

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 683 084**

51 Int. Cl.:

H03M 13/39	(2006.01)
H03M 13/11	(2006.01)
H03M 13/29	(2006.01)
H04L 1/00	(2006.01)
H04L 25/03	(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **01.07.2013 PCT/CN2013/078570**

87 Fecha y número de publicación internacional: **08.01.2015 WO15000100**

96 Fecha de presentación y número de la solicitud europea: **01.07.2013 E 13888637 (9)**

97 Fecha y número de publicación de la concesión europea: **06.06.2018 EP 3001570**

54 Título: **Método de ecualización Turbo y sistema de ecualización Turbo**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
24.09.2018

73 Titular/es:
**HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building, Bantian,
Longgang District
Shenzhen, Guangdong 518129, CN**

72 Inventor/es:
**CHANG, DEYUAN;
XIAO, ZHIYU;
YU, FAN y
ZHAO, YU**

74 Agente/Representante:
LEHMANN NOVO, María Isabel

ES 2 683 084 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

Método de ecualización Turbo y sistema de ecualización Turbo

Campo técnico

5 La presente invención está relacionada con el campo de las comunicaciones ópticas y, en particular, con un método para implementar una compensación de ecualización Turbo, un ecualizador Turbo y un sistema ecualizador Turbo.

Antecedentes

10 En la actualidad, ha aumentado la tasa de transmisión en los sistemas de transmisión de fibra óptica de alta velocidad, por ejemplo, de 40 Gb/s a 100 Gb/s, e incluso a 400 Gb/s. Sin embargo, el coste de diversos efectos en el sistema de transmisión de fibra óptica como, por ejemplo, un efecto no lineal, un efecto de dispersión por modo de polarización (PMD, polarization mode dispersion), y una codificación diferencial han limitado severamente la distancia de transmisión de los sistemas de transmisión de fibra óptica de alta velocidad. Es bien conocido que, los procesos de acción de estos efectos perjudiciales se han descrito utilizando diagramas de enrejado (trellis) y, por lo tanto, estos efectos perjudiciales se pueden compensar en alguna medida utilizando un algoritmo de compensación BCJR (Bahl, Cocke, Jelinek y Raviv), un algoritmo de operación recursiva hacia delante y hacia atrás.

20 Con el fin de compensar aún más el límite de la distancia de transmisión del sistema de transmisión de fibra óptica de alta velocidad de los efectos en el sistema de transmisión de fibra óptica, se ha propuesto que la compensación sea realizada en forma de ecualización (equalization) Turbo en un extremo receptor del sistema de transmisión de fibra óptica de alta velocidad, en otras palabras, se ha mejorado el rendimiento del sistema mediante la iteración interactiva entre un decodificador de comprobación de paridad de baja densidad (LDPC, low density parity check) y un módulo BCJR, con el fin de compensar el efecto de codificación diferencial, el efecto no lineal, el efecto PMD, etc. Dicha forma de ecualización Turbo puede mejorar enormemente el rendimiento del sistema compensando el daño en un canal. En la presente solicitud, una información tentativa de un solo bit (bit) se refiere al valor de la probabilidad de la decisión sobre si el bit es 0 ó 1. Con el fin de simplificar la operación, en general se utiliza la relación entre la probabilidad de la decisión de que el bit sea 0 y la probabilidad de

30 Además, un ecualizador Turbo que implementa la forma de ecualización Turbo anterior utiliza una estructura de realimentación y una palabra código LDPC, y un módulo BCJR del ecualizador Turbo utiliza un BCJR de ventana deslizante común con estructura serie. En general, la longitud de la palabra código LDPC utilizada en comunicación óptica alcanza diez mil bits, y la palabra código LDPC tiene que estar almacenada en el módulo BCJR. Por lo tanto, el módulo BCJR puede tener numerosos recursos de almacenamiento. Sin embargo, el ecualizador Turbo que utiliza la estructura de realimentación, la palabra código LDPC de gran longitud y el complejo módulo BCJR en su conjunto limitan el rendimiento del sistema.

35 Se puede observar que, en un sistema de transmisión de fibra óptica con una alta velocidad mayor de 100 G, para implementar un alto rendimiento mayor de 100 Gbit/s, la forma de ecualización Turbo anterior no se puede adaptar para una transmisión de alta velocidad de gran capacidad.

40 La tesis de máster Núm. 2681, 2002 de Xiao-Yu HU, École Polytechnique Fédérale de Lausanne, divulga una exposición sistemática sobre la teoría y práctica de una codificación de longitud de bloque finita. Específicamente, el capítulo 6 divulga una perspectiva de planificación de paso de mensajes sobre una ecualización Turbo. Con el fin de obtener un mejor balance entre latencia y rendimiento, propone una planificación paralela de paso de mensajes por ventanas para la detección de canal de respuesta parcial (PR) que combina las ventajas de la planificación paralela hacia delante-hacia atrás por ventanas (PWFB) con una planificación por inundación. La planificación PWFB funciona del siguiente modo. Se divide el gráfico de vectores del canal PR en ventanas no solapadas consecutivas de tamaño w . Con cada ventana, se utiliza una planificación de paso de mensajes hacia delante-hacia atrás, y la planificación se sincroniza y se ejecuta en paralelo sobre cada ventana. Se considera que la planificación de paso de mensajes PWFB utiliza ventanas no solapadas. Sin embargo, la tesis también sugiere que sería posible incorporar ventanas solapadas en una planificación en paralelo, lo cual sin embargo aumentaría la complejidad de cálculo pero se podría justificar por el aumento de la ganancia de rendimiento.

Resumen

50 La presente invención ofrece un método para implementar una compensación de ecualización Turbo y un sistema, los cuales pretenden resolver el problema de que el rendimiento se encuentra limitado cuando se implementa la compensación de ecualización Turbo en un sistema de transmisión de fibra óptica de alta velocidad.

De acuerdo con un primer aspecto, se ofrece un método para implementar una compensación de ecualización Turbo de acuerdo con la reivindicación 1.

5 Haciendo referencia al primer aspecto, en una primera forma de implementación del primer aspecto, la realización del procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos incluye: realizar concurrentemente una operación recursiva hacia delante sobre cada segmento de datos de los n segmentos de datos y realizar concurrentemente una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos.

10 Haciendo referencia al primer aspecto, en una segunda forma de implementación del primer aspecto, la realización del procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos incluye: realizar concurrentemente una operación recursiva hacia delante sobre cada segmento de datos de los n segmentos de datos.

15 Haciendo referencia al primer aspecto, en una tercera forma de implementación del primer aspecto, la realización del procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos incluye: realizar concurrentemente una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos.

20 Haciendo referencia al primer aspecto o la primera, segunda y tercera formas de implementación del primer aspecto, en una cuarta forma de implementación del primer aspecto, la realización del procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos incluye: recibir el segundo bloque de datos; realizar un procesamiento de decodificación sobre el segundo bloque de datos recibido y otros T-1 bloques de datos sobre los que se ha realizado la decodificación iterativa, donde la longitud de los datos de cada uno de los otros T-1 bloques de datos sobre los que se ha realizado la decodificación iterativa es 1/T de la longitud de código del código convolucional LDPC; y generar el tercer bloque de datos sobre el que se ha realizado el procesamiento de decodificación un número máximo de veces.

25 Haciendo referencia al primer aspecto o a la primera, segunda, tercera y cuarta formas de implementación, en una quinta forma de implementación del primer aspecto, antes de dividir el primer bloque de datos en n segmentos de datos, el método incluye, además: realizar una estimación de distribución de probabilidad de transición condicional sobre el primer bloque de datos, con el fin de determinar la información de parámetros de estimación de canal.

De acuerdo con un segundo aspecto, se ofrece un sistema ecualizador Turbo de acuerdo con la reivindicación 7.

30 Haciendo referencia al segundo aspecto, en una primera forma de implementación del segundo aspecto, la unidad OP-BCJR de cada ecualizador Turbo incluye: un módulo de segmentación, configurado para dividir el primer bloque de datos en los n segmentos de datos, donde se solapan D bits en dos segmentos de datos adyacentes en los n segmentos de datos, n es un entero positivo mayor o igual que 2, y D es un entero positivo mayor o igual que 1; un módulo de recursión, configurado para realizar el procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos; y un módulo de agrupación, configurado para agrupar los n segmentos de datos sobre los que se ha realizado el procesamiento recursivo, con el fin de obtener el segundo bloque de datos.

40 Haciendo referencia a la primera forma de implementación del segundo aspecto, en una segunda forma de implementación del segundo aspecto, el módulo de recursión está configurado para: realizar concurrentemente una operación recursiva hacia delante sobre cada segmento de datos de los n segmentos de datos y realizar concurrentemente una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos.

45 Haciendo referencia a la primera forma de implementación del segundo aspecto, en una tercera forma de implementación del segundo aspecto, el módulo de recursión está configurado para: realizar concurrentemente una operación recursiva hacia delante sobre cada segmento de datos de los n segmentos de datos.

Haciendo referencia a la primera forma de implementación del segundo aspecto, en una cuarta forma de implementación del segundo aspecto, el módulo de recursión está configurado para: realizar concurrentemente una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos.

50 Haciendo referencia al segundo aspecto o la primera, segunda, tercera y cuarta formas de implementación del segundo aspecto, en una quinta forma de implementación del segundo aspecto, la unidad de decodificación del código convolucional LDPC de cada ecualizador Turbo incluye: un módulo de recepción, configurado para recibir un segundo bloque de datos; un módulo de decodificación, configurado para realizar un procesamiento de decodificación sobre el segundo bloque de datos recibido y otros T-1 bloques de datos sobre los que se ha realizado la decodificación iterativa, donde la longitud de los datos de cada uno de los otros T-1 bloques de datos sobre los que se ha realizado la decodificación iterativa es 1/T de la longitud de código del código convolucional

LDPC; y un módulo de salida, configurado para generar el tercer bloque de datos sobre el que se ha realizado el procesamiento de decodificación un número máximo de veces.

5 Haciendo referencia al segundo aspecto o a la primera, segunda, tercera, cuarta y quinta formas de implementación del segundo aspecto, en una sexta forma de implementación del segundo aspecto, cada ecualizador Turbo incluye, además: una unidad de estimación de canal, configurada para: antes de que la unidad OP-BCJR divida el primer bloque de datos en los n segmentos de datos, realizar una estimación de distribución de probabilidad de transición condicional sobre el primer bloque de datos, con el fin de determinar la información de parámetros de estimación de canal.

