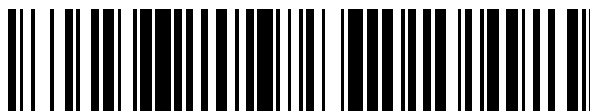


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 686 699**

51 Int. Cl.:

**H04N 21/2343** (2011.01)  
**H04N 19/44** (2014.01)  
**H04N 19/187** (2014.01)  
**H04N 21/45** (2011.01)  
**H04N 21/81** (2011.01)  
**H04N 19/70** (2014.01)  
**H04N 19/66** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **01.05.2015 PCT/US2015/028841**
- 87 Fecha y número de publicación internacional: **05.11.2015 WO15168591**
- 96 Fecha de presentación y número de la solicitud europea: **01.05.2015 E 15722859 (4)**
- 97 Fecha y número de publicación de la concesión europea: **13.06.2018 EP 3138290**

54 Título: **Procedimiento y dispositivo para descodificar datos de vídeo de múltiples capas determinando la capacidad del descodificador basándose en el perfil, grado y nivel asociados con la partición que contiene una o más capas**

30 Prioridad:

**01.05.2014 US 201461987463 P**  
**15.05.2014 US 201461993980 P**  
**30.04.2015 US 201514701004**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**19.10.2018**

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)**  
**Attn: International IP Administration, 5775**  
**Morehouse Drive**  
**San Diego, CA 92121-1714 , US**

72 Inventor/es:

**RAMASUBRAMONIAN, ADARSH KRISHNAN;**  
**WANG, YE-KUI;**  
**HENDRY, FNU y**  
**RAPAKA, KRISHNAKANTH**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

ES 2 686 699 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

Procedimiento y dispositivo para descodificar datos de vídeo de múltiples capas determinando la capacidad del decodificador basándose en el perfil, grado y nivel asociados con la partición que contiene una o más capas

## CAMPO TÉCNICO

[0001] Esta divulgación se refiere a la codificación y descodificación de vídeo.

## ANTECEDENTES

[0002] Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo digitales implementan técnicas de compresión de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), el estándar de Codificación de Vídeo de Alta Eficacia (HEVC) actualmente en desarrollo y las ampliaciones de dichos estándares. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de compresión de vídeo.

[0003] Las técnicas de compresión de vídeo realizan la predicción espacial (intra-imagen) y/o la predicción temporal (entre imágenes) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (es decir, una trama de vídeo o una parte de una trama de vídeo) puede dividirse en bloques de vídeo. Los bloques de vídeo en un fragmento intra-codificado (I) de una imagen se codifican mediante predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen. Los bloques de vídeo en un fragmento inter-codificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen, o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia.

[0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intra-codificado se codifica de acuerdo con un modo de intra-codificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio del píxel a un dominio de transformación, dando como resultado coeficientes residuales, los cuales pueden cuantificarse posteriormente. Los coeficientes cuantificados, inicialmente dispuestos en una formación bidimensional, pueden recorrerse con el fin de producir un vector unidimensional de coeficientes, y puede aplicarse la codificación por entropía para lograr aún más compresión.

[0005] Un flujo de bits de codificación de múltiples visualizaciones puede generarse codificando visualizaciones, por ejemplo, desde múltiples perspectivas. Se han desarrollado algunos estándares de vídeo tridimensionales (3D) que hacen uso de aspectos de codificación de múltiples visualizaciones. Por ejemplo, diferentes visualizaciones pueden transmitir visualizaciones de ojo izquierdo y derecho para soportar vídeo 3D. De forma alternativa, algunos procesos de codificación de vídeo 3D pueden aplicar la denominada codificación de múltiples visualizaciones más profundidad. En la codificación de múltiples visualizaciones más profundidad, un flujo de bits de vídeo 3D puede contener no solo componentes de visualización de textura, sino también componentes de visualización de profundidad. Por ejemplo, cada visualización puede comprender un componente de visualización de textura y un componente de visualización de profundidad.

## RESUMEN

[0006] En general, esta divulgación describe técnicas de descripción de propiedades de flujo de bits de vídeo usadas para determinar, basándose en el perfil, grado y nivel de información especificada para las respectivas particiones de flujo de bits de vídeo, la capacidad de un decodificador de vídeo para descodificar un flujo de bits de vídeo.

[0007] En otro ejemplo, esta divulgación describe un procedimiento de descodificación de datos de vídeo de múltiples capas, comprendiendo el procedimiento: determinar, basándose en una lista de entradas triples, si un decodificador de vídeo de múltiples capas es capaz de descodificar un flujo de bits que comprende una representación codificada de los datos de vídeo de múltiples capas, en el que cada entrada triple respectiva en la lista de entradas triples indica un perfil, un grado y un nivel para un decodificador de capa única respectivo en el

descodificador de vídeo de múltiples capas; y basándose en que el descodificador de vídeo de múltiples capas es capaz de descodificar el flujo de bits, descodificar, mediante el descodificador de vídeo de múltiples capas, el flujo de bits.

5 **[0008]** En otro ejemplo, esta divulgación describe un dispositivo para descodificar datos de vídeo de múltiples capas, comprendiendo el dispositivo: uno o más medios de almacenamiento legibles por ordenador configurados para almacenar los datos de vídeo de múltiples capas; y uno o más procesadores configurados para: determinar basándose en una lista de entradas triples, si un descodificador de múltiples capas es capaz de descodificar un flujo de bits que comprende una representación codificada de los datos de vídeo de múltiples capas, en el que cada  
10 entrada triple respectiva en la lista de las entradas triples indica un perfil, un grado y un nivel para un descodificador de capa única respectivo en el descodificador de vídeo de múltiples capas.

**[0009]** En otro ejemplo, esta divulgación describe un dispositivo para descodificar datos de vídeo, comprendiendo el dispositivo: medios para determinar, basándose en una lista de entradas triples, si el dispositivo es capaz de  
15 descodificar un flujo de bits que comprende una representación codificada de los datos de vídeo de múltiples capas, en el que cada entrada triple respectiva en la lista de entradas triples indica un perfil, un grado y un nivel para un descodificador de capa única respectivo en el dispositivo; y un medio para descodificar, basándose en que el dispositivo es capaz de descodificar el flujo de bits, el flujo de bits.

20 **[0010]** En otro ejemplo, esta divulgación describe un medio de almacenamiento de datos legible por ordenador que comprende instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que un descodificador de vídeo de múltiples capas: determine, basándose en la lista de entradas triples, si el descodificador de vídeo de múltiples capas es capaz de descodificar un flujo de bits que comprende una representación codificada de datos de vídeo de múltiples capas, en el que cada entrada triple respectiva en la lista de entradas triples indica un perfil, un grado y un  
25 nivel para un descodificador de capa única respectivo en el descodificador de vídeo de múltiples capas; y basándose en que el descodificador de vídeo de múltiples capas es capaz de descodificar el flujo de bits, descodificar el flujo de bits.

**[0011]** Los detalles de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y la descripción  
30 siguiente. Otras características, objetivos y ventajas resultarán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

## BREVE DESCRIPCIÓN DE LOS DIBUJOS

35 **[0012]**

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo de ejemplo que puede utilizar las técnicas descritas en esta divulgación.

40 La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de implementación de un descodificador de vídeo, que comprende una pluralidad de descodificadores de capa única, de acuerdo con una técnica de esta divulgación.

45 La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 4 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo que puede implementar las técnicas descritas en esta divulgación.

50 La FIG. 5A es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un codificador de vídeo, de acuerdo con una técnica de esta divulgación.

55 La FIG. 5B es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un descodificador de vídeo de acuerdo con una técnica de esta divulgación.

La FIG. 6A es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un codificador de vídeo, de acuerdo con una técnica de esta divulgación.

60 La FIG. 6B es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un descodificador de vídeo de acuerdo con una técnica de esta divulgación.

La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un descodificador de vídeo de acuerdo con una técnica de esta divulgación.

65 **DESCRIPCIÓN DETALLADA**

**[0013]** Un flujo de bits (es decir, un flujo de bits de vídeo) puede comprender una representación codificada de datos de vídeo. Más específicamente, el flujo de bits puede comprender una serie de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL puede incluir una cabecera de unidad NAL y una carga útil de secuencia de bytes sin procesar (Rbsp). En la codificación de vídeo de múltiples capas, las unidades NAL pueden corresponder a varias capas de datos de vídeo. En la codificación de vídeo escalable, las capas pueden incluir una capa base y una o más capas de mejora. En la codificación de múltiples visualizaciones, cada una de las capas puede corresponder a diferentes visualizaciones, tales como diferentes visualizaciones de una escena de vídeo.

**[0014]** Ciertos estándares de codificación de vídeo, tales como codificación de vídeo de alta eficiencia (HEVC), definen varios perfiles, grados y niveles diferentes. En general, un "perfil" de un estándar de codificación de vídeo es un subconjunto de las características y herramientas presentes en el estándar de codificación de vídeo. En otras palabras, un perfil define qué herramientas de codificación pueden usarse. Por ejemplo, para un codificador de vídeo, se puede establecer un perfil de herramientas de codificación que el codificador de vídeo puede usar para generar flujos de bits codificados que conforman dicho perfil. Para un descodificador de vídeo, un perfil puede significar el conjunto de herramientas de codificación que el descodificador de vídeo debe tener para poder descodificar flujos de bits que se dice que se ajustan a dicho perfil.

**[0015]** Un nivel es un conjunto definido de restricciones de los valores que pueden ser adoptadas por los elementos sintácticos y las variables de un estándar de compresión de vídeo. Un grado es una categoría específica de restricciones de nivel impuestas a valores de los elementos sintácticos en el flujo de bits o valores de variables, donde las restricciones de nivel están agrupadas dentro de un grado y un descodificador conforme a un cierto grado y nivel sería capaz de descodificar todos los flujos de bits que se ajustan al mismo grado o al grado inferior de ese nivel o cualquier nivel por debajo de ese nivel. De este modo, un nivel de un grado es un conjunto especificado de restricciones impuestas en los valores de los elementos sintácticos en el flujo de bits o variables utilizadas en la descodificación del flujo de bits.

**[0016]** Diferentes descodificadores de vídeo pueden tener diferentes capacidades. Por ejemplo, un descodificador de vídeo puede ser capaz de descodificar flujos de bits que se ajustan a un primer perfil, pero puede que no sea capaz de descodificar flujos de bits que se ajustan a un segundo perfil. En este ejemplo, otro descodificador de vídeo puede descodificar flujos de bits que se ajusten al primer perfil y al segundo perfil. Por ejemplo, con capas escalables de codificación de vídeo, el nivel de calidad de vídeo que se puede lograr puede escalarse a través de capas de modo que algunos dispositivos solo pueden descodificar y presentar un nivel básico de calidad de vídeo, mientras que otros dispositivos pueden descodificar y presentar una calidad de vídeo mejorada. Además, las capas escalables pueden permitir una recepción de datos de vídeo más fiable cuando el ancho de banda de canal es limitado; por ejemplo, un ancho de banda limitado puede ser suficiente para la comunicación de datos de vídeo de la capa base y canales de ancho de banda más robusto pueden permitir la comunicación de datos de vídeo de capas base y de mejora. En general, si un primer procesador puede descodificar flujos de bits que se ajustan a perfiles, grados y niveles que son un superconjunto de perfiles, grados y niveles que pueden ser descodificados por un segundo procesador, el primer procesador es más complejo y por lo tanto tiende a ser más caro.

**[0017]** De forma similar, con la codificación de múltiples visualizaciones, algunos dispositivos pueden descodificar múltiples visualizaciones, mientras que otros dispositivos pueden solo pueden tener la capacidad de descodificar un número limitado de visualizaciones. Las restricciones del ancho de banda del canal también pueden utilizarse de manera más efectiva mediante la codificación de múltiples visualizaciones, de modo que un número limitado de visualizaciones puede descodificarse cuando el ancho de banda es limitado y un mayor número de visualizaciones puede descodificarse cuando lo permite el ancho de banda.

**[0018]** En algunos casos, un descodificador de vídeo comprende múltiples núcleos de hardware (por ejemplo, unidades de procesamiento o módulos de procesamiento). En otras palabras, un descodificador de vídeo puede construirse utilizando una pluralidad de núcleos de hardware. Cada uno de los núcleos de hardware puede ser capaz de descodificar independientemente al menos elementos sintácticos de bajo nivel de un flujo de bits. Los diferentes núcleos pueden ser similares o diferentes en sus capacidades de procesamiento, pero, independientemente de ello, la presencia de múltiples núcleos puede proporcionar ventajas de procesamiento respecto a los descodificadores de núcleo único. En algunos ejemplos, un descodificador de vídeo que comprende múltiples núcleos de hardware puede descodificar múltiples partes de un flujo de bits en paralelo. Por ejemplo, algunos núcleos de procesamiento pueden implementar un descodificador de capa única. Un descodificador de capa única puede descodificar una capa única de un flujo de bits, tal como una capa base de vídeo escalable o una visualización particular en vídeo de múltiples visualizaciones. Por el contrario, un descodificador de múltiples capas puede descodificar múltiples capas de un flujo de bits simultáneamente (es decir, en paralelo). En un ejemplo, un solo núcleo de hardware es capaz de descodificar más de una capa.

**[0019]** Diferentes núcleos de hardware de un descodificador de vídeo pueden tener diferentes capacidades. Por ejemplo, un núcleo de hardware de un descodificador de vídeo puede descodificar flujos de bits que se ajusten a un primer perfil, pero no a un segundo perfil, mientras que un segundo núcleo de hardware de un descodificador de vídeo puede descodificar flujos de bits que se ajusten al primer perfil y al segundo perfil. Un descodificador de vídeo

puede tener núcleos de hardware que tienen diferentes capacidades con el fin de mantener el coste general del descodificador de vídeo lo más bajo posible, dado el propósito pretendido del descodificador de vídeo.

