

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 750 328**

51 Int. Cl.:

H04N 19/70	(2014.01)
H04N 19/132	(2014.01)
H04N 19/152	(2014.01)
H04N 19/50	(2014.01)
H04N 19/124	(2014.01)
H04N 19/176	(2014.01)
H04N 19/46	(2014.01)
H04N 19/13	(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **13.01.2011 PCT/JP2011/050462**
- 87 Fecha y número de publicación internacional: **05.04.2012 WO12042916**
- 96 Fecha de presentación y número de la solicitud europea: **13.01.2011 E 11828481 (9)**
- 97 Fecha y número de publicación de la concesión europea: **18.09.2019 EP 2624554**

54 Título: **Dispositivo de codificación de imagen dinámica, método de codificación de imagen dinámica y programa informático para codificación de imagen dinámica**

30 Prioridad:

30.09.2010 JP 2010222247

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

25.03.2020

73 Titular/es:

**FUJITSU LIMITED (100.0%)
1-1, Kamikodanaka 4-chome, Nakahara-ku
Kawasaki-shi, Kanagawa 211-8588 , JP**

72 Inventor/es:

**KAZUI, KIMIHICO;
SHIMADA, SATOSHI;
NAKAGAWA, AKIRA;
MIYOSHI, HIDENOBU y
KOYAMA, JUNPEI**

74 Agente/Representante:

MARTÍN BADAJOZ, Irene

ES 2 750 328 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

Dispositivo de codificación de imagen dinámica, método de codificación de imagen dinámica y programa informático para codificación de imagen dinámica

5

Campo

La presente invención se refiere a un aparato de codificación de vídeo en movimiento, a un método de codificación de vídeo en movimiento y a un programa informático de codificación de vídeo en movimiento para codificar cada imagen bloque por bloque, por ejemplo, dividiendo la imagen en una pluralidad de bloques.

10

Antecedentes

Según técnicas de codificación de vídeo en movimiento recientes, cada imagen se codifica usando diferentes modos de codificación que tienen diferentes eficiencias de compresión. Por ejemplo, una imagen se codifica como una imagen intrapredicha (imagen I), que es una imagen codificada usando sólo predicción intraimagen, o como una imagen predicha de manera unidireccional (imagen P) o una imagen predicha de manera bidireccional (imagen B), que es una imagen codificada usando predicción interimagen. Dado que la cantidad de código que se produce difiere según el tipo de la imagen o la complejidad de la imagen, la cantidad de código tiende a variar de una ocasión a la siguiente. Como resultado, para transmitir un flujo de datos que contiene vídeo en movimiento codificado a una tasa de transmisión constante, han de proporcionarse una memoria intermedia de transmisión y una memoria intermedia de recepción para almacenar el flujo en memoria intermedia. El retardo provocado por este almacenamiento en memoria intermedia es uno de los factores que provocan un retardo de visualización en el extremo de decodificador de vídeo en movimiento. Si se reduce el tamaño de memoria intermedia, puede reducirse el retardo de memoria intermedia en el extremo de decodificador de vídeo en movimiento, aunque la calidad de imagen tiende a degradarse, porque disminuye el grado de libertad al distribuir la cantidad de código entre las imágenes.

15

20

25

Normas internacionales de codificación de vídeo en movimiento usadas ampliamente en la actualidad, tales como MPEG-2 de vídeo (ISO/CEI 13818-2/ITU-T H.262, denominada a continuación en el presente documento MPEG-2) y MPEG-4 AVC/H.264 (ISO/CEI 14496-10/ITU-T H.264, denominada a continuación en el presente documento H.264), prevén el funcionamiento de una memoria intermedia de recepción de flujo en un decodificador idealizado denominado verificador de almacenamiento de vídeo en memoria intermedia (VBV) o memoria intermedia de imágenes codificadas (CPB), respectivamente. Un codificador de vídeo en movimiento ha de controlar la cantidad de código de modo que la memoria intermedia de recepción en el decodificador idealizado no tenga ni desbordamiento ni subdesbordamiento. Se estipula que el decodificador idealizado realiza una decodificación instantánea que tarda un tiempo cero en la realizar la decodificación. El documento de patente 1, por ejemplo, divulga un método de control de codificador de vídeo en movimiento referente al VBV. El documento de patente 2 divulga un aparato de codificación de vídeo en movimiento que comprende una unidad de control de cantidad de código que controla una cantidad de código para cada bloque contenido en un sector.

30

35

40

[Documentos de la técnica anterior]

[Documento de patente]

45

[Documento de patente 1] Publicación de patente japonesa abierta a consulta por el público n.º H03-148981; [documento de patente 2] US2006/088094

[Documentos no de patente]

50

[Documento no de patente 1] MPEG-2 Test Model 5, abril de 1993, ISO-CEI/JTC1/SC29/WG11/N0400 (<http://www.mpeg.org/MPEG/MSSG/tm5/>)

[Documento no de patente 2] JCTVC-A116, "Description of video coding technology proposal by Fraunhofer HHI", Joint Collaborative Team on Video Coding de ITU-T SG16 WP3 e ISO/CEI JTC1/SC29/WG11, abril de 2010

55

Sumario

[Problema que va a resolver la invención]

60

Con el fin de no provocar que la memoria intermedia de recepción en el decodificador idealizado tenga desbordamiento o subdesbordamiento, el codificador de vídeo en movimiento controla la cantidad de código para garantizar que todos los datos de cualquier imagen dada se hayan almacenado en la memoria intermedia de recepción para cuando el decodificador idealizado decodifique esa imagen dada.

65

Cuando el codificador de vídeo en movimiento está transmitiendo un flujo de datos a una tasa de transmisión

5 constante, si la cantidad de código de cada imagen es grande, puede producirse un subdesbordamiento de memoria intermedia de recepción que es un estado en el que algunos de los datos necesarios para decodificar la imagen no están almacenados aún en la memoria intermedia de recepción del decodificador cuando la imagen va a decodificarse y visualizarse mediante el decodificador de vídeo en movimiento. Si esto sucede, el decodificador de vídeo en movimiento pasará a no poder realizar la decodificación, y se producirá un salto de trama.

10 Por tanto, el decodificador de vídeo en movimiento visualiza la imagen después de retardar el flujo un tiempo prescrito a partir de su tiempo de recepción de modo que la decodificación puede hacerse sin provocar que la memoria intermedia de recepción tenga subdesbordamiento.

15 Tal como se describió anteriormente, se estipula que el decodificador idealizado efectúa la decodificación en un tiempo cero. Como resultado, si el tiempo de introducción de la i -ésima imagen en el codificador de vídeo en movimiento es $t(i)$, y el tiempo de decodificación de la i -ésima imagen en el decodificador idealizado es $dt(i)$, entonces el tiempo en el que la imagen pasa a estar lista para su visualización también es $dt(i)$. Dado que el periodo de visualización de imagen $\{t(i+1) - t(i)\}$ es igual a $\{dt(i+1) - dt(i)\}$ para cualquier imagen, el tiempo de decodificación $dt(i)$ viene dado como $dt(i) = t(i) + dly$, es decir, el tiempo retardado un tiempo fijo dly a partir del tiempo de introducción $t(i)$. Esto significa que el codificador de vídeo en movimiento ha de completar la transmisión de todos los datos necesarios para la decodificación a la memoria intermedia de recepción antes de que transcurra el tiempo $dt(i)$.

20 Con referencia a la figura 1, se facilitará una descripción de cómo funciona la memoria intermedia de recepción convencional. En la figura 1, la abscisa representa el tiempo y la ordenada representa la ocupación de memoria intermedia de la memoria intermedia de recepción. El gráfico 100 de línea continua representa la ocupación de memoria intermedia en función del tiempo.

25 La ocupación de memoria intermedia de la memoria intermedia de recepción se restaura a una tasa igual a la tasa de transmisión, y los datos usados para decodificar cada imagen se recuperan a partir de la memoria intermedia en el tiempo de decodificación de la imagen. Los datos de la i -ésima imagen empiezan a introducirse en la memoria intermedia de recepción en el tiempo $at(i)$, y los datos finales de la i -ésima imagen se introducen en el tiempo $ft(i)$. El decodificador idealizado completa la decodificación de la i -ésima imagen en el tiempo $dt(i)$, tras lo cual la i -ésima imagen pasa a estar lista para su visualización.

30 Mientras que el decodificador idealizado es un decodificador instantáneo, el decodificador de vídeo en movimiento real tarda un tiempo finito en decodificar una imagen; generalmente, el tiempo de procesamiento de decodificación para una imagen es menor que el periodo de visualización de imagen, pero la decodificación tarda un tiempo cercano al periodo de visualización de imagen. Los datos de la i -ésima imagen se introducen en la memoria intermedia de recepción durante el periodo que se inicia en el tiempo $at(i)$ y que finaliza en el tiempo $ft(i)$, pero no hay garantía de en cuál de los tiempos $at(i)$ a $ft(i)$ llegarán los datos necesarios para decodificar un bloque dado. Por tanto, el decodificador de vídeo en movimiento real empieza a decodificar la i -ésima imagen en el tiempo $ft(i)$. Como resultado, el decodificador de vídeo en movimiento real sólo puede garantizar que completará la decodificación antes de que transcurra el tiempo $ft(i)+ct$, donde ct es el tiempo de procesamiento de caso más desfavorable necesario para decodificar una imagen.

35 Lo que el codificador de vídeo en movimiento real garantiza es que los datos necesarios para decodificar la imagen llegan a la memoria intermedia de recepción del decodificador antes de que transcurra el tiempo $dt(i)$, es decir, $ft(i)$ no será posterior a $dt(i)$. Como resultado, cuando el retardo de $ft(i)$ es el más prolongado, $ft(i)$ pasa a ser igual a $dt(i)$. En este caso, el tiempo en el que se garantiza que se completa la decodificación es $dt(i)+ct$. Si todas las imágenes van a visualizarse a una tasa de imágenes constante, el decodificador de vídeo en movimiento ha de retardar el tiempo de visualización de cada imagen al menos ct en comparación con el decodificador idealizado.

40 Por consiguiente, un objeto de la presente invención es proporcionar un aparato de codificación de vídeo en movimiento y un método de codificación de vídeo en movimiento que pueden reducir el retardo de decodificación.

45 [Medios para resolver el problema]

50 Según una realización, se proporciona un aparato de codificación de vídeo en movimiento para codificar cada imagen contenida en datos de vídeo en movimiento dividiendo la imagen en una pluralidad de bloques. El aparato de codificación de vídeo en movimiento incluye: una unidad de determinación de grupo que determina a cuál de una pluralidad de grupos pertenece cada bloque; una unidad de cálculo de información de tiempo de decodificación de grupo que calcula un tiempo de decodificación para cada uno de los grupos; una unidad de anexado de información de grupo que anexa información de grupo que identifica el grupo al que pertenece cada bloque a datos que van a emitirse; una unidad de control de cantidad de código que controla una cantidad de código para cada bloque contenido en el grupo de modo que datos necesarios para decodificar todos los bloques contenidos en el grupo llegan a una memoria intermedia de recepción de flujo proporcionada en un aparato de decodificación de vídeo en movimiento antes de que transcurra el tiempo de decodificación del grupo calculado por la unidad de cálculo de información de tiempo de decodificación de grupo cuando los datos se transmiten al aparato de decodificación de

vídeo en movimiento a una tasa de transmisión prescrita; y una unidad de procesamiento de codificación que codifica cada bloque, basándose en información de control referente a la cantidad de código.

5 Según otra realización, se proporciona un método de codificación de vídeo en movimiento para codificar cada imagen contenida en datos de vídeo en movimiento dividiendo la imagen en una pluralidad de bloques. El método de codificación de vídeo en movimiento incluye: determinar a cuál de una pluralidad de grupos pertenece cada bloque; calcular un tiempo de decodificación para cada uno de los grupos; anexas información de grupo que identifica el grupo al que pertenece cada bloque a datos que van a emitirse; controlar una cantidad de código para cada bloque contenido en el grupo de modo que datos necesarios para decodificar todos los bloques contenidos en el grupo
10 llegan a una memoria intermedia de recepción de flujo proporcionada en un aparato de decodificación de vídeo en movimiento antes de que transcurra el tiempo de decodificación del grupo cuando los datos se transmiten al aparato de decodificación de vídeo en movimiento a una tasa de transmisión prescrita; y codificar cada bloque, basándose en información de control referente a la cantidad de código.

15 Según todavía otra realización, se proporciona un programa informático de codificación de vídeo en movimiento para codificar cada imagen contenida en datos de vídeo en movimiento dividiendo la imagen en una pluralidad de bloques. El programa informático de codificación de vídeo en movimiento incluye instrucciones para hacer que un ordenador ejecute las etapas de: determinar a cuál de una pluralidad de grupos pertenece cada bloque; calcular un tiempo de decodificación para cada uno de los grupos; anexas información de grupo que identifica el grupo al que pertenece cada bloque a datos que van a emitirse; controlar una cantidad de código para cada bloque contenido en el grupo de modo que datos necesarios para decodificar todos los bloques contenidos en el grupo llegan a una memoria intermedia de recepción de flujo proporcionada en un aparato de decodificación de vídeo en movimiento antes de que transcurra el tiempo de decodificación del grupo cuando los datos se transmiten al aparato de decodificación de vídeo en movimiento a una tasa de transmisión prescrita; y codificar cada bloque, basándose en
20 información de control referente a la cantidad de código.

El objeto y las ventajas de la invención se realizarán y conseguirán por medio de los elementos y combinaciones señalados particularmente en las reivindicaciones.

30 Ha de entenderse que tanto la descripción general anterior como la siguiente descripción detallada son a modo de ejemplo y explicativas y no restringen la invención, tal como se reivindica.

[Efecto de la invención]

35 El aparato de codificación de vídeo en movimiento y el método de codificación de vídeo en movimiento divulgados en el presente documento pueden reducir el retardo de decodificación.

Breve descripción de los dibujos

40 La figura 1 es un diagrama que ilustra la ocupación de memoria intermedia de una memoria intermedia de recepción según la técnica anterior en función del tiempo.

La figura 2 es un diagrama que ilustra esquemáticamente la configuración de un aparato de codificación de vídeo en movimiento según una realización.

45 La figura 3 es un diagrama que ilustra cómo varía en función del tiempo la ocupación de memoria intermedia de una memoria intermedia de recepción para almacenar un flujo de datos de vídeo en movimiento codificado recibido desde el aparato de codificación de vídeo en movimiento según la una realización.

50 La figura 4 es un diagrama que ilustra cómo varía en función del tiempo la ocupación de memoria intermedia de la memoria intermedia de recepción cuando se retarda el tiempo de decodificación del primer grupo.

La figura 5 es un diagrama de flujo de funcionamiento de un procedimiento de codificación de vídeo en movimiento realizado por el aparato de codificación de vídeo en movimiento según la una realización.

55 La figura 6 es un diagrama para explicar un bit final de cada bloque según una primera realización.

La figura 7 es un diagrama para explicar un bit final de cada bloque según una segunda realización.

60 La figura 8A es un diagrama que representa la estructura de datos comprimidos de un bloque según una tercera realización.

La figura 8B es un diagrama para explicar la relación entre los datos comprimidos y una secuencia de bits convertida mediante codificación de longitud variable.

65 Las figuras 9A a 9C son un diagrama de flujo para derivar la posición de bit final de un bloque codificado por longitud

variable mediante una unidad de codificación de longitud variable según la tercera realización.

La figura 10 es un diagrama que ilustra esquemáticamente la configuración de una unidad de codificación de entropía proporcionada en una unidad de codificación de longitud variable según una cuarta realización.

La figura 11 es un diagrama para explicar un bit final de cada bloque según la cuarta realización.

La figura 12 es un diagrama para explicar el comportamiento de una unidad de codificación de bin y una unidad de multiplexación en la unidad de codificación de longitud variable según la cuarta realización.

Las figuras 13A a 13C son un diagrama de flujo para derivar la posición de bit final de un bloque codificado por longitud variable mediante la unidad de codificación de longitud variable según la cuarta realización.

Descripción de realizaciones

A continuación se describirá un aparato de codificación de vídeo en movimiento según una realización con referencia a los dibujos.

La figura 2 es un diagrama que ilustra esquemáticamente la configuración del aparato de codificación de vídeo en movimiento según la una realización. El aparato 10 de codificación de vídeo en movimiento incluye una unidad 11 de procesamiento de codificación, una unidad 12 de control de cantidad de código, una unidad 13 de determinación de grupo, una unidad 14 de cálculo de información de tiempo de decodificación de grupo, una unidad 15 de anexo de información de grupo y un contador 25 de bits. Estas unidades que constituyen el aparato 10 de codificación de vídeo en movimiento se implementan cada una como un circuito independiente en el aparato 10 de codificación de vídeo en movimiento. Alternativamente, estas unidades que constituyen el aparato 10 de codificación de vídeo en movimiento pueden implementarse en el aparato 10 de codificación de vídeo en movimiento en forma de un único circuito integrado en el que se integran los circuitos que implementan las funciones de las unidades respectivas. Además, estas unidades que constituyen el aparato 10 de codificación de vídeo en movimiento pueden implementarse como módulos funcionales ejecutando un programa informático en un procesador incorporado en el aparato 10 de codificación de vídeo en movimiento.

Cada imagen contenida en datos de vídeo en movimiento que va a codificarse se divide en una pluralidad de bloques mediante una unidad de control no representada, y se suministra a la unidad 11 de procesamiento de codificación bloque por bloque. Cada bloque contiene, por ejemplo, 16×16 píxeles.

Cada bloque se introduce en una unidad 21 de transformada ortogonal. La unidad 21 de transformada ortogonal calcula un conjunto de coeficientes de frecuencia aplicando una transformada ortogonal tal como una transformada de coseno discreta (DCT) a cada bloque. El conjunto de coeficientes de frecuencia así calculado se pasa a una unidad 22 de cuantificación. Cuando se crea una imagen predicha para cada bloque mediante compensación de movimiento a partir de una imagen codificada previamente, una imagen de error de predicción generada calculando la diferencia entre el bloque y la imagen predicha puede introducirse en la unidad 21 de transformada ortogonal.

La imagen puede ser o bien una trama o bien un campo. Una trama se refiere a una imagen fija completa en los datos de vídeo en movimiento, mientras que un campo se refiere a una imagen fija obtenida extrayendo datos sólo en las líneas impares o las líneas pares a partir de la trama.

Además, el vídeo en movimiento codificado puede ser vídeo en movimiento en color o vídeo en movimiento monocromo.

La unidad 22 de cuantificación cuantifica cada coeficiente de frecuencia dividiendo el coeficiente de frecuencia por la escala de cuantificación determinada según el valor de cuantificación calculado por una unidad 24 de cálculo de valor de cuantificación proporcionada en la unidad 12 de control de cantidad de código. Con esta cuantificación, se reduce la cantidad de información necesaria para representar cada coeficiente de frecuencia. Cuanto mayor sea el valor de cuantificación, menor será la resolución de cuantificación, y la cantidad de información necesaria para representar cada coeficiente de frecuencia pasa a ser menor de manera correspondiente.

Cada coeficiente de frecuencia cuantificado por la unidad 22 de cuantificación se pasa a una unidad 23 de codificación de longitud variable. Cada coeficiente de frecuencia cuantificado puede cuantificarse de manera inversa con el fin de crear una imagen predicha para una imagen o un bloque que van a codificarse posteriormente, y además puede someterse a una transformación ortogonal inversa y almacenarse en una memoria de imágenes no representada.

La unidad 23 de codificación de longitud variable codifica por longitud variable los datos comprimidos del bloque (que incluyen los coeficientes de frecuencia cuantificados e información de cabecera de bloque tal como información de modo de codificación). Luego, la unidad 23 de codificación de longitud variable ensambla la secuencia de bits representada por el código de longitud variable para dar un flujo de datos de salida.

Además, el número de bits en la secuencia de bits generada mediante la unidad 23 de codificación de longitud variable lo suma el contador 25 de bits bloque por bloque, y el resultado se suministra como la cantidad de código generado a una unidad 26 de cálculo de ocupación de memoria intermedia proporcionada en la unidad 12 de control de cantidad de código.

Más adelante se describirán realizaciones detalladas de la unidad 23 de codificación de longitud variable.

La unidad 13 de determinación de grupo determina el grupo al que pertenece el bloque que está codificándose según un método prescrito basándose en información de recuento de bloques recibida desde la unidad de control no representada. La información de recuento de bloques es información que indica el número asignado a cada bloque dado contenido en la imagen; por ejemplo, el número "1" se asigna al bloque ubicado en la esquina superior izquierda de la imagen, y los bloques posteriores se numeran en secuencia por orden de barrido por filas. El número más alto se asigna al bloque ubicado en la esquina inferior derecha de la imagen. Alternativamente, la información de recuento de bloques puede incluir el número asignado a cada bloque dado según algún otro esquema de ordenación.

Es preferible que la unidad 13 de determinación de grupo determine una pluralidad de grupos de modo que cada grupo contenga un número igual de bloques en la medida de lo posible con el fin de hacer que el tiempo de procesamiento de decodificación sea sustancialmente igual para todos los grupos. Por ejemplo, si los bloques se agrupan por líneas de bloques, la unidad 13 de determinación de grupo puede hacer que el número de bloques contenidos en cada grupo sea igual para cualquier tamaño de imagen dado. Por ejemplo, cuando el tamaño de imagen es de 1920 × 1088 píxeles correspondientes a una imagen de televisión de alta definición (HDTV), y el tamaño de bloque es de 16 × 16 píxeles, el número de líneas de bloque es de 68. Por consiguiente, en este caso, cada bloque dado contenido en la imagen que está codificándose se clasifica como perteneciente a uno de los 68 grupos.

