

MÉTODO Y APARATO DE CODIFICACIÓN DEL CÓDIGO POLAR

CAMPO DE LA INVENCION

Modalidades de esta solicitud se refieren al campo de tecnologías de comunicación y en particular, a un método de codificación de códigos polares y un aparato.

ANTECEDENTES DE LA INVENCION

En un sistema de comunicación, para reducir la complejidad de la codificación, se propone un esquema de codificación basado en un código polar de comprobación de paridad (parity check polar codes, código PC-Polar).

El código PC-Polar puede incluir una posición de bit de información, una posición de bit congelado, una posición de bit de PC y una posición de bit acortado de emparejamiento de tasas. La posición del bit de información se utiliza para transportar un bit de información, la posición del bit congelado se utiliza para transportar un bit congelado y la posición del bit de PC se utiliza para transportar un bit de PC. La posición del bit de PC se determina basándose en la posición del bit de información, un valor del bit de PC se puede determinar utilizando una ecuación de PC basada en un valor de un bit de información situado antes de la posición del bit de PC, y no es necesario enviar a un canal la posición del bit acortado de emparejamiento de tasas.

Para secuencias de diferentes cantidades de bits de información, las posiciones de los bits de información de las secuencias son diferentes, lo que da lugar a diferentes posiciones de los bits de PC y ecuaciones de PC. En otras palabras, es necesario almacenar por separado las posiciones de los bits de PC y las ecuaciones de PC correspondientes para las secuencias de diferentes cantidades de bits de información, lo que se traduce en una mayor complejidad del hardware y una mayor sobrecarga del mismo.

BREVE DESCRIPCIÓN DE LA INVENCION

Esta solicitud proporciona un método de codificación de códigos polares y un aparato, para permitir que se construyan códigos polares basados en una misma secuencia de fiabilidad y una misma primera matriz para secuencias de diferentes cantidades de bits de información, reduciendo así la complejidad del hardware, reduciendo las sobrecargas del hardware, mejorando las características del espectro del código, reduciendo la complejidad de la decodificación y mejorando el rendimiento de la decodificación.

De acuerdo con un primer aspecto, una modalidad de esta solicitud proporciona un método de codificación de códigos polares. Un cuerpo de ejecución del método puede ser un dispositivo de extremo de transmisión, o puede ser un chip, un sistema de chip o un sistema en chip de un dispositivo de extremo de transmisión. El método incluye: El dispositivo de extremo

de transmisión mapea, basándose en una secuencia de fiabilidad, una secuencia de bits de información cuya longitud es k a k bits de una primera secuencia cuya longitud es N , para obtener una segunda secuencia; multiplica la segunda secuencia por una primera matriz para obtener una tercera secuencia; y realiza una codificación polar en la tercera secuencia para

5 obtener una cuarta secuencia, en la que una longitud de la secuencia de fiabilidad es N , la segunda secuencia incluye k posiciones de bits de información, x posiciones de bits de comprobación de paridad PC y $N-k-x$ posiciones de bits congelados, $k \leq K \leq N$, $1 \leq x$, K es un valor máximo de la longitud de la secuencia de bits de información, k , K , N y x son todos enteros positivos, la primera matriz es una matriz con todos los 0 por debajo de una diagonal y todos

10 los 1 en la diagonal, una ponderación de columna de una columna correspondiente a la posición del bit de PC en la primera matriz es mayor que 1, una ponderación de columna de una columna correspondiente a la posición del bit congelado o a la posición del bit de información es igual a 1, y una cantidad de columnas cuyas ponderaciones de columna son mayores que 1 en la primera matriz es menor o igual que $N-K$.

15 De acuerdo con el primer aspecto, esta modalidad de la presente solicitud proporciona un esquema de codificación basado en un código polar PC anidado, para construir por separado, para una secuencia de bits de información, una secuencia de fiabilidad cuya longitud es N y que tiene una característica de anidamiento. De este modo, cuando la punción se realiza en un orden de atrás hacia delante, se reduce la cantidad de puntos malos de rendimiento, se mejoran

20 las características del espectro del código y se mejora el rendimiento de la decodificación. Además, en esta modalidad de la presente solicitud, los códigos polares pueden construirse basándose en una misma primera matriz para secuencias de diferentes cantidades de bits de información, es decir, la primera matriz tiene una característica de anidamiento para las secuencias de diferentes cantidades de bits de información, y la primera matriz no cambia con

25 un cambio de una cantidad de bits de información de las secuencias; o puede describirse que las posiciones de bits de PC y las ecuaciones de PC correspondientes a las secuencias de diferentes cantidades de bits de información están anidadas, y el dispositivo de extremo de transmisión puede almacenar la misma primera matriz para las secuencias de bits de información de diferentes longitudes, y no necesita almacenar por separado las posiciones de

30 bits de PC y las ecuaciones de PC correspondientes para las secuencias de bits de información de diferentes longitudes, reduciendo así la complejidad del hardware, reduciendo las sobrecargas del hardware, mejorando las características del espectro de código, reduciendo la complejidad de la decodificación y mejorando el rendimiento de la decodificación.

35 En un posible diseño, multiplicar la segunda secuencia por la primera matriz para obtener la tercera secuencia incluye: determinar, basándose en la primera matriz, una o más posiciones de bits de PC y bits comprobados por las posiciones de bits de PC; y determinar un

valor de la posición del bit de PC en la segunda secuencia basándose en un valor del bit comprobado por la posición del bit de PC, para obtener la tercera secuencia.

Basándose en este posible diseño, cuando la segunda secuencia se procesa basándose en la primera matriz, la una o más posiciones de bits de PC de la segunda secuencia pueden determinarse basándose en la columna cuya ponderación de columna es mayor que 1 en la primera matriz, y entonces se determina un valor de cada posición de bit de PC en la segunda secuencia basándose en cada bit comprobado por la posición de bit de PC. Por ejemplo, se puede utilizar como valor de la posición de bit de PC un resultado OR exclusivo del valor del bit comprobado por la posición de bit de PC. Esto proporciona una solución viable para procesar la segunda secuencia basándose en la primera matriz para obtener la tercera secuencia.

En un diseño posible, la secuencia de fiabilidad se determina en función de uno o más de los siguientes: la fiabilidad, un espectro de código y un resultado de optimización del rendimiento.

Basándose en este posible diseño, la secuencia de fiabilidad cuya longitud es N puede construirse por separado para la secuencia de bits de información basándose en parámetros como la fiabilidad, el espectro de código y el resultado de optimización del rendimiento. Cuando una secuencia codificada se punciona posteriormente en un orden de atrás hacia delante, se puede reducir una cantidad de puntos malos de rendimiento, mejorar las características del espectro de código y mejorar el rendimiento de la decodificación.

En un diseño posible, una ponderación de columna de la primera matriz es menor o igual que un primer valor.

Basándose en este posible diseño, la ponderación de columna de la primera matriz puede ser menor o igual que el primer valor, garantizando así la escasez de la primera matriz y obteniendo mejores características del espectro de códigos.

En un diseño posible, la primera matriz es:

30

35

4

5

10

15

20

[illegible]

En un diseño posible, la primera matriz es:

25

30

35

5

5

10

15

20

[illegible]

Basándose en los dos diseños posibles anteriores, se proporciona una pluralidad de soluciones viables para diseñar la primera matriz.

25

En un diseño posible, la tercera secuencia incluye la una o más posiciones de bits de PC, y una diferencia entre un número de la posición del bit de PC y un número del bit comprobado por la posición del bit de PC es un número primo.

Basándose en este posible diseño, la diferencia entre el número de la posición del bit de PC y el número del bit comprobado por la posición del bit de PC se limita a ser un número primo.

30

Esto puede reducir en gran medida el espacio de búsqueda y reducir al máximo la pérdida de optimalidad, al tiempo que se mejoran las características del espectro de código.

En un diseño posible, la posición del bit de PC de la tercera secuencia es uno o más de los siguientes bits de la tercera secuencia: 16, 18, 21 o 25.

35

En un diseño posible, cuando la posición del bit de PC es el 16^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es el 13^{ésimo} bit de la tercera secuencia; o cuando la posición del bit de PC es el 18^{ésimo} bit de la tercera secuencia, el bit comprobado

por la posición del bit de PC es uno o más de los siguientes bits de la tercera secuencia: 11, 13 y 15; o cuando la posición del bit de PC es el 21^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es un 14^{ésimo} bit de la tercera secuencia; o cuando la posición del bit de PC es el 25^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es un 22^{ésimo} bit de la tercera secuencia.

Basándose en los dos diseños posibles anteriores, se proporciona un ejemplo preferido para la posición del bit de PC y el bit comprobado por la posición del bit de PC. Esto puede reducir en gran medida el espacio de búsqueda, reducir al máximo la pérdida de optimalidad al tiempo que se obtienen mejores características del espectro de código, reducir la complejidad de la decodificación y mejorar el rendimiento de la decodificación.

En un diseño posible, la posición del bit de PC de la tercera secuencia es uno o más de los siguientes bits de la tercera secuencia: 19, 21, 25 o 26.

En un diseño posible, cuando la posición del bit de PC es el 19^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es el 14^{ésimo} bit de la tercera secuencia; o cuando la posición del bit de PC es el 21^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es uno o más de los siguientes bits de la tercera secuencia: 11 y 14; o cuando la posición del bit de PC es el 25^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es uno o más de los siguientes bits de la tercera secuencia: 14 y 22; o cuando la posición del bit de PC es el 26^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es el 13^{ésimo} bit de la tercera secuencia.

Basándose en los dos diseños posibles anteriores, se proporciona otro ejemplo preferido para la posición del bit de PC y el bit comprobado por la posición del bit de PC. Esto puede reducir en gran medida el espacio de búsqueda, reducir al máximo la pérdida de optimalidad al tiempo que se obtienen mejores características del espectro de código, reducir la complejidad de la decodificación y mejorar el rendimiento de la decodificación.

En un posible diseño, el método incluye además: realizar un procesamiento de intercalación en la cuarta secuencia basado en un primer patrón de intercalación, para obtener una quinta secuencia; y realizar un emparejamiento de tasas en la quinta secuencia para obtener una primera secuencia de emparejamiento de tasas.

Basándose en este posible diseño, el primer patrón de intercalación puede ser un patrón de intercalación determinado en función del rendimiento de la secuencia y de una característica de anidamiento, de modo que no sea necesario almacenar por separado un patrón de intercalación para secuencias de diferentes cantidades de bits de información, reduciendo así la complejidad del hardware y las sobrecargas del mismo. Además, como el primer patrón de intercalación se determina en función del rendimiento de la secuencia, se puede garantizar que no haya ningún punto malo en el rendimiento del emparejamiento de tasas (por ejemplo, no hay

ningún punto malo en el rendimiento del emparejamiento de tasas de grano fino de 1 bit), y se mejora el rendimiento de la decodificación.

En un diseño posible, el primer patrón de intercalación indica colocar un valor de un $A_{ésimo}$ bit de la cuarta secuencia en un $B_{ésimo}$ bit de la quinta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (5, 31), (26, 30), (11, 29), (9, 28), (28, 27), (7, 26), (30, 25), (16, 24), (31, 23), (21, 22), (10, 21), (24, 20), (27, 19), (6, 18), (3, 17), (12, 16), (17, 15), (8, 14), (13, 13), (0, 12), (2, 11), (1, 10), (4, 9), (14, 8), (25, 7), (23, 6), (22, 5), y (19, 4).

En un diseño posible, el primer patrón de intercalación indica colocar un valor de un $A_{ésimo}$ bit de la cuarta secuencia en un $B_{ésimo}$ bit de la quinta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (4, 31), (22, 30), (27, 29), (17, 28), (8, 27), (2, 26), (15, 25), (21, 24), (29, 23), (23, 22), (6, 21), (26, 20), (20, 19), (7, 18), (0, 17), (13, 16), (19, 15), (11, 14), (14, 13), (16, 12), (25, 11), (3, 10), (10, 9), (5, 8), (12, 7), (1, 6), (18, 5), y (24, 4).

Basándose en los dos diseños posibles anteriores, se proporcionan dos ejemplos preferidos para el primer patrón de intercalación. Esto puede garantizar que no haya ningún punto malo en el rendimiento del emparejamiento de tasas mientras se tiene una característica de anidamiento, y mejorar el rendimiento de la decodificación.

En un posible diseño, el método incluye además: realizar un procesamiento de intercalación en la cuarta secuencia basado en un segundo patrón de intercalación, para obtener una sexta secuencia; y realizar un emparejamiento de tasas en la sexta secuencia para obtener una segunda secuencia de emparejamiento de tasas.

Basándose en este posible diseño, a diferencia del primer patrón de intercalación que tiene una característica de anidamiento en el primer posible diseño, los correspondientes segundos patrones de intercalación pueden diseñarse por separado para secuencias de diferentes cantidades de bits de información, es decir, las secuencias de diferentes cantidades de bits de información pueden corresponder a diferentes segundos patrones de intercalación, para ajustarse mejor a las secuencias de diferentes cantidades de bits de información, garantizar que no haya ningún punto malo en el rendimiento del emparejamiento de tasas y mejorar el rendimiento de la decodificación.

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A_{ésimo}$ bit de la cuarta secuencia en un $B_{ésimo}$ bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (12, 31), (9, 30), (19, 29), (18, 28), (1, 27), (15, 26), (14, 25), (29, 24), (16, 23), (10, 22), (28, 21), (23, 20), (20, 19), (6, 18), (21, 17), (7, 16), (2, 15), (25, 14), (0, 13), (3, 12), (5, 11), (31, 10), (30, 9), (4, 8), (22, 7), (13, 6), (27, 5), y (8, 4).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A_{ésimo}$ bit de la cuarta secuencia en un $B_{ésimo}$ bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (18, 31), (29, 30), (9, 29), (12, 28), (27, 27), (22, 26), (23,

25), (26, 24), (4, 23), (0, 22), (7, 21), (14, 20), (8, 19), (19, 18), (1, 17), (21, 16), (13, 15), (11, 14), (17, 13), (20, 12), (16, 11), (15, 10), (6, 9), (10, 8), (28, 7), (25, 6), y (24, 5).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A_{ésimo}$ bit de la cuarta secuencia en un $B_{ésimo}$ bit de la sexta secuencia mediante intercambio; y
5 una relación de mapeo entre A y B es: (2, 31), (20, 30), (12, 29), (25, 28), (8, 27), (22, 26), (16, 25), (5, 24), (29, 23), (14, 22), (23, 21), (15, 20), (3, 19), (11, 18), (10, 17), (17, 16), (0, 15), (30, 14), (18, 13), (27, 12), (4, 11), (19, 10), (21, 9), (6, 8), (26, 7), y (9, 6).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A_{ésimo}$ bit de la cuarta secuencia en un $B_{ésimo}$ bit de la sexta secuencia mediante intercambio; y
10 una relación de mapeo entre A y B es: (19, 31), (23, 30), (16, 29), (18, 28), (29, 27), (0, 26), (21, 25), (8, 24), (11, 23), (4, 22), (13, 21), (2, 20), (20, 19), (24, 18), (25, 17), (6, 16), (28, 15), (14, 14), (10, 13), (9, 12), (31, 11), (26, 10), (7, 9), (17, 8), y (1, 7).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A_{ésimo}$ bit de la cuarta secuencia en un $B_{ésimo}$ bit de la sexta secuencia mediante intercambio; y
15 una relación de mapeo entre A y B es: (7, 31), (9, 30), (19, 29), (26, 28), (31, 27), (21, 26), (27, 25), (2, 24), (23, 23), (17, 22), (10, 21), (30, 20), (25, 19), (24, 18), (0, 17), (16, 16), (4, 15), (11, 14), (20, 13), (12, 12), (29, 11), (18, 10), (13, 9), y (3, 8).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A_{ésimo}$ bit de la cuarta secuencia en un $B_{ésimo}$ bit de la sexta secuencia mediante intercambio; y
20 una relación de mapeo entre A y B es: (15, 31), (16, 30), (9, 29), (14, 28), (17, 27), (24, 26), (21, 25), (20, 24), (5, 23), (1, 22), (7, 21), (30, 20), (11, 19), (6, 18), (2, 17), (3, 16), (8, 15), (22, 14), (12, 13), (23, 12), (19, 11), (10, 10), y (29, 9).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A_{ésimo}$ bit de la cuarta secuencia en un $B_{ésimo}$ bit de la sexta secuencia mediante intercambio; y
25 una relación de mapeo entre A y B es: (17, 31), (21, 30), (15, 29), (14, 28), (4, 27), (24, 26), (8, 25), (29, 24), (19, 23), (16, 22), (7, 21), (2, 20), (9, 19), (25, 18), (26, 17), (23, 16), (28, 15), (30, 14), (13, 13), (6, 12), (22, 11), y (0, 10).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A_{ésimo}$ bit de la cuarta secuencia en un $B_{ésimo}$ bit de la sexta secuencia mediante intercambio; y
30 una relación de mapeo entre A y B es: (28, 31), (29, 30), (14, 29), (26, 28), (8, 27), (23, 26), (24, 25), (30, 24), (0, 23), (31, 22), (10, 21), (5, 20), (20, 19), (4, 18), (16, 17), (12, 16), (9, 15), (21, 14), (1, 13), (11, 12), y (22, 11).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A_{ésimo}$ bit de la cuarta secuencia en un $B_{ésimo}$ bit de la sexta secuencia mediante intercambio; y
35 una relación de mapeo entre A y B es: (22, 31), (19, 30), (28, 29), (9, 28), (4, 27), (6, 26), (14, 25), (11, 24), (3, 23), (13, 22), (0, 21), (15, 20), (26, 19), (24, 18), (27, 17), (18, 16), (21, 15), (17,

14), (20, 13), y (12, 12).

Basándose en los nueve diseños posibles anteriores, se proporciona por separado un posible segundo patrón de intercalación cuando k es igual a 3 a 11, es decir, se diseñan por separado los correspondientes segundos patrones de intercalación para secuencias de diferentes cantidades de bits de información, para garantizar que no haya ningún punto malo en el rendimiento del emparejamiento de tasas y mejorar el rendimiento de la decodificación.

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (5, 31), (11, 30), (24, 29), (30, 28), (13, 27), (20, 26), (22, 25), (3, 24), (4, 23), (25, 22), (31, 21), (6, 20), (10, 19), (8, 18), (21, 17), (15, 16), (9, 15), (2, 14), (0, 13), (27, 12), (14, 11), (16, 10), (17, 9), (23, 8), (28, 7), (26, 6), (1, 5), y (7, 4).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (16, 31), (28, 30), (10, 29), (30, 28), (19, 27), (7, 26), (13, 25), (9, 24), (11, 23), (6, 22), (8, 21), (26, 20), (21, 19), (20, 18), (23, 17), (25, 16), (0, 15), (17, 14), (27, 13), (22, 12), (18, 11), (15, 10), (5, 9), (12, 8), (2, 7), (29, 6), y (24, 5).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (9, 31), (19, 30), (11, 29), (12, 28), (21, 27), (0, 26), (29, 25), (2, 24), (4, 23), (31, 22), (10, 21), (1, 20), (22, 19), (7, 18), (24, 17), (30, 16), (20, 15), (3, 14), (28, 13), (13, 12), (14, 11), (8, 10), (6, 9), (5, 8), (17, 7), y (18, 6).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (11, 31), (25, 30), (31, 29), (27, 28), (14, 27), (0, 26), (22, 25), (19, 24), (18, 23), (3, 22), (20, 21), (2, 20), (4, 19), (17, 18), (23, 17), (24, 16), (12, 15), (1, 14), (7, 13), (10, 12), (28, 11), (26, 10), (6, 9), (5, 8), y (8, 7).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (7, 31), (12, 30), (2, 29), (5, 28), (26, 27), (6, 26), (27, 25), (24, 24), (4, 23), (19, 22), (1, 21), (29, 20), (9, 19), (8, 18), (13, 17), (11, 16), (28, 15), (10, 14), (25, 13), (16, 12), (30, 11), (0, 10), (21, 9), y (3, 8).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (7, 31), (30, 30), (17, 29), (16, 28), (19, 27), (6, 26), (29, 25), (18, 24), (10, 23), (3, 22), (24, 21), (9, 20), (21, 19), (28, 18), (31, 17), (12, 16), (23, 15), (2, 14), (15, 13), (4, 12), (27, 11), (20, 10), y (25, 9).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (28, 31), (25, 30), (17, 29), (6, 28), (11, 27), (21, 26), (27, 25), (30, 24), (3, 23), (1, 22), (4, 21), (8, 20), (15, 19), (26, 18), (29, 17), (18, 16), (9, 15), (19, 14), (2, 13), (20, 12), (23, 11), y (10, 10).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (23, 31), (30, 30), (3, 29), (28, 28), (14, 27), (19, 26), (9, 25), (16, 24), (29, 23), (13, 22), (26, 21), (12, 20), (18, 19), (0, 18), (15, 17), (4, 16), (20, 15), (1, 14), (25, 13), (31, 12), y (21, 11).

