

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 416 356**

51 Int. Cl.:

H04L 27/00	(2006.01)
H04L 27/26	(2006.01)
H04L 1/00	(2006.01)
H04L 5/00	(2006.01)
H03M 13/29	(2006.01)
H03M 13/25	(2006.01)
H03M 13/27	(2006.01)
H03M 13/11	(2006.01)
H04L 27/34	(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **24.10.2008 E 08253469 (4)**

97 Fecha y número de publicación de la concesión europea: **24.04.2013 EP 2056550**

54 Título: **Aparato y método de procesamiento de datos**

30 Prioridad:

30.10.2007 GB 0721269	30.10.2007 GB 0721271
30.10.2007 GB 0721272	30.10.2007 GB 0721270
19.11.2007 GB 0722645	20.11.2007 GB 0722728
26.11.2007 JP 2007304689	
26.11.2007 JP 2007304690	

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
31.07.2013

73 Titular/es:

SONY CORPORATION (100.0%)
1-7-1 KONAN MINATO-KU
TOKYO 108-0075, JP

72 Inventor/es:

TAYLOR, MATTHEW PAUL ATHOL;
ATUNGSIRI, SAMUEL ASANBENG;
WILSON, JOHN NICHOLAS;
YOKOKAWA, TAKASHI y
YAMAMOTO, MAKIKO

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 416 356 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Aparato y método de procesamiento de datos

5 CAMPO DE LA INVENCION

La presente invención se refiere a métodos y aparatos de procesamiento de datos para la recuperación de bits de datos a partir de varias señales sub-portadoras de símbolos Multiplexados por División de Frecuencias Ortogonales (OFDM) para formar un flujo de bits de salida.

10

Formas de realización de la presente invención pueden dar a conocer un receptor de OFDM.

ANTECEDENTES DE LA INVENCION

15 La norma de Difusión Vídeo Digital Terrestre (DVB-T) utiliza la Multiplexación por División de Frecuencias Ortogonales (OFDM) para comunicar datos que representan imágenes de vídeo y sonido a receptores mediante una señal de comunicaciones por radio de difusión. Se conoce, por la técnica anterior, que existen dos modos para la norma DVB-T que son conocidos como el modo de 2k y el modo de 8k. El modo de 2k proporciona 2048 sub-portadoras, mientras que el modo de 8k proporciona 8192 sub-portadoras. De modo similar, para la norma de Difusión de Vídeo Digital-Portátil (DVB-H) se ha proporciona un modo de 4k, en donde el número de sub-portadoras es 4096.

20

Los sistemas de codificación de corrección de errores, tales como la codificación LDPC/BCH, que han sido propuestos para realizar mejor DVB-T2 cuando el ruido y la degradación de los valores de símbolos, resultantes de la comunicación, no están en correlación. Los canales de difusión terrestres pueden sufrir un desvanecimiento correlacionado en los dominios del tiempo y de la frecuencia. En consecuencia, separando los bits de datos codificados en símbolos de datos diferentes y separando la comunicación de los símbolos de datos en señales de sub-portadoras diferentes del símbolo OFDM es tanto más posible, con lo que se puede aumentar el rendimiento de los sistemas de codificación de corrección de errores.

25

30 Con el fin de mejorar la integridad de los datos comunicados utilizando DVB-T o DVB-H, es conocido proporcionar un intercalador de símbolos con el fin de intercalar símbolos de datos de entrada puesto que estos símbolos son puestos en correspondencia con las señales de sub-portadoras de un símbolo OFDM. Para el modo de 2k y el modo 8k, se ha dado a conocer una disposición en donde la norma de DVB-T para generar las direcciones se pone en práctica en el establecimiento de la correspondencia. De forma análoga, para el modo de 4k de la norma DVB-H, se ha dado a conocer una disposición operativa para generar direcciones para la puesta en correspondencia y un generador de direcciones para poner en práctica esta puesta en correspondencia que se da a conocer en la solicitud de patente europea 04251667.4. El intercalador de direcciones comprende un registro de desplazamiento de realimentación lineal que utilizable para generar una secuencia de bits pseudo-aleatoria y un circuito de permutación. El circuito de permutación permuta el orden del contenido del registro de desplazamiento de realimentación lineal con el fin de generar una dirección. La dirección proporciona una indicación de una posición de memoria de la memoria de intercalador para la escritura del símbolo de datos de entrada o la lectura del símbolo de datos de entrada desde la memoria de intercalador para la puesta en correspondencia con una de las señales de sub-portadoras del símbolo OFDM. De modo similar, un generador de direcciones, en el receptor, está dispuesto para generar direcciones de la memoria del intercalador para la escritura de los símbolos de datos recibidos o la lectura de los símbolos de datos desde la memoria del intercalador para formar un flujo de símbolos de salida.

35

40

45

En un artículo titulado: "Un demapeador reconfigurable nuevo y de alta velocidad-Arquitectura de-intercalador de símbolos de dispositivos de puesta en correspondencia para DVB-T", por Howarth L. et al, publicado en ISCAS '99 en Proceedings de IEEE International Symposium, sobre circuitos y sistemas páginas 382-385, 30 de mayo de 1999, se ha dado a conocer un transmisor adecuado para la norma DVB-T que está dispuesto para poner en correspondencia palabras de bits de datos desde una fuente de entrada en símbolos de modulación y para intercalar los símbolos de modulación en las sub-portadoras de símbolos OFDM. El intercalador incluye una memoria de intercalador para la escritura de los símbolos de modulación desde el flujo de entrada de la fuente en la memoria de intercalador antes de la puesta en correspondencia de los símbolos de modulación desde la memoria de intercalador en las sub-portadoras de los símbolos OFDM. Asimismo, se da a conocer un receptor que incluye un de-intercalador de símbolos para la puesta en correspondencia de los símbolos de modulación recibidos desde los símbolos OFDM en un flujo de símbolos de salida, mediante la lectura de los símbolos de modulación y la salida de una memoria de intercalador.

50

55

En conformidad con otro desarrollo de la norma de Difusión Vídeo Digital-Terrestre, conocida como DVB-T2, existe un deseo de mejorar la comunicación de bits de datos y más en particular, proporcionar una disposición mejorada para el intercalado de bits de datos codificados con código LDPC y símbolos de datos en las señales de sub-portadoras de símbolos OFDM.

60

SUMARIO DE LA INVENCION

65

5 En conformidad con la presente invención se da a conocer un aparato de procesamiento de datos dispuesto en operación para recuperar bits de datos a partir de los símbolos de datos recibidos desde un número predeterminado de señales sub-portadoras de símbolos Multiplexados por División de Frecuencias Ortogonales (OFDM) y para formar un flujo de bits de salida. El número predeterminado de señales sub-portadoras de los símbolos OFDM se determina en conformidad con uno de entre una pluralidad de modos de funcionamiento. El aparato de procesamiento de datos incluye un desintercalador de símbolos, una unidad demapeadora, un permutador inverso y un decodificador de Control de Paridad de Baja Densidad (LDPC).

10 El desintercalador de símbolos está dispuesto en operación para recuperar primeros conjuntos de símbolos de datos a partir de los primeros símbolos OFDM en conformidad con un proceso de intercalado impar y de segundos conjuntos de símbolos de datos a partir de segundos símbolos de OFDM en conformidad con un proceso de intercalado par y para formar un flujo de símbolo de salida a partir de los primeros y segundos conjuntos de símbolos de datos.

15 La unidad demapeadora está dispuesta en operación para generar, a partir de los símbolos de datos del flujo de símbolos de salida, los bits de datos de LDPC, con paridad intercalada, codificados, convirtiendo cada uno de los símbolos de datos del flujo de símbolos de salida, que representan un símbolo de modulación de las señales sub-portadoras de OFDM, en bits de datos en conformidad con un sistema de modulación.

20 El permutador inverso está adaptado para realizar un proceso de permutación inversa para efectuar una inversión de un proceso de permutación aplicado a los bits de datos codificados de LDPC, con paridad intercalada, para permutar los bits de datos de LDPC codificados, de modo que una pluralidad de los bits de datos de LDPC codificados, que correspondan a un valor de 1 en una fila arbitraria de una matriz de información correspondiente a bits de información de un código de LDPC, que fue utilizado para codificar los bits de datos, no se incorpore en el mismo símbolo.

25 El decodificador de LDPC está adaptado para realizar la decodificación de LDPC sobre los LDPC bits de datos de LDPC codificados, en donde se haya realizado el proceso de permutación inversa para formar los bits de datos de salida. El proceso de intercalado impar del desintercalador incluye la escritura de los primeros conjuntos de símbolos de datos recuperados a partir de las señales sub-portadoras de los primeros símbolos OFDM en una memoria del intercalador de símbolos, en función de un orden definido por un código de permutación y

30 lectura de los primeros conjuntos de símbolos de datos desde la memoria del intercalador de símbolos, en función de un orden secuencial en el flujo de datos de salida. El proceso de intercalado par incluye

35 la escritura de los segundos conjuntos de símbolos de datos recuperados a partir de las señales subportadoras de los segundos símbolos OFDM, en la memoria del intercalador de símbolos, en función de un orden secuencial y

40 la lectura de los segundos conjuntos de símbolos de datos desde la memoria del intercalador de símbolos en función de un orden definido por el código de permutación en el flujo de datos de salida, en conformidad con un orden secuencial de modo que mientras los símbolos de datos, procedentes del primer conjunto, son objeto de lectura desde posiciones en la memoria de intercalador de símbolos, los símbolos de datos procedentes del segundo conjunto pueden ser objeto de escritura para las posiciones que se acaban de leer y cuando los símbolos de datos, desde el segundo conjunto, son objeto de lectura desde las posiciones en la memoria de intercalador de símbolos, los símbolos de datos, procedentes de un primer conjunto siguiente, pueden ser objeto de escritura en las posiciones que se acaban de leer. El aparato de procesamiento de datos está diseñado de modo que, cuando en uno de entre la pluralidad de modos de funcionamiento,

45 en donde el número de sub-portadoras por símbolo OFDM proporciona la mitad o menos de la mitad de un número máximo de sub-portadoras en los símbolos OFDM de cualquiera de los modos de funcionamiento, el aparato de procesamiento de datos es utilizable para intercalar los símbolos de datos en conformidad con el proceso de intercalado impar.

50 Formas de realización de la presente invención pueden dar a conocer un aparato de procesamiento de datos para recibir bits de datos a partir de símbolos OFDM en donde el intercalado/desintercalado de bits se combina con el intercalado/desintercalado de símbolos para mejorar el rendimiento de un sistema de comunicación de OFDM, que utiliza la codificación de corrección de errores del Control de Paridad de Baja Densidad (LDPC). El intercalado/desintercalado de bits incluye la permutación/permutación inversa al efecto de que, cuando dos o más bits de código de un código de LDPC se transmitan/reciban como un símbolo, un proceso de permutación que permuta los bits de código del código de LDPC, de modo que una pluralidad de bits de código que corresponda a un valor de 1 en una fila arbitraria de una matriz de información, correspondiente a bits de información del código de LDPC, no se incorpore en el mismo símbolo y luego, se recupere por la permutación inversa.

55

60 Los primeros símbolos OFDM pueden ser símbolos OFDM impares y los segundos símbolos OFDM pueden ser símbolos OFDM pares.

El aparato de procesamiento de datos puede ser un dispositivo independiente y puede ser también un bloque interno incluido en un dispositivo, tal como un receptor.

65

Los códigos LDPC pueden proporcionar un alto rendimiento de corrección de errores en rutas de comunicación, que no sean los canales de Ruido Gaussiano Blanco Aditivo, que es superior a los códigos convolucionales o códigos convolucionales de Reed Solomon (RS) concatenados. Esto puede proporcionarse en canales de comunicación que presentan ráfagas de errores, que causan borrados de información. Por ello, existe una necesidad de dar a conocer un método para aumentar la resistencia a errores de ráfagas o borrados operativos, mientras se mantiene el rendimiento de las rutas de comunicación de AWGN. Formas de realización de la presente invención dan a conocer, por lo tanto, un aparato y método de procesamiento de datos que puede aumentar la resistencia a errores en bits de código de los códigos de LDPC, tales como errores de ráfagas o borrados, combinando un intercalador de bits para los bits de datos de LDPC codificados con un intercalador de símbolos.

Es decir, en conformidad con las formas de realización de la invención, el intercalado/desintercalado de paridad se realiza en un código LDPC obtenido realizando la codificación por LDPC, en función de una matriz de control de paridad, que incluye una matriz de paridad estructurada de forma escalonada, correspondiente a bits de paridad del código LDPC, de modo que los bits de paridad del código LDPC estén intercalados para posiciones de bits de paridad diferentes.

En algunos transmisores y receptores de OFDM convencionales, que funcionan en conformidad con los modos de dos 2k y de 8k para la norma DVB-T y el modo de 4k para la norma DVB-H, se utilizan dos procesos de intercalado de símbolos en el transmisor y en el receptor, uno para símbolos OFDM pares y otro para símbolos OFDM impares. Sin embargo, el análisis correspondiente ha demostrado que los sistemas de intercalado, diseñados para los intercaladores de símbolos de 2k y de 8k para DVB-T y el intercalador de símbolos de 4k para DVB-H, funcionan mejor para símbolos impares que para símbolos pares. Formas de realización de la presente invención están diseñadas de modo que solamente el proceso de intercalado de símbolos impar se utiliza a no ser que el transmisor/receptor esté en el modo del número máximo de sub-portadoras. Por lo tanto, cuando el número de símbolos de datos que pueden transmitirse por las sub-portadoras de un símbolo OFDM, en uno de la pluralidad de modos de funcionamiento, es menor que la mitad del número de símbolos de datos, que se pueden transmitir en un modo de funcionamiento, que demuestra ser el mayor número de datos que soportan señales de sub-portadoras por símbolo OFDM; en tal caso, un intercalador del transmisor y del receptor de los símbolos OFDM está diseñado para intercalar los símbolos de datos de los primeros y segundos conjuntos con la utilización del proceso de intercalado impar. Puesto que el intercalador está realizando el intercalado de los símbolos de datos de, a la vez, los primeros y los segundos conjuntos de símbolos de datos en los símbolos OFDM, utilizando el proceso de intercalado impar, el intercalador utiliza diferentes partes de la memoria de intercalador para la escritura y lectura de los símbolos de datos. En consecuencia, en comparación con la forma de realización, a modo de ejemplo, en donde el intercalador está utilizando el proceso de intercalado impar y el proceso de intercalado par para intercalar los primeros y segundos conjuntos de símbolos de datos en primeros y segundos símbolos OFDM sucesivos, que utiliza la memoria nivel, la magnitud de capacidad de memoria utilizada es dos veces el número de símbolos de datos que pueden transmitirse por un símbolo OFDM para el intercalado solamente impar. Lo anterior se compara con un requisito de memoria de una vez el número de símbolos de datos, que se pueden transmitir en un símbolo OFDM, en el modo con el mayor número de símbolos de datos por símbolo OFDM utilizando, a la vez, el proceso de intercalado impar y el proceso de intercalado par. Sin embargo, el número de sub-portadoras por símbolo OFDM, para este modo operativo máximo, es dos veces la capacidad del siguiente mayor número de sub-portadoras por símbolo OFDM para cualquier otro modo operativo con el siguiente mayor número de sub-portadoras por símbolo OFDM.

Según otras formas de realización, a modo de ejemplo, por lo tanto, un tamaño mínimo de la memoria de intercalador puede proporcionarse en función del número máximo de símbolos de datos que pueden transmitirse sobre las sub-portadoras de los símbolos OFDM, que están disponibles para transmitir los símbolos de datos en cualquiera de los modos de funcionamiento.

En algunas formas de realización, el modo de funcionamiento que proporciona el número máximo de sub-portadoras por símbolo OFDM es un modo de 32K. Los otros modos pueden incluir uno o más de los modos de 2K, 4K, 8K y 16K. Por ello, como se apreciará por la descripción anterior, en el modo de 32K los procesos de intercalado impar y par se utilizan para intercalar los símbolos de datos, de modo que la magnitud de la memoria de intercalador pueda ser suficiente para contabilizar los 32K símbolos de datos. Sin embargo, para el modo de 16K y cualquiera de los demás modos, entonces se utiliza solamente el proceso de intercalado impar, de modo que, con el modo de 16K, se requiere un tamaño de memoria equivalente de 32K símbolos, con el modo de 4k se necesita un tamaño de memoria equivalente de 8k símbolos y con el modo de 2k se requiere un tamaño de memoria equivalente de 4k símbolos.

En algunas formas de realización, a modo de ejemplo, se utiliza un código de permutación diferente para realizar el intercalado para símbolos OFDM sucesivos. El uso de diferentes códigos de permutación para símbolos OFDM sucesivos puede proporcionar una ventaja cuando el aparato de procesamiento de datos es utilizable para intercalar los símbolos de datos a comunicarse por las sub-portadoras de los símbolos OFDM o recibidos desde las señales de sub-portadoras de cada uno de los símbolos OFDM utilizando solamente el proceso de intercalado impar. Por lo tanto, en un transmisor, un aparato de procesamiento de datos es utilizable para intercalar los símbolos de datos de entrada en las señales de sub-portadoras de los símbolos OFDM efectuando la lectura de los símbolos de datos en la memoria en un orden secuencial y efectuando la lectura de los símbolos de datos desde la memoria de intercalador en un orden determinado en función del conjunto de direcciones generadas por el generador de direcciones. En un receptor, un aparato de procesamiento de datos es utilizable para intercalar los símbolos de datos de entrada en las señales de sub-portadoras de los símbolos OFDM efectuando la lectura en memoria de los símbolos de datos recibidos desde las sub-

portadoras de los símbolos OFDM en un orden determinado en función del conjunto de direcciones generadas por el generador de direcciones y la lectura desde la memoria en un flujo de datos de salida en un orden secuencial.

5 Varios aspectos y características de la presente invención se definen en las reivindicaciones adjuntas. Aspectos adicionales de la presente invención incluyen un método de recuperación de bits de datos a partir de un número predeterminado de señales sub-portadoras de símbolos Multiplexados por División de Frecuencias Ortogonales (OFDM) así como un receptor.

10 BREVE DESCRIPCIÓN DE LOS DIBUJOS

Formas de realización de la presente invención se describirán ahora, a modo de ejemplo solamente con referencia a los dibujos adjuntos, en donde los elementos similares están provistos de referencias numéricas correspondientes y en donde:

15 La Figura 1 es un diagrama de bloques esquemático de un transmisor OFDM codificado que puede utilizarse, a modo de ejemplo, con la norma DVB-T2;

La Figura 2 ilustra una matriz de control de paridad H, a modo de ejemplo, de un código LDPC;

20 La Figura 3 es un diagrama de flujo que ilustra un procesamiento para decodificar un código LDPC;

La Figura 4 ilustra una matriz de control de paridad, a modo de ejemplo, de un código LDPC;

25 La Figura 5 ilustra un gráfico de Tanner de una matriz de control de paridad;

La Figura 6 ilustra un nodo variable;

La Figura 7 ilustra un nodo de control;

30 La Figura 8 es un diagrama de bloques esquemático que ilustra una configuración, a modo de ejemplo, de un transmisor;

La Figura 9 ilustra una matriz de control de paridad;

35 La Figura 10 ilustra una matriz de paridad;

Las Figuras 11a y 11b ilustran una matriz de control de paridad de un código LDPC y los pesos de ponderación de columnas definidos en la especificación de DVB-S.2;

40 Las Figuras 12a y 12b ilustran una disposición de puntos de señales de 16QAM;

La Figura 13 ilustra una disposición de puntos de señales de 64QAM;

La Figura 14 ilustra una disposición de puntos de señales de 64QAM;

45 La Figura 15 ilustra una disposición de puntos de señales de 64QAM;

Las Figuras 16a a 16d ilustran el funcionamiento de un demultiplexor 25;

50 Las Figuras 17a a 17d ilustran el funcionamiento del demultiplexor 25;

La Figura 18 ilustra un gráfico de Tanner para la decodificación de un código LDPC;

55 Las Figuras 19a y 19b ilustran una matriz de paridad H_T que presenta una estructura escalonada y un gráfico de Tanner correspondiente a la matriz de paridad H_T ;

La Figura 20 ilustra una matriz de paridad H_T de una matriz de control de paridad H correspondiente a un código LDPC después de que se realice una operación de intercalado de paridad en el código LDPC;

60 Las Figuras 21a y 21b ilustran una matriz de control de paridad convertida;

La Figura 22 ilustra el funcionamiento de un intercalador de columna con torsión 24;

65 La Figura 23 ilustra el número de columnas de una memoria 31 que se requiere para el intercalado de columnas con torsión y las direcciones de posiciones iniciales de escritura;

- La Figura 24 ilustra el número de columnas de la memoria 31 necesarias para el intercalado de columnas con torsión y las direcciones de las posiciones de inicio de escritura;
- 5 La Figura 25 es un diagrama de flujo que ilustra un procedimiento de transmisión;
Las Figuras 26a y 26b ilustran un modelo de una ruta de comunicación utilizada en simulaciones;
- La Figura 27 ilustra las relaciones entre las frecuencias Doppler f_d y las tasas de errores obtenidas a partir de simulaciones;
- 10 La Figura 28 ilustra las relaciones entre las frecuencias Doppler f_d y las tasas de errores obtenidas a partir de simulaciones;
- La Figura 29 es un diagrama de bloques esquemático de un receptor de OFDM codificado que puede utilizarse, a modo de ejemplo, con la norma DVB-T2;
- 15 La Figura 30 es un diagrama de flujo que ilustra un procedimiento de recepción;
- La Figura 31 ilustra una matriz de control de paridad, a modo de ejemplo, de un código LDPC;
- 20 La Figura 32 ilustra una matriz (matriz de control de paridad convertida) obtenida realizando la permutación de filas y la permutación de columnas en la matriz de control de paridad;
- La Figura 33 ilustra la matriz de control de paridad convertida dividida en unidades de matrices 5 x 5;
- 25 La Figura 34 es un diagrama de bloques que ilustra una configuración, a modo de ejemplo, de un dispositivo de decodificación que realiza cálculos de nodos P en paralelo;
- La Figura 35 representa una configuración, a modo de ejemplo, de un decodificador LDPC 56;
- 30 La Figura 36 es un diagrama de bloques que ilustra una configuración, a modo de ejemplo, de una forma de realización de un ordenador al que se aplica la invención;
- La Figura 37 es un diagrama de bloques esquemático de componentes del transmisor ilustrado en la Figura 1 en donde un dispositivo de mapeado de puesta en correspondencia de símbolos y un constructor de tramas ilustra el funcionamiento de un intercalador;
- 35 La Figura 38 es un diagrama de bloques esquemático del intercalador de símbolos representado en la Figura 37;
- La Figura 39 es un diagrama de bloques esquemático de una memoria de intercalador representada en la Figura 38 y el correspondiente de-intercalador de símbolos en el receptor;
- 40 La Figura 40 es un diagrama de bloques esquemático de un generador de direcciones ilustrado en la Figura 38 para el modo de 32k;
- 45 La Figura 41(a) es un diagrama que ilustra los resultados para un intercalador que utiliza el generador de direcciones representado en la Figura 40 para símbolos pares y la Figura 41(b) es un diagrama que ilustra los resultados de la simulación de diseño para símbolos impares, mientras que la Figura 41(c) es un diagrama que ilustra resultados comparativos para un generador de direcciones que utiliza un código de permutación diferente para símbolos pares y la Figura 41 (d) es un diagrama correspondiente para símbolos impares;
- 50 La Figura 42 es un diagrama de bloques esquemático de un de-intercalador de símbolos que se representa en la Figura 29;
- La Figura 43(a) es un diagrama que ilustra los resultados para un intercalador que utiliza el generador de direcciones representado en la Figura 40 para símbolos OFDM pares y la Figura 43(b) es un diagrama que ilustra resultados para símbolos OFDM impares. Las Figuras 44(a) y 44(b) representan trazados de la distancia, a la salida del intercalador de sub-portadoras, que eran adyacentes a la entrada del intercalador;
- 55 La Figura 44 da a conocer un diagrama de bloques esquemático del intercalador de símbolos representado en la Figura 38, que ilustra un modo de funcionamiento en el que se realiza el intercalado en conformidad con un modo de intercalado impar solamente y
- 60 La Figura 45 representa un diagrama de bloques esquemático del de-intercalador de símbolos representado en la Figura 42, que ilustra el modo de funcionamiento en el que la operación de intercalado se realiza en conformidad con el modo de intercalado impar solamente.
- 65

DESCRIPCIÓN DE FORMAS DE REALIZACIÓN PREFERIDAS

La Figura 1 representa un diagrama de bloques, a modo de ejemplo, de un transmisor OFDM que puede utilizarse, también a modo de ejemplo, para transmitir imágenes de vídeo y señales de audio en conformidad con la norma DVB-T2. En la Figura 1, una fuente de programas genera datos a transmitirse por el transmisor OFDM. Un codificador de vídeo 2, un codificador de audio 4 y un codificador de datos 6 generan imágenes de vídeo, señales de audio y otros datos a transmitirse, que se alimentan operativamente a un multiplexor de programas 10. La salida del multiplexor de programas 10 forma un flujo multiplexado con otra información requerida para comunicar las imágenes de vídeo, las señales de audio y otros datos. El multiplexor 10 proporciona un flujo continuo en un canal de conexión 13. Pueden existir bastantes flujos continuos multiplexados que se alimentan en diferentes ramas de derivación A, B, etc. Para mayor simplicidad, solamente se describirá la rama de derivación A.

Según se ilustra en la Figura 1, un transmisor OFDM 11 recibe el flujo continuo en un bloque de dispersión de energía y adaptación del multiplexor 20. El bloque de dispersión de energía y adaptación del multiplexor 20 aleatoriza los datos y alimenta los datos adecuados a un codificador de corrección de errores en sentido directo 21 que realiza la codificación de corrección de errores del flujo continuo. Un intercalador de bits 22 está provisto para intercalar los bits de datos codificados que, a modo de ejemplo, de la norma DVB-T2 es la salida del codificador LDPC. La salida desde el intercalador de bits 22 se alimenta a un dispositivo de puesta en correspondencia de constelación 26, que pone en correspondencia grupos de bits en un punto de constelación, que ha de utilizarse para transportar los bits de datos codificados. Las salidas desde el bit en el dispositivo de puesta en correspondencia de constelación 26 son etiquetas de puntos de constelación que representan componentes reales e imaginarios. Las etiquetas de puntos de constelación representan símbolos de datos formados a partir de dos o más bits, dependiendo del sistema de modulación utilizado. Estas circunstancias operativas se referirán como celdas de datos. Estas celdas de datos se hacen pasar a través de un intercalador temporal 30 cuyo efecto es intercalar celdas de datos resultantes de múltiples palabras de código LDPC. Las celdas de datos desde el intercalador temporal 30 se alimentan, luego, a un constructor de tramas y modulación 27, que establece la puesta en correspondencia de las celdas de datos con los símbolos de modulación para transmisión.

Las celdas de datos se reciben dentro de la unidad de modulación 27 por un constructor de tramas 32, con celdas de datos generadas por la rama de derivación B, etc., en la Figura 1, por intermedio de otros canales 31. El constructor de tramas 32 forma entonces, numerosas celdas de datos en secuencias a transmitirse en símbolos OFDM, en donde un símbolo OFDM comprende varias celdas de datos, estando cada celda de datos en correspondencia con una de las sub-portadoras. El número de sub-portadoras dependerá del modo de funcionamiento del sistema, que puede incluir un modo de 1k, 2k, 4k, 8k, 16k o 32k, cada uno de los cuales proporciona un número diferente de sub-portadoras en conformidad, a modo de ejemplo, con la tabla siguiente:

Modo	Sub-portadoras
1K	756
2K	1512
4K	3024
8K	6048
16K	12096
32K	24192

Número de sub-portadoras adaptadas desde la norma DVB-T/H

En consecuencia, a modo de ejemplo, el número de sub-portadoras para el modo de 32k es veinticuatro mil ciento noventa y dos. Para el sistema DVB-T2, el número de sub-portadoras por símbolo OFDM puede variar dependiendo del número de portadoras piloto u otras portadoras reservadas. Por ello, en DVB-T2, a diferencia de DVB-T, el número de sub-portadoras para transmitir datos no es fijo. Los dispositivos de difusión pueden seleccionar uno de los modos de funcionamiento desde los modos de 1k, 2k, 4k, 8k, 16k, 32k, proporcionando cada uno de ellos un margen de sub-portadoras para datos por símbolo OFDM, siendo el máximo disponible para cada uno de estos modos 1024, 2048, 4096, 8192, 16384, 32768, respectivamente. En DVB-T2, una trama de capa física está constituida por numerosos símbolos OFDM. En condiciones normales, la trama se inicia con uno o más preámbulos o símbolos OFDM P2, que luego van seguidos por una carga útil numérica que transmite símbolos OFDM. El final de la trama de capa física está marcado por símbolos de cierre de trama. Para cada modo de funcionamiento, el número de sub-portadoras puede ser diferente para cada tipo de símbolo. Además, esta circunstancia operativa puede variar para cada uno, en función de si se selecciona una extensión de ancho de banda, si se permite una reserva de tonos y en función de que se haya seleccionado una configuración de sub-portadoras piloto. En consecuencia, resulta difícil una generalización para un número específico de sub-portadoras por símbolo OFDM. Sin embargo, el intercalador de frecuencias para cada modo puede intercalar cualquier símbolo cuyo número de sub-portadoras sea más pequeño que, o el mismo que, el número disponible máximo de sub-portadoras para el modo dado. A modo de ejemplo, en el modo de 1k, el intercalador funcionaría para símbolos con el número de sub-portadoras siendo menor que o igual a 1024 y para el modo de 16k, con el número de sub-portadoras siendo menor o igual a 16384.

La secuencia de celdas de datos a transmitirse en cada símbolo OFDM se transmite luego al intercalador de símbolos 33. A continuación, se genera el símbolo OFDM por un bloque constructor de símbolos OFDM 37 que introduce señales piloto y de sincronización, alimentadas desde un conformador de señales piloto e incorporadas 36. Un modulador de OFDM 38 forma luego el símbolo OFDM en el dominio del tiempo, que se alimenta a un procesador de inserción de guarda 40 para generar un intervalo de guarda entre símbolos y luego, a un convertidor digital a analógico 42 y por último, a un amplificador de RF dentro de un extremo frontal de RF 44 para su posible difusión por el transmisor OFDM desde una antena 46.

Formas de realización de la presente invención dan a conocer un sistema de comunicación OFDM que incluye un intercalador de bits para intercalar bits codificados con un codificador LDPC en combinación con un intercalador de símbolos, que intercala símbolos que representan los uno o más bits intercalados y codificados en las señales subportadoras de un símbolo OFDM. El intercalador de bits y el intercalador de símbolos, según formas de realización, a modo de ejemplo, se describen en los apartados siguientes, comenzando con el intercalador de bits, que se describe con codificación por LDPC:

Intercalador de bits para codificación por LDPC

Código de corrección de errores de LDPC

Los códigos de LDPC presentan un alto rendimiento en corrección de errores y comenzaron recientemente a utilizarse en sistemas de comunicación, que incluyen la difusión digital vía satélite, tal como DVB-S.2, que ha introducido su uso en Europa (a modo de ejemplo, véase DVB-S.2: ETSI EN 302 307 V1.1.2 (2006-06)). Asimismo, se somete a consideración la aplicación de códigos LDPC a la difusión digital terrestre de la siguiente generación.

Recientes estudios demuestran que el rendimiento de los códigos LDPC se aproximan al denominado Límite de Shannon cuando se aumenta la longitud del codificación, de forma similar a los denominados códigos turbo. Puesto que los códigos LDPC tienen una propiedad de que la distancia mínima es proporcional a la longitud del código, los códigos LDPC tienen ventajas por cuanto que las características de probabilidad de errores en bloques son excelentes y tienen un límite inferior de errores, que es un fenómeno observado en asociación con las características de decodificación de códigos turbo o similares, lo que ocurre en raras ocasiones.

Se hará ahora referencia, en detalle, a dichos códigos LDPC. Los códigos LDPC con códigos lineales. Aunque los códigos LDPC no son necesariamente binarios, la siguiente descripción se proporcionará haciendo referencia a los códigos LDPC binarios.

La característica más importante de los códigos LDPC es que una matriz de control de paridad, que define cada código LDPC, es una matriz que tiene un muy pequeño número de elementos de "1", esto es, sus elementos son principalmente "0".

La Figura 2 ilustra una matriz de control de paridad H, a modo de ejemplo, de un código LDPC.

Cada columna de la matriz de control de paridad H de la Figura 2 tiene un peso de ponderación de 3 (esto es, 3 elementos de "1") y cada fila tiene un peso de ponderación de 6 (esto es, 6 elementos de "1").

Se realiza una codificación basada en códigos LDPC (esto es, codificación por LDPC), a modo de ejemplo, calculando una matriz de generación G basada en una matriz de control de paridad H y multiplicando la matriz de generación G por los bits de información para generar una palabra de código (código LDPC).

Más concretamente, un codificador LDPC calcula primero una matriz de generación G que satisface una ecuación $GH^T = 0$, con una matriz transpuesta H^T de la matriz de control de paridad H. En este caso, cuando la matriz de generación G es una matriz $K \times N$, el codificador multiplica la matriz de generación G por una secuencia de bits de información de K-bits (vector u) para generar una palabra de código de N-bits c ($= uG$). La palabra de código (código LDPC) generada por el codificador se recibe por un lado receptor a través de una ruta de comunicación.

El código LDPC puede decodificarse mediante un algoritmo de paso de mensajes propuesto por Gallager y denominado como un "algoritmo de decodificación probabilística". El algoritmo de paso de mensajes utiliza una, así denominada, propagación de creencia en un gráfico de Tanner incluyendo nodos de variables (también referidos como nodos de mensajes) y nodos de control. En la siguiente descripción, cada uno de los nodos de variables y de los nodos de control se referirá simplemente como un "nodo" cuando sea adecuado.

La Figura 3 ilustra un procedimiento para decodificar un código LDPC.

A continuación, un valor real que expresa, como una relación de probabilidad logarítmica, la probabilidad de que un i-ésimo bit de código de un código LDPC (una palabra de código) recibido por un lado receptor tenga un valor de "0" se refiere como un valor recibido u_{0i} cuando sea adecuado. Además, una salida de mensaje desde un nodo de control se refiere como u_j y una salida de mensaje desde un nodo de variable se refiere como v_i .

Un código LDPC se decodifica en el modo siguiente. En primer lugar, según se ilustra en la Figura 3, en la etapa S11, se recibe un código LDPC, un mensaje (mensaje de nodo de control) u_j se inicializa a "0" y una variable k , que tiene un valor de número entero como un contador de un proceso iterativo, se inicializa a "0". A continuación, el procedimiento prosigue con la etapa S12. En la etapa S12, un cálculo (cálculo de nodo variable) representado por la ecuación (1) se realiza sobre la base de un valor recibido u_{0i} obtenido mediante la recepción del código LDPC para obtener un mensaje (mensaje de nodo de variable) v_i y un cálculo (cálculo de nodo de control) representado por la ecuación (2) se realiza luego sobre la base del mensaje v_i para obtener un mensaje u_j .

ECUACIÓN 1

$$v_i = u_{0i} + \sum_{j=1}^{d_v-1} u_j \quad \dots (1)$$

ECUACIÓN 2

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

d_v y d_c en la ecuación (1) y en la ecuación (2) son parámetros arbitrariamente seleccionables que representan los números respectivos de 1 en una dirección vertical (columna) y en una dirección horizontal (fila) de la matriz de control de paridad H. A modo de ejemplo, $d_v = 3$ y $d_c = 6$ en el caso de un código (3,6).

Los márgenes respectivos para el cálculo en el cálculo de nodo de variable de la ecuación (1) y el cálculo de nodo de control de la ecuación (2) son desde d_v-1 y desde 1 a d_c-1 , puesto que un mensaje recibido desde un borde (esto es, una línea que conecta un nodo de variable y un nodo de control entre sí) que proporciona a la salida el mensaje se excluye de cada uno de los cálculos de las ecuaciones (1) y (2). En realidad, el cálculo del nodo de control de la ecuación (2) se realiza mediante utilización recursiva, según se ilustra en la ecuación (4), una tabla anteriormente creada de una función $R(v_1, v_2)$, representada por la ecuación (3), que se define como una sola salida con respecto a dos entradas v_1 y v_2 .

ECUACIÓN 3

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

ECUACIÓN 4

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

En la etapa S12, la variable k se incrementa en "1" y el procedimiento prosigue con la etapa S13. En la etapa S13, se determina si la variable k es, o no, mayor que un número predeterminado de iteraciones de decodificación C . Si se determina, en la etapa S13, que la variable k no es mayor que C , el procedimiento retorna a la etapa S12 para repetir el mismo proceso.

Si se determina, en la etapa S13, que la variable k es mayor que C , el procedimiento prosigue con la etapa S14 para realizar un cálculo representado por la ecuación (5) para obtener y proporcionar a la salida un mensaje v_i como un resultado de decodificación final. Entonces, se termina el procedimiento de decodificación de código LDPC.

ECUACIÓN 5

$$v_i = u_{0i} + \sum_{j=1}^{d_v} u_j \quad \dots (5)$$

En este caso, a diferencia del cálculo del nodo de variable de la ecuación (1), el cálculo de la ecuación (5) se realiza utilizando mensajes u_j desde todos los bordes conectados al nodo de variable.

5 La Figura 4 ilustra una matriz de control de paridad H, a modo de ejemplo, de un código LDPC (3,6) con una tasa de código de 1/2 y una longitud de código de 12.

Como en el caso de la matriz de control de paridad H de la Figura 2, la matriz de control de paridad H de la Figura 4 tiene un peso de ponderación de columna de 3 y un peso de ponderación de fila de 6.

10 La Figura 5 ilustra un gráfico de Tanner de la matriz de control de paridad H de la Figura 4.

En la Figura 5, "+" representa un nodo de control y "=" representa un nodo de variable. Los nodos de control y los nodos de variable corresponden a filas y columnas de la matriz de control de paridad H, respectivamente. Cada línea de conexión entre un par de nodos de control y de variable es un borde correspondiente a un elemento de "1" de la matriz de control de paridad H.

15 Más concretamente, cuando un elemento de la j-ésima fila y de la i-ésima columna de una matriz de control de paridad es "1", un i-ésimo nodo de variable "=" (conteo desde la parte superior) y j-ésimo nodo de control "+" (conteo desde la parte superior) se conectan a través de un borde en la Figura 5. El borde indica que un bit de código correspondiente al nodo de variable tiene una limitación correspondiente al nodo de control.

20 Un algoritmo de producto de sumas, que es un algoritmo de decodificación de códigos LDPC, realiza repetidamente un cálculo de nodo de variable y un cálculo de nodo de control.

25 La Figura 6 ilustra un cálculo de nodo de variable realizado en un nodo de variable.

Un mensaje v_i correspondiente a un borde para cálculo se obtiene según el cálculo de nodo de variable de la ecuación (1) utilizando un valor recibido u_{oi} y mensajes u_1 y u_2 desde los restantes bordes conectados al nodo de variable. Los mensajes correspondientes a otros bordes se obtienen de la misma manera.

30 La Figura 7 ilustra un cálculo de nodo de control realizado en un nodo de control.

La ecuación (2) anterior para el cálculo de nodo de control se puede describir como la ecuación (6) utilizando una ecuación de relación de $axb = \exp\{\ln(|a|) + \ln(|b|)\} \times \text{sign}(a) \times \text{sign}(b)$, en donde $\text{sign}(x)$ es 1 cuando $x \geq 0$ y -1 cuando $x < 0$.

35 ECUACIÓN 6

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

40 Además, cuando una función $\phi(x) = \ln(\tanh(x/2))$ se define cuando $x \geq 0$, se satisface una ecuación $\phi^{-1}(x) = 2 \tanh^{-1}(e^{-x})$ y por lo tanto, la ecuación (6) se puede reorganizar en la ecuación (7).

ECUACIÓN 7

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

45 En el nodo de control, el cálculo del nodo de control de la ecuación (2) se realiza aplicando la ecuación (7).

Es decir, en el nodo de control, un mensaje u_j correspondiente a un borde para el cálculo se obtiene en función del cálculo del nodo de control de la ecuación (7) utilizando los mensajes v_1, v_2, v_3, v_4 y v_5 a partir de los restantes bordes conectados al nodo de control según se ilustra en la Figura 7. Los mensajes correspondientes a otros bordes se obtienen de la misma manera.

La función $\phi(x)$ en la ecuación (7) pueden expresarse también de modo que $\phi(x)=\ln((e^x + 1)/(e^x - 1))$ y $\phi(x)=\phi^{-1}(x)$ cuando $x>0$. Cuando las funciones $\phi^{-1}(x)$ y $\phi(x)$ están incorporadas en el hardware, se pueden incorporar utilizando la misma Tabla de Consulta (LUT).

Aunque es conocido que los códigos LDPC presentan muy alto rendimiento en la ruta de comunicación de Ruido Gaussiano Blanco Aditivo (AWGN), es también conocido en los últimos años que los códigos LDPC tienen un alto rendimiento en corrección de errores en otras rutas de comunicación, en comparación con los códigos convolucionales o códigos de Reed Solomon (RS) convolucionales concatenados con anterioridad en el tiempo.

Es decir, cuando se selecciona un código que tiene un excelente rendimiento en una ruta de comunicación de AWGN, el código seleccionado suele presentar un excelente rendimiento en otras rutas de comunicación, superior a otros códigos.

A modo de ejemplo, cuando se aplican códigos LDPC a la difusión digital terrestre, se ha recomendado que los códigos LDPC, definidos en la especificación DVB-S.2 y los sistemas de modulación definidos en la especificación DVB-T se combinen y un intercalador de bits, que intercala bits de código de un código LDPC, se proporcione entre un codificador LDPC y un modulador para mejorar el rendimiento de códigos LDPC en rutas de comunicación AWGN.

Sin embargo, pueden producirse borrados operativos o errores de ráfagas en las rutas de comunicación que se asumen como siendo ondas de componente terrestre. A modo de ejemplo, en un sistema de Multiplexación por División de Frecuencias Ortogonales (OFDM), se puede borrar un símbolo específico (esto es, caer a cero en potencia) debido al retardo de un eco, que es una ruta distinta a la ruta principal, en entornos multi-ruta, en donde una relación de Deseado a Indeseado (D/U) es 0 dB, de modo que la potencia de la ruta principal, como potencia deseada, es igual a la potencia de eco como potencia indeseada.

Cuando la relación D/U es 0dB, todos los símbolos OFDM, en un tiempo concreto, pueden borrarse también (esto es, caída a cero en potencia) debido a una frecuencia Doppler en un dispositivo de fluctuación que es una ruta de comunicación a la que se aplica un eco con un frecuencia Doppler y que tiene un retardo de "0" que se ha añadido.

Además, pueden ocurrir errores de ráfagas debido a potencia inestable o condiciones indeseables de cableados desde antenas a receptores.

En la técnica relacionada, códigos correctores de errores que presentan un excelente rendimiento en rutas de comunicación AWGN, se suelen utilizar frecuentemente en rutas de comunicación en donde se producen borrados operativos o errores de ráfagas según se describió anteriormente.

Por otro lado, cuando se decodifica un código LDPC, un nodo de variable correspondiente no solamente a una columna de una matriz de control de paridad H, sino también a un bit de código del código LDPC se calcula aplicando la ecuación (1) que implica la adición de un bit de código (un valor recibido u_{0i}) de un código LDPC, según se ilustra en la Figura 6. Por lo tanto, la exactitud del mensaje obtenido se reduce si ocurre un error en un bit de código utilizado en el cálculo del nodo de variable.

Además, cuando se decodifica un código LDPC, se calcula un nodo de control aplicando la ecuación (7) utilizando un mensaje obtenido en un nodo de variable conectado al nodo de control. Por lo tanto, el rendimiento de la decodificación se reduce si un error que incluye un borrado operativo ocurre simultáneamente en (una pluralidad de bits de código del código LDPC correspondiente) una pluralidad de nodos de variables conectados a cada uno de un gran número de nodos de control.

Más concretamente, a modo de ejemplo, cuando dos o más nodos de variables conectados a un nodo de control se suprimen simultáneamente, el nodo de control reenvía un mensaje con una probabilidad de "0" que es igual a la probabilidad "1" a cada nodo de variable conectado al nodo de control. En este caso, el nodo de control que reenvía el mensaje con probabilidades iguales de "0" y "1", no contribuye a un proceso de decodificación que es un conjunto de un cálculo de nodos de variable y un cálculo de nodos de control. Esta circunstancia operativa aumenta el número de procesos de decodificación requeridos, con lo que se disminuye el rendimiento de decodificación y se aumenta el consumo de energía de un receptor que realiza la decodificación de códigos LDPC.

En consecuencia, existe una necesidad de dar a conocer un método para aumentar la resistencia a los errores de ráfagas o borrados operativos, al mismo tiempo que se mantiene el rendimiento de las rutas de comunicación de AWGN.