5 [0020] Típicamente, en casos donde un descodificador de vídeo comprende múltiples núcleos de hardware, el descodificador de vídeo determina, basándose en la información de perfil, grado y nivel señalado en un flujo de bits, si un núcleo de hardware con más capacidad del descodificador de vídeo puede descodificar el flujo de bits. Si es así, el descodificador de vídeo determina que es capaz de descodificar el flujo de bits. Este proceso no permite que el descodificador de vídeo determine si alguno de los núcleos de hardware con menos capacidad del descodificador de vídeo puede ayudar en el proceso de descodificación del flujo de bits. Como resultado, los núcleos de hardware con menos capacidad pueden estar inactivos, mientras que el núcleo de hardware con más capacidad descodifica todo el flujo de bits. Sin embargo, los núcleos de hardware con menos capacidad del descodificador de vídeo pueden, de hecho, ser capaces de descodificar capas particulares del flujo de bits en paralelo con el núcleo de hardware con más capacidad del descodificador de vídeo. Como resultado, no usar los núcleos de hardware con menos capacidad puede prolongar innecesariamente los tiempos de descodificación.

15 [0021] Algunas técnicas de esta divulgación pueden facilitar reducciones en los tiempos de descodificación en descodificadores de vídeo que tienen núcleos de hardware con diferentes capacidades. Por ejemplo, algunas técnicas de esta divulgación pueden disminuir los tiempos de descodificación en los descodificadores de vídeo contruidos basándose en múltiples descodificadores de capa única que tienen diferentes capacidades. Esta reducción en los tiempos de descodificación en relación con las técnicas más convencionales se puede lograr sin cambiar los núcleos de hardware en sí (por ejemplo, sin modificar las estructuras físicas de los núcleos de hardware). Además, cuando los núcleos de hardware tienen las mismas capacidades, algunas técnicas de esta divulgación pueden permitir a los descodificadores de vídeo descodificar más capas en paralelo.

25 [0022] De acuerdo con una técnica de ejemplo de esta divulgación, un descodificador de vídeo puede descodificar un flujo de bits que comprende una representación codificada de datos de vídeo. En este ejemplo, el flujo de bits comprende una pluralidad de capas. Como parte de la descodificación del flujo de bits, el descodificador de vídeo puede obtener, a partir del flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición. Para cada esquema de partición respectivo del uno o más esquemas de partición, el esquema de partición respectivo especifica un conjunto respectivo de particiones disjuntas cuya unión forma un conjunto de capas de salida. Cada partición respectiva del conjunto respectivo de particiones disjuntas contiene una o más de las capas. Por lo tanto, una partición puede ser un tipo particular de sub-flujo de bits del flujo de bits. El conjunto de capas de salida puede consistir en todas las capas de la pluralidad de capas o un subconjunto adecuado de la pluralidad de capas. Además, el descodificador de vídeo puede descodificar cada una de las particiones de un esquema de partición particular usando diferentes núcleos de procesamiento en una pluralidad de núcleos de hardware, siendo el esquema de partición particular uno de los uno o más esquemas de partición.

35 [0023] En algunos ejemplos, para cada partición respectiva de cada esquema de partición respectivo del uno o más esquemas de partición, el descodificador de vídeo obtiene, del flujo de bits, un elemento sintáctico respectivo que indica información de perfil, grado y nivel (PTL) respectiva para la partición respectiva. En algunos casos, el descodificador de vídeo puede seleccionar, basándose en la información de PTL para las particiones de los esquemas de partición, un esquema de partición del uno o más esquemas de partición. En algunos casos, el descodificador de vídeo puede recibir el flujo de bits de acuerdo con un esquema de partición particular como lo determinan los recursos de red disponibles. El descodificador de vídeo puede descodificar cada partición respectiva del esquema de partición particular usando uno respectivo de los núcleos de hardware. Cada uno de los respectivos núcleos de hardware puede implementar un descodificador de capa única respectivo. En algunos casos, un núcleo de hardware del descodificador de vídeo puede ser capaz de descodificar más de una capa.

40 [0024] Las técnicas adicionales de esta divulgación pueden proporcionar pruebas de conformidad de flujo de bits usando las particiones de flujo de bits. Por ejemplo, un dispositivo de procesamiento de vídeo (tal como un codificador de vídeo, descodificador de vídeo u otro dispositivo) puede descodificar un flujo de bits que comprende una representación codificada de datos de vídeo. El flujo de bits puede comprender una pluralidad de capas. Como parte de la descodificación del flujo de bits, el descodificador de vídeo puede obtener, a partir del flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición para un conjunto de capas de salida. En estos tipos de ejemplos, el conjunto de capas de salida puede consistir en todas las capas de la pluralidad de capas o un subconjunto adecuado de la pluralidad de capas (por ejemplo, no todas las capas de la pluralidad de capas). Además, para cada esquema de partición respectivo del uno o más esquemas de partición, el esquema de partición respectivo puede especificar una división de la capa de salida establecida en una o más particiones. Cada capa en el conjunto de capas de salida puede pertenecer exactamente a una partición en el esquema de partición respectivo. Para cada esquema de partición respectivo del uno o más esquemas de partición, el descodificador de vídeo puede obtener, a partir de un mensaje de información de mejora suplementario (SEI) en el flujo de bits o información de usabilidad de vídeo (VUI) en un conjunto de parámetros de vídeo (VPS) en el flujo de bits; un conjunto respectivo de elementos sintácticos que indican conjuntos de parámetros de descodificador de referencia hipotético (HRD) para el esquema de partición. Además, para cada partición respectiva del esquema de partición, el dispositivo de procesamiento de vídeo puede probar, basándose en los conjuntos de parámetros de HRD para el esquema de

partición respectivo, la partición respectiva en cuanto a conformidad con un estándar de codificación de vídeo, tal como MV-HEVC o SHVC.

5 **[0025]** De acuerdo con algunas técnicas de esta divulgación, un descodificador de vídeo de múltiples capas puede determinar, basándose en una lista de entradas triples, si el descodificador de vídeo de múltiples capas es capaz de descodificar (es decir, está configurado para descodificar) un flujo de bits que comprende una representación codificada de los datos de vídeo de múltiples capas. El número de entradas triples en la lista puede ser menor o igual que una cantidad de descodificadores de capa única en el descodificador de vídeo de múltiples capas. Cada entrada triple respectiva en la lista de entradas triples indica un perfil, un grado y un nivel que un descodificador de capa única respectivo en el descodificador de vídeo de múltiples capas es capaz de descodificar (es decir, está configurado para descodificar). Basándose en que el descodificador de vídeo de múltiples capas es capaz de descodificar el flujo de bits, el descodificador de vídeo de múltiples capas puede descodificar el flujo de bits. En algunos ejemplos, el dispositivo de procesamiento de vídeo puede determinar que el descodificador de múltiples capas es capaz de descodificar el flujo de bits determinando que el flujo de bits especifica un esquema de partición donde la información PTL de cada partición del esquema de partición indica que la partición puede descodificarse al menos uno de los descodificadores de capa única. Esto puede permitir que un descodificador de vídeo de múltiples capas que tenga múltiples descodificadores de capa única determine si el descodificador de vídeo de múltiples capas es capaz de descodificar el flujo de bits.

20 **[0026]** La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo 10 de ejemplo que puede utilizar las técnicas de esta divulgación. Tal como se utiliza en el presente documento, el término "codificador de vídeo" se refiere genéricamente tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" pueden referirse genéricamente a la codificación de vídeo o la descodificación de vídeo.

25 **[0027]** Como es muestra en la FIG. 1, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12, un dispositivo de destino 14 y un elemento de red 15. El dispositivo de origen 12 genera datos de vídeo codificados. En consecuencia, el dispositivo de origen 12 puede denominarse dispositivo de codificación de vídeo o aparato de codificación de vídeo. El dispositivo de destino 14 puede descodificar los datos de vídeo codificados, generados por el dispositivo de origen 12. En consecuencia, el dispositivo de destino 14 puede denominarse dispositivo de descodificación de vídeo o aparato de descodificación de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 pueden ser ejemplos de dispositivos de codificación de vídeo o aparatos de codificación de vídeo.

35 **[0028]** El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, dispositivos informáticos móviles, ordenadores plegables (es decir, portátiles), ordenadores de tableta, descodificadores, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores de coche o similares.

40 **[0029]** El elemento de red 15 puede recibir datos de vídeo codificados y emitir datos de vídeo codificados procesados. El elemento de red 15 puede ser un elemento de red compatible con medios (MANE), un dispositivo de red de entrega de contenido (CDN) u otro tipo de dispositivo (por ejemplo, un dispositivo informático). El dispositivo de red 15 puede realizar una o más técnicas de esta divulgación. Por ejemplo, el dispositivo de red 15 puede realizar una o más pruebas de conformidad de flujo de bits, de acuerdo con una o más técnicas de esta divulgación. El elemento de red 15, el dispositivo de origen 12, el dispositivo de destino 14 y otros tipos de dispositivo que procesan datos de vídeo pueden considerarse dispositivos de procesamiento de vídeo.

50 **[0030]** El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 a través de un canal 16. El canal 16 puede comprender uno o más medios o dispositivos capaces de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender uno o más medios de comunicación que permiten al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de origen 12 puede modular los datos de vídeo codificados de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo de destino 14. Los uno o más medios de comunicación pueden incluir medios de comunicación inalámbricos y/o alámbricos, tales como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. Los uno o más medios de comunicación pueden formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global (por ejemplo, Internet). Los uno o más medios de comunicación pueden incluir routers, conmutadores, estaciones base u otros equipos que faciliten la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

60 **[0031]** En otro ejemplo, el canal 16 puede incluir un medio de almacenamiento que almacena datos de vídeo codificados, generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder al medio de almacenamiento, por ejemplo, mediante el acceso a disco o el acceso a tarjeta. El medio de almacenamiento puede incluir varios medios de almacenamiento de datos de acceso local, tales como discos Blu-

ray, DVD, CD-ROM, memoria flash u otros medios adecuados de almacenamiento digital para almacenar datos de vídeo codificados.

5 **[0032]** En un ejemplo adicional, el canal 16 puede incluir un servidor de ficheros u otro dispositivo de almacenamiento intermedio que almacene los datos de vídeo codificados, generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder a datos de vídeo codificados almacenados en el servidor de ficheros u otro dispositivo de almacenamiento intermedio mediante transmisión continua o descarga. El servidor de ficheros puede ser un tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir los datos de vídeo codificados al dispositivo de destino 14. Entre los ejemplos de servidores de ficheros se incluyen servidores de la Red (por ejemplo, para una sede de la Red), servidores del protocolo de transferencia de ficheros (FTP), dispositivos de almacenamiento conectados a red (NAS) y unidades de disco local.

15 **[0033]** El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de una conexión de datos estándar, tal como una conexión a Internet. Entre los ejemplos de tipos de conexiones de datos pueden incluirse canales inalámbricos (por ejemplo, conexiones Wi-Fi), conexiones alámbricas (por ejemplo, línea de abonados digital (DSL), módem de cable, etc.), o combinaciones de ambos que sean adecuadas para acceder a datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el servidor de ficheros puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

20 **[0034]** Las técnicas de esta divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de una diversidad de aplicaciones multimedia, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, mediante Internet, codificación de datos de vídeo para su almacenamiento en un medio de almacenamiento de datos, descodificación de datos de vídeo almacenados en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo 10 puede configurarse para soportar la transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tales como la radiodifusión de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

25 **[0035]** La FIG. 1 es meramente un ejemplo y las técnicas de esta divulgación pueden aplicarse a configuraciones de codificación de vídeo (por ejemplo, codificación de vídeo o descodificación de vídeo) que no incluyan necesariamente ninguna comunicación de datos entre los dispositivos de codificación y descodificación. En otros ejemplos, los datos (por ejemplo, datos de vídeo) se recuperan de una memoria local, se transmiten por una red o similar. Un dispositivo de codificación de vídeo puede codificar y almacenar datos (por ejemplo, datos de vídeo) en la memoria, y/o un dispositivo de descodificación de vídeo puede recuperar y descodificar datos (por ejemplo, datos de vídeo) desde la memoria. En muchos ejemplos, la codificación y descodificación se realiza mediante dispositivos que no se comunican entre sí, sino que simplemente codifican datos en la memoria y/o recuperan y descodifican datos (por ejemplo, datos de vídeo) desde la memoria.

30 **[0036]** En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos ejemplos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. El origen de vídeo 18 puede incluir un dispositivo de captura de vídeo, por ejemplo, una videocámara, un fichero de vídeo que contiene datos de vídeo previamente capturados, una interfaz de alimentación de vídeo para recibir datos de vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de vídeo, o una combinación de dichos orígenes de datos de vídeo. Por lo tanto, en algunos ejemplos, el dispositivo de origen 12 comprende una cámara configurada para capturar datos de vídeo.

35 **[0037]** El codificador de vídeo 20 puede codificar datos de vídeo procedentes del origen de vídeo 18. En algunos ejemplos, el dispositivo de origen 12 directamente transmite los datos de vídeo codificados al dispositivo de destino 14 mediante la interfaz de salida 22. En otros ejemplos, los datos de vídeo codificados también pueden almacenarse en un medio de almacenamiento o un servidor de ficheros para un acceso posterior mediante el dispositivo de destino 14 para su descodificación y/o reproducción.

40 **[0038]** En el ejemplo de la FIG. 1, el dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos ejemplos, la interfaz de entrada 28 incluye un receptor y/o un módem. La interfaz de entrada 28 puede recibir los datos de vídeo codificados por el canal 16. El codificador de vídeo 30 puede descodificar datos de vídeo codificados. El dispositivo de visualización 32 puede mostrar los datos de vídeo descodificados. El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. El dispositivo de visualización 32 puede comprender varios dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

45 **[0039]** El codificador de vídeo 20 y el descodificador de vídeo 30 pueden, cada uno, implementarse como cualquiera entre varios circuitos adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), conjuntos de puertas programables sobre el

terreno (FPGA), lógica discreta, hardware o cualquier combinación de los mismos. Si las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado de almacenamiento no transitorio legible por ordenador, y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cualquiera de los anteriores (incluyendo hardware, software, una combinación de hardware y software, etc.) puede considerarse como uno o más procesadores. Cada uno del codificador de vídeo 20 y del descodificador de vídeo 30 puede estar incluido en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador (CODEC) combinado en un respectivo dispositivo.