En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (1, 31), (24, 30), (14, 29), (15, 28), (25, 27), (12, 26), (26, 25), (13, 24), (27, 23), (31, 22), (8, 21), (5, 20), (20, 19), (21, 18), (3, 17), (28, 16), (17, 15), (6, 14), (19, 13), y (11, 12).

Basándose en los nueve diseños posibles anteriores, se proporciona por separado otro posible segundo patrón de intercalación cuando k es igual a 3 a 11, es decir, se diseñan por separado los correspondientes segundos patrones de intercalación para secuencias de diferentes cantidades de bits de información, para garantizar que no haya ningún punto malo en el rendimiento del emparejamiento de tasas y mejorar el rendimiento de la decodificación.

De acuerdo con un segundo aspecto, una modalidad de esta solicitud proporciona un aparato de comunicación. El aparato de comunicación puede utilizarse en el dispositivo de extremo de transmisión del primer aspecto, para implementar una función realizada por el dispositivo de extremo de transmisión. El aparato de comunicación puede ser un dispositivo de extremo de transmisión, o puede ser un chip, un sistema de chip, un sistema en chip, o similares del dispositivo de extremo de transmisión. El aparato de comunicación puede realizar, por hardware o por hardware mediante la ejecución del software correspondiente, la función realizada por el dispositivo de extremo de transmisión. El hardware o el software incluyen uno o más módulos correspondientes a la función anterior, por ejemplo, un módulo transceptor y un módulo de procesamiento. El módulo transceptor puede completar de forma independiente las siguientes operaciones de recepción y envío, o puede cooperar con el módulo de procesamiento para completar las siguientes operaciones de recepción y envío. De manera correspondiente, el módulo de procesamiento puede completar de manera independiente la siguiente operación de procesamiento, o puede cooperar con el módulo transceptor para completar la siguiente operación de procesamiento. Esto no está limitado.

El módulo de procesamiento está configurado para mapear, basándose en una

- secuencia de fiabilidad, una secuencia de bits de información cuya longitud es k a k bits de una primera secuencia cuya longitud es N , para obtener una segunda secuencia; y está configurado además para: multiplicar la segunda secuencia por una primera matriz para obtener una tercera secuencia; y realizar una codificación polar en la tercera secuencia para obtener una cuarta
- 5** secuencia, en la que una longitud de la secuencia de fiabilidad es N , la segunda secuencia incluye k posiciones de bits de información, x posiciones de bits de PC de comprobación de paridad y $N-k-x$ posiciones de bits congelados, $k \leq K \leq N$, $1 \leq x$, K es un valor máximo de la longitud de la secuencia de bits de información, k , K , N y x son todos enteros positivos, la primera matriz es una matriz con todos los 0 por debajo de una diagonal y todos los 1 por encima de la diagonal,
- 10** una ponderación de columna de una columna correspondiente a la posición del bit de PC en la primera matriz es mayor que 1, una ponderación de columna de una columna correspondiente a la posición del bit congelado o a la posición del bit de información es igual a 1, y una cantidad de columnas cuyas ponderaciones de columna son mayores que 1 en la primera matriz es menor o igual que $N-K$.
- 15** En un posible diseño, el módulo de procesamiento está específicamente configurado para: determinar, basándose en la primera matriz, una o más posiciones de bits de PC y bits comprobados por las posiciones de bits de PC; y determinar un valor de la posición del bit de PC en la segunda secuencia basándose en un valor del bit comprobado por la posición del bit de PC, para obtener la tercera secuencia.
- 20** En un diseño posible, la secuencia de fiabilidad se determina en función de uno o más de los siguientes: la fiabilidad, un espectro de código y un resultado de optimización del rendimiento.
- En un diseño posible, una ponderación de columna de la primera matriz es menor o igual que un primer valor.
- 25** En un diseño posible, la tercera secuencia incluye la una o más posiciones de bits de PC, y una diferencia entre un número de la posición del bit de PC y un número del bit comprobado por la posición del bit de PC es un número primo.
- En un diseño posible, la posición del bit de PC de la tercera secuencia es uno o más de los siguientes bits de la tercera secuencia: 16, 18, 21 o 25.
- 30** En un diseño posible, cuando la posición del bit de PC es el 16^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es el 13^{ésimo} bit de la tercera secuencia; o cuando la posición del bit de PC es el 18^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es uno o más de los siguientes bits de la tercera secuencia: 11, 13 y 15; o cuando la posición del bit de PC es el 21^{ésimo} bit de la tercera secuencia, el bit
- 35** comprobado por la posición del bit de PC es un 14^{ésimo} bit de la tercera secuencia; o cuando la posición del bit de PC es el 25^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición

En un diseño posible, la posición del bit de PC de la tercera secuencia es uno o más de los siguientes bits de la tercera secuencia: 19, 21, 25 o 26.

En un diseño posible, cuando la posición del bit de PC es el 19^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es el 14^{ésimo} bit de la tercera secuencia; o cuando la posición del bit de PC es el 21^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es uno o más de los siguientes bits de la tercera secuencia: 11 y 14; o cuando la posición del bit de PC es el 25^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es uno o más de los siguientes bits de la tercera secuencia: 14 y 22; o cuando la posición del bit de PC es el 26^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es el 13^{ésimo} bit de la tercera secuencia.

En un diseño posible, cuando la posición del bit de PC es el 19^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es el 14^{ésimo} bit de la tercera secuencia; o cuando la posición del bit de PC es el 21^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es uno o más de los siguientes bits de la tercera secuencia: 11 y 14; o cuando la posición del bit de PC es el 25^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es uno o más de los siguientes bits de la tercera secuencia: 14 y 22; o cuando la posición del bit de PC es el 26^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es el 13^{ésimo} bit de la tercera secuencia.

En un diseño posible, la primera matriz es:

	10000000000000000000000000000000
	01000000000000000000000000000000
	00100000000000000000000000000000
15	00010000000000000000000000000000
	00001000000000000000000000000000
	00000100000000000000000000000000
	00000010000000000000000000000000
	00000001000000000000000000000000
	00000000100000000000000000000000
	00000000010000000000000000000000
	00000000001000000000000000000000
20	00000000000100000000000000000000
	00000000000010000001000000000000
	00000000000001000000000000000000
	00000000000000100000000000000000
	0000000000000001001010000000000000
	0000000000000000100000010000000000
	0000000000000000010010000000000000
	0000000000000000001000000000000000
	0000000000000000000100000000000000
25	0000000000000000000010000000000000
	0000000000000000000001000000000000
	0000000000000000000000100000000000
	0000000000000000000000010000000000
	0000000000000000000000001000000000
	000000000000000000000000010010000000
	000000000000000000000000001000000000
	000000000000000000000000000100000000
30	000000000000000000000000000010000000
	000000000000000000000000000001000000
	000000000000000000000000000000100000
	000000000000000000000000000000010000
	000000000000000000000000000000001000
	000000000000000000000000000000000100
	000000000000000000000000000000000010
	000000000000000000000000000000000001

En un diseño posible, la primera matriz es:

10

15

20

25

30

35

En un diseño posible, el módulo de procesamiento está configurado además para: realizar el procesamiento de intercalación en la cuarta secuencia basándose en un segundo patrón de intercalación, para obtener una sexta secuencia; y realizar el emparejamiento de tasas en la sexta secuencia para obtener una segunda secuencia de emparejamiento de tasas.

- 5** En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un A^{ésimo} bit de la cuarta secuencia en un B^{ésimo} bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (12, 31), (9, 30), (19, 29), (18, 28), (1, 27), (15, 26), (14, 25), (29, 24), (16, 23), (10, 22), (28, 21), (23, 20), (20, 19), (6, 18), (21, 17), (7, 16), (2, 15), (25, 14), (0, 13), (3, 12), (5, 11), (31, 10), (30, 9), (4, 8), (22, 7), (13, 6), (27, 5), y (8, 4); o una relación de mapeo entre A y B es: (18, 31), (29, 30), (9, 29), (12, 28), (27, 27), (22, 26), (23, 25), (26, 24), (4, 23), (0, 22), (7, 21), (14, 20), (8, 19), (19, 18), (1, 17), (21, 16), (13, 15), (11, 14), (17, 13), (20, 12), (16, 11), (15, 10), (6, 9), (10, 8), (28, 7), (25, 6), y (24, 5); o una relación de mapeo entre A y B es: (2, 31), (20, 30), (12, 29), (25, 28), (8, 27), (22, 26), (16, 25), (5, 24), (29, 23), (14, 22), (23, 21), (15, 20), (3, 19), (11, 18), (10, 17), (17, 16), (0, 15), (30, 14), (18, 13), (27, 12), (4, 11), (19, 10), (21, 9), (6, 8), (26, 7), y (9, 6); o una relación de mapeo entre A y B es: (19, 31), (23, 30), (16, 29), (18, 28), (29, 27), (0, 26), (21, 25), (8, 24), (11, 23), (4, 22), (13, 21), (2, 20), (20, 19), (24, 18), (25, 17), (6, 16), (28, 15), (14, 14), (10, 13), (9, 12), (31, 11), (26, 10), (7, 9), (17, 8), y (1, 7); o una relación de mapeo entre A y B es: (7, 31), (9, 30), (19, 29), (26, 28), (31, 27), (21, 26), (27, 25), (2, 24), (23, 23), (17, 22), (10, 21), (30, 20), (25, 19), (24, 18), (0, 17), (16, 16), (4, 15), (11, 14), (20, 13), (12, 12), (29, 11), (18, 10), (13, 9), y (3, 8); o una relación de mapeo entre A y B es: (15, 31), (16, 30), (9, 29), (14, 28), (17, 27), (24, 26), (21, 25), (20, 24), (5, 23), (1, 22), (7, 21), (30, 20), (11, 19), (6, 18), (2, 17), (3, 16), (8, 15), (22, 14), (12, 13), (23, 12), (19, 11), (10, 10), y (29, 9); o una relación de mapeo entre A y B es: (17, 31), (21, 30), (15, 29), (14, 28), (4, 27), (24, 26), (8, 25), (29, 24), (19, 23), (16, 22), (7, 21), (2, 20), (9, 19), (25, 18), (26, 17), (23, 16), (28, 15), (30, 14), (13, 13), (6, 12), (22, 11), y (0, 10); o una relación de mapeo entre A y B es: (28, 31), (29, 30), (14, 29), (26, 28), (8, 27), (23, 26), (24, 25), (30, 24), (0, 23), (31, 22), (10, 21), (5, 20), (20, 19), (4, 18), (16, 17), (12, 16), (9, 15), (21, 14), (1, 13), (11, 12), y (22, 11); o una relación de mapeo entre A y B es: (22, 31), (19, 30), (28, 29), (9, 28), (4, 27), (6, 26), (14, 25), (11, 24), (3, 23), (13, 22), (0, 21), (15, 20), (26, 19), (24, 18), (27, 17), (18, 16), (21, 15), (17, 14), (20, 13), y (12, 12).
- 10**
- 15**
- 20**
- 25**
- 30**

- En un diseño posible, el segundo patrón de intercalación indica colocar un valor de un A^{ésimo} bit de la cuarta secuencia en un B^{ésimo} bit de la sexta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (5, 31), (11, 30), (24, 29), (30, 28), (13, 27), (20, 26), (22, 25), (3, 24), (4, 23), (25, 22), (31, 21), (6, 20), (10, 19), (8, 18), (21, 17), (15, 16), (9, 15), (2, 14), (0, 13), (27, 12), (14, 11), (16, 10), (17, 9), (23, 8), (28, 7), (26, 6), (1, 5), y (7, 4); o una relación de mapeo entre A y B es: (16, 31), (28, 30), (10, 29), (30, 28), (19, 27), (7, 26), (13, 25), (9, 24),
- 35**

- (11, 23), (6, 22), (8, 21), (26, 20), (21, 19), (20, 18), (23, 17), (25, 16), (0, 15), (17, 14), (27, 13), (22, 12), (18, 11), (15, 10), (5, 9), (12, 8), (2, 7), (29, 6), y (24, 5); o una relación de mapeo entre A y B es: (9, 31), (19, 30), (11, 29), (12, 28), (21, 27), (0, 26), (29, 25), (2, 24), (4, 23), (31, 22), (10, 21), (1, 20), (22, 19), (7, 18), (24, 17), (30, 16), (20, 15), (3, 14), (28, 13), (13, 12), (14, 11),
- 5** (8, 10), (6, 9), (5, 8), (17, 7), y (18, 6); o una relación de mapeo entre A y B es: (11, 31), (25, 30), (31, 29), (27, 28), (14, 27), (0, 26), (22, 25), (19, 24), (18, 23), (3, 22), (20, 21), (2, 20), (4, 19), (17, 18), (23, 17), (24, 16), (12, 15), (1, 14), (7, 13), (10, 12), (28, 11), (26, 10), (6, 9), (5, 8), y (8, 7); o una relación de mapeo entre A y B es: (7, 31), (12, 30), (2, 29), (5, 28), (26, 27), (6, 26), (27, 25), (24, 24), (4, 23), (19, 22), (1, 21), (29, 20), (9, 19), (8, 18), (13, 17), (11, 16), (28, 15),
- 10** (10, 14), (25, 13), (16, 12), (30, 11), (0, 10), (21, 9), y (3, 8); o una relación de mapeo entre A y B es: (7, 31), (30, 30), (17, 29), (16, 28), (19, 27), (6, 26), (29, 25), (18, 24), (10, 23), (3, 22), (24, 21), (9, 20), (21, 19), (28, 18), (31, 17), (12, 16), (23, 15), (2, 14), (15, 13), (4, 12), (27, 11), (20, 10), y (25, 9); o una relación de mapeo entre A y B es: (28, 31), (25, 30), (17, 29), (6, 28), (11, 27), (21, 26), (27, 25), (30, 24), (3, 23), (1, 22), (4, 21), (8, 20), (15, 19), (26, 18), (29, 17), (18,
- 15** 16), (9, 15), (19, 14), (2, 13), (20, 12), (23, 11), y (10, 10); o una relación de mapeo entre A y B es: (23, 31), (30, 30), (3, 29), (28, 28), (14, 27), (19, 26), (9, 25), (16, 24), (29, 23), (13, 22), (26, 21), (12, 20), (18, 19), (0, 18), (15, 17), (4, 16), (20, 15), (1, 14), (25, 13), (31, 12), y (21, 11); o una relación de mapeo entre A y B es: (1, 31), (24, 30), (14, 29), (15, 28), (25, 27), (12, 26), (26, 25), (13, 24), (27, 23), (31, 22), (8, 21), (5, 20), (20, 19), (21, 18), (3, 17), (28, 16), (17, 15), (6,
- 20** 14), (19, 13), y (11, 12).

Para el efecto técnico aportado por cualquiera del segundo aspecto o los posibles diseños del segundo aspecto, remitirse al efecto técnico aportado por cualquiera del primer aspecto o los posibles diseños del primer aspecto. Los detalles no se describen de nuevo.

- De acuerdo con un tercer aspecto, una modalidad de esta solicitud proporciona un
- 25** aparato de comunicación. El aparato de comunicación incluye uno o más procesadores. El uno o más procesadores están configurados para ejecutar un programa de computadora o instrucciones. Cuando el uno o más procesadores ejecutan las instrucciones de computadora o las instrucciones, el aparato de comunicación está habilitado para realizar el método de codificación del código polar de acuerdo con cualquiera del primer aspecto o los posibles
- 30** diseños del primer aspecto.

- En un diseño posible, el aparato de comunicación incluye además una o más memorias, la una o más memorias están acopladas al uno o más procesadores, y la una o más memorias están configuradas para almacenar el programa de computadora o las instrucciones anteriores. En una implementación posible, la memoria se ubica fuera del aparato de comunicación. En
- 35** otra implementación posible, la memoria se ubica dentro del aparato de comunicación. En esta modalidad de la presente solicitud, el procesador y la memoria se pueden integrar de manera

alternativa en un componente. En otras palabras, el procesador y la memoria se pueden integrar alternativamente juntos. En una posible implementación, el aparato de comunicación incluye además un transceptor. El transceptor está configurado para recibir información y/o enviar información.

- 5** En un diseño posible, el aparato de comunicación incluye además una o más interfaces de comunicación, la una o más interfaces de comunicación están acopladas al uno o más procesadores, y la una o más interfaces de comunicación están configuradas para comunicarse con un módulo distinto del aparato de comunicación.

- 10** De acuerdo con un cuarto aspecto, una modalidad de esta solicitud proporciona un aparato de comunicación. El aparato de comunicación incluye una interfaz de entrada/salida y un circuito lógico. La interfaz de entrada/salida está configurada para introducir y/o emitir información. El circuito lógico está configurado para realizar el método de codificación del código polar de acuerdo con cualquiera del primer aspecto o los posibles diseños del primer aspecto, y realizar un procesamiento basado en la información y/o generar información.

- 15** De acuerdo con un quinto aspecto, una modalidad de esta solicitud proporciona un medio de almacenamiento legible por computadora. El medio de almacenamiento legible por computadora almacena instrucciones de computadora o un programa. Cuando las instrucciones de computadora o el programa se ejecutan en una computadora, se lleva a cabo el método de codificación del código polar de acuerdo con cualquiera del primer aspecto o los posibles diseños del primer aspecto.

- 20** De acuerdo con un sexto aspecto, una modalidad de la presente solicitud proporciona un producto de programa de computadora, que incluye instrucciones de computadora. Cuando el producto de programa de computadora se ejecuta en una computadora, se lleva a cabo el método de codificación del código polar de acuerdo con cualquiera del primer aspecto o los posibles diseños del primer aspecto.

- 25** De acuerdo con un séptimo aspecto, una modalidad de esta solicitud proporciona un programa de computadora. Cuando el programa de computadora se ejecuta en una computadora, se lleva a cabo el método de codificación del código polar de acuerdo con cualquiera del primer aspecto o los posibles diseños del primer aspecto.

- 30** De acuerdo con un octavo aspecto, una modalidad de esta solicitud proporciona un chip, incluyendo un procesador. El procesador está acoplado a una memoria, la memoria está configurada para almacenar un programa o instrucciones. Cuando el programa o las instrucciones son ejecutados por el procesador, el chip está habilitado para realizar el método de codificación del código polar de acuerdo con cualquiera del primer aspecto o los posibles diseños del primer aspecto.

