

(19)



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS
ESPAÑA



(11) Número de publicación: **2 562 031**

(51) Int. Cl.:

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

(12)

TRADUCCIÓN DE PATENTE EUROPEA

T3

(96) Fecha de presentación y número de la solicitud europea: **24.10.2008 E 11183243 (2)**

(97) Fecha y número de publicación de la concesión europea: **06.01.2016 EP 2403147**

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

(30) Prioridad:

**30.10.2007 GB 0721271 30.10.2007 GB 0721272
30.10.2007 GB 0721270 30.10.2007 GB 0721269
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:
02.03.2016

(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;
YOKOKAWA, TAKASHI y
YAMAMOTO, MAKIKO**

(74) Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 562 031 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

Campo de la invención

5 La presente invención se refiere a métodos y aparatos de procesamiento de datos para recuperar bits de datos desde un cierto número de señales de sub-portadora de un símbolo Multiplexado por División de Frecuencias Ortogonales (OFDM) para formar una corriente de bits de salida.

Realizaciones de la presente invención pueden proporcionar un receptor OFDM.

Antecedentes de la invención

10 El estándar de Difusión de Video Digital Terrestre (DVB-T) utiliza Multiplexado por División de Frecuencia Ortogonal (OFMD) para comunicar datos que representan imágenes de video y de sonido a receptores por medio de una señal de comunicación de radiodifusión. Se sabe que existen dos modos conocidos para el estándar de DVB-T que se conocen 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 manera similar, se ha proporcionado para el estándar de Radiodifusión Portátil de Video Digital (DVB-H) un modo de 4k, en el que número de sub-portadoras es de 4096.

15 15 Esquemas de codificación de corrección de error, tal como codificación LDPC/BCH, que han sido propuestos para el DVB-T2, se comportan mejor cuando el ruido y la degradación de los valores de símbolo resultantes de la comunicación están sin relacionar. Los canales de radiodifusión terrestre pueden adolecer de desvanecimiento correlacionado en los dominios tanto del tiempo como de la frecuencia. Como tales, separando bits de datos codificados en diferentes señales de sub-portadora del símbolo de OFMD tanto como sea posible, el rendimiento de 20 los esquemas de codificación de corrección de error se puede incrementar.

Con el fin de mejorar la integridad de los datos comunicados utilizando DVB-T o DVB-H, se conoce el hecho de proporcionar un intercalador de símbolo a efectos de intercalar símbolos de datos de entrada según son mapeados tales símbolos sobre las señales de sub-portadora de un símbolo de OFMD. Para el modo de 2k y el modo de 8k, se ha divulgado una disposición en el estándar de DVB-T para generar las direcciones que efectúen el mapeo. De igual modo, para el modo de 4k del estándar de DVB-H, se ha proporcionado una disposición para generar direcciones para mapeo, y un generador de dirección para implementar este mapeo ha sido divulgado en la solicitud de patente europea nº 04251667.4. El generador de dirección comprende un registro de desplazamiento de retroalimentación lineal que es operable 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 retroalimentación lineal con el fin 30 de generar una dirección. La dirección proporciona una indicación de una posición de memoria de la memoria de intercalador para escribir el símbolo de dato de entrada en, o leer el símbolo de dato de entrada desde, la memoria de intercalador para el mapeado sobre una señal de sub-portadora del símbolo de OFMD. De forma similar, un generador de dirección del receptor está dispuesto para generar direcciones de la memoria de intercalador para escribir los símbolos de datos recibidos en, o leer los símbolos de datos que salen desde, la memoria de intercalador 35 para formar una corriente de símbolos de salida.

De acuerdo con un desarrollo adicional del estándar de Radiodifusión de Video Digital Terrestre, conocido como CVB-T2, existe un deseo de mejorar la comunicación de bits de datos, y más en particular de proporcionar una disposición mejorada para intercalar bits de datos codificados con códigos LDPC y símbolos de datos sobre las señales de sub-portadora de símbolos de OFDM.

40 40 En un artículo denominado "*A novel, high-speed, reconfigurable demapper-symbol deinterleaver architecture for DVB-T*", de Horvath et al, publicado en la ISCAS '99, el acto de la Conferencia Internacional sobre Circuitos y Sistemas del IEEE de 1999, en Orlando, Florida, EE. UU., desde el 30 de mayo al 2 de junio de 1999, se divulga un algoritmo reconfigurable para el desmapeo de la señal, el cual se puede utilizar en receptores DVB-T (Difusión de Video Digital, versión Terrestre). El algoritmo/la arquitectura soporta tanto el modo de transmisión jerárquico como 45 no jerárquico, para disposiciones en las cuales se utilizan esquemas de modulación diferentes y tasas de codificación. El receptor incluye un intercalador de símbolos y un intercalador de bits.

Los documentos EP 1 463 255 y EP 1 463 256 divulan un intercalador para mapear símbolos de datos en las sub-portadoras de un símbolo de OFDM. El intercalador incluye un generador de direcciones. El intercalador introduce 50 por lectura el número predeterminado de símbolos de datos en la memoria del intercalador y extrae por lectura los símbolos de datos sobre las sub-portadoras del símbolo de OFDM, siendo el orden de salida por lectura diferente al orden de entrada por lectura, lo que se determina a partir de un conjunto de direcciones con el efecto de que los símbolos de datos sean intercalados. El conjunto de direcciones que se generan en el generador de direcciones puede proporcionar intercalación para un transmisor o receptor DVB de modo de 4K.

En un artículo denominado “*IEEE 802.16 TG4 OFDM PHY Proposal for the 802.16b PHY Layer*”, el grupo de trabajo de acceso inalámbrico de banda ancha del IEEE 802.16, de Segal Y. et al, del 4 de marzo de 2001 (2001-03-04), páginas 1-53, se divulga una especificación para una capa física que incluye una disposición para portar datos utilizando símbolos de OFDM.

- 5 El documento US 6.353.900 divulga un intercalador que incluye un generador de dirección para generar una dirección de una memoria de intercalador utilizando un generador de número pseudo-aleatorio. Los datos se escriben en la memoria de intercalador en un orden secuencial y a continuación son extraídos por lectura desde el generador de dirección utilizando direcciones especificadas por el generador de dirección.

Sumario de la invención

- 10 De acuerdo con la presente invención, se proporciona un aparato de procesamiento de datos dispuesto en funcionamiento para recuperar bits de datos desde símbolos de datos recibidos desde un número predeterminado de señales de sub-portadora de un símbolo Multiplexado por División de Frecuencias Ortogonales (OFDM) para formar una corriente de bits de salida. El aparato de procesamiento de datos comprende un desentrelazador de símbolos que puede funcionar para introducir por lectura en una memoria de entrelazador de símbolos el número predeterminado de símbolos de datos desde las señales de sub-portadora OFDM, y para extraer por lectura de la memoria de entrelazador de símbolos los símbolos de datos adentro de una corriente de símbolos de salida para efectuar el mapeo, siendo la extracción por lectura en un orden diferente a la introducción por lectura, determinándose el orden a partir de un conjunto de direcciones, con el efecto de que los símbolos de datos son desentrelazados desde las señales de sub-portadora OFDM adentro de la corriente de símbolos de salida. Una unidad de desmapeo se puede hacer funcionar para generar, a partir de los símbolos de datos de la corriente de símbolos de salida, bits de datos codificados por LDPC, entrelazados por paridad, mediante la conversión de cada uno de los símbolos de datos de la corriente de símbolos de salida que representa un símbolo de modulación de las señales de sub-portadora OFDM en bits de datos de acuerdo con un esquema de modulación. Un permutador inverso está adaptado para realizar un proceso de permutación inversa para efectuar una reversión de un proceso de permutación aplicado a los bits de datos codificados por LDPC, entrelazados por paridad, para permutar los bits de datos codificados por LDPC de manera que no se incorporan en el mismo símbolo una pluralidad de los bits de datos codificados por LDPC que corresponden a un valor de 1 en una fila arbitraria de una matriz de información que corresponde a bits de información de un código LDPC, que se usó para codificar los bits de datos. Un decodificador LDPC está adaptado para realizar una decodificación LDPC sobre los bits de datos codificados por LDPC sobre los que se ha realizado el proceso de permutación inversa para formar los bits de datos de salida. El desentrelazador de símbolos incluye un generador de direcciones que se puede hacer funcionar para generar el conjunto de direcciones, generándose una dirección para cada uno de los símbolos de datos recibido para indicar la señal de sub-portadora OFDM desde la que el símbolo de datos recibido se tiene que mapear adentro de la corriente de símbolos de salida. El generador de dirección comprende: un registro de desplazamiento de retroalimentación lineal que incluye un número predeterminado de niveles de registro y que es operable para generar una secuencia de bits pseudo-aleatoria de acuerdo con un polinomio generador;

un circuito de permutación operable para recibir el contenido de los niveles de registro de desplazamiento y para permutar los bits presentes en los niveles de registro de acuerdo con un código de permutación para formar una dirección de una de las sub-portadoras de OFDM, y

- 40 una unidad de control operable en combinación con un circuito de comprobación de dirección para regenerar una dirección cuando una dirección generada exceda de una dirección válida máxima predeterminada, y en el que:

la dirección válida máxima predeterminada es sustancialmente cuatro mil noventa y seis,

- 45 el registro de desplazamiento de retroalimentación lineal tiene once niveles de registro con un polinomio generador para el registro de desplazamiento de retroalimentación lineal de $R'_i[10] = R'_{i-1}[0] \oplus R'_{i-1}[2]$, y el orden de permutación forma una dirección de once bits $R'_i[n]$ para el símbolo de datos $i^{\text{ésimo}}$ del bit presente en el nivel de registro $n^{\text{ésimo}}$ $R'_i[n]$ de acuerdo con la tabla:

$R'_i[n]$ 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

En un ejemplo, en el que el símbolo de OFDM se genera de acuerdo con un modo de 4k, la dirección válida máxima predeterminada es sustancialmente cuatro mil noventa y seis, el registro de desplazamiento de retroalimentación lineal tiene once niveles de registro con un polinomio generador para el registro de desplazamiento de retroalimentación lineal que es $R'_i[10]=R'_{i-1}[0]\oplus R'_{i-1}[2]$, y el orden de permutación forma una dirección $R'_i[n]$ de once bits para el símbolo de datos $i^{\text{ésimo}}$ a partir del bit presente en el nivel de registro $n^{\text{ésimo}}$ $R'_i[n]$ de acuerdo con la tabla:

$R'_i[n]$ 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
---------------------	---	----	---	---	---	---	---	---	---	---	---

En otro ejemplo, en el que el símbolo de OFDM se genera de acuerdo con un modo de 32k, la dirección válida máxima predeterminada es de aproximadamente treinta y dos mil, el registro de desplazamiento de retroalimentación lineal tiene catorce niveles de registro con un polinomio generador para el registro de desplazamiento de retroalimentación lineal que es $R'_i[13] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[2] \oplus R'_{i-1}[12]$, y el código de permutación forma, con un bit adicional, una dirección $R'_i[n]$ de quince bits para el símbolo de datos $i^{\text{ésimo}}$ a partir del bit presente en el nivel de registro $n^{\text{ésimo}}$ $R'_i[n]$ de acuerdo con la tabla:

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

En otros modos, la dirección válida máxima, el número de niveles del registro de desplazamiento de retroalimentación lineal, el polinomio generador y el código de permutación pueden ser adaptados de acuerdo con el número predeterminado de señales de sub-portadora por símbolo de OFDM en cada modo.

- 10 Las realizaciones de la presente invención incluyen un intercalador de bit que combina con un intercalador de símbolo para mejorar el rendimiento de un sistema de comunicación de OFMD, utilizando codificación de corrección de error de Comprobación de Paridad de Baja Densidad (LDPC). El intercalador de bits incluye un permutador para realizar, cuando dos o más bits de código de un código de Comprobación de Paridad de Baja Densidad (LPDC) son transmitidos y recibidos como un símbolo, un proceso de permutación que permuta los bits de código del código de LPDC de modo que una pluralidad de bits de código correspondientes a un valor de 1 en una fila arbitraria de una matriz de información correspondientes a bits de información del código de LPDC no son incorporados en el mismo símbolo.
- 15

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

- 20 Los códigos de LDPC pueden proporcionar un comportamiento alto de corrección de error en trayectorias de comunicación, distintas de los canales Ruido Gaussiano Blanco Aditivo, que sea superior a códigos convolucionales o a códigos convolucionales Reed Solomon (RS) concatenados. Esto puede estar previsto en canales de comunicación que presenten errores de ráfaga que ocasionen borraduras. Así, existe una necesidad de proporcionar un método para incrementar la resistencia a los errores de ráfaga o borraduras mientras se conserva el comportamiento de las trayectorias de comunicación de AWGN.
- 25

La invención ha sido realizada en vista de las circunstancias anteriores y proporciona un aparato y un método de procesamiento de datos que puede incrementar la resistencia a los errores en bits de código de códigos de LDPC tales como los errores de ráfaga o las borraduras, combinando un intercalador de bits para los bits de datos codificados de LDPC con un intercalador de símbolo.

- 30 Es decir, de acuerdo con las realizaciones de la invención, la intercalación de paridad se realiza sobre un código de LDPC obtenido al realizar codificación de LDPC de acuerdo con una matriz de comprobación de paridad que incluye una matriz de paridad estructurada en forma gradual correspondiente a bits de paridad del código de LDPC de modo que los bits de código de LDPC son intercalados en diferentes posiciones de bits de paridad.

- 35 Se han previsto diversos modos operativos de un sistema de OFDM en los que encuentra aplicación la presente invención. Por ejemplo, con el fin de proporcionar un despliegue más disperso uniforme de transmisores de DVB dentro de una red de frecuencia única, se ha propuesto proporcionar el modo de 32k. Para implementar el modo de 32k, se debe proporcionar un intercalador de símbolo para mapear los símbolos de datos de entrada sobre las señales de sub-portadora del símbolo de OFDM.

- 40 Las realizaciones de la presente invención pueden proporcionar un aparato de procesamiento de datos operable como intercalador de símbolo para mapear símbolos de datos que han de ser comunicados sobre un símbolo de OFDM, que tenga aproximadamente treinta y dos mil señales de sub-portadora. En una realización, el número de señales de sub-portadora puede ser un valor comprendido sustancialmente entre veinticuatro mil y treinta y dos mil setecientos sesenta y ocho. Además, el símbolo de OFDM puede incluir sub-portadoras piloto, las cuales están dispuestas sobre un número de símbolos de sub-portadora piloto presentes en el símbolo de OFDM. Como tal, se puede proporcionar el modo de 32k, por ejemplo, para un estándar de DVB, tal como el DVB-72, DVB-Cable2, DVB-T o DVB-H.
- 45

El mapeo de símbolos de datos que van a ser transmitidos sobre las señales de sub-portadora de un símbolo de OFDM, donde el número de señales de sub-portadora es de aproximadamente treinta y dos mil, representa un problema técnico que requiere análisis de simulación y comprobación para establecer un polinomio generador apropiado para el registro de desplazamiento de retroalimentación lineal y el orden de permutación. Esto se debe a que el mapeo requiere que los símbolos sean intercalados sobre las señales de sub-portadora con el efecto de que los símbolos sucesivos procedentes de la corriente de datos estén separados en frecuencia mediante una cantidad

lo más grande posible con el fin de optimizar el comportamiento de los esquemas de codificación de corrección de error.

Como se explicará, se ha descubierto a partir de análisis de comportamiento de simulación que el polinomio generador para el registro de desplazamiento de retroalimentación lineal en combinación con el orden de circuito de permutación indicado anteriormente, proporciona un buen rendimiento. Además, al proporcionar una disposición que puede implementar generación de dirección para uno de entre el modo de 2k, el modo de 4k y el modo de 8k cambiando las derivaciones del polinomio generador para el registro de desplazamiento de retroalimentación lineal y para el orden de permutación, se puede proporcionar una implementación de bajo coste del intercalador de símbolo para el modo de 32k. Además, se puede cambiar un transmisor y un receptor entre el modo de 1k, el modo de 2k, el modo de 4k, el modo de 8k, el modo de 16k y el modo de 32k cambiando el polinomio generador y los órdenes de permutación. Esto puede ser efectuado en software (o mediante señalización incorporada), con lo que se proporciona una implementación flexible.

Diversos aspectos y características de la presente invención están definidos en las reivindicaciones anexas. Los aspectos adicionales de la invención incluyen un aparato de procesamiento de datos que se puede hacer funcionar para mapear símbolos recibidos desde un número predeterminado de señales de sub-portadora de un símbolo Multiplexado por División de Frecuencias Ortogonales (OFDM) adentro de una corriente de símbolos de datos, así como un receptor.

Breve descripción de los dibujos

Las realizaciones de la presente invención van a ser descritas ahora a título de ejemplo únicamente con referencia a los dibujos que se acompañan, en los que las partes iguales han sido dotadas de números de referencia correspondientes, y en los que:

la figura 1 es un diagrama de bloques esquemático de un transmisor de OFDM codificado que puede ser usado, por ejemplo, con el estándar DVB-T2;

la figura 2 ilustra un ejemplo de matriz H de comprobación de paridad de un código de LDPC;

25 la figura 3 es un diagrama de flujo que ilustra un procedimiento para la descodificación de un código de LDPC;

la figura 4 ilustra un ejemplo de matriz de comprobación de paridad de un código de LDPC;

la figura 5 ilustra un gráfico de Tanner de una matriz de comprobación de paridad;

la figura 6 ilustra un nodo de variable;

la figura 7 ilustra un nodo de comprobación;

30 la figura 8 es un diagrama de bloques esquemático que ilustra un ejemplo de configuración de un transmisor;

la figura 9 es una matriz de comprobación de paridad;

la figura 10 ilustra una matriz de paridad;

las figuras 11a y 11b ilustran una matriz de comprobación de paridad de un código de LDPC y pesos de columna definidos en la especificación DVB-S.2;

35 las figuras 12a y 12b ilustran una disposición de puntos de señal de 16QAM;

la figura 13 ilustra una disposición de puntos de señal de 64QAM;

la figura 14 ilustra una disposición de puntos de señal de 64QAM

la figura 15 ilustra una disposición de puntos de señal de 64QAM;

las figuras 16a a 16d ilustran la operación de un desmultiplexor 25;

40 las figuras 17a a 17d ilustran la operación del desmultiplexor 25;

la figura 18 ilustra un gráfico de Tanner para descodificar un código de LDPC;

las figuras 19a y 19b ilustran una matriz de paridad H_T que tiene una estructura de forma gradual y un gráfico de Tanner correspondiente a la matriz H_T de paridad;

45 la figura 20 ilustra una matriz H_T de paridad de una matriz H de comprobación de paridad correspondiente a un código de LDPC después de haber realizado intercalación de paridad sobre el código de LDPC;

- las figuras 21a y 21b ilustran una matriz de comprobación de paridad convertida;
- la figura 22 ilustra la operación de un intercalador 24 por giro de columna;
- la figura 23 ilustra el número de columnas de una memoria 31 requeridas para intercalación por giro de columna y las direcciones de las posiciones de inicio de escritura;
- 5 la figura 24 ilustra el número de columnas de la memoria 31 requeridas para intercalación por giro de columna y las direcciones de las posiciones de inicio de escritura;
- la figura 25 es un diagrama de flujo que ilustra un procedimiento de transmisión;
- las figuras 26a y 26b ilustran un modelo de trayectoria de comunicación empleada en simulaciones;
- la figura 27 ilustra relaciones entre frecuencias de Doppler f_d y tasas de error obtenidas a partir de simulaciones;
- 10 la figura 28 ilustra relaciones entre frecuencias de Doppler f_d y tasas de error obtenidas a partir de simulaciones;
- la figura 29 es un diagrama de bloques esquemático de un receptor de OFDM codificado que puede ser usado, por ejemplo, con el estándar DVB-T2;
- la figura 30 es un diagrama de flujo que ilustra un procedimiento de recepción;
- la figura 31 ilustra un ejemplo de matriz de comprobación de paridad de un código de LPDC;
- 15 la figura 32 ilustra una matriz (matriz de comprobación de paridad convertida) obtenida al realizar permutación de fila y permutación de columna sobre la matriz de comprobación de paridad;
- la figura 33 ilustra la matriz de comprobación de paridad convertida dividida en matrices unidad de 5x5;
- la figura 34 es un diagrama de bloques que ilustra un ejemplo de configuración de un dispositivo de descodificación que realiza cálculos de nodos P en paralelo;
- 20 la figura 35 muestra un ejemplo de configuración de un descodificador 56 de LDPC;
- la figura 36 es un diagrama de bloques que ilustra un ejemplo de configuración de una realización de un ordenador al que se aplica la invención;
- la figura 37 es un diagrama de bloques de partes del transmisor mostrado en la figura 1 en el que un mapeador de símbolo y un generador de trama ilustran la operación de un intercalador;
- 25 la figura 38 es un diagrama de bloques esquemático del intercalador de símbolo mostrado en la figura 37;
- la figura 39 es un diagrama de bloques esquemático de una memoria de intercalador mostrada en la figura 38, y del correspondiente símbolo desintercalador en el receptor;
- la figura 40 es un diagrama esquemático de bloques del generador de dirección mostrado en la figura 38 para el modo de 32k;
- 30 la figura 41(a) es un diagrama que ilustra los resultados para un intercalador que utiliza el generador de dirección mostrado en la figura 40 para símbolos pares, y la figura 41(b) es un diagrama que los ilustra resultados de simulación de diseño para símbolos impares, mientras que la figura 41(c) es un diagrama que ilustra resultados comparativos para un generador de dirección que utiliza un código de permutación diferente para símbolos pares, y la figura 41(d) es un diagrama correspondiente para símbolos impares.
- 35 la figura 42 es un diagrama de bloques esquemático de un desintercalador de símbolo que aparece en la figura 29;
- la figura 43(a) es un diagrama que ilustra resultados para un intercalador que utiliza el generador de dirección mostrado en la figura 40 para símbolos de OFDM pares, y la figura 43(b) es un diagrama que ilustra resultados para símbolos de OFDM impares. Las figuras 44(a) y 44(b) muestran representaciones de la distancia a la salida del intercalador de sub-portadoras que eran adyacentes a la entrada del intercalador;
- 40 la figura 44 proporciona un diagrama de bloques esquemático del intercalador de símbolo mostrado en la figura 38, que ilustra un modo operativo en el que la intercalación se realiza de acuerdo con un modo de intercalación impar solamente; y
- la figura 45 proporciona un diagrama de bloques esquemático del desintercalador de símbolo mostrado en la figura 42, que ilustra el modo operativo en el que se realiza la intercalación de acuerdo con un modo de intercalación impar únicamente.

Descripción de realizaciones preferidas

La figura 1 proporciona un ejemplo de diagrama de bloques de un transmisor de OFDM que puede ser usado, por ejemplo, para transmitir imágenes de video y señales de audio de acuerdo con el estándar DVB-T2. En la figura 1, una fuente de programa genera datos que van a ser transmitidos por el transmisor de OFDM. Un codificador 2 de video, un codificador 4 de audio y un codificador 6 de datos, generan video, audio y otros datos que van a ser transmitidos, los cuales son alimentados a un multiplexor 10 de programa. La salida del multiplexor 10 de programa forma una corriente multiplexada con otra información requerida para comunicar el video, el audio o los otros datos. El multiplexor 10 proporciona una corriente sobre un canal 13 de conexión. Pueden existir muchas corrientes multiplexadas de ese tipo que son alimentadas a diferentes ramas A, B, etc. Por simplicidad, solamente se va a describir la rama A.

Según se muestra en la figura 1, un transmisor 11 de OFDM recibe la corriente en un bloque 20 de adaptación de multiplexor y de dispersión de energía. El bloque 20 de adaptación de multiplexor y de dispersión de energía aleatoriza los datos y alimenta los datos apropiados a un codificador 21 de corrección de error adelantado que realiza codificación de corrección de error de la corriente. Un intercalador de bits 22 ha sido previsto para intercalar los bits de datos codificados que, para el ejemplo de DVB-T2, son la salida del codificador de LDPC. La salida del intercalador 22 de bits se alimenta en un bit a un mapeador 26 de constelación, el cual mapea grupos de bits en un punto de constelación, que va a ser usado para transportar los bits de datos codificados. Las salidas procedentes del bit en el mapeador 26 de constelación son etiquetas de punto de constelación que representan componentes reales e imaginarios. Las etiquetas de punto de constelación representan símbolos de datos formados a partir de dos o más bits dependiendo del esquema de modulación utilizado. Éstas serán mencionadas como células de datos. Estas células de datos se hacen pasar a través de un intercalador por tiempo 30 cuyo efecto es el de intercalar células de datos resultantes de múltiples palabras de código de LDPC. Las células de datos procedentes del intercalador por tiempo 30 son alimentadas a continuación a un generador 27 de modulación y trama, el cual mapea las células de datos sobre símbolos de modulación para su transmisión.

Las células de datos son recibidas en el interior de la unidad 27 de modulación por medio de un generador 32 de trama, con células de datos producidas por la rama B de la figura 1, a través de otros canales 31. El generador 32 de trama forma entonces muchas células de datos según secuencias que van a ser transportadas sobre símbolos de OFDM, donde un símbolo de OFDM comprende un número de células de datos, siendo cada célula de datos mapeada sobre una de las sub-portadoras. El número de sub-portadoras dependerá del modo de operación del sistema, el cual puede incluir uno de entre 1k, 2k, 4k, 8k, 16k o 32k, cada uno de los cuales proporciona un número diferente de sub-portadoras de acuerdo, por ejemplo, con la siguiente tabla:

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

Número de sub-portadoras adaptadas a partir de DVB-T/H

Así, en un ejemplo, el número de sub-portadoras para el modo de 32k es de veinticuatro mil ciento noventa y dos. Para el sistema de DVB-T2, el número de sub-portadoras por símbolo de OFDM puede variar dependiendo del número de portadoras piloto y de otras portadoras reservadas. Así, en DVB-T2, a diferencia con DVB-T, el número de sub-portadoras para portar datos no es fijo. Los radiodifusores pueden seleccionar uno de los modos operativos de entre 1k, 2k, 4k, 8k, 16k, 32k, teniendo cada uno de ellos una gama de sub-portadoras para datos por símbolo de OFDM, siendo el máximo disponible para cada uno de estos modos de 1024, 2048, 4096, 8192, 16384, 32768, respectivamente. En DVB-T2 una trama de capa física está compuesta por muchos símbolos de OFDM. Tipicamente la trama empieza con uno o más símbolos de OFDM de preámbulo o P2, los cuales van seguidos a continuación por un número de símbolos de OFDM portadores de carga útil. El final de la trama de capa física está marcado por símbolos de cierre de trama. Para cada modo operativo, el número de sub-portadoras puede ser diferente para cada tipo de símbolo. Además, esto puede variar para cada uno de acuerdo a si se elige extensión de ancho de banda, si se ha habilitado reserva de tono y de acuerdo con el patrón de sub-portadora piloto que haya sido seleccionado. Como tal, una generalización a un número específico de sub-portadoras por símbolo de OFDM es difícil. Sin embargo, el intercalador de frecuencia para cada modo puede intercalar cualquier símbolo cuyo número de sub-portadoras sea más pequeño, o igual, que el número máximo disponible de sub-portadoras para el modo dado. Por ejemplo, en el modo de 1k, el intercalador podría trabajar para símbolos con un número de sub-portadoras que sea menor que, o igual a 1024, y para el modo de 16k, con un número de sub-portadoras que sea menor que, o igual a 16384.

- La secuencia de células de datos que va a ser portada en cada símbolo de OFDM se hace pasar a continuación al intercalador 33 de símbolo. El símbolo de OFDM es generado a continuación por un bloque 37 generador de símbolo de OFDM que introduce señales piloto y de sincronización desde un generador 36 de señal piloto e incorporada. Un modulador 38 de OFDM conforma a continuación el símbolo de OFDM en el dominio del tiempo, siendo alimentado a un procesador 40 de inserción de protección para generar un intervalo de seguridad entre símbolos, y a continuación a un convertidor digital a analógico, y finalmente a un amplificador de RF dentro de un extremo 44 frontal de RF para su difusión eventual por el transmisor de OFDM desde una antena 46.
- 5

Las realizaciones de la presente invención proporcionan un sistema de comunicación de OFDM que incluye un intercalador de bit para intercalar bits codificados con un codificador de LDPC en combinación con un intercalador de símbolo, el cual intercala símbolos que representan a los uno o más bits intercalados y codificados sobre las señales de sub-portadora de un símbolo de OFDM. Tanto el intercalador de bit como el intercalador de símbolo, de acuerdo con ejemplos de realización, van a ser descritos en los párrafos siguientes, empezando por el intercalador de bit, el cual se describe con codificación de LDPC.

Intercalador de bit para codificación de LDPC

15 Códigos de corrección de errores de LDPC

Los códigos de LDPC tienen un alto comportamiento de corrección de error y han empezado recientemente a ser usados en esquemas de comunicación que incluyen radiodifusión digital por satélite tal como DVB-S.2, el cual ha entrado en uso en Europa (por ejemplo, véase DVB-S.2: ETSI EN 302 307 V1.1.2 (06-2006)). Aplicar códigos de LDPC a radiodifusión digital terrestre de próxima generación, está siendo también objeto de discusión.

20 Estudios recientes muestran que el rendimiento de los códigos de LDPC se aproxima al Límite de Shannon según se incrementa la longitud del código, de manera similar a los turbo códigos. Puesto que los códigos de LDPC tienen la propiedad de que la distancia mínima es proporcional a la longitud del código, los códigos de LDPC tienen las ventajas de que las características de probabilidad de error de bloque son excelentes y un piso de error, que es un fenómeno observado en asociación con características de descodificación de turbo códigos o similares, raramente ocurre.

25

Ahora se hará referencia en detalle a tales códigos de LDPC. Los códigos de LDPC son códigos lineales. Aunque los códigos de LDPC no son necesariamente binarios, la descripción que sigue se hará con referencia a códigos de LDPC binarios.

30 La característica más importante de los códigos de LDPC consiste en que una matriz de comprobación de paridad que define cada código de LDPC es una matriz de dispersión que tiene un número muy pequeño de elementos de "1", es decir que los elementos de la misma son en su mayor parte "0".

La figura 2 ilustra un ejemplo de matriz H de comprobación de paridad de un código de LDPC.

Cada columna de la matriz H de comprobación de paridad de la figura 2 tiene un peso de 3 (es decir, 3 elementos de "1") y cada fila tiene un peso de 6 (es decir, 6 elementos de "1").

35 La codificación basada en códigos de LDPC (es decir, codificación LDPC) se realiza, por ejemplo, calculando una matriz G de generación basada en una matriz H de comprobación de paridad y multiplicando la matriz G de generación por bits de información para generar una palabra clave (código de LDPC).

Específicamente, un codificador de LDPC calcula en primer lugar una matriz G de generación que satisface una ecuación $GH^T = 0$ con una matriz transpuesta H^T de la matriz H de comprobación de paridad. Aquí, cuando la matriz 40 G de generación es una matriz K x N, el codificador multiplica la matriz G de generación por una secuencia de bit de información de K bits (vector u) para generar un palabra clave c (= uG) N bits. La palabra clave (código de LDPC) generada por el codificador es recibida por un lado de recepción a través de una trayectoria de comunicación.

45 El código de LDPC puede ser descodificado mediante un algoritmo de paso de mensaje propuesto por Gallager y denominado "algoritmo de descodificación probabilística". El algoritmo de paso de mensaje utiliza propagación creíble sobre un gráfico de Tanner que incluye nodos de variable (también conocidos como nodos de mensaje) y nodos de comprobación. En la descripción que sigue, cada uno de los nodos de variable y de los nodos de comprobación será mencionado simplemente como un "nodo" según sea apropiado.

La figura 3 ilustra un procedimiento para descodificar un código de LDPC.

50 En lo que sigue, un valor real que exprese, a modo de relación de probabilidad logarítmica, la probabilidad de que un bit de código $i^{\text{ésimo}}$ de un código de LDPC (una palabra clave) recibido por un lado de recepción tenga un valor de "0", se menciona como valor u_{0i} recibido, según sea apropiado. Además, un mensaje de salida desde un nodo de comprobación se mencionada como u_i y un mensaje de salida desde un nodo de variable se menciona como v_i .

Un código de LDPC se descodifica de la siguiente manera. En primer lugar, según se muestra en la figura 3, en la etapa S11, se recibe un código de LDPC, se inicializa un mensaje u_i (mensaje de nodo de comprobación) en "0", y

una variable k, que es un valor entero como contador de un proceso iterativo, se inicializa en "0". A continuación, el procedimiento avanza hasta la etapa S12. En la etapa S12, se realiza un cálculo (cálculo de nodo de variable) representado por la ecuación (1) en base al valor u_{0i} recibido, obtenido por recepción del código de LDPC para obtener un mensaje v_i (mensaje de nodo de variable) y a continuación se realiza un cálculo (cálculo de nodo de comprobación) representado por la ecuación (2), en base al mensaje v_i , para obtener un mensaje u_j .

5

ECUACIÓN 1

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

ECUACIÓN 2

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

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

15 Las gamas para cálculo respectivas en el cálculo de nodo de variable de la ecuación (1) y el cálculo de nodo de comprobación de la ecuación (2) van desde 1 hasta d_v-1 y desde d_c-1 puesto que un mensaje recibido desde un borde (es decir, una línea que conecta un nodo de variable con un nodo de comprobación, cada uno con el otro) que presenta a la salida el mensaje, es excluido de cada uno de los cálculos de las ecuaciones (1) y (2). Realmente, el cálculo de nodo de comprobación de la ecuación (2) se realiza utilizando recursivamente, según se muestra en la ecuación (4), una tabla creada previamente de una función $R(v_1, v_2)$ mostrada en la ecuación (3) que se define como una salida con respecto a dos entradas v_1 y v_2 .

20 ECUACIÓN 3

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

ECUACIÓN 4

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

25 En la etapa S12, la variable k se incrementa en "1" y el procedimiento avanza hasta la etapa S13. En la etapa S13, se determina si la variable k es o no mayor que un número C predeterminado de iteraciones de descodificación. 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.

30 Si se determina en la etapa S13 que la variable k es mayor que C, el procedimiento avanza hasta la etapa S14 para realizar un cálculo representado por la ecuación (5) para obtener y presentar a la salida un mensaje v_i como resultado de la descodificación final. A continuación, termina el procedimiento de descodificación de código de LDPC.

ECUACIÓN 5

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

35 Aquí, a diferencia con el cálculo de nodo de variable de la ecuación (1), el cálculo de la ecuación (5) se realiza utilizando mensajes u_j procedentes de todos los bordes conectados al nodo de variable.

La figura 4 ilustra un ejemplo de matriz H de comprobación de paridad de un código de LDPC de (3, 6) con una tasa de código de 1/2 y una longitud de código de 12.

Al igual que con la matriz H de comprobación de paridad de la figura 2, la matriz H de comprobación de paridad de la figura 4 tiene un peso de columna de 3 y un peso de fila de 6.

La figura 5 ilustra un gráfico de Tanner de la matriz H de comprobación de paridad de la figura 4.

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

Específicamente, cuando un elemento de la fila $j^{\text{ésima}}$ y de la columna $i^{\text{ésima}}$ de una matriz de comprobación de paridad es “1”, un nodo “=” de variable $i^{\text{ésimo}}$ (contando desde la parte superior) y un nodo “+” de comprobación $j^{\text{ésimo}}$ (contando desde la parte superior) están conectados a través de un borde en la figura 5. El borde indica que un bit de código correspondiente al nodo de variable tiene una restricción correspondiente al nodo de comprobación.

Un algoritmo de suma y producto, que es un algoritmo de descodificación de código de LDPC, realiza repetitivamente un cálculo de nodo de variable y un cálculo de nodo de comprobación.

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 el cálculo, se obtiene de acuerdo con el cálculo de nodo de variable de la ecuación (1), utilizando un valor u_{0i} recibido y mensajes u_1 y u_2 procedentes de los bordes restantes conectados al nodo de variable. Los mensajes correspondientes a otros bordes se obtienen de la misma manera.

La figura 7 ilustra un cálculo de nodo de comprobación realizado en un nodo de comprobación.

La ecuación (2) anterior para el cálculo de nodo de comprobación puede ser re-escrita como ecuación (6) utilizando una ecuación de relación de $a \times b = \exp\{\ln(|a|) + \ln(|b|)\} \times \text{sign}(a) \times \text{sign}(b)$, donde $\text{sign}(x)$ es 1 cuando $x > 0$, y -1 cuando $x < 0$.

20 ECUACIÓN 6

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

Adicionalmente, cuando se define una función $\phi = \ln(\tanh(x/2))$ cuando $x > 0$, se satisface una ecuación $\phi^{-1}(x) = 2\tanh^{-1}(e^{-x})$ y por lo tanto la ecuación (6) puede ser reorganizada como ecuación (7):

ECUACIÓN 7

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

En el nodo de comprobación, el cálculo de nodo de comprobación de la ecuación (2) se realiza de acuerdo con la ecuación (7).

Es decir, en el nodo de comprobación, se obtiene un mensaje u_j correspondiente a un borde para cálculo, de acuerdo con el cálculo de nodo de comprobación de la ecuación (7) utilizando mensajes v_1, v_2, v_3, v_4 y v_5 procedentes de los bordes restantes conectados al nodo de comprobación como se muestra en la figura 7. Otros mensajes correspondientes a otros bordes son obtenidos de la misma manera.

La función $\phi(x)$ en la ecuación (7) puede ser expresada de tal modo que $\phi(x) = \ln((e^x + 1)/(e^x - 1))$ y $\phi^{-1}(x) = \phi^{-1}(x)$ cuando $x > 0$. Cuando las funciones $\phi(x)$ y $\phi^{-1}(x)$ están incorporadas en hardware, ambas pueden ser incorporadas utilizando la misma Tabla de Búsqueda Ascendente (LUT).

Aunque se sabe que los códigos de LDPC presentan un comportamiento muy alto en trayectorias de comunicación de Ruido Gaussiano Blanco Aditivo (AWGN), también se ha sabido en los últimos años que los códigos de LDPC tienen un alto rendimiento de corrección de error en otras trayectorias de comunicación, en comparación con códigos convolucionales o con códigos convolucionales de Reed Solomon (RS) concatenados en el pasado.

- 5 Es decir, cuando se elige un código que tiene un comportamiento excelente en una trayectoria de comunicación de AWGN, el código elegido muestra por lo general un excelente comportamiento en otras trayectorias de comunicación, superior a otros códigos.

Por ejemplo, puesto que se aplican códigos de LDPC a radiodifusión digital terrestre, se ha sugerido que los códigos de LPDC definidos en los esquemas de especificación y modulación de DVB-S.2 sean combinados y se proporcione un intercalador de bit que intercale bits de código de un código de LDPC entre un codificador de LDPC y un modulador para mejorar el rendimiento de códigos de LDPC en trayectorias de comunicación de AWGN.

10 Sin embargo, las borraduras o los errores de ráfaga que pueden ocurrir en las trayectorias de comunicación se supone que son ondas de superficie. Por ejemplo, en un sistema de Multiplexado por División de Frecuencia Ortogonal (OFDM), un símbolo específico puede ser borrado (es decir, llevado a potencia cero) debido al retardo de 15 un eco, lo cual es una trayectoria distinta de la trayectoria principal, en entornos multi-trayectoria en los que la Relación de Deseada respecto a Indeseada (D/U) es de 0 dB, de tal modo que la potencia de la trayectoria principal como potencia deseada es igual a la potencia de eco como potencia indeseada.

20 Cuando la D/U es de 0 dB, todos los símbolos de OFDM en un momento específico pueden ser también borrados (es decir, caer a potencia cero) debido a una frecuencia Doppler en una titilación que es una trayectoria de comunicación a la que ha sido añadido un eco con una frecuencia Doppler aplicada al mismo y que tiene un retardo de "0".

Adicionalmente, los errores de ráfaga pueden producirse debido a una potencia inestable o unas condiciones indeseadas del cableado que va desde las antenas hasta los receptores.

25 En la técnica relacionada, los códigos de corrección de error que tienen un comportamiento excelente en trayectorias de comunicación de AWGN, son también utilizados frecuentemente en trayectorias de comunicación en las que ocurren errores de ráfaga o borraduras según se ha descrito anteriormente.

30 Por otra parte, cuando se descodifica un código de LDPC, se calcula un nodo de variable correspondiente no sólo a una columna de una matriz H de comprobación de paridad sino también a un bit de código del código de LDPC de acuerdo con la ecuación (1), que incluye la adición de un bit de código (un valor u_{0i} recibido) de un código de LDPC según se muestra en la figura 6. Por lo tanto, la precisión del mensaje obtenido se reduce si se produce un error en 35 un bit de código utilizado en el cálculo de nodo de variable.

Adicionalmente, cuando se descodifica un código de LDPC, se calcula un nodo de comprobación de acuerdo con la ecuación (7) utilizando un mensaje obtenido en un nodo de variable conectado al nodo de comprobación. Por lo tanto, el rendimiento de descodificación se reduce si ocurre simultáneamente un error que incluya una borradura en 40 (una pluralidad de bits de código de LDPC correspondientes a) una pluralidad de nodos de variable conectados a cada uno de un gran número de nodos de comprobación.

