

Thus, the formerly proposed LDPC decoding which uses a conversion parity check matrix is described first with reference to FIGS. 31 to 34.

[0301]

FIG. 31 shows an example of the parity check matrix H of an LDPC code whose code length N is 90 and encoding rate is $2/3$.

[0302]

It is to be noted that, in FIG. 31, 0 is represented by a period (.) (this similarly applies also to FIGS. 32 and 33 hereinafter described).

[0303]

In the parity check matrix H of FIG. 31, the parity matrix has a staircase structure.

[0304]

FIG. 32 illustrates a parity check matrix H' obtained by applying row replacement of an expression (8) and column replacement of an expression (9) to the parity check matrix H of FIG. 31.

[0305]

Row replacement: $6s+t+1\text{th row} \rightarrow 5t+s+1\text{th row}$

... (8)

[0306]

Column replacement: $6x+y+61\text{th column} \rightarrow 5y+x+61\text{th column}$

... (9)

[0307]

However, in the expressions (8) and (9), s , t , x and y are integers within the ranges of $0 \leq s < 5$, $0 \leq t < 6$, $0 \leq x < 5$ and $0 \leq t < 6$, respectively.

[0308]

According to the row replacement of the expression (8), the replacement is carried out in such a manner that the 1st, 7th, 13th, 19th and 25th rows each of whose numbers indicates a remainder of 1 where it is divided by 6 are replaced to the 1st, 2nd, 3rd, 4th and 5th rows, and the 2nd, 8th, 14th, 20th and 26th rows each of whose numbers indicates a remainder of 2 where it is divided by 6 are replaced to 6th, 7th, 8th, 9th and 10th rows.

[0309]

On the other hand, according to the column replacement of the expression (9), the replacement is carried out for the 61st and succeeding columns (parity matrix) such that the 61st, 67th, 73rd, 79th and 85th

columns each of whose numbers indicates a remainder of 1 where it is divided by 6 are replaced to 61st, 62nd, 63rd, 64th and 65th columns, and the 62nd, 68th, 74th, 80th and 86th columns each of whose numbers indicates a remainder of 2 where it is divided by 6 are replaced to 66th, 67th, 68th, 69th and 70th columns.

[0310]

A matrix obtained by carrying out replacement of the rows and the columns for the parity check matrix H of FIG. 31 is a parity check matrix H' of FIG. 32.

[0311]

Here, even if the row replacement of the parity check matrix H is carried out, this does not have an influence on the arrangement of the code bits of the LDPC code.

[0312]

Meanwhile, the column replacement of the expression (9) corresponds to parity interleave when the information length K , the unit column number P of the cyclic structure and the divisor q ($= M/P$) of the parity length M (here, 30) in the parity interleave of interleaving the $K+qx+y+1$ th code bit to the position of the $K+Py+x+1$ th code bit are set to 60, 5 and 6, respectively.

[0313]

If the parity check matrix H' (hereinafter referred to suitably as replacement parity check matrix) of FIG. 32 is multiplied by a result of replacement same as that of the expression (9) for the LDPC code of the parity check matrix H (hereinafter referred to suitably as original parity check matrix) of FIG. 31, then the 0 vector is outputted. In particular, where a row vector obtained by applying the column replacement of the expression (9) for the row vector c as the LDPC code (one codeword) of the original parity check matrix H is represented by c' , since Hc^T becomes the 0 vector on the basis of the characteristic of the parity check matrix, also $H'c'^T$ naturally becomes the 0 vector.

[0314]

From the foregoing, the conversion parity check matrix H' of FIG. 32 becomes the parity check matrix of an LDPC code c' obtained by carrying out the column replacement of the expression (9) for the LDPC code c of the original parity check matrix H .

[0315]

Accordingly, by carrying out the column replacement of the expression (9) for the LDPC code c of the original parity check matrix H , decoding (LDPC decoding) the LDPC code c' after the column replacement using the parity

check matrix H' of FIG. 32 and then carrying out reverse replacement to the column replacement of the expression (9) for result of decoding, a decoding result similar to that obtained where the LDPC code of the original parity check matrix H is decoded using the parity check matrix H can be obtained.

[0316]

FIG. 33 shows the conversion parity check matrix H' of FIG. 32 wherein a space is provided between units of 5×5 matrices.

[0317]

In FIG. 33, the conversion parity check matrix H' is represented by a combination of a unit matrix of 5×5 elements, another matrix (hereinafter referred to suitably as quasi unit matrix) which corresponds to the unit matrix whose element or elements of 1 are changed into an element or elements of 0, a further matrix (hereinafter referred to suitably as shift matrix) which corresponds to the unit matrix or quasi unit matrix after it is cyclically shifted (cyclic shift), a still further matrix (hereinafter referred to suitably as sum matrix) of two or more of the unit matrix, quasi unit matrix and shift matrix, and a 0 matrix of 5×5 elements.

[0318]

It can be regarded that the conversion parity check matrix H' of FIG. 33 is composed of a unit matrix, a quasi unit matrix, a shift matrix, a sum matrix and a 0 matrix of 5×5 elements. Therefore, the matrices of 5×5 elements which compose the conversion parity check matrix H' are hereinafter referred to as component matrices.

[0319]

For decoding of an LDPC code represented by a parity check matrix represented by a matrix of $P \times P$ components, an architecture which carries out check node mathematical operation and variable node mathematical operation simultaneously for P check nodes and P variable nodes can be used.

[0320]

FIG. 34 is a block diagram showing an example of a configuration of a decoding apparatus which carries out such decoding as just described.

[0321]

In particular, FIG. 34 shows an example of a configuration of a decoding apparatus which carries out decoding of LDPC codes of the original parity check matrix H of FIG. 31 using the conversion parity check matrix H' of FIG. 33 obtained by carrying out at least the column replacement of the expression (9).

[0322]

The decoding apparatus of FIG. 34 includes an edge data storage memory 300 including six FIFOs 300_1 to 300_6 , a selector 301 for selecting the FIFOs 300_1 to 300_6 , a check node calculation section 302, two cyclic shift circuits 303 and 308, an edge data storage memory 304 including 18 FIFOs 304_1 to 304_{18} , a selector 305 for selecting the FIFOs 304_1 to 304_{18} , a reception data memory 306 for storing reception information, a variable node calculation section 307, a decoded word calculation section 309, a reception data re-arrangement section 310, and a decoded data re-arrangement section 311.

[0323]

First, a storage method of data into the edge data storage memories 300 and 304 is described.

[0324]

The edge data storage memory 300 includes the six FIFOs 300_1 to 300_6 the number of which is equal to a quotient when the row number 30 of the conversion parity check matrix H' of FIG. 33 is divided by the row number 5 of the component matrices. Each of the FIFOs 300_y ($y = 1, 2, \dots, 6$) has a plurality of stages of storage regions such that messages corresponding to five edges whose number is equal to the number of rows and the number of

columns of the component matrices can be read out from or written into the storage regions of each stage at the same time. Further, the number of stages of the storage regions of each FIFO 300_y is nine which is the maximum number of 1s (Hamming weight) in the row direction of the conversion parity check matrix of FIG. 33.

[0325]

In the FIFO 300₁, data (messages v_i from variable nodes) corresponding to the positions of the value 1 in the first to fifth rows of the conversion parity check matrix H' of FIG. 33 are stored in a closed form in the horizontal direction in the individual rows (in the form wherein 0 is ignored). In particular, if an element in the j row of the i th column is represented as (j, i) , then in the storage regions at the first stage of the FIFO 300₁, data corresponding to the positions of the value 1 of the unit matrix of 5×5 elements from $(1, 1)$ to $(5, 5)$ of the conversion parity check matrix H' are stored. In the storage regions at the second stage, data corresponding to the positions of the value 1 of a shift matrix from $(1, 21)$ to $(5, 25)$ of the conversion parity check matrix H' (a shift matrix obtained by cyclically shifting the unit matrix of 5×5 elements by three in the rightward direction). Also in the storage regions at

the third to eighth stages, data are stored in an associated relationship with the conversion parity check matrix H' . Then, in the storage regions at the ninth stage, data corresponding to the positions of the value of a shift matrix of (1, 86) to (5, 90) of the conversion parity check matrix H' (a shift matrix obtained by replacing the value 1 in the first row of the unit matrix of 5×5 elements with the value 0 and then cyclically shifting the unit matrix after the replacement by one in the leftward direction) are stored.

[0326]

In the FIFO 300₂, data corresponding to the positions of the value 1 from the sixth to tenth rows of the conversion parity check matrix H' of FIG. 33 are stored. In particular, in the storage region at the first stage of the FIFO 300₂, data corresponding to the positions of the value 1 of a first shift matrix which forms a sum matrix from (6, 1) to (10, 5) of the conversion parity check matrix H' (a sum matrix which is the sum of a first shift matrix obtained by cyclically shifting the unit matrix of 5×5 elements by one in the rightward direction and a second shift matrix obtained by cyclically shifting the unit matrix of 5×5 elements by two in the rightward direction) are stored. Further, in

the storage region at the second stage, data corresponding to the positions of the value 1 of the second shift matrix which forms the sum matrix from (6, 1) to (10, 5) of the conversion parity check matrix H' are stored.

[0327]

In particular, with regard to a component matrix whose weight is 2 or more, where the component matrix is represented in the form of the sum of plural ones from among a unit matrix of $P \times P$ elements having the weight 1, a quasi unit matrix which corresponds to the unit matrix whose one or more elements having the value 1 are replaced with 0 and a shift matrix obtained by cyclically shifting the unit matrix or the quasi unit matrix, data corresponding to the positions of the value 1 of the unit matrix, quasi unit matrix or shift matrix whose weight is 1 (messages corresponding to edges belonging to the unit matrix, quasi unit matrix or shift matrix) are stored into the same address (same FIFO from among the FIFOs 300₁ to 300₆).

[0328]

Also in the storage regions at the third to ninth stages, data are stored in an associated relationship with the conversion parity check matrix H' .

[0329]

Also the FIFOs 300_3 to 300_6 store data in an associated relationship with the conversion parity check matrix H' .

[0330]

The edge data storage memory 304 includes 18 FIFOs 304_1 to 304_{18} the number of which is equal to the quotient when the column number 90 of the conversion parity check matrix H' is divided by the column number 5 of the component matrix. Each edge data storage memory 304_x ($x = 1, 2, \dots, 18$) includes a plurality of stages of storage regions, and messages corresponding to five edges the number of which is equal to the number of rows and the number of columns of the conversion parity check matrix H' can be read out from or written into the storage regions of each stage at the same time.

[0331]

In the FIFO 304_1 , data corresponding to the positions of the value 1 from the first to fifth columns of the conversion parity check matrix H' of FIG. 33 (messages u_j from the check nodes) are stored in a closed form in the vertical direction in the individual columns (in the form wherein 0 is ignored). In particular, in the storage regions at the first stage of the FIFO 304_1 , data

corresponding to the positions of the value 1 of the unit matrix of 5×5 elements from (1, 1) to (5, 5) of the conversion parity check matrix H' are stored. In the storage regions at the second stage, data corresponding to the positions of the value of a first shift matrix which forms a sum matrix from (6, 1) to (10, 5) of the vertical parity check matrix H' (a sum matrix which is the sum of a first shift matrix obtained by cyclically shifting the unit matrix of 5×5 elements by one to the right and a second shift matrix obtained by cyclically shifting the unit matrix of 5×5 elements by two to the right) are stored. Further, in the storage regions at the third stage, data corresponding to the positions of the value 1 of the second shift matrix which forms the sum matrix from (6, 1) to (10, 5) of the vertical parity check matrix H' .

[0332]

In particular, with regard to a component matrix whose weight is 2 or more, where the component matrix is represented in the form of the sum of plural ones from among a unit matrix of $P \times P$ elements having the weight 1, a quasi unit matrix which corresponds to the unit matrix whose one or more elements having the value 1 are replaced with 0 and a shift matrix obtained by cyclically

shifting the unit matrix or the quasi unit matrix, data corresponding to the positions of the value 1 of the unit matrix, quasi unit matrix or shift matrix whose weight is 1 (messages corresponding to edges belonging to the unit matrix, quasi unit matrix or shift matrix) are stored into the same address (same FIFO from among the FIFOs 304_1 to 304_{18}).

[0333]

Also with regard to the storage regions at the fourth and fifth stages, data are stored in an associated relationship with the conversion parity check matrix H' . The number of stages of the storage regions of the FIFO 304_1 is 5 which is a maximum number of the number of 1s (Hamming weight) in the row direction in the first to fifth columns of the conversion parity check matrix H' .

