

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 852 748**

51 Int. Cl.:

H04L 1/00 (2006.01)

H04L 7/04 (2006.01)

H04L 12/24 (2006.01)

H04J 3/06 (2006.01)

H03M 13/15 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **09.07.2015 PCT/CN2015/083693**

87 Fecha y número de publicación internacional: **27.10.2016 WO16169133**

96 Fecha de presentación y número de la solicitud europea: **09.07.2015 E 15889622 (5)**

97 Fecha y número de publicación de la concesión europea: **16.12.2020 EP 3264652**

54 Título: **Método de tratamiento de datos y extremo de transmisión de datos**

30 Prioridad:

23.04.2015 WO PCT/CN2015/077319

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

14.09.2021

73 Titular/es:

**HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building, Huawei
Industrial Base, Bantian, Longgang District
Shenzhen 518129, CN**

72 Inventor/es:

**YANG, WENBIN;
WANG, TONGTONG y
WANG, XINYUAN**

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 852 748 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 tratamiento de datos y extremo de transmisión de datos

Campo técnico

5 Las realizaciones de la presente invención se refieren al campo de las comunicaciones y, en particular, a un método de tratamiento de datos y a un extremo de transmisión de datos.

Antecedentes

Con el desarrollo de Internet, el tráfico de una red troncal de telecomunicaciones aumenta de manera explosiva a una velocidad del 50 % al 80 % cada año, y existe una necesidad cada vez mayor de ancho de banda en una interfaz de Ethernet en el futuro.

10 Las arquitecturas propuestas en los estándares de Ethernet actuales son principalmente arquitecturas descritas en estándares tales como el IEEE 802.3ba / bj / bm. Las capas físicas de todas estas arquitecturas utilizan una interfaz de ancho de banda de 100 G para realizar la codificación y decodificación mediante corrección de errores de reenvío (Forward Error Correction, FEC, en inglés). En la técnica anterior, una arquitectura de interfaz de Ethernet utiliza un ancho de banda de tratamiento de 100 G, y los módulos principales en una subcapa física de codificación (Physical Coding Sublayer, PCS, en inglés) tienen un módulo de codificación y decodificación, un módulo de aleatorización, un módulo de suministro, un módulo de inserción de marcador de alineación, y similares. Esta solución está diseñada para un ancho de banda de 100 G.

20 En una arquitectura de Ethernet de la técnica anterior, los datos de entrada en la capa PCS que utiliza una velocidad de transmisión de 100 G son datos paralelos en una interfaz independiente de medios de Ethernet de 100 G (Century Gigabit Media Independent Interface, CGMII, en inglés) en un formato de subcapa de reconciliación (Reconciliation sublayer, RS, en inglés). Los datos en múltiples carriles se envían y, a continuación, los datos se envían a una subcapa física de conexión a medio (Physical Medium Attachment Sublayer, PMA, en inglés). Las funciones principales de un lado de recepción y un lado de transmisión en la capa PCS pueden estar basadas en la extensión de un estándar de 100 G. Un extremo de transmisión de 100 G realiza, en primer lugar, la codificación de 64b/66b de los datos y, a continuación, codifica y suministra los datos. Para resolver los problemas de alineación en carriles y el desorden entre carriles, el extremo de transmisión de 100 G necesita insertar códigos de paridad entrelazada de bits (Bit Interleaved Parity, BIP, en inglés) como marcadores de alineación (Alignment Marker, AM, en inglés) y suministra los datos a los carriles de la PMA después de insertar los AM. Un extremo de recepción de 100 G recibe, en primer lugar, los datos, realiza la alineación y el bloqueo de bloques y el bloqueo y la alineación de AM, a continuación, realiza la reasignación de carriles utilizando los AM, elimina los AM después de completar la reasignación de los carriles y, finalmente, completa la decodificación de 64b/66b. Es decir, en la técnica anterior citada anteriormente, el extremo de recepción de 100 G completa la reasignación de los carriles de acuerdo con los AM después de realizar la alineación y el bloqueo de los AM; por lo tanto, los AM pueden ser identificados por el extremo de recepción solo cuando se garantiza que los patrones de AM están completos.

35 Una arquitectura de Ethernet existente puede transmitir un AM completo cuando sea aplicable a un ancho de banda de tratamiento de 100 G; sin embargo, una estructura de la arquitectura anterior es simple, solo están soportados escenarios limitados, y la arquitectura no puede ser adaptada de manera flexible y no puede ser extendida a un ancho de banda alto. Si una interfaz de Ethernet utiliza una velocidad más alta (por ejemplo, 400 G), es probable que se produzcan problemas tales como que el ancho de banda de transmisión de un serializador / deserializador (nombre en inglés: Serdes) no coincide y que los AM no pueden ser alineados. En particular, después de introducir la corrección de errores de reenvío (Forward Error Correction, FEC, en inglés) en Ethernet, los AM originales pueden estar desorganizados; como resultado, el extremo de recepción no puede identificar correctamente los AM y, por lo tanto, no puede completar la reasignación de los carriles, y se produce el problema de que el extremo de recepción no puede realizar correctamente la decodificación.

45 MARK GUSTLIN – XILINX, JERRY PEPPER - IXIA, ANDRE SZCZEPANEK – INPHI, STEVE TROWBRIDGE - ALU, "802.3bj FEC Overview and Status 400 GbE PCS Options; gustlin_3bs_04_0714", BORRADOR del IEEE; GUSTLIN_3BS_04_0714, IEEE-SA, PISCATAWAY, Nueva Jersey, EE.UU., (20140724), vol. 802.3, páginas 1 - 18, XP068080604, da a conocer las opciones de PCS de 400 GbE para su uso con diversos PMD.

50 El documento US 2014/376566 A1 da a conocer un mecanismo por el cual las capas PCS y FEC se combinan en una sola capa en un solo chip, y el número de carriles se establece en cuatro carriles.

El documento XP011305453 de H. TOYODA ET AL., "100GbE PHY and MAC layer implementations", IEEE COMMUNICATIONS MAGAZINE, IEEE SERVICE CENTER, PISCATAWAY, EE. UU., (20100301), vol. 48, no. 3, ISSN 0163-6804, páginas S41 - S47, analiza una implementación lógica del control de acceso a medios y de la capa física de Ethernet de 100 Gb/s.

55 El documento CN 103 534 971 A da a conocer un método de tratamiento de datos codificado y decodificado mediante FEC y un dispositivo relacionado.

Compendio

La presente invención está definida en las reivindicaciones adjuntas.

Se puede ver a partir de las soluciones técnicas anteriores que las realizaciones de la presente invención tienen las siguientes ventajas:

- 5 En las realizaciones de la presente invención, los marcadores de alineación, AM, son insertados en un primer flujo de datos, donde el primer flujo de datos es un flujo de datos que está transcodificado y aleatorizado después de ser codificado en una capa física, y la velocidad de transmisión del primer flujo de datos es mayor o igual a 100 G; el primer flujo de datos que incluye los AM es asignado de manera adaptativa a múltiples carriles de la subcapa física de codificación, PCS, para obtener segundos flujos de datos; la codificación mediante corrección de errores de reenvío, FEC, se realiza en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos; y los terceros flujos de datos son enviados a múltiples carriles de la subcapa física de conexión a medio, PMA, de acuerdo con un ancho de bit de entrada de un serializador / deserializador, Serdes, donde un tercer flujo de datos suministrado a cada carril de la PMA incluye un AM completo y continuo. Debido a que, en la presente invención, los terceros flujos de datos son suministrados a los múltiples carriles de la PMA, y el AM en cada carril de la PMA puede seguir siendo continuo y completo, un extremo de recepción de datos puede obtener AM completos y continuos de los múltiples carriles de la PMA, lo que puede ser aplicable a un escenario en el que la velocidad de transmisión de un flujo de datos es bastante alta, y evitar un daño a los AM.

Breve descripción de los dibujos

20 La figura 1 es un diagrama de bloques esquemático de un procedimiento de un método de transmisión de datos, de acuerdo con una realización de la presente invención;

la figura 2-a es un diagrama esquemático de una forma de implementación de asignar de manera adaptativa un primer flujo de datos a múltiples carriles de la PCS, de acuerdo con una realización de la presente invención;

la figura 2-b es un diagrama esquemático de otra forma de implementación de la asignación adaptativa de un primer flujo de datos a múltiples carriles de la PCS, de acuerdo con una realización de la presente invención;

25 la figura 3 es un diagrama esquemático de módulos de función para realizar la codificación mediante FEC en segundos flujos de datos, de acuerdo con una realización de la presente invención;

la figura 4-a es un diagrama esquemático de un modo de tratar la concatenación de dos segmentos de palabra de código, de acuerdo con una realización de la presente invención;

30 la figura 4-b es un diagrama esquemático de otro modo de tratar la concatenación de dos segmentos de palabra de código, de acuerdo con una realización de la presente invención;

la figura 4-c es un diagrama esquemático de otro modo de tratar la concatenación de dos segmentos de palabra de código, de acuerdo con una realización de la presente invención;

la figura 4-d es un diagrama esquemático de otro modo de tratar la concatenación de dos segmentos de palabra de código, de acuerdo con una realización de la presente invención;

35 la figura 4-e es un diagrama esquemático de una forma de implementación de un primer circuito de codificación mediante FEC y un segundo circuito de codificación mediante FEC, de acuerdo con una realización de la presente invención;

40 la figura 4-f es un diagrama esquemático de otra forma de implementación de un primer circuito de codificación mediante FEC y un segundo circuito de codificación mediante FEC, de acuerdo con una realización de la presente invención;

la figura 4-g es un diagrama esquemático de otra forma de implementación de un primer circuito de codificación mediante FEC y un segundo circuito de codificación mediante FEC, de acuerdo con una realización de la presente invención;

45 la figura 4-h es un diagrama esquemático de otra forma de implementación de un primer circuito de codificación mediante FEC y un segundo circuito de codificación mediante FEC, de acuerdo con una realización de la presente invención;

la figura 5 es un diagrama esquemático de la variación del ancho de bit de una Gearbox común en la técnica anterior;

la figura 6 es un diagrama esquemático de un modo de implementación de suministro de terceros flujos de datos a múltiples carriles de la PMA, de acuerdo con una realización de la presente invención;

la figura 7 es un diagrama esquemático de un modo de manejar un desajuste entre anchos de bit de módulos en una PCS, de acuerdo con una realización de la presente invención;

la figura 8 es un diagrama de bloques esquemático de un procedimiento de otro método de transmisión de datos, de acuerdo con una realización de la presente invención;

5 la figura 9 es un diagrama esquemático de la arquitectura de un extremo de transmisión de datos y un extremo de recepción de datos, de acuerdo con una realización de la presente invención;

la figura 10 es un diagrama esquemático de conversión realizada por un módulo de cambios de 640 bits a 680 bits, de acuerdo con una realización de la presente invención;

10 la figura 11 es un diagrama estructural, esquemático, de la composición de un extremo de transmisión de datos, de acuerdo con una realización de la presente invención. y

la figura 12 es un diagrama estructural, esquemático, de la composición de un extremo de recepción de datos, de acuerdo con una realización de la presente invención.

Descripción de las realizaciones

15 Las realizaciones de la presente invención dan a conocer un método de tratamiento de datos y un extremo de transmisión de datos, que puede transmitir un flujo de datos en un caso en el que se garantiza que un AM no esté dañado, donde el extremo de recepción de datos puede decodificar correctamente el AM.

20 Para hacer que los objetivos, características y ventajas de la presente invención sean más claros y comprensibles, lo siguiente describe clara y completamente las soluciones técnicas en las realizaciones de la presente invención con referencia a los dibujos adjuntos en las realizaciones de la presente invención. Aparentemente, las realizaciones descritas a continuación son simplemente una parte y no todas las realizaciones de la presente invención.

25 En la memoria descriptiva, las reivindicaciones y los dibujos adjuntos de la presente invención, los términos “primero”, “segundo”, etc. pretenden distinguir entre objetos similares, pero no necesariamente indican un orden o secuencia específicos. Se debe entender que los términos utilizados de dicha manera son intercambiables en circunstancias adecuadas, que es simplemente una forma de discriminación que se utiliza cuando se describen objetos que tienen un mismo atributo en las realizaciones de la presente invención. Además, los términos “incluir”, “contener” y cualquier otra variante significan cubrir la inclusión no exclusiva, de modo que un proceso, método, sistema, producto o dispositivo que incluye una serie de unidades no esté limitado necesariamente a esas unidades, sino que pueda incluir otras unidades no enumeradas expresamente o inherentes a dicho proceso, método, sistema, producto o dispositivo.

A continuación, se proporcionan descripciones detalladas de manera separada.

30 Se introduce un mecanismo de codificación y decodificación mediante FEC en Ethernet. Si se introduce un estándar de ancho de banda superior a 400 G en Ethernet, una arquitectura de Ethernet completa varía mucho con respecto a una generación anterior. Sin embargo, una arquitectura de Ethernet existente solo puede ser aplicable a un ancho de banda de 100 G o menos, y, si se utiliza un estándar de 400 G, no se puede lograr un efecto óptimo en los siguientes aspectos: adaptación flexible de módulos en diversas capas, reducción de la ocupación de recursos, acortamiento del retardo de un sistema completo y mejora del rendimiento del sistema completo. En la presente invención, es necesario restablecer la arquitectura de Ethernet y, en particular, es necesario rediseñar una capa PCS para hacer que la adaptación de una red completa sea más flexible. Por ejemplo, limitado por técnicas y parámetros tales como el tamaño de una palabra de código en un algoritmo de FEC, surgen muchos problemas cuando una arquitectura actual transita de manera continua a 400 G. Por ejemplo, datos de 1 x 400 G obtenidos por codificación mediante FEC deben ser suministrados en 16 carriles (nombre en inglés: lane). Una RS de FEC puede ser seleccionada para un algoritmo de FEC, y una RS de FEC (544, 514, 15, 10) pueden ser seleccionadas como un tipo de código específico. Si los datos de 1 x 400 G obtenidos mediante la codificación mediante FEC utilizan el algoritmo de FEC, existe el siguiente problema: para un ancho de banda de 400 G, un ancho de bit de los datos tratados en cada período debe ser un múltiplo entero de 16 carriles, y los datos suministrados en cada carril deben ser un múltiplo entero de un símbolo de FEC (nombre en inglés: symbol). Por ejemplo, el tamaño de un símbolo en un algoritmo de RS de FEC actualmente posible es de 10 bits; por lo tanto, un ancho de bit de cada período debe ser un múltiplo entero de 16 x 10. Específicamente, un ancho de bit posible y una frecuencia de reloj tienen las siguientes posibilidades: el ancho de bit es de 160 bits (nombre en inglés: bit), y la correspondiente frecuencia de reloj es de 2,5 GHz; el ancho de bit es de 320 bits y la frecuencia de reloj es de 1,25 GHz; el ancho de bit es de 480 bits y la frecuencia de reloj es de 833,33 MHz; el ancho de bit es de 640 bits y la frecuencia de reloj es de 625 MHz; el ancho de bit es de 800 bits y la frecuencia de reloj es de 500 MHz; el ancho de bit es de 960 bits y la frecuencia de reloj es de 416,66 MHz; y así sucesivamente.

Tal como se muestra en la figura 1, un método de tratamiento de datos dado a conocer en una realización de la presente invención incluye las siguientes etapas:

101. Insertar múltiples marcadores de alineación (Alignment Marker, AM, en inglés) en un primer flujo de datos, donde el primer flujo de datos es un flujo de datos que es transcodificado y aleatorizado después de ser codificado en una capa física, y la velocidad de transmisión del primer flujo de datos es mayor que o igual a 100 Gbps.

5 En esta realización de la presente invención, un flujo de datos obtenido por un extremo de transmisión de datos está en un formato de datos de una capa de MAC en de Ethernet, y el formato de datos es convertido a un formato de CDGMII correspondiente después de que el flujo de datos se transmite a una interfaz de Ethernet de 400 G (CDGMII, para abreviar, en inglés). Un formato de datos específico de la interfaz CDGMII se describe como sigue: un ancho de bit de datos de la interfaz CDGMII es de $(20 \times (64 + 8))$, y el formato de la interfaz CDGMII es de veinte fragmentos de información de datos de 64 bits más un indicador de información de control de 8 bits, donde el indicador de información de control de 8 bits indica si los datos de 8 bytes de 64 bits representan datos o información de control. Los datos en el formato de datos de CDGMII anterior son transmitidos a un módulo de codificación de 64b/66b para realizar la codificación. Un formato de codificación específico de 64b/66b es un método de codificación universal en un estándar actual.

15 Por ejemplo, el flujo de datos es transcodificado después de realizar la codificación de 64b/66b en el flujo de datos, y existen los siguientes modos de transcodificación para un formato específico de un módulo de transcodificación (nombre en inglés: Transcode): 256b/257b, 512b/513b, 256b/258b, 512b/514b y similares. La presente invención no impone ninguna limitación a un modo de transcodificación específico.

20 Además, el flujo de datos transcodificados debe ser aleatorizado, y la aleatorización global o parcial se puede realizar de acuerdo con una granularidad específica. Se puede utilizar la aleatorización global unificada, o la aleatorización correspondiente a una granularidad de FEC, o la aleatorización utilizando un carril (nombre en inglés: lane) en una PCS como granularidad. Además, un valor inicial de un dispositivo de aleatorización puede ser configurado dinámicamente para garantizar el rendimiento de un sistema completo. El flujo de datos que es transcodificado y aleatorizado se define como el primer flujo de datos descrito en la presente invención, el primer flujo de datos incluye múltiples bloques de tratamiento en paralelo solo de datos y la velocidad de transmisión del primer flujo de datos en la presente invención es de 100 Gbps (Nombre en inglés: gigabit por segundo) o superior.

25 Después de obtener el primer flujo de datos, el extremo de transmisión de datos inserta los AM en el primer flujo de datos. Cabe señalar que, para un modo de insertar los AM en el primer flujo de datos, se puede hacer referencia a la técnica anterior. La inserción de AM se realiza después de la aleatorización, y un formato de inserción de AM es un formato transcodificado. Debido a que un extremo de recepción necesita realizar operaciones de alineación y reordenación de acuerdo con los AM, no se realiza una operación de aleatorización sobre los AM.

30 Cabe señalar que la codificación de la capa física puede ser una codificación de 4b/5b, una codificación de 8b/10b, una codificación de 64b/66b o una codificación de capa física de otro tipo, que se ejecuta mediante un circuito de capa física (PHY, para abreviar, PHYsical, en inglés).

35 102. Asignar de manera adaptativa el primer flujo de datos que incluye los AM a múltiples carriles de la PCS para obtener segundos flujos de datos.

En esta realización de la presente invención, después de que el primer flujo de datos es enviado desde un circuito de inserción de AM, el extremo de transmisión de datos asigna adaptativamente el primer flujo de datos que incluye los AM a los múltiples carriles de la PCS, para obtener los segundos flujos de datos, donde los segundos flujos de datos incluyen los AM citados anteriormente.

40 En esta realización de la presente invención, después de que los AM son insertados en el primer flujo de datos, el primer flujo de datos debe ser asignado de manera adaptativa a los carriles de la PCS, es decir, el primer flujo de datos en el que se insertan los AM debe ser suministrado a los múltiples carriles de la PCS. En un ejemplo en el que un ancho de bit de tratamiento del circuito de inserción de AM está representado por P y existen i carriles de la PCS, donde los carriles de la PCS son respectivamente un carril 0 de la PCS, un carril 1 de la PCS, ..., y un carril i-1 de la PCS. El primer flujo de datos cuyo tamaño de datos es P debe ser enviado al carril 0 de la PCS, al carril 1 de la PCS, ..., y al carril i-1 de la PCS. El que P sea de 640 bits e i sea 16 se utiliza como ejemplo; a continuación, se deben suministrar 640 bits del primer flujo de datos a 16 carriles de la PCS, y se suministran 40 bits del primer flujo de datos a cada carril de la PCS.

45 En algunas realizaciones de la presente invención, la etapa 102 de asignar de manera adaptativa el primer flujo de datos que incluye los AM a múltiples carriles de la subcapa física de codificación, PCS, incluye:

50 el primer flujo de datos incluye un bloque de tratamiento en paralelo solo de datos y un bloque de tratamiento en paralelo de AM; asignar de manera adaptativa el bloque de tratamiento en paralelo solo de datos a los múltiples carriles de la PCS de acuerdo con una granularidad del tratamiento en paralelo solo de datos, y asignar de manera adaptativa el bloque de tratamiento en paralelo de AM a los múltiples carriles de la PCS de acuerdo con una granularidad de tratamiento en paralelo de AM.

55 Antes de que los AM sean insertados en el primer flujo de datos, un bloque de datos incluido en el primer flujo de datos puede contener todos los datos. Una vez insertados los AM, el primer flujo de datos es un flujo de datos que se forma

mezclando todos los datos y los AM. Cuando el primer flujo de datos que incluye los AM es asignado de manera adaptativa a los múltiples carriles de la PCS, para una situación en la que el bloque de datos de todos los datos y un bloque de AM en el primer flujo de datos tienen diferentes tamaños, para distribuir uniformemente los AM a los carriles de la PCS en esta realización de la presente invención, el primer flujo de datos que incluye los AM puede ser asignado de manera adaptativa utilizando dos granularidades diferentes. Específicamente, para todos los datos en el primer flujo de datos, el bloque de tratamiento en paralelo solo de datos puede ser asignado de manera adaptativa a los múltiples carriles de la PCS, de acuerdo con la granularidad del tratamiento en paralelo solo de datos; para los AM en el primer flujo de datos, el bloque de tratamiento en paralelo de los AM es asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la granularidad de tratamiento en paralelo de los AM. Debido a que los AM son suministrados de manera separada, se puede garantizar que los AM se distribuyen uniformemente en los múltiples carriles de la PCS. Se puede comprender que, si la granularidad del tratamiento en paralelo solo de datos es igual a la granularidad del tratamiento en paralelo de los AM, el primer flujo de datos que incluye los AM puede ser asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la misma granularidad.

Tal como se muestra en la figura 2-a, la figura 2-a es un diagrama esquemático de un modo de implementación de la asignación adaptativa de un primer flujo de datos a múltiples carriles de la PCS, de acuerdo con una realización de la presente invención. Tal como se muestra en la figura 2-b, la figura 2-b es un diagrama esquemático de otro modo de implementación de asignar de manera adaptativa un primer flujo de datos a múltiples carriles de la PCS de acuerdo con una realización de la presente invención. La asignación adaptativa del primer flujo de datos a los múltiples carriles de la PCS puede ser implementada utilizando un módulo de adaptación (nombre en inglés: Adaptor) en el extremo de transmisión de datos. Una función del módulo de adaptación es suministrar uniformemente los AM a los respectivos carriles de la PCS. El módulo de adaptación realiza el tratamiento de acuerdo con un modo de tratamiento de un bloque de tratamiento en paralelo, donde el bloque de tratamiento en paralelo puede ser clasificado en dos tipos de bloques de tratamiento en paralelo de acuerdo con el contenido del primer flujo de datos: un bloque de tratamiento en paralelo solo de datos y un bloque de tratamiento en paralelo de los AM. En la presente invención, se utilizan dos granularidades para realizar respectivamente el tratamiento de suministro en los AM y todos los datos, que es diferente de una estructura de suministro en la que el suministro se puede realizar solo de acuerdo con una granularidad unificada en la técnica anterior. Sin embargo, cuando los bloques de tratamiento en paralelo de los AM y todos los datos son iguales, las dos granularidades en la presente invención también pueden referirse a la misma granularidad. Tal como se muestra en la figura 2-a, un método para tratar el bloque de tratamiento en paralelo solo de datos en la presente invención debe ser realizado en un modo de suministro de sondeo, en el que la programación por turnos (nombre en inglés: Round Robin Scheduling) se realiza de acuerdo con un símbolo (nombre en inglés: Symbol) en los carriles de la PCS o con un múltiplo entero del símbolo, y el bloque de tratamiento en paralelo solo de datos es suministrado de manera separada al carril 0 de la PCS, al carril 1 de la PCS, al carril 2 de la PCS, ..., y al carril i-1 de la PCS. Tal como se muestra en la figura 2-b, en un método para tratar el bloque de tratamiento en paralelo de los AM en el primer flujo de datos, los AM son suministrados directamente a los carriles de la PCS de acuerdo con la granularidad de tratamiento en paralelo de los AM. El suministro también se realiza en la forma de suministro de sondeo por turnos, y el bloque de tratamiento en paralelo de los AM es suministrado de manera separada al carril 0 de la PCS, al carril 1 de la PCS, al carril 2 de la PCS, ..., y al carril i-1 de la PCS. Los tamaños de los AM en los carriles se utilizan como granularidades de suministro para suministrar los AM a los carriles.

En algunas realizaciones de la presente invención, después de la etapa 102 de asignar adaptativamente el primer flujo de datos que incluye los AM a múltiples carriles de la subcapa física de codificación, PCS, y antes de la etapa 103 de realizar la codificación mediante corrección de errores de reenvío, FEC, en los segundos flujos de datos en los múltiples carriles de la PCS, el método de transmisión de datos dado a conocer en esta realización de la presente invención incluye, además, la siguiente etapa:

si un ancho de bit de tratamiento de un circuito de inserción de AM es diferente del ancho de bit de tratamiento de un circuito de codificación mediante FEC, realizar la conversión de ancho de bit en el primer flujo de datos que incluye los AM, donde un ancho de bit del primer flujo de datos convertido es igual al ancho de bit de tratamiento del circuito de codificación mediante FEC.

Específicamente, después de que los AM son insertados en el primer flujo de datos, el ancho de bit de tratamiento emitido por el circuito de inserción de AM está representado por P, el ancho de bit de tratamiento del circuito de codificación mediante FEC está representado por F, y, en un caso en el que P y F no son iguales, se debe realizar adicionalmente una conversión de ancho de bit en un ancho de bit del primer flujo de datos. Para adaptarse a un requisito del circuito de codificación mediante FEC, el ancho de bit del primer flujo de datos puede ser convertido de P a F. Específicamente, la conversión del ancho de bit se puede implementar utilizando un módulo de cambios (nombre en inglés: Gearbox) o un módulo de segmentación (Nombre en inglés: Slice) dispuesto en el extremo de transmisión de datos.

Por ejemplo, antes de que el primer flujo de datos que incluye los AM sea asignado de manera adaptativa a los múltiples carriles de la PCS, en la presente invención, si utilizar o no Gearbox y Slice para realizar la coincidencia se elige de acuerdo con una condición de coincidencia de ancho de bit de P y F. Cuando P y F no coinciden en la presente invención, Gearbox y Slice deben ser agregadas para colaborar en la conversión de anchos de bit de datos de los segundos flujos de datos de P a F, para garantizar que el tratamiento interno de FEC sea fácil.

103. Realizar la codificación mediante FEC en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos.

5 En esta realización de la presente invención, la codificación mediante FEC se realiza en los segundos flujos de datos suministrados en los múltiples carriles de la PCS. Para un proceso de realización de codificación mediante FEC en los segundos flujos de datos, se puede hacer referencia a la técnica anterior. Los terceros flujos de datos se obtienen después de la codificación mediante FEC, donde los terceros flujos de datos incluyen los AM citados anteriormente.

10 En esta realización de la presente invención, después de que los segundos flujos de datos se hacen coincidir con el circuito de codificación mediante FEC, para ser sometidos a tratamiento, para la selección de un algoritmo de FEC en la presente invención, se puede seleccionar FEC de RS o FEC de BCH. Además, los siguientes algoritmos de FEC también pueden ser seleccionados para completar la codificación mediante FEC: KR4-RS (528, 514, 7, 10), KP4-RS (544, 514, 15, 10), RS (560, 514, 23, 10), RS (576, 514, 31, 10), BCH (2858, 2570, 24), BCH (9193, 8192, 71) y similares. La arquitectura de la presente invención no impone ninguna limitación al algoritmo de FEC citado anteriormente.

15 En la presente invención, un módulo de codificación y decodificación mediante FEC puede utilizar FEC de RS o FEC de BCH, donde se seleccionan diferentes algoritmos de FEC para su uso principalmente de acuerdo con diferentes características de errores de reenvío. FEC de RS se utiliza como ejemplo, a continuación, para describir una función específica incluida en la FEC de RS específica. En referencia a la figura 3, la figura 3 muestra un diagrama esquemático de los módulos de función para realizar la codificación mediante FEC en segundos flujos de datos de acuerdo con una realización de la presente invención, donde se utiliza como ejemplo una estructura de implementación de decodificación mediante FEC de RS. FEC de RS se selecciona preferentemente para una velocidad de transmisión actual de 400 G y, por lo tanto, se utiliza una estructura de implementación de codificación mediante FEC de RS como ejemplo en la presente invención.

20 En la presente invención se utiliza una estructura de implementación de FEC de RS en paralelo, e incluye módulos de decodificación principales en la figura 3: un módulo de cálculo del factor de validación (es decir, el conocido Syndrome), un módulo KES, un módulo de búsqueda de ubicación de error (es decir, el conocido Chien) y un módulo de cálculo del valor de error (es decir, el conocido Forney).

30 KES es un módulo de resolución iterativa de coeficientes de error y, en la actualidad, se utiliza más comúnmente un algoritmo BM. Chien es el módulo de búsqueda de ubicación de errores. Forney es el módulo de cálculo de valores de error. No se enumeran otros módulos secundarios, tales como las identidades de Newton y el marcado de errores. Un ancho de bit de tratamiento en paralelo de entrada/salida mediante FEC (nombre en inglés: datawidth) está representado por F. Los parámetros de FEC se seleccionan de la siguiente manera: para un tamaño c de un bloque de codificación mediante FEC y un grado de paralelismo de FEC F, c/F puede ser un entero y no tener resto, es decir, el tratamiento de un bloque de codificación mediante FEC se puede completar en un múltiplo entero de un período; o c/F puede tener un resto, es decir, el tratamiento se completa en una cantidad no entera de períodos. Para el tamaño c del bloque de codificación mediante FEC y un grado de paralelismo S de Serdes, c/S puede ser un número entero y no tener resto, o puede tener un resto. Todos los escenarios de implementación citados anteriormente se pueden manejar utilizando el método de transmisión de datos dado a conocer en la presente invención.

40 En algunas realizaciones de la presente invención, después de la etapa 102 de asignar adaptativamente el primer flujo de datos que incluye los AM a múltiples carriles de la subcapa física de codificación, PCS, y antes de la etapa 103 de realizar la codificación mediante corrección de errores de reenvío, FEC, en los segundos flujos de datos en los múltiples carriles de la PCS, el método de transmisión de datos dado a conocer en esta realización de la presente invención puede incluir, además, la siguiente etapa:

45 Etapa A1: Asignar dos ciclos de reloj diferentes a dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj para realizar el tratamiento, e insertar datos inactivos (nombre en inglés: Idle) en los dos ciclos de reloj diferentes, donde cada uno de los dos ciclos de reloj diferentes incluye un segmento de palabra de código de los dos segmentos de palabra de código que están originalmente concatenados en un ciclo de reloj.