Más específicamente, por ejemplo cuando dos o más nodos de variable conectados a un nodo de comprobación son 45 borrados simultáneamente, el nodo de comprobación devuelve un mensaje con una probabilidad de "0" que es igual a la probabilidad de "1" en cada nodo de variable conectado al nodo de comprobación. En este caso, el nodo de comprobación que devuelve el mensaje con iguales probabilidades de "0" y de "1" no contribuye a un proceso de descodificación que consiste en un conjunto de un cálculo de nodo de variable y un cálculo de nodo de comprobación. Esto incrementa el número de procesos de descodificación requeridos, reduciendo con ello el rendimiento de descodificación e incrementando el consumo de potencia de un receptor que realiza descodificación de código de LDPC.

50 De ese modo, existe una necesidad de proporcionar un método que incremente la resistencia a los errores de ráfaga o borraduras mientras se mantiene el rendimiento de las trayectorias de comunicación de AWGN.

En este punto, puede ser posible incrementar el rendimiento de descodificación si se proporciona un intercalador de bit que intercale bits de código de un código de LDPC entre un codificador de LDPC y un modulador para mejorar el rendimiento de códigos de LDPC en trayectorias de comunicación de AWGN según se ha descrito anteriormente, y 55 si el intercalador de bit está diseñado de modo que pueda realizar intercalación que reduzca la probabilidad de que ocurra un error simultáneamente en (una pluralidad de bits de código de un código de LDPC correspondiente a) una pluralidad de nodos de variable conectados a un nodo de comprobación.

La invención ha sido realizada en vista de las circunstancias anteriores y proporciona un aparato y un método de procesamiento de datos que pueden incrementar la resistencia a los errores en los bits de código de códigos de LDPC tales como errores de ráfaga o borraduras.

- Un aparato de procesamiento de datos para intercalar datos de acuerdo con una realización de la invención incluye un intercalador de paridad para realizar intercalación de paridad sobre un código de Comprobación de Paridad de Baja Densidad (LDPC) obtenido al realizar codificación de LDPC de acuerdo con una matriz de comprobación de paridad que incluye una matriz de paridad correspondiente a bits de paridad del código de LDPC, teniendo la matriz de paridad una estructura progresiva, de modo que se intercale un bit de paridad del código de LDPC en una posición de bit de paridad diferente.
- Un método de procesamiento para un aparato de procesamiento que intercala datos de acuerdo con una realización de la invención incluye la etapa de hacer que el aparato de procesamiento de datos realice intercalación de paridad en un código de Comprobación de Paridad de Baja Densidad (LDPC) obtenido al realizar codificación de LDPC de acuerdo con una matriz de comprobación de paridad que incluye una matriz de paridad correspondiente a bits de paridad del código de LDPC, teniendo la matriz de paridad una estructura gradual, de modo que se intercale un bit de paridad del código de LDPC en una posición de bit de paridad diferente.
- Es decir, de acuerdo con las realizaciones de la invención, la intercalación de paridad se lleva a cabo sobre un código de LDPC obtenido al realizar codificación de LDPC de acuerdo con una matriz de comprobación de paridad que incluye una matriz de paridad estructurada de forma gradual, correspondiente a bits de paridad del código de LDPC de modo que se intercalan bits de paridad del código de LDPC en posiciones de bit de paridad diferentes.
- El aparato de procesamiento de datos puede ser un dispositivo independiente y puede ser también un bloque interno incluido en un dispositivo.
- Explicación detallada de un intercalador de bit de ejemplo
- La figura 8 proporciona una representación más detallada de partes del transmisor mostrado en la figura 1, las cuales ilustran la operación del intercalador de bit. En particular, el codificador 21 de LDPC va a ser descrito a continuación. El codificador 21 de LDPC codifica los datos objetivo en un bit de datos codificados de LDPC que incluyen bits de información correspondientes a los datos objetivo de acuerdo con una matriz de comprobación de paridad, en la que una matriz de paridad correspondiente a bits de paridad del código de LDPC tiene una estructura gradual.
- Específicamente, el codificador 21 de LDPC codifica datos objetivo en un código de LDPC definido, por ejemplo, de acuerdo con la especificación DVB-S.2 y presenta a la salida el código de LDPC.
- El código de LDPC definido en la especificación DVB-S.2 es un código Acumulado de Repetición Irregular (IRA), y una matriz de paridad en una matriz de comprobación del código de LDPC tiene una estructura gradual. Detalles de la matriz de paridad y de la estructura gradual de la misma van a ser descritos en lo que sigue. Un ejemplo del código IRA se encuentra descrito en "Repetición Irregular-Códigos Acumulados", H. Jin, A. Khandekar, y R. J. McEliece, en Procedimientos de 2º Simposio Internacional sobre Turbo códigos y Tópicos Relacionados, pp. 1-8, Septiembre 2000.
- La salida de código de LDPC procedente del codificador 21 de LDPC se suministra al intercalador 22 de bit.
- El intercalador 22 de bit es un aparato de procesamiento de datos que intercala datos e incluye un intercalador 23 de paridad, un intercalador 24 por giro de columna, y un desmultiplexor 25.
- El intercalador 23 de paridad realiza intercalación de paridad sobre el código de LDPC procedente del codificador 21 de LDPC para intercalar bits de paridad del código de LDPC en posiciones de bit de paridad diferentes, y proporciona el código de LDPC de paridad intercalada al intercalador 24 por giro de columna.
- El intercalador 24 por giro de columna realiza intercalación por giro de columna sobre el código de LDPC procedente del intercalador 23 de paridad y a continuación proporciona el código de LDPC por giro de columna intercalado al desmultiplexor 25.
- De ese modo, el código de LDPC es transmitido después de que dos o más bits de código del código de LDPC hayan sido mapeados respecto a un símbolo ortogonalmente modulado por medio de la unidad 26 de mapeo que se describe en lo que sigue.
- El intercalador 24 por giro de columna realiza permutación (por ejemplo, intercalación por giro de columna que se describe a continuación) sobre los bits de código del código de LDPC recibido desde el intercalador 23 de paridad de tal modo que una pluralidad de bits de código del código de LDPC correspondiente a "1" en una fila arbitraria de la matriz de comprobación de paridad utilizada por el codificador 21 de LDPC no son mapeados respecto a un símbolo.
- El desmultiplexor 25 lleva a cabo un proceso de reordenación sobre el código de LDPC recibido desde el intercalador 24 por giro de columna de tal modo que son reordenadas las posiciones de dos o más bits de código del código de LDPC que van a ser mapeados en un símbolo, obteniendo un código de LDPC de resistencia incrementada respecto a un AWGN, y a continuación proporciona el código de LDPC obtenido a la unidad 26 de

mapeo.

La unidad de mapeo 26 mapea dos o más bits de código del código de LDPC procedente del desmultiplexor 25 en cada punto de señal que se determine de acuerdo con un esquema de modulación que el modulador 27 ortogonal utiliza para realizar modulación ortogonal (modulación multi-valor).

- 5 Más específicamente, la unidad 26 de mapeo convierte el código de LDPC procedente del desmultiplexor 25 en símbolos (valores de símbolo) representados por puntos de señal determinados de acuerdo con el esquema de modulación sobre un plano IQ (constelación IQ) definido con un eje I que representa I componentes de los mismos niveles que las portadoras, y Q ejes que representan Q componentes ortogonales a las portadoras.

10 El esquema de modulación que utiliza el transmisor de OFMD de la figura 1 utilizado para llevar a cabo modulación ortogonal incluye un esquema de modulación definido en la especificación DVB-T, ejemplos del cual incluyen Modulación por Desplazamiento de Fase en Cuadratura (QPSK), Modulación de Amplitud de Cuadratura 16 (16QAM), 64QAM, 256QAM, y 4096QAM. Uno de los esquemas de modulación que utiliza el modulador 27 ortogonal para llevar a cabo modulación ortogonal está preestablecido, por ejemplo, mediante operación por un operador que opera el transmisor de la figura 1. Ejemplos de otros esquemas de modulación que pueden usar el modulador 27 ortogonal para realizar modulación ortogonal incluyen Modulación de amplitud de Pulso 4 (4PAM).

15 El símbolo obtenido en la unidad 26 de mapeo es proporcionado al intercalador por tiempo, el cual puede intercalar diferentes palabras de código de LDPC sobre diferentes símbolos de OFMD. La salida del intercalador por tiempo 30 es alimentada a continuación al generador de trama de la figura 1. Las partes restantes del transmisor mostrado en la figura 1 realizan modulación ortogonal de las señales de sub-portadora del símbolo de OFMD recibido desde la unidad 26 de mapeo para producir una señal modulada y a continuación transmitir la señal modulada.

20 La figura 9 ilustra una matriz H de comprobación de paridad que utiliza el codificador 21 de LDPC de la figura 8 para codificación de LDPC.

25 La matriz H de comprobación de paridad tiene una estructura de Matriz de Generación de Baja Densidad (LDGM) y puede ser expresada mediante la ecuación “ $H = [H_A|H_T]$ ” que incluye una matriz H_A de información como componente izquierdo y una matriz H_T de paridad como componente derecho, donde la matriz H_A de información corresponde a bits de información entre los bits de código del código de LDPC, y la matriz H_T de paridad corresponde a bits de paridad.

30 Aquí, el número de bits de información y el número de bits de paridad entre bits de código de un código de LDPC (una palabra clave) 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 N (= K + M).

La longitud K de información y la longitud M de paridad de un código de LDPC de una longitud de código N, se determinan en base a una tasa de código. De ese modo, la matriz H de comprobación de paridad es una matriz de M x N. Adicionalmente, la matriz H_A de información es una matriz de M x K y la matriz H_T de paridad es una matriz de M x M.

35 La figura 10 ilustra una matriz H_T de paridad de una matriz H de comprobación de paridad de un código de LDPC definido en la especificación DVB-S.2.

40 La matriz H_T de paridad de la matriz H de comprobación de paridad de un código de LDPC definida en la especificación DVB-S.2, tiene una estructura gradual tal que los elementos de “1” de la matriz H_T de paridad están dispuestos de una forma gradual según se muestra en la figura 10. La primera fila de la matriz H de comprobación de paridad tiene un peso de 1 y las restantes filas tienen un peso de 2. La última columna de la matriz H de comprobación de paridad tiene un peso de 1 y las restantes columnas tienen un peso de 2.

45 El código de LDPC de la matriz H de comprobación de paridad que tiene la matriz H_T de paridad de estructura gradual, puede ser generada fácilmente utilizando la matriz H de comprobación de paridad.

Más específicamente, supóngase que un vector de fila c representa un código (palabra clave) de LDPC y supóngase que C^T representa un vector columna obtenido por transposición del vector fila. Adicionalmente, supóngase que un vector fila A representa una parte de bit de información del vector fila c, el cual es el código de LDPC, y supóngase que un vector fila T represente una parte de bit de paridad del mismo.

50 En este caso, el vector fila c puede ser expresado mediante una ecuación “ $c = [A|T]$ ” que incluye un vector fila A como componente izquierdo y un vector fila T como componente derecho, donde el vector fila A corresponde a bits de información y el vector fila T corresponde a bits de paridad.

La matriz H de comprobación de paridad y el vector fila $c = [A|T]$, que corresponde al código de LDPC, necesitan satisfacer una ecuación “ $Hc^T = 0$ ”. De ese modo, el valor de cada elemento del vector fila T que corresponda a bits de paridad incluidos en el vector fila $c = [A|T]$ puede ser obtenido secuencialmente estableciendo un elemento en cada fila del vector columna Hc^T en la ecuación “ $Hc^T = 0$ ” en cero por un orden que empieza desde el elemento de la

primera fila, cuando la matriz H_T de paridad en la matriz $H = [H_A|H_T]$ de comprobación de paridad tiene una estructura gradual según se muestra en la figura 10.

Las figuras 12a y 12b ilustran una matriz H de comprobación de paridad de un código de LDPC y pesos de columna definidos en la especificación DVB-S.2.

- 5 Es decir, la figura 11A ilustra una matriz H de comprobación de paridad de un código de LDPC definido en la especificación DVB-S.2.

Antes de que las columnas $KX^{\text{ésimas}}$ de la matriz H de comprobación de paridad tengan un peso de columna de X, las siguientes columnas K3 tienen un peso de columna de 3, las siguientes columnas M-1 tienen un peso de columna de 2, y la última columna tiene un peso de columna de 1.

- 10 En este punto, la suma de los números de columnas "KX + K3 + M-1 + 1" es igual a una longitud de código de N.

En la especificación DVB-S.2, los números de columnas KX, K3 y M (longitud de paridad) y el peso de columna X se definen según se muestra en la figura 11B.

Es decir, la figura 11B ilustra los números de columnas KX, K3 y M, y el peso de columna X, para cada tasa de código de códigos de LDPC definidos en la especificación DVB-S.2.

- 15 Los códigos de LDPC con longitudes de código N respectivas de 64800 bits y 16200 bits, están definidos en la especificación DVB-S.2.

Adicionalmente, se definen 11 tasas nominales de 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9 y 9/10 para el código de LDPC cuya longitud de código N sea 64800, y se definen 10 tasas de código nominales de 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6 y 8/9 para el código de LDPC cuya longitud de código N sea 16200 bits según se ha mostrado en la figura 11B.

- 20 Para códigos de LDPC, se sabe que la tasa de error de un bit de código disminuye según se incrementa el peso de una columna correspondiente al bit de código en una matriz H de comprobación de paridad.

En el caso de una matriz H de comprobación de paridad definida en la especificación DVB-S.2 mostrada en las figuras 12a y 12b, el peso de una columna se incrementa según se reduce el número ordinal de la columna (es decir, según la columna está más cerca del extremo izquierdo de la matriz H de comprobación de paridad), y por lo tanto un bit de código de un código de LDPC correspondiente a la matriz H de comprobación de paridad es más robusto contra (resistente a) errores según disminuye el número ordinal del bit de código (es decir, el primer bit de código es el más resistente) y es más susceptible de errores según se incrementa el número ordinal del bit de código (es decir, el último bit de código es el más susceptible).

- 30 Las figuras 12A y 12B ilustran una disposición de (puntos de señal correspondientes a) 16 símbolos en un plano IQ en el caso de que se realice 16QAM en el modulador 27 ortogonal de la figura 8.

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

En 16QAM, un símbolo representa 4 bits y se proporcionan 16 ($= 2^4$) símbolos. Adicionalmente, 16 símbolos se encuentran dispuestos en un cuadrado de 4 x 4 símbolos en las direcciones I y Q, con centro en el origen del plano IQ.

En este punto, cuando y_0 , y_1 , y_2 e y_3 indican 4 bits representados por un símbolo de 16QAM, empezando secuencialmente desde el bit más significativo (MSB), la unidad 26 de mapeo de la figura 8 mapea bits de código de un código de LDPC respecto a un símbolo de 4 bits y_0 a y_3 correspondientes a 4 bits de código en el caso de que el esquema de modulación sea 16QAM.

- 40 La figura 13B muestra límites de bit de cuatro bits y_0 a y_3 representados por un símbolo de 16QAM.

En este caso, un límite de bit 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 muestra en la figura 13B, un límite correspondiente al eje Q en el plano IQ que es el único límite de bit para el primer bit (es decir, el MSB) y_0 de los cuatro bits y_0 a y_3 representados por un símbolo de 16QAM y un límite correspondiente al eje I en el plano IQ es el único límite de bit para el segundo bit (es decir, el segundo MSB) y_1 .

Adicionalmente, dos límites, uno entre la primera y la segunda columnas de símbolos (contando desde el lado izquierdo) entre los símbolos de 4 x 4 y el otro entre la tercera y la cuarta columnas, son límites de bit para el tercer bit y_2 .

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

Cada bit y_i representado por un símbolo es más resistente a errores según se incrementa el número de símbolos distantes del límite de bit, y es más susceptible de errores según se incrementa el número de símbolos cerca del límite de bit.

- 5 Cuando un bit resistente a (robusto frente a) errores se menciona como "bit fuerte" y un bit susceptible a (sensible a) errores se menciona como "bit débil", el primer bit (es decir, el MSB) y_0 y el segundo bit y_1 son bits fuertes, y el tercer bit y_2 y el cuarto bit y_3 son bits débiles según se muestra en las figuras 1A y 12B.

Las figuras 13 a 15 ilustran una disposición de (puntos de señal correspondientes a) 64 símbolos en un plano IQ en el caso de que se realice 64QAM en el modulador 27 ortogonal de la figura 8.

- 10 En 64QAM, un símbolo representa 4 bits y se proporcionan 64 ($= 2^6$) símbolos. Adicionalmente, 64 símbolos están dispuestos en un cuadrado de 8 x 8 símbolos en las direcciones I y Q, con centro en el origen del plano IQ.

En este caso, cuando y_0 , y_1 , y_2 , y_3 , y_4 e y_5 indican 6 bits representados por un símbolo de 64QAM, empezando secuencialmente desde el bit más significativo (MSB), la unidad 26 de mapeo de la figura 8 mapea 6 bits de código de un código de LDPC respecto a un símbolo de 6 bits y_0 a y_5 correspondientes a los 6 bits de código en el caso de que el esquema de modulación sea 64QAM.

- 15 La figura 13 muestra límites de bit del primer 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 muestra límites de bit del tercer y cuarto bits y_2 e y_3 , y la figura 15 muestra límites de bit del quinto y sexto bits y_4 e y_5 .

- 20 Un límite de bit está presente para cada uno del primer y segundo bits y_0 e y_1 según se muestra en la figura 14. Dos límites de bit están presentes para cada uno del tercer y cuarto bits y_2 e y_3 mostrados en la figura 14, y cuatro límites de bit están presentes para cada uno del quinto y sexto bits y_4 e y_5 según se muestra en la figura 15.

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

- 25 A partir de las figuras 13, 14 y 15 se puede ver que, en el caso de bits de símbolos modulados ortogonalmente, los bits más significativos son bits fuertes, y los menos significativos son bits débiles.

La salida de código de LDPC desde el codificador 21 de LDPC de la figura 8 incluye bits de código susceptibles a error y bits de código resistentes a error según se ha descrito en lo que antecede con referencia a la figura 11.

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

- 30 En consecuencia, cuando son mapeados bits de código susceptibles de error de un código de LDPC respecto a bits débiles de símbolos modulados ortogonalmente, se reduce la resistencia global a errores.

Por lo tanto, la invención sugiere un intercalador que intercale bits de código de un código de LDPC de modo que los bits de códigos susceptibles de error del código de LDPC sean mapeados respecto a bits fuertes de símbolos modulados ortogonalmente.

- 35 El desmultiplexor de la figura 8 realiza la función de ese intercalador.

Las figuras 16A a 16D ilustran la operación del desmultiplexor 25 de la figura 8.

Específicamente, la figura 16A ilustra un ejemplo de configuración funcional del desmultiplexor 25.

- 40 El desmultiplexor 25 incluye una memoria 31 y una unidad 32 de reordenación. Se proporciona un código de LDPC a la memoria 31. La memoria 31 tiene una capacidad de almacenamiento para almacenar mb bits en la dirección de una fila (horizontal) y almacenar N/m bits en la dirección de una columna (vertical). Los bits del código de LDPC proporcionados a la memoria 31 son escritos en la dirección de una columna en la memoria 31 y son leídos en la dirección de una fila desde la memoria 31, y los bits de código leídos son proporcionados a la unidad 32 de reordenación.

- 45 En este caso, "m" representa el número de bits de código del código de LDPC mapeados en un símbolo y "b" representa un número entero específico positivo (es decir, un factor) por el que se multiplica "m" para obtener un múltiplo entero de "m". Adicionalmente, "N" (= longitud de información K + longitud de paridad M) representa la longitud de código del código de LDPC según se ha descrito con anterioridad.

La figura 16A ilustra un ejemplo de configuración del desmultiplexor 25 cuando el esquema de modulación es 64QAM. En consecuencia, el número de bits de código "m" de un código de LDPC mapeado en un símbolo es 6.

- 50 En la figura 16A, el factor "b" es 1 y por lo tanto la memoria 31 tiene una capacidad de almacenamiento de N/(6 x 1)

x (6 x 1) bits en las direcciones de columna y fila.

En lo que sigue, una región de almacenamiento de la memoria 31, que es un bit en la dirección de una fila y se extiende en la dirección de una columna, se menciona como columna según sea apropiado. En el ejemplo de la figura 16A, la memoria 31 incluye 6 (= 6 x 1) columnas.

- 5 El desmultiplexor 25 escribe bits de código del código de LDPC en la memoria 31 en la dirección de una columna desde la parte superior a la inferior de cada columna, empezando secuencialmente desde la columna más a la izquierda hasta la derecha.

Cuando los bits de código han escrito completamente de arriba abajo la columna más a la derecha, los bits de código son leídos desde la memoria 31 en la dirección de una fila, empezando 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 son proporcionados a la unidad 32 de reordenación.

- 10 La unidad 32 de reordenación reordena posiciones de 6 bits de código recibidos desde la memoria 31 y presenta 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 símbolo de 64 QAM.

15 Más específicamente, cuando los 6 bits de código leídos a partir de la memoria 31 en la dirección de una fila son indicados como b_0, b_1, b_2, b_3, b_4 y b_5 empezando secuencialmente desde el MSB, los bits de código que incluyen "y" que son adyacentes al bit " b_5 ", son bits de código susceptibles de error de acuerdo con la relación de peso de columna que se ha descrito anteriormente con referencia a la figura 11.

- 20 La unidad 32 de reordenación reordena posiciones de los 6 bits de código b_0 a b_5 recibidos desde la memoria 31 de tal modo que los bits de código susceptibles de error entre los 6 bits de código b_0 a b_5 procedentes de la memoria 31 son asignados a bits fuertes entre los 6 bits y_0 a y_5 que representan un símbolo de 64 QAM.

Varias compañías han sugerido una diversidad de métodos para reordenar los 6 bits de código b_0 a b_5 a partir de la memoria 31 y asignarlos a 6 bits y_0 a y_5 que representan un símbolo de 64 QAM, respectivamente.

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

- 25 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 (es decir, la posición del bit de código b_i se cambia a la del bit de símbolo y_j), de manera similar a las figuras 17A y 17B que se describen más adelante.

Mientras que el primer método de reordenación de la figura 16B sugiere el uso de uno de los tres tipos de reordenación, el segundo método de reordenación de la figura 16C sugiere utilizar uno de dos tipos de reordenación.

- 30 El tercer método de reordenación de la figura 16D sugiere la selección y el uso secuencial de 6 tipos de reordenación.

Las figuras 17A y 17B ilustran un ejemplo de configuración de un desmultiplexor 25 y un cuarto método de reordenación en el caso de que el método de modulación sea 16QAM (de modo que el número de bits de código "m" de un código de LDPC mapeado respecto a un símbolo sea 6 como en la figura 16) y el factor "b" sea 2.

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

La figura 17A ilustra el orden en el que los bits de código de un código de LDPC son escritos en la memoria 31.

El desmultiplexor 25 escribe bits de código en la memoria 31 en la dirección de una columna desde la parte superior hasta la inferior de cada columna, empezando secuencialmente desde la columna más a la izquierda hasta la derecha según se ha descrito anteriormente con referencia a la figura 16A.

Cuando los bits de código han sido escritos por completo de arriba abajo de la columna más a la derecha, los bits de código son leídos desde la memoria 31 en la dirección de una fila, secuencialmente a partir de la primera fila de todas las columnas de la memoria 31 en unidades de 12 bits (es decir, mb bits), y los bits de código leídos son proporcionados a la unidad 32 de reordenación.

- 45 La unidad 32 de reordenación reordena las posiciones de 12 bits de código recibidos desde la memoria 31 de acuerdo con el cuarto método de reordenación y presenta a la salida los 12 bits reordenados como 12 bits que representan dos símbolos (es decir, símbolos b) de 6aQAM, es decir, 6 bits y_0, y_1, y_2, y_3, y_4 e y_5 que representan un símbolo de 64QAM y 6 bits y_0, y_1, y_2, y_3, y_4 e y_5 que representan un símbolo de 64QAM y 6 bits y_0, y_1, y_2, y_3, y_4 e y_5 que representan el otro símbolo.

- 50 La figura 17B ilustra el cuarto método de reordenación realizado por la unidad 32 de reordenación de la figura 17A.

El óptimo de los métodos de reordenación, que minimiza tasas de error en las trayectorias de comunicación de AWGM, depende de la tasa de código de un código de LDPC o similar.

Ahora se va a describir cómo el intercalador de paridad 23 de la figura 8 realiza el intercalado de paridad con referencia a las figuras 18 a 20.

- 5 La figura 18 ilustra (parte de) un gráfico de Tanner de una matriz de comprobación de paridad de un código de LDPC.

Si ocurre un error, tal como una borradura, simultáneamente en dos o más nodos de variable conectados a (o dos o más bits de código correspondientes a) un nodo de comprobación, entonces el nodo de comprobación devuelve un mensaje con una probabilidad de "0" que es igual a la probabilidad de "1" a cada nodo de variable conectado al nodo de comprobación según se muestra en la figura 18. Por lo tanto, el rendimiento de descodificación se reduce si son borrados múltiples nodos de variable conectados al mismo nodo de comprobación.

Un código de LDPC definido en la especificación DVB-S.2, que es presentado a la salida por el codificador 21 de LDPC de la figura 8, es un código IRA y una matriz H_T de paridad de la matriz H de comprobación de paridad tiene una estructura gradual según se ha mostrado en la figura 11.

- 15 Las figuras 19A y 19B ilustran una matriz H_T de paridad que tiene una estructura gradual y un gráfico de Tanner que corresponde a la matriz H_T de paridad.

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

- 20 Cuando la matriz H_T de paridad tiene una estructuras gradual, los nodos de variable cuyos mensajes son obtenidos utilizando bits de código adyacentes (bits de paridad) de un código de LDPC, correspondientes a columnas que incluyen elementos que tienen un valor de "1" en la matriz H_T de paridad, son conectados al mismo nodo de comprobación en el gráfico de Tanner de la matriz H_T de paridad.

- 25 En consecuencia, si un error tal como un error de ráfaga o una borradura se produce simultáneamente en bits de paridad adyacentes, el rendimiento de descodificación se reduce puesto que un nodo de comprobación conectado a nodos de variable correspondientes respectivamente a bits de paridad erróneos (es decir, nodos de variable cuyos mensajes son obtenidos utilizando los bits de paridad) devuelve un mensaje con una probabilidad de "0" que es igual a la probabilidad de "1" a cada nodo de variable conectado al nodo de comprobación. El comportamiento de descodificación se reduce también cuando la longitud de ráfaga, que es el número de bits erróneos, es grande.

- 30 A continuación, el intercalador 23 de paridad de la figura 8 realiza intercalación de paridad sobre el código de LDPC procedente del codificador 21 de LDPC para intercalar bits de paridad del código de PDPC en diferentes posiciones de bits de paridad con el fin de impedir una reducción del rendimiento de descodificación.

La figura 20 ilustra una matriz H_T de paridad de una matriz H de comprobación de paridad correspondiente a un código de LDPC después de que el intercalador de la figura 8 realice intercalación de paridad sobre el código de LDPC.

- 35 Aquí, una matriz H_A de información de la matriz H de comprobación de paridad correspondiente al código de LDPC definida en la especificación DVB-S.2 a la salida del codificador 21 de LDPC, tiene una estructura cíclica.

- 40 El término "estructura cíclica" se refiere a una estructura en la que una columna, cuando se desplaza cíclicamente, se empareja con otra columna. Ejemplos de 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 han sido desplazadas cíclicamente en la dirección de una columna mediante un valor proporcional al valor "q" obtenido al dividir la longitud de paridad "M". En lo que sigue, el número de columnas "P" de la estructuras cíclica se menciona como un número de columnas unidad que tienen una estructura cíclica según sea apropiado.

- 45 Ejemplos del código de LDPC definido en la especificación DVBN-S.2 a la salida del codificador 21 de LDPC incluyen dos tipos de códigos de LDPC con longitudes de código respectivas N de 64800 bits y 16200 bits según se ha descrito anteriormente con referencia a la figura 12.

Ahora, se va a proporcionar la descripción que sigue centrándonos sobre el tipo de códigos de LDPC que tienen una longitud de código N de 64800 bits entre los dos tipos de códigos de LDPC con longitudes de código respectivas N de 64800 bits y 16200 bits. Se han definido 11 tasas de código nominal para el código de LDPC cuya longitud de código N es de 64800 bits según se ha descrito anteriormente con referencia a la figura 12.

- 50 Para cualquier código de LDPC que tenga una longitud de código N de 64800 bits de cada una de las 11 tasas de código nominales, el número de columnas unidad P que tienen una estructura cíclica se define como "360", el cual es uno de los divisores (excluyendo 1 y M) de la longitud de paridad M, en la especificación DV-S.2.

Para un código de LDPC que tenga una longitud de código N de 64800 bits de cada una de las 11 tasas de código

nominales, la longitud de paridad M se calcula de modo que sea un valor no primo de acuerdo con una ecuación $M = q \times P = q \times 360$, utilizando un valor “q” que varía dependiendo de la tasa de código. En consecuencia, de manera similar al número de columnas unidad 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 de columnas unidad P que tienen una estructura cíclica (es decir, 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 23 de paridad realiza intercalación de paridad sobre el código de LDPC recibido desde el codificador 21 de LDPC para intercalar el bit de código $(K + qx + y + 1)^{\text{ésimo}}$ entre bits de paridad, que son $(K + 1)^{\text{ésimo}}$ a $(K + M) (= N)^{\text{ésimo}}$ bits de código del código de LDPC, en una posición $(K + Py + x + 1)^{\text{ésima}}$ de bit de código.

De acuerdo con este método de intercalación de paridad, (bits de paridad correspondientes a) nodos de variable conectados al mismo nodo de comprobación están a una distancia correspondiente al número P de columnas unidad que tienen una estructura cíclica (360 bits en este ejemplo), impidiendo con ello la ocurrencia simultánea de error en una pluralidad de nodos de variable conectados al mismo nodo de comprobación. Esto puede mejorar la resistencia a errores de ráfaga.

El código de LDPC, que ha sido sometido a operación de intercalación de paridad de tal modo que el bit de código $(K + qx + y + 1)^{\text{ésimo}}$ se intercala en la posición $(K + Py + x + 1)^{\text{ésima}}$ de bit de código, es idéntico a un código de LDPC de una matriz de comprobación de paridad (mencionada en lo que sigue como matriz de comprobación de paridad convertida) que se obtiene realizando permutación de columna sobre la matriz H de comprobación de paridad original para sustituir (específicamente, cambiar) la columna $(K + Py + x + 1)^{\text{ésima}}$ de la matriz H de comprobación de paridad original por la columna $(k + qx + y + 1)^{\text{ésima}}$.

La matriz de paridad de la matriz de comprobación de paridad convertida tiene una estructura pseudo-cíclica cuyo número “P” de columnas unidad es (“360” en la figura 20) según se ha mostrado en la figura 20.

Aquí, el término “estructura pseudo-cíclica” se refiere a una estructura en la que una porción de la matriz de paridad, excluyendo una parte específica de la matriz de paridad, tiene estructura cíclica. Una matriz de comprobación de paridad convertida obtenida realizando permutación de columna correspondiente a intercalación de paridad sobre una matriz de comprobación de paridad de un código de LDPC definido en la especificación DVB-S.2, tiene una porción de esquina derecha de 360×360 (correspondiente a una matriz desplazada que se describe más adelante) que es solamente un elemento de “1” corto de la estructura cíclica (es decir, la porción de esquina derecha de 360×360 tiene un elemento de “0” en vez de “1” que se requiere en la estructura cíclica). Puesto que la matriz de comprobación de paridad convertida no tiene una estructura cíclica (completa), se define como que tiene una “estructura pseudo-cíclica”.

Realmente, la matriz de comprobación de paridad convertida de la figura 20 se obtiene realizando permutación de fila, adicionalmente a la permutación de columna correspondiente a intercalación de paridad, sobre la matriz H de comprobación de paridad original para permitir que la matriz de comprobación de paridad convertida incluya matrices componentes que se describen más adelante.

Ahora se va a describir cómo el intercalador 24 por giro de columna de la figura 8 realiza la intercalación por giro de columna a modo de un proceso de permutación, con referencia a las figuras 21 a 24.

El transmisor 11 de la figura 8 transmite dos o más bits de código del código de LDPC como un símbolo según se ha descrito anteriormente, con el fin de mejorar la eficiencia de uso de frecuencias. Por ejemplo, se utiliza QPSK como método de modulación cuando dos bits de código son transmitidos, y se utiliza 16QAM como método de modulación cuando cuatro bits de código son transmitidos como un símbolo.

Si un error, tal como una borraradura, ocurre en un símbolo en el caso de que dos o más bits de código sean transmitidos como símbolo según se ha descrito anteriormente, todos los bits de código del símbolo resultan ser erróneos (es decir, son borrados).

En consecuencia, para mejorar el rendimiento de descodificación, con el fin de reducir la probabilidad de que (bits de código correspondientes a) nodos de variable conectados al mismo nodo de comprobación sean borrados simultáneamente, es necesario evitar que nodos de variable correspondientes a bits de código de un símbolo sean conectados al mismo nodo de comprobación.

Por otra parte, en el caso de que la matriz H de comprobación de paridad del código de LDPC definido en la especificación de DVB-S.2 esté presente a la salida del codificador 21 de LDPC, la matriz H_A de información de la matriz H de comprobación de paridad tiene una estructura cíclica y la matriz H_T de paridad tiene una estructura gradual según se ha descrito con anterioridad. En el caso de que la matriz de comprobación de paridad convertida sea la matriz de comprobación de paridad del código de LDPC que ha sido sometida a intercalación de paridad, la matriz de paridad tiene también una estructura cíclica (específicamente, una estructura pseudo-cíclica) según se ha descrito anteriormente con referencia a la figura 20.

Las figuras 21A y 21B ilustran una matriz de comprobación de paridad convertida.

Específicamente, la figura 21A ilustra una matriz de comprobación de paridad convertida de una matriz H de comprobación de paridad de un código de LDPC que tiene una longitud de código N de 64800 bits y una tasa de código (r) de 3/4.

- 5 En la figura 21A, la posición de cada elemento que tiene un valor de "1" en la matriz de comprobación de paridad convertida ha sido mostrado como un punto "•".

La figura 21B ilustra una operación que el desmultiplexor 25 de la figura 8 realiza sobre un código de LDPC de la matriz de comprobación de paridad convertida de la figura 21A, es decir, un código de LDPC que ha sido sometido a intercalación de paridad.

- 10 En la figura 21B, que utiliza 16QAM como método de modulación, bits de código del código de LDPC de paridad intercalada son escritos en la dirección de una columna a cuatro columnas que constituyen la memoria 31 del desmultiplexor 25.

Los bits de código escritos en la dirección de una columna hasta las cuatro columnas de la memoria 31, son leídos en la dirección de una fila en unidades de 4 bits a modo de un símbolo.

- 15 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 un "1" en una fila arbitraria de la matriz de comprobación de paridad convertida de la figura 21A. En este caso, nodos de variable correspondientes a los cuatro bits de código B_0 , B_1 , B_2 y B_3 están conectados al mismo nodo de comprobación.

- 20 En consecuencia, si ocurre una borradura en un símbolo en el caso en que los cuatro bits de código B_0 , B_1 , B_2 y B_3 del símbolo incluyen bits de código correspondientes a "1" en una fila arbitraria de la matriz de comprobación de paridad convertida, es difícil obtener un mensaje apropiado para el mismo nodo de comprobación conectado a nodos de variable correspondientes respectivamente a los bits de código B_0 , B_1 , B_2 y B_3 , reduciendo con ello el rendimiento de descodificación.

- 25 Cuando se emplea una tasa de código distinta de 3/4, una pluralidad de bits de código correspondientes a una pluralidad de nodos de variable conectados al mismo nodo de comprobación pueden constituir un símbolo de 16QAM.

Por lo tanto, el intercalador 24 por giro de columna realiza intercalación por giro de columna sobre el código de LDPC de paridad intercalada a partir del intercalador 23 de paridad para intercalar los bits de código del código de LDPC de paridad intercalada de modo que una pluralidad de bits de código correspondientes a "1" de una fila arbitraria de la matriz de comprobación de paridad convertida no son mapeados en un símbolo.

La figura 22 ilustra cómo se lleva a cabo la intercalación por giro de columna.

Específicamente, la figura 22 ilustra la memoria 31 del desmultiplexor 25 mostrado en las figuras 16 y 17.

- 35 La memoria 31 tiene una capacidad de almacenamiento para almacenar mb bits en la dirección de una fila (horizontal) y almacenar N/m_b bits en la dirección de una columna (vertical), e incluye m_b columnas según se ha descrito anteriormente con referencia a la figura 16. El intercalador 24 por giro de columna realiza intercalación por giro de columna controlando una posición de inicio de escritura en cada columna de la memoria 31, en la que se inicia la escritura en la columna, cuando se escriben bits de código de un código de LDPC en la memoria 31 en la dirección de una columna y son leídos desde la memoria 31 en la dirección de una fila.

- 40 Más específicamente, el intercalador 24 por giro de columna cambia apropiadamente la posición de inicio de escritura de modo que una pluralidad de bits de código leídos en la dirección de una fila para constituir un símbolo no incluyan una pluralidad de bits de código correspondientes a "1" en una fila arbitraria de la matriz de comprobación de paridad convertida. Es decir, el intercalador 24 por giro de columna permuta bits de código del código de LDPC de modo que una pluralidad de bits de código correspondientes a "1" de una fila arbitraria de la matriz de comprobación de paridad no sean incorporados en el mismo símbolo.

- 45 La figura 22 ilustra un ejemplo de configuración de la memoria 31 en caso de que se emplee 16QAM como método de modulación y el factor "b" descrito con referencia a la figura 16 sea "1". En consecuencia, el número de bits de código "m" del código de LDPC mapeado en un símbolo es 4 y la memoria 31 incluye 4 (= mb) columnas.

- 50 El intercalador 24 por giro de columna de la figura 22 (en vez del desmultiplexor 25 de la figura 176) escribe bits de código en la memoria 31 en la dirección de una columna desde la parte superior hasta la inferior de cada una de las cuatro columnas de la memoria 31, empezando secuencialmente desde la columna más a la izquierda hasta la derecha.

Cuando se han escrito completamente bits de código de arriba abajo hasta la columna más a la derecha, el intercalador 24 por giro de columna lee bits de código en unidades de 4 bits (mb bits) en la dirección de una fila,

empezando desde la primera fila de todas las columnas de la memoria 31, y presenta a la salida los bits de código leídos a modo de código de LDPC por giro de columna intercalado a la unidad 32 de reordenación del desmultiplexor 25 mostrado en las figuras 16 y 17.

- 5 Cuando la dirección de la primera (superior) posición de cada columna está representada por "0" y la dirección de cada posición a lo largo de una dirección de columna está representada por un número entero que se incrementa secuencialmente, el intercalador 24 por giro de columna de la figura 22 determina que la dirección de una posición de inicio 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 una posición de inicio de escritura en la tercera columna es "4", y la dirección de una posición de inicio de escritura en la cuarta columna es "7".
- 10 Después de que los bits de código han sido escritos en una columna que tiene una posición de inicio de escritura en una dirección distinta de "0" desde la posición superior hasta la inferior de la columna, el intercalador 24 por giro de columna retorna a la primera posición de la columna en una dirección de "0" y continúa escribiendo bits de código en la columna hasta una posición inmediatamente anterior a la posición de inicio de escritura. El intercalador 24 por giro de columna realiza entonces la escritura en la siguiente columna de la derecha.
- 15 Al realizar la intercalación por giro de columna según se ha descrito anteriormente, se puede evitar que una pluralidad de bits de código correspondientes a una pluralidad de nodos de variable conectados al mismo nodo de comprobación sean asignados a un símbolo de 16 QAM (es decir, que sean incorporados en el mismo símbolo) para un código de LDPC de cada tasa de código que tenga una longitud de código N de 647800 según está definido en la especificación DVB-S.2. Esto puede mejorar el rendimiento de descodificación en una trayectoria de comunicación
- 20 en cuanto a que ocurra una borradura.

La figura 23 ilustra el número de columnas de la memoria 31 requeridas para intercalación por giro de columna y las direcciones de las posiciones de inicio de escritura en asociación con cada método de modulación para un código de LDPC de cada una de las 11 relaciones de código que tienen una longitud de código N de 64800 según se define en la especificación DVB-S.2.

- 25 El número de bits "m" de un símbolo es 2 y el factor "b" es 1 cuando se emplea uno de los métodos primero a tercero en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea QPSK como método de modulación.

En este caso, la memoria 31 tiene 2 columnas para almacenar 2×1 (= mb) bits en la dirección de una fila, y almacena $64800/(2 \times 1)$ en la dirección de una columna según se muestra en la figura 23. La posición de inicio de escritura de la 1^a y 2^a columnas de la memoria 31 es una dirección de "0" y la posición de inicio de escritura de la 2^a columna es una dirección de "2".

Adicionalmente, el número de bits "m" de un símbolo es 2 y el factor "b" es 2 cuando se emplea el cuarto método de reordenación de la figura 17 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea QPSK como método de modulación.

- 35 En este caso, la memoria 31 tiene 4 columnas para almacenar 2×2 bits en la dirección de una fila, y almacena $64800/(2 \times 2)$ bits en la dirección de una columna según se muestra en la figura 23. La posición de inicio de escritura de la 1^a y 4^a columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2^a columna está en una dirección de "2", la posición de inicio de escritura de la 3^a columna está en una dirección de "4", y la posición de inicio de escritura de la 4^a columna está en una dirección de "7".

- 40 Adicionalmente, el número de bits "m" de un símbolo es 4 y el factor "b" es 1 cuando se emplea uno de los métodos de reordenación primero a tercero de la figura 16 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 16QAM como método de modulación.

