

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 780 006**

51 Int. Cl.:

**H04N 19/174** (2014.01)  
**H04N 19/70** (2014.01)  
**H04N 19/52** (2014.01)  
**H04N 19/597** (2014.01)  
**H04N 19/436** (2014.01)  
**H04N 19/30** (2014.01)  
**H04N 19/423** (2014.01)  
**H04N 19/46** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **19.09.2013 E 17201617 (2)**

97 Fecha y número de publicación de la concesión europea: **08.01.2020 EP 3301923**

54 Título: **Método de decodificación de imagen y aparato de decodificación de imagen**

30 Prioridad:

**26.09.2012 US 201261705846 P**  
**10.10.2012 US 201261711892 P**

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

**21.08.2020**

73 Titular/es:

**VELOS MEDIA INTERNATIONAL LIMITED**  
**(100.0%)**  
**Unit 32, the Hyde Building The Park,**  
**Carrickmines**  
**Dublin 18, IE**

72 Inventor/es:

**ESENLIK, SEMIH;**  
**NARROSCHKE, MATTHIAS y**  
**WEDI, THOMAS**

74 Agente/Representante:

**MILTENYI , Peter**

**ES 2 780 006 T3**

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

**DESCRIPCIÓN**

Método de decodificación de imagen y aparato de decodificación de imagen

**Campo técnico**

5 La presente invención se refiere a un método de codificación de imagen de codificación de una imagen y un método de decodificación de imagen para decodificar una imagen.

**Técnica anterior**

10 La mayoría de los algoritmos de codificación de video estandarizados actuales están basados en codificación de video híbrida. En los métodos de codificación de video híbridos, se usan varios esquemas de compresión con y sin pérdidas para conseguir la ganancia de compresión deseada. La codificación de video híbrida es también la base de los estándares ITU-T (estándares H.26x tales como H.261, H.263) así como los estándares ISO/IEC (estándares MPEG-X tales como MPEG-1, MPEG-2 y MPEG-4).

15 El estándar de codificación de video más reciente y avanzado es el estándar actualmente denominado como codificación de video avanzada (Advanced Video Coding, AVC) H.264/MPEG-4. Este es el resultado de los esfuerzos de normalización del equipo de video conjunto (Joint Video Team, JVT), un equipo conjunto de grupos de ITU-T y MPEG ISO/IEC.

Además, un estándar de codificación de video llamado codificación de video de alta eficiencia (High-Efficiency Video Coding, HEVC) está siendo considerado por el equipo colaborativo conjunto en codificación de video (JCT-VC), con el objetivo particular de mejorar la eficiencia con respecto a la codificación de video de alta resolución.

**Lista de referencias**

20 Bibliografía no de patente

Bibliografía no de patente 1: C. Gordon, *et al.*, "Wavefront Parallel Processing for HEVC Encoding y Decoding", JCTVC-F274-v2, de la reunión en Torino, julio del 2011, Internet <URL: <http://phenix.int-evry.fr>>

Bibliografía no de patente 2: A. Fuldseth, *et al.*, "Tiles", JCTVC-F355-v1, de la reunión en Torino, julio del 2011, Internet <URL: <http://phenix.int-evry.fr>>

25 Bibliografía no de patente 3: JCTVC-J1003\_d7, "High efficiency video coding (HEVC) text specification draft 8", julio del 2012, página 73, "dependent\_slice\_flag", Internet <URL: <http://phenix.IT-sudparis.eu/jct/>>

30 LEE T *ET AL*: "On dependent slices", 10. REUNIÓN JCT-VC; 101. REUNIÓN MPEG; 11-7-2012 - 20-7-2012; ESTOCOLMO; (JOINT COLLABORATIVE TEAM ON VIDEO CODING DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16); URL: <HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/>, n.º JCTVC-J0217, 2 de julio de 2012 (02-07-2012) se refiere a dependent\_slice\_flag, dependent\_slice\_enabled\_flag y slice\_adress.

35 SCHIERL T *ET AL*: "Dependent Slices", 9. REUNIÓN JCT-VC; 100. REUNIÓN MPEG; 27-4-2012 - 7-5-2012; GÉNOVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16); URL: <HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/>, n.º JCTVC-I0229, 16 de abril de 2012 (16-04-2012) se refiere a una "dirección de porción" y a una "bandera de porción dependiente" en la estructura de datos de "slice\_header{}" y a "dependent\_slices\_present\_flag" en la estructura de datos pic\_parameter\_set\_rbsp{}.

40 BROSS B *ET AL*: "High Efficiency Video Coding (HEVC) text specification draft 8", 10. REUNIÓN JCT-VC; 101. REUNIÓN MPEG; 11-7-2012 - 20-7-2012; ESTOCOLMO; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/ WG11 AND ITU-T SG.16); URL: <HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/>, n.º JCTVC-J1003, 23 de julio de 2012 (23-07-2012) se refiere a una semántica de cabecera de porción y a definiciones de: slice\_address, dependent\_slice\_flag y dependent\_slice\_enabled\_flag.

COBAN M *ET AL*: "AHG4: On tiles and wavefront parallel processing", 10. REUNIÓN JCT-VC; 101. REUNIÓN MPEG; 11-7-2012 - 20-7-2012; ESTOCOLMO; (JOINT COLLABORATIVE TEAM ON VIDEO CODING DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16), ITU, ISO, IEC, CH, vol. JCTVC-J0123, n.º JCTVC-J0123 3 de julio de 2012 (03-07-2012), páginas 1-6 se refiere a procesamiento paralelo de frente de onda y tejas.

45 R. SJOBERG *ET AL*: "Overview of HEVC high-level syntax and reference picture management", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, 1 de enero de 2012 (01-01-2012), páginas 1-1, ISSN: 1051-8215, DOI: 10.1109/TCSVT.2012.2223052 da una visión general de la sintaxis de alto nivel de HEVC.

50 GORDON C *ET AL*: "Wavefront Parallel Processing for HEVC Encoding and Decoding", 6. REUNIÓN JCT-VC; 97. REUNIÓN MPEG; 14-7-2011 - 22-7-2011; TORINO; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/ SC29/WG11 E ITU-T SG.16); URL: <HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/>, n.º JCTVC-

F274, 1 de enero de 2011 (01-01-2011) se refiere a procesamiento paralelo de frente de onda.

### Sumario de la invención

#### Problema técnico

5 Sin embargo, existe un problema de que un método de codificación de imagen, un método de decodificación de imagen, y similares no tienen suficiente eficiencia de procesamiento.

Por lo tanto, la presente invención proporciona un método de codificación de imagen y un método de decodificación de imagen que son capaces de aumentar la eficiencia de procesamiento.

#### Solución al problema

10 Un método de decodificación de imagen según un aspecto de la presente invención es un método de decodificación de imagen, según la reivindicación independiente 1.

Un aparato de decodificación de imagen según un aspecto de la presente invención es un aparato de decodificación de imagen, según la reivindicación dependiente 7.

15 Los aspectos generales y específicos descritos anteriormente se pueden implementar usando un sistema, un método, un circuito integrado, un programa informático, o un medio de grabación legible por un ordenador tal como un CD-ROM, o cualquier combinación de sistemas, métodos, circuitos integrados, programas informáticos, o medios de grabación legibles por un ordenador.

#### Efectos ventajosos de la invención

Un método de decodificación de imagen y un aparato de decodificación de imagen según la presente invención son capaces de aumentar la eficiencia de codificación.

### 20 Breve descripción de los dibujos

Estos y otros objetos, ventajas y características de la descripción se harán evidentes a partir de la siguiente descripción de las mismas tomadas en conjunto con los dibujos adjuntos que ilustran una realización específica de la presente invención.

La figura 1 es un diagrama de bloques que muestra un ejemplo de un codificador de acuerdo a HEVC.

25 La figura 2 es un diagrama de bloques que muestra un ejemplo de un decodificador de acuerdo a HEVC.

La figura 3 es un diagrama que muestra un ejemplo de una configuración de una imagen en el procesamiento paralelo de frente de onda (Wavefront Parallel Processing, WPP).

La figura 4 es un diagrama que muestra un ejemplo de una relación entre una porción (slice) normal y una porción dependiente en el procesamiento paralelo de frente de onda.

30 La figura 5 es un diagrama que muestra un ejemplo de una cabecera de paquete.

La figura 6 es un diagrama que muestra un ejemplo de una cabecera de porción de una porción de entropía o una porción dependiente.

La figura 7 es un diagrama que muestra las dependencias y la transmisión de señal cuando se usa una porción normal.

35 La figura 8 es una vista esquemática que muestra las dependencias y las transmisiones de señal cuando se usan una porción dependiente y una porción de entropía.

La figura 9A es un diagrama que muestra un ejemplo que muestra una implementación de sintaxis de las dependencias entre capas, dependencias temporales y dependencias entre porciones en HM8.0.

40 La figura 9B es un diagrama para explicar los pasos de análisis a ser realizados para analizar las dependencias entre capas en HM8.0.

La figura 9C es un diagrama para explicar los pasos de análisis a ser realizados para analizar las dependencias entre capas en HM8.0.

La figura 10 es un diagrama que muestra un ejemplo de la posición de la bandera `dependent_slice_flag`.

45 La figura 11 es un diagrama que muestra un ejemplo de sintaxis cuando la condición de análisis con respecto a la bandera `dependent_slice_enabled_flag` en la figura 10 se elimina.

La figura 12 es un diagrama que muestra un ejemplo de sintaxis cuando la bandera `dependent_slice_flag` se mueve antes de la bandera `first_slice_in_pic_flag`. La figura 13 es un diagrama que muestra un ejemplo de sintaxis cuando la bandera `dependent_slice_flag` se mueve antes del elemento de sintaxis `slice_address`.

5 La figura 14 es un diagrama que muestra un ejemplo de sintaxis cuando la bandera `dependent_slice_flag` se mueve dentro de la cabecera NAL.

La figura 15 es un diagrama que muestra un ejemplo de sintaxis de una cabecera de porción de una porción dependiente cuando un tipo nuevo se añade a los tipos de unidades de NAL usados para porciones dependientes.

La figura 16 es un diagrama que muestra un ejemplo de sintaxis de una cabecera de porción y de una cabecera de unidad NAL cuando se asume que la bandera `dependent_slice_flag` se establece a 1 para ciertos tipos de NALU.

10 La figura 17 muestra una configuración general de un sistema que proporciona contenido para implementar servicios de distribución de contenidos.

La figura 18 muestra una configuración general de un sistema de difusión digital.

La figura 19 muestra un diagrama de bloques que ilustra un ejemplo de una configuración de un televisor.

15 La figura 20 muestra un diagrama de bloques que ilustra un ejemplo de una configuración de una unidad de reproducción/grabación de información que lee y escribe información de y en un medio de grabación que es un disco óptico.

La figura 21 muestra un ejemplo de una configuración de un medio de grabación que es un disco óptico.

La figura 22A muestra un ejemplo de un teléfono móvil.

La figura 22B es un diagrama de bloques que muestra un ejemplo de una configuración de un teléfono móvil.

20 La figura 23 ilustra una estructura de datos multiplexados.

La figura 24 muestra esquemáticamente cómo cada flujo es multiplexado en datos multiplexados.

La figura 25 muestra cómo un flujo de video es almacenado en un flujo de paquetes PES en más detalle.

La figura 26 muestra una estructura de paquetes TS y paquetes origen en los datos multiplexados.

La figura 27 muestra una estructura de datos de una PMT.

25 La figura 28 muestra una estructura interna de información de datos multiplexados.

La figura 29 muestra una estructura interna de información de atributos de flujo.

La figura 30 muestra pasos para identificar datos de video.

30 La figura 31 muestra un ejemplo de una configuración de un circuito integrado para implementar el método de codificación de imágenes en movimiento y el método de decodificación de imágenes en movimiento según cada una de las realizaciones.

La figura 32 muestra una configuración para conmutar entre frecuencias portadoras.

La figura 33 muestra los pasos para identificar datos de video y conmutar entre frecuencias portadoras.

La figura 34 muestra un ejemplo de una tabla de búsqueda en la cual los estándares de datos de video están asociados con frecuencias portadoras;

35 La figura 35A es un diagrama que muestra un ejemplo de una configuración para compartir un módulo de una unidad de procesamiento de señal.

La figura 35B es un diagrama que muestra otro ejemplo de una configuración para compartir un módulo de la unidad de procesamiento de señal.

### Descripción de las realizaciones

40 (Base de conocimientos fundamentales de la presente descripción)

Respecto al método de codificación de imagen y el método de decodificación de imagen descritos en la sección de Antecedentes, los inventores han encontrado el siguiente problema.

Primero, se describirán un aparato de codificación de imagen y un aparato de decodificación de imagen en HEVC.

Una señal de video introducida en un aparato de codificación de imagen es una secuencia de imágenes llamadas tramas (imágenes). Cada trama incluye una matriz de dos dimensiones de píxeles. Todos los estándares mencionados anteriormente basados en codificación de video híbrida incluyen el particionado de cada trama de video individual en bloques más pequeños que incluyen una pluralidad de píxeles. El tamaño de los bloques puede variar, por ejemplo, según el contenido de la imagen. El método de codificación se puede variar normalmente sobre una base por bloque. El tamaño más grande posible para tal bloque, por ejemplo en HEVC, es 64 x 64 píxeles. Se denomina la unidad de codificación más grande (Largest Coding Unit, LCU). La LCU se puede particionar recursivamente en 4 CU.

En AVC según H.264/MPEG-4, un macro bloque (denotando usualmente un bloque de 16 x 16 píxeles) era el elemento de imagen básico, para el cual se realiza la codificación. El macro bloque puede dividirse en subbloques más pequeños. Los pasos de codificación incluidos en el método de decodificación se realizan sobre una base por subbloque.

#### 1-1. Codificación de video híbrida

A continuación se describe simplemente la codificación de video híbrida.

Normalmente, los pasos de codificación de una codificación de video híbrida incluyen una predicción espacial y/o temporal (predicción de espacio y/o predicción de tiempo). Por consiguiente, cada bloque que va a codificarse primero se predice usando bien los bloques en sus proximidades espaciales o los bloques en sus proximidades temporales, esto es, tramas de video codificadas anteriormente. Entonces se calcula un bloque residual que es una diferencia entre el bloque que va a codificarse y su resultado de predicción. A continuación, el bloque residual se transforma desde el dominio del espacio (píxel) a un dominio de la frecuencia. La transformación pretende reducir la correlación del bloque de entrada.

Además, se cuantifican los coeficientes de transformación obtenidos de la transformación. Esta cuantificación es la compresión con pérdidas (irreversible). Generalmente, los valores de los coeficientes de la transformación comprimidos se comprimen más sin pérdidas por una codificación de entropía. Además, la información adicional necesaria para la reconstrucción de la señal de video codificada se codifica y se proporciona junto con la señal de video codificada. Esto es por ejemplo información sobre la predicción espacial, la predicción temporal, y/o la cuantificación.

#### 1-2. Configuración del aparato de codificación de imagen

La figura 1 es un ejemplo de un aparato de codificación de una AVC según H.264/MPEG-4 típico y/o de un aparato de codificación de imagen HEVC (codificador 100).

Como se muestra en la figura 1, el codificador 100 incluye un sustractor 105, una unidad de transformación 110, una unidad de cuantificación 120, una unidad de transformación inversa 130, un sumador 140, un filtro de desbloqueo 150, un filtro de bucle adaptativo 160, una memoria de trama 170, una unidad de predicción 180, y un codificador de entropía 190.

La unidad de predicción 180 dirige una señal de predicción s2 mediante predicción temporal o predicción espacial. El tipo de predicción usado en la unidad de predicción 180 puede variarse sobre una base por trama o sobre una base por bloque. La predicción temporal se denomina predicción inter, y la predicción espacial se denomina predicción intra. La codificación que usa una señal de predicción s2 mediante predicción temporal se denomina codificación inter, y la codificación que usa una señal de predicción s2 mediante predicción espacial se denomina codificación intra. En la derivación de una señal de predicción que usa predicción temporal, se usan las imágenes codificadas almacenadas en una memoria. En la derivación de una señal de predicción que usa predicción espacial, se usa un valor de píxel de frontera de bloques próximos codificados o decodificados almacenados en una memoria. El número de direcciones de predicción en la predicción intra depende del tamaño de la unidad de codificación (Coding Unit, CU). Debe observarse que más adelante se describirán los detalles de la predicción.

El sustractor 105 primero determina una diferencia (señal de error de predicción e) entre un bloque actual que va a codificarse y una imagen de entrada (= señal de entrada s1) y un bloque de predicción correspondiente (= señal de predicción s2). La diferencia se usa para la predicción del bloque actual que va a codificarse. Debe observarse que la señal de error de predicción e también se denomina señal residual de predicción.

La unidad de transformación 110 transforma una señal de error de predicción e en coeficientes. Generalmente, la unidad de transformación 110 usa una transformación ortogonal tal como una transformación de coseno discreta (Discrete Cosine Transformation, DCT) de dos dimensiones o una versión entera de la misma. La transformación ortogonal puede reducir la correlación de la señal de entrada s1 (la señal de video antes de la codificación) eficientemente. Tras la transformación, los componentes de frecuencias más bajas son normalmente más importantes para la calidad de la imagen que los componentes de frecuencias altas de forma que se puede gastar más bits en codificar los componentes de baja frecuencia que en los componentes de alta frecuencia.

La unidad de cuantificación 120 cuantifica los coeficientes y deriva los coeficientes cuantificados.

5 El codificador de entropía 190 realiza la codificación de entropía en los coeficientes cuantificados. Los coeficientes cuantificados se comprimen sin pérdidas por la codificación de entropía. Además, mediante la codificación de entropía, el volumen de datos almacenado en la memoria y el volumen de datos (flujo de bits) que va a transmitirse pueden reducirse aún más. La codificación de entropía se realiza principalmente mediante la aplicación de codificación usando palabras claves de longitud variable. La longitud de una palabra clave se elige basándose en la probabilidad de su ocurrencia.

10 El codificador de entropía 190 transforma la matriz de dos dimensiones de coeficientes transformados en un vector de una dimensión. Normalmente, el codificador de entropía 190 realiza esta conversión a través de un denominado escaneo de zigzag. El escaneo de zigzag se inicia con el coeficiente de DC en la esquina superior izquierda de la matriz de dos dimensiones y escanea la matriz de dos dimensiones en una secuencia predeterminada que termina con un coeficiente de AC en la esquina inferior derecha. La energía se concentra normalmente en la parte superior izquierda de la matriz de dos dimensiones de coeficientes. Generalmente, cuando los coeficientes están ubicados en la esquina superior izquierda, son coeficientes de componentes de baja frecuencia. Cuando los coeficientes están ubicados en la esquina inferior derecha, son coeficientes de componentes de alta frecuencia. Por lo tanto, el escaneo de zigzag resulta en un vector donde normalmente los últimos valores son consecutivamente una pluralidad de unos o ceros. Esto permite una codificación eficiente que usa códigos de longitud de secuencia como parte de/antes de la codificación de entropía real.

20 Una AVC según H.264/MPEG-4 y una HEVC usan diferentes tipos de codificación de entropía. Aunque algunos elementos de sintaxis con codificados con longitud fija, la mayoría de los elementos de sintaxis se codifican con códigos de longitud variable. En particular, entre las sintaxis, los códigos de longitud variable de contexto adaptativo (CABAC) se usan para la codificación de señales de error de predicción (señales residuales de predicción). Generalmente, otros varios códigos enteros diferentes de los códigos de longitud variable de contexto adaptativo se usan para la codificación de otros elementos de sintaxis. Sin embargo, puede usarse la codificación aritmética binaria de contexto adaptativo.

25 Los códigos de longitud variable permiten una compresión sin pérdidas del flujo de bits codificado. Sin embargo, dado que las palabras clave tienen longitud variable, la decodificación debe realizarse secuencialmente en las palabras clave. En otras palabras, no es posible codificar o decodificar palabras clave antes de codificar o decodificar las palabras clave anteriores sin reiniciar (inicializar) el codificador de entropía o sin indicar por separado una posición de la palabra clave (punto de inicio) por donde empezar cuando se decodifica.

30 La codificación aritmética codifica una secuencia de bits en una única palabra clave basándose en un modelo de probabilidad predeterminado. El modelo de probabilidad predeterminado se determina según el contenido de la secuencia de video en el caso de CABAC. La codificación aritmética, y por tanto también CABAC, son más eficientes cuando la longitud del flujo de bits que va a codificarse es más grande. En otras palabras, CABAC aplicado a secuencias de bits es eficiente para bloques más grandes. Al principio de cada secuencia, se reinicia CABAC. En otras palabras, al principio de cada secuencia de video, su modelo de probabilidad se inicializa con algunos valores predefinidos o predeterminados.

El codificador de entropía 109 transmite, a un lado del decodificador, un flujo de bits que incluye coeficientes cuantificados codificados (señales de video codificadas) e información auxiliar codificada.

40 El H.264/MPEG-4 y AVC según H.264/MPEG-4 así como la HEVC incluyen dos capas funcionales, una capa de codificación de video (Video Coding Layer, VCL) y una capa de abstracción de red (Network Abstraction Layer, NAL). La VCL proporciona la funcionalidad de codificación tal como se describió anteriormente. La NAL encapsula elementos de información en unidades estandarizadas llamadas unidades de NAL según otra aplicación tal como transmisión sobre un canal o almacenamiento en un dispositivo de almacenamiento. Los elementos de información encapsulados por la NAL son, por ejemplo, (1) la señal de error de predicción codificada (datos de video comprimidos) u (2) otra información necesaria de decodificación de la señal de video tal como tipo de predicción, parámetro de cuantificación, vectores de movimiento, etc. Hay unidades de NAL de VCL que contienen los datos de video comprimidos y la información relacionada, así como unidades no VCL que encapsulan datos adicionales tales como conjunto de parámetros referentes a una secuencia de video completa, o una Información de mejora suplementaria (Supplemental Enhancement Information, SEI) que proporciona información adicional que puede usarse para mejorar el rendimiento de decodificación.

55 Algunas unidades de NAL no VCL incluyen, por ejemplo, conjuntos de parámetros. Un conjunto de parámetros es un conjunto de parámetros referentes a la codificación y decodificación de una cierta porción de la secuencia de video. Por ejemplo, hay un conjunto de parámetros de secuencia (Sequence Parameter Set, SPS) que incluye parámetros relevantes de codificación y decodificación de la secuencia de imágenes completa. En particular, el conjunto de parámetros de secuencia es una estructura de sintaxis que incluye elementos de sintaxis. En particular, los elementos de sintaxis se aplican a ninguna o más secuencias de video codificadas completas como se determina por el contenido de un seq\_parameter\_set\_id. El seq\_parameter\_set\_id es un elemento de sintaxis incluido en el conjunto de parámetros de imagen (descrito anteriormente) referido por el pic\_parameter\_set\_id. El pic\_parameter\_set\_id es un elemento de sintaxis incluido en cada cabecera de porción (slice header).

- 5 El conjunto de parámetros de imagen (Picture Parameter Set, PPS) es un conjunto de parámetros que definen parámetros aplicados a la codificación y decodificación de una imagen de secuencia de imágenes (secuencia de video). En particular, el PPS es una estructura de sintaxis que incluye elementos de sintaxis. Los elementos de sintaxis se aplican a ninguna o más imágenes codificadas completas tal como se determina mediante el pic\_parameter\_set\_id que es un elemento de sintaxis que se encuentra en cada cabecera de porción.
- Por consiguiente, es más simple seguir la pista de un SPS que del PPS. Esto es porque el PPS cambia para cada imagen, mientras que el SPS es constante para la secuencia de video completa que puede ser incluso de una duración de minutos u horas.
- 10 El codificador 100 incluye una unidad de reconstrucción (denominada unidad de decodificación) que deriva una señal reconstruida s3 (denominada una señal decodificada). Mediante la unidad de reconstrucción, una imagen reconstruida obtenida mediante la reconstrucción (decodificación) de la imagen codificada se genera y se almacena en la memoria de trama 170.
- La unidad de reconstrucción incluye la unidad de transformación inversa 130, el sumador 140, el filtro de desbloqueo 150, y el filtro de bucle adaptativo 160.
- 15 La unidad de transformación inversa 130, según los pasos de codificación descritos anteriormente, realiza la cuantificación inversa y la transformación inversa. Debe observarse que la señal de error de predicción e' derivada de la unidad de transformación inversa 130 es diferente de la señal de error de predicción e debida al error de cuantificación, llamado también ruido de cuantificación.
- 20 El sumador 140 deriva una señal reconstruida s' mediante la suma de una señal de error de predicción reconstruida e' reconstruida mediante la unidad de transformación inversa 130 a una señal de predicción s2.
- 25 El filtro de desbloqueo (deblocking filter) 150 realiza el procesamiento de filtrado de desbloqueo (deblocking filter processing) para reducir el ruido de cuantificación que se superpone a la señal reconstruida s' debido a la cuantificación. En este caso, dado que los pasos de codificación descritos anteriormente se realizan sobre una base por bloque, hay un caso donde una frontera de bloque es visible cuando el ruido se superpone (características de bloque del ruido). El ruido superpuesto se denomina ruido de bloque. En particular, cuando se realiza una cuantificación fuerte mediante la unidad de cuantificación 120, hay más fronteras de bloques visibles en la imagen reconstruida (imagen decodificada). Tal ruido de bloque tiene un efecto negativo sobre la percepción visual humana, lo que significa que una persona siente que la calidad de la imagen está deteriorada. Para reducir el ruido de bloque, el filtro de desbloqueo 150 realiza el procesamiento de filtrado de desbloqueo en cada señal reconstruida s' (bloque reconstruido).
- 30 Por ejemplo, en el procesamiento de filtrado de desbloqueo de AVC según H.264/MPEG-4, para cada área, se selecciona un procesamiento de filtrado adecuado para el área. En el caso de un grado alto de ruido de bloque, se aplica un filtro paso bajo (de banda estrecha) fuerte, mientras que para un grado bajo de ruido de bloque se aplica un filtro paso bajo (de banda ancha) más débil. La fuerza del filtro paso bajo se determina por la señal de predicción e2 y por la señal de error de predicción e'. El procesamiento de filtrado de desbloqueo generalmente suaviza los bordes de los bloques. Esto lleva a una calidad de imagen subjetiva mejorada de las señales decodificadas. La imagen filtrada se usa para la predicción de compensación de movimiento de la siguiente imagen. Dado que el procesamiento de filtrado también reduce los errores de predicción, puede mejorarse la eficiencia de codificación.
- 35 El filtro de bucle adaptativo 160 aplica un procesamiento de compensación adaptativo de muestra (Sample Adaptive Offset, SAO) y/o un procesamiento de filtrado de bucle adaptativo (Adaptive Loop Filter, ALF) a la imagen reconstruida s' tras el procesamiento de filtrado de desbloqueo en el filtro de desbloqueo 150, para derivar una señal reconstruida s3 (señal decodificada).
- 40 El procesamiento de filtrado de desbloqueo en el filtro de desbloqueo 150 pretende mejorar la calidad subjetiva. Mientras tanto, el procesamiento ALF y el procesamiento SAO en el filtro de bucle adaptativo 160 pretenden mejorar la fidelidad de píxel sabio (calidad "objetiva"). El procesamiento SAO se usa para añadir un valor de compensación a un valor de píxel para cada píxel que usa un valor de píxel de los píxeles inmediatamente próximos. El procesamiento ALF se usa para compensar la distorsión de la imagen causada por la compresión. Normalmente, el filtro usado en el procesamiento ALF es un filtro de Wiener con coeficientes del filtro determinados de forma que el error cuadrático medio (Mean Square Error, MSE) entre la señal reconstruida s' y la señal de entrada s1 se minimiza. Los coeficientes del filtro del procesamiento ALF se calculan y transmiten sobre una base trama a trama, por ejemplo. El procesamiento ALF puede aplicarse a la trama (imagen) completa o a áreas locales (bloques). La información auxiliar que indica qué áreas han de filtrarse se puede transmitir sobre una base bloque a bloque, una base trama a trama, o una base árbol cuaternario a árbol cuaternario.
- 45 El procesamiento de filtrado de desbloqueo en el filtro de desbloqueo 150 pretende mejorar la calidad subjetiva. Mientras tanto, el procesamiento ALF y el procesamiento SAO en el filtro de bucle adaptativo 160 pretenden mejorar la fidelidad de píxel sabio (calidad "objetiva"). El procesamiento SAO se usa para añadir un valor de compensación a un valor de píxel para cada píxel que usa un valor de píxel de los píxeles inmediatamente próximos. El procesamiento ALF se usa para compensar la distorsión de la imagen causada por la compresión. Normalmente, el filtro usado en el procesamiento ALF es un filtro de Wiener con coeficientes del filtro determinados de forma que el error cuadrático medio (Mean Square Error, MSE) entre la señal reconstruida s' y la señal de entrada s1 se minimiza. Los coeficientes del filtro del procesamiento ALF se calculan y transmiten sobre una base trama a trama, por ejemplo. El procesamiento ALF puede aplicarse a la trama (imagen) completa o a áreas locales (bloques). La información auxiliar que indica qué áreas han de filtrarse se puede transmitir sobre una base bloque a bloque, una base trama a trama, o una base árbol cuaternario a árbol cuaternario.
- 50 La memoria de trama (memoria intermedia de trama) 170 almacena parte de la imagen (señal reconstruida s3) codificada y reconstruida (decodificada). La imagen reconstruida almacenada se usa para decodificar un bloque codificado inter.
- 55 La unidad de predicción 180 deriva una señal de predicción s2 usando la (misma) señal que puede usarse tanto en

la parte codificadora como en la parte decodificadora, para mantener la compatibilidad entre la parte codificadora y la parte decodificadora. La señal que puede usarse tanto en la parte codificadora como en la parte decodificadora es una señal reconstruida s3 (señal de video tras el procesamiento de filtrado mediante el filtro de bucle adaptativo 160) en la parte codificadora que es codificada y luego reconstruida (decodificada), y una señal reconstruida s4 (señal de video tras el procesamiento de filtrado mediante el filtro de bucle adaptativo en la figura 2) en la parte decodificadora que es decodificada desde un flujo de bits.

La unidad de predicción 180, cuando genera una señal de predicción s2 mediante la codificación inter, predice usando la predicción de compensación de movimiento. Un estimador de movimiento de la unidad de predicción 180 (no ilustrado) encuentra el bloque que mejor se corresponde para el bloque actual de entre los bloques en las tramas de video previamente codificadas y reconstruidas. El bloque que mejor se corresponde entonces se convierte en una señal de predicción. El desplazamiento (movimiento) relativo entre el bloque actual y el bloque que mejor se corresponde se señala entonces como datos de movimiento incluidos en la información auxiliar en la forma de vectores de movimiento de tres dimensiones. La señal se transmite junto con los datos de video codificados. El vector de movimiento de tres dimensiones incluye dos vectores de movimiento de dos dimensiones espaciales y un vector de movimiento de una dimensión temporal. Con el fin de optimizar la precisión de la predicción, pueden determinarse vectores de movimiento con una resolución de subpíxel espacial, por ejemplo, resolución de medio píxel o de un cuarto de píxel. Un vector de movimiento con resolución de subpíxel espacial puede apuntar a una posición espacial en una trama ya reconstruida donde no hay disponible un valor de píxel real, esto es, una posición de subpíxel. Por lo tanto, se necesita la interpolación espacial de tales valores de píxel para realizar la predicción compensada de movimiento. Esto puede lograrse mediante un filtro de interpolación (integrado en la unidad de predicción 180 en la figura 1).

### 1-3. Configuración de un aparato decodificador de imagen

Se describirá una configuración de un decodificador (aparato decodificador de imagen) con referencia a la figura 2.

La figura 2 es un diagrama de bloques que muestra un ejemplo de un decodificador 200 según la norma de codificación de video HEVC o AVC según H.264/MPEG-4.

Tal como se muestra en la figura 2, el decodificador 200 incluye un decodificador de entropía 290, una unidad de transformación inversa 230, un sumador 240, un filtro de desbloqueo 250, un filtro de bucle adaptativo 260, una memoria de trama 270, y una unidad de predicción 280.

Un flujo de bits introducido al decodificador 200 (señal de video codificada) se transmite primero al decodificador de entropía 290.

El decodificador de entropía 290 extrae los coeficientes cuantificados codificados del flujo de bits y la información auxiliar codificada, y decodifica los coeficientes cuantificados codificados y la información auxiliar codificada. La información auxiliar, Tal como se describió anteriormente, es información necesaria para decodificar datos de movimiento (vector de movimiento) y modo de predicción (tipo de predicción).

El decodificador de entropía 290 transforma los coeficientes cuantificados decodificados en un vector de una dimensión en aquellos en un vector de dos dimensiones mediante el escaneo inverso. El decodificador de entropía 290 introduce, en la unidad de transformación inversa 230, los coeficientes cuantificados tras transformarlos en aquellos en un vector de dos dimensiones.

La unidad de transformación inversa 230 realiza la cuantificación inversa y la transformación inversa en los coeficientes cuantificados transformados en aquellos en un vector de dos dimensiones, para derivar una señal de error de predicción e'. La señal de error de predicción e' se corresponde con las diferencias obtenidas mediante la sustracción de la señal de predicción de la señal de entrada al codificador en el caso de no introducir ruido de cuantificación y de que no ocurran errores.

La unidad de predicción 280 dirige una señal de predicción s2 mediante la predicción temporal o la predicción espacial. La información tal como tipo de predicción incluida en la información auxiliar se usa en el caso de predicción intra (predicción espacial). Además, la información tal como datos de movimiento incluida en la información auxiliar se usa en el caso de predicción compensada de movimiento (predicción inter, predicción temporal).

El sumador 240 suma una señal de error de predicción e' obtenida de la unidad de transformación inversa 230 y una señal de predicción e2 obtenida de una unidad de predicción 280, para derivar una señal reconstruida s'.

El filtro de desbloqueo 250 realiza un procesamiento de filtrado de desbloqueo en una señal reconstruida s'. El filtro de bucle adaptativo 260 aplica el procesamiento SAO y el procesamiento ALF a la señal reconstruida s' para la cual el procesamiento de filtrado de desbloqueo se aplica mediante el filtro de desbloqueo 250. Una señal decodificada S4 obtenida de la aplicación del procesamiento SAO y el procesamiento ALF en el filtro de bucle adaptativo 260 se almacena en la memoria de trama 270. La señal decodificada S4 almacenada en la memoria de trama 270 se usa, en la unidad de predicción 280, para predecir el siguiente bloque actual a decodificarse o la imagen actual a ser



codificada.

1-4. Eficiencia del procesamiento

Generalmente, se considera la paralelización de procesamientos para mejorar la eficiencia de procesamiento del procesamiento de codificación y el procesamiento de decodificación.

5 En comparación con AVC según H.264/MPEG-4, HEVC tiene una función de soportar un procesamiento paralelo de alto nivel (procesamiento de paralelización) de la codificación y la decodificación. En HEVC, es posible dividir una trama en porciones, de forma similar a AVC según H.264/MPEG-4. En este caso, las porciones son grupos de LCU en el orden de escaneo. En AVC según H.264/MPEG-4, las porciones son decodificables de manera independiente, y no se aplica predicción espacial entre las porciones. Por lo tanto, el procesamiento paralelo puede realizarse sobre una base porción a porción.

Sin embargo, dado que las porciones poseen cabeceras significativamente grandes y hay falta de dependencia entre las porciones, se reduce la eficiencia de la compresión. Además, la codificación CABAC pierde eficiencia cuando se aplica a bloques de datos pequeños.