En este caso, será posible aumentar el rendimiento de decodificación si un intercalador de bits, que intercala bits de código de un código LDPC, se proporciona entre un codificador LDPC y un modulador para mejorar el rendimiento de

código de LDPC en rutas de comunicación de AWGN según se describió anteriormente y si el intercalador de bits está diseñado de modo que pueda realizar la función de intercalado para reducir la probabilidad de que un error ocurriera simultáneamente en (una pluralidad de bits de código de un código LDPC correspondiente) una pluralidad de nodos de variable conectados a un nodo de control.

5 La invención se ha realizado considerando las circunstancias anteriores y proporciona un aparato de procesamiento de datos y un método que puede aumentar la resistencia a errores en bits de código de códigos LDPC, tales como errores de ráfagas o borrados operativos.

10 Un aparato de procesamiento de datos, para intercalar datos según una forma de realización de la invención, incluye un intercalador de paridad para realizar el intercalado de paridad en un código de Control de Paridad de Baja Densidad (LDPC) obtenido realizando la codificación de LDPC en función de una matriz de control de paridad, que incluye una matriz de paridad correspondiente a los bits de paridad del código de LDPC, presentando la matriz de paridad una estructura escalonada, de modo que un bit de paridad del código LDPC se intercale a una posición de bit de paridad diferente.

15 Un método de procesamiento de datos, para un aparato de procesamiento de datos que intercala datos según una forma de realización de la invención, incluye la etapa de hacer que el aparato de procesamiento de datos realice un intercalado de paridad en un código de Control de Paridad de Baja Densidad (LDPC) obtenido realizando la codificación de LDPC en función de una matriz de control de paridad, que incluye una matriz de paridad correspondiente a bits de paridad del código LDPC, presentando la matriz de paridad una estructura escalonada, de modo que un bit de paridad del código LDPC se intercale a una posición de bit de paridad diferente.

20 Es decir, según las formas de realización de la invención, el intercalado de paridad se realiza sobre un código LDPC obtenido realizando la codificación de LDPC en función de una matriz de control de paridad, que incluye una matriz de paridad de estructura escalonada correspondiente a bits de paridad del código LDPC, de modo que los bits de paridad del código LDPC se intercalen a diferentes posiciones de bits de paridad.

25 El aparato de procesamiento de datos puede ser un dispositivo independiente y puede ser también un bloque interno incluido en un dispositivo.

Explicación detallada, a modo de ejemplo, de un intercalador de bits

35 La Figura 8 ilustra una representación más detallada de partes del transmisor representado en la Figura 1, que sirve para describir el funcionamiento del intercalador de bits. En particular, se describirá ahora el codificador LDPC 21. El codificador LDPC 21 codifica los datos objetivos en bits de datos codificados por LDPC que incluyen bits de información correspondientes a los datos objetivos en función de una matriz de control de paridad, en donde una matriz de paridad correspondiente a los bits de paridad del código LDPC tiene una estructura escalonada.

40 Más concretamente, el codificador LDPC 21 codifica datos objetivos en un código LDPC definido, a modo de ejemplo, en conformidad con la especificación DVB-S.2 y proporciona, a la salida, el código LDPC.

45 El código LDPC definido en la especificación DVB-S.2 es un código de Acumulación de Repetición Regular (IRA) y una matriz de paridad en una matriz de control de paridad del código LDPC presenta una estructura escalonada. Detalles de la matriz de paridad y de su estructura escalonada se describirán a continuación. A modo de ejemplo, el código IRA se describe en el documento "Códigos de Repetición-Acumulación Irregular", H. Jin, A. Khandekar y R. J. McEliece, en Proceedings of 2 International Symposium sobre códigos turbo y temas relacionados, páginas 1-8, septiembre 2000.

50 La salida de código LDPC, desde el codificador LDPC 21, se proporciona al intercalador de bits 22.

El intercalador de bits 22 es un aparato de procesamiento de datos que intercala datos e incluye un intercalador de paridad 23, un intercalador de columnas con torsión 24 y un demultiplexor 25.

55 El intercalador de paridad 23 realiza una operación de intercalado de paridad en el código LDPC desde el codificador LDPC 21 para intercalar bits de paridad del código LDPC a diferentes posiciones de bits de paridad y proporciona el código LDPC, intercalado en paridad, al intercalador de columnas con torsión 24.

60 El intercalador de columnas con torsión 24 realiza la operación de intercalado de columnas con torsión en el código LDPC desde el intercalador de paridad 23 y luego, proporciona el código LDPC con intercalado de columnas con torsión al demultiplexor 25.

De este modo, el código LDPC se transmite después que de dos o más bits de código del código LDPC se pongan en correspondencia con un símbolo modulado de forma ortogonal mediante la unidad de puesta en correspondencia 26 descrita a continuación.

65

- 5 El intercalador de columnas con torsión 24 realiza la permutación (a modo de ejemplo, el intercalado de columnas con torsión descrito a continuación) sobre los bits de código del código LDPC recibidos desde el intercalador de paridad 23, de modo que una pluralidad de bits de código del código LDPC correspondiente a "1" en una fila arbitraria en la matriz de control de paridad, utilizada por el codificador de LDPC 21, no sea puesta en correspondencia con un solo símbolo.
- 10 El demultiplexor 25 realiza un proceso de reordenamiento sobre el código LDPC recibido desde el intercalador de columnas con torsión 24, de modo que las posiciones de dos o más bits de códigos del código LDPC puestos en correspondencia con un solo símbolo sea reordenado, obteniendo un código LDPC con resistencia aumentada para AWGN y luego, proporciona el código LDPC obtenido para la puesta en correspondencia de la unidad 26.
- 15 La unidad de puesta en correspondencia 26 efectúa la puesta en correspondencia de dos o más bits de código del código LDPC desde el demultiplexor 25 a cada punto de señal, que se determina en conformidad con un sistema de modulación que utiliza el modulador ortogonal 27 para realizar una modulación ortogonal (modulación multivalor).
- 20 Más concretamente, la unidad de puesta en correspondencia 26 convierte el código LDPC desde demultiplexor 25 en símbolos (valores de símbolos) representados por puntos de señales determinados en conformidad con el sistema de modulación sobre un plano de IQ (denominada constelación de IQ) definido con un eje I que representa I componentes de las mismas fases que las portadoras y un eje Q que representa Q componentes ortogonales a las portadoras.
- 25 El sistema de modulación que utiliza el transmisor OFDM de la Figura 1 para realizar la modulación ortogonal incluye un sistema de modulación definido en la especificación DVB-T, de la que formas de realización, a modo de ejemplo, incluyen la Modulación por Desplazamiento de Fase en Cuadratura (QPSK), Modulación de Amplitud en Cuadratura de 16 estados (16QAM), 64QAM, 256QAM, 1024QAM y 4096QAM. Uno de los sistemas de modulación que utiliza el modulador ortogonal 27 para realizar una modulación ortogonal está preestablecido, a modo de ejemplo, mediante la operación por un operador que utiliza el transmisor de la Figura 1. Formas de realización, a modo de ejemplo, de otros sistemas de modulación que el modulador ortogonal 27 puede utilizar para realizar la modulación ortogonal incluyen la modulación en amplitud de 4 pulsos (4PAM).
- 30 El símbolo obtenido en la unidad de puesta en correspondencia 26 se proporciona al intercalador temporal, que puede intercalar diferentes palabras de código LDPC en diferentes símbolos OFDM. La salida del intercalador temporal 30 se alimenta luego al constructor de tramas representado en la Figura 1. Las partes restantes del transmisor, representado en la Figura 1, realizan la modulación ortogonal de las señales de sub-portadoras del símbolo OFDM recibido desde la unidad de puesta en correspondencia 26 para generar una señal modulada y luego, transmite dicha señal modulada.
- 35 La Figura 9 ilustra una matriz de control de paridad H que el codificador LDPC 21 de la Figura 8 utiliza para la codificación de LDPC.
- 40 La matriz de control de paridad H tiene una estructura de Matriz de Generación de Baja Densidad (LDGM) y puede expresarse por una ecuación $H=[H_A|H_T]$ que incluye una matriz de información H_A como un componente a la izquierda y una matriz de paridad H_T como una componente a la derecha, en donde la matriz de información H_A corresponde a bits de información entre los bits de código del código LDPC y la matriz de paridad H_T corresponde a bits de paridad.
- 45 En este caso, el número de bits de información el número de bits de paridad entre los bits de código de un código LDPC (una sola palabra de código) se definen como una longitud de información de K y una longitud de paridad de M y el número de los bits de código se define como una longitud de código de $N(=K+M)$.
- 50 La longitud de paridad K y la longitud de paridad M de un código LDPC de una longitud de código de N se determinan en función de una tasa de códigos. De este modo, la matriz de control de paridad H es una matriz $M \times N$. Además, la matriz de información H_A es una matriz $M \times K$ y la matriz de paridad H_T es una matriz $M \times M$.
- 55 La Figura 10 ilustra una matriz de paridad H_T de una matriz de control de paridad H de un código LDPC definido en la especificación DVB-S.2.
- 60 La matriz de paridad H_T del matriz de control de paridad H de un código LDPC definido en la especificación DVB-S.2 presenta una estructura escalonada, de modo que los elementos de "1" de la matriz de paridad H_T estén dispuestos, de una forma escalonada, según se representa en la Figura 10. La primera fila de la matriz de control de paridad H tiene un peso de ponderación de 1 y las restantes filas tienen un peso de ponderación de 2. La última columna de la matriz de control de paridad H tiene un peso de ponderación de 1 y las restantes columnas tienen un peso de ponderación de 2.
- 65 El código LDPC de la matriz de control de paridad H, que presenta la matriz de paridad puede generarse fácilmente utilizando la matriz de control de paridad H.
- Más concretamente, suponiendo que un vector de fila c representa un código LDPC (palabra de código) se supone que C^T representa un vector de columna obtenido mediante transposición del vector de fila. Además, se supone que un vector de fila A representa una parte de bits de información del vector de fila c, que es el código de LDPC y se supone que un vector de fila T representa su parte de bits de paridad.

En este caso, el vector de fila c puede expresarse por una ecuación " $c=[A|T]$ " que incluye un vector de fila A como componente a la izquierda y un vector de fila T como un componente a la derecha, en donde el vector de fila A corresponde a bits de información y el vector de fila T corresponde a bits de paridad.

5 La matriz de control de paridad H y el vector de fila $c=[A|T]$, que corresponde al código LDPC, necesitan satisfacer una ecuación " $Hc^T=0$ ". De este modo, el valor de cada elemento del vector de fila T correspondiente a los bits de paridad incluidos en el vector de fila $c=[A|T]$, puede obtenerse, de forma secuencial, estableciendo un elemento de cada fila del vector de columna Hc^T en la ecuación " $Hc^T=0$ " a cero en el orden que se inicia desde el elemento de la primera fila, cuando la matriz de paridad H_T , en la matriz de control de paridad $H=[H_A|H_T]$ tiene una estructura escalonada según se ilustra en la Figura 10.

Las Figuras 12a y 12b ilustran una matriz de control de paridad H de un código LDPC y pesos de ponderación de columnas definidos en la especificación DVB-S.2.

15 Es decir, la Figura 111A ilustra una matriz de control de paridad H de un código LDPC definido en la especificación DVB-S.2.

20 En primer lugar, las KX -ésimas columnas de la matriz de control de paridad H tienen un peso de ponderación de columna de X , las $K3$ columnas siguientes tienen un peso de ponderación de columna de 3, las $M-1$ columnas siguientes tienen un peso de ponderación de columna de 2 y la última columna tiene un peso de ponderación de columna de 1.

En este caso, la suma de los números de las columnas " $KX+K3+M-1+1$ " es igual la longitud de código de N .

25 En la especificación DVB-S.2, los números de columnas KX , $K3$ y M (longitud de paridad) y el peso de ponderación de columnas X se definen según se ilustra en la Figura 111B.

Es decir, la Figura 111B ilustra los números de columnas KX , $K3$ y M y el peso de ponderación de columna X para cada tasa de códigos de códigos LDPC definidos en la especificación de DVB-S.2.

30 Dos códigos LDPC, con respectivas longitudes de código de N de 64800 bits y 16200 bits se definen en la especificación DVB-S.2.

35 Además, 11 tasas de códigos nominales $1/4$, $1/3$, $2/5$, $1/2$, $3/5$, $2/3$, $3/4$, $4/5$, $5/6$, $8/9$ y $9/10$ se definen para el código LDPC cuya longitud de código N es 64800 bits y 10 tasas de códigos nominales $1/4$, $1/3$, $2/5$, $1/2$, $3/5$, $2/3$, $3/4$, $4/5$, $5/6$ y $8/9$ se definen para código LDPC cuya longitud de código N es 16200 bits, según se ilustra en la Figura 111B.

Para códigos LDPC, se conoce que la tasa de errores de un bit de código disminuye a medida que aumenta el peso de ponderación de una columna correspondiente al bit de código en una matriz de control de paridad H .

40 En el caso de la matriz de control de paridad H definida en la especificación DVB-S.2 ilustrada en las Figuras 12a y 12b, el peso de ponderación de una columna aumenta a medida que disminuye el número ordinal de la columna (esto es, cuando la columna está más próxima al extremo izquierdo de la matriz de control de paridad H) y, por lo tanto, un bit de código en un código LDPC correspondiente a la matriz de control de paridad H es más sólido operativamente contra (resistente a) errores a medida que disminuye el número ordinal del bit de código (es decir, el primer bit de código es el más resistente) y más susceptible a errores a medida que aumenta el número ordinal del bit de código (es decir, el último bit de código es el más susceptible).

45 Las Figuras 12A y 12B ilustran una disposición de (punto de señales correspondientes a) 16 símbolos en un plano IQ en el caso en donde se realiza 16QAM en el modulador ortogonal 27 representado en la Figura 8.

Es decir, la Figura 13A ilustra símbolos de 16QAM.

50 En 16QAM, un solo símbolo representa 4 bits y se proporcionan $16(=2^4)$ símbolos. Además, 16 símbolos están dispuestos en un cuadrado de 4×4 símbolos en las direcciones I y Q , centrados en el origen del plano IQ.

55 En este caso, cuando y_0 , y_1 , y_2 e y_3 indican 4 bits representados por un solo símbolo de 16QAM, comenzando, de forma secuencial, desde el bit más significativo (MSB), la unidad de puesta en correspondencia 26 de la Figura 8 efectúa la puesta en correspondencia de 4 bits de código de un código LDPC como un símbolo de 4 bits y_0 a y_3 correspondientes a los 4 bits de código en el caso en donde el sistema de modulación es 16QAM.

La Figura 13B muestra límites de bits de cuatro bits y_0 a y_3 representados por un símbolo 16QAM.

60 En este caso, un límite de bits de un bit y_i ($i=0,1, 2, 3$ en las Figuras 12A y 12B) es un límite entre símbolos con un bit y_i de "0" y símbolos con un bit y_i de "1".

Según se ilustra en la Figura 13B, un límite correspondiente al eje Q, en el plano IQ, es el único límite de bits para el primer bit (esto es, el bit MSB) y_0 de los 4 bits y_0 a y_3 representados por un símbolo 16QAM y un límite correspondiente al eje I, en el plano IQ, es el único límite de bits para el segundo bit (esto es, el segundo bit MSB) y_1 .

5 Además, dos límites, uno entre las primera y segunda columnas de símbolos (contando desde el lado izquierdo) entre los 4x4 símbolos y el otro entre las tercera y cuarta columnas, son límites de bits para el tercer bit y_2 .

Además, dos límites, uno entre las primera y segunda filas de símbolos (contando desde la parte superior) entre los 4x4 símbolos y el otro entre las tercera y cuarta filas, son límites de bits para el cuarto bit y_3 .

10 Cada bit y_1 , representado por un símbolo, es más resistente a errores a medida que aumenta el número de símbolos distantes desde el límite de bits y es más susceptible a errores a medida que aumenta el número de símbolos cerca del límite de bits.

15 Cuando un bit resistente a (sólido contra) errores se refiere como un “bit fuerte” y un bit susceptible a (sensible a) errores se refiere como un “bit débil”, el primer bit (es decir, el bit MSB) y_0 y el segundo bit y_1 son bit fuertes y el tercer bit y_2 y el cuarto bit y_3 , son bits débiles, según se ilustra en las Figuras 12A y 12B.

20 Las Figuras 13 a 15 ilustran una disposición de (puntos de señales correspondientes a) 64 símbolos en un plano IQ en el caso en que se realicen 64QAM en el modulador ortogonal 27 representado en la Figura 8.

En 64QAM, un solo símbolo representa 4 bits y $64(=2^6)$ símbolos se proporcionan a este respecto. Además, 64 símbolos están dispuestos en un cuadrado de 8x8 símbolos en las direcciones I y Q, centrados en el origen del plano IQ.

25 En este caso, cuando y_0, y_1, y_2, y_3, y_4 e y_5 indican 6 bits representados por un solo símbolo de 64QAM, comenzando secuencialmente desde el bits más significativo (MSB), la unidad de puesta en correspondencia 26 de la Figura 8 efectúa la puesta en correspondencia de 6 bits de código de un código LDPC con un símbolo de 6 bits y_0 a y_5 correspondiente a los 6 bits de código en el caso en donde el sistema de modulación es 64QAM.

30 La Figura 13 representa límites de bits de los primero y segundo bits y_0 e y_1 entre los 6 bits y_0 a y_5 representados por un símbolo de 64QAM, la Figura 14 representa límites de bits de los tercero y cuarto bits y_2 e y_3 y la Figura 15 representa límites de bits del quinto y sexto bits y_4 e y_5 .

35 Un límite de bits está presente para cada uno de los primero y segundo bits y_0 e y_1 según se ilustra en la Figura 14. Dos límites de bits están presentes para cada uno de los tercero y cuarto bits y_2 e y_3 según se ilustra en la Figura 14 y cuatro límites de bits están presentes para cada uno de los quinto y sexto bits, y_4 e y_5 , según se ilustra en la Figura 15.

40 En consecuencia, los primero y segundo bits y_0 e y_1 , entre los 6 bits y_0 a y_5 , representados por un símbolo de 64QAM son los bits más fuertes, los tercero y cuarto bits y_2 e y_3 son los segundos bits más fuertes y los quinto y sexto bits y_4 e y_5 son los bits débiles.

A partir de las Figuras 12, 13 y 15 se pueden deducir que, en el caso de bits de símbolos con modulación ortogonal, los bits más significativos son bits fuertes y los bits menos significativos son bits débiles.

45 La salida de código de LDPC desde el codificador LDPC 21 de la Figura 8 incluye bits de código susceptibles a errores y bits de código resistentes a errores, según se describió anteriormente con referencia a la Figura 111.

Los bits de símbolos ortogonalmente modulados por el modulador ortogonal 27 incluyen bits fuertes y bits débiles según se describió anteriormente con referencia a las Figuras 12 a 15.

50 En consecuencia, cuando los bits de códigos susceptibles a errores de un código LDPC son puestos en correspondencia con los bits débiles de símbolos ortogonalmente modulados, se reduce la resistencia global a errores.

55 Por lo tanto, la invención recomienda un intercalador que intercale bits de código de un código LDPC, de modo que los bits de códigos susceptibles a errores, del código LDPC, sean puestos en correspondencia con los bits fuertes de símbolos ortogonalmente modulados.

El demultiplexor 25 de la Figura 8 realiza la operación de este intercalador.

60 Las Figuras 16A a 16D ilustran la operación del demultiplexor 25 representado en la Figura 8.

Más concretamente, la Figura 16A ilustra una configuración funcional, a modo de ejemplo, del demultiplexor 25.

65 El demultiplexor 25 incluye una memoria 31 y una unidad de reordenamiento 32. Un código LDPC se proporciona a la memoria 31. La memoria 31 tiene una capacidad de almacenamiento para almacenar mb bits en una dirección de fila (horizontal) y almacenar N/mb bits en una dirección de columna (vertical). Los bits de código del código LDPC

proporcionados a la memoria 31 son objeto de escritura en una dirección de columna para la memoria 31 y son objeto de lectura en una dirección de fila desde la memoria 31 y los bits de código leídos se proporcionan luego a la unidad de reordenamiento 32.

5 En este caso, "m" representa el número de bits de código del código LDPC puestos en correspondencia con un solo símbolo y "b" representa un número entero positivo específico (esto es, factor) por el que se multiplica "m" para obtener un múltiplo entero de "m". Además, "N" (= longitud de información K + longitud de paridad M) representa la longitud de código del código LDPC según se describió anteriormente.

10 La Figura 16A ilustra una configuración, a modo de ejemplo, del demultiplexor 25 cuando el sistema de modulación es 64QAM. En consecuencia, el número de bits de código "m" de un código LDPC puesto en correspondencia con un solo símbolo es 6.

15 En la Figura 16A, el factor "b" es 1 y por lo tanto, la memoria 31 tiene una capacidad de almacenamiento de $N/(6 \times 1) \times (6 \times 1)$ bits en las direcciones de columnas y de filas.

20 En lo sucesivo, una zona de almacenamiento de la memoria 31, que es un solo bit en una dirección de fila y se extiende en una dirección de columna, se refiere como una columna cuando sea adecuado. En la forma de realización, a modo de ejemplo, de la Figura 16A, la memoria 31 incluye $6(=6 \times 1)$ columnas.

El demultiplexor 25 realiza la escritura del bit de código del código LDPC en la memoria 31 en una dirección de columna desde la parte superior a la parte inferior de cada columna, comenzando secuencialmente desde la columna más a la izquierda a la derecha.

25 Cuando los bits de código han sido completamente escritos hasta la parte inferior de la columna más a la derecha, los bits de código son objeto de lectura desde la memoria 31 en una dirección de fila, comenzando secuencialmente desde la primera fila de todas las columnas de la memoria 31 en unidades de 6 bits (es decir, mb bits) y los bits de código leídos se proporcionan a la unidad de reordenamiento 32.

30 La unidad de reordenamiento 32 reordena las posiciones de 6 bits de código recibidos desde la memoria 31 y proporciona, a la salida, los 6 bits reordenados como 6 bits y_0, y_1, y_2, y_3, y_4 e y_5 que representan un solo símbolo 64QAM.

35 Más concretamente, cuando los 6 bits de código leídos desde la memoria 31, en una dirección de fila, se indican por b_0, b_1, b_2, b_3, b_4 y b_5 comenzando secuencialmente desde el bit MSB, los bits de código que incluyen y son adyacentes al bit "b₀" son bits de código resistentes a errores y los bits de código que incluyen y son adyacentes al bit "b₅" son bits de código susceptibles a errores en función de la relación de peso de ponderación de columnas anteriormente descrita con referencia a la Figura 111.

40 La unidad de reordenamiento 32 reordena las posiciones de los 6 bits de código b_0 a b_5 recibidos desde la memoria 31, de modo que los bits de código susceptibles a errores, entre los 6 bits de código b_0 a b_5 desde la memoria 31, son asignados a bits fuertes entre los 6 bits y_0 a y_5 que representan un solo símbolo de 64QAM.

45 Varios operadores han recomendado una diversidad de métodos para reordenar los 6 bits de código b_0 a b_5 desde la memoria 31 y asignándoles a 6 bits y_0 a y_5 que representan un solo símbolo 64QAM, respectivamente.

La Figura 16B ilustra un primer método de reordenamiento, la Figura 16C ilustra un segundo método de reordenamiento y la Figura 16D ilustra un tercer método de reordenamiento.

50 En las Figuras 16B a 16D, una línea que conecta los bits b_i e y_j indica que un bit de código b_i está asignado a un bit de símbolo y_j (esto es, la posición del bit de código b_i se cambia a la del bit de símbolo y_j), similar a las Figuras 17A y 17B, descritas a continuación.

55 Mientras el primer método de reordenamiento, representado en la Figura 16B, recomienda utilizar uno de tres tipos de reordenamiento, el segundo método de reordenamiento, representado en la Figura 16C, recomienda utilizar uno de dos tipos de reordenamiento.

El tercer método de reordenamiento de la Figura 16D recomienda la selección secuencial y la utilización de seis tipos de reordenamiento.

60 Las Figuras 17A y 17B ilustran una configuración, a modo de ejemplo, de un demultiplexor 25 y un cuarto método de reordenamiento en el caso en donde el método de modulación es 64QAM (de modo que el número de bits de código "m" de un código LDPC puesto en correspondencia con un símbolo es 6 como en la Figura 16) y el factor "b" es 2.

65 Cuando el factor "b" es 2, la memoria 31 tiene una capacidad de almacenamiento de $N/(6 \times 2) \times (6 \times 2)$ en las direcciones de columna y de fila y tiene $12(=6 \times 2)$ columnas.

La Figura 17A ilustra el orden en el que los bits de código de un código LDPC son objeto de escritura en la memoria 31.

El demultiplexor 25 realiza la escritura de bits de código del código LDPC en la memoria 31 en una dirección de columna desde la parte superior a la parte inferior de cada columna, comenzando secuencialmente desde la columna más a la izquierda a la derecha, según se describió anteriormente con referencia a la Figura 16A.

Cuando los bits de código se han escrito completamente hasta la parte inferior de la columna más a la derecha, los bits de código son objeto de lectura desde la memoria 31 en una dirección de fila, comenzando secuencialmente desde la primera fila de todas las columnas de la memoria 31 en unidades de 12 bits (esto es, mb bits) y los bits de código leídos se proporcionan a la unidad de reordenamiento 32.

La unidad de reordenamiento 32 reordena las posiciones de 12 bits de códigos recibidos desde la memoria 31 en conformidad con el cuarto método de reordenamiento y proporciona, a la salida, los 12 bits reordenados como 12 bits que representan dos símbolos (esto es, b símbolos) de 64QAM, esto es 6 bits y_0, y_1, y_2, y_3, y_4 e y_5 que representan un símbolo 64QAM y 6 bits y_0, y_1, y_2, y_3, y_4 e y_5 que representan el otro símbolo.

La Figura 17B ilustra el cuarto método de reordenamiento realizado por la unidad de reordenamiento 32 de la Figura 17A.

El último de los métodos de reordenamiento, que reduce al mínimo la tasa de errores en las rutas de comunicación de AWGN, depende de la tasa de códigos de un código LDPC o similar.

La forma en que el intercalador de paridad 23, representado en la Figura 8, realiza el intercalado de paridad se describirá ahora con referencia a las Figuras 18 a 20.

La Figura 18 ilustra (parte de) un gráfico de Tanner de una matriz de control de paridad de un código LDPC.

Si ocurre un error, tal como un borrado operativo simultáneamente en dos o más nodos de variable conectados (o dos o más bits de código correspondientes a) un nodo de control, entonces, el nodo de control reenvía un mensaje con una probabilidad de "0" que es igual a la probabilidad de "1" para cada nodo de variable conectado al nodo de control según se representa en la Figura 18. Por lo tanto, el rendimiento de decodificación se reduce si borran múltiples nodos de variables conectados al mismo nodo de control.

Un LDPC definido en la especificación DVB-S.2, que se proporciona por el codificador de LDPC 21 de la Figura 8, es un código IRA y una matriz de paridad H_T de la matriz de control de paridad H presenta una estructura escalonada según se indica en la Figura 11.

Las Figuras 19A y 19B ilustran una matriz de paridad H_T que tiene una estructura escalonada y un gráfico de Tanner correspondiente a la matriz de paridad H_T .

Es decir, la Figura 19A ilustra una matriz de paridad con estructura escalonada H_T y la Figura 19B ilustra un gráfico de Tanner correspondiente a la matriz de paridad H_T de la Figura 19A.

Cuando la matriz de paridad H_T presenta una estructura escalonada, los nodos de variables, cuyos mensajes se obtienen utilizando bits de código adyacentes (bits de paridad) de un código LDPC, correspondientes a las columnas que incluyen elementos que tengan un valor de "1" en la matriz de paridad H_T , están conectados al mismo nodo de control en el gráfico de Tanner de la matriz de paridad H_T .

En consecuencia, si un error tal como un error de ráfagas o borrado operativo ocurre simultáneamente en bits de paridad adyacentes, disminuye el rendimiento de la decodificación puesto que un nodo de control, conectado a nodos de variables correspondientes respectivamente a los bits de paridad erróneos (esto es, nodos de variables cuyos mensajes se obtienen utilizando los bits de paridad), reenvía un mensaje con una probabilidad de "0" siendo igual a la probabilidad de "1" para cada nodo de variable conectado al nodo de control. El rendimiento de la decodificación se reduce también cuando es grande la longitud de la ráfaga, que es el número de bits erróneos debido a la ráfaga.

Entonces, el intercalador de paridad 23, representado en la Figura 8, realiza una función de intercalado de paridad en el código LDPC desde el codificador de LDPC 21 para intercalar bits de paridad del código LDPC a diferentes posiciones de bit de paridad con el fin de evitar una reducción en el rendimiento de la decodificación.

La Figura 20 ilustra una matriz de paridad H_T de una matriz de control de paridad H, correspondiente a un código LDPC, después de que el intercalador de paridad 23, representado en la Figura 8, realice una función de intercalado de paridad en el código LDPC.

En este caso, una matriz de información H_A , en la matriz de control de paridad H, correspondiente al código LDPC definido en la salida de especificación DVB-S.2, desde el codificador LDPC 21 presenta una estructura cíclica.

El término “estructura cíclica” se refiere a una estructura en donde una columna, cuando se desplaza de forma cíclica, coincide con otra columna. Formas de realización, a modo de ejemplo, de la estructura cíclica incluyen una estructura en la que la posición de un elemento de “1” de cada fila de cada P columnas corresponde a la de la primera de las P columnas que se ha desplazado, de forma cíclica, en una dirección de las columnas en un valor proporcional al valor “q” obtenido dividiendo la longitud de paridad “M”. A continuación, el número de columnas “P”, en la estructura cíclica, se refiere como un número unitario de columnas que tienen una estructura cíclica, cuando sea adecuado.

Formas de realización, a modo de ejemplo, del código LDPC definido en la especificación DVB-S.2, objeto de salida desde el codificador de LDPC 21, incluyen dos tipos de códigos LDPC con respectivas longitudes de código de N de 64800 bits y 16200 bits, según se describió anteriormente con referencia a la Figura 12.

A continuación, la siguiente descripción se concentrará en el tipo de códigos LDPC que tienen una longitud de código N de 64800 bits entre los dos tipos de códigos LDPC con longitudes de códigos respectivas de N 64800 bits y 16200 bits. 11 tasas de código nominales se definen para el código LDPC, cuya longitud de código N es 64800 bits, según se describió anteriormente con referencia a la Figura 12.

Para cualquier código LDPC que tenga una longitud de código N de 64800 bits de cada una de las 11 tasas de códigos nominales, el número unitario de las columnas P, que tiene una estructura cíclica, se define como “360”, que es uno de los divisores (excluyendo 1 y M) de la longitud de paridad M, en la especificación DVB-S.2.

Para un código LDPC que tenga una longitud de código N de 64800 bits de cada una de las 11 tasas de códigos nominales, la longitud de paridad M se calcula para ser un valor no primo según una ecuación $M = qxP = qx360$ utilizando un valor “q” que varía dependiendo de la tasa de código. En consecuencia, de forma similar al número unitario de columnas P que tienen una estructura cíclica, el valor “q” es otro de los divisores (excluyendo 1 y M) de la longitud de paridad M y se calcula dividiendo la longitud de paridad M por el número unitario de columnas P que tienen una estructura cíclica (esto es, la longitud de paridad M es el producto de los divisores “P” y “q” de la longitud de paridad M).

Cuando K es la longitud de información, x es un número entero igual o mayor que 0 y menor que P e y es un número entero igual o mayor que 0 y menor que q, el intercalador de paridad 23 realiza una función de intercalado de paridad sobre el código LDPC recibido desde el codificador LDPC 21 para intercalar un $K+qx+y+1$ -ésimo bit de código entre bits de paridad, que son $K+1$ -ésimo a $K+M (=N)$ -ésimo bits de código del código LDPC, a una posición de $K+Py+x+1$ -ésimo bit de código.

Según este método de intercalado de paridad, (los bits de paridad correspondientes a) nodos de variables, conectados al mismo nodo de control, están a una distancia correspondiente al número unitario de columnas P que presentan una estructura cíclica (360 bits, a modo de ejemplo), impidiendo, de este modo, la ocurrencia de error simultánea en una pluralidad de nodos de variables conectados al mismo nodo de control. Esta circunstancia operativa puede mejorar la resistencia a los errores de ráfagas.

El código LDPC, que ha realizado la operación de intercalado de paridad, de modo que el $K+qx+y+1$ -ésimo bit de código se intercale en la posición de $K+Py+x+1$ -ésimo bit de código, sea idéntico a un código LDPC de una matriz de control de paridad (en adelante, referida como una matriz de control de paridad convertida) que se obtiene realizando una permutación de columnas en la matriz de control de paridad H original para sustituir (más concretamente, permutar) la $K+Py+x+1$ -ésima columna de la matriz de control de paridad H original con la $K+qx+y+1$ -ésima columna.

La matriz de paridad de la matriz de control de paridad convertida presenta una estructura pseudo-cíclica cuyo número unitario de columnas es “P” (360 en la Figura 20), según se indica en la Figura 20.

En este caso, el término “estructura pseudo-cíclica” se refiere a una estructura en la que una parte de la matriz de paridad, excluyendo una parte específica de la matriz de paridad, tiene una estructura cíclica. Una matriz de control de paridad convertida obtenida realizando una permutación de columnas correspondiente al intercalado de paridad, en una matriz de control de paridad de un código LDPC definido en la especificación DVB-S.2 tiene una parte de esquina derecha de 360×360 (correspondiente a una matriz desplazada descrita a continuación) que es solamente un elemento de “1” corto de la estructura cíclica (esto es, la parte de la esquina derecha de 360×360 tiene un elemento de “0” y no de “1”, que se requiere en la estructura cíclica). Puesto que la matriz de control de paridad convertida no tiene una estructura cíclica (completa), se refiere como teniendo una “estructura pseudo-cíclica”.

En realidad, la matriz de control de paridad convertida, representada en la Figura 20, se obtiene realizando una permutación de filas, además de la permutación de columnas correspondiente a la operación de intercalado de paridad, en la matriz de control de paridad H original para permitir que la matriz de control de paridad convertida incluya matrices componentes, descritas a continuación.

Haciendo referencia a las Figuras 21 a 24, se describirá ahora la forma en la que el intercalador de columnas con torsión 24, representado en la Figura 8, realiza el intercalado de columnas con torsión como un proceso de permutación.

El transmisor 11, representado en la Figura 8, transmite dos o más bits de código del código LDPC como un solo símbolo según se describió anteriormente con el fin de mejorar la eficiencia de uso de frecuencias. A modo de ejemplo, QPSK se utiliza como un método de modulación cuando se transmiten dos bits de código como un solo símbolo y 16QAM se emplea como un método de modulación cuando cuatro bits de código se transmiten como un solo símbolo.

5 Si un error tal como un borrado operativo ocurre en un símbolo en el caso de que se transmitan dos o más bits de código como el símbolo según se describió anteriormente, todos los bits de código del símbolo se hacen erróneos (esto es, son operativamente borrados).

10 En consecuencia, para mejorar el rendimiento de la decodificación, con el fin de reducir la probabilidad de (bits de código correspondientes a) nodos de variables conectados al mismo nodo de control que se está suprimiendo simultáneamente, es necesario impedir que los nodos de variables correspondientes a bits de código de un símbolo estén conectados al mismo nodo de control.

15 Por otro lado, en el caso de la matriz de control de paridad H del código LDPC, definida en la salida, según la especificación DVB-S.2 desde el codificador LDPC 21, la matriz de información H_A en la matriz de control de paridad H, presenta una estructura cíclica y la matriz de paridad H_T presenta una estructura escalonada, según se describió anteriormente. En el caso de la matriz de control de paridad convertida, que es la matriz de control de paridad del código LDPC que se ha sometido a la función de intercalado de paridad, la matriz de paridad presenta también una estructura cíclica (más concretamente, una estructura pseudo-cíclica) según se describió anteriormente con referencia a la Figura 20.

Las Figuras 21A y 21B ilustran una matriz de control de paridad convertida.

25 Más concretamente, la Figura 21A ilustra una matriz de control de paridad convertida de una matriz de control de paridad H de un código LDPC que tiene una longitud de código N de 64800 bits y una tasa de código (r) de $\frac{3}{4}$.

En la Figura 21A, la posición de cada elemento que tiene un valor de "1", en la matriz de control de paridad convertida se representa por un punto ".".

30 La Figura 21B ilustra una operación, que el demultiplexor 25, representado en la Figura 8, realiza en un código LDPC de la matriz de control de paridad convertida de la Figura 21A, esto es, un código LDPC que se ha sometido a la operación de intercalado de paridad.

35 En la Figura 21B, utilizando 16QAM como un método de modulación, bits de código del código LDPC con intercalado de paridad son objeto de escritura en una dirección de las columnas, en cuatro columnas que constituyen la memoria 31 del demultiplexor 25.

40 Los bits de código, objeto de escritura en una dirección de columna, en las cuatro columnas de la memoria 31 se leen en una dirección de fila en unidades de 4 bits como un solo símbolo.

En este caso, los cuatro bits de código, B_0 , B_1 , B_2 y B_3 de un símbolo pueden incluir una pluralidad de bits de código, correspondientes a "1" en una fila arbitraria en la matriz de control de paridad convertida representada en la Figura 21A. En este caso, los nodos de variables correspondientes a los cuatro bits de códigos B_0 , B_1 , B_2 y B_3 están conectados al mismo nodo de control.

45 En consecuencia, si ocurre un borrado operativo en un símbolo, en el caso de que los cuatro bits de código B_0 , B_1 , B_2 y B_3 del símbolo incluyan bits de código correspondientes a "1", en una fila arbitraria en la matriz de control de paridad convertida, resulta difícil obtener un mensaje adecuado para el mismo nodo de control conectado a nodos de variables correspondientes, respectivamente, a los bits de código B_0 , B_1 , B_2 y B_3 con lo que se reduce el rendimiento de la decodificación.

50 Cuando se utiliza una tasa de código distinta de $\frac{3}{4}$, una pluralidad de bits de código correspondientes a una pluralidad de nodos de variables conectados al mismo nodo de control pueden constituir un símbolo de 16QAM.

55 Por lo tanto, el intercalador de columnas con torsión 24 realiza una operación de intercalado de columnas con torsión sobre el código LDPC, con intercalado de paridad, desde el intercalador de paridad 23 para intercalar los bits de código del código LDPC, con intercalado de paridad, de modo que una pluralidad de bits de código correspondientes a "1", en una fila arbitraria en la matriz de control de paridad convertida, no sean objeto de puesta en correspondencia con un solo símbolo.

60 La Figura 22 ilustra la forma en que se realiza el intercalado de columnas con torsión.

Más concretamente, la Figura 22 ilustra la memoria 31 del demultiplexor 25 ilustrado en las Figuras 16 y 17.

65

La memoria 31 tiene una capacidad de almacenamiento para almacenar mb bits en una dirección de fila (horizontal) y para almacenar N/mb bits en una dirección de columna (vertical) e incluye mb columnas, según se describió anteriormente con referencia a la Figura 16. El intercalador de columnas con torsión 24 realiza el intercalado de columnas con torsión controlando una posición inicial de escritura en cada columna en la memoria 31, en donde se inicia la escritura en la columna, cuando bits de código de un código LDPC son objeto de escritura en la memoria 31 en una dirección de columna y se leen desde la memoria 31 en una dirección de las filas.

Más concretamente, el intercalador de columnas con torsión 24 cambia adecuadamente la posición de inicio de escritura en donde los bits de código comienzan a escribirse en cada una de la pluralidad de columnas, de modo que una pluralidad de bits de código, leídos en una dirección de las filas para constituir un símbolo, no incluya una pluralidad de bits de código correspondientes a "1" en una fila arbitraria de la matriz de control de paridad convertida. Es decir, el intercalador de columnas con torsión 24 permuta los bits de código del código LDPC, de modo que una pluralidad de bits de código correspondientes a "1", en una fila arbitraria de la matriz de control de paridad, no se incorporen en el mismo símbolo.

La Figura 22 ilustra una configuración, a modo de ejemplo, de la memoria 31 en el caso de que 16QAM se utilice como un método de modulación y el factor "b" descrito con referencia a la Figura 16 es "1". En consecuencia el número de bits de código "m" del código LDPC, objeto de puesta en correspondencia con un símbolo es 4 y la memoria 31 incluye 4 (= mb) columnas.

El intercalador de columnas con torsión 24 de la Figura 22 (en lugar del demultiplexor 25 de la Figura 16) escribe bits de código del código LDPC en la memoria 31 en una dirección de las columnas desde la parte superior a la parte inferior de cada una de las cuatro columnas de la memoria 31, comenzando secuencialmente desde la columna más a la izquierda hacia la derecha.

Cuando los bits de código hayan sido completamente escritos en la columna más a la derecha, el intercalador de columnas con torsión 24 efectúa la lectura de bits de código en unidades de 4 bits (mb bits) en una dirección de las filas, comenzando desde la primera fila de todas las columnas de la memoria 31 y con la salida de los bits de código leídos, como un código LDPC con intercalado de columnas con torsión, a la unidad de reordenamiento 32 del demultiplexor 25, según se ilustra en las Figuras 16 y 17.

Cuando la dirección de la primera posición (superior) de cada columna se representa por "0" y la dirección de cada posición, a lo largo de una dirección de columna, se representa por un número entero secuencialmente creciente, el intercalador de columnas con torsión 24 de la Figura 22, determina que la dirección de una posición inicial de escritura en la columna más a la izquierda es "0", la dirección de una posición de inicio de escritura en la segunda columna (desde la izquierda) es "2", la dirección de posición inicial de escritura en la tercera columna es "4" y la dirección de una posición de inicio de escritura en la cuarta columna es "7".

Después de que se hayan escrito bits de código en una columna que tiene una posición de inicio de escritura, en una dirección distinta de "0", hasta la posición inferior de la columna, el intercalador de columnas con torsión 24 retorna a la primera posición de la columna en una dirección de "0" y continúa la escritura de los bits de código en la columna hasta una posición inmediatamente antes de la posición inicial de escritura, el intercalador de columnas con torsión 24 realiza, entonces, la escritura en la columna siguiente a la derecha.

La realización del intercalado de columnas con torsión, según se describió anteriormente, puede evitar que una pluralidad de bits de código correspondientes a una pluralidad de nodos de variables, conectados al mismo nodo de control, sean asignados a un solo símbolo de 16QAM (esto es, se impide que se incorpore en el mismo símbolo) para un código LDPC de cada tasa de código que tenga una longitud de código N de 64800, según se define en la especificación DVB-S.2. Esta circunstancia operativa puede mejorar el rendimiento de decodificación en una ruta de comunicación en la que ocurre un borrado operativo.

La Figura 23 ilustra el número de columnas de la memoria 31 que se requiere para el intercalado de columnas con torsión y las direcciones de posiciones de inicio de escritura en asociación con cada método de modulación para un código LDPC de cada una de las 11 tasas de código que tienen una longitud de código N de 64800, según se define en la especificación DVB-S.2.

El número de bits "m" de un símbolo es 2 y el factor "b" es 1 cuando uno de los primero a tercero métodos de reordenamiento, representados en la Figura 16, se utiliza en el proceso de reordenamiento del demultiplexor 25 ilustrado en la Figura 8 y QPSK se emplea como el método de modulación.

En este caso, la memoria 31 tiene 2 columnas para almacenar $2 \times 1 (=mb)$ bits en una dirección de las filas y almacena $64800 / (2 \times 1)$ bits en una dirección de las columnas según se ilustra en la Figura 23. La posición de inicio de la escritura de la 1ª de las 2 columnas de la memoria 31 está en una dirección de "0" y la posición de inicio de escritura de la 2ª columna está en una dirección de "2".

Además, el número de bits "m" de un símbolo es 2 y el factor "b" es 2 cuando el cuarto método de reordenamiento de la Figura 17 se utiliza en el proceso de reordenamiento del demultiplexor 25 ilustrado en la Figura 8 y QPSK se utiliza como el método de modulación.

5 En este caso, la memoria 31 tiene 4 columnas para almacenar 2x2 bits en una dirección de las filas y almacena 64800/(2x2) bits en una dirección de las columnas según se ilustra en la Figura 23. La posición inicial de escritura la 1ª de las 4 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "2", la posición inicial de escritura de la 3ª columna está en una dirección de "4" y la posición de inicio de escritura de la 4ª columna está en una dirección de "7".

10 Además, el número de bits "m" de un símbolo es 4 y el factor "b" es 1, cuando uno de los primero a tercero métodos de reordenamiento de la Figura 16 se utiliza en el proceso de reordenamiento del demultiplexor 25 ilustrado en la Figura 8 y 16QAM se emplea como el método de modulación.

15 En este caso, la memoria 31 tiene 4 columnas para almacenar 4x1 bits en una dirección de las filas y almacena 64800/(4x1) bits en una dirección de las columnas según se ilustra en la Figura 23. La posición de inicio de escritura de la 1ª de las 4 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "2", la posición de inicio de escritura de la 3ª columna está en una dirección de "4" y la posición de inicio de escritura de la cuarta columna está en una posición de "7".

20 Además, el número de bits "m" de un símbolo es 4 y el factor "b" es 2 cuando el cuarto método de reordenamiento de la Figura 17 se utiliza en el proceso de reordenamiento del demultiplexor 25 que se ilustra en la Figura 8 y la 16QAM se emplea como el método de modulación.