En este caso, la memoria 31 tiene 4 columnas para almacenar 4×1 bits en la dirección de una fila, y almacena $64800/(4 \times 1)$ bits en la dirección de una columna según se muestra en la figura 23. La posición de inicio de escritura de la 1^a de las 4 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2^a columna está en una dirección de "2", la posición de inicio de escritura de la 3^a columna está en una dirección de "4", y la posición de inicio de escritura de la 4^a columna está en una dirección de "7".

- 45 Adicionalmente, el número de bits "m" de un símbolo es 4 y el factor "b" es 2 cuando se emplea el cuarto método de reordenación de la figura 17 durante el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 16QAM como método de modulación.

En este caso, la memoria 31 tiene 8 columnas para almacenar 4×2 bits en la dirección de una fila, y almacena $64800/(4 \times 2)$ bits en la dirección de una columna según se muestra en la figura 23. La posición de inicio de escritura de la 1^a de las 8 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2^a columna está en una dirección de "0", la posición de inicio de escritura de la 3^a columna está en una dirección de "2", la posición de inicio de escritura de la 4^a columna está en una dirección de "4", la posición de inicio de escritura de la 5^a columna están en una dirección de "4", la posición de inicio de escritura de la 6^a columna está en una dirección de

"5", la dirección de inicio de escritura de la 7^a columna está en una dirección de "7", y la posición de inicio de escritura de la 8^a columna está en una dirección de "7".

Adicionalmente, el número de bits "m" de un símbolo es 6 y el factor "b" es 1 cuando se emplea uno de los métodos de reordenación primero a tercero de la figura 16 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 64QAM como método de modulación.

En este caso, la memoria 31 tiene 6 columnas para almacenar 6 x 1 bits en la dirección de una fila, y almacena $64800/(6 \times 1)$ bits en la dirección de columna según se muestra en la figura 23. La posición de inicio de escritura de la 1^a de las 6 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2^a columna está en una dirección de "2", la posición de inicio de escritura de la 3^a columna está en una dirección de "5", la posición de inicio de escritura de la 4^a columna está en una dirección de "9", la posición de inicio de escritura de las 5^a columna está en una dirección de "10", y la posición de inicio de escritura de la 6^a columna está en una dirección de "13".

Adicionalmente, el número de bits "m" de un símbolo es 6 y el factor "b" es 2 cuando se emplea el cuarto método de reordenación de la figura 17 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 64QAM como método de modulación.

En este caso, la memoria 31 tiene 12 columnas para almacenar 6 x 2 bits en la dirección de una fila, y almacena $64800/(6 \times 2)$ bits en la dirección de columna según se muestra en la figura 23. La posición de inicio de escritura de la 1^a de las 12 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2^a columna está en una dirección de "0", la posición de inicio de escritura de la 3^a columna está en una dirección de "2", la posición de inicio de escritura de la 4^a columna está en una dirección de "2", la posición de inicio de escritura de la 5^a columna está en una dirección de "3", la posición de inicio de escritura de la 6^a columna está en una dirección de "4", la posición de inicio de escritura de la 7^a columna está en una dirección de "4", la posición de inicio de escritura de la 8^a columna está en una dirección de "5", la posición de inicio de escritura de la 9^a columna está en una dirección de "5", la posición de inicio de escritura de la 10^a columna está en una dirección de "7", la posición de inicio de escritura de la 11^a columna está en una dirección de "8", y la posición de inicio de escritura de la 12^a columna está en una dirección de "9".

Adicionalmente, el número de bits "m" de un símbolo es 8 y el factor "b" es 1 cuando se emplea uno de los métodos de reordenación primero a tercero en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 256QAM como método de modulación.

En este caso, la memoria 31 tiene 8 columnas para almacenar 8 x 1 bits en la dirección de una fila, y almacena $64800/(8 \times 1)$ bits en la dirección de columna según se muestra en la figura 23. La posición de inicio de escritura de la 1^a de las 8 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2^a columna está en una dirección de "0", la posición de inicio de escritura de la 3^a columna está en una dirección de "2", la posición de inicio de escritura de la 4^a columna está en una dirección de "4", la posición de inicio de escritura de la 5^a columna está en una dirección de "4", la posición de inicio de escritura de la 6^a columna está en una dirección de "5", la posición de inicio de escritura de la 7^a columna está en una dirección de "7", y la posición de inicio de escritura de la 8^a columna está en una dirección de "7".

Adicionalmente, el número de bits "m" de un símbolo es 8 y el factor "b" es 2 cuando se emplea el cuarto método de reordenación de la figura 17 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 256QAM como método de modulación.

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

Adicionalmente, el número de bits "m" de un símbolo es 10 y el factor de "b" es 1 cuando se emplea uno de los métodos de reordenación primero a tercero de la figura 16 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 1024QAM como método de modulación.

En este caso, la memoria 31 tiene 10 columnas para almacenar 10 x 1 bits en una dirección de fila, y almacena

22^a columna está en una dirección de “41”, la posición de inicio de escritura de la 23^a columna está en una dirección de “41”, y la posición de inicio de escritura de la 24^a columna está en una dirección de “41”.

La figura 24 ilustra el número de columnas de la memoria 31 requeridas para intercalación por giro de columna y las direcciones de las posiciones de inicio de escritura en asociación con cada método de modulación para un código de

- 5 LDPC de cada una de las 10 tasas de código 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 se emplea uno de los métodos de reordenación primero a tercero de la figura 16 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea QPSK como método de modulación.

- 10 En este caso, la memoria 31 tiene 2 columnas para almacenar 2 x 1 bits en una dirección de fila, y almacena 16200/(2 x 1) bits en una dirección de columna según se muestra en la figura 24. La posición de inicio de escritura de la 1^a de las 2 columnas de la memoria 31 está en una dirección de “0”, y la posición de inicio de escritura de la 2^a columna está en una dirección de “0”.

- 15 Adicionalmente, el número de bits “m” de un símbolo es 2 y el factor “b” es 2 cuando se emplea el cuarto método de reordenación de la figura 17 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea QPSK como método de modulación.

- 20 En este caso, la memoria 31 tiene 4 columnas para almacenar 2 x 2 bits en una dirección de fila, y almacena 16200/(2 x 2) bits en una dirección de columna según se muestra en la figura 24. La posición de inicio de escritura de la 1^a de las 4 columnas de la memoria 31 está en una dirección de “0”, la posición de inicio de escritura de la 2^a columna está en una dirección de “2”, la posición de inicio de escritura de la 3^a columna está en una dirección de “3”, y la posición de inicio de escritura de la 4^a columna está en una dirección de “3”.

Adicionalmente, el número de bits “m” de un símbolo es 4 y el factor “b” es 1 cuando se emplea uno de los métodos de reordenación primero a tercero de la figura 16 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 16QAM como método de modulación.

- 25 En este caso, la memoria 31 tiene 4 columnas para almacenar 4 x 1 bits en una dirección de fila, y almacena 16200/(4 x 1) bits en una dirección de columna según se muestra en la figura 24. La posición de inicio de escritura de la 1^a de las 4 columnas de la memoria 31 está en una dirección de “0”, la posición de inicio de escritura de la 2^a columna está en una dirección de “2”, la posición de inicio de escritura de la 3^a columna está en una dirección de “3”, y la posición de inicio de escritura de la 4^a columna está en una dirección de “3”.

- 30 Adicionalmente, el número de bits “m” de un símbolo es 4 y el factor “b” es 2 cuando se emplea el cuarto método de reordenación de la figura 17 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 16QAM como método de modulación.

En este caso, la memoria 31 tiene 8 columnas para almacenar 4 x 2 bits en una dirección de fila, y almacena 16200/(4 x 2) bits en una dirección de columna según se muestra en la figura 24. La posición de inicio de escritura

- 35 de la 1^a de las 8 columnas de la memoria 31 está en una dirección de “0”, la posición de inicio de escritura de la 2^a columna está en una dirección de “0”, la posición de inicio de escritura de la 3^a columna está en una dirección de “0”, la posición de inicio de escritura de la 4^a columna está en una dirección de “1”, la posición de inicio de escritura de la 5^a columna está en una dirección de “7”, la posición de inicio de escritura de la 6^a columna está en una dirección de “20”, la posición de inicio de escritura de la 7^a columna está en una dirección de “20”, y la posición de inicio de escritura de la 8^a columna está en una dirección de “21”.

Adicionalmente, el número de bits “m” de un símbolo es 6 y el factor “b” es 1 cuando se emplea uno de los métodos de reordenación primero a tercero de la figura 16 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 64QAM como método de modulación.

- 45 En este caso, la memoria 31 tiene 6 columnas para almacenar 6 x 1 bits en una dirección de fila, y almacena 16200/(6 x 1) bits en una dirección de columna según se muestra en la figura 24. La posición de inicio de escritura de la 1^a de las 6 columnas de la memoria 31 está en una dirección de “0”, la posición de inicio de escritura de la 2^a columna está en una dirección de “0”, la posición de inicio de escritura de la 3^a columna está en una dirección de “2”, la posición de inicio de escritura de la 4^a columna está en una dirección de “3”, la posición de inicio de escritura de la 5^a columna está en una dirección de “7”, y la posición de inicio de escritura de la 6^a columna está en una dirección de “7”.

50 Adicionalmente, el número de bits “m” de un símbolo es 6 y el factor “b” es 2 cuando se emplea el cuarto método de reordenación de la figura 17 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 64QAM como método de modulación.

- 55 En este caso, la memoria 31 tiene 12 columnas para almacenar 6 x 2 bits en una dirección de fila, y almacena 16200/(6 x 2) bits en una dirección de columna según se muestra en la figura 24. La posición de inicio de escritura

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

Adicionalmente, el número de bits "m" de un símbolo es 12 y el factor "b" es 2 cuando se emplea el cuarto método de reordenación de la figura 17 en el proceso de reordenación del desmultiplexor 25 mostrado en la figura 8 y se emplea 4096QAM como método de modulación.

- 10 En este caso, la memoria 31 tiene 24 columnas para almacenar 12×2 bits en una dirección de fila, y almacena $16200/(12 \times 2)$ bits en una dirección de columna según se muestra en la figura 24. La posición de inicio de escritura de la 1^a de las 24 columnas de la memoria 31 está en una dirección de "0", la posición de inicio de escritura de la 2^a columna está en una dirección de "0", la posición de inicio de escritura de la 3^a columna está en una dirección de "0", la posición de inicio de escritura de la 4^a columna está en una dirección de "0", la posición de inicio de escritura de la 5^a columna está en una dirección de "0", la posición de inicio de escritura de la 6^a columna está en una dirección de "0", la posición de inicio de escritura de la 7^a columna está en una dirección de "0", la posición de inicio de escritura de la 8^a columna está en una dirección de "1", la posición de inicio de escritura de la 9^a columna está en una dirección de "1", la posición de inicio de escritura de la 10^a columna está en una dirección de "1", la posición de inicio de escritura de la 11^a columna está en una dirección de "2", la posición de inicio de escritura de la 12^a columna está en una dirección de "2", la posición de inicio de escritura de la 13^a columna está en una dirección de "2", la posición de inicio de escritura de la 14^a columna está en una dirección de "3", la posición de inicio de escritura de la 15^a columna está en una dirección de "7", la posición de inicio de escritura de la 16^a columna está en una dirección de "9", la posición de inicio de escritura de la 17^a columna está en una dirección de "9", la posición de inicio de escritura de la 18^a columna está en una dirección de "9", la posición de inicio de escritura de la 19^a columna está en una dirección de "10", la posición de inicio de escritura de la 20^a columna está en una dirección de "10", la posición de inicio de escritura de la 22^a columna está en una dirección de "10", la posición de inicio de escritura de la 23^a columna está en una dirección de "10", y la posición de inicio de escritura de la 24^a columna está en una dirección de "11".

30 Un procedimiento de transmisión llevado a cabo por el transmisor 11 de la figura 8 va a ser descrito en lo que sigue con referencia al diagrama de flujo de la figura 25.

El codificador 21 de LDPC espera hasta que se reciben datos objetivo y codifica los datos objetivo recibidos según un código de LDPC y proporciona el código de LDPC al intercalador 22 de bit en la etapa S101, y el procedimiento avanza hasta la etapa S102.

35 En la etapa S102, el intercalador 22 de bit realiza intercalación de bit sobre el código de LDPC procedente del codificador 21 de LDPC y proporciona el código de LDPC de bit intercalado a la unidad 26 de mapeo, y el procedimiento avanza entonces hasta la etapa S103.

Más específicamente, en la etapa S102, el intercalador 23 de paridad del intercalador 22 de bit realiza intercalación de paridad sobre el código de LDPC procedente del codificador 21 de LDPC y proporciona el código de LDPC de paridad intercalada al intercalador 24 por giro de columna.

40 El intercalador 24 por giro de columna lleva a cabo intercalación por giro de columna sobre el código de LDPC procedente del intercalador 23 de paridad, y el desmultiplexor 23 lleva a cabo un proceso sobre el código de LDPC de columna de giro intercalada por medio del intercalador 24 por giro de columna. El desmultiplexor 25 proporciona a continuación el código de LDPC reordenado a la unidad 26 de mapeo.

45 En la etapa S103, la unidad 26 de mapeo lleva a cabo el mapeo de m bits de código del código de LDPC procedente del desmultiplexor 25 en un símbolo representado por un punto de señal determinado de acuerdo con un esquema de modulación que utiliza el modulador 27 ortogonal para llevar a cabo la modulación ortogonal, y proporciona el símbolo mapeado al modulador 27 ortogonal, y el procedimiento avanza entonces hasta la etapa S104.

50 En la etapa S104, el modulador 27 ortogonal realiza modulación ortogonal de portadoras sobre el símbolo procedente de la unidad 26 de mapeo, y el procedimiento avanza a continuación hasta la etapa S105 y transmite la señal ortogonalmente modulada y el procedimiento termina a continuación.

El procedimiento de transmisión de la figura 25 se repite.

Realizar la intercalación de paridad o la intercalación por giro de columna según se ha descrito en lo que antecede, puede incrementar la resistencia a las borraduras o a los errores de ráfaga cuando una pluralidad de bits de código de un código de LDPC son transmitidos como un símbolo.

55 El intercalador 23 de paridad, que es un bloque para realizar intercalación de paridad, y el intercalador 24 por giro de columna, que es un bloque para realizar intercalación por giro de columna, pueden ser construidos integralmente

aunque el intercalador 23 de paridad y el intercalador 24 por giro de columna hayan sido mostrados como construidos por separado en la figura 8 para facilidad de la explicación.

Más específicamente, tanto el intercalador de paridad como el intercalador por giro de columna pueden escribir y leer bits de código en, y desde, la memoria y pueden ser representados por medio de una matriz que convierte una dirección (dirección de escritura) en la que un bit de código es escrito, en una dirección (dirección de lectura) en la que un bit de código es leído.

En consecuencia, es posible obtener un código de LDPC que haya sido intercalado en paridad y que haya sido a continuación intercalado en columna de giro convirtiendo bits de código con la utilización de una matriz obtenida por multiplicación de una matriz que representa intercalación de paridad y una matriz que representa intercalación por giro de columna.

El desmultiplexor 25 puede ser construido también integralmente con el intercalador 23 de paridad y con el intercalador 24 por giro de columna.

Más específicamente, un proceso de reordenación llevado a cabo por el desmultiplicador 25 puede ser representado también por una matriz que convierte una dirección de escritura de la memoria 31 que almacena un código de LDPC, en una dirección de lectura.

Por consiguiente, es posible realizar conjuntamente intercalación de paridad, intercalación por giro de columna, y un proceso de reordenación utilizando una matriz obtenida por multiplicación de una matriz que representa intercalación de paridad, una matriz que representa intercalación por giro de columna, y una matriz que representa un proceso de reordenación.

20 También es posible realizar solamente intercalación de paridad o bien intercalación por giro de columna.

Ahora se van a describir las simulaciones para la medición de tasas de error de bit que se realizaron con el transmisor 11 de la figura 8, con referencia a las figuras 26 a 28.

Las simulaciones fueron realizadas utilizando una trayectoria de comunicación con una titilación de D/U de 0 dB.

Las figuras 26A y 26B ilustran un modelo de trayectoria de comunicación empleada en las simulaciones.

25 Específicamente, la figura 26A muestra un modelo de titilador empleado en las simulaciones.

La figura 26B ilustra un modelo de trayectoria de comunicación que tiene el titilador cuyo modelo ha sido mostrado en la figura 26A.

“H” indica, en la figura 26B, el modelo de titilador de la figura 26A. “N” indica, en la figura 26B, Interferencia Inter-Portadora (ICI). En las simulaciones, una previsión $E[N^2]$ de potencia de ICI fue aproximada por AWGN.

30 Las figuras 27 y 28 ilustran relaciones entre frecuencias Doppler f_d de los titiladores y tasas de error de las simulaciones.

Más específicamente, la figura 27 muestra relaciones entre tasas de error y frecuencias Doppler f_d cuando el esquema de modulación es 16QAM, la tasa de código (r) es 3/4, y el método de reordenación es el primer método de reordenación. La figura 28 muestra relaciones entre tasas de error y frecuencias Doppler f_d cuando el esquema de modulación es 64AQAM, la tasa de código (r) es 5/6, y el método de reordenación es el primer método de reordenación.

35 En las figuras 27 y 28, una línea marcada de manera más intensa indica una relación entre tasas de error y frecuencias Doppler f_d cuando la intercalación de paridad, la intercalación por giro de columna y el proceso de reordenación fueron todos llevados a cabo, y una línea delgada indica una relación entre tasas de error y frecuencias Doppler f_d cuando solamente se llevó a cabo el proceso de reordenación de entre los tres procesos.

40 Se puede apreciar a partir de cualquiera de las figuras 27 y 28 que se mejoraron las tasas de error (es decir, se rebajaron) cuando la intercalación de paridad, la intercalación por giro de columna y el proceso de reordenación fueron todos llevados a cabo, en comparación a cuando solamente se llevó a cabo el proceso de reordenación.

Receptor

45 La figura 29 proporciona la ilustración de un ejemplo de receptor que puede ser usado para detectar símbolos de OFMD y recuperar bits de datos desde señales de sub-portadora de los símbolos de OFMD. Según se muestra en la figura 29, una señal de OFMD es recibida por una antena 500 y detectada por un sintonizador 502 y convertida en una forma digital por medio de un convertidor 504 analógico-digital. Un procesador 506 de eliminación de intervalo de seguridad retira el intervalo de seguridad del símbolo de OFDM recibido, con anterioridad a que los datos sean recuperados desde el símbolo de OFDM utilizando un procesador 508 de Transformada Rápida de Fourier (FFT) en combinación con un estimador de canal y corrección 510 en cooperación con una unidad 511 de descodificación de

señalización incorporada, de acuerdo con técnicas conocidas. Los símbolos de datos desmodulados son recuperados a partir de un desmapeador 512 y alimentados a un desintercalador 514 de símbolo, el cual actúa para efectuar el mapeo inverso de los símbolos de datos recibidos para regenerar una corriente de símbolos de salida con símbolos de datos desintercalados. El desintercalador 514 de símbolo va a ser descrito con mayor detalle.

5 Intercalador de bit y descodificador de LDPC

Según se muestra en la figura 29, el receptor incluye también una unidad 52 de desmapeo, un desintercalador 53, y un descodificador 56 de LDPC. La unidad 52 de desmapeo recibe símbolos (con valores respectivos de las direcciones de los ejes I y Q) desde el desintercalador 514 de símbolo y actúa de modo que desmapea los símbolos en bits codificados de un código de LDPC y proporciona los bits codificados del código de LDPC al desintercalador 53 de bit. El desmapeo de los símbolos de datos recibidos se efectúa identificando los bits que están representados por el símbolo de datos identificado a partir de la señal de sub-portadora del símbolo de OFMD.

El desintercalador 53 de bit incluye un desmultiplexor 54 y un desintercalador 55 por giro de columna y realiza desintercalación sobre los bits de código del código de LDPC a partir de la unidad 52 de desmapeo.

Más específicamente, el desmultiplexor 54 realiza un proceso de reordenación inversa, que es la inversa del procedimiento de reordenación realizado por el desmultiplexor 25 de la figura 8, sobre el código de LDPC procedente de la unidad 52 de desmapeo. Específicamente, el desmultiplexor 54 realiza un proceso de reordenación inversa para restaurar las posiciones de los bits de código reordenados por medio del proceso de reordenación en posiciones originales y proporciona el código de LDPC inversamente reordenado al desintercalador 55 por giro de columna.

El desintercalador 55 por giro de columna realiza un proceso de desintercalación inversa por giro de columna, el cual es la inversa de la intercalación por giro de columna como proceso de permutación realizado por el intercalador 24 por giro de columna de la figura 8, sobre el código de LDPC procedente del desmultiplexor 54. Específicamente, el desintercalador 55 por giro de columna realiza un proceso de permutación inversa (por ejemplo, desintercalación por giro de columna) para restablecer el orden original de los bits de código del código de LDPC reordenado por intercalación por giro de columna como proceso de permutación de los bits de código.

Más específicamente, el desintercalador 55 por giro de columna realiza desintercalación por giro de columna escribiendo y leyendo los bits de código del código de LDPC en, y desde, una memoria para desintercalación que está construida de una manera similar a la memoria 31 mostrada en la figura 22.

Sin embargo, el desintercalador 55 por giro de columna escribe un bit de código en una dirección de fila en la memoria para desintercalación utilizando una dirección de lectura en la que el código de bit fue leído desde la memoria 31, como dirección de escritura. Adicionalmente, el desintercalador 55 por giro de columna lee un bit de código en una dirección de columna desde la memoria para intercalación utilizando una dirección de escritura, en la que el bit de código fue escrito en la memoria 31, como dirección de lectura.

El desintercalador 55 por giro de columna proporciona el código de LDPC desintercalado por giro de columna, al descodificador 56 de LDPC.

Aunque la intercalación de paridad, la intercalación por giro de columna, y un proceso de reordenación fueran realizados secuencialmente sobre el código de LDPC proporcionado desde la unidad 52 de desmapeo al desintercalador 53, el desintercalador 53 realiza solamente los dos procesos, es decir, un proceso de reordenación inversa correspondiente al proceso de reordenación y a la desintercalación por giro de columna que corresponden a intercalación por giro de columna, sobre el código de LDPC. De ese modo, el desintercalador 53 no realiza desintercalación de paridad correspondiente a intercalación de paridad (es decir, la inversa de intercalación de paridad). Es decir, el desintercalador 53 no realiza desintercalación de paridad para restablecer el orden original de los bits de código del código de LDPC reordenado por intercalación de paridad.

En consecuencia, el código de LDPC, sobre el que se ha realizado el proceso de reordenación inversa y la desintercalación por giro de columna y no se ha realizado ninguna desintercalación de paridad, es suministrado desde (el desintercalador 55 por giro de columna de) el desintercalador 53 hasta el descodificador 56 de LDPC.

El descodificador 56 de LDPC realiza descodificación de LDPC sobre el código de LDPC procedente del desintercalador 53 utilizando una matriz de comprobación de paridad convertida, obtenida con la realización de al menos permutación de columna correspondiente a intercalación de paridad sobre la matriz H de comprobación de paridad que el codificador 21 de LDPC de la figura 8 utilizó para codificación de LDPC y a continuación presenta a la salida los datos resultantes como datos objetivo descodificados.

La figura 30 es un diagrama de flujo que ilustra un procedimiento de recepción llevado a cabo por el receptor 12 de la figura 29.

El demodulador ortogonal 51 recibe una señal modulada desde el transmisor 11 en la etapa S111. El procedimiento avanza a continuación hasta la etapa S112 para realizar demodulación ortogonal sobre la señal modulada. El

demodulador 51 ortogonal proporciona entonces un símbolo obtenido mediante demodulación ortogonal a la unidad 52 de desmapeo y el procedimiento avanza a continuación desde la etapa 112 hasta la etapa S113.

En la etapa S113, la unidad 52 de desmapeo realiza el desmapeo de los símbolos procedentes del demodulador 51 ortogonal en bits de código de un código de LDPC y proporciona los bits de código de un código de LDPC al desintercalador 53. El procedimiento avanza a continuación hasta la etapa S114.

En la etapa S114, el desintercalador 53 realiza desintercalación sobre los bits de código del código de LDPC procedente de la unidad 52 de desmapeo y el procedimiento avanza a continuación hasta la etapa S115.

Más específicamente, en la etapa S114, el desmultiplicador 54 del desintercalador 53 realiza un proceso de reordenación inversa sobre el código de LDPC procedente de la unidad 52 de desmapeo y proporciona el código de LDPC resultante al desintercalador 55 por giro de columna.

El desintercalador 55 por giro de columna realiza desintercalación por giro de columna sobre el código de LDPC procedente del desmultiplexor 54 y proporciona el código de LDPC resultante al descodificador 56 de LDPC.

En la etapa S115, el descodificador 56 de LDPC realiza descodificación de LDPC sobre el código de LDPC procedente del desintercalador 55 por giro de columna utilizando una matriz de comprobación de paridad convertida, obtenida al realizar al menos permutación de columna correspondiente a intercalación de paridad sobre la matriz H de comprobación de paridad que usó el codificador 21 de LDPC de la figura 8 para la codificación de LDPC, y a continuación proporciona los datos resultantes como datos objetivo descodificados. El procedimiento termina a continuación.

El procedimiento de recepción de la figura 30 se repite.

20 El desmultiplexor 54, que realiza un proceso de reordenación inversa, y el desintercalador 55 por giro de columna, que realiza desintercalación por giro de columna, pueden haber sido construidos integralmente, aunque el desmultiplexor 54 y el desintercalador 55 por giro de columna hayan sido mostrados como construidos por separado en la figura 29 de la misma manera que en la figura 8 para facilidad de la explicación.

25 En caso de que el transmisor 11 de la figura 8 no realice intercalación por giro de columna, no hay necesidad de proporcionar el desintercalador 55 por giro de columna en el receptor 12 de la figura 29.

Ahora se hará referencia a cómo el descodificador 56 de LDPC de la figura 29 realiza la descodificación de LDPC.

El descodificador 56 de LDPC de la figura 29 realiza descodificación de LDPC del código de LDPC procedente del desintercalador 55 por giro de columna, sobre el que han sido realizados el proceso de reordenación inversa y la desintercalación por giro de columna y no ha sido realizada ninguna desintercalación de paridad, utilizando una matriz de comprobación de paridad convertida obtenida al realizar al menos permutación correspondiente a intercalación de paridad sobre la matriz H de comprobación de paridad que el codificador 21 de LDPC utilizó para codificación de LDPC.

30 En este caso, la descodificación de LDPC, que se realiza utilizando la matriz de comprobación de paridad convertida con el fin de reducir las dimensiones de la circuitería y limitar la frecuencia operativa dentro de una gama que pueda ser completamente alcanzable, ha sido sugerida con anterioridad (por ejemplo, véase la publicación de la solicitud de Patente japonesa núm. 2004-343170).

35 En primer lugar, la descodificación de LDPC utilizando la matriz de comprobación de paridad convertida sugerida anteriormente, va a ser descrita haciendo referencia a las figuras 31 a 34.

40 La figura 31 ilustra un ejemplo de matriz H de comprobación de paridad de un código de LDPC que tiene una longitud de código N de 90 y una tasa de código de 2/3.

En la figura 31, “0” está representado por un período “.” al igual que en las figuras 32 y 33 descritas más adelante.

Una matriz de paridad de la matriz H de comprobación de paridad de la figura 31 tiene una estructura gradual.

45 La figura 32 ilustra una matriz H' de comprobación de paridad obtenida mediante realización de permutación de fila de expresión matemática (8) y permutación de columna de expresión matemática (9) sobre la matriz H de comprobación de paridad de la figura 31.

Permutación de fila: fila $(6s + t + 1)^{\text{ésima}}$ → fila $(5t + s + 1)^{\text{ésima}}$ (8)

Permutación de columna: columna $(6x + y + 61)^{\text{ésima}}$ → columna $(5y + x + 61)^{\text{ésima}}$ (9)

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

De acuerdo con la permutación de fila de la expresión matemática (8), la 1^a, 7^a, 13^a, 19^a y 25^a filas, cuyos números

ordinales producen "1" como resto cuando se dividen por 6, son cambiadas a (específicamente, intercambiadas con) las filas 1^a, 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 producen "2" como resto cuando se dividen por 6, son cambiadas a las 6^a, 7^a, 8^a, 9^a y 10^a, respectivamente.

- 5 De acuerdo con la permutación de columna de la expresión matemática (9), las columnas 61^a, 67^a, 73^a, 79^a y 89^a entre (paridad) columnas posteriores a la columna 60^a, cuyos números ordinales producen "1" de resto cuando se dividen por 6, son cambiadas a las columnas 61^a, 62^a, 63^a, 64^a y 65^a, respectivamente, y las columnas 62^a, 68^a, 74^a, 80^a y 86^a, cuyos números ordinales producen "2" como resto cuando se dividen por 6, son cambiadas a columnas 66^a, 67^a, 68^a, 69^a y 70^a, respectivamente.

- 10 Una matriz obtenida al realizar permutación de fila y de columna sobre la matriz H de comprobación de paridad de la figura 31, constituye de esta manera la matriz H' de comprobación de paridad de la figuras 32.

En este caso, realizar la permutación de fila de la matriz H de comprobación de paridad no afecta al orden de los bits de código del código de LDPC.

- 15 La permutación de columna de expresión matemática (9) corresponde a la intercalación de paridad que se realiza para intercalar el bit de código $(K + qx + y + 1)^{\text{ésimo}}$ en la posición de bit de código $(K + Py + x + 1)^{\text{ésimo}}$ según se ha descrito en lo que antecede cuando la longitud de información K es "60", el número de columnas P de la unidad que tienen estructura cíclica es "5", y el divisor $q(M/P)$ de longitud de paridad M (30 en este ejemplo) es "6".

- 20 La salida es un vector cero si la matriz H' de comprobación de paridad de la figura 32, que en lo que sigue se mencionará como "matriz de comprobación de paridad convertida" según sea apropiado, se multiplica por un código de LDPC obtenido al realizar la misma permutación que la expresión matemática (9) sobre el código de LDPC de matriz H de comprobación de paridad de la figura 31 que en lo que sigue se mencionará como "matriz de comprobación de paridad original" según sea apropiado. Más específicamente, cuando "c" representa un vector fila obtenido al realizar la permutación de columna de la expresión matemática (9) sobre un vector fila "c" como código de LDPC (una palabra clave) de la matriz H de comprobación de paridad original, Hc^T es un vector cero debido a la naturaleza de la matriz de comprobación de paridad y por lo tanto $H'c^T$ es también un vector cero.

- 25 De ese modo, la matriz H' de comprobación de paridad convertida de la figura 32 es una matriz de comprobación de paridad del código c' de LDPC obtenido mediante realización de la permutación de columna de la expresión matemática (9) sobre el código c de LDPC de la matriz h de comprobación de paridad original.

- 30 En consecuencia, el mismo código de LDPC de la matriz H de comprobación de paridad original que el obtenido mediante descodificación utilizando la matriz H de comprobación de paridad, puede ser obtenido mediante descodificación de LDPC del código c' de LDPC columna permutada, que se produjo al realizar la permutación de columna de la expresión matemática (9) sobre el código c de LDPC de la matriz H de comprobación de paridad original, utilizando la matriz H' de comprobación de paridad convertida de la figura 32, y a continuación realizando la inversa de la permutación de columna (9) sobre el código c' de LDPC descodificado.

- 35 La figura 33 ilustra la matriz H' de comprobación de paridad convertida de la figura 32, en la que los elementos han sido mostrados como elementos que están dispuestos en unidades de matrices de 5 x 5, separadas entre sí.

- 40 En la figura 33, la matriz H_j de comprobación de paridad convertida ha sido mostrada como una combinación de matrices unidad de 5 x 5, cada una de las cuales se produce reemplazando uno o más "1s" de una matriz unidad de 5 x 5, por "0s" (mencionadas en lo que sigue como "matrices quasi-unidad" según sea apropiado), matrices producidas desplazando cíclicamente matrices unidad o matrices quasi-unidad (mencionadas en lo que sigue como "matrices desplazadas", según sea apropiado), cada una de las cuales es la suma de dos o más de entre matrices unidad, matrices quasi-unidad, y matrices desplazadas (en lo que sigue, mencionadas como "matrices suma" según sea apropiado), y matrices cero de 5 x 5.

- 45 Es decir, la matriz H' de comprobación de paridad convertida de la figura 33 puede ser una matriz que incluya matrices unidad de 5 x 5, matrices quasi-unidad, matrices desplazadas, matrices suma, y matrices cero de 5 x 5. De ese modo, las matices 5 x 5, que constituyen la matriz H' de comprobación de paridad convertida, van a ser mencionadas ahora como "matrices componentes" según sea apropiado.

- 50 La descodificación de un código de LDPC mediante una matriz de comprobación de paridad representada por P x P matrices componentes, puede ser llevada a cabo utilizando una arquitectura que realice simultáneamente P cálculos de nodo de comprobación P cálculos de nodo de variable.

- 55 La figura 34 es un diagrama de bloque que ilustra un ejemplo de configuración de un dispositivo de descodificación que realiza descodificación según se ha descrito anteriormente.

Más específicamente, la figura 34 ilustra un ejemplo de configuración de un dispositivo de descodificación que realiza descodificación de un código de LDPC utilizando la matriz H' de comprobación de paridad convertida de la figura 33 obtenida al realizar al menos la permutación de columna de la expresión matemática (9) sobre la matriz H de comprobación de paridad original de la figura 31.

El dispositivo de descodificación de la figura 34 incluye una memoria de almacenamiento de datos de borde que incluye 6 FIFO 300₁ a 300₆, un selector 301 para seleccionar una de las FIFO 300₁ a 300₆. Una unidad 302 de cálculo de nodo de comprobación, dos circuitos 303 y 308 de desplazamiento cíclico, una memoria 304 de almacenamiento de datos de borde que incluye 18 FIFO 304₁ a 304₁₈, un selector 305 para seleccionar una de las

- 5 FIFO 304₁ a 304₁₈, una memoria 306 de datos recibidos para almacenar información recibida, una unidad 307 de cálculo de nodo de variable, una unidad 309 de cálculo de palabra descodificada, una unidad 310 de permutación de datos recibidos, y una unidad 311 de permutación de datos descodificados.

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

- 10 La memoria 300 de almacenamiento de datos de borde incluye el mismo número de 6 FIFO 300₁ a 300₆ que el número obtenido al dividir el número de filas “30” de la matriz H' de comprobación de paridad convertida de la figura 33 por el número de filas “5” de cada matriz componente. Cada FIFO 300_y (y = 1, 2, ..., 6) incluye regiones de almacenamiento respectivas de múltiples niveles, en o desde cada uno de las cuales pueden ser escritos o leídos simultáneamente mensajes correspondientes al mismo número de “5” bordes que el número de filas y el número de columnas de cada matriz componente. El número de niveles de regiones de almacenamiento de cada FIFO 300_y es “9”, lo que es igual que el máximo de los números de 1s (pesos de Hamming) de una dirección de fila de la matriz de comprobación convertida de la figura 33.

- 15 Los datos (es decir, los mensajes v_i procedentes de nodos de variable) correspondientes a posiciones de “1” de la primera a la quinta filas de la matriz H' de comprobación de paridad convertida de la figura 33, se almacenan en la FIFO 300₁ en una dirección horizontal en cada fila simultáneamente mientras que se descarta el “0”. Específicamente, cuando (j, i) representa un elemento de fila j^{ésima} y de columna i^{ésima}, los datos correspondientes a posiciones de “1” de una matriz unidad de 5 x 5 de (1, 1) a (5, 5) de la matriz H' de comprobación de paridad convertida se almacena en la región de almacenamiento de primer nivel de la FIFO 300₁. Los datos correspondientes a posiciones de “1” de una matriz desplazada de (1, 21) a (5, 25) de la matriz H' de comprobación de paridad convertida, que se obtiene desplazando cíclicamente una matriz unidad de 5 x 5 a la derecha en 3 elementos, se almacena en la región de almacenamiento de segundo nivel. De forma similar, los datos se almacenan en las regiones de almacenamiento de 3º a 8º niveles en asociación con la matriz H' de comprobación de paridad convertida. Los datos correspondientes a posiciones de “1” de una matriz desplazada de (1, 81) a (5, 90) de la matriz H' de comprobación de paridad convertida, que se obtiene sustituyendo “1” en la primera fila por “0” en la matriz unidad de 5 x 5 y desplazando cíclicamente la matriz unidad de 5 x 5 a la izquierda en 1 elemento, se almacena en la región de almacenamiento de noveno nivel.

- 20 Los datos correspondientes a posiciones de “1” desde la 6^a a la 10^a filas de la matriz H' de comprobación de paridad convertida de la figura 33 se almacena en la FIFO 300₂. Específicamente, los datos correspondientes a posiciones de “1” de una primera matriz desplazada incluida en una matriz suma de (6, 1) a (10, 5) de la matriz H' de comprobación de paridad convertida, que se obtiene añadiendo a la primera matriz desplazada obtenida al desplazar cíclicamente una matriz unidad de 5 x 5 a la derecha 1 elemento y una segunda matriz desplazada obtenida al desplazar cíclicamente una matriz unidad de 5 x 5 a la derecha 2 elementos, se almacenan en la región de almacenamiento de primer nivel de la FIFO 300₂. Los datos correspondientes a posiciones de “1” de la segunda matriz desplazada incluidos en la matriz suma de (6, 1) a (10, 5) de la matriz H' de comprobación de paridad convertida, se almacenan en la región de almacenamiento de segundo nivel de la FIFO 300₂.

- 25 Más específicamente, cuando una matriz componente que tiene un peso de 2 o más está representada por la suma de dos o más de entre una matriz unidad de P x P con un peso de 1, una matriz quasi-unidad producida al sustituir uno o más “1s!” de la matriz unidad por “0s”, y una matriz desplazada producida al desplazar cíclicamente en la matriz unidad o la matriz quasi-unidad, los datos correspondientes a posiciones de “1” de la matriz unidad con un peso de 1, la matriz quasi-unidad, o la matriz desplazada (es decir, mensajes correspondientes a bordes pertenecientes a la matriz unidad, la matriz quasi-unidad o la matriz desplazada) se almacenan en la misma dirección (la misma FIFO entre las FIFO 300₁ a 300₆).

- 30 Los datos se almacenan también en las regiones de almacenamiento de los niveles 3º a 9º en asociación con la matriz H' de comprobación de paridad convertida.
- 35 De forma similar, los datos se almacenan en las FIFO 300₃ a 300₆ en asociación con la matriz H' de comprobación de paridad convertida.

- 40 La memoria 304 de almacenamiento de datos de borde incluye el mismo numero de 18 FIFO 304₁ a 304₁₈ como número obtenido al dividir el número de columnas “90” de la matriz H_j de comprobación de paridad convertida por el número de columnas “5” de cada matriz componente. Cada FIFO 304_x (x = 1, 2, .., 18) incluye regiones respectivas de almacenamiento de múltiples niveles, en o desde las que pueden ser simultáneamente escritos o leídos mensajes correspondientes al mismo número de “5” bordes que el número de filas y el número de columnas de cada matriz H' componente convertida.

45 Los datos (es decir, los mensajes u_i procedentes de nodos d comprobación) correspondientes a posiciones de “1” de

la primera a quinta columnas de la matriz H' de comprobación de paridad convertida de la figura 33 se almacenan en la FIFO 304₁ en dirección vertical en cada columna simultáneamente mientras se descarta el "0". Específicamente, datos correspondientes a posiciones de "1" de una matriz unidad de 5×5 de (1, 1) a (5, 5) de la matriz H' de comprobación de paridad convertida se almacenan en la región de almacenamiento de primer nivel de la FIFO 304₁.

- 5 Datos correspondientes a posiciones de "1" de una primera matriz desplazada incluida en una matriz suma de (6, 1) a (10, 5) de la matriz H' de comprobación de paridad convertida, que se obtiene al añadir a una primer matriz desplazada producida al desplazar cíclicamente una matriz unidad de 5×5 a la derecha 1 elemento y una segunda matriz desplazada producida al desplazar cíclicamente una matriz unidad de 5×5 dos elementos a la derecha, se almacenan en la región de almacenamiento de segundo nivel. Datos correspondientes a posiciones de "1" de la segunda matriz desplazada incluida en a matriz suma de (6, 1) a (10,5) de la matriz H' de comprobación de paridad convertida, se almacenan en la región de almacenamiento de tercer nivel.

Más específicamente, cuando una matriz componente que tiene un peso de 2 o más está representada por la suma de dos o más matrices unidad de $P \times P$ con un peso de j , una matriz quasi-unidad producida por sustitución de uno o más "1s" de la matriz unidad por "0s", y una matriz desplazada producida al desplazar cíclicamente la matriz unidad o la matriz quasi-unidad, datos correspondientes a posiciones de "1" de la matriz unidad con un peso de 1, la matriz quasi-unidad o la matriz desplazada (es decir, mensajes correspondientes a bordes pertenecientes a la matriz unidad, la matriz quasi-unidad o la matriz desplazada) se almacenan en la misma dirección (la misma FIFO entre las FIFO 304₁ a 304₁₈).

- 10 Los datos se almacenan también en las regiones de almacenamiento de 4^o y 5^o niveles en asociación con la matriz H' de comprobación de paridad convertida. El número de los niveles de regiones de almacenamiento de la FIFO 304₁ es de "5", lo que es igual al máximo de los números de 1s (peso de Hamming) en una dirección de fila en la primera a quinta columnas de la matriz H' de comprobación de paridad convertida.

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

Ahora se hará referencia a la operación del dispositivo descodificador de la figura 34.