15 Para permitir un procesamiento paralelo más eficiente, se propone el procesamiento paralelo de frente de onda (Wavefront Parallel Processing, WPP). El WPP mantiene una dependencia constante que es diferente del procesamiento paralelo en el cual cada una de las porciones es independiente.

20 La siguiente descripción se hará refiriéndose al caso en el que una imagen comprende las LCU en cada una de los cuales las imágenes se disponen en una matriz y cada fila de LCU comprende una porción (véase la figura 3). En el WPP, entre las LCU que comprenden la fila 32 de la LCU actual, como el modelo de probabilidad de CABAC para resetear el estado de CABAC de la primera LCU (cabecera de la LCU), se usa el modelo de probabilidad de CABAC justo después de que el procesamiento de la segunda LCU de la fila 31 de la LCU anterior se complete. Se mantienen todas las dependencias entre bloques. Esto permite la paralelización de decodificación de las filas de la LCU. La temporización para iniciar el procesamiento de cada fila de la LCU se retrasa por dos LCU con respecto a la anterior. La información sobre los puntos de inicio para iniciar la decodificación de fila de la LCU se incluye en la cabecera de porción. El WPP se describe en detalle en la bibliografía no de patente 1.

Otro enfoque para mejorar la paralelización se denomina tejas (tiles). Por consiguiente, una trama (imagen) se particiona en tejas. Las tejas son grupos rectangulares de LCU. Las fronteras entre las tejas se establecen de manera que la imagen completa se particiona en una matriz. Las tejas se procesan en el orden de escaneo más rápido.

30 Todas las dependencias se rompen en las fronteras de las tejas. La codificación de entropía tal como CABAC también se resetea al inicio de cada teja. Solo el procesamiento de filtrado de desbloqueo y el procesamiento de compensación adaptativo de muestra pueden aplicarse sobre las fronteras de las tejas. Así, las tejas pueden codificarse y decodificarse en paralelo. Las tejas se describen en detalle en la bibliografía no de patente 2 y la bibliografía no de patente 3.

35 Además, para mejorar el concepto de porciones y hacerlo adecuado para la paralelización más que para resistencia a errores que era el propósito original de las porciones en AVC según H.264/MPEG-4, se ha propuesto el concepto de porciones dependientes y porciones de entropía.

En otras palabras, en HEVC, hay tres tipos de porciones soportadas: (1) porciones normales; (2) porciones de entropía; y (3) porciones dependientes.

40 Las porciones normales denotan porciones ya conocidas de AVC según H.264/MPEG-4. Se permite la predicción espacial entre porciones normales. En otras palabras, no se permite la predicción sobre las fronteras de las porciones. Esto significa que una porción normal se codifica sin referirse a cualquier otra porción. Para permitir decodificación independiente en tales porciones, se reinicia el CABAC al inicio de cada porción.

45 Cuando la porción que va a procesarse es una porción normal, el reinicio del CABAC incluye el procesamiento de finalización (procesamiento de terminación) procesamiento de codificación aritmética o procesamiento de decodificación aritmética en el final de la porción precedente, y el procesamiento de inicializar la tabla de contexto (tabla de probabilidad) a un valor por defecto al inicio de la porción normal.

50 Las porciones normales se usan al inicio de cada trama. En otras palabras, cada trama tiene que empezar con una porción normal. Una porción normal tiene una cabecera que incluye los parámetros necesarios para decodificar los datos de porción.

El término "porciones de entropía" denota porciones en las cuales se permite la predicción espacial entre la porción padre y la porción de entropía. El análisis de la porción padre y la porción de entropía se realiza de manera independiente.

Sin embargo, la porción padre (parent slice) es, por ejemplo, una porción normal justo antes de la porción de

- entropía. La porción padre se requiere para la reconstrucción de los valores de píxeles de la porción de entropía. Para permitir el análisis de manera independiente de las porciones de entropía, el CABAC también se reinicia al inicio de la porción. Como la cabecera de porción de las porciones de entropía, es posible usar una cabecera de porción que sea más corta que la cabecera de porción de la porción normal. Los elementos que faltan en la cabecera de porción de entropía se copian de la cabecera de porción padre.
- 5 Cuando la porción que va a procesarse es una porción de entropía, el reinicio del CABAC, de manera similar a la porción normal, incluye un procesamiento de finalización (procesamiento de terminación) en el final de la porción precedente, y un procesamiento de inicialización de la tabla de contexto (tabla de probabilidad) a un valor por defecto al inicio de la porción actual.
- 10 (3) La porción dependiente es similar a una porción de entropía, pero es parcialmente diferente en el procesamiento en el cual se reinicia el CABAC.
- 15 Cuando la porción que va a procesarse es una porción dependiente y el WPP no es efectivo, el reinicio del CABAC incluye un procesamiento de finalización en la porción precedente (procesamiento de terminación) y un procesamiento de inicialización de la tabla de contexto a un valor de estado del fin de la porción precedente. Cuando la porción que va a procesarse es una porción dependiente y el WPP no es efectivo, el reinicio del CABAC incluye un procesamiento de finalización en la porción precedente (procesamiento de terminación) y un procesamiento de inicialización de la tabla de contexto a un valor de estado tras el procesamiento de la LCU que pertenece a la porción precedente y es el segundo por el final de la izquierda y el inicio de la porción actual.
- 20 Tal como se describió anteriormente, el reinicio del CABAC siempre incluye el procesamiento de terminación. A la inversa, en el reinicio del CABAC, el estado del CABAC normalmente se retrasa.
- Las porciones dependientes no pueden analizarse sin una porción padre. Por lo tanto, las porciones dependientes no pueden decodificarse cuando no se recibe la porción padre. La porción padre normalmente es una porción precedente de las porciones dependientes en un orden de codificación, y una porción que incluye una cabecera de porción completa. Esto es lo mismo para la porción padre de la porción de entropía.
- 25 Tal como se describió anteriormente, las porciones dependientes y de entropía usan la cabecera de porción (en particular, la información de la cabecera de porción que falta en la cabecera de porción dependiente) de la porción inmediatamente precedente según el orden de codificación de las porciones. Esta regla se aplica recursivamente. La porción padre de la que depende la porción dependiente actual se reconoce como disponible para referencia. La referencia incluye el uso de predicción espacial entre las porciones, compartir los estados del CABAC, y similares.
- 30 Una porción dependiente usa las tablas de contexto del CABAC que se generan al final de la porción inmediatamente precedente. Así, una porción dependiente no inicializa las tablas del CABAC a los valores por defecto, pero, en cambio sigue usando las tablas de contexto ya desarrolladas. Se pueden encontrar más detalles con respecto a las porciones de entropía y dependientes en la bibliografía no de patente 3.
- 35 La HEVC proporciona varios perfiles. Un perfil incluye algunas configuraciones del aparato de codificación de imagen y del aparato de decodificación de imagen adecuadas para una aplicación particular. Por ejemplo, el "perfil principal" solo incluye porciones normales y dependientes, pero no porciones de entropía.
- 40 Tal como se describió anteriormente, las porciones codificadas se encapsulan en unidades de NAL, que además se encapsulan, por ejemplo, en un protocolo de tiempo real (RTP) y finalmente en paquetes del protocolo de internet (IP). Bien ésta, u otras pilas de protocolos, permiten transmitir el video codificado en redes orientadas a paquetes, tal como Internet o algunas redes propietarias.
- 45 Normalmente, las redes incluyen al menos uno o más enrutadores, que emplean hardware espacial que opera muy rápido. La función del enrutador es recibir paquetes IP, analizar las cabeceras de los paquetes IP y, en consecuencia, remitir los paquetes IP a sus destinos respectivos. Dado que los enrutadores necesitan manejar tráfico desde muchas fuentes, la lógica de manejo de paquetes necesita ser tan simple como sea posible. El requisito mínimo para el enrutador es comprobar el campo de dirección de destino en la cabecera IP para determinar qué ruta tomar para remitirlos. Para proporcionar más soporte para la calidad de servicio (Quality of Service, QoS), los enrutadores inteligentes (consciente del medio, *media-aware*) comprueban adicionalmente campos especializados en las cabeceras del protocolo de red, tal como cabecera IP, cabecera RTP, e incluso la cabecera de una NALU.
- 50 Tal como puede verse a partir de la descripción anterior de la codificación de video, los diferentes tipos de porciones definidas con el propósito de procesamiento paralelo, tal como porciones dependientes y porciones de entropía, son de importancia diferente con respecto a la distorsión de calidad sobre su daño. En particular, las porciones dependientes no pueden analizarse y decodificarse sin una porción padre. Esto es porque al inicio de la porción dependiente, el codificador o decodificador de entropía no puede reiniciarse. Por consiguiente, la porción padre es
- 55 más importante para la reconstrucción de la imagen o el video.
- En HEVC, las porciones dependientes y de entropía introducen una dimensión adicional de dependencia, a saber, la dependencia entre porciones (una dependencia dentro de la trama). Este tipo de dependencia no se considera por

los enrutadores.

5 Las dependencias descritas anteriormente y, en particular, la dependencia entre porciones no se considera a nivel de red. Sin embargo, sería deseable tener en cuenta la dependencia descrita anteriormente en el nivel de red para proporcionar un mejor soporte para la calidad de servicio. Por consiguiente, se necesita mejorar la flexibilidad del manejo de paquetes en el nivel de red mediante la consideración de las dependencias de las porciones.

(Detalles del problema)

[1-5. WPP y porciones dependientes]

10 Las porciones dependientes pueden usarse junto con las herramientas de procesamiento paralelo tales como procesamiento paralelo de frente de onda (WPP) y tejas. En particular, las porciones dependientes hacen que el frente de onda (subflujo) sea capaz de reducir el retardo de transmisión sin causar pérdida de codificación.

15 Además, las porciones dependientes sirven como puntos de inicio para los subflujos de CABAC dado que CABAC no se reinicia en las porciones dependientes. Además, la información que indica los puntos de inicio puede transmitirse en el flujo de bits para proporcionar los puntos de inicio para un posible análisis independiente. En particular, si más de dos subflujos CABAC se encapsulan en una porción normal o dependiente, los puntos de inicio se señalan explícitamente en la forma del número de bytes por subflujo. En este caso, el subflujo denota una porción de un flujo que es analizable independientemente gracias a los puntos de inicio. Adicionalmente, las porciones dependientes pueden usarse como puntos de inicio "marcadores", dado que cada porción dependiente necesita tener una cabecera de unidad de la NAL. Esto significa que los puntos de inicio pueden señalizarse con respecto a dichos marcadores.

20 Los dos enfoques, a saber la señalización de punto de inicio explícito y el marcado de los puntos de inicio a través de porciones dependientes, se usan conjuntamente.

Por lo general, el punto de inicio de cada unidad NAL (inicio de cada cabecera NAL) tiene que ser identificable. No hay requisito sobre la operación de identificación exacta. Por ejemplo, pueden aplicarse los dos siguientes métodos.

25 El primer método es un método de poner un código de inicio (por ejemplo, de 3 bytes de longitud) al inicio de cada cabecera NAL. El segundo método es un método de poner cada unidad NAL en un paquete diferente. Debido a la dependencia de las porciones, el tamaño de la cabecera de porción puede reducirse.

30 En cuanto a las porciones de entropía, el método permite análisis del CABAC en paralelo. Esto es porque el CABAC verdaderamente se reinicia al inicio de cada porción de entropía. En el caso de procesamiento paralelo del CABAC, CABAC representa un cuello de botella que puede superarse mediante el análisis del CABAC en paralelo seguido de una operación de reconstrucción de píxel secuencial. En particular, la herramienta de paralelización del WPP permite decodificar cada fila de la LCU por un núcleo de procesamiento (código de propiedad intelectual (núcleo de IP), un bloque de función). Debe observarse que la asignación de las filas de las LCU a los núcleos puede ser diferente. Por ejemplo, dos filas pueden ser asignadas a un núcleo, y una fila puede ser asignada a dos núcleos.

35 La figura 3 es un diagrama que muestra un ejemplo de una configuración de una imagen 300. En la figura 3, una imagen 300 se subdivide en filas 31 a 3m (m es el número ordinal de la LCU) de unidades de codificación más grandes (LCU). Cada una de las filas 3i (i = 1 a m) de la LCU comprende las LCU 3i1 a 3in (n es el número ordinal de columna de LCU) que se disponen en una fila. La fila 3i de la LCU corresponde al "Frente de onda i". El procesamiento paralelo puede realizarse para los frentes de onda. La flecha del estado del CABAC en la figura 3 denota una relación entre la LCU que se refiere al estado del CABAC y el destino de referencia.

40 Específicamente, en la figura 3, primero, entre las LCU incluidas en la fila 31 de la LCU, el procesamiento (codificación o decodificación) empieza para la cabeza de la LCU 311. El procesamiento de las LCU se realiza en un orden desde la LCU 311 a la 31n. Tras realizar el procesamiento en las dos primeras LCU 311 y 312 en la fila 31 de la LCU, empieza el procesamiento en la fila 32 de la LCU. En el procesamiento de la primera fila 321 de la LCU de la columna 32 de la LCU, como se muestra en la flecha del estado del CABAC en la figura 3, el estado del CABAC justo antes del procesamiento en la LCU 312 en la fila 31 de la LCU en la primera fila se usa como el estado inicial del estado del CABAC. En otras palabras, hay un retraso de dos LCU entre los dos procesamientos paralelos.

45 La figura 4 es un diagrama que muestra un ejemplo del caso donde se usa una porción dependiente que usa WPP. Las filas 41 a 43 de la LCU se corresponden con los "Frente de onda 1", "Frente de onda 2" y "Frente de onda 3", respectivamente. Las filas 41 a 43 de la LCU se procesan mediante sus respectivos núcleos independientes. En la figura 4, la fila 41 de la LCU es una porción normal, y las filas 42 a 4m de la LCU son porciones dependientes.

55 Las porciones dependientes hacen que el WPP sea capaz de reducir el retraso. Las porciones dependientes tienen una cabecera de porción no completa. Además, las porciones dependientes pueden decodificarse independientemente de las otras porciones mientras los puntos de inicio (o el punto de inicio de las porciones dependientes, que se conoce por lo general tal como se describió anteriormente) sean conocidos. En particular, las porciones dependientes pueden hacer que el WPP sea adecuado también para aplicaciones de bajo retardo sin

incurrir en pérdidas de codificación.

5 En el caso habitual de encapsulamiento de los subflujos (filas de la LCU) en porciones, es obligatorio insertar los puntos de inicio explícitos en la cabecera de porción para asegurar la codificación y decodificación de entropía paralela. Como resultado una porción está lista para la transmisión solamente después de que el último subflujo de la porción se haya codificado completamente. La cabecera de porción está completa solamente después de que se haya completado la codificación de todos los subflujos en la porción. Esto significa que la transmisión de los inicios de una porción no puede empezarse a través de la fragmentación de paquetes en la capa RTP/IP hasta que la porción completa haya terminado.

10 Sin embargo dado que las porciones dependientes pueden usarse como marcadores de puntos de inicio, no se requiere la señalización de puntos de inicio explícita. Por lo tanto, es posible dividir una porción normal en muchas porciones dependientes sin pérdida de codificación. Las porciones dependientes pueden transmitirse tan pronto como se haya completado la codificación del subflujo encapsulado (o incluso antes en el caso de fragmentación de paquetes).

15 Las porciones dependientes no rompen la dependencia de predicción espacial. Las porciones dependientes ni siquiera rompen la dependencia de análisis. Esto es porque habitualmente el análisis de la porción dependiente actual requiere el estado del CABAC de la porción anterior.

Cuando no se permiten las porciones dependientes, entonces cada fila de la LCU puede configurarse para ser una porción. Tal configuración baja el retraso de transmisión, pero al mismo tiempo, lleva a una pérdida de codificación bastante alta tal como se comentó en la sección de Antecedentes anterior.

20 Alternativamente, la trama (imagen) completa se encapsula en una única porción. En este caso, los puntos de inicio para los subflujos (filas de la LCU) necesitan señalarse en la cabecera de porción para permitir su análisis paralelo. Como resultado, hay un retraso de transmisión en el nivel de trama. En otras palabras, la cabecera necesita modificarse después de que la trama completa se codifique. Tener una imagen completa encapsulada en una única porción por sí misma no incrementa el tiempo de retraso. Por ejemplo, la transmisión de algunas partes de la porción puede empezar antes de que termine la codificación completa. Sin embargo, si se usa el WPP, entonces la cabecera de porción necesita modificarse después para escribir los puntos de inicio. Por lo tanto, la porción completa necesita retrasarse para la transmisión.

30 Por tanto, el uso de porciones dependientes permite una reducción del retraso. Como se muestra en la figura 4, se divide una imagen 400 en una fila 41 de la LCU que es una porción normal, y las filas 42 a 4m de la LCU que son porciones dependientes. Cuando cada fila de la LCU es una porción dependiente, puede lograrse un retardo de transmisión de una fila de la LCU sin pérdidas de codificación. Esto se produce por el hecho de que las porciones dependientes no rompen ninguna dependencia espacial y no reinician el motor del CABAC.

#### [1-6. Configuración de paquete]

35 Tal como se describió anteriormente, los enrutadores de red tienen que analizar las cabeceras de los paquetes para permitir proporcionar calidad de servicio. La calidad de servicio es diferente según el tipo de aplicación y/o prioridad del servicio y/o de la relevancia de los paquetes para la distorsión causada por su pérdida de paquetes.

La figura 5 es un diagrama que muestra un ejemplo de una encapsulación (paquetización) de un flujo de bits.

40 Generalmente, el protocolo de tiempo real (RTP) se usa para la paquetización. El RTP se usa generalmente para la transmisión de medios en tiempo real. Las longitudes de las cabeceras de los respectivos protocolos involucrados son básicamente fijas. Las cabeceras de protocolos tienen campos de extensión. Los campos de extensión pueden extender la longitud de las cabeceras en 4 bytes. Por ejemplo, la cabecera IP puede extenderse hasta 20 bytes. Los elementos de sintaxis en las cabeceras IP, protocolo de datagrama de usuario (User Datagram Protocol, UDP) y RTP también son fijos en su longitud.

45 La figura 5 muestra una cabecera 500 de paquete incluida en un paquete IP. La cabecera de paquete mostrada en la figura 5 incluye una cabecera IP 510, una cabecera UDP 530, una cabecera RTP 540, una cabecera de carga útil de H264 RTP 560, y una cabecera NAL 570. La cabecera IP 510 es una cabecera con una longitud de 20 bytes con un campo de extensión 520 de 4 bytes. La carga del paquete IP es un paquete UDP. El paquete UDP incluye una cabecera UDP 530 con una longitud de 8 bytes y la carga UDP. La carga UDP está formada por el paquete RTP. El paquete RTP incluye una cabecera RTP 540 con una longitud de cabecera de 12 bytes y un campo de extensión 550 de 4 bytes. El paquete RTP puede extenderse selectivamente mediante el campo de extensión. La carga del paquete RTP incluye una cabecera de carga útil H264 RTP 560 especial con una longitud de 0 a 3 bytes seguida por una cabecera NAL 570 de la HEVC cuya longitud es de 2 bytes. La carga de la NALU que incluye el paquete de video codificado sigue a las cabeceras del paquete 500 (no mostrada en la figura 5).

55 Los enrutadores que son capaces de proporcionar una calidad de servicio mejorada se denominan elementos de red conscientes del medio (Media Aware Network Elements, MANE). Los elementos de red conscientes del medio comprueban algunos de los campos de las cabeceras de paquete mostradas en la figura 5. Por ejemplo, MANE se

denomina “temporal\_id” y se incluye en la cabecera NAL 570 o el número de orden de decodificación incluido en la cabecera RTP 540 puede comprobarse para detectar pérdidas y el orden de presentación de contenidos de paquetes recibidos. Los enrutadores (elementos de red) manejan los paquetes tan rápido como es posible para permitir un alto rendimiento en la red. La lógica se requiere para acceder a los campos en las cabeceras de paquete de manera rápida y simple, para mantener la complejidad del procesamiento del elemento de red baja.

La NALU se encapsula mediante la cabecera 500. La NALU puede incluir datos de porción cuando hay presente una cabecera de porción.

La figura 6 es un diagrama que muestra un ejemplo de una sintaxis de cabecera de porción 600. El elemento de sintaxis dependent\_slice\_flag 601 es un elemento de sintaxis que indica si una porción es una porción dependiente o no. Este elemento de sintaxis puede usarse para identificar la dependencia entre porciones. Sin embargo, la cabecera de porción es el contenido de una NALU. El análisis de los elementos de sintaxis antes de la dependent\_slice\_flag 601 requiere una lógica bastante complicada. Este es un nivel que no puede considerarse eficientemente mediante los enrutadores ordinarios tal como se mostrará a continuación.

Tal como se describió anteriormente, una NALU incluye información común para una pluralidad de porciones tales como conjuntos de parámetros, o incluye porciones directamente codificadas con información necesaria de decodificación incluida en la cabecera de porción. La sintaxis de una cabecera de porción usada para una porción de entropía o dependiente se ejemplifica en la figura 6. La figura 6 muestra una tabla con una estructura de cabecera de porción. Cuando el elemento de sintaxis “dependent\_slice\_flag” se establece a 1, se requieren todas las porciones hasta la primera porción normal (una porción que no es una porción de entropía ni una porción dependiente) anteriores a la porción actual en el orden de decodificación. Cuando las porciones no se decodifican, en general, la porción dependiente actual no puede decodificarse. En algunos casos especiales, por ejemplo, la porción dependiente puede ser decodificable cuando hay disponible alguna otra información lateral señalada o derivada. El elemento de sintaxis dependent\_slice\_flag 601 se incluye aproximadamente en la mitad de la cabecera de porción. Además, la cabecera de porción incluye el número de subflujos del CABAC en la porción actual señalada por el elemento de información num\_entry\_points\_offsets 602 y el número de bytes en un subflujo 603 señalado por el elemento de sintaxis entry\_points\_offsets [i]. En este caso, el elemento de información num\_entry\_points\_offsets 602 corresponde al número de puntos de entradas. Además, i es un entero y un índice que denota los puntos de entrada particulares (compensaciones de los puntos de entrada). El número de bytes en un subflujo denotado por entry\_points\_offsets [i] 603 permite una navegación fácil en el flujo de bits.

#### [1-7. Dependencia de imagen]

Tal como se describió anteriormente, hay varios tipos de dependencias resultantes del enfoque de codificación HEVC.

La figura 7 es un diagrama que muestra las dependencias y su señalización en el caso en el que sólo se usan porciones normales, esto es, no porciones dependientes ni de entropía. La figura 7 muestra tres imágenes 710, 720 y 730.

La imagen 710 es una imagen de capa base llevada en dos NALU de la VCL, a saber unidad de NAL de VCL 1 y unidad de NAL de VCL 2. POC indica el orden en el que se van a representar las imágenes. La NALU de VCL incluye un elemento de sintaxis que indica si una imagen pertenece a una capa base o a una capa de mejora, y un elemento de sintaxis temporal\_id. El elemento de sintaxis que indica si una imagen pertenece a una capa base o a una capa de mejora se transmite bajo un estado en la cabecera NAL 570 de la cabecera de paquete 500 mostrada en la figura 5. El elemento de sintaxis “temporal\_id” también se transmite bajo un estado en la cabecera NAL 570. El elemento de sintaxis “temporal\_id” indica el grado de dependencia de las otras imágenes. Por ejemplo, las imágenes o porciones codificadas con temporal\_id=0 son decodificables independientemente de otras imágenes/porciones que tienen una temporal\_id más alta. Debe observarse que en HEVC, el temporal\_id se señala en la cabecera NAL como nuh\_temporal\_id\_plus1 (obsérvese la figura 9A). En particular, la siguiente Expresión 1 puede aplicarse a la relación entre el temporal\_id usado en estos ejemplos y el elemento de sintaxis nuh\_temporal\_id\_plus1.

[Matemáticas 1]

$$\text{temporal\_id} = \text{nuh\_temporal\_id\_plus1} - 1$$

(Expresión 1)

Las porciones con temporal\_id=1 dependen de porciones de temporal\_id con un valor más bajo. En otras palabras, el valor de temporal\_id en este caso es 0. En particular, el elemento de sintaxis temporal\_id se refiere a la estructura de predicción de la imagen. En general, las porciones con un valor particular de temporal\_id dependen solo de porciones con un valor de temporal\_id menor o igual.

Por consiguiente, puede decodificarse primero una imagen 710 en la figura 7.

Una imagen 720 es una capa de mejora a la capa base de la imagen 710. Así, hay una dependencia que requiere

que la imagen 720 se decodifique después de decodificar la imagen 710. La imagen 720 incluye dos NALU, a saber unidad de NAL de VCL 3 y unidad de NAL de VCL 4. Ambas imágenes 710 y 720 tienen su valor de POC a 0. Esto significa que las imágenes 710 y 720 pertenecen a la misma imagen que va a presentarse de una vez. Las imágenes comprenden las capas base y de mejora.

5 La imagen 730 es una capa base que incluye dos NALU, a saber unidad de NAL de VCL 5 y unidad de NAL de VCL 6. La imagen 730 tiene un valor de POC de 1. Esto significa que la imagen (porción) 730 va a presentarse tras las imágenes 720 y 710. Además, la imagen 730 tiene el valor de `temporal_id` = 1. Esto significa que la imagen 730 depende temporalmente de una imagen con `temporal_id` = 0. Por consiguiente, basándose en la dependencia señalada en la cabecera NAL, la imagen 730 depende de la imagen 710.

10 La figura 8 es un diagrama que muestra las dependencias (grado de dependencia) y su señalización en el caso en que se usen porciones dependientes y no de entropía. La figura 8 muestra tres imágenes 810, 820, y 830. La figura 8 difiere de la figura 7 descrita anteriormente en que se añaden las dependencias de las porciones dependientes y de entropía señaladas en la cabecera de porción.

15 En la figura 7, la dependencia entre capas se muestra con el ejemplo de las imágenes 710 y 720. Además, la dependencia temporal se muestra en el ejemplo de las imágenes 710 y 730. Ambas dependencias se señalan en la cabecera NAL.

20 La dependencia entre capas tal como se muestra en la figura 8 es inherente a las porciones dependientes y de entropía. En particular, la trama 810 de capa base y la trama 820 de capa de mejora tienen ambas dos porciones. De las dos porciones, una es una porción padre (porción normal) y la otra es una porción dependiente. En la trama 810, la porción de la unidad de NAL de VCL 1 es la porción padre de la unidad de NAL de VCL 2. En la trama 820, la porción de la unidad de NAL de VCL 3 es la porción padre de la unidad de NAL de VCL 4. Tal como se describió anteriormente, el término “porción padre” de una porción dependiente se refiere a una porción de la cual dependen las porciones dependientes, esto es, la porción de la cual se usa la información de la cabecera de porción por las porciones dependientes. Es una regla que la primera porción precedente sea una porción que tiene una cabecera completa. La porción que tiene una cabecera completa es una porción normal, no otra porción dependiente, por ejemplo.

La sintaxis correspondiente de la cabecera de unidad de NAL y la cabecera de porción tal como se usan actualmente en HEVC y en particular en HM8.0 se describirá con referencia a la figura 9A.

30 La figura 9A es un diagrama que muestra la sintaxis de una cabecera de unidad de NAL 910 y la sintaxis de la cabecera de porción 920. En particular, las dependencias entre capas se planean (en la normalización actual) para señalarse en la cabecera de unidad de NAL usando un elemento de sintaxis `nuh_reserved_zero_6bits`. Las dependencias temporales se señalan usando el elemento de sintaxis `nuh_temporal_id_plus1`. La cabecera de porción 920 incluye una señal que indica la indicación de dependencia entre porciones. La indicación de dependencia entre porciones se indica mediante el elemento de sintaxis `dependent_slice_flag`. En otras palabras, la dependencia entre porciones (por ejemplo, dependencia temporal) se señala en la cabecera de porción, en algún sitio en la cabecera de porción.

Para analizar este elemento de sintaxis, todos los elementos de sintaxis que preceden a `dependent_slice_flag` deben analizarse así como el conjunto de parámetros de elementos de sintaxis necesario para analizar los elementos de sintaxis de la cabecera de porción que preceden a la `dependent_slice_flag`.

40 [1-8. Procesamiento en el enrutador]

Tal como se describió anteriormente, en la determinación de la forma del tráfico, es deseable tener en cuenta las dependencias introducidas por las porciones dependientes y de entropía, además de las dependencias señaladas en la cabecera NAL. Por ejemplo, un enrutador puede implementarse como una estación base móvil consciente del medio. El ancho de banda en el enlace descendente está muy limitado y necesita gestionarse muy cuidadosamente.

45 Asumamos el siguiente caso de ejemplo. Asumamos se tira que un paquete de manera aleatoria en el flujo de subida por un enrutador normal. En este caso, un elemento de red consciente del medio (MAME) descubre la pérdida del paquete mediante la comprobación del número del paquete. Tras comprobar la pérdida del paquete, MAME tira todos los paquetes que son dependientes del paquete tirado y que le siguen. Esta es una característica deseable para elementos de red conscientes del medio. De este modo, los paquetes se pueden tirar de manera más inteligente. Cuando un enrutador se determina a tirar una unidad NAL, deducirá inmediatamente que será necesario tirar también las porciones dependientes siguientes. En la sintaxis actual introducida en la figura 9A, el acceso de la `dependent_slice_flag` requiere analizar una cantidad considerable de información. No es esencial para el enrutamiento de paquetes o para operaciones de formas de tráfico en los enrutadores. Toda la información que es necesaria para descubrir las relaciones inter-capas y inter-temporal está presente en el conjunto de parámetros de video. El conjunto de parámetros de video es el conjunto más grande en la jerarquía de conjuntos de parámetros.

55 Por consiguiente, la información descrita anteriormente se señala en la cabecera NAL 570. Sin embargo, en el caso de la cabecera NAL y la cabecera de porción mostradas en la figura 9A, el acceso a la información dependiente de la porción requiere seguir conjuntos de parámetros adicionales tales como PPS y SPS. Esto, por otro lado, reutiliza

la capacidad de puertas de enlace o enrutadores conscientes del medio. Tal como se ve a partir de la figura 9A, la cabecera de porción 920 tiene que analizarse hasta la `dependent_slice_flag` y los parámetros analizados son inútiles para la operación de red.

5 Con el fin de poder analizar la dirección de la porción que precede a la `dependent_slice_flag`, se requieren los siguientes elementos de sintaxis de los elementos de sintaxis incluidos en el SPS930 tal como se muestran en la figura 9B. La figura 9B es un diagrama que muestra un ejemplo de sintaxis incluido en SPS.

- `pic_width_in_luma_samples` (señal 931 de referencia en la figura 9B)
- `pic_height_in_luma_samples` (señal 932 de referencia en la figura 9B)
- `log2_min_coding_block_size_minus3` (señal 933 de referencia en la figura 9B)
- 10 • `log2_diff_max_min_coding_block_size` (señal 934 de referencia en la figura 9B)

Estos parámetros se muestran en la tabla derecha de la figura 9B y se necesitan para obtener los parámetros de `slice_address`. El elemento de sintaxis `slice_address` se codifica con longitud variable (tal como puede verse cuando se mira a la longitud "v" en el descriptor, segunda columna, de `slice_address` y cabecera de porción 920 en la figura 9A). Para conocer la longitud de este parámetro codificado con longitud variable, se necesitan esos elementos de sintaxis del SPS. De hecho, para ser capaz de analizar la `dependent_slice_flag`, no se necesita el valor real del elemento de sintaxis `slice_address`. Sólo debe conocerse la longitud del elemento de sintaxis que es variable para que el proceso de análisis pueda continuar.

15 Por lo tanto, el SPS necesita analizarse hasta el punto 935 de los elementos de sintaxis en SPS 930 mostrado en la figura 9B. Se requiere almacenar los cuatro elementos de sintaxis. Luego se usarán en una fórmula para calcular la longitud del elemento de sintaxis `slice_address`.

Además, para acceder a la `dependent_slice_enabled_flag` que precede también a la `dependent_slice_flag`, el PPS necesita analizarse hasta el punto 945 de los elementos de sintaxis en el PPS mostrado en la figura 9C. La figura 9C es un diagrama que muestra un ejemplo de sintaxis incluido en el PPS. Debe observarse que los elementos de sintaxis cuyos métodos de análisis se han descrito con referencia a las figuras 9A a 9C y que están ubicados en la cabecera de porción y el SPS y el PPS no se requieren para operaciones de enrutamiento comunes. Además, algunos de los elementos de sintaxis no pueden saltarse simplemente dado que algunos de los elementos de sintaxis están codificados con códigos de longitud variable. Por consiguiente, aunque se realice un salto en el flujo de bits por una cantidad predefinida de bits, no es posible saltar hasta la `dependent_slice_enabled_flag`.

25 En otras palabras, para leer la `dependent_slice_flag` (indicación de dependencia), el MAME necesita ir más allá en la cabecera de porción (obsérvese la cabecera de porción 920) cuyo análisis es bastante complicado.

Específicamente, la bandera `first_slice_in_pic_flag` tiene que analizarse. La bandera `first_slice_in_pic_flag` es una bandera que indica si una porción es la primera en la imagen o no.

Entonces, `no_output_of_prior_pics_flag` cuya presencia es condicional en el tipo de la NALU tiene que analizarse.

30 Además, el `id_conjunto_parámetros_img` codificado con longitud variable tiene que decodificarse. El elemento de sintaxis `id_conjunto_parámetros_img` es un elemento de sintaxis que indica qué conjunto de parámetros se usa (un elemento de sintaxis que identifica el conjunto de parámetros). Mediante el análisis de `id_conjunto_parámetros_img`, puede identificarse el conjunto de parámetros que va a usarse.

Finalmente, se requiere el elemento de sintaxis `slice_address`. El elemento de sintaxis `slice_address` es un elemento de sintaxis que indica la posición de inicio de la porción. El elemento de sintaxis requiere además analizar el PPS y el SPS así como computación adicional.

40 Como última etapa, el valor de `dependent_slice_enabled_flag` (bandera que habilita una porción dependiente) tiene que obtenerse del PPS, para conocer si la `dependent_slice_flag` está presente en el flujo de bits o no. Cuando la `dependent_slice_enabled_flag` = 0, significa que la porción actual es una porción normal dado que las porciones dependientes no están habilitadas. Para obtener el valor de la `dependent_slice_enabled_flag`, se requiere el PPS para analizar hasta la mitad aproximadamente.

Desgraciadamente, los elementos de sintaxis anteriores a `dependent_slice_flag` no pueden saltarse y necesitan analizarse a diferencia del caso de los datos de la cabecera NAL y RTP, en los cuales las posiciones de los datos están predefinidas. Esto se produce por el hecho de que los elementos de sintaxis en la cabecera de la porción se codifican con longitud variable. Por lo tanto, la presencia y longitud del elemento necesitan computarse para cada unidad de NAL de VCL. Además, los datos de sesión adicionales necesitan almacenarse porque son necesarios más adelante (obsérvese PPS y SPS). Además, la presencia y longitud de algunos elementos de sintaxis depende de la presencia o valor de otros elementos de sintaxis posiblemente incluidos en otras estructuras de parámetros (los elementos de sintaxis se codifican condicionalmente).

En la normalización actual, hay una propuesta para señalar la estructura de dependencia de la secuencia de video en el conjunto de parámetros de video (VPS) que describe cuántas capas están contenidas en el flujo de bits y los indicadores de dependencia para indicar las diversas dependencias entre capas. El VPS se señala en el mismo principio del video, antes del primer SPS. Múltiples SPS pueden referirse a un único VPS. Esto significa que un VPS lleva información que es válida para múltiples secuencias de video. El objetivo principal del VPS es informar al enrutador o decodificador sobre el contenido del video que incluye información. Cuántas secuencias de video están incluidas y cómo están interrelacionadas. SPS es válido sólo en una secuencia de video en donde el VPS lleva información relacionada con múltiples secuencias de video.