- 35** Para el efecto técnico aportado por cualquiera de las formas de diseño del tercer aspecto

al octavo aspecto, remitirse al efecto técnico aportado por cualquiera del primer aspecto o los posibles diseños del primer aspecto. Los detalles no se describen de nuevo.

- De acuerdo con un noveno aspecto, una modalidad de esta solicitud proporciona un sistema de comunicación. El sistema de comunicación puede incluir el aparato de comunicación de acuerdo con cualquiera del segundo aspecto o los posibles diseños del segundo aspecto.

BREVE DESCRIPCIÓN DE LAS FIGURAS

La FIGURA 1 es un diagrama de un procedimiento de decodificación de un código LTE-RM de acuerdo con una modalidad de esta solicitud;

- La FIGURA 2 es un diagrama de un código PC-Polar de acuerdo con una modalidad de esta solicitud;

La FIGURA 3 es un diagrama de comparación entre la complejidad de decodificación de un código LTE-RM y un código PC-Polar de acuerdo con una modalidad de esta solicitud;

La FIGURA 4 es un diagrama de un sistema de comunicación de acuerdo con una modalidad de esta solicitud;

- La FIGURA 5 es un diagrama de la codificación y decodificación realizadas por un dispositivo de extremo de transmisión y un dispositivo de extremo de recepción de acuerdo con una modalidad de esta solicitud;

La FIGURA 6 es un diagrama de una composición de un aparato de comunicación de acuerdo con una modalidad de esta solicitud;

- La FIGURA 7 es un diagrama de flujo de un método de codificación de códigos polares de acuerdo con una modalidad de esta solicitud;

La FIGURA 8 es un diagrama de una primera matriz de acuerdo con una modalidad de esta solicitud;

- La FIGURA 9 es un diagrama de una primera matriz de acuerdo con una modalidad de esta solicitud;

La FIGURA 10 es un diagrama de un proceso de búsqueda de un primer patrón de intercalación de acuerdo con una modalidad de esta solicitud;

La FIGURA 11 es un diagrama de un primer patrón de intercalación de acuerdo con una modalidad de esta solicitud;

- La FIGURA 12 es un diagrama de un primer patrón de intercalación de acuerdo con una modalidad de esta solicitud;

La FIGURA 13 es un diagrama de un segundo patrón de intercalación de acuerdo con una modalidad de esta solicitud;

- La FIGURA 14 es un diagrama de un segundo patrón de intercalación de acuerdo con una modalidad de esta solicitud;

La FIGURA 15 es un diagrama de una comparación de rendimiento de acuerdo con una

modalidad de esta solicitud;

La FIGURA 16 es un diagrama de una comparación de rendimiento de acuerdo con una modalidad de esta solicitud;

La FIGURA 17 es un diagrama de una comparación de rendimiento de acuerdo con una modalidad de esta solicitud;

La FIGURA 18 es un diagrama de una comparación de rendimiento de acuerdo con una modalidad de esta solicitud;

La FIGURA 19 es un diagrama de una comparación de rendimiento de acuerdo con una modalidad de esta solicitud;

La FIGURA 20 es un diagrama de un aparato de comunicación de acuerdo con una modalidad de la presente solicitud; y

La FIGURA 21 es un diagrama de una composición de un aparato de comunicación de acuerdo con una modalidad de esta solicitud.

DESCRIPCIÓN DETALLADA DE LA INVENCION

Antes de describir las modalidades de esta solicitud, se describen los términos técnicos utilizados en ellas.

Codificación de códigos de evolución a largo plazo Reed-Muller (long term evolution-reed-muller, LTE-RM): Un dispositivo de extremo de transmisión puede codificar una secuencia de bits de información muy corta de 3 a 11 bits de la siguiente manera:

Paso 1: Codificar una secuencia de bits de información $c_0, c_1, \dots$, and c_{K-1} cuya longitud es K, para obtener una secuencia codificada $d_0, d_1, \dots$, and d_{N-1} cuya longitud es N.

Por ejemplo, K puede ser cualquier valor entre 3 y 11, y N puede ser 32.

$d_i = (\sum_{k=0}^{K-1} c_k M_{i,k})$, un valor de $M_{i,k}$ se puede determinar de acuerdo con la tabla 1 siguiente, y $i=0, 1, 2, \dots$, y N-1.

Tabla 1

i	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$	$M_{i,10}$
0	1	1	0	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	0	0	1	1
2	1	0	0	1	0	0	1	0	1	1	1
3	1	0	1	1	0	0	0	0	1	0	1
4	1	1	1	1	0	0	0	1	0	0	1
5	1	1	0	0	1	0	1	1	1	0	1
6	1	0	1	0	1	0	1	0	1	1	1
7	1	0	0	1	1	0	0	1	1	0	1
8	1	1	0	1	1	0	0	1	0	1	1
9	1	0	1	1	1	0	1	0	0	1	1
10	1	0	1	0	0	1	1	1	0	1	1
11	1	1	1	0	0	1	1	0	1	0	1
12	1	0	0	1	0	1	0	1	1	1	1
13	1	1	0	1	0	1	0	1	0	1	1

i	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$	$M_{i,10}$
14	1	0	0	0	1	1	0	1	0	0	1
15	1	1	0	0	1	1	1	1	0	1	1
16	1	1	1	0	1	1	1	0	0	1	0
17	1	0	0	1	1	1	0	0	1	0	0
18	1	1	0	1	1	1	1	1	0	0	0
19	1	0	0	0	0	1	1	0	0	0	0
20	1	0	1	0	0	0	1	0	0	0	1
21	1	1	0	1	0	0	0	0	0	1	1
22	1	0	0	0	1	0	0	1	1	0	1
23	1	1	1	0	1	0	0	0	1	1	1
24	1	1	1	1	1	0	1	1	1	1	0
25	1	1	0	0	0	1	1	1	0	0	1
26	1	0	1	1	0	1	0	0	1	1	0
27	1	1	1	1	0	1	0	1	1	1	0
28	1	0	1	0	1	1	1	0	1	0	0
29	1	0	1	1	1	1	1	1	1	0	0
30	1	1	1	1	1	1	1	1	1	1	1
31	1	0	0	0	0	0	0	0	0	0	0

15 Paso 2: Realizar el emparejamiento de tasas en la secuencia codificada $d_0, d_1, \dots$, and d_{N-1} cuya longitud es N, para obtener una secuencia de emparejamiento de tasas $f_0, f_1, \dots$, and f_{E-1} cuya longitud es E.

20 E es una longitud de código de transmisión real obtenida al realizar el emparejamiento de tasas, o E se describe como una longitud de código de transmisión obtenida al realizar el emparejamiento de tasas. La longitud del código de transmisión E se puede determinar basándose en la información relacionada con el emparejamiento de tasas.

25 Cuando se determina que la longitud del código de transmisión E no es igual a la longitud codificada N (por ejemplo, E no es igual a 32), puede utilizarse la siguiente manera de emparejar la tasa: Cuando la longitud del código de transmisión E es menor que N (por ejemplo, E es inferior a 32), la punción se realiza de atrás hacia delante; o cuando la longitud del código de transmisión E es mayor que N (por ejemplo, E es mayor que 32), la repetición se realiza de delante hacia atrás.

30 Por ejemplo, la secuencia de emparejamiento de tasas $f_0, f_1, \dots$, and f_{E-1} puede obtenerse de la siguiente manera:

30 para $j=0$ a $E-1$

$$f_j = d_{j \bmod N};$$

fin para

Paso 3: Enviar la secuencia de emparejamiento de tasas $f_0, f_1, \dots$, and f_{E-1} .

35 Decodificación LTE-RM: Con referencia a un diagrama de un procedimiento de decodificación mostrado en la FIGURA 1, un dispositivo de extremo de recepción puede

decodificar un resultado de codificación de la secuencia de bits de información muy corta de 3 a 11 bits de la siguiente manera:

- 5 Paso 1: Ejecutar una decisión simple (por ejemplo, una decisión dura) sobre una secuencia recibida, y realizar un procesamiento de intercalación sobre una palabra de código (por ejemplo, una palabra de código bipolar) o una información de bits suave obtenida mediante una decisión simple, para obtener una palabra de código de recepción procesada.

La secuencia recibida puede ser la secuencia de emparejamiento de tasas.

Opcionalmente, si una longitud de la palabra de código obtenida mediante decisión simple no es igual a N, puede realizarse un procesamiento de relleno de ceros de alto orden.

- 10 Por ejemplo, si la palabra de código obtenida mediante decisión simple es $b_0, b_1, \dots$, and b_{19} cuya longitud es 20, pueden rellenarse 12 0 en los bits de alto orden para obtener una palabra de código $0, \dots, 0, b_0, b_1, \dots$, and b_{19} cuya longitud es $N=32$.

Paso 2: Realizar, basándose en un vector de máscaras, un procesamiento de intercalación en la palabra de código de recepción procesada en el paso 1.

- 15 El proceso de procesamiento de intercalación es el mismo que el proceso de procesamiento de intercalación del paso 1.

- 20 Por ejemplo, pueden generarse 128 vectores de máscaras de acuerdo con siete secuencias básicas de máscaras, y los 128 vectores de máscaras se multiplican respectivamente por la palabra de código de recepción procesada en el paso 1 (es decir, se realiza el desenmascaramiento), para obtener 128 secuencias bipolares con una longitud de 32.

Paso 3: Realizar la transformada rápida de Hadamard (fast Hadamard transform, FHT) sobre la secuencia bipolar obtenida en el paso 2 y una matriz de Hadamard (Hadamard) de 32 órdenes, para obtener una matriz de valores de correlación de 128×32 .

- 25 Paso 4: Encontrar un valor absoluto máximo a partir de la matriz de valores de correlación obtenida en el paso 3, de forma que un formato binario correspondiente a un número de fila con el valor absoluto máximo sea de 2^{ésimo} a 6^{ésimo} bits de la secuencia de bits de información, y un formato binario correspondiente a un número de columna con el valor absoluto máximo sea de 7^{ésimo} a 13^{ésimo} bits de la secuencia de bits de información.

- 30 Paso 5: Un 1^{ésimo} bit de la secuencia de bits de información se determina en función de un signo real del valor absoluto máximo. Es decir, cuando el signo es positivo, el 1^{ésimo} bit se decodifica como 0; y cuando el signo es negativo, el 1^{ésimo} bit se decodifica como 1.

- 35 El 1^{ésimo} bit al 12^{ésimo} bit en el paso 4 y el paso 5 definen la secuencia de bits de información a partir del 1^{ésimo} bit. Puede entenderse que la secuencia de bits de información puede definirse alternativamente empezando por el 0^{ésimo} bit, es decir, el 1^{ésimo} bit, el 2^{ésimo} bit, ..., y el 12^{ésimo} bit se sustituyen respectivamente por el 0^{ésimo} bit, el 1^{ésimo} bit, ..., y el 11^{ésimo} bit. Esto

no está limitado.

Sin embargo, la decodificación LTE-RM utiliza el FHT. Cuando la longitud de la secuencia de bits de información es mayor que 6 bits, es necesario enumerar los vectores de máscara y realizar el desenmascaramiento (demask), lo que supone una gran complejidad y un elevado consumo de energía para que el esquema de decodificación LTE-RM alcance un rendimiento de decodificación de máxima verosimilitud (maximum likelihood, ML). Además, cuando la longitud E de la secuencia de emparejamiento de tasas es pequeña, hay una gran cantidad de punciones, lo que provoca puntos malos en el rendimiento y afecta al rendimiento de la decodificación.

- 5** Código polar de comprobación de paridad (parity check polar codes, códigos PC-Polar): El código PC-Polar puede incluir una posición de bit de información, una posición de bit congelado, una posición de bit de PC y una posición de bit acortado de emparejamiento de tasas. La posición del bit de información se puede utilizar para transportar un bit de información, la posición del bit congelado se puede utilizar para transportar un bit congelado, la posición del bit de PC se puede utilizar para transportar un bit de PC, y la posición del bit acortado de emparejamiento de tasas no necesita enviarse a un canal.

- 15** Una parte de los conjuntos puede seleccionarse entre las posiciones de bits congelados como posiciones de bits de PC. Los valores de los bits de PC en estas posiciones de bits de PC son diferentes de los de otras posiciones de bits congelados, y no están fijados a 0, sino que se determinan utilizando una ecuación de PC basada en los valores de los bits de información de las posiciones de bits de información anteriores a las posiciones de bits de PC. Por lo tanto, las posiciones de bits de PC también pueden denominarse posiciones dinámicas de bits congelados (es decir, las posiciones son de las posiciones de bits congelados, pero los valores no están fijados a 0).

- 25** Por ejemplo, la longitud de la secuencia de bits de información es 7. El código PC-Polar correspondiente a la secuencia de bits de información puede ser un código PC-Polar mostrado en la FIGURA 2. Las posiciones de los bits congelados pueden ser 0^{ésima}, 1^{ésima}, 2^{ésima}, 4^{ésima} y 8^{ésima}, las posiciones de los bits de información pueden ser 3^{ésima}, 5^{ésima}, 6^{ésima}, 10^{ésima}, 11^{ésima}, 13^{ésima}, y 14^{ésima} bits, las posiciones de bits de PC pueden ser 9^{ésimo} y 12^{ésimo} bits, y las posiciones de bits acortados de emparejamiento de tasas pueden ser 7^{ésimo} y 15^{ésimo} bits.

- 30** Un valor del bit de PC en el 9^{ésimo} bit puede ser un OR exclusivo de un valor de un bit de información en el 3^{ésimo} bit y un valor de un bit de información en el 6^{ésimo} bit, y un valor del bit de PC en el 12^{ésimo} bit puede ser un OR exclusivo de un valor de un bit de información en el 3^{ésimo} bit y un valor de un bit de información en el 5^{ésimo} bit.

- 35** Basándose en la descripción anterior, en comparación con el esquema de codificación y decodificación LTE-RM, el uso del código PC-Polar para la codificación y decodificación puede

reducir la complejidad de la decodificación, reducir el consumo de energía y mejorar el rendimiento de la decodificación.

Por ejemplo, como se muestra en la FIGURA 3, se puede saber que para diferentes tasas de código, la complejidad de decodificación de un código PC-Polar es menor que la de un código LTE-RM.

Sin embargo, en el código PC-Polar, para secuencias de bits de información de longitudes diferentes (o descritas como cantidades diferentes de bits de información), las posiciones de los bits de información de las secuencias son diferentes, lo que da lugar a posiciones de bits de PC y ecuaciones de PC diferentes (por ejemplo, una posición del bit de PC correspondiente a una secuencia de bits de información cuya longitud es 4 puede no ser una posición del bit de PC en una secuencia de bits de información cuya longitud es 3, y las ecuaciones de PC correspondientes también son diferentes). En consecuencia, las posiciones de bits de PC y las ecuaciones de PC correspondientes deben almacenarse por separado para las secuencias de bits de información de diferentes longitudes, la complejidad del hardware es mayor y las sobrecargas del hardware son más elevadas.

Para resolver este problema técnico, una modalidad de esta solicitud proporciona un método de codificación de códigos polares. En el método, un dispositivo de extremo de transmisión puede mapear, basándose en una secuencia de fiabilidad, una secuencia de bits de información cuya longitud es k a k bits de una primera secuencia cuya longitud es N , para obtener una segunda secuencia; multiplicar la segunda secuencia por una primera matriz para obtener una tercera secuencia; y realizar una codificación polar en la tercera secuencia para obtener una cuarta secuencia, en la que una longitud de la secuencia de fiabilidad es N , la segunda secuencia incluye k posiciones de bits de información, x posiciones de bits de comprobación de paridad PC y $N-k-x$ posiciones de bits congelados, $k \leq K \leq N$, $1 \leq x$, K es un valor máximo de la longitud de la secuencia de bits de información, k , K , N y x son todos enteros positivos, la primera matriz es una matriz con todos los 0 por debajo de una diagonal y todos los 1 por encima de la diagonal, una ponderación de columna de una columna correspondiente a la posición del bit de PC en la primera matriz es mayor que 1, una ponderación de columna de una columna correspondiente a la posición del bit congelado o a la posición del bit de información es igual a 1, y una cantidad de columnas cuyas ponderaciones de columna son mayores que 1 en la primera matriz es menor o igual que $N-K$.

En esta modalidad de esta solicitud, se puede construir por separado una secuencia de fiabilidad cuya longitud sea N para la secuencia de bits de información. De este modo, cuando la punción se realiza en un orden de atrás hacia delante, se reduce la cantidad de puntos malos de rendimiento, se mejoran las características del espectro del código y se mejora el rendimiento de la decodificación. Además, en esta modalidad de la presente solicitud, los códigos polares

pueden construirse basándose en una misma primera matriz para secuencias de diferentes cantidades de bits de información, es decir, las posiciones de bits de PC y las ecuaciones de PC correspondientes a las secuencias de diferentes cantidades de bits de información están anidadas, y no es necesario almacenar por separado las posiciones de bits de PC y las ecuaciones de PC correspondientes para las secuencias de bits de información de diferentes longitudes, reduciendo así la complejidad del hardware, reduciendo las sobrecargas del hardware, mejorando las características del espectro del código, reduciendo la complejidad de la decodificación y mejorando el rendimiento de la decodificación.

A continuación se describen en detalle las modalidades de esta solicitud haciendo referencia a los dibujos que la acompañan en esta especificación.

El método de codificación de códigos polares proporcionado en las modalidades de la presente solicitud puede aplicarse a cualquier sistema de comunicación. El sistema de comunicación puede ser un sistema de comunicación del proyecto de asociación de tercera generación (third generation partnership project, 3GPP), por ejemplo, un sistema de evolución a largo plazo (long term evolution, LTE), o puede ser un sistema de comunicación móvil de quinta generación (fifth generation, 5G), un sistema con redes híbridas de LTE y 5G, un sistema de nueva radio (new radio, NR), un sistema NR de vehículo a todo (vehicle-to-everything, V2X), un sistema de comunicación dispositivo a dispositivo (dispositivo a dispositivo, D2D), un sistema de comunicación máquina a máquina (máquina a máquina, M2M), internet de las cosas (internet of things, IoT), un sistema de internet de las cosas de banda estrecha (narrowband-internet of things, NB-IoT), un sistema global de comunicaciones móviles (global system for mobile communications, GSM), un sistema de velocidad de datos mejorada para la evolución del GSM (enhanced data rate for GSM evolution, EDGE), un sistema de acceso múltiple por división de código de banda ancha (wideband code division multiple access, WCDMA), un sistema de acceso múltiple por división de código 2000 (code division multiple access, CDMA2000), un sistema de acceso múltiple por división de código síncrono por división de tiempo (time division-synchronous code division multiple access, TD-SCDMA), banda ancha móvil mejorada (enhanced mobile broadband, eMBB), comunicación ultrafiel y de baja latencia (ultra-reliable and low-latency communication, URLLC), comunicación mejorada de tipo máquina (enhanced machine-type communication, eMTC), y varios tipos de sistemas de comunicación de próxima generación, como un sistema de comunicación móvil de sexta generación (sixth generation, 6G), o puede ser un sistema de red no terrestre (non-terrestrial network, NTN), un sistema de comunicación no-3GPP, o similares. Esto no está limitado.