- 20 En la memoria 300 de almacenamiento de datos de borde que incluye 6 FIFO 300₁ a 300₆, se selecciona una FIFO para almacenamiento de datos a partir de las FIFO 300₁ a 300₆ de acuerdo con la información (datos de matriz) D312 que indica una fila en la matriz H' de comprobación de paridad convertida, a la que pertenecen 5 mensajes D311, recibidos desde el circuito 308 de desplazamiento cíclico situado corriente arriba de la memoria 300 de arranque de datos de borde, y los 5 mensajes D311 son recogidos y almacenados en la FIFO seleccionada por orden. Cuando se leen datos desde la memoria de almacenamiento de datos de borde, en primer lugar, 5 mensajes 35 D300 son leídos por orden a partir de la memoria 300₁ y son proporcionados a continuación al selector 301 situado corriente debajo de la memoria 300 de almacenamiento de borde. Después de que los mensajes son leídos completamente desde la FIFO 300₁, los mensajes son leídos por orden desde las FIFO 300₂ a 300₆ de la memoria 300 de almacenamiento de datos de borde, y son proporcionados después al selector 301 de la misma manera.

El selector 301 selecciona 5 mensajes recibidos desde una FIFO desde la que han sido leídos normalmente los datos entre las FIFO 300₁ a 300₆ de acuerdo con una señal D301 de selección, y proporciona los mensajes seleccionados como mensajes D302 a la unidad 302 d cálculo de nodo de comprobación.

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

El circuito 303 de desplazamiento cíclico desplaza cíclicamente los 5 mensajes D303₁ a D303₅ obtenidos por medio de la unidad 302 de cálculo de nodo de comprobación en base a la información (datos de matriz) D305 que incluye el número de elementos mediante el que una matriz unidad original fue desplazada cíclicamente para obtener cada borde correspondiente en la matriz H' de comprobación de paridad convertida y proporciona los mensajes desplazados cíclicamente como mensajes D304 a la memoria 304 de almacenamiento de datos de borde.

En la memoria 304 de almacenamiento de datos de borde que incluye 18 FIFO 304₁ a 304₁₈, se selecciona una FIFO para almacenar datos entre las FIFO 304₁ a 304₁₈ de acuerdo con la información D05 que indica una fila de la matriz H' de comprobación de paridad convertida, a la que pertenecen 5 mensajes D304, recibidos desde el circuito 303 de desplazamiento cíclico situado corriente arriba de la memoria 304 de almacenamiento de datos de borde, y los 5 mensajes D304 son recogidos y almacenados en la FIFO seleccionada, por orden. Cuando se leen datos a partir de la memoria 304 de almacenamiento de datos de borde, en primer lugar, 5 mensajes D306₁ son leídos por orden desde la FIFO 304₁ y a continuación son proporcionados al selector 305 situado corriente debajo de la

memoria 304 de almacenamiento de datos de borde. Después de que los datos han sido completamente leídos desde la FIFO 304, los mensajes son leídos por orden desde las FIFO 304₂ a 304₁₈ de la memoria 304 de almacenamiento de datos, y a continuación son proporcionados al selector 305 de la misma manera.

- 5 El selector 305 selecciona 5 mensajes recibidos desde una FIFO desde la que los datos están siendo leídos normalmente entre las FIFO 304₁ a 304₁₈ de acuerdo con una señal de selección D307 y proporciona los mensajes seleccionados como mensajes D308 tanto a la unidad 307 de cálculo de nodo de variable como al calculador 309 de palabra descodificada.

Por otra parte, la unidad 310 de permutación de datos recibidos realiza la permutación de columna de la Expresión 1proporciona los datos resultantes como datos D314 recibidos a la memoria 306 de datos recibidos. La memoria 306 de datos recibidos calcula y almacena una Relación de Probabilidad Logarítmica (LLR) de recepción procedente de los datos D314 recibidos, que se han recibido desde la unidad 310 de permutación de datos recibidos, y proporciona LLRs de recepción en grupos de 5 LLRs como valores D309 recibidos tanto a la unidad 307 de cálculo de nodo de variable como a la unidad 309 de cálculo de palabra descodificada.

10 15 La unidad 307 de cálculo de nodo de variable incluye 5 calculadores 307₁ a 307₅ de nodo de variable, y realiza cálculos de nodo de variable de acuerdo con la ecuación (1) utilizando los mensajes D308 (D308₁ a D308₅) (correspondiente4s a mensajes u_j en la ecuación (1)) recibidos a través del selector 305, y los 5 valores D309 recibidos (correspondientes a valores u_{oi} recibidos en la ecuación (1)) recibidos desde la memoria 306 de datos recibidos, y proporciona a continuación 5 mensajes D310 (D310₁ a D310₅) (correspondientes a mensajes v_i en la ecuación (1)) recibidos a través de los cálculos de nodo de variable, al circuito 308 de desplazamiento cíclico.

20 25 30 35 El circuito 308 de desplazamiento cíclico desplaza los 5 mensajes D310₁ a D310₅ calculados por la unidad 307 de cálculo de nodo de variable en base a información que indica el número de elementos por el que una matriz unidad original fue desplazada cíclicamente para obtener cada borde correspondiente en la matriz H' de comprobación de paridad convertida, y proporciona los mensajes cíclicamente desplazados como mensajes D11 a la memoria 300 de almacenamiento de datos de borde.

40 45 50 El código de LDPC puede ser descodificado una vez realizando las operaciones anteriores una vez. Tras la descodificación del código de LDPC un número predeterminado de veces, el dispositivo de descodificación de la figura 34 obtiene y presenta a la salida datos descodificados finales por medio de la unidad 309 de cálculo de palabra descodificada y de la unidad 311 de permutación de datos descodificados.

Más específicamente, la unidad 309 de cálculo de palabra descodificada incluye 5 calculadores 309₁ a 309₅ de palabra descodificada y realiza, como proceso final de una pluralidad de procedimientos de descodificación, el cálculo de datos descodificados (es decir, una palabra descodificada) en base a la ecuación (5) utilizando los 5 mensajes D308 (D308₁ a D308₅) (correspondientes a mensajes u_j en la ecuación (5)) salientes desde el selector 305 y los 5 valores D309 recibidos (correspondientes a valores u_{oi} recibidos en la ecuación (5)) recibidos desde la memoria 306 de datos y proporciona los datos D315 descodificados calculados a la unidad 311 de permutación de datos descodificados.

La unidad 311 de permutación de datos descodificados realiza la inversión de la permutación de columna de la expresión matemática (9) sobre los datos D315 descodificados recibidos desde la unidad 309 de cálculo de palabra descodificada para cambiar el orden de los datos D315 descodificados y presenta a continuación en la salida los datos resultantes como datos D316 descodificados finales.

Según se ha descrito con anterioridad, una o ambas de entre permutación de fila y permutación de columna, se realizan sobre la matriz de comprobación de paridad (es decir, la matriz de comprobación de paridad original) para convertirla en una matriz de comprobación de paridad (es decir, una matriz de comprobación de paridad convertida) que puede ser representada por medio de una combinación de matrices componentes, es decir, una combinación de matrices unidad P x P, una matriz quasi-unidad producida por sustitución de uno o más “1s” de la matriz unidad por “0s”, una matriz desplazada producida al desplazar cíclicamente la matriz unidad o la matriz quasi-unidad, o la matriz de desplazamiento, y una matriz cero P x P. La conversión de matriz de comprobación de paridad hace que sea posible emplear, cuando se descodifica un código de LDPC, una arquitectura que realiza simultáneamente P cálculos de nodo de comprobación y P cálculos de nodo de variable. Realizar simultáneamente P cálculos de nodo limita I frecuencia operativa dentro de una gama que puede ser completamente alcanzable, haciendo por ello que sea posible realizar descodificación un número de veces.

De manera similar al dispositivo de descodificación de la figura 34, el descodificador 56 de LDPC incluido en el receptor 12 de la figura 29 está diseñado para descodificar un código de LDPC realizando simultáneamente P cálculos de nodo de comprobación y P cálculos de nodo de variable.

Más específicamente, cuando se ha supuesto por facilidad de la explicación que la matriz de comprobación de paridad de un código de LDPC saliente del codificación 21 de LDPC incluido en el transmisor 11 de la figura 8 es una matriz H de comprobación de paridad en la que una matriz de paridad tiene una estructura gradual, por ejemplo según se muestra en la figura 31, el intercalador 23 de paridad del transmisor 11 realiza intercalación de paridad para intercalar un bit de código (K + qx + y + 1)^{ésimo} en una posición de bit de código (K + Py + x + 1)^{ésimo} con la

longitud de información K que es "60", teniendo el número de columnas unidad P que tienen una estructura cíclica que es "5", y el divisor q (= M/P) de longitud de paridad M que es "6".

Puesto que esta intercalación de paridad corresponde a la permutación de columna de la expresión matemática (9) según se ha descrito en lo que antecede, el descodificador 56 de LDPC no necesita realizar la permutación de columna de la expresión matemática (9).

Por lo tanto, en el receptor 12 de la figura 29, un código de LDPC que no está sometido a desintercalación de paridad, es decir, un código de LDPC con la permutación de columna de la expresión matemática (9) realizada, se proporciona desde el desintercalador 55 por giro de columna al descodificador 56 de LDPC según se ha descrito anteriormente. El descodificador 56 de LDPC realiza los mismos proceso que los del dispositivo de descodificación de la figura 34 salvo que la permutación de columna de la expresión matemática (9) no se realiza en el descodificador 56 de LDPC.

Más específicamente, la figura 35 muestra un ejemplo de configuración del descodificador 56 de LDPC de la figura 29.

El descodificador 56 de LDPC mostrado en la figura 35 tiene la misma configuración que la del dispositivo de descodificación de la figura 34 salvo que no se ha previsto la unidad 310 de permutación de datos recibidos de la figura 34 en el descodificador 56 de LDPC y de ese modo se omite en la presente memoria una descripción de la misma configuración y los mismos procesos.

El descodificador 56 de LDPC puede ser de tamaño reducido en comparación con el dispositivo de descodificación de la figura 34 puesto que el descodificador 56 de LDPC puede ser construido sin la unidad 310 de permutación de datos recibidos según se ha descrito anteriormente.

Aunque, por simplicidad de la explicación, las figuras 31 a 35 han sido descritas con referencia a un ejemplo en el la longitud de código N de un código de LDPC es 90, la longitud de información K es 60, el número de columnas P de la unidad que tienen 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 de columnas P de la unidad que tienen estructura cíclica, y el divisor q (= M/P) no están limitados a esos valores.

De ese modo, mientras que el codificador 21 de LDPC del transmisor 11 de la figura 8 presenta a la salida un código de LDPC que tiene, por ejemplo, una longitud de código N de 64800, una longitud de información K de N-Pq (= N-M), un número de columnas P de la unidad que tienen estructura cíclica de 360, y un divisor q de M/P, el descodificador 56 de la figura 35 puede ser aplicado a descodificación LDPC del código de LDPC realizando simultáneamente P cálculos de nodo de comprobación y P cálculos de nodo de variable.

La serie anterior de procesos puede ser llevada a cabo no solo mediante hardware sino también mediante software. Cuando la serie de procesos se realizan mediante software, un programa que implementa el software se instala en un ordenador de propósito general o similar.

35 La figura 36 ilustra un ejemplo de configuración de una realización de un ordenador con un programa para realizar la serie anterior de procesos, instalado en el mismo.

El programa puede ser grabado previamente en un disco duro 405 o una ROM 403 como medio de grabación incorporado en el ordenador.

40 El programa puede estar almacenado (o grabado) temporalmente o permanentemente en un medio 411 de grabación extraíble tal como un disco flotante, un Disco Compacto-Memoria de Solo Lectura (CD-ROM), un Disco Magneto-Óptico (MOD), un Disco Versátil Digital (DVD), un disco magnético, o una memoria de semiconductor. Este medio 411 de grabación extraíble puede ser proporcionado en lo que se conoce como paquete de software.

45 En vez de instalar el programa desde un medio 411 de grabación extraíble como se ha descrito anteriormente en un ordenador, el programa puede ser transmitido inalámbricamente desde un sitio de descarga hasta un ordenador a través de un satélite para radiodifusión digital por satélite, o puede ser transmitido por cable hasta 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 408 de comunicación y puede instalar el programa recibido en un disco duro 405 incorporado.

50 El ordenador puede incluir una Unidad Central de Proceso (CPU) 402. La CPU 402 está acoplada a una interfaz 410 de entrada/salida (IO) a través de un bus 401. La CPU 402 ejecuta un programa almacenado en la Memoria de Solo Lectura (ROM) 403 cuando un comando, que es introducido por el usuario, por ejemplo operando una unidad 407 de entrada que incluye un teclado, un ratón, un micrófono y similar, haya sido recibido a través de la interfaz 410 de IO. Alternativamente, la CPU 402 carga en una Memoria de Acceso Aleatorio (RAM) 404 y ejecuta un programa almacenado en el disco duro 405, un programa que ha sido instalado en el disco duro 405 después de ser recibido desde un satélite o una red a través de la unidad 408 de comunicación, o un programa que ha sido instalado en el

disco duro 405 después de ser leído desde el medio 411 de grabación extraíble instalado en un excitador 409. Ejecutando el programa de esa manera, la CPU 402 realiza los procesos descritos anteriormente con referencia a los diagramas de flujo o a los procesos llevados a cabo por los componentes descritos anteriormente con referencia a los diagramas de bloques. A continuación, según se necesite, la CPU 402 presenta a la salida los resultados de los procesos, por ejemplo a través de una unidad 406 de salida que incluye un Visualizador de Cristal Líquido (LCD), un altavoz, o similar a través de una interfaz 410 de I/O o transmite los resultados del proceso a través de la unidad 408 de comunicación o graba los resultados del proceso en el disco duro 405.

En la descripción anterior, debe apreciarse que las etapas que describen el programa que hace que el ordenador realiza varios tipos de procesamiento no son realizados necesariamente de forma cronológica por el orden descrito anteriormente con referencia a los diagramas de flujo, y pueden ser llevados a cabo en paralelo o individualmente (por ejemplo, mediante procesamiento paralelo o mediante procesamiento orientado al objeto).

El programa puede ser operado con un ordenador o puede ser operado con múltiples ordenadores de una manera distribuida. El programa puede ser transferido también a un ordenador remoto de modo que sea ejecutado en el ordenador remoto.

15 Los espertos en la materia podrán apreciar que las realizaciones de la invención no se limitan a las descritas en lo que antecede, y que se pueden realizar diversos cambios sin apartarse del alcance de la invención según se divulga en las reivindicaciones que se acompañan.

De manera más específica, aunque la intercalación de paridad o la intercalación por giro de columna, que es un proceso de permutación, se realice sobre un código de LDPC definido en la especificación DVB-S.2 en las 20 realizaciones anteriores, la intercalación de paridad puede ser aplicada a un código de LDPC de una matriz de comprobación de paridad en la que una matriz de información no tenga una estructura cíclica, siempre que la matriz de paridad de la matriz de comprobación de paridad tenga una estructura gradual, y la intercalación por giro de columna como proceso de permutación pueda ser aplicado, por ejemplo, a un código de LDPC de una matriz de comprobación de paridad que sea convertida en una estructura pseudo-cíclica por medio de al menos permutación 25 de columna o un código de LDPC Quasi-Cíclico (QC) de una matriz de comprobación de paridad que tenga una estructura cíclica en su totalidad.

Es decir, la matriz de comprobación de paridad de un código de LDPC que va a ser sometida a intercalación de paridad, solamente necesita incluir una matriz de paridad que tenga una estructura gradual y no necesita incluir una matriz de información que tenga una estructura cíclica.

30 La matriz de comprobación de paridad de un código de LDPC que va a ser sometida a intercalación por giro de columna como proceso de permutación, no está limitada a ninguna estructuras específica.

Adicionalmente, el proceso de permutación solamente necesita estar capacitado para permutar bits de código de un código de LDPC de tal modo que una pluralidad de bits de código correspondientes a "1" de una fila arbitraria de la matriz de comprobación de paridad no estén incorporados en el mismo símbolo, y pueda ser llevado a cabo 35 utilizando un método distinto de la intercalación por giro de columna. Más específicamente, el proceso de permutación puede ser llevado a cabo controlando direcciones de escritura y de lectura, por ejemplo utilizando una memoria en la que los datos son almacenados solamente en una dirección en vez de la memoria 31 en la que los datos son almacenados en direcciones de columna y de fila.

Intercalador de símbolo

40 Se ha propuesto que el número de modos, que están disponibles dentro del estándar DVB-T2, pueda ser ampliado de modo que incluya un modo de 1qk, un modo de 16k y un modo de 232k. La descripción que sigue se proporciona para ilustrar la operación de un intercalador de símbolo de acuerdo con la presente técnica, aunque se apreciará que el intercalador de símbolo puede ser usado con otros modos y estándares distintos del DVB.

Para crear nuevos modos, deben ser definidos varios elementos, uno de los cuales es el intercalador 33 de símbolo. 45 El mapeador 26 de bit en constelación, el intercalador 33 de símbolo y el generador 32 de trama, han sido mostrados con mayor detalle en la figura 37.

Según se ha explicado anteriormente, la presente técnica proporciona facilidad para proporcionar un mapeo quasi-óptimo de los símbolos de datos sobre señales de sub-portadora de OFMD. De acuerdo con el ejemplo de técnica, el intercalador de símbolo se proporciona para efectuar el mapeo óptimo de símbolos de datos de entrada sobre señales de sub-portadora de OFMD de acuerdo con un código de permutación y un polinomio generador, lo que ha sido verificado mediante análisis de simulación. El intercalador de símbolo combina, por lo tanto, con el intercalador de bit y con la codificación de LDPC para mejorar el rendimiento de datos en canales de comunicación tales como los propuestos para DVB.

55 Según se muestra en la figura 37, se proporciona un ejemplo de ilustración más detallada del mapeador de constelación de bit respecto a símbolo y del generador 32 de trama, para ilustrar un ejemplo de realización de la presente técnica. Bits de datos recibidos desde el intercalador 26 de bit a través de un canal 62, son agrupados en

conjuntos de bits para ser mapeados sobre una célula de datos, de acuerdo con un número de bits por símbolo proporcionado por el esquema de modulación. Los grupos de bits, que forman una palabra de datos, se alimentan en paralelo a través de canales 64 de datos al procesador 66 de mapeo. El procesador 66 de mapeo selecciona a continuación uno de los símbolos de datos, de acuerdo con un mapeo previamente asignado. El punto de constelación, está representado por un componente real y uno imaginario proporcionado al canal 29 de salida como una de un conjunto de entradas al generador 32 de trama.

El generador 32 de trama recibe las células de datos desde el mapeador 28 de bit frente a constelación a través del canal 29, junto con células de datos procedentes de otros canales 31. Después de generar una t rama de muchas secuencias de células de OFDM, las células de cada símbolo de OFDM son escritas a continuación en una memoria 100 de intercalador y extraídas por lectura de la memoria 100 de intercalador de acuerdo con direcciones de escritura y direcciones de lectura generadas por un generador 102 de dirección. De acuerdo con el orden de entrada por escritura y salida por lectura, se consigue la intercalación de células de datos, generando direcciones apropiadas. La operación del generador de dirección 102 y de la memoria 100 de intercalador va a ser descrita con mayor detalle en lo que sigue con referencia a la figuras 38, 39 y 40. Las células de datos intercaladas son combinadas a continuación con símbolos piloto y de sincronización recibidos desde el conformador 36 de señalización piloto e incorporada en un generador 37 de símbolo de OFDM, para formar el símbolo de OFDM, el cual es alimentado al modulador 38 de OFDM según se ha explicado con anterioridad.

La figura 38 proporciona un ejemplo de partes de un intercalador 33 de símbolo, la cual ilustra la presente técnica para intercalación de símbolos. En la figura 38 las células de datos de entada procedentes del generador 32 de trama son escritos en la memoria 100 de intercalador. Las células de datos son escritas en la memoria 100 de intercalador de acuerdo con una dirección de escritura alimentada desde el generador 102 de dirección por el canal 104, y extraída por lectura desde la memoria 100 de intercalador de acuerdo con una dirección de lectura suministrada desde el generador 102 de dirección por un canal 106. El generador 102 de dirección genera la dirección de escritura y la dirección de lectura según se explica más adelante, dependiendo de si el símbolo de OFDM es par o impar, el cual es identificado a partir de una señal suministrada desde un canal 108, y dependiendo de un modo seleccionado que se identifica a partir de una señal suministrada desde un canal 110. Según se ha explicado, el modo puede ser un modo de 1k, un modo de 2k, un modo de 4k, un modo de 8k, un modo de 16k o un modo de 32k. Según se explica más adelante, la dirección de escritura y la dirección de lectura son generadas de forma diferente para símbolos impares y pares según se explica en relación con la figura 39, lo que proporciona un ejemplo de implementación de la memoria 100 de intercalador.

En el ejemplo mostrado en la figura 39, la memoria de intercalador ha sido mostrada de modo que comprende una parte 100 superior que ilustra la operación de la memoria de intercalador del transmisor y una parte inferior 340, que ilustra la operación de la memoria de desintercalador en el receptor. El intercalador 100 y el desintercalador 340 se han mostrado juntos en la figura 39 con el fin de facilitar la comprensión de su funcionamiento. Según se muestra en la figura 39, una representación de la comunicación entre el intercalador 100 y el desintercalador 340 por medio de otros dispositivos y a través de un canal de transmisión ha sido simplificada y representada como sección 140 entre el intercalador 100 y el desintercalador 340. La operación del intercalador 100 se describe en los párrafos siguientes:

Aunque la figura 39 proporcione una ilustración de sólo cuatro células de datos de entrada sobre un ejemplo de cuatro señales de sub-portadora de un símbolo de OFMD, se apreciará que la técnica ilustrada en la figura 39 puede ser ampliada a un número más grande de sub-portadoras tal 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 8 k, 12096 para el modo de 16k y 24192 para el modo de 32k.

El direccionamiento de entrada y de salida de la memoria 100 de intercalador mostrado en la figura 39, es para símbolos impares y pares. Para un símbolo de OFDM par, las células de datos son tomadas desde el canal 120 de entrada y escritas en la memoria 124.1 de intercalador de acuerdo con una secuencia de direcciones 120 generadas para cada símbolo de OFMD por el generador 102 de dirección. Las direcciones de escritura son aplicadas para el símbolo par de modo que, según se ha ilustrado, la intercalación se efectúa mediante la titilación de las direcciones de entrada por escritura. Por lo tanto, cada símbolo intercalado, $y(h(q)) = y'(q)$.

Para símbolos impares se utiliza la misma memoria 124.2 de intercalador. Sin embargo, según se muestra en la figura 39 para el símbolo impar, el orden 132 de entrada por escritura está en la misma secuencia de dirección utilizada para la salida por lectura del símbolo 126 par anterior. Esta característica permite que las implementaciones de intercalador de símbolo impar y par solamente utilicen una memoria 100 de intercalador siempre que la operación de salida por lectura para una dirección dada se realice antes que la operación de entrada por escritura. Las células de datos escritas en la memoria 124 de intercalador durante los símbolos impares son extraídas por lectura a continuación en una secuencia 134 generada por el generador 102 de dirección para el siguiente símbolo par de OFMD, y así sucesivamente. De ese modo, solamente se genera una dirección por símbolo, con entrada por lectura y salida por escritura para el símbolo impar/par de OFMD que se realiza contemporáneamente.

En resumen, según se ha representado en la figura 39, una vez que el conjunto de direcciones $H(q)$ ha sido calculado para todas las sub-portadoras activas, el vector de entrada $Y' = (y_0', y_1', y_2', \dots, y_{N_{max}-1}')$ es procesado con el fin de producir el vector intercalado $Y = (y_0, y_1, y_2, \dots, y_{N_{max}-1})$ definido por:

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

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

En otras palabras, para símbolos de OFMD pares las palabras de entrada son escritas de una forma permutada en una memoria, y leídas de nuevo de una forma secuencial, mientras que para símbolos impares, son escritas secuencialmente y leídas de nuevo permutadas. En el caso anterior, la permutación $H(q)$ se define mediante la siguiente tabla:

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

Tabla 1: permutación para caso simple en el que $N_{\max} = 4$

Según se muestra en la figura 39, el desintercalador 340 opera para invertir la intercalación aplicada por el intercalador 100, aplicando el mismo conjunto de direcciones que las generadas por el generador de dirección equivalente, pero aplicando las direcciones de entrada por escritura y salida por lectura de modo inverso. Como tal, para símbolos pares, las direcciones 342 de entrada por escritura están en orden secuencial, mientras que las direcciones 344 de salida por lectura son proporcionadas por el generador de dirección. De manera correspondiente, para los símbolos impares, el orden 346 de entrada por escritura se determina a partir del conjunto de direcciones generadas por el generador de dirección, mientras que la salida 348 por lectura está en orden secuencial.

Generación de dirección para modos de operación

Un diagrama esquemático de bloques del algoritmo utilizado para generar la función $H(q)$ de permutación ha sido representado en la figura 40 para un modo de 32k. Sin embargo, como podrá apreciarse, el intercalador de modo de 32k de la figura 40 puede ser adaptado para operar como un intercalador de acuerdo con un modo de 1k, 2k, 4k, 8k o uno de 16k realizando la adaptación apropiada del polinomio generador y del código de permutación según se explica en lo que sigue.

En la figura 40, un registro de desplazamiento de retroalimentación lineal está formado por trece niveles de registro 200 y una puerta XOR 202 que está conectada a los niveles del registro de desplazamiento 200 de acuerdo con un polinomio generador. Por lo tanto, de acuerdo con el contenido del registro de desplazamiento 200, un bit siguiente del registro de desplazamiento se proporciona desde la salida de la puerta XOR 202 por aplicación de XOR del contenido de registros de desplazamiento R[0], R[1], R[2], R[12] de acuerdo con el polinomio generador:

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

De acuerdo con el polinomio generador, se genera una secuencia de bit 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 ha ilustrado, se proporciona un circuito 210 de permutación que permuta de forma efectiva el orden de los bits dentro del registro de desplazamiento 200.1 a partir de un orden $R'[n]$ hasta un orden $R[n]$ a la salida del circuito 210 de permutación. Catorce bits procedentes de la salida del circuito 210 de permutación son alimentados a continuación a un canal 212 de conexión a los que se añade un bit más significativo a través del canal 214 que es proporcionado por un circuito 218 basculador. A continuación se genera por tanto una dirección de quince bits sobre el canal 212. Sin embargo, con el fin de asegurar la autenticidad de una dirección, un circuito 216 de comprobación de dirección 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-portadora, que están disponibles para símbolos de datos dentro del símbolo de OFMD, disponible para el modo que se esté utilizando. Sin embargo, el intercalador para el modo de 32k puede ser usado también para otros modos, de modo que el generador 102 de dirección puede ser usado también para el modo de 2k, el modo de 4k, el modo de 8k, el modo de 16k y el modo de 32k, ajustando adecuadamente el número de la dirección máxima válida.

Si la dirección generada excede el valor máximo predeterminado, entonces se genera una señal d control por parte de la unidad 216 de comprobación de dirección y se alimenta a través de un canal 220 de conexión a una unidad 224 de control. Si la dirección generada excede el valor máximo predeterminado, entonces esta dirección es rechazada y se vuelve a generar una nueva dirección para ese símbolo particular.

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

El polinomio utilizado para generar esta secuencia es:

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

50 donde i varía desde 0 hasta $M_{\max}-1$.

Una vez que una palabra R'_i ha sido generada, la palabra R'_i va a producir por medio de una permutación otra palabra de $(N_r - 1)$ bits denominada R_i . R_i se deriva de R'_i mediante permutaciones de bits dadas como sigue:

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

Permutación de bit para el modo de 32k

- 5 Como ejemplo, esto significa que para el modo de 32k, el número de bit 12 de R'_i es enviado a la posición de bit número 5 de R_i .

La dirección $H(q)$ se deriva entonces de R_i a través de la siguiente ecuación:

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

La parte $(i \bmod 2) \cdot 2^{N_r-1}$ de la ecuación anterior se ha representado en la figura 40 mediante el bloque basculador T218.

- 10 A continuación se realiza una comprobación de dirección sobre $H(q)$ para verificar que la dirección generada esté dentro de la gama de direcciones aceptables: si $(H(q) < N_{\max})$, donde $N_{\max} = 24192$, por ejemplo, en el modo de 32k, entonces la dirección es válida. Si la dirección no es válida, la unidad de control es informada y tratará de generar una nueva $H(q)$ incrementando el índice i .
- 15 El papel del bloque basculador es el de asegurar que no se genera una dirección que exceda N_{\max} dos veces en una fila. En efecto, si se genera un valor que lo excede, esto significa que el MSB (es decir, el bit basculador) de la dirección $H(q)$ era uno. Así, el siguiente valor generado tendrá un MSB establecido en cero, asegurando que produce una dirección válida. El bit adicional, por lo tanto, reduce la probabilidad de que si una dirección supera la dirección válida máxima predeterminada, entonces la siguiente dirección será una dirección válida. En un ejemplo, el bit adicional es el bit más significativo.
- 20 Las ecuaciones que siguen resumen el comportamiento global y ayudan a comprender la estructura en bucle de este algoritmo:

$$q = 0$$

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

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

- 25 si $(H(q) < N_{\max})$ $q = q + 1$; }

Análisis que soporta el generador de dirección

- La selección del generador de polinomio y del código de permutación explicados en lo que antecede para el generador 102 de dirección para cada modo operativo, por ejemplo el modo de 32k, hay sido identificado siguiendo análisis de simulación del rendimiento relativo del intercalador. El rendimiento relativo del intercalador ha sido evaluado utilizando una capacidad relativa del intercalador para separar símbolos sucesivos o una "calidad de intercalación". Según se ha mencionado anteriormente, la intercalación efectiva debe ser llevada a cabo para símbolos impares y pares, con el fin de usar una única memoria de intercalador. La medición relativa de la calidad de intercalador se determina definiendo una distancia D (en número de sub-portadoras). Se elige un criterio C para identificar un número de sub-portadoras que estén a una distancia $< D$ a la salida del intercalador que estaba a distancia $< D$ 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 ambos símbolos de OFMD impares y pares. La minimización de C genera un intercalador de calidad superior:

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

- 40 donde: $N_{par}(d)$ y $N_{impar}(d)$ son números de sub-portadoras en un símbolo par e impar, respectivamente, a la salida del intercalador que permanece dentro de una separación de una sub-portadora con respecto a otra.

El análisis del intercalador identificado anteriormente para el modo de 32k para un valor de $D = 5$, ha sido mostrado

en la figura 41(a) para los símbolos de OFDM pares, y en la figura 41(b) para el símbolo de OFDM impar. De acuerdo con el análisis que antecede, el valor de C para el código de permutación identificado anteriormente para el modo de 32k produjo un valor de $C = 21,75$, que el número ponderado de sub-portadoras con símbolos están separadas por cinco o menos en la salida de acuerdo con la ecuación anterior fue de 21,75.

- 5 Se proporciona un análisis correspondiente para un código de permutación alternativo para símbolos de OFDM pares en la figura 41(c) y para símbolos de OFDM impares en la figura 41(d). Tal y como puede apreciarse en comparación con los resultados ilustrados en las figuras 41(a) y 41(b), hay presentes más componentes que representan símbolos separados por pequeñas distancias tales como $D = 1$ y $D = 2$, cuando se comparan con los resultados mostrados en la figura 41(a) y 41(b), que ilustran que el código de permutación identificado anteriormente para el intercalador de símbolo del modo de 32k produce un intercalador de calidad superior.
- 10

Códigos de permutación alternativos

Los quince códigos posibles alternativos que siguen (bits de posición $[n]R'_i$, donde $n = 1$ a 15) han sido hallados para proporcionar un intercalador de símbolo con una buena calidad según se determina mediante el criterio C identificado anteriormente.

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

- 15 Permutación de bit para el modo de 32k

Adaptación del intercalador de símbolo y del generador de dirección para otros modos

- Según se ha mencionado anteriormente, el intercalador de símbolo mostrado en la figura 40 puede ser adaptado para intercalar símbolos desde otros modos cambiando simplemente la dirección válida máxima, el número de niveles del registro de desplazamiento de retroalimentación lineal, y el código de permutación. En particular, de acuerdo con el análisis mencionado anteriormente, se ha establecido lo que sigue para cada uno de los modos de 1k, 2k, 4k, 8k y 16k.

Modo de 1k

Dirección válida máxima: aproximadamente mil

Número de niveles en el registro de desplazamiento de retroalimentación lineal: Nueve

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

Código de permutación:

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

Modo de 2k

Dirección válida máxima: aproximadamente dos mil

Número de niveles en el registro de desplazamiento de retroalimentación lineal: 10

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

- 5 Código de permutación:

Posición de bit $R'[n]$	9	8	7	6	5	4	3	2	1	0
Posición de bit $R[n]$	0	7	5	1	8	2	6	9	3	4

Modo de 4k

Dirección válida máxima: aproximadamente cuatro mil

Número de niveles en el registro de desplazamiento de retroalimentación lineal: Once

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

- 10 Código de permutación:

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

Modo de 8k

Dirección válida máxima: aproximadamente ocho mil

Número de niveles en el registro de desplazamiento de retroalimentación lineal: Doce

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

- 15 Código de permutación:

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

Modo de 16k

Dirección válida máxima: aproximadamente dieciséis mil

Número de niveles en el registro de desplazamiento de retroalimentación lineal: 13

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

- 20 Código de permutación:

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

Descripción adicional de intercalador de símbolo en el receptor

Volviendo al intercalador mostrado en la figura 29, el desintercalador 514 de símbolo ha sido formado a partir de un aparato de procesamiento de datos según se muestra en la figura 42 con una memoria 540 de intercalador y un generador 542 de dirección. La memoria 540 de intercalador es según se muestra en la figura 39 y opera como se ha explicado ya anteriormente para efectuar la desintercalación utilizando conjuntos de direcciones generadas por el generador 542 de dirección. El generador 542 de dirección se ha formado según se muestra en la figura 40 y está dispuesto para que genere direcciones correspondientes para mapear los símbolos de datos recuperados desde cada una de las señales de sub-portadora de OFDM en una corriente de datos de salida.

- 30 Las partes restantes del receptor de OFDM mostrado en la figura 29 han sido previstas para realizar descodificación 518 de corrección de error de los bits de datos codificados de LDPC, para corregir errores y recuperar una estimación de los datos fuente.

Una ventaja proporcionada por la presente técnica tanto para el receptor como para el transmisor consiste en que un intercalador de símbolo y un desintercalador de símbolo que operan en los receptores y transmisores, pueden ser conmutados entre los modos de 1k, 2k, 4k, 8k, 16k y 32k cambiando los polinomios generadores y el orden de permutación. De ahí que el generador 542 de dirección mostrado en la figura 42 incluya una entrada 544, que proporciona una indicación del modo, así como una entrada 546 que indica si existen símbolos de OFDM pares/impares. Se proporciona con ello una implementación flexible puesto que se puede formar un intercalador y desintercalador de símbolo según se muestra en las figuras 38 y 42, con un generador de dirección como el ilustrado en cualquiera de las figuras 40. El generador de dirección puede ser adaptado, por lo tanto, a los diferentes modos cambiando a los polinomios generadores y a los órdenes de permutación indicados para cada uno de los modos. Por ejemplo, esto puede ser llevado a cabo utilizando un cambi9o de software. Alternativamente, en otras realizaciones, una señal incorporada que indique el modo de transmisión de DVB-T2 puede ser detectada en el receptor de la unidad 511 de procesamiento de señalización incorporada, y utilizada para configurar automáticamente el desintercalador de símbolo de acuerdo con el modo detectado.

Alternativamente, según se ha mencionado anteriormente, se pueden usar diferentes intercaladores con diferentes modos, adaptando simplemente la dirección válida máxima de acuerdo con el modo que se esté utilizando.

Uso óptimo de intercaladores impares

Según se ha mostrado en la figura 39, dos procesos de i9ntercalación de símbolo, uno para los símbolos de OFDM pares y uno para los símbolos de OFDM impares, permiten que se reduzca la cantidad de memoria utilizada durante la intercalación. En el ejemplo mostrado en la figura 39, la escritura por orden para el símbolo impar es la misma que el orden de salida por lectura para el símbolo par; por lo tanto, mientras un símbolo impar está siendo leído desde la memoria, un símbolo par puede ser escrito en la posición que se acaba de leer; a continuación, cuando el símbolo par se ha leído en la memoria, el siguiente símbolo impar puede ser escrito en la posición que se acaba de leer.

Según se ha mencionado anteriormente, durante un análisis experimental del comportamiento de los intercaladores (utilizando el criterio C según se ha definido en lo que antecede) y para el ejemplo mostrado en la figura 43(a) y 43(b), se ha descubierto que los esquemas de intercalación diseñados para los intercaladores de símbolo de 2k y 8k para DVB-T y el intercalador de símbolo de 4k para DVB-H, trabajan mejor para símbolos impares que para símbolos pares. De ese modo, a partir de los resultados de evaluación de comportamiento de los intercaladores, según se ha ilustrado mediante las figuras 43(a) y 43(b), se ha puesto de relieve que los intercaladores impares trabajan mejor que los intercaladores pares. Esto puede ser apreciado mediante la comparación de la figura 43(a) que muestra los resultados para un intercalador para símbolos pares, con la figura 43(b) que ilustra los resultados para símbolos impares: se puede ver que la distancia media en la salida de 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.

Como se comprenderá, la cantidad de memoria de intercalador requerida para implementar un intercalador de símbolo depende del número de símbolos de datos que van a ser mapeados sobre los símbolos de portadora de OFDM. Así, un intercalador de símbolo de modo 16k, requiere la mitad de la memoria que se necesita para implementar un intercalador de símbolo de modo 32k y, de manera similar, la cantidad de memoria requerida para implementar un intercalador de símbolo de 8k es la mitad que la requerida para implementar un intercalador de 16k. Por lo tanto, un transmisor o un receptor que esté dispuesto para implementar un intercalador de símbolo de un modo, que establece del número máximo de símbolos de datos que pueden ser portados por el símbolo de OFDM, ya sea receptor o transmisor deberá incluir memoria suficiente para implementar dos procesos de intercalación impares para cualquier otro modo, lo que proporciona la mitad o menos de la mitad del número de sub-portadoras por símbolo de OFDM en ese modo máximo dado. Por ejemplo, un receptor o un transmisor que incluya un intercalador de 32k tenga memoria suficiente para albergar dos procesos de intercalación impar de 16k cada uno con su propia memoria de 16k.

Por lo tanto, con el fin de aprovechar el mejor comportamiento de los procesos de intercalación impar, un intercalador de símbolo capaz de albergar múltiples modos de modulación puede estar dispuesto de modo que solamente se utilice un proceso de intercalación de símbolo impar si es un modo que comprenda la mitad o menos de la mitad del número de sub-portadoras en un modo máximo, que represente el número máximo de sub-portadoras por símbolo de OFDM. El modo máximo establece, por lo tanto, el tamaño máximo de memoria. Por ejemplo, en un transmisor/receptor capacitado para el modo de 32k, cuando opera en un modo con menos portadoras (es decir, 16k, 8k, 4k o 1k), entonces en vez de emplear procesos separados de intercalación de símbolo impar y par se podrían usar dos intercaladores impares.

Una ilustración de una adaptación del intercalador 33 de símbolo que ha sido mostrada en la figura 38 cuando se intercalan símbolos de datos de entrada sobre sub-portadoras de símbolos de OFDM en el modo de intercalación impar solamente, ha sido mostrada en la figura 44. El intercalador 33.1 de símbolo corresponde exactamente al intercalador 33 de símbolo según se muestra en la figura 38, salvo que el generador 102.1 de dirección está adaptado para llevar a cabo el proceso de intercalación impar solamente. Para el ejemplo mostrado en la figura 44, el intercalador 33.1 de símbolo está operando en un modo en que el número de símbolos de datos que pueden ser portados por cada símbolo de OFMD es menor de la mitad del número máximo que puede ser portador por un

símbolo de OFMD en un modo operativo con un número mayor de sub-portadoras por símbolo de OFDM. Como tal, el intercalador 33.1 de símbolo ha sido dispuesto para dividir la memoria 100 de intercalador. Para la presente ilustración mostrada en la figura 44, la memoria 100 de intercalador está dividida en dos partes 601, 602. Como ilustración del intercalador 33.1 de símbolo que opera en un modo en el que los símbolos de datos son mapeados sobre símbolos de OFDM utilizando el proceso de intercalación impar, la figura 44 proporciona una vista expandida de cada mitad de la memoria 601, 602 de intercalador. La expansión proporciona una ilustración del modo de intercalación impar según se ha representado para el lado del transmisor para cuatro símbolos A, B, C, D reproducidos a partir de la figura 39. De ese modo, según se muestra en la figura 44, para conjuntos sucesivos de primeros y segundos símbolos de datos, los símbolos de datos son escritos en la memoria 601, 602 de intercalador por orden secuencial y extraídos por lectura de acuerdo con direcciones generadas por el generador 102 de dirección en un orden permutado de acuerdo con las direcciones generadas por el generador de dirección según se ha explicado anteriormente. Así, según ilustra la figura 44, puesto que se está llevando a cabo un proceso de intercalación impar para conjuntos sucesivos de primeros y segundos conjuntos de símbolos de datos, la memoria de intercalador debe ser dividida en dos partes. Los símbolos de un primer conjunto de símbolos de datos son escritos en una primera mitad de la memoria 601 de intercalador, y los símbolos de un segundo conjunto de símbolos de datos son escritos en una segunda parte de la memoria 602 de intercalador, puesto que el intercalador de símbolo no está ya capacitado para reutilizar las mismas partes de la memoria de intercalador de símbolo que Opudieron ser albergadas cuando operaba en un modo de intercalación impar y par.