10 Haciendo referencia al segundo aspecto o a la primera, segunda, tercera, cuarta, quinta y sexta formas de implementación del segundo aspecto, en una séptima forma de implementación del segundo aspecto, el sistema ecualizador Turbo incluye, además, una primera unidad independiente de decodificación de código convolucional LDPC, donde la primera unidad independiente de decodificación de código convolucional LDPC recibe el tercer bloque de datos de salida de un último ecualizador Turbo de los múltiples ecualizadores Turbo y realiza una decodificación iterativa sobre el tercer bloque de datos, para generar un cuarto bloque de datos, donde la longitud de datos del cuarto bloque de datos es $1/T$ de la longitud de código del código convolucional LDPC.

15 Haciendo referencia a la séptima forma de implementación del segundo aspecto, en una octava forma de implementación del segundo aspecto, el sistema ecualizador Turbo comprende, además, una segunda unidad independiente de decodificación de código convolucional LDPC, en donde la segunda unidad independiente de decodificación de código convolucional LDPC recibe el cuarto bloque de datos de salida de la primera unidad independiente de decodificación de código convolucional LDPC y realiza una decodificación iterativa sobre el cuarto bloque de datos, para generar un quinto bloque de datos, en donde la longitud de datos del quinto bloque de datos es $1/T$ de la longitud de código del código convolucional LDPC.

20 Los modos de realización de la presente invención se aplican a un extremo de recepción de un sistema de transmisión de fibra óptica de alta velocidad. Mediante la realización, en una unidad OP-BCJR, del procesamiento de segmentación y las operaciones recursivas hacia delante y hacia atrás sobre un bloque de datos recibido, y la realización, en una unidad de decodificación del código convolucional LDPC, del procesamiento iterativo Turbo sobre los datos obtenidos desde la unidad OP-BCJR, se puede mejorar de forma efectiva el rendimiento del sistema.

Breve descripción de los dibujos

30 Con el fin de describir con más claridad las soluciones técnicas en los modos de realización de la presente invención, a continuación se introducen brevemente los dibujos adjuntos necesarios para describir los modos de realización de la presente invención. Evidentemente, los dibujos adjuntos en la siguiente descripción únicamente muestran algunos modos de realización de la presente invención, y una persona con un conocimiento normal en la técnica puede todavía derivar sin esfuerzos creativos otros dibujos a partir de estos dibujos adjuntos.

35 La FIG. 1 es un diagrama de flujo de un método de compensación de ecualización Turbo de ejemplo;

la FIG. 2 es un diagrama esquemático de la estructura de un ecualizador Turbo de ejemplo;

la FIG. 3 es un diagrama esquemático de la estructura de una unidad OP-BCJR en un ecualizador Turbo de ejemplo;

40 la FIG. 4 es un diagrama esquemático de la estructura de una unidad de decodificación de código convolucional LDPC en un ecualizador Turbo de ejemplo;

la FIG. 5 es un diagrama esquemático de la estructura de un ecualizador Turbo de ejemplo;

la FIG. 6 es un diagrama esquemático de la estructura de un sistema ecualizador Turbo de acuerdo con un modo de realización de la presente invención;

45 la FIG. 7 es un diagrama esquemático de la estructura de un sistema ecualizador Turbo de acuerdo con otro modo de realización de la presente invención;

la FIG. 8 es un diagrama de la estructura de un sistema ecualizador Turbo de acuerdo con un modo de realización específico de la presente invención;

la FIG. 9 es un diagrama de la estructura de un ecualizador Turbo de acuerdo con un modo de realización específico de la presente invención;

50 la FIG. 10 es un diagrama esquemático de una ranura de tiempo de un procesamiento iterativo de una unidad OP-BCJR en un ecualizador Turbo de acuerdo con un modo de realización específico de la presente invención;

la FIG. 11 es un diagrama esquemático de un proceso de procesamiento específico de una unidad OP-BCJR en un ecualizador Turbo de acuerdo con un modo de realización específico de la presente invención; y

la FIG. 12 es un diagrama de la estructura de un sistema ecualizador Turbo de acuerdo con otro modo de realización específico de la presente invención.

5 Descripción de los modos de realización

A continuación, se describen claramente las soluciones técnicas en los modos de realización de la presente invención haciendo referencia a los dibujos adjuntos en los modos de realización de la presente invención. Evidentemente, los modos de realización descritos son algunos pero no todos los modos de realización de la presente invención. Cualesquiera otros modos de realización obtenidos por una persona con un conocimiento ordinario en la técnica basándose en los modos de realización de la presente invención sin esfuerzos creativos se considerarán dentro del alcance de protección de la presente invención.

Las soluciones técnicas de la presente invención se pueden aplicar a varios sistemas de comunicaciones como, por ejemplo: un sistema del Sistema Global para Comunicaciones Móviles (GSM), un sistema de Acceso Múltiple por División de Código (CDMA), un sistema de Acceso Múltiple por División de Código de Banda Ancha (WCDMA), un sistema del servicio general de radio por paquetes (GPRS) y un sistema de la Evolución a Largo Plazo (LTE).

Un equipo de usuario (UE) también se puede denominar terminal móvil o estación móvil y se puede comunicar con una o más redes troncales utilizando una red de acceso radio (RAN, Radio Access Network). El UE intercambia voz y/o datos con la red de acceso radio.

Una estación base puede ser una estación base (BTS) en GSM o CDMA, también puede ser una estación base (NodoB) en WCDMA, y también puede ser un NodoB evolucionado (eNB o e-NodoB) en LTE. Además, una estación base puede soportar/gestionar una o más celdas; cuando necesita comunicarse con una red, el UE selecciona una celda para iniciar el acceso a la red.

Con el fin de resolver el problema de que el rendimiento es limitado cuando se implementa la compensación de ecualización Turbo en un sistema de transmisión de fibra óptica de alta velocidad, los modos de realización de la presente invención ofrecen un método para implementar una compensación de ecualización Turbo que se aplica a un extremo de recepción del sistema de transmisión de fibra óptica de alta velocidad.

Por ejemplo, para un extremo de transmisión, después de que una señal enviada atraviese un entramador en una unidad de transporte óptico (OTU), la señal enviada se somete sucesivamente a una codificación de código convolucional en un codificador de código convolucional LDPC, y una codificación diferencial en un codificador diferencial, y finalmente la señal óptica es enviada por un modulador óptico a una red de transmisión de fibra óptica. Para un extremo de recepción, después de una comprobación coherente, sobre la señal óptica se realiza un muestreo en un convertidor analógico digital (ADC) y un procesamiento de ecualización normal, la señal óptica entra en un sistema ecualizador Turbo para implementar una compensación de ecualización Turbo y, por último, genera una señal recibida después de atravesar el desentramador en la OTU.

En los modos de realización de la presente invención reivindicados, el sistema ecualizador Turbo incluye múltiples ecualizadores Turbo, y cada ecualizador Turbo incluye una unidad OP-BCJR y una unidad de decodificación de código convolucional LDPC. Por otro lado, el sistema ecualizador Turbo puede incluir, además, al menos una unidad de decodificación de código convolucional LDPC independiente.

A continuación se describe, utilizando un ejemplo en el que el sistema ecualizador Turbo incluye un ecualizador Turbo, un método de ejemplo para implementar la compensación de ecualización Turbo. Refiérase a los siguientes pasos.

S11: Una unidad OP-BCJR en un ecualizador Turbo divide un primer bloque de datos en n segmentos de datos, donde se solapan D bits en dos segmentos de datos adyacentes en los n segmentos de datos, n es un entero positivo mayor o igual que 2, y D es un entero positivo mayor o igual que 1, realiza un procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos, y une los n segmentos de datos sobre los que se ha realizado el procesamiento recursivo con el fin de obtener un segundo bloque de datos.

En la presente solicitud, la longitud de datos tanto del primer bloque de datos como del segundo bloque de datos son ambas de $1/T$ de la longitud de código de un código convolucional LDPC, y T es la cantidad de capas de una matriz de comprobación escalonada del código convolucional LDPC. Además, el valor de estado del símbolo inicial de los D bits solapados tiene una distribución equiprobable. La distribución equiprobable se refiere a que la probabilidad de distribución de estado en este bit es igual en cada posible estado.

En la presente solicitud, la longitud de código del código convolucional LDPC se refiere a la longitud de datos que cumple una relación de comprobación de capa. En la presente solicitud, "el cumplimiento de una relación de comprobación de capa" se refiere a que $x \cdot H_i^t = 0$ e $i = 1, 2, \dots, T$, donde x son los datos de bit de decisión firme que cumplen la relación y H_i^t es una transposición de la i -ésima capa H_i de una matriz de comprobación del código convolucional LDPC. En la presente solicitud, H_1 a H_T constituyen la matriz de comprobación del código convolucional LDPC.

En otras palabras, T indica que un total de T bloques de palabras código se combinan para conjuntamente cumplir la relación de comprobación del código convolucional LDPC. T se determina mediante un parámetro de estructura escalonada (esto es, la cantidad de capas) de la matriz H de comprobación del código convolucional LDPC. Por ejemplo, suponiendo que la cantidad de columnas espaciadas entre sí entre la capa i -ésima H_i y la capa $(i+1)$ -ésima H_{i+1} de la matriz de comprobación del código convolucional LDPC es N_T , y la cantidad de columnas en cada capa de la matriz de comprobación del código convolucional LDPC es N , donde en general N_T y N son constantes, entonces $T=N/N_T$.

Se puede observar que, la longitud de datos del bloque de datos procesado en el ecualizador Turbo es únicamente $1/T$ de la longitud de código del código convolucional LDPC y, por lo tanto, se pueden reducir los recursos de almacenamiento necesarios por parte de la unidad OP-BCJR.

Además, la realización del procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos puede incluir: realizar concurrentemente una operación recursiva hacia delante y una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos. La realización del procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos puede también incluir: realizar concurrentemente una operación recursiva hacia delante sobre cada segmento de datos de los n segmentos de datos. La realización del procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos puede también incluir: realizar concurrentemente una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos. Opcionalmente, la realización del procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos puede incluir: realizar una operación recursiva hacia delante sobre algunos segmentos de datos en los n segmentos de datos, y realizar una operación recursiva hacia atrás en el resto de segmentos de datos.