25 En este caso, la memoria 31 tiene 8 columnas para almacenar 4x2 bits en una dirección de las filas y almacena 64800/(4x2) bits en una dirección de las columnas, según se ilustra en la Figura 23. La posición de inicio de escritura de la primera de las 8 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "0", la posición de inicio de escritura de la 3ª columna está en una dirección de "2", la posición de inicio de escritura de la 4ª columna está en una dirección de "4", la posición de inicio de escritura de la 5ª columna está en una dirección de "4", la posición de inicio de escritura de la 6ª columna está en una dirección de "5", la posición de inicio de escritura de la 7ª columna está en una dirección de "7" y la posición de inicio de escritura de la 8ª columna está en una dirección de "7".

35 Además, el número de bits "m" de un símbolo es 6 y el factor "b" es 1 cuando uno de los primero a tercero métodos de reordenamiento de la Figura 16 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 64QAM se emplea como el método de modulación.

40 En este caso, la memoria 31 tiene 6 columnas para almacenar 6x1 bits en una dirección de las filas y almacena 64800/(6x1) bits en una dirección de las columnas según se ilustra en la Figura 23. La posición de inicio de escritura de la 1ª de las 6 columnas de la memoria 31 está en un dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "2", la posición de inicio de escritura de la 3ª columna está en una dirección de "5", la posición de inicio de escritura de la 4ª columna está en una dirección de "9", la posición de inicio de escritura de la 5ª columna está en una dirección de "10" y la posición de inicio de escritura de la 6ª columna está en una dirección de "13".

45 Además, el número de bits "m" de un símbolo es 6 y el factor "b" es 2 cuando se utiliza el cuarto método de reordenamiento de la Figura 17 en el proceso de reordenamiento del demultiplexor 25 ilustrado en la Figura 8 y 64QAM se emplea como el método de reordenamiento.

50 En este caso, la memoria 31 tiene 12 columnas para almacenar 6x2 bits en una dirección de las filas y almacena 64800/(6x2) bits en una dirección de las columnas, según se ilustra en la Figura 23. La posición de inicio de escritura de la 1ª de las 12 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "0", la posición de inicio de escritura de la 3ª columna está en una dirección de "2", la posición de inicio de escritura de la 4ª columna está en una dirección de "2", la posición de inicio de escritura de la 5ª columna está en una dirección de "3", la posición de inicio de escritura de la 6ª columna está en una dirección de "4", la posición de inicio de escritura de la 7ª columna está en una dirección de "4", la posición de inicio de escritura de la 8ª columna está en una dirección de "5", la posición de inicio de escritura de la 9ª columna está en una dirección de "5", la posición de inicio de escritura de la 10ª columna está en una dirección de "7", la posición de inicio de escritura de la 11ª columna está en una dirección de "8" y la dirección de inicio de escritura de la 12ª columna está en una dirección de "9".

60 Además, el número de bits "m" de un símbolo es 8 y el factor "b" es 1 cuando uno de los primero a tercero métodos de reordenamiento de la Figura 16 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 256QAM se emplea como el método de modulación.

65 En este caso, la memoria 31 tiene 8 columnas para almacenar 8x1 bits en una dirección de las filas y almacena 64800/(8x1) bits en una dirección de las columnas, según se ilustra en la Figura 23. La posición de inicio de escritura de la 1ª de las 8 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna

está en una dirección de "0", la posición de inicio de escritura de la 3ª columna está en una dirección de "2", la posición de inicio de escritura de la 4ª columna está en una dirección de "4", la posición de inicio de escritura de la 5ª columna está en una dirección de "4", la posición de inicio de escritura de la 6ª columna está en una dirección de "5", la posición de inicio de escritura de la 7ª columna está en una dirección de "7" y la posición de inicio de escritura de la 8ª columna está en una dirección de "7".

Además, el número de bits "m" de un símbolo es 8 y el factor "b" es 2 cuando el cuarto método de reordenamiento de la Figura 17 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 256QAM se emplea como el método de modulación.

En este caso, la memoria 31 tiene 16 columnas para almacenar 8x2 bits en una dirección de las filas y almacena 64800/(8x2) bits en una dirección de las columnas, según se ilustra en la Figura 23. La posición de inicio de escritura de la 1ª de las 16 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "2", la posición de inicio de escritura de la 3ª columna está en una dirección de "2", la posición de inicio de escritura de la 4ª columna está en una dirección de "2", la posición de inicio de escritura de la 5ª columna está en una dirección de "2", la posición de inicio de escritura de la 6ª columna está en una dirección de "3", la posición de inicio de escritura de la 7ª columna está en una dirección de "7", la posición de inicio de escritura de la 8ª columna está en una dirección de "15", la posición de inicio de escritura de la 9ª columna está en una dirección de "16", la posición de inicio de escritura de la 10ª columna está en una dirección de "20", la posición de inicio de escritura de la 11ª columna está en una dirección de "22", la posición de inicio de escritura de la 12ª columna está en una dirección de "22", la posición de inicio de escritura de la 13ª columna está en una dirección de "27", la dirección de inicio de escritura de la 14ª columna está en una dirección de "27", la posición de inicio de escritura de la 15ª columna está en una dirección de "28" y la posición de inicio de escritura de la 16ª columna está en una dirección de "32".

Además, el número de bits "m" de un símbolo es 10 y el factor "b" es 1 cuando uno de los primero a tercero métodos de reordenamiento de la Figura 16 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 1024QAM se emplea como el método de modulación.

En este caso, la memoria 31 tiene 10 columnas para almacenar 10x1 bits en una dirección de las filas y almacena 64800/(10x1) bits en una dirección de las columnas, según se ilustra en la Figura 23. La posición de inicio de escritura de la 1ª de las 10 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "3", la posición de inicio de escritura de la 3ª columna está en una dirección de "6", la posición de inicio de escritura de la 4ª columna está en una dirección de "8", la posición de inicio de escritura de la 5ª columna está en una dirección de "11", la posición de inicio de escritura de la 6ª columna está en una dirección de "13", la posición de inicio de escritura de la 7ª columna está en una dirección de "15", la posición de inicio de escritura de la 8ª columna está en una dirección de "17", la posición de inicio de escritura de la 9ª columna está en una dirección de "18" y la posición de inicio de escritura de la 10ª columna está en una dirección de "20".

Además, el número de bits "m" de un símbolo es 10 y el factor "b" es 2 cuando el cuarto método de reordenamiento de la Figura 17 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 1024QAM se emplea como el método de modulación.

En este caso, la memoria 31 tiene 20 columnas para almacenar 10x2 bits en una dirección de las filas y almacena 64800/(10x2) bits en una dirección de las columnas, según se ilustra en la Figura 23. La posición de inicio de escritura de la 1ª de las 20 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "1", la posición de inicio de escritura de la 3ª columna está en una dirección de "3", la posición de inicio de escritura de la 4ª columna está en una dirección de "4", la posición de inicio de escritura de la 5ª columna está en una dirección de "5", la posición de inicio de escritura de la 6ª columna está en una dirección de "6", la posición de inicio de escritura de la 7ª columna está en una dirección de "6", la posición de inicio de escritura de la 8ª columna está en una dirección de "9", la posición de inicio de escritura de la 9ª columna está en una dirección de "13", la posición de inicio de escritura de la 10ª columna está en una dirección de "14", la posición de inicio de escritura de la 11ª columna está en una dirección de "14", la posición de inicio de escritura de la 12ª columna está en una dirección de "16", la posición de inicio de escritura de la 13ª columna está en una dirección de "21", la dirección de inicio de escritura de la 14ª columna está en una dirección de "21", la posición de inicio de escritura de la 15ª columna está en una dirección de "23", la posición de inicio de escritura de la 16ª columna está en una dirección de "25", la posición de inicio de escritura de la 17ª columna está en una dirección de "25", la posición de inicio de escritura de la 18ª columna está en una dirección de "26", la posición de inicio de escritura de la 19ª columna está en una dirección de "28" y la posición de inicio de escritura de la 20ª columna está en una dirección de "30".

Además, el número de bits "m" de un símbolo es 12 y el factor "b" es 1 cuando uno de los primero a tercero métodos de reordenamiento de la Figura 16 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 4096QAM se emplea como el método de modulación.

En este caso, la memoria 31 tiene 12 columnas para almacenar 12x1 bits en una dirección de las filas y almacena 64800/(12x1) bits en una dirección de las columnas, según se ilustra en la Figura 23. La posición de inicio de escritura de la 1ª de las 12 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª

columna está en una dirección de "0", la posición de inicio de escritura de la 3ª columna está en una dirección de "2", la posición de inicio de escritura de la 4ª columna está en una dirección de "2", la posición de inicio de escritura de la 5ª columna está en una dirección de "3", la posición de inicio de escritura de la 6ª columna está en una dirección de "4", la posición de inicio de escritura de la 7ª columna está en una dirección de "4", la posición de inicio de escritura de la 8ª columna está en una dirección de "5", la posición de inicio de escritura de la 9ª columna está en una dirección de "5", la posición de inicio de escritura de la 10ª columna está en una dirección de "7", la posición de inicio de escritura de la 11ª columna está en una dirección de "8" y la posición de inicio de escritura de la 12ª columna está en una dirección de "9".

Además, el número de bits "m" de un símbolo es 12 y el factor "b" es 2 cuando el cuarto método de reordenamiento de la Figura 17 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 4096QAM se emplea como el método de modulación.

En este caso, la memoria 31 tiene 24 columnas para almacenar 12x2 bits en una dirección de las filas y almacena 64800/(12x2) bits en una dirección de las columnas, según se ilustra en la Figura 23. La posición de inicio de escritura de la 1ª de las 24 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "5", la posición de inicio de escritura de la 3ª columna está en una dirección de "8", la posición de inicio de escritura de la 4ª columna está en una dirección de "8", la posición de inicio de escritura de la 5ª columna está en una dirección de "8", la posición de inicio de escritura de la 6ª columna está en una dirección de "8", la posición de inicio de escritura de la 7ª columna está en una dirección de "10", la posición de inicio de escritura de la 8ª columna está en una dirección de "10", la posición de inicio de escritura de la 9ª columna está en una dirección de "10", la posición de inicio de escritura de la 10ª columna está en una dirección de "12", la posición de inicio de escritura de la 11ª columna está en una dirección de "13", la posición de inicio de escritura de la 12ª columna está en una dirección de "16", la posición de inicio de escritura de la 13ª columna está en una dirección de "17", la dirección de inicio de escritura de la 14ª columna está en una dirección de "19", la posición de inicio de escritura de la 15ª columna está en una dirección de "21", la posición de inicio de escritura de la 16ª columna está en una dirección de "22", la posición de inicio de escritura de la 17ª columna está en una dirección de "23", la posición de inicio de escritura de la 18ª columna está en una dirección de "26", la posición de inicio de escritura de la 19ª columna está en una dirección de "37", la posición de inicio de escritura de la 20ª columna está en una dirección de "39", la posición de inicio de escritura de la 21ª columna está en una dirección de "40", la posición de inicio de escritura de la 22ª columna está en una dirección de "41", la posición de inicio de escritura de la 23ª columna está en una dirección de "41" y la posición de inicio de escritura de la 24ª columna está en una dirección de "41".

La Figura 24 ilustra el número de columnas de la memoria 31 que se requiere para el intercalado de columnas con torsión y las direcciones de posiciones de inicio de escritura en asociación con cada método de modulación para un código LDPC de cada una de las 10 tasas de códigos que tienen una longitud de código N de 16200 según se define en la especificación DVB-S.2.

El número de bits "m" de un símbolo es 2 y el factor "b" es 1 cuando uno de los primero a tercero métodos de reordenamiento de la Figura 16 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y QPSK se emplea como el método de modulación.

En este caso, la memoria 31 tiene 2 columnas para almacenar 2x1 bits en una dirección de las filas y almacena 16200/(2x1) bits en una dirección de las columnas, según se ilustra en la Figura 24. La posición de inicio de escritura de la 1ª de las 2 columnas de la memoria 31 está en una dirección de "0" y la posición de inicio de escritura de la 2ª columna está en una dirección de "0".

Además, el número de bits "m" de un símbolo es 2 y el factor "b" es 2 cuando el cuarto método de reordenamiento de la Figura 17 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y QPSK se emplea como el método de modulación.

En este caso, la memoria 31 tiene 4 columnas para almacenar 2x2 bits en una dirección de las filas y almacena 16200/(2x2) bits en una dirección de las columnas, según se ilustra en la Figura 24. La posición de inicio de escritura de la 1ª de las 4 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "2", la posición de inicio de escritura de la 3ª columna está en una dirección de "3" y la posición de inicio de escritura de la 4ª columna está en una dirección de "3".

Además, el número de bits "m" de un símbolo es 4 y el factor "b" es 1 cuando uno de los primero a tercero métodos de reordenamiento de la Figura 16 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 16QAM se emplea como el método de modulación.

En este caso, la memoria 31 tiene 4 columnas para almacenar 4x1 bits en una dirección de las filas y almacena 16200/(4x1) bits en una dirección de las columnas, según se ilustra en la Figura 24. La posición de inicio de escritura de la 1ª de las 4 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "2", la posición de inicio de escritura de la 3ª columna está en una dirección de "3" y la posición de inicio de escritura de la 4ª columna está en una dirección de "3".

Además, el número de bits "m" de un símbolo es 4 y el factor "b" es 2 cuando el cuarto método de reordenamiento de la Figura 17 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 16QAM se emplea como el método de modulación.

5 En este caso, la memoria 31 tiene 8 columnas para almacenar 4x2 bits en una dirección de las filas y almacena 16200/(4x2) bits en una dirección de las columnas, según se ilustra en la Figura 24. La posición de inicio de escritura de la 1ª de las 8 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "0", la posición de inicio de escritura de la 3ª columna está en una dirección de "0", la posición de inicio de escritura de la 4ª columna está en una dirección de "1", la posición de inicio de escritura de la 5ª columna está en una dirección de "7", la posición de inicio de escritura de la 6ª columna está en una dirección de "20", la posición de inicio de escritura de la 7ª columna está en una dirección de "20" y la posición de inicio de escritura de la 8ª columna está en una dirección de "21".

15 Además, el número de bits "m" de un símbolo es 6 y el factor "b" es 1 cuando uno de los primero a tercero métodos de reordenamiento de la Figura 16 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 64QAM se emplea como el método de modulación.

20 En este caso, la memoria 31 tiene 6 columnas para almacenar 6x1 bits en una dirección de las filas y almacena 16200/(6x1) bits en una dirección de las columnas, según se ilustra en la Figura 24. La posición de inicio de escritura de la 1ª de las 6 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "0", la posición de inicio de escritura de la 3ª columna está en una dirección de "2", la posición de inicio de escritura de la 4ª columna está en una dirección de "3", la posición de inicio de escritura de la 5ª columna está en una dirección de "7" y la posición de inicio de escritura de la 6ª columna está en una dirección de "7".

25 Además, el número de bits "m" de un símbolo es 6 y el factor "b" es 2 cuando el cuarto método de reordenamiento de la Figura 17 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 64QAM se emplea como el método de modulación.

30 En este caso, la memoria 31 tiene 12 columnas para almacenar 6x2 bits en una dirección de las filas y almacena 16200/(6x2) bits en una dirección de las columnas, según se ilustra en la Figura 24. La posición de inicio de escritura de la 1ª de las 12 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "0", la posición de inicio de escritura de la 3ª columna está en una dirección de "0", la posición de inicio de escritura de la 4ª columna está en una dirección de "2", la posición de inicio de escritura de la 5ª columna está en una dirección de "2", la posición de inicio de escritura de la 6ª columna está en una dirección de "2", la posición de inicio de escritura de la 7ª columna está en una dirección de "3", la posición de inicio de escritura de la 8ª columna está en una dirección de "3", la posición de inicio de escritura de la 9ª columna está en una dirección de "3", la posición de inicio de escritura de la 10ª columna está en una dirección de "6", la posición de inicio de escritura de la 11ª columna está en una dirección de "7" y la posición de inicio de escritura de la 12ª columna está en una dirección de "7".

40 Además, el número de bits "m" de un símbolo es 8 y el factor "b" es 1 cuando uno de los primero a tercero métodos de reordenamiento de la Figura 16 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 256QAM se emplea como el método de modulación.

45 En este caso, la memoria 31 tiene 8 columnas para almacenar 8x1 bits en una dirección de las filas y almacena 16200/(8x1) bits en una dirección de las columnas, según se ilustra en la Figura 24. La posición de inicio de escritura de la 1ª de las 8 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "0", la posición de inicio de escritura de la 3ª columna está en una dirección de "0", la posición de inicio de escritura de la 4ª columna está en una dirección de "1", la posición de inicio de escritura de la 5ª columna está en una dirección de "7", la posición de inicio de escritura de la 6ª columna está en una dirección de "20", la posición de inicio de escritura de la 7ª columna está en una dirección de "20" y la posición de inicio de escritura de la 8ª columna está en una dirección de "21".

55 Además, el número de bits "m" de un símbolo es 10 y el factor "b" es 1 cuando uno de los primero a tercero métodos de reordenamiento de la Figura 16 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 1024QAM se emplea como el método de modulación.

60 En este caso, la memoria 31 tiene 10 columnas para almacenar 10x1 bits en una dirección de las filas y almacena 16200/(10x1) bits en una dirección de las columnas, según se ilustra en la Figura 24. La posición de inicio de escritura de la 1ª de las 10 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "1", la posición de inicio de escritura de la 3ª columna está en una dirección de "2", la posición de inicio de escritura de la 4ª columna está en una dirección de "2", la posición de inicio de escritura de la 5ª columna está en una dirección de "3", la posición de inicio de escritura de la 6ª columna está en una dirección de "3", la posición de inicio de escritura de la 7ª columna está en una dirección de "4", la posición de inicio de escritura de la 8ª columna está en una dirección de "4", la posición de inicio de escritura de la 9ª columna está en una dirección de "5" y la posición de inicio de escritura de la 10ª columna está en una dirección de "7".

65

Además, el número de bits "m" de un símbolo es 10 y el factor "b" es 2 cuando el cuarto método de reordenamiento de la Figura 17 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 1024QAM se emplea como el método de modulación.

5 En este caso, la memoria 31 tiene 20 columnas para almacenar 10x2 bits en una dirección de las filas y almacena 16200/(10x2) bits en una dirección de las columnas, según se ilustra en la Figura 24. La posición de inicio de escritura de la 1ª de las 20 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "0", la posición de inicio de escritura de la 3ª columna está en una dirección de "0", la posición de inicio de escritura de la 4ª columna está en una dirección de "2", la posición de inicio de escritura de la 5ª columna está en una dirección de "2", la posición de inicio de escritura de la 6ª columna está en una dirección de "2", la posición de inicio de escritura de la 7ª columna está en una dirección de "2", la posición de inicio de escritura de la 8ª columna está en una dirección de "2", la posición de inicio de escritura de la 9ª columna está en una dirección de "5", la posición de inicio de escritura de la 10ª columna está en una dirección de "5", la posición de inicio de escritura de la 11ª columna está en una dirección de "5", la posición de inicio de escritura de la 12ª columna está en una dirección de "5", la posición de inicio de escritura de la 13ª columna está en una dirección de "5", la dirección de inicio de escritura de la 14ª columna está en una dirección de "7", la posición de inicio de escritura de la 15ª columna está en una dirección de "7", la posición de inicio de escritura de la 16ª columna está en una dirección de "7", la posición de inicio de escritura de la 17ª columna está en una dirección de "7", la posición de inicio de escritura de la 18ª columna está en una dirección de "8", la posición de inicio de escritura de la 19ª columna está en una dirección de "8" y la posición de inicio de escritura de la 20ª columna está en una dirección de "10".

Además, el número de bits "m" de un símbolo es 12 y el factor "b" es 1 cuando uno de los primero a tercero métodos de reordenamiento de la Figura 16 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 4096QAM se emplea como el método de modulación.

25 En este caso, la memoria 31 tiene 12 columnas para almacenar 12x1 bits en una dirección de las filas y almacena 16200/(12x1) bits en una dirección de las columnas, según se ilustra en la Figura 24. La posición de inicio de escritura de la 1ª de las 12 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "0", la posición de inicio de escritura de la 3ª columna está en una dirección de "0", la posición de inicio de escritura de la 4ª columna está en una dirección de "2", la posición de inicio de escritura de la 5ª columna está en una dirección de "2", la posición de inicio de escritura de la 6ª columna está en una dirección de "2", la posición de inicio de escritura de la 7ª columna está en una dirección de "3", la posición de inicio de escritura de la 8ª columna está en una dirección de "3", la posición de inicio de escritura de la 9ª columna está en una dirección de "3", la posición de inicio de escritura de la 10ª columna está en una dirección de "6", la posición de inicio de escritura de la 11ª columna está en una dirección de "7" y la posición de inicio de escritura de la 12ª columna está en una dirección de "7".

Además, el número de bits "m" de un símbolo es 12 y el factor "b" es 2 cuando el cuarto método de reordenamiento de la Figura 17 se utiliza en el proceso de reordenamiento del demultiplexor 25 según se ilustra en la Figura 8 y 4096QAM se emplea como el método de modulación.

40 En este caso, la memoria 31 tiene 24 columnas para almacenar 12x2 bits en una dirección de las filas y almacena 16200/(12x2) bits en una dirección de las columnas, según se ilustra en la Figura 24. La posición de inicio de escritura de la 1ª de las 24 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2ª columna está en una dirección de "0", la posición de inicio de escritura de la 3ª columna está en una dirección de "0", la posición de inicio de escritura de la 4ª columna está en una dirección de "0", la posición de inicio de escritura de la 5ª columna está en una dirección de "0", la posición de inicio de escritura de la 6ª columna está en una dirección de "0", la posición de inicio de escritura de la 7ª columna está en una dirección de "0", la posición de inicio de escritura de la 8ª columna está en una dirección de "1", la posición de inicio de escritura de la 9ª columna está en una dirección de "1", la posición de inicio de escritura de la 10ª columna está en una dirección de "1", la posición de inicio de escritura de la 11ª columna está en una dirección de "2", la posición de inicio de escritura de la 12ª columna está en una dirección de "2", la posición de inicio de escritura de la 13ª columna está en una dirección de "2", la dirección de inicio de escritura de la 14ª columna está en una dirección de "3", la posición de inicio de escritura de la 15ª columna está en una dirección de "7", la posición de inicio de escritura de la 16ª columna está en una dirección de "9", la posición de inicio de escritura de la 17ª columna está en una dirección de "9", la posición de inicio de escritura de la 18ª columna está en una dirección de "9", la posición de inicio de escritura de la 19ª columna está en una dirección de "10", la posición de inicio de escritura de la 20ª columna está en una dirección de "10", la posición de inicio de escritura de la 21ª columna está en una dirección de "10", la posición de inicio de escritura de la 22ª columna está en una dirección de "10", la posición de inicio de escritura de la 23ª columna está en una dirección de "10" y la posición de inicio de escritura de la 24ª columna está en una dirección de "11".

60 Un procedimiento de transmisión, realizado por el transmisor 11 de la Figura 8 se describirá ahora con referencia a un diagrama de flujo de la Figura 25.

65 El codificador LDPC 21 espera hasta que se reciban los datos objetivos y codifica los datos objetivos recibidos en un código LDPC y proporciona el código LDPC al intercalador de bits 22 en la etapa S101 y luego, el procedimiento prosigue con la etapa S102.

En la etapa S102, el intercalador de bits 22 realiza el intercalado de bits en el código LDPC desde el codificador LDPC 21 y proporciona el código LDPC con intercalado de bits a la unidad de puesta en correspondencia 26 y luego, el procedimiento prosigue con la etapa S103.

5 Más concretamente, en la etapa S102, el intercalador de paridad 23, en el intercalador de bits 22 realiza el intercalado de paridad en el código LDPC desde la codificación de LDPC 21 y proporciona el código LDPC, con intercalado de paridad, al intercalador de columnas con torsión 24.

10 El intercalador de columnas con torsión 24 realiza el intercalado de columnas con torsión en el código LDPC desde el intercalador de paridad 23 y el demultiplexor 25 realiza un proceso de reordenamiento en el código LDPC que es objeto de intercalado de columnas con torsión por intermedio del intercalador de columnas con torsión 24. El demultiplexor 25 proporciona, entonces, el código LDPC reordenado a la unidad de puesta en correspondencia 26.

15 En la etapa S103, la unidad de puesta en correspondencia 26 efectúa la puesta en correspondencia de m bits de código del código LDPC desde el demultiplexor 25 a un símbolo representado por un punto de señal determinado en conformidad con un sistema de modulación que el modulador ortogonal 27 utiliza para realizar la modulación ortogonal y proporciona el símbolo, puesto en correspondencia, al modulador ortogonal 27 y luego, el procedimiento prosigue con la etapa S104.

20 En la etapa S104, el modulador ortogonal 27 realiza la modulación ortogonal de portadoras en el símbolo desde la unidad de puesta en correspondencia 26 y luego, el procedimiento prosigue con la etapa S105 y transmite la señal ortogonalmente modulada y a continuación, el procedimiento se termina.

25 El procedimiento de transmisión de la Figura 25 es objeto de repetición.

La realización del intercalado de paridad o del intercalado de columnas con torsión, según se describió anteriormente, puede aumentar la resistencia a los borrados operativos o errores de ráfagas cuando una pluralidad de bits de código de un código LDPC se transmite como un símbolo.

30 El intercalado de paridad 23, que es un bloque para realizar el intercalado de paridad y el intercalador de columnas con torsión 24, que es un bloque para realizar el intercalado de columnas con torsión, pueden construirse íntegramente aunque el intercalador de paridad 23 y el intercalador de columnas con torsión 24 se ilustran como estando contruidos por separado en la Figura 8 para facilidad de explicación.

35 Más concretamente, el intercalador de paridad y el intercalador de columnas con torsión pueden efectuar las operaciones de escritura y lectura de los bits de código, a y desde la memoria, y pueden representarse con una matriz que convierta una dirección (dirección de escritura) en donde un bit de código es objeto de escritura para una dirección (dirección de lectura) en la que se lee un bit de código.

40 En consecuencia, es posible obtener un código LDPC que haya sido objeto de intercalado de paridad y luego, haya sido objeto de intercalado de columnas con torsión convirtiendo bits de código utilizando una matriz obtenida multiplicando una matriz que representa un intercalado de paridad y una matriz que representa un intercalado de columnas con torsión.

45 El demultiplexor 25 puede construirse también íntegramente con el intercalador de paridad 23 y el intercalador de columnas con torsión 24.

50 Más concretamente, un proceso de reordenamiento, realizado por el demultiplexor 25, puede representarse también por una matriz que convierte una dirección de escritura de la memoria 31, que almacena un código LDPC para una dirección de lectura.

55 En consecuencia, es posible realizar colectivamente el intercalado de paridad, el intercalado de columnas con torsión y un proceso de reordenamiento que utiliza una matriz obtenida multiplicando una matriz que representa un intercalado de paridad, una matriz que representa un intercalado de columnas con torsión y una matriz que representa un proceso de reordenamiento.

También es posible realizar un intercalado de paridad o un intercalado de columnas con torsión solamente.

60 Las simulaciones para medir las tasas de errores de bits que se realizaron con el transmisor 11 de la Figura 8 se describirán ahora con referencia a las Figuras 26 a 28.

Se realizaron simulaciones utilizando una ruta de comunicación con un dispositivo de fluctuación de 0 dB D/U.

65 Las Figuras 26A y 26B ilustran un modelo para una ruta de comunicación utilizada en las simulaciones.

Más concretamente, la Figura 26A ilustra un modelo de un dispositivo de fluctuación empleado en las simulaciones.

La Figura 26B ilustra un modelo de ruta de comunicación que tiene el dispositivo de fluctuación cuyo modelo se ilustra en la Figura 26A.

5 “H” en la Figura 26B, indica el modelo del dispositivo de fluctuación de la Figura 26A. “N” indica la Interferencia Interportadoras (ICI) en la Figura 26B. En la simulación, una expectativa $E[N^2]$ de potencia de la interferencia ICI fue aproximadamente por AWGN.

10 Las Figuras 27 y 28 ilustran relaciones entre las frecuencias Doppler f_d de dispositivo de fluctuación y tasas de errores desde las simulaciones.

Más concretamente, la Figura 27 representa las relaciones entre tasas de errores y frecuencias Doppler f_d cuando el sistema de modulación es 16QAM, la tasa de código (r) es $\frac{3}{4}$ y el método de reordenamiento es el primer método de reordenamiento. La Figura 28 ilustra las relaciones entre las tasas de errores y las frecuencias Doppler f_d cuando el sistema de modulación es 64QAM, la tasa de código (r) es $\frac{5}{6}$ y el método de reordenamiento es el primer método de reordenamiento.

20 En las Figuras 27 y 28, una línea continua indica una relación entre las tasas de errores y las frecuencias Doppler f_d cuando el intercalado de paridad, el intercalado de columnas con torsión y un proceso de reordenamiento se realizaron todos ellos y una línea delgada indica una relación entre tasas de errores y frecuencias Doppler f_d cuando solamente se realizó el proceso de reordenamiento entre los tres procesos.

25 Puede deducirse de cualquiera de las Figuras 27 y 28 que se mejoran las tasas de errores (esto es, se reducen) cuando el intercalado de paridad, el intercalado de columnas con torsión y un proceso de reordenamiento se realizaron todos ellos, en comparación a cuando solamente se realiza el proceso de reordenamiento.

Receptor

30 La Figura 29 proporciona una ilustración, a modo de ejemplo, de un receptor que puede utilizarse para detectar símbolos de OFDM y para recuperar bits de datos desde las señales de sub-portadoras de los símbolos OFDM. Según se ilustra en la Figura 29, una señal de OFDM se recibe por una antena 500 y se detecta por un sintonizador 502 y se convierte en una forma digital mediante un convertidor analógico a digital 504. Un procesador de eliminación de intervalo de guarda 506 elimina el intervalo de guarda desde un símbolo OFDM recibido, antes de que se recuperen los datos desde el símbolo OFDM utilizando un procesador de Transformada de Fourier Rápida (FFT) 508 en combinación con un estimador de canales y corrección 510, en cooperación con una unidad de decodificación de señalización incorporada 511, en conformidad con las técnicas conocidas. Los símbolos de datos demodulados se recuperan desde un demapeador 512 y se alimentan a un de-intercalador de símbolos 514, que funciona para efectuar el mapeado de puesta en correspondencia inversa de los símbolos de datos recibidos para generar un flujo de símbolos de salida con símbolos de datos de-intercalados. El de-intercalador de símbolos 514 se describirá con más detalle a continuación.

40

Intercalador de bits y codificador de LDPC

45 Según se ilustra en la Figura 29, el receptor incluye, además, una unidad demapeadora 52, un de-intercalador 53 y un decodificador LDPC 56. La unidad demapeadora 52 recibe símbolos (con valores respectivos de direcciones de ejes I y Q) desde el de-intercalador de símbolos 514 y funciona para efectuar el demapeado de los símbolos en bits codificados de un código LDPC y proporciona los bits codificados del código LDPC al de-intercalador de bits 53. La función demapeado de los símbolos de datos recibidos se realiza identificando los bits que están representados por el símbolo de datos identificado desde la señal de sub-portadora del símbolo OFDM.

50 El de-intercalador de bits 53 incluye un demultiplexor 54 y un de-intercalador de columnas con torsión 55 y realiza la operación de de-intercalado sobre los bits de código del código LDPC desde la unidad demapeadora 52.

Más concretamente, el demultiplexor 54 realiza un proceso de reordenamiento inverso, que es el inverso del proceso de reordenamiento realizado por el demultiplexor 25 de la Figura 8, sobre el código LDPC desde la unidad demapeadora 52. Más concretamente, el demultiplexor 54 realiza un proceso de reordenamiento inverso para restablecer las posiciones de los bits de código reordenados por el proceso de reordenamiento en posiciones originales y proporciona el código LDPC, con reordenamiento inverso, al de-intercalador de columnas con torsión 55.

60 El de-intercalador de columnas con torsión 55 realiza un proceso de de-intercalado de columnas con torsión inverso, que es el inverso del intercalado de columnas con torsión como el proceso de permutación realizado por el intercalador de columnas con torsión 24 de la Figura 8, sobre el código LDPC desde el demultiplexor 54. Más concretamente, el de-intercalador de columnas con torsión 55 realiza un proceso de permutación inversa (a modo de ejemplo, de-intercalado de columnas con torsión) para restablecer el orden original de los bits de código del código LDPC reordenado mediante intercalado de columnas con torsión como el proceso de permutación de los bits de código.

65

Más concretamente, el de-intercalador de columnas con torsión 55 realiza la operación de de-intercalado de columnas con torsión mediante la escritura y lectura de los bits de código del código LDPC a y desde una memoria para la operación de de-intercalado que se establece similarmente para la memoria 31 ilustrada en la Figura 22.

5 Sin embargo, el de-intercalador de columnas con torsión 55 realiza la escritura de un bit de código en una dirección de las filas en la memoria para desintercalarse utilizando una dirección de lectura, en donde el bit de código fue objeto de lectura desde la memoria 31, como una dirección de escritura. Además, el de-intercalador de columnas con torsión 55 efectúa la lectura de un bit de código en una dirección de las columnas desde la memoria para su desintercalado utilizando una dirección de escritura, en donde el bit de código fue objeto de escritura en la memoria 31, como una
10 dirección de lectura.

El de-intercalador de columnas con torsión 55 proporciona el código LDPC, con desintercalado de columnas con torsión al decodificador LDPC 56.

15 Aunque las operaciones de intercalado de paridad, intercalado de columnas con torsión y un proceso de reordenamiento se realizaron secuencialmente sobre el código LDPC provisto desde la unidad demapeadora 52 al de-intercalador 53, el decodificador 53 realiza solamente los dos procesos, esto es, un proceso de reordenamiento inverso correspondiente al proceso de reordenamiento y el desintercalado de columnas con torsión correspondiente al intercalado de columnas con torsión, sobre el código LDPC. De este modo, el de-intercalador 53 no realiza el desintercalado de paridad correspondiente al intercalado de paridad (esto es, la inversa de la operación de intercalado de paridad). Es decir, el de-
20 intercalador 53 no realiza el desintercalado de paridad para restablecer el orden original del bit de código del código LDPC reordenado mediante intercalado de paridad.

En consecuencia, el código LDPC, sobre el que se ha realizado el proceso de reordenamiento inverso y el desintercalado de columnas con torsión y no se ha realizado ningún desintercalado de paridad, se proporciona desde (el de-intercalador de columnas con torsión 55 del) de-intercalador 53 al decodificador LDPC 56.

En decodificador LDPC 56 realiza la decodificación de LDPC sobre el código LDPC desde el de-intercalador 53 utilizando una matriz de control de paridad convertida, obtenida realizando al menos una permutación de columnas correspondiente al intercalado de paridad sobre la matriz de control de paridad H que el codificador LDPC 21 de la Figura 8 utilizado para la codificación de LDPC y luego, proporciona, a la salida, los datos resultantes como datos objetivos decodificados.

La Figura 30 es un diagrama de flujo que ilustra un procedimiento de recepción realizado por el receptor 12 de la Figura 29.

El demodulador ortogonal 51 recibe una señal modulada desde el transmisor 11 en la etapa S111. A continuación, el procedimiento prosigue con la etapa S112 para realizar la demodulación ortogonal sobre la señal modulada. El demodulador ortogonal 51 proporciona, luego, un símbolo obtenido mediante la demodulación ortogonal a la unidad demapeadora 52 y luego, el procedimiento prosigue desde la etapa S112 a la etapa S113.

En la etapa S113, la unidad demapeadora 52 efectúa el demapeado del símbolo desde el demodulador ortogonal 51 en bits de código de un código LDPC y proporciona los bits de código de un código LDPC al de-intercalador 53. A continuación, el procedimiento prosigue con la etapa S114.

En la etapa S114, el de-intercalador 53 realiza la operación de desintercalado sobre los bits de código del código LDPC desde la unidad demapeadora 52 y a continuación, el procedimiento prosigue con la etapa S115.

Más concretamente, en la etapa S114, el demultiplexor 54, en el de-intercalador 53 realiza un proceso de reordenamiento inverso sobre el código LDPC desde la unidad demapeadora 52 y proporciona el código LDPC resultante al de-intercalador de columnas con torsión 55.

El de-intercalador de columnas con torsión 55 realiza la operación de desintercalado de columnas con torsión sobre el código LDPC desde el demultiplexor 54 y proporciona el código LDPC resultante al decodificador de LDPC 56.

En la etapa S115, el decodificador de LDPC 56 realiza la decodificación de LDPC sobre el código LDPC desde el de-intercalador de columnas con torsión 55 utilizando una matriz de control de paridad convertida, obtenida realizando al menos la permutación de columnas correspondiente al intercalado de paridad sobre la matriz de control de paridad H que el codificador de LDPC 21 de la Figura 8 utilizó para la codificación de LDPC y luego, proporciona los datos resultantes como datos objetivo decodificados. A continuación, se termina el procedimiento.

El procedimiento de recepción de la Figura 30 es objeto de repetición.

El demultiplexor 54, que realiza un proceso de reordenamiento inverso, y el de-intercalador de columnas con torsión 55, que realiza el desintercalado de columnas con torsión, pueden construirse de forma integrada aunque el demultiplexor 54

y el de-intercalador de columnas con torsión 55 se representan como estando contruidos de forma separada en la Figura 29 en la misma manera que en la Figura 8 para facilidad de explicación.

5 En el caso en que el transmisor 11 de la Figura 8 no realice el intercalado de columnas con torsión, no hay necesidad de proporcionar el de-intercalador de columnas con torsión 55 en el receptor 12 de la Figura 29.

Se hará referencia ahora a la forma en que el codificador de LDPC 56 de la Figura 29 realiza la decodificación de LDPC.

10 El decodificador de LDPC 56 de la Figura 29 realiza la decodificación de LDPC del código LDPC desde el de-intercalador de columnas con torsión 55, en donde se han realizado el proceso de reordenamiento inverso y la operación de desintercalado de columnas con torsión y no se ha realizado ningún desintercalado de paridad, utilizando una matriz de control de paridad convertida obtenida realizando al menos una permutación de columnas correspondiente al intercalado de paridad sobre la matriz de control de paridad H que el codificador de LDPC 21 de la Figura 8 utilizó para la codificación de LDPC.

15 En este caso, la decodificación de LDPC, que se realiza utilizando la matriz de control de paridad convertida, con el fin de reducir la magnitud de los circuitos y limitar la frecuencia operativa dentro de un margen que puede ser completamente conseguible, se ha recomendado con anterioridad (a modo de ejemplo, véase la publicación de la solicitud de patente japonesa nº 2004-343170).

20 En primer lugar, la decodificación de LDPC utilizando la matriz de control de paridad convertida, anteriormente recomendada, se describe haciendo referencia a las Figuras 31 a 34.

25 La Figura 31 ilustra una matriz de control de paridad H, a modo de ejemplo, de un código LDPC que tiene una longitud de código N de 90 y una tasa de código de 2/3.

En la Figura 31, "0" se representa por un punto "." Según se ilustra en las Figuras 32 y 33 descritas a continuación.

30 Una matriz de paridad en la matriz de control de paridad H de la Figura 31 presenta una estructura escalonada.

La Figura 32 ilustra una matriz de control de paridad H' obtenida realizando una permutación de filas de la Expresión Matemática (8) y una permutación de columnas de la Expresión Matemática (9) en la matriz de control de paridad H representada en la Figura 31.

35 Permutación de filas: $6s+t+1$ -ésima fila \rightarrow $5t+s+1$ -ésima fila(8)

Permutación de columnas: $6x+y+61$ -ésima columna \rightarrow $5y+x+61$ -ésima columna ...(9)

40 En las expresiones matemáticas (8) y (9), s, t, x e y son números enteros de modo que se tiene $0 \leq s < 5$, $0 \leq t < 6$, $0 \leq x < 5$ y $0 \leq y < 6$.

45 En conformidad con la permutación de filas de la expresión matemática (8), las filas 1^a , 7^a , 13^a , 19^a y 25^a , cuyos números ordinales proporcionan "1" como un resto cuando se dividen por 6, se cambian a (concretamente, se intercambian con) las filas 2^a , 3^a , 4^a y 5^a , respectivamente y las filas 2^a , 8^a , 14^a , 20^a y 26^a , cuyos números ordinales proporcionan "2" como un resto cuando se dividen por 6, se cambian a la fila 6^a , 7^a , 8^a , 9^a y 10^a , respectivamente.

50 En conformidad con la permutación de columnas de la expresión matemática (9), las columnas 61^a , 67^a , 73^a , 79^a y 89^a entre columnas (paridad) posteriores a la columna 60^a , cuyos números ordinales proporcionan "1" como un resto cuando se dividen por 6, se cambian a las columnas 61^a , 62^a , 63^a , 64^a y 65^a , respectivamente y columnas 62^a , 68^a , 74^a , 80^a y 86^a , cuyos números ordinales proporcionan "2" como un resto cuando se dividen por 6, se cambian a las columnas 66^a , 67^a , 68^a , 69^a y 70^a , respectivamente.

55 Una matriz obtenida realizando la permutación de filas y de columnas en la matriz de control de paridad H de la Figura 31, en esta manera, es la matriz de control de paridad H' de la Figura 32.

En este caso, la realización de la permutación de filas de la matriz de control de paridad H no afecta al orden de los bits de código del código LDPC.

60 La permutación de columnas de la expresión matemática (9) corresponde al intercalado de paridad que se realiza para intercalar el $K+qx+y+1$ -ésimo bit de código a la posición de $K+Py+x+1$ -ésimo bit de código, según se describió anteriormente cuando la longitud de información K es "60" el número unitario de columnas P que presenta una estructura cíclica es "5" y el divisor $q(M/P)$ de la longitud de paridad M (30, en este ejemplo), es "6".

65 Un vector cero es objeto de salida si la matriz de control de paridad H' de la Figura 32 que, en adelante, se refiere como "matriz de control de paridad convertida", cuando sea adecuado, se multiplica por un código LDPC obtenido realizando la misma permutación que en la expresión matemática (9) sobre el código LDPC de la matriz de control de paridad H de la

Figura 31 que, adelante, se refiere como una “matriz de control de paridad original” cuando sea adecuado. Más concretamente, cuando “c” representa un vector de fila obtenido realizando la permutación de columnas de la expresión matemática (9) sobre un vector de fila “c” como un código LDPC (una palabra de código) de la matriz de control de paridad original H , Hc^T es un vector cero debido a la naturaleza de la matriz de control de paridad y por lo tanto, Hc^T es también un vector cero.

De este modo, la matriz de control de paridad convertida H' de la Figura 32 es una matriz de control de paridad del código LDPC c' obtenido realizando la permutación de columnas de la expresión matemática (9) sobre el código c de LDPC de la matriz de control de paridad H original.

En consecuencia, el mismo código LDPC de la matriz de control de paridad H original, que se obtiene mediante decodificación utilizando la matriz de control de paridad H , se puede obtener mediante decodificación por LDPC del código c' de LDPC con permutación de columnas, que fue generado realizando la permutación de columnas de la expresión matemática (9) sobre el código c de LDPC de la matriz de control de paridad H original, utilizando la matriz de control de paridad H' convertida de la Figura 32 y luego, realizando la inversa de la permutación de columnas de la expresión matemática (9) sobre el código c' de LDPC decodificado.

La Figura 33 ilustra la matriz de control de paridad H' convertida de la Figura 32 en donde los elementos se representan como estando dispuestos en unidades de matrices 5×5 espaciadas entre sí.

En la Figura 33, la matriz de control de paridad H' convertida se representa como una combinación de matrices unitarias 5×5 , cada una de las cuales se obtiene sustituyendo uno o más “1” en una matriz unitaria 5×5 con “0” (en adelante referidas como “matrices cuasi-unitarias, cuando sea adecuado), las matrices obtenidas mediante desplazamiento cíclico de matrices unitarias o matrices casi unitarias (en adelante referidas como matrices desplazadas, cuando sea adecuado), matrices, cada una de las cuales es la suma de dos o más de una matriz unitaria, una matriz cuasi-unitaria y una matriz desplazada (en adelante, referida como “matrices sumas” cuando sea adecuado) y matrices de ceros 5×5 .

Es decir, la matriz de control de paridad H' convertida de la Figura 33 puede ser una matriz que incluya matrices unitarias 5×5 , matrices casi unitarias, matrices desplazadas, matrices sumas y matrices cero 5×5 . De este modo, las matrices 5×5 , que constituyen la matriz de control de paridad convertida H' , se referirán ahora como “matrices componentes” cuando sea adecuado.

La decodificación de un código LDPC representado por una matriz de control de paridad representada por matrices componentes $P \times P$ puede realizarse utilizando una arquitectura que realice simultáneamente P cálculos de nodos de control y P cálculos de nodos de variables.

La Figura 34 es un diagrama de bloques que ilustra una configuración, a modo de ejemplo, de un dispositivo de decodificación que realiza la decodificación según se describió anteriormente.

Más concretamente, la Figura 34 ilustra una configuración, a modo de ejemplo, de un dispositivo de decodificación que realiza la decodificación de un código LDPC utilizando la matriz de control de paridad convertida H' de la Figura 33 obtenida realizando al menos la permutación de columnas de la expresión matemática (9) en la matriz de control de paridad H original de la Figura 31.