Además, la característica de la información llevada en el VPS es especialmente informativa para los enrutadores. Por ejemplo, el VPS puede llevar información que se requiere para la configuración de la sesión de transmisión dado que el diseño no está finalizado. El enrutador analiza la información en el VPS. El enrutador, sin la necesidad de otros conjuntos de parámetros (solamente observando las cabeceras NAL), puede determinar qué paquetes de datos remitir al decodificador y qué paquetes tirar.

Sin embargo, para descubrir el VPS activo actualmente, se necesita realizar los siguientes pasos ordenados:

- analizar el id\_PPS en la cabecera de porción;
- analizar el id\_SPS en el PPS activo determinado por el id\_PPS; y
- analizar el id\_VPS en el SPS activo determinado por el id\_SPS.

Con el fin de resolver el problema descrito anteriormente, un método de codificación de imagen según un aspecto de la presente invención es un método de codificación de imagen para realizar procesamiento de codificación dividiendo una imagen en una pluralidad de porciones, comprendiendo el método de codificación de imagen transmitir un flujo de bits que incluye: una bandera que habilita una porción dependiente que indica si la imagen incluye o no una porción dependiente en la cual el procesamiento de codificación se realiza dependiendo de un resultado del procesamiento de codificación en una porción diferente de una porción actual; una dirección de porción que indica una posición de inicio de la porción actual; y una indicación de dependencia (dependent\_slice\_flag) que indica si la porción actual es la porción dependiente o no, en donde la bandera habilitada de porción dependiente está dispuesta en un conjunto de parámetros comunes a las porciones, la dirección de porción está dispuesta en una cabecera de la porción de la porción actual, y la indicación de dependencia está dispuesta en la cabecera de la porción, y está dispuesta antes de la dirección de porción y tras un elemento de sintaxis (id\_conjunto\_parámetros\_img) que identifica el conjunto de parámetros.

En el método de codificación de imagen descrito anteriormente, una indicación de dependencia de dependencia entre porciones está ubicada en una posición adecuada para analizar mediante el enrutador. Con esto, es posible codificar la indicación de dependencia la sintaxis independientemente, en otras palabras, incondicionalmente, de otros elementos de sintaxis.

Por ejemplo, la indicación de dependencia puede incluirse en el flujo de bits cuando la bandera que habilita una porción dependiente indica la inclusión de la porción dependiente.

Por ejemplo, la bandera que habilita una porción dependiente puede estar dispuesta al inicio del conjunto de parámetros.

Por ejemplo, cada una de las porciones puede incluir una pluralidad de macrobloques, y el procesamiento de codificación en la porción actual puede iniciarse después de que el procesamiento de codificación se realice en dos de los macrobloques incluidos en una porción inmediatamente precedente a la actual.

Por ejemplo, la indicación de dependencia puede no incluirse en una cabecera de porción de una porción que es la primera procesada de la imagen, de entre las porciones.

Con el fin de resolver el problema descrito anteriormente, un método de decodificación de imagen según un aspecto de la presente invención es un método de decodificación de imagen para realizar un procesamiento de decodificación dividiendo una imagen en una pluralidad de porciones, comprendiendo el método de decodificación de imagen extraer, de un flujo de bits codificado, una bandera que habilita una porción dependiente que indica si la imagen incluye o no una porción dependiente en la que el procesamiento de decodificación se realiza dependiendo de un resultado del procesamiento de decodificación de una porción diferente de la porción actual, una dirección de porción que indica un punto de inicio de la porción actual, y una indicación de dependencia que indica si la porción actual es o no la porción dependiente, donde la bandera que habilita una porción dependiente se dispone en un conjunto de parámetros común a las porciones, la dirección de porción se dispone en una cabecera de porción de la porción actual, y la indicación de dependencia se dispone en la cabecera de porción, y se dispone antes de la dirección de porción y tras un elemento de sintaxis que identifica el conjunto de parámetros.

Por ejemplo, la indicación de dependencia puede extraerse del flujo de bits cuando la bandera que habilita una porción dependiente indica la inclusión de la porción dependiente.



Por ejemplo, la bandera que habilita una porción dependiente puede disponerse al inicio del conjunto de parámetros.

Por ejemplo, cada una de las porciones puede incluir una pluralidad de macrobloques, y el procesamiento de decodificación en la porción actual puede empezar después de que el procesamiento de decodificación se realice en dos de los macrobloques incluidos en una porción inmediatamente precedente a la actual.

- 5 Por ejemplo, la indicación de dependencia puede no incluirse en una cabecera de porción de una porción que es la primera procesada de la imagen, entre las porciones.

10 Con el fin de resolver el problema descrito anteriormente, un aparato de codificación de imagen según un aspecto de la presente invención es un aparato de codificación de imagen que realiza un procesamiento de codificación dividiendo una imagen en una pluralidad de porciones, comprendiendo el aparato de codificación de imagen un codificador que transmite un flujo de bits que incluye: una bandera que habilita una porción dependiente que indica si la imagen incluye o no una porción dependiente en la que el procesamiento de codificación se realiza dependiendo de un resultado del procesamiento de codificación de una porción diferente de la porción actual; una dirección de porción que indica una posición de inicio de la porción actual; y una indicación de dependencia que indica si la porción actual es o no la porción dependiente, donde la bandera que habilita una porción dependiente se dispone en un conjunto de parámetros común a las porciones, la dirección de porción se dispone en una cabecera de porción de la porción actual, y la indicación de dependencia se dispone en la cabecera de porción, y se dispone antes de la dirección de porción y tras un elemento de sintaxis que identifica el conjunto de parámetros.

15 Con el fin de resolver el problema descrito anteriormente, un aparato de decodificación de imagen según un aspecto de la presente invención es un aparato de decodificación de imagen que realiza un procesamiento de decodificación dividiendo una imagen en una pluralidad de porciones, comprendiendo el aparato de decodificación de imagen un decodificador que extrae, de un flujo de bits codificado, una bandera que habilita una porción dependiente que indica si la imagen incluye o no una porción dependiente en la que el procesamiento de decodificación se realiza dependiendo de un resultado del procesamiento de decodificación de una porción diferente de la porción actual, una dirección de porción que indica una posición de inicio de la porción actual, y una indicación de dependencia que indica si la porción actual es o no la porción dependiente, donde la bandera que habilita una porción dependiente se dispone en un conjunto de parámetros común a las porciones, la dirección de porción se dispone en una cabecera de porción de la porción actual, y la indicación de dependencia se dispone en la cabecera de porción, y se dispone antes de la dirección de porción y tras un elemento de sintaxis que identifica el conjunto de parámetros.

20 Con el fin de resolver el problema descrito anteriormente, un aparato de codificación y decodificación de imagen según un aspecto de la presente invención incluye el aparato de codificación de imagen descrito anteriormente y el aparato de decodificación de imagen descrito anteriormente.

25 Según el método de codificación de imagen, en el método de decodificación de imagen, y similares que se configuraron anteriormente, una indicación de dependencia entre porción está ubicado en la sintaxis del flujo de bits relacionado con una porción independientemente de otros elementos. La indicación de dependencia se ubica, sin analizar innecesariamente otros elementos, separadamente de los otros elementos. En los ejemplos de HEVC anteriores, el indicador de la dependencia entre porciones `dependent_slice_flag` está señalado en una ubicación en la cual no se necesita analizar elementos de sintaxis irrelevantes para la operación de red.

30 Específicamente, la presente invención proporciona un aparato para analizar un flujo de bits de una secuencia de video de imágenes codificadas al menos parcialmente con un código de longitud variable e incluyendo unidades de datos que llevan porciones codificadas de secuencias de video. El aparato comprende un analizador para extraer del flujo de bits una indicación de dependencia que es un elemento de sintaxis que indica para una porción si la decodificación de longitud variable o análisis de la porción depende o no de otras porciones, donde la indicación de dependencia se extrae del flujo de bits de manera independiente de y sin necesidad de extraer otros elementos de sintaxis de antemano.

35 Tal aparato puede incluirse, por ejemplo, en el decodificador de entropía 290 en la figura 2. Cuando se hace referencia a extraer del flujo de bits, se entiende la extracción y, cuando sea necesario para la extracción, una decodificación de entropía. La codificación de entropía es una codificación de longitud variable, por ejemplo, la codificación aritmética tal como CABAC. Esto es, en HEVC, aplicado a la codificación de los datos de imagen. Las unidades de datos aquí se refieren, por ejemplo, a unidades de NAL o unidades de acceso. La expresión "sin necesidad de extraer otros elementos de sintaxis" se refiere a una situación en la cual la indicación de dependencia sólo está precedida por elementos, de los cuales se conoce la longitud y cuya presencia se conoce o se condiciona en elementos ya analizados o no codificados condicionalmente en absoluto.

40 La presente invención además proporciona un aparato para generar un flujo de bits de una secuencia de video codificada al menos parcialmente con un código de longitud variable y que incluye unidades de datos que llevan porciones codificadas de imágenes de video. El aparato comprende un generador de flujo de bits para incorporar en el flujo de bits una indicación de dependencia que es un elemento de sintaxis que indica para una porción si la decodificación de longitud variable de la porción depende o no de otras porciones, donde la indicación de dependencia está incrustada en el flujo de bits de manera independiente de y sin necesidad de incrustar otros

elementos de sintaxis de antemano.

Tal aparato puede incluirse, por ejemplo, en el decodificador de entropía 190 en la figura 1.

Según el método de codificación de imagen, en el método de decodificación de imagen, y similares que se configuraron anteriormente, el flujo de bits incluye datos de porción codificados y datos de cabecera con respecto a la porción, y la indicación de dependencia está ubicada en el flujo de bits al inicio de la cabecera de porción. Esto significa que la cabecera de porción empieza con los elementos de sintaxis que indican la dependencia de porción.

Debe observarse que la indicación de dependencia no tiene que estar ubicada en el mismo principio de la cabecera de porción. Sin embargo, es ventajoso cuando ningún otro elemento de sintaxis codificado condicionalmente y/o codificado con longitud variable precede la indicación de dependencia en la cabecera de porción.

Por ejemplo, la posición actual de la `dependent_slice_flag` se cambia con respecto a la técnica anterior descrita anteriormente para ubicarse al inicio de la cabecera de porción. Con este cambio, se consigue la reducción de la cantidad de elementos de sintaxis que necesitan analizarse. Se evitan operaciones de análisis complicadas de los enrutadores, tales como decodificación de longitud variable y análisis de información que requieren computación adicional y/o almacenamiento de parámetros adicionales para usos futuros y/o análisis de otros conjuntos de parámetros. Además, el número de conjuntos de parámetros a los que se requiere seguir la pista se reduce.

A continuación en el presente documento, las realizaciones se describen específicamente con referencia a los dibujos. Cada una de las realizaciones descritas a continuación muestra un ejemplo general o específico. Los valores numéricos, formas, materiales, elementos estructurales, la disposición y conexión de los elementos estructurales, etapas, el orden del procesamiento de los etapas etc. mostrados en las siguientes realizaciones son meros ejemplos, y por lo tanto no limitan el alcance de la presente invención. Por lo tanto, entre los elementos estructurales en las siguientes realizaciones, los elementos estructurales no mencionados en cualquiera de las reivindicaciones independientes se describen como elementos estructurales arbitrarios.

(Realización 1)

La figura 10 muestra un ejemplo de una sintaxis de un flujo de bits según la presente invención. Una cabecera NAL 1010 mostrada en la figura 10 es la misma que la cabecera NAL 910 mostrada en la figura 9A. En otras palabras, no hay cambios.

Sin embargo, la estructura de sintaxis de la cabecera de porción 1020 es diferente de la estructura de sintaxis de la cabecera de porción 920 en la figura 9A. En particular, en la cabecera de porción 1020, la `dependent_slice_flag` se sube en la cabecera de porción de forma que no hay elementos de sintaxis precediendo a la `dependent_slice_flag`. La `dependent_slice_flag` está codificada condicionalmente, está codificada usando un código de longitud variable, o recibe el análisis que requiere computación adicional.

Los elementos de sintaxis `bandera_primera_porción_en_imagen` y `dependent_slice_flag` realmente determinan ambos las dependencias espaciales. Los elementos de sintaxis se codifican inmediatamente después de la cabecera NAL de tal modo que no se necesita analizar ningún otro elemento de sintaxis. Dado que `bandera_primera_porción_en_imagen` también lleva información que está relacionada con las dependencias entre porciones, puede preceder la `dependent_slice_flag`. El elemento de sintaxis `bandera_primera_porción_en_imagen` es una bandera que se establece según la regla que cada trama tiene que empezar con una porción normal. Por consiguiente, cuando la `bandera_primera_porción_en_imagen` se establece, significa que la porción es una porción normal y por tanto independiente. Así, la `dependent_slice_flag` y la `bandera_primera_porción_en_imagen` pueden verse conjuntamente como una indicación de dependencias entre porciones.

En otras palabras, la indicación de dependencias puede definirse para incluir un primer indicador de porción si la porción es una primera porción en una imagen y una bandera de porción dependiente que indica si la decodificación de longitud variable de la porción depende de otras porciones. La primera porción en una imagen es siempre una porción para la que la decodificación de longitud variable no depende de otras porciones.

Ventajosamente, el flujo de bits incluye una bandera que habilita una porción dependiente si se pueden incluir porciones dependientes en el flujo de bits o no. La indicación de dependencia se incluye en el flujo de bits sólo cuando la bandera que habilita una porción dependiente indica que las porciones dependientes se pueden incluir en el flujo de bits. La bandera que habilita una porción dependiente está ubicada en el flujo de bits en un conjunto de parámetros común para una pluralidad de porciones y ubicado al inicio del conjunto de parámetros. El conjunto de parámetros puede ser, por ejemplo, el conjunto de parámetros de imagen que lleva parámetros para una única imagen. De manera alternativa, la bandera que habilita una porción dependiente está ubicada en un conjunto de parámetros de secuencia que lleva parámetros para la secuencia de imágenes (video) completa.

Sin embargo, en la presente invención, la `dependent_slice_flag` (indicación de dependencia) está codificada no condicionalmente en el elemento de sintaxis `dependent_slice_enabled_flag` (bandera que habilita una porción dependiente). En la presente realización, dado que el identificador de conjunto de parámetros de imagen está ubicado tras la indicación de dependencia, es una ventaja evitar un posible error de análisis en el caso en el que el

identificador de conjunto de parámetros de imagen esté señalado en la cabecera de porción.

Este cambio puede también verse como y/o interpolado mediante el cambio de la posición de los otros elementos de sintaxis requeridos en el conjunto de parámetros o cabeceras para reducir la cantidad de elementos de sintaxis que se requiere analizar para determinar las dependencias entre las porciones.

- 5 Por ejemplo, el elemento de sintaxis `dependent_slice_flag` en la cabecera de porción de la presente sintaxis de HM8.0 sólo está presente cuando el valor del elemento de sintaxis "`dependent_slice_enabled_flag`" indica que el uso de porciones dependientes en el flujo de bits está habilitado. La habilitación de las porciones dependientes y por tanto, también el elemento de sintaxis "`dependent_slice_enabled_flag`" están incluidos en el PPS tal como se muestra en la figura 9C. Por consiguiente, el elemento de sintaxis "`dependent_slice_enabled_flag`" en el PPS se sube en la sintaxis del PPS para simplificar su análisis necesario para analizar la `dependent_slice_flag` (por ejemplo, el inicio del conjunto de parámetros). Esto puede ser también útil cuando la `dependent_slice_flag` está codificada detrás del `id_conjunto_parámetros_img` (el elemento de sintaxis que identifica el conjunto de parámetros). Esto es porque al hacerlo, el error de análisis se evita aun cuando la bandera que habilita una porción dependiente está condicionando la presencia de la indicación de dependencia.
- 10
- 15 En vez de subir la "`dependent_slice_enabled_flag`" en el PPS, la "`dependent_slice_enabled_flag`" puede moverse del PPS al SPS y/o VPS para que no se requiera seguir la pista de los conjuntos de parámetros que están más abajo en la jerarquía.

En otras palabras, según la presente realización, la posición de los elementos de sintaxis requeridos se cambia para reducir la cantidad de conjuntos de parámetros a los que se necesita seguir la pista. Esto también reduce la complejidad del análisis. Los "parámetros requeridos" en este contexto significa los parámetros destinados que contribuyen a determinar si una porción es una porción con dependencia entre porciones o no. Una primera posibilidad aplicable directamente a la HEVC es proporcionar la indicación de dependencia al inicio de la cabecera de porción dependiente e incondicionalmente en la bandera que habilita una porción dependiente tras el indicador de conjunto de parámetros que identifica el conjunto de parámetros en el que está incluida la bandera que habilita una porción dependiente. La indicación de dependencia puede estar condicionada en la bandera que habilita una porción dependiente. Subir la bandera que habilita una porción dependiente en el PPS o mover la bandera que habilita una porción dependiente al SPS puede ser beneficioso para cualquiera de esas posibilidades. En particular, esto es beneficioso para la segunda posibilidad, en la que la bandera que habilita una porción dependiente se necesita para analizar la indicación de dependencia.

20

25

30 Tal como puede verse a partir en la figura 10, la cabecera de unidad de NAL, junto con la porción relevante de la cabecera de porción, tiene 18 bits (14 bits de la cabecera NALU y 2 bits de la cabecera de porción). Según este ejemplo, un elemento de red consciente del medio puede operar para un paquete de porción actual como sigue. Si una porción anterior se tira, que es una porción normal, de entropía o dependiente, el elemento de red comprueba los dos primeros bits de la cabecera de porción actual, que son la `first_slice_in_pic_flag` y (en el caso en el que las porciones dependientes estén permitidas en el flujo de bits) la `dependent_slice_flag`.

35

Cuando el tipo de unidad NAL es un tipo de unidad de NAL de VCL y los dos últimos bits de los 18 bits comprobados son "01", la unidad NAL se tira. En particular, cuando el primer bit de la cabecera de porción es "1", entonces es la primera porción en la imagen que es (según las reglas) una porción no dependiente. Cuando el primer bit de la cabecera de porción es "0" y el siguiente bit de la cabecera de porción es también "0", la porción es no dependiente. Por consiguiente, solo cuando los dos primeros bits de la cabecera de porción son "01", la porción es dependiente. Además, la porción debería tirarse dado que no puede decodificarse cuando la porción padre ya se ha tirado. Por consiguiente, las banderas `first_slice_in_pic_flag` y `dependent_slice_flag` pueden verse como una extensión de la cabecera NAL, aunque pertenecen a la sintaxis de cabecera de porción.

40

Por consiguiente, la presente realización también proporciona como uno de sus aspectos un enrutador de red para recibir, analizar y remitir paquetes de red a sus destinos. El enrutador incluye una unidad receptora para recibir un paquete de red que incluye una dirección de destino de paquete y una porción de flujo de bits con datos de video codificados; un analizador que incluye el aparato para analizar un flujo de bits de una secuencia de video codificada según cualquiera de las realizaciones anteriores y posteriores citadas, para determinar la dependencia de los datos de video codificados de otros paquetes; y un analizador de paquetes para analizar la dirección de destino de paquete recibida y la dependencia determinada y para juzgar cómo manejar el paquete de red.

45

50

(Realización 2)

Según la realización 2, `dependent_slice_enabled_flag` se tira del PPS. Debe observarse que `dependent_slice_enabled_flag` puede moverse al SPS, en vez de tirarla.

La figura 11 muestra un ejemplo en el cual la `dependent_slice_enabled_flag` no necesita analizarse antes de acceder a `first_slice_in_pic_flag` y `dependent_slice_flag`.

55

En este ejemplo, `dependent_slice_enabled_flag` no se usa porque no está condicionada en la presencia de la indicación de dependencia. Este ejemplo proporciona la posibilidad de tener la indicación de dependencia al inicio de

la cabecera de porción sin causar problemas de análisis debidos a una identificación no conocida del conjunto del PPS actual.

(Efecto de la realización 2, etc.)

5 En la realización 1, para analizar la `dependent_slice_flag`, debe analizarse la `dependent_slice_enabled_flag`. La `dependent_slice_enabled_flag` se señala en un PPS. Esto puede causar alguna sobrecarga de análisis tal como se comentó anteriormente, cuando la `dependent_slice_enabled_flag` se ubica lejos del inicio del PPS y los elementos de sintaxis precedentes se codifican condicionalmente.

10 Además, señalar el elemento de sintaxis `dependent_slice_flag` antes de que el elemento de sintaxis `id_conjunto_parámetros_img` en el PPS se analice, puede crear errores de análisis tal como sigue. La presencia de `dependent_slice_flag` depende de la `dependent_slice_enabled_flag` que está señalada en el PPS. Sin embargo, la identificación del PPS activo actualmente se señala tras la `dependent_slice_flag`. Por lo tanto, no es posible analizar la `dependent_slice_flag` antes de acceder a los elementos previos.

15 Por consiguiente, es ventajoso eliminar la condición de análisis en la `dependent_slice_enabled_flag`. Puede ser más beneficioso, cuando se aplica la siguiente restricción. A saber, si `dependent_slice_enabled_flag` en PPS es cero, entonces `dependent_slice_flag` deberá ser igual a cero.

Sin embargo, estas implementaciones ventajosas no limitan el alcance de la presente invención.

(Modificación 1 de las realizaciones 1 y 2)

Como una alternativa o adicionalmente para eliminar el condicionamiento en la `dependent_slice_enabled_flag`, la `dependent_slice_enabled_flag` puede moverse desde el PPS a tanto el SPS o/como al VPS.

20 Además, en vez de sólo mover la `dependent_slice_enabled_flag`, la `dependent_slice_enabled_flag` puede duplicarse en el SPS. En este caso, el indicador en el SPS y PPS podría forzarse a tener el mismo valor. O, al PPS se le podría permitir sobrescribir el indicador en el SPS.

25 Por ejemplo, cuando la `dependent_slice_enabled_flag_sps` es igual a 1, entonces la `dependent_slice_enabled_flag_pps` puede ser 0 ó 1. Entonces, `dependent_slice_enabled_flag_sps` es un indicador para habilitar las porciones dependientes para una secuencia de imágenes señaladas en el SPS, y `dependent_slice_enabled_flag_pps` es un indicador para habilitar las porciones dependientes para una imagen señalada en el PPS. Sin embargo, cuando el valor de `dependent_slice_enabled_flag` puede cambiar en el PPS, esto significa que el análisis del PPS todavía se necesita y se evita la ventaja de un seguimiento y análisis menos frecuente del PPS se evita.

30 Estas modificaciones proporcionan la ventaja de que el VPS y el SPS pueden llevar estructuras de dependencia. Llevar las estructuras de dependencia por el VSP y el SPS habilita a los elementos de red a dar forma a los flujos de bits, a saber, determinar descartar los paquetes dependientes que no pueden decodificarse de ninguna manera o para descartar las porciones dependientes más que las porciones independientes. Por lo tanto, la `dependent_slice_enabled_flag` en VPS desencadenaría que el enrutador comprobara la cabecera de porción adicionalmente o no.

35 Se observa que estas modificaciones no reducen más la complejidad del análisis si se aplica el ejemplo de las figuras 10 y 11. Sin embargo, proporciona una estructura más beneficiosa de la sintaxis para llevar las estructuras de dependencia. Resumiendo, según este ejemplo, un indicador para indicar si las porciones dependientes se habilitan para el flujo de bits se señala en un conjunto de parámetros de video. El conjunto de parámetros de video es un conjunto de parámetros que aplica a más de una porción en más de una imagen.

40 Hay dos ventajas diferentes de señalar `dependent_slice_enabled_flag` en el VPS y/o el SPS. Cuando la bandera sólo se mueve o se duplica, no se requiere analizar el PPS, reduciendo la sobrecarga de análisis. El otro beneficio es dejar que los enrutadores conozcan sobre la estructura de predicción de la secuencia de video. Esta ventaja está presente todo el tiempo. Generalmente, un enrutador puede comprobar el contenido de un VPS/SPS para conocer que recibirá.

45 El VPS es el parámetro más alto en la jerarquía. El VPS puede incluir información sobre múltiples secuencias de video, mientras que el SPS y el PPS son específicos para una única secuencia de video y una imagen, respectivamente. La información en el VPS incluye una tasa de bit, estructura temporal de capa de las secuencias de video, y similares. También incluye información sobre las dependencias entre capas (dependencias entre diferentes secuencias de video). Por consiguiente, VPS puede verse como un contenedor para múltiples secuencias de video, y da una visión de conjunto general sobre cada secuencia.

50 En la versión de la HEVC actual, la dependencia entre porciones en una trama se establece tanto por `dependent_slice_flag` como `first_slice_in_pic_flag`. Según las especificaciones actuales, las entidades de red no pueden usar dependencias entre porciones sin aplicar un análisis altamente complejo. Una solución sencilla sería, si

se ha descubierto una pérdida de paquete a través de un número de paquete que falta, tirar todos los paquetes hasta que se encuentre la `first_slice_in_pic_flag` con valor igual a 1. Esto es porque la primera porción en una imagen es siempre una porción normal.

5 Sin embargo, esta solución lleva a reducir la eficiencia de codificación. Por lo tanto, tal como se describió anteriormente puede usarse la señalización de dependencia entre porciones que habilita un análisis eficiente. Esto se consigue señalando `dependent_slice_flag` y `first_slice_in_pic_flag` en la cabecera de porción inmediatamente después de la cabecera NAL.

10 Alternativamente o además, los elementos de sintaxis relativos a las dependencias entre porciones se codifican incondicionalmente, a saber, independientemente de los otros elementos de sintaxis que pueden estar en la cabecera de porción o en el PPS.

(Modificación 2 de las realizaciones 1 y 2)

15 La figura 12 ilustra una modificación 2 alternativa a la modificación 1 discutida anteriormente. En particular, la cabecera 1210 de la unidad de la NAL es la misma que la cabecera de la unidad de la NAL mostrada en la figura 10 (cabecera 910 de la unidad de la NAL mostrada en la figura 9A). Sin embargo, la cabecera 1220 de porción y la cabecera de porción 1020 mostradas en la figura 10 son diferentes en que los elementos de sintaxis de la cabecera de porción `dependent_slice_flag` y `first_slice_in_pic_flag` se invierten en su orden. En particular, la cabecera 1220 de porción incluye la `dependent_slice_flag` como un primer elemento de sintaxis, y el elemento de sintaxis `first_slice_in_pic_flag` como un segundo elemento de sintaxis, condicionado en la presencia de `dependent_slice_flag`.

20 Como se puede ver de este ejemplo, se incluye en la sintaxis un primer indicador de porción que indica si la porción es o no una primera porción en una imagen. Una primera porción en una imagen es siempre una porción para la cual la decodificación de longitud variable no depende de otras porciones. Además, la bandera de porción dependiente se incluye en el flujo de bits delante del indicador de primera porción. El indicador de primera porción se incluye en el flujo de bits solo cuando la bandera de porción dependiente no indica una porción dependiente. Esta disposición proporciona las mismas ventajas que el condicionamiento. En otras palabras, la bandera de dependencia está condicionada en la indicación de primera porción. Como se puede ver en la figura 12, ambos elementos pueden entenderse como la indicación de dependencia y están incluidos al inicio de la cabecera de porción.

(Realización 3)

30 En la realización 3, comparada con las realizaciones 1 y 2, el método de disposición de los elementos de sintaxis se cambia para reducir el análisis de elementos de sintaxis innecesarios.

35 En las realizaciones descritas anteriormente, `dependent_slice_flag` se describe en el caso en el que `first_slice_in_pic_flag` está incluida como condición para la presencia de `dependent_slice_flag`. Sin embargo, el `first_slice_in_pic_flag` y la `dependent_slice_flag` pueden ser ambas incluidas en el flujo de bits sin estar condicionadas una con la presencia de la otra. Por ejemplo, el método de codificación de la `dependent_slice_flag` se cambia para ser independiente del elemento de sintaxis `dependent_slice_enabled_flag` según una de las modificaciones descritas anteriormente.

La figura 13 es un diagrama que muestra un ejemplo de una cabecera de porción según la presente realización. La figura 13 ilustra el caso que todavía incluye la condición de la indicación de dependencia en la bandera que habilita una porción dependiente.

40 Específicamente, en la cabecera de porción según la presente invención, la `dependent_slice_flag` está dispuesta antes de la `slice_address` comparada con la cabecera de porción existente mostrada en la figura 6. Además, en la cabecera de porción según la presente invención, comparada con los ejemplos en las figuras 10 a 12, la `dependent_slice_flag` está dispuesta detrás de `id_conjunto_parámetros_img`.

45 En la presente realización, dado que la `dependent_slice_flag` está dispuesta antes de la `slice_address`, al menos el SPS no necesita ser analizado para el análisis de `dependent_slice_flag`. Como se describió anteriormente, la `slice_address` es un elemento de sintaxis que indica el inicio de una porción. Además, la `slice_address` solo se puede analizar con la ayuda de los elementos de sintaxis señalados en el SPS (`id_conjunto_parámetros_img`).

50 Alternativamente o además, la `dependent_slice_enabled_flag` bien se sube en el PPS o se mueve al SPS y/o al VPS. Si la bandera habilitada está en el VPS y/o en el SPS, puede que no se requiera analizar y hacer seguimiento del PPS y del SPS.

(Modificación de la realización 3, efecto, y similares)

(1) El ejemplo de la figura 13 puede llevar a proporcionar un aparato para analizar un flujo de bits de una secuencia de video codificada al menos parcialmente con un código de longitud variable e incluir unidades de datos que llevan porciones codificadas de imágenes de video. En este caso, el aparato se configura para incluir un analizador que

extrae del flujo de bits los siguientes elementos de sintaxis:

una indicación de dependencia que es un elemento de sintaxis que indica para una porción en la cabecera de porción si la decodificación de longitud variable de la porción depende o no de otras porciones;

5 una bandera que habilita una porción dependiente que incluida en un conjunto de parámetros para una pluralidad de porciones y que indica si las porciones dependientes se pueden incluir o no en el flujo de bits; y

una dirección de porción que indica la posición en el flujo de bits en la cual la porción empieza.

(2) Además, en la presente realización, la indicación de dependencia está señalada en la cabecera de porción antes de la dirección de porción y tras el elemento de sintaxis que identifica el conjunto de parámetros.

10 Con esta realización, es posible configurar sin causar errores de análisis que la indicación de dependencia sea incluida en el flujo de bits solo cuando la bandera de porción de dependencia habilitada indique que las porciones dependientes se pueden incluir en el flujo de bits.

(3) En la presente realización, la bandera que habilita una porción dependiente está ubicada en el flujo de bits en un conjunto de parámetros (PPS) común para una pluralidad de porciones que forman la misma trama de imagen y está ubicado al inicio del conjunto de parámetros. Sin embargo, no está limitado a eso.

15 Alternativamente (o además), la bandera que habilita una porción dependiente está ubicada en el flujo de bits en un conjunto de parámetros (SPS) común para una pluralidad de porciones que forman la misma secuencia de imágenes. Todavía alternativamente (o además), la bandera que habilita una porción dependiente está ubicada en el flujo de bits en un conjunto de parámetros (VPS) común para una pluralidad de porciones que forman una pluralidad de secuencias de tramas de imagen.

20 (4) Además, en la presente invención, el `id_VPS` y el `id_SPS` pueden estar señalados explícitamente en un mensaje SEI. Cuando la `dependent_slice_enabled_flag` está señalada en el SPS, la `dependent_slice_flag` debe todavía seguir al `id_conjunto_parámetros_img`.

25 De otra manera, la dependencia de análisis se introduce porque el `id_SPS` está señalado en el PPS. Con la señalización de la identificación del SPS o VPS actual que lleva la `dependent_slice_enabled_flag`, la indicación de dependencia puede incluirse también antes del `id_conjunto_parámetros_img` dado que entonces el análisis del conjunto de parámetros de la imagen no es necesario. Además, tal mensaje SEI, que lleva el `id_VPS` o el `id_SPS` no es necesario para la operación de decodificación dado que estos ID también se determinan mediante el análisis del PPS. El mensaje SEI puede así ser descartado sin afectar a la operación de decodificación tras ser usado por los elementos de red.

30 (Realización 4)

En la realización 4, la información de dependencia entre porciones se duplica (de manera suplementaria a la información señalada en la cabecera de porción y/o en un conjunto de parámetros) en otra unidad de la NAL tal como un mensaje SEI.

35 Por ejemplo, puede definirse un mensaje SEI que transporta la información de dependencia entre porciones en cada unidad de acceso o antes de cada porción dependiente. El término “unidad de acceso” se refiere a una unidad de datos que está hecha de un conjunto de unidades de NAL. Una unidad de acceso incluye porciones de imágenes codificadas, esto es, NALU de la VCL. En particular, las unidades de acceso pueden definir puntos para acceso aleatorio y pueden incluir NALU de una única imagen. Sin embargo, la unidad de acceso no es necesariamente un punto de acceso aleatorio.

40 En las especificaciones de HEVC actuales, la unidad de acceso se define como un conjunto de unidades de NAL que son consecutivas en orden de decodificación y contienen exactamente una imagen codificada. Además de las unidades de NAL de porción codificada de la imagen codificada, la unidad de acceso puede también contener otras unidades de NAL que no contienen porciones de la imagen codificada. La decodificación de una unidad de acceso siempre resulta en una imagen decodificada. Sin embargo, en una extensión futura de la HEVC (como codificación multivista, (MVC) o codificación de video escalable, (SVC)), la definición de unidad de acceso puede relajarse o modificarse. Según las especificaciones actuales, la unidad de acceso está formada por un delimitador de unidad de acceso, mensajes SEI, y NALU de la VCL.

45 Según la presente invención, la indicación de dependencia está ubicada en el flujo de bits fuera de la cabecera de una porción a la cual se refiere la indicación de dependencia. Además, puede ser beneficioso cuando la indicación de dependencia está ubicada en el flujo de bits en un mensaje de información mejorada suplementaria incluido en el flujo de bits antes de la porción de dependencia o una vez por unidad de acceso.

(Realización 5)

Según la realización 5, la información de dependencia entre porciones se señala en la cabecera NAL como una

bandera o implícitamente como un tipo de la unidad de la NAL con la cual está asociado.

Por lo general, el análisis de elementos de sintaxis en la cabecera NAL no depende de otros elementos de sintaxis. Cada cabecera de la unidad de la NAL se puede analizar de manera independiente. La cabecera NAL es el lugar habitual para señalar la información de dependencia. En consecuencia, según la presente realización, también la

5 dependencia entre porciones se señala en su interior.

En otras palabras, el aparato de análisis puede adoptarse en un enrutador o en un decodificador. El aparato de análisis además incluye una unidad de capa de adaptación de red para añadir a una porción de datos de video codificado y a la cabecera de la porción una capa de adaptación de red, y una cabecera NAL. Ventajosamente, la indicación de dependencia está ubicada en el flujo de bits en la cabecera NAL y está codificada independientemente

10 de los otros elementos de sintaxis.

La indicación de dependencia puede colocarse en la cabecera NAL dado que la cabecera NAL en las especificaciones de HEVC actuales prevén algunos bits reservados que se pueden usar para eso. Un único bit sería suficiente para señalar la indicación de dependencia.

Alternativamente, la indicación de dependencia está indicada por un tipo de la unidad de la NAL y un tipo de la

15 unidad de la NAL predefinido se reserva para llevar información de dependencia.

(Realización 6)

Se nota que las cinco realizaciones anteriores se pueden combinar arbitrariamente para habilitar un análisis eficiente de la información dependiente en los elementos de red. Aun cuando su uso sea redundante, las realizaciones son combinables. En consecuencia, la duplicación de la indicación de dependencia se puede aplicar aun cuando la

20 indicación de dependencia también esté señalada al inicio de la cabecera de porción.