Además, cuando la unidad OP-BCJR realiza una operación recursiva hacia delante y una operación recursiva hacia atrás, es necesario utilizar una función de densidad de probabilidad (PDF), un parámetro de distribución de probabilidad y un parámetro de probabilidad de transición de un canal. En algunos escenarios, estos parámetros se conocen con antelación, pero en algunos escenarios de aplicación, estos parámetros únicamente se pueden obtener mediante una estimación del canal. Por lo tanto, antes de dividir el primer bloque de datos en los n segmentos de datos, también es necesario realizar una estimación de distribución de probabilidad de transición condicional sobre el primer bloque de datos, con el fin de determinar la información de los parámetros de estimación de canal.

S12: Una unidad de decodificación de código convolucional LDPC en el ecualizador Turbo realiza una decodificación iterativa sobre el segundo bloque de datos, para generar un tercer bloque de datos. En la presente solicitud, la longitud de datos del tercer bloque de datos también es $1/T$ de la longitud de código del código convolucional LDPC.

La longitud de datos del bloque de datos en el procesamiento de ecualización Turbo es siempre $1/T$ de la longitud de código del código convolucional LDPC y, por lo tanto, el rendimiento del sistema se puede mejorar de forma efectiva.

Específicamente, la realización de la decodificación iterativa sobre el segundo bloque de datos para generar un tercer bloque de datos incluye: recibir el segundo bloque de datos; realizar un procesamiento de decodificación sobre el segundo bloque de datos recibido y otros $T-1$ bloques de datos sobre los que se ha realizado la decodificación iterativa; y generar el tercer bloque de datos sobre el que se ha realizado el procesamiento de decodificación un número máximo de veces. En la presente solicitud, la longitud de datos de cada uno de los otros $T-1$ bloques de datos sobre los que se ha realizado la decodificación iterativa es $1/T$ de la longitud de código del código convolucional LDPC.

A partir de lo anterior se puede deducir que este modo de realización de ejemplo se aplica a un extremo de recepción de un sistema de transmisión de fibra óptica de alta velocidad. Mediante la realización, en una unidad OP-BCJR, del procesamiento de segmentación y las operaciones recursivas hacia delante y hacia atrás sobre un bloque de datos recibido, y la realización, en una unidad de decodificación de código convolucional LDPC, de un procesamiento Turbo iterativo sobre los datos obtenidos de la unidad OP-BCJR, el rendimiento del sistema se puede mejorar de forma efectiva.

La FIG. 2 muestra un diagrama esquemático de la estructura de un ecualizador Turbo de ejemplo para implementar una compensación de ecualización Turbo. Haciendo referencia al ecualizador Turbo de la FIG. 2, a continuación se describe detalladamente cómo implementar el método de compensación de ecualización Turbo descrito más arriba.

5 En la FIG. 2, un ecualizador Turbo 20 incluye una unidad OP-BCJR 21 y una unidad 22 de decodificación de código convolucional LDPC, donde:

la unidad OP-BCJR 21 está configurada para dividir un primer bloque de datos en n segmentos de datos, donde se solapan D bits en dos segmentos de datos adyacentes en los n segmentos de datos, n es un entero positivo mayor o igual que 2, y D es un entero positivo mayor o igual que 1, realizar un procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos, y agrupar los n segmentos de datos sobre los que se ha realizado el procesamiento recursivo con el fin de obtener un segundo bloque de datos; y

la unidad 22 de decodificación de código convolucional LDPC está conectada a la unidad OP-BCJR 21, y configurada para realizar una decodificación iterativa sobre el segundo bloque de datos para generar un tercer bloque de datos.

15 En lo anterior, las longitudes de datos del primer bloque de datos, segundo bloque de datos y tercer bloque de datos son todas $1/T$ de la longitud del código de un código convolucional de comprobación de paridad de baja densidad LDPC, y T es la cantidad de capas de una matriz de comprobación escalonada del código convolucional LDPC.

Además, tal como se muestra en la FIG. 3, la unidad OP-BCJR 21 puede incluir un módulo 211 de segmentación, un módulo 212 de recursión y un módulo 213 de agrupación, donde:

el módulo 211 de segmentación está configurado para dividir el primer bloque de datos en los n segmentos de datos, donde se solapan D bits en dos segmentos de datos adyacentes en los n segmentos de datos, n es un entero positivo mayor o igual que 2 y D es un entero positivo mayor o igual que 1;

el módulo 212 de recursión está configurado para realizar el procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos; y

el módulo 213 de agrupación está configurado para agrupar los n segmentos de datos sobre los que se ha realizado el procesamiento recursivo, con el fin de obtener el segundo bloque de datos.

Específicamente, el módulo 212 de recursión está configurado para realizar concurrentemente una operación recursiva hacia delante y una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos; o, realizar concurrentemente una operación recursiva hacia delante sobre cada segmento de datos de los n segmentos de datos, o, realizar concurrentemente una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos. Opcionalmente, el módulo 212 de recursión puede estar configurado, además, para realizar una operación recursiva hacia delante sobre algunos segmentos de datos en los n segmentos de datos, y realizar una operación recursiva hacia atrás en el resto de segmentos de datos.

Además, tal como se muestra en la FIG. 4, la unidad 22 de decodificación de código convolucional LDPC puede incluir un módulo 221 de recepción, un módulo 222 de decodificación y un módulo 223 de salida, donde:

el módulo 221 de recepción está configurado para recibir el segundo bloque de datos;

el módulo 222 de decodificación está configurado para realizar un procesamiento de decodificación sobre el segundo bloque de datos recibido y los otros $T-1$ bloques de datos sobre los que se ha realizado la decodificación iterativa, donde la longitud de datos de cada uno de los otros $T-1$ bloques de datos sobre los que se ha realizado la decodificación iterativa es $1/T$ de la longitud de código del código convolucional LDPC; y

el módulo 223 de salida está configurado para generar el tercer bloque de datos sobre el que se ha realizado el procesamiento de decodificación un número máximo de veces.

A partir de lo anterior se puede deducir que este ecualizador Turbo de ejemplo se aplica a un extremo de recepción de un sistema de transmisión de fibra óptica de alta velocidad. Mediante la realización, en una unidad OP-BCJR, del procesamiento de segmentación y las operaciones recursivas hacia delante y hacia atrás sobre un bloque de datos recibido, y la realización, en una unidad de decodificación de código convolucional LDPC, de un procesamiento Turbo iterativo sobre los datos obtenidos de la unidad OP-BCJR, el rendimiento del sistema se puede mejorar de forma efectiva.

Además de una unidad OP-BCJR 21 y una unidad 22 de decodificación de código convolucional LDPC, un ecualizador Turbo 50 que se muestra en la FIG. 5 incluye, además, una unidad 23 de estimación de canal, donde la unidad 23 de estimación de canal está configurada para: antes de que la unidad OP-BCJR divida un primer

bloque de datos en n segmentos de datos, realizar una estimación de distribución de probabilidad de transición condicional sobre el primer bloque de datos, con el fin de determinar la información de parámetros de estimación de canal.

5 De este modo, cuando se realiza una operación recursiva hacia delante y/o hacia atrás, la unidad OP-BCJR necesita utilizar un parámetro de distribución de probabilidad PDF, un parámetro de probabilidad de transición, etc., de un canal los cuales se pueden obtener mediante una estimación del canal.

En lo anterior, se proporciona la descripción completa utilizando un ecualizador Turbo

10 un primer bloque de datos en n segmentos de datos, realizar una estimación de distribución de probabilidad de transición condicional sobre el primer bloque de datos, para determinar la información de parámetros de estimación de canal.

De este modo, cuando se realiza una operación recursiva hacia delante y/o hacia atrás, la unidad OP-BCJR necesita utilizar un parámetro de distribución de probabilidad PDF, un parámetro de probabilidad de transición, etc., de un canal, los cuales se pueden obtener mediante una estimación del canal.

15 En los modos de realización anteriores, se proporciona la descripción completa utilizando a modo de ejemplo un ecualizador Turbo en el sistema ecualizador Turbo. De hecho, con el fin de hacer mejor el efecto de compensación de ecualización Turbo, en general se considera que el sistema ecualizador Turbo puede incluir al menos un ecualizador Turbo como el descrito más arriba. Alternativamente, el sistema ecualizador Turbo puede incluir al menos un ecualizador Turbo como el descrito más arriba y al menos una unidad de decodificación de código convolucional LDPC como la descrita más arriba, donde las posiciones relativas del ecualizador Turbo y la
20 unidad de decodificación de código convolucional LDPC pueden cambiar aleatoriamente y no se encuentran limitadas. Por lo tanto, se puede realizar consecutivamente un procesamiento BCJR multinivel y una decodificación de código convolucional LDPC sobre los bloques de datos cuya longitud de datos es $1/T$ de la longitud de código del código convolucional; y como la unidad OP-BCJR y la unidad de decodificación de código convolucional LDPC se encuentran conectadas en serie, se realizará sobre los bloques de datos el
25 procesamiento iterativo de ecualización Turbo.

El sistema ecualizador Turbo 60 mostrado en la FIG. 6 incluye al menos el ecualizador Turbo 20 mostrado en la FIG. 2.

30 El sistema ecualizador Turbo 70 mostrado en la FIG. 7 incluye al menos el ecualización Turbo 20 mostrado en la FIG. 2, y al menos una unidad 22 de decodificación de código convolucional LDPC. Una unidad de decodificación de código convolucional LDPC de la al menos una unidad de decodificación de código convolucional LDPC recibe el tercer bloque de datos de salida del ecualizador Turbo en el al menos un ecualizador Turbo u otra unidad de decodificación de código convolucional LDPC de la al menos una unidad de decodificación de código convolucional LDPC, y realiza una decodificación iterativa sobre el tercer bloque de datos, para generar un cuarto bloque de datos, donde la longitud de datos del cuarto bloque de datos es $1/T$ de la longitud de código del código convolucional LDPC.
35

40 Por ejemplo, tal como se muestra en el diagrama esquemático de la estructura del sistema ecualizador Turbo de la FIG. 7, múltiples ecualizadores Turbo 20 están conectados, los cuales a continuación están conectados, además, a una o más unidades 22 de decodificación de código convolucional LDPC. En otras palabras, el tercer bloque de datos de salida del primer ecualizador Turbo 20 se envía a un segundo ecualizador Turbo 20 y se utiliza como primer bloque de datos del segundo ecualizador Turbo, un tercer bloque de datos de salida del segundo ecualizador Turbo 20 se envía a un tercer ecualizador Turbo 20, y se utiliza como primer bloque de datos del tercer ecualizador Turbo, etc. Por lo tanto, el tercer bloque de datos de salida del último ecualizador Turbo 20 se envía a una primera unidad 22 de decodificación de código convolucional LDPC, y se utiliza como
45 segundo bloque de datos para la primera unidad 22 de decodificación de código convolucional LDPC para realizar una decodificación iterativa, con el fin de generar un tercer bloque de datos después de la decodificación iterativa, el tercer bloque de datos de salida de la primera unidad 22 de decodificación de código convolucional LDPC se utiliza como segundo bloque de datos para una segunda unidad 22 de decodificación de código convolucional LDPC para realizar una decodificación iterativa, con el fin de generar un tercer bloque de datos después de la decodificación iterativa, etc.

