(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 686 629**

(51) Int. Cl.:

H03M 13/05 (2006.01)
H03M 13/13 (2006.01)
H04L 1/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **23.12.2008** **E 14191987 (8)**

(97) Fecha y número de publicación de la concesión europea: **04.07.2018** **EP 2852063**

(54) Título: **Codificación de canal usando un código (32,11) de bloque y un código (20,O) de bloque con longitud variable O**

(30) Prioridad:

24.12.2007 US 16492 P
16.01.2008 US 21337 P
12.02.2008 US 28016 P
30.07.2008 KR 20080074682

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
18.10.2018

(73) Titular/es:

LG ELECTRONICS INC. (100.0%)
128, Yeoui-daero Yeongdeungpo-gu
Seoul 07336 , KR

(72) Inventor/es:

ROH, DONG WOOK;
AHN, JOON KUI;
YU, NAM YUL;
CHO, JUNG HYUN;
NOH, YU JIN;
KIM, KI JUN y
LEE, DAE WON

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 686 629 T3

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

DESCRIPCIÓN

Codificación de canal usando un código (32,11) de bloque y un código (20,O) de bloque con longitud variable O

Campo técnico

La presente invención se refiere a un procedimiento de codificación para un sistema de comunicación móvil y, más específicamente, a un procedimiento para realizar de manera efectiva un proceso de codificación de canal sobre información de longitud variable usando un código de bloque.

Técnica anterior

Por conveniencia de la descripción y para una mejor comprensión de la presente invención, algunos conceptos, requeridos para la presente invención entre varias teorías básicas de codificación, serán descritos en detalle más adelante en la presente memoria.

En general, un código general de corrección de errores binarios se indica por [n, k, d], donde “n” es el número de bits de palabras de código codificadas, “k” es el número de bits de información creados antes del proceso de codificación y “d” es una distancia mínima entre palabras de código. En este caso, solamente el código binario ha sido considerado para el precitado código de corrección de errores binarios, de modo que un número de posibles puntos de palabras de código en el espacio de códigos está indicado con 2^n y un número total de palabras de código codificadas está indicado con 2^k. Además, si la distancia mínima no es considerada esencialmente como importante, el precitado código de corrección de errores binarios también puede ser indicado por [n, k]. Si no hay ninguna mención del anterior código de corrección de errores en esta solicitud, debería observarse que los valores individuales de “n”, “k” y “d” están fijados en los valores mencionados anteriormente.

En este caso, el código de corrección de errores anteriormente mencionado no debería ser confundido con un código de bloque de tipo matricial, compuesto por un número X de filas (es decir, X filas) y un número Y de columnas (es decir, Y columnas).

Mientras tanto, la tasa R de codificación se define como el valor específico obtenido cuando el número de bits de información es dividido entre el número de bits de cada palabra de código. En otras palabras, la tasa R de codificación está indicada por “k/n”, es decir, R = k/n.

WILSON S G: “Linear block codes”, DIGITAL MODULATION AND CODING, XX, XX, de enero de 1996, páginas 411-493 (de aquí en adelante Wilson) revela códigos de bloque lineales. El documento de Wilson revela un código C lineal (n, k) sobre GF(q) que es un conjunto de q^k palabras de código $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$, x_iEGF(q), definido por la transformación lineal $\mathbf{x} = \mathbf{uG}$, en la que $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ indica una k-tupla de mensaje arbitrario desde GF(q) y **G** es una matriz de k por n compuesta de elementos de GF(q). **G** se denomina matriz generadora del código y el código C comprende todas las combinaciones lineales de filas de G: $\mathbf{x} = \sum_{i=0}^{k-1} u_i g_i = u_0 g_0 + u_1 g_1 + \ldots + u_{k-1} g_{0k-1}$, en la que g_j es la j-ésima fila vector de **G**.

TSG R1-99b60 "New optimal Coding for extended TFCI" revela un esquema de código para los bits 1-6 de TFCI usando el código (32, 6) de Reed Muller para obtener la palabra de código de TFCI con una longitud de 32 y un esquema de codificación para los bits 7-10 de TFCI usando dos códigos (16, 15) de Reed Muller para obtener las palabras de códigos de TFCI con una longitud de 32. El documento TSG R1-99b60 revela adicionalmente un esquema de codificación para los bits 6-10 TFCI usando (32, 10) subcódigos de segundo orden del código de Reed Muller para obtener palabras de código de TFCI con una longitud de 32, en la que el malteado de los bits TFCI a las palabras de código se describe en la siguiente tabla.

Bits de información	Palabra de código	Bits de información	Palabra de código	Bits de información	Palabra de código
0000000000	$C_{32.1}$	0001000000	$M_1 + C_{32.1}$	.....	.....
0000000001	$\overline{C}_{32,1}$	0001000001	$M_1 + \overline{C}_{32,1}$	1111000000	$M_{15} + C_{32,1}$
0000000010	$C_{32,2}$	0001000010	$M_1 + C_{32,2}$	1111000001	$M_{15} + \overline{C}_{32,1}$
.....	.....	.....	.....	.....	.....
0000111101	$\overline{C}_{32,31}$	0001111101	$M_1 + \overline{C}_{32,31}$	1111111101	$M_{15} + \overline{C}_{32,31}$
0000111110	$C_{32,32}$	0001111110	$M_1 + C_{32,32}$	1111111110	$M_{15} + C_{32,32}$
0000111111	$\overline{C}_{32,32}$	0001111111	$M_1 + \overline{C}_{32,32}$	1111111111	$M_{15} + \overline{C}_{32,32}$

El borrador del 3GPP, R1-02-0653, revela las secuencias básicas para el código (20, 5) para el que se extiende el código TFCI (16, 5).

El documento 3GPP TS 25.212 versión 3.11.0 edición 99 revela la codificación para un indicador (TFCI) de transporte-formato-combinación usando el código de Reed Muller (32, A). El documento 3GPP TX 25.212 versión 6. 8.0 edición 6, editado el 1 de junio de 2006, revela una codificación de TFCI usando un código de bloque (32, 10). De acuerdo con los documentos 3GPP TS 25.212 versión 3.11.0 y 3GPP TS 25.212 versión 6.8.0, el TFCI se codifica usando un subcódigo (32, 10) del código de Reed Muller de segundo orden. El TFCI (10 bits) a9 … a0 se codifica usando (32, 10) subcódigos del código de Reed Muller de segundo orden en la palabra de código de TFCI b_0 … b_{31}. Si el TFCI consiste en menos de 10 bits, se rellena con ceros hasta los 10 bits fijando los bits más significativos a cero. La longitud de la palabra de código de TFCI es de 32 bits. La palabra de código del subcódigo (32, 10) del código de Reed Muller de segundo orden es la combinación lineal de 10 secuencias básicas. Las secuencias básicas son como las de la tabla siguiente.

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

Los bits de información de TFCI a0, a1, a2, a3, a4, a5, a6, a7, a8, a9 (en los que a0 es el bit menos significativo y a9 es el bit más significativo) corresponderán al índice de TFCI (expresado en forma binaria sin signo) definida por

la capa RRC para referenciar el TFC del CCTrCH en el marco de radio DPCH asociado. Los bits b_i de la palabra de código de salida vienen dados por:

$$b_i = \sum_{n=0}^{9} (a_n \times M_{i,n}) \bmod 2,$$ en la que i = 0, …, 31.

Los bits de salida se indican por b_k, k = 0, 1, 2, …, 31.

Ahora, se describirá en detalle la distancia de Hamming en lo sucesivo en la presente memoria.

Si dos códigos binarios con el mismo número de bits incluyen algunos bits con distintos valores de bits, la distancia de Hamming anteriormente mencionada indica el número de los bits anteriores con los distintos valores de bits. En general, si la distancia "d" de Hamming está indicada por d = 2a + 1, el número "a" de errores pueden ser corregidos. Por ejemplo, si una de las dos palabras de código es 101001 y la otra es 110010, la distancia de Hamming entre las dos palabras de código es 3.

Mientras tanto, el término "distancia mínima", para su uso en la teoría de codificación, es indicativo de un valor mínimo entre dos palabras de código arbitrarias contenidas en un código. Se considera que la distancia mínima es uno de los criterios para evaluar las prestaciones de un código. Cuando más larga es la distancia entre las palabras de código generadas por el proceso de codificación, menor es la probabilidad de detectar erróneamente una correspondiente palabra de código como otra palabra de código; como resultado, mejoran las prestaciones de la codificación. Las prestaciones de un código total son estimadas por una distancia mínima entre las palabras de código con las peores prestaciones. En conclusión, si se maximiza una distancia mínima de un código específico, este código específico puede tener prestaciones superiores.

En el sistema de comunicación móvil de la próxima generación, la información de control transmite información constituyente del sistema e información del canal de transmisión, por lo que se considera como información muy importante para determinar las prestaciones de un sistema. En general, esta información de control tiene una longitud breve, para usar una cantidad relativamente pequeña de recursos del sistema. La información de control anteriormente mencionada es codificada por la técnica de codificación que tiene mucha resistencia a un error de canal, y se transmite a continuación. Se han considerado una gran variedad de esquemas de codificación para la información de control anterior en el sistema de comunicación móvil del 3GPP, por ejemplo, un código de bloque de longitud breve, basado en un código de Reed-Muller (RM), un código de convolución "que se muerde la cola" y un código de repetición de un código complejo.

Mientras tanto, la información de control para su uso en el sistema de LTE del 3GPP, que actúa como un formato mejorado del sistema de comunicación móvil anteriormente mencionado, es codificada por medio de códigos de bloque, de modo que la información de control codificado por bloques sea transmitida a continuación. Con más detalle, si la longitud de un bit de información de transmisión (Tx) es "A", el proceso de codificación de canal es realizado por un código de bloque compuesto por 20 filas y A columnas (es decir, código de bloque (20, A)) durante la transmisión de un canal específico (p. ej., el Canal Físico de Control de Enlace Ascendente (PUCCH)) y el resultado de la codificación de canal se transmite a continuación. En el sistema de LTE de 3GPP, la información de control del enlace ascendente es transmitida por el PUCCH y un PUSCH (es decir, un canal físico compartido de enlace ascendente). La información de control transmitida sobre el PUCCH y un PUSCH, (es decir, un canal físico compartido de enlace ascendente). La información de control transmitida por el PUSCH es codificada en canal por un código de bloque compuesto de 32 filas y A columnas (es decir, código de bloque (32, A)), de modo que la información de control codificada por canal se transmita a continuación.

Mientras tanto, el código de bloque (32, A) puede tener diversos formatos. Es difícil para el usuario buscar un formato óptimo después de comprobar prestaciones individuales de codificación de los bits de información de longitud variable asociados a todos los códigos de bloque.

Divulgación

Problema técnico

En consecuencia, la presente invención está orientada a un procedimiento de codificación de canal de información de longitud variable, que usa un código de bloque que elude esencialmente uno o más problemas debidos a las limitaciones y desventajas de la técnica relacionada.

Los objetos se resuelven mediante las características de las reivindicaciones independientes.

Un objeto de la presente invención es proporcionar un procedimiento efectivo de codificación de bloque (32, A) de información de longitud variable. En otras palabras, con la condición de que la longitud de un bit de información sea cambiada de diversas maneras, y de que la longitud de bits de una palabra de código codificada también sea

cambiada de diversas maneras, la presente invención proporciona el procedimiento de codificación de bloque (32, A) para prestar soporte efectivo a la combinación de longitudes variables de bits.

Mientras tanto, el número de bits codificados puede ser igual o menor que 32, y el número de bits de información puede ser cambiado de diversas maneras. Por lo tanto, según las siguientes realizaciones de la presente invención, la presente invención proporciona un procedimiento para usar efectivamente solamente algunas partes necesarias de todos los códigos de bloque propuestos, asociados al número de bits de información de longitud específica, o al número de bits de codificación de longitud específica. Por el contrario, si se necesita la codificación con una longitud mayor que la longitud específica anterior, la presente invención permite que se repita el código de bloque, en base a la longitud específica anterior, de modo que se realice una codificación de longitud larga.

Solución Técnica

Para conseguir estos objetos y otras ventajas y de acuerdo con la finalidad de la invención, según se ha realizado y ampliamente descrito en el presente documento, un procedimiento para información de codificación de canal de bits que usa una matriz de generación de código que incluye 32 filas y A columnas que corresponden a una longitud de los bits de información, comprendiendo el procedimiento: codificación de canal de los bits de información que tienen una longitud de A usando secuencias básicas que tienen una longitud de 32 bits que corresponde a columnas individuales de la matriz de generación de código y, generar el resultado de canal codificado como una secuencia de salida, en la que si el valor de A es superior a “10”, la matriz de generación de código se genera cuando un número (A-10) de secuencias básicas adicionales de entre muchas secuencias básicas adicionales se añadieron como secuencias de columna direccional a una primera o a una segunda matriz, en la que la primera matriz corresponde a una información de matriz de generación de código de Indicador de Combinación de Formato de Transporte (TFCI) compuesta de 32 filas y 10 columnas usadas para la información de codificación TFCI, la segunda matriz se realiza cuando al menos una ubicación entre la fila o una ubicación entre la columna de la primera matriz se cambió y, las secuencias básicas adicionales satisfacen una condición predeterminada en la que un valor de una distancia Hamming mínima es 10.