La figura 14 muestra un ejemplo de una cabecera 1410 de la unidad de la NAL en el cual se modifica la cabecera 910 de la unidad de la NAL mostrada en la figura 9A. La cabecera 1410 de la unidad de la NAL incluye `dependent_slice_flag`.

Además, para mover la `dependent_slice_flag` en la cabecera NAL y para mantener el tamaño de la cabecera NAL fijado debido a la compatibilidad hacia atrás, el bit necesario para la `dependent_slice_flag` se toma del elemento de sintaxis `nuh_reserved_zero_6bits` de la cabecera de la unidad de la NAL. En consecuencia, el elemento de sintaxis `nuh_reserved_zero_6bits` ahora tiene solo 5 bits. El elemento de sintaxis `nuh_reserved_zero_6bits` incluye bits reservados para usos futuros de forma que la reducción no causa ningún problema y no requiere de más modificaciones.

25

En general, una unidad de la NAL de la VCL depende de la unidad de la NAL de la VCL anterior que tiene el mismo `id_capa_temporal`. Cuando la `dependent_slice_flag` se señala en la cabecera NAL, un bit será malgastado tanto por las unidades de NAL de la VCL como las no-VCL dado que cada unidad de datos tal como una porción de imagen o conjunto de parámetros tienen la misma cabecera NAL. En consecuencia, aunque parece que la `dependent_slice_flag` también sería señalada para conjuntos de parámetros o para mensajes SEI, esto es innecesario. Además, `dependent_slice_flag` siempre necesita ser señalada aun si las porciones dependientes están deshabilitadas en el conjunto de parámetros de secuencia. Esto lleva a una sobrecarga innecesaria.

30

En todas las realizaciones anteriores, la indicación de dependencia puede ser una bandera de un bit.

(Realización 7)

Según la realización 7, la indicación de dependencia se indica por un tipo de la unidad de la NAL y un tipo de la

40 unidad de la NAL predefinido se reserva para llevar información de dependencia.

En consecuencia, un nuevo (separado) tipo de la NAL de la VCL se define con una semiótica similar a las unidades de NAL y la VCL existentes. Por ejemplo, cuando `tipo_unidad_NAL` es igual a 15 (o a otro tipo predefinido o NALU que no está reservado para otro tipo de NALU particular), entonces la unidad de la NAL de la VCL actual depende de la unidad de la NAL de la VCL anterior que tiene el mismo `id_capa_temporal`. La dependencia se refiere a la dependencia de la porción actual en la cabecera de porción de una porción precedente, como se describió anteriormente, esto es, dependencia en análisis.

45

Puede ser ventajoso en estos casos incluir el bit en la cabecera NAL para los tipos de unidades de NAL adicionales. Esto se puede usar para indicar si la porción actual es o no una porción dependiente.

Cuando la información de dependencia está señalada en la cabecera de porción además de la cabecera NAL, la señalización en la cabecera NAL se vuelve opcional. Específicamente, cuando el campo `tipo_unidad_NAL` en la cabecera NAL se configura para señalar que la porción actual es una porción dependiente, entonces no es posible señalar cualquier otra información de "tipo". Por ejemplo, en algunos casos podría ser más beneficioso transportar la información de que una porción actual es una "primera imagen en la secuencia" (`tipo_unidad_NAL` igual a 10 o 11). Cuando la información de dependencia entre porciones en la cabecera NAL es opcional (dado que está duplicada en

50

la cabecera de porción), se puede elegir señalar la información más valiosa.

Puede ser además ventajoso añadir dos o más tipos de unidades de NAL de la VCL, tales como "imagen RAP de porción dependiente" (requerida para análisis) o "imagen no RAP de porción dependiente". "RAP" denota la imagen de acceso aleatorio. Imagen de acceso aleatorio es una imagen codificada independientemente (en términos de predicción) de otras imágenes de forma que tal imagen puede usarse como un punto de inicio para codificar y decodificar. Con esto, es así adecuada como punto de acceso aleatorio.

En la cabecera de porción dependiente, el elemento de sintaxis `BanderalmgRap` se usa en el proceso de análisis. Específicamente, el elemento de sintaxis `BanderalmgRap` es un indicador que indica si la imagen actual es o no una imagen de acceso aleatorio.

El valor de `BanderalmgRap` depende del tipo de unidad de la NAL como la siguiente expresión 2.

[Matemáticas 2]

$$\text{BanderalmgRap} = (\text{tipo\_unidad\_nal} \geq 7 \ \&\& \ \text{tipo\_unidad\_nal} \leq 12)$$

(Expresión 2)

En otras palabras, en el ejemplo mostrado en la figura 15, las imágenes de acceso aleatorio son llevadas por la NALU con tipo de la NALU entre 7 y 12. Para habilitar un análisis correcto y para proporcionar una posibilidad de dependencia de porciones para las imágenes de acceso aleatorio, por lo tanto, en la presente invención, se definen dos tipos de unidades de NAL para garantizar el análisis correcto de la cabecera de porción.

Por lo general, aun cuando se define un nuevo tipo de unidad de la NAL de la VCL, el análisis de la cabecera de porción debería todavía ser posible sin problemas. Cualquiera de los múltiples tipos de NAL se define como anteriormente o la cabecera de porción dependiente se carga de tal modo que no hay problema de análisis.

Cuando un nuevo tipo de unidad de la NAL de la VCL se define para indicar la porción dependiente, la estructura de sintaxis de cabecera de porción se puede cambiar como sigue.

En el ejemplo anterior el tipo de unidad de la NAL "NUT\_DS" se usa para indicar que la unidad de la NAL de la VCL actual es una porción dependiente. Comparado con la estructura de sintaxis de la cabecera de porción del estado de la técnica que se describe en la bibliografía no de patente 3, los dos cambios siguientes se introducen en la presente invención.

(1) `bandera_no_salida_de_img_anterior` no está señalada en la cabecera de porción dependiente. En otras palabras la presencia de `bandera_no_salida_de_img_anterior` está basada en la condición de que la porción actual no es una porción dependiente. (`bandera_no_salida_de_img_anterior` se puede presentar en la cabecera de porción cuando la porción actual no es una porción dependiente).

(2) `first_slice_in_pic_flag` se señala condicionalmente en el valor de `tipo_unidad_nal`. Cuando el valor de `tipo_unidad_nal` indica que la porción actual es una porción dependiente, el elemento de sintaxis `first_slice_in_pic_flag` no se señala explícitamente y se concluye que es 0. Esto ahorra tasa de bits en la misma calidad.

Según el ejemplo, `bandera_no_salida_de_img_anterior` no se señala cuando la porción actual es una porción dependiente. Según el valor de `BanderalmgRap` no se requiere evaluar cuando la porción actual es una porción dependiente. Por lo tanto la cabecera de porción de una porción dependiente se puede analizar sin problemas. Más específicamente, la cabecera de porción de la porción dependiente se puede analizar sin hacer referencia a la cabecera de la unidad de la NAL de una cabecera de unidad precedente. Un problema ocurre cuando la unidad de la NAL precedente no está presente en el momento de decodificar.

En segundo lugar, la `first_slice_in_pic_flag` se señala basándose en el valor del `tipo_unidad_NAL`. Este cambio es el mismo que en el ejemplo descrito en la figura 12. En la figura 12, `first_slice_in_pic_flag` se señala en la cabecera de porción solo cuando la porción actual no es una porción dependiente (que se indica por la `dependent_slice_flag`). De manera similar en el ejemplo anterior `first_slice_in_pic_flag` se señala solo cuando el `tipo_unidad_nal` no es igual a "NUT\_DS", que significa que la porción actual no es una porción dependiente.

No se requiere que los dos cambios que se han presentado anteriormente se hagan juntos. También es posible realizar solo uno de los cambios en la cabecera de porción. El beneficio de cada cambio está asociado con el coste de comprobar si una porción es o no una porción dependiente. Sin embargo, cuando los dos cambios se realizan juntos, los beneficios de ambos cambios pueden venir tanto por los mismos costes como el beneficio de cada uno de los cambios individuales en el caso en el que los dos elementos de sintaxis `first_slice_in_pic_flag` y `bandera_no_salida_de_img_anterior` se codifican consecutivamente. Así, la aplicación de ambos cambios en combinación con una codificación consecutiva de los dos elementos de sintaxis mencionados da una ventaja sobre la aplicación directa de cada uno de los cambios individualmente.



En toda la explicación en las realizaciones, también es posible eliminar la `dependent_slice_enabled_flag` del flujo de bits cuando el indicador de porción dependiente no está codificado condicionalmente en él. En otras palabras, cuando por ejemplo un nuevo tipo de unidad de la NAL se usa para indicar que la porción actual es una porción dependiente, entonces la `dependent_slice_enabled_flag` se puede eliminar del flujo de bits.

- 5 La figura 15 muestra una cabecera 1510 de la unidad de la NAL que es la misma que la cabecera 910 de la unidad de la NAL mostrada en la figura 9A y una cabecera de porción 1520 que se cambia de la cabecera de porción 920 mostrada en la figura 9A. La cabecera de porción 1520 incluye la terminación del valor de `dependent_slice_flag` según el tipo de la NALU. En particular, el elemento de sintaxis `tipo_unidad_NAL` con valores 15 y 16 define porciones dependientes. Cuando `tipo_unidad_NAL` es igual a 15, el tipo de la porción es una porción dependiente de imagen de acceso aleatorio. Si, por otra parte, `tipo_unidad_NAL` es igual a 16, la porción es una porción dependiente de una imagen de acceso no aleatorio. Por lo tanto, se establece una relación de la siguiente expresión 3.

[Matemáticas 3]

$$\text{BanderaImgRap} = (\text{tipo\_unidad\_nal} \geq 7 \ \&\& \ \text{tipo\_unidad\_nal} \leq 12 \ || \ \text{tipo\_unidad\_nal} == 15)$$

(Expresión 3)

- 15 Obsérvese que los valores 15 y 16 se seleccionaron solo como un ejemplo. Como será claro para los expertos en la técnica, se puede adoptar cualquier número predefinido que no fuera a ser usado de otro modo. Específicamente, un primer tipo de la NALU se define para identificar el contenido de porción dependiente de una imagen de acceso aleatorio, y un segundo tipo de una NALU se define para identificar el contenido de porción dependiente de una imagen de acceso no aleatorio.

- 20 Además, se puede aplicar una restricción que las porciones dependientes solo se usan para RAP o solo se usan para no-RAP. En tales casos, solo es necesario un nuevo tipo de NALU.

(Realización 8)

- 25 La figura 16 es un diagrama que muestra una solución alternativa. Una cabecera de la unidad de la NAL 1610 es la misma que la cabecera 910 de la unidad de la NAL. La cabecera de porción 1620 asume la definición de `tipo_unidad_NAL` con los valores 15 y 16 de porciones dependientes señaladas como se describió anteriormente.

Sin embargo, el tipo de unidad de la NAL no se usa en el análisis de la bandera de porción dependiente. Esto habilita el uso del `tipo_unidad_NAL` de forma opcional para el codificador. En consecuencia, la ventaja de la presente realización solo se alcanza cuando el codificador está determinado para adoptar los nuevos tipos de las NALU.

- 30 Entonces, el enrutador solo necesita mirar en el tipo de la NALU. Sin embargo, cuando el codificador no usa los nuevos tipos de las NALU, el enrutador trataría las porciones dependientes como en el estado de la técnica.

- Resumiendo, la indicación de dependencia puede indicarse por un tipo de la unidad de la NAL. Un tipo de la unidad de la NAL predefinido puede reservarse para llevar porciones codificadas cuya cabecera de porción depende de la cabecera de porción de una porción precedente. Ventajosamente, se proporciona un tipo de la unidad de la NAL separado que indica la dependencia para imágenes de acceso aleatorio y para imágenes de acceso no aleatorio.

- 35 Resumiendo, las realizaciones descritas anteriormente se refieren a la sintaxis de un flujo de bits que lleva secuencias de video codificadas. En particular, las realizaciones descritas anteriormente se refieren la sintaxis referida a las porciones dependientes y de entropía, de las cuales la cabecera de porción depende de la cabecera de porción de una porción precedente. Para permitir un elemento de red consciente del medio para considerar este tipo de dependencia sin aumentar esencialmente su complejidad y retardo debido al análisis, la indicación de dependencia se señala al inicio de los paquetes o en otras palabras en la proximidad de las cabeceras o parámetros que van a ser analizados. Esto se logra por ejemplo, mediante la inclusión de la indicación de dependencia al inicio de la cabecera de porción (figuras 10 a 12), posiblemente tras el identificador del conjunto de parámetros y antes de la dirección de porción, o mediante la inclusión de la indicación de dependencia antes de la dirección de porción (figuras 10 y 11), o proporcionando la indicación de dependencia en una cabecera NAL (figura 14), en un mensaje separado o por un tipo de la NALU especial para NALU que llevan porciones dependientes (figuras 15 y 16).

(Modificaciones de las realizaciones 1 a 8, efecto, y similares)

Son posibles varios cambios sin estar limitados a las realizaciones 1 a 8, y obviamente se incluyen en el alcance de la presente invención.

- 50 Cada uno de los elementos estructurales en cada una de las realizaciones descritas anteriormente puede configurarse en la forma de un producto de hardware exclusivo (circuito de procesamiento), o se puede realizar mediante la ejecución de un programa de software adecuado para el elemento estructural. Cada uno de los elementos estructurales puede ser realizado por medio de una unidad de ejecución de programa, tal como una CPU y un procesador, que leen y ejecutan el programa de software grabado en un medio de grabación tal como un disco

duro o una memoria semiconductora

Aunque en las realizaciones 1 a 8 la descripción asume frentes de ondas, no está limitado a ellos.

5 Sin embargo, en el caso de frentes de ondas, todos los subflujos no pueden iniciarse al mismo tiempo. Como se describió anteriormente, con respecto a cada uno de los subflujos excepto del subflujo al inicio, el inicio del procesamiento (codificación o decodificación) se retrasa por dos LCU del subflujo precedente. Por lo tanto, en un frente de onda, se requiere mayor acortamiento del procesamiento. En la presente realización, mediante la ubicación de la indicación de dependencia (dependent\_slice\_flag) tras la sintaxis que identifica el PPS y antes de la dirección de porción, el número de elementos de sintaxis a ser analizados se puede reducir y así el procesamiento se reduce.

10 Además, en las realizaciones 1 a 8 anteriormente descritas, mediante la disposición de la indicación de dependencia más arriba en la cabecera de porción (notablemente al inicio), es posible, por ejemplo, comprobar si cada una de las porciones es una porción dependiente o no en una etapa temprana del procesamiento de la imagen.

15 En otras palabras, en el momento del inicio de procesamiento en una imagen (codificación o decodificación), cuando un paso de comprobación si cada una de las porciones es una porción dependiente o no, es posible extraer un punto de inicio del procesamiento paralelo en el momento del inicio del procesamiento de la imagen. En otras palabras, cuando la imagen incluye una pluralidad de porciones normales, es posible extraer un punto de inicio del procesamiento paralelo en el momento del procesamiento en una imagen o en una etapa temprana del procesamiento.

20 Aquí, convencionalmente, cuando la indicación de dependencia se dispone tras la dirección de porción, no es posible comprobar si la porción es una porción dependiente o una porción normal hasta que se completa el análisis de la dirección de porción. En este caso, el inicio del procesamiento en la porción normal en el medio de la imagen se retrasa significativamente del inicio del procesamiento en la porción normal al inicio de la imagen.

25 A la inversa, en las realizaciones 1 a 8 anteriormente descritas, dado que es posible comprobar si cada una de las porciones es una porción dependiente o no en una etapa temprana del procesamiento en una imagen, es posible facilitar el inicio del procesamiento en una porción normal en el medio de la imagen. En otras palabras, es posible iniciar el procesamiento de la porción normal en el medio de una imagen al mismo tiempo que la porción normal al inicio de la imagen.

(Realización 9)

30 El procesamiento descrito en cada una de las realizaciones se puede implementar simplemente en un sistema de ordenador independiente, mediante la grabación, en un medio de grabación, de un programa para implementar las configuraciones del método de codificación de imágenes en movimiento (método de codificación de imagen) y el método de decodificación de imágenes en movimiento (método de decodificación de imagen) descritos en cada una de las realizaciones. El medio de grabación puede ser cualquier medio de grabación mientras el programa se pueda grabar, tal como un disco magnético, un disco óptico, un disco óptico magnético, una tarjeta IC y una memoria semiconductora.

35 A continuación en el presente documento, las aplicaciones al método de codificación de imágenes en movimiento (método de codificación de imagen) y el método de decodificación de imágenes en movimiento (método de decodificación de imagen) descritos en cada una de las realizaciones y sistemas usando los mismos serán descritos. El sistema tiene una característica de tener un aparato de codificación y decodificación de imagen que incluye un aparato de codificación de imagen que usa el método de codificación de imagen y un aparato de decodificación de imagen que usa el método de decodificación de imagen. Se pueden cambiar otras configuraciones en el sistema según sea apropiado dependiendo de los casos.

45 La figura 17 ilustra una configuración general de un sistema ex100 que proporciona contenido para implementar servicios de distribución de contenidos. El área para proporcionar servicios de comunicación está dividida en celdas del tamaño deseado, y las estaciones base ex106, ex107, ex108, ex109, y ex110 que son estaciones inalámbricas fijas son emplazadas en cada celda.

El sistema ex100 que proporciona contenido está conectado a dispositivos, tales como un ordenador ex111, un asistente digital personal (PDA) ex112, una cámara ex113, un teléfono móvil ex114 y una consola de videojuegos ex115, a través de Internet ex101, un proveedor de servicios de Internet ex102, una red telefónica ex104, así como las estaciones base ex106 a ex110 respectivamente.

50 Sin embargo, la configuración del sistema ex100 que proporciona contenido no está limitada a la configuración mostrada en la figura 17, y una combinación en la cual cualquiera de los elementos está conectado es aceptable. Además, cada dispositivo puede estar directamente conectado con la red telefónica ex104, más que a través de las estaciones base ex106 a ex110, que son las estaciones inalámbricas fijas. Además, los dispositivos pueden estar interconectados entre ellos a través de una comunicación inalámbrica de corta distancia y otros.

55 La cámara ex113, tal como una video cámara digital, es capaz de capturar video. Una cámara ex116, tal como una

video cámara digital, es capaz de capturar tanto imágenes fijas como video. Además, el teléfono móvil ex114 puede ser el que cumpla cualquiera de los estándares como sistema global para comunicaciones móviles (GSM), acceso múltiple por división de código (CDMA), acceso múltiple por división de código de banda ancha (W-CDMA), evolución a largo plazo (LTE) y acceso de paquetes de alta velocidad (HSPA). Alternativamente, el teléfono móvil ex114 puede ser un sistema de teléfono útil personal (PHS).

En el sistema ex100 que proporciona contenido, un servidor de flujo de transmisión ex103 está conectado a la cámara ex113 y otros a través de la red telefónica ex104 y la estación base ex109, que permiten la distribución de imágenes en un espectáculo en vivo y otros. En tal distribución, un contenido (por ejemplo, video de un espectáculo de música en vivo) capturado por el usuario usando la cámara ex113 es codificado como se describió anteriormente en cada una de las realizaciones (esto es, las funciones de la cámara como el aparato de codificación de imágenes de la presente invención), y el contenido codificado es transmitido al servidor de flujo de transmisión ex103. Por otro lado, el servidor de flujo de transmisión ex103 lleva a cabo distribución de flujos de los datos contenidos transmitidos a los clientes a su solicitud. Los clientes incluyen el ordenador ex111, el PDA ex112, la cámara ex113, el teléfono móvil ex114, y la consola de videojuegos ex115 que son capaces de decodificar los datos codificados mencionados anteriormente. Cada uno de los dispositivos que ha recibido los datos distribuidos decodifica y reproduce los datos codificados (esto es, funcionan como el aparato de decodificación de imágenes según un aspecto de la presente invención).

Los datos capturados pueden ser codificados por la cámara ex113 o el servidor de flujo de transmisión ex103 que transmite los datos, o los procesos de codificación pueden compartirse entre la cámara ex113 y el servidor de flujo de transmisión ex103. Similarmente, los datos distribuidos pueden ser decodificados por los clientes o el servidor de flujo de transmisión ex103, o los procesos de decodificación pueden compartirse entre los clientes y el servidor de flujo de transmisión ex103. Además, los datos de las imágenes fijas y el video capturado por no solo la cámara ex113 sino también por la cámara ex116 pueden ser transmitidos al servidor de flujo de transmisión ex103 a través del ordenador ex111. Los procesos de codificación pueden realizarse mediante la cámara ex116, el ordenador ex111, o el servidor de flujo de transmisión ex103, o compartidos entre ellos.

Además, los procesos de codificación y decodificación pueden ser realizados por un LSI ex500 generalmente incluido en cada ordenador ex111 y los dispositivos. El LSI ex500 puede estar configurado de un único chip o una pluralidad de chips. El software para codificar y decodificar el video puede estar integrado en algún tipo de medio de grabación (tal como un CD-ROM, un disco flexible, y un disco duro) que es legible por el ordenador ex111 y otros, y los procesos de codificación y decodificación pueden ser realizados usando el software. Además, cuando el teléfono móvil ex114 está equipado con una cámara, los datos de imágenes obtenidas por la cámara pueden ser transmitidos. Los datos de video son datos codificados por el LSI ex500 incluido en el teléfono móvil ex114.

Además, el servidor de flujo de transmisión ex103 puede estar compuesto de servidores y ordenadores, y puede descentralizar datos y procesar los datos descentralizados, grabar, o distribuir datos.

Como se describió anteriormente, los clientes pueden recibir y reproducir los datos codificados en el sistema ex100 que proporciona contenido. En otras palabras, los clientes pueden recibir y decodificar información transmitida por el usuario, y reproducir y decodificar datos en tiempo real en el sistema ex100 que proporciona contenido, de forma que el usuario que no tienen ningún derecho y un equipo particular puede implementar difusión personal.

Aparte del ejemplo del sistema ex100 que proporciona contenido, al menos un aparato de codificación de imágenes en movimiento (aparato de codificación de imágenes) y el aparato de decodificación de imágenes en movimiento (aparato de decodificación de imágenes) descritos en cada una de las realizaciones puede ser implementado en un sistema de difusión digital ex200 ilustrado en la figura 18. Más específicamente, una estación de difusión ex201 se comunica o transmite, a través de ondas de radio a un satélite de difusión ex202, datos multiplexados obtenidos mediante la multiplexación de datos de audio y otros en datos de video. Los datos de video son datos codificados por el método de codificación de imágenes en movimiento descrito en cada una de las realizaciones (esto es, datos codificados por el aparato de codificación de imágenes de la presente invención). Tras recibir los datos multiplexados, el satélite de difusión ex202 transmite ondas de radio para difusión. Entonces, una antena de uso doméstico ex204 con una función de recepción de difusión de satélite recibe las ondas de radio. A continuación, un dispositivo como un televisor ex300 (receptor) y un decodificador de televisión (STB) ex217 decodifican los datos multiplexados recibidos, y reproducen los datos decodificados (esto es, funciona como el aparato de decodificación de imágenes según un aspecto de la presente invención).

Además, un lector/grabador ex218 (i) lee y decodifica los datos multiplexados grabados en un medio de grabación ex215, tal como un DVD y un BD, o (i) codifica señales de video en el medio de grabación ex215, y en algunos casos, escribe datos obtenidos mediante la multiplexación de una señal de audio en los datos codificados. El lector/grabador ex218 puede incluir el aparato de decodificación de imágenes en movimiento o el aparato de codificación de imágenes en movimiento como se muestra en cada una de las realizaciones. En este caso, las señales de video reproducidas son visualizadas en el monitor ex219, y pueden ser reproducidas por cualquier dispositivo o sistema que use el medio de grabación ex215 en el cual los datos multiplexados están grabados. Es también posible implementar el aparato de decodificación de imágenes en movimiento en el decodificador de televisión ex217 conectado al cable ex203 para televisión por cable o a la antena ex204 para difusión satelital y/o

terrestre, para visualizar las señales de video en el monitor ex219 del televisor ex300. El aparato de decodificación de imágenes en movimiento puede implementarse no solo en el decodificador de televisión sino en el televisor ex300.

5 La figura 19 ilustra el televisor (receptor) ex300 que usa el método de codificación de imágenes en movimiento y el método de decodificación de imágenes en movimiento descrito en cada una de las realizaciones. El televisor ex300 incluye: un sintonizador ex301 que obtiene o proporciona datos multiplexados obtenidos mediante la multiplexación de datos de audio sobre datos de video, a través de la antena ex204 o el cable ex203, etc. que recibe una difusión; una unidad ex302 de modulación/demodulación que demodula los datos multiplexados recibidos o modula datos en datos multiplexados a ser suministrados fuera; y una unidad de multiplexación/demultiplexación ex303 que demultiplexa los datos multiplexados modulados en datos de video y datos de audio, o multiplexa datos de video y datos de audio codificados por una unidad de procesamiento de señal ex306 en datos.

10 El televisor ex300 además incluye: una unidad de procesamiento de señal ex306 que incluye una unidad ex304 de procesamiento de señal de audio y una unidad de procesamiento de señal de video ex305 que decodifica datos de audio y datos de video y codifica datos de audio y datos de video, (que funciona como el aparato de codificación de imágenes y el aparato de decodificación de imágenes según los aspectos de la presente invención), respectivamente; y una unidad de salida ex309 que incluye un altavoz ex307 que proporciona la señal de audio decodificada, y una unidad de visualización ex308 que muestra la señal de video decodificada, tal como un monitor. Además, el televisor ex300 incluye una unidad de interfaz ex317 que incluye una unidad de entrada de operación ex312 que recibe una entrada de una operación de usuario. Además, el televisor ex300 incluye una unidad de control ex310 que controla en general cada elemento constituyente del televisor ex300, y una unidad de circuito de alimentación ex311 que suministra alimentación a cada uno de los elementos. Además de la unidad de entrada de operación ex312, la unidad de interfaz ex317 puede incluir: un puente ex313 que está conectado a un dispositivo externo, tal como el lector/grabador ex218; una unidad de ranura ex314 para habilitar el acoplamiento del medio de grabación ex216 tal como una tarjeta SD; un controlador ex315 a ser conectado a un medio de grabación externo, tal como un disco duro; y un módem ex316 a ser conectado a la red telefónica. Aquí, el medio de grabación ex216 puede grabar eléctricamente información usando un elemento de memoria semiconductor no volátil/volátil de almacenamiento. Los elementos constituyentes del televisor ex300 están conectados entre ellos a través de un bus síncrono.

30 Primero, se describirá la configuración en la cual el televisor ex300 decodifica datos multiplexados obtenidos desde fuera a través de la antena ex204 y otros y reproduce los datos decodificados. En el televisor ex300, tras una operación de usuario a través de un mando a distancia ex220 y otros, la unidad de multiplexación/demultiplexación ex303 demultiplexa los datos multiplexados demodulados por la unidad ex302 de modulación/demodulación, bajo el control de la unidad de control ex310 que incluye una CPU. Además, la unidad ex304 de procesamiento de señal de audio decodifica los datos de audio demultiplexados, y la unidad ex305 de procesamiento de señal de video decodifica los datos de video demultiplexados, usando el método de decodificación descrito en cada una de las realizaciones, en el televisor ex300. La unidad de salida ex309 proporciona la señal de video y la señal de audio decodificadas de salida, respectivamente. Cuando la unidad de salida ex309 proporciona la señal de video y la señal de audio, las señales pueden almacenarse temporalmente en las memorias intermedias (búfers) ex318 y ex319, y otras para que las señales se reproduzcan en sincronía entre ellas. Además, el televisor ex300 puede leer datos multiplexados no a través de una difusión y otros sino desde el medio ex215 y ex216 de grabación, tal como un disco magnético, un disco óptico, y una tarjeta SD. A continuación, se describirá una configuración en la cual el televisor ex300 codifica una señal de audio y una señal de video, y transmite los datos afuera o escribe los datos en un medio de grabación. En el televisor ex300, tras una operación de usuario a través de un mando a distancia ex220 y otros, la unidad ex304 de procesamiento de señal de audio codifica una señal de audio, y la unidad ex305 de procesamiento de señal de video codifica una señal de video, bajo el control de la unidad de control ex310 usando el método de codificación descrito en cada una de las Realizaciones. La unidad ex303 multiplexa la señal de video y la señal de audio codificadas, y proporciona la señal resultante afuera. Cuando la unidad de multiplexación/demultiplexación ex303 multiplexa la señal de video y la señal de audio, las señales pueden almacenarse temporalmente en las memorias intermedias (búfers) ex320 y ex321, y otras para que las señales se reproduzcan en sincronía entre ellas. Aquí, las memorias intermedias ex318, ex319, ex320 y ex321 pueden ser varias como se ilustra, o al menos una memoria intermedia puede ser compartida en el televisor ex300. Además, los datos pueden ser almacenados en una memoria intermedia para que se pueda evitar el exceso de flujo o la falta de flujo entre la unidad ex302 de modulación/demodulación y la unidad de multiplexación/demultiplexación ex303, por ejemplo.

55 Además, el televisor ex300 puede incluir una configuración para recibir una entrada AV desde un micrófono o una cámara además de la configuración para obtener datos de audio y video desde una difusión o un medio de grabación, y puede codificar los datos obtenidos. Aunque el televisor ex300 puede codificar, multiplexar, y proporcionar datos de salida en la descripción, puede ser solo capaz de recibir, decodificar, y proporcionar datos de salida pero no de codificar, multiplexar y proporcionar datos de salida.

60 Además, cuando el lector/grabador ex218 lee o escribe datos multiplexados desde o en un medio de grabación, uno entre el televisor ex300 y el lector/grabador ex218 puede decodificar y codificar los datos multiplexados, y el televisor ex300 y el lector/grabador ex218 pueden compartir la decodificación o codificación.

Como un ejemplo, la figura 20 ilustra una configuración de una unidad de reproducción/grabación de información ex400 cuando los datos se leen o escriben desde o en un disco óptico. La unidad de reproducción/grabación de información ex400 incluye elementos constituyentes ex401, ex402, ex403, ex404, ex405, ex406, y ex407 que serán descritos en adelante. El cabezal óptico ex401 irradia un punto láser en una superficie de grabación del medio de grabación ex215 que es un disco óptico para escribir información, y detecta luz reflejada desde la superficie de grabación en el medio de grabación ex215 para leer la información. La unidad ex402 de grabación de modulación dirige eléctricamente un láser semiconductor incluido en el cabezal óptico ex401, y modula la luz láser según los datos grabados. La unidad de demodulación de reproducción ex403 amplifica una señal de reproducción obtenida mediante la detección eléctrica de la luz reflejada desde la superficie de grabación usando un foto detector incluido en el cabezal óptico ex401, y demodula la señal de reproducción separando una componente de señal grabada en el medio de grabación ex215 para reproducir la información necesaria. La memoria ex404 intermedia mantiene temporalmente la información a ser grabada en el medio de grabación ex215 y la información reproducida desde el medio de grabación ex215. El motor ex405 de disco rota el medio de grabación ex215. La unidad de control servo ex406 mueve el cabezal óptico ex401 a una pista de información predeterminada mientras controla el controlador de rotación del motor ex405 del disco para seguir el punto láser. La unidad de control del sistema ex407 controla en general la unidad de reproducción/grabación de información ex400. Los procesos de lectura y escritura pueden ser implementados por la unidad de control del sistema ex407 usando varia información almacenada en la memoria ex404 intermedia y generando y añadiendo nueva información según sea necesario, y a través de la unidad ex402 de grabación de modulación, la unidad de demodulación de reproducción ex403, y la unidad de control servo ex406 que graba y reproduce información a través del cabezal óptico ex401 mientras son operados de manera coordinada. La unidad de control del sistema ex407 incluye, por ejemplo, un microprocesador, y ejecuta procesamiento causando que un ordenador ejecute un programa para leer y escribir.

Aunque el cabezal óptico ex401 irradia un punto láser en la descripción, puede realizar grabación de alta densidad usando luz de campo cercano.

La figura 21 ilustra el medio de grabación ex215 que es un disco óptico. En la superficie de grabación del medio de grabación ex215, guías formadas por surcos en espiral, y una pista de información ex230 graba, por adelantado, información de dirección que indica una posición absoluta en el disco según los cambios y en la forma de los surcos de guía. La información de dirección incluye información para determinar posiciones de bloques ex231 de grabación que son una unidad para grabar datos. Reproducir la pista de información ex230 y leer la información de dirección en un aparato que graba y reproduce datos puede llevar a determinar las posiciones de los bloques de grabación. Además, el medio de grabación ex215 incluye un área de grabación de datos ex233, un área de circunferencia interior ex232, y un área de circunferencia exterior ex234. El área de grabación de datos ex233 es un área para usar en la grabación de datos de usuario. El área de circunferencia interior ex232 y el área de circunferencia exterior ex234 que están dentro y fuera del área de grabación de datos ex233, respectivamente son para uso específico excepto para grabar los datos de usuario. La unidad de reproducción/grabación de información 400 lee y escribe audio codificado, datos de video codificado, o datos multiplexados obtenidos mediante la multiplexación de audio y datos de video codificados, desde y en el área de grabación de datos ex233 del medio de grabación ex215.

Aunque un disco óptico que tiene una capa, tal como un DVD y un BD se describen como un ejemplo en la descripción, el disco óptico no está limitado a tal, y puede ser un disco óptico que tenga una estructura multicapa y que sea capaz de ser grabado en otra parte aparte de la superficie. Además, el disco óptico puede tener una estructura para grabación/reproducción multidimensional, tal como grabación de información usando luces de colores con diferentes longitudes de onda en la misma porción del disco óptico y para grabar información que tenga diferentes capas desde varios ángulos.

Además, un coche ex210 que tenga una antena ex205 puede recibir datos desde el satélite ex202 y otros, y reproducir video en un dispositivo de visualización tal como un sistema de navegación del coche ex211 instalado en el coche ex210, en el sistema de difusión digital ex200. Aquí, una configuración del sistema de navegación del coche ex211 será una configuración, por ejemplo, que incluya una unidad de recepción de GPS desde la configuración ilustrada en la figura 19. Lo mismo será verdad para la configuración del ordenador ex111, el teléfono móvil ex114, y otros.

La figura 22A ilustra el teléfono móvil ex114 que usa el método de codificación de imágenes en movimiento y el método de decodificación de imágenes en movimiento descritos en las realizaciones. El teléfono móvil ex114 incluye: una antena ex350 para transmitir y recibir ondas de radio a través de la estación base ex110; una unidad de cámara ex365 capaz de capturar imágenes fijas y en movimiento; y una unidad de visualización ex358 tal como una pantalla de cristal líquido para visualizar los datos tales como video decodificado capturado por la unidad de cámara ex365 o recibidos por la antena ex350. El teléfono móvil ex114 además incluye: una unidad de cuerpo principal que incluye una unidad de teclas de operación ex366; una unidad de salida de audio ex357 tal como un altavoz para salida de audio; una unidad de entrada de audio ex356 tal como un micrófono para entrada de audio; una unidad de memoria ex367 para almacenar video o imágenes fijas, audio grabado, datos codificados o decodificados del video recibido, las imágenes fijas, correos electrónicos, u otros; y una unidad de ranura ex364 que es una unidad de interfaz para medios de grabación que almacena datos en el mismo modo que la unidad de memoria ex367.