50 Opcionalmente, el ecualizador Turbo 20 y la unidad 22 de decodificación de código convolucional LDPC en el sistema ecualizador Turbo también pueden estar conectados entre sí de forma entremezclada. Teniendo esto en cuenta se puede deducir que, la salida de un módulo de procesamiento anterior (el ecualizador Turbo 20 o la unidad 22 de decodificación de código convolucional LDPC) se utiliza como la entrada de un módulo de procesamiento posterior (el ecualizador Turbo 20 o la unidad 22 de decodificación de código convolucional LDPC), y se realiza la iteración por turno.
55

Haciendo referencia a la FIG. 8, a continuación se describe detalladamente el principio de funcionamiento del sistema ecualizador Turbo.

Tal como se muestra en la FIG. 8, en el extremo de transmisión, se realiza la codificación utilizando un codificador de código convolucional LDPC, a continuación se realiza la codificación diferencial, y después un modulador óptico envía una señal óptica a una red de transmisión de fibra óptica; y en el extremo de recepción, después de una comprobación coherente, un muestreo de un convertidor analógico digital y un procesamiento de ecualización normal de señal utilizando un ecualizador, la señal óptica entra en el sistema ecualizador Turbo.

El sistema ecualizador Turbo incluye: un ecualizador Turbo primario (esto es, un ecualizador Turbo conectado a un ecualizador común de señal) y M ecualizadores Turbo posteriores, donde la diferencia entre el ecualizador Turbo primario y los ecualizadores Turbo posteriores reside en que son diferentes las formas de establecer un valor de estado de un símbolo inicial durante el funcionamiento de la unidad OP-BCJR. Por ejemplo el valor de estado de un símbolo inicial en una unidad OP-BCJR en el ecualizador Turbo primario es un valor de estado de distribución equiprobable, y el valor de estado de un símbolo inicial en una unidad OP-BCJR en los ecualizadores Turbo posteriores es un valor de estado en un mismo bit que el obtenido desde una operación de la unidad OP-BCJR del nivel anterior y se lee de una memoria. El ecualizador Turbo primario y los ecualizadores Turbo posteriores incluyen ambos una unidad OP-BCJR y una unidad de decodificación de código convolucional LDPC, tal como se muestra en la FIG. 9.

Por otro lado, el sistema ecualizador Turbo que se muestra en la FIG. 8 incluye, además, N unidades de decodificación de código convolucional LDPC independientes. Se puede entender que las posiciones de los M ecualizadores Turbo posteriores y las N unidades de decodificación de código convolucional LDPC en el sistema ecualizador Turbo no están limitadas a aquellas que se muestran en la FIG. 8, y también se puede utilizar una conexión de forma intercalada.

Las FIG. 9 a 11 describen conjuntamente el principio de funcionamiento del ecualizador Turbo.

Tal como se muestra en la FIG. 9, en una unidad de decodificación de código convolucional LDPC, $C_1, C_2, C_3, \dots, C_T$, forman conjuntamente una secuencia de palabra código que tiene que cumplir una relación de comprobación de la k-ésima capa de una matriz de comprobación de un código convolucional LDPC, y la decodificación y el cálculo de información tentativa se realizan de acuerdo con la relación de comprobación de la capa. Al mismo tiempo, una unidad OP-BCJR divide el bloque de datos C_0 recibido en segmentos de acuerdo con un bit de estado en un diagrama de enrejado (trellis). D bits en segmentos adyacentes se solapan, los n segmentos de datos solapados se envían, respectivamente, a n unidades de procesamiento de segmentos (por ejemplo, BPU_1 a BPU_n) para un procesamiento de operación de BCJR (que incluye una operación recursiva hacia delante y/o una operación recursiva hacia atrás; para detalles, véase la FIG. 11 y la descripción asociada).

Después de completar la actualización de la información tentativa de $C_1, C_2, C_3, \dots, C_T$, la unidad de decodificación de código convolucional LDPC envía el bloque de datos C_T a un ecualizador Turbo del siguiente nivel. Al mismo tiempo, el bloque de datos C_0 que se ha procesado se recibe desde una unidad OP-BCJR en un mismo nivel C_0 y los $C_1, C_2, C_3, \dots, C_{T-1}$ que se siguen encontrando en la unidad de codificación del código convolucional LDPC forman conjuntamente una secuencia de palabra código que tiene que cumplir una relación de comprobación de una capa que es una capa mayor que la capa k-ésima de la matriz de comprobación del código convolucional LDPC, y la decodificación y el cálculo de la información tentativa se realizan de acuerdo con la relación de comprobación de la capa.

El proceso de procesamiento iterativo Turbo anterior se representa utilizando un diagrama de secuencia, tal como se muestra en la FIG. 10, y se utiliza un ejemplo en el que $T = 4$.

En un primer momento, en una unidad de decodificación de código convolucional LDPC de un módulo Turbo en el nivel (i-1)-ésimo, C_1, C_2, C_3 y C_4 forman conjuntamente una secuencia de palabra código que necesita cumplir una relación de comprobación de una capa H^3 éxima de una matriz de comprobación H_c de un código convolucional LDPC, y se realiza una decodificación y un cálculo de información tentativa de acuerdo con la relación de comprobación de la capa. Al mismo tiempo, la unidad OP-BCJR que también se encuentra en el (i-1)-ésimo nivel realiza un procesamiento de operación paralelo BCJR sobre un bloque de datos C_0 recibido de acuerdo con los segmentos solapados.

En un segundo momento, la unidad de decodificación de código convolucional LDPC en el nivel (i-1)-ésimo envía el bloque de datos C_4 a un ecualizador Turbo en el nivel i-ésimo. Al mismo tiempo, el bloque de datos C_0 que se ha procesado es recibido desde la unidad OP-BCJR en el mismo nivel, C_0 y C_1, C_2 y C_3 que siguen estando en la unidad de codificador de código convolucional LDPC forman conjuntamente una secuencia de palabra código que tiene que cumplir una relación de comprobación de una capa H^4 de la matriz de comprobación H_c de un código convolucional LDPC, y se realiza una decodificación y un cálculo de información tentativa de acuerdo con la relación de comprobación de la capa.

En la FIG. 11 se muestra un proceso de procesamiento específico de la unidad OP-BCJR paralela solapada. El bloque de datos C_0 se divide con antelación en múltiples segmentos, donde los segmentos se solapan entre sí, y los módulos BPU son responsables de procesar la actualización de información de los segmentos, por ejemplo, los BPU_1, BPU_2 y BPU_3 que están marcados en la parte inferior de la FIG. 11. Los segmentos de bits de los que cada uno de los módulos BPU es responsable y para los que realmente es necesario actualizar la información tentativa posterior son una primera parte de los módulos BPU_1, BPU_2 y BPU_3, siendo respectivamente BPU_1-1, BPU_2-1 y BPU_3-1; y las partes solapadas para las que únicamente es necesario actualizar un valor de estado utilizando el valor de estado obtenido en la iteración anterior por un segmento adyacente son una segunda parte (un valor de estado de un símbolo inicial mostrado en esta parte es un valor de estado hacia delante obtenido en la iteración previa por parte de un segmento previo) y una tercera parte (un valor de estado de un símbolo inicial mostrado en esta parte es un valor de estado hacia atrás obtenido en la iteración previa por un segmento siguiente) en los módulos BPU_1, BPU_2 y BPU_3, siendo respectivamente BPU_1-2, y BPU_2-2 y BPU_3-2, y BPU_1-3, BPU_2-3 y BPU_3-3.

El proceso de procesamiento de una unidad OP-BCJR es como sigue: (1) en cada módulo BPU, desde una memoria, se lee un valor de estado hacia delante de un símbolo inicial (una pequeña caja en blanco sobre el eje de bits) de un segmento de bits (una segunda parte) que se solapa con un segmento anterior, y se lee un valor de estado hacia atrás de un símbolo inicial (una pequeña caja sólida sobre el eje de bits) de un segmento de bits (una tercera parte) que se solapa con un segmento siguiente, donde para una unidad OP-BCJR en el ecualizador Turbo primario, un valor de estado correspondiente al símbolo inicial es un valor de estado de distribución equiprobable; (2) cada módulo BPU realiza una operación recursiva hacia delante solapada (una línea de puntos en la figura) sobre un segmento de bits solapado de la segunda parte, hasta un bit final del segmento de bits de la segunda parte, y realiza una operación recursiva hacia atrás solapada (una línea discontinua en la figura) sobre un segmento de bits solapado de la tercera parte, hasta un bit final del segmento de bits de la tercera parte; (3) utilizando los bits finales de los segmentos de bits de la segunda parte y la tercera parte como símbolos iniciales, cada módulo BPU realiza una operación recursiva hacia delante y una operación recursiva hacia atrás sobre un segmento de bits de una primera parte de la que cada módulo BPU es realmente responsable para su actualización, y calcula una información tentativa posterior de cada bit de acuerdo con los valores de estado hacia delante y hacia atrás obtenidos; y (4) cada módulo BPU necesita almacenar los valores de estado hacia delante y hacia atrás de los símbolos iniciales de la segunda parte y la tercera parte que se solapan con un segmento de bits adyacente, para ser utilizados en una operación del siguiente nivel de la unidad OP-BCJR.

Con el modo de realización de la FIG. 11, lo anterior describe un proceso para realizar un procesamiento recursivo hacia delante y un procesamiento recursivo hacia atrás sobre cada segmento de datos (esto es, un módulo BPU). Se debería entender que, con el fin de simplificar el procesamiento recursivo, también puede ser que únicamente se realice el procesamiento recursivo hacia delante o el procesamiento recursivo hacia atrás sobre cada segmento de datos (esto es, el módulo BPU); o que se realice el procesamiento recursivo hacia delante sobre algunos segmentos de datos, y se realice el procesamiento recursivo hacia atrás sobre los otros segmentos de datos.