**[0040]** Esta divulgación puede referirse en general al codificador de vídeo 20 que "señala" cierta información a otro dispositivo, tal como el descodificador de vídeo 30. El término "indicar" puede referirse en general a la comunicación de elementos sintácticos y/u otros datos usados para descodificar los datos de vídeo comprimidos. Dicha comunicación puede producirse en tiempo real o casi real. De forma alternativa, dicha comunicación puede producirse durante un tramo de tiempo, tal como podría ocurrir cuando se almacenan elementos sintácticos en un medio de almacenamiento legible por ordenador en un flujo de bits codificado en el momento de la codificación, que a continuación un dispositivo de descodificación puede recuperar en cualquier momento tras haber sido almacenado en este medio.

**[0041]** En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 funcionan de acuerdo con un estándar de compresión de vídeo, tal como ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluida su extensión de codificación de vídeo escalable (SVC), extensión de codificación de vídeo de múltiples visualizaciones (MVC) y extensión 3DV basada en MVC. En algunos casos, cualquier flujo de bits que se ajuste a la extensión 3DV basada en MVC de H.264/AVC siempre contiene un sub-flujo de bits que es compatible con la extensión MVC de H.264/AVC. El último borrador conjunto de MVC se describe en "Codificación de vídeo avanzada para servicios audiovisuales genéricos", Recomendación ITU-T H.264, marzo de 2010. Además, actualmente se están emprendiendo unas iniciativas para generar una ampliación de codificación de vídeo tridimensional (3DV) para H.264/AVC, en concreto, 3DV basada en AVC. En otros ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263 e ISO/IEC MPEG-4 Visual.

**[0042]** En otros ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con el estándar de Codificación de Vídeo de Alta Eficiencia (HEVC) elaborado por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC. Un borrador del estándar HEVC, denominado "Memoria descriptiva de borrador de HEVC" se describe en el documento Bross et al. "Informe 3 Predeterminado de Codificación de Vídeo de Alta Eficiencia (HEVC)", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 16.ª reunión, San José, Estados Unidos, enero de 2014, n.º de documento JCTVC-P1003\_v1. La memoria descriptiva de borrador de HEVC está disponible en [http://phenix.it-sudparis.eu/jct/doc\\_end\\_user/documents/16\\_San%20Jose/wg11/JCTVC-P1003-v1.zip](http://phenix.it-sudparis.eu/jct/doc_end_user/documents/16_San%20Jose/wg11/JCTVC-P1003-v1.zip).

**[0043]** Además, actualmente se están dedicando esfuerzos a la elaboración de una extensión de codificación de vídeo para HEVC. La extensión de codificación de vídeo escalable de HEVC puede denominarse SHEVC o SHVC. Adicionalmente, un Equipo de Colaboración Conjunta en Codificación de Vídeo 3D (JCT-3C) de VCEG y MPEG está elaborando un estándar 3DV basado en HEVC. Parte de los esfuerzos de estandarización para el estándar 3DV basado en HEVC incluyen la estandarización de un códec de vídeo de múltiples visualizaciones basado en HEVC (es decir, MV-HEVC). Otra parte de los esfuerzos de estandarización para el estándar 3DV basado en HEVC incluye la estandarización de una codificación de vídeo 3D basada en HEVC (es decir, 3D-HEVC). Para 3D-HEVC, se pueden incluir y soportar nuevas herramientas de codificación, incluyendo aquellas a nivel de unidad de codificación (CU) o unidad de predicción (PU), tanto para las visualizaciones de textura como de profundidad.

**[0044]** Un reciente Borrador de Trabajo (WD) de MV-HEVC, denominado MV-HEVC WD8 de aquí en adelante, está disponible en [http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/documents/8\\_Valencia/wg11/JCT3V-H1002-v1.zip](http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/8_Valencia/wg11/JCT3V-H1002-v1.zip). La extensión escalable a HEVC, llamada SHVC, también está siendo desarrollada por el JCT-VC. Un borrador de trabajo (WD) reciente de SHVC, denominado SHVC WD6 de aquí en adelante, está disponible en [http://phenix.it-sudparis.eu/jct/doc\\_end\\_user/documents/17\\_Valencia/wg11/JCTVC-Q1008-v1.zip](http://phenix.it-sudparis.eu/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1008-v1.zip).

**[0045]** En la HEVC y otras especificaciones de codificación de vídeo, una secuencia de vídeo incluye típicamente una serie de imágenes. Las imágenes también pueden denominarse "tramas". Una imagen puede incluir tres formaciones de muestras, denominadas  $S_L$ ,  $S_{Cb}$  y  $S_{Cr}$ .  $S_L$  es una formación bidimensional (es decir, un bloque) de muestras de luma.  $S_{Cb}$  es una formación bidimensional de muestras de croma Cb.  $S_{Cr}$  es una formación bidimensional de muestras de croma Cr. Las muestras de croma también se pueden denominar en el presente documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede solo incluir una formación de muestras de luma.

**[0046]** Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades de árbol de codificación (CTU). Cada una de las CTU puede comprender un bloque de árbol



de codificación de muestras de luma, dos bloques de árbol de codificación correspondientes de muestras de croma y estructuras sintácticas usadas para codificar las muestras de los bloques de árbol de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CTU puede comprender un solo bloque de árbol de codificación y estructuras sintácticas usadas para codificar las muestras del bloque de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque de muestras de tamaño NxN. Una CTU también puede denominarse "bloque de árbol" o una "máxima unidad de codificación" (LCU). Las CTU de la HEVC pueden ser análogas en gran medida a los macrobloques de otros estándares, tales como H.264/AVC. Sin embargo, una CTU no está limitada necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). Un fragmento puede incluir un número entero de CTU ordenadas consecutivamente en un orden de escaneado cuadrado.

**[0047]** Esta divulgación puede usar el término "unidad de vídeo", "bloque de vídeo" o "bloque" para referirse a uno o más bloques de muestras y estructuras sintácticas usadas para codificar muestras del uno o más bloques de muestras. Los tipos de ejemplo de unidades de vídeo pueden incluir CTU, CU, PU, unidades de transformación (TU), macrobloques, particiones de macrobloque, etc. En algunos contextos, el análisis de PU puede intercambiarse con el análisis de macrobloques o particiones de macrobloques.

**[0048]** Para generar una CTU codificada, el codificador de vídeo 20 puede llevar a cabo de forma recursiva una partición de árbol cuaternario en los bloques de árbol de codificación de una CTU para dividir los bloques de árbol de codificación en bloques de codificación, de ahí el nombre "unidades de árbol de codificación". Un bloque de codificación es un bloque de muestras de tamaño NxN. Una CU puede comprender un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene una matriz de muestras de luma, una matriz de muestras de Cb y una matriz de muestras de Cr y estructuras sintácticas usadas para codificar las muestras de los bloques de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CU puede comprender un único bloque de codificación y estructuras sintácticas usadas para codificar las muestras del bloque de codificación.

**[0049]** El codificador de vídeo 20 puede dividir un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción es un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las que se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede comprender un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma y estructuras sintácticas usadas para predecir los bloques de predicción. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una PU puede comprender un solo bloque de predicción y estructuras sintácticas usadas para predecir el bloque de predicción. El codificador de vídeo 20 puede generar bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para bloques de predicción (por ejemplo, bloques de predicción de luma, Cb y Cr) de cada PU de la CU.

**[0050]** El codificador de vídeo 20 puede usar intra-predicción o inter-predicción para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 utiliza intra-predicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU basándose en muestras descodificadas de la imagen que incluye la PU.

**[0051]** Después de que el codificador de vídeo 20 genera bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para una o más PU de una CU, el codificador de vídeo 20 puede generar uno o más bloques residuales para la CU. Por ejemplo, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra en el bloque residual de luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques predictivos de luma de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual de Cb para la CU. Cada muestra en el bloque residual de Cb de la CU puede indicar una diferencia entre una muestra de Cb en uno de los bloques predictivos de Cb de la CU y una muestra correspondiente en el bloque de codificación de Cb original de la CU. El codificador de vídeo 20 también puede generar un bloque residual de Cr para la CU. Cada muestra en el bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques de Cr predictivos de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.

**[0052]** Además, el codificador de vídeo 20 puede usar la partición en árbol cuádruple para descomponer los bloques residuales (por ejemplo, los bloques residuales de luma, Cb y Cr) de una CU en uno o más bloques de transformación (por ejemplo, bloques de transformación de luma, Cb y Cr). Un bloque de transformación es un bloque rectangular (por ejemplo, cuadrado o no cuadrado) de muestras a las que se aplica la misma transformación. Una unidad de transformación (TU) de una CU puede comprender un bloque de transformación de muestras de luma, dos bloques de transformación correspondientes de muestras de croma y estructuras sintácticas utilizadas para transformar las muestras de bloques de transformación. De este modo, cada TU de una CU puede tener un bloque de transformación de luma, un bloque de transformación de Cb y un bloque de transformación de Cr. El bloque de transformación de luma de la TU puede ser un sub-bloque del bloque residual de luma de la CU. El bloque de transformación de Cb puede ser un sub-bloque del bloque residual de Cb de la CU. El bloque de transformación de Cr puede ser un sub-bloque del bloque residual de Cr de la CU. En imágenes monocromáticas o imágenes que

tienen tres planos de color separados, una TU puede comprender un solo bloque de transformación y unas estructuras sintácticas usadas para transformar las muestras del bloque de transformación.

5 **[0053]** El codificador de vídeo 20 puede aplicar una o más transformaciones a un bloque de transformación de TU para generar un bloque de coeficientes para la TU. Por ejemplo, el codificador de vídeo 20 puede aplicar una o más transformaciones a un bloque de transformación de luma de una TU con el fin de generar un bloque de coeficientes de luma para la TU. Un bloque de coeficientes puede ser una formación bidimensional de coeficientes de transformación. Un coeficiente de transformación puede ser una cantidad escalar. El codificador de vídeo 20 puede aplicar una o más transformaciones a un bloque de transformación de Cb de una TU para generar un bloque de coeficientes de Cb para la TU. El codificador de vídeo 20 puede aplicar una o más transformaciones a un bloque de transformación de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

15 **[0054]** Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luma, un bloque de coeficientes de Cb o un bloque de coeficientes de Cr), el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. La cuantificación se refiere, en general, a un proceso en el que los coeficientes de transformación se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes de transformación, proporcionando una compresión adicional. Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos sintácticos que indican los coeficientes de transformación cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar la Codificación Aritmética Binaria Adaptativa al Contexto (CABAC) en los elementos sintácticos que indican los coeficientes de transformación cuantificados.

25 **[0055]** El codificador de vídeo 20 puede emitir un flujo de bits que incluye una secuencia de bits que forman una representación de imágenes codificadas y datos asociados. Por lo tanto, el flujo de bits comprende una representación codificada de datos de vídeo. El flujo de bits puede comprender una secuencia de unidades de capa de abstracción de red (NAL). Una unidad NAL es una estructura sintáctica que contiene una indicación del tipo de datos en la unidad NAL y los bytes que contienen dichos datos en forma de una carga útil de secuencia de bytes sin procesar (RBSP) intercalados según sea necesario con bits de prevención de emulación. Cada una de las unidades NAL incluye una cabecera de la unidad NAL y encapsula una RBSP. La cabecera de la unidad NAL puede incluir un elemento sintáctico que indica un código de tipo de unidad NAL. El código de tipo de unidad NAL especificado por la cabecera de unidad NAL de una unidad NAL indica el tipo de la unidad NAL. Una RBSP puede ser una estructura sintáctica que contiene un número entero de bytes que se encapsula dentro de una unidad NAL. En algunos casos, una RBSP incluye cero bits.

35 **[0056]** Diferentes tipos de unidades NAL pueden encapsular diferentes tipos de las RBSP. Por ejemplo, diferentes tipos de unidades NAL pueden encapsular diferentes RBSP para conjuntos de parámetros de vídeo (VPS), conjuntos de parámetros de secuencia (SPS), conjuntos de parámetros de imagen (PPS), fragmentos codificados, información de mejora suplementaria (SEI), etc. Las unidades NAL que encapsulan las RBSP para datos de codificación de vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes SEI) pueden denominarse unidades NAL de la capa de codificación de vídeo (VCL). En HEVC (es decir, HEVC no de múltiples capas), una unidad de acceso puede ser un conjunto de unidades NAL que son consecutivas en orden de descodificación y que contienen exactamente una imagen codificada. Además de las unidades NAL de fragmento codificado de la imagen codificada, la unidad de acceso también puede contener otras unidades NAL que no contienen fragmentos de la imagen codificada. En algunos ejemplos, la descodificación de una unidad de acceso siempre da como resultado una imagen descodificada. La información de mejora suplementaria (SEI) contiene información que no es necesaria para descodificar las muestras de imágenes codificadas de las unidades NAL VCL. Un SEI RBSP contiene uno o más mensajes SEI.

50 **[0057]** El descodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede analizar el flujo de bits para obtener elementos sintácticos del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basándose, al menos en parte, en los elementos sintácticos obtenidos a partir del flujo de bits. El proceso para reconstruir los datos de vídeo puede ser en general recíproco al proceso realizado por el codificador de vídeo 20. Por ejemplo, el descodificador de vídeo 30 puede usar vectores en movimiento de las PU para determinar bloques predictivos para las PU de una CU actual. Además, el descodificador de vídeo 30 puede cuantificar inversamente bloques de coeficientes de TU de la CU actual. El descodificador de vídeo 30 puede realizar transformaciones inversas en los bloques de coeficientes para reconstruir los bloques de transformación de las TU de la CU actual. El descodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para las PU de la CU actual a las muestras correspondientes de los bloques de transformación de las TU de la CU actual. Mediante la reconstrucción de los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.

65 **[0058]** Las técnicas de esta divulgación son potencialmente aplicables a la codificación de múltiples visualizaciones y/o estándares y especificaciones 3DV, incluyendo MV-HEVC y 3D-HEVC. En la codificación de múltiples visualizaciones, tal como la definida en MV-HEVC y 3D-HEVC, puede haber varias visualizaciones de la misma escena desde diferentes puntos de visualización. En el contexto de la codificación de múltiples

visualizaciones y codificación de vídeo escalable, el término "unidad de acceso" puede usarse para referirse a un conjunto de imágenes que corresponden a la misma instancia de tiempo. En algunos casos, en el contexto de codificación de múltiples visualizaciones y codificación de vídeo escalable, una unidad de acceso puede comprender un conjunto de unidades NAL que están asociadas entre sí de acuerdo con una regla de clasificación especificada, son consecutivas en orden de descodificación y contienen las unidades NAL VCL de todas las imágenes codificadas asociadas con el mismo tiempo de salida y sus unidades NAL no VCL asociadas. Así, los datos de vídeo pueden conceptualizarse como una serie de unidades de acceso que se producen a lo largo del tiempo.