50 Es decir, en esta realización de la presente invención, si el ancho de bit de tratamiento del circuito de codificación mediante FEC no es un múltiplo entero de un ancho de bit de un carril de la PCS, existe un caso en el que dos segmentos de palabra de código diferentes están concatenados en un mismo ciclo de reloj. Se puede comprender que una palabra de código puede tener múltiples segmentos y un segmento de una palabra de código ocupa un ciclo de reloj. Para el caso en el que un segmento de una palabra de código no ocupa completamente un ciclo de reloj, el segmento y un segmento de otra palabra de código necesitan ocupar por completo el ciclo de reloj de manera conjunta. Los segmentos respectivos en las dos palabras de código diferentes están concatenados en el mismo ciclo de reloj; por tanto, existe un problema de concatenación de segmentos de palabra de código. Por ejemplo, un ciclo de reloj puede referirse a un período, se genera un período concatenado para segmentos en dos palabras de código diferentes, y la cantidad de períodos ocupados por cada palabra de código ya no es una cantidad entera de períodos. Para un circuito de codificación mediante FEC, los datos de las dos palabras de código deben ser tratados en un mismo período cuando las cantidades de períodos ocupados por las dos palabras de código son cantidades no enteras de períodos. Debido a que dos partes de datos de los diferentes segmentos de palabra de código en un período pertenecen a

diferentes palabras de código y los segmentos de las dos palabras de código diferentes deben ser tratados en paralelo, se deben utilizar recursos de tratamiento lógico de dos partes diferentes en el circuito de codificación mediante FEC; de lo contrario, se produce una congestión en el tratamiento.

5 Para un algoritmo de FEC relativamente complejo, el ancho de banda de datos y una frecuencia que pueden ser compatibles con una técnica actual y una técnica de próxima generación y que relativamente ahorran recursos, pueden ser un ancho de bit de 640 bits y una frecuencia de reloj de 625 MHz. Sin embargo, el tamaño de una palabra de código (nombre en inglés: Codeword) de FEC de RS (544, 514, 15, 10) es de 5.440 bits, $5.440 / 640 = 8,5$ períodos, y, debido a un problema de una cantidad no entera de períodos, se genera un problema durante el tratamiento mediante FEC: para una cantidad no entera de períodos, se agregan recursos lógicos redundantes en un algoritmo de decodificación mediante FEC completo para adaptarse a un caso en el que dos palabras de código continuas deben ser tratadas en un período. Por ejemplo, para un circuito de codificación mediante FEC en el que el tamaño de una palabra de código es de 5.440 bits, se determina básicamente, de acuerdo con la técnica actual, que un ancho de bit de Serdes es de 640 bits. Si este ancho de bit también se utiliza para decodificar, $5.440 / 640 = 8,5$, en este caso, existe un problema de concatenación de medio período, y se deben agregar recursos lógicos de manera separada para los datos en el período concatenado, a costa de agregar una cantidad extremadamente alta de recursos lógicos. Los recursos lógicos deben ser incrementados en aproximadamente un 50 %. Otro módulo es similar y no se describe en detalle.

20 Para permitir que el ancho de bit de FEC sea tratado en una cantidad entera de períodos, puede surgir un problema de desajuste entre los anchos de bit de los módulos dentro de la PCS. Para este problema, la presente invención da a conocer una solución para asignar dos ciclos de reloj diferentes a dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj para realizar el tratamiento e insertar datos inactivos en los dos ciclos de reloj diferentes. Es decir, se asignan dos ciclos de reloj diferentes a dos segmentos de palabra de código en un ciclo de reloj concatenado para realizar el tratamiento. Por lo tanto, los datos inactivos se pueden completar en una parte en blanco en un ciclo de reloj. Después de completar los datos inactivos en el ciclo de reloj concatenado, debido a que se requieren dos ciclos de reloj para soportar los dos segmentos de la palabra de código, es necesario incrementar la frecuencia, es decir, se debe acelerar la velocidad para tratar una palabra de código, es decir, se utiliza un modo de incremento de frecuencia.

30 En un escenario implementado utilizando la etapa A1 en la presente invención, la etapa 103 de realizar la codificación mediante corrección de errores de reenvío, FEC, en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos incluye:

B1. Realizar la codificación mediante FEC en los datos, excepto en los datos inactivos insertados, en los segundos flujos de datos.

B2. Después de realizar la codificación mediante FEC en los segundos flujos de datos, eliminar los datos inactivos insertados de los segundos flujos de datos codificados para obtener los terceros flujos de datos.

35 Cabe señalar que en el escenario de implementación de la utilización de la etapa A1, los datos inactivos son insertados en los segundos flujos de datos, pero no es necesario realizar la codificación mediante FEC en los datos inactivos, es decir, en la etapa B1, se realiza la codificación mediante FEC solo en los datos, excepto los datos inactivos, en los segundos flujos de datos. Una vez realizada la codificación mediante FEC en los segundos flujos de datos, debido a que los datos inactivos son insertados en los segundos flujos de datos antes de la codificación, es necesario eliminar los datos inactivos insertados. En este caso, se obtienen los terceros flujos de datos. Una forma de eliminar los datos inactivos también se puede describir como lógica de extrusión de burbujas.

Además, en algunas realizaciones de la presente invención, la etapa A1 de asignar dos ciclos de reloj diferentes a dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj para realizar el tratamiento, e insertar datos inactivos en los dos ciclos de reloj diferentes puede incluir específicamente la siguiente etapa:

45 Etapa A11: Insertar, entre los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, datos inactivos que ocupan un ciclo de reloj, asignar el ciclo de reloj en el que los dos segmentos de palabra de código están concatenados originalmente a un segmento de palabra de código de los dos segmentos de palabra de código que están concatenados originalmente en un ciclo de reloj y una parte de los datos inactivos en un ciclo de reloj, y asignar un ciclo de reloj a continuación del ciclo de reloj en el que los dos segmentos de palabra de código están concatenados originalmente al otro segmento de palabra de código de los dos segmentos de palabra de código que están concatenados originalmente en un ciclo de reloj y la otra parte de los datos inactivos en un ciclo de reloj.

Además, en algunas realizaciones de la presente invención, la etapa A1 de asignar dos ciclos de reloj diferentes a dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj para realizar el tratamiento, e insertar datos inactivos en los dos ciclos de reloj diferentes puede incluir, específicamente, la siguiente etapa:

55 Etapa A12: Insertar datos inactivos después del segmento de palabra de código anterior en los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, de modo que el ciclo de reloj en el que los dos segmentos de palabra de código están concatenados originalmente se complete, realizar el desplazamiento de datos hacia atrás en el último segmento de palabra de código en los dos segmentos de palabra de código que son

tratados en paralelo en un mismo ciclo de reloj, e insertar datos inactivos después del último segmento de palabra de código en una palabra de código en la que está situado el último segmento de palabra de código, de modo que un ciclo de reloj en el que está situado el último segmento de la palabra de código al que pertenece el último segmento de palabra de código, se rellene.

5 Específicamente, en la etapa A11 y la etapa A12, los datos inactivos que ocupan un ciclo de reloj completo se insertan entre los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj. Los datos inactivos que ocupan un ciclo de reloj completo se dividen en dos partes, una parte de los datos inactivos se concatena con un segmento de la palabra de código y la otra parte de los mismos se concatena con el otro segmento de la palabra de código. En referencia a la figura 4-a, la figura 4-a es un diagrama esquemático de un modo de tratar la concatenación de dos segmentos de palabra de código, de acuerdo con una realización de la presente invención. En referencia a la figura 4-b, la figura 4-b es un diagrama esquemático de otro modo de tratar la concatenación de dos segmentos de palabra de código, de acuerdo con una realización de la presente invención. El siguiente es un ejemplo con fines de descripción: En un algoritmo de FEC de FEC de RS (544, 514, 15, 10), un ancho de bit de salida a 16 carriles es de 640 bits, y, debido a esto $544 * 10 / 640 = 8,5$, que es una cantidad no entera de períodos, y existe el problema de que diferentes segmentos de palabra de código deben ser tratados en un mismo ciclo de reloj. En la presente invención, una solución de tratamiento mediante FEC para dicho múltiplo no entero es la siguiente: se utiliza como ejemplo un ancho de bit de decodificación de 640 bits; se asignan de manera forzosa dos períodos a dos partes de datos utilizando un método de incremento de frecuencia, y, en la figura 4-a y la figura 4-b, se muestra un modo específico de asignación de períodos. En la figura 4-a, el tratamiento de las palabras de código C1 y C2 se utiliza como ejemplo. Un segmento de la palabra de código C1 y un segmento de la palabra de código C2 están en un ciclo de reloj P9, y el segmento de la palabra de código C1 y el segmento de la palabra de código C2 forman el ciclo de reloj P9 concatenado. En la figura 4-a, los datos inactivos son rellenos en P9, y C1 y los datos inactivos rellenan completamente P9; P10 está relleno con datos inactivos, así como C2; las dos palabras de código necesitan 17 ciclos de reloj en total antes de incrementar la frecuencia, y se necesitan 18 ciclos de reloj después del tratamiento de incremento de frecuencia. Sin embargo, las palabras de código C1 y C2 utilizan de manera separada un ciclo de reloj, la lógica de tratamiento original del circuito de codificación mediante FEC se puede utilizar directamente para la ejecución sin causar congestión, y no hay necesidad de copiar un recurso de tratamiento lógico separado. A diferencia de la figura 4-a, en la figura 4-b, después de que los datos inactivos son rellenos en P9, los segmentos de la palabra de código C2 se desplazan sucesivamente hacia atrás. Debido a que el último segmento de C2 no puede ocupar completamente un ciclo de reloj P18, se rellenan datos inactivos en el ciclo de reloj P18, de modo que las palabras de código C1 y C2 utilicen de manera separada un ciclo de reloj, y la lógica de tratamiento original del circuito de codificación mediante FEC puede ser utilizada directamente para la ejecución sin causar congestión y no es necesario copiar un recurso de tratamiento lógico separado. Una ventaja del modo de incremento de frecuencia en la figura 4-a es: después de que se completa el tratamiento de codificación mediante FEC, los datos inactivos son eliminados directamente de acuerdo con la lógica de extrusión de burbujas sin selección mediante un selector (es decir, un MUX). De esta manera, se puede resolver el problema de que existan datos de dos palabras de código en el tratamiento en paralelo de datos en cada ciclo de reloj, y no hay necesidad de copiar una lógica adicional.

En un ejemplo, un ciclo de reloj es un período. En la figura 4-a, se debe incrementar una frecuencia en el caso de que existan dos palabras de código en un mismo período en el circuito de codificación mediante FEC. Los datos inactivos de un período completo se insertan en cada período concatenado para asignar de manera forzosa dos períodos a una parte final de C1 y a una parte inicial de C2. Tal como se muestra en la figura 4-a, el período P9 es asignado a la parte final de C1, y el período P10 es asignado a la parte inicial de C2. En el modo de insertar los datos inactivos de todo el período, la lógica de extruir una burbuja de los datos inactivos después de la codificación mediante FEC es relativamente fácil, solo se requiere eliminar directamente los datos inactivos. La figura 4-b muestra un segundo modo de incremento de frecuencia en la presente invención, en el que se asignan diferentes períodos a un período concatenado de datos para realizar el tratamiento. Mediante este modo de tratamiento, los datos inactivos son insertados solo en la parte final de C1, y la palabra de código C2 está situada al comienzo de un período siguiente, sin que existan los datos inactivos adicionales que existen en el primer modo; sin embargo, es necesario agregar una lógica de MUX para realizar una operación de desplazamiento de datos. En comparación con el primer modo, mediante este modo se insertan menos datos inactivos; por lo tanto, la frecuencia que necesita ser incrementada durante el tratamiento por parte del circuito de codificación mediante FEC es menor que la del primer modo, pero es necesario agregar una operación de desplazamiento de datos de MUX. Por lo tanto, la lógica de extrusión de burbujas sigue siendo necesaria después de que se completa el tratamiento de codificación mediante FEC, y la lógica de extrusión de burbujas necesita la operación de desplazamiento de datos. La lógica de extrusión de burbujas correspondiente a los dos modos de incremento de frecuencia son procesos inversos del tratamiento de incremento de frecuencia.

La solución citada anteriormente de la presente invención amplía la velocidad de datos completa; por lo tanto, la lógica de extrusión de burbujas debe ser agregada posteriormente. Para el ancho de bit de decodificación de 640 bits, se asignan de manera forzosa dos períodos a las dos partes de datos en el período concatenado utilizando un método de incremento de frecuencia. De esta manera, un problema de período no entero se puede manejar sin agregar un recurso. En esta realización, hay un incremento de frecuencia de aproximadamente un 6 %, de 625 MHz a 661 MHz. Sin embargo, en este caso, se amplía la velocidad de datos completa y, posteriormente, se debe agregar una lógica de extrusión de burbujas relativamente pequeña. Los datos inactivos rellenos son extrudidos y se forma un formato de datos que coincide con una velocidad de un ancho de bit de salida.

En esta realización de la presente invención, la codificación mediante FEC se realiza en los segundos flujos de datos. El circuito de codificación mediante FEC utiliza un modo de codificación y decodificación de corrección de errores de extremo a extremo. La arquitectura de la presente invención soporta un modo de codificación de una o más FEC. Un ancho de bit de tratamiento de todo el circuito de codificación mediante FEC está representado por F. Si hay múltiples sub-FEC (Integer_f sub-FEC), un ancho de bit de tratamiento de cada circuito de codificación de sub-FEC se define como f; si solo hay una FEC, el ancho de bit del circuito de codificación mediante FEC es F. Cabe señalar que, en la realización anterior de la presente invención, el método de incremento de frecuencia anterior no se limita simplemente al ejemplo de este múltiplo no entero, sino que es aplicable a los casos de todos los múltiplos no enteros, y no solo es aplicable a una estructura de un circuito de codificación mediante FEC, sino que también es aplicable a un caso de múltiples circuitos de codificación mediante FEC. El método de incremento de frecuencia también es aplicable a una estructura de decodificación mediante FEC, por ejemplo, el método anterior también es aplicable en un caso en el que el ancho de banda de tratamiento de datos f de una sub-FEC y una palabra de código no están en una relación de múltiplo.

En algunas realizaciones de la presente invención, la etapa 103 de realizar la codificación mediante FEC en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos incluye, específicamente, las siguientes etapas:

obtener dos segmentos de palabra de código que están en los segundos flujos de datos y son tratados en paralelo en un mismo ciclo de reloj, donde los dos segmentos de palabra de código incluyen un primer segmento de palabra de código y un segundo segmento de palabra de código, y el volumen de datos del primer segmento de palabra de código es mayor que o igual al del segundo segmento de palabra de código; y

realizar la codificación mediante FEC en el primer segmento de palabra de código utilizando un primer circuito de codificación mediante FEC, y realizar la codificación mediante FEC en el segundo segmento de palabra de código utilizando un segundo circuito de codificación mediante FEC, donde los terceros flujos de datos se generan después de que el primer segmento de palabra de código y el segundo segmento de palabra de código son tratados respectivamente por el primer circuito de codificación mediante FEC y el segundo circuito de codificación mediante FEC, y el segundo circuito de codificación mediante FEC y el primer circuito de codificación mediante FEC utilizan circuitos lógicos de codificación configurables completamente diferentes, o el segundo circuito de codificación mediante FEC es un circuito de codificación mediante FEC obtenido después de que se realiza una operación de configuración de recursos en un circuito lógico de codificación configurable, excepto por un circuito lógico de codificación configurable que trata el primer segmento de palabra de código, en el primer circuito de codificación mediante FEC.

En esta realización de la presente invención, en un caso en el que solo existe un segmento de palabra de código en los segundos flujos de datos en un ciclo de reloj, la codificación mediante FEC se puede realizar utilizando el primer circuito de codificación mediante FEC. Sin embargo, si un ancho de bit de tratamiento del primer circuito de codificación mediante FEC no puede ser divisible por un ancho de bit de una palabra de código completa, existe un caso en el que dos segmentos de palabra de código diferentes están concatenados en un mismo ciclo de reloj. Se puede comprender que una palabra de código puede tener múltiples segmentos y un segmento de una palabra de código puede ocupar un ciclo de reloj. Para un caso en el que un segmento de una palabra de código no ocupa completamente un ciclo de reloj, el segmento y un segmento de palabra de código en otra palabra de código necesitan ocupar por completo el ciclo de reloj de manera conjunta. Los segmentos de palabra de código respectivos en las dos palabras de código diferentes están concatenados en el mismo ciclo de reloj; por tanto, existe un problema de concatenación de segmentos de palabra de código. En los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, el primer segmento de palabra de código es un segmento de palabra de código, en los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, que tiene un volumen de datos mayor o un volumen de datos igual, y el segundo segmento de palabra de código es el otro segmento de palabra de código, en los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, que tienen un volumen de datos más pequeño o un volumen de datos igual.

Tal como se muestra en la figura 4-c y la figura 4-d, la figura 4-c y la figura 4-d son diagramas esquemáticos de otro modo de tratar la concatenación de dos segmentos de palabra de código de acuerdo con una realización de la presente invención. Cuando solo existe un segmento de palabra de código en un mismo ciclo de reloj, el segmento de palabra de código es de una misma palabra de código, y la codificación mediante FEC se realiza en el segmento de palabra de código utilizando el primer circuito de codificación mediante FEC; mientras que, cuando existen dos segmentos de palabra de código en un mismo ciclo de reloj, es decir, cuando dos segmentos de palabra de código C1 y C2 están concatenados, los dos segmentos de palabra de código que existen en el mismo ciclo de reloj pertenecen a palabras de código diferentes. Por ejemplo, el primer segmento C1 de palabra de código pertenece a una primera palabra de código 1, que es la primera palabra de código, y el segundo segmento de palabra de código C2 pertenece a una segunda palabra de código 2, que es la última palabra de código. Por ejemplo, en la figura 4-c, en el ciclo de reloj en el que están concatenados las palabras de código, si $C1 < C2$, el volumen de datos de C1 es menor que el volumen de datos de C2. Se utiliza como ejemplo un período concatenado de un ciclo de reloj. El segmento de palabra de código C2 en el período concatenado se coloca en el primer circuito de codificación mediante FEC para ser sometido a codificación mediante FEC, y el segmento de palabra de código C1 en el período concatenado se coloca en el segundo circuito de codificación mediante FEC para ser sometido a codificación mediante FEC. En la figura 4-d, si

C1 \geq C2 en el período concatenado, C1 en el período concatenado se coloca en el primer circuito de codificación mediante FEC para ser sometido a codificación mediante FEC, y C2 en el período concatenado se coloca en el segundo circuito de codificación mediante FEC para ser sometido a codificación mediante FEC. En la figura 4-c y la figura 4-d, el segundo circuito de codificación mediante FEC solo está configurado para tratar un segmento de palabra de código, en dos segmentos de palabra de código que existen en un mismo ciclo de reloj, que tienen un volumen de datos menor o un volumen de datos igual; el primer circuito de codificación mediante FEC está configurado para tratar un segmento de palabra de código, en los dos segmentos de palabra de código que existen en un mismo ciclo de reloj, que tienen un volumen de datos mayor o un volumen de datos igual, y el primer circuito de codificación mediante FEC está configurado, además, para tratar un único segmento de palabra de código que existe en un ciclo de reloj.

Específicamente, un modo de implementación del primer circuito de codificación mediante FEC y el segundo circuito de codificación mediante FEC es el siguiente: el segundo circuito de codificación mediante FEC es un circuito de codificación mediante FEC independiente del primer circuito de codificación mediante FEC, o el segundo circuito de codificación mediante FEC es un circuito de codificación mediante FEC obtenido después de que se realiza una operación de configuración de recursos en un circuito lógico de codificación configurable, excepto por que un circuito lógico de codificación configurable realiza una codificación mediante FEC en un segmento de palabra de código que tiene un volumen de datos mayor o un volumen de datos igual, en el primer circuito de codificación mediante FEC. Es decir, el segundo circuito de codificación mediante FEC se puede obtener después de que se realiza una operación de configuración de recursos en una parte de los circuitos lógicos de codificación configurables en el primer circuito de codificación mediante FEC, o el segundo circuito de codificación mediante FEC puede ser independiente del primer circuito de codificación mediante FEC y se obtiene agregando directamente un recurso lógico. Específicamente, la operación de configuración de recursos en esta realización de la presente invención puede incluir operaciones tales como configuración de parámetros, establecimiento de registros y activación de datos. Además, se puede agregar otra operación a la operación de configuración de recursos de acuerdo con un escenario de aplicación específico, para implementar la configuración de recursos para un circuito lógico de codificación configurable.

A continuación, se describe la forma de implementación del primer circuito de codificación mediante FEC y del segundo circuito de codificación mediante FEC en esta realización de la presente invención utilizando ejemplos. Tal como se muestra en la figura 4-e, la figura 4-e es un diagrama esquemático de un modo de implementación de un primer circuito de codificación mediante FEC y un segundo circuito de codificación mediante FEC de acuerdo con una realización de la presente invención. El segundo circuito de codificación mediante FEC de la figura 4-e se obtiene después de que se realicen operaciones de configuración de parámetros, establecimiento de registros y activación de datos en un circuito lógico de codificación configurable inactivo en el primer circuito de codificación mediante FEC. En un ejemplo, el primer circuito de codificación mediante FEC y el segundo circuito de codificación mediante FEC se implementan con base en el módulo de cálculo del factor de validación mostrado en la figura 3, y el primer circuito de codificación mediante FEC tiene las unidades de cálculo mediante FEC S1, S2, ..., y S30. La unidad de cálculo de codificación mediante FEC S1 en la figura 4-e se utiliza como ejemplo; S1 tiene un S1_block 0, un S1_block 1, ..., y un S1_block 63. El S1_block 0 se utiliza como ejemplo. Un resultado obtenido al multiplicar la salida de datos de un ciclo de reloj 1 a un ciclo de reloj 10 por Alpha^{32} o Alpha^0 se suma a un resultado obtenido al multiplicar la salida de datos en un ciclo de reloj anterior por Alpha^{64} para obtener los datos D, y los datos D se almacenan en un registro (que está representado por SR en la figura) para obtener datos Q. Se transmiten dos palabras de código en el ciclo de reloj 1, el ciclo de reloj 2, ..., y el ciclo de reloj 10, y son respectivamente una palabra de código 1 y una palabra de código 2. La palabra de código 1 incluye los datos A0, A1, A2, ..., y A543, y la palabra de código 2 incluye los datos B0, B1, B2, ..., y B543. El ciclo de reloj 9 incluye dos segmentos de palabra de código, donde A512 a A543 son un segmento de palabra de código en la palabra de código 1, B0 a B31 son un segmento de palabra de código en la palabra de código 2, y los dos segmentos de palabra de código están concatenados en el ciclo de reloj 9. Antes del ciclo de reloj 9, todos los segmentos de palabra de código transmitidos desde el ciclo de reloj 0 al ciclo de reloj 8 pertenecen a la palabra de código 1. Todo el S1_block 0, el S1_block 1, ..., y el S1_block 63 están involucrados en el cálculo de codificación mediante FEC, es decir, Alpha^0 a Alpha^{63} se seleccionan para realizar el cálculo de codificación mediante FEC. Tal como se indica mediante una línea continua negra en la figura 4-e, cuando aparece una palabra de código de número impar (por ejemplo, la palabra de código 1 o una palabra de código 3), se configura esta línea de parámetros de registro. Existen dos segmentos de palabra de código concatenados en el ciclo de reloj 9, y se seleccionan Alpha^{32} a Alpha^{63} y Alpha^0 a Alpha^{31} para realizar el cálculo de codificación mediante FEC. Tal como se indica mediante una línea discontinua negra en la figura 4-e, S1_block 0, S1_block 1, ..., y S1_block 31 son utilizados por el primer circuito de codificación mediante FEC para realizar la codificación mediante FEC en el segmento de palabra de código de A512 a A543. Para el ciclo de reloj 9, el S1_block 32, el S1_block 33, ..., y el S1_block 63 en el primer circuito de codificación mediante FEC son circuitos lógicos de codificación configurables inactivos. El segundo circuito de codificación mediante FEC puede ser obtenido después de que se realicen las operaciones de configuración de parámetros, ajuste de registro y activación de datos en el S1_block 32, el S1_block 33, ..., y el S1_block 63. A continuación, la codificación mediante FEC se puede realizar en los segmentos de palabra de código de B0 a B31 utilizando el segundo circuito de codificación mediante FEC. En el ciclo de reloj 10, todos los segmentos de palabra de código que comienzan desde B32, de la palabra de código 2, pertenecen a la palabra de código 2, y la codificación mediante FEC se puede realizar en estos segmentos de palabra de código utilizando el primer circuito de codificación mediante FEC formado por el S1_block 0, el S1_block 1, ..., y el S1_block 63. De este ejemplo se puede aprender que el segundo circuito de codificación mediante FEC se puede obtener de un circuito lógico de codificación configurable

inactivo que existe cuando el primer circuito de codificación mediante FEC trata un segmento de palabra de código concatenado.

Tal como se muestra en la figura 4-f, la figura 4-f es un diagrama esquemático de otro modo de implementación de un primer circuito de codificación mediante FEC y un segundo circuito de codificación mediante FEC de acuerdo con una realización de la presente invención. El segundo circuito de codificación mediante FEC de la figura 4-f es independiente del primer circuito de codificación mediante FEC y se obtiene agregando directamente un recurso lógico. En un ejemplo, el primer circuito de codificación mediante FEC y el segundo circuito de codificación mediante FEC se implementan en base al módulo de cálculo del factor de validación mostrado en la figura 3, y el primer circuito de codificación mediante FEC tiene unidades de cálculo de codificación mediante FEC S1, S2, ..., y S30. La unidad de cálculo de codificación mediante FEC S1 en la figura 4-f se utiliza como ejemplo; S1 tiene un S1_block 0, un S1_block 1, ..., y un S1_block 63. El S1_block 0 se utiliza como ejemplo. Un resultado obtenido al multiplicar la salida de datos de un ciclo de reloj 1 a un ciclo de reloj 10 por Alfa^0 se suma a un resultado obtenido al multiplicar la salida de datos de un ciclo de reloj anterior por Alfa^{64} para obtener los datos D, y los datos D son almacenados en un registro (que está representado por SR en la figura) para obtener datos Q. Se transmiten dos palabras de código en el ciclo de reloj 1, el ciclo de reloj 2, ..., y el ciclo de reloj 10, y son respectivamente una palabra de código 1 y una palabra de código 2. La palabra de código 1 incluye los datos A0, A1, A2, ..., y A543, y la palabra de código 2 incluye los datos B0, B1, B2, ..., y B543. Hay dos segmentos de palabra de código en el ciclo de reloj 9, donde A512 a A543 son un segmento de palabra de código en la palabra de código 1, B0 a B31 son un segmento de palabra de código en la palabra de código 2, y los dos segmentos de palabra de código están concatenados en el ciclo de reloj 9. Antes del ciclo de reloj 9, todos los segmentos de palabra de código transmitidos desde el ciclo de reloj 0 al ciclo de reloj 8 pertenecen a la palabra de código 1. Todo el S1_block 0, el S1_block 1, ..., y el S1_block 63 están involucrados en el cálculo de codificación mediante FEC, es decir, se seleccionan Alfa^0 a Alfa^{63} para realizar el cálculo de codificación mediante FEC. Cuando aparece una palabra de código de número impar (por ejemplo, la palabra de código 1 o una palabra de código 3), se configura esta línea de parámetros de registro. Existen dos segmentos de palabra de código concatenados en el ciclo de reloj 9, y se seleccionan Alfa^0 a Alfa^{31} para realizar el cálculo de codificación mediante FEC. El primer circuito de codificación mediante FEC utiliza el bloque S1_block 0, el bloque S1_block 1, ..., y el S1_block 31 para realizar la codificación mediante FEC en el segmento de palabra de código de A512 a A543. El segundo circuito de codificación mediante FEC está configurado de manera independiente fuera del primer circuito de codificación mediante FEC, y el segundo circuito de codificación mediante FEC configurado de manera independiente incluye el S1_block 32, el S1_block 33, ..., y el S1_block 63. A continuación, la codificación mediante FEC puede ser realizada en el segmento de palabra de código de B0 a B31 utilizando el segundo circuito de codificación mediante FEC. En el ciclo de reloj 10, todos los segmentos de palabra de código que comienzan desde B32 de la palabra de código 2 pertenecen a la palabra de código 2, y la codificación mediante FEC se puede realizar en estos segmentos de palabra de código utilizando el primer circuito de codificación mediante FEC formado por el S1_block 0, el S1_block 1, ..., y el S1_block 63. De este ejemplo se puede aprender que el segundo circuito de codificación mediante FEC puede ser independiente del primer circuito de codificación mediante FEC y se puede obtener agregando directamente un recurso lógico.

Cabe señalar que, en esta realización de la presente invención, para los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, cuando los volúmenes de datos de los dos segmentos de palabra de código son iguales, los circuitos lógicos de codificación configurables del segundo circuito de codificación mediante FEC necesitan alcanzar un valor máximo. En este caso, el segundo circuito de codificación mediante FEC puede ser el 50 % de los circuitos lógicos de codificación configurables del primer circuito de codificación mediante FEC. A medida que aumenta la diferencia entre los volúmenes de datos de los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, disminuyen los circuitos lógicos de codificación configurables requeridos por el segundo circuito de codificación mediante FEC. Por ejemplo, los circuitos lógicos de codificación configurables requeridos por el segundo circuito de codificación mediante FEC pueden ser determinados específicamente de acuerdo con una relación de palabras de código de los dos segmentos de palabra de código en un período no entero, y los circuitos lógicos de codificación configurables del segundo circuito de codificación mediante FEC son el valor de los circuitos lógicos de codificación configurables que pueden tratar un segmento de palabra de código más pequeño en el período concatenado no entero.

A partir de las descripciones anteriores se puede aprender que el segundo circuito de codificación mediante FEC puede ser implementado de tal manera que un circuito lógico sea agregado de manera independiente, o se puede obtener de los circuitos del primer circuito de codificación mediante FEC cambiando un parámetro. El segundo circuito de codificación mediante FEC maneja un período no entero durante la codificación mediante FEC. Un problema que existe en el período no entero de que dos segmentos de palabra de código necesitan ser tratados en un período puede ser manejado agregando un circuito lógico de codificación configurable en el período no entero. En esta realización de la presente invención, es equivalente a agregar un segundo circuito de codificación mediante FEC fuera del primer circuito de codificación mediante FEC como circuito lógico de codificación configurable. Esta realización de la presente invención soporta, además, otro método para configurar un parámetro en el primer circuito de codificación mediante FEC para configurar directamente, en un período concatenado, un circuito lógico de codificación configurable inactivo en el primer circuito de codificación mediante FEC como el segundo circuito de codificación mediante FEC para manejar el problema del período no entero.