A continuación, se describirá un ejemplo de una configuración del teléfono móvil ex114 con referencia a la figura

22B. En el teléfono móvil ex114, una unidad de control principal ex360 diseñada para controlar en general cada unidad del cuerpo principal incluyendo la unidad de visualización ex358 además de la unidad de teclas de operación ex366 está conectada mutuamente, a través de un bus síncrono ex370, a una unidad de circuito de alimentación ex361, una unidad de control de entrada de operación ex362, una unidad de procesamiento de señal de video ex355, una unidad de interfaz de cámara ex363, una unidad de control de pantalla de cristal líquido (LCD) ex359, una unidad de modulación/demodulación ex352, una unidad de multiplexación/demultiplexación ex353, una unidad de procesamiento de señal de audio ex354, la unidad de ranura ex364 y la unidad de memoria ex367.

Cuando una tecla de fin de llamada o una tecla de encendido se activa por una operación de usuario, la unidad de circuito de alimentación ex361 suministra a las unidades respectivas alimentación desde un paquete de baterías para activar el teléfono móvil ex114.

En el teléfono móvil ex114, la unidad de procesamiento de señal de audio ex354 convierte la señal de audio recogida por la unidad de entrada de audio ex356 en modo de conversación de voz en señales de audio digitales bajo el control de la unidad de control principal ex360 incluyendo una CPU, ROM, y RAM. Entonces, la unidad de modulación/demodulación ex352 realiza procesamiento de espectro ensanchado en las señales de audio digitales, y la unidad de transmisión y recepción ex351 realiza la conversión digital a analógico y conversión de frecuencia sobre los datos, para transmitir los datos resultantes a través de la antena ex350. También, en el teléfono móvil ex114, la unidad de transmisión y recepción ex351 amplifica los datos recibidos por la antena ex350 en modo de conversación de voz y realiza conversión de frecuencia y la conversión analógica a digital sobre los datos. Entonces, la unidad de modulación/demodulación ex352 realiza el procesamiento de espectro ensanchado inverso sobre los datos, y la unidad de procesamiento de señal de audio ex354 la convierte en señales de audio analógicas, para sacarlas a través de la unidad de salida de audio ex357.

Además, cuando se transmite un correo electrónico en el modo de comunicación de datos, los datos de texto del correo electrónico introducidos mediante la operación de la unidad de teclas de operación ex366 y otros del cuerpo principal son enviados a la unidad de control principal ex360 a través de la unidad de control de entrada de operación ex362. La unidad de control principal ex360 causa que la unidad de modulación/demodulación ex352 realice procesamiento de espectro ensanchado en los datos de texto, y la unidad de transmisión y recepción ex351 realiza la conversión digital a analógico y la conversión de frecuencia en los datos resultantes para transmitir los datos a la estación base ex110 a través de la antena ex350. Cuando se recibe un correo electrónico, se realiza un procesamiento en los datos recibidos que es aproximadamente inverso al procesamiento de transmitir un correo electrónico, y los datos resultantes son proporcionados a la unidad de visualización ex358.

Cuando se transmiten video, imágenes fijas, o video y audio en el modo de comunicación de datos, la unidad de procesamiento de señal de video ex355 comprime y codifica señales de video suministradas desde la unidad de cámara ex365 usando el método de codificación de imágenes en movimiento mostrado en cada una de las realizaciones (esto es, funciona como el aparato de codificación de imágenes según el aspecto de la presente invención), y transmite los datos de video codificado a la unidad de multiplexación/demultiplexación ex353. En contraste, durante la captura de video, imágenes fijas, y otros, de la unidad de cámara ex365, la unidad de procesamiento de señal de audio ex354 codifica señales de audio recogidas por la unidad de entrada de audio ex356, y transmite los datos de audio codificados a la unidad de multiplexación/demultiplexación ex353.

La unidad de multiplexación/demultiplexación ex353 multiplexa los datos de video codificado suministrado desde la unidad de procesamiento de señal de video ex355 y los datos de audio codificado suministrados desde la unidad de procesamiento de señal de audio ex354, usando un método predeterminado. Entonces, la unidad de modulación/demodulación ex352 (unidad de circuito de modulación/demodulación) realiza procesamiento de espectro ensanchado en los datos multiplexados, y la unidad de transmisión y recepción ex351 realiza conversión digital a analógico y conversión de frecuencia en los datos para transmitir los datos resultantes a través de la antena ex350.

Cuando se reciben datos de un archivo de video que está enlazado a una página web y otros en el modo de comunicación de datos cuando se recibe un correo electrónico con video y/o audio adjunto, para decodificar los datos multiplexados recibidos a través de la antena ex350, la unidad de multiplexación/demultiplexación ex353 demultiplexa los datos multiplexados en un flujo de bits de datos de video y un flujo de bits de datos de audio, y suministra a la unidad de procesamiento de señal de video ex355 los datos de video codificados y a la unidad de procesamiento de señal de audio ex354 los datos de audio codificados, a través del bus síncrono ex370. La unidad de procesamiento de señal de video ex355 decodifica la señal de video usando un método de decodificación de imágenes en movimiento correspondiente con el método de codificación de imágenes en movimiento en cada una de las realizaciones (esto es, funciona como el aparato de decodificación de imágenes de la presente invención), y entonces la unidad de visualización ex358 muestra, por ejemplo, el video e imágenes fijas incluidas en el archivo de video enlazado a la página web a través de la unidad de control del LCD ex359. Además, la unidad de procesamiento de señal de audio ex354 decodifica la señal de audio, y la unidad de salida de audio ex357 proporciona el audio.

Además, similarmente al televisor ex300, un terminal tal como el teléfono móvil ex114 probablemente tenga 3 tipos de configuraciones de implementación incluyendo no solo (i) un terminal de transmisión y recepción que incluye

tanto un aparato de codificación como un aparato de decodificación, sino también (ii) un terminal de transmisión que incluye solo un aparato de codificación y (iii) un terminal de recepción que incluye solo un aparato de decodificación. Aunque el sistema de difusión digital ex200 recibe y transmite los datos multiplexados obtenidos mediante la multiplexación de datos de audio sobre datos de video en la descripción, los datos multiplexados pueden ser datos obtenidos mediante la multiplexación de datos que no son de audio sino datos de caracteres relacionados con el video sobre datos de video, y pueden no ser datos multiplexados sino datos de video en sí.

Como tal, el método de codificación de imágenes en movimiento y el método de decodificación de imágenes en movimiento en cada una de las realizaciones pueden ser usados en cualquiera de los dispositivos y sistemas descritos. Así, se pueden obtener las ventajas descritas en cada una de las realizaciones.

Además, la presente invención no está limitada a las realizaciones, y son posibles varias modificaciones y revisiones sin salirse del alcance de la presente invención.

(Realización 10)

Los datos de video pueden ser generados mediante la conmutación, según sea necesario, entre (i) el método de codificación de imágenes en movimiento y el aparato de codificación de imágenes en movimiento mostrado en cada una de las realizaciones y (ii) un método de codificación de imágenes en movimiento o un aparato de codificación de imágenes en movimiento en conformidad con un estándar diferente, tal como MPEG-2, MPEG-4 AVC, y VC-1.

Aquí, cuando se genera una pluralidad de datos de video conforme a los diferentes estándares y se decodifica a continuación, los métodos de decodificación necesitan ser seleccionados para ajustarse a los diferentes estándares. Sin embargo, dado que no se puede detectar con qué estándar de la pluralidad de los datos de video a ser decodificados, hay un problema de que no se pueda seleccionar un método de decodificación apropiado.

Para resolver el problema, los datos multiplexados obtenidos mediante la multiplexación de datos de audio y otros sobre datos de video tienen una estructura que incluye información de identificación que indica a qué estándar se ajustan los datos de video. En adelante se describirá la estructura específica de los datos multiplexados que incluyen los datos de video generados en el método de codificación de imágenes en movimiento y por el aparato de codificación de imágenes en movimiento mostrado en cada una de las realizaciones. Los datos multiplexados es un flujo digital en el formato de flujo de transporte MPEG-2.

La figura 23 ilustra una estructura de los datos multiplexados. Como se ilustra en la figura 23, los datos multiplexados se pueden obtener mediante la multiplexación de al menos uno entre un flujo de video, un flujo de audio, un flujo de gráficos de presentación (PG), y un flujo de gráficos interactivos. El flujo de video representa video primario y video secundario de una película, el flujo de audio (IG) representa una parte de audio primaria y una parte de audio secundaria a ser mezcladas con la parte de audio primaria, y el flujo de gráficos de representación representan subtítulos de la película. Aquí, el video primario es video normal a ser visualizado en una pantalla, y el video secundario es video a ser visualizado en una ventana más pequeña en el video primario. Además, el flujo de gráficos interactivo representa una pantalla interactiva a ser generada mediante la disposición de los componentes del GUI en una pantalla. El flujo de video es codificado en el método de codificación de imágenes en movimiento o mediante el aparato de codificación de imágenes en movimiento en cada una de las realizaciones, o en un método de codificación de imágenes en movimiento o mediante un aparato de codificación de imágenes en movimiento en conformidad con un estándar convencional, tal como MPEG-2, MPEG-4 AVC, y VC-1. El flujo de audio está codificado según un estándar, tal como Dolby-AC-3, Dolby Digital Plus, MLP, DTS, DTS-HD, y PCM lineal.

Cada flujo incluido en los datos multiplexados es identificado por PID. Por ejemplo, 0x1011 está asignado en el flujo de video a ser usado por el video de una película, 0x1100 a 0x111F están asignados a los flujos de audio, 0x1200 a 0x121F están asignados a los flujos de gráficos de presentación, 0x1400 a 0x141F están asignados a los flujos de gráficos interactivos, 0x1B00 a 0x1B1F están asignados a los flujos de video a ser usados por el video secundario de la película, y 0x1A00 a 0x1A1F están asignados a los flujos de audio a ser usados por el video secundario a ser mezclado con el audio primario.

La figura 24 ilustra esquemáticamente cómo se multiplexan los datos. Primero, un flujo de video ex235 compuesto de tramas de video y un flujo de audio ex238 compuesto de tramas de audio son transformados en un flujo de paquetes PES ex236 y un flujo de paquetes PES ex239, y además en paquetes TS ex237 y paquetes TS ex240, respectivamente. Similarmente, los datos del flujo de gráficos de presentación ex241 y datos de un flujo de gráficos interactivos ex244 son transformados en un flujo de paquetes PES ex242 y un flujo de paquetes PES ex245, y además en paquetes TS ex243 y paquetes TS ex246, respectivamente. Estos paquetes TS son multiplexados en un flujo para obtener datos multiplexados ex247.

La figura 25 ilustra cómo un flujo de video es almacenado en un flujo de paquetes PES en más detalle. La primera barra en la figura 25 muestra un flujo de tramas de video en un flujo de video. La segunda barra muestra el flujo de paquetes PES. Como se indica por las flechas denotadas como yy1, yy2, yy3 y yy4 en la figura 25, el flujo de video es dividido en imágenes como imágenes I, imágenes B, e imágenes P cada una de las cuales es una unidad de presentación de video, y las imágenes son almacenadas en una carga de cada uno de los paquetes PES. Cada uno de los paquetes PES tiene una cabecera PES, y la cabecera PES almacena una marca de tiempo de presentación

(PTS) que indica un momento de visualización de la imagen, y una marca de tiempo de decodificación (DTS) que indica un momento de decodificación de la imagen.

La figura 26 ilustra un formato de paquetes TS a ser finalmente escritos en los datos multiplexados. Cada uno de los paquetes TS es un paquete de longitud fija de 188 bytes que incluye una cabecera TS de 4 bytes que tiene información, tal como un PID para identificar un flujo y una carga de TS de 184 bytes para almacenar datos. Los paquetes PES son divididos, y almacenados en las cargas TS, respectivamente. Cuando se usa una ROM BD, a cada uno de los paquetes TS se le da una Cabecera Extra TP de 4 bytes, resultando así en paquetes origen de 192 bytes. Los paquetes origen son escritos en los datos multiplexados. Las cabeceras extra TP almacenan información tal como una marca de tiempo de llegada (ATS). La ATS muestra un momento de inicio de la transferencia en el cual cada uno de los paquetes TS ha de ser transferido a un filtro PID. Los paquetes origen son dispuestos en los datos multiplexados como se muestra en la parte inferior de la figura 26. Los números que aumentan desde la cabecera de los datos multiplexados son llamados números de paquetes origen (SPN).

Cada uno de los paquetes TS incluidos en los datos multiplexados incluye no solo flujos de audio, video, subtítulos y otros, sino también una tabla asociada del programa (PAT), una tabla de correspondencia del programa (PMT), y una referencia de reloj del programa (PCR). La PAT muestra lo que un PID en una PMT usada en los datos multiplexados indica, y un PID de la PAT en sí es registrada a cero. La PMT almacena PID de los flujos de video, audio, subtítulos y otros incluidos en los datos multiplexados, e información de atributos de los flujos correspondientes a los PID. La PMT también tiene varios descriptores relativos a los datos multiplexados. Los descriptores tienen información tal como información de control de copia que muestra si la copia de los datos multiplexados está permitida o no. La PCR almacena información de tiempo STC correspondiente a una ATS que muestra cuando el paquete PCR es transferido a un decodificador, para lograr la sincronización entre el reloj de tiempo de llegada (ATC) que es un eje de tiempo de ATS, y un reloj de hora del sistema (STC) que es un eje de tiempo de PTS y DTS.

La figura 27 ilustra la estructura de datos de una PMT en detalle. Una cabecera PMT es dispuesta en la parte superior de la PMT. La cabecera PMT describe la longitud de los datos incluidos en la PMT y otros. Una pluralidad de descriptores relativos a los datos multiplexados es dispuesta tras la cabecera PMT. Información tal como la información de control de copia es descrita en los descriptores. Tras los descriptores, se dispone una pluralidad de piezas de información de flujo relativas a los flujos incluidos en los datos multiplexados. Cada pieza de información de flujo incluye descriptores de flujo cada uno describiendo información, tal como un tipo de flujo para identificar un códec de compresión de un flujo, un PID del flujo, e información de atributos del flujo (tal como una tasa de trama o una relación de aspecto). Los descriptores de flujo son iguales en número al número de flujos en los datos multiplexados.

Cuando los datos multiplexados se graban en un medio de grabación y otros, se graban junto con archivos de información de datos multiplexados.

Cada uno de los archivos de información de datos multiplexados es información de gestión de los datos multiplexados como se muestra en la figura 28. Los archivos de información de datos multiplexados se corresponden uno a uno con los datos multiplexados, y cada uno de los archivos incluye información de datos multiplexados, información de atributos de flujo, y una correspondencia de entrada.

Como se ilustra en la figura 28, los datos multiplexados incluyen una tasa de sistema, un momento de inicio de reproducción, y un momento de fin de reproducción. La tasa de sistema indica la tasa de transferencia máxima en la cual un decodificador objetivo del sistema que será descrito más tarde transfiere los datos multiplexados a un filtro PID. Los intervalos de las ATS incluidas en los datos multiplexados son establecidos a no más altas que una tasa de sistema. El momento de inicio de reproducción indica un PTS en una trama de video en la cabecera de los datos multiplexados. Un intervalo de una trama es añadido a un PTS en una trama de video en el final de los datos multiplexados, y el PTS se establece al momento final de reproducción.

Como se muestra en la figura 29, una pieza de información de atributo es registrada en la información de atributo de flujo, para cada PID de cada flujo incluido en los datos multiplexados. Cada pieza de información de atributo tiene información diferente según si el flujo correspondiente es un flujo de video, un flujo de audio, un flujo de gráficos de representación, o un flujo de gráficos interactivos. Cada pieza de información de atributo de flujo de video porta información que incluye qué tipo de códec de compresión se usa para comprimir el flujo de video, y la resolución, relación de aspecto y tasa de trama de las piezas de datos de imágenes que se incluyen en el flujo de video. Cada pieza de información de atributo de flujo de audio porta información que incluye qué tipo de códec de compresión se usa para comprimir el flujo de audio, cuántos canales se incluyen en el flujo de audio, qué idioma soporta el flujo de audio, y cómo de alta es la frecuencia de muestreo. La información de atributo de flujo de video y la información de atributo de flujo de audio se usan para inicializar un decodificador antes de que el reproductor reproduzca la información.

En la presente realización, los datos multiplexados a ser usados son de un tipo de flujo incluido en la PMT. Además, cuando los datos multiplexados se graban en un medio de grabación, se usa la información de atributo del flujo de video incluida en la información de los datos multiplexados. Más específicamente, el método de codificación de



imágenes en movimiento o el aparato de codificación de imágenes en movimiento descritos en cada una de las realizaciones incluyen un paso o una unidad para asignar información única que indique los datos de video generados por el método de codificación de imágenes en movimiento o el aparato de codificación de imágenes en movimiento en cada una de las realizaciones, al tipo de flujo incluido en la PMT o la información de atributo de flujo de video. Con la configuración, los datos de video generados por el método de codificación de imágenes en movimiento o el aparato de codificación de imágenes en movimiento descritos en cada una de las realizaciones se pueden distinguir de datos de video que se ajustan a otro estándar.

Además, la figura 30 muestra pasos del método de decodificación de imágenes en movimiento según la presente realización. En el paso exS100, el tipo de flujo incluido en la PMT o la información de atributo de flujo de video se obtiene desde los datos multiplexados. A continuación, en el paso exS101, se determina si el tipo de flujo o la información de atributo de flujo de video indican o no que los datos multiplexados se generan por el método de codificación de imágenes en movimiento o el aparato de codificación de imágenes en movimiento en cada una de las realizaciones. Cuando se determina que el tipo de flujo o la información de atributo de flujo de video indican que los datos multiplexados se generan por el método de codificación de imágenes en movimiento o el aparato de codificación de imágenes en movimiento en cada una de las realizaciones, en el paso exS102, se realiza la decodificación mediante el método de decodificación de imágenes en movimiento en cada una de las realizaciones. Además, cuando el tipo de flujo o la información de atributo de flujo de video indican conformidad con los estándares convencionales, tales como MPEG-2, MPEG-4 AVC, y VC-1, en el paso exS103, se realiza la decodificación mediante un método de decodificación de imágenes en movimiento en conformidad con los estándares convencionales.

Como tal, asignar un nuevo valor único al tipo de flujo o la información de atributo de flujo de video permite determinar si el método de decodificación de imágenes en movimiento o el aparato de decodificación de imágenes en movimiento que son descritos en cada una de las realizaciones pueden realizar decodificación o no. Aun cuando los datos multiplexados se ajustan a un estándar diferente, se puede seleccionar un método o aparato de decodificación apropiado. Así, se vuelve posible decodificar información sin errores. Además, el método o aparato de codificación de imágenes en movimiento, o el método o aparato de decodificación de imágenes en movimiento en la presente realización se pueden usar en los dispositivos y sistemas descritos anteriormente.

(Realización 11)

Cada uno del método de codificación de imágenes en movimiento, el aparato de codificación de imágenes en movimiento, el método de decodificación de imágenes en movimiento, y el aparato de decodificación de imágenes en movimiento en cada una de las realizaciones se consigue típicamente en la forma de un circuito integrado o un circuito Integrado a gran escala (LSI). Como un ejemplo del LSI, la figura 31 ilustra una configuración del LSI ex500 que está hecho en un chip. El LSI ex500 incluye los elementos ex501, ex502, ex503, ex504, ex505, ex506, ex507, ex508, y ex509 que serán descritos a continuación, y los elementos están conectados entre sí a través de un bus ex510. La unidad ex505 de circuito de alimentación se activa por el suministro a cada uno de los elementos cuando la unidad ex505 de circuito de alimentación se enciende.

Por ejemplo, cuando se realiza la codificación, el LSI ex500 recibe una señal AV desde un micrófono ex117, una cámara ex113, y otros a través de un AV IO ex509 bajo el control de una unidad de control ex501 que incluye una CPU ex502, un controlador de memoria ex503, un controlador de flujo ex504, y una unidad de control de frecuencia portadora ex512. La señal AV recibida es almacenada temporalmente en una memoria externa ex511 tal como una SDRAM. Bajo el control de la unidad de control ex501, los datos almacenados son segmentados en porciones de datos según la cantidad y velocidad de procesamiento a ser transmitidos a una unidad de procesamiento de señal ex507. Entonces, la unidad de procesamiento de señal ex507 codifica una señal de audio y/o una señal de video. Aquí, la codificación de la señal de video es la codificación descrita en cada una de las realizaciones. Además, la unidad de procesamiento de señal ex507 a veces multiplexa los datos de audio codificados y los datos de video codificados, y un flujo IO ex506 proporciona los datos multiplexados hacia afuera. Los datos multiplexados proporcionados son transmitidos a la estación base ex107, o escritos en el medio de grabación ex215. Cuando los conjuntos de datos son multiplexados, los datos deberían ser almacenados temporalmente en la memoria intermedia ex508 para que los conjuntos de datos se sincronicen entre sí.

Aunque la memoria ex511 es un elemento fuera del LSI ex500, se puede incluir en el LSI ex500. La memoria intermedia ex508 no está limitada a una memoria intermedia, sino que puede estar compuesta de memorias intermedias. Además, el LSI ex500 puede estar hecho en un chip o una pluralidad de chips.

Además, aunque la unidad de control ex501 incluye la CPU ex502, el controlador de memoria ex503, el controlador de flujo ex504, la unidad de control de frecuencia portadora ex512, la configuración de la unidad de control ex501 no está limitada a tales. Por ejemplo, la unidad de procesamiento de señal ex507 puede además incluir una CPU. La inclusión de otra CPU en la unidad de procesamiento de señal ex507 puede mejorar la velocidad de procesamiento. Además, como otro ejemplo, la CPU ex502 puede servir como o ser una parte de la unidad de procesamiento de señal ex507, y, por ejemplo, puede incluir una unidad de procesamiento de señal de audio. En tal caso, la unidad de control ex501 incluye la unidad de procesamiento de señal ex507 o la CPU ex502 incluyendo una parte de la unidad de procesamiento de señal ex507.

El nombre usado aquí es LSI, pero también puede llamarse IC, sistema LSI, super LSI, o ultra LSI dependiendo del grado de integración.

Además, los modos de conseguir la integración no están limitados al LSI, y un circuito especial o un procesador de propósito general etcétera pueden también conseguir la integración. La matriz de puertas programables en campo (FPGA) puede ser programada para permitir la reconfiguración de la conexión o se puede usar la configuración de un LSI para el mismo propósito. Tal dispositivo de lógica programable puede ejecutar típicamente el método de codificación de imágenes en movimiento y/o el método de decodificación de imágenes en movimiento según cualquiera de las realizaciones anteriores, mediante la carga o lectura de una memoria o similar de uno o más programas que están incluidos en el software o firmware.

En el futuro, con los adelantos en tecnología de semiconductores, una nueva tecnología puede reemplazar al LSI. Los bloques funcionales se pueden integrar usando tal tecnología. La posibilidad es que la presente invención sea aplicada en biotecnología.

(Realización 12)

Cuando los datos de video generados en el método de codificación de imágenes en movimiento o mediante el aparato de codificación de imágenes en movimiento descrito en cada una de las realizaciones son decodificados, comparado con cuando los datos de video se ajustan a un estándar convencional, tal como MPEG-2, MPEG-4 AVC y VC-1 son decodificados, la cantidad de procesamiento probablemente aumenta. Así, el LSI ex500 necesita ser configurado para una frecuencia portadora más alta que la del CPU ex502 para usarse cuando los datos de video son decodificados ajustándose a un estándar convencional. Sin embargo, cuando la frecuencia portadora se configura más alta, hay un problema de que el consumo de potencia aumente.

Para solucionar el problema, el aparato de decodificación de imágenes en movimiento, tal como el televisor ex300 y el LSI ex500 son configurados para determinar a qué estándar se ajustan los datos de video, y conmutar entre las frecuencias portadoras según el estándar determinado. La figura 32 ilustra una configuración ex800 en la presente realización. Una unidad de conmutación de frecuencia portadora ex803 establece una frecuencia portadora a una frecuencia portadora más alta cuando los datos de video se generan por el método de codificación de imágenes en movimiento o el aparato de codificación de imágenes en movimiento descritos en cada una de las realizaciones. Entonces, la unidad de conmutación de frecuencia portadora ex803 da instrucciones a una unidad de procesamiento de decodificación ex801 que ejecuta el método de decodificación de imágenes en movimiento descrito en cada una de las realizaciones para decodificar los datos de video. Cuando los datos de video se ajustan al estándar convencional, la unidad de conmutación de frecuencia portadora ex803 establece una frecuencia portadora a una frecuencia portadora más baja que aquella de los datos de video generados por el método de codificación de imágenes en movimiento o el aparato de codificación de imágenes en movimiento descritos en cada una de las realizaciones. Entonces, la unidad de conmutación de frecuencia portadora ex803 da instrucciones a la unidad de procesamiento de decodificación ex802 que se ajusta al estándar convencional para decodificar los datos de video.

Más específicamente, la unidad de conmutación de frecuencia portadora ex803 incluye la CPU ex502 y la unidad de control de frecuencia portadora ex512 en la figura 31. Aquí, cada una de la unidad de procesamiento de decodificación ex801 que ejecuta el método de decodificación de imágenes en movimiento descrito en cada una de las realizaciones y la unidad de procesamiento de decodificación ex802 que se ajusta al estándar convencional se corresponden con la unidad de procesamiento de señal ex507 en la figura 31. La CPU ex502 determina a qué estándar se ajustan los datos de video. Entonces, la unidad de control de frecuencia portadora ex512 determina una frecuencia portadora basada en una señal de la CPU ex502. Además, la unidad de procesamiento de señal ex507 decodifica los datos de video basado en la señal de la CPU ex502. Por ejemplo, la información de identificación descrita en la realización 10 se usa probablemente para identificar los datos de video. La información de identificación no está limitada a la descrita en la realización 10 sino que puede ser cualquier información mientras que la información indique a qué estándar se ajustan los datos de video. Por ejemplo, cuando se puede determinar a qué estándar se ajustan los datos de video basándose en una señal externa para determinar que los datos de video se usan para un televisor o un disco, etc., la determinación puede hacerse basándose en tal señal externa. Además, la CPU ex502 selecciona una frecuencia portadora basándose en, por ejemplo, una tabla de búsqueda en la cual se asocian los estándares de los datos de video con las frecuencias portadoras como se muestra en la figura 34. La frecuencia portadora se puede seleccionar mediante el almacenamiento de la tabla de búsqueda en la memoria intermedia ex508 y en una memoria interna de un LSI, y con referencia a la tabla de búsqueda por la CPU ex502.

La figura 33 ilustra los pasos para ejecutar un método en la presente realización. Primero, en el paso exS200, la unidad de procesamiento de señal ex507 obtiene información de identificación desde los datos multiplexados. A continuación, en el paso exS201, la CPU ex502 determina si los datos de video se generan o no por el método de codificación y el aparato de codificación descritos en cada una de las realizaciones, basándose en la información de identificación. Cuando se generan los datos de video mediante el método de codificación de imágenes en movimiento y el aparato de codificación de imágenes en movimiento descritos en cada una de las realizaciones, en el paso exS202, la CPU ex502 transmite una señal para establecer la frecuencia portadora a una frecuencia portadora más alta a la unidad de control de frecuencia portadora ex512. Entonces, la unidad de control de frecuencia portadora ex512 establece la frecuencia portadora a la frecuencia portadora más alta. Por otro lado,

cuando la información de identificación indica que los datos de video se ajustan al estándar convencional, tal como MPEG-2, MPEG-4 AVC, y VC-1, en el paso exS203, la CPU ex502 transmite una señal para establecer la frecuencia portadora a una frecuencia portadora más baja a la unidad de control de frecuencia portadora ex512. Entonces, la unidad de control de frecuencia portadora ex512 establece la frecuencia portadora a la frecuencia portadora más baja que en el caso en el que los datos de video se generan por el método de codificación de imágenes en movimiento y el aparato de codificación de imágenes en movimiento descritos en cada una de las realizaciones.

Además, junto con la conmutación de las frecuencias portadoras, el efecto de conservación de potencia se puede mejorar cambiando el voltaje a aplicar al LSI ex500 o a un aparato que incluye el LSI ex500. Por ejemplo, cuando la frecuencia portadora se establece más baja, el voltaje a aplicar al LSI ex500 o al aparato que incluye el LSI ex500 se establece probablemente a un voltaje más bajo que en el caso en el que la frecuencia portadora se establece más alta.

Además, cuando la cantidad de procesamiento para decodificar es mayor, la frecuencia portadora puede establecerse más alta, y cuando la cantidad de procesamiento para decodificar es menor, la frecuencia portadora puede establecerse más baja como el método para establecer la frecuencia portadora. Así, el método de establecimiento no está limitado a los descritos anteriormente. Por ejemplo, cuando la cantidad de procesamiento para decodificar datos de video en conformidad con MPEG-4 AVC es mayor que la cantidad de procesamiento para decodificar datos de video generados por el método de codificación de imágenes en movimiento y el aparato de codificación de imágenes en movimiento descritos en cada una de las realizaciones, la frecuencia portadora se establece probablemente en orden inverso a los establecimientos descritos anteriormente.

Además, el método para establecer la frecuencia portadora no está limitado al método para establecer la frecuencia portadora más baja. Por ejemplo, cuando la información de identificación indica que los datos de video se generan por el método de codificación de imágenes en movimiento y el aparato de codificación de imágenes en movimiento descritos en cada una de las realizaciones, el voltaje a aplicar al LSI ex500 o el aparato que incluye el LSI ex500 se establece probablemente más alto. Cuando la información de identificación indica que los datos de video se ajustan al estándar convencional, tal como MPEG-2, MPEG-4 AVC, y VC-1, el voltaje a ser aplicado al LSI ex500 o al aparato que incluye el LSI ex500 se establece probablemente más bajo. Como otro ejemplo, cuando la información de identificación indica que los datos de video se generan por el método de codificación de imágenes en movimiento y el aparato de codificación de imágenes en movimiento descritos en cada una de las realizaciones, no es probable que tenga que suspenderse la conducción de la CPU ex502. Cuando la información de identificación indica que los datos de video se ajustan al estándar convencional, tal como MPEG-2, MPEG-4 AVC, y VC-1, la conducción de la CPU ex502 se suspende probablemente en un momento dado porque la CPU ex502 tiene capacidad de procesamiento extra. Aun cuando la información de identificación indica que los datos de video se generan por el método de codificación de imágenes en movimiento y el aparato de codificación de imágenes en movimiento descritos en cada una de las realizaciones, en el caso en el que la CPU ex502 tiene capacidad de procesamiento extra, la conducción de la CPU ex502 se suspende probablemente en un momento dado. En tal caso, el tiempo de suspensión probablemente se establece más corto que aquel en el caso en el que la información de identificación indica que los datos de video se ajustan al estándar convencional, tal como MPEG-2, MPEG-4 AVC, y VC-1.

En consecuencia, el efecto de conservación de potencia puede mejorarse mediante la conmutación entre las frecuencias portadoras según el estándar al cual se ajusten los datos de video. Además, cuando el LSI ex500 o el aparato que incluye el LSI ex500 son conducidos usando una batería, la vida de la batería puede extenderse con el efecto de conservación de potencia.

(Realización 13)

Hay casos donde una pluralidad de datos de video que se ajustan a diferentes estándares, son proporcionados a los dispositivos y sistemas, tal como un televisor y un teléfono móvil. Para permitir la decodificación de la pluralidad de datos de video que se ajustan a diferentes estándares, la unidad de procesamiento de señal ex507 del LSI ex500 necesita ajustarse a los diferentes estándares. Sin embargo, los problemas de aumentar la escala del circuito del LSI ex500 y aumentar el coste surgen con el uso individual de las unidades de procesamiento de señal ex507 que se ajustan a los estándares respectivos.

Para resolver el problema, lo que se concibe es una configuración en la cual la unidad de procesamiento de decodificación para implementar el método de decodificación de imágenes en movimiento descrito en cada una de las Realizaciones y la unidad de procesamiento de decodificación que se ajusta al estándar convencional, tal como MPEG-2, MPEG-4 AVC, y VC-1 son compartidas parcialmente. Ex900 en la figura 35A muestra un ejemplo de la configuración. Por ejemplo, el método de decodificación de imágenes en movimiento descrito en cada una de las realizaciones y el método de decodificación de imágenes en movimiento que se ajusta a MPEG-4 AVC tienen, parcialmente en común, los detalles del procesamiento, tal como codificación de entropía, cuantificación inversa, filtrado de desbloqueo, y predicción de compensación de movimiento. Es probable que una unidad de procesamiento de decodificación ex902 que se ajusta a MPEG-4 AVC sea compartida por operaciones de procesamiento comunes, y que una unidad de procesamiento de decodificación dedicada ex901 se use para un procesamiento que es único en un aspecto de la presente invención y no se ajuste a MPEG-4-AVC. Es probable que una unidad de procesamiento de decodificación ex902 que se ajusta a MPEG-4 AVC sea compartida por operaciones de

procesamiento comunes, y que una unidad de procesamiento de decodificación dedicada ex901 se use para el procesamiento que es único en un aspecto de la presente invención y que no se ajusta a MPEG-4 AVC. La unidad de procesamiento de decodificación para implementar el método de decodificación de imágenes en movimiento descrito en cada una de las realizaciones puede compartirse para compartir el procesamiento, y una unidad de procesamiento de decodificación dedicada puede usarse para procesar únicamente la del MPEG-4 AVC.

Además, ex1000 en la figura 35B muestra otro ejemplo en el que se comparte parcialmente el procesamiento. Este ejemplo usa una configuración que incluye una unidad de procesamiento de decodificación dedicada ex1001 que soporta el procesamiento único a la presente invención, una unidad de procesamiento de decodificación dedicada ex1002 que soporta el procesamiento único a otra estándar convencional, y una unidad de procesamiento de decodificación ex1003 que soporta el procesamiento compartido entre el método de decodificación de imágenes en movimiento en la presente invención y el método de decodificación de imágenes en movimiento convencional. Aquí, las unidades de procesamiento de decodificación ex1001 y ex1002 no están necesariamente especializadas para el procesamiento de la presente invención y el procesamiento de los estándares convencionales, respectivamente, y pueden ser las capaces de implementar procesamiento general. Además, la configuración de la presente realización puede implementarse mediante el LSI ex500.

Como tal, reducir la escala del circuito del LSI y reducir el coste es posible compartiendo la unidad de procesamiento de decodificación para el procesamiento a ser compartido entre el método de decodificación de imágenes en movimiento en la presente invención y el método de decodificación de imágenes en movimiento en conformidad con el estándar convencional.

A continuación, se describen características de algunas realizaciones preferidas adicionales.

Preferiblemente, en el método de codificación de imagen según un aspecto de la invención, la indicación de dependencia se incluye en el flujo de datos cuando la bandera que habilita una porción dependiente indica la inclusión de la porción dependiente.

También preferiblemente, la bandera que habilita una porción dependiente se dispone en un inicio del conjunto de parámetros.

También preferiblemente, cada una de las porciones incluye una pluralidad de macro bloques y el procesamiento de codificación sobre la porción actual se inicia después de que el procesamiento de codificación se realice sobre dos de los macro bloques incluidos en una porción actual inmediatamente precedente.

También preferiblemente, la indicación de dependencia no se incluye en una cabecera de porción de una porción que se procesa en primer lugar para la imagen, de entre las porciones.

Preferiblemente, en el método de decodificación de imagen según un aspecto de la invención, la indicación de dependencia se extrae del flujo de bits cuando la bandera que habilita una porción dependiente indica la inclusión de la porción dependiente.

También preferiblemente, la bandera que habilita una porción dependiente se dispone en un inicio del conjunto de parámetros.