[0334]

Also the FIFOs 304_2 and 304_3 store data in an associated relationship with the conversion parity check matrix H' similarly, and each length (stage number) of the FIFOs 304_2 and 304_3 is 5. Also the FIFOs 304_4 to 304_{12} store data in an associated relationship with the conversion parity check matrix H' similarly, and each length of the FIFOs 304_4 to 304_{12} is 3. Also the FIFOs 304_{13} to 304_{18} store data in an associated relationship

with the conversion parity check matrix H' similarly, and each length of the FIFOs 304_{13} to 304_{18} is 2.

[0335]

Now, operation of the decoding apparatus of FIG. 34 is described.

[0336]

The edge data storage memory 300 includes the six FIFOs 300_1 to 300_6 , and FIFOs into which data are to be stored are selected from among the FIFOs 300_1 to 300_6 in accordance with information (Matrix data) $D312$ representing to which row of the conversion parity check matrix H' five messages $D311$ supplied from the cyclic shift circuit 308 at the preceding stage belong. Then, the five messages $D311$ are stored collectively and in order into the selected FIFOs. Further, when data are to be read out, the edge data storage memory 300 reads out five messages $D300_1$ in order from the FIFO 300_1 and supplies the five messages $D300_1$ to the selector 301 at the succeeding stage. After the reading out of the messages from the FIFO 300_1 ends, the edge data storage memory 300 reads out the messages in order also from the FIFOs 300_2 to 300_6 and supplies the read out messages to the selector 301.

[0337]

The selector 301 selects the five messages from that FIFO from which data are currently read out from among the FIFOs 300_1 to 300_6 in accordance with a select signal D301 and supplies the five messages as messages D302 to the check node calculation section 302.

[0338]

The check node calculation section 302 includes five check node calculators 302_1 to 302_5 and carries out the check node mathematical operation in accordance with the expression (7) using the messages D302 ($D302_1$ to $D302_5$) (messages v_i of the expression (7)) supplied thereto through the selector 301. Then, the check node calculation section 302 supplies five messages D303 ($D303_1$ to $D303_5$) (messages u_j of the expression (7)) obtained as a result of the check node mathematical operation to the cyclic shift circuit 303.

[0339]

The cyclic shift circuit 303 cyclically shifts the five messages $D303_1$ to 303_5 determined by the check node calculation section 302 based on information (Matrix data) D305 regarding by what number of original unit matrices the corresponding edges are cyclically shifted in the conversion parity check matrix H' , and supplies a result of the cyclic shift as a message D304 to the edge

data storage memory 304.

[0340]

The edge data storage memory 304 includes 18 FIFOs 304₁ to 304₁₈. The edge data storage memory 304 selects a FIFO into which data are to be stored from among the FIFOs 304₁ to 304₁₈ in accordance with the information D305 regarding to which row of the conversion parity check matrix H' the five messages D304 supplied from the cyclic shift circuit 303 at the preceding stage belong and collectively stores the five messages D304 in order into the selected FIFO. On the other hand, when data are to be read out, the edge data storage memory 304 reads out five messages D306₁ in order from the FIFO 304₁ and supplies the messages D306₁ to the selector 305 at the succeeding stage. After the reading out of data from the FIFO 304₁ ends, the edge data storage memory 304 reads out messages in order also from the FIFOs 304₂ to 304₁₈ and supplies the messages to the selector 305.

[0341]

The selector 305 selects the five messages from the FIFO from which data are currently read out from among the FIFOs 304₁ to 304₁₈ in accordance with a select signal D307 and supplies the selected messages as messages D308 to the variable node calculation section 307 and the

decoded word calculation section 309.

[0342]

On the other hand, the reception data re-arrangement section 310 carries out the column replacement of the expression (9) to re-arrange an LDPC code D313 received through a communication path and supplies the re-arranged LDPC code D313 as reception data D314 to the reception data memory 306. The reception data memory 306 calculates and stores a reception LLR (logarithmic likelihood ratio) from the reception data D314 supplied thereto from the reception data re-arrangement section 310 and collects and supplies every five ones of the reception LLRs as reception values D309 to the variable node calculation section 307 and the decoded word calculation section 309.

[0343]

The variable node calculation section 307 includes five variable node calculators 307_1 to 307_5 and carries out variable node mathematical operation in accordance with the expression (1) using the messages D308 (308_1 to 308_5) (messages u_j of the expression (1)) supplied thereto through the selector 305 and the five reception values D309 (reception values u_{oi} of the expression (1)) supplied thereto from the reception data memory 306. Then, the

variable node calculation section 307 supplies messages D310 (D301₁ to D310₅) (messages v_i of the expression (1)) obtained as a result of the mathematical operation to the cyclic shift circuit 308.

[0344]

The cyclic shift circuit 308 cyclically shifts messages D310₁ to D310₅ calculated by the variable node calculation section 307 based on information regarding by what number of original unit matrices the corresponding edge is cyclically shifted in the conversion parity check matrix H' , and supplies a result of the cyclic shifting as a message D311 to the edge data storage memory 300.

[0345]

By carrying out the sequence of operations described above, decoding in one cycle of an LDPC code can be carried out. In the decoding apparatus of FIG. 34, after an LDPC code is decoded by a predetermined number of times, a final decoding result is determined by the decoded word calculation section 309 and the decoded data re-arrangement section 311 and then outputted.

[0346]

In particular, the decoded word calculation section 309 includes five decoded word calculators 309₁ to 309₅ and acts as a final stage in a plurality of cycles of

decoding to calculate a decoding result (decoded word) in accordance with the expression (5) using the five messages D308 ($D308_1$ to $D308_5$) (messages u_j of the expression (5)) outputted from the selector 305 and the five reception values D309 (reception values u_{0i} of the expression (5)) outputted from the reception data memory 306. Then, the decoded word calculation section 309 supplies decoded data D315 obtained as a result of the calculation to the decoded data re-arrangement section 311.

[0347]

The decoded data re-arrangement section 311 carries out reverse replacement to the column replacement of the expression (9) for the decoded data D315 supplied thereto from the decoded word calculation section 309 to re-arrange the order of the decoded data D315 and outputs the re-arranged decoded data D315 as a decoding result D316.

[0348]

As described above, by applying one or both of row replacement and column replacement to a parity check matrix (original parity check matrix) to convert the parity check matrix into a parity check matrix (conversion parity check matrix) which can be represented

by a combination of a unit matrix of $P \times P$ elements, a quasi unit matrix which corresponds to the unit matrix whose element or elements of 1 are changed into an element or elements of 0, a shift matrix which corresponds to the unit matrix or quasi unit matrix after it is cyclically shifted, a sum matrix of two or more of the unit matrix, quasi unit matrix and shift matrix, and a 0 matrix of $P \times P$ elements as described above, it becomes possible to adopt for LDPC code decoding an architecture which carries out check node mathematical operation and variable node mathematical operation simultaneously for P check nodes and P variable nodes. Consequently, by carrying out the node mathematical operation simultaneously for P nodes, it is possible to suppress the operation frequency within an implementable range to carry out LDPC decoding.

[0349]

The LDPC decoding section 56 which composes the reception apparatus 12 of FIG. 29 carries out check node mathematical operation and variable node mathematical operation simultaneously for P check nodes and P variable nodes to carry out LDPC decoding similarly to the decoding apparatus of FIG. 34.

[0350]

In particular, it is assumed now to simplify description that the parity check matrix of an LDPC code outputted from the LDPC encoding section 21 which composes the transmission apparatus 11 of FIG. 8 is, for example, the parity check matrix H wherein the parity matrix has a staircase structure shown in FIG. 31. In this instance, the parity interleaver 23 of the transmission apparatus 11 carries out parity interleave for interleaving the $K+qx+y+1$ th code bit to the position of the $K+Py+x+1$ th code bit with the information length K set to 60, with the unit column number P of the cyclic structure set to 5 and with the divisor q ($= M/P$) of the parity length M to 6.

[0351]

Since this parity interleave corresponds to the column replacement of the expression (9), the LDPC decoding section 56 need not carry out the column replacement of the expression (9).

[0352]

Therefore, in the reception apparatus 12 of FIG. 29, an LDPC code for which parity deinterleave has not been carried out, that is, an LDPC code in a state wherein the column replacement of the expression (9) is carried out, is supplied from the column twist deinterleaver 55 to the

LDPC decoding section 56 as described above. The LDPC decoding section 56 carries out processing similar to that of the decoding apparatus of FIG. 34 except that the column replacement of the expression (9) is not carried out.

[0353]

In particular, FIG. 35 shows an example of a configuration of the LDPC decoding section 56 of FIG. 29.

[0354]

Referring to FIG. 35, the LDPC decoding section 56 is configured similarly to that of the decoding apparatus of FIG. 34 except that the reception data re-arrangement section 310 of FIG. 34 is not provided and carries out processing similar to that of the decoding apparatus of FIG. 34 except that the column replacement of the expression (9) is not carried out. Therefore, description of the LDPC decoding section 56 is omitted herein.

[0355]

Since the LDPC decoding section 56 can be configured without including the reception data re-arrangement section 310 as described above, it can be reduced in scale in comparison with the decoding apparatus of FIG. 34.

[0356]

It is to be noted that, while, in FIGS. 31 to 35, it is assumed that the code length N of the LDPC code is 90; the information length K is 60; the unit column number P (row number and column number of a component matrix) of the cyclic structure is 5; and the divisor q ($= M/P$) of the parity length M is 6, for simplified description, the code length N , information length K , unit column number P of the cyclic structure and the divisor q ($= M/P$) are not individually limited to the specific values given above.

[0357]

In particular, while the LDPC encoding section 21 in the transmission apparatus 11 of FIG. 8 outputs an LDPC code wherein, for example, the code length N is 64,800, the information length K is $N - Pq$ ($= N - M$), the unit column number P of the cyclic structure is 360 and the divisor q is M/P , the LDPC decoding section 56 shown in FIG. 35 can be applied also where LDPC decoding is carried out by carrying out the check node mathematical operation and the variable node mathematical operation simultaneously for P check nodes and P variable nodes in regard to such an LDPC code as just described.

[0358]

While the series of processes described above can

be executed by hardware, it may otherwise be executed by software. Where the series of processes is executed by software, a program which constructs the software is installed into a computer for universal use or the like.

[0359]

FIG. 36 shows an example of a configuration of an embodiment of a computer into which a program for executing the series of processes described hereinabove is installed.

[0360]

The program can be recorded in advance on a hard disk 405 or in a ROM 403 as a recording medium built in the computer.

[0361]

Or, the program can be stored (recorded) temporarily or permanently on or in a removable recording medium 411 such as a flexible disk, a CD-ROM (Compact Disc Read Only Memory), an MO (Magneto Optical) disc, a DVD (Digital Versatile Disc), a magnetic disc or a semiconductor memory. Such a removable recording medium 411 as just described can be provided as so-called package software.

[0362]

It is to be noted that the program not only can be

installed from such a removable recording medium 411 as described above into the computer but also can be installed into the hard disk 405 built in the computer where it is transferred thereto and received by a communication section 408. In this instance, the program may be transferred to the computer by wireless communication from a download site through an artificial satellite for digital satellite broadcasting or transferred to the computer by wire communication through a network such as a LAN (Local Area Network) or the Internet.

[0363]

The computer has a CPU (Central Processing Unit) 402 built therein. An input/output interface 410 is connected to the CPU 402 by a bus 401, and if an instruction is inputted to the CPU 402 through the input/output interface 410 when an inputting section 407 configured from a keyboard, a mouse, a microphone and so forth is operated by a user or in a like case, the CPU 402 executes the program stored in the ROM (Read Only Memory) 403. Or, the CPU 402 loads a program stored on the hard disk 405, a program transferred from a satellite or a network, received by the communication section 408 and installed in the hard disk 405 or a program read out

from the removable recording medium 411 loaded in a drive 409 and installed in the hard disk 405 into a RAM (Random Access Memory) 404 and executes the program. Consequently, the CPU 402 carries out processing in accordance with the flow chart described hereinabove or processing carried out by the configuration of the block diagram described hereinabove. Then, the CPU 402 outputs a result of the processing from an outputting section 406 configured from an LCD (Liquid Crystal Display), a speaker and so forth and transmits the processing result from the communication section 408 through the input/output interface 410 or records the processing result on the hard disk 405 as occasion demands.