Si se utiliza el modo de agregar de manera independiente el segundo circuito de codificación mediante FEC, en esta realización de la presente invención se puede agregar un máximo de la mitad de los recursos de circuitos lógicos de codificación configurables requeridos por el primer circuito de codificación mediante FEC. A continuación, se describe, utilizando un ejemplo, un problema de tratar dos segmentos de palabra de código en un período cuando hay una cantidad no entera de períodos. Para un FEC en el que el tamaño de una palabra de código es 5.440 bits, se determina básicamente, de acuerdo con una técnica de un circuito integrado específico de la aplicación (ASIC – Application Specific Integrated Circuit, en inglés) actual, que un ancho de bit de un Serdes es de 640 bits. Si este ancho de bit también se utiliza para la codificación mediante FEC, $5.440 / 640 = 8,5$, existe un problema de concatenación de medio período y es necesario agregar el segundo circuito de codificación mediante FEC a los datos en el período concatenado. Se utiliza como referencia un ancho de bit de codificación de 640 bits; un circuito lógico de codificación configurable del segundo circuito de codificación mediante FEC que necesita ser agregado es la mitad de los circuitos lógicos de codificación configurables del primer circuito de codificación mediante FEC. Como ejemplo adicional, si un algoritmo de FEC de RS (544, 514, 15, 10) se implementa mediante la utilización de una técnica de tratamiento de matrices de puertas programables en campo (FPGA – Field Programmable Gate Array, en inglés), un ancho de bit de Serdes es de 1.280 bits. Si el primer circuito de codificación mediante FEC utiliza este ancho de bit, $5.440 / 1.280 = 4,25$, y, de esta manera, también surge un problema de concatenación de períodos no enteros. Si el segundo circuito de codificación mediante FEC se obtiene mediante la utilización de un método, en la presente invención, de agregar de manera independiente un circuito lógico de codificación configurable, se utiliza como referencia un ancho de bit de codificación de 1.280 bits, y un circuito lógico de codificación configurable del segundo circuito de codificación mediante FEC que debe ser agregado es una cuarta parte de los circuitos lógicos de codificación configurables del primer circuito de codificación mediante FEC.

Si el segundo circuito de codificación mediante FEC se obtiene a partir de los circuitos lógicos de codificación configurables del primer circuito de codificación mediante FEC, y si hay dos segmentos de palabra de código en un ciclo de reloj, el registro debe ser establecido en períodos diferentes. Una regla de configuración específica es: cuando los segmentos de palabra de código en dos períodos continuos provienen de diferentes palabras de código, es necesario realizar una operación de configuración en el registro. Se puede obtener un segundo circuito de codificación mediante FEC en un período no entero mediante operaciones de configuración de parámetros, establecimiento de registros y establecimiento de puertas para los datos. Una parte lógica principal del segundo circuito de codificación mediante FEC sigue siendo de los circuitos lógicos de codificación configurables del primer circuito de codificación mediante FEC; por lo tanto, tampoco es necesario agregar un circuito lógico de codificación configurable adicional. Si el segundo circuito de codificación mediante FEC se obtiene configurando un parámetro en el primer circuito de codificación mediante FEC, el volumen de recursos que necesita ser incrementado no excede el 10 % de los circuitos lógicos de codificación configurables requeridos por el primer circuito de codificación mediante FEC.

Cabe señalar que, en los ejemplos anteriores de la presente invención, la concatenación de los segmentos de palabra de código C1 y C2 en un ciclo de reloj se utiliza como ejemplo tanto en la figura 4-c como en la figura 4-d. Cuando los dos segmentos de palabra de código están concatenados en un ciclo de reloj, el segundo circuito de codificación mediante FEC puede ser configurado de manera independiente del circuito lógico de codificación configurable inactivo en el primer circuito de codificación mediante FEC de acuerdo con una relación de un volumen de datos de C1 con respecto a un volumen de datos de C2, o el segundo circuito de codificación mediante FEC puede ser agregado de manera independiente fuera del primer circuito de codificación mediante FEC de acuerdo con una relación de un volumen de datos de C1 con respecto a un volumen de datos de C2, lo que no está específicamente limitado en la presente invención. Sin embargo, cuando dos palabras de código C2 y C3 están concatenadas en otro ciclo de reloj, la implementación específica del segundo circuito de codificación mediante FEC aún puede ser determinada de acuerdo con una relación de un volumen de datos de C2 con respecto a un volumen de datos de C3. Además, cuando dos palabras de código C3 y C4 están concatenadas en otro ciclo de reloj, la implementación específica del segundo circuito de codificación mediante FEC todavía puede ser determinada de acuerdo con una relación de un volumen de datos de C3 con respecto a un volumen de datos de C4. Cabe señalar que, para obtener el segundo circuito de codificación mediante FEC en un modo de agregar un circuito lógico de codificación configurable, el circuito lógico de codificación configurable que debe ser agregado debe ser un valor más grande en tres segmentos de palabra de código que tienen un volumen de datos más pequeño en tres períodos concatenados (C1:C2, C2:C3 y C3:C4), para determinar el segundo circuito de codificación mediante FEC.

Tal como se muestra en la figura 4-g, la figura 4-g es un diagrama esquemático de otra forma de implementación de un primer circuito de codificación mediante FEC y un segundo circuito de codificación mediante FEC de acuerdo con una realización de la presente invención. El segundo circuito de codificación mediante FEC de la figura 4-g se obtiene después de que las operaciones de configuración de parámetros, establecimiento de registros y activación de datos se realicen en un circuito lógico de codificación configurable inactivo en el primer circuito de codificación mediante FEC. En un ejemplo, el primer circuito de codificación mediante FEC y el segundo circuito de codificación mediante FEC se implementan en base al módulo de cálculo del factor de validación mostrado en la figura 3, y el primer circuito de codificación mediante FEC tiene unidades de cálculo de codificación mediante FEC S1, S2, ..., y S30. La unidad de cálculo de codificación mediante FEC S1 en la figura 4-g se utiliza como ejemplo; S1 tiene un S1_block 0, un S1_block 1, ..., y un S1_block 127. El S1_block 0 se utiliza como ejemplo. Un resultado obtenido al multiplicar la salida de datos de un ciclo de reloj 1 a un ciclo de reloj 17 por Alpha^{32} o Alpha^{64} o Alpha^{96} o Alpha^0 se agrega a un resultado obtenido al multiplicar la salida de datos en un ciclo de reloj anterior por Alpha^{128} para obtener los

datos D, y los datos D se almacenan en un registro (que está representado por SR en la figura) para obtener los datos Q. Se transmiten cuatro palabras de código en el ciclo de reloj 1, el ciclo de reloj 2, ..., y el ciclo de reloj 17, y son respectivamente una palabra de código 1, una palabra de código 2, una palabra de código 3 y una palabra de código 4. La palabra de código 1 incluye los datos A0, A1, A2, ..., y A543, la palabra de código 2 incluye datos B0, B1, B2, ..., y B543, la palabra de código 3 incluye los datos C0, C1, C2, ..., y C543, y la palabra de código 4 incluye los datos D0, D1, D2, ..., y D543. Hay dos segmentos de palabra de código en cada uno de los ciclos de reloj 5, 9 y 13. El tratamiento en el ciclo de reloj 5 se utiliza como ejemplo, donde A512 a A543 son un segmento de palabra de código en la palabra de código 1, B0 a B95 son un segmento de palabra de código en la palabra de código 2, y los dos segmentos de palabra de código están concatenados en el ciclo de reloj 5. Antes del ciclo de reloj 5, todos los segmentos de palabra de código transmitidos desde el ciclo de reloj 0 al ciclo de reloj 4 pertenecen a la palabra de código 1. Todos los S1_block 0, el S1_block 1, ..., y S1_block 127 están involucrados en el cálculo de codificación mediante FEC, es decir, Alpha^0 a Alpha^{63} se seleccionan para realizar el cálculo de codificación mediante FEC. Tal como se indica mediante una línea negra continua en la figura 4-g, cuando se produce una palabra de código de número impar (por ejemplo, la palabra de código 1 o una palabra de código 3), se configura esta línea de parámetros de registro. Existen dos segmentos de palabra de código concatenados en el ciclo de reloj 5, y Alpha^{96} a Alpha^{127} , Alpha^0 a Alpha^{31} , Alpha^{32} a Alpha^{63} y Alpha^{64} a Alpha^{95} se seleccionan para realizar el cálculo de la codificación mediante FEC. Tal como se indica mediante una línea discontinua negra en la figura 4-g, el primer circuito de codificación mediante FEC utiliza el S1_block 32, el S1_block 33, ... y el S1_block 127 para realizar la codificación mediante FEC en el segmento de palabra de código de B0 a B95. El S1_block 0, el S1_block 1, ... y el S1_block 32 en el primer circuito de codificación mediante FEC son circuitos lógicos de codificación configurables inactivos. El segundo circuito de codificación mediante FEC puede ser obtenido después de que se realicen las operaciones de configuración de parámetros, ajuste de registro y activación de datos en el S1_block 0, el S1_block 1, ... y el S1_block 32. Entonces, la codificación mediante FEC se puede realizar en el segmento de palabra de código de A512 a A543 utilizando el segundo circuito de codificación mediante FEC. En el ciclo de reloj 6, todos los segmentos de palabra de código que comienzan desde B32 de la palabra de código 2 pertenecen a la palabra de código 2, y la codificación mediante FEC puede ser realizada en estos segmentos de palabra de código utilizando el primer circuito de codificación mediante FEC formado por el S1_block 0, el S1_block 1 y el S1_block 127. A partir de este ejemplo se puede aprender que el segundo circuito de codificación mediante FEC puede ser obtenido de un circuito lógico de codificación configurable inactivo que existe cuando el primer circuito de codificación mediante FEC trata un segmento de palabra de código concatenado.

Tal como se muestra en la figura 4-h, la figura 4-h es un diagrama esquemático de otro modo de implementación de un primer circuito de codificación mediante FEC y un segundo circuito de codificación mediante FEC de acuerdo con una realización de la presente invención. El segundo circuito de codificación mediante FEC en la figura 4-h es independiente del primer circuito de codificación mediante FEC y se obtiene agregando directamente un recurso lógico. En un ejemplo, el primer circuito de codificación mediante FEC y el segundo circuito de codificación mediante FEC se implementan en base al módulo de cálculo del factor de validación mostrado en la figura 3, y el primer circuito de codificación mediante FEC tiene unidades de cálculo de codificación mediante FEC S1, S2, ..., y S30. La unidad de cálculo de codificación mediante FEC S1 en la figura 4-h se utiliza como ejemplo; S1 tiene un S1_block 0, un S1_block 1, ..., y un S1_block 127. El S1_block 0 se utiliza como ejemplo. Un resultado obtenido al multiplicar la salida de datos de un ciclo de reloj 1 a un ciclo de reloj 6 por Alfa^0 se suma a un resultado obtenido al multiplicar la salida de datos en un ciclo de reloj anterior por Alfa^a para obtener los datos D, y los datos D son almacenados en un registro (que está representado por SR en la figura) para obtener datos Q. Se transmiten dos palabras de código en el ciclo de reloj 1, el ciclo de reloj 2, ..., y el ciclo de reloj 6, y son respectivamente una palabra de código 1 y una palabra de código 2. La palabra de código 1 incluye los datos A0, A1, A2, ..., y A543, y la palabra de código 2 incluye los datos B0, B1, B2, ..., y B543. El ciclo de reloj 5 incluye dos segmentos de palabra de código, donde A512 a A543 son un segmento de palabra de código en la palabra de código 1, B0 a B95 son un segmento de palabra de código en la palabra de código 2, y los dos segmentos de palabra de código están concatenados en el ciclo de reloj 5. Antes del ciclo de reloj 5, todos los segmentos de palabra de código transmitidos desde el ciclo de reloj 0 al ciclo de reloj 4 pertenecen a la palabra de código 1. Todos, el S1_block 0, el S1_block 1, ..., y el S1_block 127 están involucrados en el cálculo de codificación mediante FEC, es decir, se seleccionan Alpha^0 a Alpha^{127} para realizar el cálculo de codificación mediante FEC. Cuando aparece una palabra de código de número impar (por ejemplo, la palabra de código 1 o una palabra de código 3), se configura esta línea de parámetros de registro. Existen dos segmentos de palabra de código concatenados en el ciclo de reloj 5, y se seleccionan Alpha^0 a Alpha^{31} para realizar el cálculo de codificación mediante FEC. El primer circuito de codificación mediante FEC utiliza el S1_block 0, el S1_block 1, ..., y el S1_block 95 para realizar la codificación mediante FEC en el segmento de palabra de código de B0 a B95. El segundo circuito de codificación mediante FEC se configura de manera independiente fuera del primer circuito de codificación mediante FEC, y el segundo circuito de codificación mediante FEC configurado de manera independiente incluye el S1_block 0, el S1_block 1, ..., y el S1_block 31. A continuación, la codificación mediante FEC se puede realizar en el segmento de palabra de código de A512 a A543 utilizando el segundo circuito de codificación mediante FEC. En el ciclo de reloj 6, todos los segmentos de palabra de código que comienzan desde B32 de la palabra de código 2 pertenecen a la palabra de código 2, y la codificación mediante FEC puede ser realizada en estos segmentos de palabra de código utilizando el primer circuito de codificación mediante FEC formado por el S1_block 0, el S1_block 1, ..., y el S1_block 127. A partir de este ejemplo, se puede aprender que el segundo circuito de codificación mediante FEC puede ser independiente del primer circuito de codificación mediante FEC y se puede obtener agregando directamente un recurso lógico.

En algunas realizaciones de la presente invención, después de la etapa 103 de realizar la codificación mediante corrección de errores de reenvío, FEC, en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos, y antes de la etapa 104 de suministrar los terceros flujos de datos a múltiples carriles de la subcapa física de conexión a medio, PMA, de acuerdo con un ancho de bit de entrada de un serializador / deserializador, Serdes, el método de transmisión de datos dado a conocer en esta realización de la presente invención incluye, además, la siguiente etapa:

si un ancho de bit de tratamiento de un circuito de inserción de AM es diferente del ancho de bit de tratamiento de un circuito de codificación mediante FEC, realizar la conversión de ancho de bit en los terceros flujos de datos, donde los anchos de bit de los terceros flujos de datos convertidos son iguales al ancho de bit de tratamiento del circuito de inserción de AM.

Específicamente, después de que se han obtenido los terceros flujos de datos después de que se ha realizado la codificación mediante FEC en los segundos flujos de datos, el ancho de bit de tratamiento emitido por el circuito de inserción de AM está representado por P, el ancho de bit de tratamiento del circuito de codificación mediante FEC está representado por F, y en un caso en el que P y F no sean iguales, la conversión de ancho de bit debe ser realizada adicionalmente en anchos de bit de los terceros flujos de datos. Para adaptarse a un requisito del circuito de codificación mediante FEC, los anchos de bit de los terceros flujos de datos son convertidos de P a F. Una vez completada la codificación mediante FEC, los anchos de bit de los terceros flujos de datos deben ser convertidos además al ancho de bit original P. Específicamente, la conversión de ancho de bit se puede implementar utilizando un módulo de cambios (nombre en inglés: Gearbox) o un módulo de segmentación (nombre en inglés: Slice) dispuesto en el extremo de transmisión de datos. Por ejemplo, antes de que el primer flujo de datos que incluye los AM sea asignado de manera adaptativa a los múltiples carriles de la PCS, en la presente invención, si se utiliza Gearbox y el segmento para realizar la coincidencia se elige de acuerdo con una condición de coincidencia de ancho de bit de P y F. Cuando P y F no coinciden en la presente invención, Gearbox y Slice deben ser agregadas para colaborar en la conversión de anchos de bit de datos de los terceros flujos de datos de P a F, para garantizar la integridad de los AM.

104. Suministrar los terceros flujos de datos a múltiples carriles de la PMA de acuerdo con un ancho de bit de entrada de un serializador / deserializador (nombre en inglés: Serdes) para obtener múltiples cuartos flujos de datos, donde los múltiples cuartos flujos de datos están en una correspondencia de uno a uno con los múltiples carriles de la PMA, cada cuarto flujo de datos incluye, como mínimo, un AM completo y continuo, y el, como mínimo, un AM es un AM en los múltiples AM.

En esta realización de la presente invención, después de que se han obtenido los terceros flujos de datos mediante codificación mediante FEC, debido a que los terceros flujos de datos incluyen los AM, el extremo de transmisión de datos debe suministrar los terceros flujos de datos a los múltiples carriles de la PMA para obtener los múltiples cuartos flujos de datos, donde los múltiples cuartos flujos de datos están en una correspondencia de uno a uno con los múltiples carriles de la PMA, cada cuarto flujo de datos incluye, como mínimo, un AM completo y continuo, y el, como mínimo, un AM es un AM en los múltiples AM. Se requiere la segmentación en los terceros flujos de datos, donde Slice puede dañar la integridad de los AM; como resultado, el lado del decodificador de datos no puede detectar correctamente los AM. Por lo tanto, en esta realización de la presente invención, se utiliza un modo de suministrar los terceros flujos de datos de acuerdo con el ancho de bit de entrada de los Serdes, y un tercer flujo de datos suministrado a cada carril de la PMA incluye un AM completo y continuo. Si el tercer flujo de datos en cada carril de la PMA incluye un AM completo y continuo, un extremo de recepción de datos puede realizar el bloqueo y la detección de AM, y realizar la reasignación en los carriles de acuerdo con los AM después de encontrar los AM.

El, como mínimo, un AM está incluido en los datos suministrados por un circuito de suministro en un ciclo de reloj a un carril de la PMA utilizado para recibir el, como mínimo, un AM. El carril de la PMA utilizado para recibir, como mínimo, un AM es un carril de la PMA en los múltiples carriles de la PMA. El circuito de suministro está configurado para suministrar los terceros flujos de datos a los múltiples carriles de la PMA. En un ciclo de reloj, el circuito de suministro puede suministrar datos a un carril de la PMA en los múltiples carriles de la PMA. Por ejemplo, el circuito de suministro puede suministrar los terceros flujos de datos a los múltiples carriles de la PMA en un modo de programación por turnos. Por ejemplo, los terceros flujos de datos pueden incluir múltiples bloques que ejecutan la codificación mediante FEC, donde los bloques que ejecutan la codificación mediante FEC pueden ser bloques de datos de 5.280 bits.

Cabe señalar que un AM es un bloque de datos. Una cantidad de bits incluidos en el AM es igual a un múltiplo entero positivo de una cantidad de bits incluidos en un bloque de datos en el primer flujo de datos. Por ejemplo, el AM es un bloque de datos de 64 bits o un bloque de datos de 128 bits. Además, se insertan múltiples AM en el primer flujo de datos, y los múltiples AM están en una correspondencia de uno a uno con los múltiples carriles de la subcapa física de codificación, PCS.

Cabe señalar que, en esta realización de la presente invención, se introduce un códec de FEC en la arquitectura de Ethernet. Los terceros flujos de datos se obtienen después de la codificación mediante FEC, y los terceros flujos de datos se suministran de acuerdo con el ancho de bit de entrada de los Serdes. El ancho de banda de tratamiento utilizado para transmitir datos a los múltiples carriles de la PMA puede ser determinado de acuerdo con el ancho de bit de entrada de los Serdes. El extremo de recepción de datos puede recibir, utilizando los carriles de la PMA, los

terceros flujos de datos transmitidos por el extremo de transmisión de datos. Debido a que los terceros flujos de datos se suministran a los múltiples carriles de la PMA en la presente invención, un AM incluido en el tercer flujo de datos suministrado a cada carril de la PMA es continuo y completo, de modo que los AM no se dañen.

5 Cabe señalar que el suministro de los terceros flujos de datos a múltiples carriles de la PMA de acuerdo con un ancho de bit de entrada de un serializador / deserializador (nombre en inglés: Serdes) para obtener múltiples cuartos flujos de datos puede ser específicamente: suministrar uniformemente los terceros flujos de datos a los múltiples carriles de la PMA de acuerdo con el ancho de bit de entrada del serializador / deserializador (nombre en inglés: Serdes) para obtener los múltiples cuartos flujos de datos, donde el suministro uniforme de los terceros flujos de datos pueden ser las cantidades (nombre en inglés: cantidad) de datos recibidos por diferentes carriles de la PMA en los múltiples carriles de la PMA en un mismo período de tiempo (nombre en inglés: período de tiempo) son iguales.

10 En algunas realizaciones de la presente invención, la etapa 104 de suministrar los terceros flujos de datos a múltiples carriles de la subcapa física de conexión a medio, PMA, de acuerdo con un ancho de bit de entrada de un serializador / deserializador, Serdes, puede incluir, específicamente, las siguientes etapas:

15 D1. Determinar un tamaño de segmento de acuerdo con una cantidad de carriles de la PCS y un tamaño de datos de un solo carácter en cada carril de la PCS, y segmentar los terceros flujos de datos de acuerdo con el tamaño de segmento para obtener los terceros flujos de datos segmentados, donde cada segmento incluye múltiples terceros bloques de datos.

20 D2. Determinar, de acuerdo con los anchos de bit de los terceros flujos de datos, el ancho de bit de entrada de los Serdes y el tamaño del segmento, una cantidad de segmentos que deben ser suministrados, extraer, de los terceros flujos de datos segmentado, los terceros bloques de datos en los segmentos correspondientes. a la cantidad de segmentos, y a continuación suministrar los terceros bloques de datos a un mismo carril de la PMA en los múltiples carriles de la PMA después de superponer terceros bloques de datos que están en la misma ubicación en los segmentos correspondientes a la cantidad de segmentos.

25 Se obtiene la cantidad de carriles de la PCS, es decir, una cantidad de carriles de la PCS en los que se transmiten los terceros flujos de datos, y a continuación, se obtiene el tamaño de datos de un solo carácter soportado en cada carril de la PCS. Para garantizar que los AM sean continuos y completos, el tamaño de segmento utilizado para segmentar los terceros flujos de datos se determina de acuerdo con la cantidad de carriles de la PCS y el tamaño de datos de un solo carácter soportado en cada carril de la PCS. Por ejemplo, el tamaño del segmento se puede determinar multiplicando la cantidad de carriles de la PCS por el tamaño de los datos de un solo carácter soportado en cada carril de la PCS. El tamaño del segmento es un intervalo de segmento utilizado para segmentar los terceros flujos de datos, es decir, una cantidad de bloques de datos que deben ser segmentados como un segmento cada vez. Después de que se ha determinado el tamaño del segmento, los terceros flujos de datos pueden ser segmentados de acuerdo con el tamaño del segmento. Después de que se hayan segmentado los terceros flujos de datos, los datos incluidos en cada segmento se definen como un tercer bloque de datos. Opcionalmente, el tamaño del segmento puede ser un múltiplo entero de un resultado obtenido al multiplicar la cantidad de carriles de la PCS por el tamaño de datos de un solo carácter soportado en cada carril de la PCS. De esta forma, un AM en un segmento obtenido mediante segmentación es continuo y completo.

40 Después de que se han segmentado los terceros flujos de datos, la cantidad de segmentos que deben ser suministrados se determina de acuerdo con los anchos de bit de los terceros flujos de datos, el ancho de bit de entrada del Serdes y el tamaño del segmento. Es decir, una cantidad de segmentos que deben ser suministrados a cada carril de la PMA se determina de acuerdo con una relación entre los anchos de bit de los terceros flujos de datos, el ancho de bit de entrada de los Serdes y el tamaño del segmento. Después de que se ha obtenido la cantidad de segmentos mediante el cálculo, los terceros bloques de datos en los segmentos correspondientes a la cantidad de segmentos se extraen de los terceros flujos de datos segmentados, y, a continuación, los terceros bloques de datos son enviados al mismo carril de la PMA en los múltiples carriles de la PMA después de los terceros bloques de datos que están en la misma ubicación en los segmentos correspondientes a la cantidad de segmentos. Por ejemplo, la cantidad de segmentos es 3, a continuación, se obtienen tres segmentos, los terceros bloques de datos que están en la misma ubicación se extraen de los segmentos y los terceros bloques de datos se envían a un carril de la PMA. Los terceros bloques de datos que están situados en diferentes ubicaciones en los segmentos son enviados a diferentes carriles de la PMA. De esta manera, se puede garantizar que los terceros bloques de datos que están en la misma ubicación en los segmentos estén solo en un carril de la PMA; por lo tanto, un tercer flujo de datos en cada carril de la PMA todavía incluye un AM completo y continuo, y el extremo de recepción de datos puede recibir los AM completos y continuos.

55 Además, en algunas realizaciones de la presente invención, la etapa D1 de determinar un tamaño de segmento de acuerdo con una cantidad de carriles de la PCS y un tamaño de datos de un solo carácter soportado en cada carril de la PCS, y segmentar los terceros flujos de datos de acuerdo con el tamaño de segmento incluye:

la cantidad de carriles de la PCS es i , el tamaño de datos de un solo carácter soportado en cada carril de la PCS es m , el tamaño de un tercer bloque de datos en cada segmento obtenido después de que se han segmentado los terceros

flujos de datos es $i \times m$, y los anchos de bit de los terceros flujos de datos son $i \times m \times \text{Entero } 1$, donde el Entero 1 es un entero positivo preestablecido;

etapa D2 de determinar, de acuerdo con los anchos de bit de los terceros flujos de datos, el ancho de bit de entrada de los Serdes y el tamaño del segmento, una cantidad de segmentos incluye:

- 5 el ancho de bit de entrada del Serdes es $i \times m \times \text{Entero } 2$, el tamaño del segmento es $i \times m \times \text{Entero } S$, y, por lo tanto, tanto Entero 1 como Entero 2 son múltiplos enteros de Entero S, y la cantidad de segmentos es $\text{Entero } 2 / \text{Entero } S$, donde Entero 2 es un entero positivo preestablecido; y

10 la etapa D2 de extraer, de los terceros flujos de datos segmentados, terceros bloques de datos segmentados correspondientes a la cantidad de segmentos, y a continuación suministrar los terceros bloques de datos a un mismo carril de la PMA en los múltiples carriles de la PMA después de superponer terceros bloques de datos que están en una misma ubicación en los segmentos correspondiente a la cantidad de segmentos incluye:

extraer el tercer bloque de datos de orden n de cada segmento, y suministrar al $\text{Entero } 2 / \text{Entero } S$ extraído terceros bloques de datos de orden n en total hasta el carril de la PMA de orden n después de superponer al $\text{Entero } 2 / \text{Entero } S$ extraído terceros bloques de datos de orden n , donde el valor de n es cualquier valor comprendido entre 0 e $i-1$.

- 15 En esta realización de la presente invención, los terceros flujos de datos obtenidos después de que se realiza el tratamiento de codificación mediante FEC se deben transmitir los Serdes. Si F y S no coinciden, no se puede utilizar una tecnología Gearbox simple existente, porque la integridad y continuidad de un patrón de AM puede resultar dañada.

20 Un caso de una Gearbox común no se puede adaptar a la integridad y continuidad de un patrón de AM en un proceso de coincidencia de varios anchos de bit. Tal como se muestra en la figura 5, la figura 5 es un diagrama esquemático de la variación del ancho de bit de una Gearbox común en la técnica anterior. Por ejemplo, para un ancho de bit que puede ser seleccionado en una técnica actual y una cantidad de AM y un tamaño de AM que se definen en un estándar actual, se produce un problema de que la integridad y la continuidad de un patrón de AM se dañan cuando la Gearbox realiza la conversión. Se supone que los terceros flujos de datos obtenidos después de realizar la codificación mediante FEC tienen 16 AM en total, que son respectivamente AM 0, AM 1, ..., y AM 15, y además de los 16 AM, el resto son bloques solo de datos. Si se utiliza un módulo de cambios común, después de convertir un ancho de bit de 640 bits a 480 bits, cada bloque de datos sigue siendo de 40 bits y por lo tanto existen 12 carriles después del cambio. Los AM originales se dividen en AM 0_P1, ..., AM 11_P1, AM 12_P1, AM 13_P1, ..., AM 7_P2, AM 8_P2, AM 9_P2, ..., y un bloque solo de datos, que puede estar situado en un tercer tubo. Entonces, se produce el problema de que los AM son discontinuos, los AM están dañados y el extremo de recepción de datos no puede recibir los AM normalmente.

25 Se puede observar en la figura 5 que se produce un problema de daño en un patrón de AM cuando Gearbox realiza una conversión de ancho de bit en información que lleva el patrón de AM, y no se puede garantizar que un patrón de AM se distribuya uniformemente en cada carril. Sin embargo, la tecnología de segmentación en la presente invención puede garantizar la coincidencia de anchos de bit y garantizar que un patrón de AM distribuido en cada carril sea continuo y completo. Por ejemplo, la tecnología de segmentación en la presente invención garantiza la distribución de un AM en cada carril, y una tecnología de segmentación y reenvío en la presente invención es similar a la conversión de ancho de bit, pero es diferente de una tecnología Gearbox existente. La técnica anterior no puede garantizar que los AM sean distribuidos uniformemente en los carriles de salida de FEC. La tecnología Slice y reenvío de la presente invención puede garantizar eficazmente la corrección de los AM.

- 30 La técnica anterior citada anteriormente es que la Gearbox común resuelve un desajuste entre los anchos de bit. Sin embargo, debido a un múltiplo no entero, los AM no se pueden distribuir uniformemente en el nuevo ancho de banda, y no se puede implementar una función de alineación de los AM, lo que afecta a un módulo de alineación y bloqueo y un módulo de reordenamiento en el extremo de recepción, es decir, un extremo de par no puede recuperar datos.

35 Haciendo referencia a la figura 6, la figura 6 muestra un diagrama esquemático de un modo de implementación de suministrar terceros flujos de datos a múltiples carriles de la PMA de acuerdo con una realización de la presente invención. De acuerdo con la tecnología Slice en la presente invención, los terceros flujos de datos obtenidos después de la codificación mediante FEC y de los cuales el ancho total del bit de datos es F , son convertidos en primer lugar mediante Gearbox a un formato en el lado izquierdo de la figura, es decir, los terceros flujos de datos son segmentados en un formato de datos correspondiente a i carriles. Los anchos de bit de los terceros flujos de datos son $i * m * \text{Entero } 1$. Los datos son segmentados utilizando la tecnología Slice de la presente invención en segmentos de datos cuyos tamaños son $i \times m \times \text{Entero } S$, y a continuación se superponen los segmentos de datos. Solo se muestran tres segmentos en la figura 6, y los datos de todos los fragmentos se representan respectivamente mediante $\text{sym } 1$, $\text{sym } 2$ y $\text{sym } 3$. Existen m fragmentos de $\text{sym } 1$ en total en el primer segmento; del mismo modo, existen m fragmentos de $\text{sym } 2$ en total en el segundo segmento, y existen m fragmentos de $\text{sym } 3$ en total en el tercer segmento. En la conversión mostrada en la figura 6, los segmentos se superponen y un ancho de bit de salida a Serdes es $S = i * m * \text{Entero } 2$. Un patrón, en cada carril, del ancho de bit del Serdes que se asigna superponiendo los segmentos anteriores es un patrón de AM completo y continuo. Entero 1 como Entero 2 son múltiplos enteros de Entero S, y una cantidad de segmentos es $\text{Entero } 2 / \text{Entero } S$, donde Entero 2 es un entero positivo preestablecido. El de orden n de los

terceros bloques de datos es extraído de cada segmento, Entero 2/ Entero S bloque de orden n de los terceros bloques de datos en total se superponen y, a continuación, son enviados al carril de la PMA de orden n, donde el valor de n es cualquier valor comprendido entre 0 e i-1. En la figura 6, el primer símbolo 1, el primer símbolo 2 y el primer símbolo 3 son extraídos respectivamente del primer segmento, el segundo segmento y el tercer segmento, y se suministran a una interfaz 0 del serializador / deserializador (Serdes); el segundo símbolo 1, el segundo símbolo 2 y el segundo símbolo 3 son extraídos respectivamente del primer segmento, el segundo segmento y el tercer segmento y se suministran a una interfaz 1 del serializador / deserializador (Serdes);...; y el sym 1 de orden i-1, el sym 2 de orden i-1 y el sym 3 de orden i-1 son extraídos respectivamente del primer segmento, el segundo segmento y el tercer segmento y se suministran a una interfaz i-1 del serializador / deserializador (Serdes).