Un ejemplo correspondiente de intercalador del receptor, que aparece en la figura 42 pero que está adaptado para operar con un proceso de intercalación impar solamente, ha sido mostrado en la figura 45. Según se muestra en la figura 45, la memoria 540 de intercalador está dividida en dos mitades 710, 712, y el generador 542 de dirección está adaptado para escribir símbolos de datos en la memoria de intercalador y leer símbolos de datos desde la memoria de intercalador en las partes respectivas de la memoria 710, 712 para conjuntos sucesivos de símbolos de datos para implementar un proceso de intercalación impar solamente. Por lo tanto, en correspondencia con la representación mostrada en la figura 44, la figura 45 muestra el mapeo del proceso de intercalación que se lleva a cabo en el receptor y que se ha ilustrado en la figura 39 como vista expandida que es operativa tanto para la primera como para la segunda mitades de la memoria 710, 712 de intercalación. Así, un primer conjunto de símbolos de datos son escritos en una primera parte de la memoria 710 de intercalador en un orden permutado definido de acuerdo con las direcciones generadas por el generador 542 de dirección según haya sido ilustrado por el orden de escritura de los en los símbolos de datos que proporciona una secuencia de escritura de 1, 3, 0, 2. Según se ilustra, los símbolos de datos son a continuación extraídos por lectura de la primera parte de la memoria 710 de intercalador en un orden secuencial recuperando así la secuencia original A, B, C, D.

De manera correspondiente, un segundo conjunto subsiguiente de símbolos de datos que son recuperados a partir de un símbolo sucesivo de OFDM, son escritos en la segunda mitad de la memoria 712 de intercalador de acuerdo con las direcciones generadas por el generador 542 de dirección en un orden permutado, y extraídos por lectura en la corriente de datos de salida en un orden secuencial.

En un ejemplo, las direcciones generadas para un primer conjunto de símbolos de datos para su escritura en la primera mitad de la memoria 710 de intercalador pueden ser reutilizados para escribir un segundo conjunto subsiguiente de símbolos de datos en la memoria 712 de intercalador. De manera correspondiente, el transmisor puede reutilizar también direcciones generadas para una mitad del intercalador para un primer conjunto de símbolos de datos para la salida por lectura de un segundo conjunto de símbolos de datos que han sido escritos en la segunda mitad de la memoria en orden secuencial.

Utilización de una secuencia de permutaciones

En un ejemplo, el generador de dirección puede aplicar un código de permutación diferente procedente de un conjunto de códigos de permutación para símbolos de OFMD sucesivos. La utilización de una secuencia de permutaciones en el generador de dirección de intercalador, reduce la probabilidad de que cualquier bit de entrada de datos al intercalador no module siempre la misma sub-portadora en el símbolo de OFDM. En otro ejemplo, podrían usarse dos generadores de dirección, generando una dirección para el primer conjunto de símbolos de datos y la primera mitad de la memoria, y generando el otro una secuencia diferente de direcciones para el segundo conjunto de símbolos de datos y la segunda mitad de la memoria. Los dos generadores de dirección podrían diferir en su elección de código de permutación en la tabla de buenas permutaciones anteriores, por ejemplo.

Por ejemplo, se podría usar una secuencia cíclica, de modo que un código de permutación diferente en un conjunto de códigos de permutación de una secuencia sea usado para sucesivos símbolos de OFDM y después se repita. Esta secuencia cíclica podría ser, por ejemplo, de longitud dos o cuatro. Para el ejemplo del intercalador de símbolo de 16k, una secuencia de dos códigos de permutación que sean sometidos a operación cíclica, podría ser por ejemplo:

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

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

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

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

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

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

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

La conmutación de un código de permutación a otro podría ser efectuada en respuesta a un cambio en la señal de Impar/Par indicado en el canal 108 de control. En respuesta, la unidad 224 de control cambia el código de permutación en el circuito 210 de código de permutación a través de la línea 111 de control

Para el ejemplo de un intercalador de símbolo de 1k, dos códigos de permutación podrían ser:

10 4 3 2 1 0 5 6 7 8

3 2 5 0 1 4 7 8 6

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

15 7 5 3 8 2 6 1 4 0

1 6 8 2 5 3 4 0 7

Otras combinaciones de secuencias pueden ser posibles para modos de portadora de 2k, 4k y 8k, o en verdad un modo de portadora de 0,5k. Por ejemplo, los códigos de permutación siguientes para cada uno de los modos de 0,5k, 2k, 4k y 8k proporciona una buena des-correlación de símbolos por medio de un generador de dirección para cada uno de los modos respectivos:

Modo de 2k:

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

25 7 0 4 8 3 6 9 1 5 2

Modo de 4k:

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

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

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

30 1 4 1 0 3 9 7 2 6 5 0 8

Modo de 8k:

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

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

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

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

Para los códigos de permutación indicados anteriormente, los dos primeros podrían ser usados en un ciclo de secuencia dos, mientras que la totalidad de los cuatro podrían ser usados para un ciclo de secuencia cuatro. Adicionalmente, secuencias adicionales de cuatro códigos de permutación, que son sometidos a operaciones cíclicas para proporcionar desviación en el generador de dirección para producir una buena des-correlación en los símbolos intercalados (algunos son comunes a los anteriores), se proporcionan a continuación:

40 Modo de 0,5k:

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

5 Modo de 2k:

0 7 5 1 8 2 6 9 3 4*

3 2 7 0 1 5 8 4 9 6

4 8 3 2 9 0 1 5 6 7

7 3 9 5 2 1 0 6 4 8

10 Modo de 4k:

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

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

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

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

15 Modo de 8k:

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

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

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

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

20 * Éstas son las permutaciones en el estándar DVB-T

** Éstas son las permutaciones en el estándar DVB-H

Ejemplos de generadores de dirección, y de intercaladores correspondientes, para los modos de 2k, 4k y 8k han sido divulgados en la solicitud de Patente Europea número 04251667.4. Un generador de dirección para el modo de 0,5k ha sido divulgado en la solicitud de Patente en trámite en UK número 0722553.5.

25 Según podrá ser apreciado, el transmisor y el receptor mostrados en las figuras 1 y 7, respectivamente, han sido facilitados como ilustraciones solamente y no se pretende que sean limitaciones. Por ejemplo, se podrá apreciar que la posición del intercalador y el desintercalador de símbolo con respecto, por ejemplo, al intercalador de bit y al mapeador y al desmapeador, pueden ser cambiados. Según podrá apreciarse, el efecto del intercalador y del desintercalador se mantiene sin cambio en base a sus posiciones relativas, aunque el intercalador pueda estar intercalando símbolos de I/Q en vez de vectores de v-bit. Un cambio correspondiente podría ser realizado en el receptor. En consecuencia, el intercalador y el desintercalador pueden estar operando sobre diferentes tipos de datos, y pueden ser posicionados de manera diferente a la posición descrita en los ejemplos de realización.

35 Según se ha explicado anteriormente, los códigos de permutación y el polinomio generador del intercalador, que han sido descritos con referencia a una implementación de un modo particular, pueden ser aplicados igualmente a otros modos, cambiando la dirección permitida máxima predeterminada de acuerdo con el número de portadoras para ese modo.

De acuerdo con una implementación de un receptor, se incluye un aparato de procesamiento de datos operable para mapear símbolos de datos recibidos a partir de un número predeterminado de señales de sub-portadora de símbolos Multiplexados por División de Frecuencia Ortogonal OFDM en una corriente de datos de salida.

40 Según se ha mencionado anteriormente, las realizaciones de la presente invención encuentran aplicación con estándares DVB tales como DVB-T, DVB-T2 y DVB-H, que se incorporan a la presente por referencia. Por ejemplo, las realizaciones de la presente invención pueden ser usadas en un transmisor o un receptor que operen de acuerdo con el estándar DVB-T2 según se ha especificado de acuerdo con el estándar EN 302 755 de ETSI, aunque se podrá apreciar que la presente invención no se limita a su aplicación con DVB y puede ser extendida a otros estándares para transmisión o recepción, tanto fijos como móviles. En otros ejemplos de realización de la presente

invención, encuentran aplicación con el estándar de transmisión por cable conocido como DVB-C2.

Adicionalmente a los ejemplos de realización descritos en lo que antecede y a los aspectos y características de la invención que se definen en las reivindicaciones anexas, otras realizaciones pueden proporcionar un aparato de procesamiento de datos operable para mapear símbolos de entrada que van a ser comunicados sobre un número predeterminado de señales de sub-portadora de un símbolo Multiplexado por División de Frecuencia Ortogonal. El número predeterminado de señales de sub-portadora 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 operable para realizar un primer proceso de intercalación que intercala símbolos de datos de entrada impares en las señales de sub-portadora y un proceso de intercalación par que intercala símbolos de datos de entrada pares en las señales de sub-portadora, el primer proceso de intercalación impar y el proceso de intercalación par que introduce por lectura y extrae por lectura los símbolos de datos para mapear sobre las señales de portadora de OFDM en una memoria de intercalador la salida por lectura que está en un orden diferente al de la entrada por lectura de tal modo que mientras se está leyendo un símbolo impar desde una posición de la memoria, un símbolo par puede ser escrito en la posición justamente leída desde, y cuando, un símbolo par es leído desde la

5 posición de la memoria, un siguiente símbolo impar puede ser escrito en la posición desde la que se acaba de leer, en el que el proceso de intercalación introduce por lectura y extrae por lectura símbolos de datos impares desde la memoria de intercalador de acuerdo con un esquema de intercalación impar, y el proceso de intercalación par introduce por lectura y extrae por lectura símbolos de datos pares desde la memoria de intercalador de acuerdo con un esquema de intercalación par. Cuando el modo de modulación es un modo que incluye la mitad o menos de la

10 mitad de sub-portadoras de un número total de sub-portadoras que pueden ser albergadas en la memoria, el aparato de datos es operable para asignar una porción de la memoria de intercalación al primer proceso de intercalación impar y asignar una segunda porción de la memoria de intercalación a un segundo proceso de intercalación impar de acuerdo con el primero, el segundo proceso de intercalación impar que intercala los símbolos de entrada pares.

15 25 De acuerdo con otro ejemplo de realización, un aparato de procesamiento de datos es operable para mapear símbolos de entrada que van a ser comunicados sobre un número predeterminado de señales de sub-portadora de un símbolo Multiplexado por División de Frecuencia Ortogonal (OFDM). El número predeterminado de señales de sub-portadora corresponde a un modo de modulación y los símbolos de entrada incluyen primeros símbolos de datos para el mapeo sobre un primer símbolo de OFDM y segundos símbolos de datos para el mapeo sobre un

20 30 segundo símbolo de OFDM. El aparato de procesamiento de datos comprende un intercalador operable para realizar un proceso de intercalación impar que intercala primeros símbolos de datos de entrada sobre las señales de sub-portadora y un proceso de intercalación de datos que intercala segundos símbolos de datos de entrada sobre las señales de sub-portadora, escribiendo el proceso de intercalación impar los primeros símbolos de datos de entrada en una memoria de intercalador de acuerdo con un orden secuencial de los primeros símbolos de datos de entrada y

35 35 extrayendo por lectura los primeros símbolos de datos de entrada desde la memoria de intercalador sobre las señales de sub-portadora de acuerdo con un orden definido por un código de permutación, escribiendo el proceso de intercalación par los segundos símbolos de datos de entrada en una memoria de intercalador de acuerdo con un orden definido por el código de permutación y extrayendo por lectura los segundos símbolos de datos desde la memoria de intercalador sobre las señales de sub-portadora de acuerdo con un orden secuencial tal que mientras

40 40 un primer símbolo de datos de entrada está siendo leído desde una posición de la memoria de intercalador, un segundo símbolo puede ser escrito en la posición exacta leída, y cuando un segundo símbolo es leído desde la posición de la memoria de intercalador, un siguiente primer símbolo puede ser escrito en posición desde la que se acaba de leer. Cuando el modo de modulación es un modo que incluye la mitad o menos de la mitad de un número

45 45 de señales de sub-portadora de un número total de sub-portadoras que pueden ser albergadas en la memoria de intercalador, el aparato de datos es operable para intercalar tanto primeros como segundos símbolos de entrada de acuerdo con el proceso de intercalación impar.

Otro ejemplo de realización puede proporcionar un método de mapeo símbolos de entrada que van a ser comunicados sobre un número predeterminado de señales de sub-portadora de un símbolo Multiplexado por División de Frecuencia Ortogonal (=FMD). El método comprende mapear primeros símbolos de datos sobre un primer símbolo de OFDM y mapear segundos símbolos de datos sobre un segundo símbolo de OFDM.

50

REIVINDICACIONES

1.- Un aparato de procesamiento de datos dispuesto en funcionamiento para recuperar bits de datos desde símbolos de datos recibidos desde un número predeterminado de señales de sub-portadora de un símbolo Multiplexado por División de Frecuencias Ortogonales, OFDM, y formar una corriente de bits de salida, comprendiendo el aparato de procesamiento de datos:

5 un desentrelazador (514) de símbolos que puede funcionar para introducir por lectura en una memoria (100) de entrelazador de símbolos el número predeterminado de símbolos de datos desde las señales de sub-portadora OFDM, y para extraer por lectura de la memoria (100) de entrelazador de símbolos los símbolos de datos adentro de una corriente de símbolos de salida para efectuar el mapeo, siendo la extracción por lectura en un orden diferente a

10 la introducción por lectura, determinándose el orden a partir de un conjunto de direcciones, con el efecto de que los símbolos de datos son desentrelazados desde las señales de sub-portadora OFDM adentro de la corriente de símbolos de salida,

15 una unidad (52) de desmapeo que se puede hacer funcionar para generar, a partir de los símbolos de datos de la corriente de símbolos de salida, bits de datos codificados por LDPC, entrelazados por paridad, mediante la conversión de cada uno de los símbolos de datos de la corriente de símbolos de salida que representa un símbolo de modulación de las señales de sub-portadora OFDM en bits de datos de acuerdo con un esquema de modulación,

20 un permutador inverso (53) adaptado para realizar un proceso de permutación inversa para efectuar una reversión de un proceso de permutación aplicado a los bits de datos codificados por LDPC, entrelazados por paridad, para permutar los bits de datos codificados por LDPC de manera que no se incorporan en el mismo símbolo una pluralidad de los bits de datos codificados por LDPC que corresponden a un valor de 1 en una fila arbitraria de una matriz de información que corresponde a bits de información de un código LDPC, que se usó para codificar los bits de datos, y

25 un decodificador LDPC (50) adaptado para realizar una decodificación LDPC sobre los bits de datos codificados por LDPC sobre los que se ha realizado el proceso de permutación inversa para formar los bits de datos de salida;

25 en el que el desentrelazador (514) de símbolos incluye:

un generador (102) de direcciones que se puede hacer funcionar para generar el conjunto de direcciones, generándose una dirección para cada uno de los símbolos de datos recibido para indicar la señal de sub-portadora OFDM desde la que el símbolo de datos recibido se tiene que mapear adentro de la corriente de símbolos de salida, comprendiendo el generador (102) de direcciones:

30 un registro (200, 202) de desplazamiento de retroalimentación lineal, que incluye un número predeterminado de niveles de registro y que es operable para generar una secuencia de bit pseudo-aleatoria de acuerdo con un polinomio generador,

35 un circuito (210) de permutación operable para recibir el contenido de los niveles del registro de desplazamiento y para permutar los bits presentes en los niveles de registro de acuerdo con un código de permutación para formar una dirección de una de las sub-portadoras de OFMD, y

una unidad (224) de control operable en combinación con un circuito de comprobación de dirección (216) para re-generar una dirección cuando una dirección generada exceda de una dirección válida máxima predeterminada;

y en el que:

la dirección válida máxima predeterminada es sustancialmente cuatro mil noventa y seis,

40 el registro (200, 202) de desplazamiento de retroalimentación lineal tiene once niveles de registro con un polinomio generador para el registro de desplazamiento de retroalimentación lineal de $R'_i[10] = R'_{i-1}[0] \oplus R'_{i-1}[2]$, y el orden de permutación forma una dirección de once bits $R_i[n]$ para el símbolo de datos $i^{\text{ésimo}}$ del bit presente en el nivel de registro $n^{\text{ésimo}}$ $R'_i[n]$ de acuerdo con la tabla:

$R'_i[n]$ 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

45 2.- Un aparato de procesamiento de datos según la reivindicación 1, en el que el proceso de permutación inversa realizado por el permutador inverso (53) sobre los datos codificados por LDPC tiene por efecto revertir una permutación de los bits de datos codificados que fue realizada por un correspondiente permutador en un transmisor, habiendo realizado el correspondiente permutador un entrelazamiento por paridad de los bits de datos codificados por LDPC obtenidos mediante la realización de una codificación LDPC de acuerdo con una matriz de comprobación de paridad de un código LDPC, incluyendo la matriz de comprobación de paridad una matriz de paridad que

50 corresponde a bits de paridad del código LDPC, teniendo la matriz de paridad una estructura escalonada, de modo

que un bit de paridad de los bits de datos codificados por LDPC se entrelaza a una posición de bit de paridad diferente, y después la realización de un proceso de permutación sobre los bits de datos codificados por LDPC para permutar los bits de código de los bits de datos codificados por LDPC de modo que no se incorporan en el mismo símbolo una pluralidad de los bits de datos codificados por LDPC del código LDPC que corresponden a un valor de 1 en una fila arbitraria de una matriz de información que corresponde a bits de información de los bits codificados por LDPC, y en el que el decodificador LDPC (56) realiza una decodificación LDPC del código LDPC, sobre el que se ha realizado el proceso de permutación inversa y no se ha realizado un desentrelazamiento por paridad correspondiente al entrelazamiento por paridad, usando una matriz de comprobación de paridad convertida obtenida mediante la realización de al menos permutación de columnas correspondiente al entrelazamiento por paridad en la matriz de comprobación de paridad.

3.- Un aparato de procesamiento de datos según la reivindicación 1 ó 2, en el que el símbolo de OFMD incluye sub-portadoras piloto que están dispuestas para portar símbolos conocidos, y la dirección válida máxima predeterminada depende de un número de símbolos de sub-portadora piloto presentes en el símbolo de OFMD.

4.- Un aparato de procesamiento de datos según la reivindicación 1, 2 ó 3, en el que la memoria (100) de entrelazador de símbolos está dispuesta para efectuar el mapeo de los símbolos de datos recibidos desde las señales de sub-portadora hasta la corriente de símbolos de salida para símbolos OFDM pares mediante la introducción por lectura de los símbolos de datos de acuerdo con un orden secuencial y la extracción por lectura de los símbolos de datos desde la memoria (100) de entrelazador de símbolos de acuerdo con el conjunto de direcciones generado por el generador (102) de direcciones, y para símbolos OFDM impares mediante la introducción por lectura de los símbolos de datos adentro de la memoria (100) de entrelazador de símbolos de acuerdo con el conjunto de direcciones generado por el generador (102) de direcciones y la extracción por lectura de los símbolos de datos desde la memoria de entrelazador de símbolos de acuerdo con un orden secuencial.

5.- Un aparato de procesamiento de datos según cualquier reivindicación anterior, en el que el circuito de permutación (210) está dispuesto para cambiar el código de permutación a uno o más códigos de permutación diferentes, los cuales permulan el orden de los bits de los niveles de registro para formar el conjunto de direcciones de un símbolo de OFDM a otro, y donde uno de entre el o los códigos de permutación diferentes es:

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

6.- Un receptor para recuperar bits de datos desde un número predeterminado de señales de sub-portadora de un símbolo Multiplexado por División de Frecuencias Ortogonales para formar una corriente de bits de salida, comprendiendo el receptor:

30 un desentrelazador (514) de símbolos que puede funcionar para introducir por lectura en una memoria (100) de entrelazador de símbolos el número predeterminado de símbolos de datos desde las señales de sub-portadora OFDM, y para extraer por lectura de la memoria (100) de entrelazador de símbolos los símbolos de datos adentro de una corriente de símbolos de salida para efectuar el mapeo, siendo la extracción por lectura en un orden diferente a la introducción por lectura, determinándose el orden a partir de un conjunto de direcciones, con el efecto de que los símbolos de datos son desentrelazados desde las señales de sub-portadora OFDM adentro de la corriente de símbolos de salida,

40 una unidad (52) de desmapeo que se puede hacer funcionar para generar, a partir de los símbolos de datos de la corriente de símbolos de salida, bits de datos codificados por LDPC, entrelazados por paridad, mediante la conversión de cada uno de los símbolos de datos de la corriente de símbolos de salida representado por un símbolo de modulación de las señales de sub-portadora OFDM en bits de datos de acuerdo con un esquema de modulación utilizado,

45 un permutador inverso (53) adaptado para realizar un proceso de permutación inversa para efectuar una reversión de un proceso de permutación aplicado a los bits de datos codificados por LDPC, entrelazados por paridad, para permutar los bits de datos codificados por LDPC de manera que no se incorporan en el mismo símbolo una pluralidad de bits de códigos de los bits de datos codificados por LDPC que corresponden a un valor de 1 en una fila arbitraria de una matriz de información que corresponde a bits de información de un código LDPC utilizado para codificar los bits de datos codificados por LDPC; y

un decodificador LDPC (56) que realiza una decodificación LDPC sobre los bits de datos codificados por LDPC sobre los que se ha realizado el proceso de permutación inversa para formar los bits de datos de salida;

50 en el que el desentrelazador (514) de símbolos incluye:

un generador (102) de direcciones que se puede hacer funcionar para generar el conjunto de direcciones, generándose una dirección para cada uno de los símbolos de datos recibido para indicar la señal de sub-portadora OFDM desde la que el símbolo de datos recibido se tiene que mapear adentro de la corriente de símbolos de salida, comprendiendo el generador (102) de direcciones:

un registro (200, 202) de desplazamiento de retroalimentación lineal, que incluye un número predeterminado de niveles de registro y que es operable para generar una secuencia de bit pseudo-aleatoria de acuerdo con un polinomio generador,

- 5 un circuito (210) de permutación operable para recibir el contenido de los niveles del registro de desplazamiento y para permutar los bits presentes en los niveles de registro de acuerdo con un código de permutación para formar una dirección de una de las sub-portadoras de OFMD, y

una unidad (224) de control operable en combinación con un circuito (216) de comprobación de dirección para re-generar una dirección cuando una dirección generada exceda de una dirección válida máxima predeterminada;

y en el que:

- 10 la dirección válida máxima predeterminada es sustancialmente cuatro mil noventa y seis,

el registro (200, 202) de desplazamiento de retroalimentación lineal tiene once niveles de registro con un polinomio generador para el registro de desplazamiento de retroalimentación lineal de $R'_i[10]=R'_{i-1}[0]\oplus R'_{i-1}[2]$, y el orden de permutación forma una dirección de once bits $R_i[n]$ para el símbolo de datos $i^{\text{ésimo}}$ del bit presente en el nivel de registro $n^{\text{ésimo}}$ de acuerdo con la tabla:

$R'_i[n]$ 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

- 15 7.- Un receptor según la reivindicación 6, en el que el circuito de permutación (210) está dispuesto para cambiar el código de permutación a uno o más códigos de permutación diferentes, los cuales permutan el orden de los bits de los niveles de registro para formar el conjunto de direcciones de un símbolo de OFDM a otro, y donde uno de entre el o los códigos de permutación diferentes es:

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

- 20 8.- Un receptor según la reivindicación 6 ó 7, en el que los bits de datos han sido modulados en los símbolos OFMD de acuerdo con una norma de Difusión de Vídeo Digital tal como la de Difusión de Vídeo Digital - Terrestre, la de Difusión de Vídeo Digital - Portátil, la norma de Difusión de Vídeo Digital - Terrestre 2 o la norma de Difusión de Vídeo Digital - Cable 2.

- 25 9.- Un método de recuperación de bits de datos desde símbolos de datos recibidos desde un número predeterminado de señales de sub-portadora de un símbolo Multiplexado por División de Frecuencias Ortogonales, OFDM, para formar una corriente de bits de salida, comprendiendo el método:

introducir por lectura en una memoria (100) de entrelazador de símbolos el número predeterminado de símbolos de datos desde las señales de sub-portadora OFDM,

- 30 extraer por lectura de la memoria (100) de entrelazador de símbolos los símbolos de datos adentro de una corriente de símbolos de salida para efectuar un desentrelazamiento de los símbolos de datos desde las señales de sub-portadora del símbolo OFDM, siendo la extracción por lectura en un orden diferente a la introducción por lectura, determinándose el orden a partir de un conjunto de direcciones, con el efecto de que los símbolos de datos son desentrelazados desde las señales de sub-portadora OFDM adentro de la corriente de símbolos de salida,

- 35 generar, a partir de los símbolos de datos de la corriente de símbolos de salida, bits de datos codificados por LDPC, entrelazados por paridad, mediante la conversión de cada símbolo de datos representado por un símbolo de modulación de las señales de sub-portadora OFDM en bits de datos codificados entrelazados de acuerdo con un esquema de modulación,

- 40 realizar un proceso de permutación inversa para efectuar una reversión de un proceso de permutación aplicado a los bits de datos codificados por LDPC, entrelazados por paridad, para permutar los bits de datos codificados por LDPC de manera que no se incorporan en el mismo símbolo una pluralidad de los bits de datos codificados por LDPC que corresponden a un valor de 1 en una fila arbitraria de una matriz de información que corresponde a bits de información de un código LDPC, y

- 45 realizar una decodificación LDPC sobre los bits de datos codificados por LDPC sobre los que se ha realizado el proceso de permutación inversa para formar los bits de datos de salida; en el que la introducción por lectura en la memoria de entrelazador de símbolos y la extracción por lectura de la memoria (100) de entrelazador de símbolos incluye:

generar el conjunto de direcciones, generándose una dirección para cada uno de los símbolos de datos recibido

para indicar la señal de sub-portadora OFDM desde la que el símbolo de datos recibido se tiene que mapear adentro de la corriente de símbolos de salida, comprendiendo la generación del conjunto de direcciones:

utilización de un registro (200, 202) de desplazamiento de retroalimentación lineal que incluye un número predeterminado de niveles de registro para generar una secuencia de bit pseudo-aleatoria de acuerdo con un polinomio generador,

utilización de un circuito de permutación para recibir el contenido de los niveles de registro y permutar los bits presentes en los niveles de registro de acuerdo con un código de permutación para formar una dirección, y

re-generación de una dirección cuando una dirección generada exceda una dirección válida máxima predeterminada, siendo sustancialmente la dirección válida máxima predeterminada cuatro mil noventa y seis,

el registro (200, 202) de desplazamiento de retroalimentación lineal teniendo once niveles de registro con un polinomio generador para el registro de desplazamiento de retroalimentación lineal de $R'_i[10] = R'_{i-1}[0] \oplus R'_{i-1}[2]$, y el orden de permutación formando una dirección de once bits $R'_i[n]$ para el símbolo de datos $i^{\text{ésimo}}$ del bit presente en el nivel de registro $n^{\text{ésimo}}$ $R'_i[u]$ de acuerdo con la tabla:

$R'_i[n]$ 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

10.- Un método según la reivindicación 9, en el que la realización del proceso de permutación inversa sobre los datos codificados por LDPC entrelazados tiene por efecto revertir una permutación de los bits codificados por LDPC, entrelazados por paridad, que fueron obtenidos realizando una codificación LDPC de acuerdo con una matriz de comprobación de paridad del código LDPC, incluyendo la matriz de comprobación de paridad una matriz de paridad que corresponde a bits de paridad de un código LDPC, teniendo la matriz de paridad una estructura escalonada, de modo que un bit de paridad del código LDPC se entrelaza a una posición de bit de paridad diferente, y realizando después un proceso de permutación sobre los bits de datos codificados por LDPC para permutar bits de código de los bits de datos codificados por LDPC de modo que no se incorporan en el mismo símbolo una pluralidad de bits de código de los bits de datos codificados por LDPC que corresponden a un valor de 1 en una fila arbitraria de una matriz de información que corresponde a bits de información del código LDPC, y en el que se realiza la decodificación LDPC de los bits codificados por LDPC, sobre los que se ha realizado el proceso de permutación inversa y no se ha realizado un desentrelazamiento por paridad correspondiente al entrelazamiento por paridad, usando una matriz de comprobación de paridad convertida obtenida mediante la realización de al menos permutación de columnas correspondiente al entrelazamiento por paridad en la matriz de comprobación de paridad.

20 11.- Un método según la reivindicación 9 ó 10, en el que el símbolo de OFDM incluye sub-portadoras piloto, las cuales están dispuestas para portar símbolos conocidos, y la dirección válida máxima predeterminada depende del número de símbolos de sub-portadora piloto presentes en el símbolo de OFDM.

25 12.- Un método según la reivindicación 9, 10 u 11, en el que la introducción por lectura y la extracción por lectura de los símbolos de datos en y desde la memoria (100) de entrelazador de símbolos incluye:

para símbolos OFDM pares, la introducción por lectura de los símbolos de datos de acuerdo con un orden secuencial y la extracción por lectura de los símbolos de datos desde la memoria de entrelazador de símbolos de acuerdo con el conjunto de direcciones generado por el generador de direcciones, y

30 para símbolos OFDM impares, la introducción por lectura de los símbolos de datos adentro de la memoria de entrelazador de símbolos de acuerdo con el conjunto de direcciones generado por el generador de direcciones y la extracción por lectura de los símbolos de datos desde la memoria de entrelazador de símbolos de acuerdo con un orden secuencial.

35 40 13.- Un método según cualquiera de las reivindicaciones 9-12, en el que la utilización del circuito de permutación incluye cambiar el código de permutación a uno o más códigos de permutación diferentes, los cuales permutan el orden de los bits de los niveles de registro para formar el conjunto de direcciones de un símbolo de OFDM a otro, y donde uno de entre el o los códigos de permutación diferentes es:

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

45 14. Un método para recibir bits de datos desde un número predeterminado de señales de sub-portadora de un símbolo Multiplexado por División de Frecuencias Ortogonales y formar una corriente de bits de salida, comprendiendo el método

introducir por lectura en una memoria (100) de entrelazador de símbolos el número predeterminado de símbolos de datos desde las señales de sub-portadora OFDM,

- extraer por lectura de la memoria (100) de entrelazador de símbolos los símbolos de datos adentro de una corriente de símbolos de salida para efectuar un desentrelazamiento de los símbolos de datos desde las señales de sub-portadora del símbolo OFDM, siendo la extracción por lectura en un orden diferente a la introducción por lectura, determinándose el orden a partir de un conjunto de direcciones, con el efecto de que los símbolos de datos son desentrelazados desde las señales de sub-portadora OFDM adentro de la corriente de símbolos de salida,
- 5 generar, a partir de los símbolos de datos de la corriente de símbolos de salida, bits de datos codificados por LDPC, entrelazados por paridad, mediante la conversión de cada símbolo de datos representado por un símbolo de modulación de las señales de sub-portadora OFDM en bits de datos codificados entrelazados de acuerdo con un esquema de modulación,
- 10 realizar un proceso de permutación inversa para efectuar una reversión de un proceso de permutación aplicado a los bits de datos codificados por LDPC, entrelazados por paridad, para permutar los bits de datos codificados por LDPC de manera que no se incorporan en el mismo símbolo una pluralidad de los bits de datos codificados por LDPC que corresponden a un valor de 1 en una fila arbitraria de una matriz de información que corresponde a bits de información de un código LDPC, y
- 15 realizar una decodificación LDPC sobre los bits de datos codificados por LDPC sobre los que se ha realizado el proceso de permutación inversa para formar los bits de datos de salida; en el que la introducción por lectura en la memoria (100) de entrelazador de símbolos y la extracción por lectura de la memoria (100) de entrelazador de símbolos incluye:
- 20 generar el conjunto de direcciones, generándose una dirección para cada uno de los símbolos de datos recibido para indicar la señal de sub-portadora OFDM desde la que el símbolo de datos recibido se tiene que mapear adentro de la corriente de símbolos de salida, comprendiendo la generación del conjunto de direcciones:
- utilización de un registro (200, 202) de desplazamiento de retroalimentación lineal que incluye un número predeterminado de niveles de registro para generar una secuencia de bit pseudo-aleatoria de acuerdo con un polinomio generador,
- 25 utilización de un circuito (210) de permutación para recibir el contenido de los niveles de registro y permutar los bits presentes en los niveles de registro de acuerdo con un código de permutación para formar una dirección, y
- re-generación de una dirección cuando una dirección generada exceda una dirección válida máxima predeterminada, siendo sustancialmente la dirección válida máxima predeterminada cuatro mil noventa y seis,
- 30 el registro (200, 202) de desplazamiento de retroalimentación lineal teniendo once niveles de registro con un polinomio generador para el registro de desplazamiento de retroalimentación lineal de $R'[10] = R'_{i-1}[0] \oplus R'_{i-1}[2]$, y el orden de permutación formando una dirección de once bits $R'[n]$ para el símbolo de datos $i^{\text{ésimo}}$ del bit presente en el nivel de registro $n^{\text{ésimo}}$ $R'[u]$ de acuerdo con la tabla:
- | | | | | | | | | | | | |
|--------------------|----|----|---|---|---|---|---|---|---|---|---|
| $R'[n]$ para $n =$ | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| $R[n]$ para $n =$ | 7 | 10 | 5 | 8 | 1 | 2 | 4 | 9 | 0 | 3 | 6 |
- 35 15.- Un método para recibir según la reivindicación 14, en el que la utilización del circuito de permutación (210) incluye cambiar el código de permutación a uno o más códigos de permutación diferentes, los cuales permutan el orden de los bits de los niveles de registro para formar el conjunto de direcciones de un símbolo de OFDM a otro, y donde uno de entre el o los códigos de permutación diferentes es:
- | | | | | | | | | | | | |
|--------------------|----|---|---|----|---|---|---|---|---|---|---|
| $R'[n]$ para $n =$ | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| $R[n]$ para $n =$ | 6 | 2 | 7 | 10 | 8 | 0 | 3 | 4 | 1 | 9 | 5 |

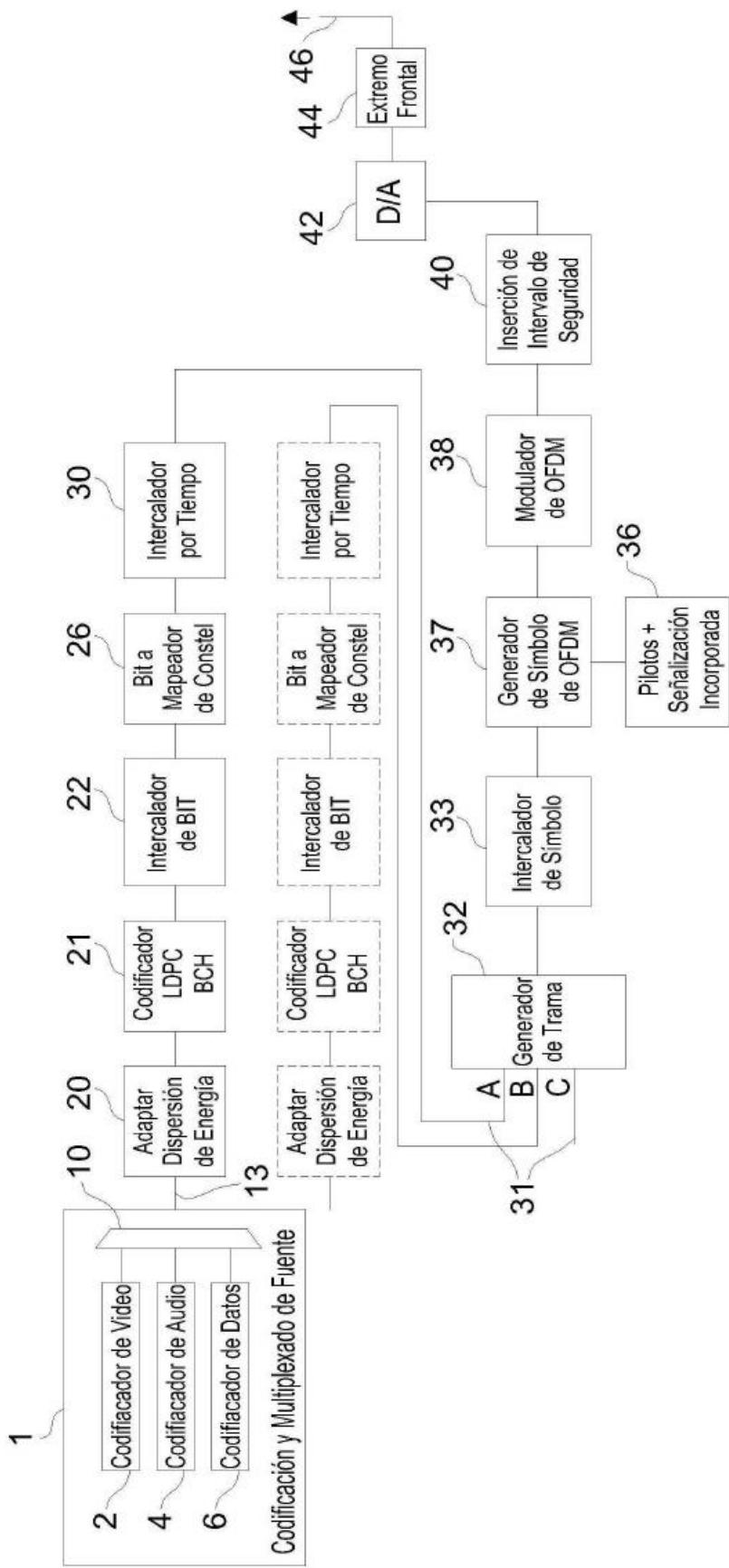


FIG. 1

FIG. 2

$$H = \begin{bmatrix} & & & & \\ & \vdots & & & \\ & 1 & \cdots & 1 & \cdots & 1 & \cdots & 1 \\ \cdots & 1 & \cdots & 1 & 1 & \cdots & 1 & \cdots & 1 \\ & \vdots & & & & & & & \\ & 1 & \cdot & & & & & & \\ & \vdots & & & & & & & \\ & 1 & & & & & & & \end{bmatrix}$$