El número de bloques contenidos en cada grupo puede ser cualquier número entre 1 y el número total de bloques contenidos en toda la imagen.

La unidad 13 de determinación de grupo envía información de identificación que identifica el grupo al que pertenece el bloque que está codificándose a la unidad 26 de cálculo de ocupación de memoria intermedia en la unidad 12 de control de cantidad de código. Además, la unidad 13 de determinación de grupo notifica a la unidad 14 de cálculo de información de tiempo de decodificación de grupo y a la unidad 15 de anexado de información de grupo el número total N de grupos. La unidad 13 de determinación de grupo puede notificar a la unidad 15 de anexado de información de grupo el índice del bloque ubicado en el encabezamiento de cada grupo.

A continuación se facilitará una descripción tomando como ejemplo el caso en el que los bloques se dividen por igual entre N grupos por líneas de bloques, donde M indica el número de bloques contenidos horizontalmente en la imagen que está codificándose y N el número de bloques contenidos verticalmente en la misma.

Suponiendo que la imagen que está codificándose es la i-ésima imagen en el orden de codificación, la unidad 14 de cálculo de información de tiempo de decodificación de grupo calcula el tiempo de decodificación $dgt(i,n)$, es decir, el tiempo en el que el n-ésimo grupo va a decodificarse, basándose en el tiempo de decodificación $dt(i) \{= t(i)+dly\}$ de la i-ésima imagen que es posterior según un tiempo de retardo predeterminado dly al tiempo de introducción $t(i)$ de la i-ésima imagen. En vez de $dgt(i,n)$, la unidad 14 de cálculo de información de tiempo de decodificación de grupo puede calcular $\{dgt(i,n) - dgt(i,n-1)\}$ como el tiempo de decodificación que es equivalente a $dgt(i,n)$. Además, la unidad 14 de cálculo de información de tiempo de decodificación de grupo puede redondear el tiempo de decodificación a una unidad apropiada, por ejemplo, a un múltiplo de 1/90000 segundos.

La unidad 14 de cálculo de información de tiempo de decodificación de grupo determina el tiempo de decodificación para cada grupo, por ejemplo, dividiendo el tiempo necesario para decodificar una imagen por igual entre el número N de grupos de modo que el tiempo necesario para decodificar cada bloque pasa a ser igual. En este caso, el tiempo de decodificación del n-ésimo grupo ($n = 1, 2, \dots, N$) se determina según la siguiente ecuación.

$$dgt(i, n) = dt(i - 1) + \{dt(i) - dt(i - 1)\} \cdot n / N \quad (1)$$

No es necesario hacer que el tiempo de procesamiento de decodificación sea necesariamente igual para cada grupo. Por ejemplo, en el caso del primer grupo que va a codificarse y decodificarse, la unidad 14 de cálculo de información de tiempo de decodificación de grupo puede determinar el tiempo de decodificación $dgt(i,1)$ tal como se indica mediante la siguiente ecuación, es decir, de tal manera que se retarda en relación con el tiempo de decodificación que se determinaría dividiendo por igual el tiempo necesario para decodificar una imagen.

$$dgt(i,1) \geq dt(i - 1) + \{dt(i) - dt(i - 1)\} / N \quad (2)$$

donde $dt(i)$ y $dt(i-1)$ son los tiempos de decodificación para las imágenes i -ésima e $(i-1)$ -ésima, respectivamente.

Además, para cada grupo que se codifica y decodifica posteriormente, la unidad 14 de cálculo de información de tiempo de decodificación de grupo puede determinar el tiempo de decodificación $dgt(i,n)$ ($n \geq 2$) tal como se indica mediante la siguiente ecuación.

$$dgt(i, n) = dgt(i,1) + \{dt(i) - dgt(i,1)\} \cdot (n-1)/(N-1) \quad (n \geq 2) \quad (3)$$

Determinando así los tiempos de decodificación, la unidad 14 de cálculo de información de tiempo de decodificación de grupo puede aumentar la ocupación de memoria intermedia que va a calcularse por la unidad 26 de cálculo de ocupación de memoria intermedia antes de que se inicie la codificación del primer grupo. Esto proporciona una mayor libertad de control de cantidad de código.

La unidad 14 de cálculo de información de tiempo de decodificación de grupo pasa la información de tiempo de decodificación que contiene el tiempo de decodificación de cada grupo tanto a la unidad 26 de cálculo de ocupación de memoria intermedia como a la unidad 15 de anexo de información de grupo.

La unidad 26 de cálculo de ocupación de memoria intermedia calcula una estimación de la ocupación de memoria intermedia de la memoria intermedia de recepción de flujo en el decodificador idealizado. La tasa de bits de transmisión va a indicarse mediante R y la ocupación de memoria intermedia mediante d . Cuando se completa la codificación de longitud variable del bloque actual, la cantidad del código generado, b , se notifica desde el contador 25 de bits hasta la unidad 26 de cálculo de ocupación de memoria intermedia. Luego, la unidad 26 de cálculo de ocupación de memoria intermedia resta b de d .

Después de que se ha codificado el último bloque en cada grupo, la unidad 26 de cálculo de ocupación de memoria intermedia restaura la ocupación de memoria intermedia, d , según la siguiente ecuación.

$$d = d + (dgt(i, n+1) - dgt(i, n)) \cdot R \quad (4)$$

Cada vez que se completa la codificación de longitud variable de un bloque, la unidad 26 de cálculo de ocupación de memoria intermedia notifica la ocupación de memoria intermedia calculada, d , a la unidad 24 de cálculo de valor de cuantificación.

Basándose en la ocupación de memoria intermedia, la unidad 24 de cálculo de valor de cuantificación calcula el valor de cuantificación para ese bloque. En este momento, la unidad 24 de cálculo de valor de cuantificación controla el valor de cuantificación para cada bloque de modo que la cantidad total de código que resulta de codificar todos los bloques en el grupo no excede la ocupación de memoria intermedia, d , calculada justo antes de iniciar la codificación del primer bloque en ese grupo, es decir, el valor de d no pasa a ser negativo durante el procedimiento de codificación.

La unidad 24 de cálculo de valor de cuantificación calcula el valor de cuantificación de acuerdo, por ejemplo, con el método de cálculo de valor de cuantificación definido en un software de referencia de MPEG-2, el modelo de prueba 5 (véase el documento no de patente 1). A continuación, la unidad 24 de cálculo de valor de cuantificación compara la ocupación de memoria intermedia, d , con un umbral predeterminado DTH1. Cuando el valor de cuantificación es el valor máximo que puede adoptar dentro de un intervalo prescrito, el umbral DTH1 viene dado mediante la siguiente ecuación.

$$DTH1 = b0 \cdot M0 + \text{compensación} \quad (5)$$

donde $b0$ es la cantidad máxima de código que puede resultar de cada bloque, y $M0$ es el número de bloques que aún han de codificarse en el grupo al que pertenece el bloque que está codificándose. En la ecuación (5), la compensación es un término de margen. Cuando d se compara con el umbral DTH1, si d es menor que DTH1, el valor de cuantificación se establece en el valor máximo.

Alternativamente, puede hacerse que $b0$ represente la cantidad de código que resulta de cada bloque cuando se supone que todos los coeficientes de frecuencia de ese bloque son cero. En este caso, si d es menor que DTH1, la unidad 24 de cálculo de valor de cuantificación determina el valor de cuantificación de modo que todos los coeficientes de frecuencia del bloque que está codificándose se cuantifican a cero. Realizando así el control, la memoria intermedia de recepción en el decodificador hipotético no tiene subdesbordamiento, siempre y cuando el valor promedio de la cantidad de código que resulta de la codificación de los bloques que quedan por codificarse en el grupo no exceda $b0$.

De esta manera, transmitiendo realmente el flujo de salida desde el aparato de codificación de vídeo en movimiento a la tasa prescrita R hasta el aparato de decodificación de vídeo en movimiento, la unidad 12 de control de cantidad de código puede controlar la cantidad de código de los datos de vídeo en movimiento de modo que la memoria

intermedia de recepción en el aparato de decodificación de vídeo en movimiento no tiene subdesbordamiento.

La unidad 24 de cálculo de valor de cuantificación pasa el valor de cuantificación calculado a la unidad 22 de cuantificación.

5 Con el fin de compartir información referente al grupo al que pertenece cada bloque y el tiempo de decodificación de cada grupo con el aparato de decodificación de vídeo en movimiento, el aparato 10 de codificación de vídeo en movimiento anexa al menos información de grupo que identifica los bloques pertenecientes a cada grupo e información de tiempo de decodificación que contiene el tiempo de decodificación de cada grupo al flujo de datos que va a emitirse. El aparato 10 de codificación de vídeo en movimiento envía la información de grupo y la información de tiempo de decodificación al aparato de decodificación de vídeo en movimiento.

10 Con este propósito, la unidad 15 de anexado de información de grupo anexa la información de grupo a la información de cabecera del flujo de datos de salida, por ejemplo, para cada imagen o para cada número predeterminado de imágenes.

15 La cabecera de secuencia de MPEG-2 o el conjunto de parámetros de secuencia o la información de mejora complementaria de H.264, por ejemplo, pueden usarse como información de cabecera. El tiempo de decodificación de cada grupo puede anexarse a la información de cabecera, tal como la cabecera de imagen de MPEG-2 o la cabecera de sector de H.264, que se adjunta invariablemente a cada imagen.

20 Cuando los grupos se determinan de modo que cada grupo contiene un número igual de bloques, el aparato 10 de codificación de vídeo en movimiento notifica al aparato de decodificación de vídeo en movimiento que todos los bloques se han dividido por igual entre los N grupos. Con este propósito, la unidad 13 de determinación de grupo envía información acerca del número de grupos, N, como información de grupo a la unidad 15 de anexado de información de grupo. La unidad 15 de anexado de información de grupo codifica esta información de grupo. En MPEG-2 y H.264, la codificación se realiza bloque por bloque, denominados macrobloques cada uno de los cuales comprende 16×16 píxeles, y habitualmente el número de bloques no excede un intervalo que puede expresarse mediante 20 bits. Dado que el número máximo N de grupos es a lo sumo igual al número máximo de bloques, N también puede codificarse con una longitud de bits fija.

25 Por otra parte, cuando el número de bloques contenidos en cada grupo no es siempre igual, la unidad 13 de determinación de grupo envía información de índice que indica el bloque de inicio en cada grupo, junto con el número de grupos, N, como información de grupo a la unidad 15 de anexado de información de grupo. La unidad 15 de anexado de información de grupo codifica en primer lugar el número de grupos, N, y luego codifica la información de índice del bloque de inicio en cada grupo en secuencia. Un esquema de codificación de longitud de bits fija, por ejemplo, también se usa para codificar la información de índice del bloque de inicio. Alternativamente, la unidad 15 de anexado de información de grupo puede usar otros esquemas de codificación, por ejemplo, un esquema de codificación de longitud variable tal como la codificación de Huffman, para codificar el número de grupos, N, y la información de índice del bloque de inicio en cada grupo.

30 La unidad 14 de cálculo de información de tiempo de decodificación de grupo notifica a la unidad 15 de anexado de información de grupo el tiempo de decodificación calculado para cada grupo. En este caso, la unidad 14 de cálculo de información de tiempo de decodificación de grupo calcula el tiempo de decodificación en forma del valor de diferencia $dgt(i,n) - dgt(i,n-1)$ para cada uno de los N grupos empezando con el primer grupo y finalizando con el último N-ésimo grupo, y envía la información de tiempo de decodificación así calculada a la unidad 15 de anexado de información de grupo. Para el tiempo de decodificación del primer grupo, $dgt(i,0)$ se establece igual al tiempo de decodificación $dgt(i-1, N) = dt(i-1)$ del último grupo en la imagen inmediatamente anterior. La unidad 15 de anexado de información de grupo codifica el tiempo de decodificación de cada grupo, y anexa el tiempo de decodificación codificado a los datos de imagen actuales que se envían al aparato de decodificación de vídeo en movimiento. La unidad 15 de anexado de información de grupo puede cuantificar en primer lugar cada valor de diferencia con una resolución de cuantificación adecuada, por ejemplo, una resolución de 1/90000 segundos, y luego codificarlo con una longitud de bits fija de 32 bits más o menos. Mientras que lo anterior se ha descrito como ejemplo de codificación usando una longitud de bits fija, la unidad 15 de anexado de información de grupo puede codificar el valor de diferencia que representa el tiempo de decodificación de cada grupo usando un esquema de codificación de longitud variable adecuado.

35 Basándose en el número de grupos, N, y la información de tiempo de decodificación calculada para cada grupo, el aparato de decodificación de vídeo en movimiento calcula la cantidad de retardo de visualización. Cuando los grupos se determinan de modo que cada grupo contiene un número igual de bloques, la cantidad de retardo de visualización es ct/N , donde ct es el tiempo de procesamiento de caso más desfavorable necesario para decodificar una imagen.

40 A continuación, con referencia a las figuras 3 y 4, se facilitará una descripción de cómo el aparato 10 de codificación de vídeo en movimiento según la realización anterior puede reducir el retardo que se produce cuando el aparato de decodificación de vídeo en movimiento decodifica el vídeo en movimiento.

La figura 3 es un diagrama que ilustra cómo la ocupación de memoria intermedia de la memoria intermedia de recepción para almacenar el flujo de datos de vídeo en movimiento codificado recibido desde el aparato 10 de codificación de vídeo en movimiento varía en función del tiempo. La figura 4 es un diagrama que ilustra cómo la ocupación de memoria intermedia de la memoria intermedia de recepción varía en función del tiempo cuando se retarda el tiempo de decodificación del primer grupo. En las figuras 3 y 4, la abscisa representa el tiempo y la ordenada representa la ocupación de memoria intermedia. Se supone que el número de grupos, N, es 4. En la figura 3, el gráfico 300 representa la variación de la ocupación de memoria intermedia en función del tiempo. Cada periodo indicado mediante una flecha 301 representa el tiempo máximo ct/N que puede ser necesario para decodificar el grupo correspondiente. De manera similar, en la figura 4, el gráfico 400 representa la variación de la ocupación de memoria intermedia en función del tiempo.

El aparato 10 de codificación de vídeo en movimiento controla la cantidad de código para cada bloque de modo que los datos necesarios para decodificar los bloques contenidos en el n-ésimo grupo llegan a la memoria intermedia de recepción antes de que transcurra el tiempo $dgt(i,n)$ dado por la ecuación (1) mencionada anteriormente. Se mantiene la siguiente relación entre el tiempo de procesamiento de caso más desfavorable ct necesario para decodificar una imagen y el tiempo de decodificación $dt(i-1)=dgt(i-1,N)$ de la (i-1)-ésima imagen y el tiempo de decodificación $dt(i)=dgt(i,N)$ del último grupo en la i-ésima imagen.

$$ct \geq \{dt(i) - dt(i-1)\} \quad (6)$$

Cuando cada grupo contiene un número igual de bloques, dado que se mantiene la siguiente relación tal como se ilustra en la figura 3, el procedimiento de decodificación para los bloques contenidos en los grupos primero a (N-1)-ésimo en la i-ésima imagen se completará antes de que transcurra el tiempo de decodificación $dgt(i,N)$ del último grupo en la i-ésima imagen.

$$dgt(i, n+1) \geq dgt(i, n) + ct / N \quad (7)$$

Supóngase que los datos necesarios para decodificar los bloques contenidos en el N-ésimo grupo llegan a la memoria intermedia de recepción en el aparato de decodificación de vídeo en movimiento en el tiempo $dgt(i,N)$, tras lo cual se inicia la decodificación del N-ésimo grupo; en este caso, el tiempo de procesamiento necesario para la decodificación es ct/N . Esto significa que la decodificación de todos los bloques se completará en el tiempo dado por la siguiente ecuación, haciendo por tanto que la imagen esté lista para su visualización. De esta manera, el tiempo de retardo de visualización en relación con el decodificador idealizado se reduce desde ct hasta ct/N .

$$dgt(i, N) + ct / N = dt(i) + ct / N \quad (8)$$

Cuando el tiempo de decodificación se calcula grupo por grupo, el tiempo de decodificación $dgt(i,1)$ del primer grupo en la i-ésima imagen pasa a ser anterior según $\{dt(i)-dt(i-1)\}(N-1)/N$ al tiempo de decodificación $dt(i)$ de la i-ésima imagen. Como resultado, tal como puede observarse a partir de la ocupación de memoria intermedia por imagen representada en la figura 1 e indicada mediante la línea 302 discontinua en la figura 3, la ocupación de la memoria intermedia de recepción se reduce en comparación con la técnica anterior, y la cantidad de código que puede usarse para cada grupo disminuye, mientras que al mismo tiempo, el grado de libertad para distribuir la cantidad de código entre los grupos disminuye. En el caso de decodificar cada imagen con un breve retardo aproximadamente igual al periodo de visualización de una imagen después de que los datos de la imagen se han introducido en la memoria intermedia de recepción, tal como se ilustra en la figura 1, los efectos de una ocupación de memoria intermedia reducida pasan a ser más pronunciados dado que la ocupación de memoria intermedia pasa a ser menor.

Para evitar un problema de este tipo, es preferible retardar el tiempo de decodificación del primer grupo, tal como se indica mediante la ecuación (2) dada anteriormente, con el propósito de restaurar la ocupación de memoria intermedia. Retardando el tiempo de decodificación del primer grupo, el número de bits que pueden transmitirse a la memoria intermedia aumenta, y el grado de libertad para controlar la cantidad de código generado para cada imagen puede aumentarse. En este caso, el último tiempo de compleción de decodificación viene dado mediante la siguiente ecuación.

$$\min(dgt(i,1) + ct, dt(i) + ct / N) \quad (9)$$

La función $\min(x,y)$ es una función que devuelve el valor de la variable x o y , la que sea menor.

Tal como puede observarse comparando la ecuación (9) con la ecuación (8), si se retarda el tiempo de decodificación del primer grupo, el retardo no aumenta en el caso de un aparato de decodificación de vídeo en movimiento en el que ct es suficientemente breve, mientras que en el caso de un aparato de decodificación de vídeo en movimiento en el que ct es prolongado, el retardo pasa a ser mayor que cuando no se retarda el tiempo de decodificación del primer grupo.

5 La figura 5 es un diagrama de flujo de funcionamiento del procedimiento de codificación de vídeo en movimiento realizado por el aparato 10 de codificación de vídeo en movimiento según la una realización. El aparato 10 de codificación de vídeo en movimiento realiza el siguiente procedimiento de codificación de vídeo en movimiento en cada imagen.

10 La unidad 13 de determinación de grupo determina el grupo al que pertenece cada bloque (etapa S101). La unidad 13 de determinación de grupo envía información de identificación que identifica el grupo al que pertenece el bloque a la unidad 26 de cálculo de ocupación de memoria intermedia en la unidad 12 de control de cantidad de código. Además, la unidad 13 de determinación de grupo notifica a la unidad 14 de cálculo de información de tiempo de decodificación de grupo y la unidad 15 de anexado de información de grupo el número total N de grupos. La unidad 13 de determinación de grupo puede notificar a la unidad 15 de anexado de información de grupo el índice del bloque ubicado en el encabezamiento de cada grupo.

15 A continuación, la unidad 14 de cálculo de información de tiempo de decodificación de grupo calcula el tiempo de decodificación para cada grupo (etapa S102). La unidad 14 de cálculo de información de tiempo de decodificación de grupo pasa información que indica el tiempo de decodificación de cada grupo a la unidad 26 de cálculo de ocupación de memoria intermedia y la unidad 15 de anexado de información de grupo.

20 La unidad 15 de anexado de información de grupo codifica la información de tiempo de decodificación y la información de grupo que identifica el grupo al que pertenece cada bloque, y anexa la información de tiempo de decodificación y la información de grupo codificadas al flujo de datos que va a emitirse (etapa S103).

25 Por otra parte, la unidad 26 de cálculo de ocupación de memoria intermedia realiza una estimación de la ocupación de memoria intermedia que se espera para la memoria intermedia de recepción en el tiempo de decodificación de cada grupo para el caso en el que los datos necesarios para decodificar todos los bloques contenidos en el grupo se transmiten al aparato de decodificación de vídeo en movimiento a una tasa de transmisión predeterminada (etapa S104). Con este propósito, la unidad 26 de cálculo de ocupación de memoria intermedia se refiere a la información recibida desde el contador 25 de bits que representa la cantidad de código de los bloques codificados hasta el momento dentro del grupo. Luego, la unidad 26 de cálculo de ocupación de memoria intermedia pasa información que representa la ocupación de memoria intermedia estimada a la unidad 24 de cálculo de valor de cuantificación en la unidad 12 de control de cantidad de código. La unidad 24 de cálculo de valor de cuantificación controla la cantidad de código para cada bloque contenido en el grupo de modo que todos los datos necesarios para decodificar el grupo lleguen a la memoria intermedia de recepción de flujo en el aparato de decodificación de vídeo en movimiento antes de que transcurra el tiempo de decodificación de ese grupo (etapa S105). Luego, la unidad 24 de cálculo de valor de cuantificación calcula la información de control de cantidad de código calculando el valor de cuantificación que coincide con la cantidad de código, y pasa el valor de cuantificación así calculado a la unidad 22 de cuantificación en la unidad 11 de procesamiento de codificación.

40 La unidad 21 de transformada ortogonal en la unidad 11 de procesamiento de codificación calcula un conjunto de coeficientes de frecuencia para cada bloque realizando una transformada ortogonal en cada bloque (etapa S106). La unidad 21 de transformada ortogonal pasa el conjunto de coeficientes de frecuencia de cada bloque a la unidad 22 de cuantificación.