Preferentemente, la secuencia básica adicional incluye el número 10 de valores “0”,

Preferentemente, la segunda matriz se realiza cuando al menos una ubicación entre filas o una ubicación entre columnas de la primera matriz se cambió e incluye 12 filas inferiores a eliminar para generar una tercera matriz, de tal manera que la tercera matriz corresponde a otra matriz de generación de código para transmisión de Canal de Control de Enlace Ascendente Físico (PUCCH).

En otro aspecto de la presente invención, se proporciona un procedimiento para bits de información de codificación de canal usando una matriz de generación de código que incluye 32 filas y A columnas que corresponden a una longitud de los bits de información, comprendiendo el procedimiento: codificación de canal de los bits de información que tienen la longitud de A usando secuencias básicas que tienen una longitud de 32 bits que corresponden a columnas individuales de la matriz de generación de código y generar el resultado de canal codificado como una secuencia de salida en la que, si el valor A es superior a “10”, la matriz de generación de código se genera cuando el número (A-10) de las secuencias básicas adicionales de entre muchas secuencias básicas adicionales se añadieron como secuencias de dirección de columna a una primera o segunda matriz, en la que la segunda matriz se realiza cuando al menos una de una ubicación entre filas o una ubicación entre columnas de la primera matriz se cambió, la primera matriz se representa mediante la siguiente tabla,

[Tabla]

i	$M_{i.0}$	$M_{i.1}$	$M_{i.2}$	$M_{i.3}$	$M_{i.4}$	$M_{i.5}$	$M_{i.6}$	$M_{i.7}$	$M_{i.8}$	$M_{i.9}$
0	1	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	1
2	1	0	0	1	0	0	1	0	1	1
3	1	0	1	1	0	0	0	0	1	0
4	1	1	1	1	0	0	0	1	0	0
5	1	1	0	0	1	0	1	1	1	0
6	1	0	1	0	1	0	1	0	1	1

(continuación)

i	$M_{i.0}$	$M_{i.1}$	$M_{i.2}$	$M_{i.3}$	$M_{i.4}$	$M_{i.5}$	$M_{i.6}$	$M_{i.7}$	$M_{i.8}$	$M_{i.9}$
i	Mi.0	Mi.1	Mi.2	Mi.3	Mi.4	Mi.5	Mi.6	Mi.7	Mi.8	Mi.9
7	1	0	0	1	1	0	0	1	1	0
8	1	1	0	1	1	0	0	1	0	1
9	1	0	1	1	1	0	1	0	0	1
10	1	0	1	0	0	1	1	1	0	1
11	1	1	1	0	0	1	1	0	1	0
12	1	0	0	1	0	1	0	1	1	1
13	1	1	0	1	0	1	0	1	0	1
14	1	0	0	0	1	1	0	1	0	0
15	1	1	0	0	1	1	1	1	0	1
16	1	1	1	0	1	1	1	0	0	1
17	1	0	0	1	1	1	0	0	1	0
18	1	1	0	1	1	1	1	1	0	0
19	1	0	0	0	0	1	1	0	0	0
20	1	0	1	0	0	0	1	0	0	0
21	1	1	0	1	0	0	0	0	0	1
22	1	0	0	0	1	0	0	1	1	0
23	1	1	1	0	1	0	0	0	1	1
24	1	1	1	1	1	0	1	1	1	1
25	1	1	0	0	0	1	1	1	0	0
26	1	0	1	1	0	1	0	0	1	1
27	1	1	1	1	0	1	0	1	1	1
28	1	0	1	0	1	1	1	0	1	0
29	1	0	1	1	1	1	1	1	1	0
30	1	1	1	1	1	1	1	1	1	1
31	1	0	0	0	0	0	0	0	0	0

, y las varias secuencias básicas adicionales son iguales a
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0],
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0],
[0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,0, 0, 1,0, 1, 0, 1, 1, 0, 0, 0, 1, 1,0], y
[0, 0, 1, 1, 1,0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,0, 1, 1,0, 0, 1,0, 1,0, 0, 1], respectivamente.

En otro aspecto de la presente invención, se proporciona un procedimiento para bits de información de codificación de canal usando una matriz de generación de código que incluye 32 filas y A columnas que corresponden a una longitud de los bits de información, comprendiendo el procedimiento: codificar el canal de los bits de información que tienen la longitud de A usando secuencias básicas que tienen una longitud de 32 bits que corresponden a columnas individuales de la matriz de generación de código y, generar el resultado de canal codificado como una secuencia generada, en la que la matriz de generación de código corresponde a una décima matriz que tiene 32

filas y A columnas, en la que una cuarta matriz corresponde a una matriz compuesta de 20 filas y A columnas en la que el número A de secuencias básicas se seleccionó secuencialmente de un lado izquierdo de una quinta matriz representada por la siguiente tabla

[Tabla]

i	$M_{i.0}$	$M_{i.1}$	$M_{i.2}$	$M_{i.3}$	$M_{i.4}$	$M_{i.5}$	$M_{i.6}$	$M_{i.7}$	$M_{i.8}$	$M_{i.9}$	$M_{i.10}$	$M_{i.11}$	$M_{i.12}$	$M_{i.13}$
0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
1	1	1	1	0	0	0	0	0	0	1	1	1	0	0
2	1	0	0	1	0	0	1	0	1	1	1	1	1	1
3	1	0	1	1	0	0	0	0	1	0	1	1	1	1
4	1	1	1	1	0	0	0	1	0	0	1	1	1	1
5	1	1	0	0	1	0	1	1	1	0	1	1	1	0
6	1	0	1	0	1	0	1	0	1	1	1	1	1	0
7	1	0	0	1	1	0	0	1	1	0	1	1	1	1
8	1	1	0	1	1	0	0	1	0	1	1	1	1	1
9	1	0	1	1	1	0	1	0	0	1	1	1	1	1
10	1	0	1	0	0	1	1	1	0	1	1	1	1	1
11	1	1	1	0	0	1	1	0	1	0	1	1	1	1
12	1	0	0	1	0	1	0	1	1	1	1	1	1	1
13	1	1	0	1	0	1	0	1	0	1	1	1	1	1
14	1	0	0	0	1	1	0	1	0	0	1	0	1	1
15	1	1	0	0	1	1	1	1	0	1	1	0	1	1
16	1	1	1	0	1	1	1	0	0	1	0	1	1	1
17	1	0	0	1	1	1	0	0	1	0	0	1	1	1
18	1	1	0	1	1	1	1	1	0	0	0	0	0	1
19	1	0	0	0	0	1	1	0	0	0	0	0	0	1

, y una sexta matriz es una matriz realizada cuando al menos una ubicación entre filas o una ubicación entre columnas de la cuarta matriz se cambió, y una séptima matriz es una matriz realizada cuando 12 bits adicionales se añadieron a cada una de las secuencias básicas de la cuarta o la sexta matriz, en la que el componente de la séptima matriz corresponde a una matriz compuesta de 12 filas y A columnas en la que el número A de secuencias básicas se seleccionó secuencialmente de un lado izquierdo de una octava matriz representada por la siguiente tabla

[Tabla]

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

(continuación)

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

, y una novena matriz se genera cuando al menos una ubicación entre filas o entre columnas de la séptima matriz se cambió y la décima matriz es una matriz de generación de código generada cuando las secuencias básicas que corresponden al número A de las secuencias básicas de un lado izquierdo de la séptima o la novena matriz se seleccionó.

Preferentemente, si el valor A es igual o inferior a “14”, un número predeterminado de secuencias de columna direccional que corresponden a la longitud A basándose en un lado izquierdo de secuencias de columna direccional de una undécima matriz corresponde secuencialmente a las secuencias básicas de la matriz de generación de código, en la que la undécima matriz está representada por la siguiente tabla.

[Tabla]

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

Preferentemente, si el valor A es igual a o inferior a “11”, un número predeterminado de secuencias de columna direccional que corresponden a la longitud A basándose en un lado izquierdo de las secuencias de columna

direccional de una duodécima matriz que corresponde secuencialmente a las secuencias básicas de la matriz de generación de código, en la que la duodécima matriz se representa por la siguiente tabla.

[Tabla]

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
6	1	0	1	0	1	0	1	0	1	1	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
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
19	1	0	0	0	0	1	1	0	0	0	0
7	1	0	0	1	1	0	0	1	1	0	1
14	1	0	0	0	1	1	0	1	0	0	1
5	1	1	0	0	1	0	1	1	1	0	1
18	1	1	0	1	1	1	1	1	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

(continuación)

i	$M_{i.0}$	$M_{i.1}$	$M_{i.2}$	$M_{i.3}$	$M_{i.4}$	$M_{i.5}$	$M_{i.6}$	$M_{i.7}$	$M_{i.8}$	$M_{i.9}$	$M_{i.10}$
30	1	1	1	1	1	1	1	1	1	1	1
31	1	0	0	0	0	0	0	0	0	0	0

Preferentemente, el procedimiento comprende, además: si el número de bits (es decir, un número de bit) de la secuencia generada es al menos 32 (es decir, al menos 32 bits), repetir cada secuencia básica de la matriz de generación de código un número de veces predeterminado y, realizar un procedimiento de codificación de canal usando una parte específica que tiene una longitud predeterminada que corresponde al número de bit de la secuencia generada de entre la secuencia básica repetida.

Preferentemente, si el número de bit de la secuencia generada es superior a 32 (es decir, 32 bits), la secuencia generada se adquiere cuando el resultado del canal codificado se repite cíclicamente.

Preferentemente, el bit de información corresponde a al menos uno de Información de Calidad de Canal (CQI) y un Índice de Matriz de Precodificación (PMI).

Preferentemente, la secuencia generada se transmite sobre un Canal Compartido de Enlace Ascendente Físico (PUSCH).

Debe entenderse que tanto la descripción general anterior como la siguiente descripción detallada de la presente invención son ejemplares y explicativas y pretenden proporcionar una explicación adicional de la invención reivindicada.

Efectos Ventajosos

Según las realizaciones precitadas de la presente invención, la presente invención reutiliza una matriz de generación de código usada para la codificación de información de TFCI (Indicador de Combinación de Formato de Transporte) de un sistema convencional de 3GPP y / o la matriz de generación de código (20, A) para la transmisión del PUCCH, de modo que pueda implementar fácilmente la codificación de bloque (32, k). Como resultado, aumenta una distancia mínima entre las palabras de código generadas, dando como resultado unas prestaciones aumentadas del sistema.

Descripción de los dibujos

Los dibujos adjuntos, que son incluidos para proporcionar una comprensión adicional de la invención, ilustran realizaciones de la invención y, junto con la descripción, sirven para explicar el principio de la invención.

En los dibujos:

la FIG. 1 es un diagrama conceptual que ilustra un procedimiento para generar el código de bloque (32, 14) usando el código de bloque (32, 10) usado para una codificación convencional de información de TFCI y el código (20, 14) usado para la transmisión del PUCCH según la presente invención; y
la FIG. 2 es un gráfico que ilustra las prestaciones de distancia mínima obtenidas cuando el código de bloque (40, k), el código de bloque (52, k) y el código de bloque (64, k) son generados, a condición de que el código bloque (20, k) y el código de bloque (32, k) sean usados como códigos de base.

Modalidad óptima

Se hará ahora referencia en detalle a las realizaciones preferidas de la presente invención, cuyos ejemplos están ilustrados en los dibujos adjuntos. Toda vez que sea posible, los mismos números de referencia serán usados en toda la extensión de los dibujos para referirse a las mismas partes, o a partes similares.

Antes de describir la presente invención, debería observarse que la mayoría de los términos revelados en la presente invención corresponden a términos generales bien conocidos en la técnica, pero algunos términos han sido seleccionados por el solicitante como necesarios, y serán revelados más adelante en la presente memoria, en la siguiente descripción de la presente invención. Por lo tanto, es preferible que los términos definidos por el solicitante sean entendidos en base a sus significados en la presente invención. Por ejemplo, aunque la siguiente descripción se refiere a un ejemplo detallado aplicado al sistema LTE 3GPP (Evolución a Largo Plazo del Proyecto de Colaboración de 3ª Generación), la presente invención también puede ser aplicada, no solamente al anterior sistema LTE de 3GPP, sino también a otros sistemas arbitrarios de comunicación que necesitan realizar el proceso de codificación de canal sobre información de control de longitud variable, usando el código de bloque.

Por conveniencia de la descripción y para una mejor comprensión de la presente invención, las estructuras y dispositivos generales bien conocidos en la técnica serán omitidos, o serán indicados por un diagrama de bloques o un diagrama de flujo. Toda vez que sea posible, los mismos números de referencia serán usados en toda la extensión de los dibujos para referirse a las mismas partes, o a partes similares.

En primer lugar, los elementos que han de ser considerados en común por la presente invención serán descritos en detalle en adelante en la presente memoria.

El código [n, k] es un código específico en el cual el número de bits de codificación es “n” y el número de bits de información es “k”.