El dispositivo de decodificación de la Figura 34 incluye una memoria de almacenamiento de datos de borde 300 que comprende 6 FIFOs 300_1 a 300_6 , un selector 301, para seleccionar una de entre FIFOs 300_1 a 300_6 , una unidad de cálculo de nodo de control 302, dos circuitos de desplazamiento cíclicos 303 y 308, una memoria de almacenamiento de datos de borde 304 que incluye 18 FIFOs 304_1 a 304_{18} , un selector 305 para seleccionar una de las FIFOs 304_1 a 304_{18} , una memoria de datos recibidos 306 para almacenar información recibida, una unidad de cálculos de nodos de variables 307, una unidad de cálculo de palabras decodificadas 309, una unidad de permutación de datos recibidos 310 y una unidad de permutación de datos decodificados 311.

En primer lugar, se hace referencia a un método para almacenar datos en la memoria de almacenamiento de datos de borde 300 y 304.

La memoria de almacenamiento de datos de borde 300 incluye el mismo número de 6 FIFOs 300_1 a 300_6 , como un número obtenido dividiendo el número de filas “30” de la matriz de control de paridad convertida H' de la Figura 33 por el número de filas “5” de cada matriz componente. Cada FIFO 300_y ($y = 1, 2, \dots, 6$) incluye zonas de almacenamiento respectivas de múltiples etapas, a o desde cada una de las cuales pueden escribirse o leerse mensajes correspondientes al mismo número de “5” bordes como el número de filas y el número de columnas de cada matriz componente. El número de las etapas de zonas de almacenamiento de cada FIFO 300_y es “9”, que es igual al máximo de los números de 1 (pesos de ponderación de Hamming) en una dirección de las filas de la matriz de control de paridad convertida representada en la Figura 33.

Los datos (esto es, mensajes v_i desde nodos de variables) correspondientes a las posiciones de “1” de la primera a quinta filas de la matriz de control de paridad convertida H' de la Figura 33 se memoriza en FIFO 300_1 en una dirección

horizontal en cada fila simultáneamente mientras se desechan los "0". Más concretamente, cuando (j, i) representa un elemento de la j -ésima fila y i -ésima columna, los datos correspondientes a las posiciones de "1" de una matriz unitaria 5×5 de $(1,1)$ a $(5,5)$ de la matriz de control de paridad convertida H' se memoriza en la zona de almacenamiento de la primera etapa de FIFO 300₁. Los datos correspondientes a las posiciones de "1" de una matriz desplazada de $(1, 21)$ a $(5, 25)$ de la matriz de control de paridad convertida H' , que se obtiene mediante el desplazamiento cíclico de una matriz unitaria 5×5 a la derecha en 3 elementos se memoriza en la zona de almacenamiento de la segunda etapa. De modo similar, los datos se memorizan en la zona de almacenamiento de las 3ª a 8ª etapas en asociación con la matriz de control de paridad convertida H' . Los datos correspondientes a las posiciones de "1" de una matriz desplazada de $(1, 81)$ a $(5, 90)$ de la matriz de control de paridad convertida H' , que se obtiene sustituyendo "1" en la primera fila con "0" en la matriz unitaria 5×5 y con el desplazamiento cíclico de la matriz unitaria 5×5 a la izquierda en un elemento, se memoriza en la zona de almacenamiento de la novena etapa.

Los datos correspondientes a las posiciones de "1" desde la 6ª a 10ª filas de la matriz de control de paridad convertida H' de la Figura 33 se almacenan en FIFO 300₂. Más concretamente, los datos correspondientes a las posiciones de "1" de una primera matriz desplazada incluida en una matriz suma de $(6, 1)$ a $(10, 5)$ de la matriz de control de paridad convertida H' , que se obtiene añadiendo la primera matriz desplazada obtenida mediante desplazamiento cíclico de una matriz unitaria 5×5 a la derecha en un elemento y una segunda matriz desplazada, obtenida desplazando cíclicamente una matriz unitaria 5×5 a la derecha en 2 elementos, se memoriza en la zona de almacenamiento de la primera etapa de FIFO 300₂. Los datos correspondientes a las posiciones de "1" de la segunda matriz desplazada, incluida en la matriz suma de $(6,1)$ a $(10, 5)$ de la matriz de control de paridad convertida H' , se memorizan en la zona de almacenamiento de la segunda etapa de FIFO 300₂.

Más concretamente, cuando una matriz componente que tiene un peso de ponderación de 2 o más, se representa por la suma de dos o más de una matriz unitaria $P \times P$ con un peso de ponderación de 1, una matriz cuasi-unitaria obtenida sustituyendo uno o más "1" de la matriz unitaria con "0" y una matriz desplazada obtenida mediante el desplazamiento cíclico de la matriz unitaria o la matriz casi unitaria, los datos correspondientes a las posiciones de "1" de la matriz unitaria con un peso de ponderación de 1, la matriz casi unitaria, o la matriz desplazada (esto es, mensajes correspondientes a bordes que pertenecen a la matriz unitaria, la matriz cuasi-unitaria o la matriz desplazada) se memorizan en la misma dirección (la misma FIFO entre las FIFO 300₁ a 300₆).

Los datos se memorizan también en las zonas de almacenamiento de la 3ª a 9ª etapas en asociación con la matriz de control de paridad convertida H' .

De modo similar, los datos se memorizan en las FIFOs 300₃ a 300₆ en asociación con la matriz de control de paridad convertida H' .

La memoria de almacenamiento de datos de bordes 304 incluye el mismo número de 18 FIFOs 304₁ a 304₁₈, como un número obtenido dividiendo el número de columnas "90" de la matriz de control de paridad convertida H' por el número de columnas "5" de cada matriz componente. Cada FIFO 304_x ($x=1, 2, \dots, 18$) incluye zonas de almacenamiento respectivas de múltiples etapas, a o desde cada una de las cuales pueden escribirse o leerse simultáneamente mensajes correspondientes al mismo número de "5" bordes como el número de filas y el número de columnas de cada matriz componente convertida H' .

Los datos (esto es, mensajes u_i desde nodos de control) correspondientes a las posiciones de "1" de la primera a quinta columnas de la matriz de control de paridad convertida H' de la Figura 33 se memorizan en FIFO 304₁ en una dirección vertical en cada columna simultáneamente mientras se desechan los "0". Más concretamente, cuando los datos correspondientes a las posiciones de "1" de una matriz unitaria 5×5 de $(1,1)$ a $(5,5)$ de la matriz de control de paridad convertida H' se memoriza en la zona de almacenamiento de la primera etapa de FIFO 304₁. Los datos correspondientes a las posiciones de "1" de una primera matriz desplazada incluida en una matriz suma de $(6, 1)$ a $(10, 5)$ de la matriz de control de paridad convertida H' , que se obtiene añadiendo una primera matriz desplazada generada por el desplazamiento cíclico de una matriz unitaria 5×5 a la derecha en 1 elemento y una segunda matriz desplazada obtenida por desplazamiento cíclico de una matriz unitaria 5×5 , a la derecha en 2 elementos, se memorizan en la zona de almacenamiento de la segunda etapa. Los datos correspondientes a las posiciones de "1" de la segunda matriz desplazada incluida en la matriz suma de $(6, 1)$ a $(10, 5)$ de la matriz de control de paridad convertida H' , se memorizan en la zona de almacenamiento de la tercera etapa.

Más concretamente, cuando una matriz componente que tenga un peso de ponderación de 2 o más, se representa por la suma de dos o más de una matriz unitaria $P \times P$ con un peso de ponderación de 1, una matriz cuasi-unitaria obtenida sustituyendo uno o más "1" de la matriz unitaria con "0" y una matriz desplazada obtenida mediante el desplazamiento cíclico de la matriz unitaria o de la matriz cuasi-unitaria, los datos correspondientes a las posiciones de "1" de la matriz unitaria con un peso de ponderación de 1, la matriz cuasi-unitaria, o la matriz desplazada (esto es, mensajes correspondientes a bordes que pertenecen a la matriz unitaria, la matriz cuasi-unitaria o la matriz desplazada) se memorizan en la misma dirección (la misma FIFO entre las FIFOs 304₁ a 304₁₈).

Los datos se memorizan también en las zonas de almacenamiento de la 4ª a 5ª etapas en asociación con la matriz de control de paridad convertida H' . El número de las etapas de la zona de almacenamiento de FIFO 304₁ es "5", que es

igual al máximo de los números de "1" (pesos de ponderación de Hamming) en una dirección de las filas en las primera a quinta columnas de la matriz de control de paridad convertida H'.

5 De modo similar, los datos se memorizan en las FIFOs 304₂ y 304₃ en asociación con la matriz de control de paridad convertida H' y la longitud (es decir, el número de etapas) de cada FIFO es "5". De modo similar, los datos se memorizan en las FIFOs 304₄ a 304₁₂ en asociación con la matriz de control de paridad convertida H' y la longitud de cada FIFO es "3". De modo similar los datos se memorizan en las FIFOs 304₁₃ a 304₁₈ en asociación con la matriz de control de paridad convertida H' y la longitud de cada FIFO es "2".

10 Se ahora referencia al funcionamiento del dispositivo de decodificación de la Figura 34.

15 En la memoria de almacenamiento de datos de borde 300, que incluye 6 FIFOs 300₁ a 300₆, una FIFO para almacenar datos se selecciona desde entre las FIFOs 300₁ a 300₆, en función de la información (datos de matrices) D312 que indica una fila en la matriz de control de paridad convertida H', a la que pertenecen 5 mensajes D311 recibidos desde el circuito de desplazamiento cíclico 308 situado flujo arriba de la memoria de almacenamiento de datos de borde 300 y los 5 mensajes D311 se recogen y memorizan en la FIFO seleccionada en orden. Cuando se leen datos desde la memoria de almacenamiento de datos de bordes 300, en primer lugar, 5 mensajes D300₁ se leen en orden desde la FIFO 300₁ y luego, se proporcionan al selector 301 situado flujo debajo de la memoria de almacenamiento de datos de borde 300. Después de que se lean completamente los mensajes desde la FIFO 300₁ se leen mensajes en orden desde las FIFOs 300₂ a 300₆ en la memoria de almacenamiento de datos de borde 300 y luego, se proporcionan al selector 301 en la misma manera.

20 El selector 301 selecciona 5 mensajes recibidos desde una FIFO desde la que se están leyendo actualmente datos entre las FIFOs 300₁ a 300₆ en función de una señal de selección D301 y proporciona los mensajes seleccionados como mensajes D302 a la unidad de cálculo del nodo de control 302.

25 La unidad de cálculo del nodo de control 302 incluye 5 calculadores de nodos de control 302₁ a 302₅ y realiza los cálculos del nodo de control en conformidad con la ecuación (7) utilizando los mensajes D302 (D302₁ a D302₅) (correspondientes a mensajes v_i en la ecuación (7)) recibidos a través del selector 301 y proporciona 5 mensajes D303 (D303₁ a D303₅) (correspondientes a mensajes u_i en la ecuación (7)) obtenidos a través de los cálculos del nodo de control para el circuito de desplazamiento cíclico 303.

30 El circuito de desplazamiento cíclico 303 desplaza, de forma cíclica, los 5 mensajes D303₁ a D303₅ por la unidad de cálculo del nodo de control 302 en función de la información (datos de matrices) D305 que indica el número de elementos mediante los cuales una matriz unitaria original fue desplazada, de forma cíclica para obtener cada borde correspondiente en la matriz de control de paridad convertida H' y proporciona los mensajes, con desplazamiento cíclico, como mensajes D304 a la memoria de almacenamiento de datos de borde 304.

35 En la memoria de almacenamiento de datos de borde 304, que incluye 18 FIFOs 304₁ a 304₁₈, una FIFO para almacenar datos se selecciona de entre las FIFOs 304₁ a 304₁₈ en función de la información D305 que indica una fila en la matriz de control de paridad convertida H', a la que pertenecen 5 mensajes D304, recibidos desde el circuito de desplazamiento cíclico 303 situado flujo arriba de la memoria de almacenamiento de datos de borde 304 y los 5 mensajes D304 son recogidos y memorizados en la FIFO seleccionada en orden. Cuando se leen datos desde la memoria de almacenamiento de datos de borde 304, en primer lugar, 5 mensajes D306₁ son leídos en orden desde la FIFO 304₁ y luego, se proporcionan al selector 305 situado flujo abajo de la memoria de almacenamiento de datos de borde 304. Una vez que se hayan leído completamente los datos desde la FIFO 304₁, se leen mensajes en orden desde las FIFOs 304₂ a 304₁₈ en la memoria de almacenamiento de datos de borde 304 y luego, se proporcionan al selector 305 en la misma manera.

40 El selector 305 selecciona 5 mensajes recibidos desde una FIFO desde la que se están leyendo actualmente datos entre las FIFOs 304₁ a 304₁₈, en función de una señal de selección D307 y proporciona los mensajes seleccionados como mensajes D308 a la unidad de cálculo de nodo de variable 307 y la unidad de cálculo de palabras decodificadas 309.

45 Por otro lado, la unidad de permutación de datos recibidos 310 realiza la permutación de columnas de la expresión matemática (9) para permutar un código LDPC D313 recibido a través de una ruta de comunicación y proporciona los datos resultantes como datos recibidos D314 a la memoria de datos recibidos 306. La memoria de datos recibidos 306 calcula y memoriza una Relación de Probabilidad Logarítmica (LLR) de recepción a partir de los datos recibidos D314 recibidos desde la unidad de permutación de datos recibidos 310 y proporciona LLRs de recepción en grupos de 5 LLRs como valores recibidos D309 para la unidad de cálculo del nodo de variable 307 y la unidad de cálculo de palabras decodificadas 309.

50 La unidad de cálculo de nodos de variables 307 incluye 5 unidades de cálculo de nodos de variables 307₁ a 307₅ y realiza cálculos de nodos de variables según la ecuación (1) utilizando los mensajes D308 (D308₁ a D308₅) (correspondientes a mensajes u_j en la ecuación (1)) recibidos a través del selector 305 y los 5 valores recibidos D309 (dientes a los valores recibidos u_{oi} en la ecuación (1)) recibidos desde la memoria de datos recibidos 306 y luego, proporciona 5 mensajes

D310 (D310₁ a D310₅) (correspondientes a mensajes v_i en la ecuación (1)) obtenidos a través de los cálculos de nodos de variables para el circuito de desplazamiento cíclico 308.

5 El circuito de desplazamiento cíclico 308 desplaza, de forma cíclica, los 5 mensajes D310₁ a D310₅ calculados por la unidad de cálculo de nodos de variables 307 en función de la información que indica el número de elementos mediante los cuales una matriz unitaria original fue desplazada, de forma cíclica, para obtener cada borde correspondiente en la matriz de control de paridad convertida H' y proporciona los mensajes desplazados, de forma cíclica, como mensajes D311 a la memoria de almacenamiento de datos de borde 300.

10 El código LDPC se puede decodificar una vez realizando las operaciones anteriores una vez. Después de decodificar el código LDPC un número de veces predeterminado, el dispositivo de decodificación de la Figura 34 obtiene y proporciona, a la salida, datos decodificados finales mediante la unidad de cálculo de palabras decodificadas 309 y la unidad de permutación de datos decodificados 311.

15 Más concretamente, la unidad de cálculo de palabras decodificadas 309 incluye 5 unidades de cálculo de palabras decodificadas 309₁ a 309₅ y realiza, como un proceso final de una pluralidad de procedimientos de decodificación, el cálculo de los datos decodificados (esto es, una palabra decodificada) en función de la ecuación (5) utilizando los 5 mensajes D308 (D308₁ a D308₅) (correspondientes a los mensajes u_j en la ecuación (5)) que es objeto de salida desde el selector 305 y los 5 valores recibidos D309 (correspondientes a los valores recibidos u_{oi} en la ecuación (5)) recibidos desde la memoria de datos recibidos 306 y proporciona los datos decodificados calculados D315 a la unidad de permutación de datos decodificados 311.

20 La unidad de permutación de datos decodificados 311 realiza la inversa de la permutación de columnas de la expresión matemática (9) sobre los datos decodificados D315 recibidos desde la unidad de cálculo de palabras decodificadas 309 para cambiar el orden de los datos decodificados D315 y luego, proporciona, a la salida, los datos resultantes como datos decodificados finales D316.

30 Según se describió anteriormente, una o ambas permutaciones de filas y permutaciones de columnas se realizan sobre la matriz de control de paridad (esto es, la matriz de control de paridad original) para convertirla en una matriz de control de paridad (esto es, matriz de control de paridad convertida) que puede representarse por una combinación de matrices componentes, esto es, una combinación de una matriz unitaria P x P, una matriz cuasi-unitaria obtenida sustituyendo uno o más "1" de la matriz unitaria con "0", una matriz desplazada obtenida mediante el desplazamiento, de forma cíclica, de la matriz unitaria o de la matriz cuasi-unitaria, una matriz suma obtenida añadiendo dos o más de las matrices unitarias, la matriz cuasi-unitaria o la matriz de desplazamiento y una matriz de ceros P x P. Esta conversión de matriz de control de paridad hizo posible emplear, cuando se decodifica un código LDPC, una arquitectura que realiza simultáneamente P cálculos de nodos de control y P cálculos de nodos de variables. La realización simultánea de P cálculos de nodos limita la frecuencia operativa dentro de un margen que puede ser completamente alcanzable, con lo que se hace posible realizar la decodificación un número de veces.

40 De modo similar, al dispositivo de decodificación representado en la Figura 34, el decodificador de LDPC 56 incluido en el receptor 12 de la Figura 29, está diseñado para decodificar un código LDPC realizando simultáneamente P cálculos de nodos de control y P cálculos de nodos de variables.

45 Más concretamente, cuando se supone, para facilidad de explicación, que la matriz de control de paridad de una salida de código LDPC desde el codificador de LDPC 21 incluido en el transmisor 11 de la Figura 8 es una matriz de control de paridad H, en donde una matriz de paridad tiene una estructura escalonada, a modo de ejemplo, según se ilustra en la Figura 31, el intercalador de paridad 23, en el transmisor 11, realiza un intercalado de paridad para intercalar un $K+qx+y+1$ -ésimo bit de código a una posición de $K+Py+x+1$ -ésimo bit de código con la longitud de información K siendo "60", el número unitario de columnas P teniendo una estructura cíclica de "5" y el divisor $q=(M/P)$ de la longitud de paridad M siendo "6".

50 Puesto que este intercalado de paridad corresponde a la permutación de columnas de la expresión matemática (9) según se describió anteriormente, el decodificador de LDPC 56 no necesita realizar la permutación de columnas de la expresión matemática (9).

55 Por lo tanto, en el receptor 12, de la Figura 29, un código de LDPC que no se haya sometido a un desintercalado de paridad, es decir, un código LDPC con permutación de columnas realizada de la expresión matemática (9), se proporciona desde el de-intercalador de columnas con torsión 55 al decodificador LDPC 56, según se describió con anterioridad. El decodificador LDPC 56 realiza los mismos procesos que los realizados por el dispositivo de decodificación de la Figura 34 con la excepción de que la permutación de columnas de la expresión matemática (9) no se realiza en el decodificador de LDPC 56.

60 Más concretamente, la Figura 35 representa una configuración, a modo de ejemplo, del decodificador LDPC 56 de la Figura 29.

65

5 El decodificador de LDPC 56, ilustrado en la Figura 35, tiene la misma configuración que el dispositivo de decodificación representado en la Figura 34, con la excepción de que la unidad de permutación de datos recibidos 310 de la Figura 34 no se proporciona y realiza los mismos procesos que los del dispositivo de decodificación de la Figura 34, con la excepción de la permutación de columnas de la expresión matemática (9) que no se realiza en el decodificador de LDPC 56 y en consecuencia, se omite aquí una descripción de la misma configuración y procesos.

10 El decodificador de LDPC 56 se puede reducir en tamaño en comparación con el dispositivo de decodificación de la Figura 34 puesto que el decodificador de LDPC 56 se puede construir sin la unidad de permutación de datos recibidos 310, según se describió anteriormente.

15 Aunque, para facilidad de explicación, las Figuras 31 a 35 se han descrito con referencia a una forma de realización, a modo de ejemplo, en donde la longitud de código N de un código LDPC es 90, la longitud de información K es 60, el número unitario de columnas P que presenta una estructura cíclica (es decir, el número de filas y el número de columnas de una matriz componente) es 5 y el divisor $q(=M/P)$ de la longitud de paridad M es 6, la longitud de código N , la longitud de información K , el número unitario de columnas P que tiene una estructura cíclica y el divisor $q(=M/P)$ no están limitados a estos valores.

20 Por lo tanto, mientras el codificador LDPC 21, en el transmisor 11 de la Figura 8, proporciona, a la salida, un código LDPC, a modo de ejemplo, teniendo una longitud de código N de 64800, una longitud de información K de $N-Pq(=N-M)$, un número unitario de columnas P que tienen una estructura cíclica de 360 y un divisor q de M/P , el decodificador de LDPC 56 de la Figura 35 puede aplicar al LDPC, con la decodificación del código LDPC realizando simultáneamente P cálculos de nodos de control y P cálculos de nodos de variables.

25 Las anteriores series de procesos se pueden realizar no solamente por hardware sino también por software. Cuando las series de procesos se realizan por software, un programa que pone en práctica el software se instala en un ordenador de uso general o dispositivo similar.

30 La Figura 36 ilustra una configuración, a modo de ejemplo, de una forma de realización de un ordenador con un programa para realizar las series anteriores de procesos allí instalados.

El programa se puede grabar anteriormente en un disco duro 405 o memoria ROM 403 como un medio de registro incorporado en el ordenador.

35 El programa puede también memorizarse de forma temporal o permanente (o grabarse) en un medio de grabación extraíble 411, tal como un disco flexible, una memoria de lectura solamente-disco compacto (CR-ROM), un disco magneto-óptico (MOD), un disco versátil digital (DVD), un disco magnético o una memoria de semiconductores. Este medio de registro extraíble 411 se puede proporcionar como un así denominado paquete de software.

40 En lugar de instalar el programa desde el medio de registro extraíble 411 según se describió anteriormente para un ordenador, el programa puede transmitirse, de forma inalámbrica, desde un emplazamiento de descarga a un ordenador a través de un satélite para difusión vía satélite digital o puede transmitirse por cableado a un ordenador a través de una red tal como una red de área local (LAN) o Internet y el ordenador puede recibir el programa transmitido a través de una unidad de comunicación 408 y puede instalar el programa recibido en un disco duro incorporado 405.

45 El ordenador puede incluir una unidad central de procesamiento (CPU) 402. La unidad CPU 402 está acoplada a una interfaz 410 de entrada/salida (I/O) a través de un bus 401. La CPU 402 ejecuta un programa memorizado en la memoria de lectura solamente (ROM) 403 cuando una orden, que el usuario ha introducido, a modo de ejemplo, operando una unidad de entrada 407 que incluye un teclado, un ratón, un micrófono y dispositivos similares se ha recibido a través de la interfaz I/O 410. Como alternativa, la unidad CPU 402 se carga, en una memoria de acceso directorio (RAM) 404, y ejecuta un programa memorizado en el disco duro 405, que es un programa que ha sido instalado en el disco duro 405 después de recibirse desde un satélite o red a través de la unidad de comunicación 408 o un programa que ha sido instalado en el disco duro 405 después de leerse desde el medio de registro extraíble 411 instalado en una unidad de disco 409. Al ejecutar el programa de esta manera, la CPU 402 realiza los procesos anteriormente descrito con referencia a los diagramas de flujo o los procesos realizados por los componentes anteriormente descritos con referencia a los diagramas de bloques. A continuación, cuando se necesita, la CPU 402 proporciona, a la salida, los resultados de los procesos, a modo de ejemplo, a través de una unidad de salida 406 que incluye una pantalla de cristal líquido (LCD), un altavoz o dispositivos similares a través de la interfaz I/O 410 o transmite los resultados de los procesos a través de la unidad de comunicación 408 o registra los resultados de los procesos en el disco duro 405.

60 En la descripción anterior, conviene señalar que las etapas que describen el programa que hace que el ordenador realice varios tipos de procesamiento no están necesariamente realizadas, de forma cronológica, en el orden descrito anteriormente con referencia a los diagramas de flujo y se pueden realizar en paralelo o de forma individual (a modo de ejemplo, a través del procesamiento de paralelo o procesamiento orientado al objeto).

El programa puede ejecutarse con un ordenador o puede ejecutarse con múltiples ordenadores en una manera distribuida. El programa puede transferirse también a un ordenador distante con el fin de ejecutarse en dicho ordenador distante.

5 Los expertos en esta técnica apreciarán que las formas de realización de la invención no están limitadas a las anteriormente descritas y se pueden realizar varios cambios sin desviarse por ello del alcance de la invención según se da a conocer en las reivindicaciones adjuntas.

10 Más concretamente, aunque el intercalado de paridad o intercalado de columnas con torsión, que es un proceso de permutación, se realiza sobre un código LDPC definido en la especificación DVB-S.2 en las anteriores formas de realización, el intercalado de paridad puede aplicarse a un código LDPC de una matriz de control de paridad en donde una matriz de información no tiene una estructura cíclica, a condición de que una matriz de paridad en la matriz de control de paridad tenga una estructura escalonada y el intercalado de columnas con torsión, como un proceso de permutación, pueda aplicarse a, a modo de ejemplo, un código LDPC de una matriz de control de paridad que se
15 convierte en una estructura pseudo-cíclica mediante al menos una permutación de columnas de un código LDPC cuasi cíclico (QC) de una matriz de control de paridad que tiene una estructura cíclica en su integridad.

Es decir, la matriz de control de paridad de un código LDPC que ha de someterse a un intercalado de paridad, solamente necesita incluir una matriz de paridad que tenga una estructura escalonada y no necesita incluir una matriz de
20 información que presente una estructura cíclica.

La matriz de control de paridad de un código LDPC que ha de someterse a un intercalado de columnas con torsión, como un proceso de permutación, no está limitada a ninguna estructura específica.

25 Además, el proceso de permutación solamente necesita ser capaz de permutar bits de código de un código LDPC, tal como una pluralidad de bits de código correspondientes a "1" en una fila arbitraria de la matriz de control de paridad que no están incorporados en el mismo símbolo y pueden realizarse utilizando un método distinto al intercalado de columnas con torsión. Más concretamente, el proceso de permutación puede realizarse controlando las direcciones de escritura y de lectura, a modo de ejemplo, utilizando una memoria en la que se almacenan datos en solamente una dirección en
30 lugar de la memoria 31 en la que los datos se almacenan en las direcciones de filas y de columnas.

Intercalador de símbolos

35 Se ha propuesto que el número de modos, que están disponibles dentro de la norma DVB-T2 debe ampliarse para incluir un modo de 1k, un modo de 16k y un modo de 32k. La siguiente descripción se proporciona para ilustrar el funcionamiento de un intercalador de símbolos en conformidad con la presente técnica, aunque se apreciará que el intercalador de símbolos puede utilizarse con otros modos y otras normas DVB.

40 Para crear nuevos modos, han de definirse varios elementos, uno de los cuales es el intercalador de símbolos 33. El mapeador de bits a constelación 26, el intercalador de símbolos 33 y el constructor de tramas 32 se ilustran con más detalle en la Figura 37.

45 Según se explicó anteriormente, la presente técnica da a conocer medios para proporcionar un mapeo cuasi óptimo de los símbolos de datos en las señales de sub-portadoras de OFDM. Según la técnica, a modo de ejemplo, el intercalador de símbolos se proporciona para efectuar el mapeado óptimo de símbolos de datos de entrada en señales de sub-portadoras de OFDM, en función de un código de permutación y de un polinomio generador, que se ha verificado mediante análisis de simulación. Por lo tanto, el intercalador de símbolos se combina con el intercalador de bits y la codificación por LDPC para mejorar el rendimiento de la comunicación de datos en canales de comunicación tales como los propuestos para la norma DVB.
50

Según se ilustra en la Figura 37, una ilustración más detallada, a modo de ejemplo, del mapeador de constelación de bits a símbolos 26 y el constructor de tramas 32 se proporcionan para ilustrar una forma de realización, a modo de ejemplo, de la presente técnica. Los bits de datos recibidos desde el intercalador de bits 26 a través de un canal 62 se agrupan en conjuntos de bits para ser mapeados en una celda de datos, en función del número de bits por símbolo proporcionados por el sistema de modulación. Los grupos de bits, que forman una palabra de datos, se alimentan en paralelo a través de canales de datos 64 a un procesador de mapeado 66. El procesador de mapeado 66 selecciona luego uno de los símbolos de datos, en conformidad con un mapeado preasignado. El punto de constelación, que se representa por un componente real y un componente imaginario, se proporciona al canal de salida 29 como una de un conjunto de entrada al constructor de tramas 32.
55
60

El constructor de tramas 32 recibe las celdas de datos desde el mapeador de bits a constelación 28 a través del canal 29, junto con las celdas de datos desde los demás canales 31. Después de construir una trama de numerosas secuencias de llamadas OFDM, las celdas de cada símbolo OFDM son luego objeto de escritura en una memoria de intercalador 100 y de lectura a la salida de la memoria de intercalador 100, en función de las direcciones de escritura y de las direcciones de lectura generadas por un generador de direcciones 102. En conformidad con el orden de escritura y de lectura, se consigue el intercalado de las celdas de datos, generando direcciones adecuadas. El funcionamiento de
65

generador de direcciones 102 y de la memoria de intercalador 100 se describirán con más detalle, a continuación, con referencia a las Figuras 38, 39 y 40. Las celdas de datos intercaladas se combinan, luego, con los símbolos pilotos y de sincronización recibidos desde el conformador de señalización piloto e incorporado 36 en un constructor de símbolos OFDM 37 para formar el símbolo OFDM, que se alimenta al modulador OFDM 38 según se explicó anteriormente.

5 La Figura 38 proporciona una forma de realización, a modo de ejemplo, de partes del intercalador de símbolos 33, que ilustra la presente técnica para intercalar símbolos. En la Figura 38, las celdas de datos de entrada desde el constructor de tramas 32 se escriben en la memoria de intercalador 100. Las celdas de datos se escriben en la memoria de intercalador 100 en función de una dirección de escritura alimentada desde el generador de direcciones 102 en el canal 104 y son objeto de lectura desde la memoria de intercalador 100 en función de una dirección de lectura alimentada desde el generador de direcciones 102 en un canal 106. El generador de direcciones 102 genera la dirección de lectura y la dirección de escritura según se explica a continuación, dependiendo de si el símbolo OFDM es impar o par, lo que se identifica a partir de una señal alimentada desde un canal 108 y que depende de un modo selecciona, que se identifica a partir de una señal alimentada a partir de un canal 110. Según se explicó anteriormente, el modo puede ser uno de entre un modo de 1k, modo de 2k, modo de 4k, modo de 8k, modo de 16k o modo de 32k. Según se explica a continuación, la dirección de escritura y la dirección de lectura se generan de forma diferente para símbolos impares y pares, según se explica con referencia a la Figura 39, que proporciona una puesta en práctica, a modo de ejemplo, de la memoria de intercalador 100.

20 En la forma de realización, a modo de ejemplo, ilustrada en la Figura 39, la memoria de intercalador se ilustra para comprender una parte superior 100 que representa el funcionamiento de la memoria de intercalador en el transmisor y una parte inferior 340, que ilustra el funcionamiento de la memoria del de-intercalador en el receptor. El intercalador 100 y el de-intercalador 340 se ilustran juntos en la Figura 39 con el fin de facilitar el entendimiento de su operación. Según se ilustra en la Figura 39, una representación de la comunicación entre el intercalador 100 y el de-intercalador 340, a través de otros dispositivos y a través de un canal de transmisión, se ha simplificado y representado como una sección 140 entre el intercalador 100 y el de-intercalador 340. El funcionamiento del intercalador 100 se describe en los apartados siguientes:

30 Aunque la Figura 39 proporciona una ilustración de solamente cuatro celdas de datos de entrada en una forma de realización, a modo de ejemplo, de cuatro señales de sub-portadoras de un símbolo OFDM, se apreciará que la técnica ilustrada en la Figura 39 se puede extender a un mayor número de sub-portadoras tales como 756 para el modo de 1k, 1512 para el modo de 2k, 3024 para el modo de 4k y 6048 para el modo de 8k, 12096 para el modo de 16k y 24192 por el modo de 32k.

35 El direccionamiento de entrada y de salida de la memoria de intercalador 100, que se ilustra en la Figura 39, es para símbolos impares y pares. Para un símbolo OFDM par, las celdas de datos se toman desde el canal de entrada 120 y se escriben en la memoria de intercalador 124.1 en conformidad con una secuencia de direcciones 120 generadas para cada símbolo OFDM por el generador de direcciones 102. Las direcciones de escritura se aplican para el símbolo par, de modo que el intercalado ilustrado se efectúa mediante la mezcla aleatoria, shuffling, de las direcciones de escritura. Por lo tanto, para cada símbolo intercalado se tiene $y(h(q)) = y'(q)$.

45 Para símbolos impares, se utiliza la misma memoria de intercalador 124.2. Sin embargo, según se ilustra en la Figura 39, para el símbolo impar, el orden de escritura 132 está en la misma secuencia de direcciones utilizada para la lectura del símbolo par anterior 126. Esta característica permite que el intercalador de símbolos impares y pares se ponga en práctica para solamente utilizar una memoria de intercalador 100 a condición de que la operación de lectura para una dirección dada se realice antes de la operación de escritura. Las celdas de datos escritas en la memoria de intercalador 124, durante los símbolos impares, se leen luego en una secuencia 134 generada por el generador de direcciones 102 para el siguiente símbolo OFDM par y así sucesivamente. De este modo, solamente una dirección se genera por símbolo, con la escritura y lectura para el símbolo OFDM impar/par siendo realizadas al mismo tiempo.

50 En resumen, según se representa en la Figura 39, una vez que se ha calculado el conjunto de direcciones $H(q)$ para todas las sub-portadoras activas, el vector de entrada $Y' = (y_0', y_1', y_2', \dots, y_{N_{max}-1}')$ se procesa para obtener el vector intercalado $Y = (y_0, y_1, y_2, \dots, y_{N_{max}-1})$ definido por:

55 $yH(q) = y'q$ para símbolos pares para $q = 0, \dots, N_{max}-1$

$yq = y'H(q)$ para símbolos impares para $q = 0, \dots, N_{max}-1$

60 Dicho de otro modo, para símbolos OFDM pares, las palabras de entrada se escriben en una forma permutada en una memoria y se vuelven a leer en una forma secuencial, mientras que para los símbolos impares, se escriben secuencialmente y luego se leen de forma permutada. En el caso anterior, la permutación $H(q)$ se define por la tabla siguiente:

q	0	1	2	3
H(q)	1	3	0	2

Tabla 1: permutación para caso simple en donde Nmax = 4

Según se ilustra en la Figura 39, el de-intercalador 340 funciona para invertir el intercalado aplicado por el intercalador 100, aplicando el mismo conjunto de direcciones que se genera por un generador de direcciones equivalente, pero aplicando las direcciones de lectura y de escritura en orden inverso. En consecuencia, para símbolos pares, las direcciones de escritura 342 están en orden secuencial, mientras que las direcciones de lectura 344 se proporcionan por el generador de direcciones. En correspondencia, para los símbolos impares, el orden de escritura 346 se determina a partir del conjunto de direcciones generadas por el generador de direcciones, mientras que la lectura 348 está en orden secuencial.

Generador de direcciones para modos de funcionamiento

Un diagrama de bloques esquemático del algoritmo utilizado para generar la función de permutación H(q) se representa en la Figura 40 para un modo de 32K. Sin embargo, como se apreciará, el intercalador del modo de 32K, representado en la Figura 40, se puede adaptar para funcionar como un intercalador según un modo de 1K, 2K, 4K, 8K o 16K haciendo una adaptación adecuada del polinomio generador y del código de permutación según se explica a continuación.

En la Figura 40, un registro de desplazamiento de realimentación lineal se forma mediante trece etapas de registro 200 y una puerta lógica XOR 202 que está conectada a las etapas del registro de desplazamiento 200 en conformidad con un polinomio generador. Por lo tanto, en función del contenido del registro de desplazamiento 200, un bit siguiente del registro de desplazamiento se proporciona desde la salida de la puerta lógica XOR 202 efectuando dicha función lógica del contenido del registro de desplazamiento R[0], R[1], R[2], R[12] en función del polinomio generador.

$$R'_i[13] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[2] \oplus R'_{i-1}[12]$$

En conformidad con el polinomio generador, se genera una secuencia de bits pseudo-aleatoria a partir del contenido del registro de desplazamiento 200. Sin embargo, con el fin de generar una dirección para el modo de 32k según se ilustra, se proporciona un circuito de permutación 210 que permuta efectivamente el orden de los bits dentro del registro de desplazamiento 200,1 desde un orden $R'_i[n]$ a un orden $R'_i[n]$ a la salida del circuito de permutación 210. Catorce bits desde la salida del circuito de permutación 210 se alimentan luego en un canal de conexión 212 al que se añade un bit más significativo a través de un canal 214 que se proporciona por un circuito de permutación 218. Una dirección de quince bits se genera, por lo tanto, en el canal 212. Sin embargo, con el fin de asegurar la autenticidad de una dirección, un circuito de control de direcciones 216 analiza la dirección generada para determinar si excede un valor máximo predeterminado. El valor máximo predeterminado puede corresponder al número máximo de señales de sub-portadoras, que están disponibles para símbolos de datos dentro del símbolo OFDM, disponibles para el modo que se está utilizando. Sin embargo, el intercalador para el modo de 32k puede utilizarse también para otros modos, de modo que el generador de direcciones 102 puede utilizarse también para el modo de 2k, modo de 4k, modo de 8k, modo de 16k y el modo de 32k, ajustando, en consecuencia, el número de la dirección válida máxima.

Si la dirección generada supera el valor máximo predeterminado, en tal caso, se genera una señal de control por la unidad de control de dirección 216 y se alimenta, a través de un canal de conexión 220 a una unidad de control 224. Si la dirección generada excede el valor máximo predeterminado, entonces, esta dirección se rechaza y una nueva dirección se regenera para ese símbolo particular.

Para el modo de 32k, se define una palabra de (N_r-1) bits R'_i , con $N_r = \log_2 M_{max}$, en donde $M_{max} = 32768$ utilizando un registro LFSR (Registro de Desplazamiento de Realimentación Lineal).

El polinomio utilizado para generar esta secuencia es:

Modo de 32k: $R'_i[13] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[2] \oplus R'_{i-1}[12]$

en donde i varía desde 0 a $M_{max}-1$

Una vez que se haya generado una palabra R'_i , la palabra R'_i mediante una permutación para generar otra palabra de (N_r-1) bits denominada R_j . R_j se deriva de R'_i mediante las permutaciones de bits dadas a continuación.

Posiciones de bits de R'_i	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Posiciones de bits de R_j	6	5	0	10	8	1	11	12	2	9	4	3	13	7

Permutación de bits para el modo de 32k

A modo de ejemplo, esto significa que para el modo de 32K, el número de bits 12 de R'_i se envía al número de posición de bits 5 de R_j .

La dirección H(q) se deriva, entonces, de R_i mediante la aplicación de la ecuación siguiente:

$$H(q) = (i \bmod 2) \cdot 2^{N_r-1} + \sum_{j=0}^{N_r-2} R_i(j) \cdot 2^j$$

5 La parte de $(i \bmod 2) \cdot 2^{N_r-1}$ de la ecuación anterior se representa en la Figura 40 por el bloque basculante T 218.

A continuación, se realiza un control de dirección sobre H(q) para verificar que la dirección generada está dentro del margen de direcciones aceptables: si $H(q) < N_{max}$, en donde $N_{max} = 24192$, a modo de ejemplo, en el modo de 32K, entonces, la dirección es válida. Si la dirección no es válida, se informa a la unidad de control e intentará generar una nueva H(q) incrementando el índice i.

15 La función del bloque basculante es cerciorarse de que no generamos una dirección que excede N_{max} dos veces en una fila. En efecto, si se generó un valor excedente, esto significa que el bit MSB (esto es, el bit basculante) de la dirección H(q) era uno. Por ello, el siguiente valor generado tendrá un bit MSB puesto a cero, con lo que se asegura la generación de una dirección válida. El bit adicional, por lo tanto, reduce una probabilidad de que si una dirección excede la dirección válida máxima predeterminada, entonces la siguiente dirección será una dirección válida. A modo de ejemplo, el bit adicional es el bit más significativo.

20 Las siguientes ecuaciones resumen el comportamiento global y ayudan a entender la estructura en bucle de este algoritmo:

$$q = 0;$$

25 para $(i = 0; i < M_{max}; i = i+1)$

$$\{ H(q) = (i \bmod 2) \cdot 2^{N_r-1} + \sum_{j=0}^{N_r-2} R_i(j) \cdot 2^j; \\ \text{si } (H(q) < N_{max}) \text{ } q = q+1; \}$$

30 **Análisis que soporta el generador de direcciones**

La selección del polinomio generador y del código de permutación, anteriormente escritos, para el generador de direcciones 102 para cada modo de funcionamiento, a modo de ejemplo, el modo de 32k, se ha identificado después del análisis de simulación del rendimiento relativo del intercalador. El rendimiento relativo del intercalador se ha evaluado utilizando una capacidad relativa del intercalador para separar símbolos sucesivos o una "calidad de intercalado". Según se indicó anteriormente, efectivamente el intercalado debe realizarse para los símbolos impares y pares, a la vez, con el fin de utilizar una memoria de intercalador única. La medida relativa de la calidad del intercalador se determina definiendo una distancia D (en número de sub-portadoras). Un criterio C se elige para identificar un número de sub-portadoras que están a una distancia $\leq D$, a la salida del intercalador que estaban a una distancia $\leq D$ a la entrada del intercalador, siendo, entonces, el número de sub-portadoras, para cada distancia D, ponderado con respecto a la distancia relativa. El criterio C se evalúa para los símbolos OFDM impares y pares. Al minimizar C se produce un intercalador de calidad superior.

$$C = \sum_1^{d=D} N_{even}(d) / d + \sum_1^{d=D} N_{odd}(d) / d$$

45 en donde $N_{even}(d)$ y $N_{odd}(d)$ son los números de sub-portadoras en un símbolo par e impar, respectivamente, a la salida del intercalador que permanece dentro del espaciado de sub-portadoras d entre sí.

50 El análisis del intercalador, anteriormente identificado para el modo de 32k, para un valor de $D = 5$ se ilustra en la Figura 41(a) para los símbolos de OFDM pares y en la Figura 41(b) para los símbolos de OFDM impares. En conformidad con el análisis anterior, el valor de C para el código de permutación anteriormente identificado para el modo de 32k generó un valor de $C = 21.75$, es decir, el número ponderado de sub-portadoras con símbolos que se separan por cinco o menos en la salida, en conformidad con la ecuación anterior, era 21.75.

55 Un análisis correspondiente se proporciona para un código de permutación alternativo para símbolos OFDM pares en la Figura 41(c) y para símbolos de OFDM impares en la Figura 41(d). Según puede deducirse al comparar los resultados ilustrados en las Figuras 41(a) y 41(b), existen más componentes presentes que representan símbolos separados por pequeñas distancias, tales como $D = 1$ y $D = 2$, cuando se comparan con los resultados mostrados en las Figuras 41(a) y

41(b), lo que ilustra que el código de permutación anteriormente identificado, para el intercalador de símbolos del modo de 32k, proporciona un intercalador de calidad superior.

Códigos de permutación alternativos

5 Los siguientes quince códigos posibles alternativos (posiciones de $[n]R_i$ bits, en donde $n = 1$ a 15) se han encontrado para proporcionar un intercalador de símbolos con una buena calidad según se determina por el criterio C anteriormente identificado.

Posiciones de bits de R'_i	13	12	11	10	9	8	7	6	5	4	3	2	1	0
[1] posiciones de bits de R_i	0	6	1	7	2	11	12	5	9	8	3	10	4	13
[2] posiciones de bits de R_i	9	5	0	7	2	8	3	6	12	11	4	1	10	13
[3] posiciones de bits de R_i	9	12	0	1	2	13	5	8	6	3	7	4	10	11
[4] posiciones de bits de R_i	13	8	1	12	11	0	9	5	3	7	6	2	10	4
[5] posiciones de bits de R_i	5	8	7	0	3	2	11	4	13	6	1	10	12	9
[6] posiciones de bits de R_i	8	9	5	13	0	10	7	1	12	3	2	4	11	6
[7] posiciones de bits de R_i	11	10	0	7	2	9	8	1	5	3	6	4	12	13
[8] posiciones de bits de R_i	11	4	0	13	10	12	5	7	2	8	3	1	6	9
[9] posiciones de bits de R_i	4	0	5	1	12	2	10	3	13	9	6	11	8	7
[10] posiciones de bits de R_i	4	7	0	8	10	1	6	3	2	9	11	12	13	5
[11] posiciones de bits de R_i	4	6	0	13	12	1	11	2	8	3	10	7	9	5
[12] posiciones de bits de R_i	0	5	1	9	2	12	3	6	8	7	4	10	11	13
[13] posiciones de bits de R_i	12	4	2	11	10	1	13	6	0	9	3	8	5	7
[14] posiciones de bits de R_i	10	6	0	13	12	11	8	5	2	4	3	1	9	7
[15] posiciones de bits de R_i	7	6	0	1	10	3	9	4	2	5	8	11	12	13

10 Permutación de bits para el modo de 32K.

Adaptación del intercalador de símbolos y generador de direcciones para otros modos

15 Según se indicó anteriormente, el intercalador de símbolos, representado en la Figura 40 puede adaptarse para intercalar símbolos desde otros modos cambiando simplemente la dirección válida máxima, el número de etapas en el registro de desplazamiento de realimentación lineal y el código de permutación. En particular, en conformidad con el análisis antes citado, se ha establecido lo siguiente para cada uno de los modos de 1K, 2K, 4K, 8K y 16K.