[0364]

Here, in the present specification, processing steps which describe the program for causing the computer to carry out various processes need not necessarily be processed in a time series in accordance with the order described as a flow chart but include those processes to be executed in parallel or individually (for example, parallel processes or processes by an object).

[0365]

Further, the program may be processed by a single computer or may be processed by distributed processing by

a plurality of computers. Further, the program may be transferred to and executed by a computer at a remote place.

[0366]

It is to be noted that the embodiment of the present invention is not limited to the specific embodiment described hereinabove but may be altered in various manners without departing from the subject matter of the present invention.

[0367]

In particular, while, in the present embodiment, parity interleave and column twist interleave as a re-arrangement process are carried out for an LDPC code prescribed in DVB-S.2, the parity interleave can be applied to an LDPC code of a parity check matrix whose information matrix does not have a cyclic structure if the parity matrix has a staircase structure. Meanwhile, the column twist interleave as the re-arrangement process can be applied also to the LDPC code of a parity check matrix which can be varied so as to have a pseudo cyclic structure at least by column replacement, the QC-LDPC code (QC: Quasi-Cyclic) and so forth where the entire parity check matrix has a cyclic structure.

[0368]

In particular, only it is required for the parity check matrix of LDPC codes which are made an object of parity interleave that the parity matrix thereof has a staircase structure while there is no necessity that the information matrix has a cyclic structure.

[0369]

Further, the parity check matrix of an LDPC code which is made an object of column twist interleave as the re-arrangement process is not limited particularly in terms of the structure.

[0370]

It is to be noted that only it is necessary for the re-arrangement process that it can re-arrange code bits of an LDPC code such that a plurality of code bits corresponding to the value 1 included in one arbitrary row of the parity check matrix are not included in the same symbol and can be carried out by a method other than the column twist interleave. In particular, the re-arrangement process may be carried out by using not the memory 31 for storing data in the column direction and the row direction but, for example, a memory for storing data only in one direction and controlling the write address and the read address of the memory.

[0371]

Now, a process for LDPC encoding by the LDPC encoding section 21 of the transmission apparatus 11 is described further.

[0372]

For example, in the DVB-S.2 standard, LDPC encoding of the two different code lengths N of 64,800 bits and 16,200 bits are prescribed.

[0373]

And, for the LDPC code whose code length N is 64,800 bits, the 11 encoding rates $1/4$, $1/3$, $2/5$, $1/2$, $3/5$, $2/3$, $3/4$, $4/5$, $5/6$, $8/9$ and $9/10$ are prescribed, and for the LDPC code whose code length N is 16,200 bits, the 10 encoding rates $1/4$, $1/3$, $2/5$, $1/2$, $3/5$, $2/3$, $3/4$, $4/5$, $5/6$ and $8/9$ are prescribed.

[0374]

The LDPC encoding section 21 carries out encoding (error correction encoding) into LDPC codes of the different encoding rates whose code length N is 64,800 bits or 16,200 bits in accordance with a parity check matrix H prepared for each code length N and for each encoding rate.

[0375]

In particular, the LDPC encoding section 21 stores a parity check matrix initial value table hereinafter

described for producing a parity check matrix H for each code length N and for each encoding rate.

[0376]

Here, in the DVB-S.2 standard, LDPC codes of the two different code lengths N of 64,800 bits and 16,200 bits are prescribed as described hereinabove, and the 11 different encoding rates are prescribed for the LDPC code whose code length N is 64,800 bits and the 10 different encoding rates are prescribed for the LDPC code whose code length N is 16,200 bits.

[0377]

Accordingly, where the transmission apparatus 11 is an apparatus which carries out processing in compliance with the DVB-S.2 standard, parity check matrix initial value tables individually corresponding to the 11 different encoding rates for the LDPC code whose code length N is 64,800 bits and parity check matrix initial value tables individually corresponding to the 10 different encoding rates for the LDPC code whose code length N is 16,200 bits are stored in the LDPC encoding section 21.

[0378]

The LDPC encoding section 21 sets a code length N and an encoding rate r for LDPC codes, for example, in

response to an operation of an operator. The code length N and the encoding rate r set by the LDPC encoding section 21 are hereinafter referred to suitably as set code length N and set encoding rate r , respectively.

[0379]

The LDPC encoding section 21 places, based on the parity check matrix initial value tables corresponding to the set code length N and the set encoding rate r , elements of the value 1 of an information matrix H_A corresponding to an information length K ($= Nr = \text{code length } N - \text{parity length } M$) corresponding to the set code length N and the set encoding rate r in a period of 360 columns (unit column number P of the cyclic structure) in the column direction to produce a parity check matrix H .

[0380]

Then, the LDPC encoding section 21 extracts information bits for the information length K from object data which are an object of transmission such as image data or sound data supplied from the transmission apparatus 11. Further, the LDPC encoding section 21 calculates parity bits corresponding to the information bits based on the parity check matrix H to produce a codeword (LDPC code) for one code length.

[0381]

In other words, the LDPC encoding section 21 successively carries out mathematical operation of a parity bit of the codeword c which satisfies the following expression.

[0382]

$$Hc^T = 0$$

[0383]

Here, in the expression above, c indicates a row vector as the codeword (LDPC code), and c^T indicates inversion of the row vector c .

[0384]

Where, from within the row vector c as an LDPC code (one codeword), a portion corresponding to the information bits is represented by a row vector A and a portion corresponding to the parity bits is represented by a row vector T , the row vector c can be represented by an expression $c = [A|T]$ from the row vector A as the information bits and the row vector T as the parity bits.

[0385]

Meanwhile, the parity check matrix H can be represented, from the information matrix H_A of those of the code bits of the LDPC code which correspond to the information bits and the parity matrix H_T of those of the code bits of the LDPC code which correspond to the parity

bits by an expression $H = [H_A|H_T]$ (matrix wherein the elements of the information matrix H_A are elements on the left side and the elements of the parity matrix H_T are elements on the right side).

[0386]

Further, for example, in the DVB-S.2 standard, the parity check matrix H_T of the parity check matrix $H = [H_A|H_T]$ has a staircase structure.

[0387]

It is necessary for the parity check matrix H and the row vector $c = [A|T]$ as an LDPC code to satisfy the expression $Hc^T = 0$, and where the parity matrix H_T of the parity check matrix $H = [H_A|H_T]$ has a staircase structure, the row vector T as parity bits which configures the row vector $c = [A|T]$ which satisfies the expression $Hc^T = 0$ can be determined sequentially by setting the elements of each row to zero in order beginning with the elements in the first row of the column vector Hc^T in the expression $Hc^T = 0$.

[0388]

If the LDPC encoding section 21 determines a parity bit T for an information bit A , then it outputs a codeword $c = [A|T]$ represented by the information bit A and the parity bit T as an LDPC encoding result of the

information bit A.

[0389]

As described above, the LDPC encoding section 21 stores the parity check matrix initial value tables corresponding to the code lengths N and the encoding rates r in advance therein and carries out LDPC encoding of the set code length N and the set encoding rate r using a parity check matrix H produced from the parity check matrix initial value tables corresponding to the set code length N and the set encoding rate r.

[0390]

Each parity check matrix initial value table is a table which represents the position of elements of the value 1 of the information matrix H_A corresponding to the information length K corresponding to the code length N and the encoding rate r of the LDPC code of the parity check matrix H (LDPC code defined by the parity check matrix H) for every 360 rows (unit column number P of the periodic structure), and is produced in advance for a parity check matrix H for each code length N and each encoding rate r.

[0391]

FIGS. 37 to 82 illustrate the parity check matrix initial value tables for producing various parity check

matrices H including parity check matrix initial value tables prescribed in the DVB-S.2 standard.

[0392]

In particular, FIG. 37 shows the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 16,200 bits and an encoding rate r of 2/3.

[0393]

FIGS. 38 to 40 show the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 64,800 bits and an encoding rate r of 2/3.

[0394]

It is to be noted that FIG. 39 is a view continuing from FIG. 38 and FIG. 40 is a view continuing from FIG. 39.

[0395]

FIG. 41 shows the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 16,200 bits and an encoding rate r of 3/4.

[0396]

FIGS. 42 to 45 show the parity check matrix initial value table for a parity check matrix H prescribed in the

DVB-S.2 standard and having a code length N of 64,800 bits and an encoding rate r of $3/4$.

[0397]

It is to be noted that FIG. 43 is a view continuing from FIG. 42 and FIG. 44 is a view continuing from FIG. 43. Further, FIG. 45 is a view continuing from FIG. 44.

[0398]

FIG. 46 shows the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 16,200 bits and an encoding rate r of $4/5$.

[0399]

FIGS. 47 to 50 show the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 64,800 bits and an encoding rate r of $4/5$.

[0400]

It is to be noted that FIG. 48 is a view continuing from FIG. 47 and FIG. 49 is a view continuing from FIG. 48. Further, FIG. 50 is a view continuing from FIG. 49.

[0401]

FIG. 51 shows the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 16,200 bits

and an encoding rate r of $5/6$.

[0402]

FIGS. 52 to 55 show the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 64,800 bits and an encoding rate r of $5/6$.

[0403]

It is to be noted that FIG. 53 is a view continuing from FIG. 52 and FIG. 54 is a view continuing from FIG. 53. Further, FIG. 55 is a view continuing from FIG. 54.

[0404]

FIG. 56 shows the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 16,200 bits and an encoding rate r of $8/9$.

[0405]

FIGS. 57 to 60 show the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 64,800 bits and an encoding rate r of $8/9$.

[0406]

It is to be noted that FIG. 58 is a view continuing from FIG. 57 and FIG. 59 is a view continuing from FIG. 58. Further, FIG. 60 is a view continuing from FIG.

59.

[0407]

FIGS. 61 to 64 show the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 64,800 bits and an encoding rate r of $9/10$.

[0408]

It is to be noted that FIG. 62 is a view continuing from FIG. 61 and FIG. 63 is a view continuing from FIG. 62. Further, FIG. 64 is a view continuing from FIG. 63.

[0409]

FIGS. 65 and 66 show the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 64,800 bits and an encoding rate r of $1/4$.

[0410]

It is to be noted that FIG. 66 is a view continuing from FIG. 65.

[0411]

FIGS. 67 and 68 show the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 64,800 bits and an encoding rate r of $1/3$.

[0412]

It is to be noted that FIG. 68 is a view continuing from FIG. 67.

[0413]

FIGS. 69 and 70 show the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 64,800 bits and an encoding rate r of $2/5$.

[0414]

It is to be noted that FIG. 70 is a view continuing from FIG. 69.

[0415]

FIGS. 71 to 73 show the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 64,800 bits and an encoding rate r of $1/2$.

[0416]

It is to be noted that FIG. 72 is a view continuing from FIG. 71 and FIG. 73 is a view continuing from FIG. 72.

[0417]

FIGS. 74 to 76 show the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 64,800

bits and an encoding rate r of $3/5$.

[0418]

It is to be noted that FIG. 75 is a view continuing from FIG. 74 and FIG. 76 is a view continuing from FIG. 75.

[0419]

FIG. 77 shows the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 16,200 bits and an encoding rate r of $1/4$.

[0420]

FIG. 78 shows the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 16,200 bits and an encoding rate r of $1/3$.

[0421]

FIG. 79 shows the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 16,200 bits and an encoding rate r of $2/5$.

[0422]

FIG. 80 shows the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 16,200 bits

and an encoding rate r of $1/2$.

[0423]

FIG. 81 shows the parity check matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 16,200 bits and an encoding rate r of $3/5$.

[0424]

FIG. 82 shows the parity check matrix initial value table for a parity check matrix H having a code length N of 16,200 bits and an encoding rate r of $3/5$, which can be used in place of the parity check matrix initial value table of FIG. 81.

[0425]

The LDPC encoding section 21 of the transmission apparatus 11 determines a parity check matrix H in the following manner using the parity check matrix initial value tables.

[0426]

In particular, FIG. 83 illustrates a method for determining a parity check matrix H from a parity check matrix initial value table.

[0427]

It is to be noted that the parity check matrix initial value table of FIG. 83 indicates the parity check

matrix initial value table for a parity check matrix H prescribed in the DVB-S.2 standard and having a code length N of 16,200 bits and an encoding rate r of $2/3$ shown in FIG. 37.