**[0059]** En codificación 3DV, como la definida en 3D-HEVC, una "componente de visualización" puede ser una representación codificada de una visualización en una única unidad de acceso. Un componente de visualización puede contener un componente de visualización en profundidad y un componente de visualización en textura. Un componente de visualización de profundidad puede ser una representación codificada de la profundidad de una visualización en una única unidad de acceso. Un componente de visualización de textura puede ser una representación codificada de la textura de una visualización en una única unidad de acceso. En esta divulgación, una "vista" puede referirse a una secuencia de componentes de visualización asociadas con el mismo identificador de visualización.

**[0060]** La componente de visualización de textura y la componente de visualización de profundidad de un conjunto de imágenes de una visualización pueden considerarse como correspondientes entre sí. Por ejemplo, la componente de visualización de textura de un conjunto de imágenes de una visualización se considera correspondiente al componente de visualización de profundidad del conjunto de las imágenes de la visualización, y viceversa (es decir, la componente de visualización de profundidad corresponde a su componente de visualización de textura en el conjunto, y viceversa). Como se usa en esta divulgación, una componente de visualización de textura que corresponde a una componente de visualización de profundidad puede considerarse como la componente de visualización de textura y la componente de visualización de profundidad que son parte de una misma visualización de una única unidad de acceso.

**[0061]** La componente de visualización de textura incluye el contenido de imagen real que se muestra. Por ejemplo, la componente de visualización de textura puede incluir las componentes de luma (Y) y croma (Cb y Cr). El componente de visualización de profundidad puede indicar profundidades relativas de los píxeles en su componente de visualización de textura correspondiente. Como un ejemplo, el componente de visualización de profundidad puede comprender una imagen de escala de grises que incluye solo valores de luma, pero los valores de luma en un componente de visualización de profundidad pueden representar un nivel de profundidad de un píxel o conjunto de píxeles, a diferencia de un nivel de intensidad de píxel que sería transmitido por un componente de luma de una visualización de textura. En otras palabras, la componente de visualización de profundidad tal vez no transmita ningún contenido de imagen, pero en lugar de eso proporcione una medida de las profundidades relativas de los píxeles en la componente de visualización de textura. La componente de visualización de profundidad que utiliza solo valores de luma (por ejemplo, valores de intensidad) para identificar la profundidad se proporciona con fines ilustrativos y no debe considerarse limitativa. En otros ejemplos, puede utilizarse cualquier técnica para indicar las profundidades relativas de los píxeles en la componente de visualización de textura.

**[0062]** En la codificación de múltiples visualizaciones y la codificación SVC, un flujo de bits puede tener una pluralidad de capas. En codificación de múltiples visualizaciones, cada una de las capas puede corresponder a una visualización diferente. Una capa puede denominarse "capa base" si un descodificador de vídeo (por ejemplo, descodificador de vídeo 30) puede descodificar imágenes en la capa sin referencia a imágenes de ninguna otra capa. Una capa puede denominarse "capa no base" o "capa dependiente" si la descodificación de la capa depende de la descodificación de las imágenes de una o más capas diferentes. Cuando se codifica una imagen en una de las capas no base, un codificador de vídeo (como el codificador de vídeo 20 o el descodificador de vídeo 30) puede agregar una imagen a una lista de imágenes de referencia si la imagen está en una visualización diferente pero en la misma instancia de tiempo (es decir, unidad de acceso) que la imagen que el codificador de vídeo está codificando actualmente. Al igual que otras imágenes de referencia de inter-predicción, el codificador de vídeo puede insertar una imagen de referencia de predicción entre visualizaciones en cualquier posición de una lista de imágenes de referencia.

**[0063]** En SVC, las capas distintas de la capa base pueden denominarse "capas de mejora" y pueden proporcionar información que mejora la calidad visual de los datos de vídeo descodificados a partir del flujo de bits. La SVC puede mejorar la resolución espacial, la relación señal-ruido (es decir, la calidad) o la velocidad temporal. En la codificación de vídeo escalable (por ejemplo, SHVC), una "representación de capa" puede ser una representación codificada de una capa espacial en una única unidad de acceso. Para facilitar la explicación, esta divulgación puede referirse a componentes de visualización y/o representaciones de capa como "componentes de visualización/representaciones de capa" o simplemente "imágenes".

**[0064]** Para implementar las capas, las cabeceras de las unidades NAL pueden incluir elementos sintácticos `nuh_reserved_zero_6bits`, que también pueden denominarse elementos sintácticos `nuh_layer_id`. Las unidades NAL que tienen `nuh_reserved_zero_6bit` elementos sintácticos que especifican valores diferentes que pertenecen a diferentes capas de un flujo de bits. Por lo tanto, en la codificación de múltiples visualizaciones (por ejemplo, MV-

HEVC), 3DV (por ejemplo, 3D-HEVC) o SVC (por ejemplo, SHVC), el elemento sintáctico `nuh_reserved_zero_6bits` de una unidad NAL especifica un identificador de capa (es decir, una ID de capa) de la unidad NAL. En algunos ejemplos, el elemento sintáctico `nuh_reserved_zero_6bits` de una unidad NAL es igual a 0 si la unidad NAL se relaciona con una capa base en codificación de múltiples visualizaciones (por ejemplo MV-HEVC), codificación 3DV (por ejemplo, 3D-HEVC) o SVC (por ejemplo, SHVC). Los datos en una capa base de un flujo de bits pueden descodificarse sin referencia a los datos en cualquier otra capa del flujo de bits. Si la unidad NAL no se relaciona con una capa base en codificación de múltiples visualizaciones, 3DV o SVC, el elemento sintáctico `nuh_reserved_zero_6bits` del elemento sintáctico puede tener otros valores distintos de cero. Como se indicó anteriormente, en la codificación de múltiples visualizaciones y codificación 3DV, diferentes capas de un flujo de bits pueden corresponder a diferentes visualizaciones.

**[0065]** Además, algunos componentes de visualización/representaciones de capa dentro de una capa pueden descodificarse sin referencia a otros componentes de visualización/representaciones de capa dentro de la misma capa. Por lo tanto, las unidades NAL que encapsulan datos de ciertos componentes de visualización/representaciones de una capa pueden eliminarse del flujo de bits sin afectar a la capacidad de descodificación de otros componentes de visualización/representaciones de capa en la capa. La eliminación de unidades NAL que encapsulan datos de tales componentes de visualización/representaciones de capa puede reducir la velocidad de tramas del flujo de bits. Un subconjunto de componentes de visualización/representaciones de capa dentro de una capa que puede descodificarse sin referencia a otros componentes de visualización/representaciones de capa dentro de la capa se puede denominar en el presente documento una "sub-capa" o una "sub-capa temporal".

**[0066]** Las unidades NAL pueden incluir elementos sintácticos `temporal_id` que especifiquen identificadores temporales (es decir, `TemporalIds`) de las unidades NAL. El identificador temporal de una unidad NAL identifica una subcapa a la que pertenece la unidad NAL. Por lo tanto, cada subcapa de un flujo de bits puede tener un identificador temporal diferente. En general, si el identificador temporal de una primera unidad NAL es menor que el identificador temporal de una segunda unidad NAL, los datos encapsulados por la primera unidad NAL pueden descodificarse sin referencia a los datos encapsulados por la segunda unidad NAL.

**[0067]** Tal como se indica brevemente más arriba, las unidades NAL pueden encapsular RBSP para conjuntos de parámetros de vídeo (VPS), conjuntos de parámetros de secuencia (SPS), conjuntos de parámetros de imagen (PPS). Un VPS es una estructura sintáctica que comprende elementos sintácticos que se aplican a cero o más secuencias de vídeo codificadas completas (CVS). Un SPS también es una estructura sintáctica que comprende elementos sintácticos que se aplican a cero o más CVS completos. Un SPS puede incluir un elemento sintáctico que identifica un VPS que está activo cuando el SPS está activo. Por lo tanto, los elementos sintácticos de un VPS pueden ser más en general aplicables que los elementos sintácticos de un SPS. Un PPS es una estructura sintáctica que comprende elementos sintácticos que se aplican a cero o más imágenes codificadas. Un PPS puede incluir un elemento sintáctico que identifica un SPS que está activo cuando el PPS está activo. Una cabecera de fragmento de un fragmento puede incluir un elemento sintáctico que indica un PPS que está activo cuando el fragmento está siendo codificado.

**[0068]** Los conjuntos de parámetros, tales VPS, SPS y PPS pueden incluir estructuras sintácticas de extensión específicas para las extensiones del estándar HEVC. Por ejemplo, los conjuntos de parámetros pueden incluir extensiones específicas para SHVC, MV-HEVC, 3D-HEVC, etc. Las estructuras sintácticas de extensión pueden incluir elementos sintácticos aplicables a las extensiones del estándar HEVC. La Tabla 1 a continuación muestra una parte de la extensión de VPS definida en SHVC WD6 y MV-HEVC WD8.

**TABLA 1**

<code>vps_extension() {</code>	Descriptor
<code>    avc_base_layer_flag</code>	<code>u(1)</code>
<code>    ...</code>	<code>u(1)</code>
<code>    vps_num_profile_tier_level_minus1</code>	<code>ue(v)</code>
<code>    for(ptIdx = 1; ptIdx &lt;= vps_num_profile_tier_level_Minus1; ptIdx++) {</code>	
<code>        vps_profile_present_flag[ptIdx]</code>	<code>u(1)</code>
<code>        profile_tier_level(vps_profile_resent_flag[ptIdx], vps_max_sub_layers_minus1)</code>	
<code>    }</code>	
<code>    ...</code>	
<code>    for(i = 1; i &lt; NumOutputLayerSets; i++) {</code>	

	Descriptor
vps_extension( ) {	
if(i > vps_num_layer_sets_minus1)	
<b>output_layer_set_idx_minus1[i]</b>	u(v)
if( i > vps_num_layer_sets_minus1    defaultTargetOutputLayerIdc == 2)	
for(j = 0; j < NumLayersInIdList[LayerSetIdxForOutputLayerSet[i]]; j++)	
<b>output_layer_flag[i][j]</b>	u(1)
<b>profile_level_tier_idx[i]</b>	u(v)
if(NumOutputLayersInOutputLayerSet[i] == 1	
&& NumDirectRefLayers[OlsHighestOutputLayerId[i]] > 0)	
<b>alt_output-layer_flag[i]</b>	u(1)
}	
...	
<b>vps_vui_present_flag</b>	
if(vps_vui_present_flag) {	
while(!byte_aligned( ))	
<b>vps_vui_alignment_bit_equal_to_one</b>	u(1)
vps_vui( )	
}	
}	

5 [0069] En el ejemplo de la Tabla 1 anterior y en otras tablas sintácticas de esta divulgación, los elementos sintácticos con descriptor de tipo ue(v) pueden ser números enteros de longitud variable sin signo codificados utilizando codificación Golomb exponencial (Exp-golomb) de orden 0 con el bit izquierdo primero. En el ejemplo de la Tabla 1 y las siguientes tablas, los elementos sintácticos que tienen descriptores de la forma u(n), donde n es un entero no negativo, son valores sin signo de longitud n.

10 [0070] La parte de la extensión de VPS definida en la Tabla 1 incluye elementos sintácticos que definen uno o más conjuntos de capas de salida. Un conjunto de capas de salida es un conjunto de capas que consta de las capas de uno de los conjuntos de capas especificados en el VPS, donde se indica que una o más capas del conjunto de capas serán capas de salida. Como se define en SHVC WD6, una capa de salida es una capa de un conjunto de capas de salida que se emite cuando TargetOptLayerSetIdx es igual al índice del conjunto de capas de salida. TargetOptLayerSetIdx especifica una lista de identificadores de capa de un punto de funcionamiento en uso y un conjunto de capas de salida objetivo. Un punto de funcionamiento puede referirse a un subconjunto (por ejemplo, un subconjunto adecuado) de unidades NAL en el flujo de bits. Además, como se define en SHVC WD6, una capa de salida objetivo es una capa que se va a emitir y es una de las capas de salida del conjunto de capas de salida con el índice olsIdx de forma que TargetOptLayerSetIdx sea igual a olsIdx. Por lo tanto, en general, el conjunto de capas de salida objetivo puede ser uno de los conjuntos de capas de salida en particular. Cada capa del conjunto de capas de salida objetivo se emite (por ejemplo, se proporciona como salida de un descodificador de vídeo). En algunos casos, el descodificador de vídeo 30 puede descodificar una capa particular de un flujo de bits, pero no emite datos de vídeo descodificados de la capa particular. En lugar de eso, el descodificador de vídeo 30 puede usar los datos de vídeo descodificados de la capa particular para descodificar los datos de vídeo de las capas que están en el conjunto de capas de salida objetivo.

25 [0071] Además, la parte de la extensión de VPS definida en la Tabla 1 incluye elementos sintácticos que definen una o más estructuras sintácticas de perfil, grado y nivel (PTL). Cada estructura sintáctica de PTL especifica un perfil, grado y nivel para cada secuencia de vídeo codificada. Como se muestra adicionalmente en la Tabla 1, la extensión de VPS incluye profile\_level\_tier\_idx elementos sintácticos que identifican las estructuras sintácticas PTL aplicables a los conjuntos de capas de salida respectivos. De esta forma, la extensión VPS especifica información PTL para cada uno de los conjuntos de capas de salida.

35 [0072] En general, un "perfil" es un subconjunto de las características y herramientas presentes en un estándar de codificación de vídeo (por ejemplo, HEVC o una extensión del mismo). Dentro de los límites impuestos por un perfil dado, todavía es posible requerir una gran variación en el rendimiento de los codificadores y descodificadores, dependiendo de los valores tomados por los elementos sintácticos en el flujo de bits, tales como el tamaño especificado de las imágenes descodificadas. En muchas aplicaciones, puede no ser ni práctico ni económico

implementar un descodificador capaz de tratar con todos los usos hipotéticos de la sintaxis dentro de un perfil particular. Para tratar con este problema, se especifican "grados" y "niveles" dentro de cada perfil.