45 La unidad 22 de cuantificación cuantifica el conjunto de coeficientes de frecuencia de cada bloque, basándose en el valor de cuantificación calculado como la información de control de cantidad de código (etapa S107). La unidad 22 de cuantificación pasa cada coeficiente de frecuencia cuantificado a la unidad 23 de codificación de longitud variable en la unidad 11 de procesamiento de codificación. La unidad 23 de codificación de longitud variable codifica por longitud variable cada coeficiente de frecuencia cuantificado (etapa S108). La unidad 23 de codificación de longitud variable emite el código de longitud variable resultante. Este código de longitud variable se almacena en el flujo de datos de salida junto con la información de grupo, la información de tiempo de decodificación, etc., emitidas desde la unidad 15 de anexado de información de grupo. La unidad 23 de codificación de longitud variable suministra la cantidad de código para cada bloque al contador 25 de bits.

50 Después de la etapa S108, el aparato 10 de codificación de vídeo en movimiento termina el procedimiento de codificación de vídeo en movimiento.

60 A continuación se facilitará una descripción de cuatro realizaciones de la unidad 23 de codificación de longitud variable y la definición de un bit final de grupo según cada realización. El bit final de un grupo se define como el bit final del último bloque en el grupo.

65 La unidad 23 de codificación de longitud variable según la primera realización codifica por longitud variable datos de bloque comprimidos en conformidad con la codificación de longitud variable adaptativa basada en contexto (CAVLC) definida en la norma MPEG-4 AVC/H.264.

La figura 6 es un diagrama para explicar el bit final de cada bloque en el flujo de datos de salida generado mediante

la unidad 23 de codificación de longitud variable según la primera realización. En la figura 6, los datos 1100 comprimidos de un bloque representado en la parte superior de la figura pueden contener una serie 1110 de omisión de macrobloque (denominada a continuación en el presente documento MbSkipRun) y una capa 1111 de macrobloque (denominada a continuación en el presente documento MacroblockLayer).

La MbSkipRun 1110 indica el número de bloques omitidos entre el bloque precedente y el bloque actual. Por consiguiente, cuando la MbSkipRun 1110 es 0, no se omite ninguno de los bloques entre el bloque precedente y el bloque actual. Por otra parte, la MacroblockLayer 1111 son datos comprimidos de bloque que contienen información tal como el tipo de codificación de bloque, información de vector de movimiento y coeficientes de DCT cuantificados. Un bloque omitido significa que su MacroblockLayer 1111 no se incluye en el flujo de datos de salida.

Cuando la imagen es una imagen intracodificada, la MbSkipRun 1110 siempre es "0" porque no se permite una omisión de bloque. Por tanto, en este caso, la MbSkipRun 1110 no se incluye en el flujo de datos de salida.

En primer lugar, se facilitará una descripción para el caso en el que el bloque para el que la MbSkipRun 1110 es 0 (incluyendo cualquier bloque contenido en una imagen intracodificada) es el último bloque en el grupo. En este caso, una secuencia 1200 de bits en el flujo de datos de salida contiene una secuencia 1202 de bits referente a los datos comprimidos del último bloque x y una secuencia 1201 de bits para los bloques (hasta $(x-1)$) que preceden al último bloque x (x es un número entero no menor que 1).

La secuencia 1202 de bits es una secuencia de bits generada codificando por longitud variable la MbSkipRun (=0) y MacroblockLayer del último bloque x . En la secuencia 1202 de bits, Código 0 es una secuencia de bits generada codificando por longitud variable la MbSkipRun, y Código M ($M = [1, N[x]]$) es una secuencia de bits generada codificando por longitud variable cada elemento de la MacroblockLayer. $N[x]$ representa el número de elementos contenidos en la MacroblockLayer del último bloque x .

En el ejemplo ilustrado, el bit final del último bloque x es el bit final de la secuencia de bits Código $N[x]$ que se indica mediante la flecha 1210.

A continuación, se facilitará una descripción del bit final para el caso en el que la MbSkipRun es $(L-1)$ (donde $(L-1) > 0$) y donde la MacroblockLayer del bloque sucesivo aparece antes de que la siguiente cabecera de imagen o cabecera de sector aparezca en el flujo de datos de salida. En este caso, una secuencia 1300 de bits en el flujo de datos de salida contiene una secuencia 1301 de bits referente al bloque $(x-L)$ inmediatamente anterior al bloque omitido y una secuencia 1302 de bits referente al bloque x que sigue inmediatamente al bloque omitido. La secuencia 1301 de bits es una secuencia de bits generada codificando por longitud variable la MbSkipRun y MacroblockLayer del bloque $(x-L)$. De manera similar, la secuencia 1302 de bits es una secuencia de bits generada codificando por longitud variable la MbSkipRun y MacroblockLayer del bloque x . El Código 0 correspondiente a la MbSkipRun contenida en la secuencia 1302 de bits es una secuencia de bits generada codificando por longitud variable el valor $(L-1)$.

En el ejemplo ilustrado, hay $(L-1)$ bloques $(x-L+1)$ a $(x-1)$, cuya MacroblockLayer no se incluye en el flujo de datos de salida, entre el bloque $(x-L)$ y el bloque x . Por consiguiente, cuando uno de los bloques $(x-L+1)$ a $(x-1)$ es el último bloque, el bit final de tales bloques es el bit final del código Código 0 correspondiente a la MbSkipRun = $(L-1)$ que se indica mediante la flecha 1310. Cuando el bloque x es el último bloque, el bit final del bloque x es el bit final del Código $N[x]$ contenido en la secuencia 1302 de bits que se indica mediante la flecha 1311.

A continuación, se facilitará una descripción del bit final para el caso en el que la MbSkipRun es $(L-1)$ (donde $(L-1) > 0$) y en el que la siguiente cabecera de sector perteneciente a la misma imagen aparece antes que la MacroblockLayer del bloque sucesivo en el flujo de datos de salida. En este caso, una secuencia 1400 de bits en el flujo de datos de salida contiene una secuencia 1401 de bits referente al bloque e inmediatamente anterior al bloque omitido y una secuencia 1402 de bits generada codificando por longitud variable la MbSkipRun correspondiente al bloque siguiente al bloque omitido $(y+1)$. La secuencia 1401 de bits es una secuencia de bits generada codificando por longitud variable la MbSkipRun y MacroblockLayer del bloque y . Por otra parte, el Código 0 contenido en la secuencia 1402 de bits es una secuencia de bits generada codificando por longitud variable el valor $(L-1)$.

La secuencia 1400 de bits contiene además una secuencia 1403 de bits (denominada a continuación en el presente documento RbspTrailingBits) que indica el final del sector y una cabecera 1404 de sector (denominada a continuación en el presente documento SliceHeader) que es una secuencia de bits que indica el inicio del siguiente sector.

La RbspTrailingBits 1403 es una secuencia de bits que tiene un valor predeterminado que indica el final de la secuencia de bits del sector que precede a la RbspTrailingBits. La RbspTrailingBits 1403 se establece de modo que el aparato de decodificación de vídeo en movimiento puede identificar el final del sector haciendo referencia al menos al primer bit de la RbspTrailingBits 1403. La cabecera 1404 de sector contiene un parámetro FirstMbInSlice que indica la dirección de bloque de inicio del sector. En el ejemplo ilustrado, se supone que el parámetro FirstMbInSlice es z .

En este caso, hay $(L-1)$ bloques $(y+1)$ a $(y+L-1 (=z-1))$, cuya MacroblockLayer no se incluye en el flujo de datos de salida, entre el bloque y y el bloque de inicio z del siguiente sector. Por consiguiente, cuando uno de los bloques $(y+1)$ a $(y+L-1 (=z-1))$ es el último bloque, el bit final de tales bloques es el bit inicial de la secuencia de bits RbspTrailingBits 1403 que se indica mediante la flecha 1410. En este caso, el bit final puede establecerse en alguna posición de bit en la secuencia de bits RbspTrailingBits 1403 distinta de la posición de bit inicial.

A continuación, se facilitará una descripción del bit final para el caso en el que MbSkipRun es $(L-1)$ (donde $(L-1) > 0$) y donde una cabecera de sector o cabecera de imagen perteneciente a otra imagen aparece antes que la MacroblockLayer del bloque sucesivo en el flujo de datos de salida. En este caso, una secuencia 1500 de bits en el flujo de datos de salida contiene una secuencia 1501 de bits referente al bloque e inmediatamente anterior al bloque omitido y una secuencia 1502 de bits generada codificando por longitud variable la MbSkipRun correspondiente al bloque siguiente al bloque omitido $(y+1)$. La secuencia 1501 de bits es una secuencia de bits generada codificando por longitud variable la MbSkipRun y MacroblockLayer del bloque y . Por otra parte, el Código 0 contenido en la secuencia 1502 de bits es una secuencia de bits generada codificando por longitud variable el valor $(L-1)$.

La secuencia 1500 de bits contiene además una secuencia de bits RbspTrailingBits 1503 que indica el final del sector, una cabecera 1504 de imagen que indica el inicio de la siguiente imagen, y una SliceHeader 1505 que es una secuencia de bits que indica el inicio del siguiente sector. En este caso, dado que el bloque de inicio del siguiente sector es el bloque de inicio de la imagen, el valor del parámetro FirstMbInSlice contenido en la SliceHeader 1505 es 0.

En el ejemplo ilustrado, hay $(L-1)$ bloques $(y+1)$ a $(y+L-1 (=w-1))$, cuya MacroblockLayer no se incluye en el flujo de datos de salida, entre el bloque y y el último bloque $(w-1)$ de la imagen, donde w es el número total de bloques contenidos en la imagen. Por consiguiente, cuando uno de los bloques $(y+1)$ a $(y+L-1 (=w-1))$ es el último bloque, el bit final de tales bloques es el bit inicial de la secuencia de bits RbspTrailingBits 1503 que se indica mediante la flecha 1510. En este caso, el bit final puede establecerse en alguna posición de bit en la secuencia de bits RbspTrailingBits 1503 distinta de la posición de bit inicial.

A continuación, se describirá la unidad 23 de codificación de longitud variable según la segunda realización. La unidad 23 de codificación de longitud variable según la segunda realización también codifica por longitud variable datos de bloque comprimidos en conformidad con la CAVLC definida en la norma MPEG-4 AVC/H.264. Sin embargo, la segunda realización difiere de la primera realización en la estructura de datos de bloque comprimidos; más específicamente, en vez de la MbSkipRun que indica el número de bloques omitidos, la segunda realización usa una bandera de omisión de macrobloque que indica si el bloque actual se omite o no. Además, los datos comprimidos de cada bloque se anexan con una bandera de división que indica si el bloque se divide o no.

La figura 7 es un diagrama para explicar el bit final de cada bloque en el flujo de datos de salida generado por la unidad 23 de codificación de longitud variable según la segunda realización. En la figura 7, los datos 2100 comprimidos de un bloque pueden contener la bandera 2110 de división (denominada a continuación en el presente documento MbSplitFlag), la bandera 2111 de omisión (denominada a continuación en el presente documento MbSkipFlag) y la MacroblockLayer 2112 en este orden a partir del encabezamiento del bloque.

La MbSplitFlag 2110 indica si el bloque actual (por ejemplo, un macrobloque de 16×16 píxeles) se divide en cuatro subbloques (por ejemplo, bloques de 8×8 píxeles). Por ejemplo, cuando se divide el bloque actual, la MbSplitFlag 2110 se establece en "1"; por otra parte, cuando no se divide el bloque actual o se omite el bloque actual, la MbSplitFlag 2110 se establece en "0".

La MbSkipFlag 2111 indica si el bloque actual se omite o no. Por ejemplo, cuando se omite el bloque actual, la MbSkipFlag 2111 se establece en "1"; por otra parte, cuando no se omite el bloque actual, la MbSkipFlag 2111 se establece en "0". La MacroblockLayer 2112 son datos comprimidos de bloque que contienen información tal como el tipo de codificación de bloque, información de vector de movimiento y coeficientes de DCT cuantificados. Un bloque omitido significa que su MacroblockLayer 2112 no se incluye en el flujo de datos de salida.

Cuando se divide el bloque actual, se considera que la MbSkipFlag 2111 se establece en "0", y por tanto, la MbSkipFlag 2111 puede suprimirse del flujo de datos de salida. En este caso, los datos comprimidos de los cuatro subbloques se almacenan en el flujo de datos de salida, inmediatamente a continuación de la MbSplitFlag 2110.

Cuando la imagen es una imagen intracodificada, la MbSkipFlag 2111 siempre es "0" porque no se permite una omisión de bloque. Por tanto, en este caso, la MbSkipFlag 2111 no se incluye en el flujo de datos de salida.

En primer lugar, se facilitará una descripción para el caso en el que el bloque para el que la MbSkipFlag 2111 es 0 (incluyendo cualquier bloque contenido en una imagen intracodificada) es el último bloque en el grupo. En este caso, una secuencia 2200 de bits en el flujo de datos de salida contiene una secuencia 2202 de bits referente a los datos comprimidos del último bloque x y una secuencia 2201 de bits para los bloques (hasta $(x-1)$) que preceden al último bloque x (x es un número entero no menor que 1). El Código 0 que sigue a la secuencia 2202 de bits indica una

secuencia de bits referente a los datos comprimidos del bloque (x+1).

La secuencia 2202 de bits es una secuencia de bits generada codificando por longitud variable la MbSplitFlag (=0), MbSkipFlag (=0) y MacroblockLayer del último bloque x. En la secuencia 2202 de bits, Sp es una secuencia de bits generada codificando por longitud variable la MbSplitFlag, y Sk una secuencia de bits generada codificando por longitud variable la MbSkipFlag. Además, el Código M ($M = [1, N[x]]$) es una secuencia de bits generada codificando por longitud variable cada elemento de la MacroblockLayer. N[x] representa el número de elementos contenidos en la MacroblockLayer del último bloque x.

En el ejemplo ilustrado, el bit final del último bloque x es el bit final de la secuencia de bits Código N[x] que se indica mediante la flecha 2210.

A continuación, se facilitará una descripción del bit final para el caso en el que una serie de (L-1) bloques (donde $(L-1) > 0$) para los que el valor de MbSkipFlag es "1", es decir, bloques omitidos, va seguida por un bloque para el que el valor de MbSkipFlag es "0", es decir, un bloque no omitido. Una secuencia 2300 de bits en el flujo de datos de salida contiene una secuencia 2301 de bits referente a los datos comprimidos de un bloque no omitido (x-L). La secuencia 2300 de bits contiene además, a continuación de la secuencia 2301 de bits, una secuencia 2302 de bits referente a la MbSplitFlag (=0) y MbSkipFlag (=1) de cada uno de los bloques omitidos (x-L+1) a (x-1) y los datos comprimidos del bloque sucesivo x. El Código 0 que sigue a la secuencia 2302 de bits indica una secuencia de bits referente a los datos comprimidos del bloque (x+1).

En el ejemplo ilustrado, hay (L-1) bloques (x-L+1) a (x-1), cuya MacroblockLayer no se incluye en el flujo de datos de salida, entre el bloque (x-L) y el bloque x. Por consiguiente, cuando uno de los bloques (x-L+p) ($p = \{1, \dots, L-1\}$) es el último bloque, el bit final de tales bloques es el bit final del p-ésimo código desde la izquierda entre los (L-1) códigos de MbSkipFlag (=1). Cuando el bloque x es el último bloque, el bit final del bloque x es el bit final del Código N[x] contenido en la secuencia 2302 de bits que se indica mediante la flecha 2311.

A continuación, se facilitará una descripción del bit final para el caso en el que una serie de (L-1) bloques (donde $(L-1) > 0$) para los que el valor de MbSkipFlag es "1" va seguida por la siguiente cabecera de sector perteneciente a la misma imagen. En este caso, una secuencia 2400 de bits en el flujo de datos de salida contiene una secuencia 2401 de bits referente a los datos comprimidos del bloque y que precede inmediatamente a la serie de bloques omitidos. La secuencia 2400 de bits contiene además, a continuación de la secuencia 2401 de bits, una secuencia 2402 de bits generada codificando por longitud variable la MbSplitFlag (=0) y MbSkipFlag (=1) de cada uno de los bloques omitidos que se inicia con el bloque (y+1). La secuencia 2401 de bits es una secuencia de bits generada codificando por longitud variable la MbSplitFlag (=0), MbSkipFlag (=0) y MacroblockLayer del bloque y.

La secuencia 2400 de bits contiene además una RbspTrailingBits 2403 que indica el final del sector y una SliceHeader 2404 que indica el inicio del siguiente sector. La RbspTrailingBits 2403 es una secuencia de bits que tiene un valor predeterminado que indica el final de la secuencia de bits del sector que precede a la RbspTrailingBits. La RbspTrailingBits 2403 se establece de modo que el aparato de decodificación de video en movimiento puede identificar el final del sector haciendo referencia al menos al primer bit de la RbspTrailingBits 2403. La cabecera 2404 de sector contiene un parámetro FirstMbInSlice que indica la dirección de bloque de inicio del sector. En el ejemplo ilustrado, se supone que el parámetro FirstMbInSlice es z.

En este caso, hay (L-1) bloques (y+1) a (y+L-1 (=z-1)), cuya MacroblockLayer no se incluye en el flujo de datos de salida, entre el bloque y el bloque de inicio z del siguiente sector. Por consiguiente, cuando uno de los bloques (y+p) ($p = \{1, \dots, L-2\}$) es el último bloque, el bit final de tales bloques es el bit final del p-ésimo código desde la izquierda entre los (L-2) códigos de MbSkipFlag (=1). Cuando el bloque (z-1) es el último bloque, el bit final es el bit inicial de la secuencia de bits RbspTrailingBits 2403 que se indica mediante la flecha 2411. En este caso, el bit final puede establecerse en alguna posición de bit en la secuencia de bits RbspTrailingBits 2403 distinta de la posición de bit inicial.

A continuación, se facilitará una descripción del bit final para el caso en el que una serie de (L-1) bloques (donde $(L-1) > 0$) para los que el valor de MbSkipFlag es "1" va seguida por una cabecera de sector o cabecera de imagen perteneciente a otra imagen. En este caso, una secuencia 2500 de bits en el flujo de datos de salida contiene una secuencia 2501 de bits referente a los datos comprimidos del bloque y que precede inmediatamente a la serie de bloques omitidos. La secuencia 2500 de bits contiene además, a continuación de la secuencia 2501 de bits, una secuencia 2502 de bits generada codificando por longitud variable la MbSplitFlag (=0) y MbSkipFlag (=1) de cada uno de los bloques omitidos que se inicia con el bloque (y+1). La secuencia 2501 de bits es una secuencia de bits generada codificando por longitud variable la MbSplitFlag (=0), MbSkipFlag (=0) y MacroblockLayer del bloque y.

La secuencia 2500 de bits contiene además una RbspTrailingBits 2503 que indica el final del sector, una cabecera 2504 de imagen que indica el inicio de la siguiente imagen y una SliceHeader 2505 que es una secuencia de bits que indica el inicio del siguiente sector. En este caso, dado que el bloque de inicio del siguiente sector es el bloque de inicio de la imagen, el valor del parámetro FirstMbInSlice contenido en la SliceHeader 2505 es 0.

En el ejemplo ilustrado, hay (L-1) bloques (y+1) a (y+L-1 (=w-1)), cuya MacroblockLayer no se incluye en el flujo de datos de salida, entre el bloque y el último bloque (w-1) de la imagen, donde w es el número total de bloques contenidos en la imagen. Por consiguiente, cuando uno de los bloques (y+p) ($p = \{1, \dots, L-2\}$) es el último bloque, el bit final de tales bloques es el bit final del p-ésimo código desde la izquierda entre los (L-2) códigos de MbSkipFlag (=1). Cuando el bloque (w-1) es el último bloque, el bit final es el bit inicial de la secuencia de bits RbspTrailingBits 2503 que se indica mediante la flecha 2511. En este caso, el bit final puede establecerse en alguna posición de bit en la secuencia de bits RbspTrailingBits 2503 distinta de la posición de bit inicial.

Según un ejemplo modificado de la segunda realización, puede suprimirse MbSplitFlag. Si se suprime MbSplitFlag, el bit final del último bloque se determina de la misma manera que se describió anteriormente con referencia a la figura 7.

A continuación, se describirá la unidad 23 de codificación de longitud variable según la tercera realización. La unidad 23 de codificación de longitud variable según la tercera realización codifica por longitud variable datos de bloque comprimidos en conformidad con la codificación aritmética binaria adaptativa basada en contenido (CABAC) definida en la norma MPEG-4 AVC/H.264. La CABAC es una forma de codificación aritmética.

A continuación, con referencia a las figuras 8A y 8B, se facilitará una explicación del bit final de cada bloque en el flujo de bits de datos de salida. La figura 8A es un diagrama que representa la estructura de datos comprimidos de un bloque. Tal como se representa en la figura 8A, los datos comprimidos 3100 de un bloque pueden contener una MbSkipFlag 3110, una MacroblockLayer 3111 y una bandera 3112 de fin de sector (denominada a continuación en el presente documento EndOfSliceFlag) en este orden a partir del encabezamiento del bloque.

La MbSkipFlag 3110 indica si el bloque actual se omite o no. Por ejemplo, cuando se omite el bloque actual, la MbSkipFlag 3110 se establece en "1"; por otra parte, cuando no se omite el bloque actual, la MbSkipFlag 3110 se establece en "0". La MacroblockLayer 3111 son datos comprimidos de bloque que contienen información tal como el tipo de codificación de bloque, información de vector de movimiento y coeficientes de DCT cuantificados. Un bloque omitido significa que su MacroblockLayer 3111 no se incluye en el flujo de datos de salida.

Cuando la imagen es una imagen intracodificada, la MbSkipFlag 3110 siempre es "0" porque no se permite una omisión de bloque. Por tanto, en este caso, la MbSkipFlag 3110 no se incluye en el flujo de datos de salida.