El método de codificación del código polar proporcionado en las modalidades de esta solicitud puede aplicarse a diversos escenarios de comunicación, y es particularmente aplicable a un escenario de codificación de canal en el que se utiliza un código polar como código corto

en un sistema de comunicación. Por ejemplo, el método de codificación del código polar puede aplicarse a uno o más de los siguientes escenarios de comunicación: codificación del canal de control, codificación del canal de datos y similares. Esto no está limitado.

Lo siguiente utiliza la FIGURA 4 como ejemplo para describir el sistema de comunicación proporcionado en las modalidades de esta solicitud.

La FIGURA 4 es un diagrama de un sistema de comunicación de acuerdo con una modalidad de esta solicitud. Como se muestra en la FIGURA 4, el sistema de comunicación puede incluir al menos un dispositivo terminal y al menos un dispositivo de red.

En la FIGURA 4, el dispositivo terminal puede estar situado en la cobertura de haz/célula del dispositivo de red, y el dispositivo de red puede proporcionar un servicio de comunicación para el dispositivo terminal. Por ejemplo, el dispositivo de red puede codificar los datos de enlace descendente mediante la codificación de canal y transmitir los datos de enlace descendente al dispositivo terminal a través de una interfaz aérea tras la modulación de constelación (es decir, el dispositivo de red es un dispositivo de extremo de transmisión y el dispositivo terminal es un dispositivo de extremo de recepción). Alternativamente, el dispositivo terminal puede codificar los datos de enlace ascendente mediante la codificación de canal y enviar los datos de enlace ascendente al dispositivo de red a través de una interfaz aérea tras la modulación de constelación (es decir, el dispositivo terminal es un dispositivo de extremo de transmisión y el dispositivo de red es un dispositivo de extremo de recepción). Puede entenderse que, cuando el dispositivo de red se comunica con el dispositivo de red, o el dispositivo terminal se comunica con el dispositivo terminal, la comunicación puede realizarse mediante la codificación de canales. En otras palabras, tanto el dispositivo de extremo de transmisión como el dispositivo de extremo de recepción pueden ser dispositivos de red, o ambos pueden ser dispositivos terminales. Esto no está limitado.

El dispositivo terminal de la FIGURA 4 puede ser un dispositivo con una función de transceptor inalámbrico o un chip o un sistema de chips que puede disponerse en el dispositivo, puede permitir a un usuario acceder a una red y es un dispositivo configurado para proporcionar conectividad de voz y/o datos al usuario. El dispositivo terminal también puede denominarse equipo de usuario (user equipment, UE), unidad de abonado (subscriber unit), terminal (terminal), estación móvil (mobile station, MS), terminal móvil (mobile terminal, MT), o similares.

Por ejemplo, el dispositivo terminal de la FIGURA 4 puede ser un teléfono móvil (mobile phone), una computadora tipo tableta o una computadora con función de transceptor inalámbrico. Alternativamente, el dispositivo terminal puede ser una estación de usuario, una estación móvil, una estación remota, un dispositivo terminal remoto, un dispositivo terminal móvil, un dispositivo terminal de usuario, un dispositivo de comunicación inalámbrica, un agente de usuario, un aparato de usuario, un teléfono móvil, un teléfono inalámbrico, un teléfono con

protocolo de inicio de sesión (session initiation protocol, SIP), un bucle local inalámbrico (wireless local loop, WLL), un procesamiento digital personal (asistente digital personal, PDA), un dispositivo portátil con función de comunicación inalámbrica, un dispositivo informático, un dispositivo de procesamiento conectado a un módem inalámbrico, un dispositivo montado en un

5 vehículo, un dispositivo para llevar puesto, un dispositivo terminal en el internet de las cosas, un electrodoméstico, una terminal de realidad virtual (virtual reality, VR), una terminal de realidad aumentada (augmented reality, AR), una terminal inalámbrica de control industrial, una terminal inalámbrica de conducción autónoma, una terminal inalámbrica de telemedicina, una terminal inalámbrica de red inteligente, una terminal inalámbrica de ciudad inteligente (smart city), una

10 terminal inalámbrica de hogar inteligente (smart home), un vehículo con capacidad de comunicación vehículo a vehículo (vehicle-to-vehicle, V2V), un vehículo inteligente conectado, un vehículo aéreo no tripulado con capacidad de comunicación entre vehículos aéreos no tripulados (UAV to UAV, U2U), un dispositivo terminal en una red futura, un dispositivo terminal en una futura red móvil terrestre pública evolucionada (public land mobile network, PLMN), o

15 similares. Esto no está limitado.

El dispositivo de red de la FIGURA 4 puede ser cualquier dispositivo que se despliegue en una red de acceso y que pueda realizar comunicaciones inalámbricas con un dispositivo terminal, o puede ser un chip o un sistema de chips que pueda disponerse en el dispositivo, o puede ser un nodo lógico o un módulo lógico o una función implementada por software, y puede

20 estar configurado para implementar funciones como una función de control físico de radio, programación de recursos y gestión de recursos de radio, control de acceso de radio y gestión de la movilidad. En concreto, el dispositivo de red puede ser un dispositivo que admita el acceso por cable o puede ser un dispositivo que admita el acceso inalámbrico.

Por ejemplo, el dispositivo de red puede incluir uno o más nodos de red de acceso

25 (access network, AN) o de red de acceso por radio (radio access network, RAN). El nodo AN/RAN puede ser un NodoB evolucionado continuo (gNB), un punto de recepción de transmisión (transmission reception point, TRP), un NodoB evolucionado (evolved NodeB, eNB), un controlador de red de radio (radio network controller, RNC), un NodoB (NodeB, NB), un controlador de estación base (base station controller, BSC), una estación transceptora base

30 (base transceiver station, BTS), una estación base doméstica (por ejemplo, un NodoB evolucionado doméstico o un NodoB doméstico, HNB), una unidad de banda base (baseband unit, BBU), un punto de acceso (access point, AP) de fidelidad inalámbrica (wireless fidelity, Wi-Fi) o similares.

En otro ejemplo, el dispositivo de red puede incluir una unidad de banda base (baseband

35 unit, BBU) y una unidad de radio remota (remote radio unit, RRU). La BBU y la RRU pueden colocarse en lugares diferentes. Por ejemplo, la RRU es remota y se coloca en una zona de

mucho tráfico, y la BBU se coloca en una sala de equipos central. Alternativamente, la BBU y la RRU pueden colocarse en una misma sala de equipos. Alternativamente, la BBU y la RRU pueden ser componentes diferentes en un mismo bastidor.

En otro ejemplo, el dispositivo de red puede ser alternativamente un dispositivo que
 5 incluya un nodo de unidad central (central unit, CU), que incluya un nodo de unidad distribuida (distributed unit, DU) o que incluya un nodo CU y un nodo DU. Por ejemplo, el dispositivo de red puede dividirse en una CU y una DU desde una perspectiva de funciones lógicas. Algunas funciones de la capa de protocolo están controladas de forma centralizada por la CU, y una parte o todas las funciones restantes de la capa de protocolo están distribuidas en la DU, y la
 10 CU controla de forma centralizada la DU. Además, la unidad central CU puede dividirse a su vez en un plano de control (CU-CP) y un plano de usuario (CU-UP). En diferentes sistemas, una CU (incluyendo una CU-CP o una CU-UP) o una DU pueden tener diferentes nombres. Por ejemplo, en un sistema de red de acceso por radio abierta (open radio access network, O-RAN), la CU también puede denominarse O-CU (open CU), la DU también puede denominarse O-DU,
 15 la CU-CP también puede denominarse O-CU-CP y la CU-UP también puede denominarse O-CU-UP.

Basándose en las descripciones anteriores del dispositivo terminal y del dispositivo de red, opcionalmente, el método de codificación de códigos polares proporcionado en las modalidades de esta aplicación puede ser implementado por el dispositivo terminal o el
 20 dispositivo de red, o puede ser implementado por un componente del dispositivo terminal o del dispositivo de red, por ejemplo, implementado por un circuito integrado específico de la aplicación (application-specific integrated circuit, ASIC), una matriz de puertas programables en campo (field programmable gate array, FPGA), o software (por ejemplo, código de programa en una memoria) desplegado en el dispositivo terminal o el dispositivo de red. Esto no está limitado.

25 Opcionalmente, en las modalidades de esta solicitud, el dispositivo de extremo de transmisión (o referido como fuente) y el dispositivo de extremo de recepción (o referido como destino) pueden realizar la codificación y decodificación mediante un procedimiento que se muestra en la FIGURA 5.

El dispositivo de extremo de transmisión puede realizar la codificación de la fuente en
 30 los bits generados por el dispositivo de extremo de transmisión, para obtener un flujo de bits de la fuente, realizar la codificación del canal en el flujo de bits de la fuente mediante la codificación del canal y, a continuación, enviar un símbolo de modulación al dispositivo de extremo de recepción a través de un canal ruidoso después de la modulación. Al recibir el símbolo de modulación a través del canal ruidoso, el dispositivo de extremo de recepción puede realizar la
 35 demodulación y, a continuación, llevar a cabo la decodificación del canal para restaurar el flujo de bits de la fuente y, después, realizar la decodificación de la fuente para obtener un resultado

de decodificación.

Durante la implementación específica, como se muestra en la FIGURA 4, por ejemplo, cada dispositivo terminal o cada dispositivo de red puede utilizar una estructura de composición mostrada en la FIGURA 6, o incluir los componentes mostrados en la FIGURA 6. La FIGURA 6 es un diagrama de una composición de un aparato de comunicación 600 de acuerdo con una modalidad de esta solicitud. El aparato de comunicación 600 puede ser un dispositivo terminal, o un chip o un sistema en chip en el dispositivo terminal, o puede ser un dispositivo de red, o un chip o un sistema en chip en el dispositivo de red. Como se muestra en la FIGURA 6, el aparato de comunicación 600 incluye un procesador 601, un transceptor 602 y una línea de comunicación 603.

Además, el aparato de comunicación 600 puede incluir además una memoria 604. El procesador 601, la memoria 604 y el transceptor 602 pueden estar conectados a través de la línea de comunicación 603.

El procesador 601 es una unidad central de procesamiento (central processing unit, CPU), un procesador de propósito general, un procesador de red (network processor, NP), un procesador digital de señales (digital signal processing, DSP), un microprocesador, un microcontrolador, un dispositivo lógico programable (programmable logic device, PLD), o cualquier combinación de los mismos. Alternativamente, el procesador 601 puede ser otro aparato que tenga una función de procesamiento, por ejemplo, un circuito, un componente o un módulo de software. Esto no está limitado.

El transceptor 602 está configurado para comunicarse con otro dispositivo u otra red de comunicación. La otra red de comunicación puede ser una Ethernet, una red de acceso por radio (radio access network, RAN), una red de área local inalámbrica (wireless local area network, WLAN), o similares. El transceptor 602 puede ser un módulo, un circuito, un transceptor o cualquier aparato que pueda implementar la comunicación.

La línea de comunicación 603 está configurada para transmitir información entre los componentes incluidos en el aparato de comunicación 600.

La memoria 604 está configurada para almacenar instrucciones. Las instrucciones pueden ser programas de computadora.

La memoria 604 puede ser una memoria de sólo lectura (read-only memory, ROM) u otro tipo de dispositivo de almacenamiento estático que pueda almacenar información y/o instrucciones estáticas, o puede ser una memoria de acceso aleatorio (random access memory, RAM) u otro tipo de dispositivo de almacenamiento dinámico que pueda almacenar información y/o instrucciones, o puede ser una memoria de sólo lectura programable eléctricamente borrable (electrically erasable programmable read-only memory, EEPROM), una memoria de sólo lectura en disco compacto (compact disc read-only memory, CD-ROM) u otro tipo de almacenamiento

en disco compacto, un almacenamiento en disco óptico (incluidos un disco compacto, un disco láser, un disco óptico, un disco versátil digital, un disco Blu-ray, y similares), un medio de almacenamiento en disco u otro dispositivo de almacenamiento magnético, o similares. Esto no está limitado.

- 5** Cabe señalar que la memoria 604 puede existir independientemente del procesador 601, o puede estar integrada con el procesador 601. La memoria 604 puede estar configurada para almacenar las instrucciones, el código de programa, algunos datos o similares. La memoria 604 puede estar situada en el aparato de comunicación 600, o puede estar situada fuera del aparato de comunicación 600. Esto no está limitado. El procesador 601 está configurado para
- 10** ejecutar las instrucciones almacenadas en la memoria 604, para realizar el método de codificación de códigos polares proporcionado en las siguientes modalidades de esta solicitud.

En un ejemplo, el procesador 601 puede incluir una o más CPU, por ejemplo, una CPU 0 y una CPU 1 en la FIGURA 6.

- En una implementación opcional, el aparato de comunicación 600 incluye una pluralidad
- 15** de procesadores. Por ejemplo, además del procesador 601 de la FIGURA 6, el aparato de comunicación 600 puede incluir además un procesador 607.

- En una implementación opcional, el aparato de comunicación 600 incluye además un dispositivo de salida 605 y un dispositivo de entrada 606. Por ejemplo, el dispositivo de entrada 606 es un dispositivo como un teclado, un ratón, un micrófono o una palanca, y el dispositivo de
- 20** salida 605 es un dispositivo como una pantalla o un altavoz (speaker).

- Cabe señalar que, el aparato de comunicación 600 puede ser una computadora de escritorio, una computadora portátil, un servidor de red, un teléfono móvil, una tableta, una terminal inalámbrica, un dispositivo integrado, un sistema de chip o un dispositivo que tenga una estructura similar en la FIGURA 6. Además, la estructura de la composición mostrada en la
- 25** FIGURA 6 no constituye una limitación del aparato de comunicación. Además de los componentes mostrados en la FIGURA 6, el aparato de comunicación puede incluir más o menos componentes que los mostrados en la figura, o combinar algunos componentes o tener disposiciones diferentes de los componentes.

- En esta modalidad de esta solicitud, el sistema de chip puede incluir un chip, o puede
- 30** incluir un chip y otro dispositivo discreto.

- Además, las acciones, términos y similares en las modalidades de esta solicitud pueden referenciarse mutuamente. Esto no está limitado. En las modalidades de esta solicitud, los nombres de los mensajes intercambiados entre dispositivos, los nombres de los parámetros de los mensajes o similares son meros ejemplos. Alternativamente, pueden utilizarse otros
- 35** nombres durante la implementación específica. Esto no está limitado.

Con referencia al sistema de comunicación mostrado en la FIGURA 4, a continuación

se describe un método de codificación de códigos polares proporcionado en una modalidad de esta solicitud con referencia a la FIGURA 7. El dispositivo de extremo de transmisión puede ser cualquier dispositivo terminal o dispositivo de red del sistema de comunicación mostrado en la FIGURA 4, y el dispositivo de extremo de recepción también puede ser cualquier dispositivo terminal o dispositivo de red del sistema de comunicación mostrado en la FIGURA 4. El dispositivo de extremo de transmisión o el dispositivo de extremo de recepción descritos en la siguiente modalidad pueden tener los componentes mostrados en la FIGURA 6.

La FIGURA 7 es un diagrama de flujo de un método de codificación de códigos polares de acuerdo con una modalidad de esta solicitud. Como se muestra en la FIGURA 7, el método puede incluir los siguientes pasos.

Paso 701: Un dispositivo de extremo de transmisión mapea, basándose en una secuencia de fiabilidad, una secuencia de bits de información cuya longitud es k a k bits de una primera secuencia cuya longitud es N , para obtener una segunda secuencia.

La secuencia de fiabilidad puede indicar la fiabilidad correspondiente a cada bit de la secuencia. Un valor mayor de la fiabilidad indica un bit más fiable correspondiente a la fiabilidad. La longitud de la secuencia de fiabilidad puede ser N , y N es un número entero positivo.

Opcionalmente, N es inferior a 1024.

Opcionalmente, la secuencia de fiabilidad se puede determinar en función de uno o varios de los siguientes: la fiabilidad, un espectro de códigos y un resultado de optimización del rendimiento.

En esta modalidad de esta solicitud, la secuencia de fiabilidad cuya longitud es N puede construirse por separado para la secuencia de bits de información basándose en parámetros como la fiabilidad, el espectro de código y el resultado de la optimización del rendimiento. Cuando una secuencia codificada se puncia posteriormente en un orden de atrás hacia delante, se puede reducir una cantidad de puntos malos de rendimiento, mejorar las características del espectro de código y mejorar el rendimiento de la decodificación.

Por ejemplo, la longitud N de la secuencia de fiabilidad es 32, y la secuencia de fiabilidad puede ser una secuencia de fiabilidad mostrada en la Tabla 2 siguiente, donde $W(Q_i^{Nmax})$ representa la fiabilidad, y Q_i^{Nmax} representa un bit correspondiente a la fiabilidad:

Tabla 2

$W(Q_i^{Nr})$	Q_i^{Nm}	$W(Q_i^{Nr})$	Q_i^{Nm}	$W(Q_i^{Nr})$	Q_i^{Nm}	$W(Q_i^{Nr})$	Q_i^{Nm}	$W(Q_i^{Nr})$	Q_i^{Nm}	$W(Q_i^{Nr})$	Q_i^{Nm}	$W(Q_i^{Nr})$	Q_i^{Nm}	$W(Q_i^{Nr})$	Q_i^{Nm}
0	0	4	8	8	9	12	$\frac{1}{8}$	16	7	20	$\frac{2}{5}$	24	$\frac{1}{3}$	28	$\frac{2}{7}$
1	1	5	$\frac{1}{6}$	9	6	13	$\frac{1}{2}$	17	$\frac{2}{6}$	21	$\frac{2}{8}$	25	$\frac{2}{2}$	29	$\frac{2}{9}$
2	2	6	3	10	$\frac{1}{7}$	14	$\frac{2}{0}$	18	$\frac{1}{9}$	22	11	26	$\frac{1}{5}$	30	$\frac{3}{0}$

3	4	7	5	11	$\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}$	15	$\begin{smallmatrix} 2 \\ 4 \end{smallmatrix}$	19	$\begin{smallmatrix} 2 \\ 1 \end{smallmatrix}$	23	$\begin{smallmatrix} 1 \\ 4 \end{smallmatrix}$	27	$\begin{smallmatrix} 2 \\ 3 \end{smallmatrix}$	31	$\begin{smallmatrix} 3 \\ 1 \end{smallmatrix}$
---	---	---	---	----	--	----	--	----	--	----	--	----	--	----	--

Puede entenderse que la tabla 2 se define a partir de 0 bits, o puede definirse a partir de 1 bit, es decir, 0, 1, ..., y 31 descritos anteriormente pueden sustituirse respectivamente por 1, 2, ..., y 32. Esto no está limitado.

Refiriéndose a la secuencia de fiabilidad, el dispositivo de extremo de transmisión puede mapear la secuencia de bits de información cuya longitud es k a los k bits de la primera secuencia cuya longitud es N , para obtener la segunda secuencia.

$k \leq K \leq N$, K es un valor máximo de la longitud de la secuencia de bits de información, y k , K y N son todos números enteros positivos.

Por ejemplo, la longitud k de la secuencia de bits de información puede ser cualquiera de los valores siguientes: 3, 4, 5, 6, 7, 8, 9, 10, y 11. En este caso, K es igual a 11.

Por ejemplo, la longitud N de la primera secuencia puede ser 32.

El dispositivo de extremo de transmisión puede seleccionar los primeros k bits de la primera secuencia como posiciones de bits de información basándose en la secuencia de fiabilidad en orden descendente de fiabilidad, y mapear secuencialmente la secuencia de bits de información cuya longitud es k a los k bits, es decir, mapear la secuencia de bits de información a las k posiciones de bits de información de la primera secuencia, y poner a 0 los valores de los $N-k$ bits restantes de la primera secuencia, para obtener la segunda secuencia.