Por lo tanto, en este modo de realización, mediante la realización, en una unidad OP-BCJR, del procesamiento de segmentación y las operaciones recursivas hacia delante y hacia atrás sobre un bloque de datos recibido, y la realización, en una unidad de decodificación de código convolucional LDPC, del procesamiento iterativo Turbo sobre los datos obtenidos de la unidad OP-BCJR, se mejora de forma efectiva el rendimiento de la compensación de ecualización Turbo y se reducen los recursos de almacenamiento necesarios.

La FIG. 12 muestra otro modo de realización específico de un sistema ecualizador Turbo de acuerdo con un modo de realización de la presente invención. Es necesario que una señal de salida de un ecualizador en la técnica anterior pase a través de una unidad de estimación de canal (la cual es un estimador de distribución de probabilidad de transición condicional en la FIG. 12), para entrar en un ecualizador Turbo primario únicamente después de haber determinado un parámetro de estimación de canal (por ejemplo, un parámetro de distribución de probabilidad PDF, un parámetro de probabilidad de transición, etc., de un canal). Por lo tanto, es necesario estimar, de acuerdo con una secuencia de entrenamiento en el sistema, la distribución de probabilidad de transición condicional que la unidad OP-BCJR tiene que utilizar en el ecualizador Turbo primario. En otras palabras, se compensa el daño producido en un canal de fibra óptica mediante un efecto no lineal y un efecto PMD.

Obviamente, en este modo de realización, mediante la realización, en una unidad OP-BCJR, del procesamiento de segmentación y/o las operaciones recursivas hacia delante y hacia atrás sobre un bloque de datos recibido, y mediante la realización, en una unidad de decodificación de código convolucional LDPC, de un procesamiento Turbo iterativo sobre los datos obtenidos de la unidad OP-BCJR, se mejora el rendimiento de la compensación de ecualización Turbo, se reducen los recursos de almacenamiento necesarios y se puede compensar el daño producido en un canal de fibra óptica mediante un efecto no lineal y un efecto PMD.

Se debería entender que, la solución descrita en cada una de las reivindicaciones de la presente invención también se debería considerar como un modo de realización, y las características en las reivindicaciones se pueden combinar, por ejemplo, los diferentes pasos de bifurcación realizados después de determinar pasos en la presente invención se pueden utilizar como diferentes modos de realización.

5 Una persona con un conocimiento normal en la técnica puede ser consciente de que, en combinación con los ejemplos descritos en los modos de realización divulgados en esta memoria descriptiva, se pueden implementar unidades y pasos de algoritmo mediante hardware electrónico o una combinación de software informático y hardware electrónico. El que las funciones se realizan mediante hardware o software depende de las aplicaciones concretas y las condiciones de limitación de diseño de las soluciones técnicas. Una persona experimentada en la
10 técnica puede utilizar diferentes métodos para implementar las funciones descritas para cada aplicación concreta, pero no se debería considerar que la implementación va mas allá del alcance de la presente invención.

Una persona experimentada en la técnica debería entender claramente que, para el propósito de una descripción conveniente y breve, para un proceso de trabajo detallado del sistema, equipo y unidad anteriores, se puede hacer referencia a un proceso correspondiente en los modos de realización del método anterior, y no se vuelven
15 a describir los detalles en la presente solicitud.

En los varios modos de realización proporcionados en la presente solicitud, se debería entender que el sistema, equipo y método divulgados se pueden implementar de otras formas. Por ejemplo, el modo de realización del equipo descrito es únicamente un ejemplo. Por ejemplo, la división en unidades es únicamente una división de
20 función lógica y puede tener otra división en una implementación real. Por ejemplo, una pluralidad de unidades o componentes se puede combinar o integrar en otro sistema, o algunas características se pueden ignorar o no llevar a cabo. Además, los acoplamientos mutuos o acoplamientos directos o conexiones de comunicación mostrados o discutidos se pueden implementar mediante algunas interfaces. Los acoplamientos indirectos o las conexiones de comunicación entre los equipos o unidades se pueden implementar de forma electrónica, mecánica u otras formas.

25 Las unidades descritas como componentes separados pueden estar o no separados físicamente, y los componentes mostrados como unidades pueden ser o no unidades físicas, pueden estar localizados en una posición, o se pueden distribuir en una pluralidad de unidades de red. Algunas o todas las unidades se pueden seleccionar de acuerdo con las necesidades reales para conseguir los objetivos de las soluciones de los modos de realización.

30 Además, las unidades funcionales en los modos de realización de la presente invención pueden estar integradas en una unidad de procesamiento, o cada una de las unidades puede existir físicamente separadas, o dos o más unidades se encuentran integradas en una unidad.

Cuando las funciones se implementan en forma de una unidad funcional software y se comercializa o utiliza como un producto independiente, las funciones pueden encontrarse almacenadas en un medio de almacenamiento legible por un ordenador. Basándose en dicho conocimiento, las soluciones técnicas de la presente invención esencialmente, o la parte que contribuye a la técnica anterior, o algunas de las soluciones técnicas se pueden
35 implementar en forma de producto software. El producto software está almacenado en un medio de almacenamiento, e incluye varias instrucciones para gestionar un dispositivo informático (el cual puede ser un ordenador personal, un servidor o un dispositivo de red) con el fin de realizar todos o algunos de los pasos de los métodos descritos en los modos de realización de la presente invención. El medio de almacenamiento anterior incluye: cualquier medio que pueda almacenar código de programa como, por ejemplo, un disco flash USB, un disco duro extraíble, una memoria de sólo lectura (ROM, Read-Only Memory), una memoria de acceso aleatorio (RAM, Random Access Memory), un disco magnético, o un disco óptico.
40

Las descripciones anteriores son únicamente formas de implementación específicas de la presente invención, y no pretenden limitar el alcance de protección de la presente invención. Cualquier variación o sustitución fácilmente entendida por una persona experimentada en la técnica dentro del alcance técnico divulgado en la presente invención se considerará dentro del alcance de protección de la presente invención. Por lo tanto, el
45 alcance de protección de la presente invención estará sujeto al alcance de protección de las reivindicaciones.

REIVINDICACIONES

1. Un método para implementar una compensación de ecualización Turbo en un sistema ecualizador Turbo (60, 70) que comprende múltiples ecualizadores Turbo (20, 50), comprendiendo cada ecualizador Turbo (20, 50)

5 una unidad BCJR paralela solapada, OP-BCJR, (21) para realizar un procesamiento de operación paralela de acuerdo con segmentos solapados y una unidad (22) de decodificación de código convolucional de comprobación de paridad de baja densidad, LDPC, para decodificación iterativa, comprendiendo el método los pasos siguientes realizados por cada uno de los múltiples ecualizadores Turbo (20, 50):

10 dividir, por parte de la unidad OP-BCJR (21), un primer bloque de datos en n segmentos de datos, en donde D bits en dos segmentos de datos adyacentes en los n segmentos de datos se solapan, n es un entero positivo mayor o igual que 2, y D es un entero positivo mayor o igual que 1, realizar de forma concurrente, por parte de la unidad OP-BCJR (21), un procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos, y agrupar, por parte de la unidad OP-BCJR (21), los n segmentos de datos sobre los que se ha realizado el procesamiento recursivo, con el fin de obtener un segundo bloque de datos; y

15 realizar, por parte de la unidad (22) de decodificación de código convolucional LDPC, una decodificación iterativa sobre el segundo bloque de datos y los otros $T-1$ bloques de datos, los cuales conjuntamente forman una secuencia de palabra código de T bloques de palabra de código correspondientes a una relación de comprobación de una i -ésima capa H_i de una matriz de comprobación H de un código convolucional LDPC, para generar un tercer bloque de datos,

20 en donde las longitudes de datos del primer bloque de datos, el segundo bloque de datos, y el tercer bloque de datos son todas de $1/T$ de la longitud de código N de la secuencia de palabra código del código convolucional LDPC, y T se determina mediante un parámetro de estructura escalonada de la matriz de comprobación H del código convolucional LDPC con $T=N/N_T$, donde N_T y N son constantes y representan la cantidad de columnas espaciadas entre sí entre la i -ésima capa H_i y la $(i+1)$ -ésima capa H_{i+1} de la matriz de comprobación H del código convolucional LDPC y la cantidad de columnas de cada capa H_i de la matriz de comprobación H del código convolucional LDPC, respectivamente; y

en donde el tercer bloque de datos de salida de un primer ecualizador Turbo (20) se le envía a un segundo ecualizador Turbo (20) y se utiliza como primer bloque de datos del segundo ecualizador Turbo (20).

30 2. El método de acuerdo con la reivindicación 1, en donde la realización del procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos comprende:

realizar concurrentemente una operación recursiva hacia delante sobre cada segmento de datos de los n segmentos de datos y realizar concurrentemente una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos.

35 3. El método de acuerdo con la reivindicación 1, en donde la realización del procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos comprende:

realizar concurrentemente una operación recursiva hacia delante sobre cada segmento de datos de los n segmentos de datos.

4. El método de acuerdo con la reivindicación 1, en donde la realización del procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos comprende:

40 realizar concurrentemente una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos.

5. El método de acuerdo con una cualquiera de las reivindicaciones 1 a 4, en donde la realización de la decodificación iterativa sobre el segundo bloque de datos, para generar un tercer bloque de datos comprende:

recibir el segundo bloque de datos;

45 realizar un procesamiento de decodificación sobre el segundo bloque de datos recibido y los otros $T-1$ bloques de datos sobre los que se ha realizado la decodificación iterativa, en donde una longitud de datos de cada uno de los otros $T-1$ bloques de datos sobre los que se ha realizado la decodificación iterativa es $1/T$ de la longitud de código del código convolucional LDPC; y

50 generar el tercer bloque de datos sobre el que el procesamiento de decodificación se ha realizado un número máximo de veces.

6. El método de acuerdo con una cualquiera de las reivindicaciones 1 a 5, en donde antes de dividir un primer bloque de datos en n segmentos de datos, el método comprende, además:

realizar una estimación de distribución de probabilidad de transición condicional sobre el primer bloque de datos, para determinar una información de parámetros de estimación de canal.