La EndOfSliceFlag 3112 indica si el bloque actual es el último bloque del sector. Por ejemplo, cuando el bloque actual es el último bloque del sector, la EndOfSliceFlag 3112 se establece en "1"; por otra parte, cuando el bloque actual no es el último bloque, la EndOfSliceFlag 3112 se establece en "0".

La figura 8B es un diagrama para explicar la relación entre los datos comprimidos y la secuencia de bits convertida mediante codificación de longitud variable. A diferencia de la CAVLC, la CABAC aplica una codificación aritmética a cada bit de los datos comprimidos. Por consiguiente, los bits en la secuencia de bits convertida no se corresponden uno a uno con los bits en los datos comprimidos antes de la conversión. Un bit en la secuencia de bits convertida puede corresponder a más de un bit en los datos comprimidos.

Tal como se ilustra en la figura 8B, una secuencia 3210 de bits como un ejemplo de la secuencia de bits en el flujo de datos de salida contiene 12 bits. En este caso, un conjunto 3211 de puntos representa bins contenidos en los datos comprimidos que corresponden a los bits respectivos en la secuencia 3210 de bits y que se obtienen decodificando aritméticamente la secuencia 3210 de bits. Los bins son representaciones binarias de los datos comprimidos, y cada uno representa un bit o una secuencia de bits en los datos comprimidos que corresponde a una palabra de código. En el ejemplo ilustrado, la secuencia 3210 de bits se corresponde con los bins 0 a 7. Los bins 0 a 2 están contenidos en el bloque 0, los bins 3 a 5 están contenidos en el bloque 1, el bin 6 está contenido en el bloque 2 y el bin 7 está contenido en el bloque 3. Cada punto dado en el conjunto 3211 de puntos indica que el bit contenido en la secuencia 3210 de bits convertida y ubicado directamente por encima de ese punto dado se usa para decodificar el bin ubicado a la izquierda de ese punto dado. Por ejemplo, el bin 0 se decodifica usando los bits de orden cero y primero. Igualmente, el bin 1 se decodifica usando el primer bit. Por otra parte, el bin 5 se decodifica usando los bits tercero a 11°.

El bit final de cada bloque codificado mediante CABAC es el último bit en la secuencia de bits usada para decodificar el bin x contenido en los datos comprimidos de ese bloque. En este caso el bin x corresponde al último bit cuando los datos comprimidos del bloque se binarizan en secuencia. En el ejemplo ilustrado en la figura 8B, el bit final del bloque 0 es el tercer bit en la secuencia 3210 de bits que se indica mediante la flecha 3250. Por otra parte, el bit final de cada uno de los bloques 1 a 3 es el 11° bit en la secuencia 3210 de bits que se indica mediante la flecha 3251. De esta manera, una pluralidad de bloques puede compartir el mismo bit final.

Cuando cada bloque se codifica mediante CABAC, el bit final de la versión binarizada de los datos comprimidos del bloque es invariablemente una EndOfSliceFlag. Cuando la imagen contiene w bloques, el flujo de datos de salida invariablemente contiene w MbSkipFlags y w EndOfSliceFlags. Como resultado, el bit final del x-ésimo bloque en la imagen es el último bit en la secuencia de bits que se usa para decodificar la x-ésima EndOfSliceFlag.

Las figuras 9A a 9C son un diagrama de flujo que ilustra el procedimiento para identificar el bit final de cada grupo dado en el flujo de bits de datos de salida según la tercera realización. La unidad 23 de codificación de longitud variable en el aparato de codificación de vídeo en movimiento deriva, según el siguiente diagrama de flujo, la posición $\text{BitEnd}(i,j)$ del bit final del último bloque perteneciente al j -ésimo grupo $G(i,j)$ en la i -ésima imagen en el flujo de bits de datos de salida. Cuando el bit correspondiente a la posición $\text{BitEnd}(i,j)$ se introduce en la memoria intermedia de transmisión en el aparato 10 de codificación de vídeo en movimiento, el aparato 10 de codificación de vídeo en movimiento determina que todos los bits necesarios para decodificar el grupo $G(i,j)$ se han introducido en la memoria intermedia de transmisión en el aparato 10 de codificación de vídeo en movimiento.

El diagrama de flujo se inicia inmediatamente después de que se haya codificado la cabecera (cabecera de imagen y cabecera de sector) de la i -ésima imagen. En primer lugar, la unidad 23 de codificación de longitud variable inicializa las variables j , m y c (etapa S201). Las variables j , m y c representan el índice de un grupo dado en la imagen, el índice de un bloque dado en la imagen y un recuento de acarreo. El recuento de acarreo c se describirá más adelante.

A continuación, realizando una compresión del bloque de índice m y una binarización y un modelado de contexto de los datos comprimidos, la unidad 23 de codificación de longitud variable obtiene una matriz de bins $B[]$ a la que va a aplicarse una codificación aritmética (correspondiente a una $\text{EncodingDecision}()$ definida en la norma MPEG-4 AVC/H.264) (etapa S202). En este caso, la compresión es un procedimiento para eliminar una EndOfSliceFlag que indica si el bloque es el último bloque en el sector. El número total de bins en el bloque m al que va a aplicarse la codificación aritmética se expresa como $\text{getNumBins}(m)$. La unidad 23 de codificación de longitud variable establece el índice $b0$ del bin objetivo en 0 y el índice $b1$ del último bin en el bloque m en $\text{getNumBins}(m)$.

La unidad 23 de codificación de longitud variable obtiene la posición de bit final p del flujo de bits antes de la codificación aritmética del $b0$ -ésimo bin (etapa S203). Luego, la unidad 23 de codificación de longitud variable codifica aritméticamente el $b0$ -ésimo bin (etapa S204). Como resultado, un bit recién generado, por ejemplo, de entre 0 a 7 bits, se añade al final del flujo de bits. Luego, la unidad 23 de codificación de longitud variable obtiene la posición de bit final q del flujo de bits después de la codificación aritmética del $b0$ -ésimo bin (etapa S205).

Después de la etapa S205, la unidad 23 de codificación de longitud variable determina si el recuento de acarreo c es 1 o mayor o no (etapa S206). El recuento de acarreo c es un recuento que se incrementa en 1 cuando no se ha añadido un nuevo bit al flujo de bits después de codificar aritméticamente la EndOfSliceFlag del último bloque en un grupo dado. Es decir, cuando el recuento de acarreo c es 1 o mayor, el recuento de acarreo c indica que todos los bits necesarios para decodificar el grupo x ($x = j-1, \dots, j-c$) no se han emitido en el flujo de bits.

Si el recuento de acarreo c es 1 o mayor (Sí en la etapa S206), la unidad 23 de codificación de longitud variable compara la posición de bit final p del flujo de bits antes de la codificación aritmética del $b0$ -ésimo bin con la posición de bit final q del flujo de bits después de la codificación aritmética del $b0$ -ésimo bin (etapa S207).

Si la posición de bit p es diferente de la posición de bit q (Sí en la etapa S207), esto significa que se genera un nuevo bit codificando aritméticamente el $b0$ -ésimo bin. En este caso, el $b0$ -ésimo bin y el bit correspondiente a la EndOfSliceFlag del último bloque en el grupo x ($x = j-1, \dots, j-c$) que se codificó aritméticamente con anterioridad pero no se emitió en el flujo de bits se emiten en el flujo de bits. Es decir, todos los bits necesarios para decodificar el grupo x se han emitido en la posición q . Luego, la unidad 23 de codificación de longitud variable establece, como la posición de bit q , la posición de bit final $\text{BitEnd}()$ de la secuencia de bits necesaria para decodificar el grupo que se ha emitido en este caso (etapa S208). Los detalles del procesamiento en la etapa S208 se describirán más adelante.

Todos los bits necesarios para decodificar cada grupo cuyo índice es menor que j se han emitido ahora en el flujo de bits. Por tanto, la unidad 23 de codificación de longitud variable restablece el recuento de acarreo c en 0 (etapa S209).

Después de la etapa S209, la unidad 23 de codificación de longitud variable determina si todos los bins en la matriz de bins $B[]$ que van a codificarse aritméticamente se han procesado, es decir, si $b0$ es igual a $b1$ (etapa S210). Si $b0$ es menor que $b1$ (No en la etapa S210), la unidad 23 de codificación de longitud variable incrementa $b0$ en 1, y repite el procedimiento a partir de la etapa S203. Por tanto, la unidad 23 de codificación de longitud variable realiza un procesamiento en cualquier bin que queda en la matriz de bins $B[]$. Por otra parte, si $b0$ es igual a $b1$ (Sí en la etapa S210), la unidad 23 de codificación de longitud variable incrementa $b0$ en 1, y pasa a la etapa S211. Después de una comparación con $b1$, $b0$ se incrementa en 1, ya sea menor que o igual a $b1$, pero cuando $b0$ es igual a $b1$, $b0$ se restablece posteriormente en 0 en la etapa S202.

Por otra parte, si el recuento de acarreo c es 0 en la etapa S206 (No en la etapa S206), la unidad 23 de codificación de longitud variable pasa directamente a la etapa S210 sin las etapas de procesamiento S208 a S209. Además, si la posición de bit p es la misma que la posición de bit q en la etapa S207 (No en la etapa S207), no se genera ningún nuevo bit codificando aritméticamente el $b0$ -ésimo bin. También en este caso, la unidad 23 de codificación de longitud variable pasa a la etapa S210 sin las etapas de procesamiento S208 y S209.

5 Tal como se ilustra en la figura 9B, la unidad 23 de codificación de longitud variable obtiene la posición de bit final p del flujo de bits antes de la codificación aritmética de EndOfSliceFlag (etapa S211). Luego, la unidad 23 de codificación de longitud variable codifica aritméticamente EndOfSliceFlag (etapa S212), y obtiene la posición de bit final q del flujo de bits después de la codificación aritmética de EndOfSliceFlag (etapa S213).

10 La unidad 23 de codificación de longitud variable determina si el recuento de acarreo c es 1 o mayor o no y si la posición de bit final p del flujo de bits antes de la codificación aritmética de EndOfSliceFlag es diferente de la posición de bit final q del flujo de bits después de la codificación aritmética de EndOfSliceFlag (etapa S214). Si el recuento de acarreo c es 1 o mayor, y si la posición de bit final p es diferente de la posición de bit final q (Sí en la etapa S214), esto significa que la EndOfSliceFlag del último bloque en el grupo x ($x = j-1, \dots, j-c$) se emite en el flujo de bits. Dicho de otro modo, todos los bits necesarios para decodificar el grupo x se han emitido en el flujo de bits. Luego, la unidad 23 de codificación de longitud variable establece, como la posición de bit q, la posición de bit final BitEnd() de la secuencia de bits necesaria para decodificar el grupo que se ha emitido en este caso (etapa S215), y después de eso restablece el recuento de acarreo c en 0 (etapa S216).

20 Después de la etapa S216, o después de que se determina en la etapa S214 que el recuento de acarreo c es 0 o la posición de bit final p es la misma que la posición de bit final q (No en la etapa S214), la unidad 23 de codificación de longitud variable pasa a determinar si el bloque actual m es el último bloque en el grupo j (etapa S217). GetLastBlock(j) representa el índice del último bloque en el grupo j. Si el bloque actual m no es el último bloque en el grupo j (No en la etapa S217), esto significa que quedan uno o más bloques por procesarse en el grupo j. Luego, la unidad 23 de codificación de longitud variable incrementa m en 1 y repite el procedimiento a partir de la etapa S202.

25 Por otra parte, si el bloque actual m es el último bloque en el grupo j (Sí en la etapa S217), la unidad 23 de codificación de longitud variable incrementa m en 1. Luego, la unidad 23 de codificación de longitud variable compara la posición de bit final p del flujo de bits antes de la codificación aritmética de EndOfSliceFlag con la posición de bit final q del flujo de bits después de la codificación aritmética de EndOfSliceFlag (etapa S218).

30 Si la posición de bit final p es diferente de la posición de bit final q (Sí en la etapa S218), esto significa que se genera un nuevo bit codificando aritméticamente la EndOfSliceFlag. Por tanto, esto significa que todos los bits necesarios para decodificar el último bloque en el grupo j se han emitido en el flujo de bits. Luego, la unidad 23 de codificación de longitud variable establece, como la posición de bit q, la posición de bit final BitEnd() de la secuencia de bits necesaria para decodificar el grupo j que se ha emitido (etapa S219), e incrementa el índice j del grupo en 1 (etapa S220).

35 Por otra parte, si la posición de bit final p es la misma que la posición de bit final q (No en la etapa S218), no se genera ningún nuevo bit codificando aritméticamente la EndOfSliceFlag. Por consiguiente, la unidad 23 de codificación de longitud variable incrementa el recuento de acarreo c en 1 y también incrementa el índice de grupo j en 1 (etapa S221).

40 Después de la etapa S220 o S221, la unidad 23 de codificación de longitud variable determina si el bloque actual m es el último bloque en la imagen (etapa S222). GetNumBlockInPic() representa el número total de bloques en la imagen.

45 Si el bloque actual m no es el último bloque en la imagen (No en la etapa S222), esto significa que quedan uno o más bloques por procesarse en la imagen, y la unidad 23 de codificación de longitud variable repite el procedimiento a partir de la etapa S202. Por otra parte, si el bloque actual m es el último bloque en la imagen (Sí en la etapa S222), la unidad 23 de codificación de longitud variable termina el procedimiento de identificación de bit final determinando que la codificación de todos los bloques en la imagen se ha completado.

50 A continuación se describirá el flujo del procedimiento de establecimiento de posición de bit final realizado en las etapas S208, S215 y S219 con referencia a la figura 9C.

55 La unidad 23 de codificación de longitud variable establece, como la posición q, la posición de bit final BitEnd(i,(j-c)) de la secuencia de bits necesaria para decodificar el (j-c)-ésimo grupo en la i-ésima imagen (etapa S250). Después de eso, la unidad 23 de codificación de longitud variable determina si el recuento de acarreo c es 0 o menor o no (etapa S251). Si el recuento de acarreo c es mayor de 0 (No en la etapa S251), esto significa que queda un grupo para el que aún ha de establecerse la posición de bit final BitEnd(). Luego, la unidad 23 de codificación de longitud variable decrementa el recuento de acarreo en 1, y después de eso lleva a cabo la etapa S250 una vez más. Por otra parte, si el recuento de acarreo c es 0 o menor (Sí en la etapa S251), ya no queda ningún grupo para el que aún ha de establecerse la posición de bit final BitEnd(). Por tanto, la unidad 23 de codificación de longitud variable termina el procedimiento de establecimiento de posición de bit final.

65 A continuación, se describirá la unidad 23 de codificación de longitud variable según la cuarta realización. La unidad 23 de codificación de longitud variable según la cuarta realización codifica por longitud variable datos de bloque comprimidos según el método divulgado en el documento no de patente 2 (JCTVC-A116, "Description of video

coding technology proposal by Fraunhofer HHI”, Joint Collaborative Team on Video Coding de ITU-T SG16 WP3 y ISO/CEI JTC1/SC29/WG11, abril de 2010).

El método divulgado en el documento no de patente 2 usa básicamente una tecnología de codificación aritmética al igual que la CABAC. El método divulgado, sin embargo, usa una pluralidad de codificadores aritméticos basándose en las probabilidades estimadas de los bits individuales en datos comprimidos binarizados, en vez de codificar aritméticamente los datos comprimidos binarizados bit por bit. A cada codificador aritmético se le asigna una probabilidad estimada diferente, y cada bit en los datos comprimidos binarizados se codifica aritméticamente mediante el codificador aritmético correspondiente a su probabilidad estimada.

Según este método, con el fin de procesar una secuencia de bits de una probabilidad estimada fija, cada codificador aritmético puede codificar por longitud variable los datos comprimidos mapeando n bits de introducción para dar m bits de emisión, en vez de realizar realmente operaciones de codificación aritmética. En este caso, m y n son cada uno un número entero no menor de 1. Por motivos de conveniencia, en la presente memoria descriptiva, los m bits de salida se denominan una palabra. Los datos comprimidos codificados por longitud variable se emiten en el flujo de bits de datos de salida palabra por palabra.

En el método divulgado en el documento no de patente 2, al igual que en la CABAC, los bits en la secuencia de bits convertida no se corresponden uno a uno con los bits en los datos comprimidos antes de la conversión. Además, una palabra correspondiente a un bin que se introdujo en el codificador aritmético en un momento anterior puede aparecer en el flujo de bits de datos de salida en una posición posterior a la de una palabra correspondiente a un bin que se introdujo en el codificador aritmético en un momento posterior.

La figura 10 es un diagrama que ilustra esquemáticamente la configuración de una unidad 700 de codificación de entropía proporcionada en la unidad 23 de codificación de longitud variable según la cuarta realización. La unidad 700 de codificación de entropía codifica por entropía los datos comprimidos del bloque de introducción, y emite el flujo resultante. Con este propósito, la unidad 700 de codificación de entropía incluye una unidad 701 de binarización, una unidad 702 de modelado de contexto, una unidad 703 de cuantificación de probabilidad, un número, K , de unidades 704-1 a 704- K de codificación de bin (donde K es un número entero no menor de 2) y una unidad 705 de multiplexación.

La unidad 701 de binarización binariza cada símbolo contenido en los datos comprimidos del bloque de introducción, y expresa cada símbolo mediante una secuencia de bits. Con este propósito, la unidad 701 de binarización binariza cada símbolo en conformidad, por ejemplo, con la norma MPEG-4 AVC/H.264. Cuando el tipo de macrobloque $mType$ en un sector I es un símbolo “ $IN \times N$ ” que indica una intrapredicción 4×4 o una intrapredicción 8×8 , por ejemplo, el símbolo se expresa mediante “0”. Por otra parte, cuando el tipo de macrobloque $mType$ es un símbolo “ $IPCM$ ” que indica una codificación PCM, el símbolo se expresa mediante “11”. La secuencia de bits emitida desde la unidad 701 de binarización se introduce en la unidad 702 de modelado de contexto.

La unidad 702 de modelado de contexto asigna un contexto a cada bit individual en la secuencia de bits introducida desde la unidad 701 de binarización. El contexto se determina, por ejemplo, basándose en los datos comprimidos de bloques cercanos. Usando un contexto diferente cuando los datos comprimidos del bloque que va a codificarse por entropía son similares a los datos comprimidos de sus bloques cercanos y cuando no son similares, mejora la eficiencia de la codificación aritmética. Para cada bit en la secuencia de bits, la unidad 702 de modelado de contexto emite su valor (0 o 1), el bit menos probable (LPB) del contexto asignado, y la probabilidad de LPB, y el bit, LPB, y la probabilidad de LPB así emitidos se introducen en la unidad 703 de cuantificación de probabilidad. El LPB representa un valor cuya frecuencia de aparición es menor entre los bits en la secuencia de bits que se han emitido previamente al emparejarse con ese contexto. La probabilidad de LPB es la probabilidad de aparición de LPB. Por ejemplo, supóngase que los 10 bits contenidos en la secuencia de bits emitida previamente son [0, 0, 0, 1, 1, 1, 1, 1, 1, 1]; entonces, el LPB es 0, y la probabilidad de LPB es 0,3.

La unidad 703 de cuantificación de probabilidad cuantifica la probabilidad de LPB de cada bit emitido desde la unidad 702 de modelado de contexto. Por ejemplo, la unidad 703 de cuantificación de probabilidad cuantifica el valor de probabilidad continuo desde 0 hasta 1 para dar un conjunto de K valores discretos. Por ejemplo, cuando la probabilidad de LPB es 0,3, y K es 4, la unidad 703 de cuantificación de probabilidad cuantifica el valor de probabilidad de LPB de 0,3 a 0,25 (= 1/4). Cada bit emitido desde la unidad 703 de cuantificación de probabilidad se introduce en una de las K unidades 704-1 a 704- K de codificación de bin que corresponde al valor cuantificado de la probabilidad de LPB de ese bit.

Las unidades 704-1 a 704- K de codificación de bin codifican aritméticamente los bits correspondientes a los valores cuantificados de probabilidades de LPB diferentes respectivamente. Por ejemplo, cuando el valor cuantificado de la probabilidad de LPB es 1/4, tal como se describió anteriormente, el bit correspondiente a esa probabilidad de LPB se introduce en una de las unidades 704-1 a 704- K de codificación de bin que corresponde al valor de probabilidad discreto de 1/4. Luego, la unidad de codificación de bin añade el bit a la secuencia de bits almacenada en una entrada de probabilidad de LPB cuantificada asociada con esa unidad de codificación de bit entre las secuencias de bits almacenadas en entradas designadas respectivamente en una memoria intermedia mantenida en la unidad 705

de multiplexación. La unidad de codificación de bin convierte la secuencia de bits en una palabra cada vez que la secuencia de bits pasa a estar lista para la conversión.

5 La unidad 705 de multiplexación tiene, por ejemplo, una memoria intermedia y almacena en la memoria intermedia la secuencia de bits escrita por cada unidad de codificación de bin y la palabra correspondiente a esa secuencia de bits. La unidad 705 de multiplexación retiene la palabra en la memoria intermedia hasta que la palabra pasa a estar lista para su emisión. Cuando la palabra pasa a estar lista para su emisión, la unidad 705 de multiplexación emite la palabra como un flujo de bits de datos de salida.