20 **Modo de 1K**

Dirección válida máxima: aproximadamente mil

Número de etapas en el registro de desplazamiento de realimentación lineal: Nueve

25

Polinomio generador: $R'_i[8] = R'_{i-1}[0] \oplus R'_{i-1}[4]$

Código de permutación:

Posiciones de bits de R'_i	8	7	6	5	4	3	2	1	0
Posiciones de bits de R_i	4	3	2	1	0	5	6	7	8

30

Modo de 2K

Dirección válida máxima: aproximadamente dos mil

35 Número de etapas en el registro de desplazamiento de realimentación lineal: 10

Polinomio generador: $R'_i[9] = R'_{i-1}[0] \oplus R'_{i-1}[3]$

Código de permutación:

5

Posiciones de bits de $R'_i[n]$	9	8	7	6	5	4	3	2	1	0
Posiciones de bits de $R_i[n]$	0	7	5	1	8	2	6	9	3	4

Modo de 4K

Dirección válida máxima: aproximadamente cuatro mil

10

Número de etapas en el registro de desplazamiento de realimentación lineal: Once

Polinomio generador: $R'_i[10] = R'_{i-1}[0] \oplus R'_{i-1}[2]$

15 Código de permutación:

$R'_i[n]$ para n =	10	9	8	7	6	5	4	3	2	1	0
$R_i[n]$ para n =	7	10	5	8	1	2	4	9	0	3	6

Modo de 8K

20 Dirección válida máxima: aproximadamente ocho mil

Número de etapas en el registro de desplazamiento de realimentación lineal: Doce

Polinomio generador: $R'_i[11] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[6]$

25

Código de permutación:

Posiciones de bits de R'_i	11	10	9	8	7	6	5	4	3	2	1	0
Posiciones de bits de R_i	5	11	3	0	10	8	6	9	2	4	1	7

Modo de 16K

30

Dirección válida máxima: aproximadamente dieciséis mil

Número de etapas en el registro de desplazamiento de realimentación lineal: 13

35 Polinomio generador: $R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11]$

Código de permutación:

Posiciones de bits de R'_i	12	11	10	9	8	7	6	5	4	3	2	1	0
Posiciones de bits de R_i	8	4	3	2	0	11	1	5	12	10	6	7	9

40 Descripción adicional del intercalador de símbolos en el receptor

Volviendo al intercalador representado en la Figura 29, el de-intercalador de símbolos 514 está formado a partir de un aparato de procesamiento de datos según se representa en la Figura 42 con una memoria de intercalador 540 y un generador de direcciones 542. La memoria de intercalador 540 es según se representa en la Figura 39 y funciona como ya se explicó anteriormente para efectuar la operación de desintercalado utilizando conjuntos de direcciones generados por el generador de direcciones 542. El generador de direcciones 542 está formado según se ilustra en la Figura 40 y está dispuesto para generar direcciones correspondientes para establecer la correspondencia de mapeado de los símbolos de datos recuperados de cada señal de sub-portadora de OFDM en un flujo continuo de datos de salida.

45

Las partes restantes del receptor OFDM, ilustrado en la Figura 29, se proporcionan para efectuar la decodificación de corrección de errores 518 de los bits de datos codificados por LDPC para corregir errores y recuperar una estimación de los datos de origen.

5 Una ventaja proporcionada por la presente técnica, para el receptor y el transmisor, a la vez, es que un intercalador de símbolos y un de-intercalador de símbolos, que funcionan en los receptores y transmisores, se pueden conmutar entre los modos de 1k, 2k, 4k, 8k, 16k y 32k cambiando los polinomios generadores y el orden de permutación. En consecuencia, el generador de direcciones 542, ilustrado en la Figura 42, incluye una entrada 544, que proporciona una indicación del modo así como una entrada 546 que indica si existen símbolos OFDM impares/pares. De este modo, se proporciona una puesta en práctica flexible porque un intercalador y un de-intercalador de símbolos se pueden formar según se ilustra en las Figuras 38 y 42, con un generador de direcciones según se representa en una u otra de las Figuras 40. Por lo tanto, el generador de direcciones se puede adaptar a los diferentes modos cambiando a los polinomios generadores y los órdenes de permutación indicados para cada uno de los modos. A modo de ejemplo, esta operación se puede efectuar utilizando un cambio de software. Como alternativa, en otras formas de realización, una señal incorporada, que indica el modo de la transmisión de DVB-T2, se puede detectar en el receptor en la unidad de procesamiento de señalización incorporada 511 y se utiliza para configurar automáticamente el de-intercalador de símbolos en conformidad con el modo detectado.

20 Como alternativa, según se indicó anteriormente, se pueden utilizar diferentes intercaladores con diferentes modos, simplemente adaptando la dirección válida máxima en conformidad con el modo que se está utilizando.

Uso óptimo de intercaladores impares

25 Según se ilustra en la Figura 39, dos procesos de intercalado de símbolos, uno para símbolos OFDM pares y otro para símbolos OFDM impares permite la reducción de la cantidad de memoria utilizada durante el intercalado. En la forma de realización, a modo de ejemplo, ilustrada en la Figura 39, el orden de escritura para el símbolo impar que es el mismo que el orden de lectura para el símbolo par y por lo tanto, mientras un símbolo impar está siendo objeto de lectura desde la memoria, un símbolo par puede escribirse en la posición que acaba de leerse; posteriormente, cuando ese símbolo par se lee desde la memoria, el siguiente símbolo impar puede ser objeto de escritura en la posición que se acaba de leer.

30 Según se indicó anteriormente, durante un análisis experimental del rendimiento de los intercaladores (utilizando el criterio C según se definió anteriormente) y que se representa, a modo de ejemplo, en la Figura 43(a) y la Figura 43(b), se ha descubierto que los sistemas de intercalado diseñados para los intercaladores de símbolos de 2k y de 8k para el DVB-T y el intercalador de símbolos de 4k para DVB-H funcionan mejor para los símbolos impares que para los símbolos pares. En consecuencia, a partir de los resultados de evaluación del rendimiento de los intercaladores, a modo de ejemplo, según se ilustra en las Figuras 43(a) y 43(b), se ha descubierto que los intercaladores impares funcionan mejor que los intercaladores pares. Esta circunstancia puede deducirse al comparar la Figura 43(a) que muestra resultados para un intercalador para símbolos pares y la Figura 43(b) que ilustra resultados para símbolos impares: puede deducirse que la distancia media a la salida del intercalador de sub-portadoras que eran adyacentes a la entrada del intercalador es mayor para un intercalador para símbolos impares que para un intercalador para símbolos pares.

45 Como se entenderá, la cantidad de memoria de intercalador requerida para poner en práctica un intercalador de símbolos, depende del número de símbolo de datos objeto de mapeado en los símbolos de portadoras de OFDM. De este modo, un intercalador de símbolos, en el modo de 16k, requiere la mitad de la memoria necesaria para poner en práctica un intercalador de símbolos del modo de 32k y de forma similar, la cantidad de memoria necesaria para poner en práctica un intercalador de símbolos de 8k es la mitad de la requerida para poner en práctica un intercalador de 16k. Por lo tanto, un transmisor o receptor que esté diseñado para poner en práctica un intercalador de símbolos de un modo, que establece el número máximo de símbolos de datos que se pueden transmitirse por símbolos OFDM, entonces, ese receptor o transmisor incluirá memoria suficiente para poner en práctica dos procesos de intercalado impar para cualquier otro modo, que proporcionan la mitad o menos de la mitad del número de sub-portadoras por símbolos OFDM en ese modo máximo dado. A modo de ejemplo, un receptor o un transmisor, que incluye un intercalador de 32k, tendrán suficiente memoria para admitir dos procesos de intercalado impar de 16k, cada uno con su propia memoria de 16k.

55 Por lo tanto, con el fin de obtener el mejor rendimiento de los procesos de intercalado impar, un intercalador de símbolos, capaz de admitir múltiples modos de modulación, puede diseñarse de modo que solamente se utilice un proceso de intercalado de símbolos impar, si en un modo que comprende la mitad o menos de la mitad del número de sub-portadoras en un modo máximo, que representa el número máximo de sub-portadoras por símbolo OFDM. Este modo máximo, por lo tanto, establece la magnitud de memoria máxima. A modo de ejemplo, en un transmisor/receptor capaz del modo de 32k, cuando se funciona en un modo con menos portadoras (esto es, 16k, 8k, 4k o 1k), entonces en lugar de emplear procesos de intercalado de símbolos impares y pares por separado, utilizaría dos intercaladores impares.

60 Una ilustración de una adaptación del intercalador de símbolos 33, que se representa en la Figura 38, cuando se intercalan símbolos de datos a la entrada en las sub-portadoras de símbolos OFDM, en el modo de intercalado impar, solamente se ilustra en la Figura 44. El intercalador de símbolos 33.1 corresponde exactamente al intercalador de símbolos 33 según se ilustra en la Figura 38, con la excepción de que el generador de direcciones 102.1 está adaptado para realizar solamente procesos de intercalado impar. Para la forma de realización, a modo de ejemplo, ilustrada en la

Figura 44, el intercalador de símbolos 33.1 está funcionando en un modo en donde el número de símbolos de datos, que pueden transmitirse por el símbolo OFDM, es menor que la mitad del número máximo que puede transmitirse por un símbolo OFDM en un modo operativo con el mayor número de sub-portadoras por símbolo OFDM. En consecuencia, el intercalador de símbolos 33.1 se ha dispuesto para la partición de la memoria de intercalador 100. Para la presente ilustración, representada en la Figura 44, la memoria de intercalador 100 se divide, entonces, en dos partes, 601, 602. A modo de ilustración del intercalador de símbolos 33.1, que funciona en un modo en el que los símbolos de datos son puestos en correspondencia con los símbolos OFDM que utilizan el proceso de intercalado impar, la Figura 44 proporciona una vista ampliada de cada mitad de la memoria de intercalador 601, 602. La vista ampliada proporciona una ilustración del modo de intercalado impar según se representa para el lado del transmisor para cuatro símbolos A, B, C, D reproducidos a partir de la Figura 39. En consecuencia, según se ilustra en la Figura 44, para conjuntos sucesivos de primeros y segundos símbolos de datos, los símbolos de datos son objeto de escritura en la memoria de intercalador 601, 602 en un orden secuencial y objeto de lectura en función de las direcciones generadas por el generador de direcciones 102 en un orden permutado, en función de las direcciones generadas por el generador de direcciones según se explicó anteriormente. De este modo, según se ilustra en la Figura 44, puesto que se está realizando un proceso de intercalado impar para conjuntos sucesivos de primeros y segundos conjuntos de símbolos de datos, la memoria del intercalador debe ser objetivo de partición en dos partes. Los símbolos desde un primer conjunto de símbolos de datos son objeto de escritura en una primera mitad de la memoria de intercalador 601 y los símbolos desde un segundo conjunto de símbolos de datos son objeto de escritura una segunda parte de la memoria de intercalador 602, porque el intercalador de símbolos ya no es capaz de reutilizar las mismas partes de la memoria de intercalador de símbolos que las que se pueden admitir cuando se funciona en un modo de intercalado impar y par.

Una forma de realización, a modo de ejemplo, correspondiente del intercalador en el receptor, que se representa en la Figura 42, pero adaptada para funcionar con un proceso de intercalado impar solamente, se ilustra en la Figura 45. Según se representa en la Figura 45, la memoria de intercalador 540 se divide en dos mitades 710, 712 y el generador de direcciones 542 está adaptado para escritura de símbolos de datos en la memoria de intercalador y para leer símbolos de datos desde la memoria de intercalador en las respectivas partes de la memoria 710, 712, para conjuntos sucesivos de símbolos de datos para poner en práctica solamente un proceso de intercalado impar. Por lo tanto, en correspondencia con la representación ilustrada en la Figura 44, la Figura 45 ilustra el mapeado de puesta en correspondencia del proceso de intercalado, que se realiza en el receptor y se ilustra en la Figura 39 como una vista ampliada que se utiliza para las primera y segunda mitades de la memoria de intercalador 710, 712. Por lo tanto, un primer conjunto de símbolos de datos son objeto de escritura en una primera parte de la memoria de intercalador 710 en un orden permutado definido en función de las direcciones generadas por el generador de direcciones 542, según se ilustra por el orden de escritura de los símbolos de datos que proporcionan una secuencia de escritura de 1, 3, 0, 2. Según se ilustra, los símbolos de datos son objeto de posterior lectura desde la primera parte de la memoria de intercalador 710 en un orden secuencial, con lo que se recupera la secuencia original A, B, C, D.

En correspondencia, un segundo conjunto de símbolos de datos subsiguiente, que se recuperan a partir de un símbolo OFDM sucesivo, son objeto de escritura en la segunda mitad en la memoria de intercalador 712 en función de las direcciones generadas por el generador de direcciones 542 en un orden permutado y son objeto de lectura en el flujo de datos de salida en un orden secuencial.

En una forma de realización, a modo de ejemplo, las direcciones generadas para un primer conjunto de símbolos de datos para la escritura en la primera mitad de la memoria de intercalador 710 se pueden reutilizar para la escritura de un segundo conjunto posterior de símbolos de datos en la memoria de intercalador 712. En correspondencia, el transmisor puede reutilizar también direcciones generadas para una mitad del intercalador para un primer conjunto de símbolos de datos para lectura de un segundo conjunto de símbolos de datos que han sido objeto de escritura en la segunda mitad de la memoria en orden secuencial.

Utilización de una secuencia de permutaciones

En una forma de realización, a modo de ejemplo, el generador de direcciones puede aplicar un código de permutación diferente a partir de un conjunto de códigos de permutación para símbolos OFDM sucesivos. Utilizando una secuencia de permutaciones, en el generador de direcciones del intercalador, se reduce la probabilidad de que cualquier bit de entrada de datos al intercalador no module siempre la misma sub-portadora en el símbolo OFDM. A modo de otro ejemplo, dos generadores de direcciones se podrían utilizar, uno de ellos para generar direcciones para el primer conjunto de símbolos de datos y la primera mitad de la memoria y el otro para generar una secuencia diferente de direcciones para el segundo conjunto de símbolos de datos y la segunda mitad de la memoria. Los dos generadores de direcciones podrían diferir en su elección de código de permutación desde la tabla de permutaciones adecuadas anteriores, a modo de ejemplo.

A modo de otro ejemplo, se podría utilizar una secuencia cíclica, de modo que un código de permutación diferente, en un conjunto de códigos de permutación, en una secuencia, se utilice para símbolos OFDM sucesivos y luego, se repita el proceso. Esta secuencia cíclica podría ser, a modo de ejemplo, de una longitud de dos o cuatro. Para las formas de realización, a modo de ejemplo, del intercalador de símbolos de 16k, una secuencia de dos códigos de permutación que sean objeto de ciclo a través de un símbolo OFDM podría ser, a modo de ejemplo:

ES 2 416 356 T3

8 4 3 2 0 1 1 1 5 12 10 6 7 9
7 9 5 3 1 1 1 4 0 2 12 10 8 6

mientras que una secuencia de cuatro códigos de permutación podría ser:

5

8 4 3 2 0 1 1 1 5 12 10 6 7 9
7 9 5 3 1 1 1 4 0 2 12 10 8 6
6 1 1 7 5 2 3 0 1 10 8 12 9 4
5 12 9 0 3 10 2 4 6 7 8 11 1

10

La conmutación de un código de permutación a otro podría efectuarse en respuesta a un cambio en la señal impar/par indicada en el canal de control 108. En respuesta, la unidad de control 224, cambia el código de permutación en el circuito de código de permutación 210 por intermedio de la línea de control 111.

15 Para la realización, a modo de ejemplo, de un intercalador de símbolos de 1k, dos códigos de permutación podrían ser:

4 3 2 1 0 5 6 7 8
3 2 5 0 1 4 7 8 6

20 mientras que cuatro códigos de permutación podrían ser:

4 3 2 1 0 5 6 7 8
3 2 5 0 1 4 7 8 6
7 5 3 8 2 6 1 4 0
1 6 8 2 5 3 4 0 7

25

Otras combinaciones de secuencias pueden ser posibles para modos de portadoras de 2k, 4k y 8k o en realidad, un modo de portadora de 0.5k. A modo de ejemplo, los siguientes códigos de permutación para cada uno de los modos de 0.5k, 2k, 4k y 8k proporcionan una de-correlación adecuada de símbolos y se pueden utilizar, de forma cíclica, para generar la compensación para la dirección generada por un generador de direcciones para cada uno de los modos respectivos:

30

Modo de 2k:

35 0 7 5 1 8 2 6 9 3 4 *
4 8 3 2 9 0 1 5 6 7
8 3 9 0 2 1 5 7 4 6
7 0 4 8 3 6 9 1 5 2

40 Modo de 4k:

45 7 10 5 8 1 2 4 9 0 3 6 **
6 2 7 10 8 0 3 4 1 9 5
9 5 4 2 3 10 1 0 6 8 7
1 4 10 3 9 7 2 6 5 0 8

Modo de 8k:

50 5 11 3 0 10 8 6 9 2 4 1 7 *
10 8 5 4 2 9 1 0 6 7 3 11
11 6 9 8 4 7 2 1 0 10 5 3
8 3 11 7 9 1 5 6 4 0 2 10

55 Para los códigos de permutación indicados anteriormente, los dos primeros podrían utilizarse en un ciclo de dos secuencias, mientras que los cuatro podrían usarse para un ciclo de cuatro secuencias. Además, algunas secuencias adicionales de cuatro códigos de permutación, que son objeto de ciclo operativo para proporcionar la compensación en un generador de direcciones para obtener una de-correlación adecuada en los símbolos intercalados (algunos son comunes a los anteriores) se proporcionan a continuación:

60 Modo de 0.5k:

65 3 7 4 6 1 2 0 5
4 2 5 7 3 0 1 6
5 3 6 0 4 1 2 7
6 1 0 5 2 7 4 3

Modo de 2k:

0 7 5 1 8 2 6 9 3 4 *
 3 2 7 0 1 5 8 4 9 6
 5 4 8 3 2 9 0 1 5 6 7
 7 3 9 5 2 1 0 6 4 8

Modo de 4k:

10 7 10 5 8 1 2 4 9 0 3 6 **
 6 2 7 10 8 0 3 4 1 9 5
 10 3 4 1 2 7 0 6 8 5 9
 0 8 9 5 10 4 6 3 2 1 7

15 Modo de 8k:

5 11 3 0 10 8 6 9 2 4 1 7 *
 8 10 7 6 0 5 2 1 3 9 4 11
 20 11 3 6 9 2 7 4 10 5 1 0 8
 10 8 1 7 5 6 0 11 4 2 9 3

* éstas son las permutaciones en la norma de DVB-T

** éstas son las permutaciones en la norma de DVB-H.

25 Formas de realización, a modo de ejemplo, de generadores de direcciones y de los intercaladores correspondientes, para los modos de 2k, 4k y 8k se dan a conocer en la solicitud de patente europea número 04251667.4. Un generador de direcciones para el modo de 0.5k se da a conocer en nuestra solicitud de patente en el Reino Unido co-pendiente número 0722553.5.

30 Otros diversos aspectos en las características de la presente invención se definen en las reivindicaciones independientes. Varias modificaciones se pueden realizar a las formas de realización anteriormente descritas sin desviarse por ello del alcance de protección de la presente invención. En particular, la representación, a modo de ejemplo, del polinomio generador y el orden de permutación, que se han utilizado para representar aspectos de la idea inventiva no están previstos para ser limitadores y se extienden a formas equivalentes del polinomio generador y del orden de permutación.

35 Según se apreciará, el transmisor y el receptor ilustrados en la Figuras 1 y 7, respectivamente, se proporcionan a modo de ilustración solamente y no están previstos para ser limitadores en modo alguno. A modo de ejemplo, se apreciará que la posición del intercalador de símbolos y del de-intercalador con respecto, a modo de ejemplo, al intercalador de bits y al mapeador y de-mapeador se puede cambiar. Como se apreciará, el efecto del intercalador y del de-intercalador no se puede cambiar, por su posición relativa, aunque el intercalador puede intercalar símbolos de IQ en lugar de vectores de v-bits. Un cambio correspondiente se puede realizar en el receptor. En consecuencia, el intercalador y el de-intercalador pueden funcionar con diferentes tipos de datos y pueden situarse, en forma distinta, a la posición descrita en las formas de realización a modo de ejemplo.

40 Según se explicó anteriormente, los códigos de permutación y el polinomio generador del intercalador, que se han descrito con referencia a una puesta en práctica de un modo particular, pueden aplicarse igualmente a otros modos, cambiando la dirección permitida máxima predeterminada, en función del número de portadoras para ese modo operativo.

45 En conformidad con una puesta en práctica de un receptor, se incluye un aparato de procesamiento de datos utilizable para realizar un mapeado de correspondencia símbolos datos recibidos desde un número predeterminado de señales de sub-portadoras de símbolos Multiplexados por División de Frecuencias Ortogonales OFDM en un flujo continuo de datos de salida.

50 Según se indicó anteriormente, las formas de realización de la presente invención encuentran aplicación con normas DVB tales como DVB-T, DVB-T2 y DVB-H. A modo de ejemplo, formas de realización de la presente invención se pueden utilizar en un transmisor o un receptor que funcionen en conformidad con la norma DVB-T2, según se especifica en conformidad con la norma EN 302 755 de ETSI, aunque se apreciará que la presente invención no está limitada a la aplicación con la norma DVB y puede extenderse a otras normas para transmisión o recepción, fijas y móviles. En otras formas de realización, a modo de ejemplo, la presente invención encuentra aplicación con la norma de transmisión por cable conocida como DVB-C2.

65 Además de las formas de realización, a modo de ejemplo, anteriormente descritas y los aspectos y características de la idea inventiva que se definen en las reivindicaciones adjuntas, otras formas de realización pueden dar a conocer un

aparato de procesamiento de datos utilizable para efectuar la puesta en correspondencia de símbolos de entrada a comunicarse en un número predeterminado de señales de sub-portadoras para un símbolo Multiplexado por División de Frecuencias Ortogonales (OFDM). El número predeterminado de señales de sub-portadoras corresponde a un modo de modulación y los símbolos de entrada incluyen símbolos de datos impares y símbolos de datos pares. El aparato de procesamiento de datos comprende un intercalador utilizable para realizar un primer proceso de intercalado, que intercala símbolos de datos de entrada impares en las señales de sub-portadoras y un proceso de intercalado par que intercala símbolos de datos de entrada pares en las señales de sub-portadoras, estando los primeros procesos de intercalado impar y el proceso de intercalado par los que realizan las operaciones de lectura y escritura, respectivamente, de los símbolos de datos para el mapeado de correspondencia en las señales de sub-portadoras OFDM para una memoria de intercalador, siendo la salida de lectura realizada en un orden diferente que la operación de lectura a la entrada, de modo que mientras un símbolo impar es objeto de lectura desde una posición en la memoria, un símbolo par puede ser objeto de escritura en la posición que acaba de leerse desde y cuando un símbolo par es objeto de lectura desde la posición en la memoria, un siguiente símbolo impar puede ser objeto de escritura en la posición de donde acaba de leerse, con el proceso de intercalado impar realizando la lectura a la entrada y la lectura a la salida de los símbolos de datos impares desde la memoria de intercalador en conformidad con un sistema de intercalado impar y el proceso de intercalado par realizando las operaciones de lectura a la entrada y salida de símbolos de datos desde la memoria de intercalador en conformidad con un sistema de intercalado par. Cuando el modo de modulación es un modo que incluye la mitad o menos de la mitad de las señales de sub-portadoras que un número total de sub-portadoras que pueden admitirse por la memoria de intercalador, el aparato de procesamiento de datos es utilizable para asignar una parte de la memoria de intercalador al primero proceso de intercalado impar y asignar una segunda parte de la memoria del intercalador a un segundo proceso de intercalado impar que opera en conformidad con los primeros, el segundo proceso de intercalado impar que intercala los símbolos de entrada pares.

Según otra forma de realización, a modo de ejemplo, un aparato de procesamiento de datos es utilizable para la puesta en correspondencia de símbolos de entrada a comunicarse en un número predeterminado de señales de sub-portadoras de un símbolo Multiplexado por División de Frecuencias Ortogonales (OFDM). El número predeterminado de señales de sub-portadoras corresponde a un modo de modulación y los símbolos de entrada incluyen primeros símbolos de datos para la puesta en correspondencia con un primer símbolo OFDM y segundos símbolos de datos para la puesta en correspondencia con un segundo símbolo OFDM. El aparato de procesamiento de datos comprende un intercalador utilizable para realizar un proceso de intercalado impar que intercala primeros símbolos de datos de entrada en las señales de sub-portadoras y un proceso de intercalado par que intercala segundos símbolos de datos de entrada en las señales de sub-portadoras, realizando el proceso de intercalado impar la escritura de los primeros símbolos de datos de entrada en una memoria de intercalador en conformidad con un orden secuencial de los primeros símbolos de datos de entrada y la lectura, a la salida, de los primeros símbolos de datos desde una memoria de intercalador en las señales de sub-portadoras en conformidad con un orden definido por un código de permutación, la escritura por el proceso de intercalado par de los segundos símbolos de datos de entrada en una memoria de intercalador en conformidad con un orden definido por el código de permutación y la lectura, a la salida, de los segundos símbolos de datos desde la memoria de intercalador en las señales de sub-portadoras en conformidad con un orden secuencial de modo que mientras un primer símbolo de datos de entrada está siendo objeto de lectura desde una posición en la memoria de intercalador, un segundo símbolo puede ser objeto de escritura en la posición desde donde acaba de leerse y cuando un segundo símbolo es objeto de lectura desde la posición en la memoria de intercalador, un siguiente primer símbolo puede ser objeto de escritura en la posición desde donde acaba de leerse. Cuando el modo de modulación es un modo que incluye la mitad o menos de la mitad de un número de señales de sub-portadoras que un número total de sub-portadoras que pueden admitirse por la memoria de intercalador, el aparato de procesamiento de datos es utilizable para intercalar primeros y segundos símbolos de entrada en conformidad con el proceso de intercalado impar.

Otra forma de realización, a modo de ejemplo, puede dar a conocer un método de puesta en correspondencia de los símbolos de entrada a comunicarse en un número predeterminado de señales de sub-portadoras de un símbolo Multiplexado por División de Frecuencias Ortogonales (OFDM). El método comprende la puesta en correspondencia de los primeros símbolos de datos con un primer símbolo OFDM y la puesta en correspondencia de los segundos símbolos de datos en un segundo símbolo OFDM.

REIVINDICACIONES

1. Un receptor dispuesto en operación para la recuperación de bits de datos a partir de símbolos de datos recibidos a partir de un número predeterminado de señales sub-portadoras de símbolos Multiplexados por División de Frecuencias Ortogonales (OFDM) y para formar un flujo de bits de salida, siendo el número predeterminado de señales sub-portadoras de los símbolos OFDM que se determina en función de uno entre una pluralidad de modos de funcionamiento, comprendiendo dicho receptor:
- 5 un desintercalador de símbolos (514) dispuesto en operación para la recuperación de primeros conjuntos de símbolos de datos a partir de primeros símbolos OFDM en función de un proceso de intercalado impar y de segundos conjuntos de símbolos de datos a partir de segundos símbolos OFDM en función de un proceso de intercalado par y para formar un flujo de símbolos de salida a partir de los primeros y segundos conjuntos de símbolos de datos,
- 10 una unidad demapeadora (52) dispuesta en operación para generar, a partir de los símbolos de datos del flujo de símbolos de salida, los bits de datos codificados del Control de Paridad de Baja Densidad (LDPC), con paridad intercalada, convirtiendo cada uno de los símbolos de datos del flujo de símbolos de salida que representa un símbolo de modulación de las señales sub-portadoras de OFDM en bits de datos en función de un sistema de modulación,
- 15 un permutador inverso (53, 55) adaptado para realizar un proceso de permutación inversa para efectuar una inversión de un proceso de permutación aplicado a los bits de datos de LDPC, de paridad intercalada, codificados, para permutar los bits de datos de LDPC codificados, de modo que una pluralidad de los bits de datos de LDPC codificados que corresponda a un valor de 1 en una fila arbitraria de una matriz de información correspondiente a bits de información de un código de LDPC, que se utilizó para codificar los bits de datos, no se incorpore en el mismo símbolo y
- 20 un decodificador de LDPC (56) adaptado para realizar una decodificación de LDPC sobre los bits de datos de LDPC codificados, en donde el proceso de permutación inversa se haya realizado para formar los bits de datos de salida, en donde el proceso de intercalado impar incluye:
- 25 la escritura de los primeros conjuntos de símbolos de datos recuperados a partir de las señales sub-portadoras de los primeros símbolos OFDM en una memoria del intercalador de símbolos en función de un orden definido por un código de permutación y
- 30 la lectura de los primeros conjuntos de símbolos de datos a partir de la memoria del intercalador de símbolos (100) en función de un orden secuencial en el flujo de datos de salida y
- 35 el proceso de intercalado par incluye
- la escritura de los segundos conjuntos de símbolos de datos recuperados a partir de las señales sub-portadoras de los segundos símbolos OFDM en la memoria del intercalador de símbolos (540) en función de un orden secuencial y
- 40 la lectura los segundos conjuntos de símbolos de datos a partir de la memoria del intercalador de símbolos (540) en función de un orden definido por el código de permutación en el flujo de datos de salida, de tal modo que mientras los símbolos de datos desde el primer conjunto esté siendo objeto de lectura a partir de posiciones en la memoria del intercalador de símbolos (540), los símbolos de datos a partir del segundo conjunto puedan ser objeto de escritura en las
- 45 posiciones desde donde acaban de leerse y cuando los símbolos de datos desde el segundo conjunto estén siendo leídos a partir de las posiciones en la memoria del intercalador de símbolos (540), los símbolos de datos a partir de un primer conjunto siguiente puedan ser objeto de escritura en las posiciones desde las que acaban de leerse, en donde
- 50 cuando en una de la pluralidad de modos de funcionamiento en donde el número de sub-portadoras por símbolo de OFDM proporcione la mitad, o menos de la mitad, de un número máximo de sub-portadoras en los símbolos OFDM de cualquiera de los modos de funcionamiento, el aparato de procesamiento de datos sea utilizable para el desintercalado de los símbolos de datos en función de solamente el proceso de intercalado impar.
2. Un receptor según la reivindicación 1, en donde el proceso de permutación inversa realizado por el permutador inverso (53, 55) sobre los datos de LDPC codificados tiene un efecto de inversión de una permutación de los bits de datos codificados, que se realizó por un permutador correspondiente en un transmisor, con el permutador correspondiente habiendo realizado el intercalado de paridad en los bits de datos de LDPC codificados, que se obtuvieron realizando la codificación de LDPC en función de una matriz de control de paridad de un código de LDPC, incluyendo la matriz de control de paridad una matriz de paridad correspondiente a los bits de paridad del código de LDPC, presentando la matriz de paridad una estructura escalonada, de modo que un bit de paridad de los bits de datos de LDPC codificados se intercale para una posición de bit de paridad diferente y realizando luego un proceso de permutación sobre los bits de datos codificados de LDPC para permutar los bits de código de los bits de datos codificados de LDPC, de modo que una pluralidad de los bits de datos codificados de LDPC del código de LDPC que corresponda a un valor de 1 en una fila arbitraria de una matriz de información, correspondiente a los bits de información de los bits codificados de LDPC, no se incorpore en el mismo símbolo y en donde el decodificador de LDPC (56) realice la decodificación de LDPC del código de LDPC, en donde se haya realizado el proceso de permutación inversa y no se
- 55
- 60
- 65

haya realizado el desintercalado de paridad correspondiente al intercalado de paridad, utilizando una matriz de control de paridad convertida obtenida efectuando al menos una permutación de columnas correspondiente al intercalado de paridad en la matriz de control de paridad.

- 5 **3.** Un receptor según la reivindicación 1 o 2, en donde el desintercalador de símbolos (514) incluye un controlador, un generador de direcciones (102) y la memoria del intercalador de símbolos (540), siendo el controlador (544, 546) utilizable para controlar el generador de direcciones (102) para generar direcciones, durante el proceso de intercalado impar para realizar la escritura de los primeros y segundos conjuntos de símbolos de datos a partir de las señales sub-portadoras de los primeros y segundos símbolos OFDM en la memoria del intercalador de símbolos, en función de un orden definido por el código de permutación, antes de la lectura de los segundos conjuntos de símbolos de datos, desde la memoria del intercalador de símbolos, en función del orden secuencial en el flujo de datos de salida.
- 10
- 4.** Un receptor según la reivindicación 3, en donde el generador de direcciones comprende:
- 15 un registro de desplazamiento de realimentación lineal (200) que comprende un número predeterminado de etapas de registro y siendo utilizable para generar una secuencia de bits pseudo-aleatoria en conformidad con un polinomio generador,
- 20 un circuito de permutación (210) utilizable para recibir el contenido de las etapas del registro de desplazamiento y permutar los bits presentes en las etapas del registro, en conformidad con el código de permutación con el fin de formar las direcciones de una de las portadoras de OFDM y
- 25 una unidad de control (224) utilizable, en combinación con un circuito de control de dirección (216), para regenerar una dirección cuando una dirección generada supera una dirección válida máxima predeterminada, estando la dirección válida máxima predeterminada definida en conformidad con el modo de funcionamiento.
- 5.** Un receptor según cualquiera de las reivindicaciones precedentes, en donde una capacidad mínima de la memoria del intercalador de símbolos (540) puede proporcionarse en función del número máximo de símbolos de datos que se pueden transmitir en las sub-portadoras de los símbolos OFDM que estén disponibles para transmitir los símbolos de datos en cualquiera de los modos de funcionamiento.
- 30
- 6.** Un receptor según la reivindicación 5, en donde cuando se utiliza en el modo de funcionamiento que proporciona el número máximo de sub-portadoras por símbolo de OFDM, el desintercalador es utilizable para usar la memoria del intercalador de símbolos disponible (540) en función del proceso de intercalado impar y de los procesos de intercalado par al efecto de la lectura de símbolos de datos desde las posiciones en la memoria del intercalador de símbolos (540) y la escritura de los símbolos de datos desde las posiciones de donde acaban de leerse y cuando se opera en cualquier otro modo en el que el número de sub-portadoras es una mitad, o menos de una mitad, del número de sub-portadoras para transmitir los símbolos de datos por símbolo OFDM, siendo el desintercalador de símbolos (514) utilizable en el proceso de intercalado impar para leer los primeros conjuntos de símbolos de datos a partir de las primeras posiciones en la memoria del intercalador de símbolos (540) y para la escritura de los segundos conjuntos de símbolos de datos en la memoria del intercalador de símbolos (540) en las segundas posiciones, siendo las segundas posiciones diferentes con respecto a las primeras posiciones.
- 35
- 40
- 7.** Un receptor según la reivindicación 6, en donde el modo de funcionamiento que proporciona el número máximo de sub-portadoras por símbolo OFDM es un modo de 32K y los otros modos incluyen uno o más de los modos de 2K, 4K, 8K y 16K.
- 45
- 8.** Un receptor según una cualquiera de las reivindicaciones 3 a 7, en donde el aparato de procesamiento de datos es utilizable para cambiar el código de permutación que se utiliza para formar las direcciones de un símbolo OFDM a otro.
- 50
- 9.** Un receptor según la reivindicación 8, en donde los bits de datos se reciben a partir de los símbolos OFDM en función de una norma de Difusión de Vídeo Digital, tal como la norma de Difusión de Vídeo Digital-Terrestre, la norma de Difusión de Vídeo Digital-Portátil, la norma de Difusión de Vídeo Digital-Terrestre 2 o la norma de Difusión de Vídeo Digital por Cable 2.
- 55
- 10.** Un método de recepción de bits de datos a partir de símbolos de datos recibidos desde un número predeterminado de señales sub-portadoras de símbolos Multiplexados por División de Frecuencias Ortogonales (OFDM) para formar un flujo de bits de salida, siendo el número predeterminado de señales sub-portadoras de los símbolos OFDM determinado en función de uno de entre una pluralidad de modos de funcionamiento, comprendiendo dicho método:
- 60 la recuperación de los primeros conjuntos de símbolos de datos a partir de primeros símbolos OFDM en función de un proceso de intercalado impar y segundos conjuntos de símbolos de datos a partir de segundos símbolos OFDM en función de un proceso de intercalado par,
- 65 la formación de un flujo de símbolos de salida a partir de los primeros y segundos conjuntos de símbolos de datos,

la generación, a partir de los símbolos de datos del flujo de símbolos de salida, de bits de datos codificados del Control de Paridad de Baja Densidad (LDPC), con paridad intercalada, convirtiendo cada símbolo de datos representado por un símbolo de modulación de las señales sub-portadoras de OFDM en bits de datos codificados intercalados en conformidad con un sistema de modulación,

5 la realización de un proceso de permutación inversa para efectuar una inversión de un proceso de permutación aplicado a los bits de datos de LDPC, con paridad intercalada, codificados para permutar los bits de datos codificados de LDPC, de modo que una pluralidad de los bits de datos codificados de LDPC que corresponda a un valor de 1 en un fila arbitraria de una matriz de información, correspondiente a los bits de información del código de LDPC, no se incorpore en el mismo símbolo y

la realización de una decodificación de LDPC sobre los bits de datos de LDPC codificados, en donde se haya realizado el proceso de permutación inversa para formar los bits de datos de salida, en donde el proceso de intercalado impar incluye

15 la escritura de los primeros conjuntos de símbolos de datos recuperados a partir de las señales sub-portadoras de los primeros símbolos OFDM en una memoria del intercalador de símbolos (540) en función de un orden definido por un código de permutación y

20 la lectura de los primeros conjuntos de símbolos de datos desde la memoria del intercalador de símbolos (540), en función de un orden secuencial en el flujo de datos de salida y

el proceso de intercalado par incluye:

25 la escritura de los segundos conjuntos de símbolos de datos recuperados a partir de las señales sub-portadoras de los segundos símbolos OFDM en la memoria del intercalador de símbolos (540) en función de un orden secuencial y

30 la lectura de los segundos conjuntos de símbolos de datos a partir de la memoria del intercalador de símbolos (540) en función de un orden definido por el código de permutación en el flujo de datos de salida, de tal modo que mientras los símbolos de datos desde el primer conjunto esté siendo objeto de lectura a partir de posiciones en la memoria del intercalador de símbolos (540), los símbolos de datos a partir del segundo conjunto puedan ser objeto de escritura en las posiciones desde donde acaban de leerse y cuando los símbolos de datos desde el segundo conjunto estén siendo leídos a partir de las posiciones en la memoria del intercalador de símbolos (540), los símbolos de datos a partir de un primer conjunto siguiente puedan ser objeto de escritura en las posiciones desde las que acaban de leerse, en donde

35 cuando en una de la pluralidad de modos de funcionamiento en donde el número de sub-portadoras por símbolo de OFDM proporcione la mitad, o menos de la mitad, de un número máximo de sub-portadoras en los símbolos OFDM de cualquiera de los modos de funcionamiento, el intercalado comprende el intercalado de los símbolos de datos en función de solamente el proceso de intercalado impar.

40 **11.** Un método según la reivindicación 10, en donde la realización del proceso de permutación inversa sobre los datos de LDPC intercalados codificados tiene un efecto de inversión de una permutación de los bits codificados de LDPC, de paridad intercalada, que se obtuvieron realizando la codificación de LDPC en función de una matriz de control de paridad del código de LDPC, incluyendo la matriz de control de paridad a una matriz de paridad correspondiente a los bits de paridad de un código de LDPC, presentando la matriz de paridad una estructura escalonada, de modo que un bit de paridad del código de LDPC se intercale para una posición de bit de paridad diferente y realizando luego un proceso de permutación sobre los bits de datos codificados de LDPC para permutar los bits de código de los bits de datos codificados de LDPC, de modo que una pluralidad de los bits de código de los bits de datos codificados de LDPC que correspondan a un valor de 1, en una fila arbitraria de una matriz de información correspondiente a los bits de información del código LDPC, no se incorpore en el mismo símbolo y en donde la decodificación de LDPC de los bits de LDPC codificados y en donde se haya realizado el proceso de permutación inversa y no se haya realizado el desintercalado de paridad correspondiente al intercalado de paridad, incluye la utilización de una matriz de control de paridad convertida obtenida efectuando al menos una permutación de columnas correspondiente al intercalado de paridad en la matriz de control de paridad.

55 **12.** Un método según la reivindicación 10 u 11, en donde el intercalado impar incluye:

60 la generación de direcciones utilizando un generador de direcciones (542) basado en el código de permutación para la lectura de los primeros y segundos conjuntos de símbolos de datos recuperados a partir de las señales sub-portadoras de los primeros y segundos símbolos OFDM en la memoria del intercalador de símbolos (540), antes de la lectura de los símbolos de datos en el flujo de símbolo de salida en función de un orden secuencial.

13. Un método según la reivindicación 12, en donde la generación de las direcciones utilizando el generador de direcciones (542) comprende:

65 la generación de una secuencia bits pseudo-aleatoria utilizando un registro de desplazamiento de realimentación lineal (200) que comprende un número predeterminado de etapas del registro y un polinomio generador,

la permutación de los bits presentes en las etapas del registro, en conformidad con el código de permutación, con el fin de formar las direcciones de una de las sub-portadoras de OFDM y

- 5 la regeneración de una dirección cuando una dirección generada sobrepasa una dirección válida máxima predeterminada, estando definida la dirección válida máxima predeterminada en conformidad con el modo de funcionamiento.
- 10 **14.** Un método según cualquiera de las reivindicaciones 10 a 13, en donde una capacidad mínima de la memoria del intercalador de símbolos (540) puede proporcionarse en función del número máximo de símbolos de datos que se puedan transmitir en las sub-portadoras de los símbolos OFDM que estén disponibles para transmitir los símbolos de datos en cualquiera de los modos de funcionamiento.
- 15 **15.** Un método según la reivindicación 14, en donde el desintercalado incluye:
- 20 cuando se opera en el modo de funcionamiento que proporciona el número máximo de sub-portadoras por símbolo OFDM, el uso de la memoria del intercalador de símbolos disponible (540) en función del proceso de intercalado impar y del proceso de intercalado par al efecto de la lectura de símbolos de datos desde las posiciones en la memoria del intercalador de símbolos (540) y la escritura de símbolos de datos en la memoria del intercalador de símbolos (540) desde las posiciones de donde acaban de leerse y
- 25 cuando se opera en cualquier otro modo en el que el número de sub-portadoras es una mitad, o menos de una mitad, del número de sub-portadoras para transmitir los símbolos de datos por símbolo OFDM, el desintercalado según el proceso de intercalado impar para leer los primeros conjuntos de símbolos de datos a partir de las primeras posiciones en la memoria del intercalador y para la escritura de los segundos conjuntos de símbolos de datos en la memoria del intercalador en segundas posiciones, siendo las segundas posiciones diferentes con respecto a las primeras posiciones.
- 30 **16.** Un método según la reivindicación 15, en donde el modo de funcionamiento que proporciona el número máximo de sub-portadoras por símbolo OFDM es un modo de 32K y los otros modos incluyen uno o más de los modos de 2K, 4K, 8K y 16K.
- 17.** Un método según una cualquiera de las reivindicaciones 10 a 16, que comprende la modificación del código de permutación para formar las direcciones desde un símbolo OFDM a otro.
- 35 **18.** Un método según cualquiera de las reivindicaciones 10 a 17, el método comprendiendo recibir los símbolos OFDM, en conformidad con una norma de Difusión de Vídeo Digital tal como la norma de Difusión de Vídeo Digital Terrestre, la norma de Difusión de Vídeo Digital Portátil, la norma de Difusión de Vídeo Digital Terrestre 2 o la norma de Difusión de Vídeo Digital por Cable 2.