También preferiblemente, cada una de las porciones incluye una pluralidad de macro bloques, y el procesamiento de decodificación sobre la porción actual se inicia después del procesamiento de decodificación se realiza sobre dos de los macro bloques incluidos en una porción actual inmediatamente precedente.

También preferiblemente, la indicación de dependencia no se incluye en una cabecera de porción de una porción que se procesa en primer lugar para la imagen, de entre las porciones.

Un aparato de codificación de imagen según un aspecto de la presente invención es un aparato de codificación de imagen que realiza el procesamiento de codificación dividiendo una imagen en una pluralidad de porciones y comprende un codificador que transmite un flujo de bits que incluye una bandera que habilita una porción dependiente que indica si la imagen incluye o no una porción dependiente sobre la que se realiza el procesamiento de codificación dependiendo del resultado del procesamiento de codificación sobre una porción diferente de una porción actual, una dirección de porción que indica una posición de inicio de la porción actual, y una indicación de dependencia que indica si la porción actual es o no la porción dependiente. La bandera que habilita una porción dependiente se dispone en un conjunto de parámetros común a las porciones. La dirección de porción se dispone en una cabecera de porción de la porción actual. La indicación de dependencia se dispone en la cabecera de porción y se dispone antes de la dirección de porción y después de un elemento de sintaxis que identifica el conjunto de parámetros.

Un aparato de decodificación de imagen según un aspecto de la presente invención es un aparato de decodificación de imagen que realiza el procesamiento de decodificación dividiendo una imagen en una pluralidad de porciones y comprende un decodificador que extrae, de un flujo de bits codificado, una bandera que habilita una porción dependiente que indica si la imagen incluye o no una porción dependiente sobre la que se realiza el procesamiento

de decodificación dependiendo del resultado el procesamiento de decodificación sobre una porción diferente de una porción actual, una dirección de porción que indica una posición de inicio de la porción actual, y una indicación de dependencia que indica si la porción actual es o no la porción dependiente. La bandera que habilita una porción dependiente se dispone en un conjunto de parámetros común a las porciones. La dirección de porción se dispone en una cabecera de porción de la porción actual. La indicación de dependencia se dispone en la cabecera de porción y se dispone antes de la dirección de porción y después de un elemento de sintaxis que identifica el conjunto de parámetros.

Un aparato de decodificación y codificación de imagen según un aspecto de la presente invención comprende el aparato de codificación de imagen descrito anteriormente según un aspecto de la presente invención y el aparato de decodificación de imagen descrito anteriormente según un aspecto de la invención.

**Aplicabilidad industrial**

Un método de codificación de imágenes en movimiento y el método de decodificación de imágenes en movimiento según la presente invención se pueden aplicar a varios datos multimedia. El método de codificación de imagen y el método de decodificación de imagen según la presente invención son útiles como un método de codificación de imagen y un método de decodificación de imagen en almacenamiento, transmisión, comunicación, y similares usando un teléfono móvil, un dispositivo de DVD, un ordenador personal, y similares.

**Lista de signos de referencia**

- 100 codificador
- 105 sustractor
- 20 110 unidad de transformación
- 120 unidad de cuantificación
- 130, 230unidad de transformación inversa
- 140, 240sumador
- 150, 250filtro de desbloqueo
- 25 160, 260filtro de bucle adaptativo
- 170, 270memoria de trama
- 180, 280unidad de predicción
- 190 codificador de entropía
- 200 decodificador
- 30 290 decodificador de entropía
- 300, 400, 710 imagen
- 31, 32, 3i, 41, 42 fila de la LCU
- 311, 312, 3i1, 321 LCU
- 500 cabecera de paquete
- 35 510 cabecera IP
- 520, 550campo de extensión
- 530 cabecera UDP
- 540 cabecera RTP
- 560 cabecera de carga útil
- 40 570 cabecera NAL
- s1 señal de entrada
- s2 señal de predicción

$e, e'$  señal de error de predicción

$s', s'', s_3$  señal reconstruida

**REIVINDICACIONES**

1. Método de decodificación de imagen de realización de procesamiento de decodificación de una imagen sobre una base por porción (31, 32, 41, 42, 43, 44, ... 4m), comprendiendo el método de decodificación de imagen
  - 5 extraer, de un flujo de bits codificado:
    - una bandera que habilita una porción dependiente que indica si está habilitada o no una porción dependiente (42, 43, 44) para la imagen, en el que la porción dependiente es una porción en la que se realiza el procesamiento de decodificación usando parcialmente información de control incluida en una cabecera de una porción diferente de la porción dependiente,
    - 10 una dirección de porción que indica una posición de inicio de la porción actual de la imagen, y
      - una indicación de dependencia (601) que indica si la porción actual es la porción dependiente (42, 43, 44) o no,
      - en el que la bandera que habilita una porción dependiente se dispone en un conjunto de parámetros común a todas las porciones de la imagen,
      - 15 la dirección de porción se dispone en una cabecera de porción (1310) de la porción actual, y
        - la indicación de dependencia (601) se dispone en la cabecera de porción (1310) de la porción actual antes de la dirección de porción y tras un elemento de sintaxis que identifica el conjunto de parámetros.
  2. Método de decodificación de imagen según la reivindicación 1, en el que la indicación de dependencia (601) se extrae del flujo de bits cuando la bandera que habilita una porción dependiente indica que la porción dependiente (42, 43, 44) está habilitada para la imagen.
  3. Método de decodificación de imagen según una de las reivindicaciones 1 y 2, en el que la bandera que habilita una porción dependiente se dispone en un inicio del conjunto de parámetros.
  4. Método de decodificación de imagen según una de las reivindicaciones 1 a 3, en el que la indicación de dependencia (601) no está incluida en una cabecera de porción (1520) de una porción (31, 41) que se procesa primero para la imagen (300, 400).
  5. Método de decodificación de imagen según una de las reivindicaciones 1 a 4, en el que la decodificación de la porción dependiente no inicializa tablas de contexto del CABAC.
  6. Método de decodificación de imagen según una de las reivindicaciones 1 a 5, en el que la decodificación de la porción diferente de la porción dependiente inicializa tablas de contexto del CABAC.
  7. Aparato de decodificación de imagen que realiza el procesamiento de decodificación de una imagen sobre una base por porción (31, 32, 41, 42, 43, 44, ... 4m), comprendiendo el aparato de decodificación de imagen:
    - un decodificador (290), que extrae de un flujo de bits codificado:
      - 35 una bandera que habilita una porción dependiente que indica si está habilitada o no una porción dependiente (42, 43, 44) para la imagen, en el que la porción dependiente es una porción en la que se realiza el procesamiento de decodificación usando parcialmente información de control incluida en una cabecera de una porción diferente de la porción dependiente,
      - una dirección de porción que indica una posición de inicio de una porción actual de la imagen, y
        - 40 una indicación de dependencia (601) que indica si la porción actual es la porción dependiente (42, 43, 44) o no;
        - en el que la bandera que habilita una porción dependiente se dispone en un conjunto de parámetros común a todas las porciones de la imagen,
        - la dirección de porción se dispone en una cabecera de porción (1320) de la porción actual, y
          - 45 la indicación de dependencia (601) se dispone en la cabecera de porción (1310) de la porción actual antes de la dirección de porción y tras un elemento de sintaxis que identifica el conjunto de parámetros.

FIG. 1

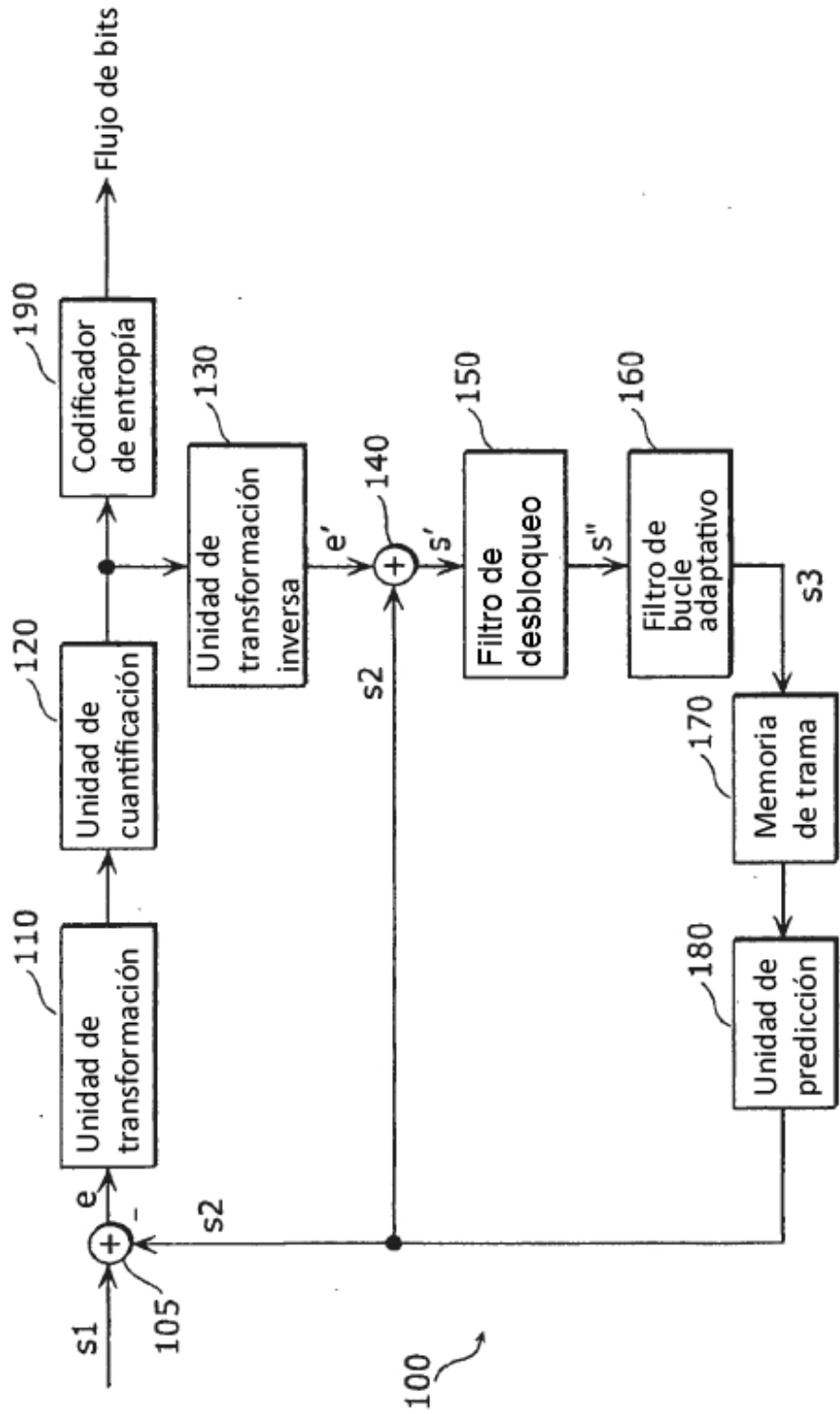




FIG. 2

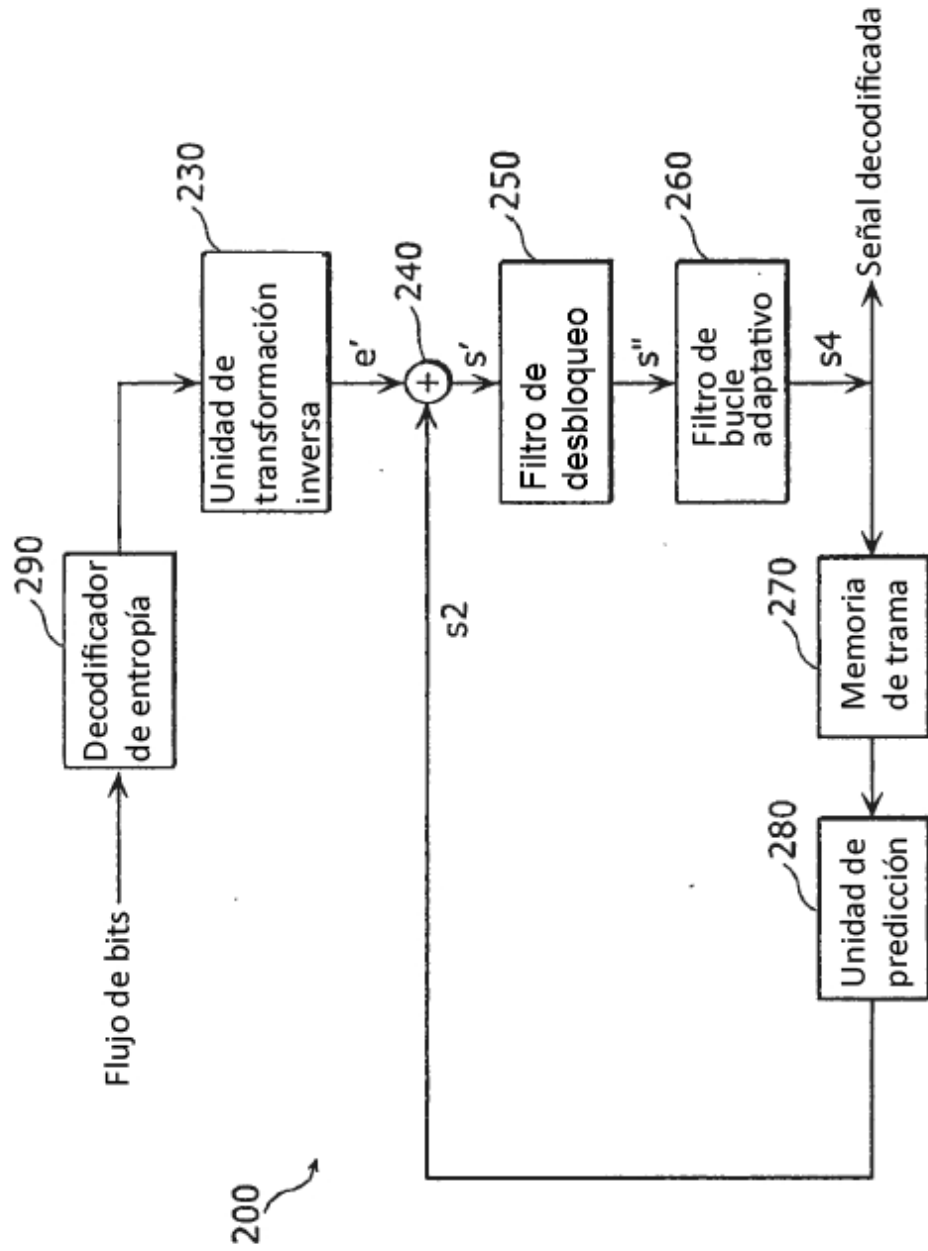


FIG. 3

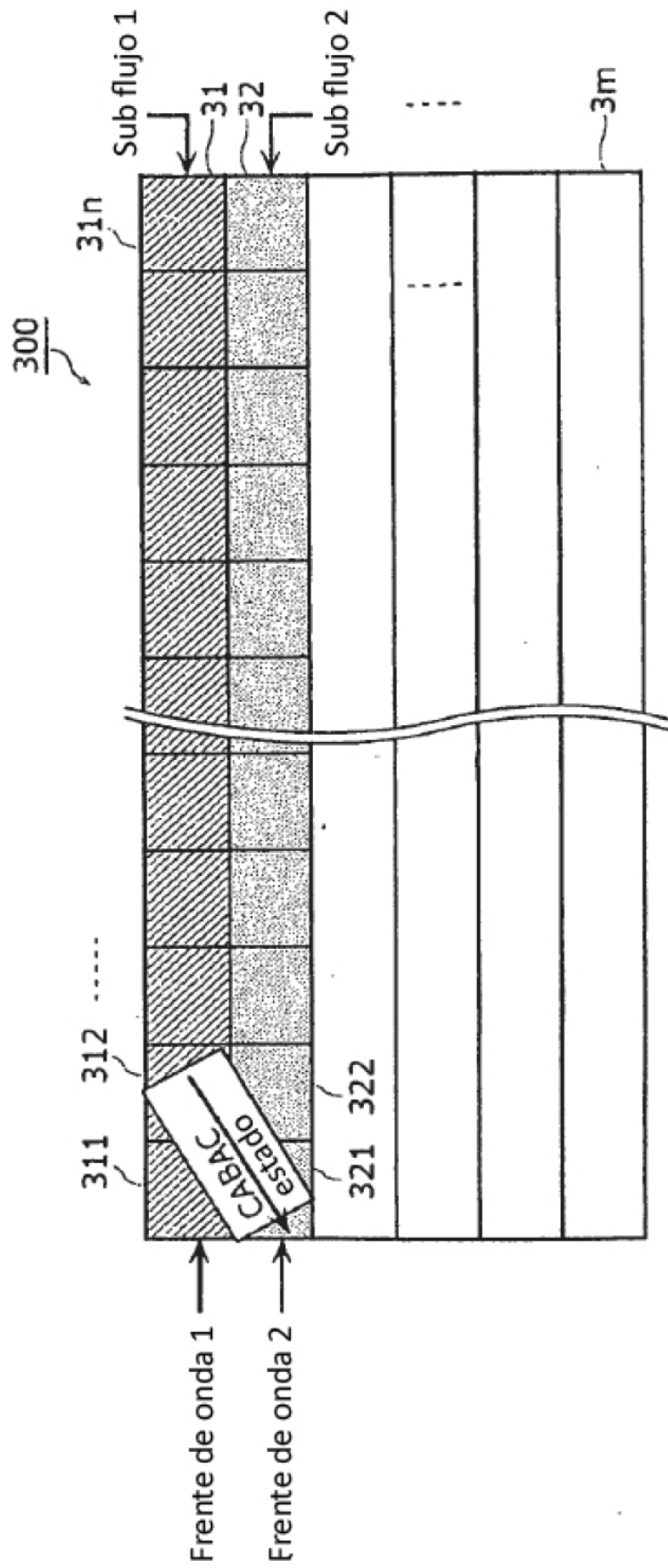


FIG. 4

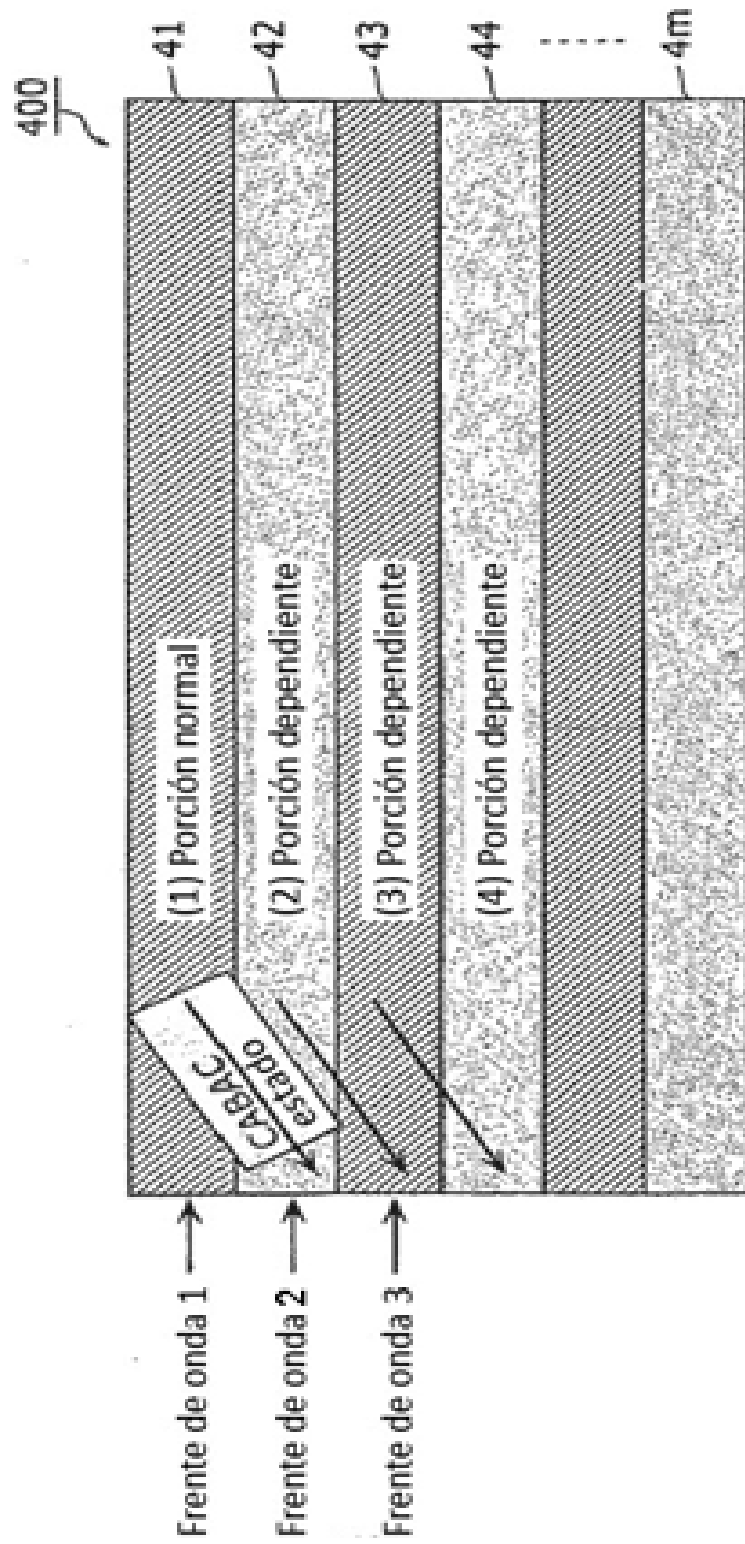


FIG. 5

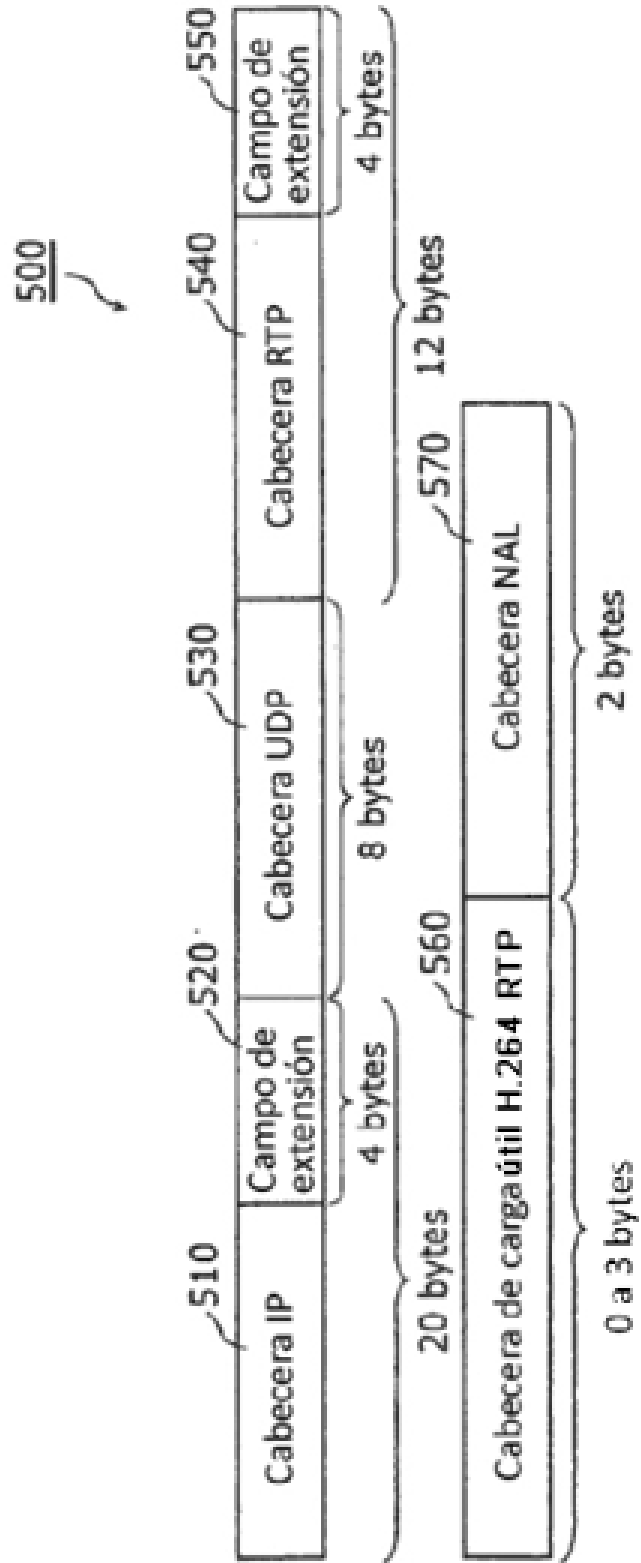


FIG. 6

600

	Descriptor
slice_header() {	
<b>first_slice_in_pic_flag</b>	u(1)
si( RapPicFlag )	
<b>no_output_of_prior_pics_flag</b>	u(1)
<b>pic_parameter_set_id</b>	ue(v)
si( !first_slice_in_pic_flag )	
<b>slice_address</b>	u(v)
si( dependent_slice_enabled_flag && !first_slice_in_pic_flag )	
<b>dependent_slice_flag</b>	u(1)
si( dependent_slice_flag == 0 ) slice_header_info	
si( tiles_or_entropy_coding_sync_idc == 1    tiles_or_entropy_coding_sync_idc == 2 ) {	
<b>num_entry_point_offsets</b>	ue(v)
si( num_entry_point_offsets > 0 ) {	
<b>offset_len_minus1</b>	ue(v)
para( i = 0; i < num_entry_point_offsets; i++ )	
<b>entry_point_offset[ i ]</b>	u(v)
}	
}	
}	

Bandera de porción independiente

601

Número de flujos CABAC en la porción

602

Número de bytes del subflujo (la navegación es más fácil en el flujo de bits)

603

FIG. 7

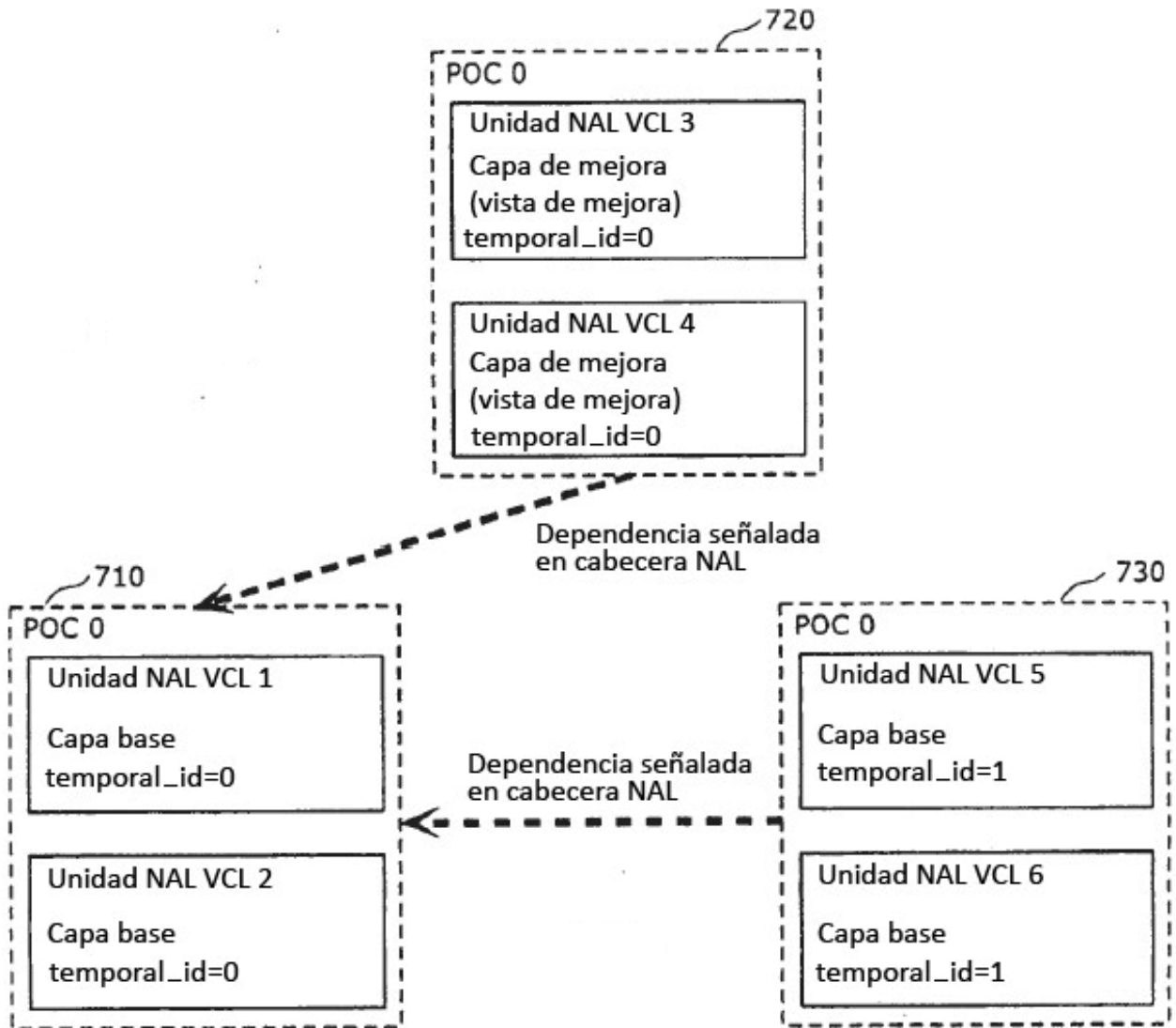


FIG. 8

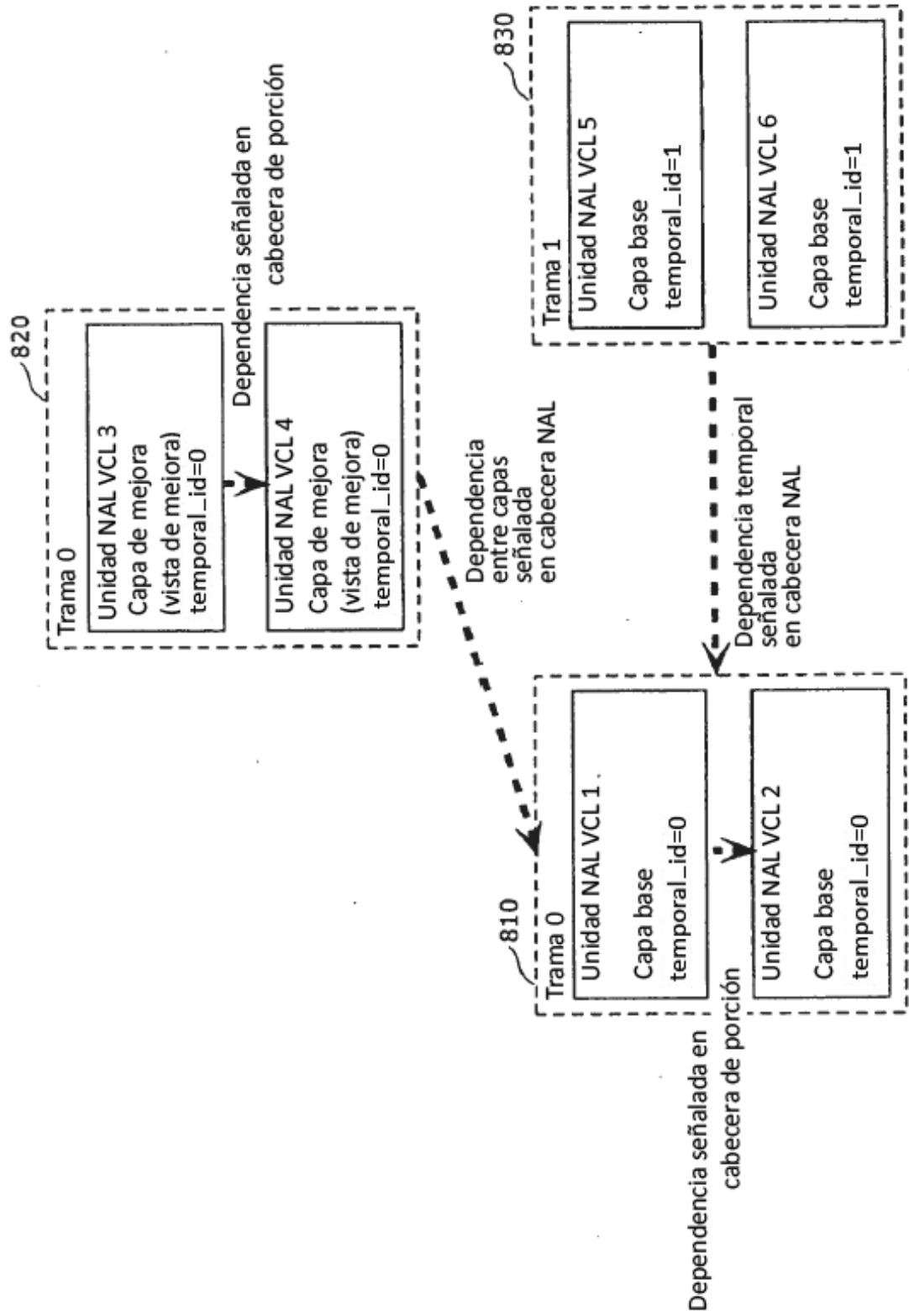


FIG. 9A

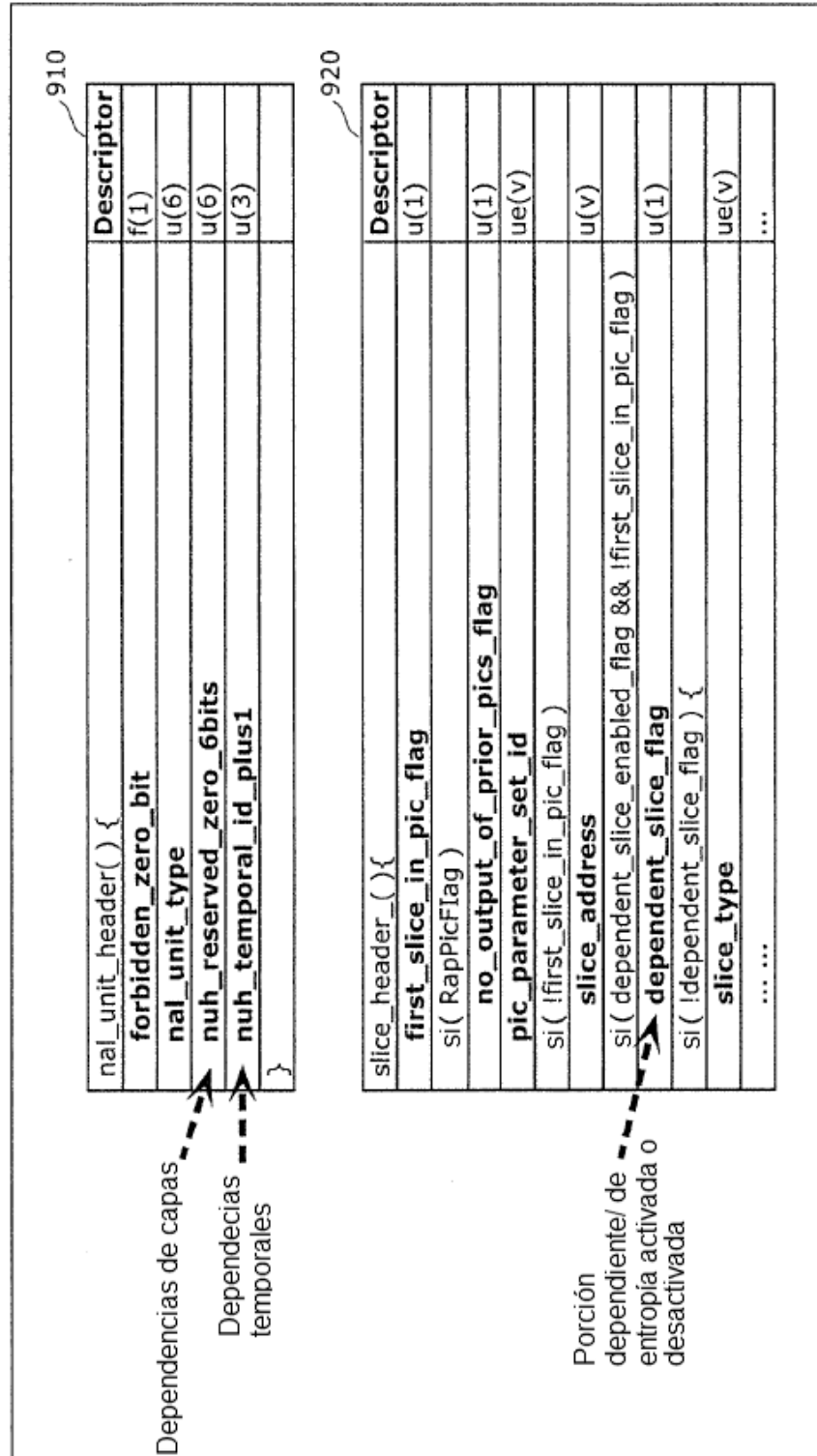




FIG. 9B

	Descriptor
seq_parameter_set_rbsp() {	
video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_reserved_zero_bit	u(1)
profile_and_level(1, sps_max_sub_layers_minus1)	
seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
si( chroma_format_idc == 3 )	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples ~931	ue(v)
pic_height_in_luma_samples ~932	ue(v)
pic_cropping_flag	u(1)
si( pic_cropping_flag ) {	
pic_crop_left_offset	ue(v)
pic_crop_right_offset	ue(v)
pic_crop_top_offset	ue(v)
pic_crop_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
pcm_enabled_flag	u(1)
si( pcm_enabled_flag ) {	
pcm_sample_bit_depth_luma_minus1	u(4)
pcm_sample_bit_depth_chroma_minus1	u(4)
}	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
para( i = 0; i <= sps_max_sub_layers_minus1; i++ ) {	
sps_max_dec_pic_buffering[ i ]	ue(v)
sps_max_num_reorder_pics[ i ]	ue(v)
sps_max_latency_increase[ i ]	ue(v)
}	
restricted_ref_pic_lists_flag	u(1)
si( restricted_ref_pic_lists_flag )	
lists_modification_present_flag	u(1)
log2_min_coding_block_size_minus3 ~933	ue(v)
log2_diff_max_min_coding_block_size ~934	ue(v)
... ... ... }	

930

935

```

slice_header_( ) {
  first_slice_in_pic_flag
  si( RapPicFlag )
  no_output_of_prior_pics_flag
  pic_parameter_set_id
  si( !first_slice_in_pic_flag )
  slice_address
  si( dependent_slice_enabled_flag && !first_slice_in_pic_flag )
  dependent_slice_flag
  si( !dependent_slice_flag ) {
    slice_type
    ....
  }
}
    
```

920

Para analizar slice\_address se necesitan elementos de sintaxis del SPS. Por lo tanto, SPS necesita ser analizado hasta este punto.