5 7. Un sistema ecualizador Turbo (60, 70), que comprende múltiples ecualizadores Turbo (20, 50), comprendiendo cada ecualizador Turbo (20, 50):

10 una unidad BCJR paralela solapada, OP-BCJR, (21) para realizar un procesamiento de operación paralela sobre un primer bloque de datos de acuerdo con segmentos solapados, configurada para dividir el primer bloque de datos en n segmentos de datos, en donde D bits en dos segmentos de datos adyacentes en los n segmentos de datos se solapan, n es un entero positivo mayor o igual que 2, y D es un entero positivo mayor o igual que 1, realizar de forma concurrente un procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos, y agrupar los n segmentos de datos sobre los que se ha realizado el procesamiento recursivo, con el fin de obtener un segundo bloque de datos; y

15 una unidad (22) de decodificación de código convolucional de comprobación de paridad de baja densidad, LDPC, conectada a la unidad OP-BCJR (21), y configurada para realizar una decodificación iterativa sobre el segundo bloque de datos y los otros $T-1$ bloques de datos, los cuales conjuntamente forman una secuencia de palabra código de T bloques de palabra de código correspondientes a una relación de comprobación de una i -ésima capa H_i de una matriz de comprobación H de un código convolucional LDPC, para generar un tercer bloque de datos,

20 en donde las longitudes de datos del primer bloque de datos, el segundo bloque de datos, y el tercer bloque de datos son todas de $1/T$ de la longitud de código N de la secuencia de palabra código del código convolucional LDPC, y T se determina mediante un parámetro de estructura escalonada de la matriz de comprobación H del código convolucional LDPC con $T=N/N_T$, donde N_T y N son constantes y representan la cantidad de columnas espaciadas entre sí entre la i -ésima capa H_i y la $(i+1)$ -ésima capa H_{i+1} de la matriz de comprobación H del código convolucional LDPC y la cantidad de columnas de cada capa H_i de la matriz de comprobación H del código convolucional LDPC, respectivamente; y

en donde el tercer bloque de datos de salida de un primer ecualizador Turbo (20) se le envía a un segundo ecualizador Turbo (20) y se utiliza como primer bloque de datos del segundo ecualizador Turbo (20).

30 8. El sistema ecualizador Turbo (60, 70) de acuerdo con la reivindicación 7, en donde la unidad OP-BCJR (21) de cada ecualizador Turbo (20, 50) comprende:

un módulo de segmentación (211), configurado para dividir el primer bloque de datos en los n segmentos de datos, en donde D bits en dos segmentos de datos adyacentes en los n segmentos de datos se solapan, n es un entero positivo mayor o igual que 2, y D es un entero positivo mayor o igual que 1;

35 un módulo de recursión (212), configurado para realizar un procesamiento recursivo sobre cada segmento de datos de los n segmentos de datos; y

un módulo de agrupación (213), configurado para agrupar los n segmentos de datos sobre los que se ha realizado el procesamiento recursivo, con el fin de obtener el segundo bloque de datos.

9. El sistema ecualizador Turbo (60, 70) de acuerdo con la reivindicación 8, en donde el módulo de recursión (212) está configurado para:

40 realizar concurrentemente una operación recursiva hacia delante sobre cada segmento de datos de los n segmentos de datos y realizar concurrentemente una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos.

10. El sistema ecualizador Turbo (60, 70) de acuerdo con la reivindicación 8, en donde el módulo de recursión (212) está configurado para:

45 realizar concurrentemente una operación recursiva hacia delante sobre cada segmento de datos de los n segmentos de datos.

11. El sistema ecualizador Turbo (60, 70) de acuerdo con la reivindicación 8, en donde el módulo de recursión (212) está configurado para:

50 realizar concurrentemente una operación recursiva hacia atrás sobre cada segmento de datos de los n segmentos de datos.

12. El sistema ecualizador Turbo (60, 70) de acuerdo con una cualquiera de las reivindicaciones 8 a 11, en donde la unidad (22) de decodificación de código convolucional LDPC de cada ecualizador Turbo (20, 50) comprende:

un módulo de recepción (221), configurado para recibir el segundo bloque de datos;

5 un módulo de decodificación (222), configurado para realizar un procesamiento de decodificación sobre el segundo bloque de datos recibido y los otros T-1 bloques de datos sobre los que se ha realizado la decodificación iterativa, en donde una longitud de datos de cada uno de los otros T-1 bloques de datos sobre los que se ha realizado la decodificación iterativa es $1/T$ de la longitud de código del código convolucional LDPC; y

10 un módulo de salida (223), configurado para generar el tercer bloque de datos sobre el que el procesamiento de decodificación se ha realizado un número máximo de veces.

13. El sistema ecualizador Turbo (60, 70) de acuerdo con una cualquiera de las reivindicaciones 7 a 12, comprendiendo cada ecualizador Turbo (50), además:

15 una unidad (23) de estimación de canal, configurada para: antes de que la unidad OP-BCJR (21) divida el primer bloque de datos en los n segmentos de datos, realizar una estimación de distribución de probabilidad de transición condicional sobre el primer bloque de datos, con el fin de determinar una información de parámetros de estimación de canal.

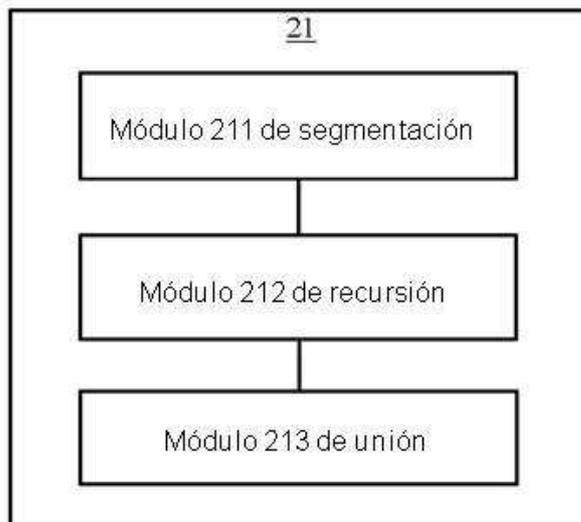
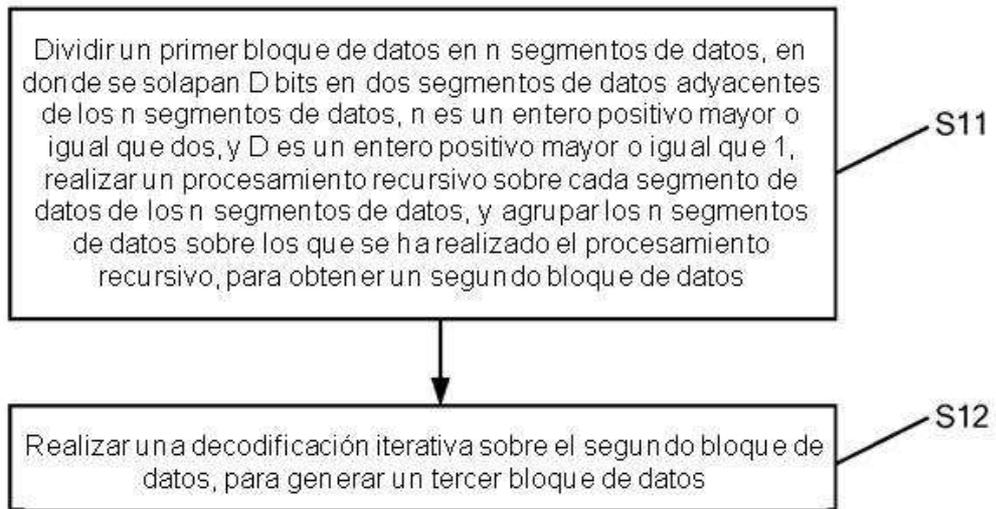
14. El sistema ecualizador Turbo (70) de acuerdo con la reivindicación 7, que comprende, además:

20 una primera unidad (22) independiente de decodificación de código convolucional LDPC, en donde la primera unidad (22) independiente de decodificación de código convolucional LDPC recibe el tercer bloque de datos de salida del último ecualizador Turbo (20) de los múltiples ecualizadores Turbo (20) y realiza una decodificación iterativa sobre el tercer bloque de datos, para generar un cuarto bloque de datos, en donde la longitud de datos del cuarto bloque de datos es $1/T$ de la longitud de código del código convolucional LDPC.

15. El sistema ecualizador Turbo (70) de acuerdo con la reivindicación 14, que comprende, además:

25 una segunda unidad (22) independiente de decodificación de código convolucional LDPC, en donde la segunda unidad (22) independiente de decodificación de código convolucional LDPC recibe el cuarto bloque de datos de salida de la primera unidad (22) independiente de decodificación de código convolucional LDPC, y realiza una decodificación iterativa sobre el cuarto bloque de datos, para generar un quinto bloque de datos, en donde la longitud de datos del quinto bloque de datos es $1/T$ de la longitud de código del código convolucional LDPC.