**[0073]** En algunos ejemplos, un nivel es un conjunto definido de restricciones en los valores que pueden ser tomados por los elementos sintácticos y variables de un estándar de codificación de vídeo (por ejemplo, HEVC o una extensión del mismo). Además, en algunos ejemplos, un grado es una categoría especificada de restricciones de nivel impuestas a los valores de los elementos sintácticos en el flujo de bits o valores de variables, donde las restricciones de nivel están agrupadas dentro de un grado y un descodificador conforme a un cierto grado y nivel sería capaz de descodificar todos los flujos de bits que se ajusten al mismo grado o al grado inferior de ese nivel o cualquier nivel debajo del mismo. De este modo, un nivel de un grado es un conjunto especificado de restricciones impuestas en los valores de los elementos sintácticos en el flujo de bits o los valores de variables. Estas restricciones pueden ser simples limitaciones de valores. De forma alternativa, pueden adoptar la forma de restricciones sobre combinaciones aritméticas de valores (por ejemplo, el ancho de imagen multiplicado por la altura de la imagen multiplicada por el número de imágenes descodificadas por segundo). Un nivel especificado para un grado inferior es más restringido que un nivel especificado para un grado superior. En algunos contextos, el término "nivel" también se puede usar para referirse al valor de un coeficiente de transformación antes del escalado.

**[0074]** Como se muestra adicionalmente en la Tabla 1 anterior, la extensión VPS puede incluir una `vps_vui()` estructura sintáctica. La `vps_vui()` estructura sintáctica contiene parámetros de información de usabilidad de vídeo (VUI). El párrafo F.7.3.2.1.4 de SHVC WD6 describe la sintaxis de una `vps_vui()` estructura sintáctica. Como se especifica en la memoria descriptiva de borrador de HEVC, los parámetros VUI no son necesarios para construir las muestras de luma o croma mediante el proceso de descodificación. No se requieren descodificadores conformes para procesar esta información para el cumplimiento de la orden de salida de esta memoria descriptiva (ver el Anexo C para la especificación de la conformidad de la orden de salida). Se requieren algunos parámetros VUI para comprobar la conformidad del flujo de bits y la conformidad del descodificador de sincronización de salida. En algunos casos, los parámetros VUI también se pueden especificar en SPS.

**[0075]** Una `vps_vui()` la estructura sintáctica puede incluir una `vps_vui_bsp_hrd_parameters()` estructura sintáctica. La `vps_vui_bsp_hrd_parameters()` la estructura sintáctica incluye elementos sintácticos que especifican parámetros hipotéticos de descodificador de referencia (HRD) para las particiones de flujo de bits. Como se define en SHVC WD6, una partición de flujo de bits es una secuencia de bits en la forma de un flujo de unidad NAL o un flujo de bytes, que es un subconjunto de un flujo de bits de acuerdo con una partición.

**[0076]** Al menos algunos estándares de codificación de vídeo especifican modelos de memoria intermedia de vídeo. En H.264/AVC y HEVC, un modelo de almacenamiento en memoria intermedia se denomina "descodificador de referencia hipotético" o "HRD". En la memoria descriptiva de borrador de HEVC se describe el HRD en el anexo C. El HRD describe cómo se almacenarán en memoria intermedia los datos para la descodificación y cómo se almacenan en memoria intermedia los datos descodificados para la emisión. Por ejemplo, el HRD describe el funcionamiento de una memoria intermedia de imágenes codificadas ("CPB"), una memoria intermedia de imágenes descodificadas ("DPB") y un proceso de descodificación de vídeo. La CPB es una memoria intermedia de primero en entrar, primero en salir, que contiene unidades de acceso en un orden de descodificación especificado por HRD. La DPB es una memoria intermedia que contiene imágenes descodificadas para referencia, reordenamiento de salida o retardo de salida especificado por el HRD. Los comportamientos de la CPB y DPB se pueden especificar matemáticamente. El HRD puede imponer directamente restricciones en el tiempo, los tamaños de la memoria intermedia y las velocidades de transferencia de bits. Además, el HRD puede imponer restricciones indirectamente a varias características y estadísticas de flujo de bits.

**[0077]** En la H.264/AVC y la HEVC, la conformidad del flujo de bits y la conformidad del descodificador se especifican como partes de la especificación del HRD. En otras palabras, el modelo de HRD especifica pruebas para determinar si un flujo de bits cumple con el estándar y pruebas para determinar si un descodificador cumple con el estándar. Aunque el HRD se denomina como algún tipo de descodificador, los codificadores de vídeo típicamente usan el HRD para garantizar la conformidad del flujo de bits, mientras que los descodificadores de vídeo típicamente no necesitan el HRD.

**[0078]** H.264/AVC y HEVC especifican dos tipos de conformidad de HRD o flujo de bits; esto es, Tipo I y Tipo II. Un flujo de bits de Tipo I es un flujo de unidad NAL que contiene solo las unidades NAL VCL y la unidad NAL de datos de relleno para todas las unidades de acceso en el flujo de bits. En al menos algunos ejemplos, un flujo de unidades NAL es una secuencia de unidades NAL. Un flujo de bits de Tipo II es un flujo de unidad NAL que contiene, además de las unidades NAL VCL y unidades NAL de datos de relleno para todas las unidades de acceso en el flujo de bits, al menos uno de los siguientes: unidades NAL no VCL adicionales distintas de unidades NAL de datos de relleno; y todos los elementos sintácticos `leading_zero_8bits`, `zero_byte`, `start_coded_prefix_one_3bytes` y `trailing_zero_8bits` que forman una secuencia de bytes desde la secuencia de la unidad NAL.

**[0079]** Cuando un dispositivo realiza una prueba de conformidad de flujo de bits que determina si un flujo de bits se ajusta a un estándar de codificación de vídeo, el dispositivo puede seleccionar un punto de funcionamiento del flujo de bits. El dispositivo puede entonces determinar un conjunto de parámetros de HRD aplicables al punto de

funcionamiento seleccionado. El dispositivo puede usar el conjunto de parámetros de HRD aplicable al punto de funcionamiento seleccionado para configurar el comportamiento del HRD. Más particularmente, el dispositivo puede usar el conjunto aplicable de parámetros de HRD para configurar los comportamientos de componentes particulares del HRD, tales como un programador de flujo hipotético (HSS), la CPB, un proceso de descodificación, la DPB, etc. Posteriormente, el HSS puede inyectar datos de vídeo codificados del flujo de bits en la CPB del HRD de acuerdo con un programa particular. Además, el dispositivo puede invocar un proceso de descodificación que descodifica los datos de vídeo codificados en la CPB. El proceso de descodificación puede enviar imágenes descodificadas a la DPB. A medida que el dispositivo mueve datos a través del HRD, el dispositivo puede determinar si sigue cumpliéndose un conjunto particular de restricciones. Por ejemplo, el dispositivo puede determinar si se produce una condición de sobreflujo o sub-flujo en la CPB o DPB mientras el HRD está descodificando el punto de funcionamiento del punto de funcionamiento seleccionado. El dispositivo puede seleccionar y procesar cada punto de funcionamiento del flujo de bits de esta manera. Si ningún punto de funcionamiento del flujo de bits hace que se violen las restricciones, el dispositivo puede determinar que el flujo de bits se ajusta al estándar de codificación de vídeo.

**[0080]** H.264/AVC y HEVC especifican dos tipos de conformidad de descodificador; esto es, la conformidad de descodificador de sincronización de salida y la conformidad de descodificador de orden de salida. Un descodificador que alega conformidad con un perfil, grado y nivel específico, es capaz de descodificar con éxito todos los flujos de bits que se ajustan a los requisitos de conformidad de flujo de bits de un estándar de codificación de vídeo, como HEVC.

**[0081]** Cuando un dispositivo realiza una prueba de conformidad de descodificador para determinar si un descodificador bajo prueba (DUT) se ajusta a un estándar de codificación de vídeo, el dispositivo puede proporcionar, tanto al HRD como al DUT, un flujo de bits que se ajusta al estándar de codificación de vídeo. El HRD puede procesar el flujo de bits de la manera descrita anteriormente con respecto a la prueba de conformidad de flujo de bits. El dispositivo puede determinar que el DUT se ajusta al estándar de codificación de vídeo si el orden de las imágenes descodificadas emitidas por el DUT coincide con el orden de las imágenes descodificadas emitidas por el HRD. Además, el dispositivo puede determinar que el DUT se ajusta al estándar de codificación de vídeo si el tiempo con el cual el DUT envía imágenes descodificadas coincide con el tiempo con el cual el HRD emite las imágenes descodificadas.

**[0082]** En los modelos H.264/AVC y HEVC HRD, la descodificación o eliminación de CPB pueden basarse en unidad de acceso. Es decir, se supone que el HRD descodifica unidades de acceso completas de una vez y elimina unidades de acceso completas de la CPB. Además, en los modelos H.264/AVC y HEVC HRD, se supone que la descodificación de imágenes es instantánea. El codificador de vídeo 20 puede indicar, en los mensajes SEI de sincronización de imágenes, los tiempos de descodificación para comenzar la descodificación de las unidades de acceso. En aplicaciones prácticas, si un descodificador de vídeo en conformidad sigue estrictamente los tiempos de descodificación indicados para iniciar la descodificación de unidades de acceso, el primer momento posible para emitir una imagen descodificada específica es igual al momento de descodificación de esa imagen particular, más el tiempo necesario para descodificar esa imagen particular. Sin embargo, en el mundo real, el tiempo necesario para descodificar una imagen no puede ser igual a cero.

**[0083]** Los parámetros de HRD pueden controlar diversos aspectos del HRD. En otras palabras, el HRD puede confiar en los parámetros de HRD. Los parámetros de HRD pueden incluir un retardo inicial de eliminación de la CPB, un tamaño de CPB, una velocidad de transferencia de bits, un retardo inicial de salida de la DPB y un tamaño de DPB. El codificador de vídeo 20 puede señalar estos parámetros de HRD en una estructura sintáctica `hrd_parameters()` especificada en un conjunto de parámetros de vídeo (VPS) y/o un conjunto de parámetros de secuencia (SPS). Los VPS y/o SPS individuales pueden incluir múltiples estructuras sintácticas `hrd_parameters()` para diferentes conjuntos de parámetros de HRD. En algunos ejemplos, el codificador de vídeo 20 puede indicar parámetros de HRD en mensajes SEI de período de almacenamiento en memoria intermedia o mensajes SEI de sincronización de imágenes.

**[0084]** Un punto de funcionamiento de un flujo de bits está asociado con un conjunto de identificadores de capa (es decir, un conjunto de valores `nuh_reserved_zero_6bits`, que también se conocen como valores `nuh_layer_id`) y un identificador temporal. Un punto de funcionamiento puede incluir cada unidad NAL que está asociada con el punto de funcionamiento. Un punto de funcionamiento puede tener una velocidad de tramas y/o velocidad de transferencia de bits diferente a la de un flujo de bits original. Esto se debe a que el punto de funcionamiento puede no incluir algunas imágenes y/o algunos de los datos del flujo de bits original. Por lo tanto, si el descodificador de vídeo 30 fuera a eliminar datos de la CPB, y/o la DPB a una velocidad particular cuando procesa el flujo de bits original y si el descodificador de vídeo 30 fuera a eliminar datos de la CPB y/o la DPB a la misma velocidad cuando procesa un punto de funcionamiento, el descodificador de vídeo 30 puede eliminar demasiados o muy pocos datos de la CPB y/o la DPB. En consecuencia, el codificador de vídeo 20 puede indicar diferentes conjuntos de parámetros de HRD para diferentes puntos de funcionamiento. Por ejemplo, el codificador de vídeo 20 puede incluir, en un VPS, múltiples estructuras sintácticas `hrd_parameters()` que incluyen parámetros de HRD para diferentes puntos de funcionamiento.

[0085] La Tabla 2, a continuación, es una sintaxis de ejemplo para una estructura sintáctica hrd\_parameters( ) en HEVC.

5

**TABLA 2- Parámetros de HRD**

	<b>Descriptor</b>
hrd_parameters(commonInfPresentFlag, MaxNumSubLayersMinus1) {	
if(commonInfPresentFlag) {	
<b>timing_info_present_flag</b>	u(1)
if(timing_info_present_flag) {	
<b>num_units_in_tick</b>	u(32)
<b>time_scale</b>	u(32)
}	
<b>nal_hrd_parameters_present_flag</b>	u(1)
<b>vcl_hrd_parameters_present_flag</b>	u(1)
if(nal_hrd_parametersresent_flag	
vcl_hrd_parameters_present_flag) {	
<b>sub_pic_cpb_params_present_flag</b>	u(1)
if(sub_pic_cpb_params_present_flag) {	
<b>tick_divisor_minus2</b>	u(8)
<b>du_cpb_removal_delay_length_minus1</b>	u(5)
}	
<b>bit_rate_scale</b>	u(4)
<b>cpb_size_scale</b>	u(4)
<b>initial_cpb_removal_delay_length_minus1</b>	u(5)
<b>cpb_removal_delay_length_minus1</b>	u(5)
<b>dpb_output_delay_length_minus1</b>	u(5)
}	
}	
for(i = 0; i <= MaxNumSubLayersMinus1; i++) {	
<b>fixed_pic_rate_flag[i]</b>	u(1)
if(fixed_pic_rate_flag[i])	
<b>pic_duration_in_tc_minus1[i]</b>	ue(v)
<b>low_delay_hrd_flag[i]</b>	u(1)
<b>cpb_cnt_minus1[i]</b>	ue(v)
if(nal_hrd_parameters_present_flag)	
sub_layer_hrd_parameters(i)	
if(vcl_hrd_parameters_present_flag)	
sub_layer_hrd_parameters(i)	
}	
}	

[0086] En algunos ejemplos, el decodificador de vídeo 30 comprende una pluralidad de núcleos de procesamiento. Cada uno de los núcleos de procesamiento comprende una unidad de procesamiento independiente, tal como un ASIC o un microprocesador. Cada uno de los núcleos de procesamiento puede implementar un



descodificador de capa única. Un descodificador de capa única es un codificador de vídeo capaz de descodificar unidades NAL que pertenecen a una capa única de un flujo de bits. En dichos ejemplos, dado que el descodificador de vídeo 30 comprende múltiples núcleos de procesamiento, y por lo tanto múltiples descodificadores de capa única, el descodificador de vídeo 30 puede descodificar múltiples capas en paralelo. En algunos casos, un núcleo de hardware individual del descodificador de vídeo 30 puede ser capaz de descodificar más de una capa.