10 Dado que puede suponerse que cada bit introducido en la misma unidad de codificación de bin tiene el mismo LPB, cada unidad de codificación de bin y la unidad de multiplexación pueden codificar por longitud variable la secuencia de bits de la pluralidad de bits de introducción como una secuencia de bins, en vez de codificar aritméticamente cada bit individual. En ese caso, una tabla de mapeo que proporciona un mapeo de secuencias de bits (secuencias de bins) a palabras se almacena en cada unidad de codificación de bin. Luego, haciendo referencia a la tabla de mapeo, la unidad de codificación de bin convierte la secuencia de bins en la palabra correspondiente. Por ejemplo, cuando el LPB es "1", y la probabilidad de LPB es 0,15 (es decir, la probabilidad de aparición de "0" es 0,85), la secuencia de bins de introducción "0000" se convierte en "1", y la secuencia de bins de introducción "11" en "00001".

15 A continuación, con referencia a la figura 11, se facilitará una explicación del bit final de cada bloque en el flujo de bits de datos de salida. La figura 11 es un diagrama para explicar la relación entre los datos comprimidos y la secuencia de bits convertida mediante codificación de longitud variable. La estructura de los datos comprimidos del bloque es la misma que la representada en la figura 8A.

20 Tal como se ilustra en la figura 9, una secuencia 4100 de bits como ejemplo de la secuencia de bits en el flujo de datos de salida contiene 15 bits. En este caso, un conjunto de puntos 4101 representa bins contenidos en los datos comprimidos que corresponden a los bits respectivos en la secuencia 4100 de bits y que se obtienen decodificando aritméticamente la secuencia 4100 de bits. En el ejemplo ilustrado, la secuencia 4100 de bits se corresponde con los bins 0 a 10. Los bins 0 a 2 están contenidos en el bloque 0, los bins 3 a 5 están contenidos en el bloque 1, los bins 6 y 7 están contenidos en el bloque 2 y los bins 8 a 10 están contenidos en el bloque 3. Cada punto dado en el conjunto de puntos 4101 indica que el bit contenido en la secuencia 4100 de bits convertida y ubicado directamente por encima de ese punto dado se usa para decodificar el bin ubicado a la izquierda de ese punto dado. Por ejemplo, los bins 0, 5, y 7 se decodifican cada uno usando los bits de orden cero a tercero. Dicho de otro modo, tres bits en los bins 0, 5 y 7 se convierten en los cuatro bits en las posiciones 0 a 3 en la secuencia 4100 de bits.

25 Tal como puede observarse a partir del conjunto de puntos 4101, la secuencia de bits (bit 0 a bit 3) necesaria para decodificar el último bin 5 perteneciente al bloque 1 aparece en el flujo 4100 de bits de datos de salida anterior a la secuencia de bits (bit 7 y bit 8) necesaria para decodificar el primer bin 3 perteneciente al bloque 1. De esta manera, a diferencia de la tercera realización, en la cuarta realización, hay casos en los que el orden de los bins en cada bloque se invierte con respecto al orden de los bits correspondientes en el flujo de bits de datos de salida.

30 El bit final de cada bloque es de la siguiente manera.

35 Tal como en el caso en el que cada bloque se codifica mediante CABAC, el primer bit y el bit final de la versión binarizada de los datos comprimidos del bloque son invariablemente una MbSkipFlag y una EndOfSliceFlag, respectivamente. Cuando la imagen contiene w bloques, el flujo de bits de datos de salida contiene invariablemente w MbSkipFlags y w EndOfSliceFlags.

40 En la presente realización, sin embargo, el bit final del bloque x codificado por longitud variable es el último bit de la palabra A cuya posición en el flujo de bits de datos de salida es la última entre las palabras necesarias para decodificar los bits respectivos en los datos comprimidos binarizados del bloque x. En el ejemplo ilustrado en la figura 11, el bit final de cada uno de bloques 0, 1 y 3 es el octavo bit en la secuencia 4100 de bits que se indica mediante la flecha 4200. Igualmente, el bit final de bloque 3 es el 14º bit en la secuencia 4100 de bits que se indica mediante la flecha 4201.

45 La figura 12 es un diagrama para explicar el comportamiento de las unidades de codificación de bin y la unidad de multiplexación en la unidad de codificación de longitud variable según la cuarta realización. Las unidades 704-1 a 704-K de codificación de bin convierten cada una secuencia de bins de introducción compuesta por bits que tienen la misma probabilidad de LPB cuantificada en una palabra haciendo referencia a un mapa 800 de conversión para convertir secuencias de bins de introducción en palabras de salida correspondientes. Por motivos de simplicidad de explicación, se supone que la unidad 705 de multiplexación usa el mismo mapa de conversión para todas las secuencias de bins introducidas desde las unidades 704-1 a 704-K de codificación de bin en la unidad 705 de multiplexación. Por ejemplo, la secuencia de bins "000" se convierte en una palabra "1".

50 En el ejemplo ilustrado, un índice 802 de probabilidad que indica la probabilidad de LPB cuantificada se anexa a cada bit en la secuencia 801 de bits de introducción mediante la unidad 703 de cuantificación de probabilidad. En el ejemplo ilustrado, se supone que la probabilidad de LPB se cuantifica para uno de cuatro valores, y cada

probabilidad de LPB cuantificada se asocia con uno de los índices de probabilidad 0 a 3. Los bits que constituyen la secuencia 801 de bits de introducción se introducen en secuencia, empezando con el primer bit 850, en las unidades de codificación de bin respectivas correspondientes a los índices de probabilidad de los bits respectivos. Por ejemplo, cada bit cuyo índice de probabilidad es "0" se introduce en la unidad 704-1 de codificación de bin, y cada bit cuyo índice de probabilidad es "1" se introduce en la unidad 704-2 de codificación de bin.

Las tablas 860 a 863 representan los datos almacenados en la memoria intermedia de la unidad 705 de multiplexación cuando los bits 850 a 853 se introducen en las unidades de codificación de bin correspondientes.

En las tablas 860 a 863, cada fila corresponde a una entrada almacenada en la memoria intermedia de la unidad 705 de multiplexación. En cada tabla, las entradas están dispuestas de arriba abajo en el orden en que se crearon las entradas. La unidad 705 de multiplexación emite las palabras desde las entradas respectivas en la memoria intermedia secuencialmente, cada una en forma de un flujo de bits de datos de salida, yendo de arriba abajo. En cada tabla, "N.º" indica el índice de cada entrada en la memoria intermedia. Además, "Índ." indica el índice de la unidad de codificación de bin correspondiente; en el ejemplo ilustrado, para facilitar la comprensión, el índice de cada unidad de codificación de bin tiene el mismo valor que el índice de probabilidad anexado al bit de introducción que se introduce en esa unidad de codificación de bin. "INTROD." indica la secuencia de bits de introducción (secuencia de bins). "PALABRA" indica la palabra correspondiente a la secuencia de bits de introducción. Si no hay una palabra correspondiente para la secuencia de bits de introducción (por ejemplo, cuando la secuencia de bits de introducción es "00"), la palabra se designa "N/A" en cada tabla.

Se supone que la memoria intermedia de la unidad 705 de multiplexación está vacía hasta que el bit 805 se introduce en la correspondiente de las unidades 704-1 a 704-K de codificación de bin.

Cuando el bit se introduce en la unidad de codificación de bin correspondiente mientras la memoria intermedia de la unidad 705 de multiplexación está vacía, la unidad de codificación de bin crea una nueva entrada en la memoria intermedia de la unidad 705 de multiplexación. En el ejemplo ilustrado, cuando el bit 850 cuyo índice de probabilidad es "0" y cuyo valor es "0" se introduce en la unidad 704-1 de codificación de bin, la primera entrada cuyo índice de entrada es "0" se crea en la memoria intermedia de la unidad 705 de multiplexación, tal como se representa en la tabla 860. Dado que el índice de probabilidad del bit 850 es "0", el "Índ." de esta entrada es "0". Además, dado que el valor del bit 850 es "0", la secuencia de bits de introducción en la primera entrada es "0". La palabra en esta entrada es "N/A" porque la palabra no se ha completado aún.

A continuación, el bit 851 cuyo índice de probabilidad es "1" y cuyo valor es "0" se introduce en la unidad 704-2 de codificación de bin. En este caso, dado que la entrada correspondiente al índice de probabilidad de este bit no existe en la memoria intermedia de la unidad 705 de multiplexación, se crea una nueva entrada, tal como se representa en la tabla 861. El "Índ." de esta segunda entrada es "1". La palabra en la segunda entrada es "N/A" porque la palabra no se ha completado aún.

Después de eso, cada vez que se introducen nuevos bits en las unidades de codificación de bin correspondientes, cada una de las unidades 704-1 a 704-K de codificación de bin y la unidad 705 de multiplexación convierten la secuencia de bins almacenada en la entrada correspondiente en la memoria intermedia de la unidad 705 de multiplexación en una palabra según las siguientes reglas 1) a 4), y emiten la palabra cuando pasa a estar lista para emitirse desde la entrada.

1) La unidad, 704-1 a 704-K, de codificación de bin busca en las entradas correspondientes al índice de probabilidad del bit de introducción en orden ascendente de los índices de entrada en la memoria intermedia de la unidad 705 de multiplexación.

2) Si hay alguna entrada en la que la palabra sea "N/A" entre las entradas correspondientes al índice de probabilidad del bit de introducción, la unidad, 704-1 a 704-K, de codificación de bin actualiza la secuencia de bits de introducción almacenada en esa entrada. Más específicamente, la unidad, 704-1 a 704-K, de codificación de bin añade el nuevo bit de introducción al final de la secuencia de bits de introducción almacenada en esa entrada. La unidad 705 de multiplexación realiza una comprobación para ver si la secuencia de bits actualizada coincide con una de las secuencias de bins en el mapa 800 de conversión. Si hay una secuencia de bins coincidente, la unidad, 704-1 a 704-K, de codificación de bin determina la palabra en esa entrada tomando la palabra correspondiente a esa secuencia de bins.

3) Si no hay ninguna entrada en la que la palabra sea "N/A" y que corresponda al índice de probabilidad del bit de introducción, la unidad, 704-1 a 704-K, de codificación de bin crea una nueva entrada para ese índice de probabilidad.

4) Cualquier entrada en la que la palabra no sea "N/A" puede emitirse como un flujo de bits cuando pasa a estar lista para su emisión, pero las entradas se emiten en orden descendente de los índices de entrada en la memoria intermedia de la unidad 705 de multiplexación. Esta es una limitación necesaria para garantizar que las palabras multiplexadas se decodifiquen correctamente en el aparato de decodificación de vídeo en movimiento (conversión

inversa desde las palabras hasta las secuencias de bins correspondientes). Si cualquier entrada cuyo índice de entrada es menor que el de la entrada de interés, y cuya palabra es "N/A", queda en la memoria intermedia de la unidad 705 de multiplexación, no se permite que la unidad 705 de multiplexación emita la entrada de interés aunque la palabra en esa entrada no sea "N/A".

5 Cuando se emite cualquier entrada dada, la unidad 705 de multiplexación añade la palabra almacenada en esa entrada dada al flujo de bits de datos de salida. La entrada así emitida se elimina de la memoria intermedia de multiplexación.

10 La tabla 862 ilustra el estado de la memoria intermedia de la unidad 705 de multiplexación cuando el bit 852 cuyo índice de probabilidad es "0" y cuyo valor es "0" se introduce en la unidad 704-1 de codificación de bin. En este momento, las entradas cuyos índices de entrada son de "1" a "4" están listas para su emisión. Sin embargo, dado que la palabra en la entrada cuyo índice de entrada es "0" es "N/A", no se permite aún que la unidad 705 de multiplexación emita todas las palabras de entrada como el flujo de bits. Esto provoca un retardo en la codificación de longitud variable.

15 La tabla 863 ilustra el estado de la memoria intermedia de la unidad 705 de multiplexación cuando el bit 853 cuyo índice de probabilidad es "0" y cuyo valor es "1" se introduce en la unidad 704-1 de codificación de bin. En este momento, la palabra en la entrada cuyo índice de entrada es "0" ya no es "N/A". Ahora se permite que la unidad 705 de multiplexación emita todas las entradas cuyos índices de entrada son de "0" a "4".

20 Cuando se añade una nueva entrada inmediatamente después de emitir todas las entradas cuyos índices de entrada son de "0" a "4", se asigna un índice "5" a la nueva entrada. Es decir, el índice de entrada de cualquier nueva entrada se determina añadiendo 1 al mayor índice entre los índices de todas las entradas incluyendo las entradas que se han emitido. La unidad 705 de multiplexación añade 1 al índice de entrada de la última entrada que se ha emitido en el flujo de bits, y almacena el valor resultante como el primer índice de entrada FirstIndex emitido del siguiente flujo de bits. FirstIndex se restablece en 0 al inicio del procedimiento de codificación de imagen.

25 Las figuras 13A a 13C son un diagrama de flujo que ilustra el procedimiento para identificar el bit final de cada grupo dado en el flujo de bits de datos de salida según la cuarta realización. La unidad 23 de codificación de longitud variable en el aparato de codificación de vídeo en movimiento deriva, según el siguiente diagrama de flujo, la posición BitEnd(i,j) del bit final del último bloque perteneciente al j-ésimo grupo G(i,j) en la i-ésima imagen en el flujo de bits de datos de salida. Cuando el bit correspondiente a la posición BitEnd(i,j) se introduce en la memoria intermedia de transmisión en el aparato 10 de codificación de vídeo en movimiento, el aparato 10 de codificación de vídeo en movimiento determina que todos los bits necesarios para decodificar el grupo G(i,j) se han introducido en la memoria intermedia de transmisión en el aparato 10 de codificación de vídeo en movimiento.

30 Las etapas en el diagrama de flujo ilustrado en las figuras 13A a 13C son las mismas que las etapas correspondientes en el diagrama de flujo ilustrado en las figuras 9A a 9C, excepto las etapas descritas a continuación en el presente documento. Por ejemplo, el procedimiento de la etapa S301 es el mismo que el procedimiento de la etapa S201. Por tanto, para los detalles de las etapas distintas de las etapas descritas a continuación en el presente documento, véase la descripción correspondiente en el diagrama de flujo ilustrado en las figuras 9A a 9C.

35 Una correspondiente de las unidades 704-1 a 704-K de codificación de bin en la unidad 23 de codificación de longitud variable codifica el b0-ésimo bin, y almacena el b0-ésimo bin (si la secuencia de bits que contiene el b0-ésimo bin puede convertirse, entonces la palabra correspondiente) en la memoria intermedia de la unidad 705 de multiplexación (etapa S304). Después de eso, la unidad 23 de codificación de longitud variable obtiene la posición de bit final q de la secuencia de bits que la unidad 705 de multiplexación emitió como un flujo de bits (etapa S305). Además, la unidad 23 de codificación de longitud variable obtiene el primer índice de entrada FirstIndex del siguiente flujo de bits desde la unidad 705 de multiplexación, y este índice se indica mediante t.

40 Después de eso, si el recuento de acarreo c es 1 o mayor en la etapa S306, la unidad 23 de codificación de longitud variable compara la posición de bit final p del flujo de bits antes de la codificación aritmética del b0-ésimo bin con la posición de bit final q del flujo de bits después de la codificación aritmética del b0-ésimo bin. Además, la unidad 23 de codificación de longitud variable determina si el primer índice de entrada t del siguiente flujo de bits es mayor que el índice s (etapa S307). El índice s representa el mayor índice de entrada entre los índices de las entradas correspondientes a EndOfSliceFlags que ya se han codificado aritméticamente pero no se han emitido aún en el flujo de bits.

45 Si la posición de bit final p es diferente de la posición de bit final q, y si el índice t es mayor que el índice s (Sí en la etapa S307), esto significa que se genera un nuevo bit codificando aritméticamente el b0-ésimo bin. En este caso, la última palabra correspondiente a la EndOfSliceFlag del último bloque en el grupo x ($x = j-1, \dots, j-c$) que se codificó aritméticamente con anterioridad pero no se emitió en el flujo de bits se emite en el flujo de bits. Es decir, todos los bits necesarios para decodificar el grupo x se han emitido en la posición q. Luego, la unidad 23 de codificación de longitud variable establece, como la posición de bit q, la posición de bit final BitEnd() de la secuencia de bits

necesaria para decodificar el grupo que se ha emitido en este caso (etapa S308).

Por otra parte, si la posición de bit final p es la misma que la posición de bit final q , o si el índice t es igual al índice s (No en la etapa S307), no se genera ningún nuevo bit. Por tanto, la unidad 23 de codificación de longitud variable pasa a la etapa S310 sin las etapas de procesamiento S308 y S309.

Tal como se ilustra en la figura 13B, la unidad 23 de codificación de longitud variable codifica aritméticamente EndOfSliceFlag (etapa S312), y después de eso obtiene la posición de bit final q del flujo de bits después de la codificación aritmética de EndOfSliceFlag (etapa S313). Además, la unidad 23 de codificación de longitud variable obtiene el primer índice de entrada FirstIndex del siguiente flujo de bits desde la unidad 705 de multiplexación, y este índice se indica mediante t . La unidad 23 de codificación de longitud variable obtiene además, desde la unidad 705 de multiplexación, el mayor índice de entrada entre los índices de las entradas correspondientes a EndOfSliceFlags que ya se han codificado aritméticamente pero no se han emitido aún en el flujo de bits; este índice se indica mediante s .

La unidad 23 de codificación de longitud variable determina si el recuento de acarreo c es 1 o mayor o no y si la posición de bit final p del flujo de bits antes de la codificación aritmética de EndOfSliceFlag es diferente de la posición de bit final q del flujo de bits después de la codificación aritmética de EndOfSliceFlag. Además, la unidad 23 de codificación de longitud variable determina si el índice t es mayor que el índice s (etapa S314). Si el recuento de acarreo c es 1 o mayor, y p es diferente de q , y si t es mayor que s (Sí en la etapa S314), esto significa que la EndOfSliceFlag del último bloque en el grupo x ($x = j-1, \dots, j-c$) se emite en el flujo de bits. Es decir, todos los bits necesarios para decodificar el grupo x se han emitido en el flujo de bits. Luego, la unidad 23 de codificación de longitud variable establece, como la posición de bit q , la posición de bit final BitEnd() de la secuencia de bits necesaria para decodificar el grupo que se ha emitido en este caso (etapa S315).

Por otra parte, si el recuento de acarreo c es 0, o p es la misma que q , o si t es igual a s (No en la etapa S314), esto significa que todos los bits necesarios para decodificar el grupo x no se emiten aún. Por tanto, la unidad 23 de codificación de longitud variable omite las etapas S315 y S316, y pasa a determinar si el bloque actual m es el último bloque en el grupo j (etapa S317). Si el bloque actual m no es el último bloque en el grupo j (No en la etapa S317), esto significa que quedan uno o más bloques por procesarse en el grupo j . Luego, la unidad 23 de codificación de longitud variable incrementa m en 1 y repite el procedimiento a partir de la etapa S302.

Por otra parte, si el bloque actual m es el último bloque en el grupo j (Sí en la etapa S317), la unidad 23 de codificación de longitud variable incrementa m en 1. Luego, la unidad 23 de codificación de longitud variable compara la posición de bit final p del flujo de bits antes de la codificación aritmética de EndOfSliceFlag con la posición de bit final q del flujo de bits después de la codificación aritmética de EndOfSliceFlag, y también compara el índice t con el índice s (etapa S318).

Si p es diferente de q , y si t es mayor que s (Sí en la etapa S318), esto significa que se genera un nuevo bit codificando aritméticamente la EndOfSliceFlag. Por tanto, esto significa que todos los bits necesarios para decodificar el último bloque en el grupo j se han emitido en el flujo de bits. Luego, la unidad 23 de codificación de longitud variable establece, como la posición de bit q , la posición de bit final BitEnd() de la secuencia de bits necesaria para decodificar el grupo j que se ha emitido en este caso (etapa S319).

Por otra parte, si p es la misma que q , o si t es igual a s (No en la etapa S318), no se genera ningún nuevo bit codificando aritméticamente la EndOfSliceFlag. Por consiguiente, la unidad 23 de codificación de longitud variable incrementa el recuento de acarreo c en 1 y también incrementa el índice de grupo j en 1 (etapa S321).

Tal como se ha descrito anteriormente, el aparato de codificación de vídeo en movimiento puede reducir el retardo de decodificación dividiendo la imagen en bloques y clasificando los bloques en grupos y ajustando la cantidad de código de los bloques en cada grupo según el tiempo de decodificación estimado para ese grupo.

Un programa informático que se ejecuta en un ordenador para implementar las funciones de las diversas unidades que constituyen el aparato de codificación de vídeo en movimiento según cada una de las realizaciones anteriores o sus ejemplos modificados puede distribuirse en la forma almacenada en una memoria de semiconductor o en la forma grabada en un medio de grabación tal como un medio de grabación óptica.

El aparato de codificación de vídeo en movimiento según cada una de las realizaciones anteriores o sus ejemplos modificados se usa en diversas aplicaciones. Por ejemplo, el aparato de codificación de vídeo en movimiento se incorpora en una cámara de vídeo, un aparato de transmisión de vídeo, un aparato de recepción de vídeo, un sistema de videoteléfono, un ordenador o un teléfono móvil.

Todos los ejemplos y el lenguaje condicional mencionados en el presente documento están destinados a propósitos pedagógicos para ayudar al lector a entender la invención y los conceptos que aporta el inventor para promover la técnica, y debe interpretarse que no existe limitación con respecto a tales ejemplos y condiciones mencionados específicamente, y la organización de tales ejemplos en la memoria respectiva tampoco está relacionada con una

demostración de superioridad e inferioridad de la invención. Aunque las realizaciones de la presente invención se han descrito con detalle, debe entenderse que los diversos cambios, sustituciones y alteraciones pueden realizarse a las mismas sin apartarse del alcance de la invención.