La segunda secuencia puede incluir k posiciones de bits de información, x posiciones de bits de PC, y $N-k-x$ posiciones de bits congelados, $1 \leq x$, y x es un número entero positivo.

Opcionalmente, en un ejemplo en el que la definición comienza a partir de 0 bits, la secuencia de bits de información puede denotarse como u_0^{k-1} , y la segunda secuencia puede denotarse como a_0^{N-1} . Puede entenderse que la definición puede comenzar alternativamente a partir de 1 bit, es decir, la secuencia de bits de información puede denotarse como u_1^k , y la segunda secuencia puede denotarse como a_1^N . Esto no está limitado.

A continuación se describe un proceso de generación de la segunda secuencia mediante un ejemplo en el que la definición parte de 0 bits.

Por ejemplo, la secuencia de fiabilidad es la secuencia de fiabilidad cuya longitud es 32 mostrada en la Tabla 2, y la longitud de la primera secuencia es 32. Cuando la longitud de la secuencia de bits de información es 3, se puede determinar, basándose en la secuencia de fiabilidad mostrada en la Tabla 2, que los 3 primeros bits en orden descendente de fiabilidad de la primera secuencia son {31 30 29}, y se puede determinar que las posiciones de los bits de información son {31 30 29}. La secuencia de bits de información cuya longitud es 3 se mapea a los 3 bits de la primera secuencia, y los valores de los 29 bits restantes de la primera secuencia

se ponen a 0, para obtener la segunda secuencia.

En otro ejemplo, la secuencia de fiabilidad es la secuencia de fiabilidad cuya longitud es 32 mostrada en la Tabla 2, y la longitud de la primera secuencia es 32. Cuando la longitud de la secuencia de bits de información es 11, se puede determinar, basándose en la secuencia de fiabilidad mostrada en la Tabla 2, que los 11 primeros bits en orden descendente de fiabilidad de la primera secuencia son {31 30 29 27 23 15 22 13 14 11 28}, y se puede determinar que las posiciones de los bits de información son {31 30 29 27 23 15 22 13 14 11 28}. La secuencia de bits de información cuya longitud es 11 se mapea a los 11 bits de la primera secuencia, y los valores de los 21 bits restantes de la primera secuencia se ponen a 0, para obtener la segunda secuencia.

Paso 702: El dispositivo de extremo de transmisión multiplica la segunda secuencia por una primera matriz para obtener una tercera secuencia.

La primera matriz puede ser una matriz con todos los 0 por debajo de una diagonal y todos los 1 en la diagonal, una ponderación de columna de una columna correspondiente a la posición del bit de PC en la primera matriz es mayor que 1, una ponderación de columna de una columna correspondiente a la posición del bit congelado o a la posición del bit de información es igual a 1, y una cantidad de columnas cuyas ponderaciones de columna son mayores que 1 en la primera matriz es menor o igual que $N-K$. La primera matriz también puede denominarse matriz triangular superior, matriz de pretransformación (pre-transform) o similar. La primera matriz también puede denotarse como T_{pre} . Esto no está limitado.

Por ejemplo, N es 32 y K es 11. La primera matriz puede ser una matriz de 32×32 , y la cantidad de columnas cuyas ponderaciones de columna son mayores que 1 en la primera matriz es menor o igual que 21.

Opcionalmente, la primera matriz puede ser una matriz dispersa.

Opcionalmente, una ponderación de columna de la primera matriz puede ser inferior o igual a un primer valor, garantizando así la escasez de la primera matriz y obteniendo mejores características del espectro del código.

Por ejemplo, el primer valor puede ser cualquiera de los siguientes: 4, 5, 6 o 7.

Basándose en la primera matriz, el dispositivo de extremo de transmisión puede determinar, basándose en la primera matriz, una o más posiciones de bits de PC y bits comprobados por las posiciones de bits de PC; y determinar un valor de la posición del bit de PC en la segunda secuencia basándose en un valor del bit comprobado por la posición del bit de PC, para obtener la tercera secuencia. La tercera secuencia puede incluir la una o más posiciones de bits de PC.

Opcionalmente, un valor de la posición del bit de PC puede ser un OR exclusivo de un valor del bit comprobado por la posición del bit de PC.

Opcionalmente, se utiliza un ejemplo en el que la definición comienza a partir de 0 bits, y la tercera secuencia puede denotarse como v_0^{N-1} , y $v_0^{N-1} = a_0^{N-1}T_{pre}$. Puede entenderse que la definición puede comenzar alternativamente a partir de 1 bit, es decir, la tercera secuencia puede denotarse como v_1^N , y $v_1^N = a_1^N T_{pre}$. Esto no está limitado.

5 Opcionalmente, una diferencia entre un número de la posición del bit de PC y un número del bit comprobado por la posición del bit de PC puede ser un número primo. La diferencia entre el número de la posición del bit de PC y el número del bit comprobado por la posición del bit de PC está limitada a ser un número primo. Esto puede reducir en gran medida el espacio de búsqueda y reducir al máximo la pérdida de optimalidad, al tiempo que se mejoran las características del espectro de código.

10 Por ejemplo, N es 32, la longitud k de la secuencia de bits de información es de 3 a 11, y la secuencia de bits de información se define empezando por 0 bits. Cuando k=11, las posiciones de los bits de información correspondientes a la secuencia de bits de información son {31 30 29 27 23 15 22 13 14 11 28}. Cuando k=10, las posiciones de los bits de información correspondientes a la secuencia de bits de información son {31 30 29 27 23 15 22 13 14 11}.

15 Cuando k=9, las posiciones de los bits de información correspondientes a la secuencia de bits de información son {31 30 29 27 23 15 22 13 14}. Cuando k=8, las posiciones de los bits de información correspondientes a la secuencia de bits de información son {31 30 29 27 23 15 22 13}.

20 Cuando k=7, las posiciones de los bits de información correspondientes a la secuencia de bits de información son {31 30 29 27 23 15 22}. Cuando k=6, las posiciones de bits de información correspondientes a la secuencia de bits de información son {31 30 29 27 23 15}.

25 Cuando k=5, las posiciones de bits de información correspondientes a la secuencia de bits de información son {31 30 29 27 23}. Cuando k=4, las posiciones de los bits de información correspondientes a la secuencia de bits de información son {31 30 29 27}. Cuando k=3, las posiciones de los bits de información correspondientes a la secuencia de bits de información son {31 30 29}.

Dado que la posición del bit de PC es una posición de bit congelado, y que hay al menos una posición de bit de información antes de la posición del bit de PC, para la secuencia de bits de información con k=11, las posiciones potenciales del bit de PC pueden ser los 10 bits siguientes: {12 16 17 18 19 20 21 24 25 26}, y las posiciones de bits de información que pueden ser comprobadas por las 10 posiciones potenciales de bits de PC son las posiciones de bits de información que se muestran en la Tabla 3:

Tabla 3

	Posición del bit de PC	Posición del bit de información comprobada por la posición del bit de PC	Cantidad de candidatos a la función de PC (PC function candidates num)
5	12	11	2
	16	11 13 14 15	16
	17	11 13 14 15	16
	18	11 13 14 15	16
	19	11 13 14 15	16
10	20	11 13 14 15	16
	21	11 13 14 15	16
	24	11 13 14 15 22 23	64
	25	11 13 14 15 22 23	64
	26	11 13 14 15 22 23	64

En la tabla 3, hay un total de $2 \times (16^6) \times (64^3) = 8,7961 \times 10^{12}$ funciones de PC potenciales. Si la diferencia entre el número de la posición del bit de PC y el número del bit comprobado por la posición del bit de PC se limita a ser un número primo, como se muestra en la tabla 4, hay un total de $1 \times (2^3) \times (4^4) \times (8^2) = 131072$ funciones de PC potenciales. Esto puede reducir en gran medida el espacio de búsqueda y reducir al máximo la pérdida de optimalidad, al tiempo que se mejoran las características del espectro de código.

Tabla 4

	Posición del bit de PC	Posición del bit de información comprobada por la posición del bit del PC	Cantidad de candidatos a la función de PC (PC function candidates num)
20	12	nulo	1
	16	11 13	4
	17	14	2
	18	11 13 15	8
	19	14	2
25	20	13 15	4
	21	14	2
	24	11 13	4
	25	14 22	4
	26	13 15 23	8

En comparación con $k=11$ descrito anteriormente, como se muestra en la Tabla 5 siguiente, para una secuencia de bits de información con $k=10$, puede añadirse una posición del bit de PC 28 sobre la base de $k=11$ mostrada en la Tabla 4; para una secuencia de bits de información con $k=9$, puede añadirse una posición del bit de PC 11 sobre la base de $k=10$; para una secuencia de bits de información con $k=8$, puede añadirse una posición del bit de PC 14 sobre la base de $k=9$; para una secuencia de bits de información con $k=7$, puede añadirse una posición del bit de PC 13 sobre la base de $k=8$; para una secuencia de bits de información con $k=6$, puede añadirse una posición del bit de PC 22 sobre una base de $k=7$; para una secuencia

de bits de información con $k=5$, puede añadirse una posición del bit de PC 15 sobre una base de $k=6$; para una secuencia de bits de información con $k=4$, puede añadirse una posición del bit de PC 23 sobre una base de $k=5$; y para una secuencia de bits de información con $k=3$, puede añadirse una posición del bit de PC 27 sobre una base de $k=4$. Esto puede reducir en gran medida el espacio de búsqueda y reducir al máximo la pérdida de optimalidad, al tiempo que se mejoran las características del espectro de código.

Tabla 5

k	posición del bit de PC	Posición del bit de información comprobada por la posición del bit del PC
10	28	11 15 23
9	11	nulo
8	14	nulo
7	13	nulo
6	22	15
5	15	nulo
4	23	nulo
3	27	nulo

Basándose en la posición del bit de PC y en la posición del bit de información comprobada por la posición del bit de PC mostrada en la Tabla 4 y en la Tabla 5, en un primer ejemplo posible, la primera matriz puede ser una matriz de 32×32 a continuación, y la primera matriz puede representarse alternativamente como una matriz mostrada en la FIGURA 8:

5

10

15

20

25

30

35

[illegible]

Basándose en la primera matriz, la posición del bit de PC de la tercera secuencia obtenida de acuerdo con el paso 702 puede ser uno o más de los siguientes bits de la tercera secuencia: 16, 18, 21 o 25.

Quando la posición del bit de PC es el 16^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es el 13^{ésimo} bit de la tercera secuencia; o cuando la posición del bit de PC es el 18^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es uno o más de los siguientes bits de la tercera secuencia: 11, 13 y 15; o cuando la posición del bit de PC es el 21^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es un 14^{ésimo} bit de la tercera secuencia; o cuando la posición del bit de PC es el 25^{ésimo} bit de la tercera secuencia, el bit comprobado por la posición del bit de PC es un 22^{ésimo} bit de la tercera secuencia.

En un segundo ejemplo posible, la primera matriz puede ser alternativamente una matriz de 32×32 a continuación, y la primera matriz puede representarse alternativamente como la matriz mostrada en la FIGURA 9:

representarse alternativamente mediante el uso de un conjunto de ecuaciones de PC, o se describe como que la primera matriz se representa mediante el uso de un conjunto de ecuaciones de PC.

El conjunto de ecuaciones de PC puede incluir una o más ecuaciones de PC, y una
5 cantidad de ecuaciones de PC incluidas en el conjunto de ecuaciones de PC es igual a una cantidad de posiciones de bits de PC (o puede describirse como una cantidad de columnas cuyas ponderaciones de columna son superiores a 1 en la primera matriz). Cada ecuación de PC incluye al menos dos elementos, un valor máximo de un elemento en cada ecuación de PC representa una posición de bit de PC (o puede describirse como que el valor máximo representa
10 un número de columna cuya ponderación de columna es mayor que 1 en la primera matriz), y un elemento distinto de la posición de bit de PC en cada ecuación de PC representa un bit comprobado por la posición de bit de PC.

En un primer ejemplo posible, si el conjunto de ecuaciones de PC es {[13 16]}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 16 y
15 el bit comprobado por la posición del bit de PC es 13. Si el conjunto de ecuaciones de PC es {[11 13 15 18]}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 18 y los bits comprobados por la posición del bit de PC son 11, 13 y 15. Si el conjunto de ecuaciones de PC es {[14 21]}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 21 y el bit comprobado por la posición del bit de
20 PC es 14. Si el conjunto de ecuaciones de PC es {[22 25]}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 25 y el bit comprobado por la posición del bit de PC es 21. Si el conjunto de ecuaciones de PC es {[13 16], [11 13 15 18], [14 21], [22 25]}, se puede determinar que la cantidad de posiciones de bits de PC es 4, las posiciones de bits de PC son 16, 18, 21 y 25, el bit comprobado por la posición del bit de PC 16
25 es 13, los bits comprobados por la posición del bit de PC 18 son 11, 13 y 15, el bit comprobado por la posición del bit de PC 21 es 14 y el bit comprobado por la posición del bit de PC 25 es 22.

En un segundo ejemplo posible, si el conjunto de ecuaciones de PC es {[14 19]}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 19 y el bit comprobado por la posición del bit de PC es 14. Si el conjunto de ecuaciones de PC
30 es {[11 14 21]}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 21 y los bits comprobados por la posición del bit de PC son 11 y 14. Si el conjunto de ecuaciones de PC es {[14 22 25]}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 25 y los bits comprobados por la posición del bit de PC son 14 y 22. Si el conjunto de ecuaciones de PC es {[13 26]}, se puede
35 determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 26 y el bit comprobado por la posición del bit de PC es 13. Si el conjunto de ecuaciones de PC es

{[14 19], [11 14 21], [14 22 25], [13 26]}, se puede determinar que la cantidad de posiciones de bits de PC es 4, las posiciones de bits de PC son 19, 21, 25 y 26, el bit comprobado por la posición del bit de PC 19 es 14, los bits comprobados por la posición del bit de PC 21 son 11 y 14, los bits comprobados por la posición del bit de PC 25 son 14 y 22, y el bit comprobado por la posición del bit de PC 26 es 13.

Basándose en el primer diseño posible, cuando la primera matriz se representa utilizando el conjunto de ecuaciones de PC, multiplicar la segunda secuencia a_0^{N-1} por la primera matriz T_{pre} para obtener la tercera secuencia $v_0^{N-1} = a_0^{N-1} T_{pre}$ puede representarse como:

```

10      para n=0 a N-1;
          si n ∈ F
              para p=0: a |PF|
                  pcldx= max( PF(p) );
                  chkldx=los otros índices en PF(p);
15      si n == pcldx
                   $v_n = \text{mod}(\sum_{i \in \text{chkldx}} a_i, 2)$ 
                  si no
                      ~0
                  fin
20      fin
          si no
                   $v_n = a_n$ ;
          fin

```

N representa una longitud de secuencia (por ejemplo, 32, y N-1 puede ser 31), F representa un conjunto de posiciones de bits congelados, PF representa un conjunto de ecuaciones de PC, pcldx representa una posición del bit de PC, PF(p) representa una p^{ésima} ecuación de PC en el conjunto de ecuaciones de PC, chkldx representa un bit comprobado por la posición del bit de PC, v_n representa un n^{ésimo} valor en la tercera secuencia, a_i representa un i^{ésimo} valor en la segunda secuencia, y a_n representa un n^{ésimo} valor en la segunda secuencia.

En un segundo diseño posible, a diferencia de representar, utilizando la primera matriz o el conjunto de ecuaciones de PC, la posición del bit de PC y el bit comprobado por la posición del bit de PC descritos anteriormente, la posición del bit de PC y el bit comprobado por la posición del bit de PC pueden representarse alternativamente utilizando un conjunto de posiciones del bit de PC y un conjunto de bits correspondiente a la posición del bit de PC, o se describe como que la primera matriz o el conjunto de ecuaciones de PC se representa utilizando

un conjunto de posiciones del bit de PC y un conjunto de bits correspondiente a la posición del bit de PC.

El conjunto de posiciones de bits de PC incluye una o más posiciones de bits de PC (o se describe como que un elemento del conjunto de posiciones de bits de PC representa un número de columna cuya ponderación de columna es mayor que 1 en la primera matriz, y una cantidad de elementos incluidos en el conjunto de posiciones de bits de PC es una cantidad de columnas cuyas ponderaciones de columna son mayores que 1 en la primera matriz), y el conjunto de bits correspondiente a la posición del bit de PC incluye uno o más bits comprobados por la posición del bit de PC.

- En un primer ejemplo posible, si el conjunto de posiciones de bits de PC es {16}, y el conjunto de bits correspondiente a la posición del bit de PC es {13}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 16, y el bit comprobado por la posición del bit de PC es 13. Si el conjunto de posiciones de bits de PC es {18}, y el conjunto de bits correspondiente a la posición del bit de PC es {11 13 15}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 18, y los bits comprobados por la posición del bit de PC son 11, 13 y 15. Si el conjunto de posiciones de bits de PC es {21}, y el conjunto de bits correspondiente a la posición del bit de PC es {14}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 21, y el bit comprobado por la posición del bit de PC es 14. Si el conjunto de posiciones de bits de PC es {25}, y el conjunto de bits correspondiente a la posición del bit de PC es {22}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 25, y el bit comprobado por la posición del bit de PC es 21. Si el conjunto de posiciones de bits de PC es {16 18 21 25}, y el conjunto de bits correspondiente a la posición del bit de PC es {[13], [11 13 15], [14], [22]}, se puede determinar que la cantidad de posiciones de bits de PC es 4, las posiciones de bits de PC son 16, 18, 21 y 25, el bit comprobado por la posición del bit de PC 16 es 13, los bits comprobados por la posición del bit de PC 18 son 11, 13 y 15, el bit comprobado por la posición del bit de PC 21 es 14 y el bit comprobado por la posición del bit de PC 25 es 22.

- En un segundo ejemplo posible, si el conjunto de posiciones de bits de PC es {19}, y el conjunto de bits correspondiente a la posición del bit de PC es {14}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 19, y el bit comprobado por la posición del bit de PC es 14. Si el conjunto de posiciones de bits de PC es {21}, y el conjunto de bits correspondiente a la posición del bit de PC es {11 14}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 21, y los bits comprobados por la posición del bit de PC son 11 y 14. Si el conjunto de posiciones de bits de PC es {25}, y el conjunto de bits correspondiente a la posición del bit de PC es {14 22}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 25, y

los bits comprobados por la posición del bit de PC son 14 y 22. Si el conjunto de posiciones de bits de PC es {26}, y el conjunto de bits correspondiente a la posición del bit de PC es {13}, se puede determinar que la cantidad de posiciones de bits de PC es 1, la posición del bit de PC es 26, y el bit comprobado por la posición del bit de PC es 13. Si el conjunto de posiciones de bits de PC es {19 21 25 26}, y el conjunto de bits correspondiente a la posición del bit de PC es {[14], [11 14], [14 22], [13]}, se puede determinar que la cantidad de posiciones de bits de PC es 4, las posiciones de bits de PC son 19, 21, 25 y 26, el bit comprobado por la posición del bit de PC 19 es 14, los bits comprobados por la posición del bit de PC 21 son 11 y 14, los bits comprobados por la posición del bit de PC 25 son 14 y 22, y el bit comprobado por la posición del bit de PC 26 es 13.