30



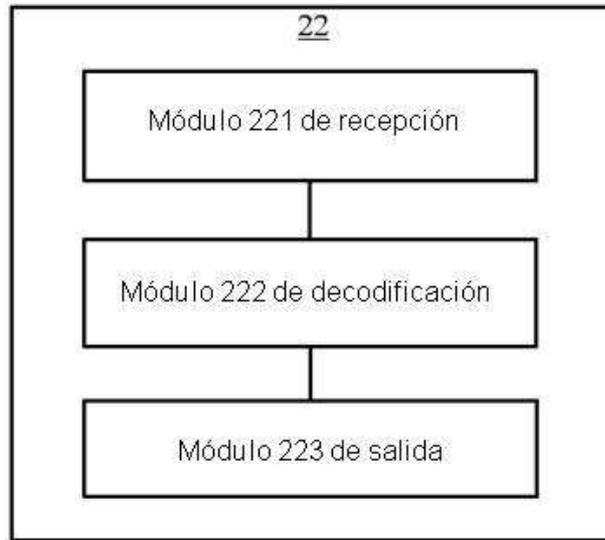


FIG. 4



FIG. 5

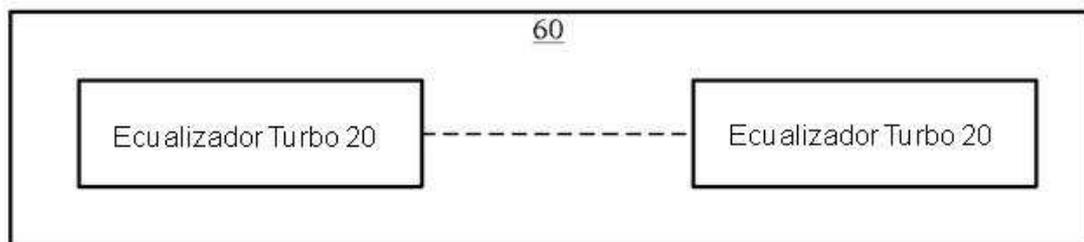


FIG. 6

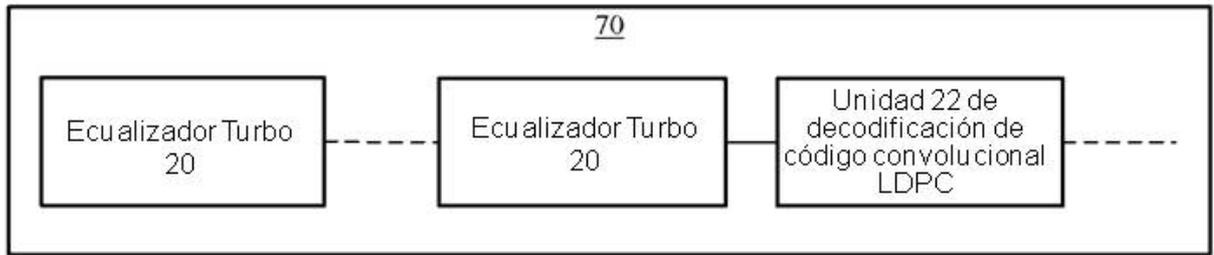


FIG. 7

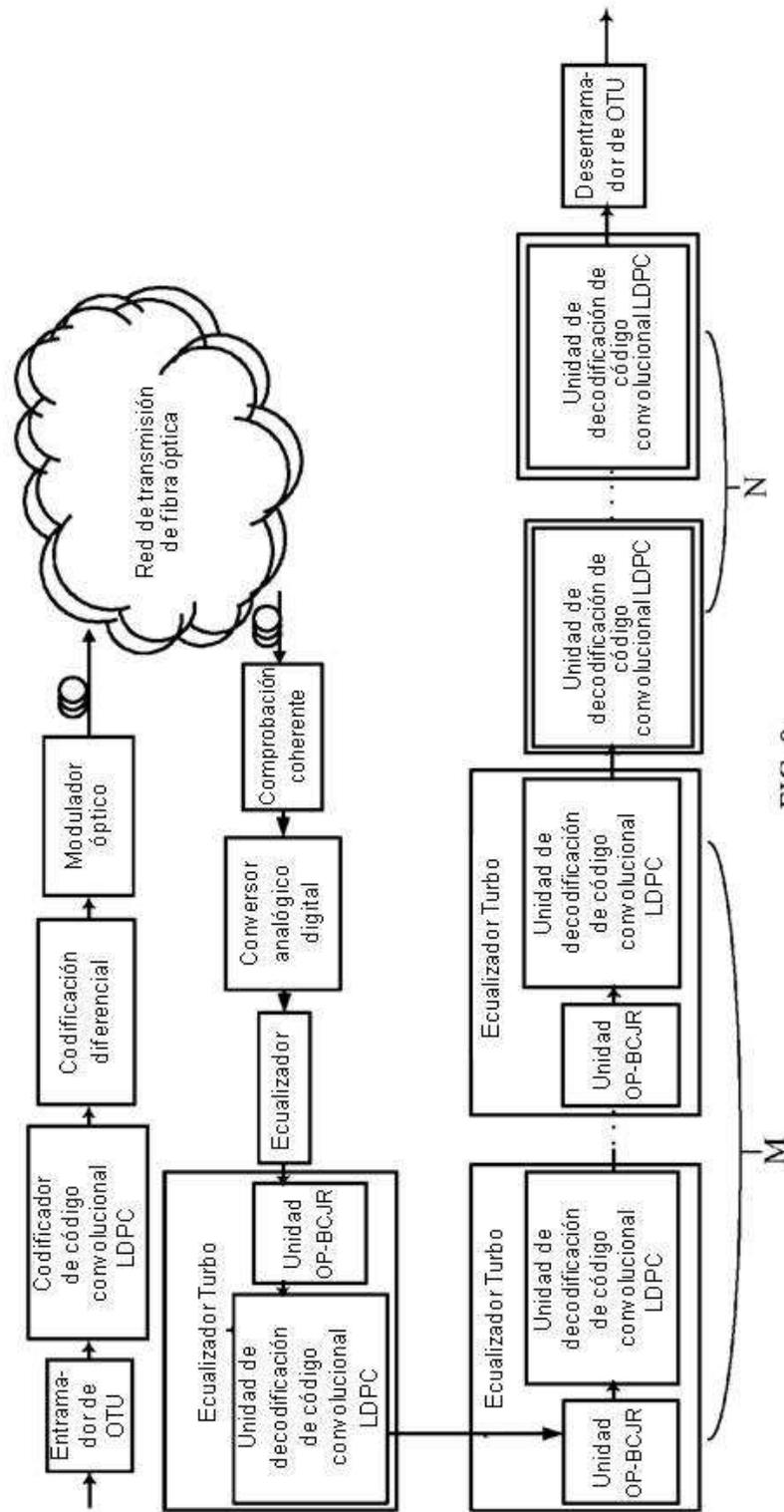


FIG. 8

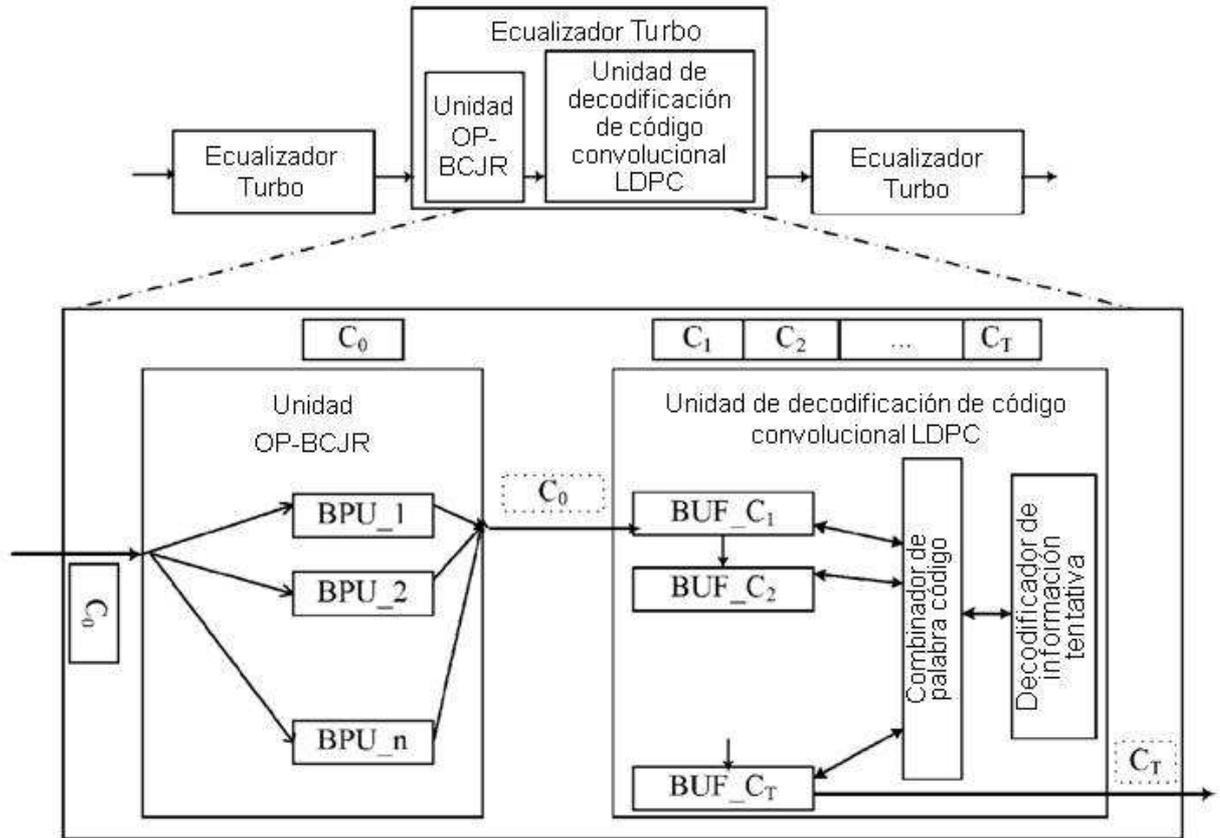
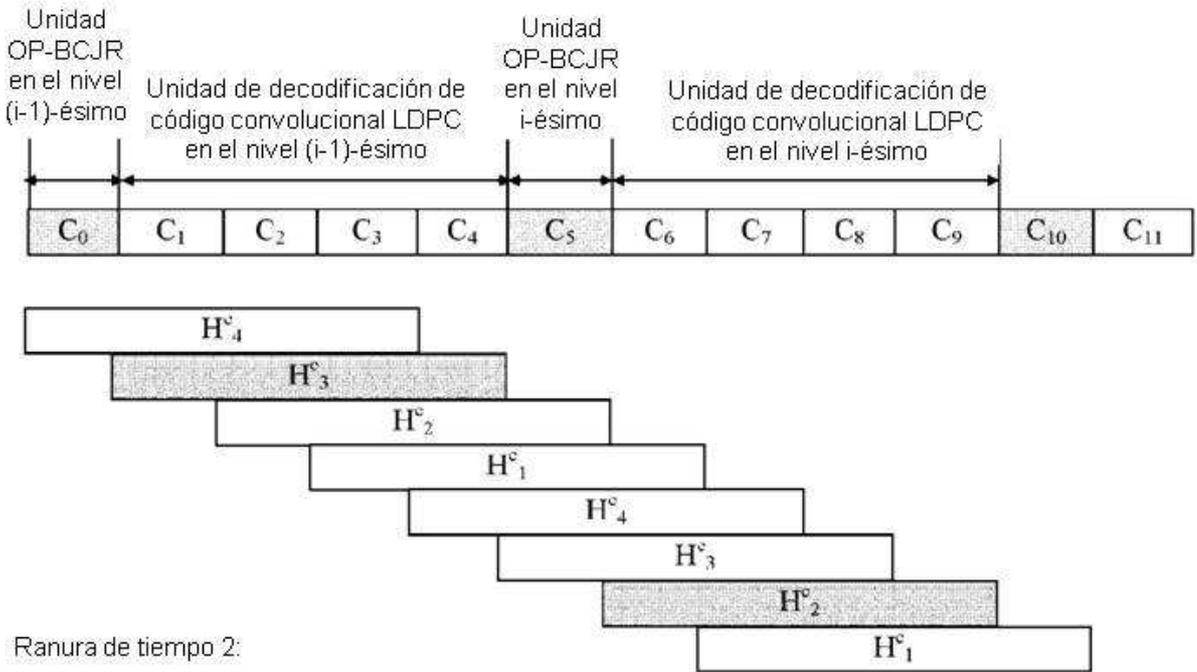