[0428]

As described above, the parity check matrix initial value table is a table which represents the position of elements of the value 1 of a information matrix H_A corresponding to the information length K corresponding to the code length N and the encoding rate r of the LDPC code for every 360 columns (for every unit column number P of the cyclic structure), and in the first row of the parity check matrix initial value table, a number of row numbers of elements of the value 1 in the $1+360 \times (i-1)$ th column of the parity check matrix H (row numbers where the row number of the first row of the parity check matrix H is 0) equal to the number of column weights which the $1+360 \times (i-1)$ th column has.

[0429]

Here, it is assumed that the parity matrix H_T of the parity check matrix H corresponding to the parity length M has a staircase structure and is determined in advance. According to the parity check matrix initial value table, the information matrix H_A corresponding to

the information length K from within the parity check matrix H is determined.

[0430]

The row number $k+1$ of the parity check matrix initial value table differs depending upon the information length K .

[0431]

The information length K and the row number $k+1$ of the parity check matrix initial value table satisfy a relationship given by the following expression.

[0432]

$$K = (k + 1) \times 360$$

[0433]

Here, 360 in the expression above is the unit column number P of the cyclic structure.

[0434]

In the parity check matrix initial value table of FIG. 83, 13 numerical values are listed in the first to third rows, and three numerical values are listed in the fourth to $k+1$ th (in FIG. 83, 30th) rows.

[0435]

Accordingly, the number of column weights in the parity check matrix H determined from the parity check matrix initial value table of FIG. 83 is 13 in the first

to $1+360 \times (3-1)-1$ th rows but is 3 in the $1+360 \times (3-1)$ th to Kth rows.

[0436]

The first row of the parity check matrix initial value table of FIG. 83 includes 0, 2084, 1613, 1548, 1286, 1460, 3196, 4297, 2481, 3369, 3451, 4620 and 2622, and this indicates that, in the first column of the parity check matrix H, the elements in rows of the row numbers of 0, 2084, 1613, 1548, 1286, 1460, 3196, 4297, 2481, 3369, 3451, 4620 and 2622 have the value 1 (and besides the other elements have the value 0).

[0437]

Meanwhile, the second row of the parity check matrix initial value table of FIG. 83 includes 1, 122, 1516, 3448, 2880, 1407, 1847, 3799, 3529, 373, 971, 4358 and 3108, and this indicates that, in the 361st ($= 1+360 \times (2-1)$ th) column of the parity check matrix H, the elements in rows of the row numbers of 1, 122, 1546, 3448, 2880, 1407, 1847, 3799, 3529, 373, 971, 4358 and 3108 have the value 1.

[0438]

As given above, the parity check matrix initial value table represents the position of elements of the value 1 of the information matrix H_A of the parity check

matrix H for every 360 columns.

[0439]

Each of the columns of the parity check matrix H other than the $1+360 \times (i-1)$ th column, that is, each of the columns from $2+360 \times (i-1)$ th to $360 \times i$ th columns, includes elements of the value of 1 obtained by cyclically shifting the elements of the value of 1 of the $1+360 \times (i-1)$ th column which depend upon the parity check matrix initial value table periodically in the downward direction (in the downward direction of the column) in accordance with the parity length M.

[0440]

In particular, for example, the $2+360 \times (i-1)$ th column is a column obtained by cyclically shifting the $1+360 \times (i-1)$ th column in the downward direction by $M/360$ ($= q$), and the next $3+360 \times (i-1)$ th is a column obtained by cyclically shifting the $1+360 \times (i-1)$ th column in the downward direction by $2 \times M/360$ ($= 2 \times q$) and then cyclically shifting the cyclically shifted column ($2+360 \times (i-1)$ th column) in the downward direction by $M/360$ ($= q$).

[0441]

Now, if it is assumed that the numeral value in the j th column (j th from the left) in the i th row (i th row

from above) of the parity check matrix initial value table is represented by $b_{i,j}$ and the row number of the j th element of the value 1 in the w th column of the parity check matrix H is represented by H_{w-j} , then the row number H_{w-j} of the element of the value 1 in the w th column which is a column other than the $1+360 \times (i-1)$ th column of the parity check matrix H can be determined in accordance with the following expression.

[0442]

$$H_{w-j} = \text{mod}\{h_{i,j} + \text{mod}((w-1), P) \times q, M\}$$

[0443]

Here, $\text{mod}(x,y)$ signifies a remainder when x is divided by y .

[0444]

Meanwhile, P is a unit number of columns of the cyclic structure described hereinabove and is, for example, in the DVB-S.2 standard, 360. Further, q is a value $M/360$ obtained by dividing the parity length M by the unit column number P ($= 360$) of the cyclic structure.

[0445]

The LDPC encoding section 21 specifies the row number of the elements of the value 1 in the $1+360 \times (i-1)$ th column of the parity check matrix H from the parity check matrix initial value table.

[0446]

Further, the LDPC encoding section 21 determines the row number H_{w-j} of the element of the value 1 in the w th column which is a column other than the $1+360 \times (i-1)$ th column of the parity check matrix H and produces a parity check matrix H in which the elements of the row numbers obtained by the foregoing have the value 1.

[0447]

Now, variations of the method of replacement of code bits of an LDPC code in the replacement process by the replacement section 32 of the demultiplexer 25 in the transmission apparatus 11, that is, of the allocation pattern (hereinafter referred to as bit allocation pattern) of code bits of an LDPC code and symbol bits representative of a symbol, are described.

[0448]

In the demultiplexer 25, the code bits of the LDPC code are written in the column direction of the memory 31, which stores $(N/(mb)) \times (mb)$ bits in the column direction $\times$ row direction. Thereafter, the code bits are read out in a unit of mb bits in the row direction. Further, in the demultiplexer 25, the replacement section 32 replaces the mb code bits read out in the row direction of the memory 31 and determines the code bits after the

replacement as mb symbol bits of (successive) b symbols.

[0449]

In particular, the replacement section 32 determines the $i+1$ th bit from the most significant bit of the mb code bits read out in the row direction of the memory 31 as the code bit b_i and determines the $i+1$ th bit from the most significant bit of the mb symbol bits of the b (successive) symbols as the symbol bit y_i , and then replaces the mb code bits b_0 to b_{mb-1} in accordance with a predetermined bit allocation pattern.

[0450]

FIG. 84 shows an example of a bit allocation pattern which can be adopted where the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $5/6$ or $9/10$ and besides the modulation method is 4096QAM and the multiple b is 1.

[0451]

Where the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $5/6$ or $9/10$ and besides the modulation method is 4096QAM and the multiple b is 1, in the demultiplexer 25, the code bits written in the memory 31 for storing $(64,800/(12 \times 1)) \times (12 \times 1)$ bits in the column direction $\times$ row direction are read out in a unit of 12×1 (= mb) bits in the row

direction and supplied to the replacement section 32.

[0452]

The replacement section 32 replaces 12×1 (= mb) code bits b_0 to b_{11} such that the 12×1 (= mb) code bits b_0 to b_{11} to be read out from the memory 31 may be allocated to the 12×1 (= mb) symbol bits y_0 to y_{11} of one (= b) symbol as seen in FIG. 84.

[0453]

In particular, according to FIG. 84, the replacement section 32 carries out, with regard to both of an LDPC code having the encoding rate of 5/6 and an LDPC code having the encoding rate of 9/10 from among LDPC codes having the code length N of 64,800 bits, replacement for allocating

- the code bit b_0 to the symbol bit y_8 ,
- the code bit b_1 to the symbol bit y_0 ,
- the code bit b_2 to the symbol bit y_6 ,
- the code bit b_3 to the symbol bit y_1 ,
- the code bit b_4 to the symbol bit y_4 ,
- the code bit b_5 to the symbol bit y_5 ,
- the code bit b_6 to the symbol bit y_2 ,
- the code bit b_7 to the symbol bit y_3 ,
- the code bit b_8 to the symbol bit y_7 ,
- the code bit b_9 to the symbol bit y_{10} ,

the code bit b_{10} to the symbol bit y_{11} , and
the code bit b_{11} to the symbol bit y_9 .

[0454]

FIG. 85 shows an example of a bit allocation pattern which can be adopted where the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $5/6$ or $9/10$ and besides the modulation method is 4096QAM and the multiple b is 2.

[0455]

Where the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $5/6$ or $9/10$ and besides the modulation method is 4096QAM and the multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(64,800/(12 \times 2)) \times (12 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of 12×2 (= mb) bits in the row direction and supplied to the replacement section 32.

[0456]

The replacement section 32 replaces 12×2 (= mb) code bits b_0 to b_{23} such that the 12×2 (= mb) code bits b_0 to b_{23} to be read out from the memory 31 may be allocated to the 12×2 (= mb) symbol bits y_0 to y_{23} of two (= b) successive symbols as seen in FIG. 85.

[0457]

In particular, according to FIG. 85, the replacement section 32 carries out, with regard to both of an LDPC code having the encoding rate of 5/6 and an LDPC code having the encoding rate of 9/10 from among LDPC codes having the code length N of 64,800 bits, replacement for allocating

the code bit b_0 to the symbol bit y_8 ,
the code bit b_2 to the symbol bit y_0 ,
the code bit b_4 to the symbol bit y_6 ,
the code bit b_6 to the symbol bit y_1 ,
the code bit b_8 to the symbol bit y_4 ,
the code bit b_{10} to the symbol bit y_5 ,
the code bit b_{12} to the symbol bit y_2 ,
the code bit b_{14} to the symbol bit y_3 ,
the code bit b_{16} to the symbol bit y_7 ,
the code bit b_{18} to the symbol bit y_{10} ,
the code bit b_{20} to the symbol bit y_{11} ,
the code bit b_{22} to the symbol bit y_9 ,
the code bit b_1 to the symbol bit y_{20} ,
the code bit b_3 to the symbol bit y_{12} ,
the code bit b_5 to the symbol bit y_{18} ,
the code bit b_7 to the symbol bit y_{13} ,
the code bit b_9 to the symbol bit y_{16} ,
the code bit b_{11} to the symbol bit y_{17} ,

the code bit b_{13} to the symbol bit y_{14} ,
the code bit b_{15} to the symbol bit y_{15} ,
the code bit b_{17} to the symbol bit y_{19} ,
the code bit b_{19} to the symbol bit y_{22} ,
the code bit b_{21} to the symbol bit y_{23} , and
the code bit b_{23} to the symbol bit y_{21} .

[0458]

Here, the bit allocation pattern of FIG. 85 utilizes the bit allocation pattern of FIG. 84 wherein the multiple b is 1 without any modification. In particular, the allocation of the code bits $b_0, b_2, \dots, b_{22}$ to the symbol bits y_i and the allocation of the $b_1, b_3, \dots, b_{23}$ to the symbol bits y_i are similar to the allocation of the code bits b_0 to b_{11} to the symbol bits y_i of FIG. 84.

[0459]

FIG. 86 shows an example of a bit allocation pattern which can be adopted where the modulation method is 1024QAM and the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is $3/4$, $5/6$ or $8/9$ and besides the multiple b is 2 and also where the modulation method is 1024QAM and the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding length is $3/4$, $5/6$ or $9/10$ and besides the multiple b is 2.

[0460]

Where the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is $3/4$, $5/6$ or $8/9$ and the modulation method is 1024QAM and besides the multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(16,200/(10 \times 2)) \times (10 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of 10×2 (= mb) bits in the row direction and supplied to the replacement section 32.

[0461]

On the other hand, where the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $3/4$, $5/6$ or $9/10$ and the modulation method is 1024QAM and besides the multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(64,800/(10 \times 2)) \times (10 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of 10×2 (= mb) bits in the row direction and supplied to the replacement section 32.

[0462]

The replacement section 32 replaces 10×2 (= mb) code bits b_0 to b_{19} such that the 10×2 (= mb) code bits b_0 to b_{19} to be read out from the memory 31 may be

allocated to the 10×2 (= mb) symbol bits y_0 to y_{19} of two (= b) successive symbols as seen in FIG. 86.