En esta realización de la presente invención, Adaptor + Gearbox + Slice se utilizan para el ancho de bit F (F no es igual a S) emitido por la codificación mediante FEC, se utiliza un formato de tratamiento de datos de período entero para el tratamiento de codificación mediante FEC (sin embargo, en la presente invención, no se requiere de manera forzosa que un bloque de FEC utilice el formato de tratamiento de datos de período entero, es decir, Gearbox y Slice colaboran para igualar el ancho de banda, pero no es necesario que FEC procese un período entero o un período no entero, y, por lo tanto, Gearbox se utiliza para convertir un ancho de bit de los datos para realizar la transmisión en un carril de salida del Serdes. En referencia a la figura 7, figura 7 muestra un diagrama esquemático de un modo de manejar un desajuste entre los anchos de bit de los módulos en una PCS de acuerdo con una realización de la presente invención.

Gearbox + Slice en la presente invención se utilizan principalmente para resolver un problema de desajuste entre los anchos de bit. Además, para garantizar una implementación fácil, en un caso en el que exista Gearbox + Slice, se debe seleccionar un ancho de bit de un período entero como un ancho de bit de un FEC (que no es necesario de manera forzosa) de modo que el FEC no necesite realizar una operación de incremento de frecuencia y extrusión de burbujas. En este modo de implementación, se utiliza un ancho de bit de decodificación de 680 bits que puede ser divisible por una palabra de código de FEC, y los datos decodificados de salida se convierten en un ancho de bit de datos de salida de 640 bits (16 carriles y 40 bits en cada carril) utilizando Gearbox. De esta manera, el formato de tratamiento de datos de período entero se utiliza para el tratamiento por parte de un módulo de FEC (que tiene palabras de código de 5.440 bits en total), y no es necesario copiar un recurso de un módulo de tratamiento interno de la FEC. A continuación, el ancho de bit de salida F de FEC es convertido a P utilizando Gearbox, y a continuación el módulo de segmentación hace coincidir, mediante segmentación, el ancho de bit de datos con el carril de salida (el ancho de bit de Serdes $S = 640 = i * d * m = 16 * 4 * 10$ o $S = 480 = i * d * m = 16 * 3 * 10$) para realizar la transmisión. De esta manera, se resuelve el problema de una falta de coincidencia entre los anchos de bit, no es necesario agregar un recurso lógico adicional de copia a la lógica de codificación mediante FEC y se resuelve el problema de que una Gearbox común no puede garantizar la integridad y continuidad de un patrón de AM.

La tecnología Slice y la tecnología Gearbox pueden colaborar de manera flexible y no se limitan simplemente a la relación jerárquica anterior. El tratamiento de segmentos también se puede realizar antes de la FEC, y se selecciona y adapta de acuerdo con un requisito de simplificación de la arquitectura específica y, en particular, una condición de ancho de bit. Además, para Slice y Gearbox y el incremento de frecuencia, estas tecnologías innovadoras se implementan mediante la colaboración, no tienen una dependencia forzosa y pueden colaborar de manera flexible para simplificar una estructura de diseño. Slice + Gearbox se utilizan principalmente para igualar los anchos de bit de diversos módulos. El incremento de frecuencia más la lógica de extrusión de burbujas es una solución eficaz que se utiliza en el caso de un período no entero causado por un ancho de bit de FEC.

Se puede aprender a partir de las descripciones de la realización anterior de la presente invención que, en un primer flujo de datos, están insertados marcadores de alineación, AM, donde el primer flujo de datos es un flujo de datos que es transcodificado y aleatorizado después de realizar la codificación mediante 64b / 66b, y una velocidad de transmisión del primer flujo de datos es mayor o igual a 100 G; el primer flujo de datos que incluye los AM se asigna de manera adaptativa a múltiples carriles de la subcapa física de codificación, PCS, para obtener segundos flujos de datos; la codificación mediante corrección de errores de reenvío, FEC, se realiza en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos; y los terceros flujos de datos son enviados a múltiples carriles de la subcapa física de conexión a medio, PMA, de acuerdo con un ancho de bit de entrada de un serializador / deserializador, Serdes, donde un tercer flujo de datos suministrado a cada carril de la PMA incluye un AM completo y continuo. Debido a que en la presente invención, los terceros flujos de datos se suministran a los múltiples carriles de la PMA, y el AM en cada carril de la PMA puede seguir siendo continuo y completo, un extremo de recepción de datos puede obtener AM completos y continuos de los múltiples carriles de la PMA, lo que puede ser aplicable a un escenario en el que la velocidad de transmisión de un flujo de datos es extremadamente alta, y evitar daños a los AM.

En referencia a la figura 8, la figura 8 muestra un método de transmisión de datos desde una perspectiva de un extremo de recepción de datos dado a conocer en una realización de la presente invención. El método puede incluir las siguientes etapas:

801. Recibir cuartos flujos de datos de múltiples carriles de la subcapa física de conexión a medio, PMA, y realizar la demodulación de PMA en los cuartos flujos de datos para obtener los terceros flujos de datos que se obtienen después de que se realiza la demodulación de PMA, donde los múltiples cuartos flujos de datos están en correspondencia de

uno a uno con los múltiples carriles de la PMA, cada cuarto flujo de datos incluye, como mínimo, un AM completo y continuo, y el, como mínimo, un AM es un AM en múltiples AM.

5 En esta realización de la presente invención, después de que un extremo de transmisión de datos transmite los cuartos flujos de datos utilizando los múltiples carriles de la PMA, el extremo de recepción de datos puede recibir los cuartos flujos de datos desde los múltiples carriles de la PMA.

En algunas realizaciones de la presente invención, la etapa 801 de realizar la demodulación de PMA en los cuartos flujos de datos para obtener los terceros flujos de datos que se obtienen después de que se realiza la demodulación de PMA, incluye:

10 E1. Determinar, de acuerdo con los anchos de bit de los cuartos flujos de datos, un ancho de bit de entrada de un serializador / deserializador, Serdes, y un tamaño de segmento, una cantidad de segmentos que deben ser combinados, obtener, de cada carril de la PMA, terceros bloques de datos que estén en una misma ubicación en segmentos correspondientes a la cantidad de segmentos, suministrar respectivamente los terceros bloques de datos a los carriles de la PCS y combinar terceros bloques de datos en cada carril de la PCS para obtener los terceros flujos de datos combinados.

15 Además, la etapa E1 de determinar, de acuerdo con los anchos de bit de los cuartos flujos de datos, un ancho de bit de entrada de un Serdes y un tamaño de segmento, una cantidad de segmentos que deben ser combinados, obtener, de cada carril de la PMA, terceros bloques de datos que están en la misma ubicación en segmentos correspondientes a la cantidad de segmentos, suministrar respectivamente los terceros bloques de datos a los carriles de la PCS, y
20 combinar terceros bloques de datos en cada carril de la PCS para obtener los terceros flujos de datos combinados que incluyen:

una cantidad de carriles de la PCS es i , un tamaño de datos de un solo carácter soportado en cada carril de la PCS es m , y los anchos de bit de los terceros flujos de datos son $i \times m \times \text{Entero } 1$, donde Entero 1 es un entero positivo preestablecido;

25 el ancho del bit de entrada de Serdes es $i \times m \times \text{Entero } 2$, el tamaño del segmento es $i \times m \times \text{Entero } S$, y, por lo tanto, tanto Entero 1 como Entero 2 son múltiplos enteros de Entero S, y la cantidad de segmentos es $\text{Entero } 1 / \text{Entero } S$, donde Entero 2 es un entero positivo preestablecido; y

30 el tamaño de un tercer bloque de datos que está situado en cada segmento y se obtiene de cada carril de la PMA es $i \times m$; obtener $\text{Entero } 1 / \text{Entero } S$ el bloque de orden n de los terceros bloques de datos en total desde el carril de la PMA de orden n , suministrar respectivamente $\text{Entero } 1 / \text{Entero } S$ bloques de orden n de los terceros bloques de datos a los carriles de la PCS, y combinar los terceros bloques de datos en cada carril de la PCS para obtener los terceros flujos de datos cuyos anchos de bit son $i \times m \times \text{Entero } 1$.

35 Cabe señalar que, en esta realización, en el presente documento se da a conocer un proceso en el que el extremo de recepción de datos recibe los terceros flujos de datos de los múltiples carriles de la PMA, y este proceso es un proceso inverso a la etapa 104, en la realización anterior, en la que el extremo de transmisión de datos suministra los terceros flujos de datos a los múltiples carriles de la PMA. Para descripciones detalladas, se puede hacer referencia a las descripciones de la figura 5 en la realización anterior.

802. Identificar los AM completos y continuos de los terceros flujos de datos y realizar la alineación y la reasignación en los terceros flujos de datos en los carriles de la subcapa física de codificación, PCS, de acuerdo con los AM para obtener los terceros flujos de datos alineados y reasignados.

40 En esta realización de la presente invención, después de obtener los terceros flujos de datos que se obtienen después de que se realiza la demodulación de PMA, el extremo de recepción de datos necesita realizar la alineación y reasignación en los terceros flujos de datos en los carriles de la PCS. Se puede aprender del método de transmisión de datos ejecutado por el extremo de transmisión de datos en la realización anterior que los terceros flujos de datos transmitidos por el extremo de transmisión de datos en los múltiples carriles de la PMA llevan los AM completos y
45 continuos; por lo tanto, el extremo de recepción de datos puede identificar los AM completos y continuos de los terceros flujos de datos. Para realizar la alineación y la reasignación en los carriles utilizando los AM, se puede hacer referencia a la técnica anterior, y los detalles no se describen de nuevo en el presente documento.

50 En algunas realizaciones de la presente invención, antes de la etapa 803 de realizar la decodificación mediante corrección de errores de reenvío, FEC, en los terceros flujos de datos alineados y reasignados, el método de transmisión de datos dado a conocer en la presente invención incluye, además, la siguiente etapa:

si un ancho de bit de tratamiento de un circuito de eliminación de AM es diferente del ancho de bit de tratamiento de un circuito de decodificación mediante FEC, realizar la conversión de ancho de bit en los terceros flujos de datos alineados y reasignados, donde los anchos de bit de los terceros flujos de datos cuyos anchos de bit se convierten son iguales al ancho de bit de tratamiento del circuito de decodificación mediante FEC.

Específicamente, el ancho de bit de tratamiento emitido por el circuito de eliminación de AM está representado por P, y el ancho de bit de tratamiento del circuito de decodificación mediante FEC está representado por F. En un caso en el que P y F no son iguales, la conversión de ancho de bit debe ser realizada, además, en los anchos de bit de los terceros flujos de datos. Para evitar dañar la continuidad y la integridad de un patrón de AM, los anchos de bit de los terceros flujos de datos transmitidos por el extremo de transmisión de datos han sido convertidos a P, y para poder adaptarse a un requisito del circuito de decodificación mediante FEC, el extremo de recepción de datos necesita convertir los anchos de bit de los terceros flujos de datos de P a F. Después de que se ha completado la decodificación mediante FEC, los anchos de bit de los terceros flujos de datos deben convertirse además al ancho de bit original P. Específicamente, se puede implementar la conversión de ancho de bit utilizando un módulo de cambios (nombre en inglés: Gearbox) o un módulo de segmentación (nombre en inglés: Slice) dispuesto en el extremo de recepción de datos. Por ejemplo, en la presente invención, el utilizar o no Gearbox y Slice para realizar el ajuste se elige de acuerdo con una condición de coincidencia de ancho de bit de P y F. Cuando P y F no coinciden en la presente invención, Gearbox y Slice deben ser agregadas para colaborar en la conversión de anchos de bit de datos de los terceros flujos de datos de P a F, a fin de garantizar la integridad de los AM.

803. Realizar la decodificación mediante FEC en los terceros flujos de datos alineados y reasignados para obtener segundos flujos de datos, donde los segundos flujos de datos incluyen los AM.

En esta realización de la presente invención, la decodificación mediante FEC realizada en los terceros flujos de datos es un proceso inverso a realizar la codificación mediante FEC en los segundos flujos de datos en la realización anterior, y un modo de implementación específico es inverso al de la codificación mediante FEC.

En algunas realizaciones de la presente invención, antes de la etapa 803 de realizar la decodificación mediante de corrección de errores de reenvío, FEC, en los terceros flujos de datos alineados y reasignados, el método de transmisión de datos dado a conocer en la presente invención incluye, además, la siguiente etapa:

asignar dos ciclos de reloj diferentes a dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj para realizar el tratamiento, e insertar datos inactivos en los dos ciclos de reloj diferentes, donde los dos ciclos de reloj diferentes incluyen cada uno un segmento de palabra de código de los dos segmentos de palabra de código que originalmente están concatenados en un ciclo de reloj.

Es decir, en esta realización de la presente invención, si el ancho de bit de tratamiento del circuito de decodificación mediante FEC no es un múltiplo entero de un ancho de bit de un carril de la PCS, existe un caso en el que dos segmentos de palabra de código diferentes están concatenados en un mismo ciclo de reloj. Se puede comprender que una palabra de código puede tener múltiples segmentos, y un segmento de una palabra de código ocupa un ciclo de reloj. Para el caso en el que un segmento de una palabra de código no ocupa completamente un ciclo de reloj, el segmento y un segmento de otra palabra de código necesitan ocupar por completo el ciclo de reloj de manera conjunta. Los segmentos respectivos en las dos palabras de código diferentes están concatenados en el mismo ciclo de reloj; por tanto, existe un problema de concatenación de segmentos de palabra de código. Por ejemplo, un ciclo de reloj puede referirse a un período, se genera un período concatenado para segmentos en dos palabras de código diferentes, y una cantidad de períodos ocupados por cada palabra de código ya no es una cantidad entera de períodos. Para un circuito de decodificación mediante FEC, los datos de las dos palabras de código deben ser tratados en un mismo período cuando las cantidades de períodos ocupados por las dos palabras de código son cantidades de períodos no enteras. Debido a que dos partes de datos de los diferentes segmentos de palabra de código en un período pertenecen a diferentes palabras de código, y los segmentos de las dos palabras de código diferentes deben ser tratados en paralelo, los recursos de tratamiento lógico de dos partes diferentes deben ser utilizados en el circuito de decodificación mediante FEC; de lo contrario, se produce una congestión en el tratamiento.

Para un algoritmo de FEC relativamente complejo, el ancho de banda de datos y una frecuencia que pueden ser compatibles con una técnica actual y una técnica de próxima generación y que relativamente ahorran recursos, pueden ser un ancho de bit de 640 bits y una frecuencia de reloj de 625 MHz. Sin embargo, el tamaño de una palabra de código (nombre en inglés: Codeword) de FEC de RS (544, 514, 15, 10) es de 5.440 bits, $5.440 / 640 = 8,5$ períodos, y debido a un problema de una cantidad no entera de períodos, se genera un problema durante el tratamiento mediante FEC: para una cantidad no entera de períodos, se agregan recursos lógicos redundantes en un algoritmo de decodificación mediante FEC completo para adaptarse a un caso en el que dos palabras de código continuas deben ser tratadas en un período. Por ejemplo, para un circuito de decodificación mediante FEC cuyo tamaño de una palabra de código es 5.440 bits, se determina básicamente, de acuerdo con la técnica actual, que un ancho de bit de Serdes es de 640 bits. Si este ancho de bit también se utiliza para decodificar, $5.440 / 640 = 8,5$, en este caso, existe un problema de concatenación de medio período, y se deben agregar recursos lógicos de manera separada a los datos en el período concatenado a costa de agregar una cantidad extremadamente grande de recursos lógicos. Los recursos lógicos deben ser incrementados en aproximadamente un 50 %. Otro módulo es similar y no se describe en detalle.

En un escenario de implementación en el que se utiliza una forma de incremento de frecuencia para realizar la decodificación en un ciclo de reloj concatenado en la presente invención, la etapa 803 de realizar la decodificación mediante corrección de errores de reenvío, FEC, en los terceros flujos de datos alineados y reasignados para obtener segundos flujos de datos, incluye:

realizar una decodificación mediante FEC sobre los datos, excepto sobre los datos inactivos insertados, en los terceros flujos de datos alineados y reasignados; y

después de realizar la decodificación mediante FEC en los terceros flujos de datos, eliminar los datos inactivos insertados de los terceros flujos de datos decodificados para obtener los segundos flujos de datos.

- 5 Cabe señalar que los datos inactivos se insertan en los terceros flujos de datos, pero no es necesario realizar la decodificación mediante FEC sobre los datos inactivos, es decir, la decodificación mediante FEC se realiza solo en los datos, excepto en los datos inactivos, en los terceros flujos de datos. Después de realizar la decodificación mediante FEC en los terceros flujos de datos, debido a que los datos inactivos se insertan en los terceros flujos de datos antes de la decodificación, los datos inactivos insertados deben ser eliminados. En este caso, se obtienen los segundos flujos de datos. Un modo de eliminar los datos inactivos también se puede describir como lógica de extrusión de burbujas.

Además, en algunas realizaciones de la presente invención, la asignación de dos ciclos de reloj diferentes a dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj para realizar el tratamiento, y la inserción de datos inactivos en los dos ciclos de reloj diferentes incluye:

- 15 insertar, entre los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, datos inactivos que ocupan un ciclo de reloj, asignar el ciclo de reloj en el que los dos segmentos de palabra de código están concatenados originalmente a un segmento de palabra de código de los dos segmentos de palabra de código que son originalmente concatenados en un ciclo de reloj y una parte de los datos inactivos en un ciclo de reloj, y asignar un ciclo de reloj a continuación del ciclo de reloj en el que los dos segmentos de palabra de código están concatenados originalmente con el otro segmento de palabra de código de los dos segmentos de palabra de código que están originalmente concatenados en un ciclo de reloj y la otra parte de los datos inactivos en un ciclo de reloj.

En algunas otras realizaciones de la presente invención, la asignación de dos ciclos de reloj diferentes a dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj para realizar el tratamiento, e insertar datos inactivos en los dos ciclos de reloj diferentes incluye:

- 25 insertar datos inactivos después del segmento de palabra de código anterior en los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, de modo que el ciclo de reloj en el que los dos segmentos de palabra de código están concatenados originalmente se complete, realizando el desplazamiento de datos hacia atrás en el último segmento de palabra de código en los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, e insertar datos inactivos después del último segmento de palabra de código en una palabra de código en la que está situado el último segmento de palabra de código, de modo que se rellena un ciclo de reloj en el que está situado el último segmento de palabra de código en la palabra de código a la que pertenece el último segmento de palabra de código.

- 30 Cabe señalar que un proceso de implementación del modo de incremento de frecuencia anterior es similar al del modo de incremento de frecuencia descrito en la figura 4-a y la figura 4-b, y se puede hacer referencia a las descripciones de la realización anterior.

En algunas realizaciones de la presente invención, la etapa 803 de realizar la decodificación mediante FEC en los terceros flujos de datos alineados y reasignados para obtener segundos flujos de datos incluye, específicamente, las siguientes etapas:

- 40 obtener dos segmentos de palabra de código que están en los terceros flujos de datos y son tratados en paralelo en un mismo ciclo de reloj, donde los dos segmentos de palabra de código incluyen un primer segmento de palabra de código y un segundo segmento de palabra de código, y el volumen de datos del primer segmento de palabra de código es mayor que o igual al del segundo segmento de palabra de código; y

- 45 realizar una decodificación mediante FEC en el primer segmento de palabra de código utilizando un primer circuito de decodificación mediante FEC, y realizar una decodificación mediante FEC en el segundo segmento de palabra de código utilizando un segundo circuito de decodificación mediante FEC, donde los segundos flujos de datos se generan después de que el primer segmento de palabra de código y el segundo segmento de palabra de código son tratados respectivamente por el primer circuito de decodificación mediante FEC y el segundo circuito de decodificación mediante FEC, y el segundo circuito de decodificación mediante FEC y el primer circuito de decodificación mediante FEC utilizan circuitos lógicos de decodificación configurables completamente diferentes, o el segundo circuito de decodificación mediante FEC es un circuito de decodificación mediante FEC obtenido después de que se realiza una operación de configuración de recursos en un circuito lógico de decodificación configurable, excepto por un circuito lógico de decodificación configurable que trata el primer segmento de palabra de código, en el primer circuito de decodificación mediante FEC.

- 55 En esta realización de la presente invención, en un caso en el que solo existe un segmento de palabra de código en los terceros flujos de datos en un ciclo de reloj, la decodificación mediante FEC se puede realizar utilizando el primer circuito de decodificación mediante FEC. Sin embargo, si un ancho de bit de tratamiento del primer circuito de decodificación mediante FEC no puede ser divisible por un ancho de bit de una palabra de código completa, existe un

caso en el que dos segmentos de palabra de código diferentes están concatenados en un mismo ciclo de reloj. Se puede comprender que una palabra de código puede tener múltiples segmentos y un segmento de una palabra de código puede ocupar un ciclo de reloj. Para un caso en el que un segmento de una palabra de código no ocupa completamente un ciclo de reloj, el segmento y un segmento de palabra de código en otra palabra de código necesitan ocupar por completo el ciclo de reloj de manera conjunta. Los segmentos respectivos en las dos palabras de código diferentes están concatenados en el mismo ciclo de reloj; por tanto, existe un problema de concatenación de segmentos de palabra de código. En los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, el primer segmento de palabra de código es un segmento de palabra de código, en los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, que tiene un volumen de datos mayor o un volumen de datos igual, y el segundo segmento de palabra de código es el otro segmento de palabra de código, en los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, que tienen un volumen de datos más pequeño o un volumen de datos igual.

Tal como se muestra en la figura 4-c y la figura 4-d, la figura 4-c y la figura 4-d son diagramas esquemáticos de otro modo de tratar la concatenación de dos segmentos de palabra de código de acuerdo con una realización de la presente invención. Cuando solo existe un segmento de palabra de código en un mismo ciclo de reloj, el segmento de palabra de código es de la misma palabra de código y la decodificación se realiza en el segmento de palabra de código utilizando el primer circuito de decodificación mediante FEC; mientras que, cuando existen dos segmentos de palabra de código en un mismo ciclo de reloj, es decir, cuando dos segmentos de palabra de código C1 y C2 están concatenados, los dos segmentos de palabra de código que existen en el mismo ciclo de reloj pertenecen a palabras de código diferentes. Por ejemplo, el primer segmento de palabra de código C1 pertenece a una primera palabra de código 1, que es la primera palabra de código, y el segundo segmento de palabra de código C2 pertenece a una segunda palabra de código 2, que es la última palabra de código. Por ejemplo, en la figura 4-c, en el ciclo de reloj en el que están concatenadas las palabras de código, si $C1 < C2$, un volumen de datos de C1 es menor que un volumen de datos de C2. Se utiliza como ejemplo un período concatenado de un ciclo de reloj. C2 en el período concatenado es colocado en el primer circuito de decodificación mediante FEC para ser sometido a una decodificación mediante FEC, y C1 en el período concatenado es colocado en el segundo circuito de decodificación mediante FEC para ser sometido a una decodificación mediante FEC. En la figura 4-d, si $C1 \geq C2$ en el período concatenado, C1 en el período concatenado es colocado en el primer circuito de decodificación mediante FEC para ser sometido a decodificación mediante FEC, y C2 en el período concatenado es colocado en el segundo circuito de decodificación mediante FEC para ser sometido a una decodificación mediante FEC. En la figura 4-c y la figura 4-d, el segundo circuito de decodificación mediante FEC solo está configurado para tratar un segmento de palabra de código, en dos segmentos de palabra de código que existen en un mismo ciclo de reloj, que tienen un volumen de datos menor o un volumen de datos igual; el primer circuito de decodificación mediante FEC está configurado para tratar un segmento de palabra de código, en los dos segmentos de palabra de código que existen en un mismo ciclo de reloj, que tienen un volumen de datos mayor o un volumen de datos igual, y el primer circuito de decodificación mediante FEC se configura, además, para tratar un único segmento de palabra de código que existe en un ciclo de reloj.

Específicamente, un modo de implementación del primer circuito de decodificación mediante FEC y el segundo circuito de decodificación mediante FEC es el siguiente:

el segundo circuito de decodificación mediante FEC es un circuito de decodificación mediante FEC independiente del primer circuito de decodificación mediante FEC, o el segundo circuito de decodificación mediante FEC es un circuito de decodificación mediante FEC obtenido después de que las operaciones de configuración de parámetros, ajuste de registro y activación de datos se realicen en un circuito lógico de decodificación configurable, excepto por que un circuito lógico de decodificación configurable que realiza decodificación mediante FEC en un segmento de palabra de código tiene un volumen de datos mayor o un volumen de datos igual, en el primer circuito de decodificación mediante FEC. Es decir, el segundo circuito de decodificación mediante FEC se puede obtener después de que se realiza una operación de configuración de recursos en una parte de los circuitos lógicos de decodificación configurables en el primer circuito de decodificación mediante FEC, o el segundo circuito de decodificación mediante FEC puede ser independiente del primer circuito de decodificación mediante FEC y se obtiene agregando directamente un recurso lógico. Específicamente, la operación de configuración de recursos en esta realización de la presente invención puede incluir operaciones tales como configuración de parámetros, establecimiento de registros y activación de datos. Además, se puede agregar otra operación a la operación de configuración de recursos de acuerdo con un escenario de aplicación específico para implementar la configuración de recursos para un circuito lógico de decodificación configurable.

Cabe señalar que, en esta realización de la presente invención, para los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, cuando los volúmenes de datos de los dos segmentos de palabra de código son iguales, los circuitos lógicos de decodificación configurables del segundo circuito de decodificación mediante FEC necesitan alcanzar un valor máximo. En este caso, el segundo circuito de decodificación mediante FEC puede ser el 50 % de los circuitos lógicos de decodificación configurables del primer circuito de decodificación mediante FEC. A medida que aumenta la diferencia entre los volúmenes de datos de los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, disminuyen los circuitos lógicos de decodificación configurables requeridos por el segundo circuito de decodificación mediante FEC. Por ejemplo, los circuitos lógicos de decodificación configurables requeridos por el segundo circuito de decodificación mediante FEC se determinan específicamente de acuerdo con una relación de palabra de código de los dos segmentos de palabra de código en un

período no entero, y los circuitos lógicos de decodificación configurables del segundo circuito de decodificación mediante FEC son el valor de los circuitos lógicos de decodificación configurables que pueden tratar un segmento de palabra de código más pequeño en el período concatenado no entero.

5 Se puede aprender a partir de las descripciones anteriores que el segundo circuito de decodificación mediante FEC puede ser implementado agregando de manera independiente un circuito lógico, o puede ser obtenido a partir de los circuitos del primer circuito de decodificación mediante FEC cambiando un parámetro. El segundo circuito de decodificación mediante FEC maneja un período no entero durante la decodificación mediante FEC. Un problema que existe en el período no entero y que sea necesario tratar dos segmentos de palabra de código en un período se puede manejar agregando lógica de tratamiento en el período no entero. En esta realización de la presente invención, es equivalente a agregar un segundo circuito de decodificación mediante FEC fuera del primer circuito de decodificación mediante FEC como circuito lógico de decodificación configurable. Esta realización de la presente invención soporta, además, otro método para configurar un parámetro en el primer circuito de decodificación mediante FEC para configurar directamente, en un período concatenado, un circuito lógico de decodificación configurable inactivo en el primer circuito de decodificación mediante FEC como el segundo circuito de decodificación mediante FEC para manejar el problema del período no entero.

Si se utiliza el modo de agregar de manera independiente el segundo circuito de decodificación mediante FEC, se puede agregar un máximo de la mitad de la lógica de tratamiento original en la presente invención. A continuación, se describe, utilizando un ejemplo, un problema de tratar dos segmentos de palabra de código en un período cuando hay una cantidad no entero de períodos. Para un FEC cuyo tamaño de una palabra de código es de 5.440 bits, se determina básicamente, de acuerdo con una técnica de un ASIC actual, que un ancho de bit de un Serdes es de 640 bits. Si este ancho de bit también se utiliza para la decodificación mediante FEC, $5.440 / 640 = 8,5$, existe un problema de concatenación de medio período y es necesario agregar el segundo circuito de decodificación mediante FEC para los datos en el período concatenado. Se utiliza como referencia un ancho de bit de decodificación de 640 bits; un circuito lógico de decodificación configurable del segundo circuito de decodificación mediante FEC que necesita ser agregado es la mitad de los circuitos lógicos de decodificación configurables del primer circuito de decodificación mediante FEC. Como ejemplo adicional, si se implementa un algoritmo de FEC, FEC de RS (544, 514, 15, 10) utilizando una técnica de tratamiento de FPGA, un ancho de bit del Serdes es de 1.280 bits. Si el primer circuito de decodificación mediante FEC utiliza este ancho de bit, $5.440 / 1.280 = 4,25$, y de esta manera, también surge un problema de concatenación de períodos no enteros. Si el segundo circuito de decodificación mediante FEC se obtiene mediante la utilización de un método, en la presente invención, de agregar de manera independiente un circuito lógico de decodificación configurable, se utiliza como referencia un ancho de bit de decodificación de 1.280 bits, y un circuito lógico de decodificación configurable que es requerido por el segundo circuito de decodificación mediante FEC y debe ser agregado es una cuarta parte del primer circuito de decodificación mediante FEC.

Si el segundo circuito de decodificación mediante FEC se obtiene a partir de los circuitos lógicos de decodificación configurables del primer circuito de decodificación mediante FEC, y si hay dos segmentos de palabra de código en un ciclo de reloj, el registro debe ser establecido en períodos diferentes. Una regla de configuración específica es: cuando los segmentos de palabra de código en dos períodos continuos provienen de diferentes palabras de código, es necesario realizar una operación de configuración en el registro. Se puede obtener un segundo circuito de decodificación mediante FEC en un período no entero mediante operaciones de configuración de parámetros, establecimiento de registros y activación de datos. Una parte lógica principal del segundo circuito de decodificación mediante FEC sigue siendo de los circuitos lógicos de decodificación configurables del primer circuito de decodificación mediante FEC; por lo tanto, tampoco es necesario agregar un circuito lógico de decodificación configurable adicional. Si el segundo circuito de decodificación mediante FEC se obtiene configurando un parámetro en el primer circuito de decodificación mediante FEC, el volumen de recursos que debe ser aumentado no excede el 10 % de los circuitos lógicos de decodificación configurables requeridos por el primer circuito de decodificación mediante FEC.