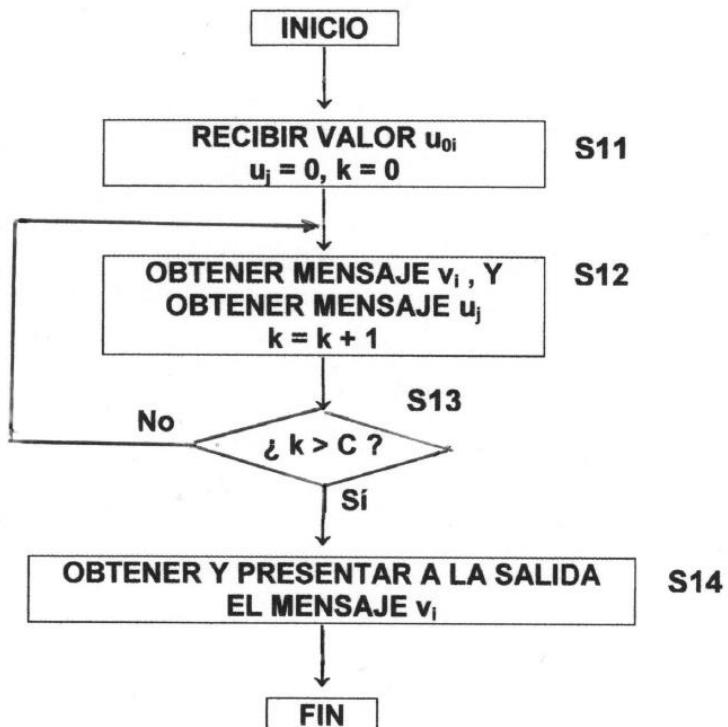
FIG. 3

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

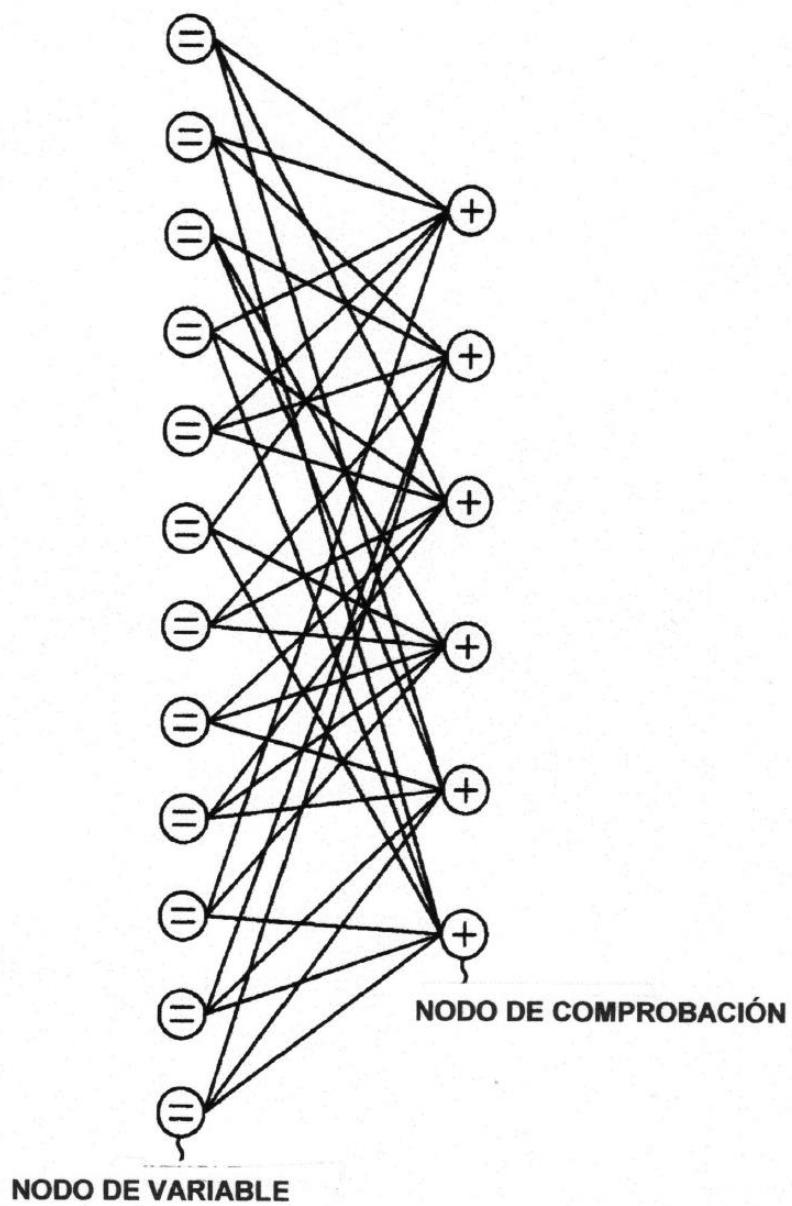


FIG. 6

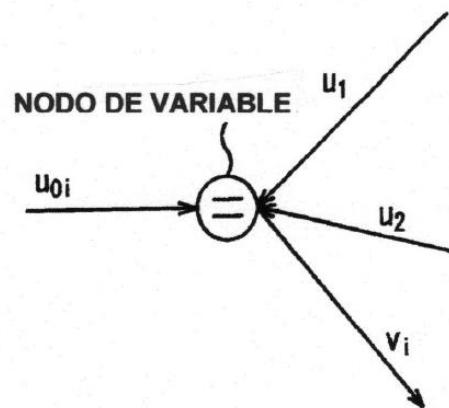


FIG. 7

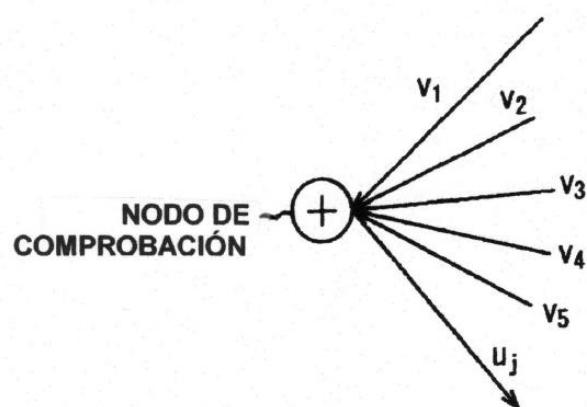


FIG. 8

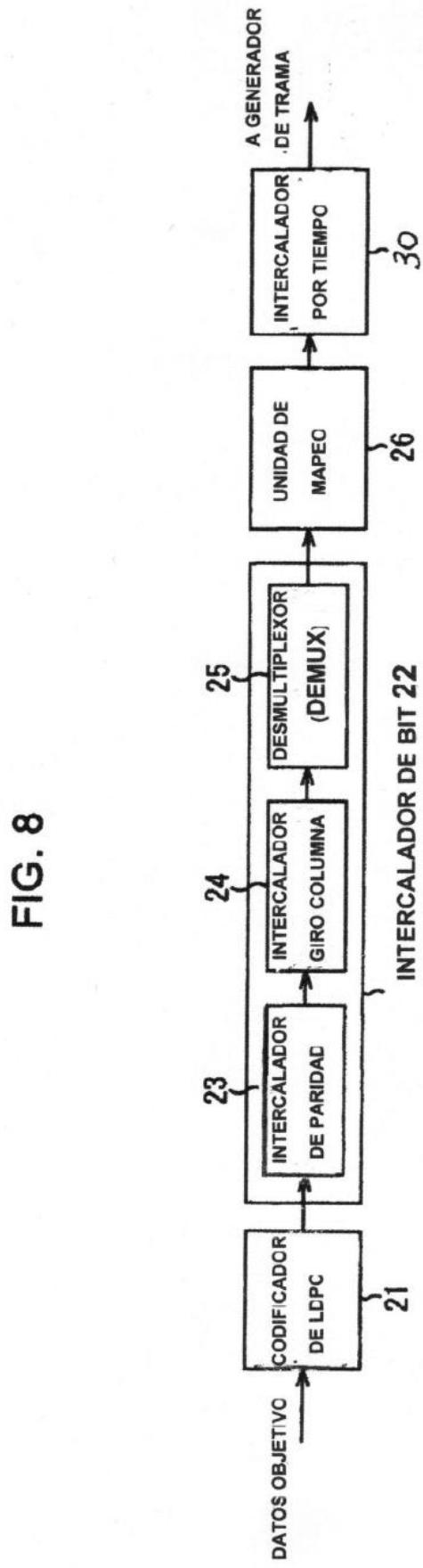


FIG. 9

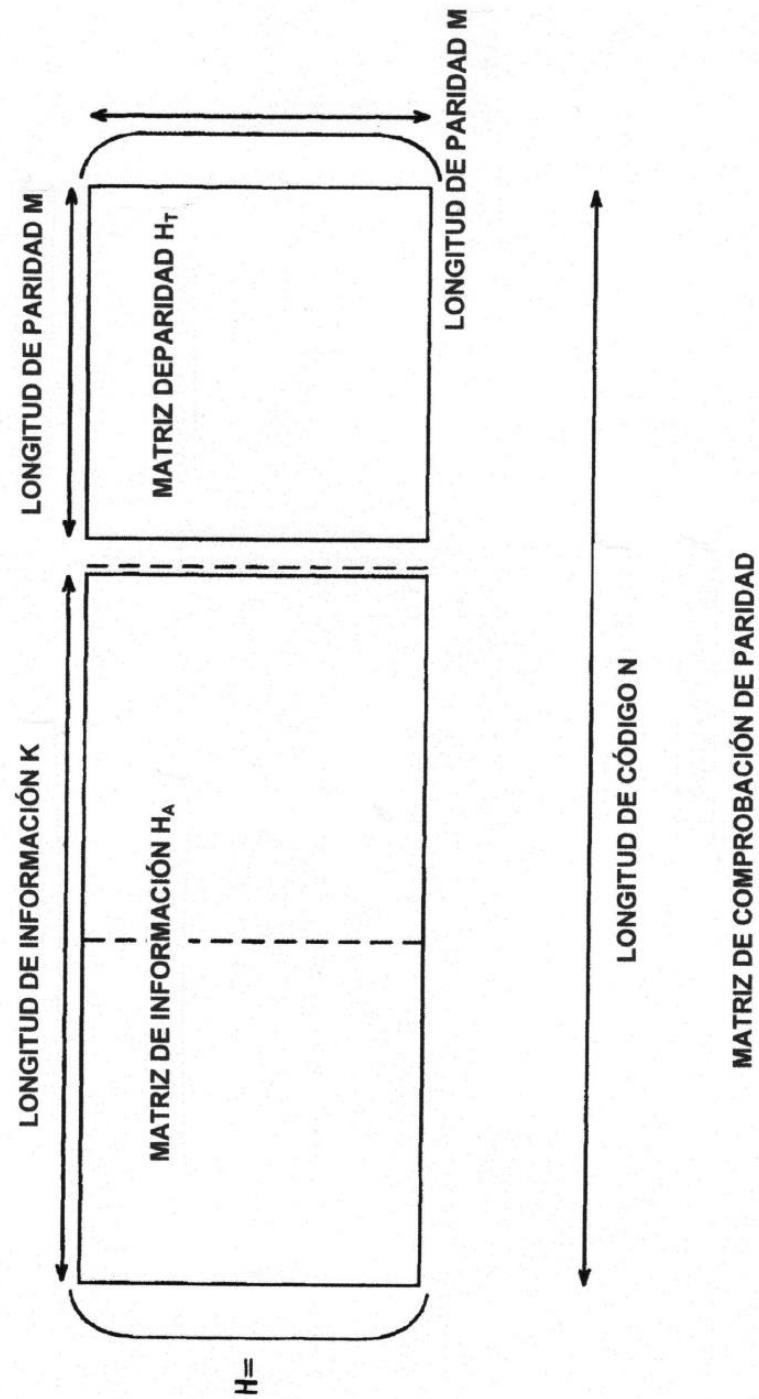
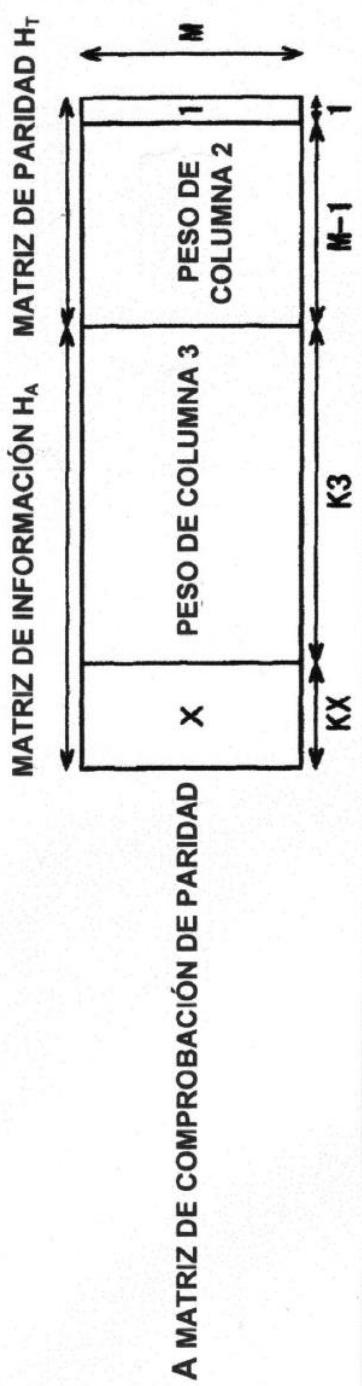


FIG. 10

$$H_T = \begin{pmatrix} 1 & & & & \\ 11 & 0 & & & \\ 11 & & 0 & & \\ & & & 11 & \\ & & & 11 & \\ & & & 11 & \end{pmatrix}$$

MATRIZ DE PARIDAD H_T

FIG. 11



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

B
NÚMERO DE COLUMNAS
DE CADA PESO
DE COLUMNAS

FIG. 12

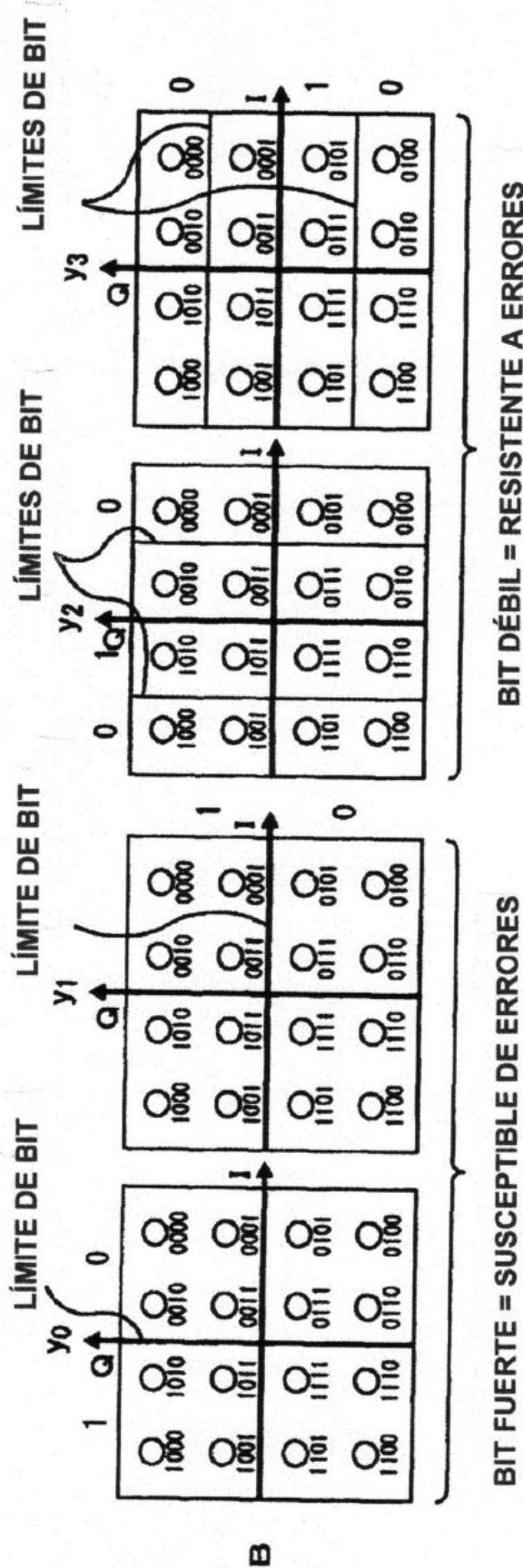
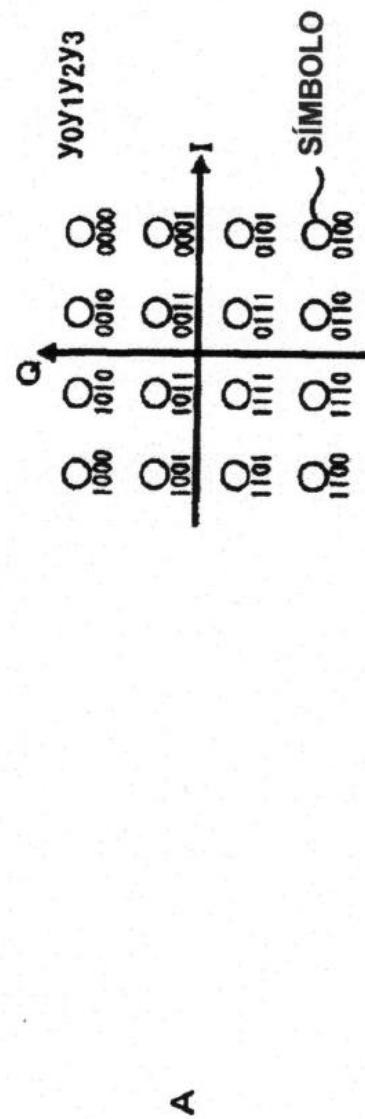


FIG. 13

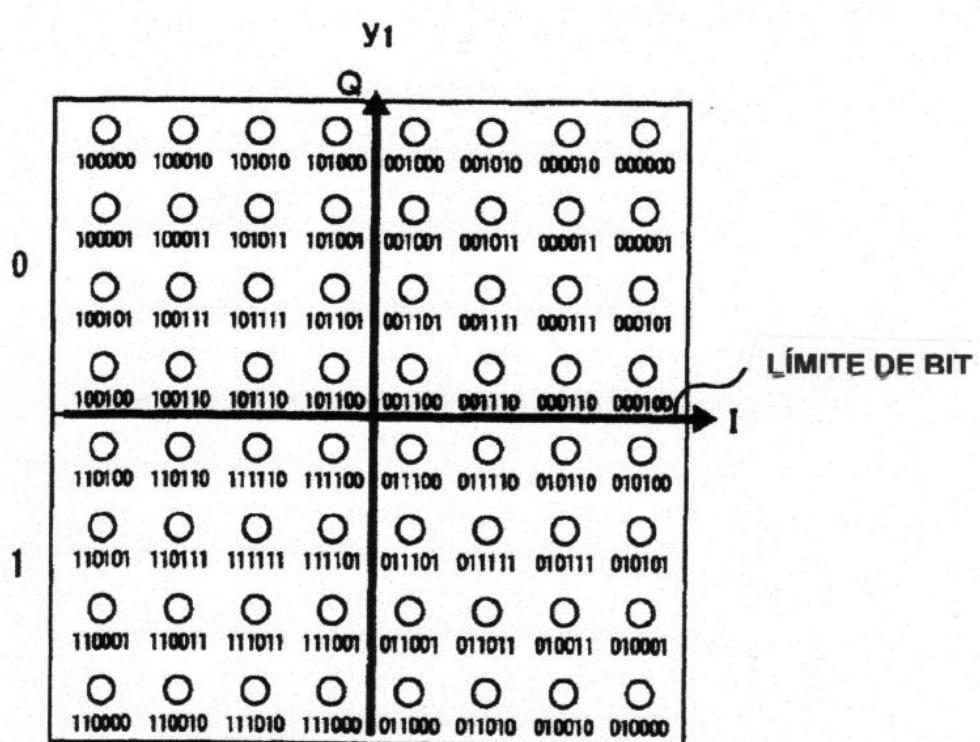
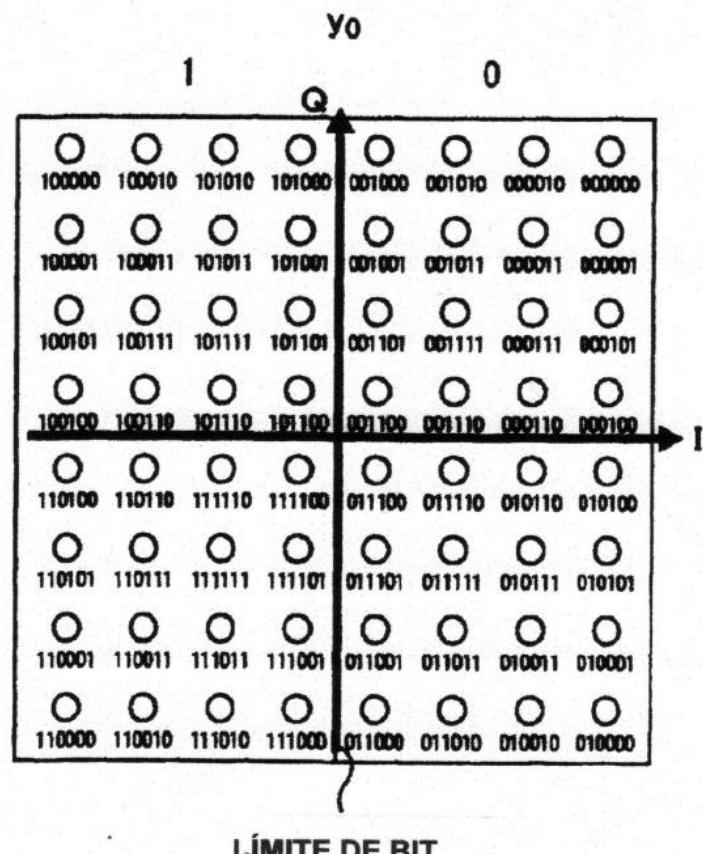


FIG. 14

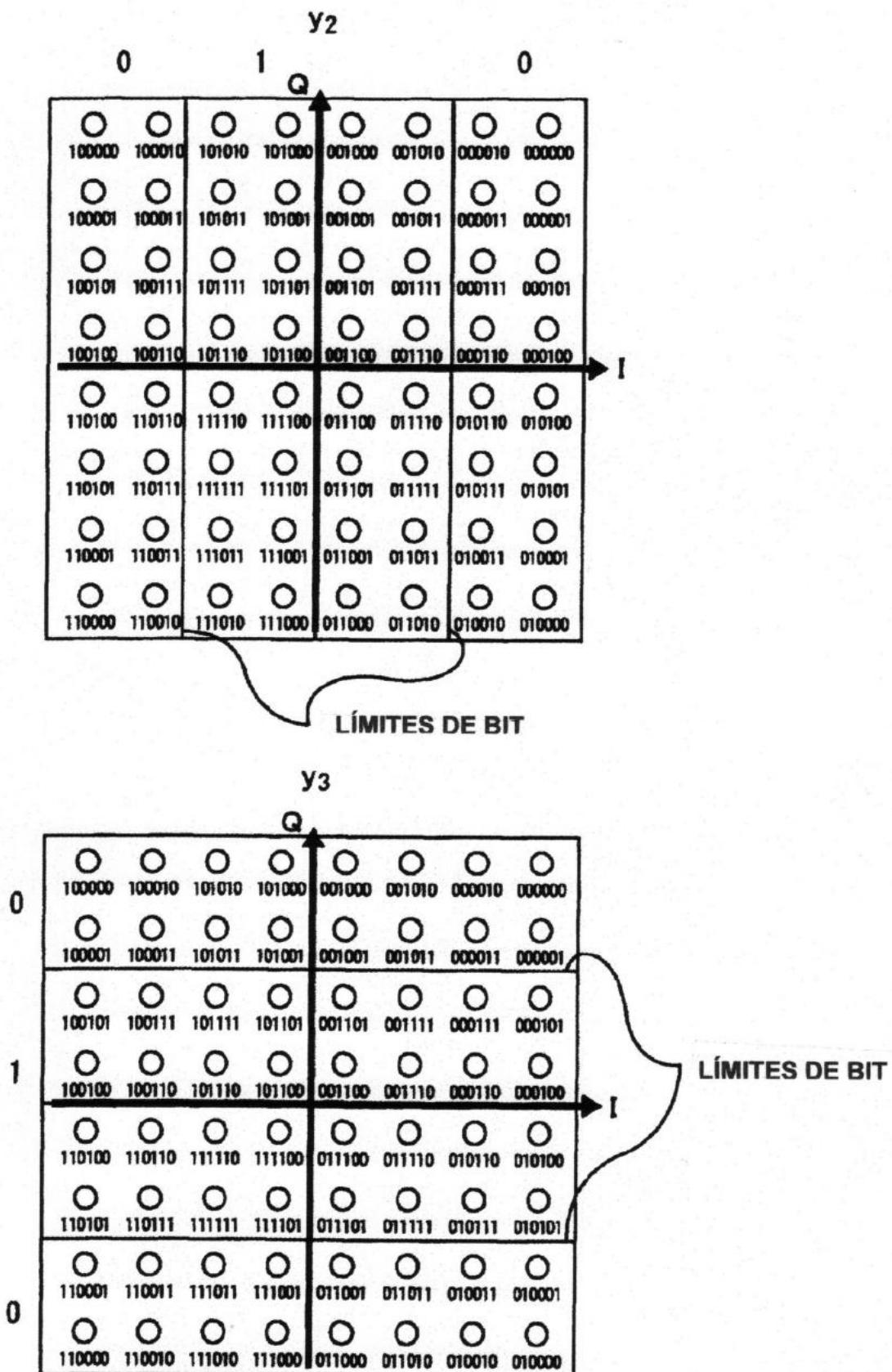


FIG. 15

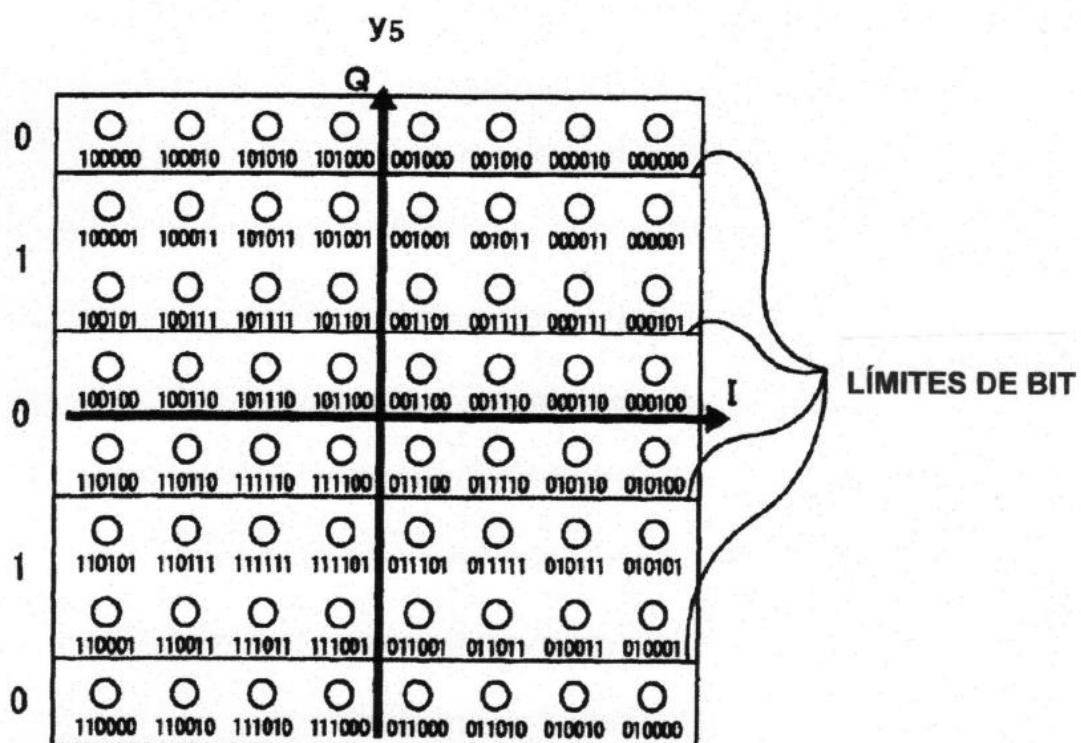
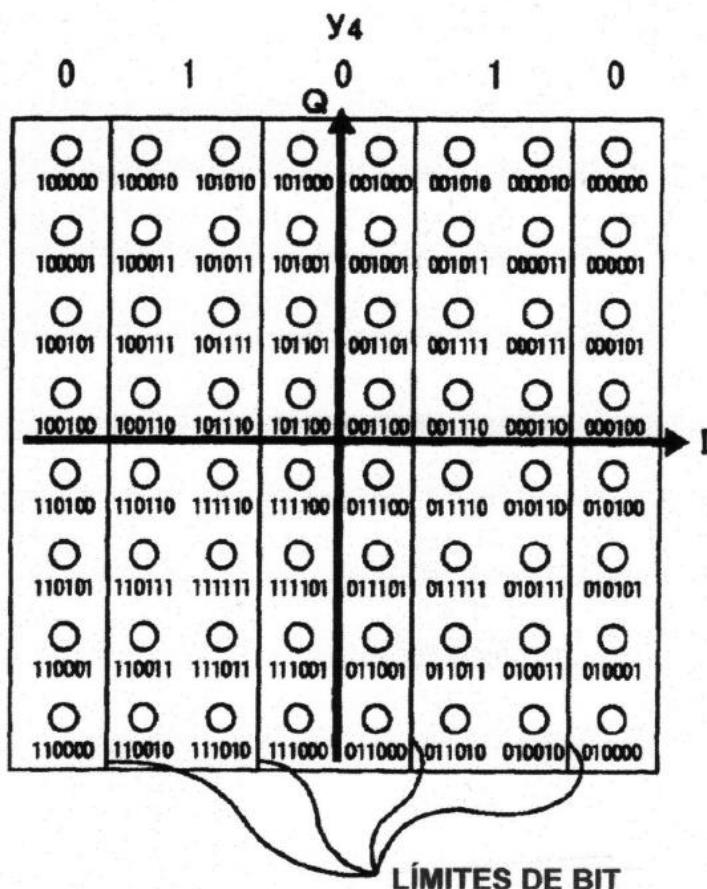


FIG. 16

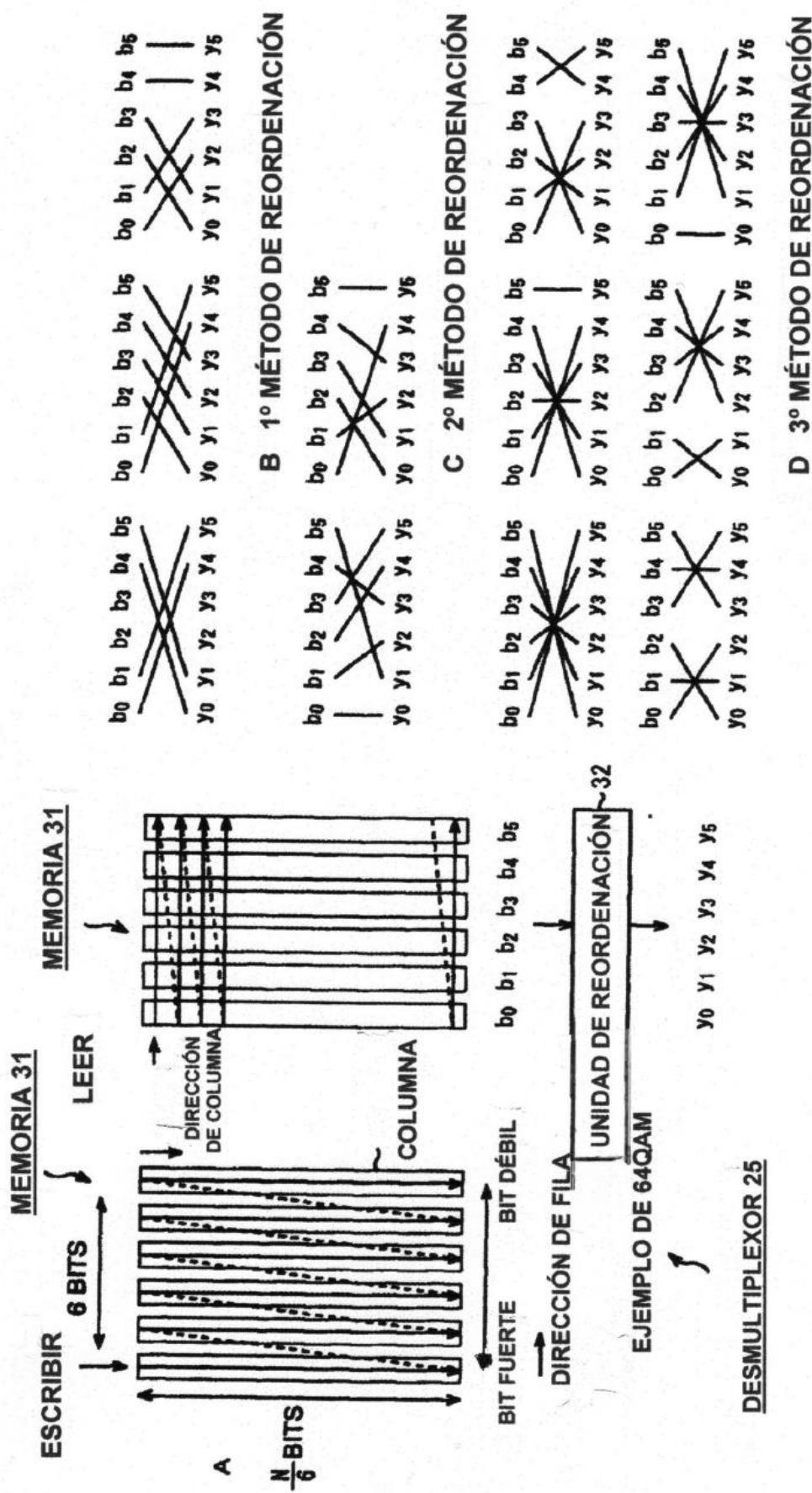


FIG. 17

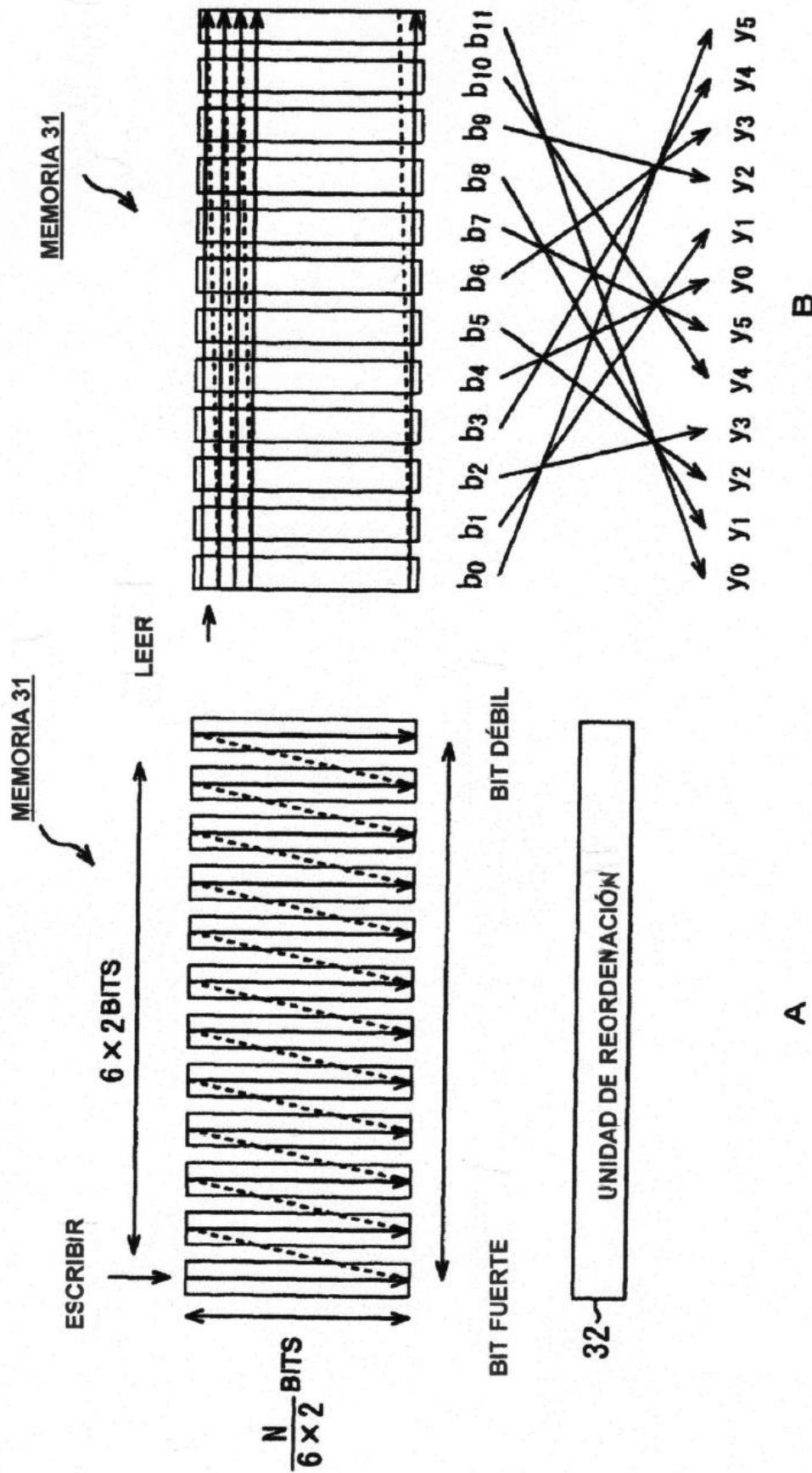


FIG. 18

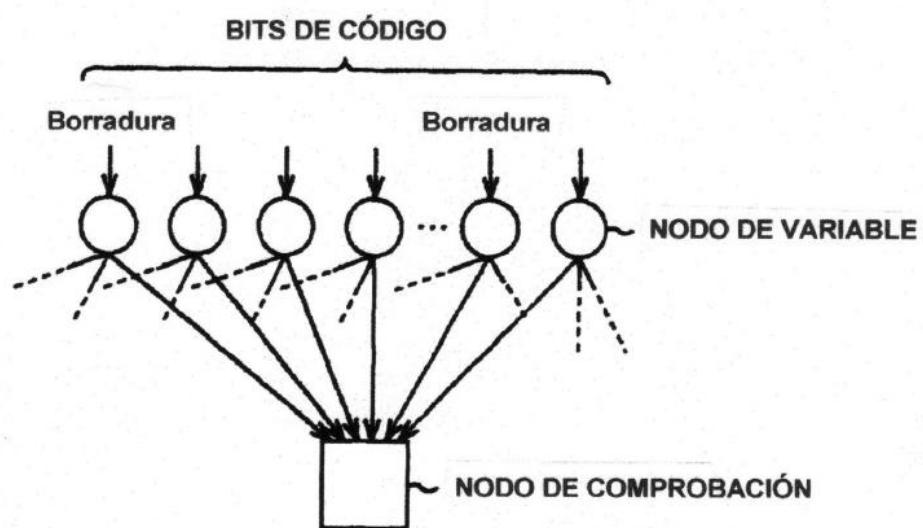
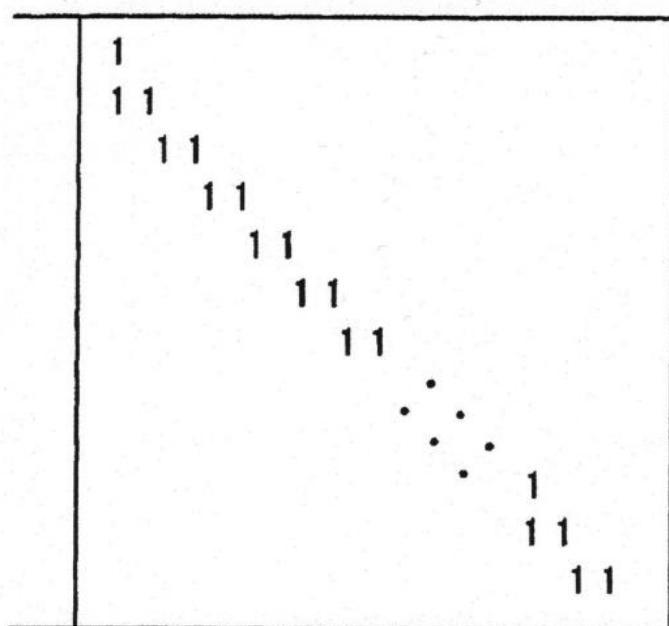
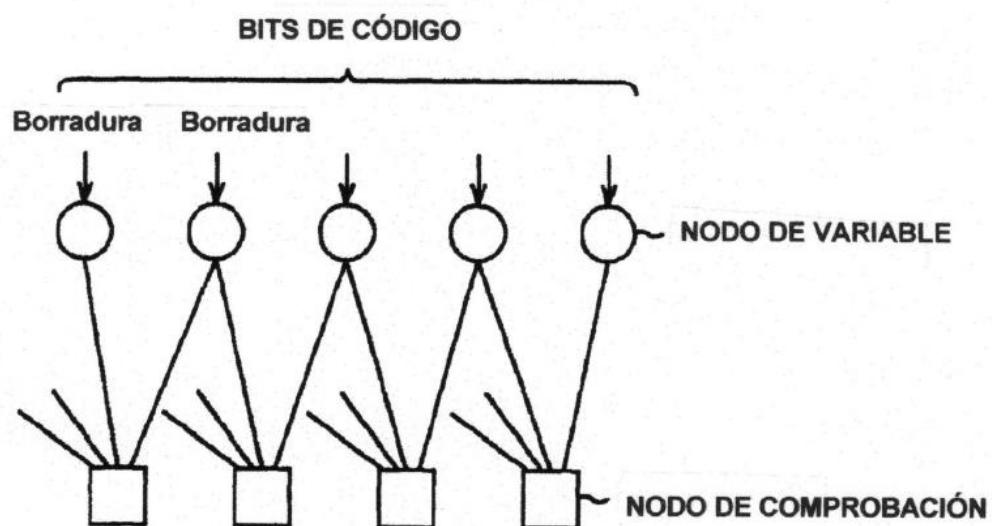