[0463]

In particular, according to FIG. 86, the replacement section 32 carries out, with regard to all of the LDPC codes having the encoding rate of $3/4$, LDPC codes having the encoding rate of $5/6$ and LDPC codes having a further encoding rate of $8/9$ from among LDPC codes having the code length of 16,200 bits as well as LDPC code having the encoding rate of $3/4$, LDPC codes having the encoding rate of $5/6$ and LDPC codes having a further encoding rate of $9/10$ from among LDPC codes having another code length N of 64,800, replacement for allocating

- the code bit b_0 to the symbol bit y_8 ,
- the code bit b_1 to the symbol bit y_3 ,
- the code bit b_2 to the symbol bit y_7 ,
- the code bit b_3 to the symbol bit y_{10} ,
- the code bit b_4 to the symbol bit y_{19} ,
- the code bit b_5 to the symbol bit y_4 ,
- the code bit b_6 to the symbol bit y_9 ,
- the code bit b_7 to the symbol bit y_5 ,
- the code bit b_8 to the symbol bit y_{17} ,
- the code bit b_9 to the symbol bit y_6 ,

the code bit b_{10} to the symbol bit y_{14} ,
 the code bit b_{11} to the symbol bit y_{11} ,
 the code bit b_{12} to the symbol bit y_2 ,
 the code bit b_{13} to the symbol bit y_{18} ,
 the code bit b_{14} to the symbol bit y_{16} ,
 the code bit b_{15} to the symbol bit y_{15} ,
 the code bit b_{16} to the symbol bit y_0 ,
 the code bit b_{17} to the symbol bit y_1 ,
 the code bit b_{18} to the symbol bit y_{13} , and
 the code bit b_{19} to the symbol bit y_{12} .

[0464]

FIG. 87 shows an example of a bit allocation pattern which can be adopted where the modulation method is 4096QAM and the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is $5/6$ or $8/9$ and besides the multiple b is 2 and also where the modulation method is 4096QAM and the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $5/6$ or $9/10$ and besides the multiple b is 2.

[0465]

Where the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is $5/6$ or $8/9$ and the modulation method is 4096QAM and besides the

multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(16,200/(12 \times 2)) \times (12 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of 12×2 (= mb) bits in the row direction and supplied to the replacement section 32.

[0466]

On the other hand, where the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is 5/6 or 9/10 and the modulation method is 4096QAM and besides the multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(64,800/(12 \times 2)) \times (12 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of 12×2 (= mb) bits in the row direction and supplied to the replacement section 32.

[0467]

The replacement section 32 replaces 12×2 (= mb) code bits b_0 to b_{23} such that the 12×2 (= mb) bits to be read out from the memory 31 may be allocated to the 12×2 (= mb) symbol bits y_0 to y_{23} of two (= b) successive symbols as seen in FIG. 87.

[0468]

In particular, according to FIG. 87, the replacement section 32 carries out, with regard to all of

the LDPC codes having the encoding rate of $5/6$ and LDPC codes having the encoding rate of $8/9$ from among LDPC codes having the code length of 16,200 bits as well as LDPC codes having the encoding rate of $5/6$ and LDPC codes having the encoding rate of $9/10$ from among LDPC codes having another code length N of 64,800, replacement for allocating

the code bit b_0 to the symbol bit y_{10} ,
the code bit b_1 to the symbol bit y_{15} ,
the code bit b_2 to the symbol bit y_4 ,
the code bit b_3 to the symbol bit y_{19} ,
the code bit b_4 to the symbol bit y_{21} ,
the code bit b_5 to the symbol bit y_{16} ,
the code bit b_6 to the symbol bit y_{23} ,
the code bit b_7 to the symbol bit y_{18} ,
the code bit b_8 to the symbol bit y_{11} ,
the code bit b_9 to the symbol bit y_{14} ,
the code bit b_{10} to the symbol bit y_{22} ,
the code bit b_{11} to the symbol bit y_5 ,
the code bit b_{12} to the symbol bit y_6 ,
the code bit b_{13} to the symbol bit y_{17} ,
the code bit b_{14} to the symbol bit y_{13} ,
the code bit b_{15} to the symbol bit y_{20} ,
the code bit b_{16} to the symbol bit y_1 ,

the code bit b_{17} to the symbol bit y_3 ,
the code bit b_{18} to the symbol bit y_9 ,
the code bit b_{19} to the symbol bit y_2 ,
the code bit b_{20} to the symbol bit y_7 ,
the code bit b_{21} to the symbol bit y_8 ,
the code bit b_{22} to the symbol bit y_{12} , and
the code bit y_{23} to the symbol bit y_0 .

[0469]

According to the bit allocation patterns shown in FIGS. 84 to 87, the same bit allocation pattern can be adopted for a plurality of kinds of LDPC codes, and besides, the tolerance to errors can be set to a desired performance with regard to all of the plural kinds of LDPC codes.

[0470]

In particular, FIGS. 88 to 91 illustrates results of simulations of the BER (Bit Error Rate) where a replacement process is carried out in accordance with the bit allocation patterns of FIGS. 84 to 87.

[0471]

It is to be noted that, in FIGS. 88 to 91, the axis of abscissa represents E_s/N_0 (signal power to noise power ratio per one symbol) and the axis of ordinate represents the BER.

[0472]

Further, a solid line curve represents the BER where a replacement process is carried out and an alternate long and short dash line represents the BER where a replacement process is not carried out.

[0473]

FIG. 88 illustrates the BER where a replacement process in accordance with the bit allocation pattern of FIG. 84 is carried out for LDPC codes whose code length N is 64,800 and whose encoding rate is $5/6$ and $9/10$ adopting 4096QAM as the modulation method and setting the multiple b to 1.

[0474]

FIG. 89 illustrates the BER where a replacement process in accordance with the bit allocation pattern of FIG. 85 is carried out for LDPC codes whose code length N is 64,800 and whose encoding rate is $5/6$ and $9/10$ adopting 4096QAM as the modulation method and setting the multiple b to 2.

[0475]

It is to be noted that, in FIGS. 88 and 89, a graph having a triangular mark applied thereto represents the BER regarding the LDPC code having the encoding rate of $5/6$, and a graph having an asterisk applied thereto

represents the BER regarding the LDPC code having the encoding rate of 9/10.

[0476]

FIG. 90 illustrates the BER where a replacement process in accordance with the bit allocation pattern of FIG. 86 is carried out for LDPC codes whose code length N is 16,200 and whose encoding rate is $3/4$, $5/6$ and $8/9$ and for LDPC codes whose code length N is 64,800 and whose encoding rate is $3/4$, $5/6$ and $9/10$ adopting 1024QAM as the modulation method and setting the multiple b to 2.

[0477]

It is to be noted that, in FIG. 90, a graph having an asterisk applied thereto represents the BER regarding the LDPC code having the code length N of 64,800 and the encoding rate of $9/10$, and a graph having an upwardly directed triangular mark applied thereto represents the BER regarding the LDPC codes having the code length N of 64,800 and the encoding rate of $5/6$. Further, a graph having a square mark applied thereto represents the BER regarding the LDPC code having the code length N of 64,800 and the encoding rate of $3/4$.

[0478]

Further, in FIG. 90, a graph having a round mark applied thereto represents the BER regarding the LDPC

code having the code length N of 16,200 and the encoding rate of $8/9$, and a graph having a downwardly directed triangular mark applied thereto represents the BER regarding the LDPC code having the code length N of 16,200 and the encoding rate of $5/6$. Further, a graph having a plus mark applied thereto represents the BER regarding the LDPC code having the code length N of 16,200 and the encoding rate of $3/4$.

[0479]

FIG. 91 illustrates the BER where a replacement process in accordance with the bit allocation pattern of FIG. 87 is carried out for LDPC codes whose code length N is 16,200 and whose encoding rate is $5/6$ and $8/9$ and for LDPC codes whose code length N is 64,800 and whose encoding rate is $5/6$ and $9/10$ adopting 4096QAM as the modulation method and setting the multiple b to 2.

[0480]

It is to be noted that, in FIG. 91, a graph having an asterisk applied thereto represents the BER regarding the LDPC code having the code length N of 64,800 and the encoding rate of $9/10$, and a graph having an upwardly directed triangular mark applied thereto represents the BER regarding the LDPC codes having the code length N of 64,800 and the encoding rate of $5/6$.

[0481]

Further, in FIG. 91, a graph having a round mark applied thereto represents the BER regarding the LDPC code having the code length N of 16,200 and the encoding rate of $8/9$, and a graph having a downwardly directed triangular mark applied thereto represents the BER regarding the LDPC code having the code length N of 16,200 and the encoding rate of $5/6$.

[0482]

According to FIGS. 88 to 91, the same bit allocation pattern can be adopted with regard to a plurality of kinds of LDPC codes. Besides, the tolerance to errors can be set to a desired performance with regard to all of the plural kinds of LDPC codes.

[0483]

In particular, where a bit allocation pattern for exclusive use is adopted for each of a plurality of kinds of LDPC codes which have different code lengths and different encoding rates, the tolerance to an error can be raised to a very high performance. However, it is necessary to change the bit allocation pattern for each of a plurality of kinds of LDPC codes.

[0484]

On the other hand, according to the bit allocation

patterns of FIGS. 84 to 87, the same bit allocation pattern can be adopted for a plurality of kinds of LDPC codes which have different code lengths and different encoding rates, and the necessity to change the bit allocation pattern for each of a plurality of kinds of LDPC codes as in a case wherein a bit allocation pattern for exclusive use is adopted for each of a plurality of kinds of LDPC codes is eliminated.

[0485]

Further, according to the bit allocation patterns of FIGS. 84 to 87, the tolerance to errors can be raised to a high performance although it is a little lower than that where a bit allocation pattern for exclusive use is adopted for each of a plurality of kinds of LDPC codes.

[0486]

In particular, for example, where the modulation method is 4096QAM, the same bit allocation pattern in FIG. 84 or 85 can be used for all of the LDPC codes which have the code length N of 64,800 and the encoding rate of $5/6$ and $9/10$. Even where the same bit allocation pattern is adopted in this manner, the tolerance to errors can be raised to a high performance.

[0487]

Further, for example, where the modulation method

is 1024QAM, the same bit allocation pattern of FIG. 86 can be adopted for all of the LDPC codes which have the code length N of 16,200 and the encoding rate of $3/4$, $5/6$ and $8/9$ and the LDPC codes which have the code length N of 64,800 and the encoding rate of $3/4$, $5/6$ and $9/10$. Then, even if the same bit allocation pattern is adopted in this manner, the tolerance to errors can be raised to a high performance.

[0488]

Meanwhile, for example, where the modulation method is 4096QAM, the same bit allocation pattern of FIG. 87 can be adopted for all of the LDPC codes which have the code length N of 16,200 and the encoding rate of $5/6$ and $8/9$ and the LDPC codes which have the code length N of 64,800 and the encoding rate of $5/6$ and $9/10$. Then, even if the same bit allocation pattern is adopted in this manner, the tolerance to errors can be raised to a high performance.

[0489]

Variations of the bit allocation pattern are further described.

[0490]

FIG. 92 illustrates an example of a bit allocation pattern which can be adopted where the LDPC code is any

LDPC code which has the code length N of 16,200 or 64,800 bits and one of the encoding rates for the LDPC code defined by a parity check matrix H produced, for example, from any of the parity check matrix initial value tables shown in FIGS. 37 to 82 other than the encoding rate of $3/5$ and besides the modulation method is QPSK and the multiple b is 1.

[0491]

Where the LDPC code is an LDPC code which has the code length N of 16,200 or 64,800 bits and has the encoding rate other than $3/5$ and besides the modulation method is QPSK and the multiple b is 1, the demultiplexer 25 reads out code bits written in the memory 31 for storing $(N/(2 \times 1)) \times (2 \times 1)$ bits in the column direction $\times$ row direction in a unit of 2×1 (= mb) bits in the row direction and supplies the read out code bits to the replacement section 32.

[0492]

The replacement section 32 replaces the 2×1 (= mb) code bits b_0 and b_1 read out from the memory 31 in such a manner that the 2×1 (= mb) code bits b_0 and b_1 are allocated to the 2×1 (= mb) symbol bits y_0 and y_1 of one (= b) symbol as seen in FIG. 92.

[0493]

In particular, according to FIG. 92, the replacement section 32 carries out replacement for allocating

the code bit b_0 to the symbol bit y_0 and

the code bit b_1 to the symbol bit y_1 .

[0494]

It is to be noted that, in this instance, also it is possible to consider that replacement is not carried out and the code bits b_0 and b_1 are determined as they are as the symbol bits y_0 and y_1 , respectively.

[0495]

FIG. 93 shows an example of a bit allocation pattern which can be adopted where the LDPC code is an LDPC code which has the code length N of 16,200 or 64,800 bits and has the encoding rate other than $3/5$ and besides the modulation method is 16QAM and the multiple b is 2.