En este caso, si no hay ninguna mención en una matriz de generación de cada código, esta matriz de generación está representada por una tabla de tipo de secuencia básico. De hecho, el procedimiento de codificación es similar al de un código de TFCI del 3GPP edición 99 explicado en la técnica antecedente. En otras palabras, los bits de información son adjudicados secuencialmente a la secuencia de base del lado izquierdo, y una secuencia correspondiente al producto de la secuencia de base y los bits de información son añadidos por operaciones binarias (es decir, suma por O Exclusivo) correspondientes al número de bits de información, de modo que se generen bits de codificación.
Si el código está representado según el procedimiento precitado, aunque el número de bits de información (denominado en adelante en la presente memoria “número de bit de información” es variable, la presente invención tiene la ventaja de que puede realizar el proceso de codificación de datos en base a una tabla de secuencias de base de tipo matricial. La presente invención es capaz de prestar soporte a una gran variedad de números de bit de información usando las ventajas precitadas. Por lo tanto, una tabla de secuencias de base o una matriz de generación de código está representada en consideración de un número de bit de información de tamaño máximo. Si el número máximo de bits de información necesarios para la aplicación efectiva es menor que el siguiente tamaño propuesto, es preferible que se use una tabla que no tenga ninguna secuencia de base para al menos un correspondiente número máximo de bit (es decir, un correspondiente máximo número de bits).
Mientras tanto, la operación de reemplazar un código de generación “0” por otro código de generación “1”, según la teoría de codificación, no tiene ninguna influencia sobre las características del código. Por lo tanto, aunque el valor de “0” sea reemplazado por el otro valor de “1” en la tabla de secuencias de base, el resultado reemplazado puede indicar el mismo bit de información de código se añade mediante operaciones binarias (es decir, Exclusiva o suma) que corresponde al número de bits de información, tal como un bit de codificación se genera.

Si el código está representado de acuerdo con el procedimiento anteriormente mencionado, aunque el número de bits de información (en lo sucesivo, denominado como “número de bit de información”) es variable, la presente invención tiene una ventaja en que puede realizar el procedimiento de codificación de datos basándose en una tabla de secuencia básica de tipo matriz. La presente invención es capaz de soportar una variedad de números de bits de información usando las ventajas anteriormente mencionadas. Por lo tanto, una tabla de secuencias básicas o una matriz de generación de código se representa en consideración de un número de bits de información dimensionado al máximo. Si el número máximo de bits de información necesario para la aplicación real es inferior al tamaño propuesto siguiente, es preferente que una tabla que no tiene secuencias básicas para al menos para al menos un número de bits máximo correspondiente (es decir, un número máximo correspondiente de bits) se use.

Mientras tanto, la operación de reemplazar un código de generación de “0” con otro código de generación de “1” de acuerdo con la teoría de codificación no tiene influencia en las características de codificación. Por lo tanto, aunque el valor de “0” se reemplaza con el otro valor de “1” basándose en la tabla de secuencia, el resultado reemplazado puede indicar la misma característica de código. Además, aunque el orden de los bits de codificación sea reemplazado por otro orden según la teoría de la codificación, el resultado reemplazado también puede indicar la misma característica del código. Por lo tanto, aunque la ubicación de la fila sea reemplazada por otra ubicación de fila en la tabla de secuencias de base, el resultado reemplazado también puede indicar la misma característica del código. También, aunque el orden de los bits de codificación se sustituya con otro orden de acuerdo con la teoría de codificación, el resultado sustituido puede indicar también la misma característica del código. Por lo tanto, aunque la localización de la fila se sustituya con otra localización de filas en la tabla de secuencia base, al resultado sustituido puede indicar también la misma característica del código.

La secuencia de base propuesta por las siguientes realizaciones está diseñada para tener un número variable de bits de información (es decir, un número de bit de información variable), y también está diseñada para tener un número variable de bits de codificación (es decir, un número de bit de codificación variable). Por lo tanto, en el caso de idear el concepto inventivo de la presente invención, fue pre-considerado un código específico en el cual una columna específica es eliminada de una tabla específica de secuencias de base. Por ejemplo, si una tabla de secuencias de base es indicada por un formato específico tal como (32, 14), la tabla de secuencias de base de formato (20, 11) es uno de los ejemplos de aplicación de la tabla anterior de secuencias de base de formato (32, 14). Con más detalle, si 12 columnas son sucesivamente eliminadas del extremo inferior de la tabla de secuencias de base de formato (32, 14), y 3 filas son eliminadas a la derecha de la tabla de secuencias de base de formato

(32, 14), puede obtenerse la precitada tabla de secuencias de base de formato (20, 11). En breve, la fila y columna de la tabla de secuencias de base según la presente invención han sido decididas en base al tamaño más grande. En el caso de la fila y columna de pequeño tamaño, las filas y columnas de la tabla de secuencias de base del mayor tamaño son eliminadas secuencialmente de la derecha o el extremo inferior de la tabla anterior, de modo que se use el resultado eliminado. No hace falta decir que, según lo anteriormente expresado en lo que antecede, aunque la ubicación de fila pueda ser reemplazada por la ubicación de columna en la tabla de secuencias de base de menor tamaño, o aunque la ubicación de “0” sea reemplazada por la otra ubicación de “1” en la tabla de secuencias de base, los anteriores resultados reemplazados pueden tener los mismos códigos.

Por conveniencia de la descripción y para una mejor comprensión de la presente invención, en el caso de indicar el orden de datos cualesquiera de la presente invención, los bits de información proceden secuencialmente en la dirección desde la columna izquierda hasta la columna derecha en la anterior tabla de secuencias de base. Los bits de codificación proceden secuencialmente en la dirección desde la fila de más arriba hasta la fila de más abajo en la anterior tabla de secuencias de base. El término “tabla de secuencias de base” también puede llamarse “Matriz de generación de código”, o con otros términos, según sea necesario.

Mientras tanto, es preferible que no esté disponible una secuencia de base de patrón específico para su uso en un procedimiento específico de estimación de canal. En este caso, una tabla específica, de la cual se elimina una secuencia de base específica, en base a un tipo de sistema, puede ser seleccionada entre distintas tablas propuestas por las siguientes realizaciones. Desde el punto de vista del proceso de codificación, una correspondiente secuencia de base siempre puede ser considerada como “0”, de modo que se produzcan las mismas prestaciones de codificación, pero solamente el número de bits de información se reduce.

Por lo tanto, según la tabla de secuencias de base propuesta por las siguientes realizaciones, una tabla específica de secuencias de base, que no tenga ninguna secuencia de base específica o matriz de generación de código, ya ha sido considerada en el caso del diseño de la presente invención.

Según el aspecto único precitado de la presente invención, la presente invención proporciona el procedimiento de codificación de bloque de formato (32, A). Con más detalle, la presente invención proporciona el procedimiento de codificación de bloque de formato (32, 14), en consideración de la máxima longitud (es decir, 14 bits) de los bits de información. Sin embargo, debería observarse que solamente algunas secuencias de base entre la matriz de generación de código de formato (32, 14) son usadas según la longitud de bits de información, según lo anteriormente expresado en lo que antecede.

Aunque pueden usarse una gran variedad de procedimientos para los fines precitados, las siguientes realizaciones proporcionan un procedimiento de diseño de código capaz de mantener un máximo punto común con los códigos convencionales, usando los códigos de bloque convencionales. Con más detalle, los códigos convencionales proporcionan un primer procedimiento de uso del código de bloque de formato (32, 10) para la codificación de información de TFCI usada en el 3GPP edición 99, y un segundo procedimiento de uso del código de bloque de formato (20, 14) para la transmisión del PUCCH. En este caso, el precitado código de bloque de formato (20, 14) ha sido revelado en la Solicitud Provisional Estadounidense Nº 61 / 016.492 usada como prioridad de la presente invención, titulada “PROCEDIMIENTO DE GENERACIÓN DE DIVERSOS CÓDIGOS DE BLOQUE DE BREVE LONGITUD CON ESTRUCTURA ANIDADA, PUNZANDO UN CÓDIGO DE BASE”, presentada por el mismo solicitante de la presente invención, que es incorporada a la presente memoria por referencia. Las descripciones detalladas del código de bloque de formato (32, 10) y del código de bloque de formato (20, 14) serán descritas en detalle más adelante en la presente memoria.

El código de bloque de formato (32, 10) para la codificación de información de TFCI y el código de bloque de formato (20, 14) para la transmisión del PUCCH serán descritos en detalle más adelante en la presente memoria.

Código de bloque de TFCI (32, 10) y código de bloque (20, 14)

El código de bloque (20, 14), para su uso en esta realización, será generado de la siguiente manera. En primer lugar, el código de bloque (20, 10) es generado a partir de la matriz de generación de código de TFCI estructurada en formato (32, 10), y las 4 secuencias de base son añadidas al código de bloque (20, 10), de modo que sea generado el código de bloque (20, 14).

El código de bloque (20, 10) se basa en la matriz de generación de código (32, 10) usada para la codificación de canal de la información de TFCI (Indicador de Combinación de Formato de Transporte) en el 3GPP Edición 99. Como resultado, el código de bloque (20, 10) puede ser diseñado para ser punzado según la longitud de una palabra de código a codificar.

La reutilización del código de información de TFCI (32, 10) tiene una gran variedad de ventajas. Por ejemplo, el código de información de TFCI ha sido diseñado en base al código de Reed-Muller (RM), de modo que el código de TFCI punzado puede tener una estructura modificada de código de Reed-Muller (RM). Este código basado en el de

Reed-Muller (RM) tiene la ventaja de que puede ser rápidamente descodificado por un procedimiento de transformación rápida de Hadamard durante el proceso de descodificación. Como otro ejemplo, el procedimiento de codificación de TFCI presta soporte a un bit de información de longitud variable y a un bit de codificación de longitud variable. De esta manera, la longitud del bit de información puede ser cambiada de varias maneras, de modo que puedan ser bien satisfechos los requisitos para la transmisión de CQI de la LTE del 3GPP.

La siguiente Tabla 1 muestra la matriz de generación de código (32, 10) usada para la codificación de canal de la información de TFCI en el 3GPP Edición 99. En este caso, la matriz de generación de código (32, 10) genera una palabra de código específica que tiene una longitud de 32 bits y el valor d_{min} = 12.

[Tabla 1]

Índice de TFCI	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$
0	1	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	1	0	0	0
2	1	1	0	0	0	1	0	0	0	1
3	0	0	1	0	0	1	1	0	1	1
4	1	0	1	0	0	1	0	0	0	1
5	0	1	1	0	0	1	0	0	1	0
6	1	1	1	0	0	1	0	1	0	0
7	0	0	0	1	0	1	0	1	1	0
8	1	0	0	1	0	1	1	1	1	0
9	0	1	0	1	0	1	1	0	1	1
10	1	1	0	1	0	1	0	0	1	1
11	0	0	1	1	0	1	0	1	1	0
12	1	0	1	1	0	1	0	1	0	1
13	0	1	1	1	0	1	1	0	0	1
14	1	1	1	1	0	1	1	1	1	1
15	1	0	0	0	1	1	1	1	0	0
16	0	1	0	0	1	1	1	1	0	1
17	1	1	0	0	1	1	1	0	1	0
18	0	0	1	0	1	1	0	1	1	1
19	1	0	1	0	1	1	0	1	0	1
20	0	1	1	0	1	1	0	0	1	1
21	1	1	1	0	1	1	0	1	1	1
22	0	0	0	1	1	1	0	1	0	0
23	1	0	0	1	1	1	1	1	0	1
24	0	1	0	1	1	1	1	0	1	0
25	1	1	0	1	1	1	1	0	0	1
26	0	0	1	1	1	1	0	0	1	0
27	1	0	1	1	1	1	1	1	0	0
28	0	1	1	1	1	1	1	1	1	0
29	1	1	1	1	1	1	1	1	1	1
30	0	0	0	0	0	1	0	0	0	0
31	0	0	0	0	1	1	1	0	0	0

En general, como es bien conocido en la técnica, aunque la ubicación entre filas o la ubicación entre columnas sea

reemplazada por otras ubicaciones en el código de bloque, no hay ninguna diferencia en las prestaciones entre las palabras de código generadas. La siguiente Tabla 2 muestra un código de bloque específico, que es equivalente al código de bloque (32, 10) usado para la precitada codificación de información de TFCI que usa la ventaja precitada.

[Tabla 2]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
$g_{10por32}[1]$	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
$g_{10por32}[2]$	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
$g_{10por32}[3]$	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
$g_{10por32}[4]$	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
$g_{10por32}[5]$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$g_{10por32}[6]$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$g_{10por32}[7]$	1	1	1	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	0	0	0	1	1	0	0	0	0	1	0	1	0	0
$g_{10por32}[8]$	1	1	1	0	0	0	1	1	1	0	1	1	0	1	1	0	1	0	1	1	0	0	1	1	1	0	0	0	0	0	0	0
$g_{10por32}[9]$	1	1	0	1	0	1	0	0	1	1	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	1	0	1	0	0	0	0
$g_{10por32}[10]$	1	0	0	0	1	0	1	0	1	1	1	1	0	1	0	0	1	1	1	0	1	1	0	0	0	0	1	1	1	0	0	0

Como puede apreciarse en el código de bloque mostrado en la Tabla 2, las ubicaciones de fila y de columna del código (32, 10) usado para la codificación de TFCI están cambiadas por otras ubicaciones, y las ubicaciones entre algunas columnas (o ubicaciones entre algunas filas en base al código de información de TFCI) están intercambiadas entre sí.