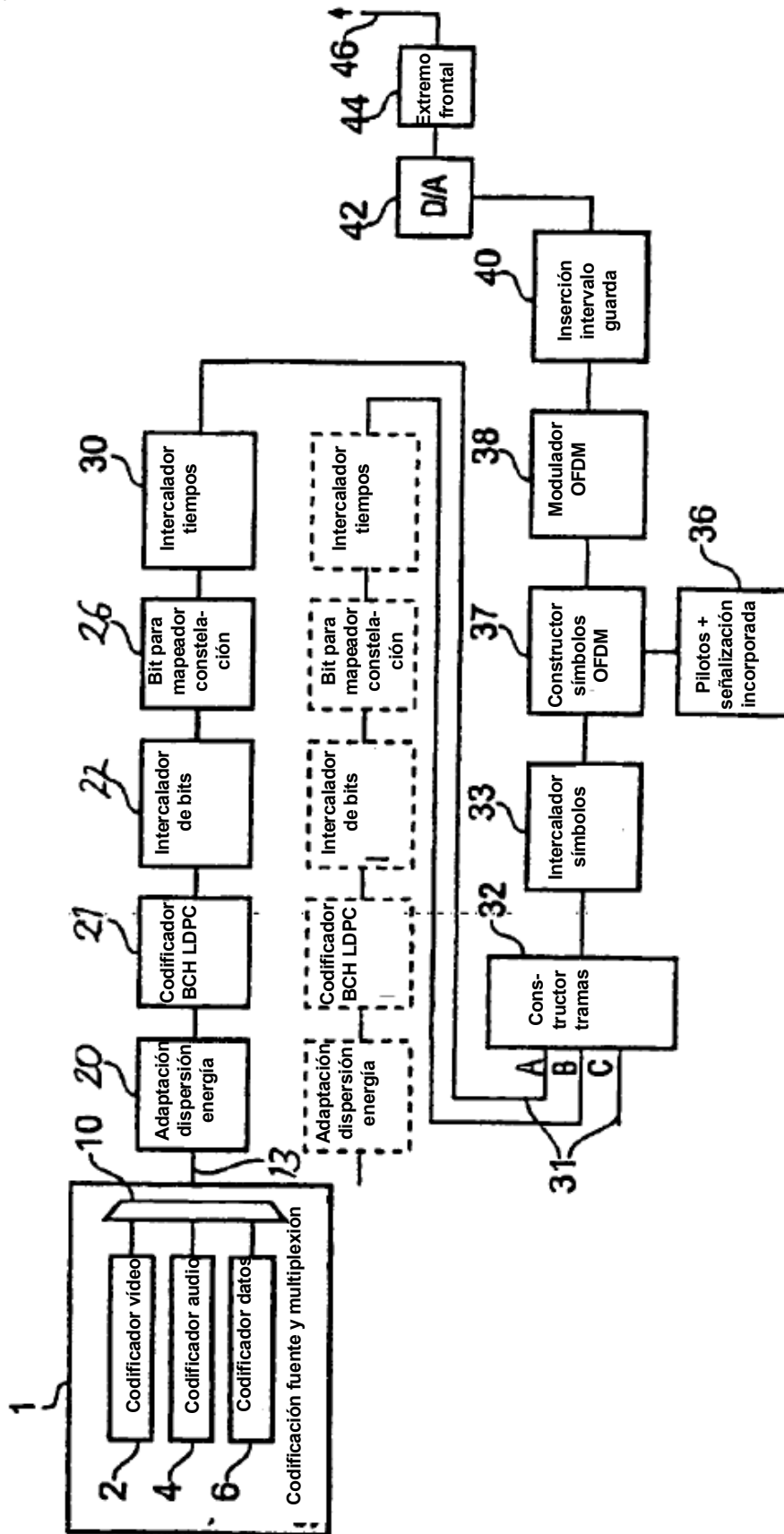


FIG. 1

FIG. 2

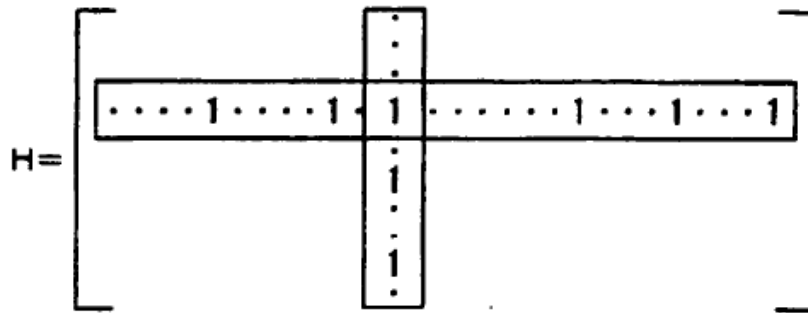


FIG. 3

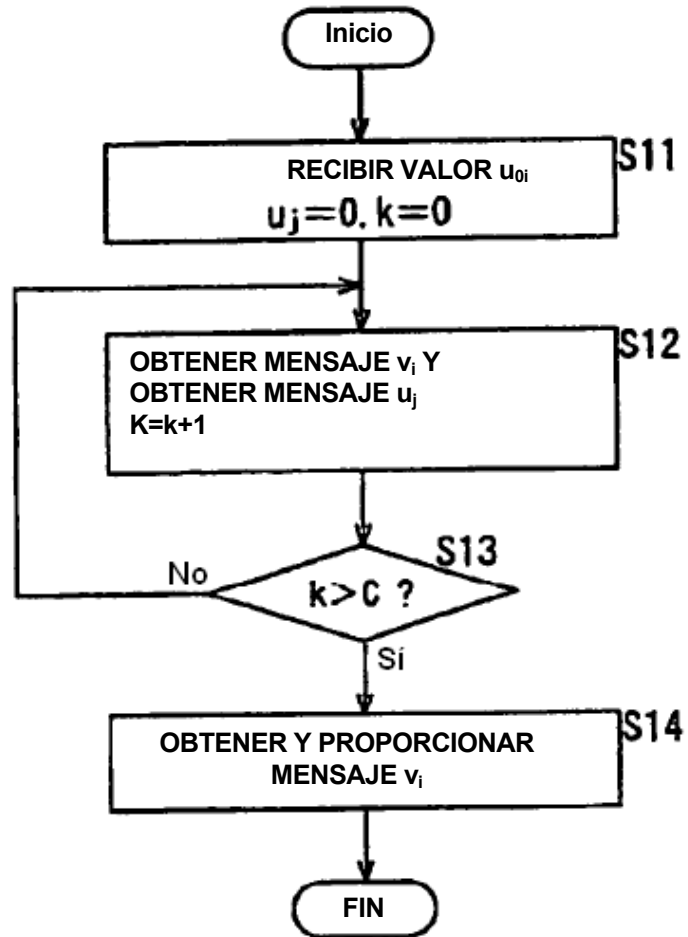


FIG. 4

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

FIG. 5

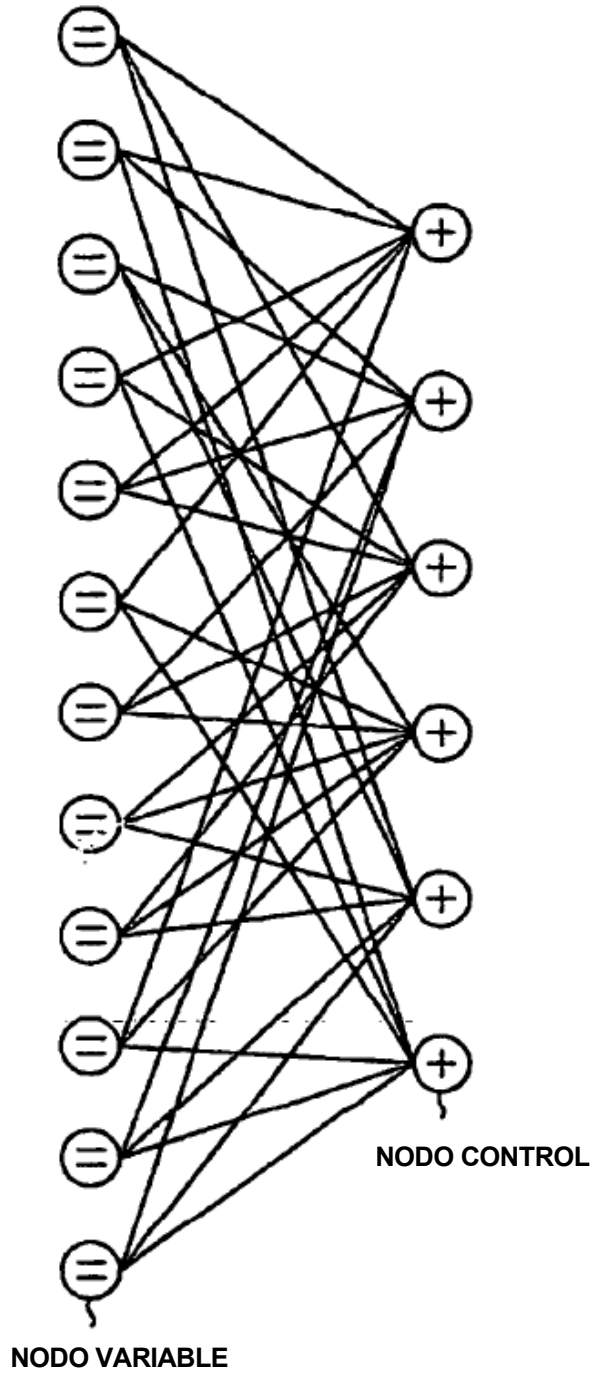


FIG. 6

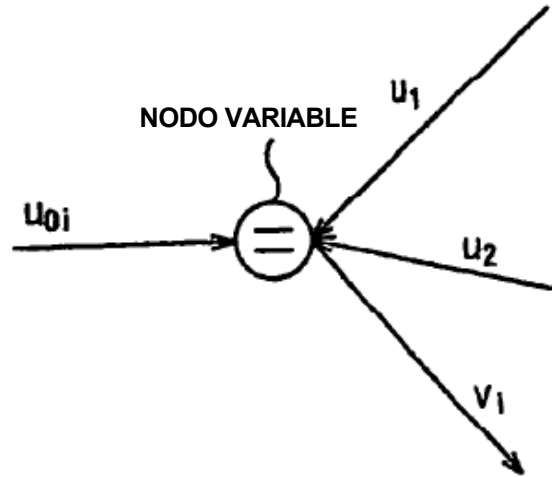


FIG. 7

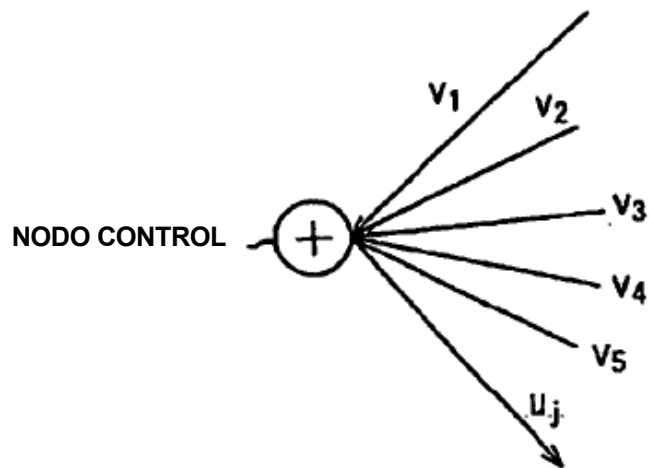


FIG. 8

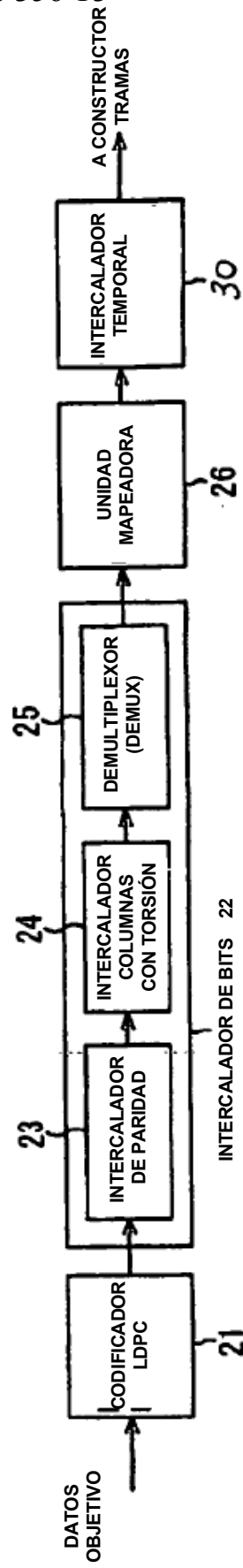


FIG. 9

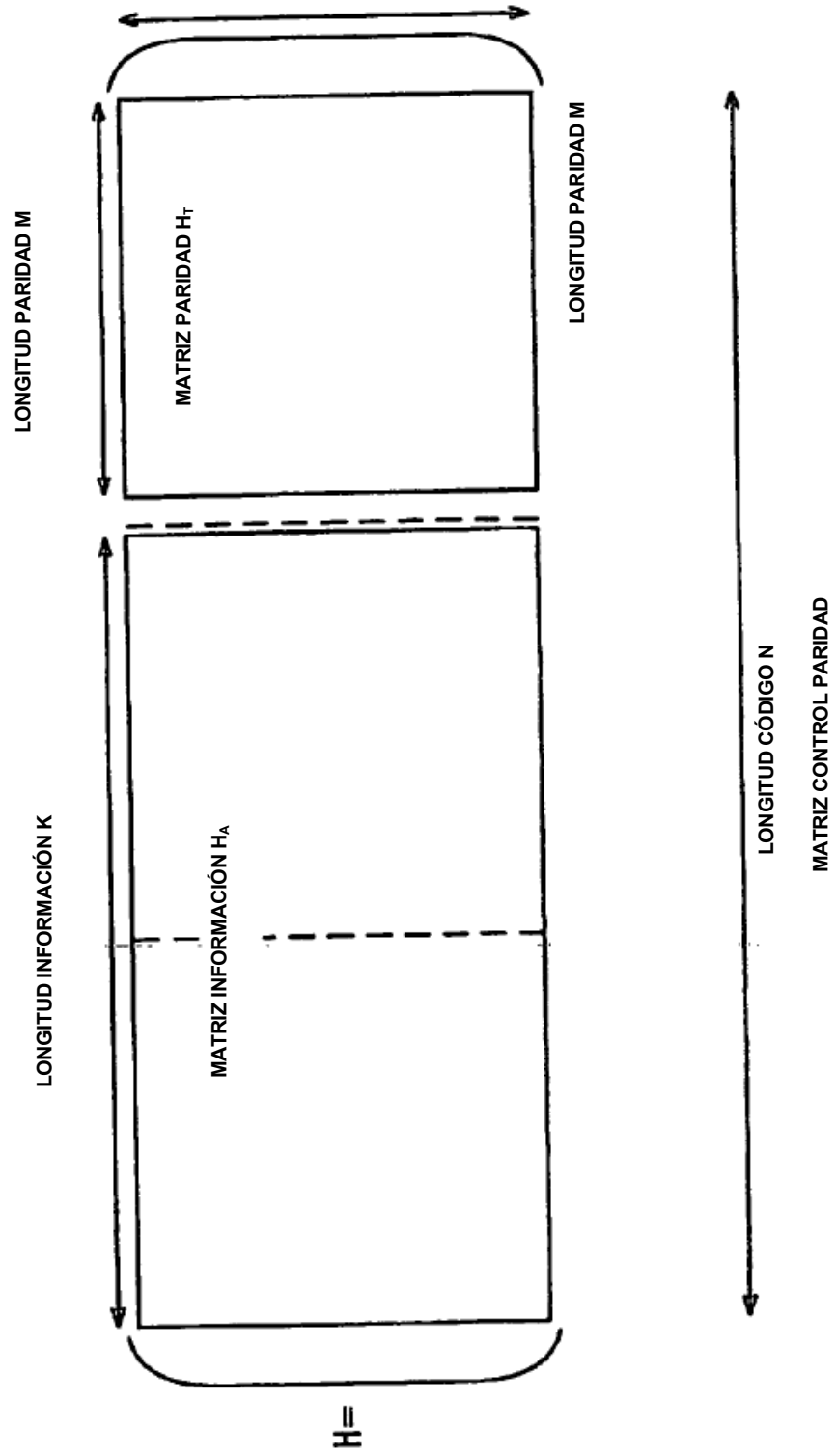


FIG. 10

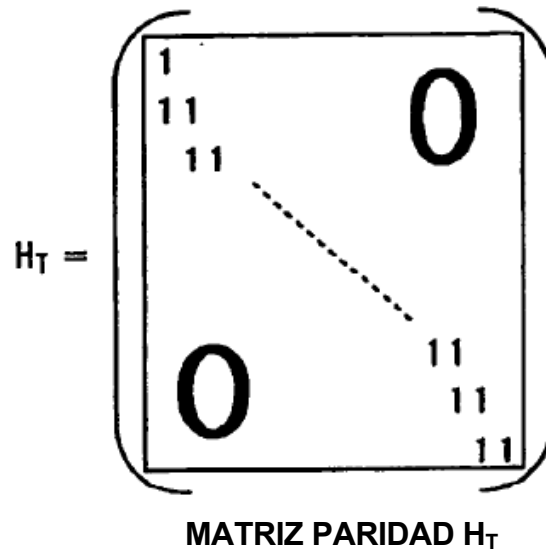
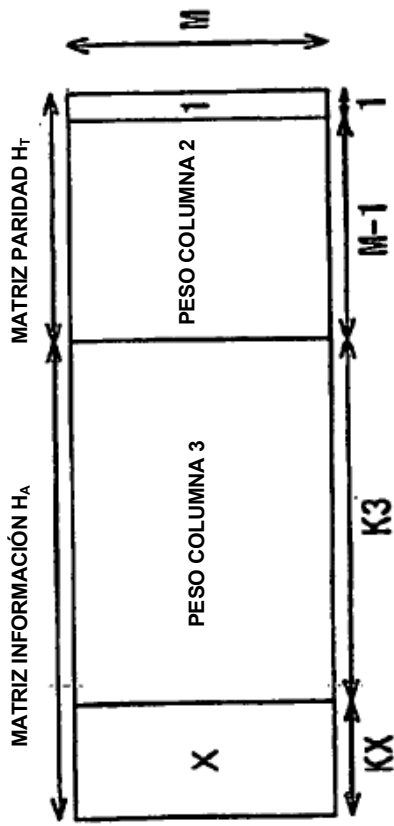


FIG. 11



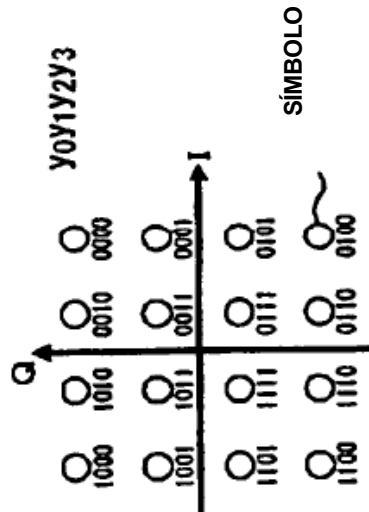
A MATRIZ CONTROL PARIDAD

TASA CÓDIGO NOMINAL	N=64800				N=16200			
	X	KX	K3	M	x	KX	K3	M
1/4	12	5400	10800	48600	12	1440	1800	12960
1/3	12	7200	14400	43200	12	1800	3600	10800
2/5	12	8640	17280	38880	12	2160	4320	9720
1/2	8	12960	19440	32400	8	1800	5400	9000
3/5	12	12960	25920	25920	12	3240	6480	6480
2/3	13	4320	38880	21600	13	1080	9720	5400
3/4	12	5400	43200	16200	12	360	11520	4320
4/5	11	6480	45360	12960	-	0	12600	3600
5/6	13	5400	48600	10800	13	360	12960	2880
8/9	4	7200	50400	7200	4	1800	12600	1800
9/10	4	6480	51840	6480	---	---	---	---

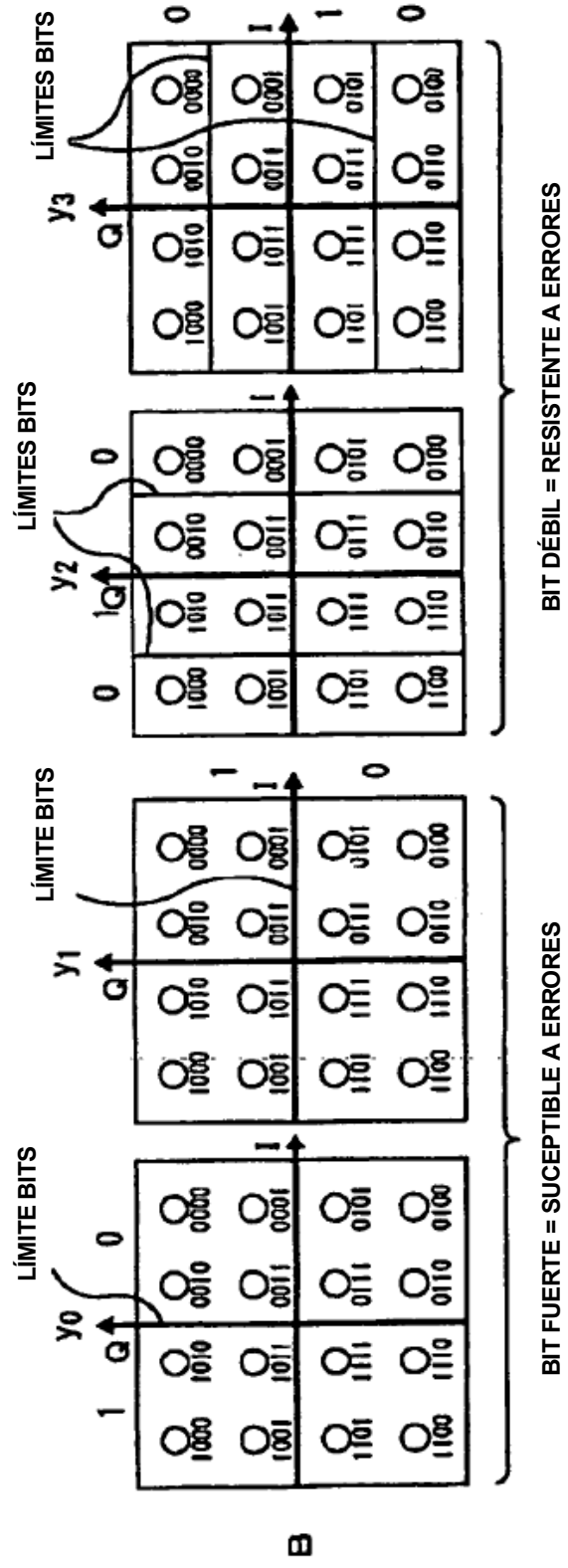
B

EL NÚMERO DE COLUMNAS DE CADA PESO DE PONDERACIÓN DE COLUMNA

FIG. 12

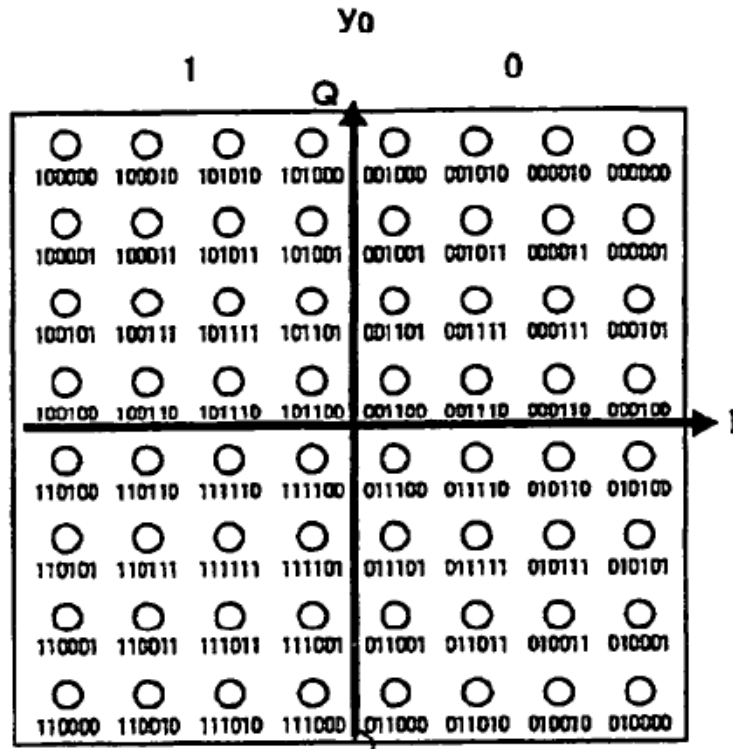


A

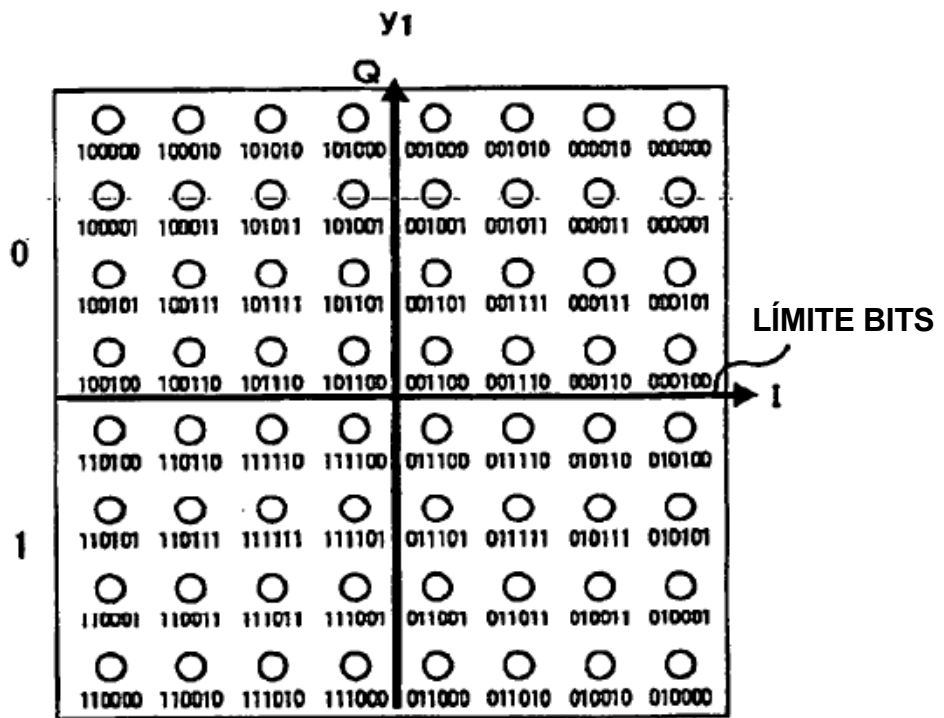


B

FIG. 13



LÍMITE BITS



LÍMITE BITS

FIG. 14

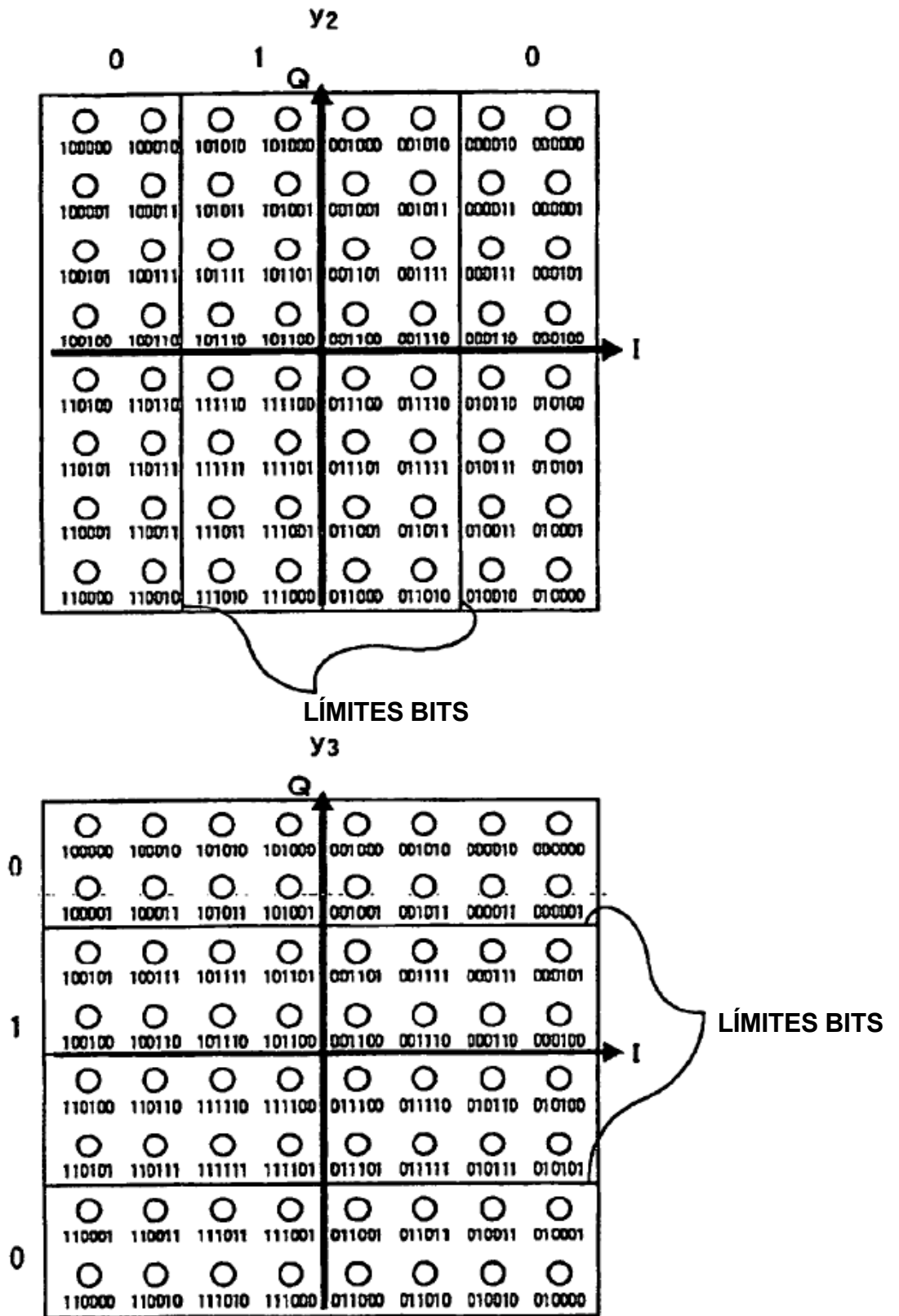
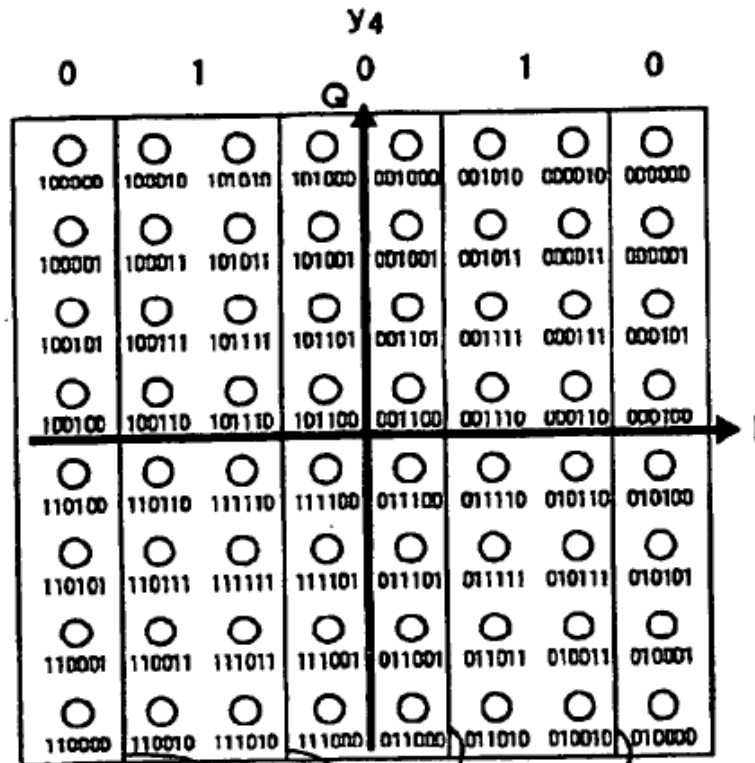
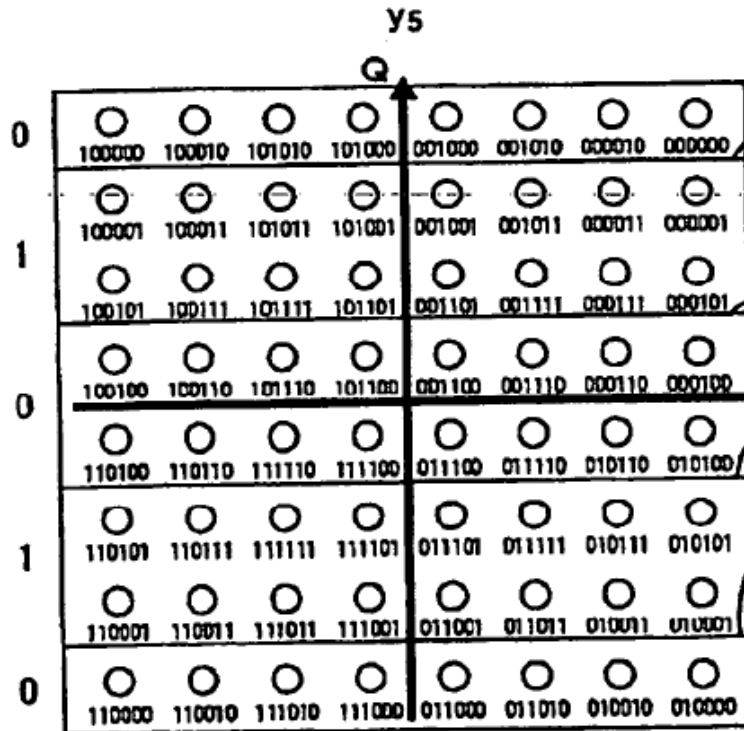


FIG. 15



LÍMITES BITS



LÍMITES BITS

FIG. 16

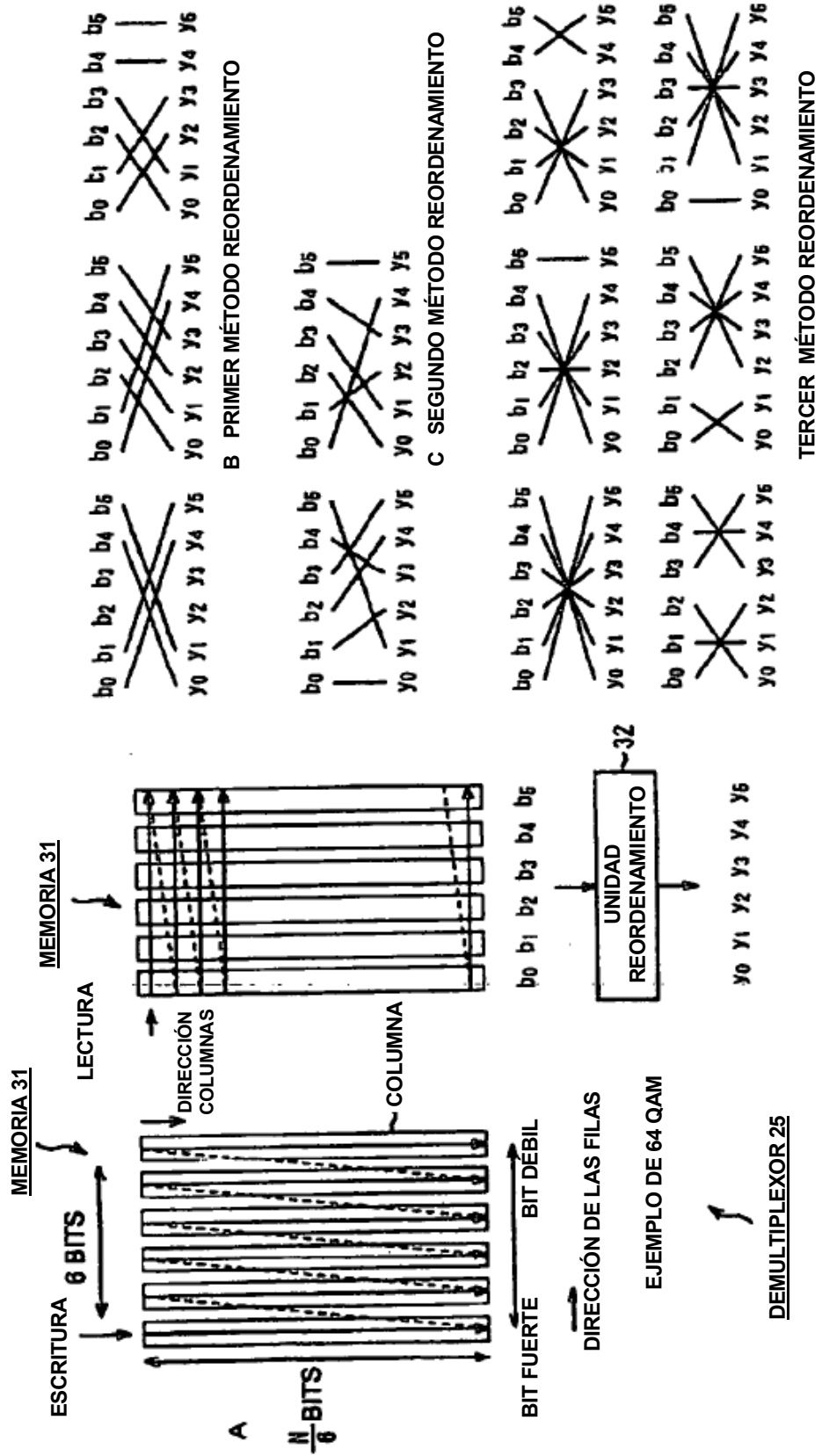


FIG. 17

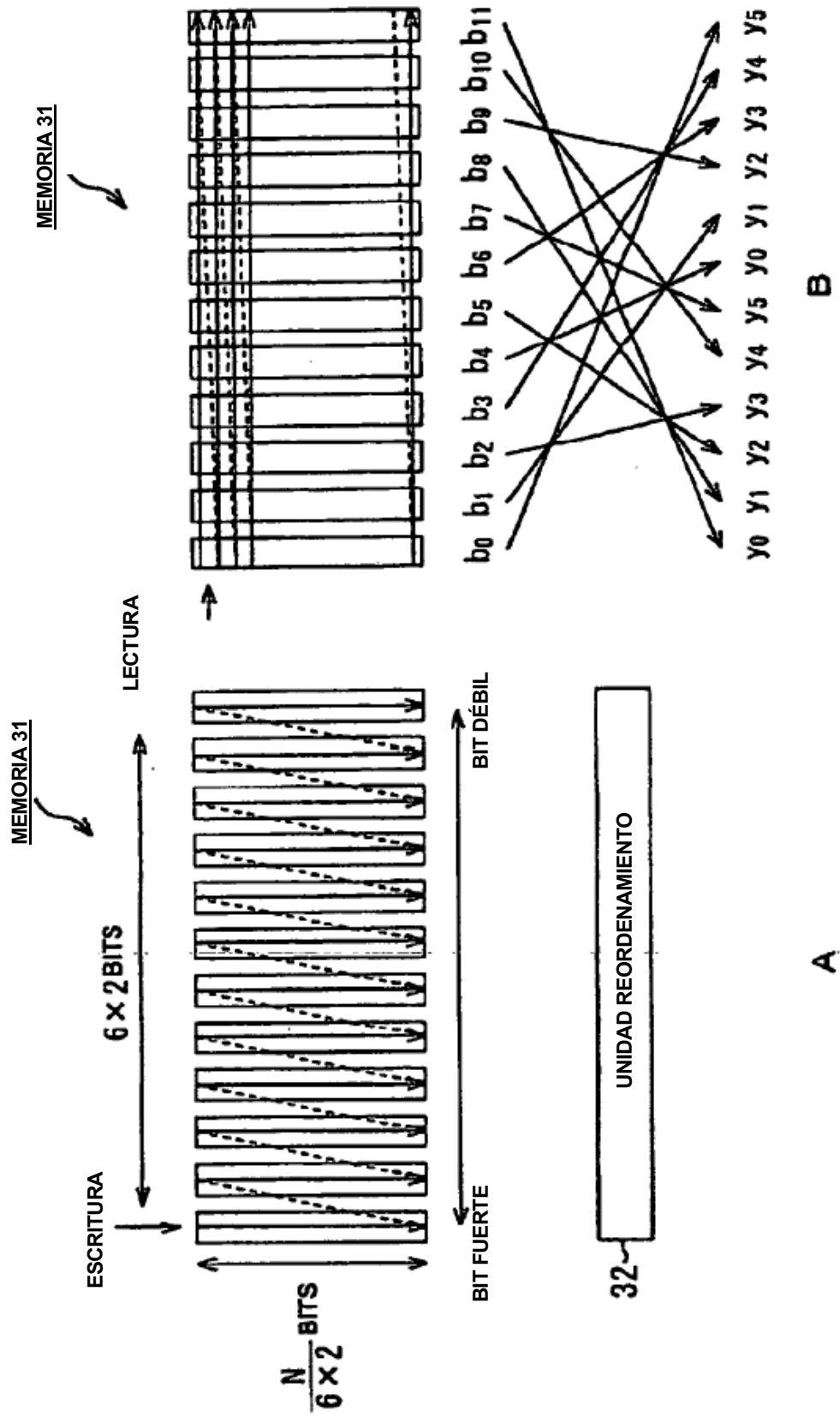


FIG. 18

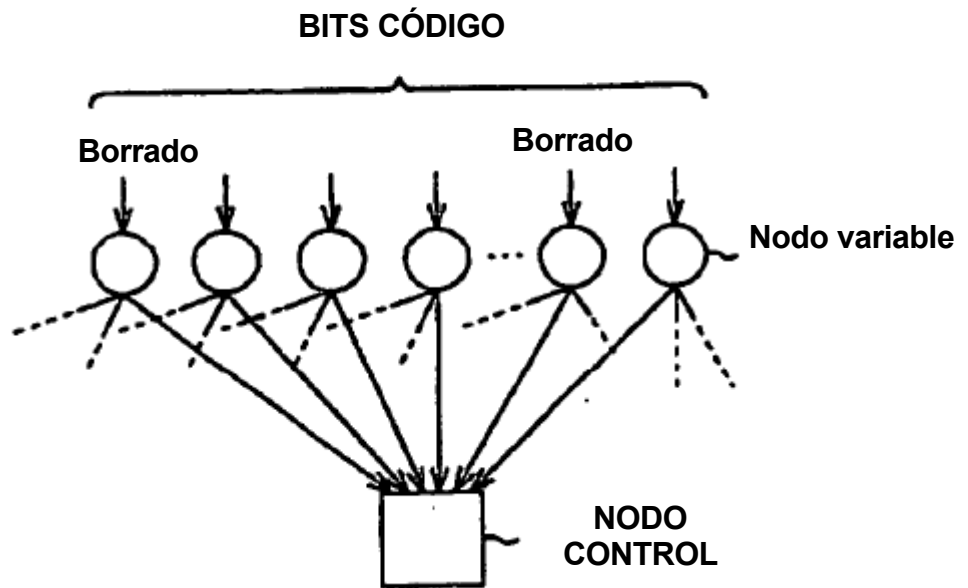
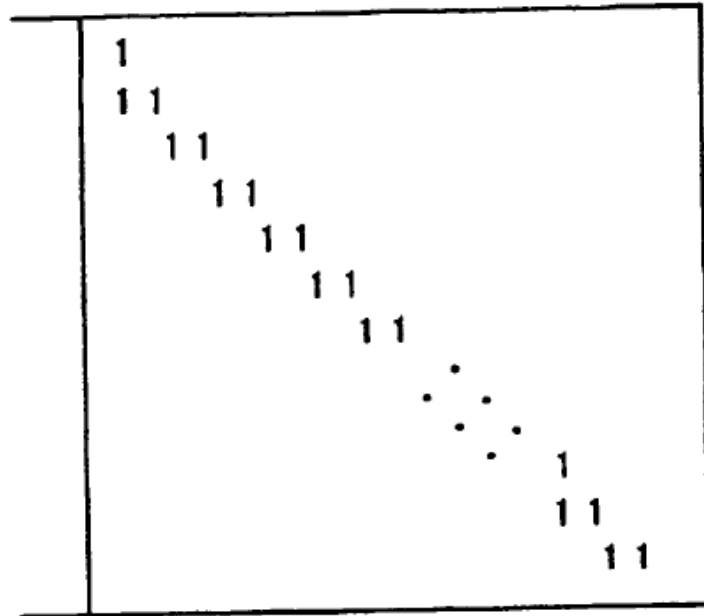