5 **[Descripción de los números de referencia]**

- 10... APARATO DE CODIFICACIÓN DE VÍDEO EN MOVIMIENTO
- 11... UNIDAD DE PROCESAMIENTO DE CODIFICACIÓN
- 10 12... UNIDAD DE CONTROL DE CANTIDAD DE CÓDIGO
- 13... UNIDAD DE DETERMINACIÓN DE GRUPO
- 15 14... UNIDAD DE CÁLCULO DE INFORMACIÓN DE TIEMPO DE DECODIFICACIÓN DE GRUPO
- 15... UNIDAD DE ANEXADO DE INFORMACIÓN DE GRUPO
- 21... UNIDAD DE TRANSFORMADA ORTOGONAL
- 20 22... UNIDAD DE CUANTIFICACIÓN
- 23... UNIDAD DE CODIFICACIÓN DE LONGITUD VARIABLE
- 25 24... UNIDAD DE CÁLCULO DE VALOR DE CUANTIFICACIÓN
- 25... CONTADOR DE BITS
- 26... UNIDAD DE CÁLCULO DE OCUPACIÓN DE MEMORIA INTERMEDIA
- 30 700... UNIDAD DE CODIFICACIÓN DE ENTROPÍA
- 701... UNIDAD DE BINARIZACIÓN
- 35 702... UNIDAD DE MODELADO DE CONTEXTO
- 703... UNIDAD DE CUANTIFICACIÓN DE PROBABILIDAD
- 704-1 A 704-K... UNIDAD DE CODIFICACIÓN DE BIN
- 40 705... UNIDAD DE MULTIPLEXACIÓN

REIVINDICACIONES

1. Aparato de codificación de vídeo en movimiento para codificar cada imagen contenida en datos de vídeo en movimiento dividiendo la imagen en una pluralidad de bloques, que comprende:
- 5 una unidad (13) de determinación de grupo que determina, para cada imagen, a cuál de una pluralidad de grupos dentro de la imagen pertenece cada bloque, y determina el tamaño de cada uno de la pluralidad de grupos;
- 10 una unidad (14) de cálculo de información de tiempo de decodificación de grupo que calcula un tiempo de decodificación para cada uno de los grupos;
- 15 una unidad (15) de anexo de información de grupo que anexa información de grupo que identifica el grupo al que pertenece cada bloque e información de tiempo de decodificación que representa el tiempo de decodificación para cada uno de los grupos a datos que van a emitirse;
- 20 una unidad (12) de control de cantidad de código que controla una cantidad de código para cada bloque contenido en el grupo de modo que datos necesarios para decodificar todos los bloques contenidos en el grupo llegan a una memoria intermedia de recepción de flujo proporcionada en un aparato de decodificación de vídeo en movimiento antes de que transcurra el tiempo de decodificación del grupo calculado por la unidad de cálculo de información de tiempo de decodificación de grupo cuando los datos se transmiten al aparato de decodificación de vídeo en movimiento a una tasa de transmisión prescrita; y
- 25 una unidad (11) de procesamiento de codificación que codifica cada bloque, basándose en información de control referente a la cantidad de código.
2. Aparato de codificación de vídeo en movimiento según la reivindicación 1, en el que la unidad (14) de cálculo de información de tiempo de decodificación de grupo calcula el tiempo de decodificación para cada uno de los grupos de modo que un valor de diferencia calculado entre el tiempo de decodificación del primer grupo en una imagen que está codificándose y el tiempo de decodificación del último grupo en una imagen inmediatamente anterior a la imagen que está codificándose pasa a ser mayor que un valor de diferencia calculado entre dos grupos sucesivos cualesquiera entre el segundo grupo y grupos posteriores en la imagen que está codificándose.
- 30 3. Método de codificación de vídeo en movimiento para codificar cada imagen contenida en datos de vídeo en movimiento dividiendo la imagen en una pluralidad de bloques, que comprende:
- 35 determinar, para cada imagen, a cuál de una pluralidad de grupos dentro de la imagen pertenece cada bloque, y determinar el tamaño de cada uno de la pluralidad de grupos;
- 40 calcular un tiempo de decodificación para cada uno de los grupos;
- 45 anexar información de grupo que identifica el grupo al que pertenece cada bloque e información de tiempo de decodificación que representa el tiempo de decodificación para cada uno de los grupos a datos que van a emitirse;
- 50 controlar una cantidad de código para cada bloque contenido en el grupo de modo que datos necesarios para decodificar todos los bloques contenidos en el grupo llegan a una memoria intermedia de recepción de flujo proporcionada en un aparato de decodificación de vídeo en movimiento antes de que transcurra el tiempo de decodificación del grupo cuando los datos se transmiten al aparato de decodificación de vídeo en movimiento a una tasa de transmisión prescrita; y
- 55 codificar cada bloque, basándose en información de control referente a la cantidad de código.
4. Método de codificación de vídeo en movimiento según la reivindicación 3, en el que el tiempo de decodificación para cada uno de los grupos se calcula de modo que un valor de diferencia calculado entre el tiempo de decodificación del primer grupo en una imagen que está codificándose y el tiempo de decodificación del último grupo en una imagen inmediatamente anterior a la imagen que está codificándose pasa a ser mayor que un valor de diferencia calculado entre dos grupos sucesivos cualesquiera entre el segundo grupo y grupos posteriores en la imagen que está codificándose.
- 60 5. Programa informático de codificación de vídeo en movimiento para hacer que un ordenador ejecute un procedimiento para codificar cada imagen contenida en datos de vídeo en movimiento dividiendo la imagen en una pluralidad de bloques, comprendiendo el procedimiento:
- 65 determinar, para cada imagen, a cuál de una pluralidad de grupos dentro de la imagen pertenece cada

bloque, y determinar el tamaño de cada uno de la pluralidad de grupos;

calcular un tiempo de decodificación para cada uno de los grupos;

5 anexar información de grupo que identifica el grupo al que pertenece cada bloque e información de tiempo de decodificación que representa el tiempo de decodificación para cada uno de los grupos a datos que van a emitirse;

10 controlar una cantidad de código para cada bloque contenido en el grupo de modo que datos necesarios para decodificar todos los bloques contenidos en el grupo lleguen a una memoria intermedia de recepción de flujo proporcionada en un aparato de decodificación de vídeo en movimiento antes de que transcurra el tiempo de decodificación del grupo cuando los datos se transmiten al aparato de decodificación de vídeo en movimiento a una tasa de transmisión prescrita; y

15 codificar cada bloque, basándose en información de control referente a la cantidad de código.

6. Programa informático de codificación de vídeo en movimiento según la reivindicación 5, en el que el tiempo de decodificación para cada uno de los grupos se calcula de modo que un valor de diferencia calculado entre el tiempo de decodificación del primer grupo en una imagen que está codificándose y el tiempo de decodificación del último grupo en una imagen inmediatamente anterior a la imagen que está codificándose pasa a ser mayor que un valor de diferencia calculado entre dos grupos sucesivos cualesquiera entre el segundo grupo y grupos posteriores en la imagen que está codificándose.