FIG. 19



ESTRUCTURA GRADUAL DE MATRIZ DE PARIDAD

A



PARTE DE GRÁFICO DE TANNER QUE TIENE ESTRUCTURA GRADUAL

B

FIG. 20

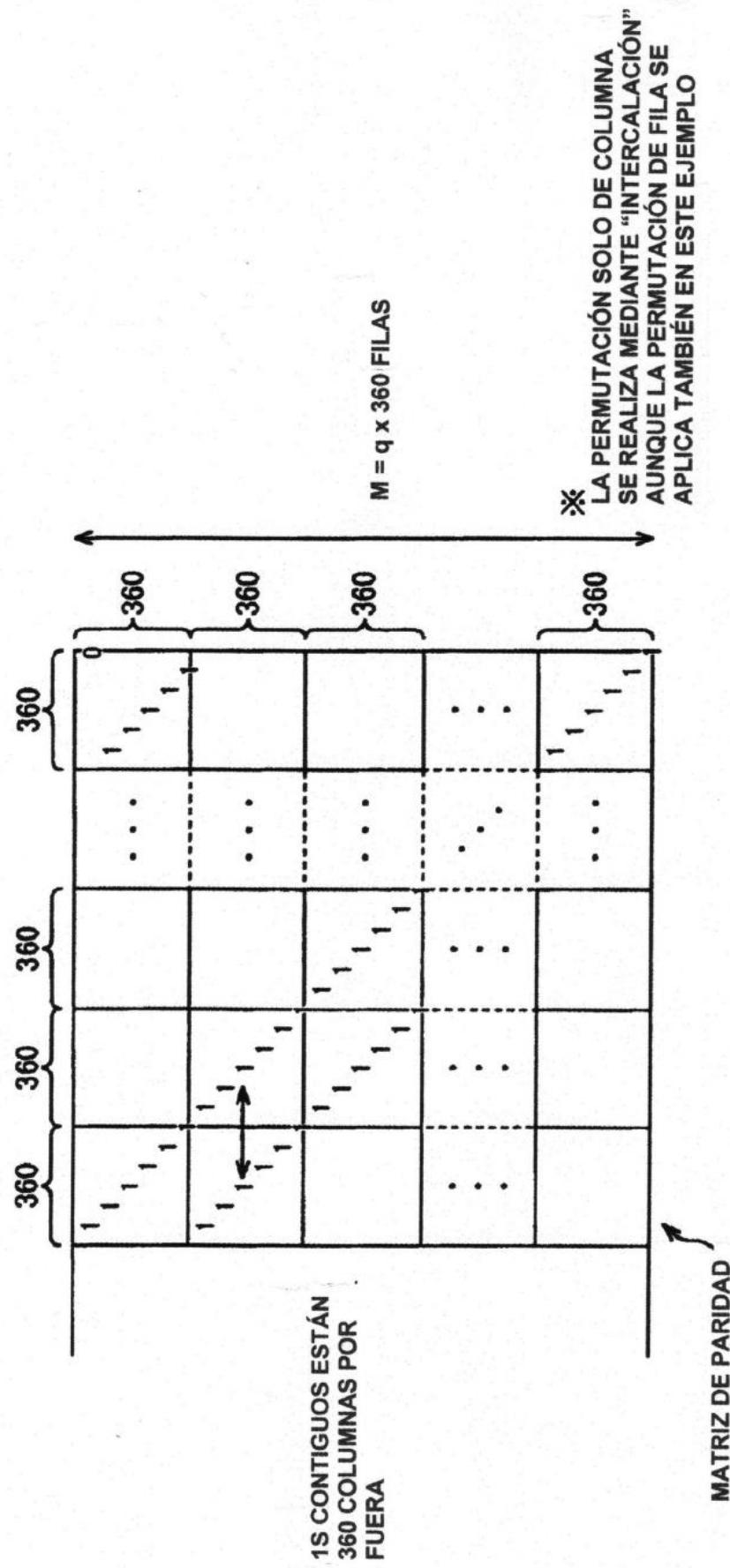
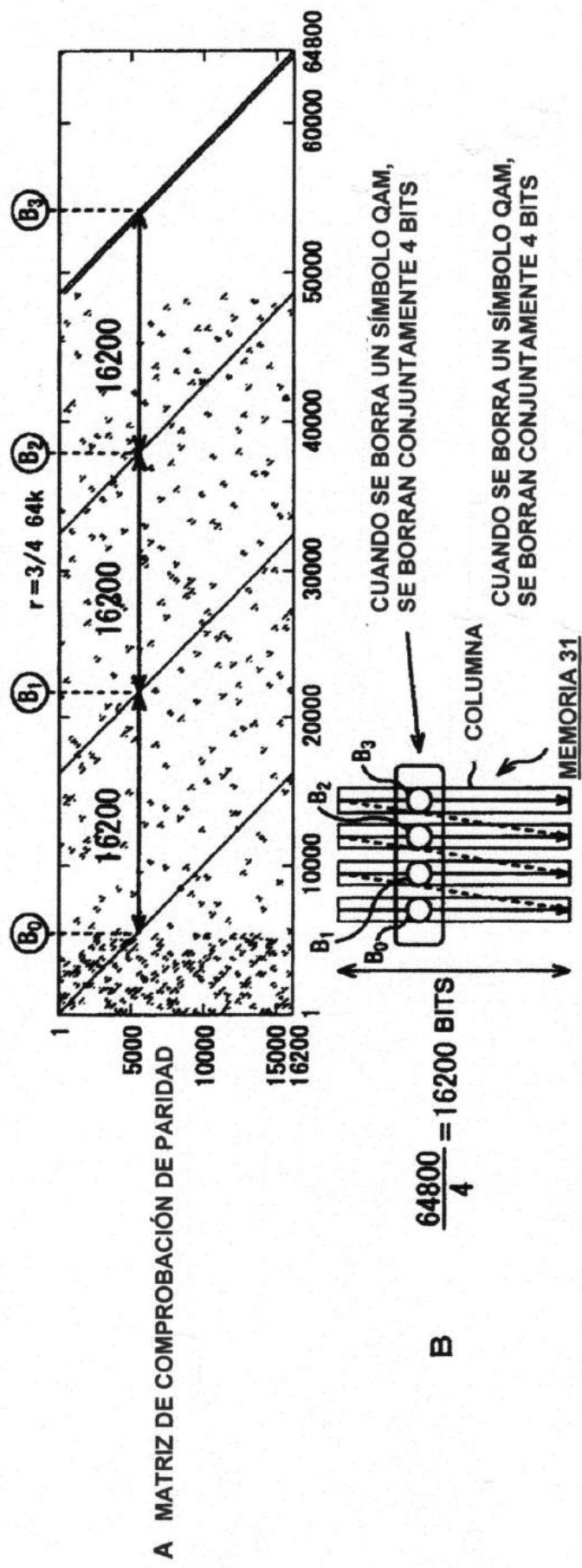


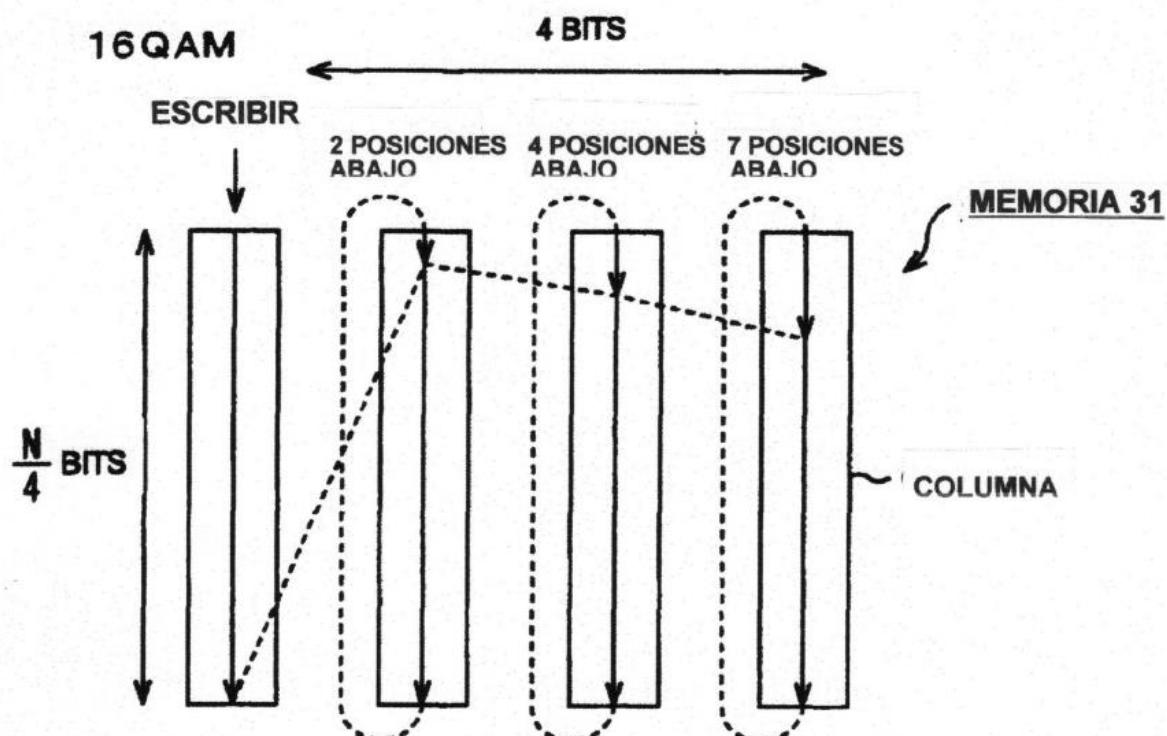
FIG. 21

16QAM, $r = 3/4$ (TRAS INTERCALACIÓN DE PARIDAD)



LA INTERCALACIÓN ES DESVENTAJOSA EN CANALES CON BORRADURAS

FIG. 22



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

FIG. 23

		POSICIONES RESPECTIVAS DE INICIO DE ESCRITURA DE mb COLUMNAS																								
Número de columnas de memoria requeridas "mb"	Primer a tercero métodos de reordenación	Cuarto método de reordenación	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	3	7	8	9												
16		256QAM	0	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	23	25	25	26	28	30					
24		4096QAM	0	5	8	8	8	10	10	10	12	13	16	17	19	21	22	23	26	37	39	40	41	41	41	

FIG. 24

		POSICIONES RESPECTIVAS DE INICIO DE ESCRITURA DE mb COLUMNAS																								
Número de columnas de memoria requeridas "mb"	Primeros a tercero métodos de ordenación	Cuarto método de ordenación	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	4	5	7														
12	4096QAM	64QAM	0	0	0	2	2	3	3	3	6	7														
20		1024QAM	0	0	0	2	2	2	2	5	5	5	7	7	7	7	7	7	7	8	8	10				
24		4096QAM	0	0	0	0	0	0	1	1	1	2	2	3	7	9	9	8	10	10	10	10	10	10	11	

FIG. 25

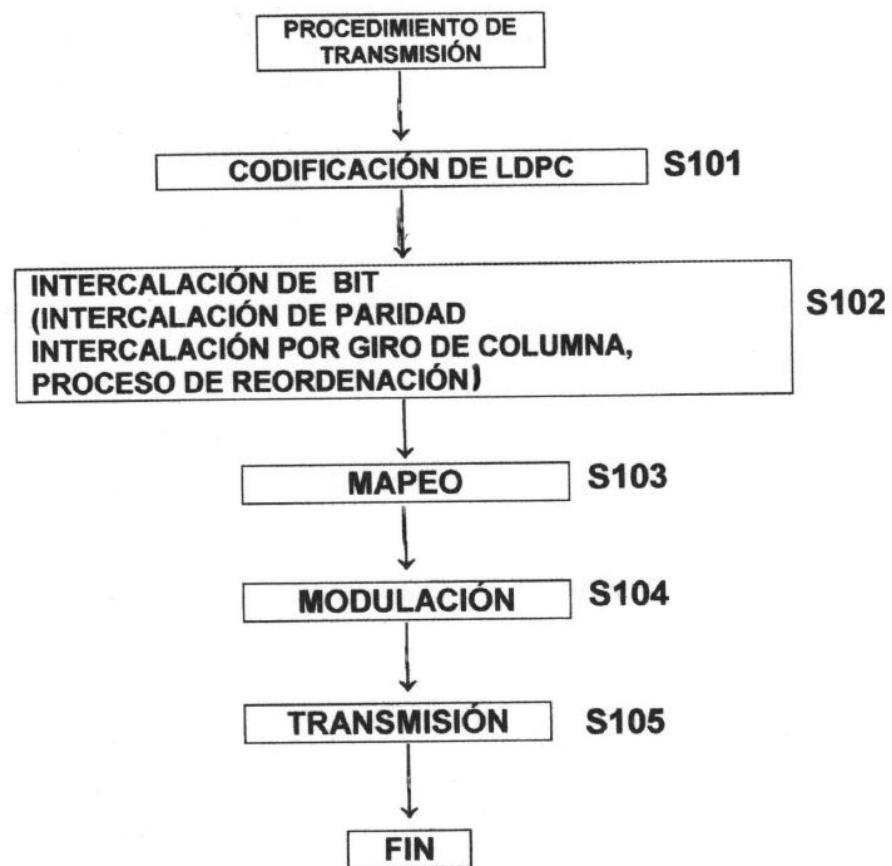
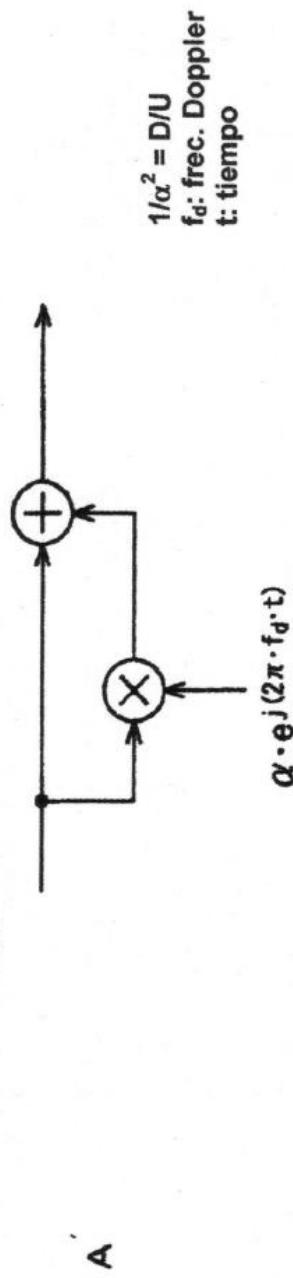
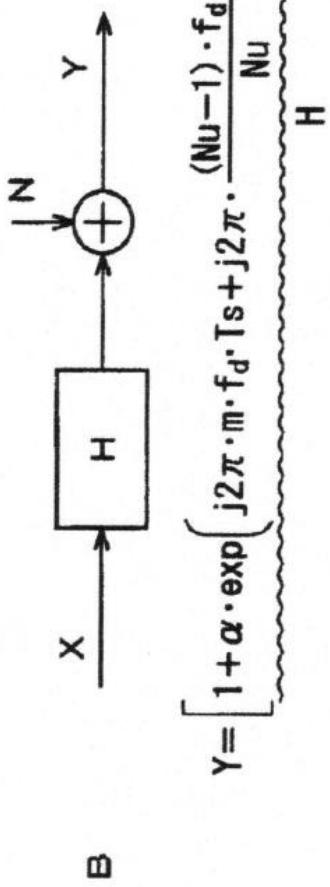


FIG. 26

MODELO DE TITILADOR REDUCIDO EQUIVALENTE



LAS SIMULACIONES SE REALIZARON UTILIZANDO UN MODELO CORRESPONDIENTE A UNA PORTADORA EXTRAIDA EN EL LADO DE RECEPCIÓN DESPUÉS DE QUE SE REALIZARA FFT SOBRE UN SÍMBOLO DE OFDM QUE FUE TRANSMITIDO A TRAVÉS DE ESTE CANAL.



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

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

POTENCIA DE |C|; APROXIMADA POR AWGN

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

FIG. 27

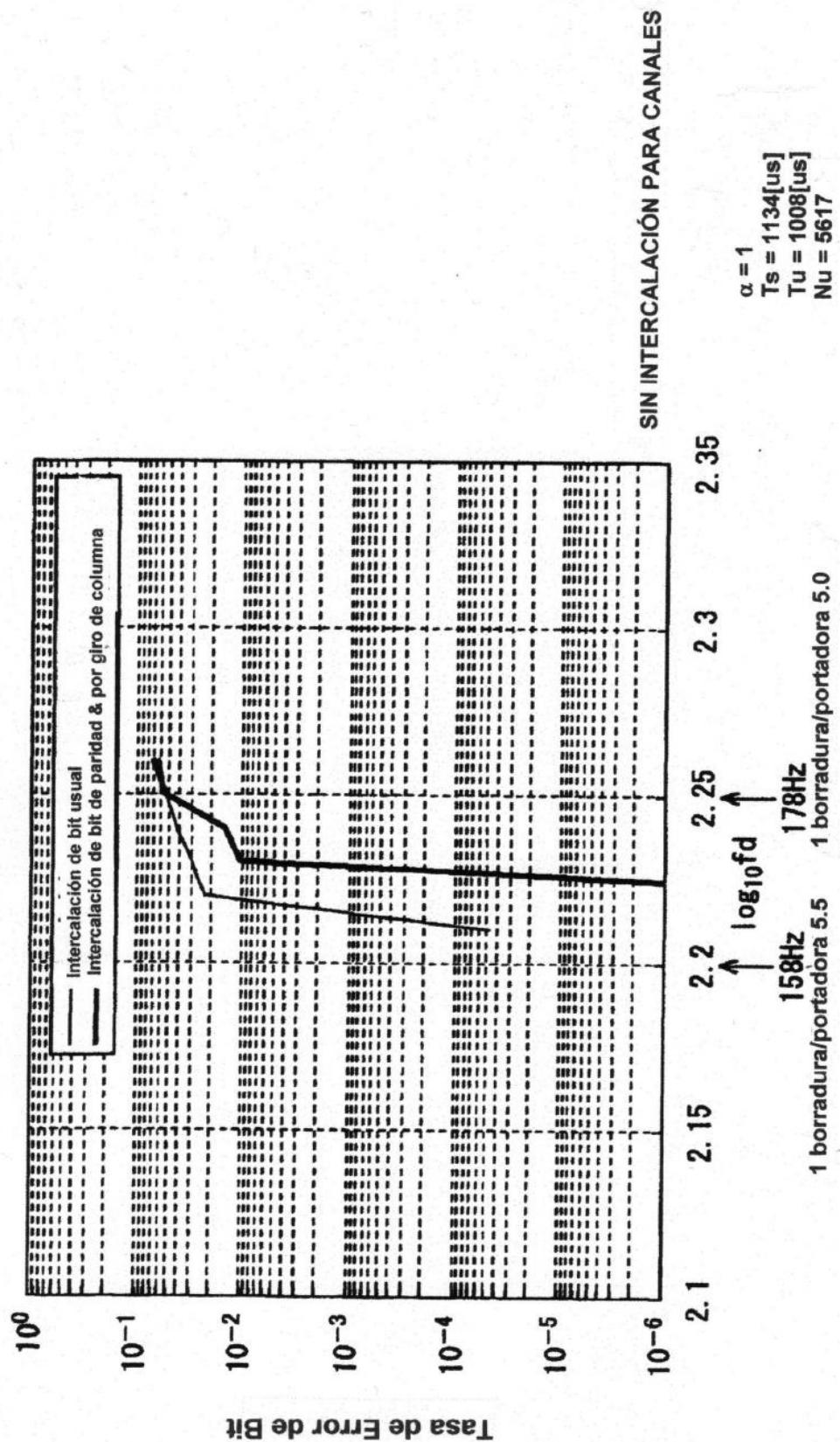
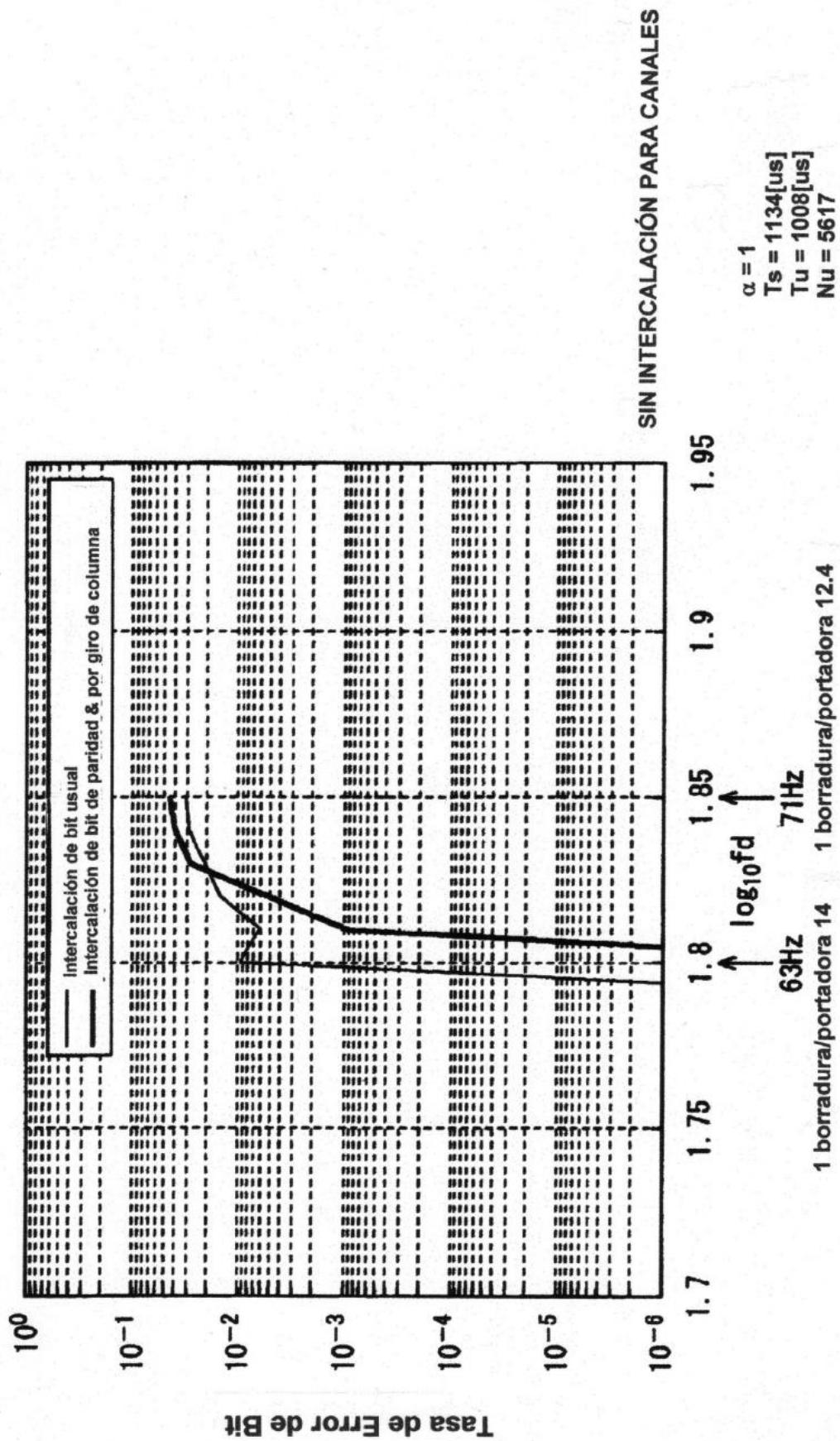