FIG. 9

Ranura de tiempo 1:



Ranura de tiempo 2:

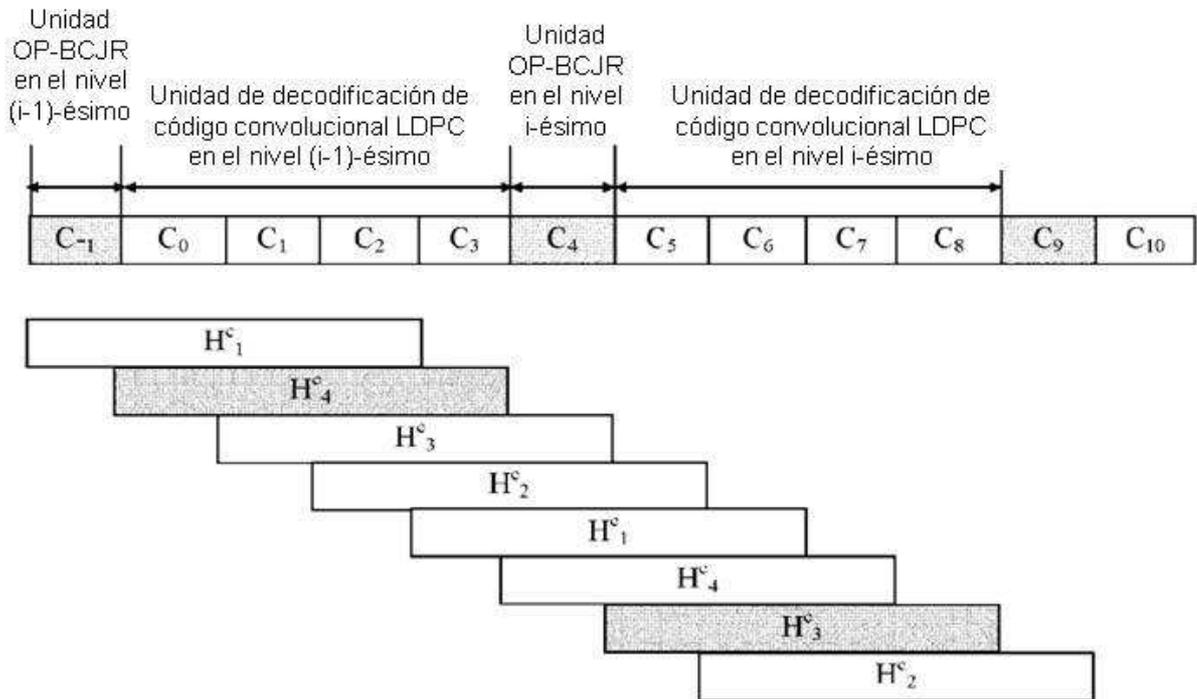


FIG. 10

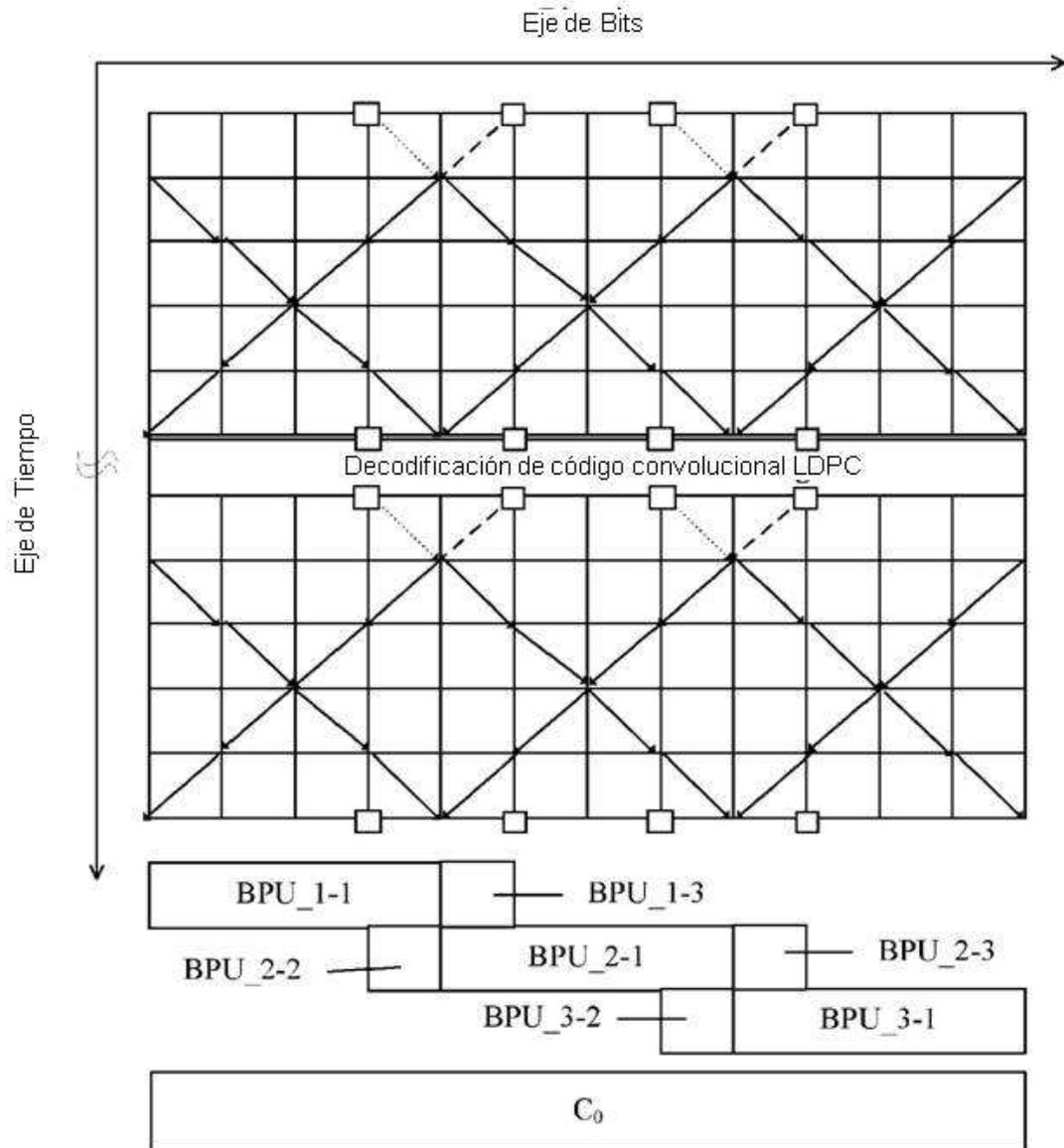


FIG. 11

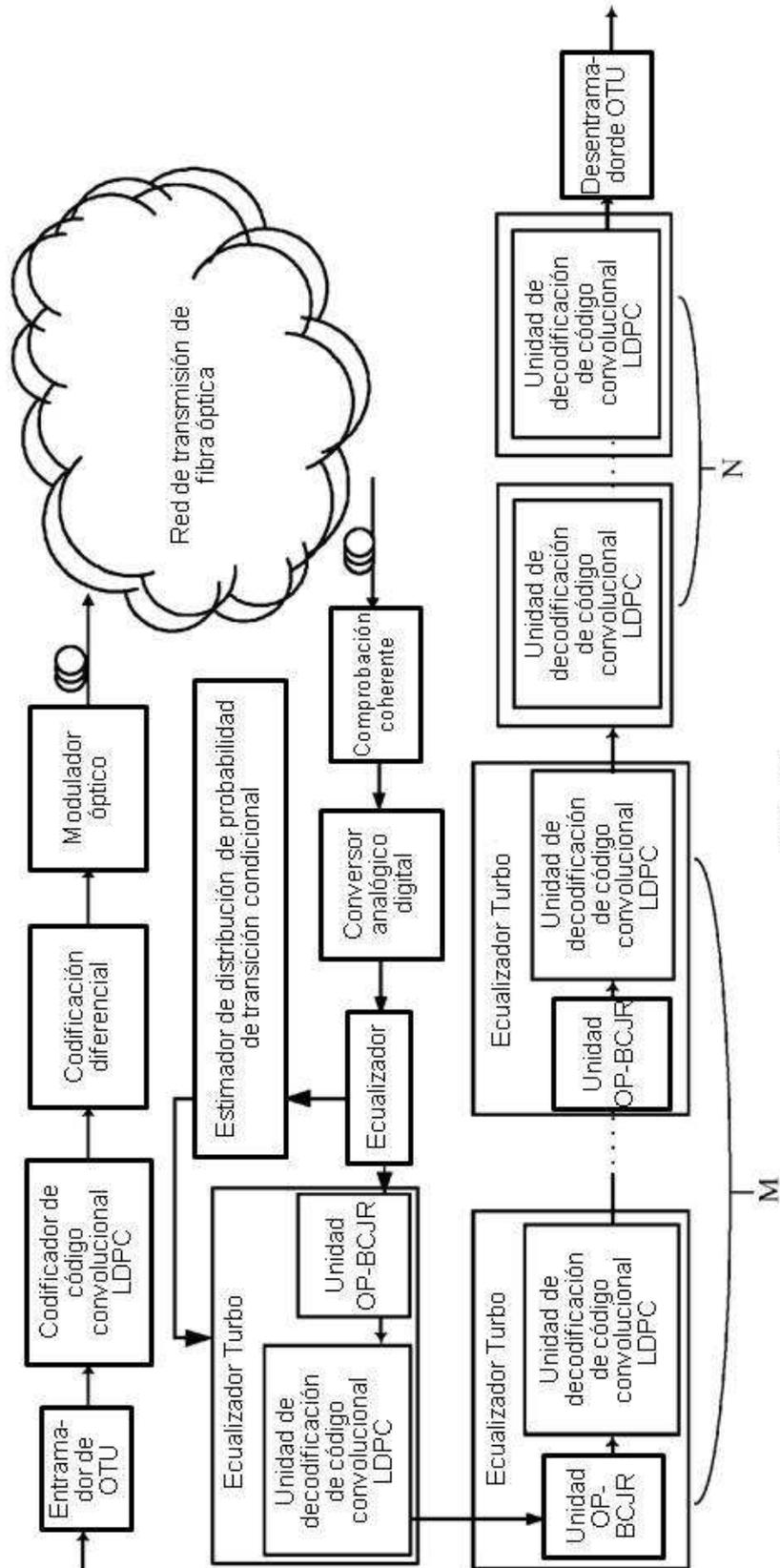


FIG. 12