Cabe señalar que, en los ejemplos anteriores de la presente invención, la concatenación de los segmentos de palabra de código C1 y C2 en un ciclo de reloj se utiliza como ejemplo tanto en la figura 4-c como en la figura 4-d. Cuando los dos segmentos de palabra de código están concatenados en un ciclo de reloj, el segundo circuito de decodificación mediante FEC puede ser configurado de manera independiente del circuito lógico de decodificación configurable inactivo en el primer circuito de decodificación mediante FEC de acuerdo con una relación de un volumen de datos de C1 con respecto a un volumen de datos de C2, o el segundo circuito de decodificación mediante FEC puede ser agregado de manera independiente fuera del primer circuito de decodificación mediante FEC de acuerdo con una relación de un volumen de datos de C1 con respecto a un volumen de datos de C2, que no está específicamente limitada en la presente invención. Sin embargo, cuando dos palabras de código C2 y C3 están concatenadas en otro ciclo de reloj, la implementación específica del segundo circuito de decodificación mediante FEC aún puede ser determinada de acuerdo con una relación de un volumen de datos de C2 con respecto a un volumen de datos de C3. Además, cuando dos palabras de código C3 y C4 están concatenadas en otro ciclo de reloj, la implementación específica del segundo circuito de decodificación mediante FEC aún puede ser determinada de acuerdo con una relación de un volumen de datos de C3 con respecto a un volumen de datos de C4. Cabe señalar que, para obtener el segundo circuito de decodificación mediante FEC en un modo de agregar un circuito lógico de decodificación configurable, el circuito lógico de decodificación configurable que debe ser agregado debe ser un valor mayor en tres segmentos de palabra de código que tienen un volumen de datos menor en tres períodos concatenados (C1:C2, C2:C3 y C3:C4), para determinar el segundo circuito de decodificación mediante FEC.

Cabe señalar que la figura 4-e, la figura 4-f, la figura 4-g y la figura 4-h en la realización anterior describen el primer circuito de decodificación mediante FEC y el segundo circuito de decodificación mediante FEC utilizando ejemplos. Un modo de implementación del primer circuito de decodificación mediante FEC y el segundo circuito de decodificación mediante FEC en esta realización es similar a la leyenda de los circuitos de decodificación mediante FEC anteriores.

5 Una diferencia es que un circuito lógico de decodificación configurable implementa la decodificación mediante FEC, lo que es diferente de que el circuito lógico de decodificación configurable anterior implemente la decodificación mediante FEC.

804. Combinar los segundos flujos de datos que están situados en los carriles de la PCS y se obtienen mediante decodificación mediante FEC para obtener un primer flujo de datos que incluye los AM.

10 En esta realización de la presente invención, el extremo de recepción de datos realiza la decodificación mediante FEC sobre los terceros flujos de datos en los carriles de la PCS para obtener los segundos flujos de datos, combina los segundos flujos de datos en los carriles de la PCS para obtener el primer flujo de datos que incluye los AM. Un proceso en el que el extremo de recepción de datos combina los segundos flujos de datos en los carriles de la PCS y la etapa 102 de asignar adaptativamente el primer flujo de datos a múltiples carriles de la PCS para obtener segundos flujos de datos en la realización anterior, son dos procesos inversos.

15

En algunas realizaciones de la presente invención, la etapa 804 de combinar los segundos flujos de datos que están situados en los carriles de la PCS y que se han obtenido después de la decodificación para obtener un primer flujo de datos que incluye los AM, puede incluir específicamente la siguiente etapa:

20 extraer un bloque de tratamiento en paralelo solo de datos de acuerdo con una granularidad de tratamiento en paralelo solo de datos, extraer un bloque de tratamiento en paralelo de AM de acuerdo con una granularidad de tratamiento en paralelo de AM y combinar el bloque de tratamiento en paralelo solo de datos y el bloque de tratamiento en paralelo de AM para obtener el primer flujo de datos que incluye los AM.

Antes de que los AM sean insertados en el primer flujo de datos, un bloque de datos incluido en el primer flujo de datos puede ser solo de datos. Una vez insertados los AM, el primer flujo de datos es un flujo de datos que se forma mezclando todos los datos y los AM. Cuando el primer flujo de datos que incluye los AM es asignado de manera adaptativa a los múltiples carriles de la PCS, para una situación en la que el bloque de datos solo de datos y un bloque de AM en el primer flujo de datos tienen diferentes tamaños, distribuir uniformemente los AM a los carriles de la PCS en esta realización de la presente invención, el primer flujo de datos que incluye los AM puede ser asignado de manera adaptativa utilizando dos granularidades diferentes. Específicamente, para todos los datos en el primer flujo de datos, el bloque de tratamiento en paralelo solo de datos puede ser asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la granularidad del tratamiento en paralelo solo de datos; para los AM en el primer flujo de datos, el bloque de tratamiento en paralelo de AM es asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la granularidad del tratamiento en paralelo de AM. Debido a que los AM se suministran de manera separada, se puede garantizar que los AM se distribuyan uniformemente en los múltiples carriles de la PCS. Se puede comprender que, si la granularidad del tratamiento en paralelo solo de datos es igual a la granularidad del tratamiento en paralelo de los AM, el primer flujo de datos que incluye los AM puede ser asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la misma granularidad. El extremo de recepción de datos debe utilizar, al combinar los segundos flujos de datos, un modo inverso al utilizado por el extremo de transmisión de datos, y combinar los segundos flujos de datos de acuerdo con la granularidad del tratamiento en paralelo solo de datos y la granularidad del tratamiento en paralelo de los AM, con el fin de obtener el primer flujo de datos que incluye los AM.

25

30

35

40

805. Eliminar los AM del primer flujo de datos que incluye los AM para obtener un primer flujo de datos original utilizado para codificar en el lado del codificador de datos.

Después de completar la alineación y reasignación de los carriles utilizando los AM, se completa una función de alineación de los AM. El extremo de recepción de datos necesita eliminar los AM insertados por el lado del codificador de datos en el primer flujo de datos, para restaurar el primer flujo de datos original utilizado para codificar en el lado del codificador de datos. El primer flujo de datos original es un flujo de datos que se obtiene mediante transcodificación y aleatorización después de que el extremo de transmisión de datos realiza la codificación mediante 64b / 66b. Una velocidad de transmisión del primer flujo de datos es mayor o igual a 100 G, por ejemplo, el primer flujo de datos utiliza una velocidad de transmisión de 400 G o mayor.

45

50 Cabe señalar que, en algunas realizaciones de la presente invención, antes de la etapa 805 de eliminar los AM del primer flujo de datos que incluye los AM, el método de transmisión de datos dado a conocer en la presente invención puede incluir, además, la siguiente etapa:

si un ancho de bit de tratamiento de un circuito de eliminación de AM es diferente del ancho de bit de tratamiento de un circuito de decodificación mediante FEC, realizar la conversión de ancho de bit en el primer flujo de datos que incluye los AM, donde un ancho de bit del primer flujo de datos convertido es igual al ancho de bit del circuito de eliminación de AM.

55

Específicamente, el ancho de bit de tratamiento emitido por el circuito de eliminación de AM está representado por P, y el ancho de bit de tratamiento del circuito de decodificación mediante FEC está representado por F. En un caso en

el que P y F no son iguales, la conversión de ancho de bit debe ser realizada, además, en un ancho de bit del primer flujo de datos. Para adaptarse a un requisito del circuito de decodificación mediante FEC, debido a que el ancho de bit del primer flujo de datos en el que se realiza el tratamiento de decodificación mediante FEC es F, el ancho de bit debe ser convertido de F a P, para que el circuito de eliminación de AM pueda eliminar los AM incluidos en el primer flujo de datos. Específicamente, la conversión de ancho de bit se puede implementar utilizando un módulo de cambios (nombre en inglés: Gearbox) o un módulo de segmentación (nombre en inglés: Slice) dispuesto en el extremo de transmisión de datos.

Se puede aprender, a partir de las descripciones de la realización anterior de la presente invención, que los cuartos flujos de datos se reciben desde múltiples carriles de la subcapa física de conexión a medio, PMA, y la demodulación PMA se realiza en los cuartos flujos de datos para obtener los terceros flujos de datos que se obtienen después de que se realiza la demodulación de la PMA; los AM completos y continuos son identificados a partir de los terceros flujos de datos, y la alineación y reasignación se realizan en los terceros flujos de datos en los carriles de la PCS de acuerdo con los AM para obtener los terceros flujos de datos alineados y reasignados; la decodificación mediante FEC se realiza sobre los terceros flujos de datos alineados y reasignados para obtener segundos flujos de datos, donde los segundos flujos de datos incluyen los AM; los segundos flujos de datos que están en los carriles de la PCS y se obtienen mediante decodificación mediante FEC son combinados para obtener un primer flujo de datos que incluye los AM; y los AM son eliminados del primer flujo de datos que incluye los AM para obtener un primer flujo de datos original utilizado para codificar en el lado del codificador de datos. Debido a que, en la presente invención, los terceros flujos de datos se suministran a los múltiples carriles de la PMA, y un AM en cada carril de la PMA puede permanecer continuo y completo, un extremo de recepción de datos puede obtener AM completos y continuos de los múltiples carriles de la PMA, que pueden ser aplicables a un escenario en el que la velocidad de transmisión de un flujo de datos es bastante alta, y evitar un daño a los AM.

Para una mejor comprensión e implementación de las soluciones anteriores en las realizaciones de la presente invención, a continuación, se proporcionan descripciones específicas utilizando escenarios de aplicación correspondientes como ejemplo.

En referencia a la figura 9, la figura 9 es un diagrama esquemático de la arquitectura de un extremo de transmisión de datos y un extremo de recepción de datos, de acuerdo con una realización de la presente invención. La figura 9 es un diagrama esquemático de una arquitectura general de todo el extremo de transmisión de datos y todo el extremo de recepción de datos. La arquitectura general describe principalmente una capa PHY (que es principalmente una capa PCS y una capa PMA) de Ethernet. El lado izquierdo es un diagrama de los módulos principales del extremo de transmisión de datos y el lado derecho es un diagrama de los módulos principales del extremo de recepción de datos.

Los datos de un extremo de transmisión de datos en la arquitectura de la presente invención están en un formato de datos de una capa de MAC en Ethernet, y el formato de datos es convertido a un formato CDGMII correspondiente después de que los datos son transmitidos a una interfaz CDGMII. Un formato de datos específico de la interfaz CDGMII se describe a continuación: Un ancho de bit de datos de la interfaz CDGMII es $(20 * (64 + 8))$, y el formato de la interfaz CDGMII es veinte fragmentos de información de datos de 64 bits más un indicador de información de control de 8 bits, donde el indicador de información de control de 8 bits indica si los datos de 8 bytes de 64 bits representan datos o información de control. Los datos en el formato de datos de CDGMII anterior se transmiten a un módulo de codificación mediante 64b / 66b para realizar la codificación.

Un formato específico de codificación mediante 64b / 66b es un método de codificación universal en un estándar actual.

Los datos son transcodificados después de ser codificados por el módulo de codificación mediante 64b / 66b, y existen los siguientes modos de transcodificación para un formato específico de un módulo de transcodificación (nombre en inglés: Transcode): 256b / 257b, 512b / 513b, 256b / 258b, 512b / 514b y similares. La presente invención no impone ninguna limitación a un modo de transcodificación específico.

Los datos transcodificados son aleatorizados mediante un módulo de aleatorización, y la codificación global o parcial se puede realizar de acuerdo con una granularidad específica. Se puede utilizar la codificación global unificada, o codificación correspondiente a una granularidad de FEC, o codificación utilizando un carril como granularidad. Además, un valor inicial de un dispositivo de aleatorización puede ser configurado dinámicamente para garantizar el rendimiento de todo un sistema.

La inserción de los AM se realiza mediante un módulo de inserción de AM después de la aleatorización, y un formato de inserción de AM es un formato transcodificado. Debido a que el extremo de recepción de datos necesita realizar operaciones de alineación y reordenamiento de acuerdo con los AM, no se realiza una operación de aleatorización sobre los AM.

Una vez que se han insertado los marcadores de alineación, AM, un módulo de codificación mediante FEC realiza la codificación mediante FEC. El módulo de codificación mediante FEC utiliza una forma de codificación y decodificación de corrección de errores de extremo a extremo. La arquitectura de la presente invención soporta un modo de codificación de una o más FEC. Un ancho de bit de tratamiento de toda la FEC en la figura anterior está representado

por F . Si hay múltiples sub-FEC (Integer f sub-FEC), un ancho de bit de tratamiento de cada sub-FEC se define como f ; si solo hay una FEC, un ancho de bit de la FEC es F . En la solución de la presente invención, la FEC puede elegir lógica de incremento de frecuencia y extrusión de burbujas de acuerdo con una condición de coincidencia de ancho de bit para realizar una operación de coincidencia de ancho de bit de acuerdo con lógica simplificada.

- 5 Para los datos codificados, se puede elegir un módulo de cambios y un módulo de segmentación de acuerdo con la condición de coincidencia de ancho de bit para convertir los datos a un ancho de bit de salida de un Serdes y garantizar que un patrón de AM en cada carril FEC sea completo y continuo, y finalmente, un módulo de transferencia de PMA en el extremo de transmisión de datos transmite los datos.

10 En la presente invención, las operaciones realizadas por el extremo de recepción de datos son inversas a las operaciones realizadas por el extremo de transmisión de datos. En primer lugar, un módulo de recepción de PMA realiza la demodulación de la capa de PMA en los datos recibidos, y, a continuación, los datos demodulados son transmitidos a la capa de PCS. Un módulo de bloqueo y alineación de AM realiza en primer lugar las operaciones de bloqueo y alineación de AM, identifica todos los carriles y realiza una operación de alineación. Un módulo de reordenamiento de carril realiza una operación de reordenamiento de carril sobre los datos alineados, un módulo de decodificación mediante FEC realiza la decodificación, un módulo de eliminación de AM realiza la eliminación de AM, un módulo de decodificación realiza la decodificación, un módulo de re-transcodificación realiza la re-transcodificación y un módulo de decodificación mediante 64b / 66b realiza la decodificación. Los módulos principales, tales como un módulo de conversión de formato de interfaz CDGMII, restauran el formato de datos de la capa de MAC en Ethernet. Asimismo, de acuerdo con una condición de coincidencia de ancho de bit, Slice y Gearbox son insertadas para la adaptación de los módulos en diversas capas.

La figura 9 muestra un diseño de una arquitectura de PCS (PMA) general, que es una arquitectura de implementación actual en la presente invención. Un objetivo principal de la presente invención es el diseño de una capa de PCS. El extremo de recepción, es decir, tres módulos de la figura 9, el módulo de eliminación de AM, el módulo de decodificación mediante FEC y el módulo de recepción PMA, se utilizan como ejemplo. El ancho de bit de tratamiento del módulo de eliminación de AM está representado por P , un ancho de bit de tratamiento del circuito de decodificación mediante FEC está representado por F , un ancho de bit de tratamiento total de Serdes está representado por S , una cantidad de carriles de salida de FEC está representada mediante i , una cantidad de símbolos transmitidos en paralelo en cada carril está representada mediante d , el tamaño de un símbolo está representado mediante m , y el tamaño de una palabra de código FEC está representado mediante n . El extremo de transmisión de datos se utiliza como ejemplo. Un proceso de transmisión de datos ejecutado puede ser el que se muestra en la figura 7. De manera similar, un proceso de transmisión de datos en el extremo de recepción de datos puede ser inverso al proceso de la figura 7. Para más detalles, véanse las descripciones de la realización anterior. En esta realización de la presente invención, se utilizan Adaptor + Gearbox + Slice para el ancho de bit F (F no es igual a S) emitido por la codificación mediante FEC, se utiliza un formato de tratamiento de datos de período entero para el tratamiento de codificación mediante FEC (sin embargo, en la presente invención, no se requiere de manera forzosa que un bloque de FEC utilice el formato de tratamiento de datos de período entero, es decir, Gearbox y Slice colaboran para igualar el ancho de banda, pero no se requiere que FEC procese un período entero o un período no entero), y, por lo tanto, Gearbox se utiliza para convertir un ancho de bit de los datos para realizar la transmisión en un carril de salida del Serdes. En referencia a la figura 7, la figura 7 muestra un diagrama esquemático de un modo de manejar un desajuste entre los anchos de bit de los módulos en una PCS de acuerdo con una realización de la presente invención.

Los anchos de bit de los diversos módulos en la PCS se utilizan como ejemplo para la descripción, y, a continuación, se describe en detalle de cómo garantizar la continuidad e integridad de un patrón de AM cuando los anchos de bit son incoherentes.

Realización 1 de la presente invención:

- 45 Se utiliza como ejemplo el hecho de que FEC de RS (544, 514, 30, 10) sea seleccionado como un algoritmo de FEC. Si los siguientes parámetros, $P = 640$, $F = 680$ y $S = 480$, se seleccionan como anchos de bit de tratamiento de una PCS completa, el tratamiento de ancho de bit completo en la presente invención se muestra en la figura 7.

50 En primer lugar, un adaptador necesita suministrar datos, particularmente un patrón de AM, a 16 carriles, utilizando dos granularidades de suministro. Tal como se muestra en la figura 2-a y la figura 2-b, se utiliza una granularidad de 64 bits para el suministro en un método para tratar una trama solo de datos en la presente invención, y los datos deben ser suministrados de acuerdo con una granularidad de codificación de datos o una granularidad de transcodificación de datos (que es 64 bits en esta realización) en un modo de suministro de sondeo por turnos (nombre en inglés: Round Robin).

55 Para una parte de AM en el flujo de datos, el adaptador suministra directamente la parte de AM de acuerdo con una granularidad de AM, y el suministro también se realiza en el modo de suministro de sondeo por turnos. El AM debe ser suministrado en cada carril de acuerdo con un tamaño, que es una granularidad de 120 bits, del AM en cada carril.

En algunas realizaciones de la presente invención, el suministro de un primer flujo de datos que incluye AM a múltiples carriles de la subcapa física de codificación, PCS, se utiliza como ejemplo e incluye:

el primer flujo de datos incluye un bloque de tratamiento en paralelo solo de datos y un bloque de tratamiento en paralelo de AM; asignar de manera adaptativa el bloque de tratamiento en paralelo solo de datos a los múltiples carriles de la PCS de acuerdo con una granularidad de tratamiento en paralelo solo de datos, y asignar de manera adaptativa el bloque de tratamiento en paralelo de AM a los múltiples carriles de la PCS de acuerdo con una granularidad de tratamiento en paralelo de los AM.

Antes de que los AM sean insertados en el primer flujo de datos, un bloque de datos incluido en el primer flujo de datos puede solo datos. Una vez insertados los AM, el primer flujo de datos es un flujo de datos que se forma mezclando todos los datos y los AM. Cuando el primer flujo de datos que incluye los AM es asignado de manera adaptativa a los múltiples carriles de la PCS, para una situación en la que el bloque de datos solo de datos y un bloque de AM en el primer flujo de datos tienen diferentes tamaños, para distribuir uniformemente los AM a los carriles de la PCS en esta realización de la presente invención, el primer flujo de datos que incluye los AM puede ser asignado de manera adaptativa utilizando dos granularidades diferentes. Específicamente, para todos los datos en el primer flujo de datos, el bloque de tratamiento en paralelo solo de datos puede ser asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la granularidad del tratamiento en paralelo solo de datos; para los AM en el primer flujo de datos, el bloque de tratamiento en paralelo de AM se asigna de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la granularidad de tratamiento en paralelo de los AM. Debido a que los AM se suministran de manera separada, se puede garantizar que los AM se distribuyan uniformemente en los múltiples carriles de la PCS. Se puede comprender que, si la granularidad del tratamiento en paralelo solo de datos es igual a la granularidad del tratamiento en paralelo de los AM, el primer flujo de datos que incluye los AM puede ser asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la misma granularidad.

Tal como se muestra en la figura 2-a, la figura 2-a es un diagrama esquemático de un modo de implementación de la asignación adaptativa de un primer flujo de datos a múltiples carriles de la PCS de acuerdo con una realización de la presente invención. Tal como se muestra en la figura 2-b, la figura 2-b es un diagrama esquemático de otro modo de implementación de asignar de manera adaptativa un primer flujo de datos a múltiples carriles de la PCS de acuerdo con una realización de la presente invención. La asignación adaptativa del primer flujo de datos a los múltiples carriles de la PCS puede ser implementada utilizando un módulo de adaptación (nombre en inglés: Adaptor) en el extremo de transmisión de datos. Una función del módulo de adaptación es suministrar uniformemente los AM a los respectivos carriles de la PCS. El módulo de adaptación realiza el tratamiento de acuerdo con un modo de tratamiento de un bloque de tratamiento en paralelo, donde el bloque de tratamiento en paralelo puede ser clasificado en dos tipos de bloques de tratamiento en paralelo de acuerdo con el contenido del primer flujo de datos: un bloque de tratamiento en paralelo solo de datos y un bloque de tratamiento en paralelo de los AM. En la presente invención, se utilizan dos granularidades para realizar respectivamente el tratamiento de suministro en los AM y en todos los datos, que es diferente de una estructura de suministro en la que el suministro se puede realizar solo de acuerdo con una granularidad unificada en la técnica anterior. Sin embargo, cuando los bloques de tratamiento en paralelo de los AM y todos los datos son iguales, las dos granularidades en la presente invención también se pueden referir a la misma granularidad. Tal como se muestra en la figura 2-a, un método para tratar el bloque de tratamiento en paralelo solo de datos en la presente invención debe ser realizado en un modo de suministro de sondeo en el que la programación por turnos (nombre en inglés: Round Robin Scheduling) se realiza de acuerdo con un símbolo (nombre en inglés: Symbol) en los carriles de la PCS, o un múltiplo entero del símbolo, y el bloque de tratamiento en paralelo solo de datos se suministrado por separado al carril 0 de la PCS, al carril 1 de la PCS, al carril 2 de la PCS, ..., y al carril i-1 de la PCS. Tal como se muestra en la figura 2-b, en un método para tratar el bloque de tratamiento en paralelo de AM en el primer flujo de datos, los AM se suministran directamente a los carriles de la PCS de acuerdo con la granularidad de tratamiento en paralelo de los AM. El suministro también se realiza en el modo de suministro de sondeo de Round Robin, y el bloque de tratamiento en paralelo de los AM es suministrado de manera separada al carril 0 de la PCS, al carril 1 de la PCS, al carril 2 de la PCS, ..., y al carril i-1 de la PCS. Los tamaños de los AM en los carriles se utilizan como granularidades de suministro para suministrar los AM a los carriles.

Después del suministro mediante Adaptor, en la presente invención, se elige si utilizar una Gearbox para realizar la coincidencia de acuerdo con una condición de coincidencia de ancho de bit. Cuando P y F no coinciden en la presente invención, es necesario agregar Gearbox para convertir un ancho de bit de datos de P a F, para garantizar que el tratamiento interno de FEC sea fácil.

A continuación, Gearbox necesita convertir los datos de 640 bits a 680 bits para realizar el tratamiento de FEC. Gearbox es un Gearbox común. En referencia a la figura 10, la figura 10 es un diagrama esquemático de una conversión realizada por un módulo de cambios de 640 bits a 680 bits de acuerdo con una realización de la presente invención. Antes de la conversión realizada por el módulo de cambios, hay 16 AM en un ciclo de reloj, que son, respectivamente, AM 0_0, ..., y AM 0_15. Después de la conversión, hay 18 AM en un ciclo de reloj, que son, respectivamente, AM 0_0, ..., AM 0_15, AM 1_0 y AM 1_1. Además, antes de la conversión, las palabras de código en un ciclo de reloj son las mismas y, después de la conversión, se produce un caso en el que están concatenados un AM y todos los datos.

El tratamiento mediante FEC se realiza en los datos en los que se realiza la conversión de ancho de bit. Los datos de una palabra de código tienen una cantidad entera de periodos: $544 / 68 = 8$ periodos. El ancho de bit de los datos codificados de salida es de 680 bits, y Gearbox convierte los datos en datos de 640 bits para su salida (la conversión realizada por Gearbox es inversa a la conversión realizada por el módulo de cambios en la figura 10). A continuación,

el tratamiento de segmentación debe ser realizado por Slice. El módulo de segmentación mostrado en la figura 6 segmenta los datos de acuerdo con 160 bits, se superponen los bloques de datos que son de tres fragmentos y se suministran a los carriles, y a continuación se envían los datos de Serdes cuyo ancho de bit es de 480 bits. La integridad y continuidad de los patrones de AM de salida en 16 carriles no se daña.

5 Realización 2 de la presente invención:

Se utiliza como ejemplo el hecho de que FEC de RS (544, 514, 30, 10) sea seleccionado como un algoritmo de FEC. Si los siguientes parámetros, $P = 640$, $F = 640$ y $S = 640$, se seleccionan como anchos de bit de tratamiento de una PCS completa, el tratamiento de ancho de bit completo en la presente invención se muestra en la figura 7.

10 En primer lugar, Adaptor necesita suministrar datos, particularmente un patrón de AM, a 16 carriles utilizando dos granularidades de suministro. Tal como se muestra en la figura 2-a y la figura 2-b, se utiliza una granularidad de 256 / 257 bits para el suministro en un método para tratar una trama solo de datos en la presente invención, y los datos deben ser suministrados de acuerdo con una granularidad de codificación de datos o una granularidad de transcodificación (que es 256 / 257 bits, en esta realización) en un modo de suministro de sondeo por turnos. Para una parte de AM en una trama híbrida, Adaptor suministra directamente la parte de AM de acuerdo con una granularidad de AM, y el suministro también se realiza en el modo de suministro de sondeo por turnos. El AM debe ser suministrado en cada carril de acuerdo con un tamaño, que es una granularidad de 320 bits, del AM en cada carril.

15 Después del suministro por parte de Adaptor, en la presente invención, se elige si utilizar una Gearbox para realizar la coincidencia de acuerdo con una condición de coincidencia de ancho de bit. Cuando P y F no coinciden en la presente invención, es necesario agregar Gearbox para convertir un ancho de bit de datos de P a F , para garantizar que el tratamiento interno de FEC sea fácil.

20 En un algoritmo de FEC, FEC de RS (544, 514, 15, 10), un ancho de bit de salida a 16 carriles es de 640 bits. Debido a que $544 \cdot 10 / 640 = 8,5$, que es una cantidad no entera de períodos, un problema, descrito en la figura 4-a y la figura 4b, existen datos de dos palabras de código en un mismo período. Una solución en la presente invención es la siguiente:

25 para un ancho de bit de decodificación de 640 bits, se utiliza un método de incremento de frecuencia para asignar dos períodos a las dos partes de datos. De esta manera, los datos que son tratados en paralelo en cada período no incluyen datos de dos palabras de código y no es necesario copiar una lógica adicional. Alternativamente, tal como se muestra en la figura 4-b, se utiliza un segundo método de concatenación de incremento de frecuencia. De acuerdo con la solución anterior en la presente invención, se amplía una velocidad de datos completa, y la lógica de extrusión de burbujas necesita ser agregada posteriormente, donde la lógica de extrusión de burbujas es un proceso inverso a la lógica de incremento de frecuencia citada anteriormente. Para el ancho de bit de decodificación de 640 bits, se asignan de manera forzosa dos períodos a las dos partes de datos en el período concatenado utilizando un método de incremento de frecuencia. De esta manera, un problema de un período no entero se puede manejar sin agregar un recurso (en el primer modo de esta realización, se requiere un incremento de frecuencia de aproximadamente 6 %, de 625 MHz a 661 MHz). Sin embargo, en este caso, se amplía la velocidad de datos completa, y posteriormente se debe agregar la lógica de extrusión de burbujas correspondiente. El IDLE lleno se extruye y se forma un formato de datos que coincide con una velocidad de un ancho de bit de salida. Una vez que se completa el tratamiento mediante FEC, un ancho de bit en paralelo de salida al Serdes es de 640 bits. En esta realización, debido a que el ancho de bit de salida de FEC es igual al ancho de bit de salida de Serdes, no se requiere una operación realizada mediante Slice.

40 Realización 3 de la presente invención:

Se utiliza como ejemplo el hecho de que FEC de RS (544, 514, 30, 10) sea seleccionado como un algoritmo de FEC. Si los siguientes parámetros, $P = 640$, $F = 680$ y $S = 640$, se seleccionan como anchos de bit de tratamiento de una PCS completa, el tratamiento de ancho de bit completo en la presente invención se muestra en la figura 7.

45 En primer lugar, Adaptor suministrar datos, particularmente un patrón de AM, a 16 carriles, utilizando dos granularidades de suministro. Tal como se muestra en la figura 2-a y la figura 2-b, se utiliza una granularidad de 64 bits para el suministro en un método para tratar una trama solo de datos en la presente invención, y los datos deben ser suministrados de acuerdo con una granularidad de codificación de datos o una granularidad de transcodificación de datos (que es de 64 bits en esta realización) en un modo de suministro de sondeo por turnos. Para una parte de AM en una trama híbrida, Adaptor suministra directamente la parte de AM de acuerdo con una granularidad de AM, y el suministro también se realiza en el modo de suministro de sondeo por turnos. El AM debe ser suministrado en cada carril de acuerdo con un tamaño, que es una granularidad de 120 bits, del AM en cada carril.

50 Después del suministro por parte de Adaptor, en la presente invención, se elige si utilizar una Gearbox para realizar la coincidencia de acuerdo con una condición de coincidencia de ancho de bit. Cuando P y F no coinciden en la presente invención, es necesario agregar Gearbox para convertir un ancho de bit de datos de P a F , para garantizar que el tratamiento interno de FEC sea fácil.

55 A continuación, Gearbox necesita convertir los datos de 640 bits a 680 bits para realizar el tratamiento de FEC. En el tratamiento de FEC posterior, hay una cantidad entera de períodos $544 / 68 = 8$ períodos. El ancho de bit de los datos

codificados de salida es de 680 bits, y Gearbox convierte los datos en datos de 640 bits para su salida. Debido a que Gearbox convierte la salida de datos de FEC a un ancho de bit de salida de 640 bits, que es igual al ancho de bit de salida del Serdes, y la integridad y continuidad de los patrones AM de salida en 16 carriles no se dañan, no es necesario un módulo de segmentación en esta realización.

5 Realización 4 de la presente invención:

P = 640, F = 510 y S = 480. Se utiliza como ejemplo el hecho de que FEC de RS (544, 514, 30, 10) sea seleccionado como un algoritmo de FEC. Si los siguientes parámetros, P = 640, F = 510 y S = 480, se seleccionan como anchos de bit de tratamiento de una PCS completa, el tratamiento de ancho de bit completo en la presente invención se muestra en la figura 7.

10 En primer lugar, Adaptor necesita suministrar datos, particularmente un patrón de AM, a 16 carriles utilizando dos granularidades de suministro. Tal como se muestra en la figura 2-a y la figura 2-b, se utiliza una granularidad de 256 / 257 bits para el suministro en un método para tratar una trama solo de datos en la presente invención, y los datos deben ser suministrados de acuerdo con una granularidad de codificación de datos o una granularidad de transcodificación de datos (que es 256 / 257 bits en esta realización) en un modo de suministro de sondeo por turnos.
15 Para una parte de AM en una trama híbrida, Adaptor suministra directamente la parte de AM de acuerdo con una granularidad de AM, y el suministro también se realiza en el modo de suministro de sondeo por turnos. El AM debe ser suministrado en cada carril de acuerdo con un tamaño, que es una granularidad de 320 bits, del AM en cada carril.