En otras palabras, según esta realización de la presente invención, pueden punzarse 12 filas, ya sea en el código de información de TFCI de formato (32, 10) (Tabla 1) o en su matriz equivalente (Tabla 2), o bien se seleccionan 20 filas del código de información de TFCI de formato (32, 10) (Tabla 1) o de su matriz equivalente (Tabla 2), de modo que se configure el código de bloque (20, 10), En este caso, desde el punto de vista del código de bloque de la Tabla 2, pueden punzarse 12 columnas, y pueden seleccionarse 20 columnas. No hay ninguna diferencia en las prestaciones de la modalidad entre el primer caso del uso de la Tabla 1 y el segundo caso del uso de la Tabla 2, Por conveniencia de la descripción y para una mejor comprensión de la presente invención, se supone que la presente invención usa el formato equivalente (Véase la Tabla 2) del código de información de TFCI si no hay ninguna mención en la descripción precitada.

Mientras tanto, el código (32, 10) usado para codificar la información de TFCI ha sido generado en base al código de Reed-Muller (RM). En este caso, a fin de implementar una realización de la corrección de errores, es muy importante para el anterior código (32, 10) buscar un patrón de punción que permita que una palabra de código tenga la distancia (d_{min}) más larga.

En comparación con esta realización, se describirá en detalle a continuación en la presente memoria una búsqueda exhaustiva, capaz de buscar un patrón óptimo de punción en una matriz de generación del código (32, 10) usado para la codificación de TFCI. Siempre que el número de columnas de una matriz de generación a punzar en la matriz de (32x10) (también indicada como matriz de (32*10)) esté fijado en "p", el número de todos los patrones disponibles de punción está indicado por $\binom{32}{p}$. En este caso, $\binom{32}{p}$ es indicativo del número de casos, cada uno de los cuales selecciona las p columnas entre 32 columnas.

Por ejemplo, si el valor de "p" es 12 (p = 12), hay distintas matrices de generación (10x20) (es decir, $\binom{32}{12}$ = 225.792.840 = número de matrices de generación (10x20)), y la información de 10 bits (es decir, 2^{10} = 1.024 = número de segmentos de información) se codifica en una palabra de código de 20 bits. Se calcula una distancia mínima de Hamming (d_{min}) entre palabras de código generadas por matrices individuales, de modo que se halle una matriz de generación con el valor más alto en la anterior distancia mínima de Hamming (d_{min}). Si se usa un patrón de punción para formar la matriz de generación que tenga el máximo valor (d_{min}), este patrón de punción es considerado como el último patrón finalmente hallado. Sin embargo, la generación del código óptimo de bloque (20, 10), en base a las etapas anteriores, requiere un gran número de cálculos, lo que da como resultado una

mayor incomodidad de uso.

Por lo tanto, esta realización de la presente invención añade condiciones específicas de restricción al proceso para decidir el patrón de punción, de modo que reduzca la gama de un espacio de búsqueda para obtener un valor (d_{min}) óptimo.

A continuación, se describirá en detalle, en adelante en la presente memoria, un procedimiento para buscar más efectivamente una matriz de generación del código (20, 10) que genera una palabra de código con (d_{min}) = d. A condición de que un valor (d_{min}) de destino está indicado por d, el peso de Hamming $w(g_{10por20}[i])$ de cada vector fila $g_{10por20}[i]$ ($1 \leq i \leq 10$) de la matriz de generación de código (20, 10) tiene algunos requisitos mostrados en la siguiente Ecuación 1.

[Ecuación 1]

$$d \leq w(g_{10por20}[i]) \leq 20 - d$$

$$i = 0, 1, ..., 10$$

$$i \neq 6 \text{ (Hay todos unos en este vector fila)}$$

Por ejemplo, si el valor de d es 6 (d = 6), la Ecuación 1 puede ser representada por la siguiente Ecuación 2.

[Ecuación 2]

$$6 \leq w(g_{10por20}[i]) \leq 14$$

Por lo tanto, si la restricción anterior de la Ecuación 2 es añadida a los vectores fila individuales $g_{10por20}[i]$ de la matriz (10*20) generada a partir del precitado proceso de búsqueda exhaustiva, el resultado añadido puede reducir el número

$(N << \binom{32}{12})$ de espacios de búsqueda capacitados para la búsqueda de una matriz de generación que genere una palabra de código con d_{min} = 6. En general, en base a diversas referencias citadas, es bien conocido en la técnica que un valor máximo (d_{min}) del código (20, 10) es 6, y su descripción detallada ha sido revelada en "La teoría de códigos correctores de errores (de F. J. MacWilliams y N. J. A. Sloane)". Por lo tanto, la condición de d_{min} = 6 se aplica a la condición de la Ecuación 1, por lo que pueden hallarse 360 patrones que indican prestaciones óptimas.

Los anteriores 360 patrones de punción han sido revelados en el Anexo "A" de la Solicitud Provisoria Estadounidense Nº 61/016.492, titulada "PROCEDIMIENTO DE GENERACIÓN DE DIVERSOS CÓDIGOS DE BLOQUE DE LONGITUD BREVE CON ESTRUCTURA ANIDADA, PUNZANDO UN CÓDIGO BASE", presentada por el mismo solicitante de la presente invención. Específicamente, los índices de las columnas punzadas, en base a la Tabla 2, han sido revelados en el Anexo "A". Por conveniencia de la descripción, se omitirán en la presente memoria los índices precitados.

El siguiente patrón, entre los anteriores 360 patrones de punción, será descrito como ejemplo a continuación en la presente memoria.

La siguiente Tabla 3 muestra un patrón específico, que indica la distribución de un peso específico de Hamming entre los 360 patrones.

[Tabla 3]

matriz 10x32	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Tabla 2	0	0	1	1	1	0	1	1	0	0	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0

Los patrones de punción mostrados en la Tabla 3 corresponden al sexto índice de la Tabla A.2 del Anexo "A" de la Solicitud Provisoria Estadounidense Nº 61 / 016.492. En este caso, el valor "0" de la Tabla 3 indica que una columna correspondiente a la ubicación del "0" está punzada. El valor "1" de la Tabla 3 indica que una columna correspondiente a la ubicación del "1" no está punzada, sino seleccionada para el código de bloque (20, 10).

En el caso en que el patrón de punción de la Tabla 3 se aplica a la Tabla 2, el resultado se muestra como la siguiente Tabla 4.

[Tabla 4]

Índice	$Mi_{,0}$	$Mi_{,1}$	$Mi_{,2}$	$Mi_{,3}$	$Mi_{,4}$	$Mi_{,5}$	$Mi_{,6}$	$Mi_{,7}$	$Mi_{,8}$	$Mi_{,9}$	punción para (20,10)
0	1	1	1	1	1	1	1	1	1	1	Punzado
1	0	1	1	1	1	1	1	1	1	0	Punzado
2	1	0	1	1	1	1	1	1	0	0	
3	0	0	1	1	1	1	0	0	1	0	
4	1	1	0	1	1	1	1	0	0	1	
5	0	1	0	1	1	1	1	0	1	0	Punzado
6	1	0	0	1	1	1	1	1	0	1	
7	0	0	0	1	1	1	0	1	0	0	
8	1	1	1	0	1	1	0	1	1	1	Punzado
9	0	1	1	0	1	1	0	0	1	1	Punzado
10	1	0	1	0	1	1	0	1	0	1	
11	0	0	1	0	1	1	0	1	1	1	
12	1	1	0	0	1	1	1	0	1	0	
13	0	1	0	0	1	1	1	1	0	1	
14	1	0	0	0	1	1	1	1	0	0	Punzado
15	0	0	0	0	1	1	1	0	0	0	
16	1	1	1	1	0	1	1	1	1	1	Punzado
17	0	1	1	1	0	1	1	0	0	1	
18	1	0	1	1	0	1	0	1	0	1	
19	0	0	1	1	0	1	0	1	1	0	
20	1	1	0	1	0	1	0	0	1	1	Punzado
21	0	1	0	1	0	1	1	0	1	1	
22	1	0	0	1	0	1	1	1	1	0	
23	0	0	0	1	0	1	0	1	1	0	Punzado
24	1	1	1	0	0	1	0	1	0	0	

(continuación)

Índice	$Mi_{,0}$	$Mi_{,1}$	$Mi_{,2}$	$Mi_{,3}$	$Mi_{,4}$	$Mi_{,5}$	$Mi_{,6}$	$Mi_{,7}$	$Mi_{,8}$	$Mi_{,9}$	punción para (20,10)
25	0	1	1	0	0	1	0	0	1	0	
26	1	0	1	0	0	1	0	0	0	1	Punzado
27	0	0	1	0	0	1	1	0	1	1	
28	1	1	0	0	0	1	0	0	0	1	
29	0	1	0	0	0	1	1	0	0	0	Punzado
30	1	0	0	0	0	1	0	0	0	0	
31	0	0	0	0	0	1	0	0	0	0	Punzado

En este caso, en comparación con la Tabla 2, las direcciones de fila y columna de la Tabla 4 están cambiadas, pero la Tabla 4 tiene el mismo significado que la Tabla 2. 12 filas punzadas entre las secuencias individuales con dirección de fila se muestran en el lado derecho de la Tabla 4. Como resultado, el código de bloque (20, 10) generado puede ser representado por la Tabla 4.

[Tabla 5]

Índice	$M_{i.0}$	$M_{i.1}$	$M_{i.2}$	$M_{i.3}$	$M_{i.4}$	$M_{i.5}$	$M_{i.6}$	$M_{i.7}$	$M_{i.8}$	$M_{i.9}$
2	1	0	1	1	1	1	1	1	0	0
3	0	0	1	1	1	1	0	0	1	0
4	1	1	0	1	1	1	1	0	0	1
6	1	0	0	1	1	1	1	1	0	1
7	0	0	0	1	1	1	0	1	0	0
10	1	0	1	0	1	1	0	1	0	1
11	0	0	1	0	1	1	0	1	1	1
12	1	1	0	0	1	1	1	0	1	0
13	0	1	0	0	1	1	1	1	0	1
15	0	0	0	0	1	1	1	0	0	0
17	0	1	1	1	0	1	1	0	0	1
18	1	0	1	1	0	1	0	1	0	1
19	0	0	1	1	0	1	0	1	1	0
21	0	1	0	1	0	1	1	0	1	1
22	1	0	0	1	0	1	1	1	1	0
24	1	1	1	0	0	1	0	1	0	0
25	0	1	1	0	0	1	0	0	1	0
27	0	0	1	0	0	1	1	0	1	1
28	1	1	0	0	0	1	0	0	0	1
30	1	0	0	0	0	1	0	0	0	0

Mientras tanto, hay una pequeña diferencia entre el orden de las filas de la Tabla 4 o la Tabla 5 y el orden matricial de la codificación de TFCI basada en el 3GPP. Como se ha descrito anteriormente, aunque la ubicación de cada fila está cambiada por otra ubicación según la precitada teoría de codificación, no hay ninguna diferencia en prestaciones entre las palabras de código generadas. Si el orden de las filas de la Tabla 4 o la Tabla 5 es ajustado de la misma manera que en la matriz de código de TFCI, se obtiene la siguiente Tabla 6.

[Tabla 6]

Índice de Tabla 4	Índice de TFCI	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$	Punción para (20,10)
30	**0**	1	0	0	0	0	1	0	0	0	0	
29	**1**	0	1	0	0	0	1	1	0	0	0	Punzado
28	**2**	1	1	0	0	0	1	0	0	0	1	
27	**3**	0	0	1	0	0	1	1	0	1	1	
26	**4**	1	0	1	0	0	1	0	0	0	1	Punzado
25	**5**	0	1	1	0	0	1	0	0	1	0	
24	**6**	1	1	1	0	0	1	0	1	0	0	
23	**7**	0	0	0	1	0	1	0	1	1	0	Punzado
22	**8**	1	0	0	1	0	1	1	1	1	0	
21	**9**	0	1	0	1	0	1	1	0	1	1	
20	**10**	1	1	0	1	0	1	0	0	1	1	Punzado
19	**11**	0	0	1	1	0	1	0	1	1	0	
18	**12**	1	0	1	1	0	1	0	1	0	1	
17	**13**	0	1	1	1	0	1	1	0	0	1	
16	**14**	1	1	1	1	0	1	1	1	1	1	Punzado
14	**15**	1	0	0	0	1	1	1	1	0	0	Punzado
13	**16**	0	1	0	0	1	1	1	1	0	1	
12	**17**	1	1	0	0	1	1	1	0	1	0	
11	**18**	0	0	1	0	1	1	0	1	1	1	
10	**19**	1	0	1	0	1	1	0	1	0	1	
9	**20**	0	1	1	0	1	1	0	0	1	1	Punzado
8	**21**	1	1	1	0	1	1	0	1	1	1	Punzado
7	**22**	0	0	0	1	1	1	0	1	0	0	
6	**23**	1	0	0	1	1	1	1	1	0	1	
5	**24**	0	1	0	1	1	1	1	0	1	0	Punzado
4	**25**	1	1	0	1	1	1	1	0	0	1	
3	**26**	0	0	1	1	1	1	0	0	1	0	
2	**27**	1	0	1	1	1	1	1	1	0	0	
1	**28**	0	1	1	1	1	1	1	1	1	0	Punzado
0	**29**	1	1	1	1	1	1	1	1	1	1	Punzado
31	**30**	0	0	0	0	0	1	0	0	0	0	Punzado
15	**31**	0	0	0	0	1	1	1	0	0	0	

Como se ha descrito anteriormente, solamente el orden de las filas de la Tabla 6 es distinto al de la Tabla 4, pero las restantes características de la Tabla 6 son exactamente iguales a las de la Tabla 4. El procedimiento de representación de la Tabla 6 tiene la ventaja de que los últimos 2 bits son punzados durante el tiempo de punción desde el código (20, 10) hasta el código (18, 10).