Basándose en el segundo diseño posible, cuando la primera matriz se representa utilizando el conjunto de posiciones de bits de PC y el conjunto de bits correspondiente a la posición del bit de PC, multiplicar la segunda secuencia a_0^{N-1} por la primera matriz T_{pre} para obtener la tercera secuencia $v_0^{N-1} = a_0^{N-1} T_{pre}$ puede representarse como:

$$v_i = \begin{cases} a_i, & i \notin \mathbf{PBS} \\ a_i = \sum_{j,j \in \mathbf{CBS}} a_j & a_{i,j \in \mathbf{PBS}}, & i \in \mathbf{PBS}, \end{cases} \quad i = \{0, 1, 2, \dots, N-1\}.$$

N representa una longitud de secuencia (por ejemplo, 32, y N-1 puede ser 31), v_i representa un $i^{\text{ésimo}}$ valor en la tercera secuencia, a_i representa un $i^{\text{ésimo}}$ valor en la segunda secuencia, a_j representa un $j^{\text{ésimo}}$ valor en la segunda secuencia, PBS representa un conjunto de posiciones de bits de PC, y CBS representa un conjunto de bits correspondiente a la posición del bit de PC.

Basándose en la descripción anterior de la posición del bit de PC y del bit comprobado por la posición del bit de PC, en los ejemplos anteriores, la posición del bit de PC y el bit comprobado por la posición del bit de PC se definen a partir de 0 bits. Puede entenderse que la posición del bit de PC y el bit comprobado por la posición del bit de PC pueden definirse alternativamente a partir de 1 bit, es decir, 0, 1, ..., y 31 descritos anteriormente pueden sustituirse respectivamente por 1, 2, ..., y 32. Esto no está limitado.

Paso 703: El dispositivo de extremo de transmisión realiza una codificación polar en la tercera secuencia para obtener una cuarta secuencia.

El dispositivo de extremo de transmisión puede multiplicar la tercera secuencia por una matriz de codificación polar, para obtener la cuarta secuencia.

Por ejemplo, la matriz de codificación polar puede ser G_N , G_N es un $\log_2 N$ producto de Kronecker de G_2 , $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, and $G_N = G_2^{\otimes \log_2 N}$.

Opcionalmente, se utiliza un ejemplo en el que la definición comienza a partir de 0 bits, y la cuarta secuencia puede denotarse como c_0^{N-1} , y $c_0^{N-1} = v_0^{N-1} G_N$. Puede entenderse que la

definición puede comenzar alternativamente a partir de 1 bit, es decir, la tercera secuencia puede denotarse como c_1^N , y $c_1^N = v_1^N G_N$. Esto no está limitado.

Basándose en el método mostrado en la FIGURA 7, una modalidad de la presente solicitud proporciona un esquema de codificación basado en un código polar PC anidado, para
 5 construir por separado, para una secuencia de bits de información, una secuencia de fiabilidad cuya longitud es N y que tiene una característica de anidamiento. De este modo, cuando la punción se realiza en un orden de atrás hacia delante, se reduce la cantidad de puntos malos de rendimiento, se mejoran las características del espectro del código y se mejora el rendimiento de la decodificación. Además, en esta modalidad de la presente solicitud, los códigos polares
 10 pueden construirse basándose en una misma primera matriz para secuencias de diferentes cantidades de bits de información, es decir, la primera matriz tiene una característica de anidamiento para las secuencias de diferentes cantidades de bits de información, y la primera matriz no cambia con un cambio de una cantidad de bits de información de las secuencias; o puede describirse que las posiciones de bits de PC y las ecuaciones de PC correspondientes a
 15 las secuencias de diferentes cantidades de bits de información están anidadas, y no es necesario almacenar por separado las posiciones de bits de PC y las ecuaciones de PC correspondientes para las secuencias de bits de información de diferentes longitudes, con lo que se reduce la complejidad del hardware, se reducen las sobrecargas del hardware, se mejoran las características del espectro de código, se reduce la complejidad de la decodificación
 20 y se mejora el rendimiento de la decodificación (por ejemplo, se multiplexa un módulo de decodificación de lista de cancelación sucesiva (successive cancellation list, SCL) y se mejora el rendimiento de decodificación del módulo de decodificación SCL (como SCL8)).

Basándose en el método mostrado en la FIGURA 7, tras determinar la cuarta secuencia basándose en la codificación polar, el dispositivo de extremo de transmisión puede realizar
 25 además un procesamiento de intercalación en la cuarta secuencia basándose en un patrón de intercalación, para obtener una secuencia intercalada.

En un primer diseño posible, el procesamiento de intercalación puede realizarse en la cuarta secuencia basándose en un primer patrón de intercalación, para obtener una quinta secuencia.

30 El primer patrón de intercalación se puede determinar basándose en el rendimiento de la secuencia y en una característica de anidamiento mediante el diseño de un algoritmo de búsqueda, es decir, el primer patrón de intercalación tiene una característica de anidamiento para secuencias de diferentes cantidades de bits de información, y no cambia con el cambio de una longitud de la secuencia de bits de información. En comparación con el diseño del primer patrón de intercalación de una manera de enumeración en la que hay que enumerar N! casos
 35

(por ejemplo, cuando N es igual a 32, hay que enumerar $32! = 2.6313 \times 10^{35}$ casos), en esta modalidad de esta solicitud, el primer patrón de intercalación se diseña mediante el diseño de un algoritmo de búsqueda. Esto puede reducir enormemente la complejidad al tiempo que garantiza un rendimiento óptimo de la secuencia y la característica de anidamiento.

5 Por ejemplo, la búsqueda de la posición de punción puede realizarse en una pluralidad de muestras de la cuarta secuencia (la pluralidad de muestras de la cuarta secuencia puede corresponder a secuencias de bits de información de diferentes longitudes, y una longitud de cada secuencia de bits de información es menor o igual que K) utilizando un algoritmo de búsqueda de listas. Cada vez que una cantidad de punciones aumenta en 1, se calcula el
10 rendimiento medio de la secuencia de todas las secuencias, se utiliza el rendimiento medio de la secuencia como métrica (Metric), y se determina una posición correspondiente al rendimiento medio óptimo de la secuencia como posición de punción, para generar el primer patrón de intercalación. El rendimiento medio es un valor medio de $SNR@BLER=1E-2$ de las secuencias.

15 Específicamente, como se muestra en la FIGURA 10, el proceso de búsqueda puede incluir:

Paso 1: Inicializar el espacio de búsqueda de la posición de punción de 0 a N-1, y la cantidad de punciones a 0.

Paso 2: Seleccionar una posición del espacio de búsqueda actual como nueva posición de punción añadida.

20 Paso 3: Obtener el rendimiento de todas las secuencias mediante simulación basada en la posición de punción actual.

Paso 4: Tomar un valor medio del rendimiento de la secuencia de todas las secuencias, para obtener métricas correspondientes a diferentes posiciones de punción en un caso de la cantidad actual de punciones.

25 Paso 5: Reservar una lista de posiciones de punción correspondientes a métricas óptimas como posiciones de punción candidatas en el caso de la cantidad actual de punciones.

Paso 6: Actualizar el espacio de búsqueda de candidatos de cada lista eliminando la posición de punción actual del espacio de búsqueda.

Paso 7: Aumentar la cantidad de punciones en 1.

30 Paso 8: Determinar si los datos punzonados son inferiores a una cantidad máxima de punciones; y si los datos punzonados son inferiores a la cantidad máxima de punciones, continuar realizando desde el Paso 2 hasta que la cantidad de punciones sea igual a un máximo de datos punzonados; o si los datos punzonados no son inferiores a la cantidad máxima de punciones, emita el primer patrón de intercalación (o puede denominarse patrón de punción
35 óptimo).

El primer patrón de intercalación determinado basándose en el proceso de búsqueda

anterior también puede describirse como sigue: El primer patrón de intercalación puede indicar colocar un valor de un $y^{\text{ésimo}}$ bit de la cuarta secuencia en un $(N-y-1)^{\text{ésimo}}$ bit de la quinta secuencia mediante intercambio, donde $0 \leq y \leq N-k-2$, y es un número entero, y $k+1 \leq N-y-1 \leq N-1$.

Los valores de $k+1$ bits distintos del $y^{\text{ésimo}}$ bit de la cuarta secuencia pueden colocarse aleatoriamente del $0^{\text{ésimo}}$ bit al $k^{\text{ésimo}}$ bit de la quinta secuencia. Esto no está limitado.

El $y^{\text{ésimo}}$ bit se puede determinar basándose en el rendimiento medio de la secuencia obtenido al eliminar un mismo bit de $N-y-1$ bits de la pluralidad de muestras de la cuarta secuencia. Las longitudes de una pluralidad de muestras de secuencias de bits de información correspondientes a la pluralidad de cuartas muestras de secuencias son menores o iguales que K , las longitudes de la pluralidad de muestras de secuencias de bits de información incluyen k , y los $N-y-1$ bits no incluyen los primeros $y-1$ bits del $y^{\text{ésimo}}$ bit.

Opcionalmente, después de eliminar un mismo bit de los $N-y-1$ bits de la pluralidad de muestras de la cuarta secuencia, cuando el rendimiento medio de la secuencia es óptimo, el bit eliminado es el $y^{\text{ésimo}}$ bit.

Por ejemplo, a continuación se utiliza un ejemplo en el que las muestras de la cuarta secuencia incluyen una cuarta secuencia correspondiente a una secuencia de bits de información cuya longitud es $k=k_1$ y una cuarta secuencia correspondiente a una secuencia de bits de información cuya longitud es $k=k_2$ para describir en detalle una manera de generar el primer patrón de intercalación que tiene una característica de anidamiento.

Paso 1: Generar aleatoriamente una primera secuencia aleatoria (es decir, una secuencia de bits de información) cuya longitud sea k_1 y una segunda secuencia aleatoria (es decir, una secuencia de bits de información) cuya longitud sea k_2 , donde k_1 y k_2 son longitudes de secuencias de bits de información por codificar, y k_1 y k_2 no son iguales.

Paso 2: Realizar la codificación polar en la primera secuencia aleatoria y en la segunda secuencia aleatoria por separado de acuerdo con el método mostrado en la FIGURA 7, para obtener una primera secuencia aleatoria de palabras de código (es decir, una cuarta secuencia) y una segunda secuencia aleatoria de palabras de código (es decir, una cuarta secuencia) cuyas longitudes son N .

Paso 3: Determinar un valor de primer rendimiento de una secuencia obtenida por punción de una $g^{\text{ésima}}$ posición de la primera secuencia aleatoria de palabras de código y un valor de primer rendimiento de una secuencia obtenida por punción de una $g^{\text{ésima}}$ posición de la segunda secuencia aleatoria de palabras de código, donde $0 \leq g \leq N-1$.

Paso 4: Determinar un valor medio A_g de un primer valor y un segundo valor, donde el primer valor es el valor del primer rendimiento de la secuencia obtenido puncionando la $g^{\text{ésima}}$ posición de la primera secuencia aleatoria de palabras de código, y el segundo valor es el valor del primer rendimiento de la secuencia obtenido puncionando la $g^{\text{ésima}}$ posición de la segunda

secuencia aleatoria de palabras de código.

Paso 5: Determinar un $y^{\text{ésimo}}$ bit ($y=0$), donde el $y^{\text{ésimo}}$ bit ($y=0$) es una posición correspondiente a un valor mínimo en A_0 a A_{N-1} , y el $y^{\text{ésimo}}$ bit ($y=0$) es una 1^{ésima} posición de punción.

- 5 Paso 6: Determinar que una cantidad 1 de posiciones de punción es igual a $(N-E)$, y generar un primer patrón de intercalación, donde el primer patrón de intercalación indica que después de intercalar la cuarta secuencia basándose en el primer patrón de intercalación, un valor del $y^{\text{ésimo}}$ bit ($y=0$) de la cuarta secuencia se coloca en el último bit de la quinta secuencia mediante intercambio. E representa la longitud de una secuencia de emparejamiento de tasas.

- 10 Alternativamente,

Paso 7: Determinar que una cantidad 1 de posiciones de punción es menor que $(N-E)$, y generar aleatoriamente una tercera secuencia aleatoria (es decir, una secuencia de bits de información) cuya longitud sea k_1 y una cuarta secuencia aleatoria (es decir, una secuencia de bits de información) cuya longitud sea k_2 .

- 15 Paso 8: Realizar la codificación polar en la tercera secuencia aleatoria y la cuarta secuencia aleatoria por separado de acuerdo con el método mostrado en la FIGURA 7, para obtener una tercera secuencia aleatoria de palabras de código (es decir, una cuarta secuencia) y una cuarta secuencia aleatoria de palabras de código (es decir, una cuarta secuencia) cuyas longitudes son N .

- 20 Paso 9: Determinar una quinta secuencia aleatoria de palabras de código y una sexta secuencia aleatoria de palabras de código, donde la quinta secuencia aleatoria de palabras de código es una secuencia obtenida mediante la punción de un bit en el $y^{\text{ésimo}}$ bit ($y=0$) de la tercera secuencia aleatoria de palabras de código, y la sexta secuencia aleatoria de palabras de código es una secuencia obtenida mediante la punción de un bit en el $y^{\text{ésimo}}$ bit ($y=0$) de la cuarta secuencia aleatoria de palabras de código.
- 25

Paso 10: Determinar un valor de primer rendimiento de una secuencia obtenida por punción de una $h^{\text{ésima}}$ posición de la quinta secuencia aleatoria de palabras de código y un valor de primer rendimiento de una secuencia obtenida por punción de una $h^{\text{ésima}}$ posición de la sexta secuencia aleatoria de palabras de código, donde $0 \leq h \leq N-2$.

- 30 Paso 11: Determinar un valor medio A_h de un tercer valor y un cuarto valor, donde el tercer valor es el valor del primer rendimiento de la secuencia obtenida puncionando la $h^{\text{ésima}}$ posición de la quinta secuencia aleatoria de palabras de código, y el cuarto valor es el valor del primer rendimiento de la secuencia obtenida puncionando la $h^{\text{ésima}}$ posición de la sexta secuencia aleatoria de palabras de código.

- 35 Paso 12: Determinar un $y^{\text{ésimo}}$ bit ($y=1$), donde el $y^{\text{ésimo}}$ bit ($y=1$) es una posición correspondiente a un valor mínimo en A_1 a A_{N-2} , y el $y^{\text{ésimo}}$ bit ($y=1$) es una 2^{ésima} posición de

punción.

Paso 13: Determinar que una cantidad 2 de posiciones de punción es igual a $(N-E)$, y generar un primer patrón de intercalación, donde el primer patrón de intercalación indica que después de intercalar la cuarta secuencia basándose en el primer patrón de intercalación, un
 5 valor del $y^{\text{ésimo}}$ bit ($y=0$) de la cuarta secuencia se coloca en el último bit de la quinta secuencia mediante intercambio, y un valor del $y^{\text{ésimo}}$ bit ($y=1$) de la cuarta secuencia se coloca en el penúltimo bit de la quinta secuencia mediante intercambio.

Alternativamente,

Paso 14: Determinar que una cantidad 2 de posiciones de punción es menor que $(N-E)$,
 10 y realizar repetidamente las operaciones anteriores hasta que la cantidad de posiciones de punción sea igual a E , para generar un primer patrón de intercalación correspondiente.

Lo anterior se describe utilizando sólo un ejemplo en el que el primer patrón de intercalación que tiene la característica de anidamiento se genera basándose en la cuarta secuencia correspondiente a k_1 y la cuarta secuencia correspondiente a k_2 (es decir, el
 15 procesamiento de intercalación puede realizarse tanto en la cuarta secuencia correspondiente a k_1 como en la cuarta secuencia correspondiente a k_2 basándose en el primer patrón de intercalación anterior). Puede entenderse que para una pluralidad de cuartas secuencias correspondientes a diferentes k_s , el primer patrón de intercalación que tiene la característica de anidamiento puede generarse de la manera anterior.

Por ejemplo, se utiliza un ejemplo en el que k es igual a 3, 4, ..., u 11. El primer patrón de intercalación que tiene la característica de anidamiento puede generarse de la manera anterior basándose en una cuarta secuencia correspondiente a k que sea 3, 4, ..., u 11. En concreto, el procesamiento de intercalación puede realizarse en la cuarta secuencia correspondiente a k igual a 3, la cuarta secuencia correspondiente a k igual a 4, ..., y la cuarta
 25 secuencia correspondiente a k igual a 11 basándose en el primer patrón de intercalación.

En un ejemplo, el valor máximo de la longitud k de la secuencia de bits de información es 11, el valor mínimo es 3 y N es 32. El primer patrón de intercalación puede indicar colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la quinta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (5, 31), (26, 30), (11, 29), (9, 28), (28, 27),
 30 (7, 26), (30, 25), (16, 24), (31, 23), (21, 22), (10, 21), (24, 20), (27, 19), (6, 18), (3, 17), (12, 16), (17, 15), (8, 14), (13, 13), (0, 12), (2, 11), (1, 10), (4, 9), (14, 8), (25, 7), (23, 6), (22, 5), y (19, 4).

La primera A (es decir, 5) es el $y^{\text{ésimo}}$ bit en un caso de $y=0$, la segunda A (es decir, 26) es el $y^{\text{ésimo}}$ bit en un caso de $y=1$, ..., el $(N-K-1=32-11-1=20)^{\text{ésimo}}$ A (es decir, 0) es el $y^{\text{ésimo}}$ bit en un caso de $y=N-K-2=32-11-2=19$, ..., y el $(N-k-1=32-3-1=28)^{\text{ésimo}}$ A (es decir, 19) es el $y^{\text{ésimo}}$ bit
 35 en un caso de $y=N-k-2=32-11-2=27$.

El primer patrón de intercalación puede representarse como un patrón de intercalación

mostrado en la tabla 6, o puede representarse como un patrón de intercalación mostrado en la FIGURA 11.

Tabla 6

B	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A
0	29	4	19	8	14	12	0	16	12	20	24	24	16	28	9
1	20	5	22	9	4	13	13	17	3	21	10	25	30	29	11
2	18	6	23	10	1	14	8	18	6	22	21	26	7	30	26
3	15	7	25	11	2	15	17	19	27	23	31	27	28	31	5

Puede entenderse que los cuatro primeros grupos de relaciones de mapeo entre A y B en la tabla 6 pueden determinarse aleatoriamente, es decir, 29, 20, 18 y 15 en A pueden corresponder aleatoriamente a 0, 1, 2 y 3 en B. Esto no es limitativo.

En otro ejemplo, el valor máximo de la longitud k de la secuencia de bits de información es 11, el valor mínimo es 3 y N es 32. El primer patrón de intercalación puede indicar colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la quinta secuencia mediante intercambio; y una relación de mapeo entre A y B es: (4, 31), (22, 30), (27, 29), (17, 28), (8, 27), (2, 26), (15, 25), (21, 24), (29, 23), (23, 22), (6, 21), (26, 20), (20, 19), (7, 18), (0, 17), (13, 16), (19, 15), (11, 14), (14, 13), (16, 12), (25, 11), (3, 10), (10, 9), (5, 8), (12, 7), (1, 6), (18, 5), y (24, 4).

La primera A (es decir, 4) es el $y^{\text{ésimo}}$ bit en un caso de $y=0$, la segunda A (es decir, 22) es el $y^{\text{ésimo}}$ bit en un caso de $y=1$, ..., el $(N-K-1=32-11-1=20)^{\text{ésimo}}$ A (es decir, 16) es el $y^{\text{ésimo}}$ bit en un caso de $y=N-K-2=32-11-2=19$, ..., y el $(N-k-1=32-3-1=28)^{\text{ésimo}}$ A (es decir, 24) es el $y^{\text{ésimo}}$ bit en un caso de $y=N-k-2=32-11-2=27$.