[0496]

Where the LDPC code is an LDPC code which has the code length N of 16,200 or 64,800 bits and has the encoding rate other than $3/5$ and besides the modulation method is 16QAM and the multiple b is 2, the demultiplexer 25 reads out the code bits written in the memory 31 for storing $(N/(4 \times 2)) \times (4 \times 2)$ bits in the column direction $\times$ row direction in a unit of 4×2 (=

mb) bits in the row direction and supplies the read out code bits to the replacement section 32.

[0497]

The replacement section 32 replaces the 4×2 (= mb) code bits b_0 to b_7 read out from the memory 31 in such a manner that the 4×2 (= mb) code bits are allocated to the 4×2 (= mb) symbol bits y_0 to y_7 of two (= b) successive symbols as seen in FIG. 93.

[0498]

In particular, according to FIG. 93, the replacement section 32 carries out replacement for allocating

the code bit b_0 to the symbol bit y_7 ,
the code bit b_1 to the symbol bit y_1 ,
the code bit b_2 to the symbol bit y_4 ,
the code bit b_3 to the symbol bit y_2 ,
the code bit b_4 to the symbol bit y_5 ,
the code bit b_5 to the symbol bit y_3 ,
the code bit b_6 to the symbol bit y_6 , and
the code bit b_7 to the symbol bit y_0 .

[0499]

FIG. 94 shows an example of a bit allocation pattern which can be adopted where the modulation method is 64QAM and the LDPC code is an LDPC code whose code

length N is 16,200 or 64,800 bits and whose encoding rate is any other than $3/5$ and besides the multiple b is 2.

[0500]

Where the LDPC code is an LDPC code whose code length N is 16,200 or 64,800 bits and whose encoding rate is any other than $3/5$ and the modulation method is 64QAM and besides the multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(N/(6 \times 2)) \times (6 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of 6×2 (= mb) bits in the row direction and supplied to the replacement section 32.

[0501]

The replacement section 32 replaces the 6×2 (= mb) code bits b_0 to b_{11} read out from the memory 31 such that the 6×2 (= mb) code bits b_0 to b_{11} may be allocated to the 6×2 (= mb) symbol bits y_0 to y_{11} of two (= b) successive symbols as seen in FIG. 94.

[0502]

In particular, according to FIG. 94, the replacement section 32 carries out replacement for allocating

the code bit b_0 to the symbol bit y_{11} ,

the code bit b_1 to the symbol bit y_7 ,

the code bit b_2 to the symbol bit y_3 ,
 the code bit b_3 to the symbol bit y_{10} ,
 the code bit b_4 to the symbol bit y_6 ,
 the code bit b_5 to the symbol bit y_2 ,
 the code bit b_6 to the symbol bit y_9 ,
 the code bit b_7 to the symbol bit y_5 ,
 the code bit b_8 to the symbol bit y_1 ,
 the code bit b_9 to the symbol bit y_8 ,
 the code bit b_{10} to the symbol bit y_4 , and
 the code bit b_{11} to the symbol bit y_0 .

[0503]

FIG. 95 shows an example of a bit allocation pattern which can be adopted where the modulation method is 256QAM and the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is any other than $3/5$ and besides the multiple b is 2.

[0504]

Where the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is any other than $3/5$ and the modulation method is 256QAM and besides the multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(64,800/(8 \times 2)) \times (8 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of $8 \times 2 (= mb)$ bits in

the row direction and supplied to the replacement section 32.

[0505]

The replacement section 32 replaces the 8×2 (= mb) code bits b_0 to b_{15} read out from the memory 31 such that the 8×2 (= mb) code bits b_0 to b_{15} may be allocated to the 8×2 (= mb) symbol bits y_0 to y_{15} of two (= b) successive symbols as seen in FIG. 95.

[0506]

In particular, according to FIG. 95, the replacement section 32 carries out replacement for allocating

the code bit b_0 to the symbol bit y_{15} ,
the code bit b_1 to the symbol bit y_1 ,
the code bit b_2 to the symbol bit y_{13} ,
the code bit b_3 to the symbol bit y_3 ,
the code bit b_4 to the symbol bit y_8 ,
the code bit b_5 to the symbol bit y_{11} ,
the code bit b_6 to the symbol bit y_9 ,
the code bit b_7 to the symbol bit y_5 ,
the code bit b_8 to the symbol bit y_{10} ,
the code bit b_9 to the symbol bit y_6 ,
the code bit b_{10} to the symbol bit y_4 ,
the code bit b_{11} to the symbol bit y_7 ,

the code bit b_{12} to the symbol bit y_{12} ,
the code bit b_{13} to the symbol bit y_2 ,
the code bit b_{14} to the symbol bit y_{14} , and
the code bit b_{15} to the symbol bit y_0 .

[0507]

FIG. 96 shows an example of a bit allocation pattern which can be adopted where the modulation method is 256QAM and the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is any other than $3/5$ and besides the multiple b is 1.

[0508]

Where the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is any other than $3/5$ and the modulation method is 256QAM and besides the multiple b is 1, in the demultiplexer 25, the code bits written in the memory 31 for storing $(16,200/(8 \times 1)) \times (8 \times 1)$ bits in the column direction $\times$ row direction are read out in a unit of 8×1 (= mb) bits in the row direction and supplied to the replacement section 32.

[0509]

The replacement section 32 replaces the 8×1 (= mb) code bits b_0 to b_7 read out from the memory 31 such that the 8×1 (= mb) code bits b_0 to b_7 may be allocated

to the 8×1 (= mb) symbol bits y_0 to y_7 of one (= b) symbol as seen in FIG. 96.

[0510]

In particular, according to FIG. 96, the replacement section 32 carries out replacement for allocating

the code bit b_0 to the symbol bit y_7 ,
the code bit b_1 to the symbol bit y_3 ,
the code bit b_2 to the symbol bit y_1 ,
the code bit b_3 to the symbol bit y_5 ,
the code bit b_4 to the symbol bit y_2 ,
the code bit b_5 to the symbol bit y_6 ,
the code bit b_6 to the symbol bit y_4 , and
the code bit b_7 to the symbol bit y_0 .

[0511]

FIG. 97 shows an example of a bit allocation pattern which can be adopted where the LDPC code is an LDPC code whose code length N is 16,200 or 64,800 bits and whose encoding rate is any other than $3/5$ and besides the modulation method is QPSK and the multiple b is 1.

[0512]

Where the LDPC code is an LDPC code whose code length N is 16,200 or 64,800 bits and whose encoding rate is any other than $3/5$ and besides the modulation method

is QPSK and the multiple b is 1, in the demultiplexer 25, the code bits written in the memory 31 for storing $(N/(2 \times 1)) \times (2 \times 1)$ bits in the column direction $\times$ row direction are read out in a unit of $2 \times 1 (= mb)$ bits in the row direction and supplied to the replacement section 32.

[0513]

The replacement section 32 replaces the $2 \times 1 (= mb)$ code bits b_0 and b_1 read out from the memory 31 such that the $2 \times 1 (= mb)$ code bits b_0 and b_1 may be allocated to the $2 \times 1 (= mb)$ symbol bits y_0 and y_1 of one $(= b)$ symbol as seen in FIG. 97.

[0514]

In particular, according to FIG. 97, the replacement section 32 carries out replacement for allocating

the code bit b_0 to the symbol bit y_0 , and

the code bit b_1 to the symbol bit y_2 .

[0515]

It is to be noted that, in this instance, also it is possible to consider that replacement is not carried out and the code bits b_0 and b_1 are determined as they are as the symbol bits y_0 and y_1 , respectively.

[0516]

FIG. 98 shows an example of a bit allocation pattern which can be adopted where the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $3/5$ and besides the modulation method is 16QAM and the multiple b is 2.

[0517]

Where the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $3/5$ and besides the modulation method is 16QAM and the multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(64,800/(4 \times 2)) \times (4 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of $4 \times 2 (= mb)$ bits in the row direction and supplied to the replacement section 32.

[0518]

The replacement section 32 replaces the $4 \times 2 (= mb)$ code bits b_0 to b_7 read out from the memory 31 such that the $4 \times 2 (= mb)$ code bits b_0 to b_7 may be allocated to the $4 \times 2 (= mb)$ symbol bits y_0 to y_7 of two $(= b)$ successive symbols as seen in FIG. 98.

[0519]

In particular, according to FIG. 98, the replacement section 32 carries out replacement for allocating

the code bit b_0 to the symbol bit y_0 ,
the code bit b_1 to the symbol bit y_5 ,
the code bit b_2 to the symbol bit y_1 ,
the code bit b_3 to the symbol bit y_2 ,
the code bit b_4 to the symbol bit y_4 ,
the code bit b_5 to the symbol bit y_7 ,
the code bit b_6 to the symbol bit y_3 , and
the code bit b_7 to the symbol bit y_6 .

[0520]

FIG. 99 shows an example of a bit allocation pattern which can be adopted where the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is $3/5$ and besides the modulation method is 16QAM and the multiple b is 2.

[0521]

Where the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is $3/5$ and besides the modulation method is 16QAM and the multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(16,200/(4 \times 2)) \times (4 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of 4×2 (= mb) bits in the row direction and supplied to the replacement section 32.

[0522]

The replacement section 32 replaces the 4×2 (= mb) code bits b_0 to b_7 read out from the memory 31 such that the 4×2 (= mb) code bits b_0 to b_7 may be allocated to the 4×2 (= mb) symbol bits y_0 to y_7 of two (= b) successive symbols as seen in FIG. 99.

[0523]

In particular, according to FIG. 99, the replacement section 32 carries out replacement for allocating

the code bit b_0 to the symbol bit y_7 ,
the code bit b_1 to the symbol bit y_1 ,
the code bit b_2 to the symbol bit y_4 ,
the code bit b_3 to the symbol bit y_2 ,
the code bit b_4 to the symbol bit y_5 ,
the code bit b_5 to the symbol bit y_3 ,
the code bit b_6 to the symbol bit y_6 , and
the code bit b_7 to the symbol bit y_0 .

[0524]

FIG. 100 shows an example of a bit allocation pattern which can be adopted where the modulation method is 64QAM and the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $3/5$ and besides the multiple b is 2.

[0525]

Where the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $3/5$ and the modulation method is 64QAM and besides the multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(64,800/(6 \times 2)) \times (6 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of $6 \times 2 (= mb)$ bits in the row direction and supplied to the replacement section 32.

[0526]

The replacement section 32 replaces the $6 \times 2 (= mb)$ code bits b_0 to b_{11} read out from the memory 31 such that the $6 \times 2 (= mb)$ code bits b_0 to b_{11} may be allocated to the $6 \times 2 (= mb)$ symbol bits y_0 to y_{11} of two $(= b)$ successive symbols as seen in FIG. 100.

[0527]

In particular, according to FIG. 100, the replacement section 32 carries out replacement for allocating

- the code bit b_0 to the symbol bit y_2 ,
- the code bit b_1 to the symbol bit y_7 ,
- the code bit b_2 to the symbol bit y_6 ,
- the code bit b_3 to the symbol bit y_9 ,
- the code bit b_4 to the symbol bit y_0 ,
- the code bit b_5 to the symbol bit y_3 ,

the code bit b_6 to the symbol bit y_1 ,
the code bit b_7 to the symbol bit y_8 ,
the code bit b_8 to the symbol bit y_4 ,
the code bit b_9 to the symbol bit y_{11} ,
the code bit b_{10} to the symbol bit y_5 , and
the code bit b_{11} to the symbol bit y_{10} .

[0528]

FIG. 101 shows an example of a bit allocation pattern which can be adopted where the modulation method is 64QAM and the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is $3/5$ and besides the multiple b is 2.

[0529]

Where the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is $3/5$ and the modulation method is 64QAM and besides the multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(16,200/(6 \times 2)) \times (6 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of $6 \times 2 (= mb)$ bits in the row direction and supplied to the replacement section 32.

[0530]

The replacement section 32 replaces the $6 \times 2 (= mb)$ code bits b_0 to b_{11} read out from the memory 31 such

that the 6×2 (= mb) code bits b_0 to b_{11} may be allocated to the 6×2 (= mb) symbol bits y_0 to y_{11} of two (= b) successive symbols as seen in FIG. 101.

[0531]

In particular, according to FIG. 101, the replacement section 32 carries out replacement for allocating

the code bit b_0 to the symbol bit y_{11} ,
the code bit b_1 to the symbol bit y_7 ,
the code bit b_2 to the symbol bit y_3 ,
the code bit b_3 to the symbol bit y_{10} ,
the code bit b_4 to the symbol bit y_6 ,
the code bit b_5 to the symbol bit y_2 ,
the code bit b_6 to the symbol bit y_9 ,
the code bit b_7 to the symbol bit y_5 ,
the code bit b_8 to the symbol bit y_1 ,
the code bit b_9 to the symbol bit y_8 ,
the code bit b_{10} to the symbol bit y_4 , and
the code bit b_{11} to the symbol bit y_0 .