20

FIG. 1

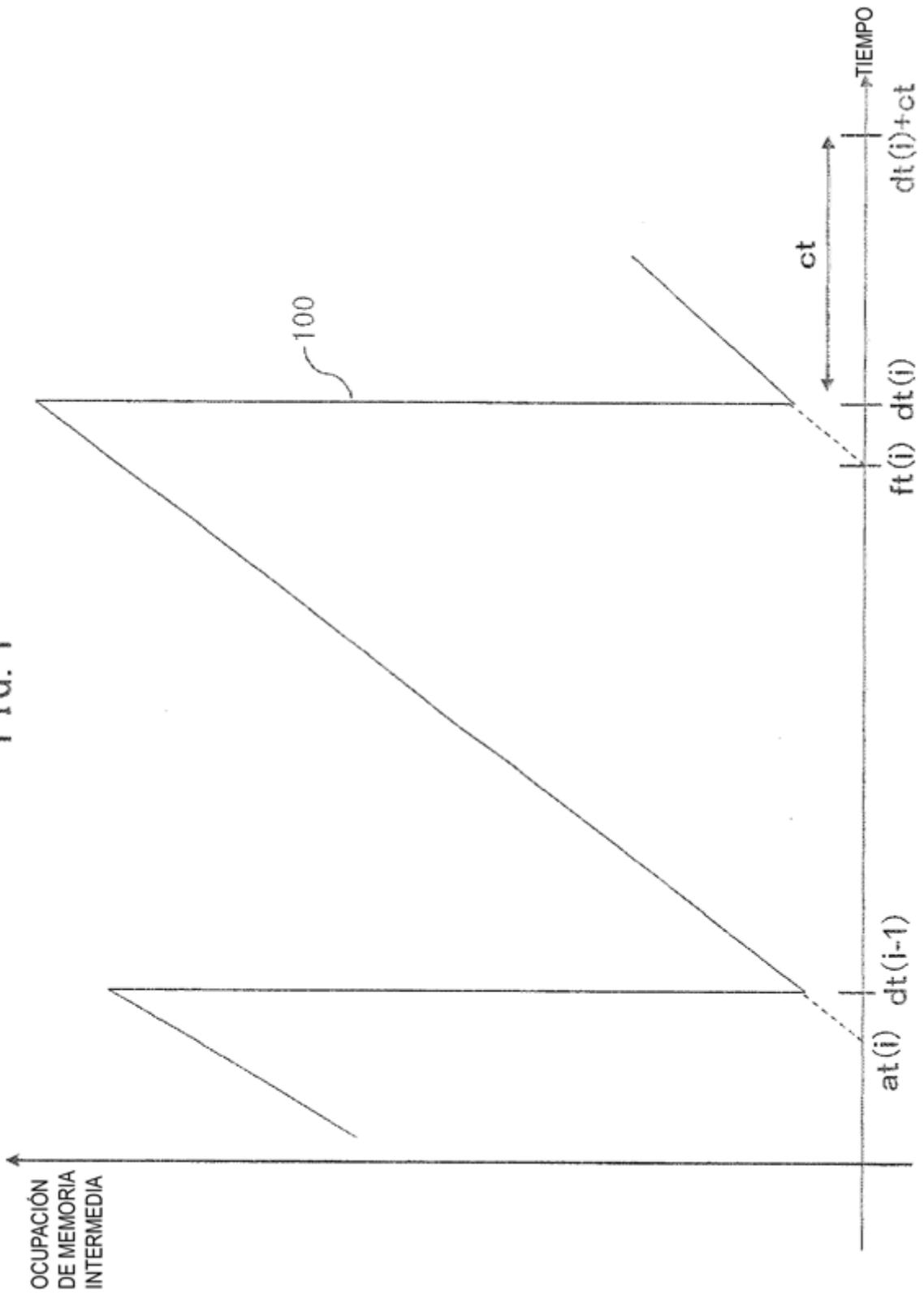


FIG. 2

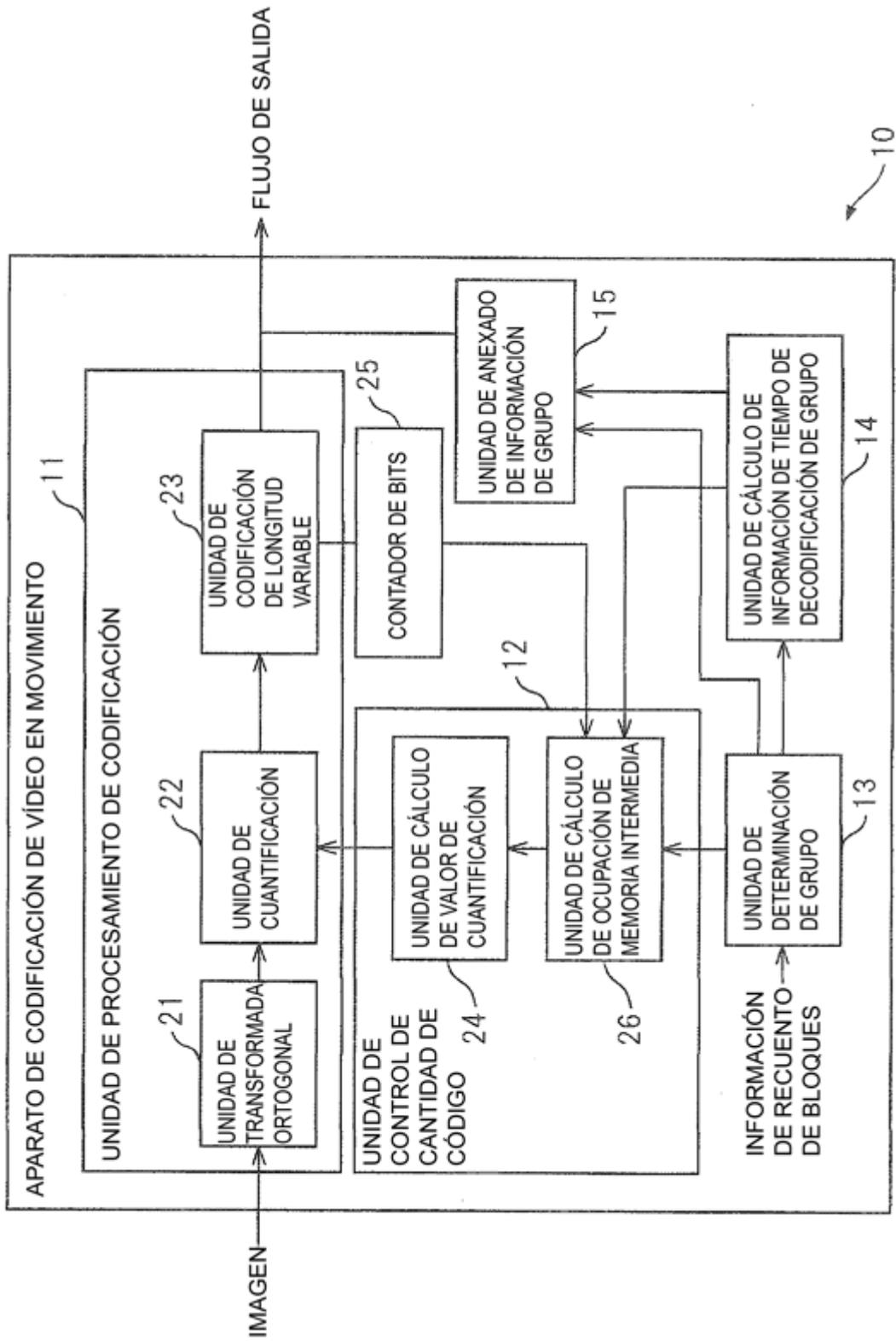


FIG. 3

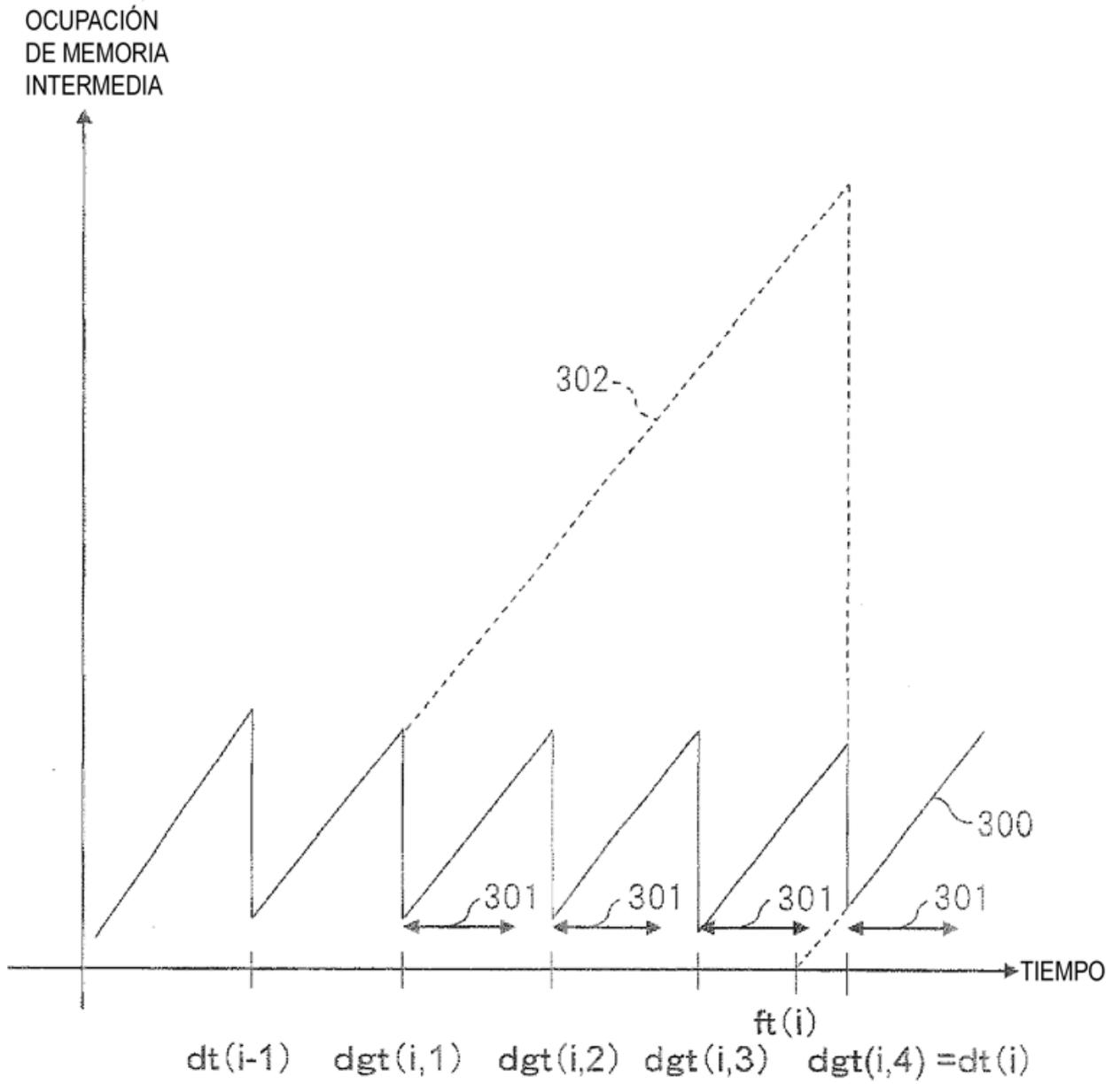


FIG. 4

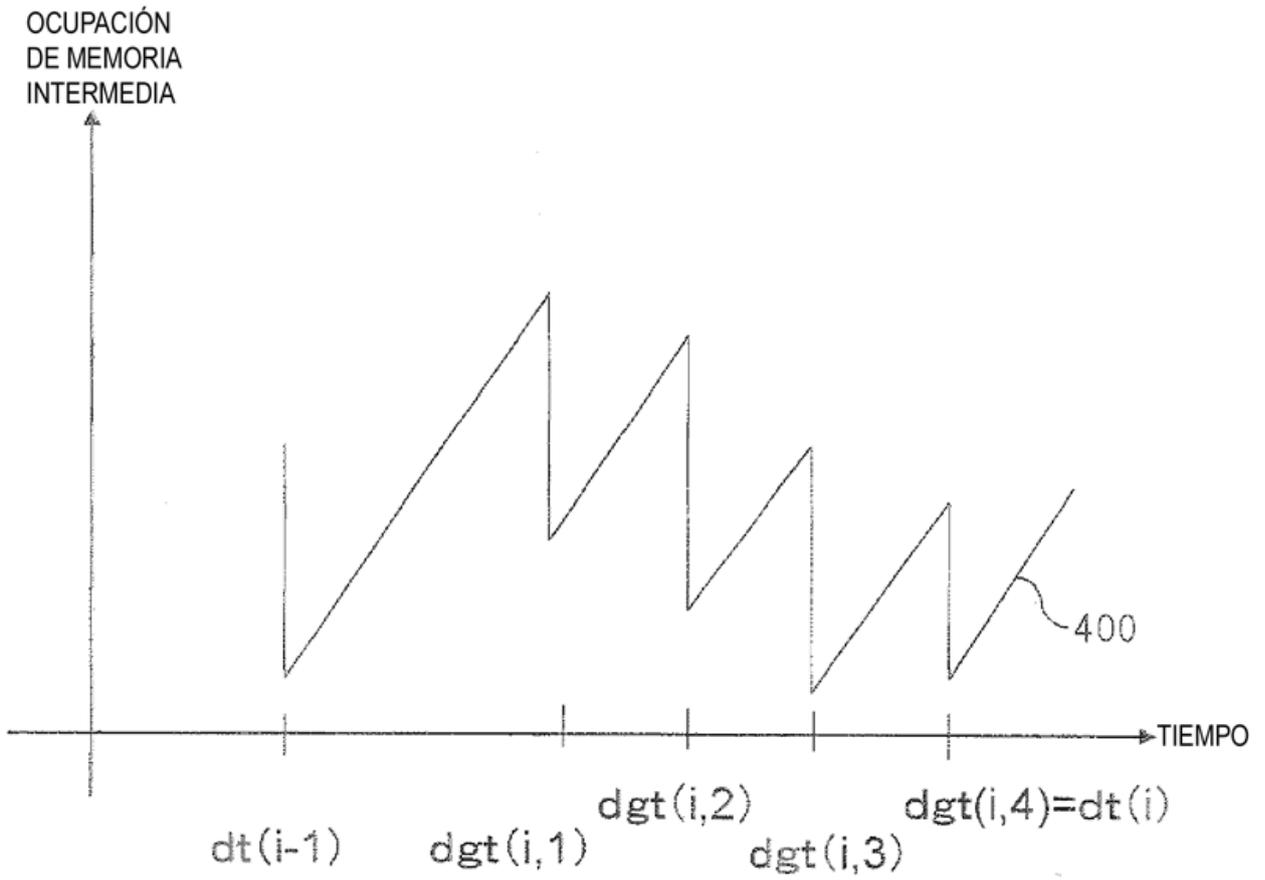


FIG. 5

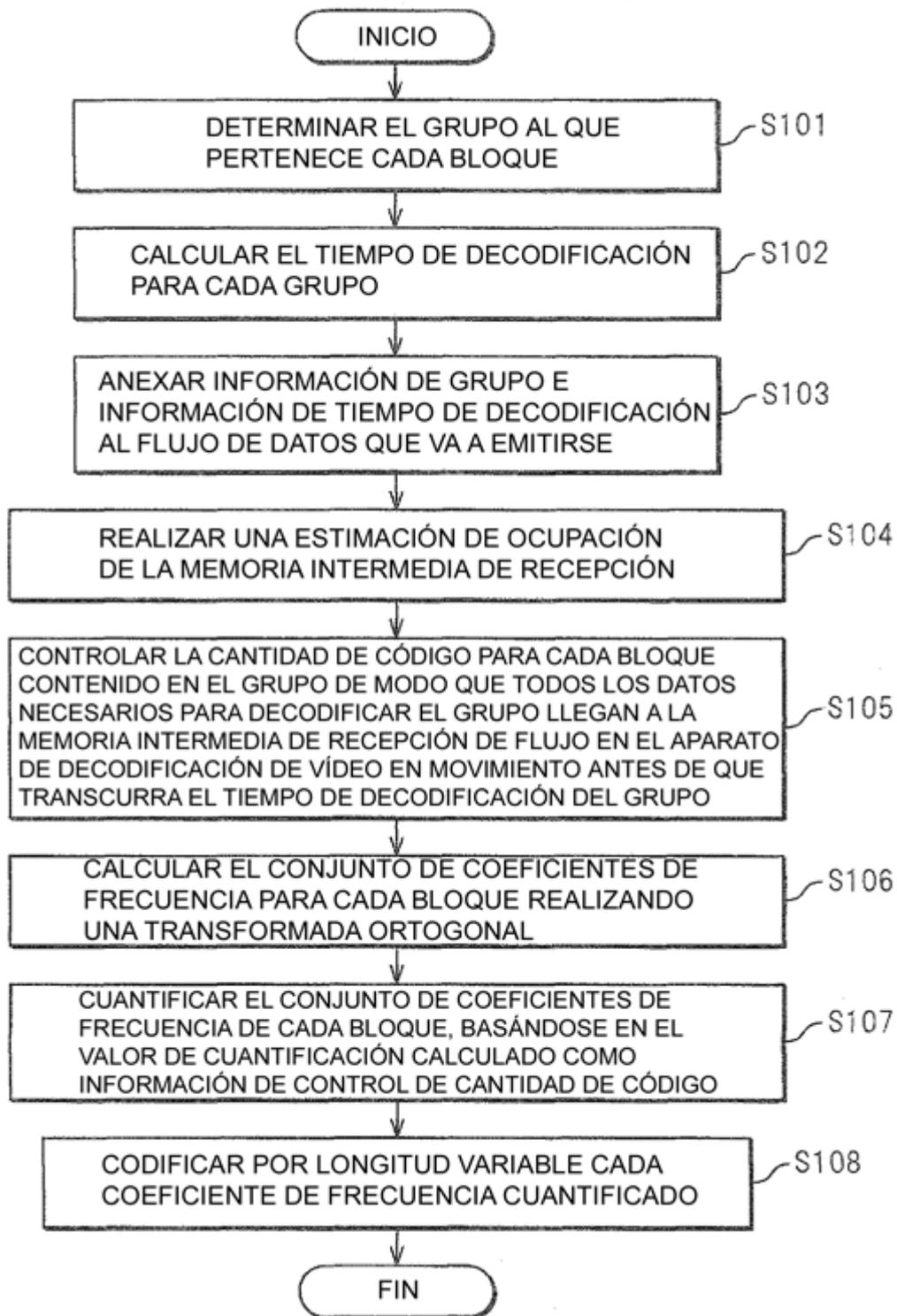


FIG. 6

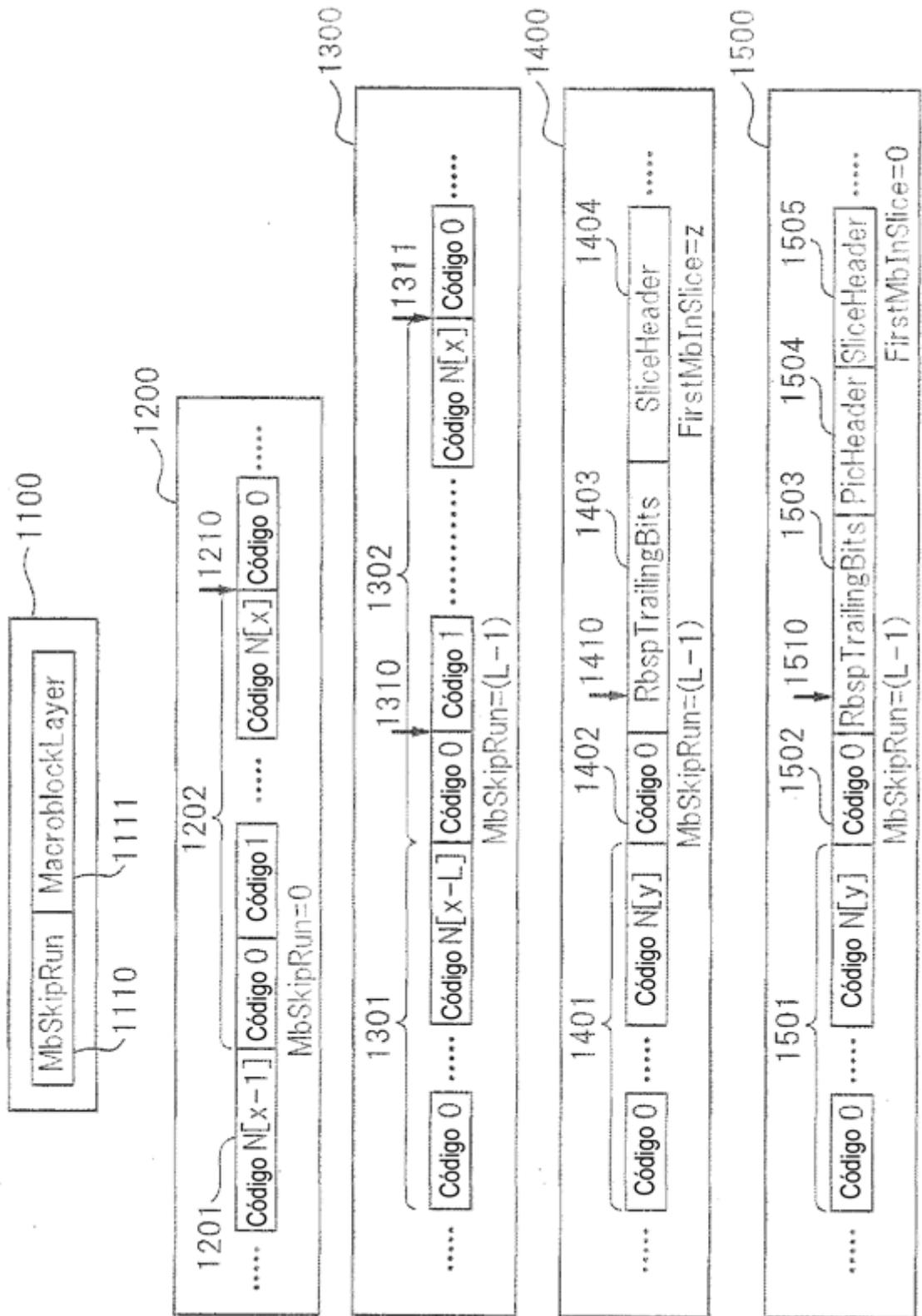


FIG. 7

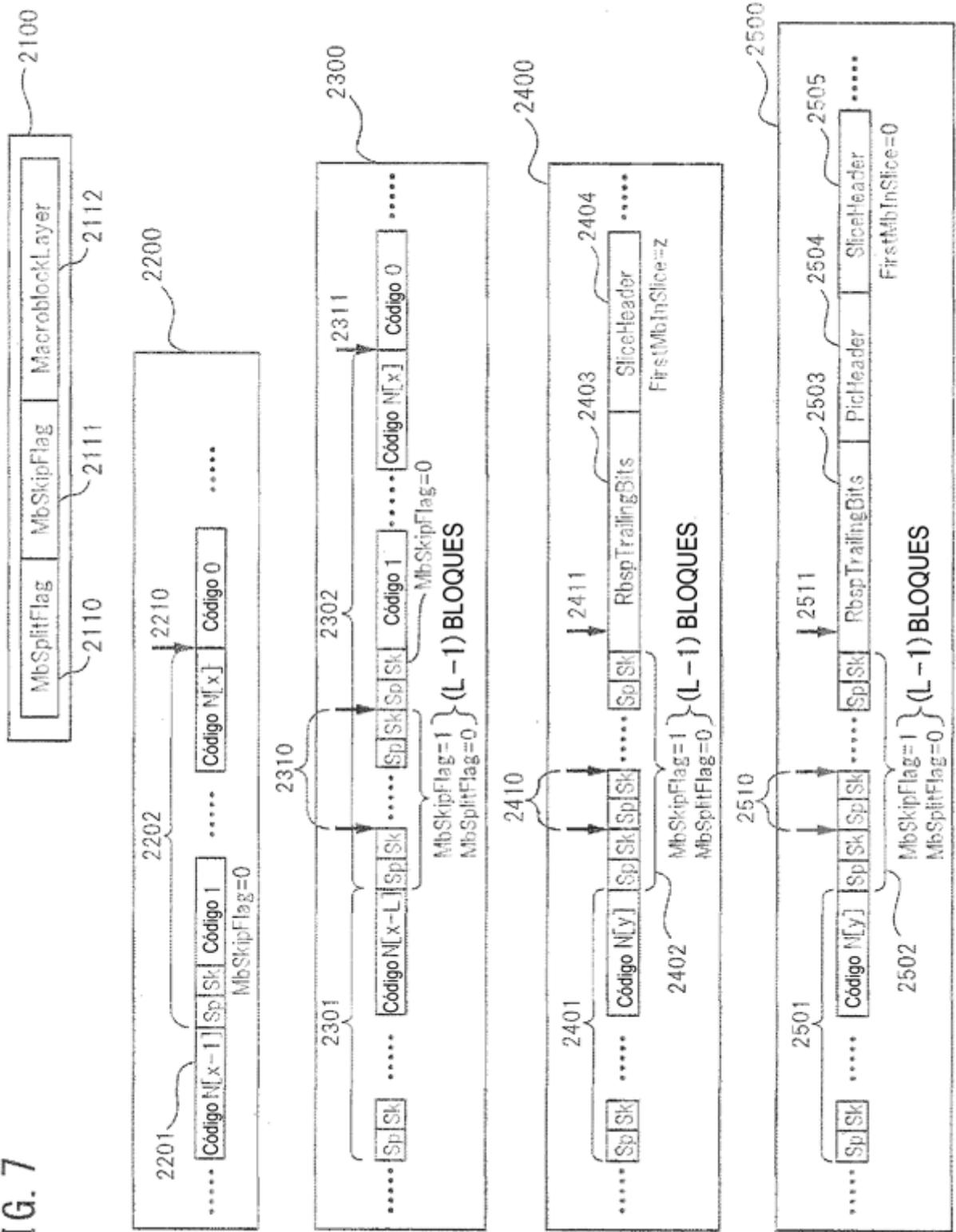


FIG. 8A

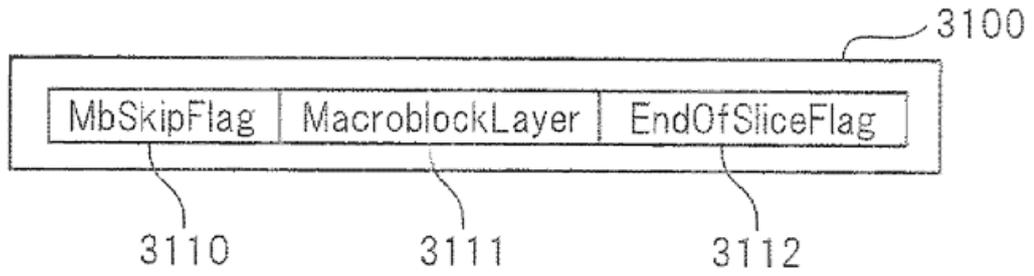


FIG. 8B

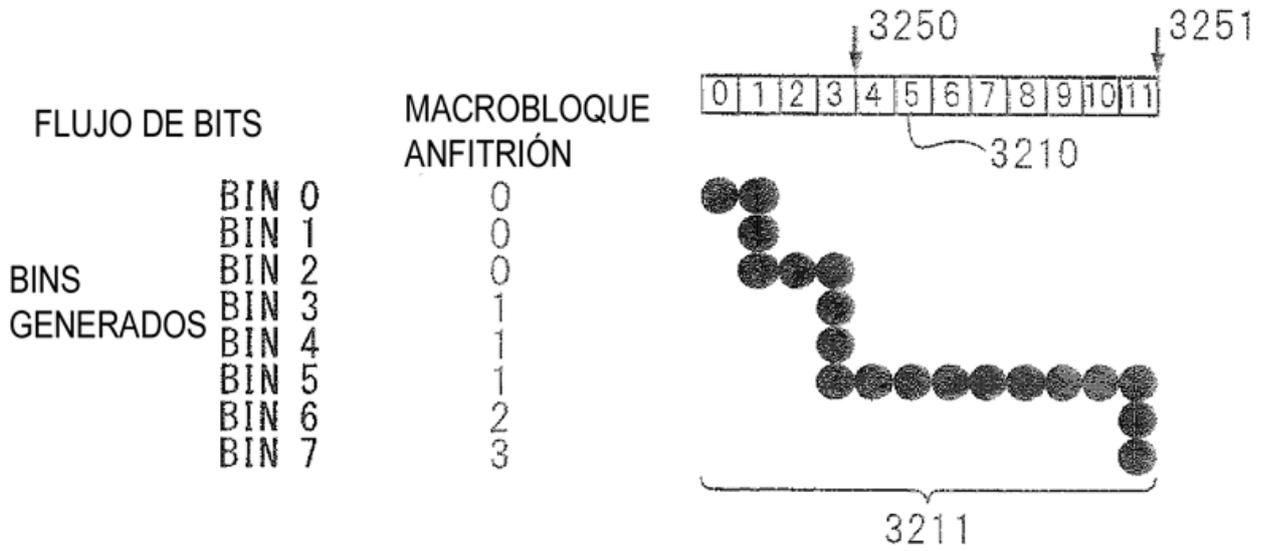