El primer patrón de intercalación puede ser un patrón de intercalación mostrado en la tabla 7, o puede representarse como un patrón de intercalación mostrado en la FIGURA 12.

Tabla 7

B	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A
0	9	4	24	8	5	12	16	16	13	20	26	24	21	28	17
1	28	5	18	9	10	13	14	17	0	21	6	25	15	29	27
2	30	6	1	10	3	14	11	18	7	22	23	26	2	30	22
3	31	7	12	11	25	15	19	19	20	23	29	27	8	31	4

Puede entenderse que los cuatro primeros grupos de relaciones de mapeo entre A y B en la tabla 7 pueden determinarse aleatoriamente, es decir, 9, 28, 30 y 31 en A pueden corresponder aleatoriamente a 0, 1, 2 y 3 en B. Esto no es limitativo.

Basándose en el primer patrón de intercalación anterior, debido a que el primer patrón de intercalación tiene la característica de anidamiento, no es necesario almacenar por separado un patrón de intercalación para secuencias de diferentes cantidades de bits de información, reduciendo así la complejidad del hardware y reduciendo las sobrecargas del hardware. Dado

que el primer patrón de intercalación se determina en función del rendimiento de la secuencia, se puede garantizar que no haya ningún punto malo en el rendimiento del emparejamiento de tasas (por ejemplo, no hay ningún punto malo en el rendimiento del emparejamiento de tasas de grano fino de 1 bit), y se mejora el rendimiento de la decodificación.

5 Puede entenderse que los patrones de intercalación pueden almacenarse alternativamente por separado para las secuencias de diferentes cantidades de bits de información, es decir, los primeros patrones de intercalación se almacenan por separado para las secuencias de diferentes cantidades de bits de información. Esto no está limitado.

10 En un segundo diseño posible, el procesamiento de intercalación puede realizarse alternativamente en la cuarta secuencia basándose en el segundo patrón de intercalación, para obtener una sexta secuencia.

15 A diferencia del primer patrón de intercalación que tiene la característica de anidamiento en el primer diseño posible, en el segundo diseño posible, las secuencias de diferentes cantidades de bits de información pueden corresponder a diferentes segundos patrones de intercalación, es decir, los segundos patrones de intercalación se diseñan por separado para las secuencias de diferentes cantidades de bits de información.

En esta modalidad de esta solicitud, el segundo patrón de intercalación puede diseñarse con referencia al algoritmo de búsqueda en el primer diseño posible. Esto puede reducir enormemente la complejidad al tiempo que garantiza un rendimiento óptimo de la secuencia.

20 Por ejemplo, la búsqueda de la posición de punción puede realizarse en una pluralidad de muestras de la cuarta secuencia (la pluralidad de muestras de la cuarta secuencia puede corresponder a secuencias de bits de información de una misma longitud, y una longitud de la secuencia de bits de información es menor o igual que K) haciendo referencia al algoritmo de búsqueda de listas. Cada vez que una cantidad de punciones aumenta en 1, se calcula el **25** rendimiento medio de la secuencia de todas las secuencias, se utiliza el rendimiento medio de la secuencia como métrica (Metric), y se determina una posición correspondiente al rendimiento medio óptimo de la secuencia como posición de punción, para generar el segundo patrón de intercalación. El rendimiento medio es un valor medio de $\text{SNR@BLER}=1\text{E-}2$ de las secuencias.

30 El segundo patrón de intercalación determinado basándose en el proceso de búsqueda anterior también puede describirse como sigue: El segundo patrón de intercalación indica colocar un valor de un $y^{\text{ésimo}}$ bit de la cuarta secuencia en un $(N-y-1)^{\text{ésimo}}$ bit de la sexta secuencia mediante intercambio, donde $1 \leq y \leq N-k-2$, y es un entero positivo, y $k+1 \leq N-y-1 \leq N-1$.

Los valores de $k+1$ bits distintos del $y^{\text{ésimo}}$ bit de la cuarta secuencia pueden colocarse aleatoriamente del $0^{\text{ésimo}}$ bit al $k^{\text{ésimo}}$ bit de la sexta secuencia. Esto no está limitado.

35 El $y^{\text{ésimo}}$ bit se determina basándose en el rendimiento medio de la secuencia obtenido al eliminar 1 bit de $N-y-1$ bits de la pluralidad de muestras de la cuarta secuencia. Las longitudes

de una pluralidad de muestras de secuencias de bits de información correspondientes a la pluralidad de cuartas muestras de secuencias son todas k , y los $N-y-1$ bits no incluyen los primeros $y-1$ bits del $y^{\text{ésimo}}$ bit.

- 5 Opcionalmente, después de eliminar 1 bit de los $N-y-1$ bits de las muestras de la cuarta secuencia, cuando el rendimiento medio de la secuencia es óptimo, el bit eliminado es el $y^{\text{ésimo}}$ bit.

En un ejemplo, el valor máximo de la longitud k de la secuencia de bits de información es 11, el valor mínimo es 3 y N es 32. El segundo patrón de intercalación puede indicar colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la sexta secuencia mediante
10 intercambio.

Cuando k es 3, una relación de mapeo entre A y B es: (12, 31), (9, 30), (19, 29), (18, 28), (1, 27), (15, 26), (14, 25), (29, 24), (16, 23), (10, 22), (28, 21), (23, 20), (20, 19), (6, 18), (21, 17), (7, 16), (2, 15), (25, 14), (0, 13), (3, 12), (5, 11), (31, 10), (30, 9), (4, 8), (22, 7), (13, 6), (27, 5), y (8, 4).

- 15 Alternativamente, cuando k es 4, una relación de mapeo entre A y B es: (18, 31), (29, 30), (9, 29), (12, 28), (27, 27), (22, 26), (23, 25), (26, 24), (4, 23), (0, 22), (7, 21), (14, 20), (8, 19), (19, 18), (1, 17), (21, 16), (13, 15), (11, 14), (17, 13), (20, 12), (16, 11), (15, 10), (6, 9), (10, 8), (28, 7), (25, 6), y (24, 5).

- 20 Alternativamente, cuando k es 5, una relación de mapeo entre A y B es: (2, 31), (20, 30), (12, 29), (25, 28), (8, 27), (22, 26), (16, 25), (5, 24), (29, 23), (14, 22), (23, 21), (15, 20), (3, 19), (11, 18), (10, 17), (17, 16), (0, 15), (30, 14), (18, 13), (27, 12), (4, 11), (19, 10), (21, 9), (6, 8), (26, 7), y (9, 6).

- 25 Alternativamente, cuando k es 6, una relación de mapeo entre A y B es: (19, 31), (23, 30), (16, 29), (18, 28), (29, 27), (0, 26), (21, 25), (8, 24), (11, 23), (4, 22), (13, 21), (2, 20), (20, 19), (24, 18), (25, 17), (6, 16), (28, 15), (14, 14), (10, 13), (9, 12), (31, 11), (26, 10), (7, 9), (17, 8), y (1, 7).

- 30 Alternativamente, cuando k es 7, una relación de mapeo entre A y B es: (7, 31), (9, 30), (19, 29), (26, 28), (31, 27), (21, 26), (27, 25), (2, 24), (23, 23), (17, 22), (10, 21), (30, 20), (25, 19), (24, 18), (0, 17), (16, 16), (4, 15), (11, 14), (20, 13), (12, 12), (29, 11), (18, 10), (13, 9), y (3, 8).

Alternativamente, cuando k es 8, una relación de mapeo entre A y B es: (15, 31), (16, 30), (9, 29), (14, 28), (17, 27), (24, 26), (21, 25), (20, 24), (5, 23), (1, 22), (7, 21), (30, 20), (11, 19), (6, 18), (2, 17), (3, 16), (8, 15), (22, 14), (12, 13), (23, 12), (19, 11), (10, 10), y (29, 9).

- 35 Alternativamente, cuando k es 9, una relación de mapeo entre A y B es: (17, 31), (21, 30), (15, 29), (14, 28), (4, 27), (24, 26), (8, 25), (29, 24), (19, 23), (16, 22), (7, 21), (2, 20), (9, 19), (25, 18), (26, 17), (23, 16), (28, 15), (30, 14), (13, 13), (6, 12), (22, 11), y (0, 10).

Alternativamente, cuando k es 10, una relación de mapeo entre A y B es: (28, 31), (29, 30), (14, 29), (26, 28), (8, 27), (23, 26), (24, 25), (30, 24), (0, 23), (31, 22), (10, 21), (5, 20), (20, 19), (4, 18), (16, 17), (12, 16), (9, 15), (21, 14), (1, 13), (11, 12), y (22, 11).

Alternativamente, cuando k es 11, una relación de mapeo entre A y B es: (22, 31), (19, 30), (28, 29), (9, 28), (4, 27), (6, 26), (14, 25), (11, 24), (3, 23), (13, 22), (0, 21), (15, 20), (26, 19), (24, 18), (27, 17), (18, 16), (21, 15), (17, 14), (20, 13), y (12, 12).

El primer A es el $y^{\text{ésimo}}$ bit en un caso de $y=0$, el segundo A es el $y^{\text{ésimo}}$ bit en un caso de $y=1$, ..., el $(N-K-1=32-11-1=20)^{\text{ésimo}}$ A (es decir, 0) es el $y^{\text{ésimo}}$ bit en un caso de $y=N-K-2=32-11-2=19$, ..., y el $(N-k-1)^{\text{ésimo}}$ A es el $y^{\text{ésimo}}$ bit en un caso de $y=N-k-2$.

Opcionalmente, el segundo patrón de intercalación puede representarse alternativamente como un patrón de intercalación mostrado en la FIGURA 13.

Opcionalmente, el segundo patrón de intercalación puede representarse alternativamente como un patrón de intercalación similar al mostrado en la Tabla 6 o la Tabla 7. Por ejemplo, k es 3, y el segundo patrón de intercalación puede ser un patrón de intercalación mostrado en la Tabla 8 a continuación.

Tabla 8

B	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A
0	25	4	8	8	4	12	3	16	7	20	23	24	29	28	18
1	16	5	27	9	30	13	0	17	21	21	28	25	14	29	19
2	23	6	13	10	31	14	25	18	6	22	10	26	15	30	9
3	10	7	22	11	5	15	2	19	20	23	16	27	1	31	12

Puede entenderse que los cuatro primeros grupos de relaciones de mapeo entre A y B en la tabla 8 pueden determinarse aleatoriamente, es decir, 25, 16, 23 y 10 en A pueden corresponder aleatoriamente a 0, 1, 2 y 3 en B . Esto no es limitativo.

En otro ejemplo, el valor máximo de la longitud k de la secuencia de bits de información es 11, el valor mínimo es 3 y N es 32. El segundo patrón de intercalación puede indicar colocar un valor de un $A^{\text{ésimo}}$ bit de la cuarta secuencia en un $B^{\text{ésimo}}$ bit de la sexta secuencia mediante intercambio.

Cuando k es 3, una relación de mapeo entre A y B es: (5, 31), (11, 30), (24, 29), (30, 28), (13, 27), (20, 26), (22, 25), (3, 24), (4, 23), (25, 22), (31, 21), (6, 20), (10, 19), (8, 18), (21, 17), (15, 16), (9, 15), (2, 14), (0, 13), (27, 12), (14, 11), (16, 10), (17, 9), (23, 8), (28, 7), (26, 6), (1, 5), y (7, 4).

Cuando k es 4, una relación de mapeo entre A y B es: (16, 31), (28, 30), (10, 29), (30, 28), (19, 27), (7, 26), (13, 25), (9, 24), (11, 23), (6, 22), (8, 21), (26, 20), (21, 19), (20, 18), (23, 17), (25, 16), (0, 15), (17, 14), (27, 13), (22, 12), (18, 11), (15, 10), (5, 9), (12, 8), (2, 7), (29, 6), y (24, 5).

Cuando k es 5, una relación de mapeo entre A y B es: (9, 31), (19, 30), (11, 29), (12, 28), (21, 27), (0, 26), (29, 25), (2, 24), (4, 23), (31, 22), (10, 21), (1, 20), (22, 19), (7, 18), (24, 17), (30, 16), (20, 15), (3, 14), (28, 13), (13, 12), (14, 11), (8, 10), (6, 9), (5, 8), (17, 7), y (18, 6).

Cuando k es 6, una relación de mapeo entre A y B es: (11, 31), (25, 30), (31, 29), (27, 28), (14, 27), (0, 26), (22, 25), (19, 24), (18, 23), (3, 22), (20, 21), (2, 20), (4, 19), (17, 18), (23, 17), (24, 16), (12, 15), (1, 14), (7, 13), (10, 12), (28, 11), (26, 10), (6, 9), (5, 8), y (8, 7).

Cuando k es 7, una relación de mapeo entre A y B es: (7, 31), (12, 30), (2, 29), (5, 28), (26, 27), (6, 26), (27, 25), (24, 24), (4, 23), (19, 22), (1, 21), (29, 20), (9, 19), (8, 18), (13, 17), (11, 16), (28, 15), (10, 14), (25, 13), (16, 12), (30, 11), (0, 10), (21, 9), y (3, 8).

Cuando k es 8, una relación de mapeo entre A y B es: (7, 31), (30, 30), (17, 29), (16, 28), (19, 27), (6, 26), (29, 25), (18, 24), (10, 23), (3, 22), (24, 21), (9, 20), (21, 19), (28, 18), (31, 17), (12, 16), (23, 15), (2, 14), (15, 13), (4, 12), (27, 11), (20, 10), y (25, 9).

Cuando k es 9, una relación de mapeo entre A y B es: (28, 31), (25, 30), (17, 29), (6, 28), (11, 27), (21, 26), (27, 25), (30, 24), (3, 23), (1, 22), (4, 21), (8, 20), (15, 19), (26, 18), (29, 17), (18, 16), (9, 15), (19, 14), (2, 13), (20, 12), (23, 11), y (10, 10).

Cuando k es 10, una relación de mapeo entre A y B es: (23, 31), (30, 30), (3, 29), (28, 28), (14, 27), (19, 26), (9, 25), (16, 24), (29, 23), (13, 22), (26, 21), (12, 20), (18, 19), (0, 18), (15, 17), (4, 16), (20, 15), (1, 14), (25, 13), (31, 12), y (21, 11).

Cuando k es 11, una relación de mapeo entre A y B es: (1, 31), (24, 30), (14, 29), (15, 28), (25, 27), (12, 26), (26, 25), (13, 24), (27, 23), (31, 22), (8, 21), (5, 20), (20, 19), (21, 18), (3, 17), (28, 16), (17, 15), (6, 14), (19, 13), y (11, 12).

El primer A es el $y^{\text{ésimo}}$ bit en un caso de $y=0$, el segundo A es el $y^{\text{ésimo}}$ bit en un caso de $y=1$, ..., el $(N-K-1=32-11-1=20)^{\text{ésimo}}$ A (es decir, 0) es el $y^{\text{ésimo}}$ bit en un caso de $y=N-K-2=32-11-2=19$, ..., y el $(N-k-1)^{\text{ésimo}}$ A es el $y^{\text{ésimo}}$ bit en un caso de $y=N-k-2$.

Opcionalmente, el segundo patrón de intercalación puede representarse alternativamente como un patrón de intercalación mostrado en la FIGURA 14.

Opcionalmente, el segundo patrón de intercalación puede representarse alternativamente como un patrón de intercalación similar al mostrado en la Tabla 6, la Tabla 7 o la Tabla 8. Por ejemplo, k es 3, y el segundo patrón de intercalación puede ser un patrón de intercalación mostrado en la Tabla 9 a continuación.

Tabla 9

B	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A
0	12	4	7	8	23	12	27	16	15	20	6	24	3	28	30
1	18	5	1	9	17	13	0	17	21	21	31	25	22	29	24
2	19	6	26	10	16	14	2	18	8	22	25	26	20	30	11
3	29	7	28	11	14	15	9	19	10	23	4	27	13	31	5

Puede entenderse que los cuatro primeros grupos de relaciones de mapeo entre A y B en la tabla 9 pueden determinarse aleatoriamente, es decir, 12, 18, 19 y 29 en A pueden corresponder aleatoriamente a 0, 1, 2 y 3 en B. Esto no es limitativo.

En los dos diseños posibles anteriores, se definen varios ejemplos a partir de 0 bits.

- 5** Puede entenderse que la definición puede comenzar alternativamente a partir de 1 bit, es decir, 0, 1, ..., y 31 descritos anteriormente pueden sustituirse respectivamente por 1, 2, ..., y 32. Esto no está limitado.

- 10** Basándose en la descripción anterior, después de realizar el procesamiento de intercalación en la cuarta secuencia, el dispositivo de extremo de transmisión puede realizar además el emparejamiento de tasas en la cuarta secuencia (por ejemplo, la quinta secuencia o la sexta secuencia) obtenida mediante el procesamiento de intercalación.

Una longitud de la quinta secuencia y una longitud de la sexta secuencia son ambas N.

- 15** Puede determinarse una longitud de código de transmisión E (es decir, una longitud de la secuencia de emparejamiento de tasas) basándose en la información relacionada con el emparejamiento de tasas. Cuando se determina que la longitud del código de transmisión E no es igual a N, se puede utilizar la siguiente manera de emparejamiento de tasas: Cuando la longitud del código de transmisión E es menor que N, la punción se realiza de atrás hacia delante; o cuando la longitud del código de transmisión E es mayor que N, la repetición se realiza de delante hacia atrás.

- 20** Es decir, el dispositivo de extremo de transmisión puede realizar la emparejamiento de tasas en la quinta secuencia cuya longitud es N, para obtener una primera secuencia de emparejamiento de tasas cuya longitud es E. Alternativamente, el emparejamiento de tasas se realiza en la sexta secuencia cuya longitud es N, para obtener una segunda secuencia de emparejamiento de tasas cuya longitud es E.

- 25** Por ejemplo, la cuarta secuencia es c_0^{N-1} . El procesamiento intercalación puede realizarse sobre la cuarta secuencia basándose en un patrón de intercalación (por ejemplo, el primer patrón de intercalación o el segundo patrón de intercalación) con referencia a la siguiente descripción, para obtener una secuencia e_0^{N-1} obtenida mediante el procesamiento de intercalación:

- 30** para $n=0$ a $N-1$

$$e_n = d_{P(n)}$$

fin para

e_n representa un $n^{\text{ésimo}}$ valor en una secuencia obtenida mediante un procesamiento de intercalación, y $P(n)$ representa un patrón de intercalación.

- 35** El emparejamiento de tasas se realiza en la secuencia e_0^{N-1} obtenida mediante el

procesamiento de intercalación, para obtener la secuencia de emparejamiento de tasas f_0^{E-1} con referencia a la siguiente descripción:

para $j=0$ a $E-1$

$$f_j = e_{j \bmod N}$$

5 fin para

En un ejemplo, la longitud k de la secuencia de bits de información correspondiente a la cuarta secuencia cuya longitud es 32 es 7, y el dispositivo de extremo de transmisión realiza el procesamiento de intercalación basándose en el segundo patrón de intercalación mostrado en la FIGURA 13. Se supone que la longitud E de la secuencia de emparejamiento de tasas es 22, los valores de 7^{ésimo}, 9^{ésimo}, 19^{ésimo}, 26^{ésimo}, 31^{ésimo}, 21^{ésimo}, 27^{ésimo}, 2^{ésimo}, 23^{ésimo}, y 17^{ésimo} bits de la cuarta secuencia necesitan ser colocados secuencialmente en el 31^{ésimo} bit al 22^{ésimo} bit de la sexta secuencia, los valores de los 22 bits restantes de la cuarta secuencia se colocan aleatoriamente en el 0^{ésimo} bit hasta el 21^{ésimo} bit de la sexta secuencia, y los valores del 22^{ésimo} bit hasta el 31^{ésimo} bit de la sexta secuencia se puncionan, para obtener la segunda secuencia de emparejamiento de tasas. Alternativamente, puede describirse como sigue: Los valores de 7^{ésimo}, 9^{ésimo}, 19^{ésimo}, 26^{ésimo}, 31^{ésimo}, 21^{ésimo}, 27^{ésimo}, 2^{ésimo}, 23^{ésimo}, y 17^{ésimo} bits de la cuarta secuencia se puncionan, y los valores de los 22 bits restantes se ordenan aleatoriamente, para obtener la segunda secuencia de emparejamiento de tasas.