FIG. 9C

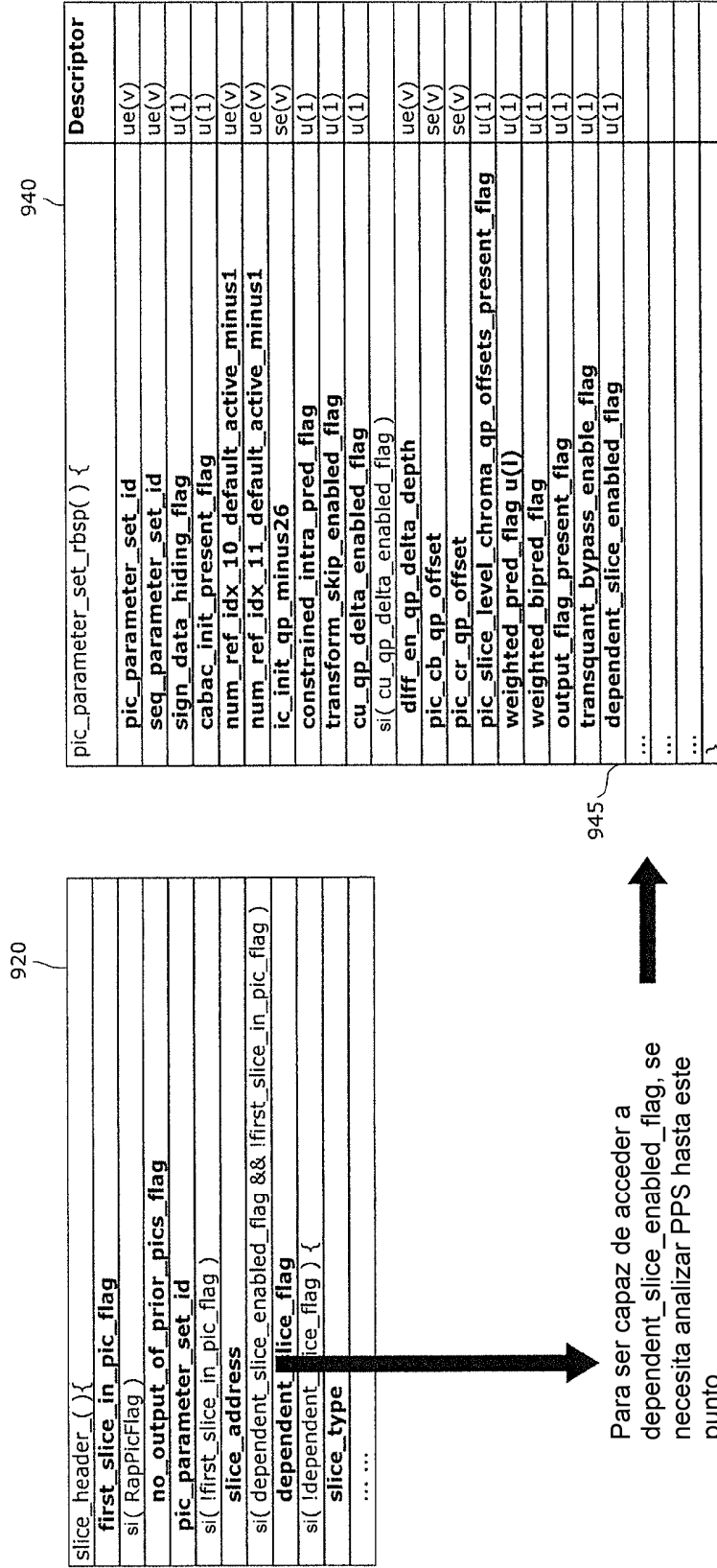


FIG. 10

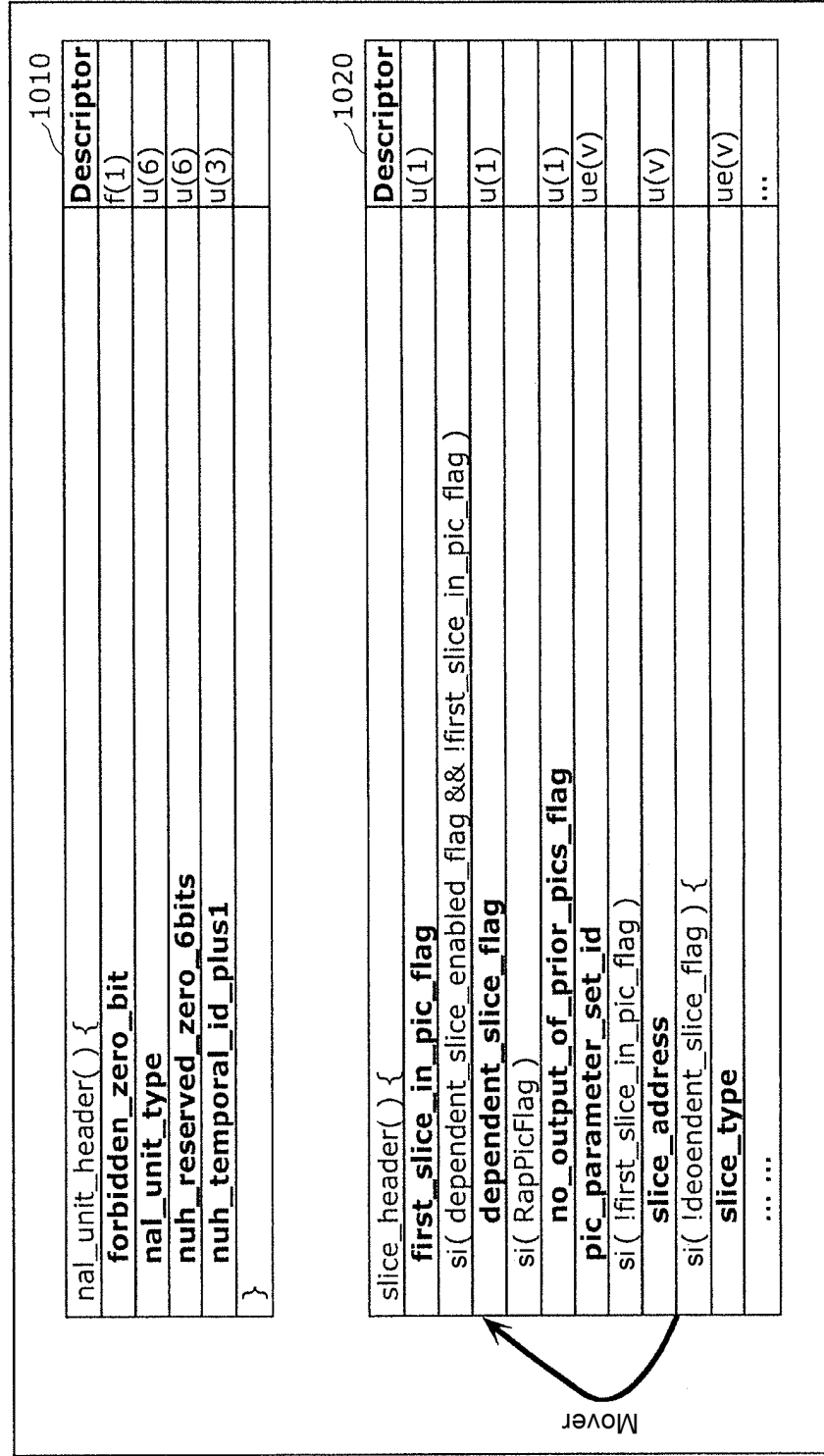


FIG. 11

	1110	<b>Descriptor</b>
<code>slice_header() {</code>		<code>u(1)</code>
<code><b>first_slice_in_pic_flag</b></code>		
<code>si( <del>dependent_slice_enabled_flag</del> &amp;&amp; !first_slice_in_pic_flag )</code>		
<code><b>dependent_slice_flag</b></code>		<code>u(1)</code>
<code>si( RapPicFlag )</code>		
<code><b>no_output_of_prior_pics_flag</b></code>		<code>u(1)</code>
<code><b>pic_parameter_set_id</b></code>		<code>ue(v)</code>
<code>si( !first_slice_in_pic_flag )</code>		
<code><b>slice_address</b></code>		<code>u(v)</code>
<code>si( !dependent_slice_flag ) {</code>		
<code><b>slice_type</b></code>		<code>ue(v)</code>
<code>... ..</code>		<code>...</code>

FIG. 12

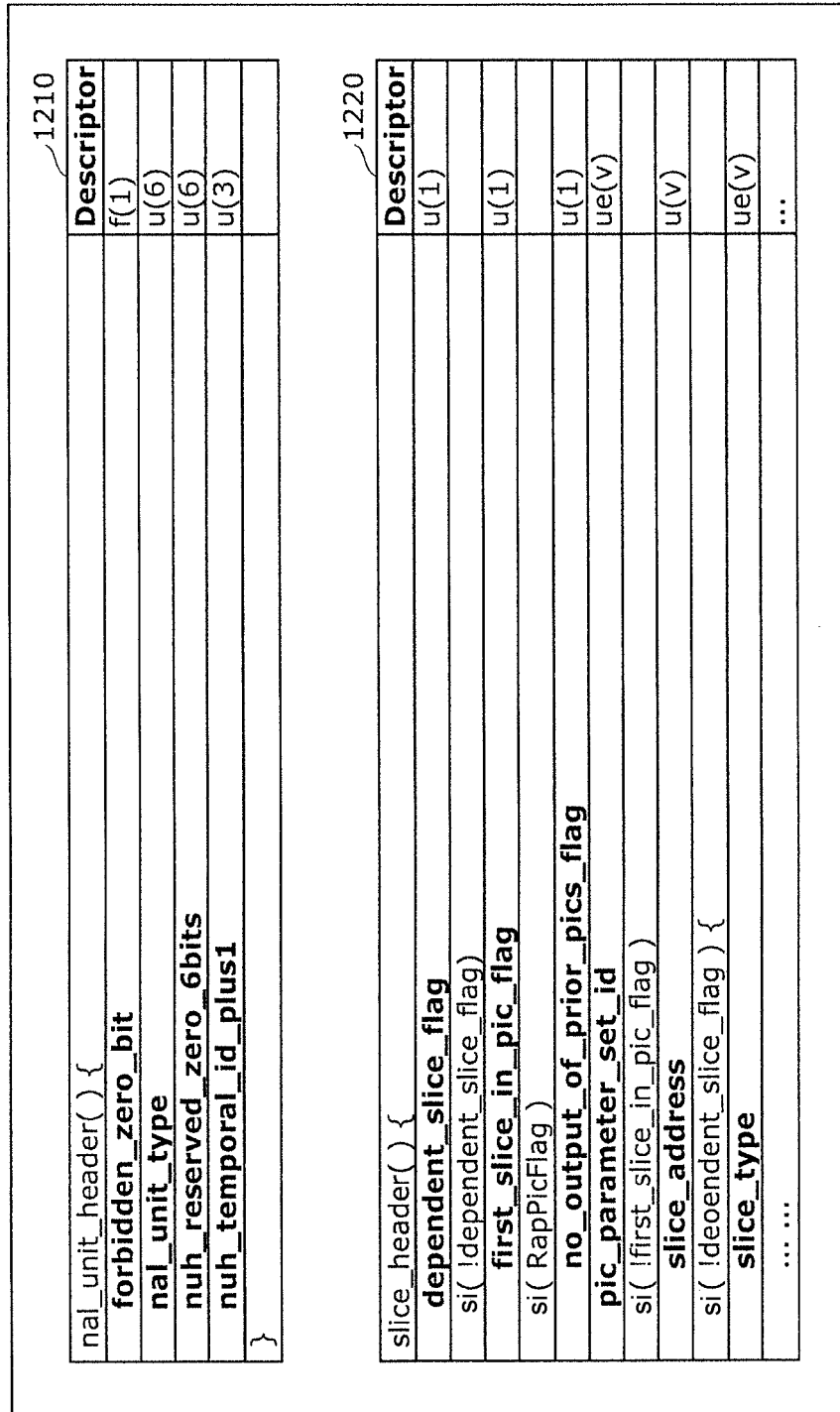


FIG. 13

slice_header ( ) {	
first_slice_in_pic_flag	u(1)
si( RapPicFlag )	
no_output_of_prior_pics_flag	u(1)
pic_parameter_set_id	ue(v)
si( dependent_slice_enabled_flag && !first_slice_in_pic_flag )	
dependent_slice_flag	u(1)
si( !first_slice_in_pic_flag )	
slice_address	u(v)
si( !dependent_slice_flag ) {	
slice_type	ue(v)
... ..	...

1310

FIG. 14

nal_unit_header() {	
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
nuh_reserved_zero_6bits	u(5)
nuh_temporal_id_plus1	u(3)
dependent_slice_flag	u(1)
}	

1410

FIG. 15

1510	
nal_unit_header() {	<b>Descriptor</b>
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
nuh_reserved_zero_6bits	u(6)
nuh_temporal_id_plus1	u(3)
}	
1520	
slice_header() {	<b>Descriptor</b>
first_slice_in_pic_flag	u(1)
si( RapPicFlag )	
no_output_of_prior_pics_flag	u(1)
pic_parameter_set_id	ue(v)
si( !first_slice_in_pic_flag )	
<b>slice_address</b>	u(v)
si(nal_unit_type == 15    nal_unit_type == 16)	
dependent_slice_flag == 1	
si( !dependent_slice_flag ) {	
<b>slice_type</b>	ue(v)
... ..	...



FIG. 16

1610	
	Descriptor
<code>nal_unit_header() {</code>	
<code>forbidden_zero_bit</code>	<code>f(1)</code>
<code>nal_unit_type</code>	<code>u(6)</code>
<code>nuh_reserved_zero_6bits</code>	<code>u(6)</code>
<code>nuh_temporal_id_plus1</code>	<code>u(3)</code>
<code>}</code>	

1620	
	Descriptor
<code>slice_header() {</code>	
<code>first_slice_in_pic_flag</code>	<code>u(1)</code>
<code>si( RapPicFlag )</code>	
<code>no_output_of_prior_pics_flag</code>	<code>u(1)</code>
<code>pic_parameter_set_id</code>	<code>ue(v)</code>
<code>si( !first_slice_in_pic_flag )</code>	
<code>slice_address</code>	<code>u(v)</code>
<code>si( dependent_slice_enabled_flag &amp;&amp; !first_slice_in_pic_flag )</code>	
<code>dependent_slice_flag</code>	<code>u(1)</code>
<code>si( !dependent_slice_flag ) {</code>	
<code>slice_type</code>	<code>ue(v)</code>
<code>... ..</code>	<code>...</code>