Después del suministro por parte de Adaptor, en la presente invención, se elige si se utiliza una Gearbox (640 a 510) para realizar la coincidencia de acuerdo con una condición de coincidencia de ancho de bit. Cuando P y F no coinciden en la presente invención, es necesario agregar Gearbox para convertir un ancho de bit de datos de P a F, para garantizar que el tratamiento interno de FEC sea fácil.
20

En un algoritmo de FEC, FEC de RS (544, 514, 15, 10), un ancho de bit paralelo interno de FEC es de 510 bits. Debido a que $544 * 10 / 510 = 10,67$, es una cantidad no entera de períodos, existe el problema descrito en la figura 4-a y la figura 4-b. Una solución en esta realización es la siguiente: para un ancho de bit de decodificación de 510 bits, se utiliza un método de incremento de frecuencia para asignar de manera forzosa dos períodos a dos partes de datos. De esta manera, los datos que son tratados en paralelo en cada período no incluyen datos de dos palabras de código y no es necesario copiar una lógica adicional. Se muestra un modo de implementación de la presente invención en la figura 4-a, o un modo de concatenación de datos después del incremento de frecuencia se muestra en la figura 4-b. De acuerdo con la solución anterior en la presente invención, se incrementa una velocidad de datos completa y posteriormente se debe agregar la lógica de extrusión de burbujas. Para el ancho de bit de decodificación de 510 bits, se asignan de manera forzosa dos períodos a las dos partes de datos en el período concatenado utilizando un método de incremento de frecuencia. Existen dos formas específicas de incremento de frecuencia, tal como se muestra en la figura 4-a y la figura 4-b. De este modo, un problema de período no entero se puede manejar sin agregar un recurso. Sin embargo, en este caso, se amplía la velocidad de datos completa, y posteriormente se debe agregar una lógica de extrusión de burbujas relativamente simple. El IDLE relleno se extruye y se forma un formato de datos que coincide con una velocidad de un ancho de bit de salida. Un ancho de bit paralelo de salida al Serdes es de 510 bits.
25
30
35

Debido a que el ancho de bit de los datos codificados emitidos por FEC es de 510 bits, una Gearbox debe convertir los datos a un formato de datos de salida de 640 bits. A continuación, el tratamiento de segmentación debe ser realizado mediante Slice. Los datos son segmentados de acuerdo con 160 bits, los bloques de datos que son de tres segmentos y se envían a los carriles se superponen, y a continuación se emiten los datos de Serdes cuyo ancho de bit es de 480 bits. La integridad y continuidad de los patrones de AM de salida en 16 carriles no se dañan. La figura 6 muestra la tecnología de división en segmentos de la presente invención.
40

Realización 5 de la presente invención:

Si los anchos de bit de tratamiento de los módulos en una capa de PCS son los siguientes: P = 640, F = 480 y S = 480, se utiliza como ejemplo el hecho de que FEC de RS (544, 514, 30, 10) sea seleccionado como un algoritmo de FEC. Si los siguientes parámetros, P = 640, F = 480 y S = 480, se seleccionan como anchos de bit de tratamiento de una PCS completa, el tratamiento de ancho de bit completo en la presente invención se muestra en la figura 7.
45

En primer lugar, una Adaptor necesita suministrar datos, particularmente un patrón de AM, a 16 carriles utilizando dos granularidades de suministro. Tal como se muestra en la figura 2-a y la figura 2-b, se utiliza una granularidad de 256 / 257 bits para el suministro en un método para tratar una trama solo de datos en la presente invención, y los datos deben ser suministrados de acuerdo con una granularidad de codificación de datos o una granularidad de transcodificación de datos (que es 256 / 257 bits en esta realización) en un modo de suministro de sondeo por turnos. Para una parte de AM en una trama híbrida, una Adaptor suministra directamente la parte de AM de acuerdo con una granularidad de AM, y el suministro también se realiza en el modo de suministro de sondeo por turnos. El AM debe ser suministrado en cada carril de acuerdo con un tamaño, que es una granularidad de 320 bits, del AM en cada carril.
50
55

Después del suministro por parte de la Adaptor, en la presente invención, se elige Slice (640 a 480) de acuerdo con una condición de coincidencia de ancho de bit para realizar la coincidencia de ancho de bit. Debido a que el ancho de bit de tratamiento FEC es igual al ancho de bit de tratamiento de Serdes, en esta realización, antes de la codificación

mediante FEC, un segmento de datos puede ser convertido directamente en un ancho de bit de datos correspondiente a 480 bits. De esta manera, la integridad y la continuidad de los patrones de AM en 16 carriles de Serdes se garantizan indirectamente y no es necesario realizar una operación de conversión de ancho de bit posteriormente.

5 Un ancho de bit de datos de 480 bits emitidos por Slice es tratado utilizando un algoritmo de FEC, FEC de RS (544, 514, 15, 10). Un ancho de bit paralelo interno de FEC es de 480 bits. Puesto que $544 * 10 / 480 = 11,34$, que es una cantidad no entera de períodos, existe el problema de la figura 2. Una solución en esta realización es la siguiente: para un ancho de bit de decodificación de 480 bits, se utiliza un método de incremento de frecuencia para asignar de manera forzosa dos períodos a dos partes de datos. De esta manera, los datos que son tratados en paralelo en cada período no incluyen datos de dos palabras de código y no es necesario copiar una lógica adicional. En la siguiente figura se muestra un modo de implementación de la presente invención.

10 La solución anterior de la presente invención amplía la velocidad de datos completa; por lo tanto, la lógica de extrusión de burbujas debe ser agregada posteriormente. Para el ancho de bit de decodificación de 480 bits, se asignan de manera forzosa dos períodos a las dos partes de datos en el período concatenado utilizando un método de incremento de frecuencia. Existen dos formas específicas de incremento de frecuencia, tal como se muestra en la figura 2-a y la figura 2-b. De esta manera, un problema de período no entero se puede manejar sin agregar un recurso. Sin embargo, en este caso, se amplía la velocidad de datos completa y posteriormente se debe agregar una lógica de extrusión de burbujas relativamente simple. El IDLE relleno se extruye y se forma un formato de datos que coincide con una velocidad de un ancho de bit de salida. IDLE es eliminado después de la codificación mediante FEC y se emiten datos con un ancho de bit de 480 bits. Debido a que Slice trata previamente los datos, los datos pueden ser enviados directamente a los Serdes cuyo ancho de bit en paralelo es de 480 bits, y los AM no resultan afectados.

15 Se puede aprender a partir de las descripciones de las realizaciones anteriores de la presente invención que la presente invención soporta la coincidencia de diferentes anchos de bit, lo que hace que la implementación sea más flexible y ahorra recursos. La presente invención soporta una Adaptor que utiliza un modo de suministro de diferentes granularidades. En la presente invención, se puede utilizar una Slice para realizar el tratamiento de segmentación, lo que facilita la coincidencia de ancho de bit, garantiza la integridad y continuidad de un patrón de AM y resuelve un problema de conversión de un ancho de bit de un múltiplo no entero. En la presente invención, FEC soporta dos modos de incremento de frecuencia, en los que se pueden ahorrar eficazmente recursos en un caso de tratamiento mediante FEC de un múltiplo no entero. La arquitectura de una solución completa se puede simplificar mediante la utilización flexible de Slice y Gearbox. La presente invención da a conocer una arquitectura de Ethernet que puede soportar la correspondencia de diferentes anchos de bit. La presente invención da a conocer una arquitectura que puede soportar el tratamiento mediante FEC de un múltiplo entero o no entero. La presente invención soporta el incremento de frecuencia para resolver un problema de incremento de recursos que existe en el caso de un ancho de bit de un múltiplo no entero. El método Slice en la presente invención garantiza la variación flexible de un ancho de bit, así como la integridad de un AM. La presente invención da a conocer una solución que soporta una combinación flexible de Slice y Gearbox. La presente invención soporta modos de suministro de diferentes granularidades para garantizar la distribución de un patrón de AM.

20 Cabe señalar que, para una breve descripción, las realizaciones del método citadas anteriormente se representan como una serie de acciones. Sin embargo, los expertos en la técnica deben apreciar que la presente invención no está limitada al orden descrito de las acciones, porque de acuerdo con la presente invención, algunas etapas pueden ser realizadas en otro orden o simultáneamente. Además, los expertos en la técnica también deben comprender que las realizaciones descritas en esta memoria descriptiva pertenecen todas a realizaciones a modo de ejemplo, y las acciones y módulos involucrados no son necesariamente obligatorios para la presente invención.

25 Para implementar mejor las soluciones anteriores en las realizaciones de la presente invención, a continuación, se da a conocer, además, un aparato relacionado utilizado para implementar las soluciones anteriores.

30 En referencia a la figura 11, una realización de la presente invención da a conocer un extremo de transmisión 1100 de datos, que puede incluir un módulo de inserción de AM 1101, un módulo de adaptación 1102, un módulo de codificación mediante FEC 1103 y un módulo de suministro de flujo de datos 1104.

35 El módulo de inserción de AM 1101 está configurado para insertar múltiples marcadores de alineación, AM, en un primer flujo de datos, donde el primer flujo de datos es un flujo de datos que es transcodificado y aleatorizado después de ser codificado en una capa física, y una velocidad de transmisión del primer flujo de datos es mayor o igual a 100 G.

40 El módulo de adaptación 1102 está configurado para asignar de manera adaptativa el primer flujo de datos que incluye los AM a múltiples carriles de la subcapa física de codificación, PCS, para obtener segundos flujos de datos.

45 El módulo de codificación mediante FEC 1103 está configurado para realizar la codificación mediante corrección de errores de reenvío, FEC, en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos.

50 El módulo de suministro de flujo de datos 1104 está configurado para suministrar los terceros flujos de datos a múltiples carriles de la subcapa física de conexión a medio, PMA, de acuerdo con un ancho de bit de entrada de un serializador

/ deserializador, Serdes, para obtener múltiples cuartos flujos de datos, donde los múltiples cuartos flujos de datos están en una correspondencia de uno a uno con los múltiples carriles de la PMA, cada cuarto flujo de datos incluye, como mínimo, un AM completo y continuo, y el, como mínimo, un AM es un AM en los múltiples AM.

5 En algunas realizaciones de la presente invención, el módulo de inserción de AM es específicamente un circuito de inserción de AM, y el módulo de codificación mediante FEC puede ser específicamente un circuito de codificación mediante FEC.

10 En algunas realizaciones de la presente invención, el módulo de adaptación 1102 está configurado específicamente para: el primer flujo de datos incluye un bloque de tratamiento en paralelo solo de datos y un bloque de tratamiento en paralelo de AM; asignar de manera adaptativa el bloque de tratamiento en paralelo solo de datos a los múltiples carriles de la PCS de acuerdo con una granularidad de tratamiento en paralelo solo de datos, y asignar de manera adaptativa el bloque de tratamiento en paralelo de AM a los múltiples carriles de la PCS de acuerdo con una granularidad de tratamiento en paralelo de los AM.

En algunas realizaciones de la presente invención, la asignación adaptativa del primer flujo de datos que incluye los AM a múltiples carriles de la subcapa física de codificación, PCS, se utiliza como ejemplo e incluye:

15 el primer flujo de datos incluye un bloque de tratamiento en paralelo solo de datos y un bloque de tratamiento en paralelo de los AM; asignar de manera adaptativa el bloque de tratamiento en paralelo solo de datos a los múltiples carriles de la PCS de acuerdo con una granularidad de tratamiento en paralelo solo de datos, y asignar de manera adaptativa el bloque de tratamiento en paralelo de los AM a los múltiples carriles de la PCS de acuerdo con una granularidad de tratamiento en paralelo de los AM.

20 Antes de que los AM sean insertados en el primer flujo de datos, un bloque de datos incluido en el primer flujo de datos puede contener solo datos. Una vez insertados los AM, el primer flujo de datos es un flujo de datos que se forma mezclando todos los datos y los AM. Cuando el primer flujo de datos que incluye los AM es asignado de manera adaptativa a los múltiples carriles de la PCS, para una situación en la que el bloque de datos solo de datos y un bloque de AM en el primer flujo de datos tienen diferentes tamaños, distribuir uniformemente los AM a los carriles de la PCS en esta realización de la presente invención, el primer flujo de datos que incluye los AM puede ser asignado de manera adaptativa utilizando dos granularidades diferentes. Específicamente, para todos los datos en el primer flujo de datos, el bloque de tratamiento en paralelo solo de datos puede ser asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la granularidad del tratamiento en paralelo solo de datos; para los AM en el primer flujo de datos, el bloque de tratamiento en paralelo de los AM es asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la granularidad de tratamiento en paralelo de los AM. Debido a que los AM son suministrados de manera separada, se puede garantizar que los AM sean distribuidos uniformemente en los múltiples carriles de la PCS. Se puede comprender que, si la granularidad del tratamiento en paralelo solo de datos es igual a la granularidad del tratamiento en paralelo de los AM, el primer flujo de datos que incluye los AM puede ser asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la misma granularidad.

35 Tal como se muestra en la figura 2-a, la figura 2-a es un diagrama esquemático de un modo de implementación de la asignación adaptativa de un primer flujo de datos a múltiples carriles de la PCS de acuerdo con una realización de la presente invención. Tal como se muestra en la figura 2-b, la figura 2-b es un diagrama esquemático de otro modo de implementación de asignar de manera adaptativa un primer flujo de datos a múltiples carriles de la PCS de acuerdo con una realización de la presente invención. La asignación adaptativa del primer flujo de datos a los múltiples carriles de la PCS puede ser implementada utilizando un módulo de adaptación (nombre en inglés: Adaptor) en el extremo de transmisión de datos. Una función del módulo de adaptación es suministrar uniformemente los AM a los respectivos carriles de la PCS. El módulo de adaptación realiza el tratamiento de acuerdo con un modo de tratamiento de un bloque de tratamiento en paralelo, donde el bloque de tratamiento en paralelo se puede clasificar en dos tipos de bloques de tratamiento en paralelo de acuerdo con el contenido del primer flujo de datos: un bloque de tratamiento en paralelo solo de datos y un bloque de tratamiento en paralelo de los AM. En la presente invención, se utilizan dos granularidades para realizar respectivamente el tratamiento de suministro en los AM y en todos los datos, que es diferente de una estructura de suministro en la que el suministro se puede realizar solo de acuerdo con una granularidad unificada en la técnica anterior. Sin embargo, cuando los bloques de tratamiento en paralelo de los AM y todos los datos son iguales, las dos granularidades en la presente invención también se pueden referir a la misma granularidad. Tal como se muestra en la figura 2-a, un método para tratar el bloque de tratamiento en paralelo solo de datos en la presente invención debe ser realizado en un modo de suministro de sondeo en el que la programación por turnos se realiza de acuerdo con un símbolo en los carriles de la PCS o un múltiplo entero del símbolo, y el bloque de tratamiento en paralelo solo de datos se suministra de manera separada al carril 0 de la PCS, al carril 1 de la PCS, al carril 2 de la PCS, ..., y al carril i-1 de la PCS. Tal como se muestra en la figura 2-b, en un método para tratar el bloque de tratamiento en paralelo de AM en el primer flujo de datos, los AM se suministran directamente a los carriles de la PCS de acuerdo con la granularidad de tratamiento en paralelo de los AM. El suministro también se realiza en el modo de suministro de sondeo por turnos, y el bloque de tratamiento en paralelo de los AM se suministra de manera separada al carril 0 de la PCS, al carril 1 de la PCS, al carril 2 de la PCS, ..., y al carril i-1 de la PCS. Los tamaños de los AM en los carriles se utilizan como granularidades de suministro para suministrar los AM a los carriles.

5 En algunas realizaciones de la presente invención, el extremo de transmisión de datos incluye, además: un módulo de transformación de ancho de bit, configurado para: después de que el módulo de adaptación asigne de manera adaptativa el primer flujo de datos que incluye los AM a los múltiples carriles de la subcapa física de codificación, PCS, y antes de que el módulo de codificación mediante FEC realice la codificación mediante corrección de errores de reenvío, FEC, en los segundos flujos de datos en los múltiples carriles de la PCS, si un ancho de bit de tratamiento de un circuito de inserción de AM es diferente del ancho de bit de tratamiento de un circuito de codificación mediante FEC, realizar la conversión de ancho de bit en el primer flujo de datos que incluye los AM, donde un ancho de bit del primer flujo de datos convertido es igual al ancho de bit de tratamiento del circuito de codificación mediante FEC.

10 En algunas realizaciones de la presente invención, el extremo de transmisión de datos incluye, además: un módulo de incremento de frecuencia, configurado para: después de que el módulo de adaptación asigne de manera adaptativa el primer flujo de datos que incluye los AM a los múltiples carriles de la subcapa física de codificación, PCS, y antes de que el módulo de codificación mediante FEC realice la codificación mediante corrección de errores de reenvío, FEC, en los segundos flujos de datos en los múltiples carriles de la PCS, asigna dos ciclos de reloj diferentes a dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj para realizar el tratamiento, e inserta datos inactivos en los dos ciclos de reloj diferentes, donde los dos ciclos de reloj diferentes incluyen cada uno un segmento de palabra de código de los dos segmentos de palabra de código que están originalmente concatenados en un ciclo de reloj.

20 En algunas realizaciones de la presente invención, el módulo de codificación mediante FEC está configurado específicamente para: realizar la codificación mediante FEC en datos, excepto por los datos inactivos insertados, en los segundos flujos de datos; y, después de realizar la codificación mediante FEC en los segundos flujos de datos, eliminar los datos inactivos insertados de los segundos flujos de datos codificados para obtener los terceros flujos de datos.

25 En algunas realizaciones de la presente invención, el módulo de incremento de frecuencia está configurado, específicamente para: insertar, entre los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, datos inactivos que ocupan un ciclo de reloj, asignar el ciclo de reloj en el que los dos segmentos de palabra de código están concatenados originalmente a un segmento de palabra de código de los dos segmentos de palabra de código que están concatenados originalmente en un ciclo de reloj y una parte de los datos inactivos en un ciclo de reloj, y asignar un ciclo de reloj a continuación del ciclo de reloj en el que los dos segmentos de palabra de código están concatenados originalmente al otro segmento de palabra de código de los dos segmentos de palabra de código que están concatenados originalmente en un ciclo de reloj, y la otra parte de los datos inactivos en un ciclo de reloj.

30 En algunas realizaciones de la presente invención, el módulo de incremento de frecuencia está configurado específicamente para: insertar datos inactivos después del segmento de palabra de código anterior en los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, de modo que el ciclo de reloj en el que los dos segmentos de palabra de código originalmente concatenados son rellenados, realizar el desplazamiento de datos hacia atrás en el último segmento de palabra de código en los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, e insertar datos inactivos después del último segmento de palabra de código en una palabra de código en la que el último segmento de palabra de código está situado, de modo que se rellene un ciclo de reloj al que pertenece el último segmento de palabra de código en la palabra de código en la que está situado el último segmento de palabra de código.

35 En algunas realizaciones de la presente invención, el módulo de codificación mediante FEC está configurado específicamente para: obtener dos segmentos de palabra de código que están en los segundos flujos de datos y son tratados en paralelo en un mismo ciclo de reloj, donde los dos segmentos de palabra de código incluyen un primer segmento de palabra de código y un segundo segmento de palabra de código, y un volumen de datos del primer segmento de palabra de código es mayor o igual que el del segundo segmento de palabra de código; y realizar la codificación mediante FEC en el primer segmento de palabra de código utilizando un primer circuito de codificación mediante FEC, y realizar la codificación mediante FEC en el segundo segmento de palabra de código utilizando un segundo circuito de codificación mediante FEC, donde los terceros flujos de datos son generados después de que el primer segmento de palabra de código y el segundo segmento de palabra de código son tratados respectivamente por el primer circuito de codificación mediante FEC y el segundo circuito de codificación mediante FEC, y el segundo circuito de codificación mediante FEC y el primer circuito de codificación mediante FEC utilizan circuitos lógicos de codificación configurables completamente diferentes, o el segundo circuito de codificación mediante FEC es un circuito de codificación mediante FEC obtenido después de que se realiza una operación de configuración de recursos en un circuito lógico de codificación configurable, excepto por un circuito lógico de codificación configurable que trata el primer segmento de palabra de código, en el primer circuito de codificación mediante FEC.

40 En algunas realizaciones de la presente invención, el extremo de transmisión de datos incluye, además: un módulo de conversión de ancho de bit, configurado para: después de que el módulo de codificación mediante FEC realiza la codificación mediante corrección de errores de reenvío, FEC, en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos, y antes de que el módulo de suministro de PMA suministre los terceros flujos de datos a los múltiples carriles de la subcapa física de conexión a medio, PMA, de acuerdo con el ancho de bit de entrada del serializador / deserializador, Serdes, si un ancho de bit de tratamiento de un circuito de

inserción de AM es diferente de un ancho de bit de tratamiento de un circuito de codificación mediante FEC, realice la conversión de ancho de bit en los terceros flujos de datos, donde los anchos de bit de los terceros flujos de datos convertidos son iguales al ancho de bit de tratamiento del circuito de inserción de AM.

5 En algunas realizaciones de la presente invención, el módulo de suministro de flujo de datos está configurado específicamente para: determinar un tamaño de segmento de acuerdo con una cantidad de carriles de la PCS y un tamaño de datos de un solo carácter soportado en cada carril de la PCS, y segmentar el tercer flujo de datos de acuerdo con el tamaño del segmento para obtener los terceros flujos de datos segmentados, donde cada segmento incluye múltiples terceros bloques de datos; y determinar, de acuerdo con los anchos de bit de los terceros flujos de datos, con el ancho de bit de entrada de los Serdes y con el tamaño del segmento, una cantidad de segmentos que
10 deben ser suministrados, extraer, de los terceros flujos de datos segmentados, terceros bloques de datos segmentados correspondientes a la cantidad de segmentos, y, a continuación, suministrar los terceros bloques de datos a un mismo carril de la PMA en los múltiples carriles de la PMA después de superponer terceros bloques de datos que están en una misma ubicación en los segmentos correspondientes a la cantidad de segmentos.

15 En algunas realizaciones de la presente invención, el módulo de suministro de flujo de datos está configurado específicamente para: la cantidad de carriles de la PCS es i , el tamaño de datos de un solo carácter soportado en cada carril de la PCS es m , un tamaño de un tercer bloque de datos en cada segmento obtenido después de que los terceros flujos de datos sean segmentados es $i \times m$, y los anchos de bit de los terceros flujos de datos son $i \times m \times$ Entero 1, donde el Entero 1 es un entero positivo preestablecido; el ancho de bit de entrada de Serdes es $i \times m \times$ Entero 2, el tamaño del segmento es $i \times m \times$ Entero S, y por lo tanto, tanto Entero 1 como Entero 2 son múltiplos
20 enteros de Entero S, y la cantidad de segmentos es Entero 2 / Entero S, donde Entero 2 es un entero positivo preestablecido; y extraer el tercer bloque de datos de orden n de cada segmento, y suministrar Entero 2 / Entero S terceros bloques de datos de orden n extraídos en total hasta el carril de la PMA de orden n después de superponer el Entero 2 / Entero S terceros bloques de datos de orden n extraídos, donde el valor de n es cualquier valor de 0 a $i-1$.

25 Se puede aprender, a partir de las descripciones de la realización anterior de la presente invención que los marcadores de alineación, AM, son insertados en un primer flujo de datos, donde el primer flujo de datos es un flujo de datos que es transcodificado y aleatorizado después de realizar la codificación mediante 64b / 66b, y la velocidad de transmisión del primer flujo de datos es mayor o igual a 100 G; el primer flujo de datos que incluye los AM es asignado de manera adaptativa a múltiples carriles de la subcapa física de codificación, PCS, para obtener segundos flujos de datos; la
30 codificación mediante corrección de errores de reenvío, FEC, se realiza en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos; y los terceros flujos de datos son enviados a múltiples carriles de la subcapa física de conexión a medio, PMA, de acuerdo con un ancho de bit de entrada de un serializador / deserializador, Serdes, donde un tercer flujo de datos suministrado a cada carril de la PMA incluye un AM completo y continuo. Debido a que, en la presente invención, los terceros flujos de datos se suministran a los múltiples carriles de la PMA, y el AM en cada carril de la PMA puede seguir siendo continuo y completo, un extremo de recepción de datos puede obtener AM completos y continuos de los múltiples carriles de la PMA, lo que pueden ser aplicable a un
35 escenario en el que la velocidad de transmisión de un flujo de datos es bastante alta, y evitar un daño a los AM.

40 En referencia a la figura 12, una realización no reivindicada de la presente invención da a conocer un extremo de recepción de datos 1200, que puede incluir un módulo de demodulación de PMA 1201, un módulo de alineación y reasignación de AM 1202, un módulo de decodificación mediante FEC 1203, un módulo de adaptación 1204 y un módulo de eliminación de AM 1205.

El módulo de demodulación de PMA 1201 está configurado para: recibir cuartos flujos de datos de múltiples carriles de la subcapa física de conexión a medio, PMA, y realizar la demodulación de PMA en los cuartos flujos de datos para obtener los terceros flujos de datos que se obtienen después de que se realiza la demodulación de PMA.

45 El módulo de alineación y reasignación de AM 1202 está configurado para: identificar los AM de marcadores de alineación completos y continuos de los terceros flujos de datos, y realizar la alineación y reasignación en los terceros flujos de datos en los carriles de la subcapa física de codificación, PCS, de acuerdo con los AM para obtener los terceros flujos de datos alineados y reasignados.

50 El módulo de decodificación mediante FEC 1203 está configurado para realizar la decodificación mediante corrección de errores de reenvío, FEC, en los terceros flujos de datos alineados y reasignados para obtener segundos flujos de datos, donde los segundos flujos de datos incluyen los AM.

El módulo de adaptación 1204 está configurado para: combinar los segundos flujos de datos que están en los carriles de la PCS y son obtenidos mediante decodificación mediante FEC para obtener un primer flujo de datos que incluye los AM.

55 El módulo de eliminación de AM 1205 está configurado para eliminar los AM del primer flujo de datos que incluye los AM para obtener un primer flujo de datos original utilizado para codificar en un lado del codificador de datos.

En algunas realizaciones de la presente invención, el módulo de demodulación de PMA está configurado específicamente para: determinar, de acuerdo con los anchos de bit de los cuartos flujos de datos, un ancho de bit de

5 entrada de un serializador / deserializador, Serdes, y un tamaño de segmento, la cantidad de segmentos que deben ser combinados, obtener, de cada carril de la PMA, terceros bloques de datos que están en la misma ubicación en segmentos correspondientes a la cantidad de segmentos, suministrar respectivamente los terceros bloques de datos a los carriles de la PCS y combinar terceros bloques de datos en cada carril de la PCS para obtener los terceros flujos de datos combinados.

10 En algunas realizaciones de la presente invención, el módulo de demodulación de PMA está configurado específicamente para: una cantidad de carriles de la PCS es i , un tamaño de datos de un solo carácter soportados en cada carril de la PCS es m y los anchos de bit de los terceros flujos de datos es $i \times m \times \text{Entero } 1$, donde Entero 1 es un entero positivo preestablecido; el ancho de bit de entrada de Serdes es $i \times m \times \text{Entero } 2$, el tamaño del segmento es $i \times m \times \text{Entero } S$, y por lo tanto, tanto Entero 1 como Entero 2 son múltiplos enteros de Entero S, y la cantidad de segmentos es $\text{Entero } 1 / \text{Entero } S$, donde Entero 2 es un entero positivo preestablecido; el tamaño de un tercer bloque de datos que está situado en cada segmento y se obtiene de cada carril de la PMA es $i \times m$; y obtener Entero 1 / Entero S terceros bloques de datos de orden n en total desde el carril de la PMA, de orden n suministran respectivamente Entero 1 / Entero S terceros bloques de datos de orden n a los carriles de la PCS, y combinar los terceros bloques de datos en cada carril de la PCS para obtener los terceros flujos de datos cuyos anchos de bit son $i \times m \times \text{Entero } 1$.

20 En algunas realizaciones de la presente invención, el extremo de recepción de datos incluye, además, un módulo de conversión de ancho de bit, configurado para: antes de que el módulo de decodificación mediante FEC realice la decodificación mediante corrección de errores de reenvío, FEC, en los terceros flujos de datos alineados y reasignados, si un ancho de bit de tratamiento de un circuito de eliminación de AM es diferente del ancho de bit de tratamiento de un circuito de decodificación mediante FEC, realizar la conversión de ancho de bit en los terceros flujos de datos alineados y reasignados, donde los anchos de bit de los terceros flujos de datos cuyos anchos de bit son convertidos son iguales al ancho de bit de tratamiento del circuito de decodificación mediante FEC.

25 En algunas realizaciones de la presente invención, el extremo de recepción de datos incluye, además: un módulo de incremento de frecuencia, configurado para: antes de que el módulo de decodificación mediante FEC esté configurado para realizar la decodificación mediante corrección de errores de reenvío, FEC, en los terceros flujos de datos alineados y reasignados, asignar dos ciclos de reloj diferentes a dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj para realizar el tratamiento, e insertar datos inactivos en los dos ciclos de reloj diferentes, donde los dos ciclos de reloj diferentes incluyen cada uno un segmento de palabra de código de los dos segmentos de palabra de código que están originalmente concatenados en un ciclo de reloj.

30 En algunas realizaciones de la presente invención, el módulo de decodificación mediante FEC está configurado específicamente para: realizar decodificación mediante FEC sobre datos, excepto los datos inactivos insertados, en los terceros flujos de datos alineados y reasignados; y después de realizar la decodificación mediante FEC en los terceros flujos de datos, eliminar los datos inactivos insertados de los terceros flujos de datos decodificados para obtener los segundos flujos de datos.

35 En algunas realizaciones de la presente invención, el módulo de incremento de frecuencia está configurado, específicamente, para: insertar, entre los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, datos inactivos que ocupan un ciclo de reloj, asignar el ciclo de reloj en el que el dos segmentos de palabra de código están concatenados originalmente a un segmento de palabra de código de los dos segmentos de palabra de código que están concatenados originalmente en un ciclo de reloj y una parte de los datos inactivos en un ciclo de reloj, y asignar un ciclo de reloj a continuación del ciclo de reloj en el que los dos segmentos de palabra de código están concatenados originalmente al otro segmento de palabra de código de los dos segmentos de palabra de código que están concatenados originalmente en un ciclo de reloj y la otra parte de los datos inactivos en un ciclo de reloj.

40 En algunas realizaciones de la presente invención, el módulo de incremento de frecuencia está configurado específicamente para: insertar datos inactivos después del segmento de palabra de código anterior en los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, de modo que el ciclo de reloj en el que los dos segmentos de palabra de código originalmente concatenados son rellenos, realizan el desplazamiento de datos hacia atrás en el último segmento de palabra de código en los dos segmentos de palabra de código que son tratados en paralelo en un mismo ciclo de reloj, e inserta datos inactivos después del último segmento de palabra de código en una palabra de código en la que el último segmento de palabra de código segmento está situado, de modo que se rellene un ciclo de reloj al que pertenece el último segmento de palabra de código en la palabra de código en la que está situado el último segmento de palabra de código.