A continuación, se describirá en detalle, en adelante en la presente memoria, un procedimiento para extender el precitado código (20, 10) a un máximo del código (20, 14).

En cuanto al código de bloque (20, A), según esta realización, se supone que un valor de CQI que indica la información de calidad de canal (CQI) del sistema de LTE de 3GPP es aplicado al procedimiento de codificación de canal para la transmisión del PUCCH. Además, en el caso de generar el código (20, 10), el número de bits de la información de CQI del sistema de LTE de 3GPP puede ser decidido en la gama entre 4 bits hasta 10 bits, de modo que pueda ser extendido máximamente hasta el código de bloque (20, 10). Sin embargo, en el caso del sistema de MIMO (Múltiples Entradas y Múltiples Salidas), el número de bit de la información de CQI puede ser mayor que 10 bits, según sea necesario. Sin embargo, la magnitud de una transmisión (Tx) de CQI efectiva se decide según un procedimiento de generación de CQI. Para el proceso de codificación, la presente invención puede

considerar un procedimiento para prestar soporte a todos los números de bits de información que oscilan entre 4 bits y 14 bits.

Por lo tanto, se describirá a continuación en la presente memoria el procedimiento de codificación de bloque (20, 14) que es capaz de prestar soporte a un máximo de 14 bits, añadiendo una columna, correspondiente al número de bit de información, al anterior código de bloque (20, 10).

A fin de buscar la columna añadida durante la búsqueda exhaustiva, deben ser llevados a cabo un gran número de cálculos. Por lo tanto, la ejecución de la búsqueda exhaustiva en todos los casos puede ser no efectiva, o indeseable.

En esta etapa, debería observarse que la sexta columna de la Tabla 6 está fijada en “1” y que es usada como una secuencia de base. Por lo tanto, si la columna añadida debe satisfacer la distancia mínima “*d*”, un número mínimo de “0” debe ser igual o mayor que “*d*”. En este ejemplo, el número “0” es igual a una distancia mínima entre palabras de código. Más detalladamente, una diferencia entre la columna añadida y la vieja sexta columna, compuesta por “1”, es indicativa de una distancia entre dos palabras de código, de modo que el número “0” contenido en la columna añadida sea igual a la distancia entre las palabras de código.

En general, un máximo de una distancia mínima disponible para el código (20, 10) es 6. En la presente invención, un mínimo de una máxima distancia disponible para el código (20, 11) correspondiente a una versión extendida del código (20, 10) es 4. Más detalladamente, las características de distancia máxima, o mínima, basadas en diversos números de bit de información de la palabra de código de 20 bits pueden ser representadas por la Tabla 7.

[Tabla 7]

n / k	4	5	6	7	8	9	10	11	12	13	14
20	8	8	8	6	6	6	6	4	4	4	4

Por lo tanto, esta realización de la presente invención proporciona un procedimiento de agregado de columna, que permite que la distancia máxima, o mínima, de la columna añadida sea “4”. Según este procedimiento de agregado de columna, una columna que tenga al menos cuatro valores “0” es añadida a la columna añadida.

A fin de minimizar el número de momentos de búsqueda, se supone que la columna añadida de esta realización incluye 4 valores de “0” (es decir, cuatro valores “0”). De esta manera, si la columna añadida incluye los cuatro valores “0” y 16 valores de “1”, esta columna añadida puede ser configurada de varias maneras. Un ejemplo representativo de la columna añadida se muestra en la siguiente Tabla 8. Si el código (20, 10) de la Tabla 6 es extendido al código (20, 14), puede obtenerse la Tabla 8.

[Tabla 8]

i	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$	$M_{i,10}$	$M_{i,11}$	$M_{i,12}$	$M_{i,13}$
0	1	0	0	0	0	1	0	0	0	0	1	1	0	0
2	1	1	0	0	0	1	0	0	0	1	1	1	0	0
3	0	0	1	0	0	1	1	0	1	1	1	1	1	1
5	0	1	1	0	0	1	0	0	1	0	1	1	1	1
6	1	1	1	0	0	1	0	1	0	0	1	1	1	1
8	1	0	0	1	0	1	1	1	1	0	1	1	1	0
9	0	1	0	1	0	1	1	0	1	1	1	1	1	0
11	0	0	1	1	0	1	0	1	1	0	1	1	1	1
12	1	0	1	1	0	1	0	1	0	1	1	1	1	1
13	0	1	1	1	0	1	1	0	0	1	1	1	1	1
16	0	1	0	0	1	1	1	1	0	1	1	1	1	1
17	1	1	0	0	1	1	1	0	1	0	1	1	1	1
18	0	0	1	0	1	1	0	1	1	1	1	1	1	1
19	1	0	1	0	1	1	0	1	0	1	1	1	1	1
22	0	0	0	1	1	1	0	1	0	0	1	0	1	1
23	1	0	0	1	1	1	1	1	0	1	1	0	1	1
25	1	1	0	1	1	1	1	0	0	1	0	1	1	1
26	0	0	1	1	1	1	0	0	1	0	0	1	1	1
27	1	0	1	1	1	1	1	1	0	0	0	0	0	1
31	0	0	0	0	1	1	1	0	0	0	0	0	0	1

Con referencia a la Tabla 8, las cuatro columnas añadidas son indicativas de cuatro columnas situadas en el lado derecho. En cada columna añadida, el “0” está indicado por un trazo en negrilla.

En base a los hechos precitados, se describirá en detalle un procedimiento para modificar u optimizar la Tabla 8, a continuación en la presente memoria.

Según esta realización, la sexta columna de la Tabla 8, es decir, todos los bits de la columna $M_{i,5}$, se fija en “1”, de modo que la sexta columna sea considerada una secuencia de base. Esta secuencia de base contribuye en gran medida a todas las palabras de código de un bit correspondiente. Por lo tanto, desde el punto de vista del correspondiente bit, el uso de la secuencia de base con muchos pesos puede ser deseable.

Sin embargo, varios bits en todas las palabras de código son objeto de la operación O exclusivo (XOR), por lo que debe ser considerado el resultado de su combinación. Por tanto, a fin de reducir el número de los casos de combinación, la secuencia de base en la cual todos los bits tienen el valor “1” se desplaza a la columna más al frente, dando como resultado el aumento de una tasa de contribución. De esta manera, si los datos son codificados con un pequeño número de bits (es decir, un número pequeño de bit), la presente invención puede considerar el procedimiento anterior para desplazar la secuencia de base en la cual todos los bits son “1” a la columna más al frente, y el resultado desplazado está representado por la siguiente Tabla 9.

[Tabla 9]

i	$M_{i.0}$	$M_{i.1}$	$M_{i.2}$	$M_{i.3}$	$M_{i.4}$	$M_{i.5}$	$M_{i.6}$	$M_{i.7}$	$M_{i.8}$	$M_{i.9}$	$M_{i.10}$	$M_{i.11}$	$M_{i.12}$	$M_{i.13}$
0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
1	1	1	1	0	0	0	0	0	0	1	1	1	0	0
2	1	0	0	1	0	0	1	0	1	1	1	1	1	1
3	1	0	1	1	0	0	0	0	1	0	1	1	1	1
4	1	1	1	1	0	0	0	1	0	0	1	1	1	1
5	1	1	0	0	1	0	1	1	1	0	1	1	1	0
6	1	0	1	0	1	0	1	0	1	1	1	1	1	0
7	1	0	0	1	1	0	0	1	1	0	1	1	1	1
8	1	1	0	1	1	0	0	1	0	1	1	1	1	1
9	1	0	1	1	1	0	1	0	0	1	1	1	1	1
10	1	0	1	0	0	1	1	1	0	1	1	1	1	1
11	1	1	1	0	0	1	1	0	1	0	1	1	1	1
12	1	0	0	1	0	1	0	1	1	1	1	1	1	1
13	1	1	0	1	0	1	0	1	0	1	1	1	1	1
14	1	0	0	0	1	1	0	1	0	0	1	0	1	1
15	1	1	0	0	1	1	1	1	0	1	1	0	1	1
16	1	1	1	0	1	1	1	0	0	1	0	1	1	1
17	1	0	0	1	1	1	0	0	1	0	0	1	1	1
18	1	1	0	1	1	1	1	1	0	0	0	0	0	1
19	1	0	0	0	0	1	1	0	0	0	0	0	0	1

Con referencia a la Tabla 9, una sexta secuencia (es decir, cada bit de la sexta secuencia tiene el valor “1”) entre las secuencias de base originales con dirección de columna, se desplaza a la ubicación de una primera secuencia de base, y el orden de las otras secuencias no se cambia por otro.

En la siguiente descripción, se supone que la estructura de código de bloque (20, 14) para la transmisión del PUCCH usa el código de bloque de la Tabla 9. Si el número de bits de información para la transmisión del PUCCH está limitado a un máximo de 13 bits, o menos, la secuencia de base situada en la parte más a la derecha de la Tabla 9 puede ser omitida según sea necesario, y el resultado omitido puede ser representado por la siguiente Tabla 10.

[Tabla 10]

i	$M_{i.0}$	$M_{i.1}$	$M_{i.2}$	$M_{i.3}$	$M_{i.4}$	$M_{i.5}$	$M_{i.6}$	$M_{i.7}$	$M_{i.8}$	$M_{i.9}$	$M_{i.10}$	$M_{i.11}$	$M_{i.12}$
0	1	1	0	0	0	0	0	0	0	0	1	1	0
1	1	1	1	0	0	0	0	0	0	1	1	1	0
2	1	0	0	1	0	0	1	0	1	1	1	1	1
3	1	0	1	1	0	0	0	0	1	0	1	1	1
4	1	1	1	1	0	0	0	1	0	0	1	1	1
5	1	1	0	0	1	0	1	1	1	0	1	1	1
6	1	0	1	0	1	0	1	0	1	1	1	1	1
7	1	0	0	1	1	0	0	1	1	0	1	1	1
8	1	1	0	1	1	0	0	1	0	1	1	1	1
9	1	0	1	1	1	0	1	0	0	1	1	1	1
10	1	0	1	0	0	1	1	1	0	1	1	1	1
11	1	1	1	0	0	1	1	0	1	0	1	1	1
12	1	0	0	1	0	1	0	1	1	1	1	1	1
13	1	1	0	1	0	1	0	1	0	1	1	1	1
14	1	0	0	0	1	1	0	1	0	0	1	0	1
15	1	1	0	0	1	1	1	1	0	1	1	0	1
16	1	1	1	0	1	1	1	0	0	1	0	1	1
17	1	0	0	1	1	1	0	0	1	0	0	1	1
18	1	1	0	1	1	1	1	1	0	0	0	0	0
19	1	0	0	0	0	1	1	0	0	0	0	0	0

Ahora, en base a la descripción precitada, se describirá en detalle un procedimiento para generar el código de bloque (n, k) (donde n $\leq$ 32 y k $\leq$ 14), a continuación en la presente memoria.

<u>Código de bloque (n, k) (n < 32, k < 14)</u>

Un código de bloque con un tamaño máximo de (32, 14) se describirá en detalle a continuación en la presente memoria. En otras palabras, según esta realización, un tamaño máximo del número de bit de codificación es 32, y un tamaño máximo del número de bit de información es 14. El diseño de codificación puede ser implementado de varias maneras, pero un procedimiento de codificación según esta realización debe estar diseñado para buscar un número máximo de puntos comunes con los códigos convencionales.

A fin de generar el código de bloque (32, 14), la presente invención considera el código de bloque (20, 14) (Véase la Tabla 9) obtenido a partir del código de bloque de TFCI (32, 10) del 3GPP Edición 99, y a la vez considera el precitado código de bloque de TFCI (32, 10). A fin de generar el código de bloque (32, 14) usando el código de bloque (20, 14) y el código de bloque de TFCI (32, 10), una parte “TBD” (A definir) de la FIG. 1 necesita ser definida adicionalmente.