**[0087]** Diferentes núcleos de procesamiento pueden tener diferentes capacidades. Por ejemplo, diferentes núcleos de procesamiento pueden descodificar datos de vídeo que pertenecen a diferentes perfiles, grados o niveles. Por ejemplo, un primer núcleo de procesamiento del descodificador de vídeo 30 puede procesar datos de vídeo que pertenecen a un primer nivel, pero no a un segundo nivel. En este ejemplo, un segundo núcleo de procesamiento del descodificador 30 de vídeo puede ser capaz de procesar datos de vídeo que pertenecen al primer nivel y al segundo nivel. En general, los núcleos de procesamiento capaces de descodificar datos de vídeo que pertenecen a perfiles, grados y niveles superiores son más complejos y, por lo tanto, más costosos que los núcleos de procesamiento que solo pueden descodificar datos de vídeo que pertenecen a perfiles, grados y niveles inferiores. En consecuencia, un fabricante de descodificador de vídeo 30 puede elegir incluir en el descodificador de vídeo 30 núcleos de procesamiento que tengan las capacidades más bajas adecuadas para manejar una tarea deseada.

**[0088]** Wang et al., "MV-HEVC/SHVC HLS: On level definition," Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11 17.<sup>a</sup> reunión: Valencia, ES, 27 de marzo - 4 de abril de 2014, documento n.º JCTVC-Q0145 (en adelante, "JCTVC-Q0145") describe varios problemas relacionados con las definiciones de niveles en SHVC.

**[0089]** En primer lugar, si se necesitan más de 4 capas de relación señal/ruido (SNR) de 720p de resolución (o un número equivalente de píxeles luma, por ejemplo, más capas con capas espaciales y SNR combinadas), deberá utilizarse el nivel 5 o superior. En consecuencia, el valor de CtbSizeY será igual a 32 o 64 (por elemento e de los límites generales). Para una resolución como 720p o inferior, esta restricción no permite el uso de tamaños de bloques de árbol de codificación más pequeños, como 16x16, y por lo tanto puede dar como resultado una eficiencia de codificación inferior a la óptima.

**[0090]** En segundo lugar, si se implementa un descodificador SHVC de una manera que consta de 4 núcleos de descodificadores HEVC de nivel 3.1, y es capaz de descodificar 4 capas SNR de 720p, según la definición de nivel actual, debería decirse que se ajusta al nivel 4 o más. En consecuencia, el descodificador puede necesitar poder descodificar cualquier flujo de bits de Nivel 4. Sin embargo, con solo cambios sintácticos de alto nivel (es decir, sin cambios en los núcleos de hardware), dicho descodificador no podría descodificar un flujo de bits de nivel 4 de SHVC con 2 capas de SNR de resolución de 1080p.

**[0091]** En tercer lugar, si un descodificador SHVC no se implementa reutilizando múltiples núcleos de descodificadores HEVC existentes, sino que se implementa de una manera que puede descodificar tanto un flujo de bits HEVC de capa única de 1080p como un flujo de bits SHVC de dos capas de 720p, de acuerdo con la definición de nivel actual se etiquetaría como un nivel 3.1. Sin embargo, entonces falta la expresión de la otra capacidad.

**[0092]** En cuarto lugar, si se implementa un descodificador SHVC de una manera que consta de 4 núcleos de descodificadores HEVC 3.1 para poder descodificar 4 capas SNR de 720p, según la definición de nivel actual, se diría que se ajusta al Nivel 4 o superior. En consecuencia, cada imagen de capa de mejora puede tener más de 3 filas de elementos y más de 3 columnas de elementos, por ejemplo, 5 filas de elementos y 5 columnas de elementos, con cada elemento con un ancho de 256 muestras de luma y una altura de 144 muestras de luma. Sin embargo, esto iría más allá de los límites del Nivel 3.1 y, en consecuencia, los núcleos descodificadores para descodificar las capas de mejora tendrían problemas.

**[0093]** En quinto lugar, todos los elementos en la subcláusula A.4.1 de la memoria descriptiva de borrador de HEVC están actualmente especificados para ser aplicados a cada capa. Sin embargo, algunos de estos elementos no son directamente aplicables a cada capa. Para el elemento d en el tamaño de DPB, el elemento sintáctico SPS no se aplica a las capas de mejora. Además, la DPB en el SHVC WD6 es un diseño compartido-sub-DPB, por lo que el elemento d no se puede aplicar directamente a cada capa. Para los elementos h e i en el tamaño de CPB, para los funcionamientos de CPB específicos del flujo de bits, el parámetro no se puede aplicar a cada capa.

**[0094]** En sexto lugar, se necesitan restricciones específicas del flujo de bits en el tamaño de CPB (mediante los puntos h e i en la subcláusula A.4.1 de la memoria descriptiva de borrador de HEVC). Sin embargo, los elementos h e i en la subcláusula A.4.1 de la memoria descriptiva de borrador de HEVC no pueden aplicarse directamente a nivel de flujo de bits, porque si se aplican directamente, el mismo límite de tamaño de CPB para flujos de bits de capa única también sería el límite para flujos de bits de múltiples capas. Esto no es escalable a la cantidad de capas y solo permite una baja calidad de imagen cuando hay muchas capas.

**[0095]** En séptimo lugar, las restricciones mediante los elementos b, c, d, g, h, i y j en la subcláusula A.4.2 de la memoria descriptiva de borrador de HEVC están especificadas actualmente para ser específicas de la capa

solamente. Sin embargo, se debe especificar la restricción específica de flujo de bits mediante estos elementos, independientemente de si se especifican sus contrapartes específicas de capa.

5 **[0096]** JVTVC-Q0145 propuso un enfoque para definir los niveles y las capacidades del descodificador, así como otros cambios para resolver los problemas anteriores. El diseño en JCTVC-Q0145 está asociado con al menos los siguientes problemas. En primer lugar, el diseño no es óptimo para los descodificadores SHVC/MV-HEVC, cada uno de los cuales se construye basándose en múltiples descodificadores HEVC de capa única sin cambiar los núcleos de hardware (por ejemplo, sin modificar las capacidades de los núcleos de hardware) donde los múltiples descodificadores de capa única de HEVC tienen diferentes capacidades, por ejemplo, diferentes perfiles, grados o niveles. En segundo lugar, los niveles se pueden definir en términos de restricciones. MV-HEVC WD8 y SHVC WD6 definen muchas de estas restricciones en términos de los elementos sintácticos o variables que tienen o no tienen valores particulares al descodificar un flujo de bits. En otras palabras, algunas de las restricciones de nivel (por ejemplo, velocidad de bits y frecuencia de muestreo) se especifican para ser específicas de flujo de bits. Sin embargo, para habilitar un descodificador HEVC de capa única para ser parte de un descodificador SHVC/MV-HEVC sin cambiar el núcleo del hardware y descodificar un subconjunto de capas (es decir, una o más capas, pero no todas las capas de un conjunto de capas de salida), tales restricciones de nivel pueden necesitar ser específicas para el subconjunto de capas.

20 **[0097]** A continuación se proporciona una descripción general de diversas características de acuerdo con diversas técnicas de esta divulgación. Algunas implementaciones detalladas de acuerdo con uno o más ejemplos de esta divulgación se proporcionan en secciones posteriores de esta divulgación. Algunas de las características indicadas a continuación pueden aplicarse independientemente y algunas de las características indicadas a continuación se pueden aplicar en combinación como reconocería un experto en la materia.

25 **[0098]** Una técnica de ejemplo de esta divulgación proporciona la señalización de uno o más esquemas de partición de capas para cada conjunto de capas de salida, con cada esquema de partición que contiene un conjunto de particiones disjuntas cuya unión forma el conjunto de capas de salida y cada partición que contiene una o más capas. En otras palabras, cada esquema de partición de una capa de salida específica una forma diferente de dividir las capas del conjunto de capas de salida en particiones que no se superponen. En algunos ejemplos, las particiones se asocian con conjuntos de capas de salida en lugar de conjuntos de capas.

30 **[0099]** En algunos ejemplos, un esquema de partición y una unidad de partición se definen de la forma siguiente: Un conjunto de capas de salida puede tener uno o más esquemas de partición. Un esquema de partición de un conjunto de capas de salida divide capas en el conjunto de capas de salida en una o más particiones. Cada capa en un conjunto de capas de salida debe pertenecer exactamente a una partición en un esquema de partición.

35 **[0100]** Por lo tanto, en un ejemplo, el codificador de vídeo 20 genera un flujo de bits que comprende una representación codificada de datos de vídeo. Como parte de generar el flujo de bits, el codificador de vídeo 20 incluye, en el flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición. Para cada esquema de partición respectivo del uno o más esquemas de partición, el esquema de partición respectivo especifica un conjunto respectivo de particiones disjuntas cuya unión forma un conjunto de capas de salida. En este ejemplo, cada partición respectiva del conjunto respectivo de particiones disjuntas contiene una o más capas, el conjunto de capas de salida consiste en todas las capas de la pluralidad de capas o un subconjunto adecuado de la pluralidad de capas. Además, el codificador de vídeo 20 emite el flujo de bits.

40 **[0101]** En un ejemplo similar, el descodificador de vídeo 30 descodifica un flujo de bits que comprende una representación codificada de datos de vídeo. En este ejemplo, el flujo de bits comprende una pluralidad de capas. Como parte de la descodificación del flujo de bits en este ejemplo, el descodificador de vídeo 30 obtiene, a partir del flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición. En algunos ejemplos, el descodificador de vídeo 30 descodifica cada una de las particiones de un esquema de partición particular usando diferentes núcleos de procesamiento en una pluralidad de núcleos de hardware, siendo el esquema de partición particular uno de los uno o más esquemas de partición.

45 **[0102]** Además, en algunos ejemplos, el codificador de vídeo 20 puede incluir, en un flujo de bits, elementos sintácticos que indican una pluralidad de conjuntos de capas de salida. Para cada conjunto de capas de salida respectivo de la pluralidad de conjuntos de capas de salida, el codificador de vídeo 20 puede incluir, en el flujo de bits, elementos sintácticos que indican uno o más esquemas de partición para el conjunto de capas de salida respectivo. En un ejemplo similar, el descodificador de vídeo 30 puede obtener, a partir del flujo de bits, elementos sintácticos que indican una pluralidad de conjuntos de capas de salida, con la pluralidad de conjuntos de capas de salida que incluye el primer conjunto de capas de salida. En este ejemplo, para cada conjunto de capas de salida respectivo de la pluralidad de conjuntos de capas de salida, el descodificador de vídeo 30 puede obtener, a partir del flujo de bits, elementos sintácticos que indican uno o más esquemas de partición para el conjunto de capas de salida respectivo.

60 **[0103]** De acuerdo con algunos ejemplos de esta divulgación, los términos "esquema de partición" y "unidad de partición" tienen las siguientes definiciones:

esquema de partición: Una división de capas en un conjunto de capas de salida en una o más particiones de modo que cada capa en el conjunto de capas de salida se incluye exactamente en una partición del esquema de partición y cada partición del esquema de partición contiene una o más capas.

5 unidad de partición: Un subconjunto de unidad de acceso que contiene un conjunto de unidades NAL de acuerdo con un esquema de partición.

10 **[0104]** Como se indicó anteriormente, una extensión de VPS puede especificar información de PTL respectiva para cada conjunto de capas de salida respectivo. Por ejemplo, la extensión VPS puede incluir, para cada conjunto de capas de salida respectivo, un elemento sintáctico `profile_level_tier_idx` para el conjunto de capas de salida respectivo que identifica una estructura sintáctica de PTL para el conjunto de capas de salida respectivo. La estructura sintáctica de PTL para el conjunto de capas de salida respectivo especifica la información de PTL para el conjunto de capas de salida respectivo.

15 **[0105]** Debido a que el descodificador de vídeo 30 puede usar la información de PTL de un conjunto de capas de salida para determinar si el descodificador de vídeo 30 puede descodificar el conjunto de capas de salida, la información de PTL del conjunto de capas de salida especifica típicamente el perfil, grado y nivel más bajos necesarios para descodificar el conjunto de capas de salida como un todo. Sin embargo, el perfil, el grado y el nivel de algunas particiones del conjunto de capas de salida pueden ser más bajos que el perfil, el grado y el nivel del conjunto de capas de salida como un todo. En otras palabras, el perfil, grado y nivel del conjunto de capas de salida puede ser el perfil, grado y nivel más alto de la partición en el esquema de partición del conjunto de capas de salida, donde cada una de las capas del conjunto de capas de salida pertenece a una partición.

25 **[0106]** Por lo tanto, cuando las diferentes particiones de un conjunto de capas de salida tienen información de PTL diferente, la información de PTL para el conjunto de capas de salida puede no ser suficiente para expresar la diversidad de información de PTL de las particiones del conjunto de capas de salida. Una o más técnicas de esta divulgación pueden proporcionar la señalización de un perfil, un grado y un nivel para cada partición. Por ejemplo, para cada partición respectiva de cada esquema de partición respectivo señalado en un flujo de bits, el codificador de vídeo 20 puede incluir, en la información de flujo de bits, el perfil, grado y nivel (PTL) para la partición respectiva. De forma similar, para cada partición respectiva de cada esquema de partición respectivo señalado en un flujo de bits, el descodificador de vídeo 30 puede obtener, a partir del flujo de bits, un elemento sintáctico respectivo (por ejemplo, `profile_level_tier_partition_idx`) que indica la información de PTL respectiva para la partición respectiva.