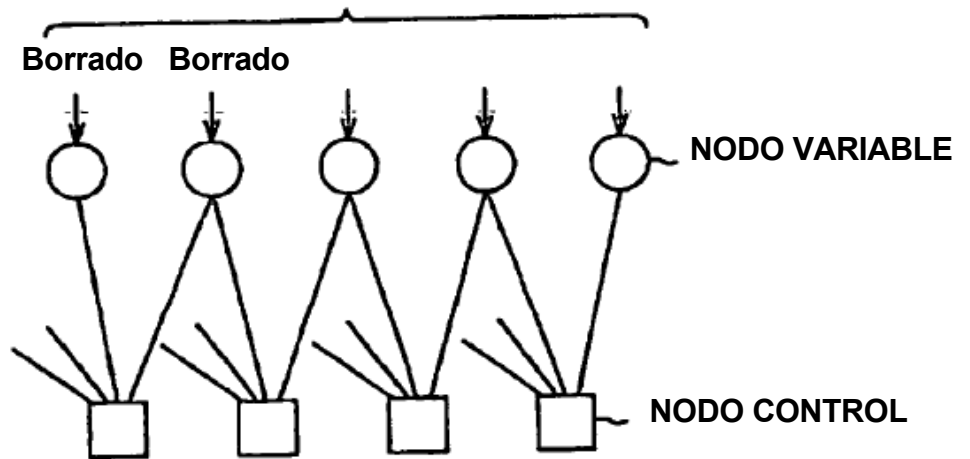
FIG. 19



ESTRUCTURA ESCALONADA DE MATRIZ DE PARIDAD

A

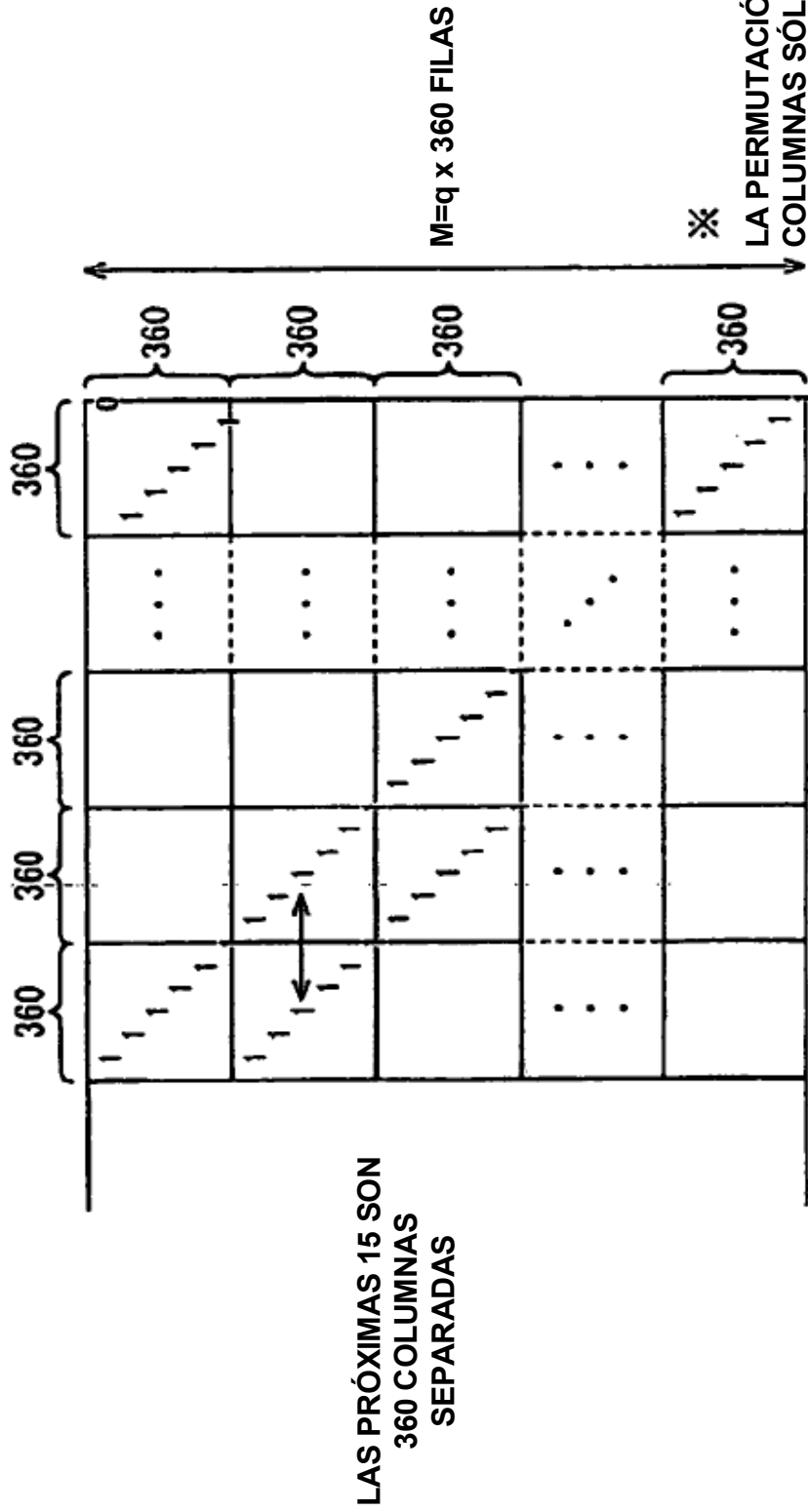
BITS CÓDIGO



PARTE DE GRÁFICO DE TANNER PRESENTANDO ESTRUCTURA ESCALONADA

B

FIG. 20



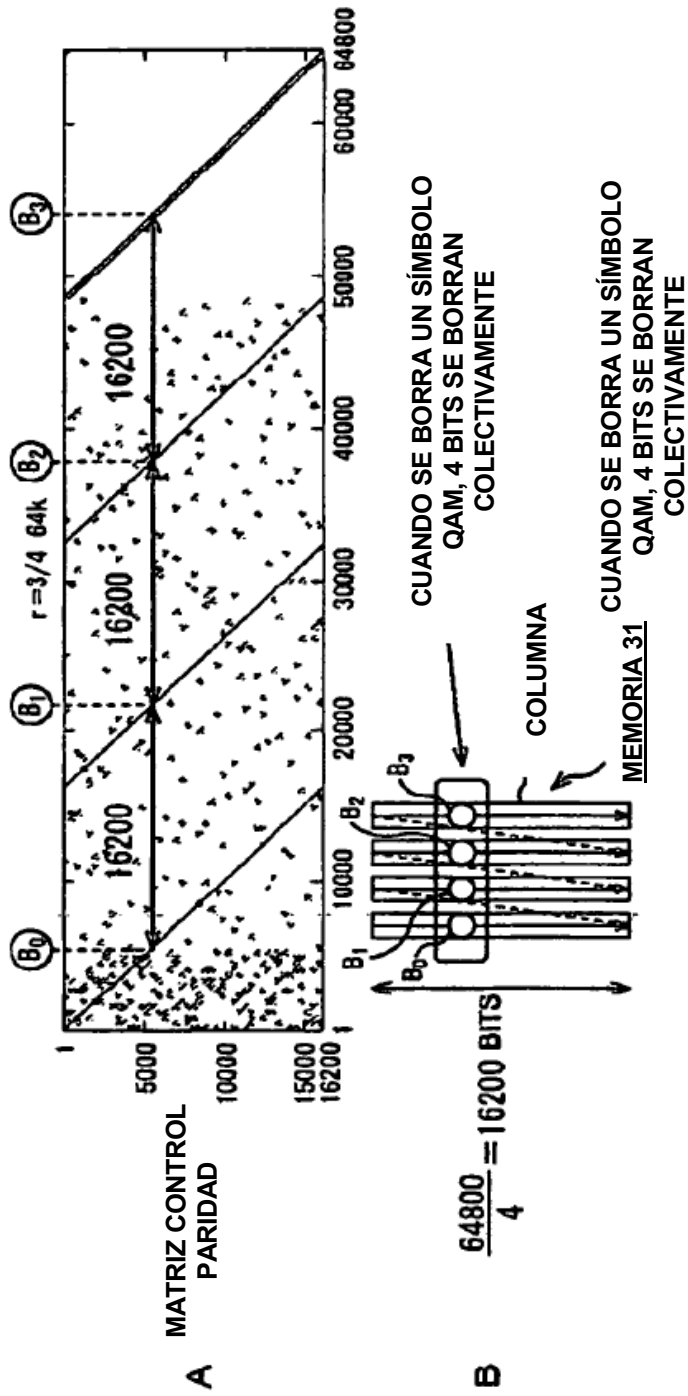
LAS PRÓXIMAS 15 SON
360 COLUMNAS
SEPARADAS

MATRIZ DE PARIDAD

LOS BITS DE PARIDAD SE HACEN RESISTENTES A ERRORES DE
RÁFAGAS SOLAMENTE CON ESTA DISPOSICIÓN OPERATIVA

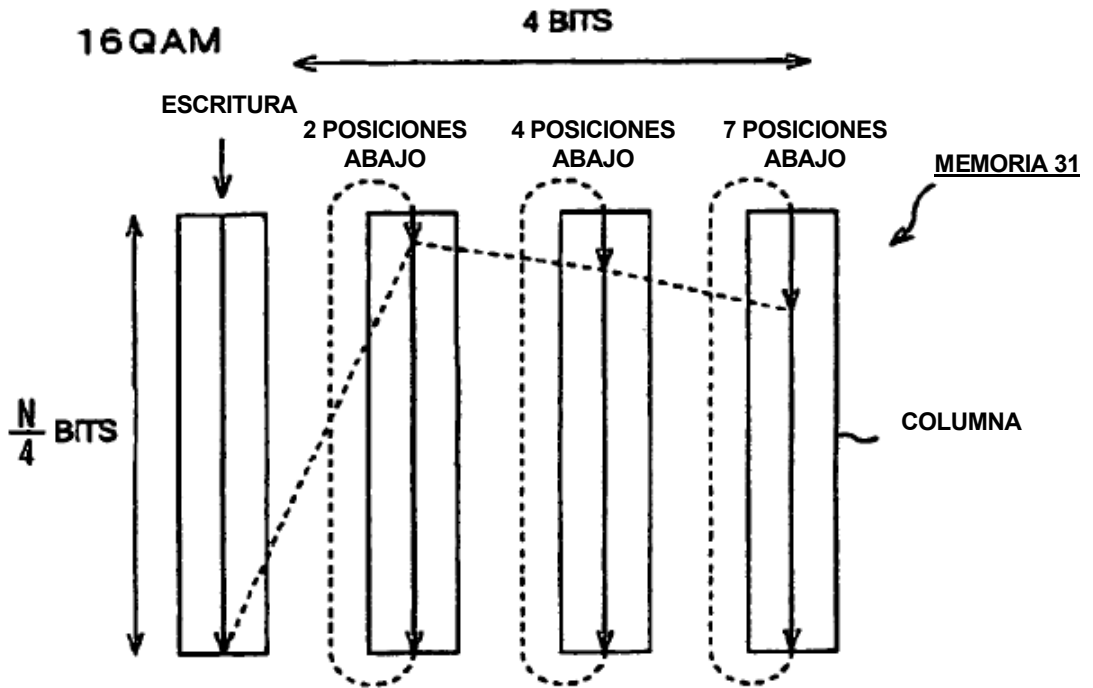
FIG. 21

16QAM, $r=3/4$ (DESPUÉS DE INTERCALADO PARIDAD)



ESTE INTERCALADO ES INCONVENIENTE EN CANALES CON BORRADOS OPERATIVOS

FIG. 22



LOS BITS DE CÓDIGO PERTENECIENTES AL MISMO NODO DE CONTROL NO SON INCORPORADOS EN EL MISMO SÍMBOLO QAM PARA TODOS LOS 11 CÓDIGOS DE 64K

FIG. 23

		POSICIONES INICIO ESCRITURA RESPECTIVAS DE mb COLUMNAS																								
EL NUMERO DE COLUMNAS REQUERIDA DE MEMORIA (mb)	PRIMERA TERCER METODOS DE REORDENAMIENTO	CUARTO METODO REORDENAMIENTO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	QPSK		0	2																						
4	16QAM	QPSK	0	2	4	7																				
6	64QAM		0	2	5	9	10	13																		
8	256QAM	16QAM	0	0	2	4	4	5	7	7																
10	1024QAM		0	3	6	8	11	13	15	17	18	20														
12	4096QAM	64QAM	0	0	2	2	3	4	4	5	5	7	8	9												
16		256QAM	0	2	2	2	2	3	7	15	16	20	22	22	27	27	28	32								
20		1024QAM	0	1	3	4	5	6	6	9	13	14	14	16	21	21	23	25	26	26	28	30				
24		4096QAM	0	5	8	8	8	8	10	10	10	12	13	16	17	19	21	22	23	26	26	37	39	40	41	41

FIG. 24

EL NÚMERO DE COLUMNAS REQUERIDA DE MEMORIA (mb)	PRIMER A TERCER METODOS DE REORDENAMIENTO	CUARTO METODO REORDENAMIENTO	POSICIONES INICIO ESCRITURA RESPECTIVAS DE mb COLUMNAS																							
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	QPSK		0	0																						
4	16QAM	QPSK	0	2	3	3																				
6	64QAM		0	0	2	3	7	7																		
8	256QAM	16QAM	0	0	0	1	7	20	20	21																
10	1024QAM		0	1	2	2	3	3	4	4	5	7														
12	4096QAM	64QAM	0	0	0	2	2	2	3	3	3	6	7	7												
20		1024QAM	0	0	0	2	2	2	2	2	2	5	5	5	5	7	7	7	7	8	8	10				
24		4096QAM	0	0	0	0	0	0	0	0	1	1	1	2	2	2	3	3	3	7	9	10	10	10	10	11

FIG. 25

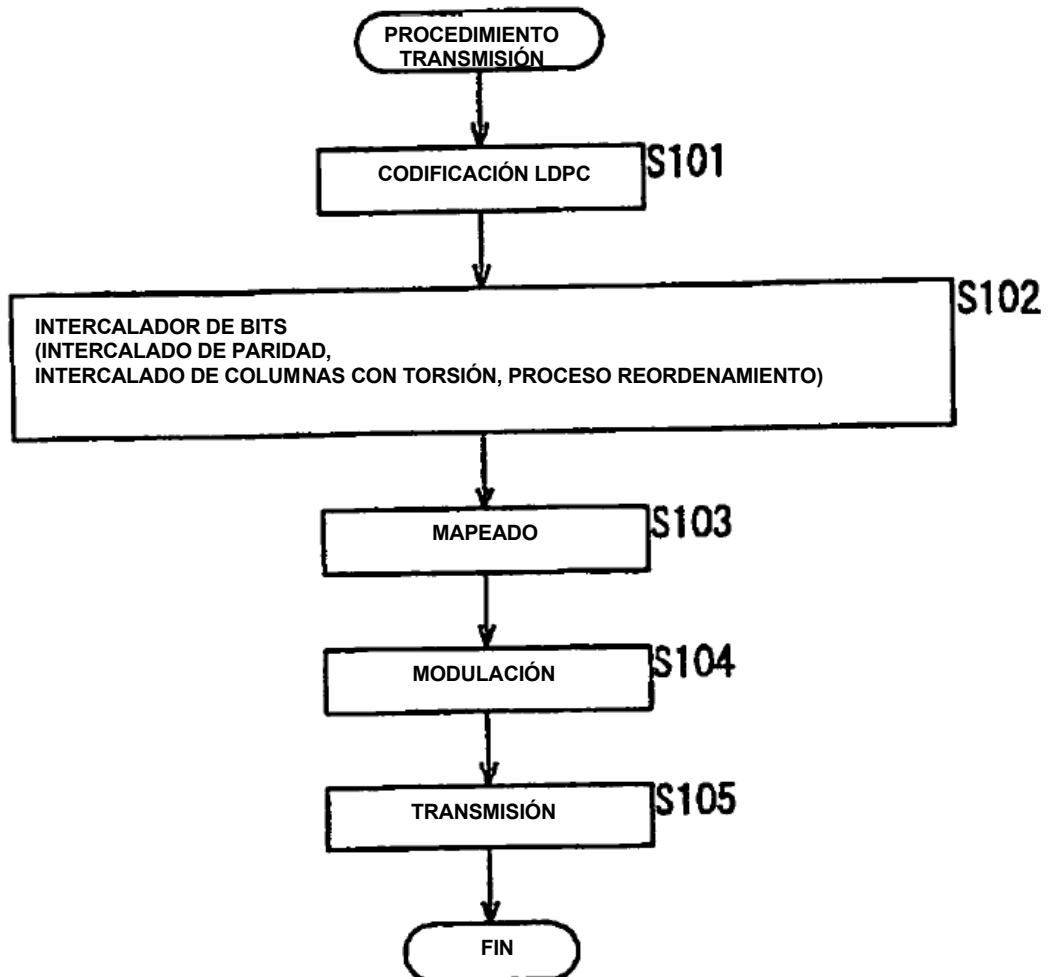
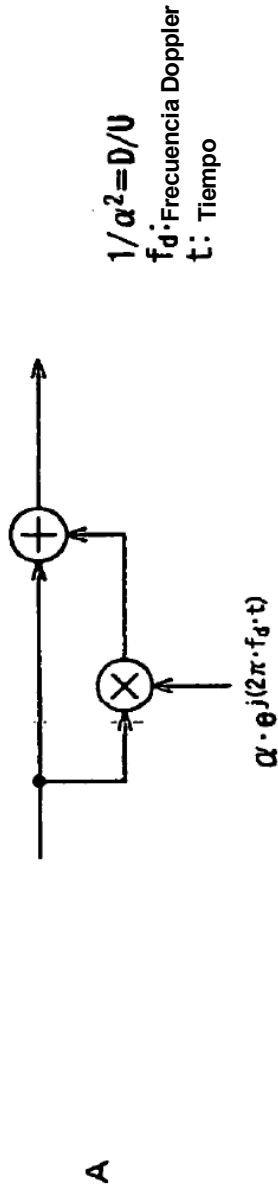
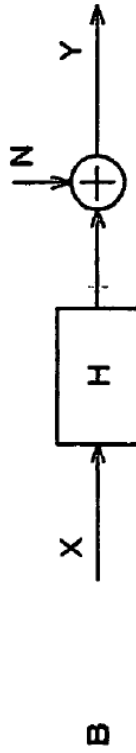


FIG. 26

MÓDELO DE FLUTTER REDUCIDO EQUIVALENTE



SE REALIZARON SIMULACIONES USANDO UN MODELO CORRESPONDIENTE A UNA PORTADORA EXTRAIDA EN EL LADO DE RECEPCIÓN DESPUÉS DE QUE SE REALICE FFT EN UN SÍMBOLO OFDM QUE FUE TRANSMITIDO A TRAVÉS DE ESTE CANAL



$$Y = \left[1 + \alpha \cdot \exp \left(j2\pi \cdot m \cdot f_d \cdot Ts + j2\pi \cdot \frac{(Nu-1) \cdot f_d \cdot Tu}{Nu} \right) \cdot \frac{\text{sinc}(\pi \cdot f_d \cdot Tu)}{\text{sinc}(\pi \cdot f_d \cdot Tu / Nu)} \right] \cdot X + N$$

m: número símbolo
 Ts: longitud símbolo (seg.)
 Tu: longitud símbolo efectiva (seg.)
 Nu: número de portadoras OFDM

$$E[N^2] = \alpha^2 \cdot \left(1 - \left| \frac{\text{sinc}(\pi \cdot f_d \cdot Tu)}{\text{sinc}(\pi \cdot f_d \cdot Tu / Nu)} \right|^2 \right)$$

POTENCIA DE $|C|$: APROXIMADA POR AWGN

FIG. 27

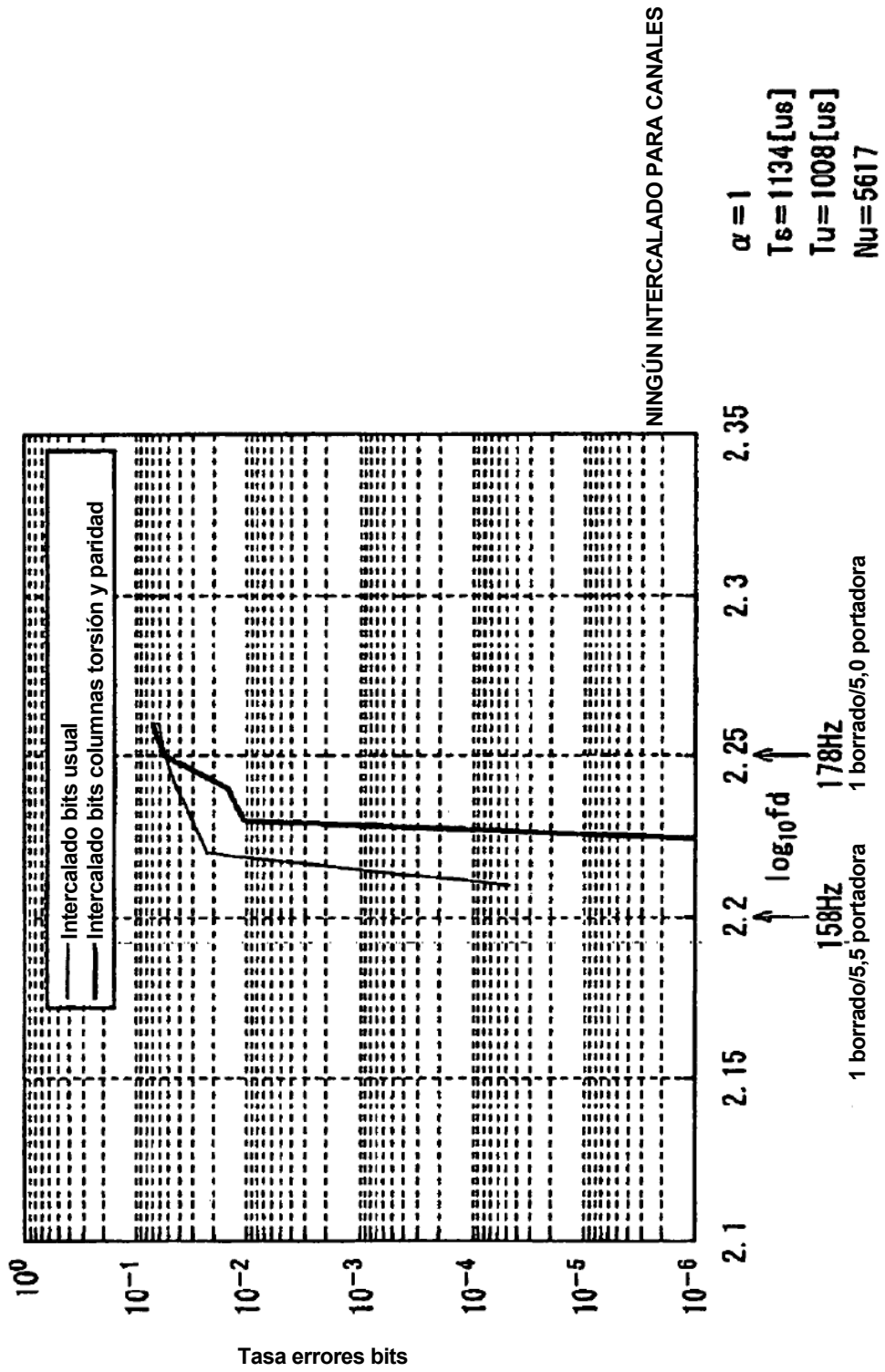
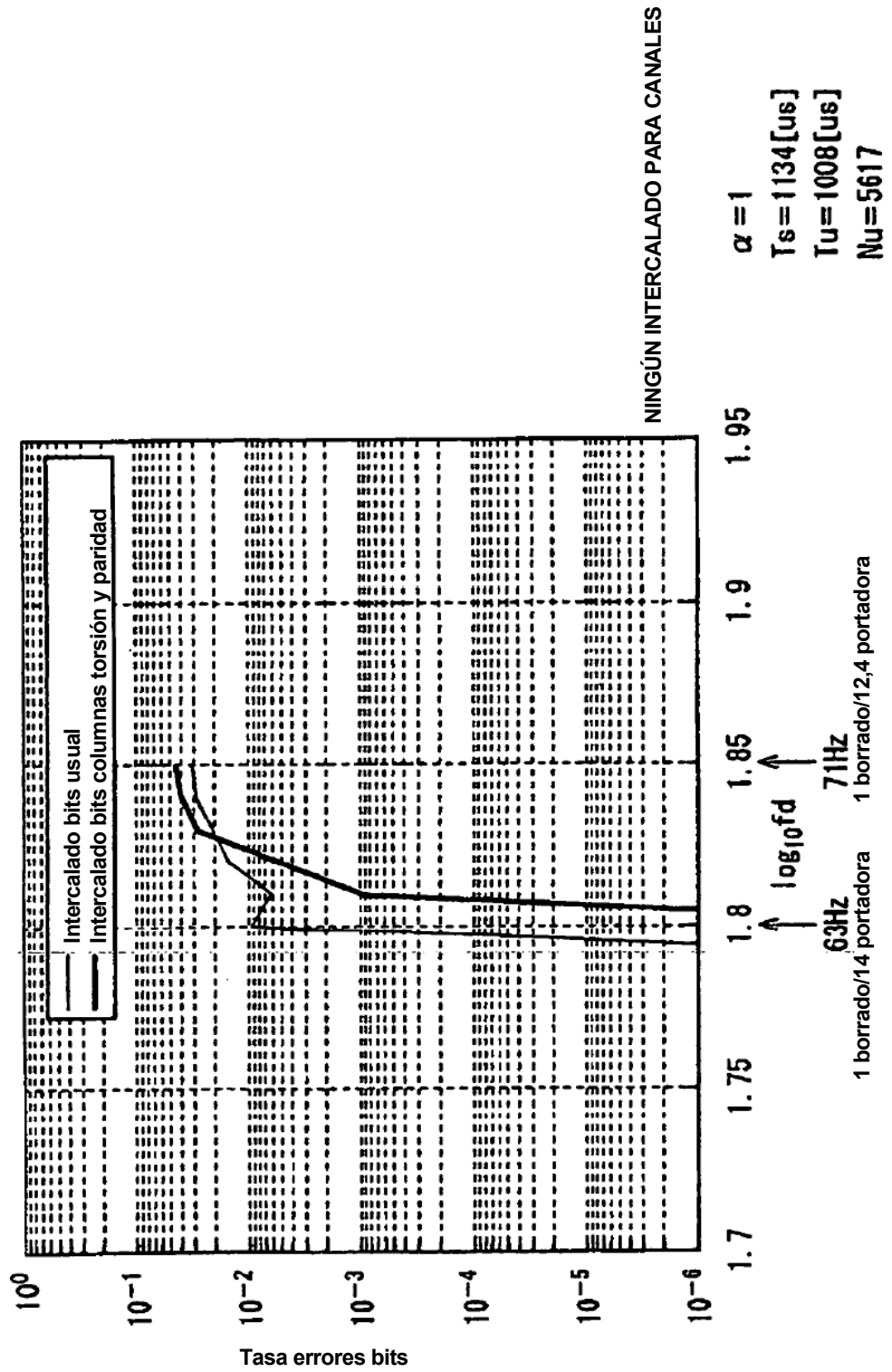