45 En algunas realizaciones de la presente invención, el módulo de decodificación mediante FEC está configurado específicamente para: obtener dos segmentos de palabra de código que están en los terceros flujos de datos y son tratados en paralelo en un mismo ciclo de reloj, donde los dos segmentos de palabra de código incluyen un primer segmento de palabra de código y un segundo segmento de palabra de código, y un volumen de datos del primer segmento de palabra de código es mayor o igual que el del segundo segmento de palabra de código; y realizar la decodificación mediante FEC en el primer segmento de palabra de código utilizando un primer circuito de decodificación mediante FEC, y realizar la decodificación mediante FEC en el segundo segmento de palabra de código

utilizando un segundo circuito de decodificación mediante FEC, donde los segundos flujos de datos son generados después de que el primer segmento de palabra de código y el segundo segmento de palabra de código sean tratados respectivamente por el primer circuito de decodificación mediante FEC y el segundo circuito de decodificación mediante FEC, y el segundo circuito de decodificación mediante FEC y el primer circuito de decodificación mediante FEC utilizan circuitos lógicos de decodificación configurables completamente diferentes, o el segundo circuito de decodificación mediante FEC es un circuito de decodificación mediante FEC obtenido después una operación de configuración de recursos que se realiza en un circuito lógico de decodificación configurable, excepto por un circuito lógico de decodificación configurable que trata el primer segmento de palabra de código, en el primer circuito de decodificación mediante FEC.

En algunas realizaciones de la presente invención, el extremo de recepción de datos incluye, además: un módulo de conversión de ancho de bit, configurado para: antes de que el módulo de eliminación de AM elimine los AM del primer flujo de datos que incluye los AM, si un ancho de bit de tratamiento de un circuito de eliminación de AM es diferente del ancho de bit de tratamiento de un circuito de decodificación mediante FEC, realizar la conversión de ancho de bit en el primer flujo de datos que incluye los AM, donde un ancho de bit del primer flujo de datos convertido es igual al ancho de bit del circuito de eliminación de AM.

En algunas realizaciones de la presente invención, el módulo de adaptación está configurado específicamente para: extraer un bloque de tratamiento en paralelo solo de datos de acuerdo con una granularidad de tratamiento en paralelo solo de datos, extraer un bloque de tratamiento en paralelo de los AM de acuerdo con una granularidad de tratamiento en paralelo de los AM y combinar los bloques de tratamiento en paralelo solo de datos y el bloque de tratamiento en paralelo de los AM para obtener el primer flujo de datos que incluye los AM.

En algunas realizaciones de la presente invención, el suministro de un primer flujo de datos que incluye AM a múltiples carriles de la subcapa física de codificación, PCS, se utiliza como ejemplo, e incluye:

el primer flujo de datos incluye un bloque de tratamiento en paralelo solo de datos y un bloque de tratamiento en paralelo de los AM; asignar de manera adaptativa el bloque de tratamiento en paralelo solo de datos a los múltiples carriles de la PCS de acuerdo con una granularidad de tratamiento en paralelo solo de datos, y asignar de manera adaptativa el bloque de tratamiento en paralelo de los AM a los múltiples carriles de la PCS de acuerdo con una granularidad de tratamiento en paralelo de los AM.

Antes de que los AM sean insertados en el primer flujo de datos, un bloque de datos incluido en el primer flujo de datos puede contener solo datos. Una vez insertados los AM, el primer flujo de datos es un flujo de datos que se forma mezclando todos los datos y los AM. Cuando el primer flujo de datos que incluye los AM es asignado de manera adaptativa a los múltiples carriles de la PCS, para una situación en la que el bloque de datos solo de datos y un bloque de AM en el primer flujo de datos tienen diferentes tamaños, distribuir uniformemente los AM a los carriles de la PCS en esta realización de la presente invención, el primer flujo de datos que incluye los AM puede ser asignado de manera adaptativa utilizando dos granularidades diferentes. Específicamente, para todos los datos en el primer flujo de datos, el bloque de tratamiento en paralelo solo de datos puede ser asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la granularidad del tratamiento en paralelo solo de datos; para los AM en el primer flujo de datos, el bloque de tratamiento en paralelo de los AM es asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la granularidad de tratamiento en paralelo de los AM. Debido a que los AM son suministrados de manera separada, se puede garantizar que los AM se distribuyan uniformemente en los múltiples carriles de la PCS. Se puede comprender que, si la granularidad del tratamiento en paralelo solo de datos es igual a la granularidad del tratamiento en paralelo de los AM, el primer flujo de datos que incluye los AM puede ser asignado de manera adaptativa a los múltiples carriles de la PCS de acuerdo con la misma granularidad.

Tal como se muestra en la figura 2-a, la figura 2-a es un diagrama esquemático de un modo de implementación de la asignación adaptativa de un primer flujo de datos a múltiples carriles de la PCS de acuerdo con una realización de la presente invención. Tal como se muestra en la figura 2-b, la figura 2-b es un diagrama esquemático de otro modo de implementación de asignar de manera adaptativa un primer flujo de datos a múltiples carriles de la PCS de acuerdo con una realización de la presente invención. La asignación adaptativa del primer flujo de datos a los múltiples carriles de la PCS puede ser implementada utilizando un módulo de adaptación en el extremo de transmisión de datos. Una función del módulo de adaptación es suministrar uniformemente los AM a los respectivos carriles de la PCS. El módulo de adaptación realiza el tratamiento de acuerdo con un modo de tratamiento de un bloque de tratamiento en paralelo, donde el bloque de tratamiento en paralelo se puede clasificar en dos tipos de bloques de tratamiento en paralelo de acuerdo con el contenido del primer flujo de datos: un bloque de tratamiento en paralelo solo de datos y un bloque de tratamiento en paralelo de los AM. En la presente invención, se utilizan dos granularidades para realizar respectivamente el tratamiento de suministro sobre los AM y sobre todos los datos, que es diferente de una estructura de suministro en la que la suministro se puede realizar solo de acuerdo con una granularidad unificada en la técnica anterior. Sin embargo, cuando los bloques de tratamiento en paralelo de los AM y de todos los datos son iguales, las dos granularidades en la presente invención también pueden referirse a la misma granularidad. Tal como se muestra en la figura 2-a, un método para tratar el bloque de tratamiento en paralelo solo de datos en la presente invención debe ser realizado en un modo de suministro de sondeo en el que la programación por turnos se realiza de acuerdo con un símbolo en los carriles de la PCS o un múltiplo entero del símbolo, y el bloque de tratamiento en paralelo solo de datos se suministra por separado al carril 0 de la PCS, al carril 1 de la PCS, al carril 2 de la PCS, ..., y al carril i-1

de la PCS. Tal como se muestra en la figura 2-b, en un método para tratar el bloque de tratamiento en paralelo de los AM en el primer flujo de datos, los AM se suministran directamente a los carriles de la PCS de acuerdo con la granularidad de tratamiento en paralelo de los AM. El suministro también se realiza en el modo de suministro de sondeo por turnos, y el bloque de tratamiento en paralelo de los AM se suministra de manera separada al carril 0 de la PCS, al carril 1 de la PCS, al carril 2 de la PCS, ..., y al carril i-1 de la PCS. Los tamaños de los AM en los carriles se utilizan como granularidades de suministro para suministrar los AM a los carriles.

Se puede aprender a partir de las descripciones de la realización anterior de la presente invención que los cuartos flujos de datos son recibidos desde múltiples carriles de la subcapa física de conexión a medio, PMA, y la demodulación de PMA se realiza en los cuartos flujos de datos para obtener los terceros flujos de datos que se obtienen después de que se realice la demodulación de la PMA; los AM completos y continuos se identifican a partir de los terceros flujos de datos, y la alineación y reasignación se realiza en los terceros flujos de datos en los carriles de la PCS de acuerdo con los AM para obtener los terceros flujos de datos alineados y reasignados; la decodificación mediante FEC se realiza en los terceros flujos de datos alineados y reasignados para obtener segundos flujos de datos, donde los segundos flujos de datos incluyen los AM; los segundos flujos de datos que están en los carriles de la PCS y se obtienen mediante decodificación mediante FEC son combinados para obtener un primer flujo de datos que incluye los AM; y los AM son eliminados del primer flujo de datos que incluye los AM para obtener un primer flujo de datos original utilizado para codificar en el lado del codificador de datos. Debido a que en la presente invención, los terceros flujos de datos se suministran a los múltiples carriles de la PMA, y un AM en cada carril de la PMA puede seguir siendo continuo y completo, un extremo de recepción de datos puede obtener AM completos y continuos de los múltiples carriles de la PMA, lo que puede ser aplicable a un escenario en el que la velocidad de transmisión de un flujo de datos es bastante alta, y evitar un daño a los AM.

Cabe señalar que el contenido, tal como el intercambio de información entre los módulos / unidades del aparato y los procesos de ejecución del mismo, se basa en la misma idea que los modos de realización del método de la presente invención, y produce los mismos efectos técnicos que los modos de realización del método de la presente invención. Para el contenido específico, se puede hacer referencia a la descripción anterior en las realizaciones del método de la presente invención, y los detalles no se describen nuevamente en el presente documento.

Además, cabe señalar que la realización del aparato descrita es meramente a modo de ejemplo. Las unidades descritas como partes separadas pueden estar o no físicamente separadas, y las partes mostradas como unidades pueden o no ser unidades físicas, pueden estar situadas en una posición o pueden estar distribuidas en una pluralidad de unidades de la red. Algunos o todos los módulos pueden ser seleccionados de acuerdo con las necesidades reales para conseguir los objetivos de las soluciones de las realizaciones. Además, en los dibujos adjuntos de las realizaciones del aparato dadas a conocer por la presente invención, las relaciones de conexión entre módulos indican que los módulos tienen conexiones de comunicación entre sí, que pueden ser implementadas específicamente como uno o más buses de comunicaciones o cables de señal. Los expertos en la técnica pueden comprender e implementar las realizaciones de la presente invención sin esfuerzos creativos.

Basándose en la descripción de los modos de implementación citados anteriormente, las personas expertas en la técnica pueden comprender claramente que la presente invención puede ser implementada mediante software además del hardware universal necesario, o mediante un hardware exclusivo, incluyendo un circuito integrado exclusivo, una CPU exclusiva, una memoria exclusiva, un componente exclusivo y similares. En general, cualquier función que pueda ser realizada por un programa informático puede ser implementada fácilmente utilizando el hardware correspondiente. Además, una estructura de hardware específica utilizada para conseguir una misma función puede ser de diversas formas, por ejemplo, en forma de circuito analógico, circuito digital, circuito exclusivo o similar. Sin embargo, como para la presente invención, la implementación del programa de software es un mejor modo de implementación en la mayoría de los casos. Sobre la base de dicha comprensión, las soluciones técnicas de la presente invención esencialmente, o la parte que contribuye a la técnica anterior, se pueden implementar en forma de un producto de software. El producto de software se almacena en un medio de almacenamiento legible, tal como un disquete, una unidad flash de USB, un disco duro extraíble, una memoria de solo lectura (ROM, Read Only Memory, en inglés), una memoria de acceso aleatorio (RAM, Random Access Memory, en inglés), un disco magnético o un disco óptico de un ordenador, e incluye varias instrucciones para indicar a un dispositivo informático (que puede ser un ordenador personal, un servidor, un dispositivo de red o similar) que realice los métodos descritos en las realizaciones de la presente invención.

Las realizaciones anteriores pretenden simplemente describir las soluciones técnicas de la presente invención, pero no limitar la presente invención. Aunque la presente invención se describe en detalle con referencia a las realizaciones anteriores, las personas con conocimientos ordinarios en la técnica deben comprender que aún se pueden realizar modificaciones a las soluciones técnicas descritas en las realizaciones anteriores o hacer sustituciones equivalentes a algunas características técnicas de las mismas, cuando las modificaciones y las sustituciones equivalentes se encuentren dentro del alcance de las reivindicaciones.

REIVINDICACIONES

1. Un método de tratamiento de datos, que comprende:

5 insertar (101), mediante un circuito de inserción de marcadores de alineación, AM, múltiples AM en un primer flujo de datos, en donde el primer flujo de datos es un flujo de datos que es transcodificado y aleatorizado después de ser codificado en una capa física, y la velocidad de transmisión del primer flujo de datos es mayor o igual a 100 Gbps;

10 asignar (102) de manera adaptativa el primer flujo de datos que comprende los AM a múltiples carriles de la subcapa física de codificación, PCS, para obtener segundos flujos de datos, en donde un ancho de bit de tratamiento del circuito de inserción de AM está representado por P y hay i carriles de la PCS, donde los carriles de la PCS son el carril 0 de la PCS al carril i-1 de la PCS y el primer flujo de datos cuyo tamaño de datos es P es enviado al carril 0 de la PCS al carril i-1 de la PCS respectivamente;

realizar (103), mediante un circuito de codificación mediante corrección de errores de reenvío, FEC, codificación mediante FEC en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos, en donde un ancho de bit de tratamiento del circuito de codificación mediante FEC está representado por F; y

15 suministrar (104) los terceros flujos de datos a múltiples carriles de la subcapa física de conexión a medio, PMA, de acuerdo con un ancho de bit de entrada de un serializador / deserializador, Serdes, para obtener múltiples cuartos flujos de datos, en donde los múltiples cuartos flujos de datos están en una correspondencia de uno a uno con los múltiples carriles de la PMA, cada cuarto flujo de datos comprende, como mínimo, un AM completo y continuo, y el, como mínimo, un AM es un AM en los múltiples AM;

20 en donde después de la asignación adaptativa del primer flujo de datos que comprende los AM a múltiples carriles de la subcapa física de codificación, PCS, y antes de realizar la codificación mediante FEC en los segundos flujos de datos en los múltiples carriles de la PCS, el método comprende, además:

25 si el ancho de bit de tratamiento P del circuito de inserción de AM es diferente del ancho de bit de tratamiento F del circuito de codificación mediante FEC, realizar la conversión de ancho de bit en el primer flujo de datos que comprende los AM, en donde un ancho de bit del primer flujo de datos convertido es igual al ancho de bit de tratamiento F del circuito de codificación mediante FEC;

en donde, después de realizar la codificación mediante FEC en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos, y antes de suministrar los terceros flujos de datos a múltiples carriles de la PMA de acuerdo con un ancho de bit de entrada de un serializador / deserializador, Serdes, el método comprende, además:

30 si el ancho de bit de tratamiento P del circuito de inserción de AM es diferente del ancho de bit de tratamiento F del circuito de codificación mediante FEC, realizar la conversión de ancho de bit en los terceros flujos de datos, en donde los anchos de bit de los terceros flujos de datos convertidos son iguales al ancho de bit de tratamiento P del circuito de inserción de AM.

2. Un extremo de transmisión de datos (1100), que comprende:

35 un módulo de inserción de marcador de alineación, AM, (1101), configurado para insertar múltiples marcadores de alineación, AM, en un primer flujo de datos, en donde el primer flujo de datos es un flujo de datos que es transcodificado y aleatorizado después de ser codificado en una capa física, y una velocidad de transmisión del primer flujo de datos es mayor o igual a 100 Gbps;

40 un módulo de adaptación (1102), configurado para asignar adaptativamente el primer flujo de datos que comprende los AM a múltiples carriles de la subcapa física de codificación, PCS, para obtener segundos flujos de datos, en donde el ancho de bit de tratamiento del módulo de inserción de AM está representado por P y existen i carriles de la PCS, donde los carriles de la PCS son el carril 0 de la PCS al carril i-1 de la PCS, y el primer flujo de datos cuyo tamaño de datos es P es suministrado al carril 0 de la PCS al carril i-1 de la PCS, respectivamente;

45 un módulo de codificación mediante corrección de errores de reenvío, FEC, (1103), configurado para realizar la codificación mediante FEC en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos, en donde el ancho de bit de tratamiento del módulo de codificación mediante FEC está representado por F; y

50 un módulo de suministro de flujo de datos (1104), configurado para suministrar el tercer flujo de datos a múltiples carriles de la subcapa física de conexión a medio, PMA, de acuerdo con un ancho de bit de entrada de un serializador / deserializador, Serdes, para obtener múltiples cuartos flujos de datos, en donde los múltiples cuartos flujos de datos están en correspondencia de uno a uno con los múltiples carriles de la PMA, cada cuarto flujo de datos comprende, como mínimo, un AM completo y continuo, y el, como mínimo, un AM es un AM en los múltiples AM;

en donde el extremo de transmisión de datos (1100) comprende, además: un módulo de transformación de ancho de bit, configurado para: después de que el módulo de adaptación asigne de manera adaptativa el primer flujo de datos

5 que comprende los AM a los múltiples carriles de la PCS, y antes de que el módulo de codificación mediante FEC realice la codificación mediante FEC en los segundos flujos de datos en los múltiples carriles de la PCS, si el ancho de bit de tratamiento P del módulo de inserción de AM es diferente del ancho de bit de tratamiento F del módulo de codificación mediante FEC, realizar la conversión del ancho de bit en el primer flujo de datos que comprende los AM, en donde el ancho de bit del primer flujo de datos convertido es igual al ancho de bit de tratamiento F del módulo de codificación mediante FEC; y

10 en donde el extremo de transmisión de datos (1100) comprende, además: un módulo de conversión de ancho de bit, configurado para: después de que el módulo de codificación mediante FEC realiza la codificación mediante FEC en los segundos flujos de datos en los múltiples carriles de la PCS para obtener los terceros flujos de datos, y antes del módulo de suministro de PMA suministre los terceros flujos de datos a los múltiples carriles de la PMA de acuerdo con el ancho de bit de entrada del serializador / deserializador, Serdes, si el ancho de bit de tratamiento P del módulo de inserción de AM es diferente del ancho de bit de tratamiento F del módulo de codificación mediante FEC, realizar la conversión de ancho de bit en los terceros flujos de datos, en donde los anchos de bit de los terceros flujos de datos convertidos son iguales al ancho de bit de tratamiento P del módulo de inserción de AM.