FIG. 9A

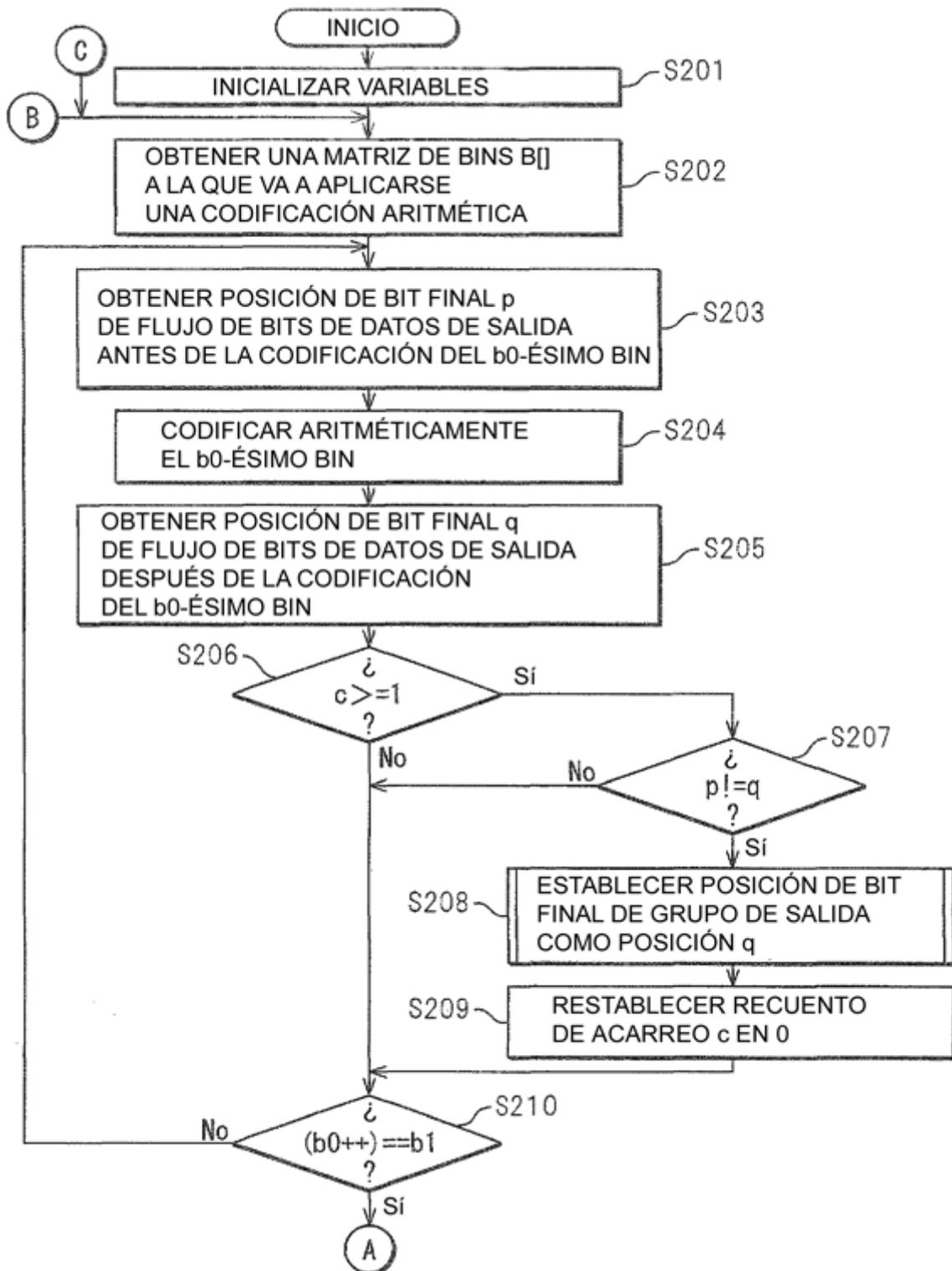


FIG. 9B

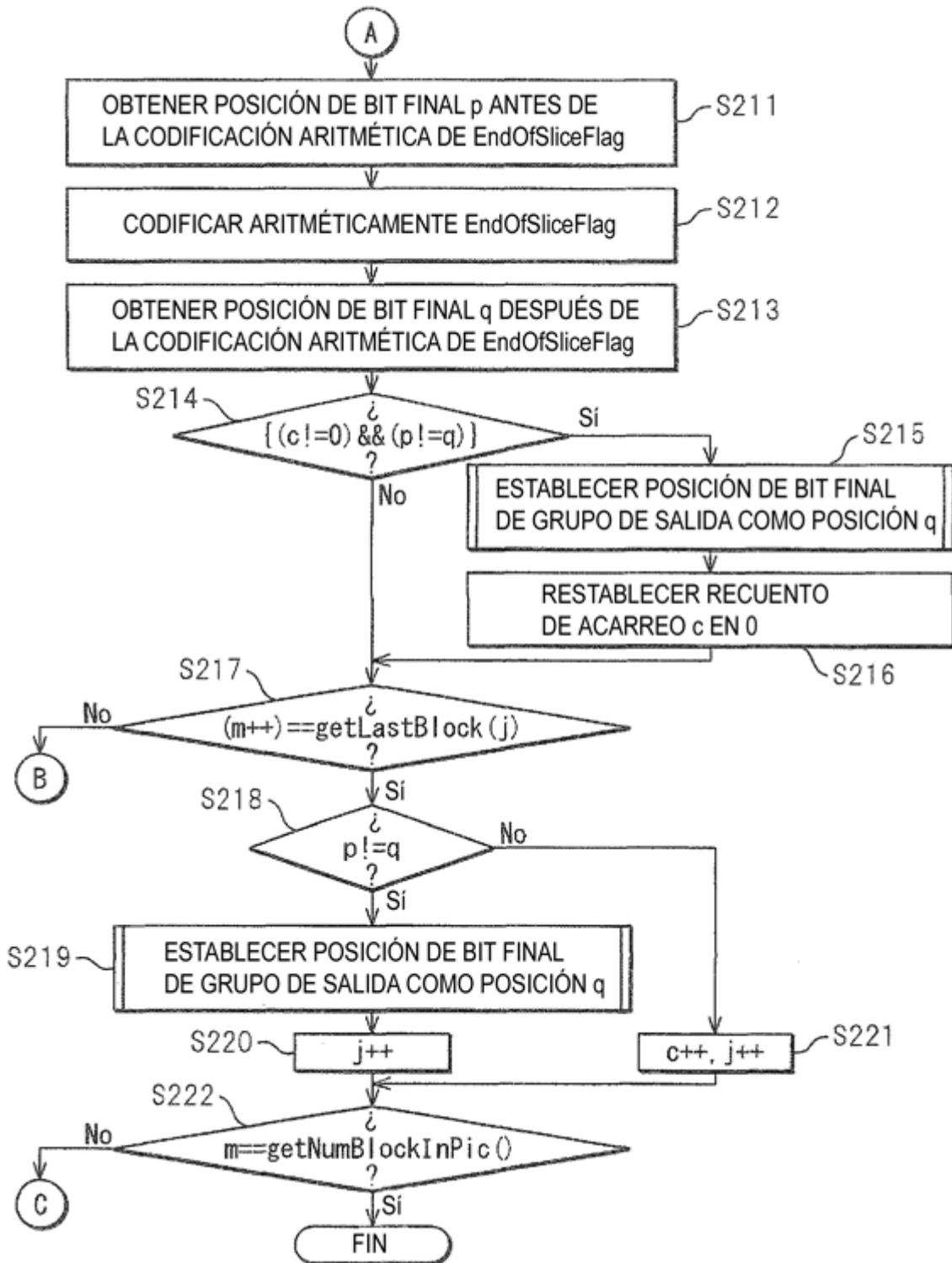


FIG. 9C

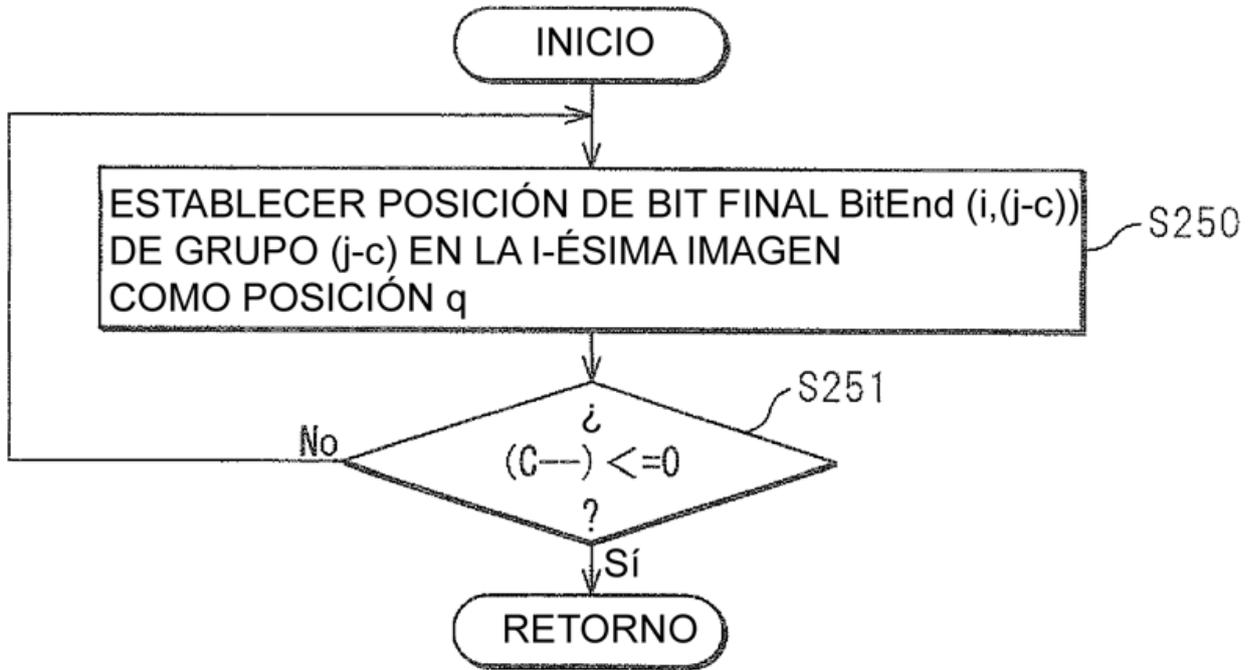


FIG. 10

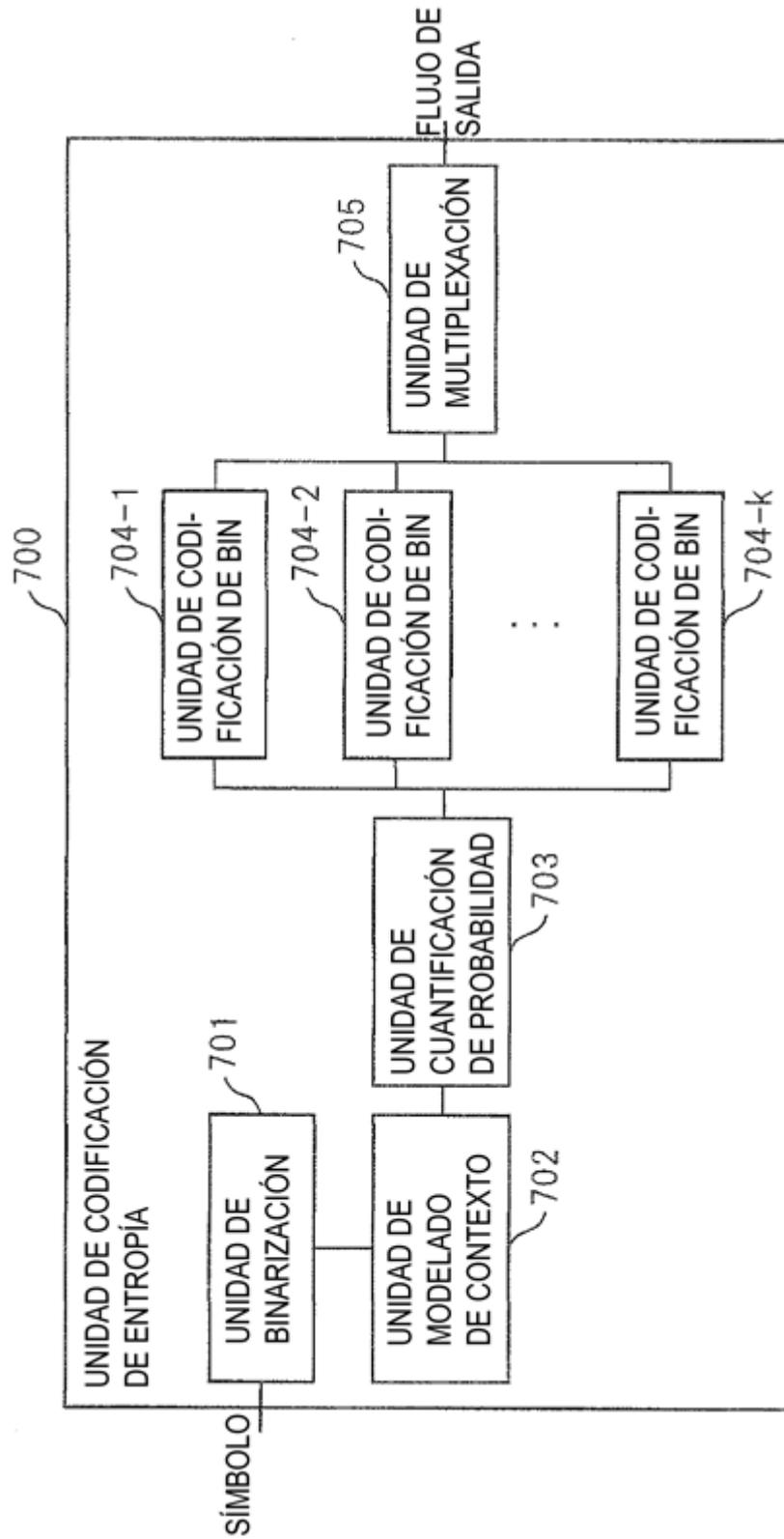


FIG. 11

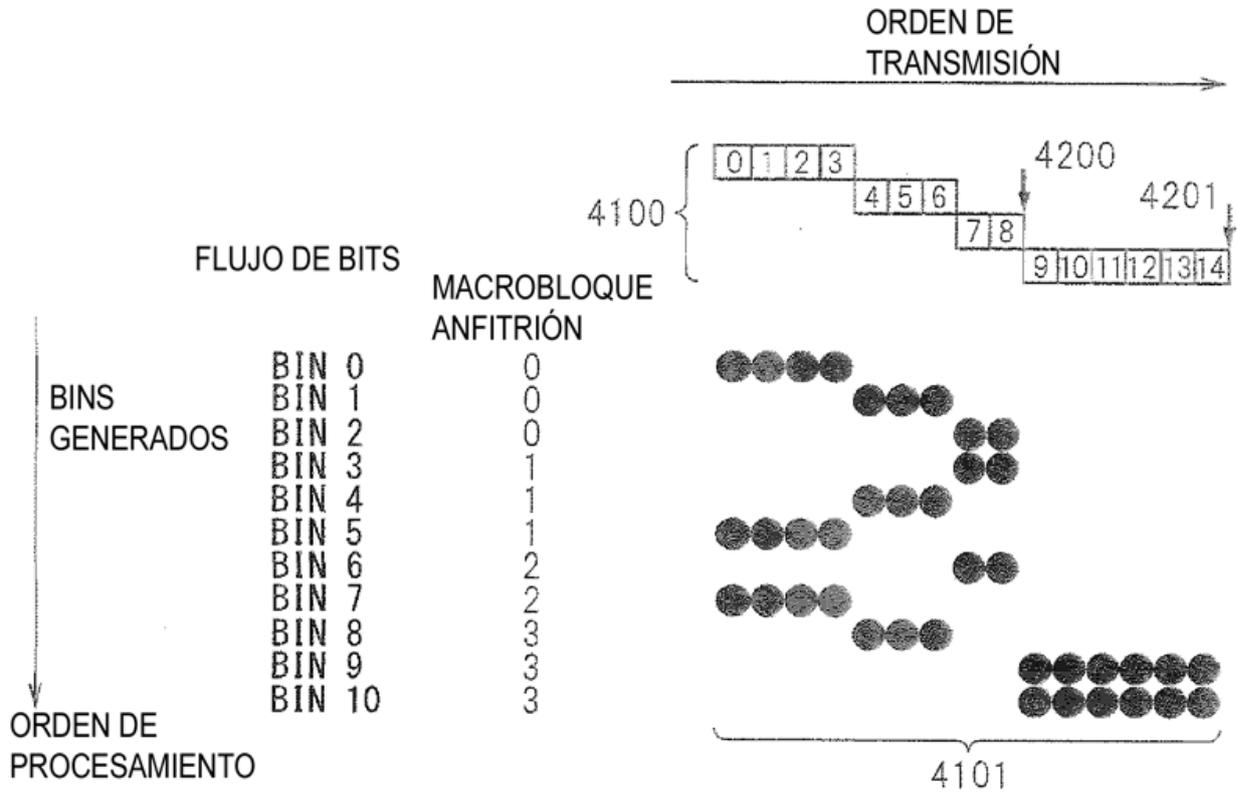


FIG. 12

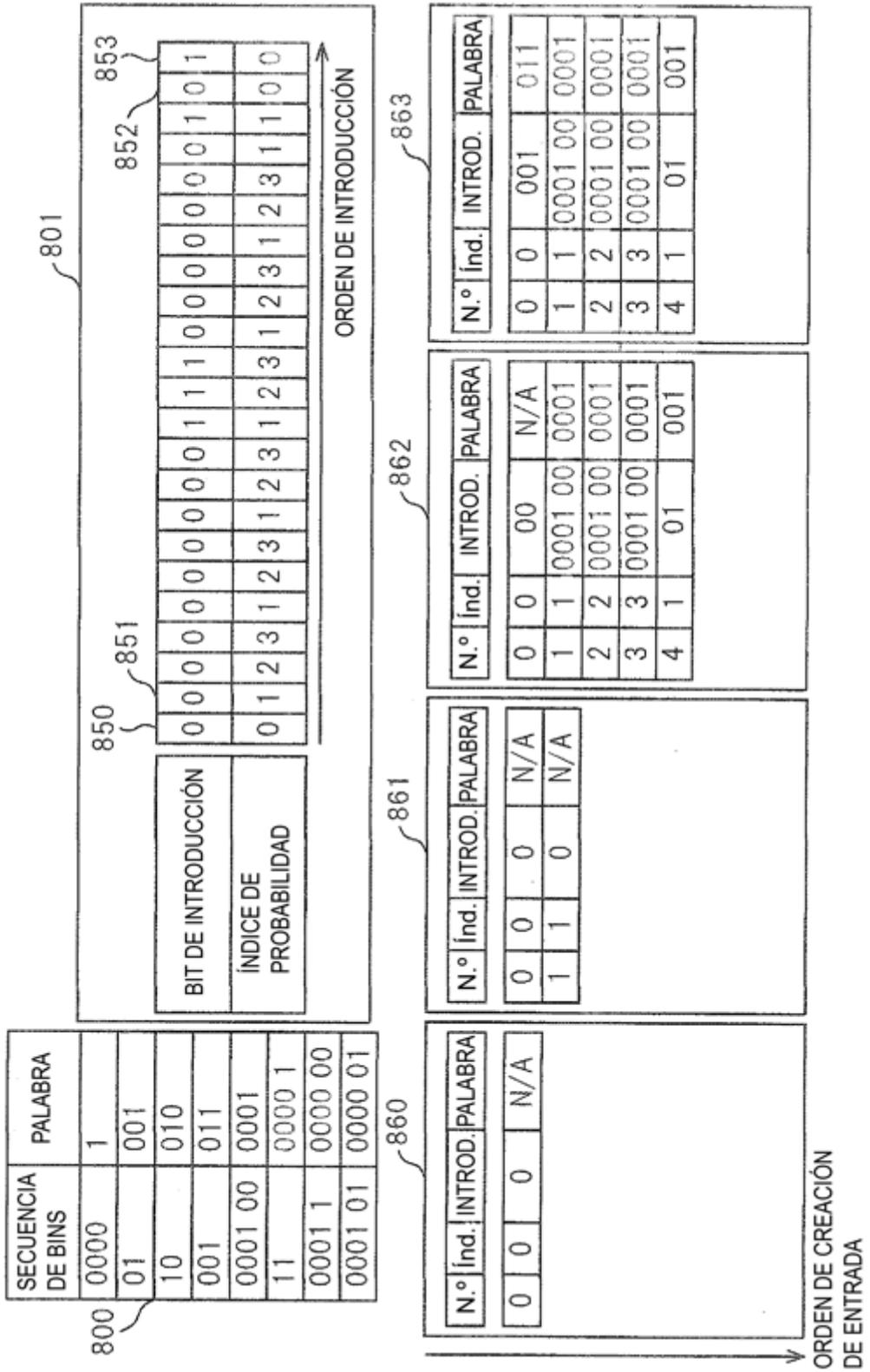


FIG. 13A

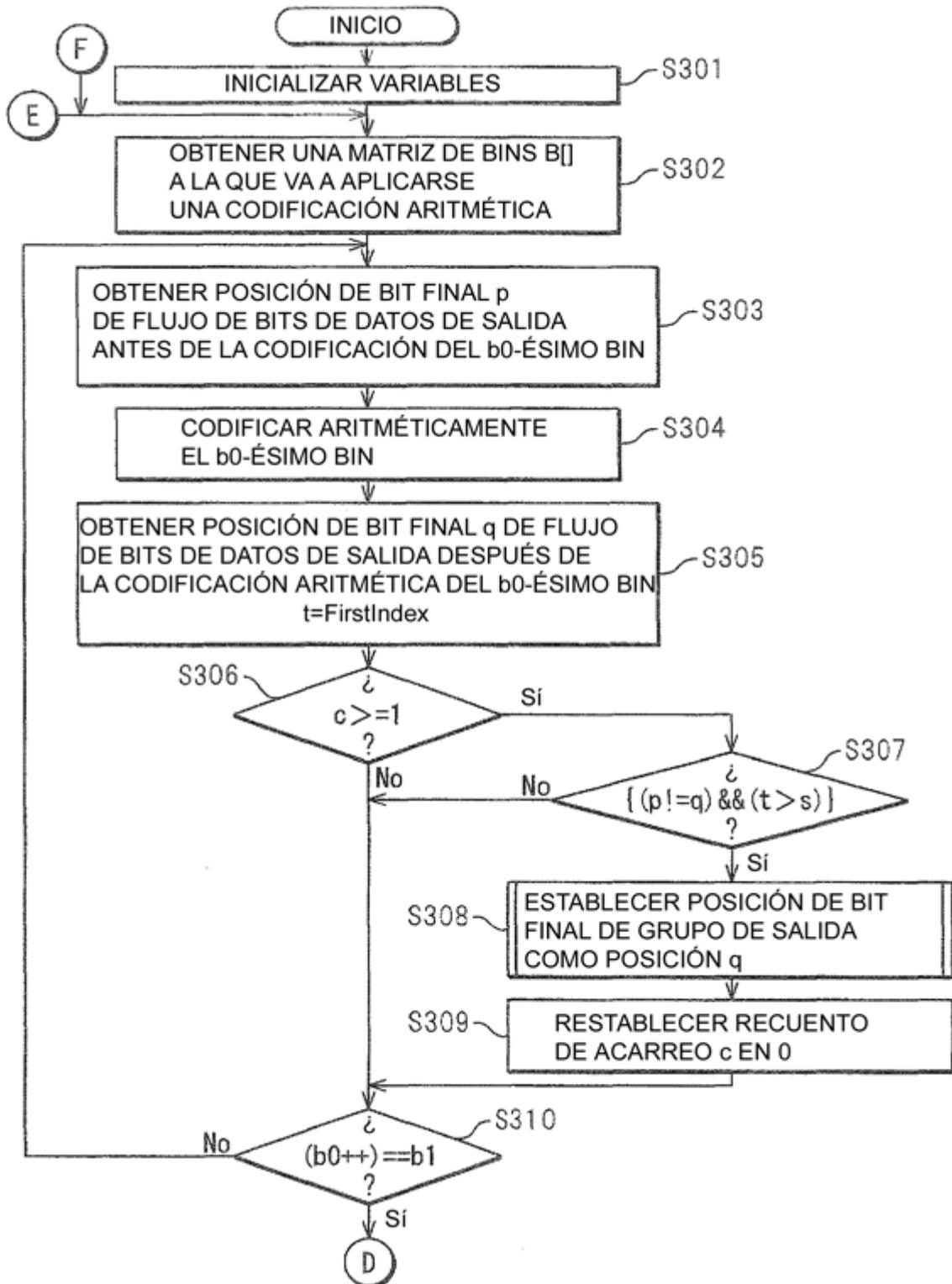


FIG. 13B

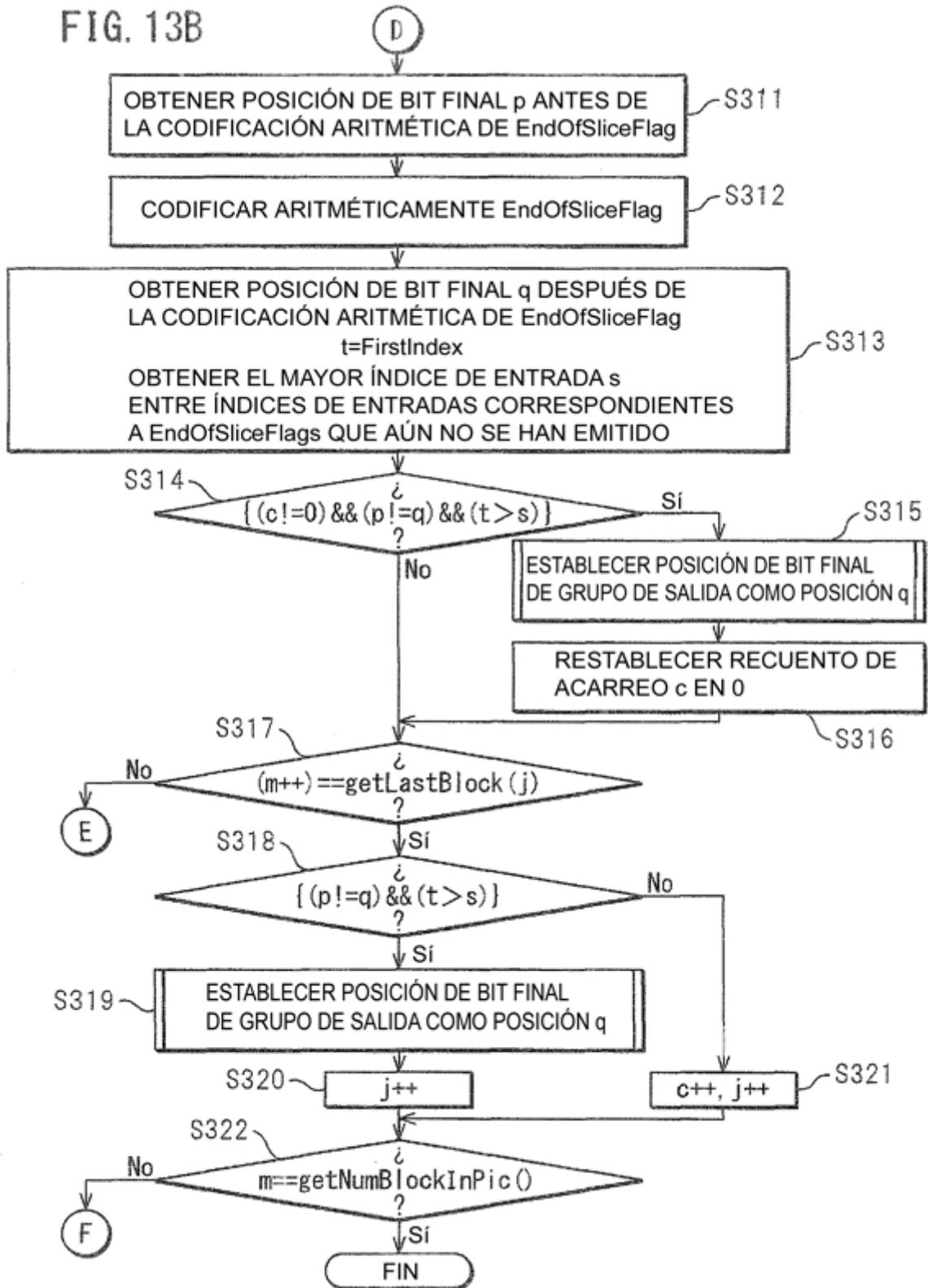


FIG. 13C