35 **[0107]** En algunos ejemplos, el descodificador de vídeo 30 determina, basándose en la información de PTL para las particiones de los esquemas de partición, un esquema de partición del uno o más esquemas de partición. Además, en algunos ejemplos, el esquema de partición se determina basándose en las condiciones de la red que se usa para enviar el flujo de bits al descodificador. El descodificador de vídeo 30 puede descodificar cada partición respectiva del esquema de partición determinante usando un núcleo de hardware respectivo. Cada uno de los núcleos de hardware puede implementar un descodificador de capa única respectivo. En algunos ejemplos, los núcleos de hardware individuales descodifican más de una capa (por ejemplo, implementan un descodificador de múltiples capas). Además, diferentes núcleos de hardware pueden tener diferentes capacidades. Por ejemplo, al menos un primer núcleo de hardware de una pluralidad de núcleos de hardware del descodificador de vídeo 30 puede soportar un perfil, grado o nivel diferente de un segundo núcleo de hardware de la pluralidad de núcleos de hardware del descodificador de vídeo 30. Por lo tanto, en un ejemplo, el descodificador de vídeo 30 puede comprender un primer núcleo de hardware y un segundo núcleo de hardware. En este ejemplo, el primer núcleo de hardware puede implementar un descodificador de nivel 3 (es decir, un descodificador capaz de descodificar HEVC de nivel 3 e inferior) y el segundo núcleo de hardware puede implementar un descodificador de nivel 2 (es decir, un descodificador capaz de descodificar HEVC de nivel 2 e inferior). En este ejemplo, si la información PTL de una primera partición indica que la primera partición puede descodificarse mediante un descodificador de nivel 3 o superior y si la información PTL indica que una segunda partición puede descodificarse mediante un descodificador de nivel 2 o superior, el descodificador de vídeo 30 puede usar el primer núcleo de hardware para descodificar la primera partición y puede usar el segundo núcleo de hardware para descodificar la segunda partición. En algunos casos, cuando un descodificador de vídeo de múltiples capas que incluye uno o más núcleos de hardware para descodificar las particiones, y al menos uno de los núcleos del descodificador es conforme a uno de los perfiles escalables (por ejemplo, escalable principal o escalable principal 10) especificados, puede incluirse la capacidad para realizar un muestreo o un muestreo superior de imágenes. Esta capacidad de muestreo puede implementarse en cada núcleo que es capaz de descodificar un perfil escalable, o implementarse como bloque común en el descodificador de múltiples capas.

60 **[0108]** Tal como se describió brevemente anteriormente, el codificador de vídeo 20 puede incluir, en un VPS, elementos sintácticos que identifican conjuntos de capas de salida. Por lo tanto, los conjuntos de capas de salida pueden señalarse en un VPS. En algunos ejemplos, los conjuntos de capas de salida también pueden señalarse mediante medios externos. Los medios externos pueden ser algún mecanismo que no sea el flujo de bits de vídeo. Por lo tanto, un conjunto de capas de salida señalado por un medio externo y que comprende capas de salida de un flujo de bits no se señala necesariamente en el flujo de bits.

[0109] De acuerdo con una o más técnicas de esta divulgación, para cada esquema de partición respectivo de cada conjunto de capas de salida, un mensaje de VPS VUI o SEI de un flujo de bits puede identificar un conjunto de parámetros de HRD para el esquema de partición respectivo. Por lo tanto, en algunos ejemplos de esta divulgación, los parámetros de HRD para cada esquema de partición para cada conjunto de capas de salida señalado en el VPS o mediante medios externos se señalan en la VUI de VPS o en los mensajes SEI. Debido a que los conjuntos de parámetros de HRD se identifican para cada esquema de partición, el descodificador de vídeo 30 u otro dispositivo puede usar los parámetros de HRD para el esquema de partición para probar si el flujo de bits es un flujo de bits conforme cuando se divide de acuerdo con el esquema de partición.

[0110] En uno de tales ejemplos, el codificador de vídeo 20 genera un flujo de bits que comprende una representación codificada de los datos de vídeo. En este ejemplo, el flujo de bits comprende una pluralidad de capas. Como parte de generar el flujo de bits en este ejemplo, el codificador de vídeo 20 incluye, en el flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición para un conjunto de capas de salida. En este ejemplo, el conjunto de capas de salida consiste en todas las capas de la pluralidad de capas o un subconjunto adecuado de la pluralidad de capas. Para cada esquema de partición respectivo del uno o más esquemas de partición, el esquema de partición respectivo especifica una división del conjunto de capas de salida en una o más particiones, en el que cada capa en el conjunto de capas de salida pertenece exactamente a una partición en el esquema de partición respectivo. Además, como parte de generar el flujo de bits en este ejemplo, el codificador de vídeo 20 puede incluir, para cada esquema de partición respectivo, en un mensaje SEI en el flujo de bits o VUI en un VPS en el flujo de bits, un conjunto respectivo de elementos sintácticos que indican los conjuntos de parámetros de HRD para el esquema de partición respectivo. El codificador de vídeo 20 puede emitir el flujo de bits.

[0111] En un ejemplo similar, un dispositivo de procesamiento de vídeo, tal como el codificador de vídeo 20 o el descodificador de vídeo 30, puede descodificar un flujo de bits que comprende una representación codificada de los datos de vídeo. En este ejemplo, el flujo de bits comprende una pluralidad de capas. Como parte de la descodificación del flujo de bits en este ejemplo, el dispositivo de procesamiento de vídeo obtiene, a partir del flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición para un conjunto de capas de salida. Adicionalmente, para cada esquema de partición respectivo, el descodificador de vídeo 30 puede obtener, a partir de un mensaje SEI en el flujo de bits o información de usabilidad de vídeo (VUI) en un VPS en el flujo de bits, un conjunto respectivo de conjuntos de parámetros de HRD para el esquema de partición respectivo. Para cada partición respectiva del esquema de partición respectivo, el dispositivo de procesamiento de vídeo prueba, basándose en los parámetros de HRD para el esquema de partición respectivo, la partición respectiva en cuanto a conformidad con un estándar de codificación de vídeo.

[0112] En al menos algunos documentos estándar, como SHVC WD6, MV-HEVC WD8 y la memoria descriptiva de borrador de HEVC, los perfiles, grados y niveles se definen en términos de restricciones y límites de nivel. Un límite de nivel puede especificar un límite en el valor de un elemento sintáctico o variable. Una restricción puede especificar un requisito o restricción. Por ejemplo, JCTV3V-H1002 define un perfil "estéreo principal" e incluye, entre otras restricciones, una restricción que requiere que el flujo de bits contenga una capa con `nuh_layer_id` igual a *i* para la cual `ViewScalExtLayerFlag[i]` es igual a 1. Muchos de los límites y restricciones de nivel especificados en SHVC WD6, MV-HEVC WD8 y la memoria descriptiva de borrador de HEVC son aplicables a los flujos de bits. Por ejemplo, el párrafo G.11.1.2 de MV-HEVC WD8 describe el perfil "estéreo principal" y especifica, con énfasis añadido, "Los **flujos de bits conforme** al perfil estéreo principal deberán cumplir las siguientes restricciones..."

[0113] Sin embargo, en algunos ejemplos de esta divulgación, diferentes particiones de flujo de bits pueden tener diferente información de PTL. Para soportar diferentes particiones de flujo de bits que tienen diferente información de PTL, las definiciones de los límites de nivel y las restricciones pueden expresarse en términos de particiones en lugar de flujos de bits. Así, de acuerdo con algunas técnicas de ejemplo de esta divulgación, todos los límites y restricciones de nivel, excepto los tres de ancho de la imagen, altura de la imagen y tamaño de la memoria intermedia de imágenes subdescodificadas (DPB) que son específicos de cada capa, se especifican como específico de partición. Por ejemplo, todos los límites y restricciones de nivel de un perfil, grado o nivel, excepto los límites y restricciones del ancho de la imagen, la altura de la imagen y el tamaño de sub-DPB, se definen como específicos de la partición.

[0114] Como se indicó anteriormente, las restricciones de tamaño de sub-DPB son específicas de cada capa. Para elaborar, la DPB puede dividirse virtualmente en sub-DPB. Las imágenes de cada capa se almacenan en una sub-DPB independiente y cada sub-DPB contiene una capa. Por ejemplo, un conjunto de capas de salida que contiene dos capas, la DPB puede dividirse en dos sub-DPB, con la primera sub-DPB que contiene la primera capa y la segunda sub-DPB que contiene la segunda capa. En lugar de especificar el tamaño de DPB como un total, el tamaño de sub-DPB (el requerido para cada capa) se especifica en el flujo de bits. En dichos ejemplos, los límites de nivel ponen indirectamente algunas restricciones en el tamaño general de DPB.

[0115] De acuerdo con una o más técnicas de esta divulgación, se introducen una o más restricciones adicionales. Una posible restricción, por ejemplo, es que para cada conjunto de capas de salida, solo habrá un esquema de partición que contenga solo una partición y esa partición contiene todas las capas en el conjunto de capas de salida.

Cada esquema de partición está dando flexibilidad de un programa de entregas, es decir, qué capas se pueden entregar juntas. Cuando todas las capas están en una partición, la única forma en que puede tener diferentes programas de entregas es teniendo diferentes velocidades de transmisión de bits, por ejemplo. Como resultado, esta restricción ayudaría a reducir alguna sintaxis adicional.

5 [0116] En otra restricción de ejemplo, la suma de los tamaños de imagen de una partición está restringida a ser menor que las muestras de luma máximas especificadas para el nivel de la partición. Esta restricción puede asegurar que si una partición con múltiples capas es descodificada por un único núcleo, no se exceden los tamaños generales de DPB del núcleo actual. Además, esta restricción puede asegurar que el tamaño total de las imágenes de todas las capas no exceda el tamaño de la imagen de una capa que puede haber sido descodificable por el descodificador de capa única. Esta restricción puede permitir una extensión más fácil de núcleos de capa única para propósitos de múltiples capas (por ejemplo, sin tener que agregar potencialmente memoria adicional).

10 [0117] En otra restricción de ejemplo, los tripletes señalados de perfil, grado y nivel asociados con el conjunto de capas de salida también están asociados con la partición que contiene todas las capas en el conjunto de capas de salida.

15 [0118] Además, de acuerdo con una técnica de ejemplo de esta divulgación, una capacidad de descodificación de un descodificador se expresa como conforme a una lista de tripletes de perfil, grado y nivel. En algunos ejemplos, el número de entradas en la lista indica el número de descodificadores de capa única utilizados para construir el descodificador de múltiples capas, y cada triplete de perfil, grado y nivel indica el perfil, grado y nivel de uno de los descodificadores de capa única.

20 [0119] Así, en un ejemplo, un dispositivo de procesamiento de vídeo, como el descodificador de vídeo 30 u otro dispositivo, puede determinar, basándose en una lista de entradas triples, si un descodificador de vídeo de múltiples capas (por ejemplo, descodificador de vídeo 30) es capaz de descodificar una flujo de bits que comprende una representación codificada de los datos de vídeo de múltiples capas. El número de entradas triples en la lista puede ser menor o igual que una cantidad de descodificadores de capa única en el descodificador de vídeo de múltiples capas. Cada entrada triple respectiva en la lista de entradas triples indica un perfil, un grado y un nivel para un descodificador de capa única respectivo en el descodificador de vídeo de múltiples capas. En algunos ejemplos, el dispositivo de procesamiento de vídeo puede determinar que el descodificador de múltiples capas es capaz de descodificar el flujo de bits determinando que el flujo de bits especifica un esquema de partición donde la información PTL de cada partición del esquema de partición indica que la partición puede descodificarse al menos uno de los descodificadores de capa única. Basándose en que el descodificador de vídeo de múltiples capas es capaz de descodificar el flujo de bits, el descodificador de vídeo de múltiples capas puede descodificar el flujo de bits.

25 [0120] En algunos ejemplos de esta divulgación, se requiere un descodificador que se ajuste a una lista de tripletes de perfil, grado y nivel para poder descodificar cualquier conjunto de capas de salida conformes para el que haya al menos un esquema de partición que satisfaga la siguiente condición: para cada partición del esquema de partición, uno de los descodificadores de capa única del descodificador de múltiples capas se puede asignar para descodificar la partición. Cuando a un descodificador de capa única se le asigna una partición para descodificación, el descodificador de capa única puede descodificar la partición. En algunos ejemplos, cada una de las particiones puede asignarse a uno diferente de los descodificadores de capa única. En otros ejemplos, dos o más particiones pueden asignarse a uno solo de los descodificadores de capa única. Además, en algunos ejemplos, el número de tripletes PTL anunciados por el descodificador y su asociación de cada triplete a un núcleo es solo una posibilidad. En dichos ejemplos, el descodificador puede tener cualquier implementación interna (por ejemplo, solo un supernúcleo) pero todavía publicitar múltiples tripletes PTL.

30 [0121] Por lo tanto, en algunos ejemplos, un dispositivo de procesamiento de vídeo, tal como un descodificador de vídeo 30 u otro dispositivo, puede obtener, a partir de un flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición. Para determinar si un descodificador de vídeo de múltiples capas, como el descodificador de vídeo 30, es capaz de descodificar el flujo de bits, el dispositivo de procesamiento de vídeo puede determinar, basándose en que se cumpla una condición, que el descodificador de vídeo de múltiples capas es capaz de descodificar el flujo de bits. En este ejemplo, la condición se satisface basándose en que existe un esquema de partición adecuado en el uno o más esquemas de partición. Para cada partición respectiva del esquema de partición adecuado, se puede asignar uno de los descodificadores de capa única del descodificador de vídeo de múltiples capas para descodificar la partición respectiva.

35 [0122] De acuerdo con otras técnicas de ejemplo de esta divulgación, las pruebas de conformidad de flujo de bits se especifican para probar la conformidad de cada partición, de forma que se supone que las imágenes descodificadas y otra información necesaria para todas las capas de referencia directas o indirectas de las capas en la partición están disponibles de manera similar a la especificación actual en las especificaciones recientes de SHVC/MV-HEVC para la capa base cuando se proporciona externamente.

60  
65

**[0123]** En algunos ejemplos, se definen las pruebas de conformidad para una partición de un flujo de bits, y la conformidad del flujo de bits se define basándose en la conformidad de todas las particiones de cada esquema de partición especificado para el flujo de bits. En algunos ejemplos, cuando se selecciona una partición para la prueba, la descodificación de las otras particiones en el esquema de partición respectivo se considera descodificada por un HRD de acuerdo con los parámetros especificados para esas particiones.