La FIG. 1 es un diagrama conceptual que ilustra un procedimiento para generar el código de bloque (32, 14) usando el código de bloque (32, 10) usado para una codificación convencional de información de TFCI y el código (20, 14) usado para la transmisión del PUCCH, según la presente invención.

Con referencia a la FIG. 1, si el código 101 de bloque (32, 10), usado para la codificación de información de TFCI, y el código 102 de bloque (20, 14) son usados para generar el código 104 de bloque (32, 14), la parte 103 TBD debe ser definida adicionalmente. En un aspecto de cualquiera de los códigos de bloque, la precitada definición puede ser analizada de varias maneras. Esto es, el código 104 de bloque (32, 14) según esta realización es generado cuando cuatro secuencias de base (correspondientes a la parte de combinación entre la parte 102a y la parte 103 de la FIG. 1) son añadidas al lado derecho del código convencional de bloque (32, 10). En otro aspecto de los anteriores códigos de bloque, 12 secuencias con dirección de fila (correspondientes a la parte de

combinación entre la parte 101a y la parte 103 de la FIG. 1) son añadidas, bien al código de bloque de la Tabla 9 o bien a su equivalente el código 102 de bloque (20, 14), de modo que el código 104 de bloque (32, 14) también pueda ser generado.

En este caso, el código convencional de información de TFCI mostrado en la Tabla 1, y su código equivalente, también puede ser usado como el código 101 de bloque (32, 10). El código de bloque de la Tabla 9 y su código equivalente también pueden ser usados como el código 102 de bloque (20, 14). En este caso, si la ubicación entre las filas y / o la ubicación entre las columnas del código de bloque convencional es / son cambiada(s) por otra(s), se forma el anterior código equivalente.

Preferiblemente, si el código es diseñado, el código diseñado debe permitir que la parte TBD 103 tenga las mejores prestaciones a una distancia mínima. En general, según diversas longitudes de bits de información y diversas longitudes de bits de codificación, se obtienen las siguientes prestaciones de distancia mínima, según se muestra en la Tabla 11.

[Tabla 11]

N / k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
20	20	10	8	8	8	8	6	6	6	6	4	4	4	4
32	32	16	16	16	16	16	12	12	12	12	10	10	8	8

Con referencia a la Tabla 11, en el caso de la codificación de bloque (32, A), si “A” es mayor que “10”, un valor máximo de una distancia mínima de Hamming de la secuencia de base está limitado a “10”.

Por lo tanto, según una realización preferida de la presente invención, la parte 103 TBD de la FIG. 1, junto con la parte 102a del código 102 de bloque (20, 14), debe permitir que un valor máximo de la mínima distancia de Hamming de cada secuencia de base (correspondiente a la parte de combinación entre la parte 102a y la parte 103 de la FIG. 1) sea “10”. En este caso, la parte 102a corresponde a los bits de información de más de 10 bits. Con más detalle, si el número de secuencias de base adicionales es 1 o 2 (es decir, si “A” es 11 o 12), esto significa que una o dos secuencias de base son añadidas para permitir que una distancia mínima de Hamming de cada secuencia de base sea “10”. Si el número de secuencias de base adicionales es 3 o 4 (es decir, si “A” es 13 o 14), esto significa que tres o cuatro secuencias de base son añadidas para permitir que una distancia mínima de Hamming de cada secuencia de base sea “8”. El código de bloque (32, 10) para la codificación de información de TFCI incluye una secuencia de base en la cual todos y cada uno de los componentes son “1”, por lo que cada una de las secuencias de base adicionales puede incluir 10 valores “0”. En otras palabras, si el código de bloque (20, 14) es usado como el código de bloque de la Tabla 9, las secuencias de base individuales correspondientes a la parte 102a de la FIG. 1 incluyen 4 valores “0”, de modo que la parte de la secuencia de base correspondiente a la parte TBD pueda incluir 6 valores “0”.

Un ejemplo para satisfacer la condición precitada se muestra en la siguiente Tabla 12.

[Tabla 12]

i	TFCI	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$	$M_{i,10}$	$M_{i,11}$	$M_{i,12}$	$M_{i,13}$
0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
1	2	1	1	1	0	0	0	0	0	0	1	1	1	0	0
2	3	1	0	0	1	0	0	1	0	1	1	1	1	1	1
3	5	1	0	1	1	0	0	0	0	1	0	1	1	1	1
4	6	1	1	1	1	0	0	0	1	0	0	1	1	1	1
5	8	1	1	0	0	1	0	1	1	1	0	1	1	1	0
6	9	1	0	1	0	1	0	1	0	1	1	1	1	1	0
7	11	1	0	0	1	1	0	0	1	1	0	1	1	1	1
8	12	1	1	0	1	1	0	0	1	0	1	1	1	1	1
9	13	1	0	1	1	1	0	1	0	0	1	1	1	1	1
10	16	1	0	1	0	0	1	1	1	0	1	1	1	1	1
11	17	1	1	1	0	0	1	1	0	1	0	1	1	1	1
12	18	1	0	0	1	0	1	0	1	1	1	1	1	1	1
13	19	1	1	0	1	0	1	0	1	0	1	1	1	1	1
14	22	1	0	0	0	1	1	0	1	0	0	1	0	1	1
15	23	1	1	0	0	1	1	1	1	0	1	1	0	1	1
16	25	1	1	1	0	1	1	1	0	0	1	0	1	1	1
17	26	1	0	0	1	1	1	0	0	1	0	0	1	1	1
18	27	1	1	0	1	1	1	1	1	0	0	0	0	0	1
19	31	1	0	0	0	0	1	1	0	0	0	0	0	0	1
20	1	1	0	1	0	0	0	1	0	0	0	1	0	1	1
21	4	1	1	0	1	0	0	0	0	0	1	1	1	0	0
22	7	1	0	0	0	1	0	0	1	1	0	1	0	1	1
23	10	1	1	1	0	1	0	0	0	1	1	1	1	0	1
24	14	1	1	1	1	1	0	1	1	1	1	0	0	1	0
25	15	1	1	0	0	0	1	1	1	0	0	1	1	1	0
26	20	1	0	1	1	0	1	0	0	1	1	0	0	0	1
27	21	1	1	1	1	0	1	0	1	1	1	0	1	0	0
28	24	1	0	1	0	1	1	1	0	1	0	0	1	0	1
29	28	1	0	1	1	1	1	1	1	1	0	0	1	1	0
30	29	1	1	1	1	1	1	1	1	1	1	1	0	1	0
31	30	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Como puede apreciarse en la Tabla 12, si 12 filas del extremo inferior de la Tabla 12 son eliminadas, se forma el código (20, 14) para la transmisión del PUCCH. Si 14 filas del extremo inferior de la Tabla 12 son eliminadas, se forma el código (18, 14). La aplicación variable del bit de información puede ser implementada fácilmente, y se obtienen de la Tabla 12 tantas secuencias de base de la Tabla 12 como el número de los correspondientes bits de información, de modo que las secuencias de base obtenidas sean usadas para el proceso de codificación. Si un número máximo de bits de información (es decir, un máximo número de bit de información) de la FIG. 12 es menor que 14 (es decir, 14 bits), tantas secuencias de base como el número predeterminado, innecesarias para la tabla de secuencias de base, tal como la Tabla 9, pueden ser eliminadas de la columna derecha. Esto significa que pueden ser añadidas tantas secuencias de base como el número de bits de información requeridos para el código

de bloque (32, 10).

Por ejemplo, a condición de que un máximo número de bit de información está limitado a 11 bits, puede usarse el siguiente código de bloque (32, 11) de la Tabla 13.

[Tabla 13]

i	TFCI	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$	$M_{i,10}$
0	0	1	1	0	0	0	0	0	0	0	0	1
1	2	1	1	1	0	0	0	0	0	0	1	1
2	3	1	0	0	1	0	0	1	0	1	1	1
3	5	1	0	1	1	0	0	0	0	1	0	1
4	6	1	1	1	1	0	0	0	1	0	0	1
5	8	1	1	0	0	1	0	1	1	1	0	1
6	9	1	0	1	0	1	0	1	0	1	1	1
7	11	1	0	0	1	1	0	0	1	1	0	1
8	12	1	1	0	1	1	0	0	1	0	1	1
9	13	1	0	1	1	1	0	1	0	0	1	1
10	16	1	0	1	0	0	1	1	1	0	1	1
11	17	1	1	1	0	0	1	1	0	1	0	1
12	18	1	0	0	1	0	1	0	1	1	1	1
13	19	1	1	0	1	0	1	0	1	0	1	1
14	22	1	0	0	0	1	1	0	1	0	0	1
15	23	1	1	0	0	1	1	1	1	0	1	1
16	25	1	1	1	0	1	1	1	0	0	1	0
17	26	1	0	0	1	1	1	0	0	1	0	0
18	27	1	1	0	1	1	1	1	1	0	0	0
19	31	1	0	0	0	0	1	1	0	0	0	0
20	1	1	0	1	0	0	0	1	0	0	0	1
21	4	1	1	0	1	0	0	0	0	0	1	1
22	7	1	0	0	0	1	0	0	1	1	0	1
23	10	1	1	1	0	1	0	0	0	1	1	1
24	14	1	1	1	1	1	0	1	1	1	1	0
25	15	1	1	0	0	0	1	1	1	0	0	1
26	20	1	0	1	1	0	1	0	0	1	1	0
27	21	1	1	1	1	0	1	0	1	1	1	0
28	24	1	0	1	0	1	1	1	0	1	0	0
29	28	1	0	1	1	1	1	1	1	1	0	0
30	29	1	1	1	1	1	1	1	1	1	1	1
31	30	1	0	0	0	0	0	0	0	0	0	0

Mientras tanto, se describirá en detalle, a continuación en la presente memoria, un procedimiento para disponer el código de bloque (32, k), en consideración del código de bloque (16, k).

Un procedimiento para generar un código óptimo del código de bloque (32, k) de la Tabla 12, o la Tabla 13, es el

siguiente. En el caso de generar el código de bloque (20, k) o (18, k), si las filas restantes de la fila más baja son eliminadas, puede generarse el anterior código óptimo del código de bloque (32, k). Sin embargo, si se necesita el código de bloque (16, k), se necesita mucha consideración. Por lo tanto, según esta realización de la presente invención, el orden de las filas del anterior código de bloque (32, k) se cambia en consideración del código de bloque (16, k). Si se necesita el código de bloque (16, k), aunque algunas filas de la fila más baja del código de bloque (32, k) sean eliminadas y usadas, la presente invención es capaz de generar un código necesario. Un ejemplo representativo se muestra en la siguiente Tabla 14.

[Tabla 14]

i	TFCI	$M_{i.0}$	$M_{i.1}$	$M_{i.2}$	$M_{i.3}$	$M_{i.4}$	$M_{i.5}$	$M_{i.6}$	$M_{i.7}$	$M_{i.8}$	$M_{i.9}$	$M_{i.10}$	$M_{i.11}$	$M_{i.12}$	$M_{i.13}$
0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
1	2	1	1	1	0	0	0	0	0	0	1	1	1	0	0
2	3	1	0	0	1	0	0	1	0	1	1	1	1	1	1
3	5	1	0	1	1	0	0	0	0	1	0	1	1	1	1
4	6	1	1	1	1	0	0	0	1	0	0	1	1	1	1
6	9	1	0	1	0	1	0	1	0	1	1	1	1	1	0
8	12	1	1	0	1	1	0	0	1	0	1	1	1	1	1
9	13	1	0	1	1	1	0	1	0	0	1	1	1	1	1
10	16	1	0	1	0	0	1	1	1	0	1	1	1	1	1
11	17	1	1	1	0	0	1	1	0	1	0	1	1	1	1
12	18	1	0	0	1	0	1	0	1	1	1	1	1	1	1
13	19	1	1	0	1	0	1	0	1	0	1	1	1	1	1
15	23	1	1	0	0	1	1	1	1	0	1	1	0	1	1
16	25	1	1	1	0	1	1	1	0	0	1	0	1	1	1
17	26	1	0	0	1	1	1	0	0	1	0	0	1	1	1
19	31	1	0	0	0	0	1	1	0	0	0	0	0	0	1
7	11	1	0	0	1	1	0	0	1	1	0	1	1	1	1
14	22	1	0	0	0	1	1	0	1	0	0	1	0	1	1
5	8	1	1	0	0	1	0	1	1	1	0	1	1	1	0
18	27	1	1	0	1	1	1	1	1	0	0	0	0	0	1
20	1	1	0	1	0	0	0	1	0	0	0	1	0	1	1
21	4	1	1	0	1	0	0	0	0	0	1	1	1	0	0
22	7	1	0	0	0	1	0	0	1	1	0	1	0	1	1
23	10	1	1	1	0	1	0	0	0	1	1	1	1	0	1
24	14	1	1	1	1	1	0	1	1	1	1	0	0	1	0
25	15	1	1	0	0	0	1	1	1	0	0	1	1	1	0
26	20	1	0	1	1	0	1	0	0	1	1	0	0	0	1
27	21	1	1	1	1	0	1	0	1	1	1	0	1	0	0
28	24	1	0	1	0	1	1	1	0	1	0	0	1	0	1
29	28	1	0	1	1	1	1	1	1	1	0	0	1	1	0
30	29	1	1	1	1	1	1	1	1	1	1	1	0	1	0
31	30	1	0	0	0	0	0	0	0	0	0	0	0	0	1

En la Tabla 14, durante la conversión de código desde el código (32, k) al código (20, k), si se eliminan 12 filas de la Tabla 14 en base a la fila más baja, se genera un código óptimo. Si se eliminan 14 filas de la Tabla 14 en base a la fila más baja, se genera el código (18, k). Si se eliminan 16 filas de la Tabla 14 en base a la fila más baja, se genera el código (16, k).