[0532]

FIG. 102 shows an example of a bit allocation pattern which can be adopted where the modulation method is 256QAM and the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $3/5$

and besides the multiple b is 2.

[0533]

Where the LDPC code is an LDPC code whose code length N is 64,800 bits and whose encoding rate is $3/5$ and the modulation method is 256QAM and besides the multiple b is 2, in the demultiplexer 25, the code bits written in the memory 31 for storing $(64,800/(8 \times 2)) \times (8 \times 2)$ bits in the column direction $\times$ row direction are read out in a unit of 8×2 (= mb) bits in the row direction and supplied to the replacement section 32.

[0534]

The replacement section 32 replaces the 8×2 (= mb) code bits b_0 to b_{15} read out from the memory 31 such that the 8×2 (= mb) code bits b_0 to b_{15} may be allocated to the 8×2 (= mb) symbol bits y_0 to y_{15} of two (= b) successive symbols as seen in FIG. 102.

[0535]

In particular, according to FIG. 102, the replacement section 32 carries out replacement for allocating

- the code bit b_0 to the symbol bit y_2 ,
- the code bit b_1 to the symbol bit y_{11} ,
- the code bit b_2 to the symbol bit y_3 ,
- the code bit b_3 to the symbol bit y_4 ,

the code bit b_4 to the symbol bit y_0 ,
 the code bit b_5 to the symbol bit y_9 ,
 the code bit b_6 to the symbol bit y_1 ,
 the code bit b_7 to the symbol bit y_8 ,
 the code bit b_8 to the symbol bit y_{10} ,
 the code bit b_9 to the symbol bit y_{13} ,
 the code bit b_{10} to the symbol bit y_7 ,
 the code bit b_{11} to the symbol bit y_{14} ,
 the code bit b_{12} to the symbol bit y_6 ,
 the code bit b_{13} to the symbol bit y_{15} ,
 the code bit b_{14} to the symbol bit y_5 , and
 the code bit b_{15} to the symbol bit y_{12} .

[0536]

FIG. 103 shows an example of a bit allocation pattern which can be adopted where the modulation method is 256QAM and the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is $3/5$ and besides the multiple b is 1.

[0537]

Where the LDPC code is an LDPC code whose code length N is 16,200 bits and whose encoding rate is $3/5$ and the modulation method is 256QAM and besides the multiple b is 1, in the demultiplexer 25, the code bits written in the memory 31 for storing $(16,200/(8 \times 1)) \times$

(8×1) bits in the column direction $\times$ row direction are read out in a unit of 8×1 (= mb) bits in the row direction and supplied to the replacement section 32.

[0538]

The replacement section 32 replaces the 8×1 (= mb) code bits b_0 to b_7 read out from the memory 31 such that the 8×1 (= mb) code bits b_0 to b_7 may be allocated to the 8×1 (= mb) symbol bits y_0 to y_7 of one (= b) symbol as seen in FIG. 103.

[0539]

In particular, according to FIG. 103, the replacement section 32 carries out replacement for allocating

the code bit b_0 to the symbol bit y_7 ,
the code bit b_1 to the symbol bit y_3 ,
the code bit b_2 to the symbol bit y_1 ,
the code bit b_3 to the symbol bit y_5 ,
the code bit b_4 to the symbol bit y_2 ,
the code bit b_5 to the symbol bit y_6 ,
the code bit b_6 to the symbol bit y_4 , and
the code bit b_7 to the symbol bit y_0 .

[0540]

Now, the deinterleaver 53 which composes the reception apparatus 12 is described.

[0541]

FIG. 104 is a view illustrating processing of the multiplexer 54 which composes the deinterleaver 53.

[0542]

In particular, A of FIG. 104 shows an example of a functional configuration of the multiplexer 54.

[0543]

The multiplexer 54 is composed of a reverse replacement section 1001 and a memory 1002.

[0544]

The multiplexer 54 determines symbol bits of symbols supplied from the demapping section 52 at the preceding stage as an object of processing thereof and carries out a reverse replacement process corresponding to the replacement process carried out by the demultiplexer 25 of the transmission apparatus 11 (process reverse to the replacement process), that is, a reverse replacement process of returning the positions of the code bits (symbol bits) of the LDPC code replaced by the replacement process. Then, the multiplexer 54 supplies an LDPC code obtained as a result of the reverse replacement process to the column twist deinterleaver 55 at the succeeding stage.

[0545]

In particular, in the multiplexer 54, mb symbol bits $y_0, y_1, \dots, y_{mb-1}$ of b symbols are supplied in a unit of b (successive) symbols to the reverse replacement section 1001.

[0546]

The reverse replacement section 1001 carries out reverse replacement of returning the arrangement of the mb symbol bits y_0 to y_{mb-1} to the original arrangement of the mb code bits $b_0, b_1, \dots, b_{mb-1}$ (arrangement of the code bits b_0 to b_{mb-1} before the replacement by the replacement section 32 which composes the demultiplexer 25 on the transmission apparatus 11 side is carried out). The reverse replacement section 1001 outputs code bits b_0 to b_{mb-1} obtained as a result of the reverse replacement.

[0547]

The memory 1002 has a storage capacity of storing mb bits in the row (horizontal) direction and storing $N/(mb)$ bits in the column (vertical) direction similarly to the memory 31 which composes the demultiplexer 25 of the transmission apparatus 11 side. In other words, the reverse replacement section 1001 is configured from mb columns each of which stores $N/(mb)$ bits.

[0548]

However, in the memory 1002, writing of the code

bits of LDPC codes outputted from the reverse replacement section 1001 is carried out in a direction in which reading out of code bits from the memory 31 of the demultiplexer 25 of the transmission apparatus 11 is carried out, and reading out of code bits written in the memory 1002 is carried out in a direction in which writing of code bits into the memory 31 is carried out.

[0549]

In particular, the multiplexer 54 of the reception apparatus 12 successively carries out writing of code bits of an LDPC code outputted from the reverse replacement section 1001 in a unit of mb bits in the row direction beginning with the first row of the memory 1002 toward a lower low as seen in A of FIG. 104.

[0550]

Then, when the writing of code bits for one code length ends, the multiplexer 54 reads out the code bits in the column direction from the memory 1002 and supplies the code bits to the column twist deinterleaver 55 at the succeeding stage.

[0551]

Here, B of FIG. 104 is a view illustrating reading out of the code bits from the memory 1002.

[0552]

The multiplexer 54 carries out reading out of code bits of an LDPC code in a downward direction (column direction) from above of a column which composes the memory 1002 beginning with a leftmost column toward a right side column.

[0553]

Now, processing of the column twist deinterleaver 55 which composes the deinterleaver 53 of the reception apparatus 12 is described with reference to FIG. 105.

[0554]

FIG. 105 shows an example of a configuration of the memory 1002 of the multiplexer 54.

[0555]

The memory 1002 has a storage capacity for storing mb bits in the column (vertical) direction and stores $N/(mb)$ bits in the row (horizontal) direction and is composed of mb columns.

[0556]

The column twist deinterleaver 55 writes code bits of an LDPC code in the row direction into the memory 1002 and controls the position at which reading out is started when the code bits are read out in the column direction to carry out column twist deinterleave.

[0557]

In particular, the column twist deinterleaver 55 carries out a reverse re-arrangement process of suitably changing the reading out starting position at which reading out of code bits with regard to each of a plurality of columns is to be started to return the arrangement of code bits re-arranged by the column twist interleave to the original arrangement.

[0558]

Here, FIG. 105 shows an example of a configuration of the memory 1002 where the modulation method is 16QAM and the multiple b is 1. Accordingly, the bit number m of one symbol is 4 bits, and the memory 1002 includes four ($= mb$) columns.

[0559]

The column twist deinterleaver 55 carries out (in place of the multiplexer 54), writing of code bits of an LDPC code outputted from the replacement section 1001 in the row direction successively into the memory 1002 beginning with the first row toward a lowermost row.

[0560]

Then, if writing of code bits for one code length ends, then the column twist deinterleaver 55 carries out reading out of code bits in the downward direction (column direction) from a top of the memory 1002

beginning with a leftmost column toward a right side column.

[0561]

However, the column twist deinterleaver 55 carries out reading out of the code bits from the memory 1002 determining the writing starting position upon writing of the code bits by the column twist interleaver 24 on the transmission apparatus 11 side to a reading out starting position of the code bits.

[0562]

In particular, if the address of the position of the top of each column is determined as 0 and the address of each position in the column direction is represented by an integer given in an ascending order, then where the modulation method is 16QAM and the multiple b is 1, the column twist deinterleaver 55 sets the reading out starting position for the leftmost column to the position whose address is 0, sets the reading out starting position for the second column (from the left) to the position whose address is 2, sets the reading out starting position for the third column to the position whose address is 4, and sets the reading out starting position for the fourth column to the position whose address is 7.

[0563]

It is to be noted that, with regard to each of those columns whose reading out starting position has an address other than 0, reading out of code bits is carried out such that, after such reading out is carried out down to the lowermost position, the reading out position is returned to the top (position whose address is 0) of the column and the reading out is carried out downwardly to the position immediately preceding to the reading out starting position. Then, after that, reading out is carried out from the next (right) column.

[0564]

By carrying out such column twist interleave as described above, the arrangement of the code bits rearranged by the column twist interleave is returned to the original arrangement.

[0565]

FIG. 106 is a block diagram showing another example of the configuration of the reception apparatus 12.

[0566]

Referring to FIG. 106, the reception apparatus 12 is a data processing apparatus which receives a modulation signal from the transmission apparatus 11 and includes an orthogonal demodulation section 51, a

demapping section 52, a deinterleaver 53 and an LDPC decoding section 1021.

[0567]

The orthogonal demodulation section 51 receives a modulation signal from the transmission apparatus 11, carries out orthogonal demodulation and supplies symbols (values in the I and Q axis directions) obtained as a result of the orthogonal demodulation to the demapping section 52.

[0568]

The demapping section 52 carries out demapping of converting the symbols from the orthogonal demodulation section 51 into code bits of an LDPC code and supplies the code bits to the deinterleaver 53.

[0569]

The deinterleaver 53 includes a multiplexer (MUX) 54, a column twist deinterleaver 55 and a parity deinterleaver 1011 and carries out deinterleave of the code bits of the LDPC code from the demapping section 52.

[0570]

In particular, the multiplexer 54 determines an LDPC code from the demapping section 52 as an object of processing thereof and carries out a reverse replacement process corresponding to the replacement process carried

out by the demultiplexer 25 of the transmission apparatus 11 (reverse process to the replacement process), that is, a reverse replacement process of returning the positions of the code bits replaced by the replacement process to the original positions. Then, the multiplexer 54 supplies an LDPC code obtained as a result of the reverse replacement process to the column twist deinterleaver 55.

[0571]

The column twist deinterleaver 55 determines the LDPC code from the multiplexer 54 as an object of processing and carries out column twist deinterleave corresponding to the column twist interleave as a rearrangement process carried out by the column twist interleaver 24 of the transmission apparatus 11.

[0572]

The LDPC code obtained as a result of the column twist deinterleave is supplied from the column twist deinterleaver 55 to the parity deinterleaver 1011.

[0573]

The parity deinterleaver 1011 determines the code bits after the column twist deinterleave by the column twist deinterleaver 55 as an object of processing thereof and carries out parity deinterleave corresponding to the parity interleave carried out by the parity interleaver

23 of the transmission apparatus 11 (reverse process to the parity interleave), that is, parity deinterleave of returning the arrangement of the code bits of the LDPC code whose arrangement was changed by the parity interleave to the original arrangement.

[0574]

The LDPC code obtained as a result of the parity deinterleave is supplied from the parity deinterleaver 1011 to the LDPC decoding section 1021.

[0575]

Accordingly, in the reception apparatus 12 of FIG. 106, the LDPC code for which the reverse replacement process, column twist deinterleave and parity deinterleave have been carried out, that is, an LDPC code obtained by LDPC coding in accordance with the parity check matrix H , is supplied to the LDPC decoding section 1021.