FIG. 17

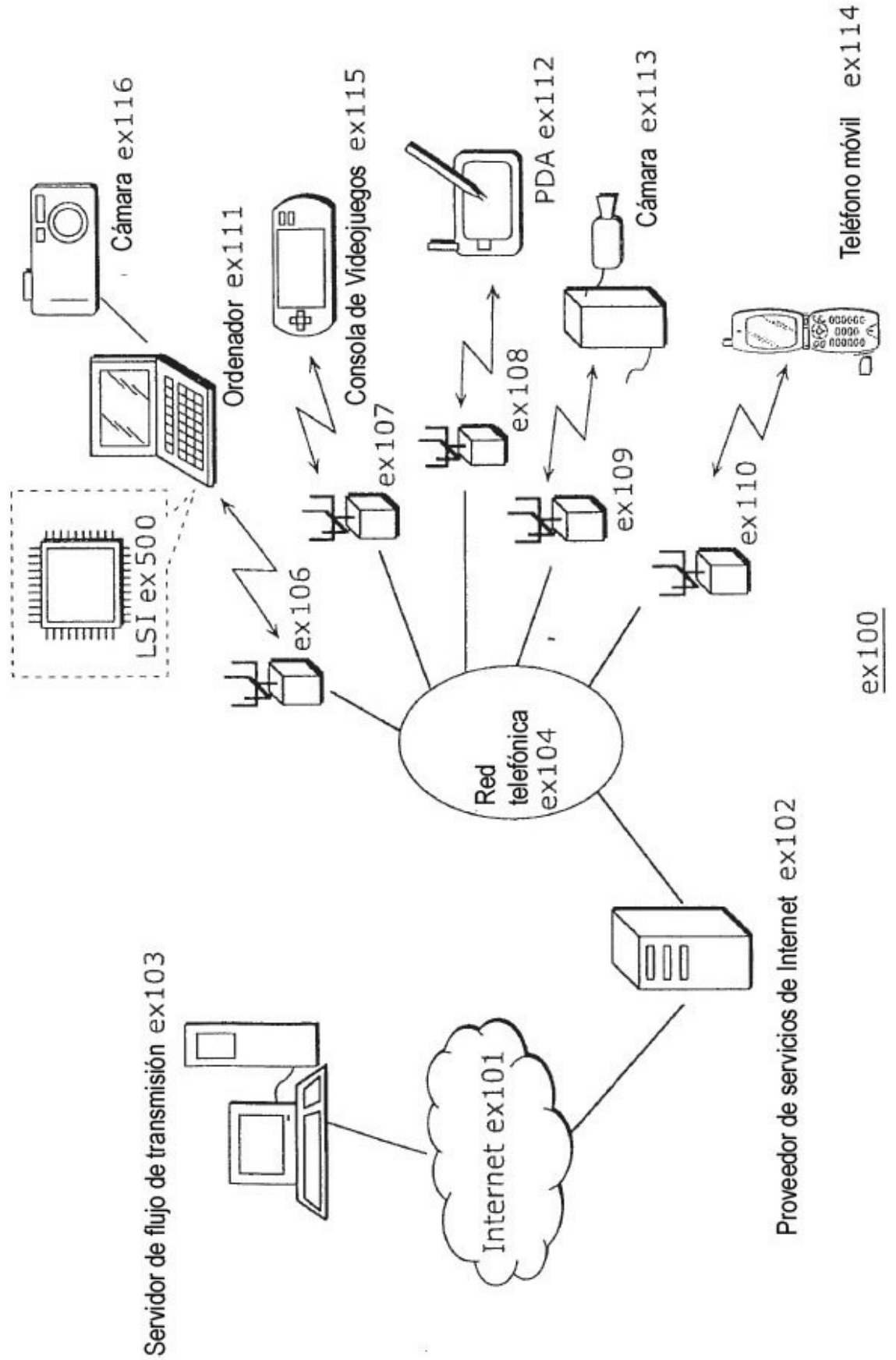


FIG. 18

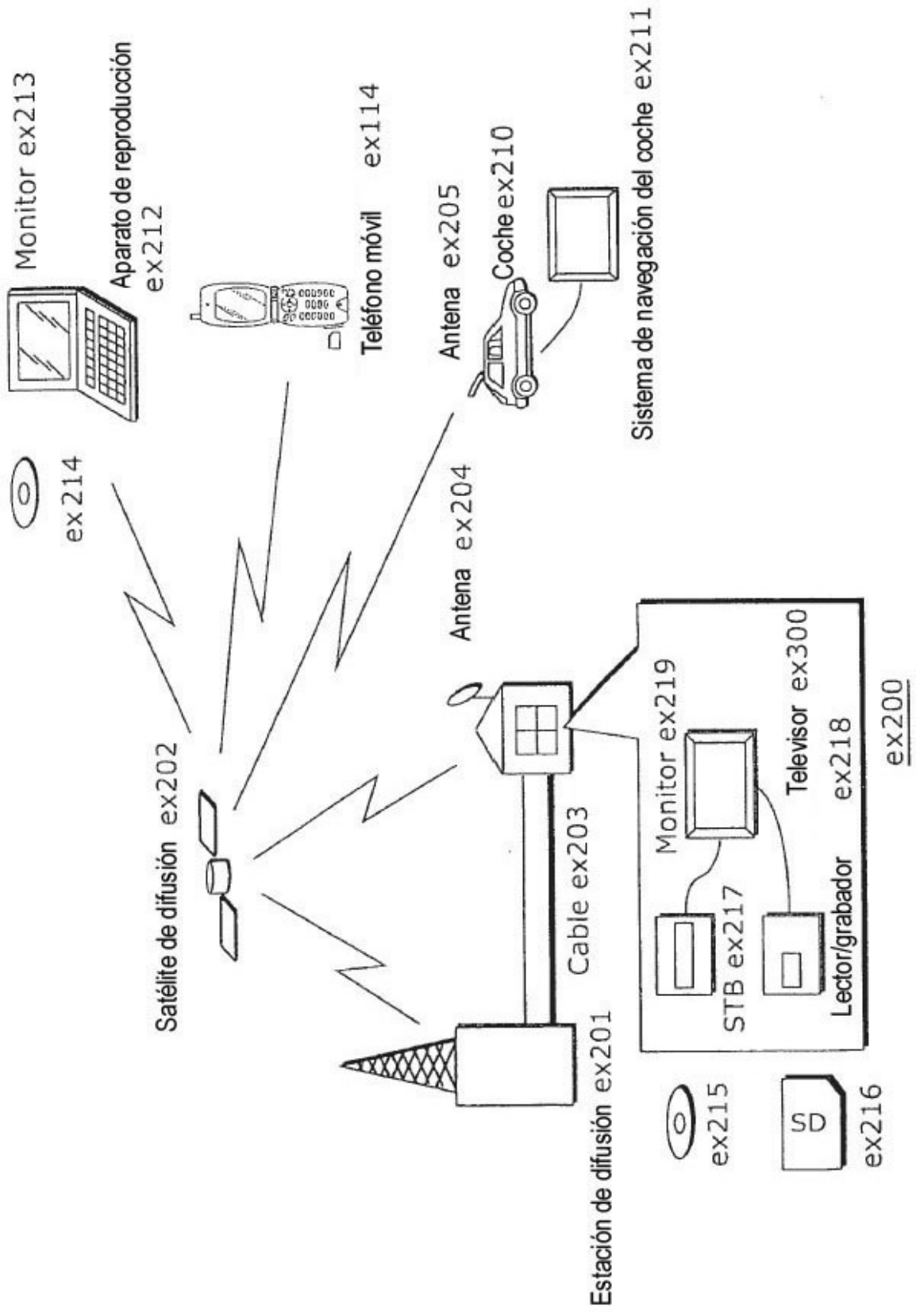


FIG. 19

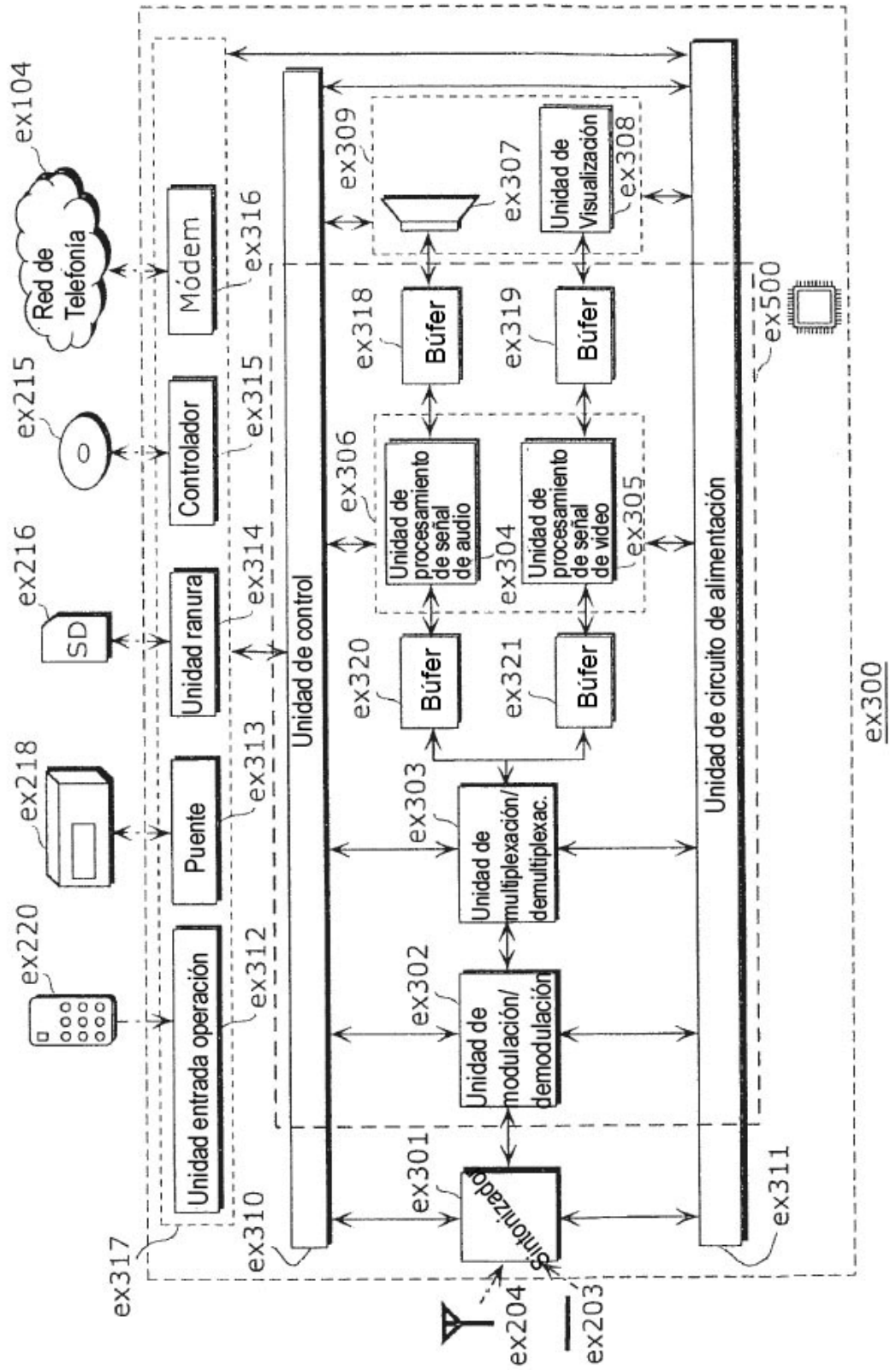


FIG. 20

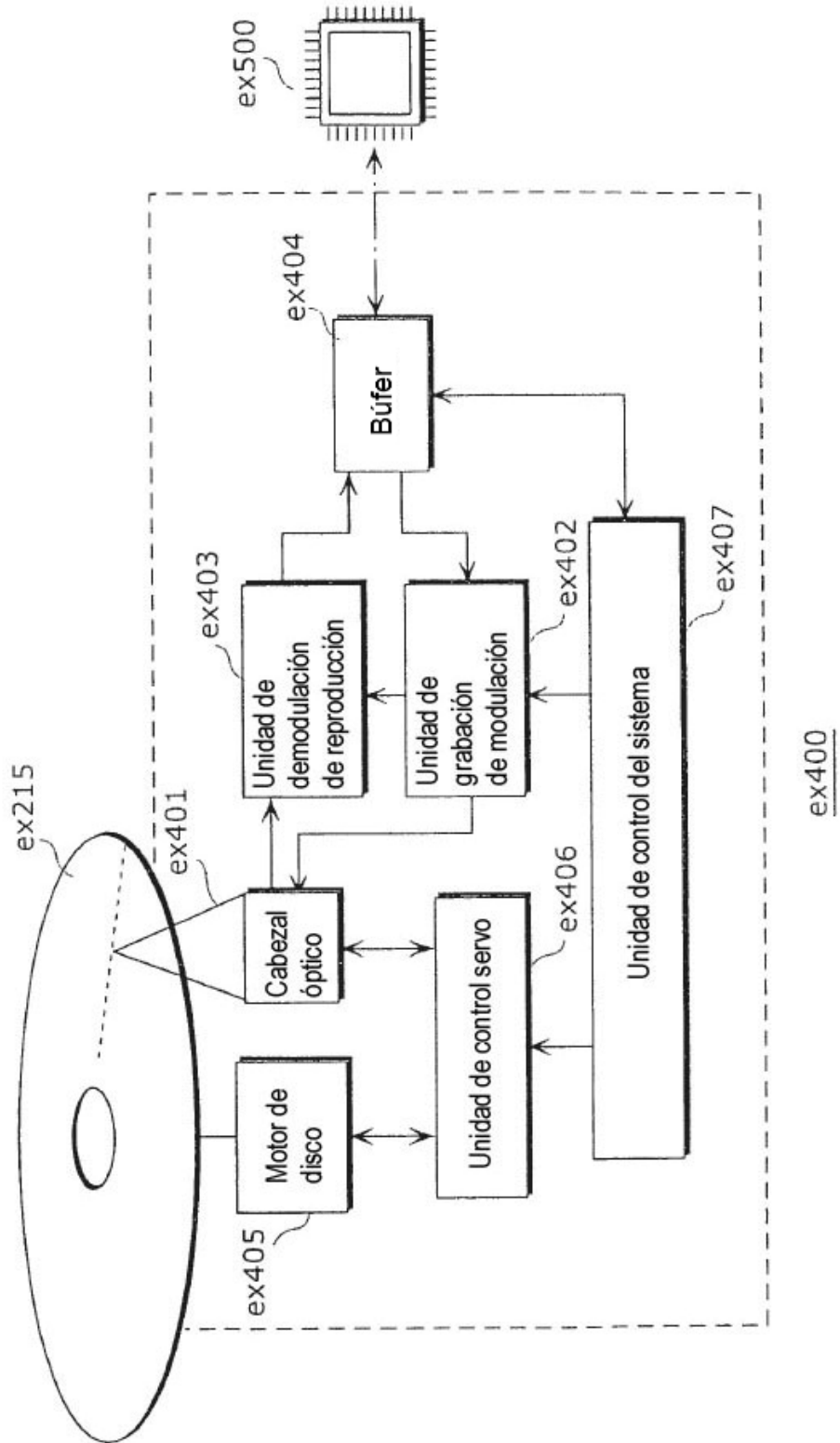


FIG. 21

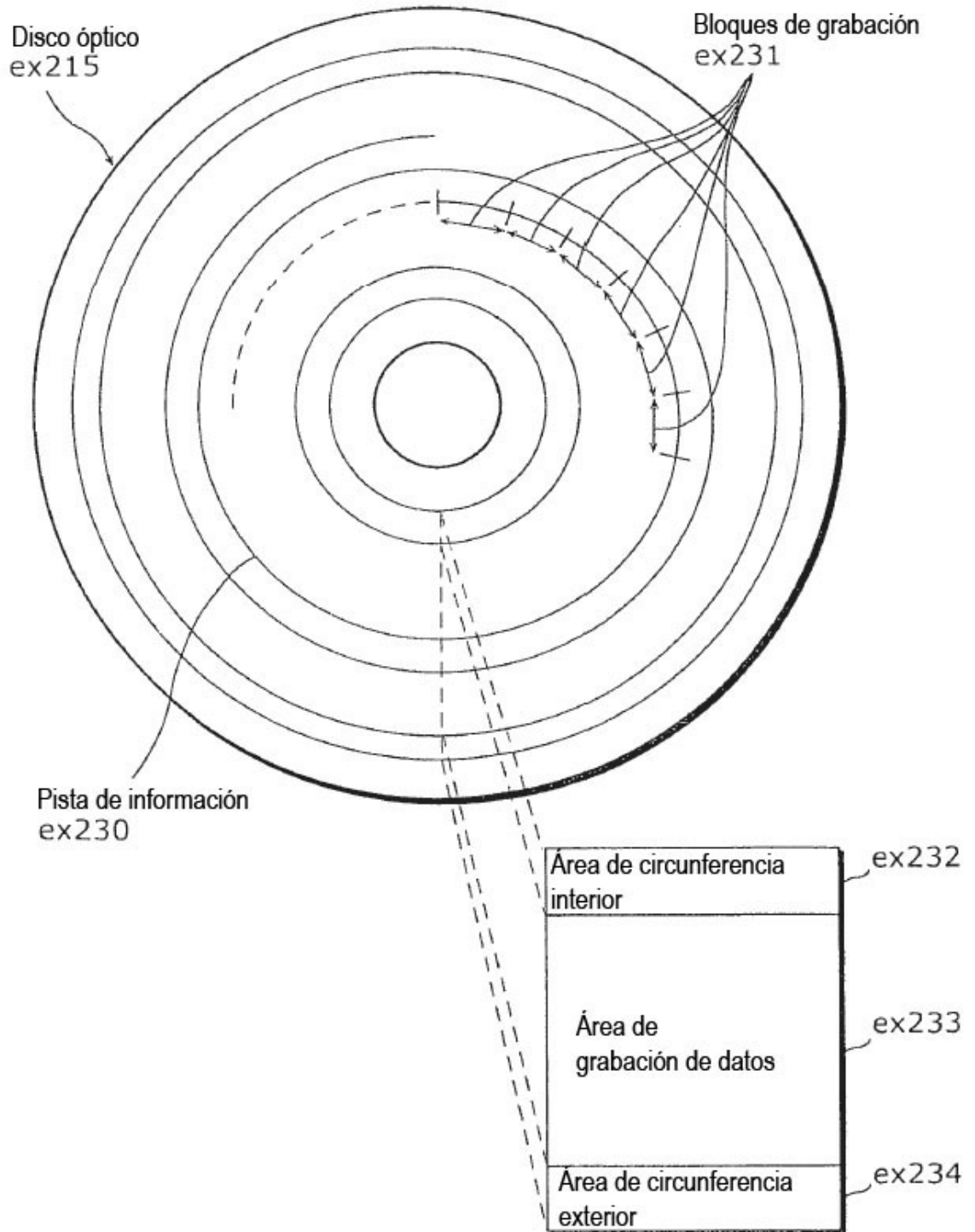


FIG. 22A

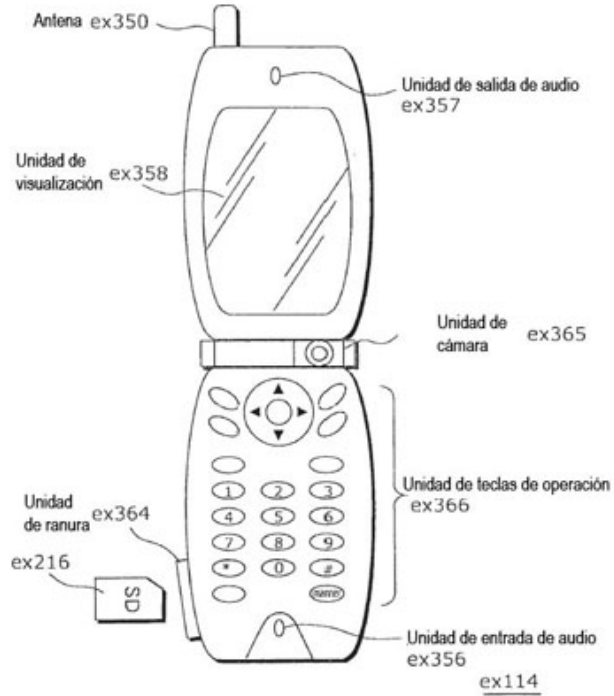


FIG. 22B

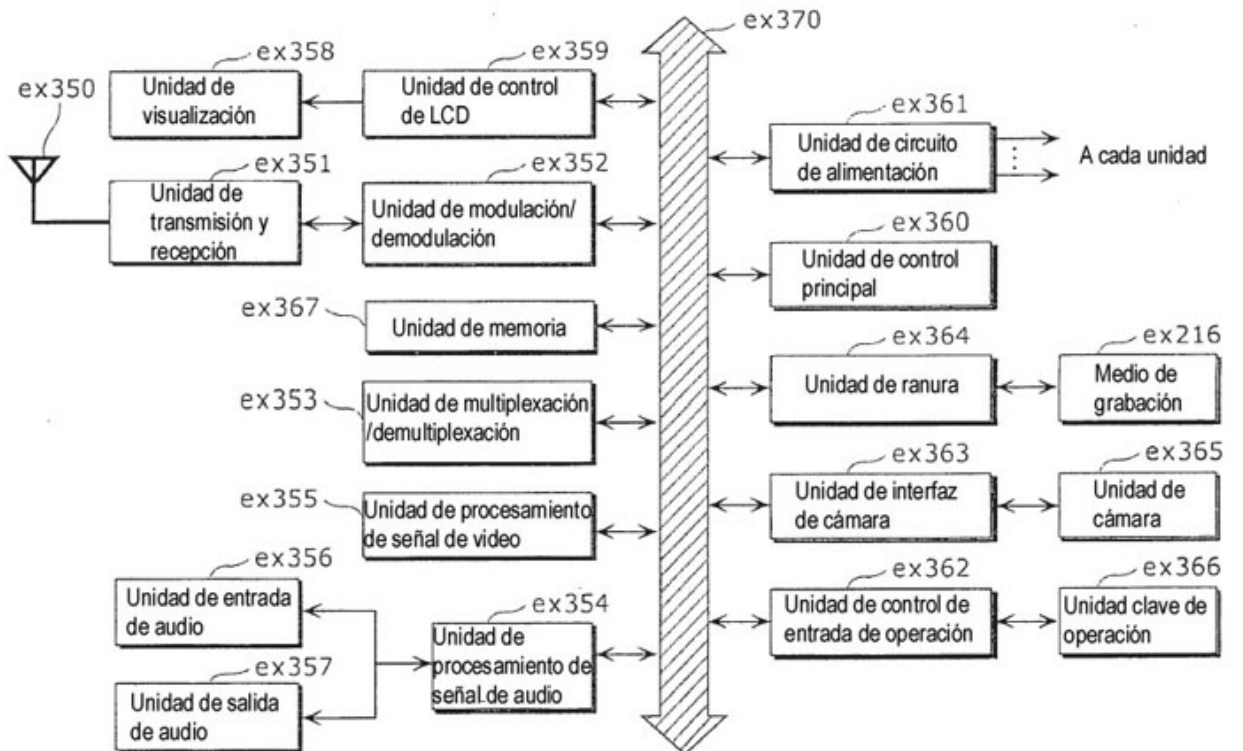


FIG. 23

Flujo de video (PID=0x1011, Video primario )
Flujo de audio (PID=0x1100)
Flujo de audio (PID=0x1101)
Flujo de gráficos de presentación (PID=0x1200)
Flujo de gráficos de presentación (PID=0x1201)
Flujo de gráficos interactivos (PID=0x1400)
Flujo de video (PID=0x1B00, Video secundario )
Flujo de video (PID=0x1B01, Video secundario )



FIG. 24

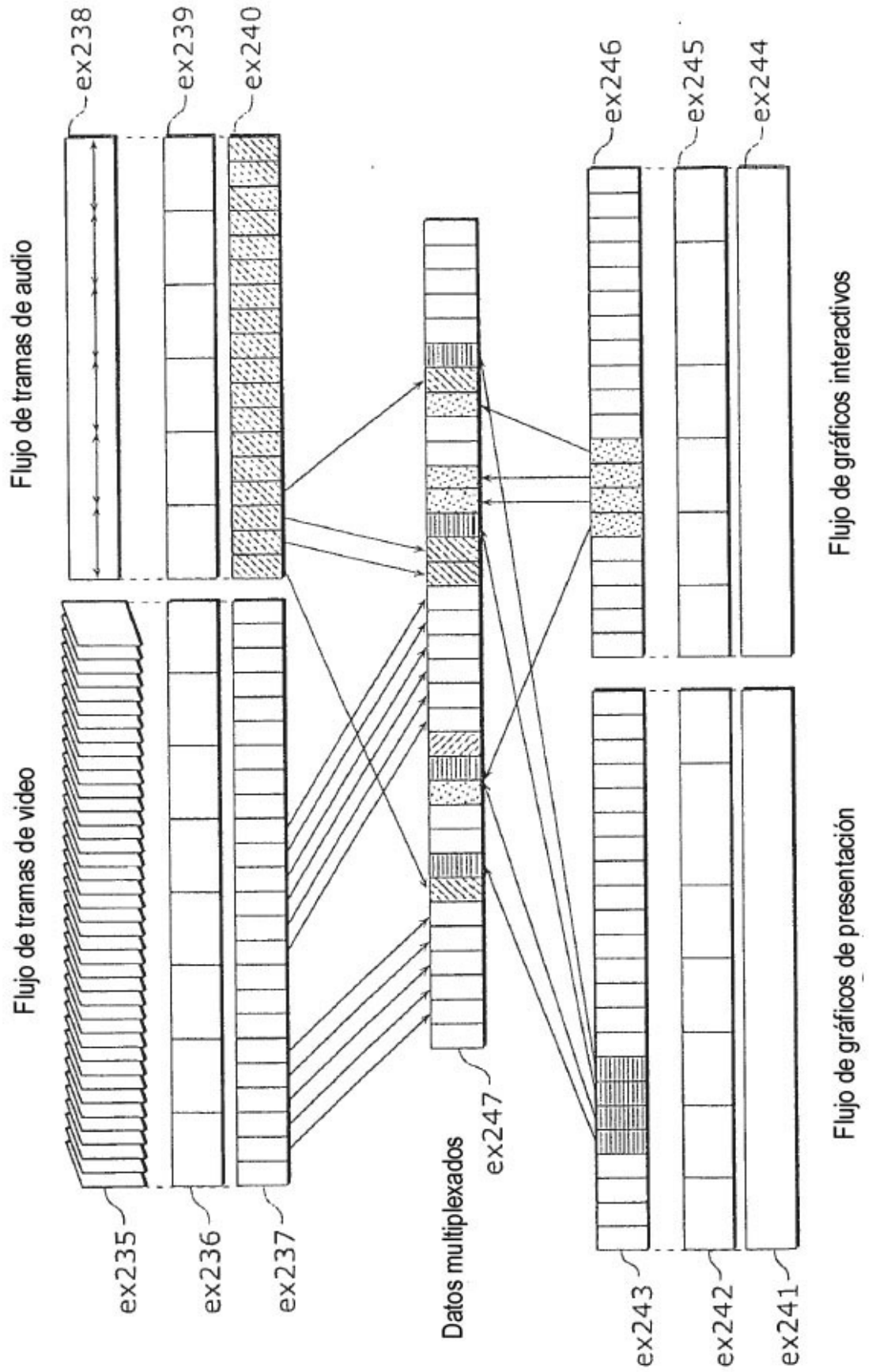


FIG. 25

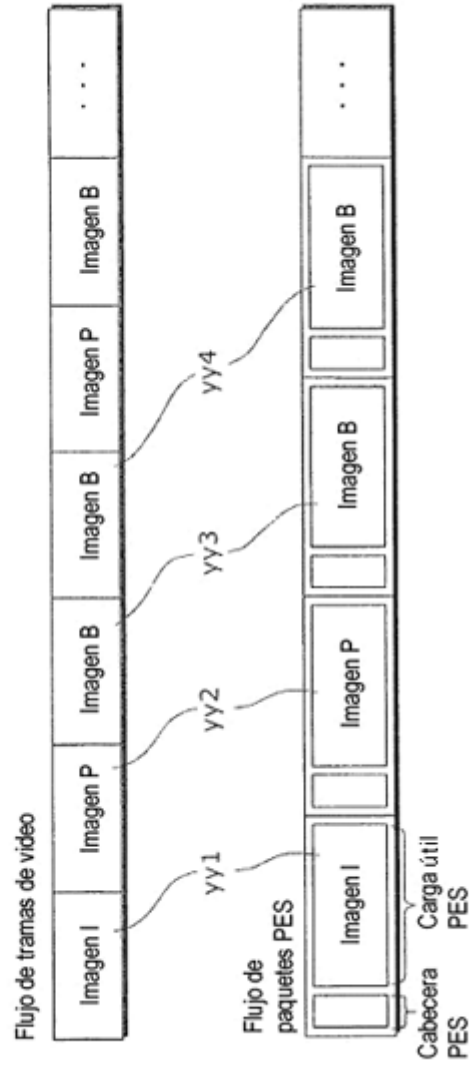


FIG. 26

Flujo de paquetes TS

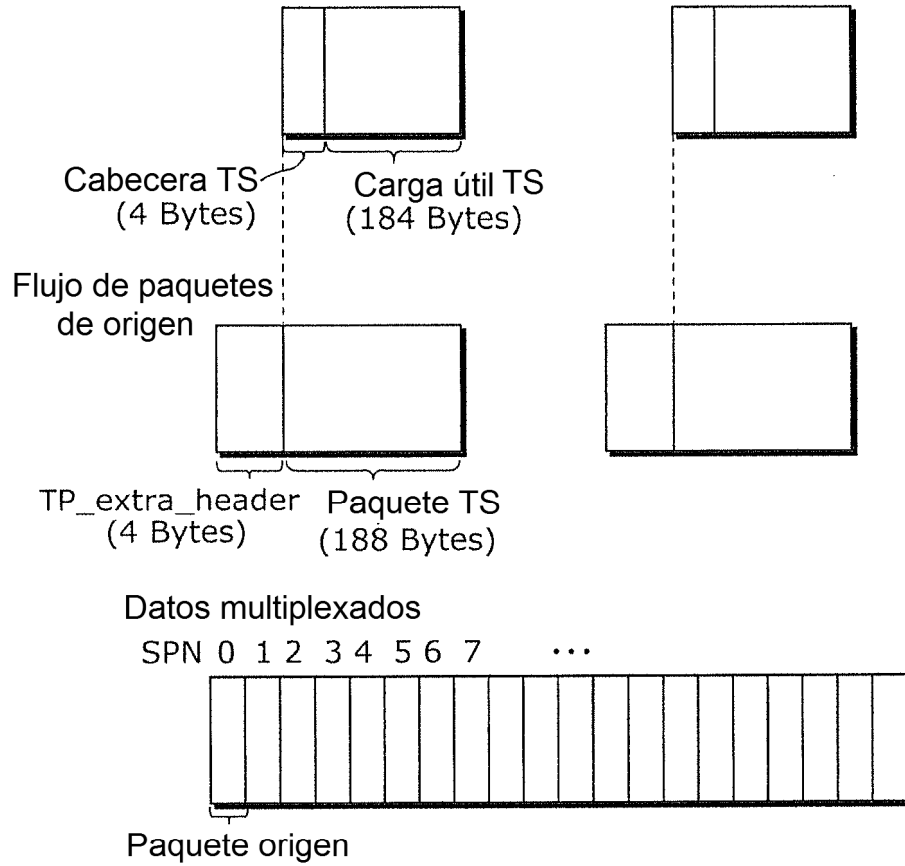


FIG. 27

Estructura de datos de la PMT

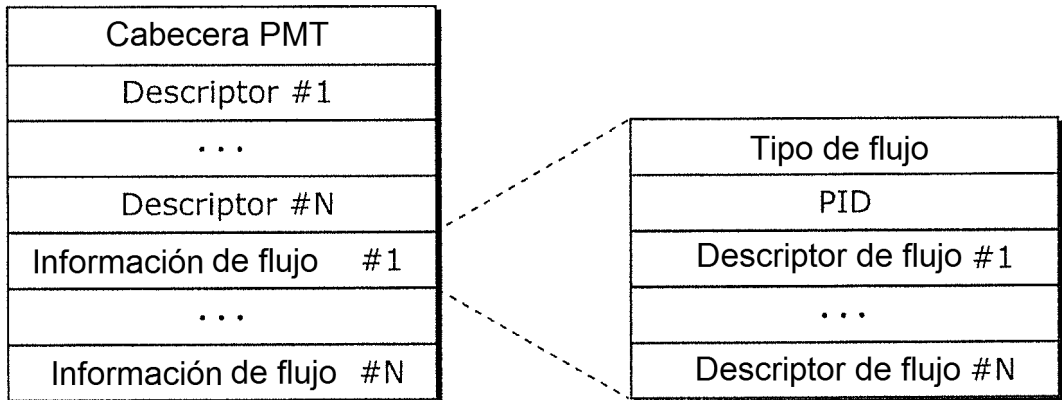


FIG. 28

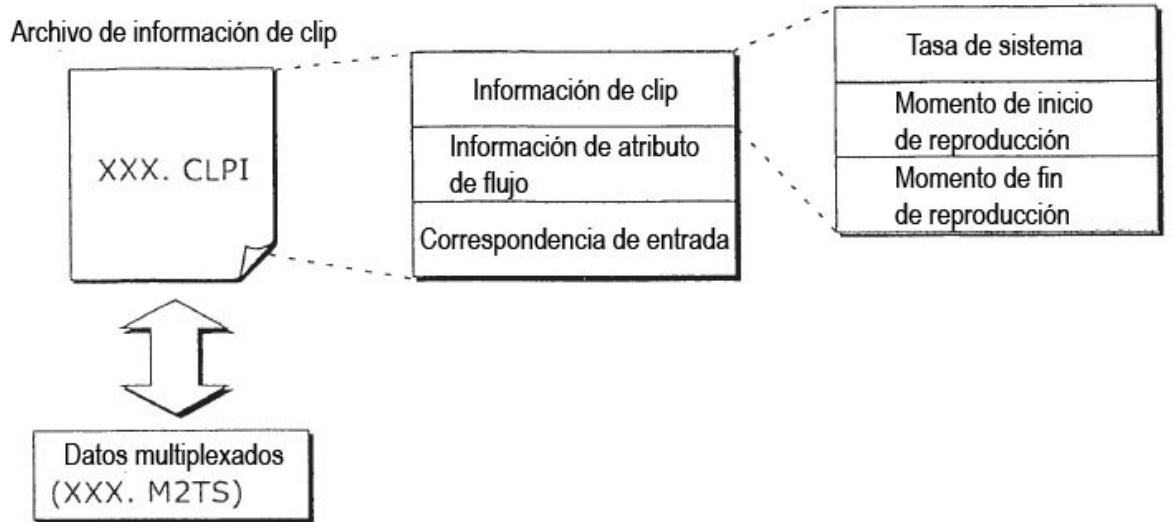


FIG. 29

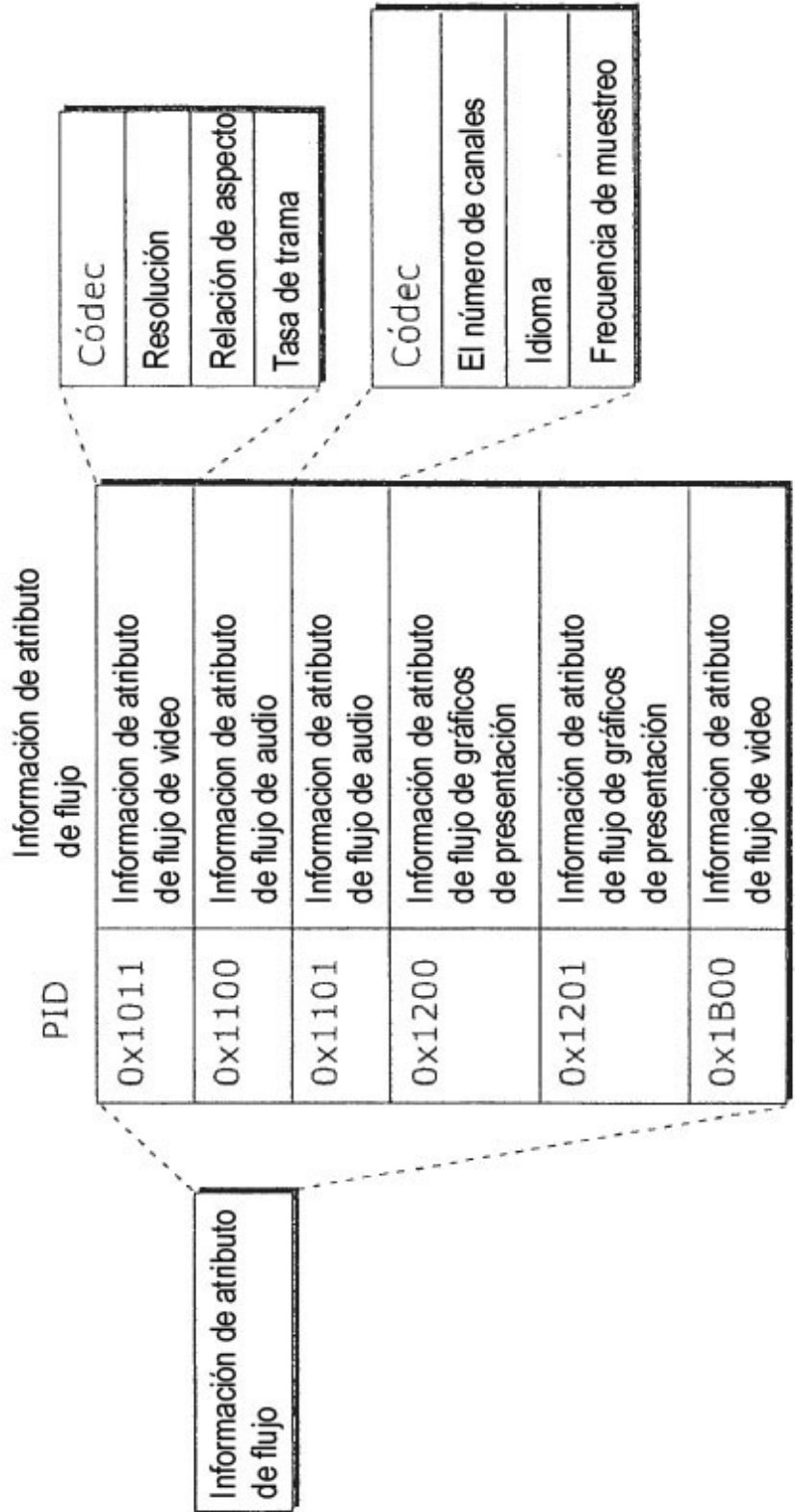


FIG. 30

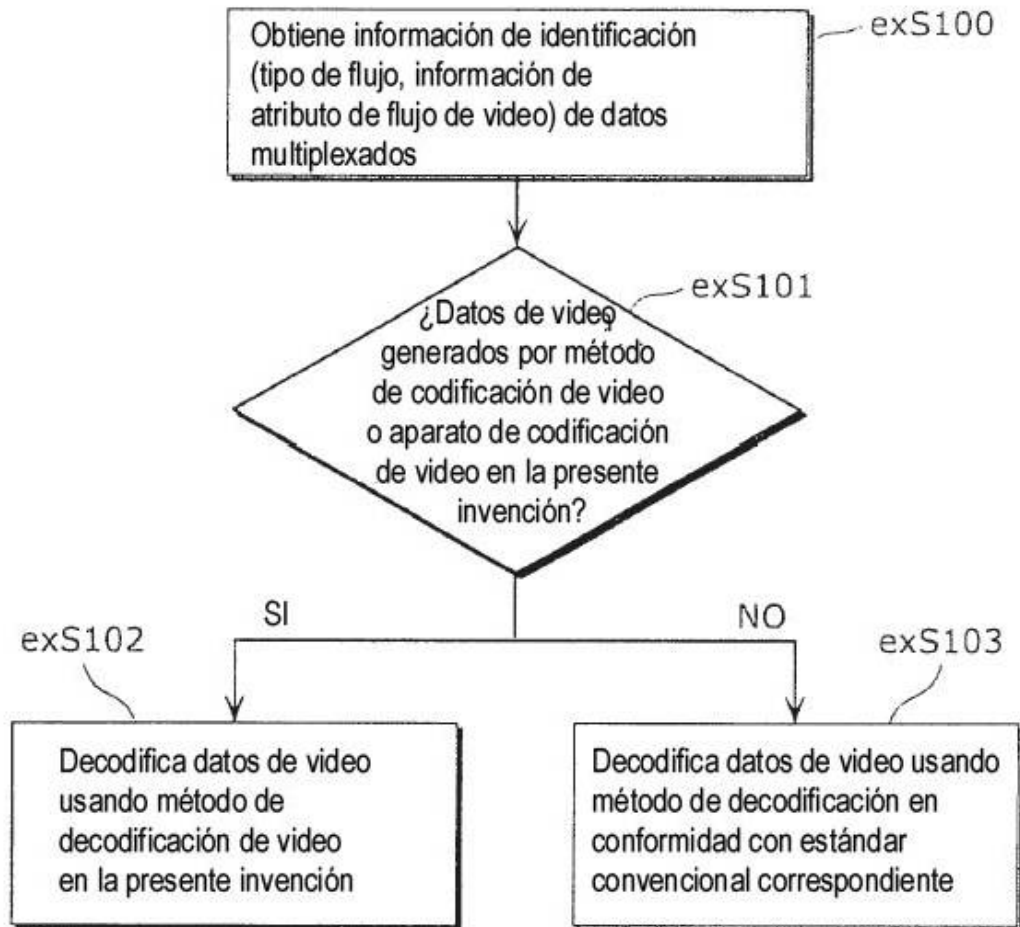


FIG. 31

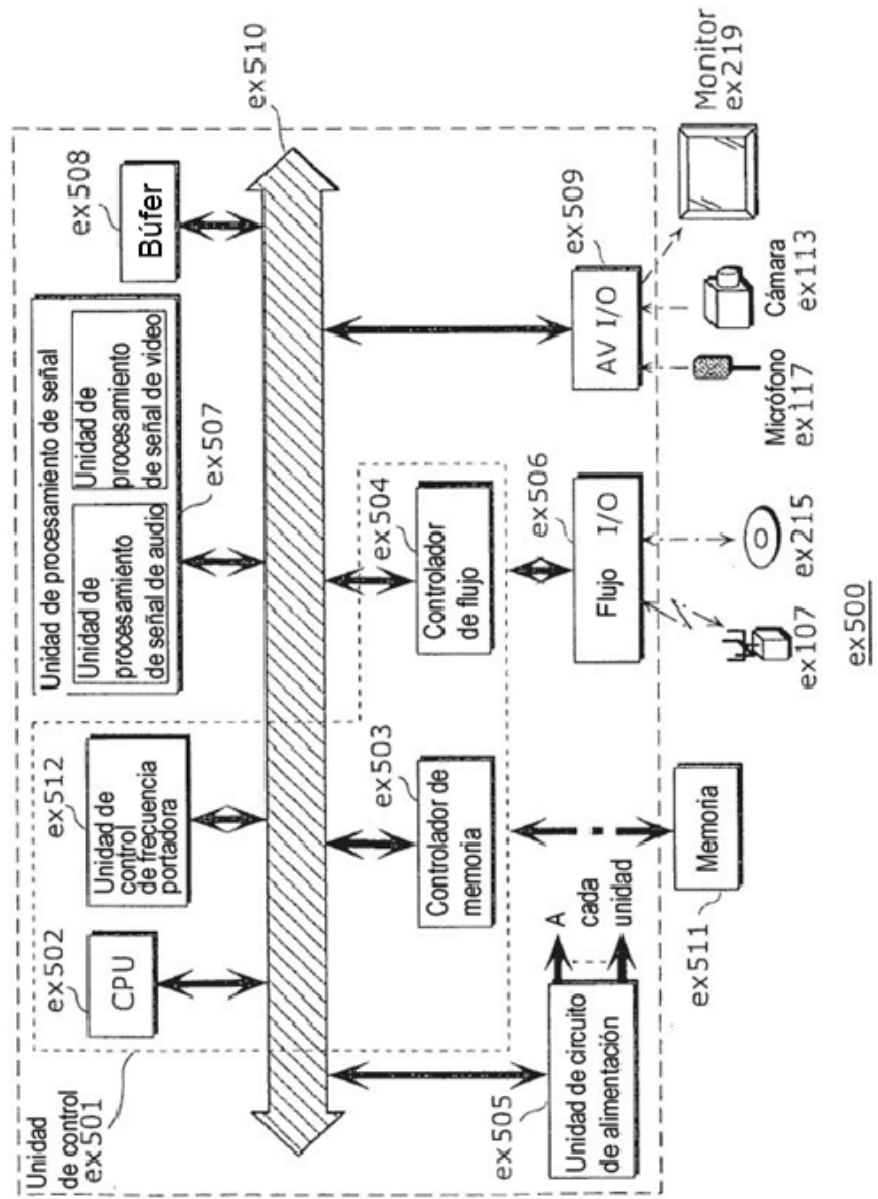


FIG. 32

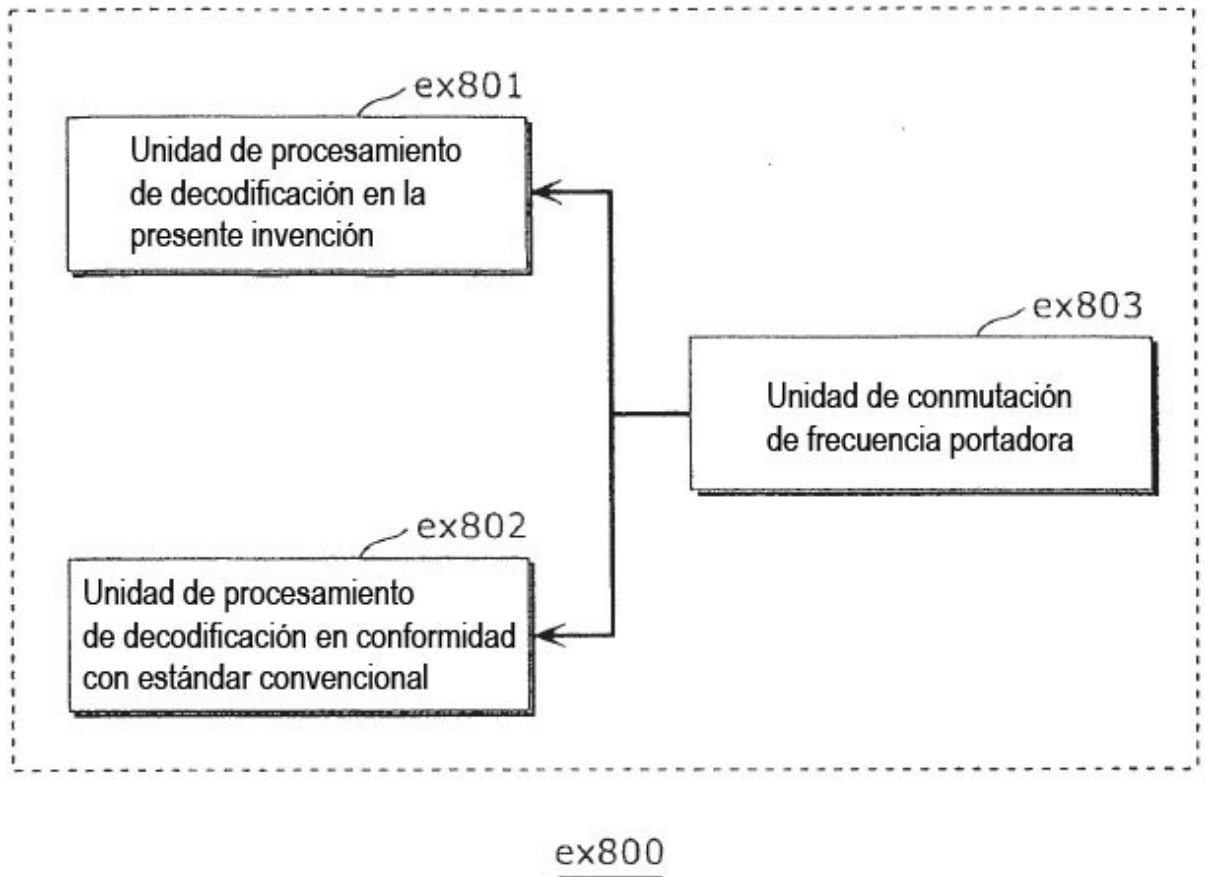




FIG. 33

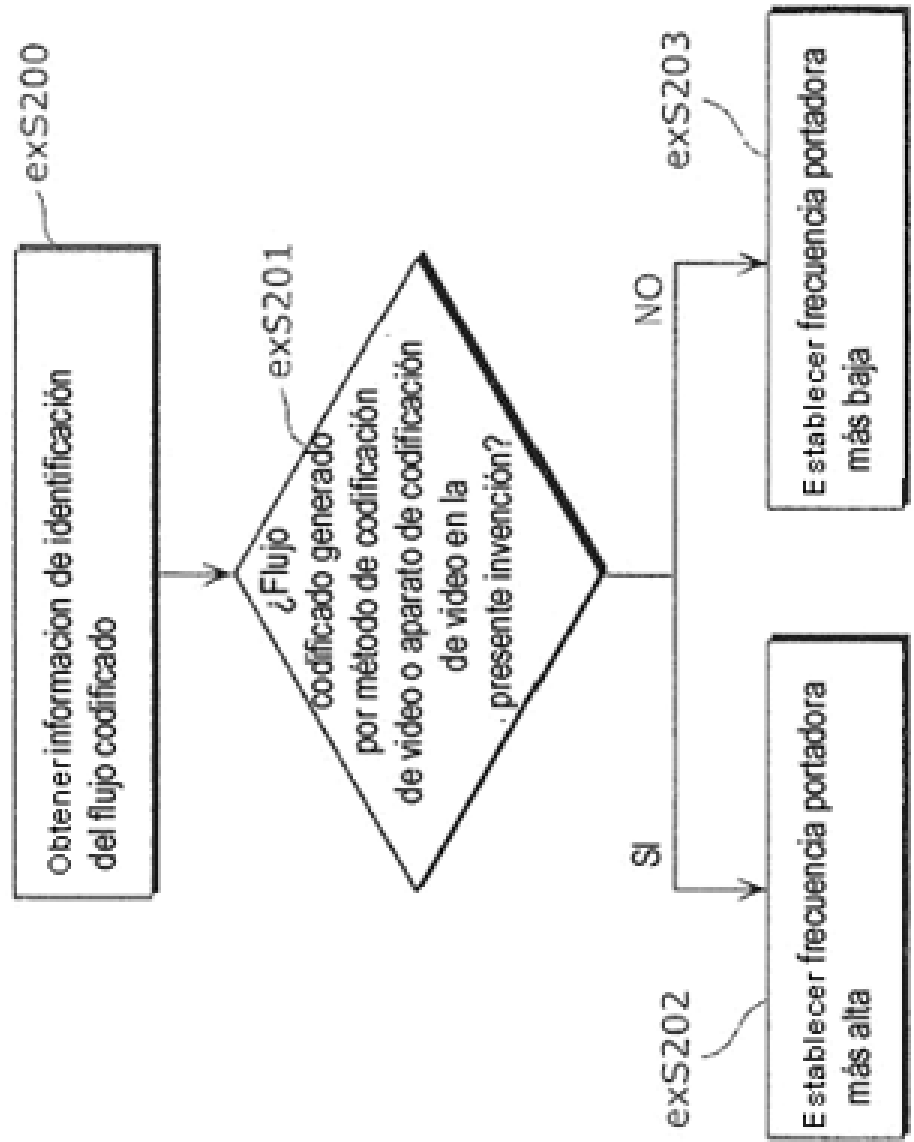


FIG. 34

Estándar correspondiente	Frecuencia portadora
MPEG-4 AVC	500 MHz
MPEG-2	350 MHz
⋮	⋮

FIG. 35A

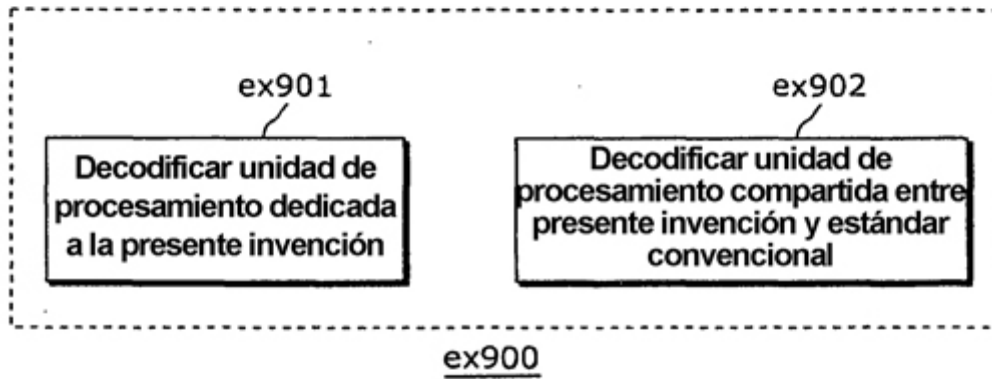


FIG. 35B