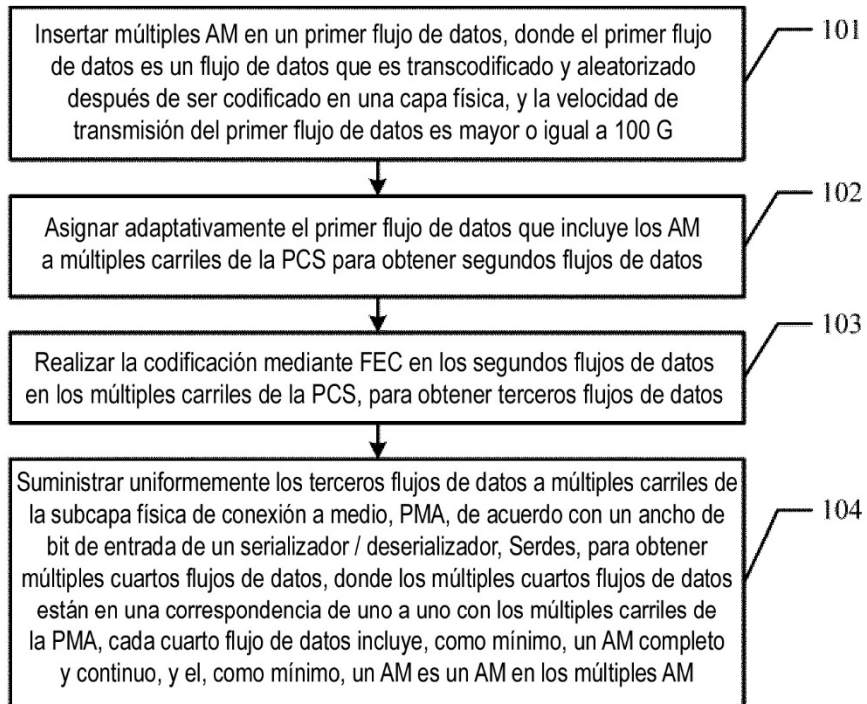


FIG. 1

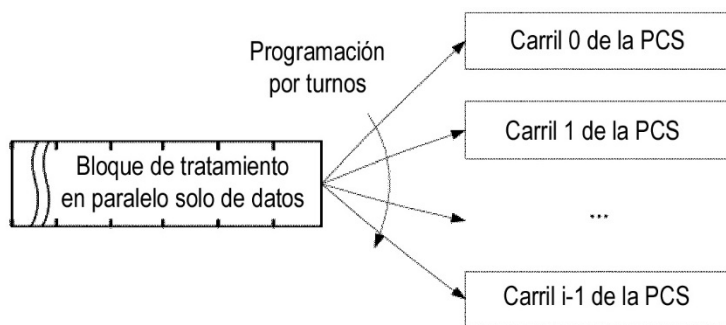


FIG. 2-a

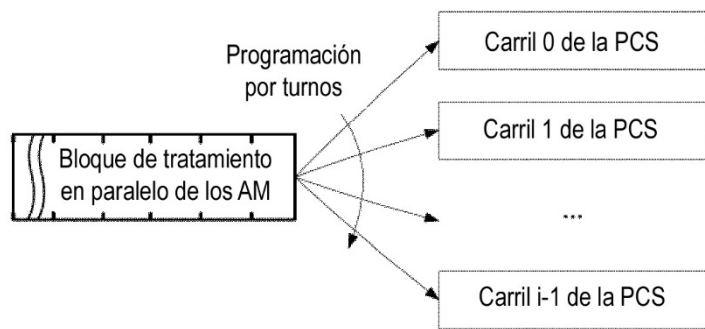


FIG. 2-b

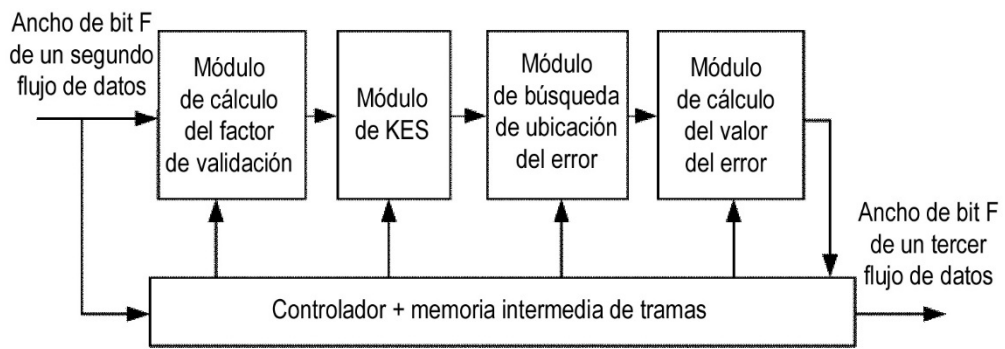


FIG. 3

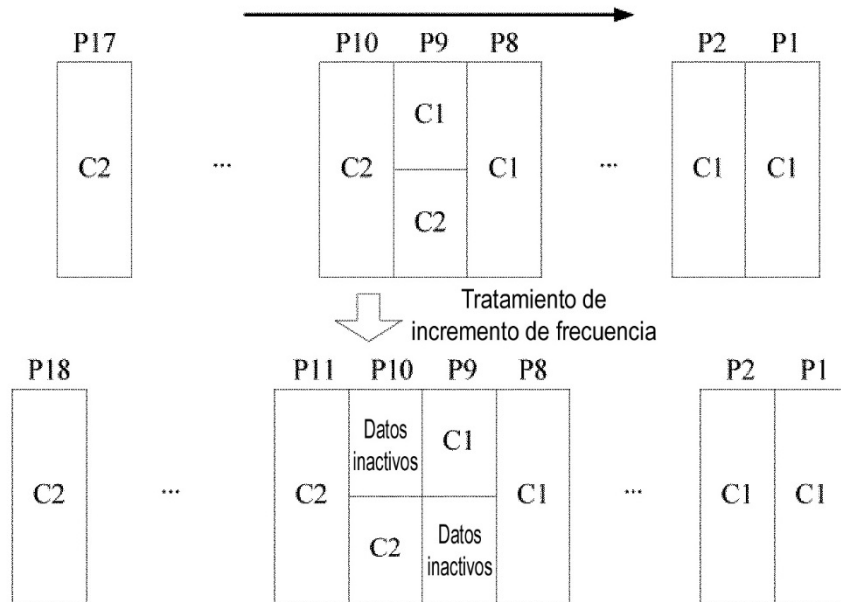


FIG. 4-a

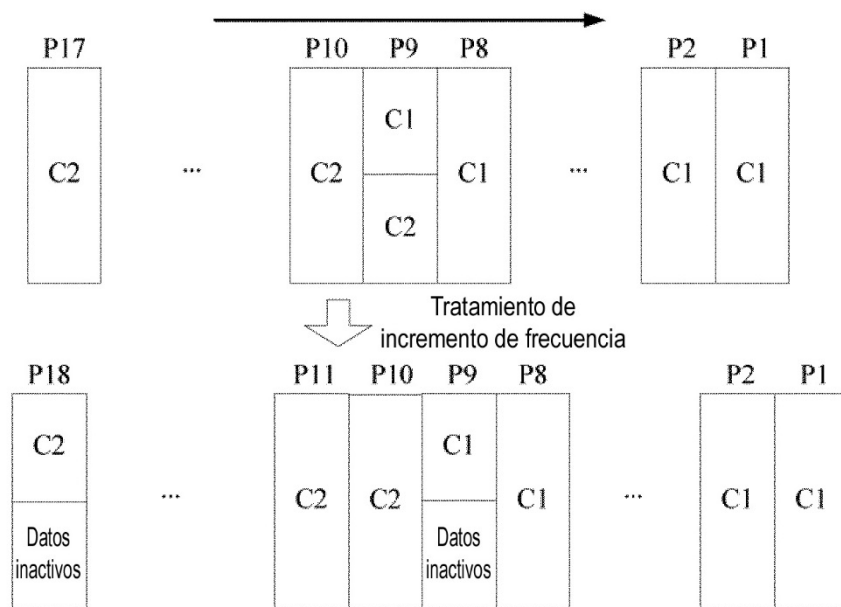


FIG. 4-b

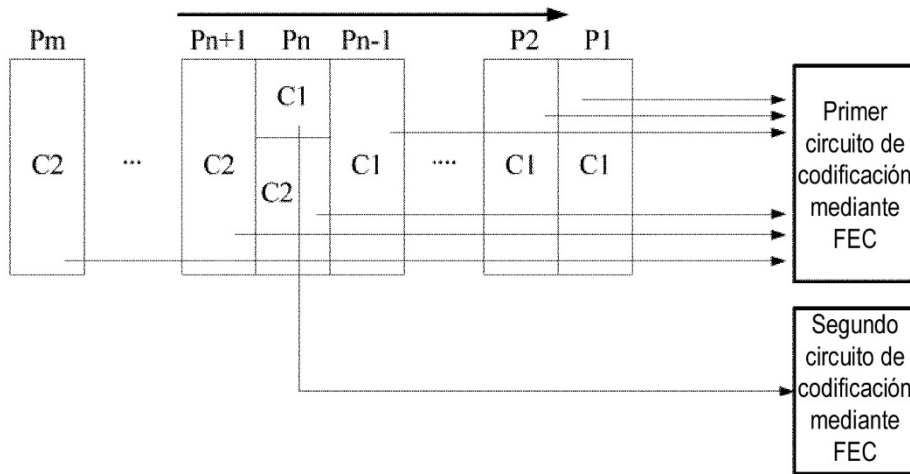


FIG. 4-c

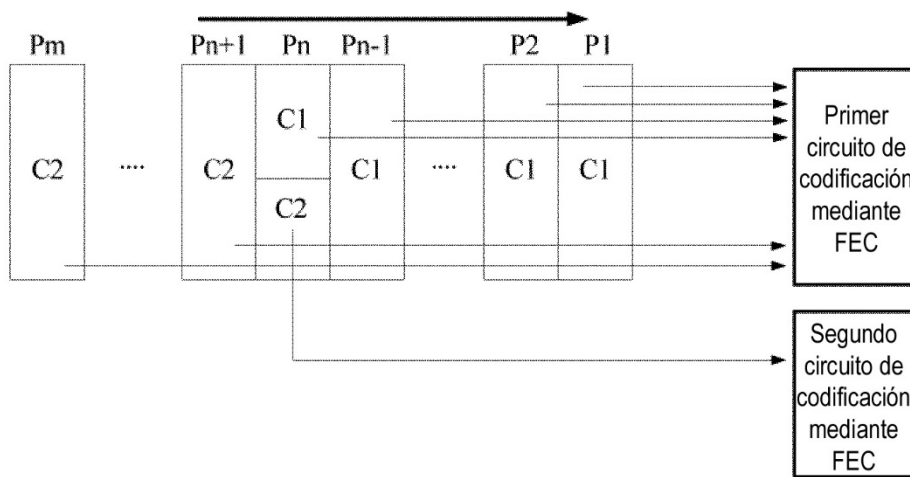


FIG. 4-d

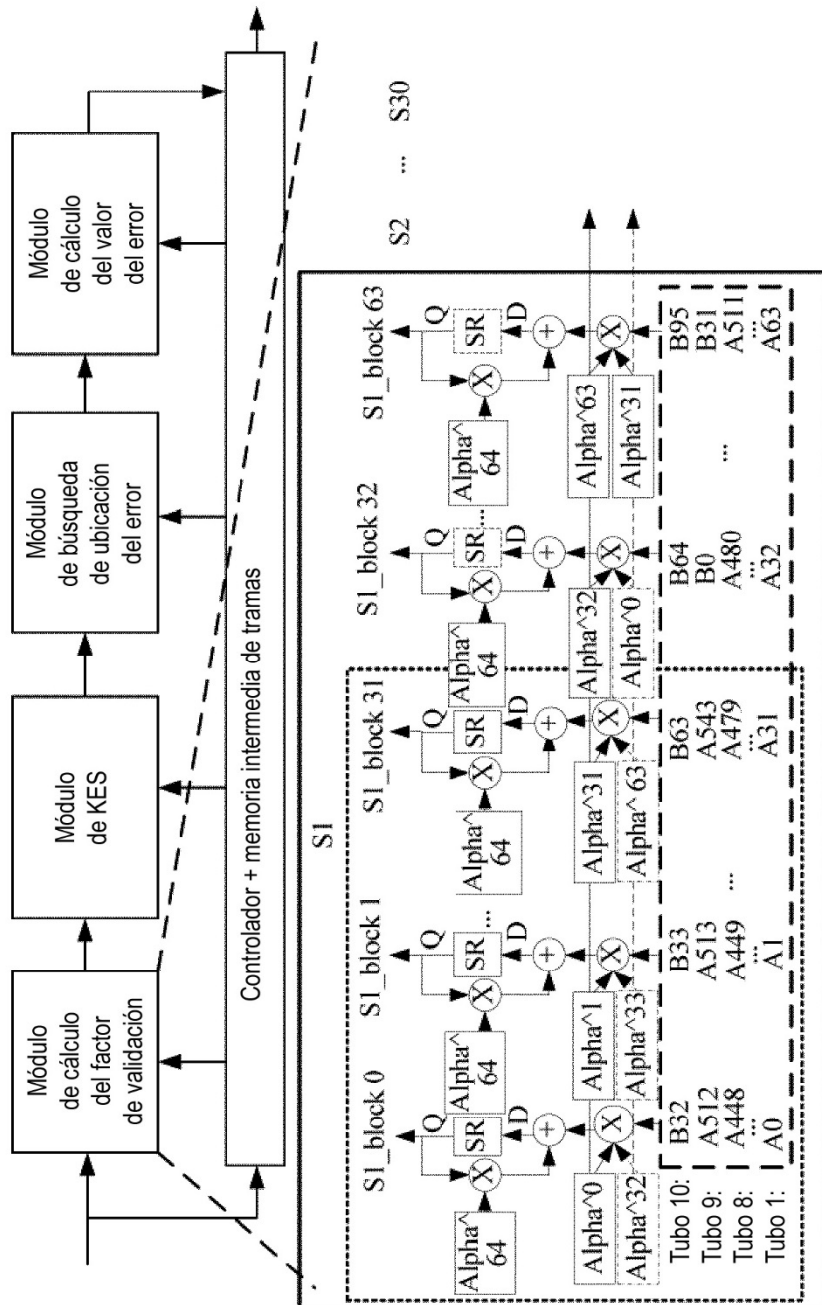


FIG 4-e

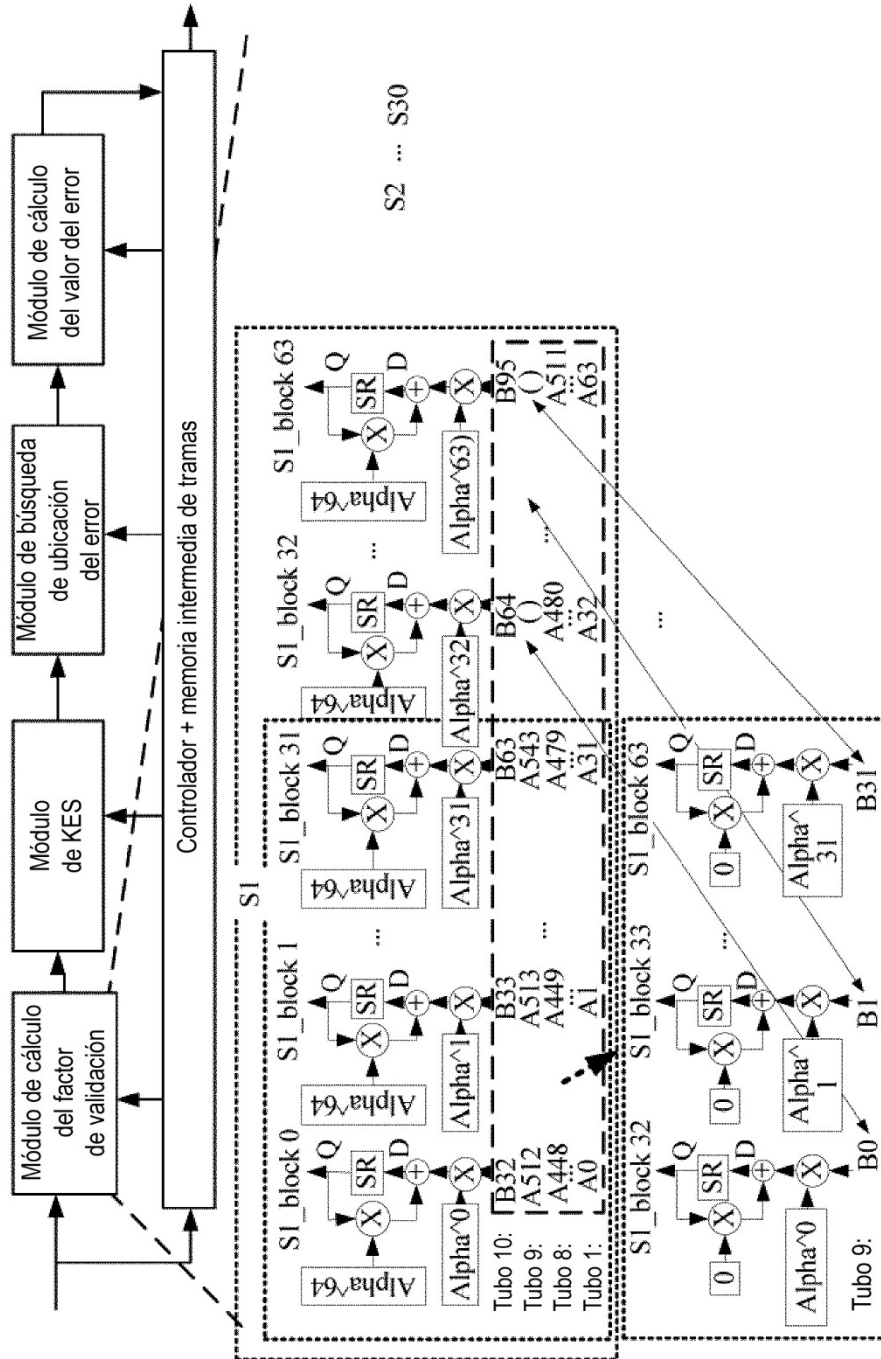


FIG. 4-f

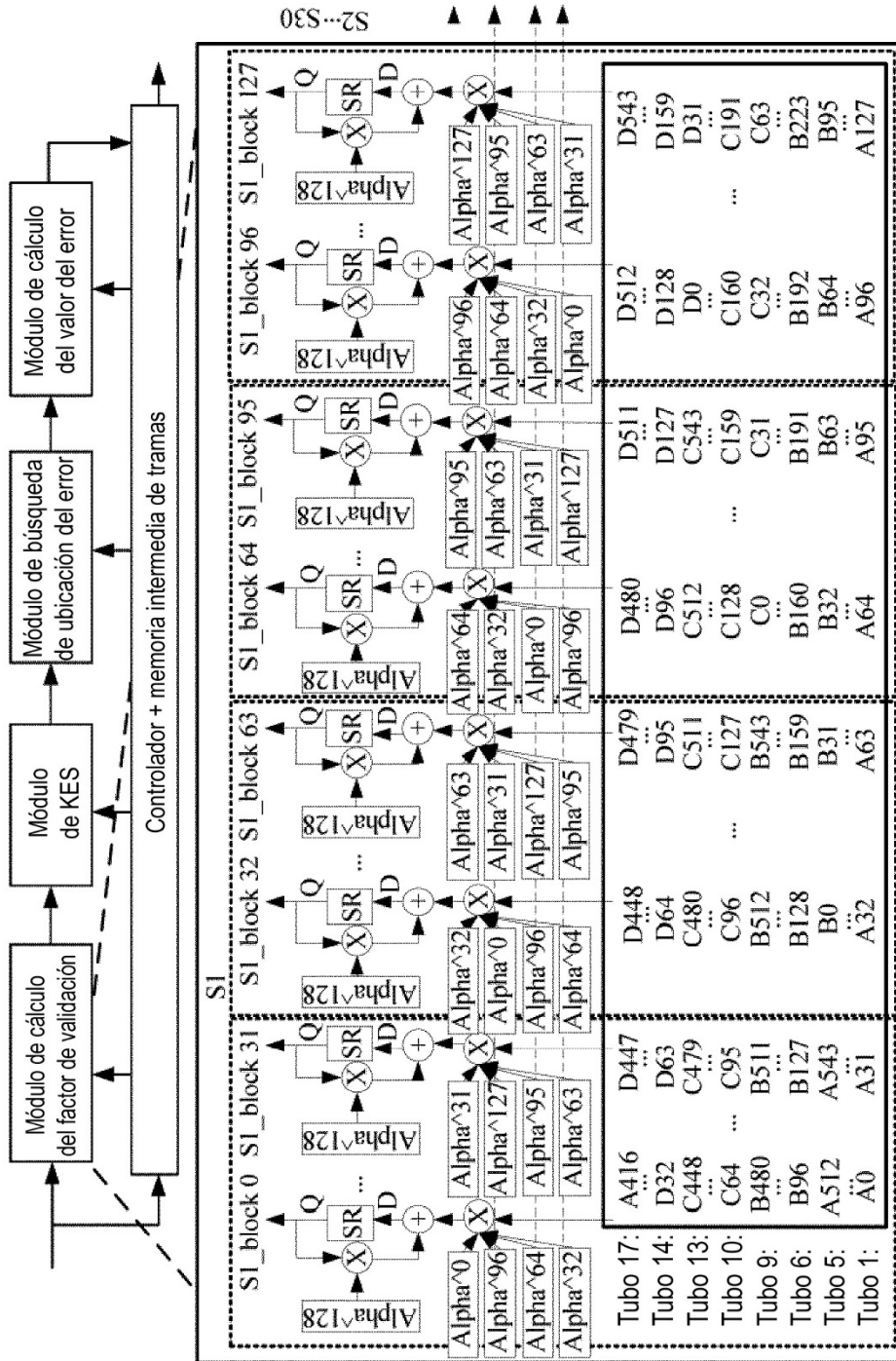


FIG. 4-g

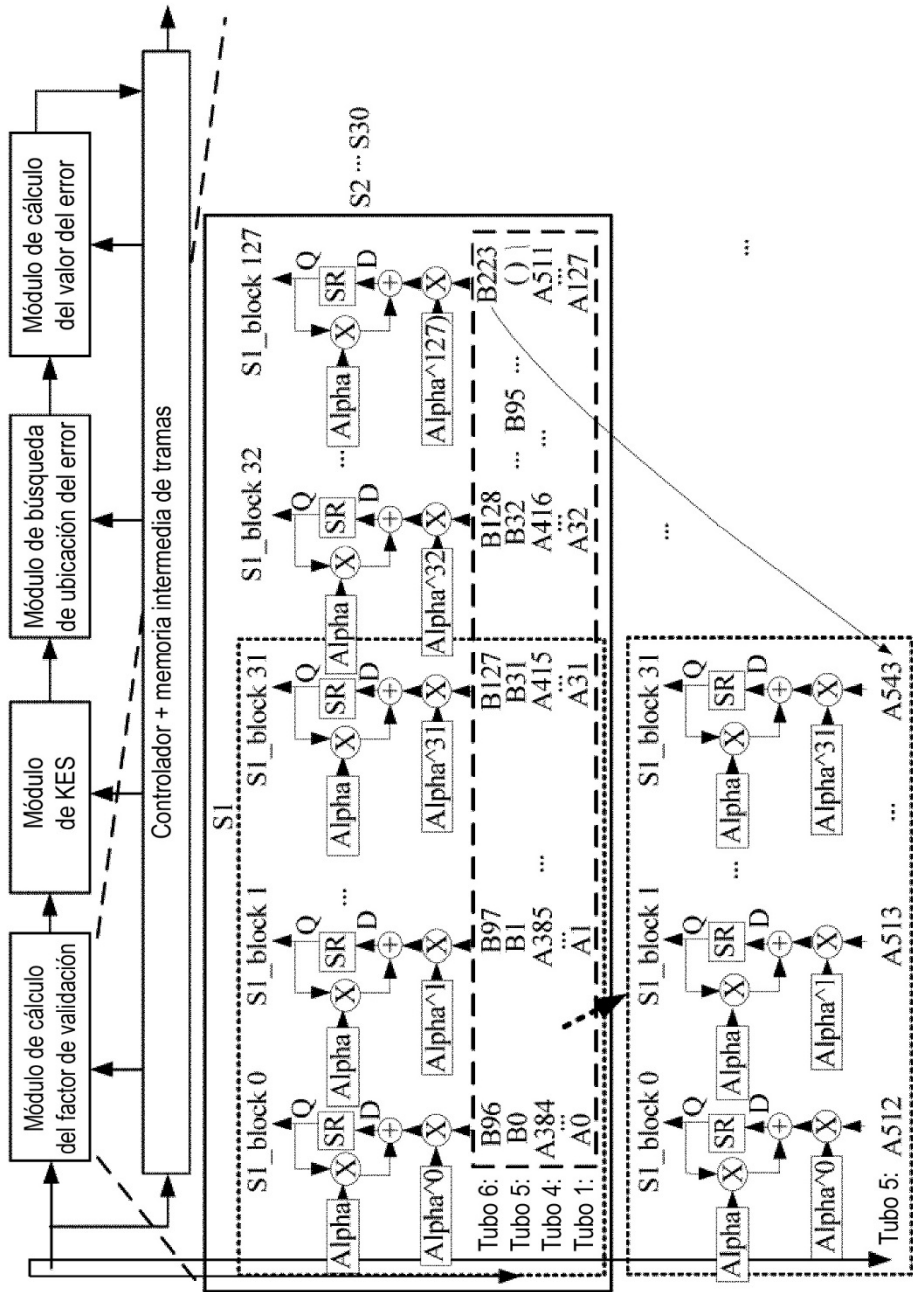


FIG 4-h

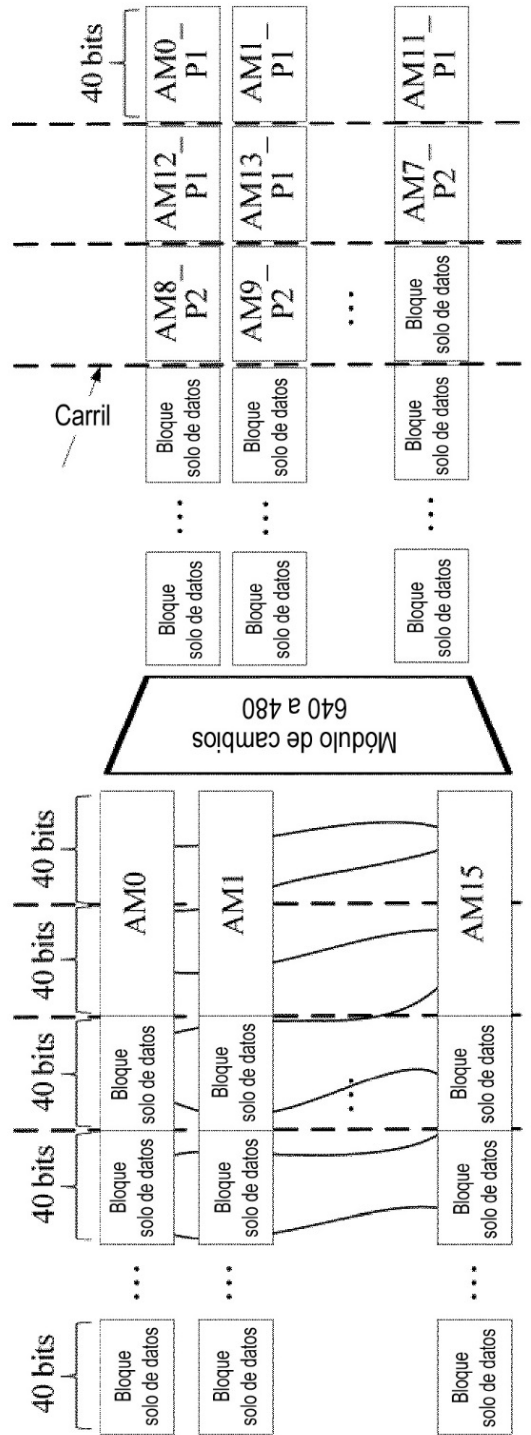


FIG. 5

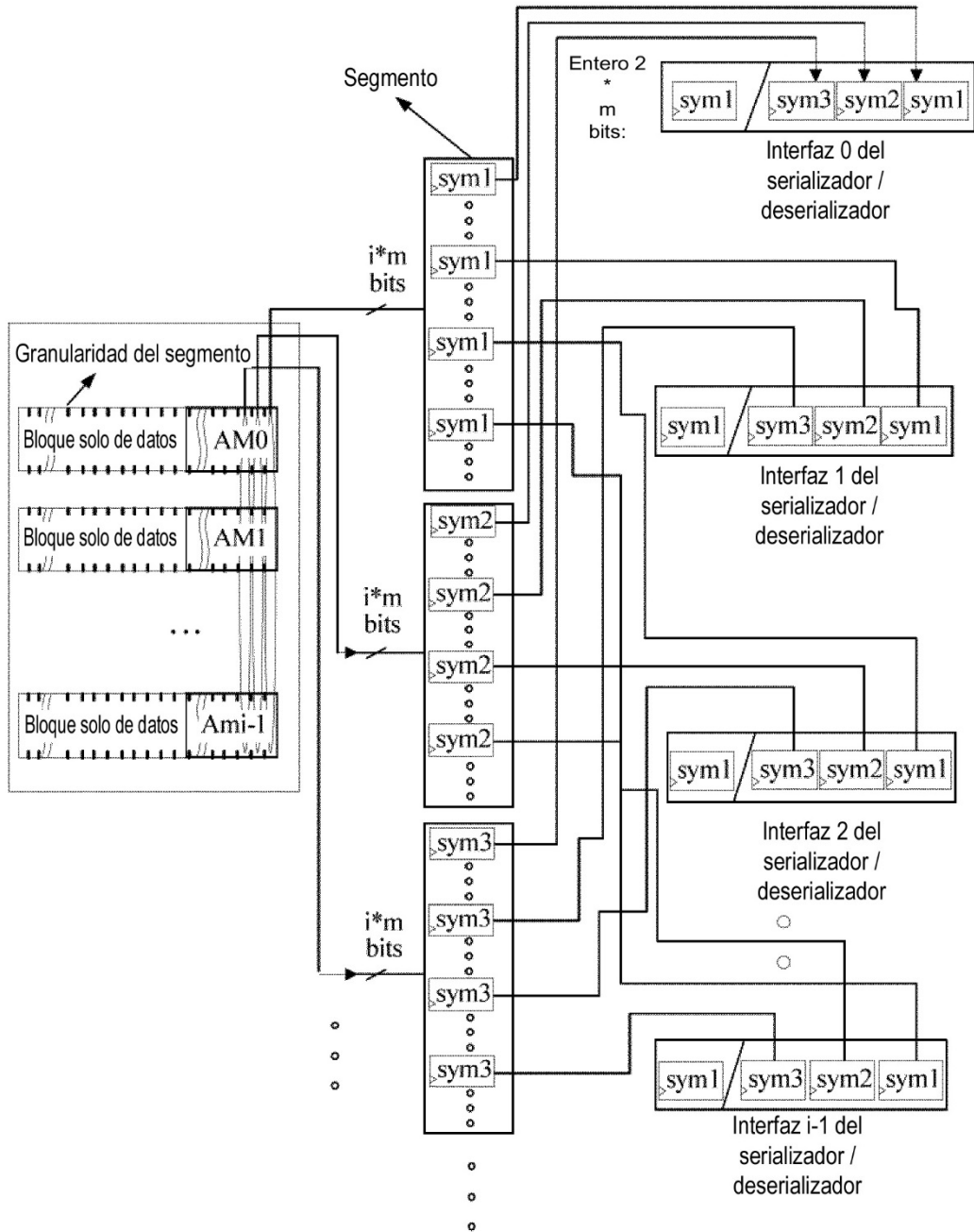


FIG. 6

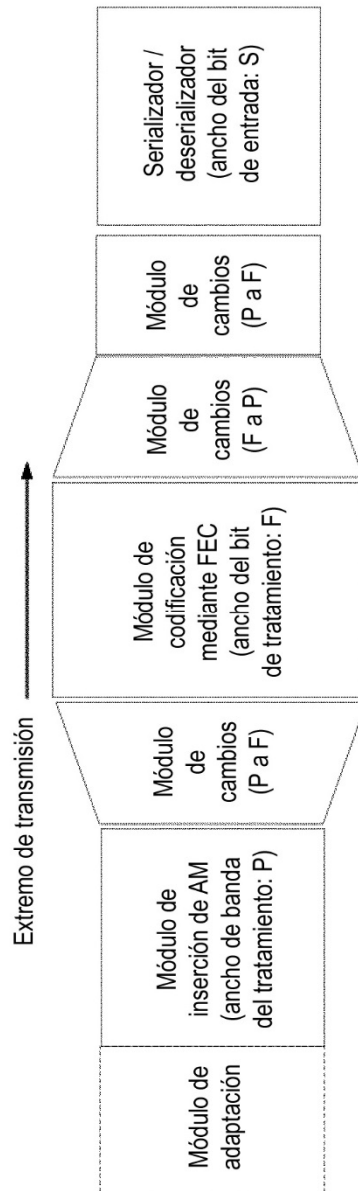


FIG. 7

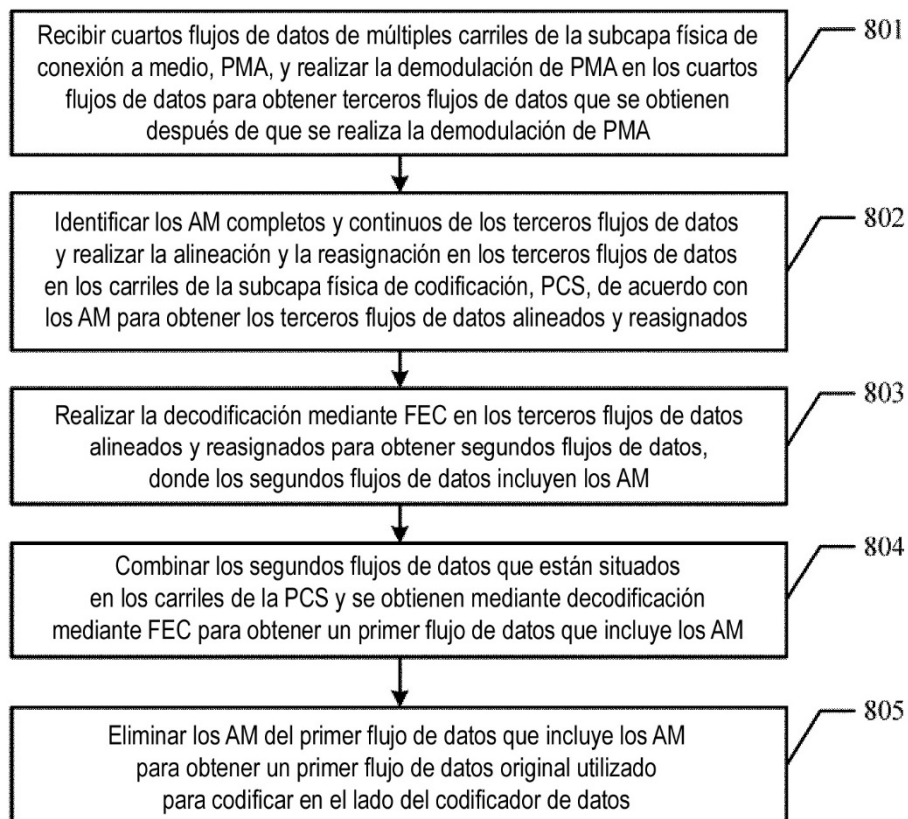


FIG. 8

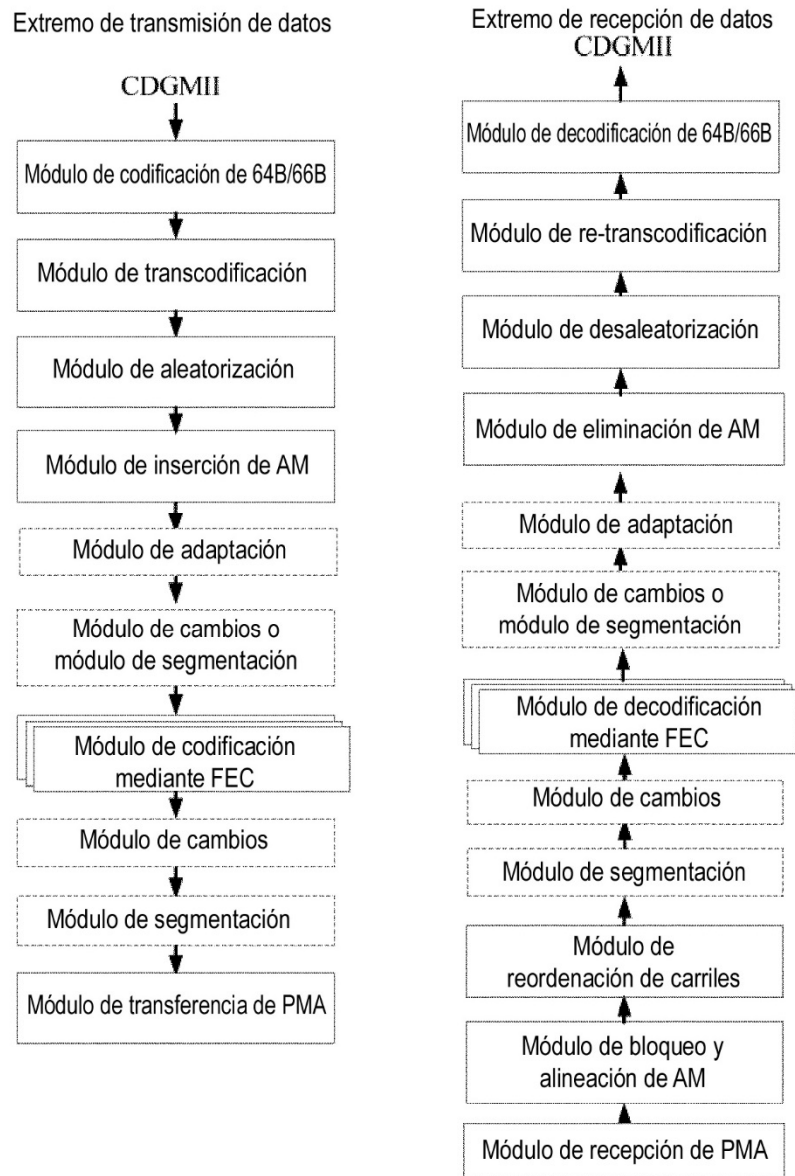


FIG. 9

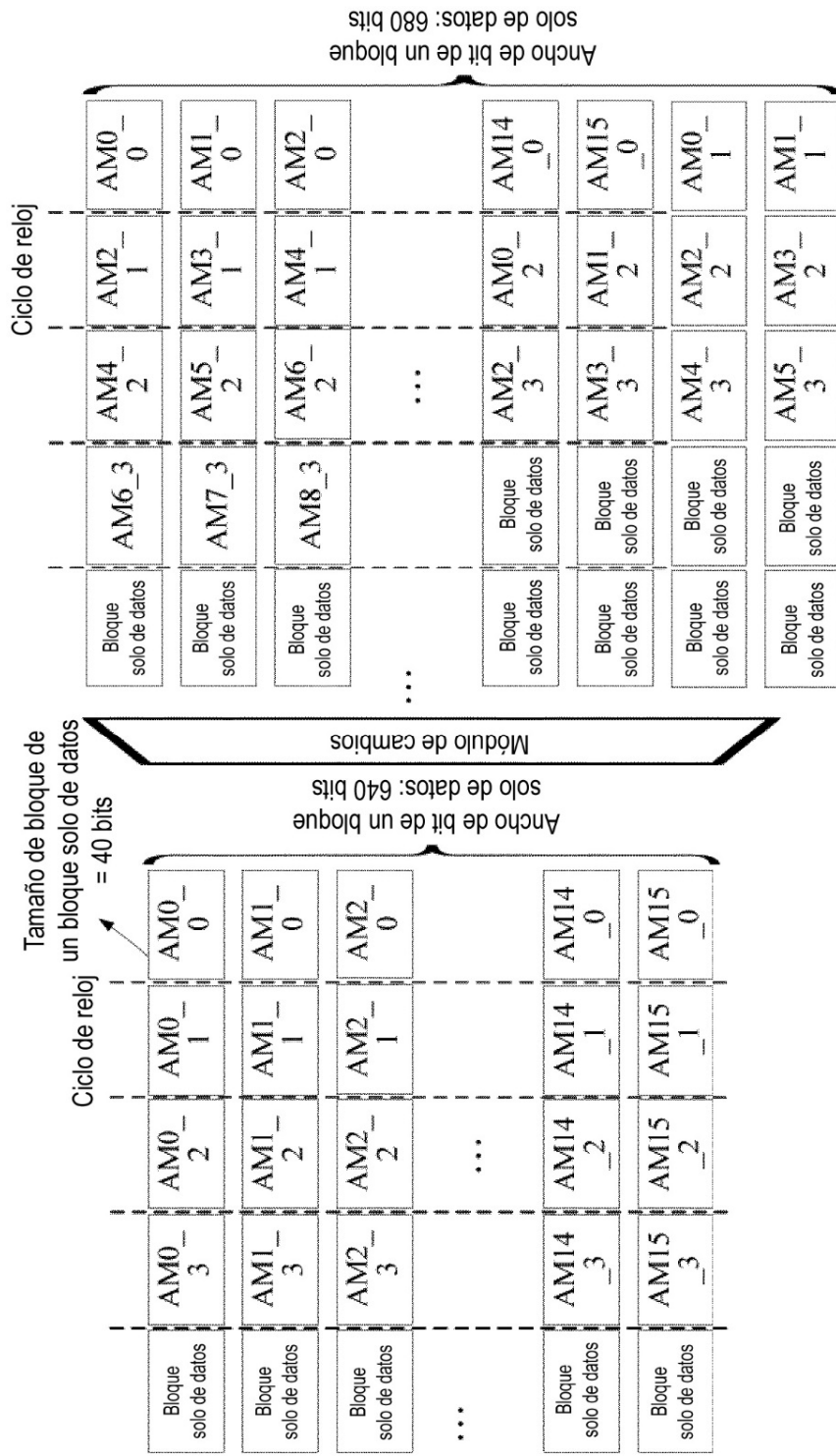


FIG. 10

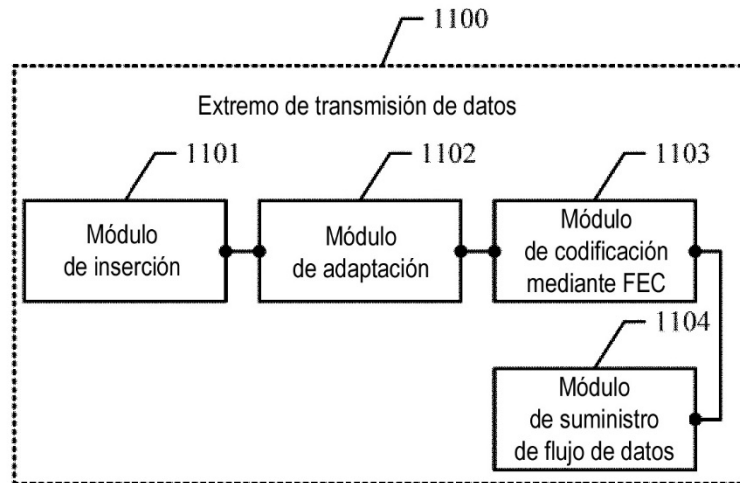


FIG. 11

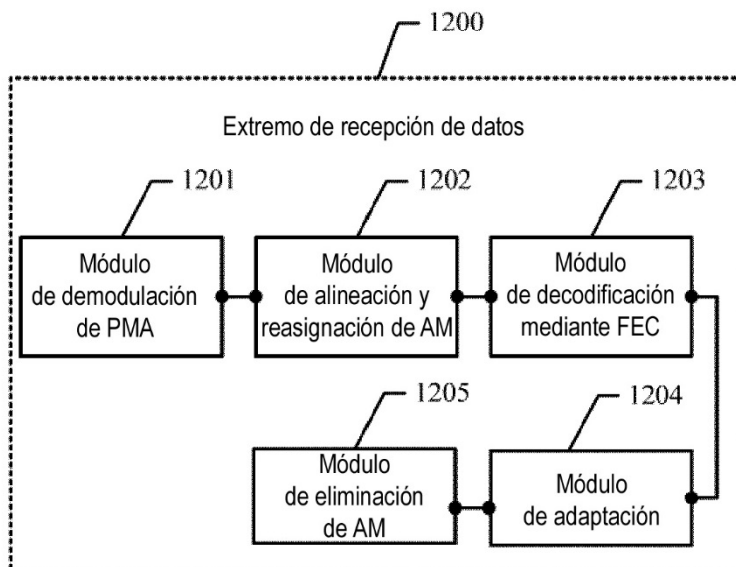


FIG. 12