[0576]

The LDPC decoding section 1021 carries out LDPC decoding of the LDPC code from the deinterleaver 53 using the parity check matrix H itself used for LDPC encoding by the LDPC encoding section 21 of the transmission apparatus 11 or a conversion parity check matrix obtained by carrying out at least column conversion corresponding

to the parity interleave for the parity check matrix H .
Then, the LDPC decoding section 1021 outputs data
obtained by the LDPC decoding as a decoding result of the
object data.

[0577]

Here, in the reception apparatus 12 of FIG. 106,
since an LDPC code obtained by LDPC encoding in
accordance with the parity check matrix H is supplied
from the (parity deinterleaver 1011 of) the deinterleaver
53 to the LDPC decoding section 1021, where the LDPC
decoding of the LDPC code is carried out using the parity
check matrix H itself used for the LDPC encoding by the
LDPC encoding section 21 of the transmission apparatus 11,
the LDPC decoding section 1021 can be configured, for
example, from a decoding apparatus which carries out LDPC
decoding in accordance with a full serial decoding method
wherein mathematical operation of messages (check node
messages and variable node messages) is carried out for
one by one node or another decoding apparatus wherein
LDPC decoding is carried out in accordance with a full
parallel decoding method wherein mathematical operation
of messages are carried out simultaneously (in parallel)
for all nodes.

[0578]

Further, where LDPC decoding of an LDPC code is carried out using a conversion parity check matrix obtained by carrying out at least column replacement corresponding to the parity interleave for the parity check matrix H used in the LDPC encoding by the LDPC encoding section 21 of the transmission apparatus 11, the LDPC decoding section 1021 can be confirmed from a decoding apparatus of an architecture which carries out the check node mathematical operation and the variable node mathematical operation simultaneously for P (or a divisor of P other than 1) check nodes and P variable nodes and which has a reception data re-arrangement section 310 for carrying out column replacement similar to the column replacement for obtaining a conversion parity check matrix for the LDPC code to re-arrange the code bits of the LDPC codes.

[0579]

It is to be noted that, while, in FIG. 106, the multiplexer 54 for carrying out the reverse replacement process, column twist deinterleaver 55 for carrying out the column twist deinterleave and parity deinterleaver 1011 for carrying out the parity deinterleave are configured separately from each other for the convenience of description, two or more of the multiplexer 54, column

twist deinterleaver 55 and parity deinterleaver 1011 can be configured integrally similarly to the parity interleaver 23, column twist interleaver 24 and demultiplexer 25 of the transmission apparatus 11.

[0580]

FIG. 107 is a block diagram showing a first example of a configuration of a reception system which can be applied to the reception apparatus 12.

[0581]

Referring to FIG. 107, the reception system includes an acquisition section 1101, a transmission line decoding processing section 1102 and an information source decoding processing section 1103.

[0582]

The acquisition section 1101 acquires a signal including an LDPC code obtained at least by LDPC encoding object data such as image data and music data of a program through a transmission line such as, for example, terrestrial digital broadcasting, satellite digital broadcasting, a CATV network, the Internet or some other network. Then, the acquisition section 1101 supplies the acquired signal to the transmission line decoding processing section 1102.

[0583]

Here, where the signal acquired by the acquisition section 1101 is broadcast, for example, from a broadcasting station through ground waves, satellite waves, a CATV (Cable Television) or the like, the acquisition section 1101 is configured from a tuner, an STB (Set Top Box) or the like. On the other hand, where the signal acquired by the acquisition section 1101 is transmitted in a multicast state as in the IPTV (Internet Protocol Television), for example, from a web server, the acquisition section 11 is configured from a network I/F (Inter face) such as, for example, an NIC (Network Interface Card).

[0584]

The transmission line decoding processing section 1102 carries out a transmission line decoding process including at least a process for correcting errors produced in the transmission line for the signal acquired through the transmission line by the acquisition section 1101, and supplies a signal obtained as a result of the transmission line decoding process to the information source decoding processing section 1103.

[0585]

In particular, the signal acquired through the transmission line by the acquisition section 1101 is a

signal obtained by carrying out at least error correction encoding for correcting errors produced in the transmission line, and for such a signal as just described, the transmission line decoding processing section 1102 carries out a transmission line decoding process such as, for example, an error correction process.

[0586]

Here, as the error correction encoding, for example, LDPC encoding, Reed-Solomon encoding and so forth are available. Here, as the error correction encoding, at least LDPC encoding is carried out.

[0587]

Further, the transmission line decoding process sometimes includes demodulation of a modulation signal and so forth.

[0588]

The information source decoding processing section 1103 carries out an information source decoding process including at least a process for decompressing compressed information into original information for the signal for which the transmission line decoding process has been carried out.

[0589]

In particular, the signal acquired through the

transmission line by the acquisition section 1101 has sometimes been processed by compression encoding for compressing information in order to reduce the data amount such as images, sound and so forth as information. In this instance, the information source decoding processing section 1103 carries out an information source decoding process such as a process (decompression process) for decompressing the compressed information into original information for a signal for which the transmission line decoding process has been carried out.

[0590]

It is to be noted that, where the signal acquired through the transmission line by the acquisition section 1101 has not been carried out compression encoding, the information source decoding processing section 1103 does not carry out the process of decompressing the compressed information into the original information.

[0591]

Here, as the decompression process, for example, MPEG decoding and so forth are available. Further, the transmission line decoding process sometimes includes descrambling in addition to the decompression process.

[0592]

In the reception system configured in such a manner

as described above, the acquisition section 1101 receives a signal obtained by carrying out compression encoding such as MPEG encoding for data of, for example, images, sound and so forth and further carrying out error correction encoding such as LDPC encoding for the compression encoded data through a transmission line. The signal is supplied to the transmission line decoding processing section 1102.

[0593]

In the transmission line decoding processing section 1102, processes similar to those carried out, for example, by the orthogonal demodulation section 51, demapping section 52, deinterleaver 53 and LDPC decoding section 56 (or LDPC decoding section 1021) are carried out as the transmission line decoding process for the signal from the acquisition section 1101. Then, a signal obtained as a result of the transmission line decoding process is supplied to the information source decoding processing section 1103.

[0594]

In the information source decoding processing section 1103, an information source decoding process such as MPEG decoding is carried out for the signal from the transmission line decoding processing section 1102, and

an image or sound obtained as a result of the information decoding process is outputted.

[0595]

Such a reception system of FIG. 107 as described above can be applied, for example, to a television tuner for receiving television broadcasting as digital broadcasting and so forth.

[0596]

It is to be noted that it is possible to configure the acquisition section 1101, transmission line decoding processing section 1102 and information source decoding processing section 1103 each as an independent apparatus (hardware (IC (Integrated Circuit) or the like) or a software module).

[0597]

Further, as regards the acquisition section 1101, transmission line decoding processing section 1102 and information source decoding processing section 1103, a set of the acquisition section 1101 and transmission line decoding processing section 1102, another set of the transmission line decoding processing section 1102 and information source decoding processing section 1103 or a further set of the acquisition section 1101, transmission line decoding processing section 1102 and information

source decoding processing section 1103 can be configured as a single independent apparatus.

[0598]

FIG. 108 is a block diagram showing a second example of the configuration of the reception system which can be applied to the reception apparatus 12.

[0599]

It is to be noted that, in FIG. 108, elements corresponding those in FIG. 107 are denoted by like reference numerals, and description of them is suitably omitted in the following description.

[0600]

The reception system of FIG. 108 is common to that of FIG. 107 in that it includes an acquisition section 1101, a transmission line decoding processing section 1102 and an information source decoding processing section 1103 but is different from that of FIG. 107 in that it newly includes an outputting section 1111.

[0601]

The outputting section 1111 is, for example, a display apparatus for displaying an image or a speaker for outputting sound and outputs an image, a sound of the like as a signal outputted from the information source decoding processing section 1103. In other words, the

outputting section 1111 displays an image or outputs sound.

[0602]

Such a reception system of FIG. 108 as described above can be applied, for example, to a TV (television receiver) for receiving a television broadcast as a digital broadcast, a radio receiver for receiving a radio broadcast and so forth.

[0603]

It is to be noted that, where the signal acquired by the acquisition section 1101 is not in a form wherein compression encoding is not applied, a signal outputted from the transmission line decoding processing section 1102 is supplied to the outputting section 1111.

[0604]

FIG. 109 is a block diagram showing a third example of the configuration of the reception system which can be applied to the reception apparatus 12.

[0605]

It is to be noted that, in FIG. 109, corresponding elements to those of FIG. 107 are denoted by like reference numerals, and in the following description, description of them is suitably omitted.

[0606]

The reception system of FIG. 109 is common to that of FIG. 107 in that it includes an acquisition section 1101 and a transmission line decoding processing section 1102.

[0607]

However, the reception system of FIG. 109 is different from that of FIG. 107 in that it does not include the information source decoding processing section 1103 but newly includes a recording section 1121.

[0608]

The recording section 1121 records (stores) a signal (for example, a TS packet of a TS of MPEG) outputted from the transmission line decoding processing section 1102 on or into a recording (storage) medium such as an optical disk, a hard disk (magnetic disk) or a flash memory.

[0609]

Such a reception system of FIG. 109 as described above can be applied to a recorder for recording a television broadcast or the like.

[0610]

It is to be noted that, in FIG. 109, the reception system may include the information source decoding processing section 1103 such that a signal after an

information source decoding process has been carried out by the information source decoding processing section 1103, that is, an image or sound obtained by decoding, is recorded by the recording section 1121.

Claims

1. A data processing apparatus which interleaves data, comprising:

parity interleave means for carrying out parity interleave of interleaving parity bits of an LDPC code obtained by carrying out LDPC encoding in accordance with a parity check matrix in which a parity matrix which is a portion corresponding to parity bits of an LDPC (Low Density Parity Check) code has a staircase structure individually to positions of other ones of the parity bits.

2. The data processing apparatus according to claim 1, wherein:

the bit number M of the parity bits of the LDPC code is a value other than the prime numbers; and

where two ones of devisors of the bit number M of the parity bits other than 1 and M whose product is equal to the bit number M of the parity bits are represented by P and q ,

the bit number of information bits of the LDPC code is represented by K ,

an integer equal to or higher than 0 but lower than P is represented by x , and

another integer equal to or higher than 0 but lower

than q is represented by y ,

said parity interleave means interleaves a $K+qx+y+1$ th code bit from among the parity bits which are $K+1$ th to $K+M$ th code bits of the LDPC code to the position of the $K+Py+x+1$ th code bit.

3. The data processing apparatus according to claim 2, further comprising

re-arrangement means for carrying out, where the LDPC code is transmitted such that two or more ones of the code bits of the LDPC code are transmitted as one symbol, a re-arrangement process for re-arranging the code bits of the LDPC code after the parity interleave such that a plurality of code bits corresponding to the value 1 included in one arbitrary row of the parity check matrix are not included in the same symbol.

4. The data processing apparatus according to claim 2, wherein:

the LDPC code is an LDPC code wherein an information matrix which is a portion of the parity check matrix of the LDPC code which corresponds to the information bits of the LDPC code has a cyclic structure; and

said data processing apparatus further comprises
re-arrangement means for carrying out column

twist interleave for changing, where a symbol is formed from code bits of the LDPC code written in a column direction of storage means for storing the code bits of the LDPC code in a row direction and the column direction and read out in the row direction, the writing starting position when the code bits of the LDPC code are written in the column direction of said storage means is changed for each of the columns of said storage means as a rearrangement process for re-arranging the code bits of the LDPC code.

5. The data processing apparatus according to claim 4, wherein the parity matrix in the parity check matrix of the LDPC code has a pseudo cyclic structure in which the parity matrix has a portion having a cyclic structure except part thereof, the structure being obtained by applying row replacement corresponding to the parity interleave.

6. The data processing apparatus according to claim 5, wherein:

where one symbol is formed from m ones of the code bits of the LDPC code, and

where the code length of the LDPC code is N bits and a predetermined positive integer is represented by b ,
said storage means stores mb bits in the row

direction and stores $N/(mb)$ bits in the column direction;

the code bits of the LDPC code being written in the column direction of said storage means, whereafter the code bits are read out in the row direction,

the mb code bits read out in the row direction of said storage means being converted into b symbols.

7. A data processing method for a data processing apparatus which interleaves data, comprising:

a step carried out by the data processing apparatus of carrying out parity interleave of interleaving parity bits of an LDPC code obtained by carrying out LDPC encoding in accordance with a parity check matrix in which a parity matrix which is a portion corresponding to parity bits of an LDPC (Low Density Parity Check) code has a staircase structure individually to positions of other ones of the parity bits.