Mientras tanto, en base a la descripción precitada, si se necesita una palabra de código de más de 32 bits, puede llevarse a cabo el siguiente procedimiento de codificación de canal.

Si se necesita una palabra de código de más de 32 bits:

Si se necesita una palabra de código de más de 32 bits, la presente invención proporciona un procedimiento para generar el código (n, k) de longitud larga, repitiendo la anterior secuencia de base sobre la base del código (32, k) o (20, k) usado como un código de base.

El código (32, k) o (20, k) puede ser fácilmente generado sobre la base de la Tabla 12 o la Tabla 14. Mientras tanto, a fin de asignar una capacidad de corrección de errores más potente, o más alta, a un bit de transmisión (Tx), la presente invención puede aumentar el número de bits de codificación (es decir, el número de bit de codificación). En este caso, es preferible que sea generado un nuevo código, correspondiente al número aumentado de bits de codificación, pero es muy difícil para un diseñador de código diseñar un nuevo código toda vez que aumenta el número de bits de codificación. Por lo tanto, uno de los procedimientos sencillos de generación es repetir el código de base en una longitud deseada. Si la longitud deseada no está exactamente representada por un múltiplo entero del código de base, el código de base se repite en al menos la longitud deseada, y pueden eliminarse tantos códigos de base como el número de bits en exceso. En este caso, aunque la presente invención es capaz de buscar un patrón óptimo de punción cada vez, la presente invención puede considerar esencialmente un procedimiento sencillo de punción en base a un bloque de correlación de tasa.

En este caso, la presente invención puede considerar el código (32, k) o (20, k) como el código de base. Por conveniencia de la descripción, la presente invención considera solamente un caso específico en el cual una longitud deseada tiene el tamaño de un múltiplo entero del código de base. En los casos restantes, se supone que la presente invención es capaz de obtener un código necesario usando el procedimiento de punción. Además, la presente invención es capaz de usar una cierta variedad de valores de “k”. En este caso, por conveniencia de la descripción, se supone que el tamaño máximo de “k” está fijado en 14. Aunque no hay ninguna mención del tamaño más pequeño, menor que 14, en la presente invención, es bien conocido para los expertos en la técnica que se selecciona y usa una secuencia de base correspondiente a una longitud correspondiente.

Por ejemplo, si se necesita el código (64, 14), el código (32, 14) puede ser repetido solamente dos veces. Si se necesita el código (40, 14), el código (20, 14) puede ser repetido solamente dos veces. Además, el código (32, 14) y el código (20, 14) son considerados simultáneamente como el código de base, por lo que el código (32, 14) y el código (20, 14) son adosados secuencialmente para configurar el código (52, 14).

Para concluir, puede determinarse que la combinación de códigos disponibles sea el código (a*32+b*20, 14) (donde “a” o “b” es un número entero mayor o igual que “0”.

Si el último número necesario de bit de codificación no está indicado por un múltiplo entero del código de base, el código de base se repite para que sea más largo que la longitud deseada, las partes innecesarias pueden ser cortadas de la parte final del código de tamaño más largo, o pueden ser punzadas usando los bloques de correlación de tasa.

Mientras tanto, según otro aspecto de la presente invención, después de que el orden de los bits de información ha sido invertido, puede repetirse a continuación el código de base.

Según lo descrito anteriormente, si el código de base se repite continuamente, las características de distancia mínima del código de base se mantienen sin ningún cambio, por lo que se repiten las características resultantes de distancia mínima. Por lo tanto, si el código con una distancia mínima original de 4 se repite dos veces, la distancia mínima aumenta dos veces, por lo que la distancia mínima del código resultante llega a “8”.

Sin embargo, a condición de que cualquier variación sea aplicada al bit de información durante la repetición anterior, aunque se use el bit de información que forma una palabra de código con una distancia mínima, una palabra de código generada por otro bit de información, cambiado por la repetición anterior, es capaz de formar otra palabra de código con la distancia más larga que la precitada distancia mínima. Si la distancia mínima no se repite según los principios precitados, puede diseñarse un código de modo que las características de distancia mínima puedan ser mayores que las de un múltiplo simple.

La presente invención puede considerar una cierta variedad de procedimientos para cambiar el bit de información. Por ejemplo, a fin de cambiar el bit de información precitado, puede usarse un procedimiento para usar una

versión invertida de una unidad de bit, y un procedimiento para permitir que el bit de información atraviese una secuencia aleatoria, tal como una secuencia de PN (Seudo Ruido), en la presente invención.

Además, la presente invención puede asignar distintas variaciones del bit de información a tiempos individuales de repetición, de modo que el bit de información sea cambiado de manera distinta toda vez que ocurre la repetición. Sin embargo, en este caso, aumenta la complejidad de un final de transmisión / recepción. Si la repetición es realizada varias veces, la presente invención puede no cambiar el bit de información la primera vez, y puede cambiar el bit de información la próxima vez.

Con más detalle, en el caso de repetir el código según esta realización, el bit de información no se cambia en la repetición par, sino que se cambia en la repetición impar. En otras palabras, la presente invención controla la variación del bit de información para que sea alternado toda vez que ocurre la repetición.

Por ejemplo, a condición de que el código (20, k) y el código (32, k) sean usados como los códigos de base, el código (40, k), el código (52, k) y el código (64, k) pueden ser generados según la presente invención. Una descripción detallada de los mismos se describirá a continuación en la presente memoria.

La FIG. 2 es un gráfico que ilustra prestaciones de distancia mínima obtenidas cuando el código de bloque (40, k), el código de bloque (52, k) y el código de bloque (64, k) son generados, a condición de que el código de bloque (20, k) y el código de bloque (32, k) sean usados como códigos de base.

Con referencia a la FIG. 2, “(40, k)_20+20” indica que el código (20, k) que actúa como el código de base se repite dos veces. “(20, k)_20rev” indica que el bit de información es invertido a partir del código (20, k) que actúa como el código de base. “(32, k)_32rev” puede ser analizado en el mismo procedimiento que el procedimiento precitado.

“(40, k)_20+20rev” indica que el código (20, k) usado como el código de base fue repetido dos veces, y que un bit de información fue cambiado según el procedimiento de alternación en el primer momento de repetición, y el bit de información fue a continuación invertido en el segundo momento de repetición. De esta manera, “(64, k)_32+32rev” también puede ser analizado en el mismo procedimiento que el anterior. Mientras tanto, “(52, k)_0+32rev” indica que los códigos (32, k) y (20, k) fueron seleccionados como códigos de base, que cada uno de los códigos de base seleccionados fue repetido solamente una vez y que el código (32, k) que actúa como el segundo código de base tuvo sus bits invertidos y que el bit de información fue a continuación aplicado al resultado de bits invertidos.

En la FIG. 2, si el valor de k está indicado por $k \geq 4$, el código “(40, k)_20” tiene prestaciones bajas, menores que las del código (32, k). Por lo tanto, puede reconocerse que la operación de usar el código (32, k) con un número menor de bits de codificación tiene buenas prestaciones, mayores que las de la operación de repetir el código (20, k) dos veces. En otras palabras, la repetición del código (32, k) con un número menor de bits de codificación tiene altas prestaciones, superiores a las de la repetición del código (20, k). Por lo tanto, la realización preferida de la presente invención proporciona un procedimiento para usar repetidamente el código (32, k) durante el proceso de codificación para un bit de codificación de longitud larga. En otras palabras, según esta realización de la presente invención, a condición de que el código (32, k) fuera seleccionado como el código de base, el número de bit (es decir, el número de bits) de la última secuencia de salida fuera igual o mayor que 32 (es decir, 32 bits) y que un múltiplo entero del código de base no fuera implementado, el código (32, k) se repite para que sea más largo que una longitud deseada, y las partes innecesarias (es decir, las correspondientes al número de bit de la secuencia de salida necesaria) son cortadas del final del resultado repetido.

La realización precitada puede ser analizada de la siguiente manera. Con más detalle, si la longitud de la secuencia de salida necesaria es de al menos 32 bits, puede reconocerse que el resultado codificado del código de bloque (32, k) fue repetido cíclicamente para obtener la secuencia anterior de salida. Esto es, a condición de que la palabra de código de 32 bits de longitud, que ha sido codificada en el canal por el código de bloque (32, k), está representada por b_0, b_1, b_2, b_3, ..., b_{B-1} (donde B = 32), y que la secuencia de salida más larga que la longitud de 32 bits (p. ej., la longitud de bits de “Q”) está representada por q_0, q_1, q_2, q_3, ..., q_{Q-1}, la siguiente relación entre la secuencia de salida y la palabra de código de 32 bits de longitud, codificada en el canal, puede ser representada por la siguiente Ecuación 3.

[Ecuación 3]

$$q_i = b_{(i \bmod B)}, \text{ donde } i = 0, 1, 2, ..., Q\text{-}1$$

Como puede apreciarse en la Ecuación 3, el componente de secuencia de salida con el índice “*i*” corresponde a un componente de palabra de código que tiene un índice correspondiente al valor del resultado de la operación módulo. En este caso, el valor del resultado de la operación módulo se obtiene cuando el índice “*i*” fue objeto de la operación módulo con el valor 32 de “B”. Si el valor de Q es mayor que 32, la secuencia de salida se obtiene cuando la secuencia codificada en el canal fue repetida cíclicamente. Esto también significa que el código (32, k) fue repetido un número predeterminado de veces y que fue seleccionada y usada la parte correspondiente a la

longitud de una palabra de código necesaria. Como es bien conocido para los expertos en la técnica, los resultados de la operación precitada han sido esencialmente iguales entre sí, pero han sido analizados de distintas maneras.

Mientras tanto, como puede apreciarse en la FIG. 2, las prestaciones del código “(52, k)_20+32rev” son iguales o mayores que las del código “(52, k)_20+32”, y las prestaciones del código “(40, k)_20+20rev” son iguales o mayores que las del “(40, k)_20+20”. Por lo tanto, a fin de mejorar las características de distancia mínima, el bit de información se repite sin ningún cambio en el primer momento de repetición, y a continuación es invertido en el siguiente momento de repetición.

Finalmente, si el número del último bit de información no está indicado por un múltiplo entero del código de base, el código de base se repite para que sea más largo que una longitud deseada, y las partes innecesarias pueden ser cortadas de la parte final del resultado repetido, o pueden ser punzadas por el procedimiento de bloques de correlación de tasa.

Mientras tanto, el procedimiento precitado para realizar la inversión de bit del bit de información, según el esquema de alternación, también puede ser aplicado, no solamente a la repetición del código de base precitado, sino también a otras repeticiones de diversos códigos de base. Por ejemplo, cuando se realiza la codificación repetitiva del código simplex de la información de control ACK / NACK, el bit de información es codificado sin ningún cambio en el primer momento de repetición, y el bit de información de la inversión de bits de la información de control ACK / NACK es a continuación codificado en el segundo momento de repetición.

Deberá observarse que la mayor parte de la terminología revelada en la presente invención está definida en consideración de funciones de la presente invención, y puede ser determinada de manera distinta según la intención de los expertos en la técnica, o las prácticas usuales. Por lo tanto, es preferible que la terminología precitada sea entendida en base a todos los contenidos revelados en la presente invención. Por ello, se pretende que la presente invención cobra las modificaciones y variaciones de la presente invención a condición de que estén incluidas en el alcance de las reivindicaciones adjuntas y sus equivalentes.

Es decir, la presente invención no está limitada solamente a las realizaciones descritas en la presente memoria, e incluye la más amplia gama equivalente a los principios y características revelados en la presente memoria.

Aplicabilidad industrial

Como es evidente a partir de la descripción anterior, el procedimiento de codificación de canal según la presente invención puede ser fácilmente aplicado a la codificación del otro canal del sistema de LTE de 3GPP, que transmite información de CQI / PMI a un enlace ascendente, mediante un canal PUSCH. Sin embargo, los procedimientos precitados no están limitados solamente al anterior sistema LTE de 3GPP, sino que también pueden ser aplicados a una gran variedad de esquemas de comunicación, cada uno de los cuales realiza la codificación de bloque sobre información de longitud variable.