FIG. 28



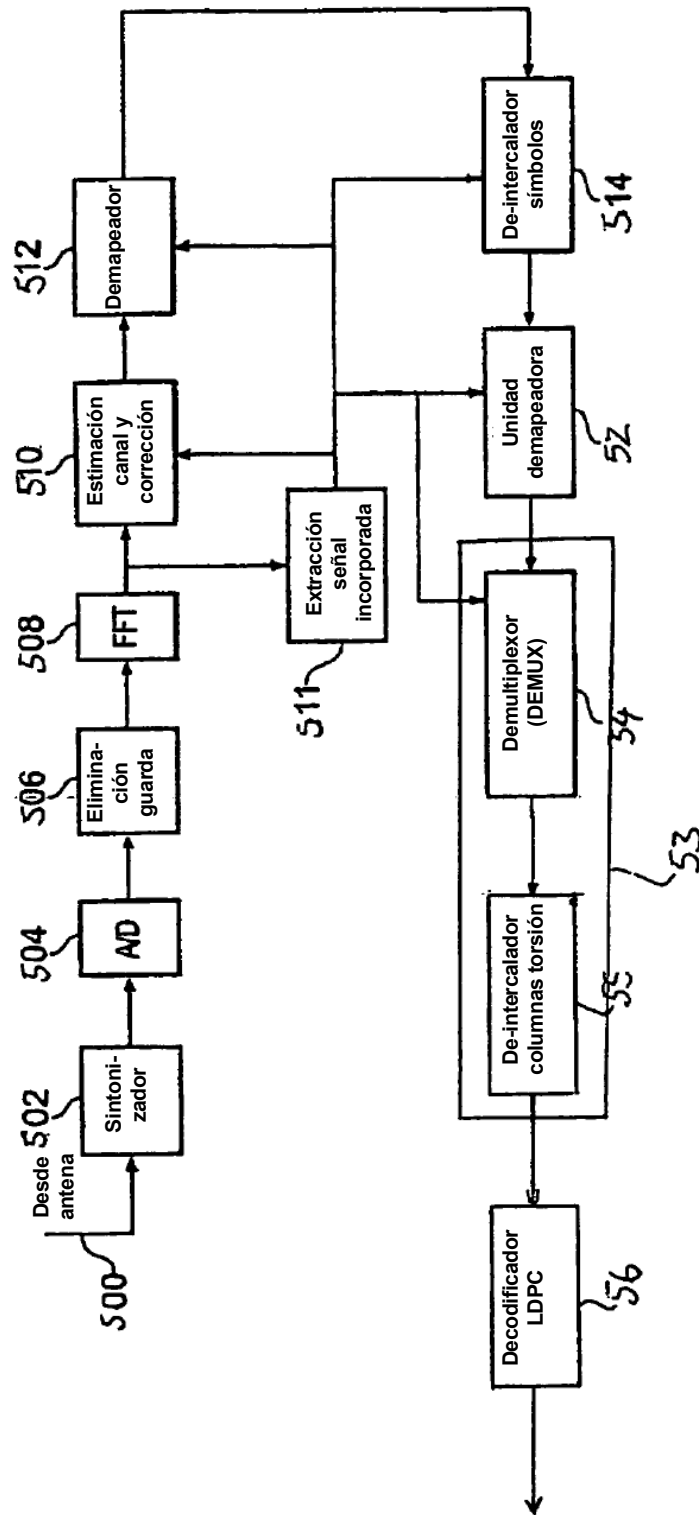


FIG. 29

FIG. 30

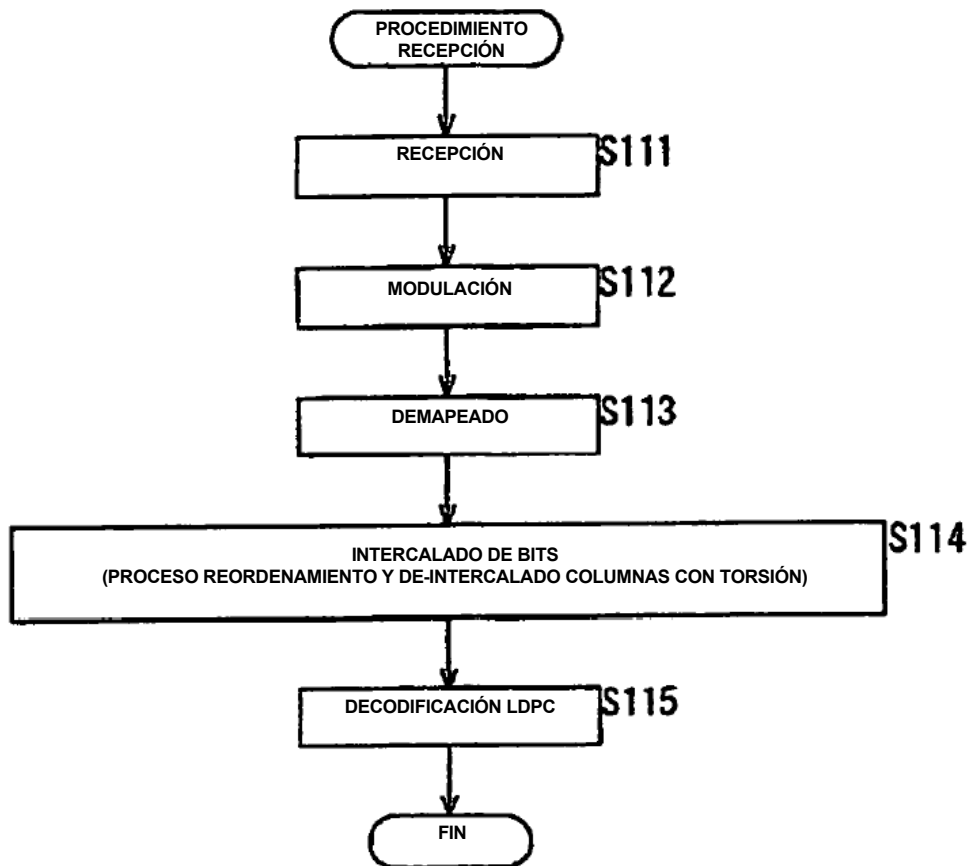


FIG. 31

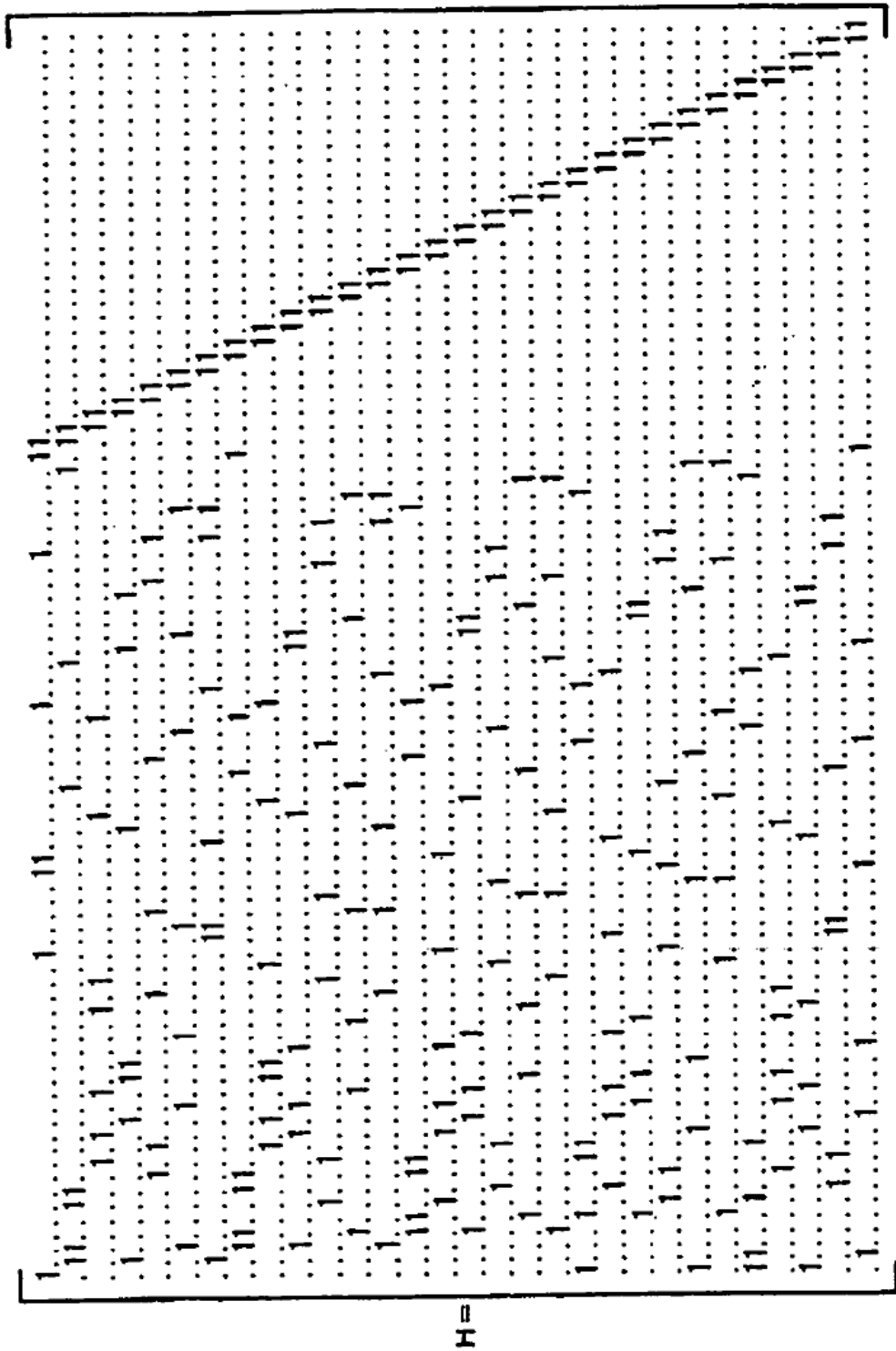


FIG. 32

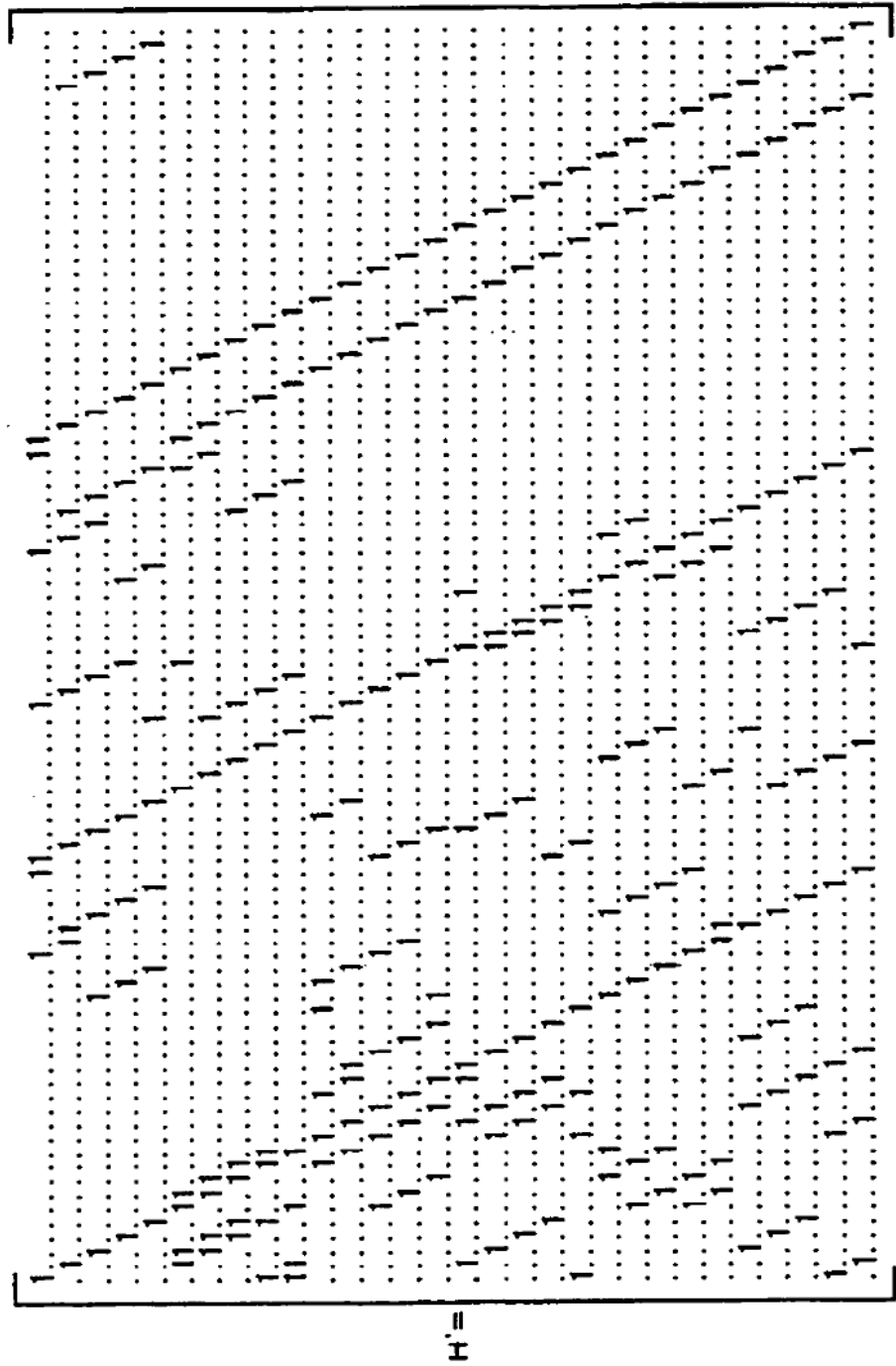


FIG. 33

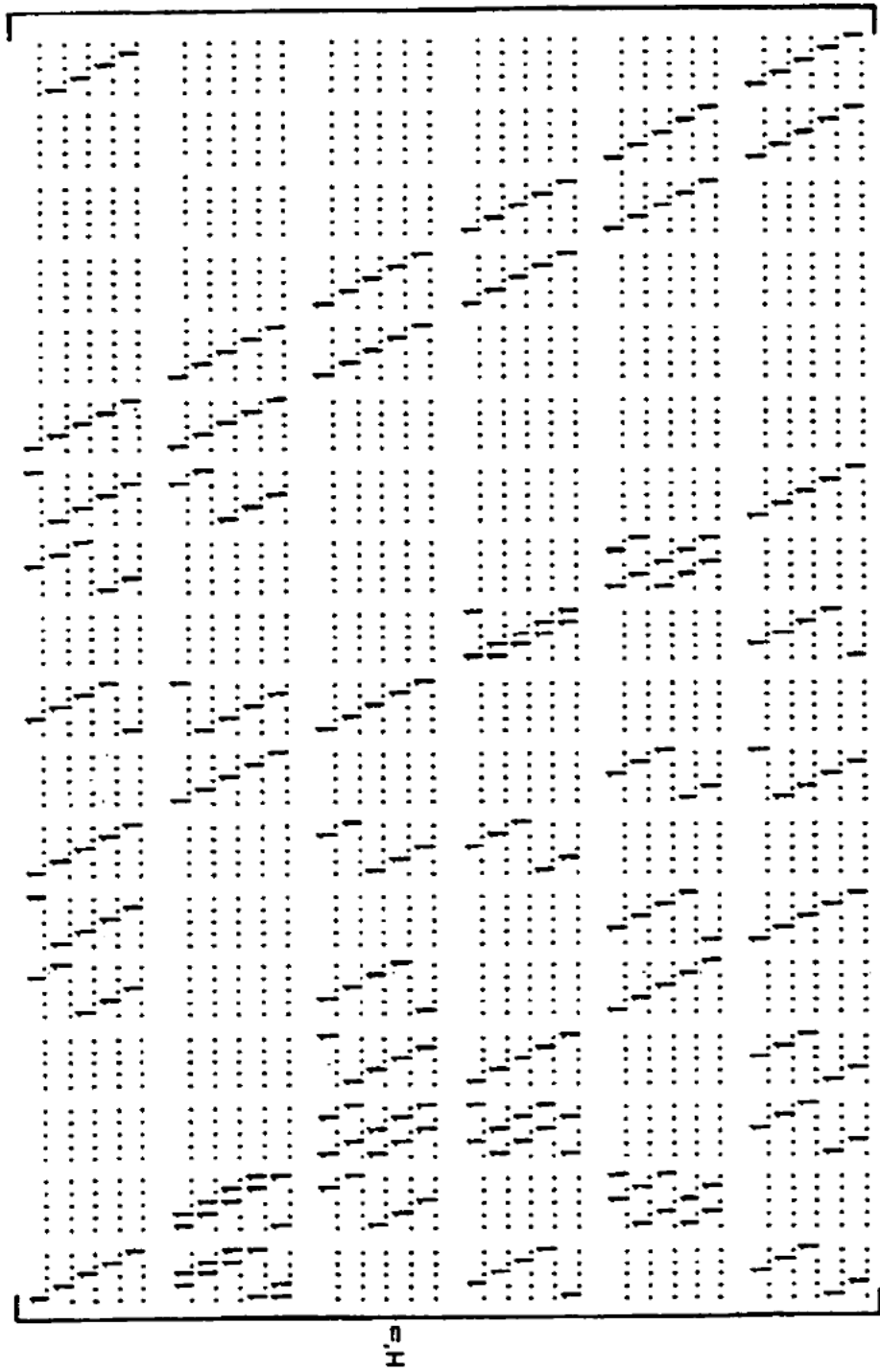


FIG. 34

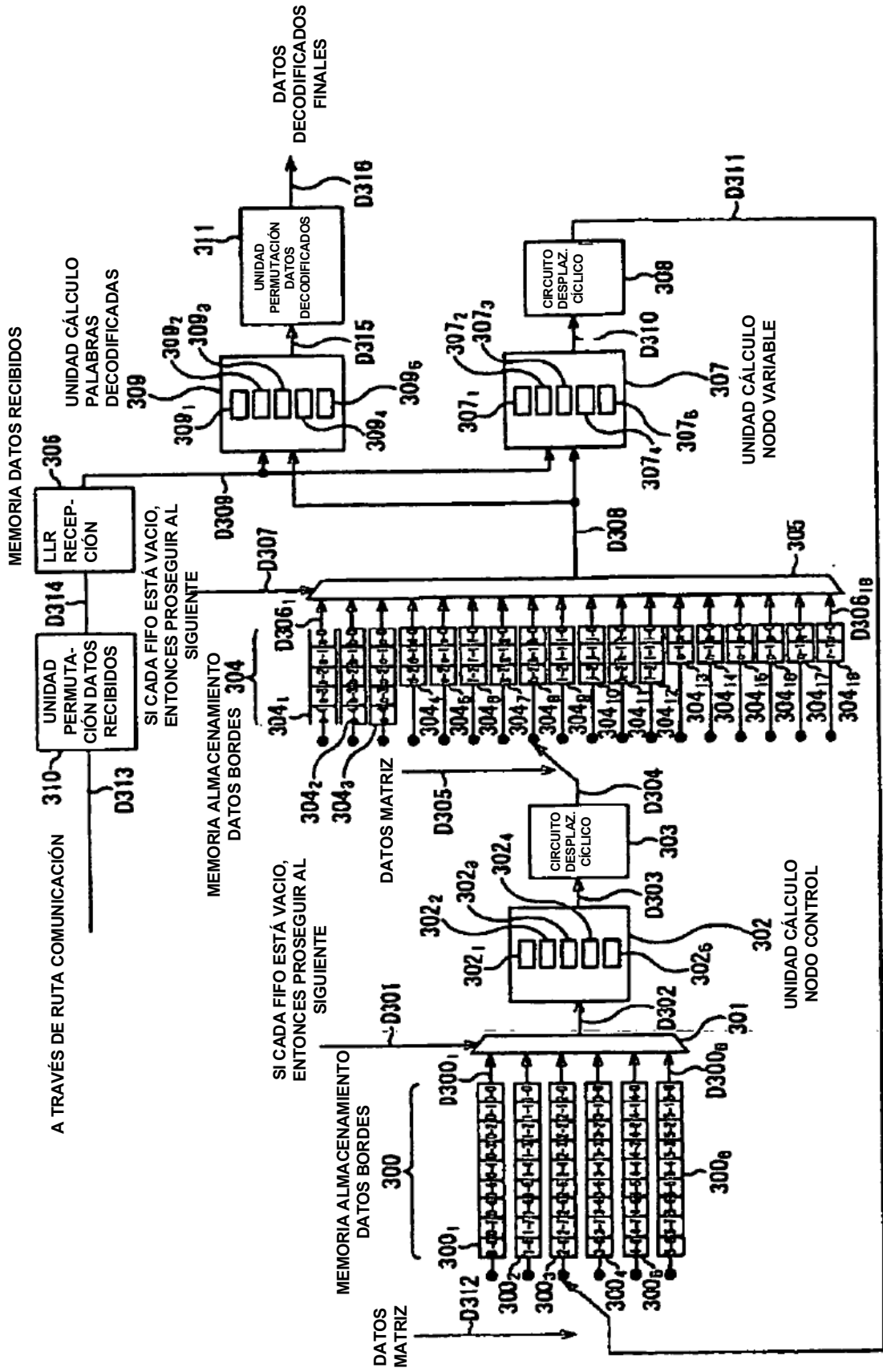


FIG. 35

DECODIFICADOR LDPC 56

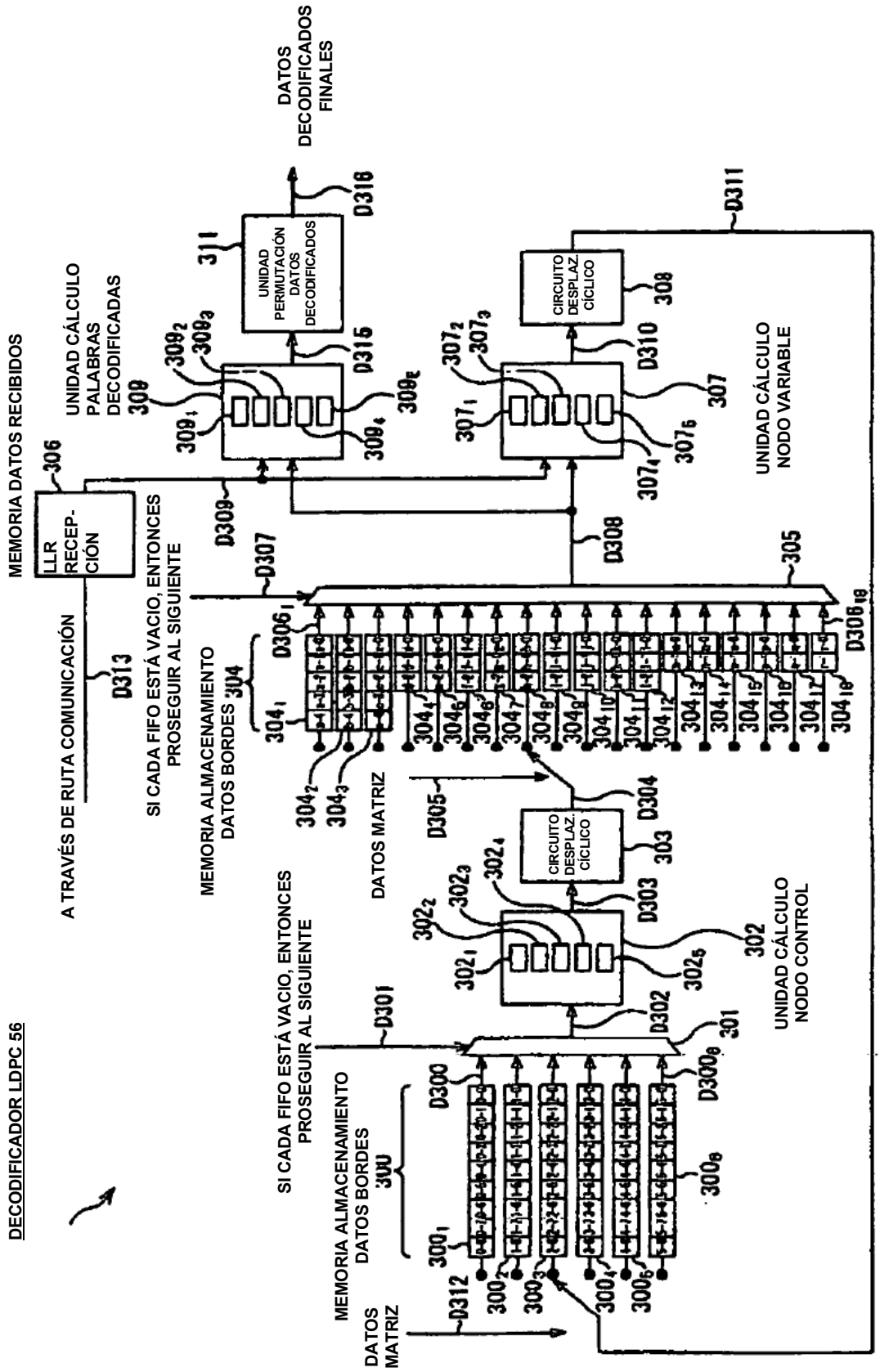
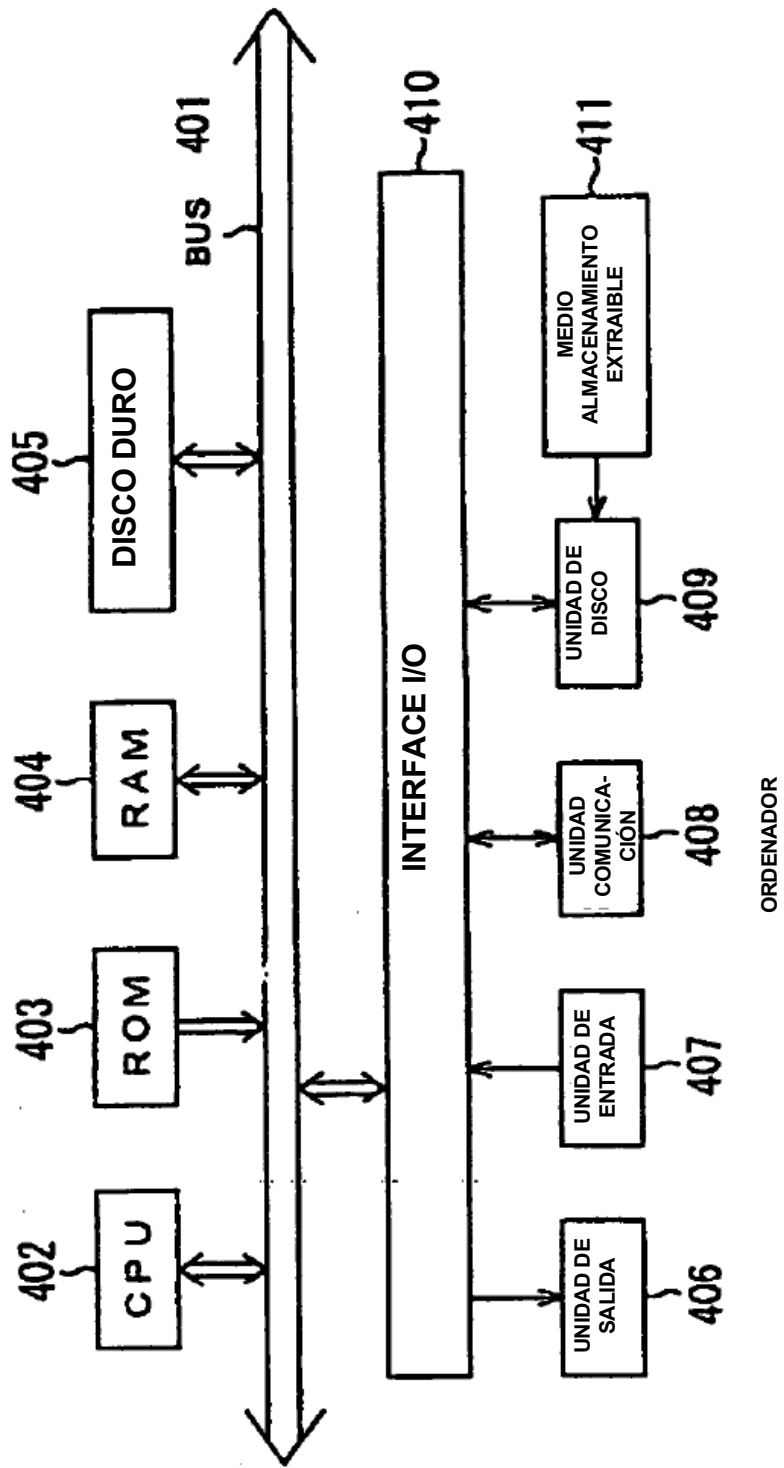


FIG. 36



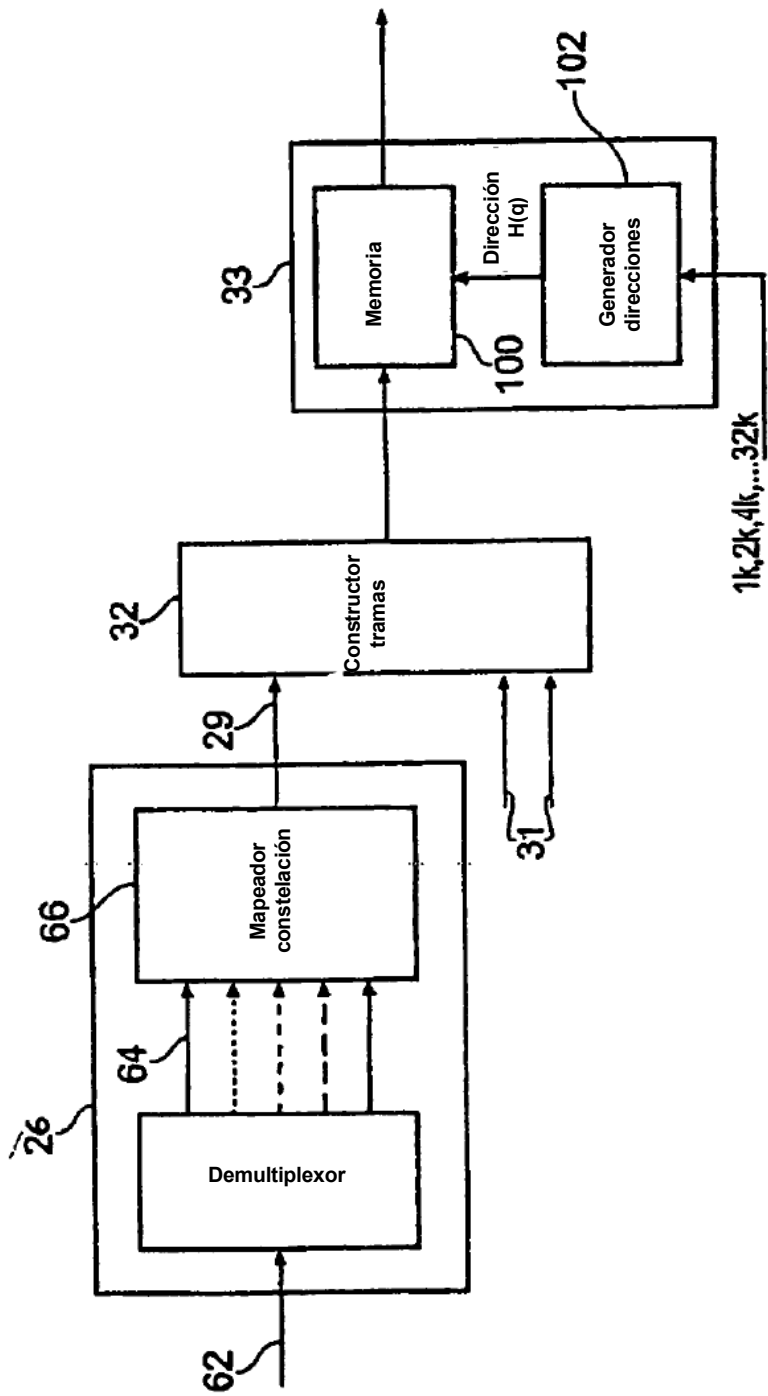


FIG.37

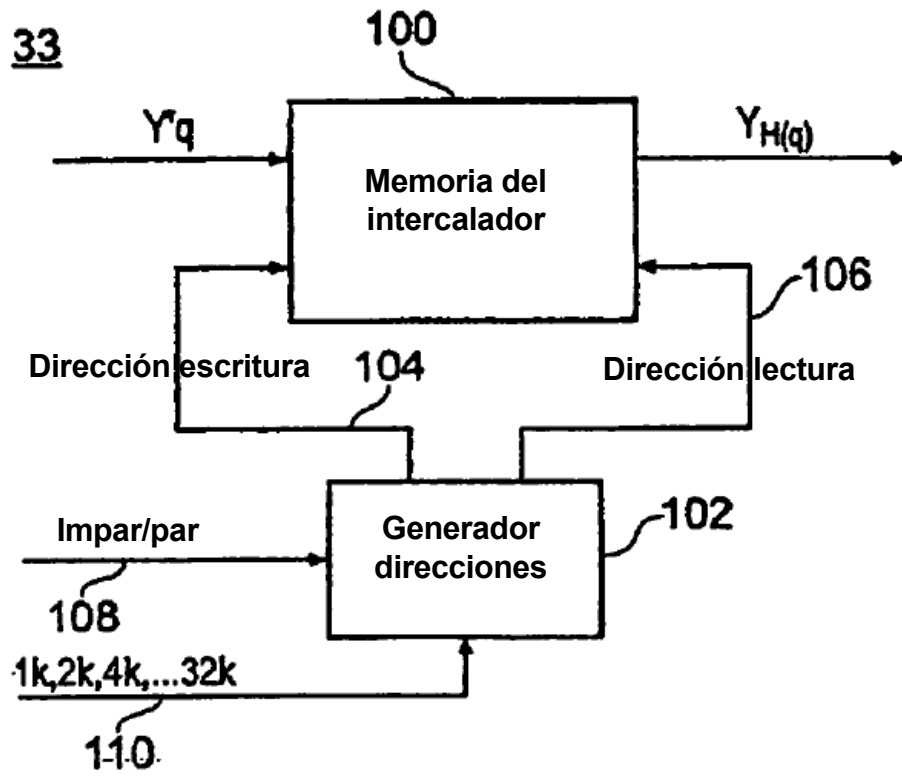


FIG. 38

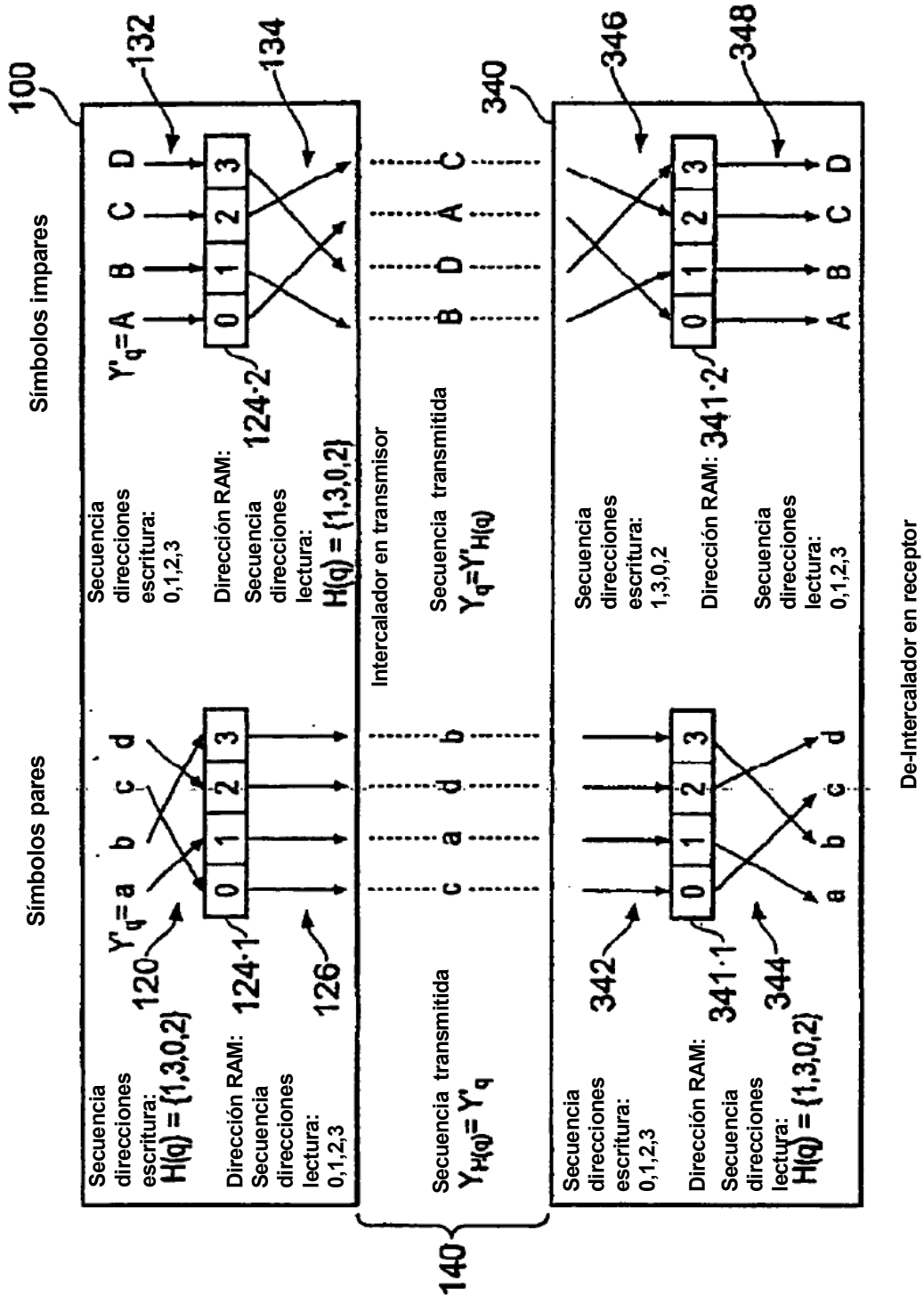


FIG. 39

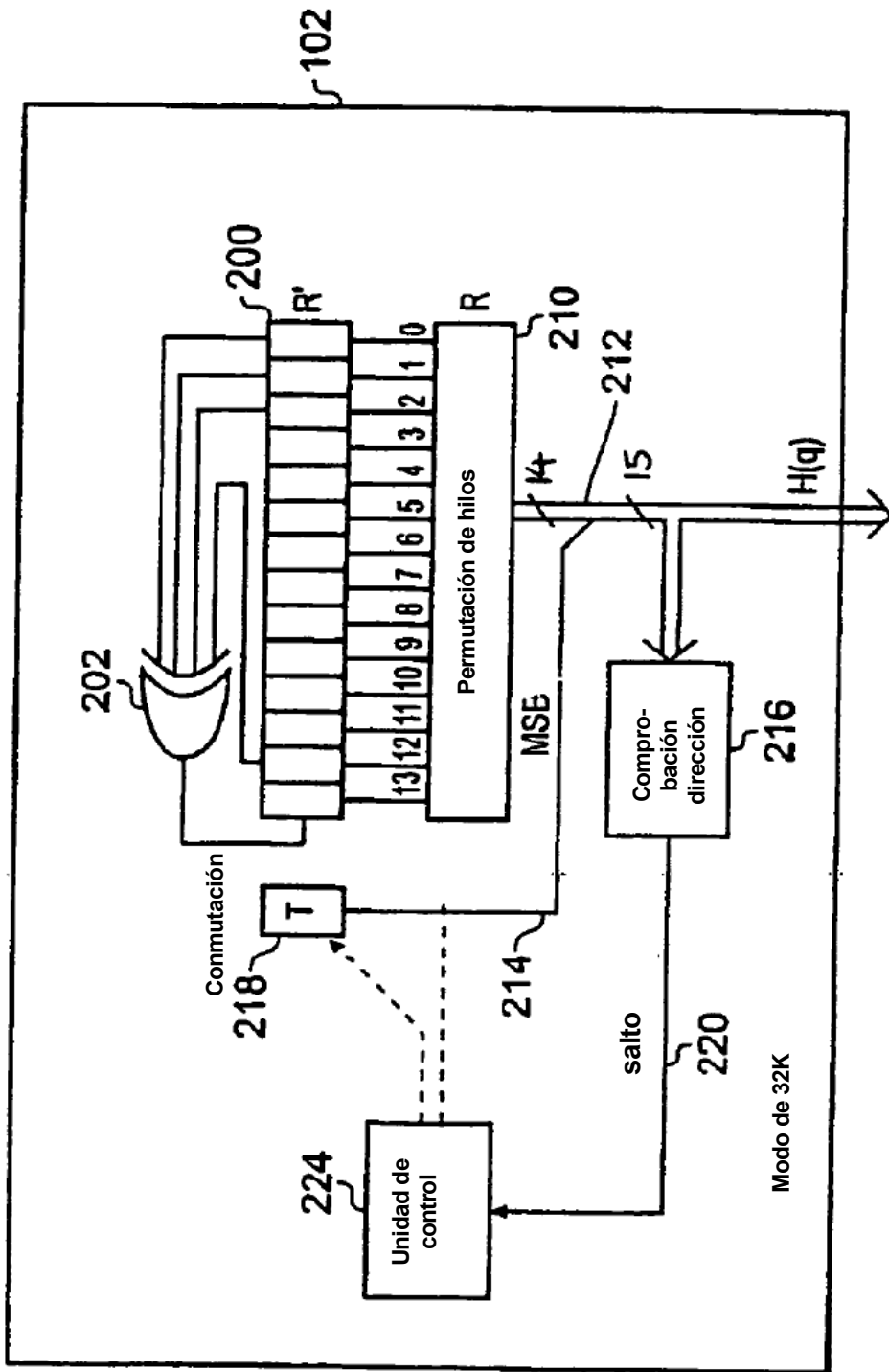
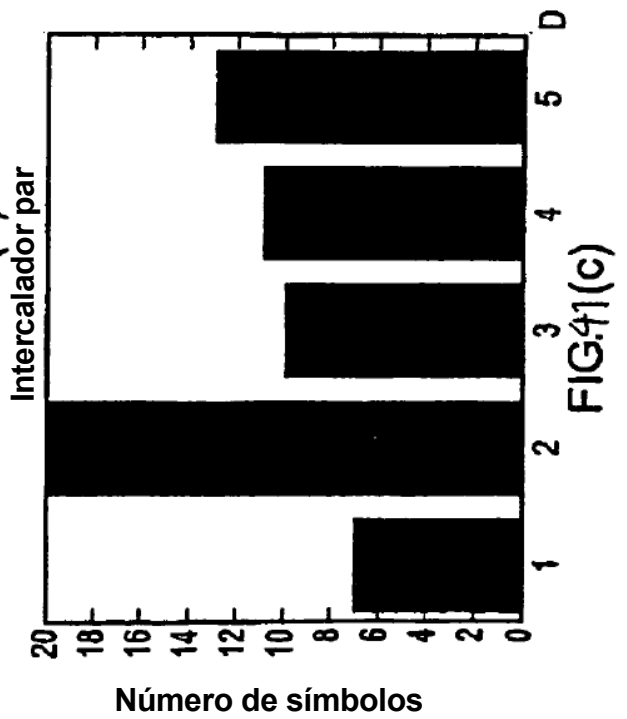
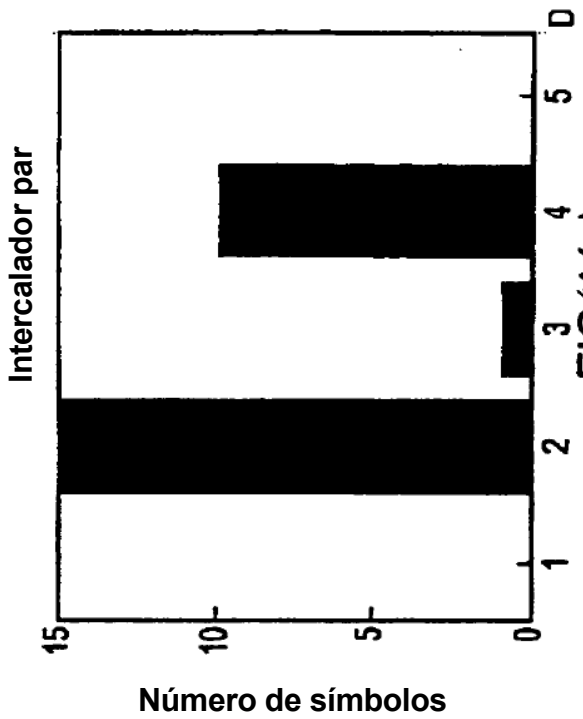
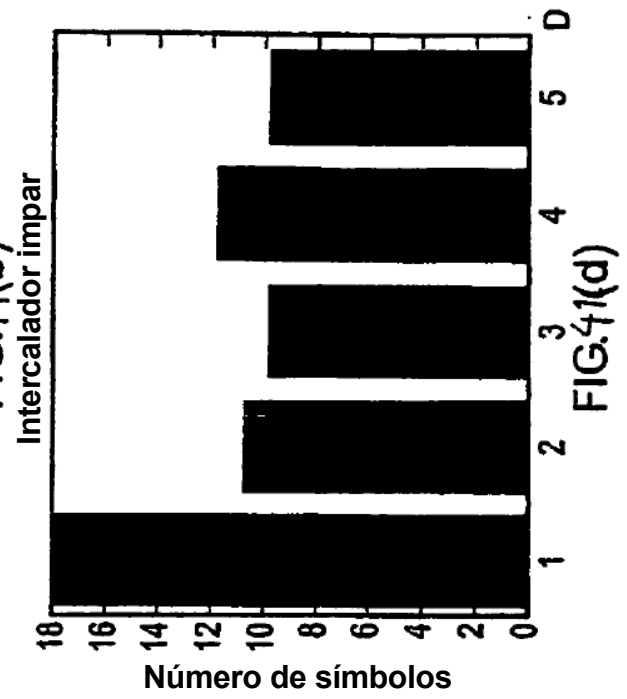
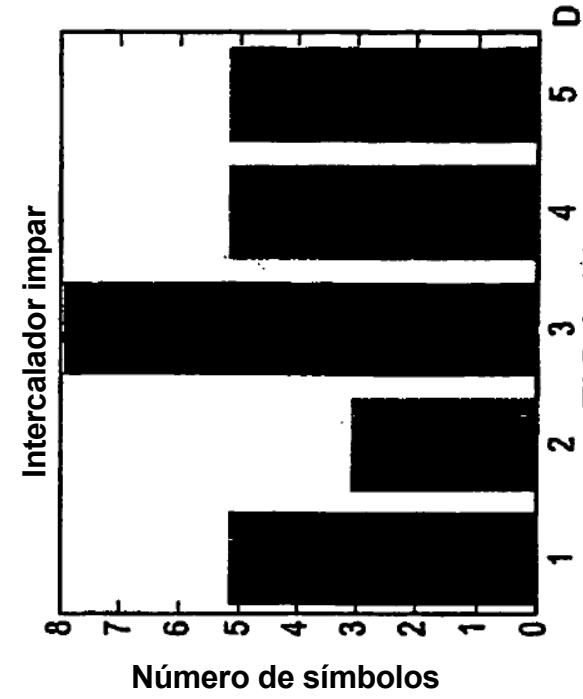


FIG. 40



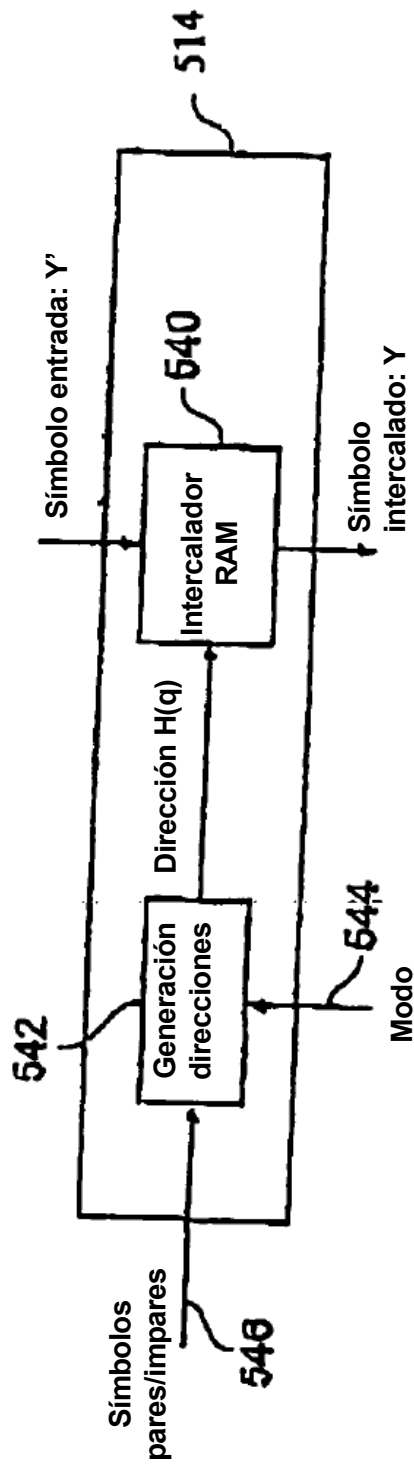


FIG. 42

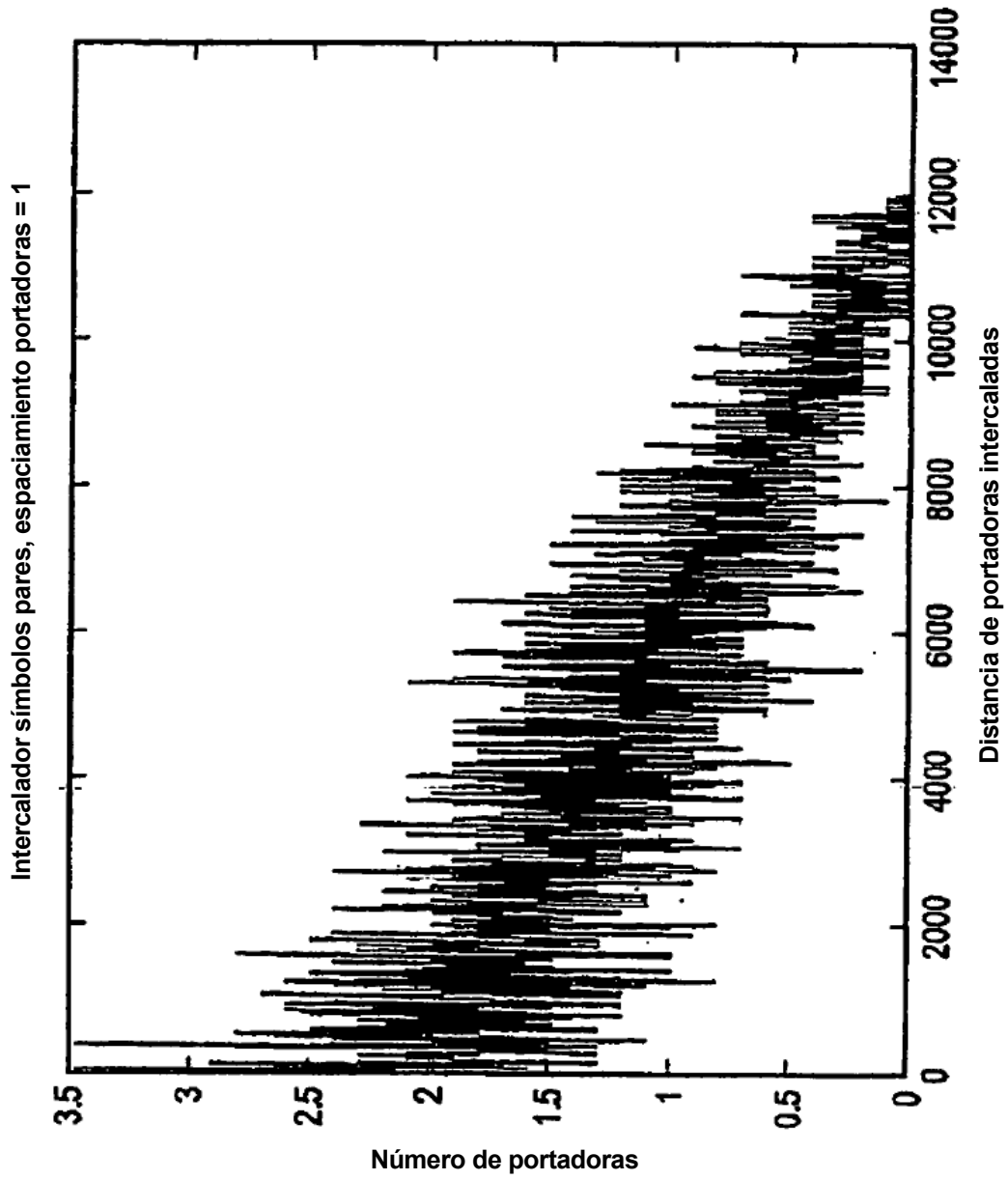


FIG.43(a)

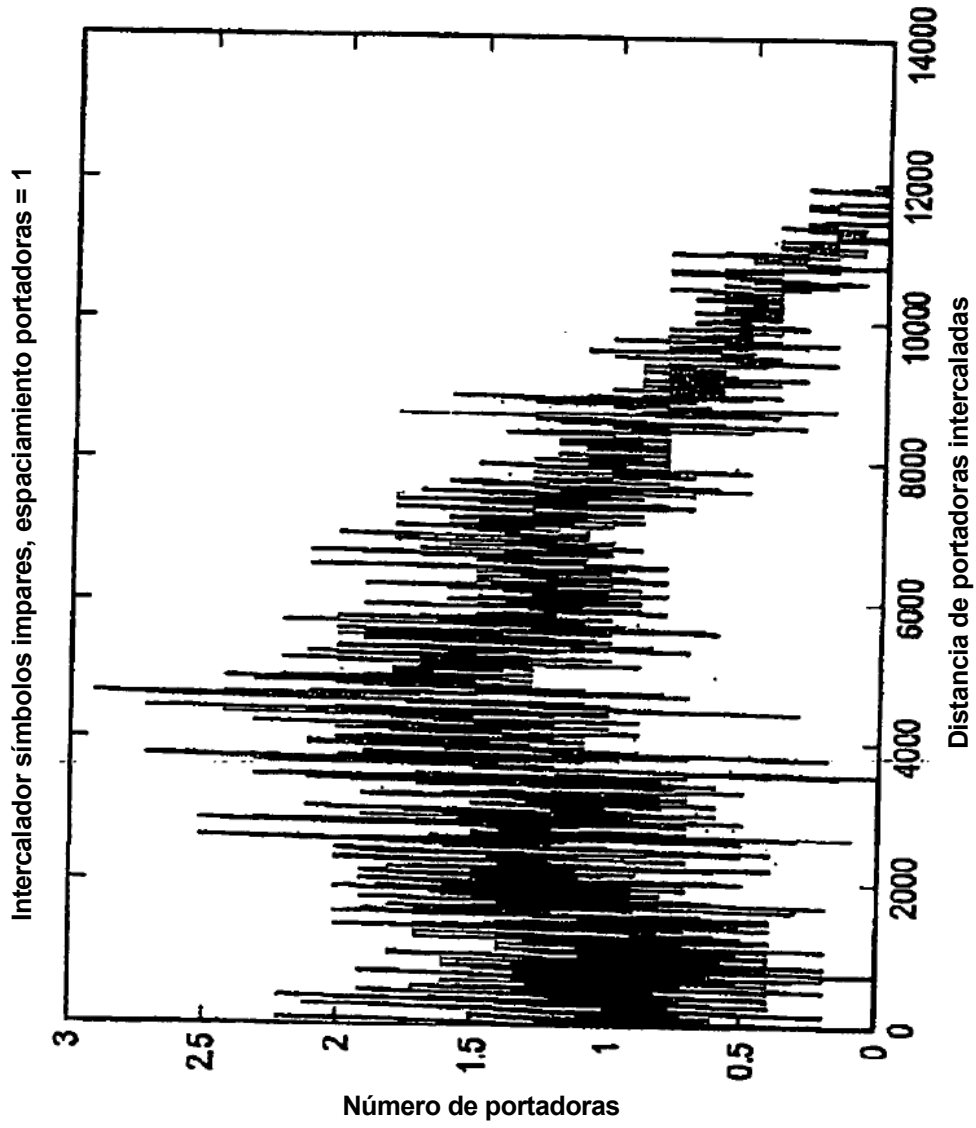


FIG. 43(b)

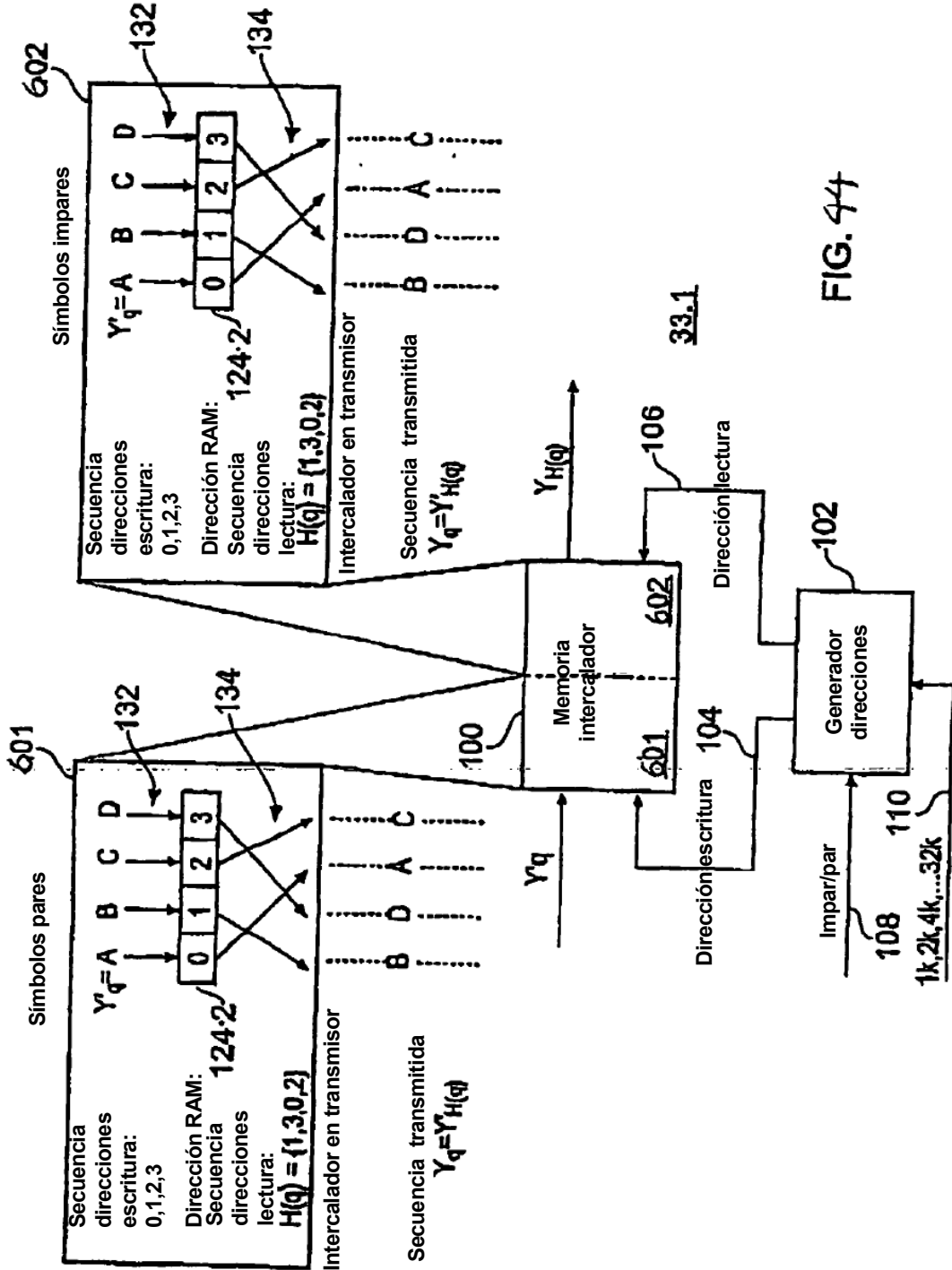


FIG. 44

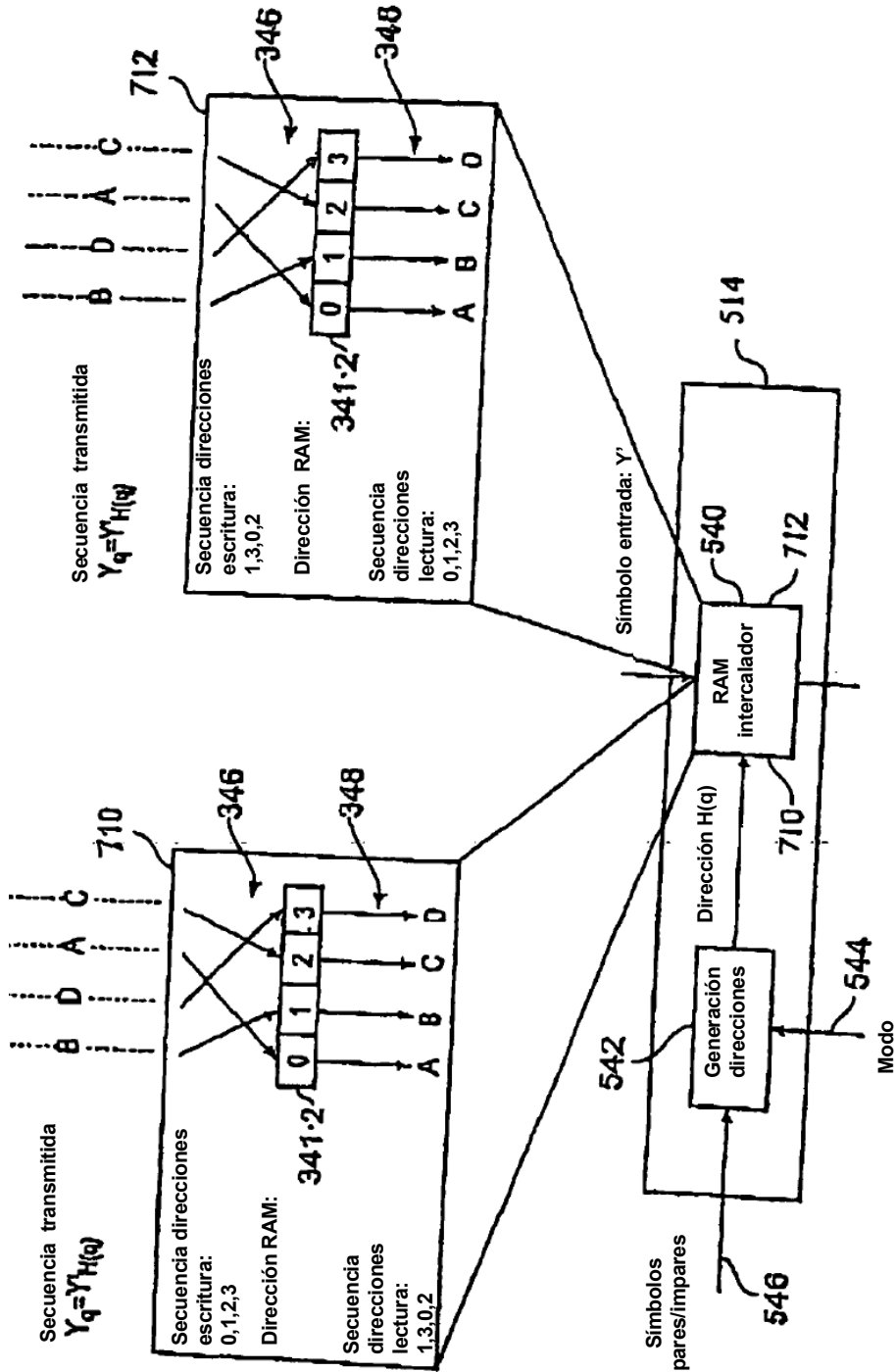


FIG. 45 Símbolo intercalado: Y