FIG. 28



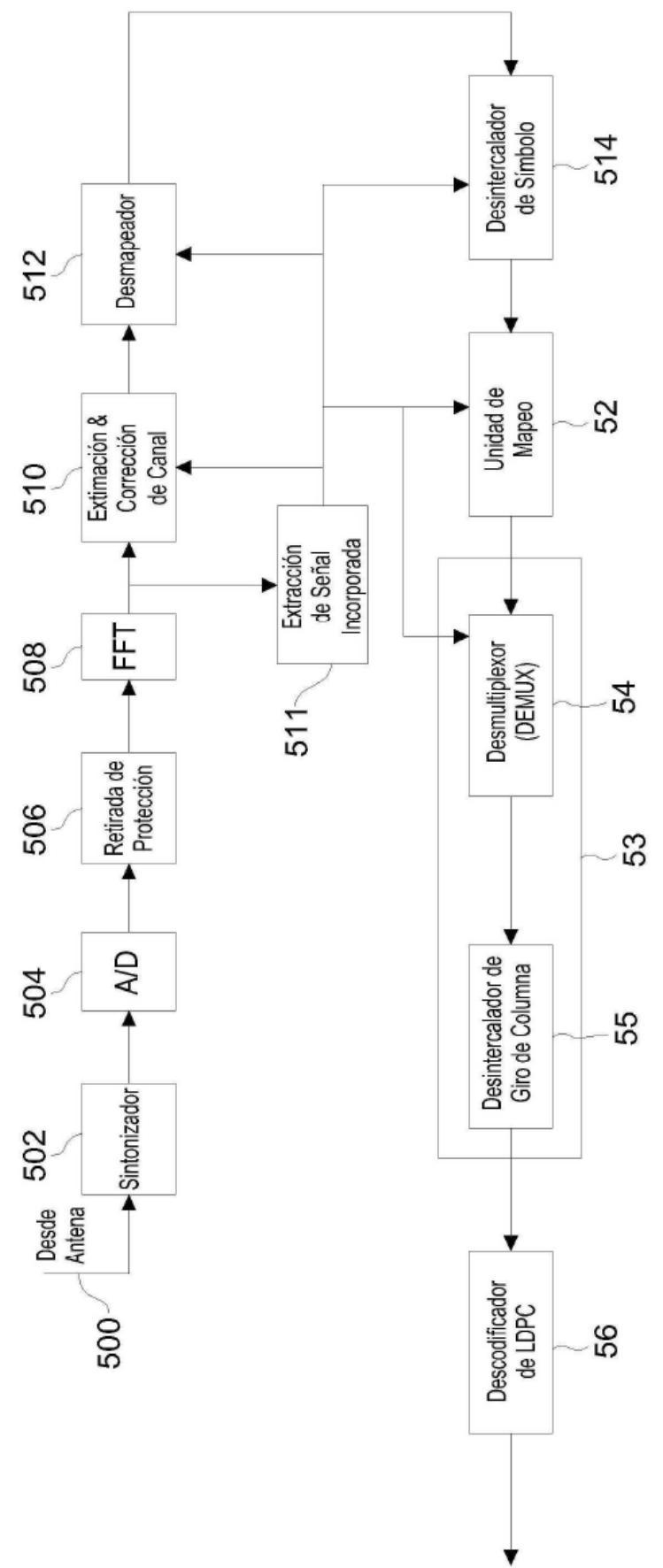


FIG. 29

FIG. 30

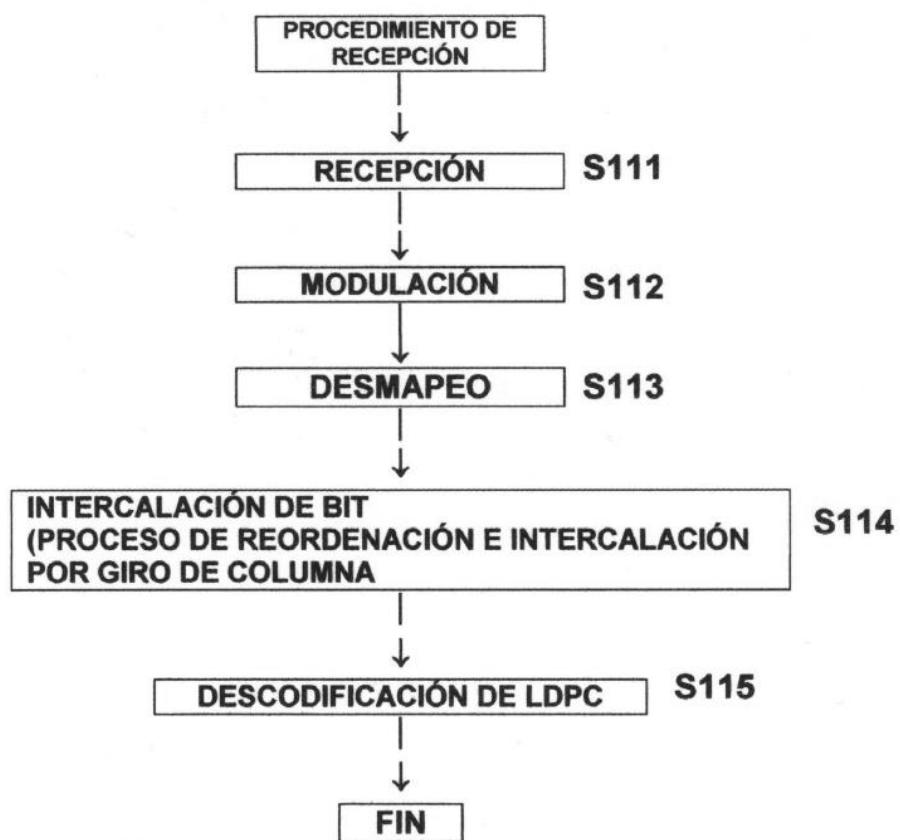


FIG. 31

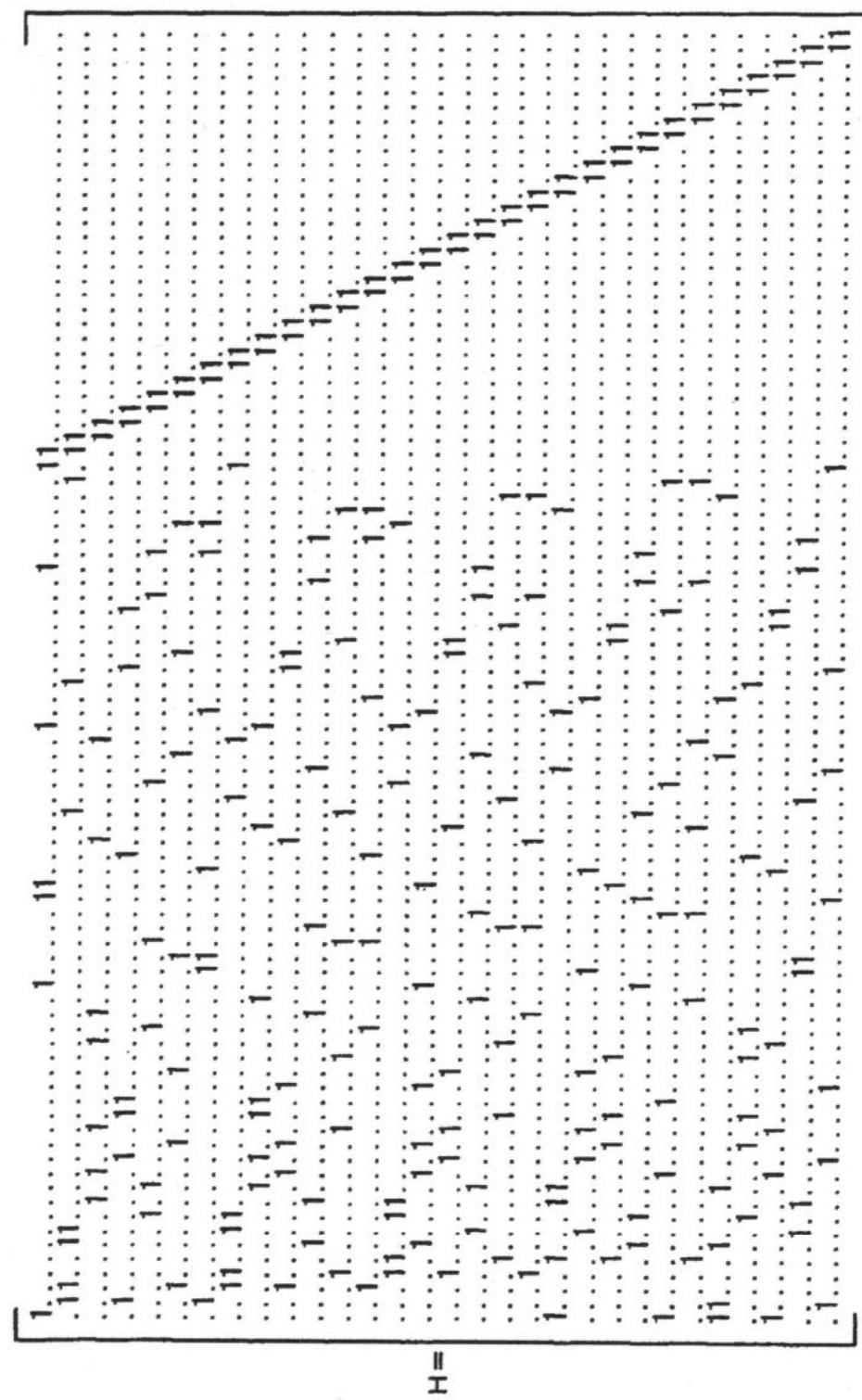


FIG. 32

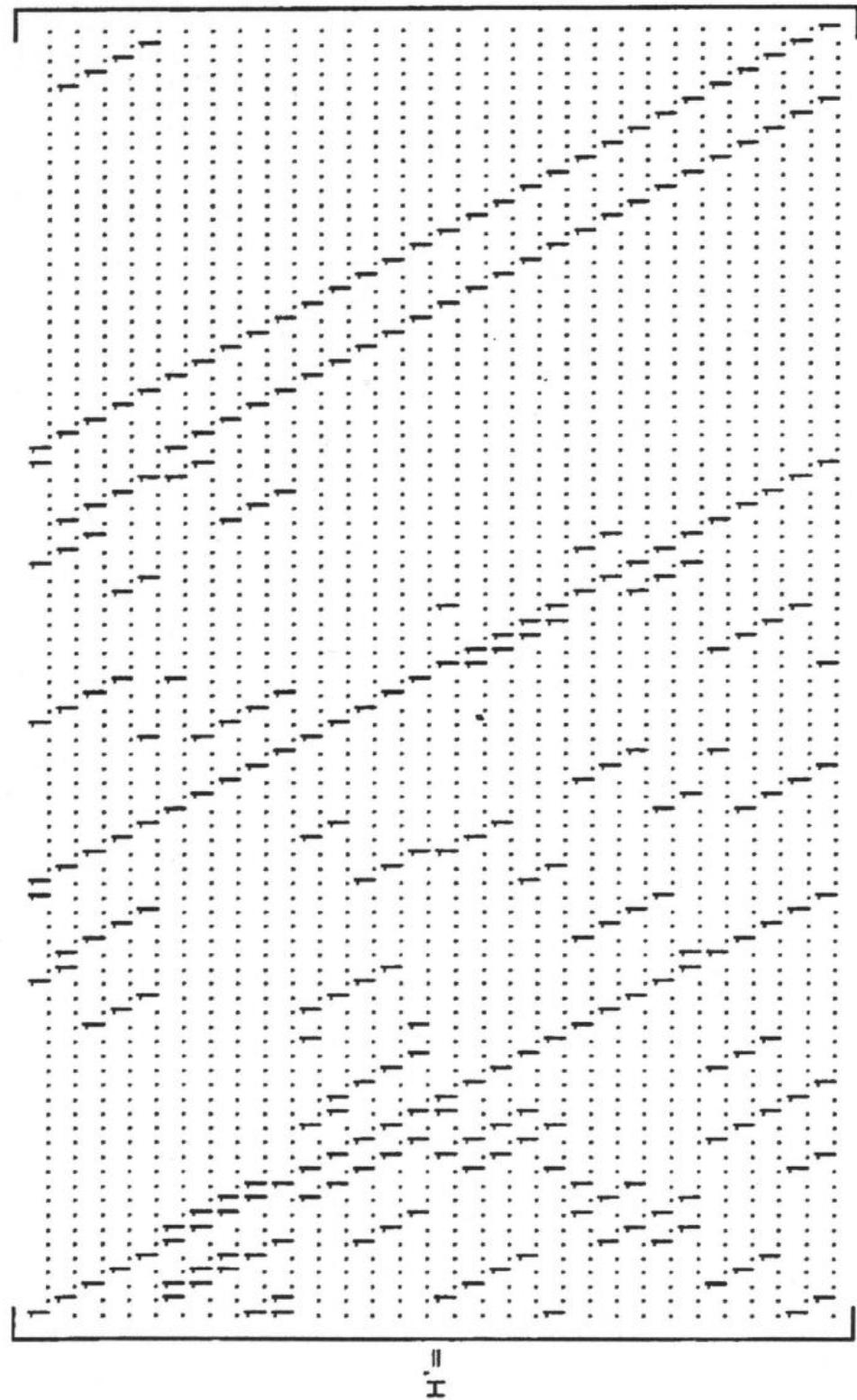


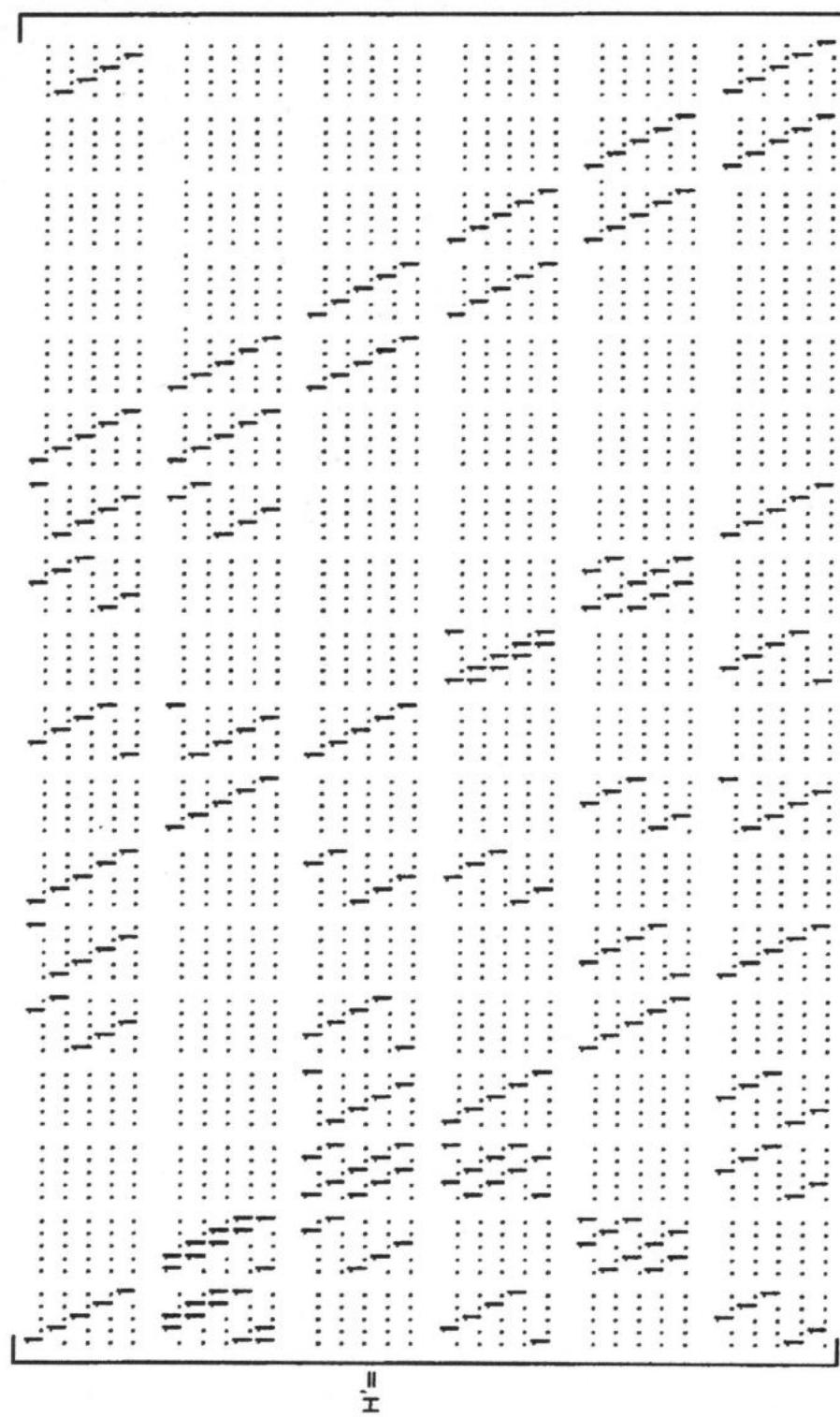
FIG. 33

FIG. 34

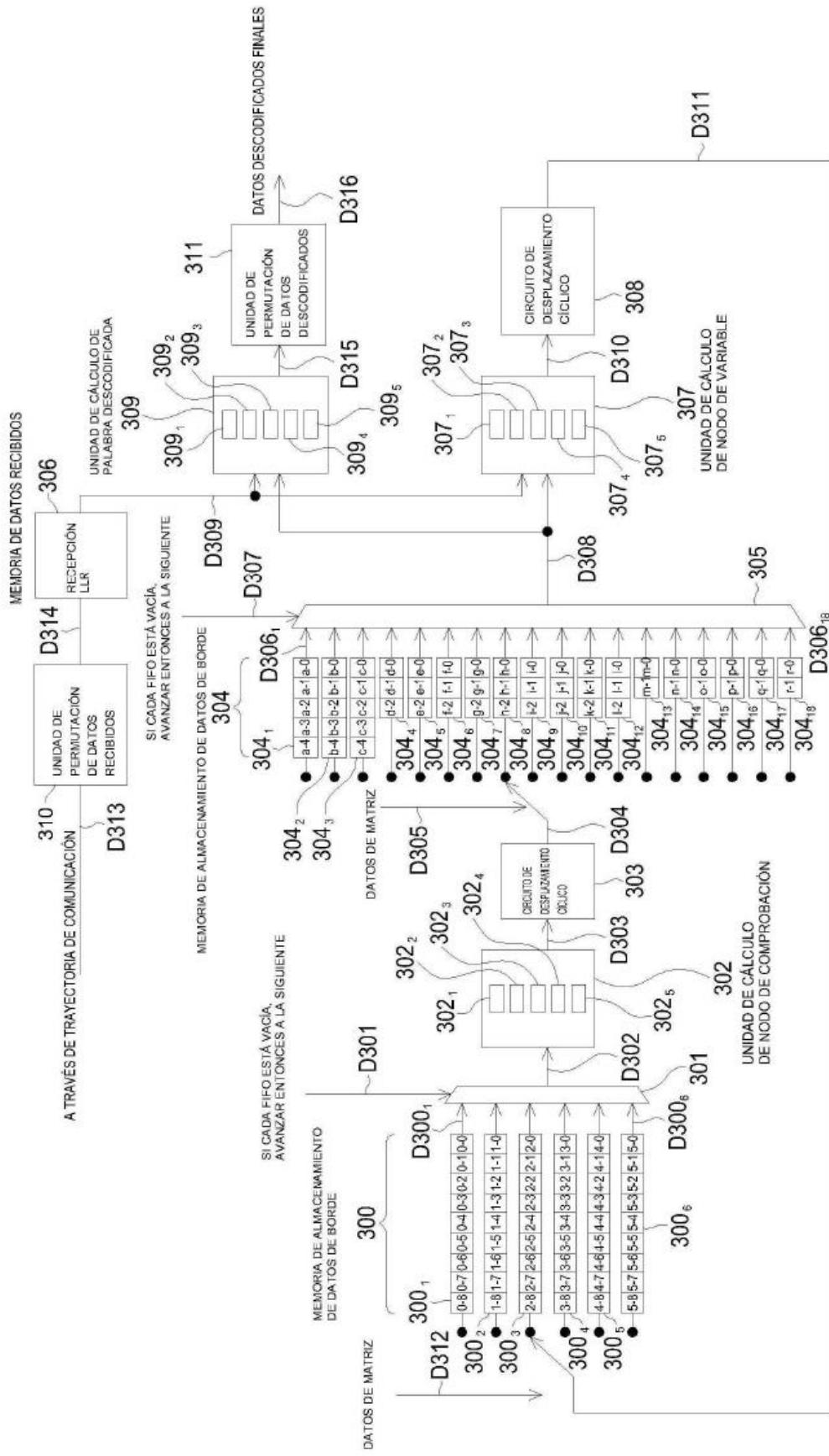


FIG. 35

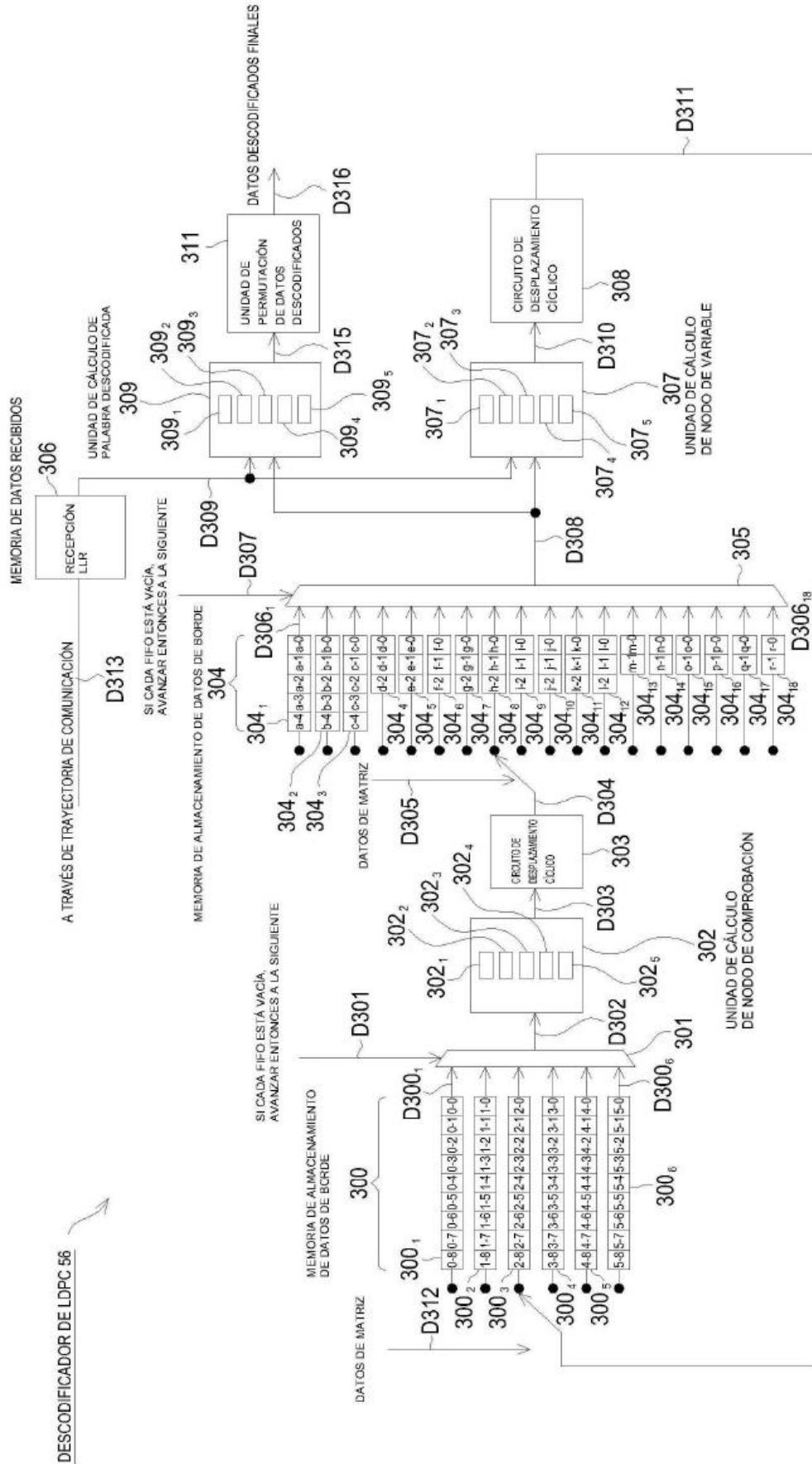
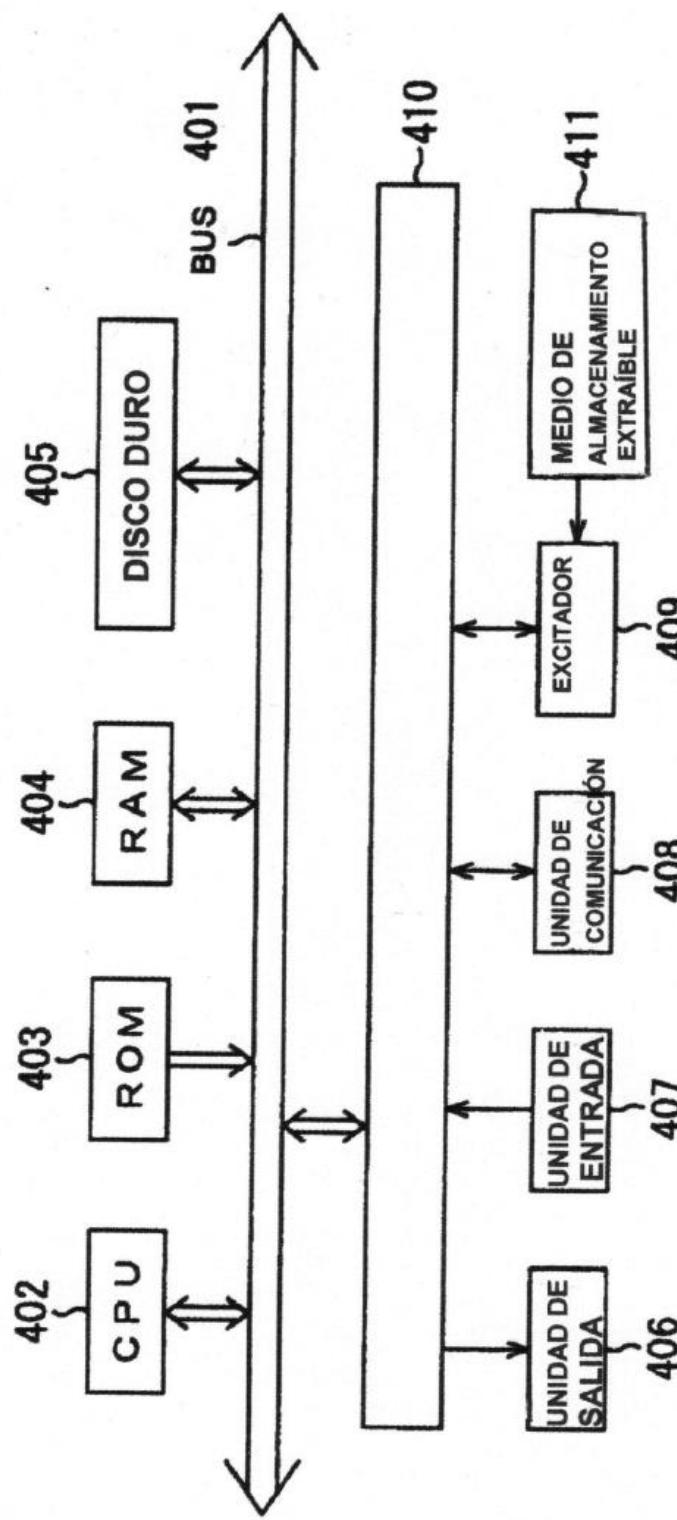


FIG. 36



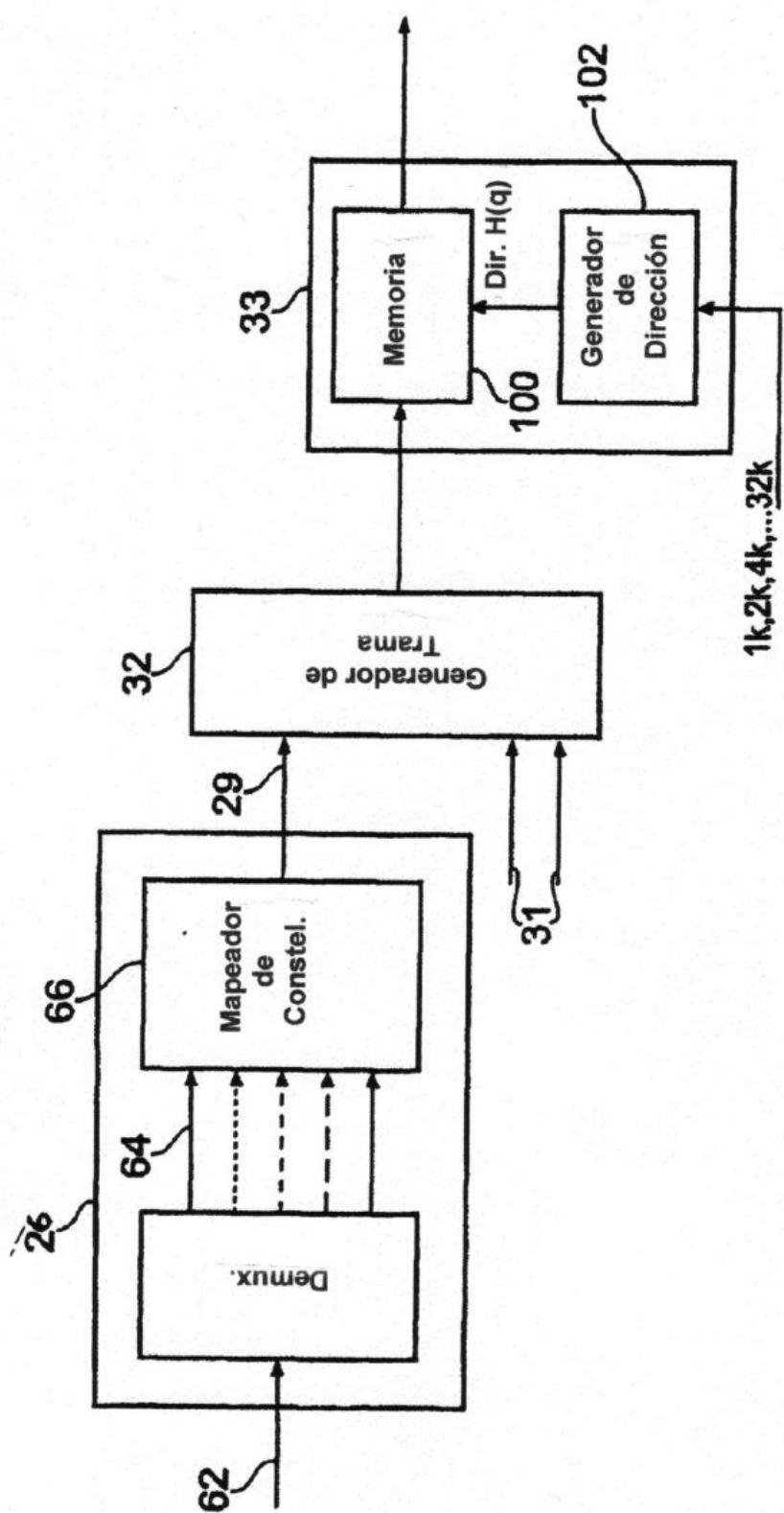


FIG. 37

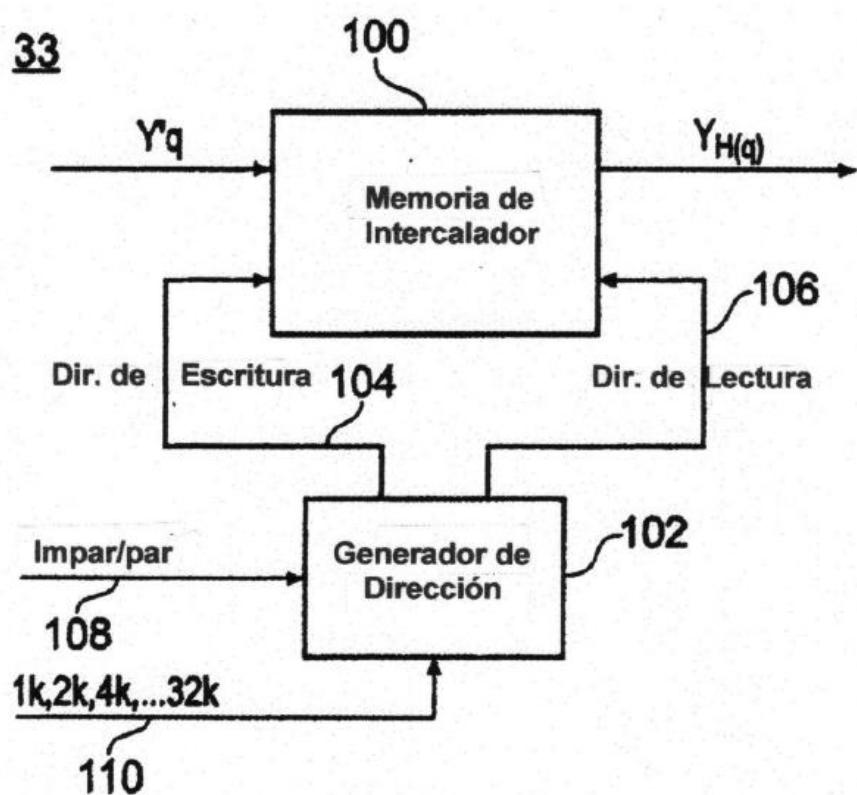


FIG. 38

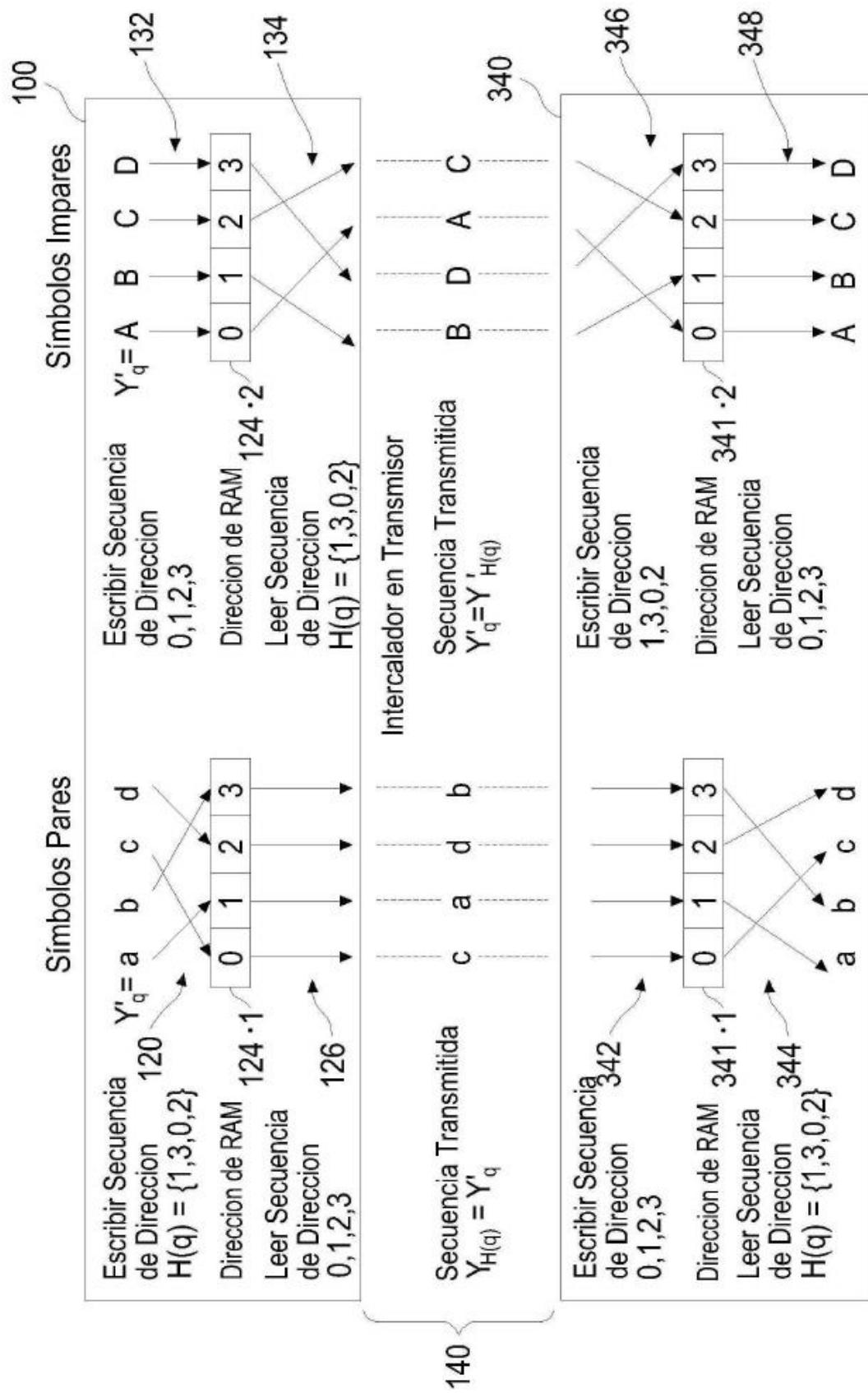


FIG. 39

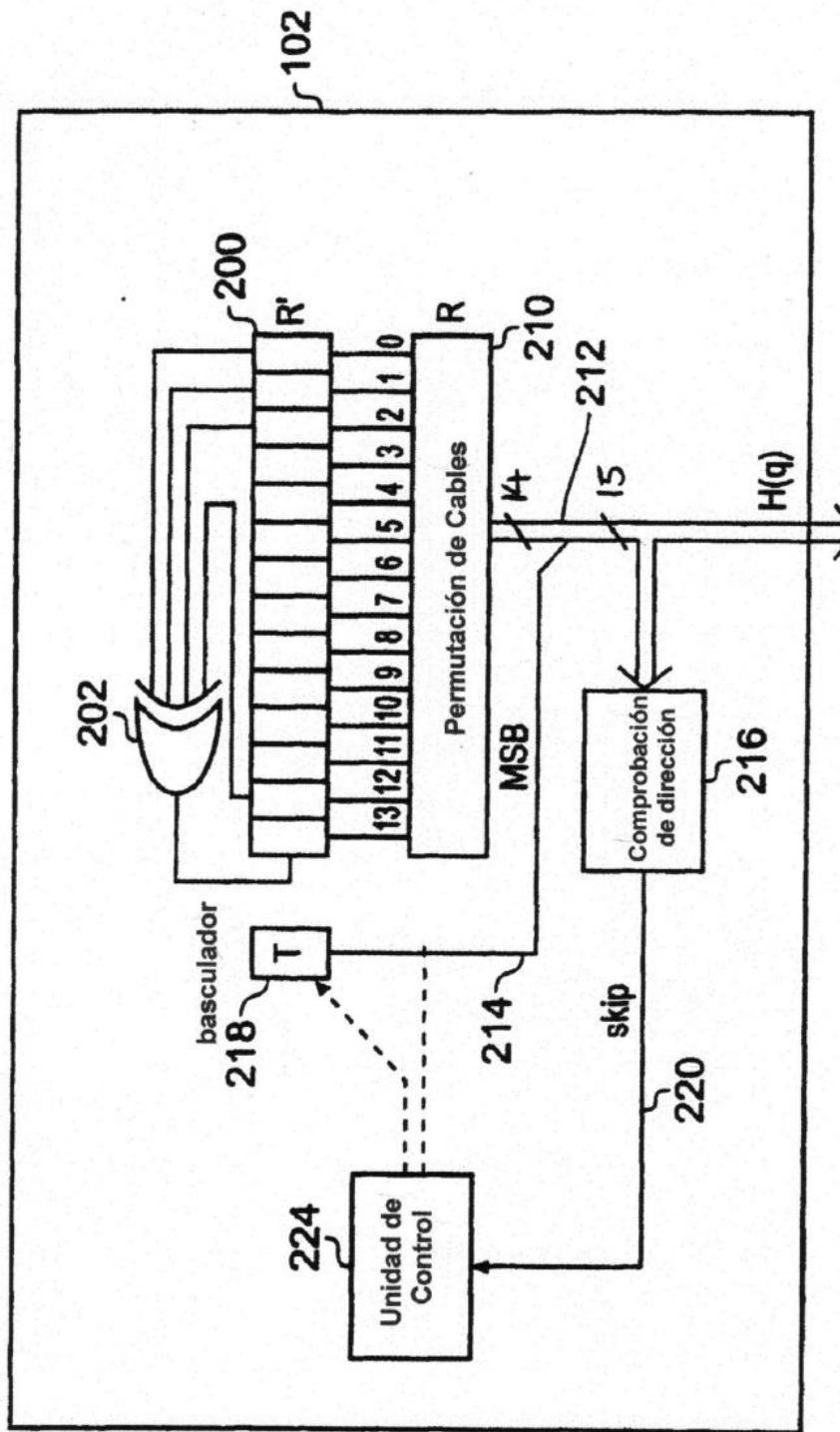
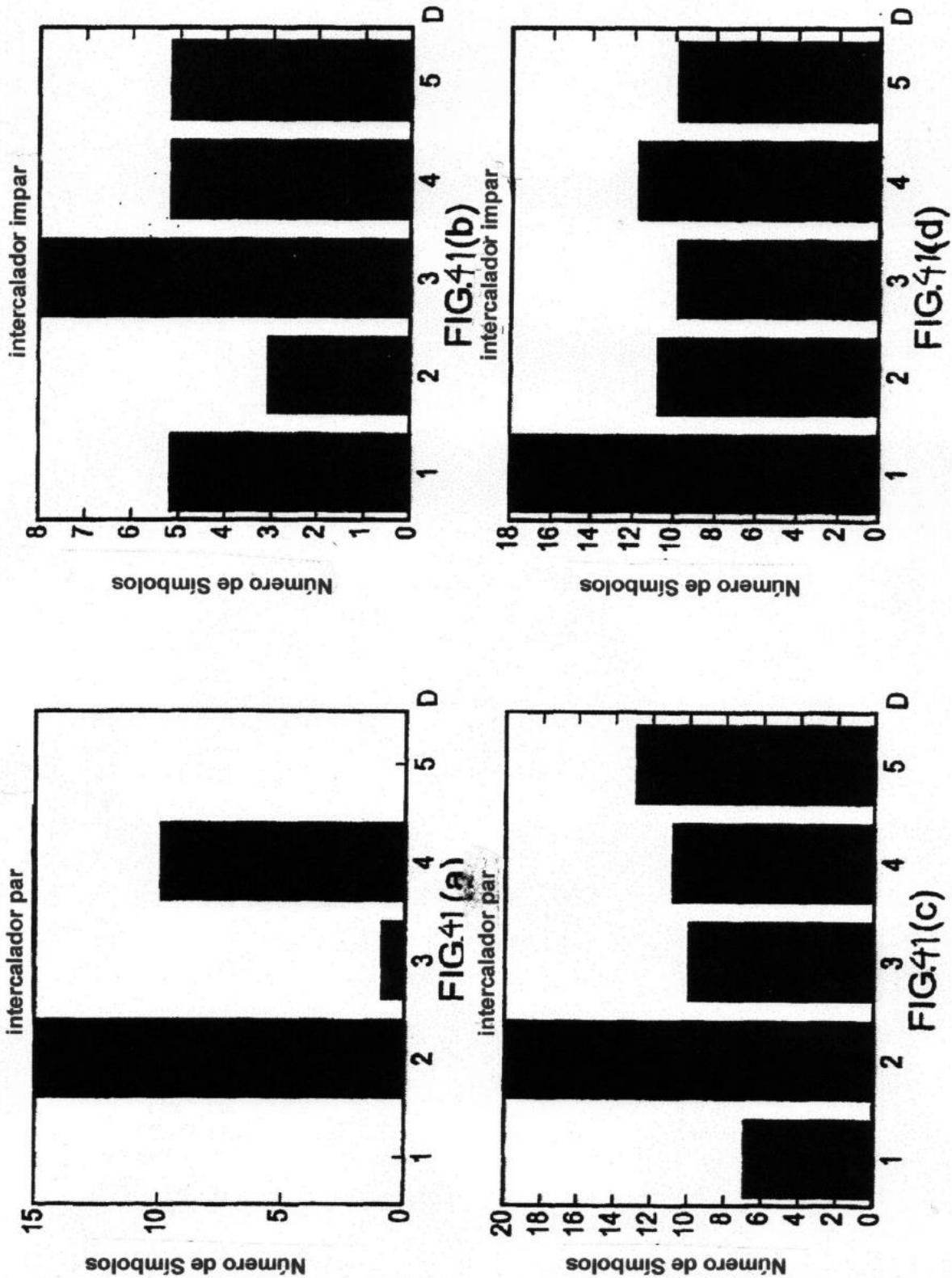


FIG. 40

Modo de 32k



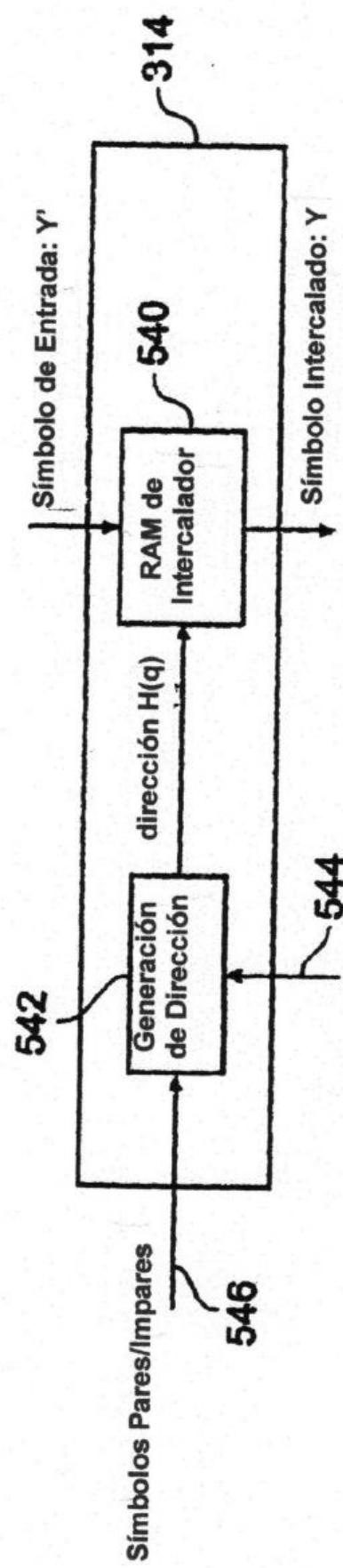


FIG. 42

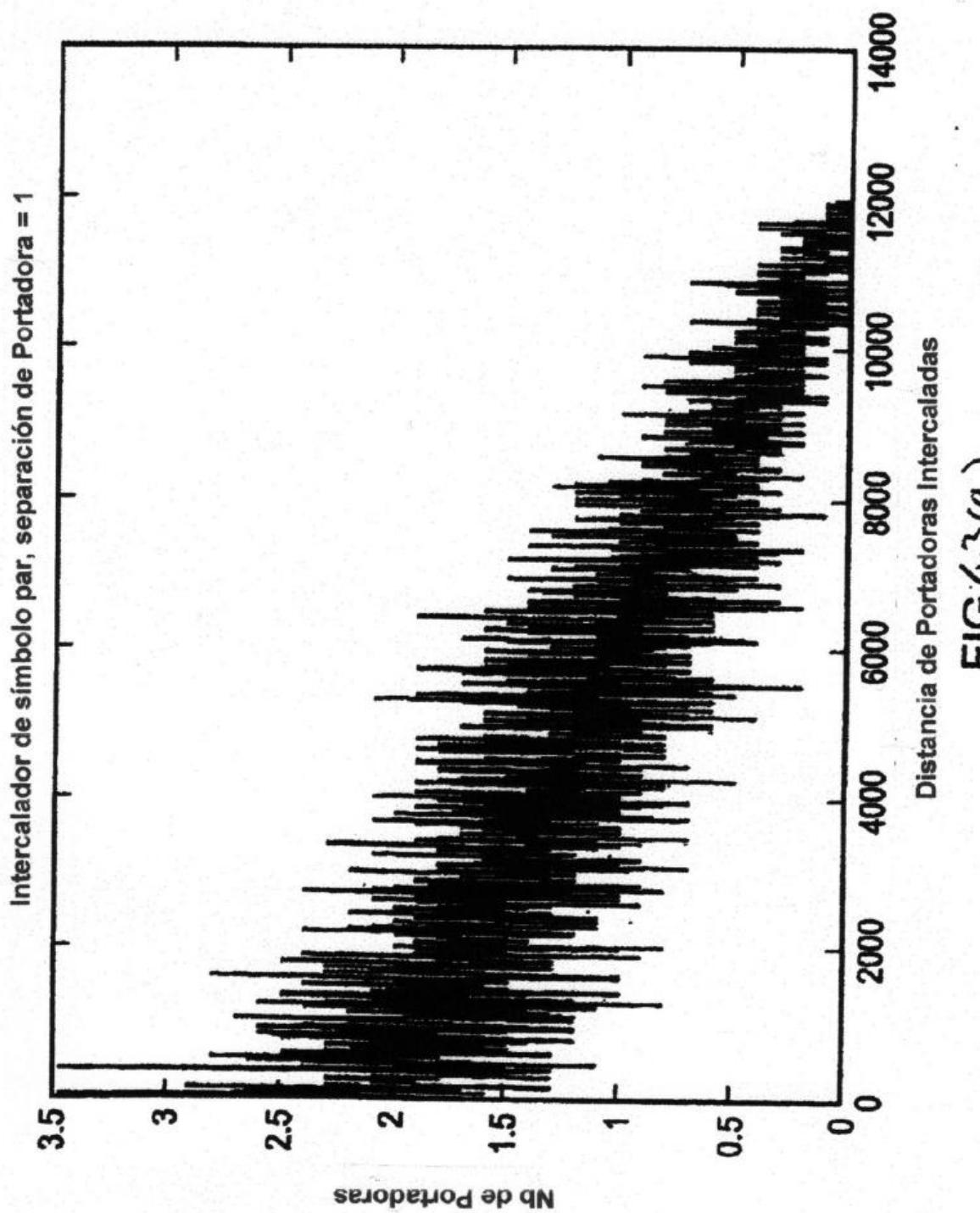


FIG.4.3(a)

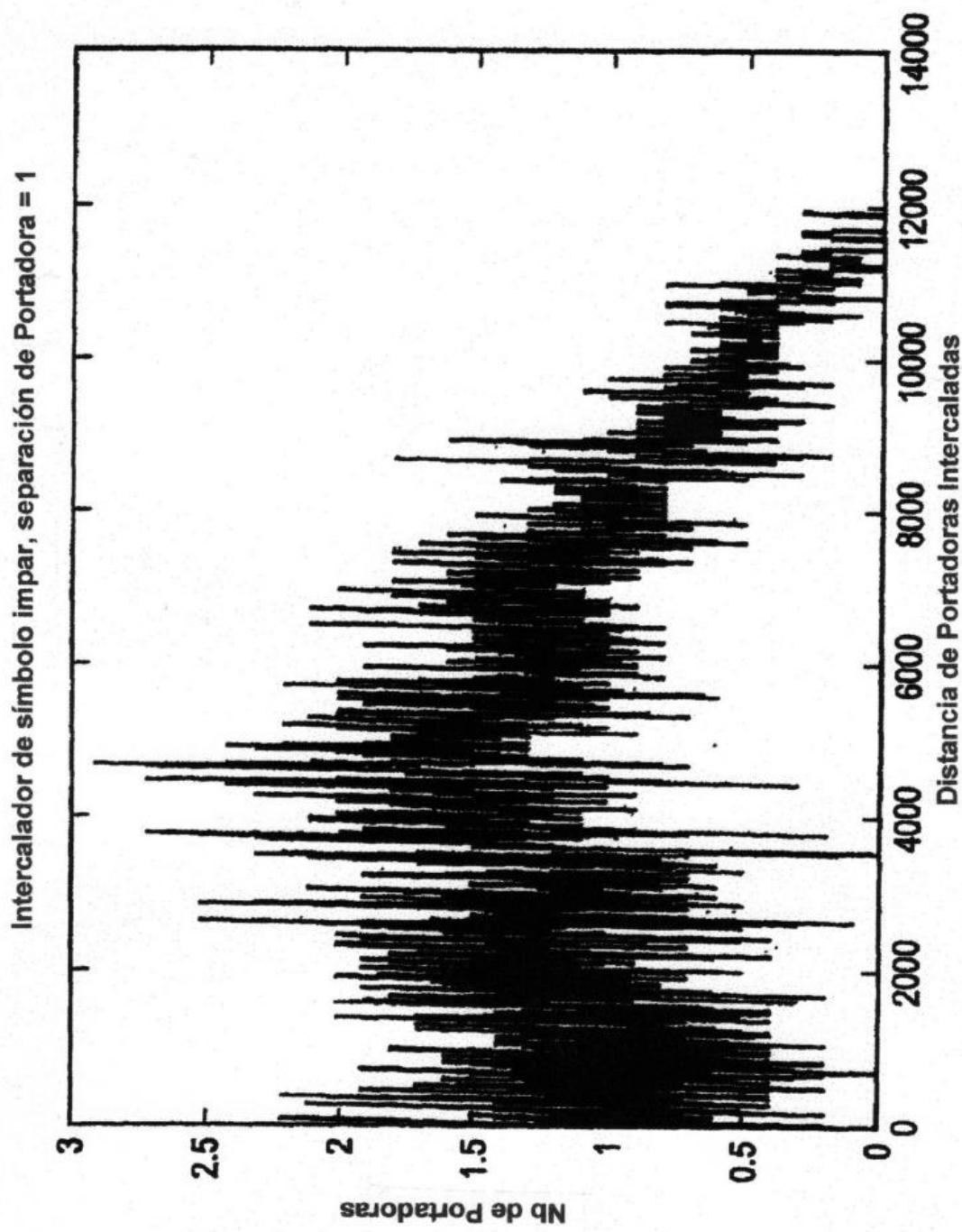


FIG. 43(b)

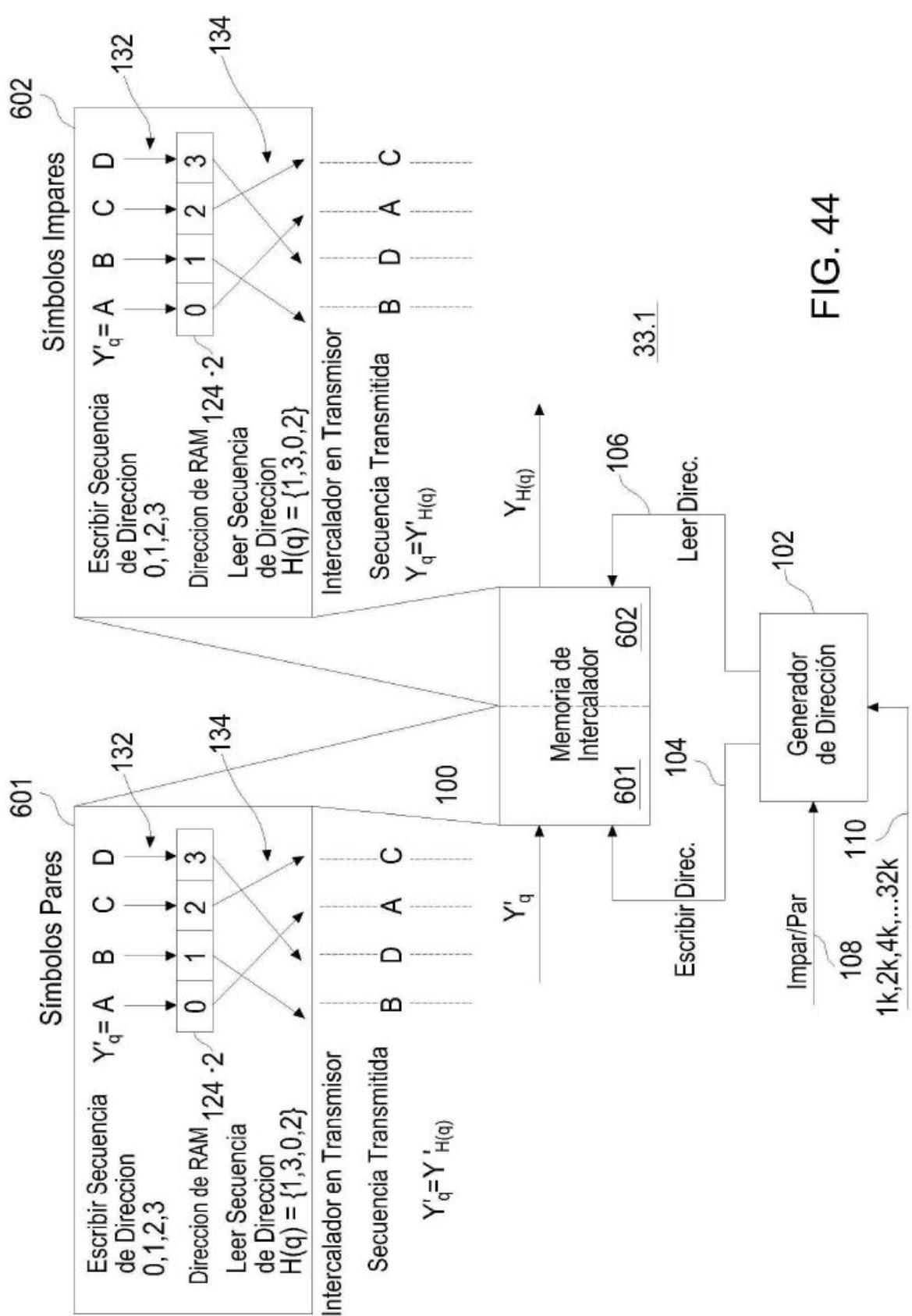


FIG. 44

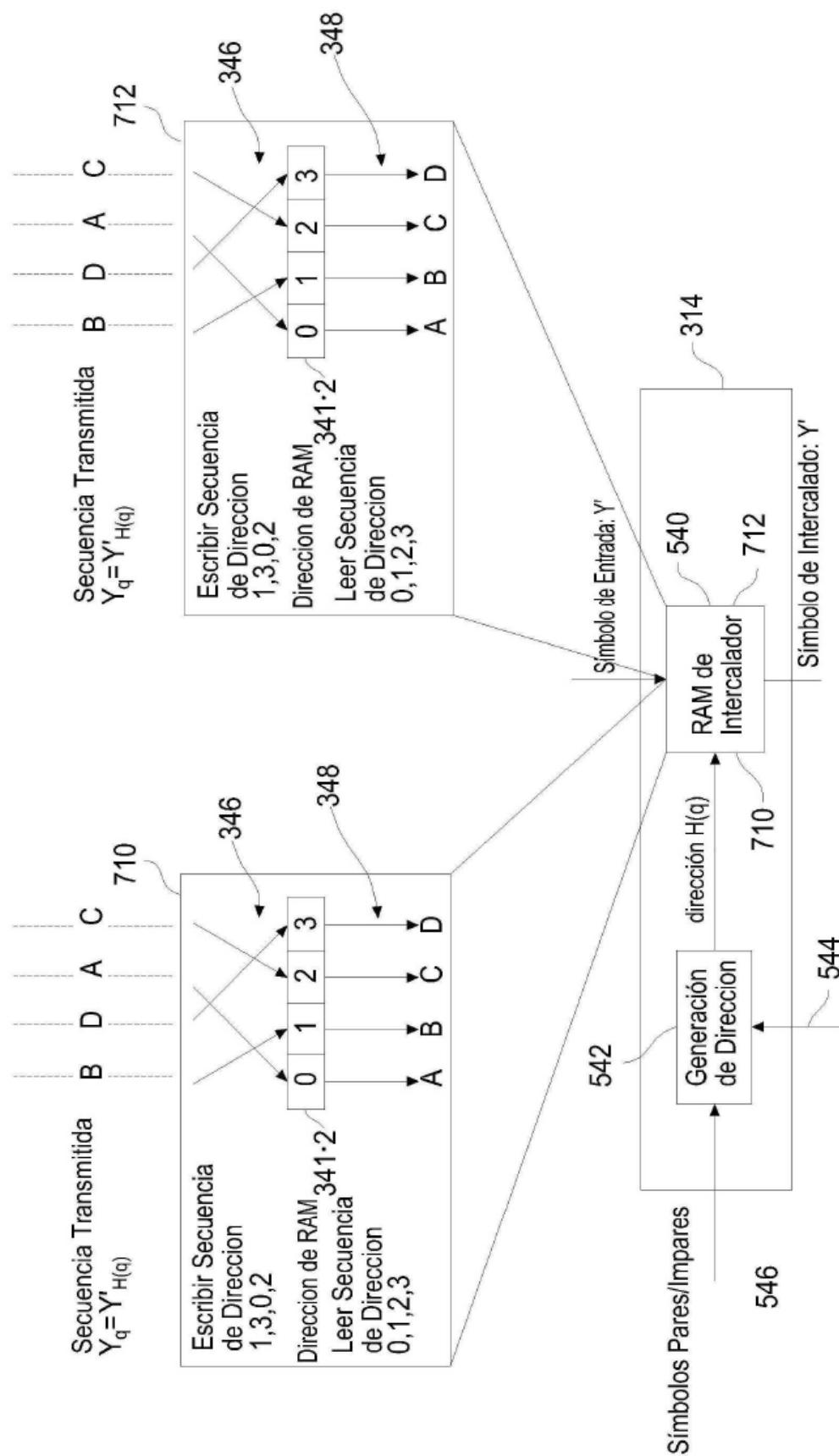


FIG. 45