En otro ejemplo, la longitud k de la secuencia de bits de información correspondiente a la cuarta secuencia cuya longitud es 32 es 3, y el dispositivo de extremo de transmisión realiza el procesamiento de intercalación basándose en el segundo patrón de intercalación mostrado en la FIGURA 13. Se supone que la longitud E de la secuencia de emparejamiento de tasas es 4, valores de 12^{ésimo}, 9^{ésimo}, 19^{ésimo}, 18^{ésimo}, 1^{ésimo}, 15^{ésimo}, 14^{ésimo}, 29^{ésimo}, 16^{ésimo}, 10^{ésimo}, 28^{ésimo}, 23^{ésimo}, 20^{ésimo}, 6^{ésimo}, 21^{ésimo}, 7^{ésimo}, 2^{ésimo}, 25^{ésimo}, 0^{ésimo}, 3^{ésimo}, 5^{ésimo}, 31^{ésimo}, 30^{ésimo}, 4^{ésimo}, 22^{ésimo}, 13^{ésimo}, 27^{ésimo}, y 8^{ésimo} bits de la cuarta secuencia deben colocarse secuencialmente del 31^{ésimo} bit al 4^{ésimo} bit de la sexta secuencia, los valores de los 4 bits restantes de la cuarta secuencia se colocan aleatoriamente en el 0^{ésimo} bit al 3^{ésimo} bit de la sexta secuencia, y se puncionan los valores del 4^{ésimo} bit al 31^{ésimo} bit de la sexta secuencia, para obtener la segunda secuencia de emparejamiento de tasas. Alternativamente, puede describirse como sigue: Valores de 12^{ésimo}, 9^{ésimo}, 19^{ésimo}, 18^{ésimo}, 1^{ésimo}, 15^{ésimo}, 14^{ésimo}, 29^{ésimo}, 16^{ésimo}, 10^{ésimo}, 28^{ésimo}, 23^{ésimo}, 20^{ésimo}, 6^{ésimo}, 21^{ésimo}, 7^{ésimo}, 2^{ésimo}, 25^{ésimo}, 0^{ésimo}, 3^{ésimo}, 5^{ésimo}, 31^{ésimo}, 30^{ésimo}, 4^{ésimo}, 22^{ésimo}, 13^{ésimo}, 27^{ésimo}, y 8^{ésimo} bits de la cuarta secuencia se puncionan, y los valores de los 4 bits restantes se ordenan aleatoriamente, para obtener la segunda secuencia de emparejamiento de tasas.

Basándose en la descripción anterior, puede comprobarse que un valor mayor de k

indica una menor cantidad de punciones admitidas, y puede admitirse un máximo de $N-1-k$ posiciones a puncionar.

Basándose en las descripciones anteriores de la codificación polar, la intercalación y el emparejamiento de tasas, la FIGURA 15 a la FIGURA 19 son, respectivamente, diagramas de comparación del rendimiento entre los efectos de simulación de un código LTE-RM y un código polar proporcionado en las modalidades de esta solicitud en diferentes longitudes de código y tasas de código.

Un canal de simulación es un canal AWGN, un esquema de modulación es la modulación QPSK, un esquema de codificación es el código LTE-RM y un esquema de decodificación es la decodificación FHT; o un esquema de codificación es el código polar y un esquema de decodificación es la decodificación SCL8.

Cada diagrama de comparación de rendimiento tiene dos curvas en total. La curva 1 es el rendimiento obtenido mediante el emparejamiento de tasas realizado en una manera de código LTE-RM, y la curva 2 es el rendimiento obtenido mediante el emparejamiento de tasas realizado en una manera de código polar (es decir, de acuerdo con el método mostrado en la FIGURA 7 y basado en el primer patrón de intercalación) proporcionado en esta modalidad de esta solicitud.

Un eje horizontal en el diagrama de comparación de rendimiento es una longitud de código E tras la punción, y un valor del eje horizontal va de $k+1$ a N (por ejemplo, 32), y un eje vertical es una SNR necesaria para alcanzar $BLER=0.01$. Una curva más baja indica un mejor rendimiento.

Se puede aprender de la FIGURA 15 a la FIGURA 17 que el método de codificación del código polar proporcionado en las modalidades de esta solicitud puede mejorar en gran medida el rendimiento de la codificación y la decodificación.

Cabe señalar que, en el método anterior, se utiliza como ejemplo para su descripción un código muy corto (very short codes) cuya longitud de una secuencia de bits de información es de 3 a 11. Puede entenderse que la longitud del código muy corto no se limita de 3 a 11, o puede tener otra longitud. La secuencia de bits de información puede ser alternativamente un código polar pretransformado (pre-transform polar), un código convolucional ajustado a la polarización (polarization adjusted convolutional, PAC), o similares. Esto no está limitado.

Debe tenerse en cuenta que las modalidades de esta solicitud pueden implementarse por separado, o pueden implementarse en combinación. Esto no está limitado. A menos que se indique lo contrario o exista un conflicto lógico, los términos y/o descripciones de las diferentes modalidades proporcionadas por esta solicitud son coherentes y pueden referenciarse mutuamente, y las características técnicas de las diferentes modalidades pueden combinarse en una nueva modalidad basada en una relación lógica interna de las mismas.

Puede entenderse que, en las modalidades de esta solicitud, un órgano de ejecución puede llevar a cabo una parte o la totalidad de los pasos de las modalidades de esta solicitud. Los pasos u operaciones son meros ejemplos. En algunas modalidades de esta solicitud, pueden llevarse a cabo otras operaciones o variaciones de diversas operaciones. Además, los

5 pasos pueden realizarse en una secuencia diferente a la presentada en las modalidades de esta solicitud, y no todas las operaciones de las modalidades de esta solicitud pueden llevarse a cabo necesariamente.

Las soluciones aportadas en las modalidades de esta solicitud se describen principalmente desde la perspectiva de la interacción entre los dispositivos. Puede entenderse

10 que, para implementar las funciones anteriores, cada dispositivo incluye una estructura de hardware correspondiente y/o un módulo de software correspondiente para realizar cada función. Una persona con conocimientos ordinarios en la materia debería ser fácilmente consciente de que, en combinación con los algoritmos y los pasos de los ejemplos descritos en las modalidades divulgadas en esta especificación, esta solicitud puede implementarse

15 mediante hardware o una combinación de hardware y software de computadora. Si una función se realiza por hardware o hardware accionado por software de computadora depende de las aplicaciones particulares y restricciones de diseño de las soluciones técnicas. Una persona experta en la técnica puede usar diferentes métodos para implementar las funciones descritas para cada aplicación particular, pero no es necesario considerar que la implementación va más

20 allá del alcance de esta solicitud.

En modalidades de la presente solicitud, los dispositivos pueden dividirse en módulos funcionales con base en los ejemplos de métodos anteriores. Por ejemplo, cada módulo funcional puede obtenerse mediante división basada en cada función correspondiente, o pueden integrarse dos o más funciones en un módulo de procesamiento. El módulo integrado

25 se puede implementar en forma de hardware, o se puede implementar en forma de un módulo funcional de software. Se debe observar que, en las modalidades de la presente solicitud, la división en módulos es un ejemplo y es meramente una división de funciones lógicas. En la implementación real, se puede utilizar otra forma de división.

Cuando cada módulo funcional se obtiene mediante la división basada en cada función correspondiente, FIGURA 20 muestra un aparato de comunicación 200. El aparato de comunicación 200 puede llevar a cabo acciones realizadas por el dispositivo de extremo de transmisión en los métodos mostrados en la FIGURA 7 a la FIGURA 19. Todo el contenido relacionado de los pasos de las modalidades de métodos anteriores puede citarse en las descripciones funcionales de los módulos funcionales correspondientes. Para conocer los

30 efectos técnicos que puede lograr el aparato de comunicación 200, consultar las modalidades de métodos anteriores. Los detalles no se describen de nuevo aquí.

35

El aparato de comunicación 200 puede incluir un módulo de transceptor 2001 y un módulo de procesamiento 2002. Por ejemplo, el aparato de comunicación 200 puede ser un dispositivo de comunicación, o puede ser un chip utilizado en el dispositivo de comunicación, u otro dispositivo o componente combinado que tenga una función del aparato de comunicación.

- 5** Cuando el aparato de comunicación 200 es un dispositivo de comunicación, el módulo transceptor 2001 puede ser un transceptor, donde el transceptor puede incluir una antena, un circuito de radiofrecuencia y similares; y el módulo de procesamiento 2002 puede ser un procesador (o un circuito de procesamiento), por ejemplo, un procesador de banda base, donde el procesador de banda base puede incluir una o más CPU. Cuando el aparato de comunicación
- 10** 200 es un componente que tiene una función del aparato de comunicación, el módulo transceptor 2001 puede ser una unidad de radiofrecuencia, y el módulo de procesamiento 2002 puede ser un procesador (o un circuito de procesamiento), por ejemplo, un procesador de banda base. Cuando el aparato de comunicación 200 es un sistema de chip, el módulo transceptor 2001 puede ser una interfaz de entrada/salida de un chip (por ejemplo, un chip de banda base);
- 15** y el módulo de procesamiento 2002 puede ser un procesador (o un circuito de procesamiento) del sistema de chip, y puede incluir una o más unidades centrales de procesamiento. Debe entenderse que el módulo transceptor 2001 en las modalidades de esta solicitud puede implementarse mediante un transceptor o un componente de circuito relacionado con el transceptor, y el módulo de procesamiento 2002 puede implementarse mediante un procesador
- 20** o un componente de circuito relacionado con el procesador (o denominado circuito de procesamiento).

- Por ejemplo, el módulo transceptor 2001 puede estar configurado para realizar todas las operaciones de recepción y envío realizadas por el aparato de comunicación en las modalidades mostradas en la FIGURA 7 a la FIGURA 19, y/o configurado para soportar otro proceso de la
- 25** tecnología descrita en esta especificación. El módulo de procesamiento 2002 puede estar configurado para realizar todas las operaciones, distintas de las de recepción y envío, realizadas por el aparato de comunicación en las modalidades mostradas en la FIGURA 7 a la FIGURA 19, y/o configurado para soportar otro proceso de la tecnología descrita en esta especificación.

- En otra posible implementación, el módulo transceptor 2001 de la FIGURA 20 puede
- 30** sustituirse por un transceptor, y una función del módulo transceptor 2001 puede integrarse en el transceptor. El módulo de procesamiento 2002 puede sustituirse por un procesador, y una función del módulo de procesamiento 2002 puede integrarse en el procesador. Además, el aparato de comunicación 200 mostrado en la FIGURA 20 puede incluir además una memoria.

- Alternativamente, cuando el módulo de procesamiento 2002 se sustituye por un
- 35** procesador, y el módulo transceptor 2001 se sustituye por un transceptor, el aparato de comunicación 200 en las modalidades de esta solicitud puede ser un aparato de comunicación

210 mostrado en la FIGURA 21. El procesador puede ser un circuito lógico 2101, y el transceptor puede ser un circuito de interfaz 2102. Además, el aparato de comunicación 210 mostrado en la FIGURA 21 puede incluir además una memoria 2103.

Una modalidad de esta solicitud proporciona además un producto de programa de computadora. Cuando el producto de programa de computadora es ejecutado por una computadora, puede implementarse una función de cualquiera de las modalidades de método anteriores.

Una modalidad de esta solicitud proporciona además un programa de computadora. Cuando el programa de computadora es ejecutado por una computadora, puede implementarse una función de cualquiera de las modalidades de métodos anteriores.

Una modalidad de esta solicitud proporciona además un medio de almacenamiento legible por computadora. Todos o algunos de los procedimientos de las modalidades de métodos anteriores pueden implementarse mediante un programa de computadora que instruya al hardware relacionado. El programa puede almacenarse en el medio de almacenamiento legible por computadora anterior. Cuando se ejecuta el programa, se pueden realizar los procedimientos de las modalidades de métodos anteriores. El medio de almacenamiento legible por computadora puede ser una unidad de almacenamiento interna de la terminal en cualquiera de las modalidades anteriores (incluyendo un extremo de transmisión de datos y/o un extremo de recepción de datos), por ejemplo, un disco duro o la memoria de la terminal. El anterior medio de almacenamiento legible por computadora puede ser alternativamente un dispositivo de almacenamiento externo de la terminal anterior, por ejemplo, un disco duro conectable, una tarjeta de memoria inteligente (smart media card, SMC), una tarjeta digital segura (secure digital, SD) o una tarjeta flash (flash card) configurada en la terminal anterior. Además, el medio de almacenamiento legible por computadora puede incluir tanto una unidad de almacenamiento interna como un dispositivo de almacenamiento externo de la terminal anterior. El medio de almacenamiento legible por computadora está configurado para almacenar el programa de computadora y otros programas y datos requeridos por la terminal anterior. El medio de almacenamiento legible por computadora puede estar configurado además para almacenar temporalmente los datos que se han emitido o se van a emitir.

Debe tenerse en cuenta que en la especificación, las reivindicaciones y los dibujos adjuntos de esta solicitud, los términos "primero", "segundo" pretenden distinguir entre objetos diferentes pero no indican una secuencia particular. "Primero" y "segundo" solamente se contemplan para descripción y no deben entenderse como una indicación o implicación de importancia relativa o indicación implícita de una cantidad de atributos técnicos indicados. Por lo tanto, un atributo limitante por "primero" o "segundo" puede incluir explícita o implícitamente uno o más atributos. En las descripciones de las modalidades, a menos que se especifique lo

contrario, "una pluralidad de" significa dos o más.

Adicionalmente, los términos "incluyendo" y "teniendo" y cualesquiera otras variantes de los mismos pretenden abarcar la inclusión no exclusiva. Por ejemplo, un proceso, un método, un sistema, un producto o un dispositivo que incluya una serie de pasos o unidades no se limita a los pasos o unidades enumerados, sino que opcionalmente además incluye un paso o unidad no enumerado, u opcionalmente además incluye otro paso o unidad inherente al proceso, el método, el producto o el dispositivo.

Debe entenderse que en esta solicitud, "al menos un (elemento)" significa uno o más. "Una pluralidad de" significa dos o más de dos. "Al menos dos (elementos)" significa dos, tres o más de tres. "Y/o" se utiliza para describir una relación de asociación entre objetos asociados e indica que pueden existir tres relaciones. Por ejemplo, "A y/o B" pueden representar los siguientes tres casos: Sólo existe A, sólo existe B y existen tanto A como B, donde A y B pueden ser singulares o plurales. El carácter "/" en general indica una relación "o" entre los objetos asociados. "Al menos uno de los siguientes elementos (piezas)" o una expresión similar de los mismos significa cualquier combinación de estos elementos, incluyendo cualquier combinación de elementos individuales (pieza) o pluralidad de elementos (piezas). Por ejemplo, al menos un elemento (pieza) de a, b o c puede indicar: a, b, c, a y b, a y c, b y c, o a, b y c, donde a, b y c puede ser singular o plural. Tanto "cuando..." como "si" significan que el procesamiento correspondiente se realiza en un caso objetivo, no pretenden limitar el tiempo, no requieren una acción determinante durante la implementación y no significan que exista otra limitación.

Además, en modalidades de esta solicitud, la palabra "ejemplo" o "por ejemplo" se usa para dar un ejemplo, una ilustración o una descripción. Cualquier modalidad o esquema de diseño descrito como "ejemplo" o "por ejemplo" en las modalidades de esta solicitud no debe explicarse como más preferida o con más ventajas que otra modalidad o esquema de diseño. Exactamente, el uso de términos como "ejemplo" o "por ejemplo" pretende presentar un concepto relacionado de forma específica para facilitar su comprensión.

En esta solicitud, "enviar información a ... (un dispositivo terminal)" puede entenderse como que un destino de la información es el dispositivo terminal, y puede incluir el envío directo o indirecto de información al dispositivo terminal. "Recibir información de ... (un dispositivo terminal)" puede entenderse como que una fuente de la información es el dispositivo terminal, y puede incluir la recepción directa o indirecta de la información desde el dispositivo terminal. La información puede sufrir el procesamiento necesario, por ejemplo, un cambio de formato, entre la fuente de envío de la información y el destino. Sin embargo, el destino puede entender información válida de la fuente.

Basándose en las descripciones anteriores de las implementaciones, un experto en la materia puede entender claramente que, a efectos de descripciones convenientes y breves, la

división en los módulos funcionales anteriores se utiliza meramente como ejemplo para las descripciones. En la aplicación real, las funciones anteriores se pueden asignar a diferentes módulos funcionales para implementación basándose en un requisito, es decir, una estructura interna de un aparato se divide en diferentes módulos funcionales para implementar todas o algunas de las funciones descritas anteriormente.

En las diversas modalidades proporcionadas en esta solicitud, debe entenderse que el aparato y método divulgados pueden implementarse de otras maneras. Por ejemplo, la modalidad del aparato descrito es meramente un ejemplo. Por ejemplo, el módulo o división en las unidades es una mera división de función lógica y puede ser otra división en la implementación real. Por ejemplo, una pluralidad de unidades o componentes se pueden combinar o integrar en otro aparato, o algunas características se pueden ignorar o no realizar. Además, los acoplamientos mutuos o acoplamientos visualizados o discutidos o los acoplamientos directos o conexiones de comunicación pueden implementarse al utilizar algunas interfaces. Los acoplamientos indirectos o las conexiones de comunicación entre los aparatos o unidades pueden implementarse en formas electrónicas, mecánicas u otras.

Las unidades descritas como partes separadas pueden o no estar físicamente separadas, y las partes mostradas como unidades pueden ser una o más unidades físicas, pueden estar ubicadas en un lugar o pueden estar distribuidas en diferentes lugares. Algunas o todas las unidades pueden seleccionarse con base en los requisitos reales para lograr los objetivos de las soluciones de las modalidades.

Además, las unidades funcionales en las modalidades de esta solicitud pueden estar integradas en una unidad de procesamiento, cada una de las unidades puede existir sola físicamente, o dos o más unidades pueden estar integradas en una unidad. La unidad integrada se puede implementar en forma de hardware, o se puede implementar en forma de unidad funcional de software.

Cuando la unidad integrada se implementan en forma de una unidad funcional de software y se venden o utilizan como un producto independiente, la unidad integrada puede almacenarse en un medio de almacenamiento legible por computadora. Basándose en dicha comprensión, las soluciones técnicas de esta solicitud esencialmente, o todas o algunas de las soluciones técnicas pueden implementarse en forma de producto de software. El producto de software se almacena en un medio de almacenamiento e incluye varias instrucciones para instruir a un dispositivo (que puede ser una microcomputadora de un solo chip, un chip o similar) o un procesador (processor) para realizar todos o algunos de los pasos de los métodos descritos en las modalidades de esta solicitud. El medio de almacenamiento anterior incluye cualquier medio que pueda almacenar código de programa, como una memoria USB, un disco duro extraíble, una ROM, una RAM, un disco magnético o un disco óptico.