REIVINDICACIONES

1. Un procedimiento de codificación de canal de bits de entrada, usando un código de bloque (32, A), y un código de bloque (20, O), estando el procedimiento **caracterizado por**:

Cuando se usa el código de bloque (32,A), codificación del canal de bits de entrada usando las secuencias base A, teniendo una longitud cada una de 32 y, cuando se usa el código de bloque (20, O), codificación del canal de bits de entrada usando secuencias básicas O M_{i0} to M_{iO-1}, teniendo cada una una longitud de 20 para generar bits codificados cuya longitud es 20 y transmitir los bits codificados con longitud 20,
en el que las secuencias básicas O son $M_{i,0}$ a $M_{i,10}$ tal como se definen en la siguiente tabla:

i	$M_{I,0}$	$M_{I,1}$	$M_{I,2}$	$M_{I,3}$	$M_{I,4}$	$M_{I,5}$	$M_{I,6}$	$M_{I,7}$	$M_{I,8}$	$M_{I,9}$	$M_{I,10}$	$M_{I,11}$	$M_{I,12}$
0	1	1	0	0	0	0	0	0	0	0	1	1	0
1	1	1	1	0	0	0	0	0	0	1	1	1	0
2	1	0	0	1	0	0	1	0	1	1	1	1	1
3	1	0	1	1	0	0	0	0	1	0	1	1	1
4	1	1	1	1	0	0	0	1	0	0	1	1	1
5	1	1	0	0	1	0	1	1	1	0	1	1	1
6	1	0	1	0	1	0	1	0	1	1	1	1	1
7	1	0	0	1	1	0	0	1	1	0	1	1	1
8	1	1	0	1	1	0	0	1	0	1	1	1	1
9	1	0	1	1	1	0	1	0	0	1	1	1	1
10	1	0	1	0	0	1	1	1	0	1	1	1	1
11	1	1	1	0	0	1	1	0	1	0	1	1	1
12	1	0	0	1	0	1	0	1	1	1	1	1	1
13	1	1	0	1	0	1	0	1	0	1	1	1	1
14	1	0	0	0	1	1	0	1	0	0	1	0	1
15	1	1	0	0	1	1	1	1	0	1	1	0	1
16	1	1	1	0	1	1	1	0	0	1	0	1	1
17	1	0	0	1	1	1	0	0	1	0	0	1	1
18	1	1	0	1	1	1	1	1	0	0	0	0	0
19	1	0	0	0	0	1	1	0	0	0	0	0	0

, en la que las secuencias básicas A son $M_{i,0}$ a $M_{i,A-1}$ como se define en la siguiente tabla:

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
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

en la que O = 11, 12 o 13 y A = 11, y
en la que M_{oo-1} a $M_{19,o-1}$ para O = 11, 12, o 13 tiene cuatro "0", y M_{20A-i} a M_{31A-1} para A = 11 tiene seis "0".

2. El procedimiento de acuerdo con la reivindicación 1, en el que transmitir los bits codificados comprende:

transmitir una secuencia generada que incluye los bits codificados b_0, b_1, b_2, b_3…, b_{B-1}, donde B=32, en el que la secuencia de salida q_0, q_1, q_2, q_3…, q_{Q-1} se genera usando la siguiente ecuación:

$$q_i = b_{(i \bmod 32)^*}$$

donde i = 0, 1, 2…, Q-1 y Q es superior a 32.

3. Un dispositivo para codificación de canal de bits de entrada que usa un código de bloque (32, A) y un código de bloque (20, O), estando el dispositivo **caracterizado por**:

un codificador de canal configurado para:

cuando se usa el código de bloque (32,A), el código del canal de los bits de entrada que usa las secuencias básicas A, teniendo cada una una longitud de 32 para generar bits codificados cuya longitud es 32 y
cuando se usa el código de bloque (20, O), el código del canal de bits de entrada que usan las secuencias básicas $M_{i,0}$ a M_{iO-1}, teniendo cada una una longitud de 20 para generar bits codificados cuya longitud es 20

y

un transmisor configurado para transmitir los bits codificados,
en el que las secuencias básicas O son $M_{i,0}$ a $M_{i,O.1}$, como se define en la siguiente tabla:

i	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$	$M_{i,10}$	$M_{i,11}$	$M_{i,12}$
0	1	1	0	0	0	0	0	0	0	0	1	1	0
1	1	1	1	0	0	0	0	0	0	1	1	1	0
2	1	0	0	1	0	0	1	0	1	1	1	1	1
3	1	0	1	1	0	0	0	0	1	0	1	1	1
4	1	1	1	1	0	0	0	1	0	0	1	1	1
5	1	1	0	0	1	0	1	1	1	0	1	1	1
6	1	0	1	0	1	0	1	0	1	1	1	1	1
7	1	0	0	1	1	0	0	1	1	0	1	1	1
8	1	1	0	1	1	0	0	1	0	1	1	1	1
9	1	0	1	1	1	0	1	0	0	1	1	1	1
10	1	0	1	0	0	1	1	1	0	1	1	1	1
11	1	1	1	0	0	1	1	0	1	0	1	1	1
12	1	0	0	1	0	1	0	1	1	1	1	1	1
13	1	1	0	1	0	1	0	1	0	1	1	1	1
14	1	0	0	0	1	1	0	1	0	0	1	0	1
15	1	1	0	0	1	1	1	1	0	1	1	0	1
16	1	1	1	0	1	1	1	0	0	1	0	1	1
17	1	0	0	1	1	1	0	0	1	0	0	1	1
18	1	1	0	1	1	1	1	1	0	0	0	0	0
19	1	0	0	0	0	1	1	0	0	0	0	0	0

en la que las secuencias básicas A son Mi,0 a Mi,A-1 como se define en la siguiente tabla:

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
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

en la que O = 11, 12 o 13 y A = 11, y
en la que $M_{0,O-1}$ a $M_{19,O-1}$ para O = 11, 12 o 13 tiene cuatro “0” y $M_{20,A-1}$ a $M_{31,A-1}$ para A = 11 tiene seis “0”.

4. El dispositivo de acuerdo con la reivindicación 3, en el que el transmisor se configura para:

transmitir una secuencia generada que incluye los bits codificados b_0, b_1, b_2, b_3…, b_{31}, donde B=32,
en el que la secuencia q_0, q_1, q_2, q_3…, q_{Q-1} se genera usando la siguiente ecuación:

$$q_i = b_{(i \bmod 32)*}$$

donde i = 0, 1, 2…, Q-1 y Q es superior a 32.

FIG. 1

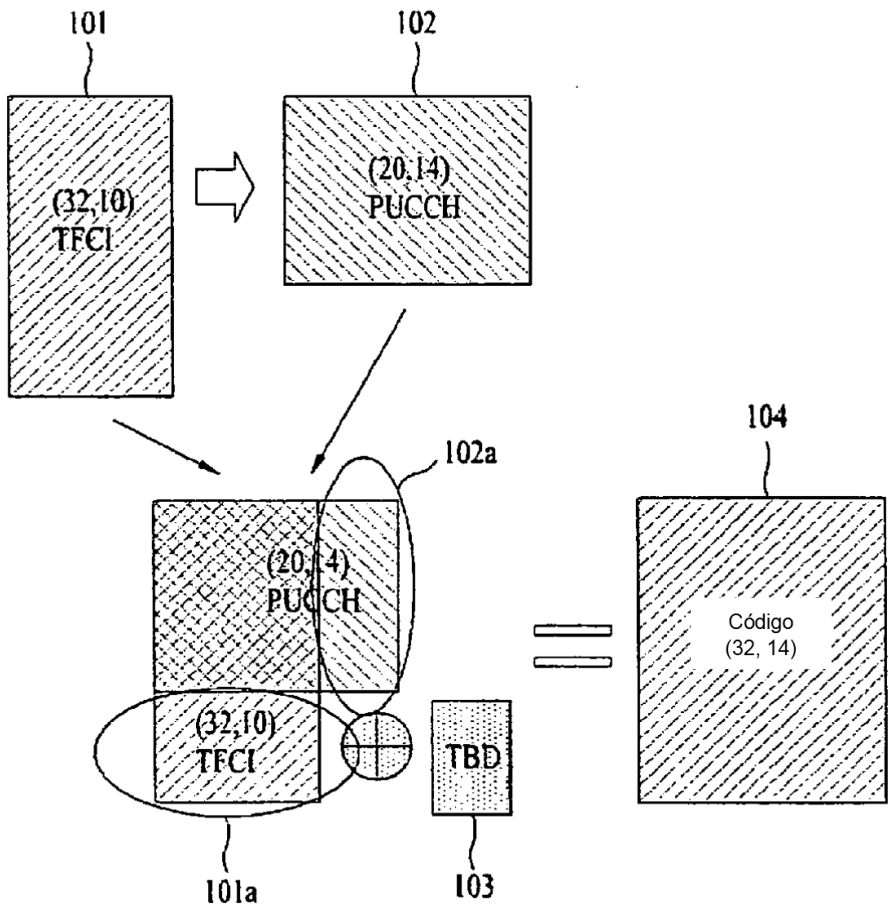
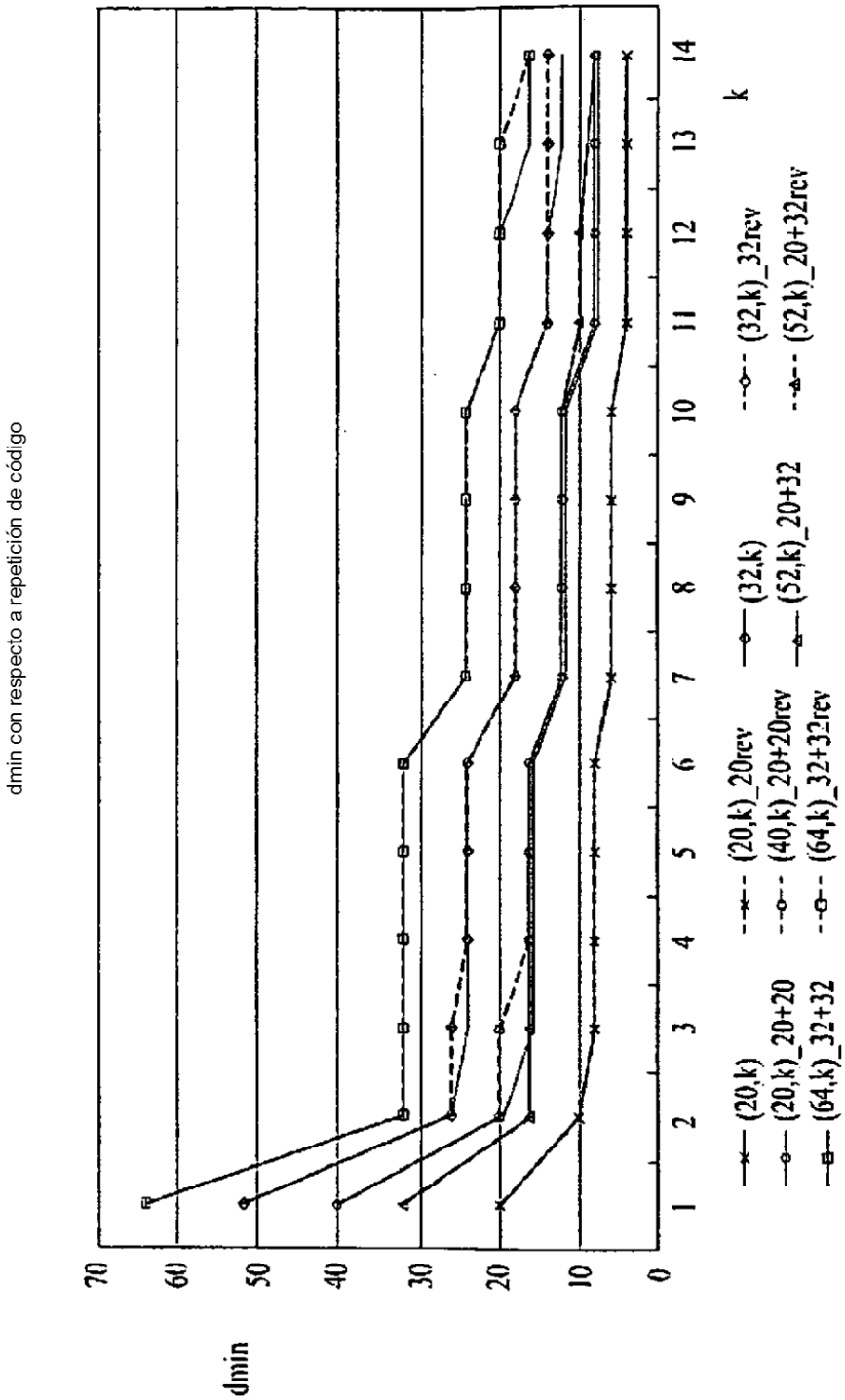

101
102
(32,10)
TFCI
(20,14)
PUCCH
102a
104
(20,14)
PUCCH
Código
(32, 14)
(32,10)
TFCI
TBD
101a
103

FIG. 2


dmin con respecto a repetición de código
dmin
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
k
(20,k)
(20,k)_20+20
(64,k)_32+32
(20,k)_20rev
(40,k)_20+20rev
(64,k)_32+32rev
(32,k)
(52,k)_20+32
(32,k)_32rev
(52,k)_20+32rev