**[0124]** La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de implementación de un descodificador de vídeo 30, que comprende una pluralidad de descodificadores de capa única, de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 2, el descodificador de vídeo 30 comprende los descodificadores de capa única 50A-50D (colectivamente, "descodificadores de capa única 50"). Cada uno de los descodificadores de capa única 50 puede implementarse mediante un núcleo de hardware independiente respectivo. Distintos de los descodificadores de capa única 50 pueden tener capacidades diferentes. Por ejemplo, diferentes descodificadores de capa única 50 pueden descodificar flujos de bits que se ajustan a diferentes perfiles, grados y niveles. En algunos ejemplos, el descodificador de vídeo 30 puede determinar, basándose en la información de PTL para las particiones de los esquemas de partición, un esquema de partición particular a partir de uno o más esquemas de partición. El descodificador de vídeo 30 puede descodificar cada partición respectiva del esquema de partición particular usando un descodificador de capa única 50 respectivo.

**[0125]** La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas de esta divulgación. La FIG. 3 se proporciona con fines de explicación y no debería considerarse limitadora de las técnicas tales como las ampliamente ejemplificadas y descritas en esta divulgación. Con fines de explicación, esta divulgación describe el codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros estándares o procedimientos de codificación.

**[0126]** En el ejemplo de la FIG. 3, el codificador de vídeo 20 incluye una unidad de procesamiento de predicción 200, una unidad de generación residual 202, una unidad de procesamiento de transformación 204, una unidad de cuantificación 206, una unidad de cuantificación inversa 208, una unidad de procesamiento de transformación inversa 210, una unidad de reconstrucción 212, una unidad de filtro 214, una memoria intermedia de imágenes descodificadas 216, y una unidad de codificación por entropía 218. La unidad de procesamiento de predicción 200 incluye una unidad de procesamiento de inter-predicción 220 y una unidad de procesamiento de intra-predicción 226. La unidad de procesamiento de inter-predicción 220 incluye una unidad de estimación de movimiento 222 y una unidad de compensación de movimiento 224. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales.

**[0127]** El codificador de vídeo 20 puede recibir datos de vídeo. La memoria de datos de vídeo 201 puede almacenar datos de vídeo a ser codificados por los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 201 se pueden obtener, por ejemplo, del origen de vídeo 18. La memoria intermedia de imágenes descodificadas 216 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo mediante el codificador de vídeo 20, por ejemplo, en modos de intra-codificación o inter-codificación. La memoria de datos de vídeo 201 y la memoria intermedia de imágenes descodificadas 216 pueden estar formadas por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 201 y la memoria intermedia de imágenes descodificadas 216 pueden proporcionarse mediante el mismo dispositivo de memoria o dispositivos de memoria independientes. En varios ejemplos, la memoria de datos de vídeo 201 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera de chip con respecto a esos componentes.

**[0128]** El codificador de vídeo 20 puede codificar cada CTU en un fragmento de una imagen de los datos de vídeo. Cada una de las CTU puede tener bloques de árbol de codificación (CTB) de luma de igual tamaño y los CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 200 puede realizar la partición de árbol cuádruple para dividir los CTB de la CTU en bloques progresivamente más pequeños. Los bloques más pequeños pueden ser bloques de codificación de CU. Por ejemplo, la unidad de procesamiento de predicción 200 puede dividir un CTB de una CTU en cuatro sub-bloques de igual tamaño, dividir uno o más de los sub-bloques en cuatro sub-sub-bloques de igual tamaño, etc.

**[0129]** El codificador de vídeo 20 puede codificar las CU de una CTU para generar representaciones codificadas de las CU (es decir, CU codificadas). Como parte de la codificación de una CU, la unidad de procesamiento de predicción 200 puede dividir los bloques de codificación de la CU entre una o más PU de la CU. Por lo tanto, cada PU puede tener un bloque de predicción de luma y los correspondientes bloques de predicción de croma. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden dar soporte a las PU de varios tamaños. Como se ha indicado anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU y el tamaño de una PU puede referirse al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular sea  $2N \times 2N$ , el codificador de vídeo 20 y el descodificador de vídeo 30 pueden dar soporte a tamaños de PU de  $2N \times 2N$  o  $N \times N$  para la intra-predicción, y a tamaños de PU simétricas de  $2N \times 2N$ ,  $2N \times N$ ,

$N \times 2N$ ,  $N \times N$  o similares para la inter-predicción. El codificador de vídeo 20 y el decodificador de vídeo 30 también pueden dar soporte a la partición asimétrica para tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$ , para la inter-predicción.

5 **[0130]** La unidad de procesamiento de inter-predicción 220 puede generar datos predictivos para una PU realizando una inter-predicción en cada PU de una CU. Los datos predictivos para la PU pueden incluir bloques predictivos de la PU e información de movimiento para la PU. La unidad de procesamiento de inter-predicción 220 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, un fragmento P o un fragmento B. En un fragmento en I, todas las PU son intra-predichas. Por lo tanto, si la PU está en  
10 un fragmento I, la unidad de procesamiento de inter-predicción 220 no realiza inter-predicción en la PU.

**[0131]** Si una PU está en un fragmento P, la unidad de estimación de movimiento 222 puede buscar las imágenes de referencia en una lista de imágenes de referencia (por ejemplo, "RefPicList0") para una región de referencia para la PU. La región de referencia para la PU puede ser una región, dentro de una imagen de referencia, que contiene bloques de muestra que se corresponden más estrictamente con los bloques de muestra de la PU. La unidad de estimación de movimiento 222 puede generar un índice de referencia que indique una posición en RefPicList0 de la imagen de referencia que contiene la región de referencia para la PU. Además, la unidad de estimación de movimiento 222 puede generar un vector de movimiento que indica un desplazamiento espacial entre un bloque de codificación de la PU y una ubicación de referencia asociada con la región de referencia. Por ejemplo, el vector de movimiento puede ser un vector bidimensional que proporciona una desviación respecto a las coordenadas en la imagen actual a las coordenadas en una imagen de referencia. La unidad de estimación de movimiento 222 puede emitir el índice de referencia y el vector de movimiento como la información de movimiento de la PU. La unidad de compensación de movimiento 224 puede generar los bloques predictivos de la PU basándose en muestras reales o interpoladas en la ubicación de referencia indicada por el vector de movimiento de la PU.  
15  
20  
25

**[0132]** Si una PU está en un fragmento B, la unidad de estimación de movimiento 222 puede llevar a cabo predicción unidireccional o predicción bidireccional para la PU. Para realizar predicción unidireccional para la PU, la unidad de estimación de movimiento 222 puede buscar las imágenes de referencia de RefPicList0 o una segunda lista de imágenes de referencia ("RefPicList1") para una región de referencia para la PU. La unidad de estimación de movimiento 222 puede emitir, como la información de movimiento de la PU, un índice de referencia que indica una posición en RefPicList0 o RefPicList1 de la imagen de referencia que contiene la región de referencia, un vector de movimiento que indica un desplazamiento espacial entre un bloque de predicción de la PU y una ubicación de referencia asociada con la región de referencia, y uno o más indicadores de dirección de predicción que indican si la imagen de referencia está en RefPicList0 or RefPicList1. La unidad de compensación de movimiento 224 puede generar los bloques predictivos de la PU basándose al menos en parte en muestras reales o interpoladas en la ubicación de referencia indicada por el vector de movimiento de la PU.  
30  
35

**[0133]** Para realizar inter-predicción bidireccional para una PU, la unidad de estimación de movimiento 222 puede buscar las imágenes de referencia en RefPicList0 para una región de referencia para la PU y también puede buscar las imágenes de referencia en RefPicList1 para otra región de referencia para la PU. La unidad de estimación de movimiento 222 puede generar índices de referencia que indiquen posiciones en RefPicList0 y RefPicList1 de las imágenes de referencia que contienen las regiones de referencia. Además, la unidad de estimación de movimiento 222 puede generar vectores de movimiento que indican desplazamientos espaciales entre las ubicaciones de referencia asociadas con las regiones de referencia y un bloque de predicción de la PU. La información de movimiento de la PU puede incluir los índices de referencia y los vectores de movimiento de la PU. La unidad de compensación de movimiento 224 puede generar los bloques predictivos de la PU basándose al menos en parte en muestras reales o interpoladas en las ubicaciones de referencia indicadas por los vectores de movimiento de la PU.  
40  
45

**[0134]** La unidad de procesamiento de intra-predicción 226 puede generar datos predictivos para una PU realizando una intra-predicción en la PU. Los datos predictivos para la PU pueden incluir bloques predictivos para la PU y varios elementos sintácticos. La unidad de procesamiento de intra-predicción 226 puede realizar la intra-predicción en las PU en fragmentos I, fragmentos P y fragmentos B.  
50

**[0135]** Para realizar la intra-predicción en una PU, la unidad de procesamiento de intra-predicción 226 puede usar múltiples modos de intra-predicción para generar múltiples conjuntos de bloques predictivos para la PU. Cuando se realiza la intra-predicción usando un modo particular de intra-predicción, la unidad de procesamiento de intra-predicción 226 puede generar bloques predictivos para la PU usando un conjunto particular de muestras de bloques contiguos. Los bloques contiguos pueden estar arriba, arriba y a la derecha, arriba y a la izquierda, o a la izquierda de los bloques de predicción de la PU, suponiendo un orden de codificación de izquierda a derecha, de arriba a abajo, para las PU, CU y CTU. La unidad de procesamiento de intra-predicción 226 puede usar varios números de modos de intra-predicción, por ejemplo, 33 modos de intra-predicción direccional. En algunos ejemplos, el número de modos de intra-predicción puede depender del tamaño de los bloques de predicción de la PU.  
55  
60

**[0136]** La unidad de procesamiento de predicción 200 puede seleccionar los datos predictivos para las PU de una CU entre los datos predictivos generados por la unidad de procesamiento de inter-predicción 220 para las PU o los datos predictivos generados por la unidad de procesamiento de intra-predicción 226 para las PU. En algunos  
65

ejemplos, la unidad de procesamiento de predicción 200 selecciona los datos predictivos para las PU de la CU basándose en las métricas de velocidad/distorsión de los conjuntos de datos predictivos. Los bloques predictivos de los datos predictivos seleccionados pueden denominarse en el presente documento bloques predictivos seleccionados.

5 **[0137]** La unidad de generación residual 202 puede generar, basándose en los bloques de codificación de una CU y los bloques predictivos seleccionados de las PU de la CU, boques residuales de la CU. Por ejemplo, la unidad de generación residual 202 puede generar, basándose en los bloques de codificación de luma, Cb y Cr de una CU y los bloques de luma, Cb y Cr predictivos seleccionados de las PU de la CU, bloques residuales de luma, Cb y Cr de la CU. En algunos ejemplos, la unidad de generación residual 202 puede generar los bloques residuales de la CU de manera que cada muestra en los bloques residuales tenga un valor igual a una diferencia entre una muestra en un bloque de codificación de la CU y una muestra correspondiente en un correspondiente bloque predictivo seleccionado de una PU de la CU.

15 **[0138]** La unidad de procesamiento de transformación 204 puede realizar partición de árbol cuádruple para dividir los bloques residuales de una CU en bloques de transformación de las TU de la CU. De este modo, una TU puede tener un bloque de transformación de luma y dos bloques de transformación de croma correspondientes. Los tamaños y las posiciones de los bloques de transformación de las TU de una CU pueden o no basarse en los tamaños y posiciones de los bloques de predicción de las PU de la CU.

20 **[0139]** La unidad de procesamiento de transformación 204 puede generar bloques de coeficientes de transformación para cada TU de una CU aplicando una o más transformaciones a los bloques de transformación de la TU. La unidad de procesamiento de transformación 204 puede aplicar varias transformaciones a un bloque de transformación de una TU. Por ejemplo, la unidad de procesamiento de transformación 204 puede aplicar una transformación de coseno discreta (DCT), una transformación direccional o una transformación conceptualmente similar al bloque de transformación. En algunos ejemplos, la unidad de procesamiento de transformación 204 no aplica transformaciones a un bloque de transformación. En dichos ejemplos, el bloque de transformación puede tratarse como un bloque de coeficientes de transformación.

25 **[0140]** La unidad de cuantificación 206 puede cuantificar los coeficientes de transformación en un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits de algunos o a la totalidad de los coeficientes de transformación. Por ejemplo, un coeficiente de transformación de  $n$  bits puede redondearse hacia abajo hasta un coeficiente de transformación de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ . La unidad de cuantificación 206 puede cuantificar un bloque de coeficientes de una TU de una CU basándose en un valor de parámetro de cuantificación (QP) para la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de una CU, ajustando el valor del QP de la CU. La cuantificación puede introducir pérdida de información, por lo que los coeficientes de transformación cuantificados pueden tener una precisión inferior a los originales.

30 **[0141]** La unidad de cuantificación inversa 208 y la unidad de procesamiento de transformación inversa 210 pueden aplicar la cuantificación inversa y transformaciones inversas al bloque de coeficientes, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficientes. La unidad de reconstrucción 212 puede añadir el bloque residual reconstruido a las muestras correspondientes de uno o más bloques predictivos generados por la unidad de procesamiento de predicción 200 para producir un bloque de transformación reconstruido de una TU. Mediante la reconstrucción de bloques de transformación para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la CU.

35 **[0142]** La unidad de filtro 214 puede realizar una o más operaciones de desbloqueo para reducir las distorsiones de bloqueo en los bloques de codificación de una CU. La memoria intermedia de imágenes descodificadas 216 puede almacenar los bloques de codificación reconstruidos después de que la unidad de filtro 214 realice las una o más operaciones de desbloqueo en los bloques de codificación reconstruidos. La unidad de procesamiento de inter-predicción 220 puede utilizar una imagen de referencia que contiene los bloques de codificación reconstruidos para realizar la inter-predicción en las PU de otras imágenes. Además, la unidad de procesamiento de intra-predicción 226 puede usar bloques de codificación reconstruidos en la memoria intermedia de imágenes descodificadas 216 para realizar intra-predicción en otras PU en la misma imagen que la CU.

40 **[0143]** La unidad de codificación por entropía 218 puede recibir datos desde otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación por entropía 218 puede recibir bloques de coeficientes desde la unidad de cuantificación 206 y puede recibir elementos sintácticos desde la unidad de procesamiento de predicción 200. La unidad de codificación por entropía 218 puede realizar una o más operaciones de codificación por entropía en los datos para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 218 puede realizar una operación de codificación de longitud variable adaptable al contexto (CAVLC), una operación de CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una operación de codificación por entropía de partición de rango de probabilidad (PIPE), una operación de codificación de Golomb exponencial u otro tipo de operación de codificación por entropía en los datos. El codificador de vídeo 20 puede



emitir un flujo de bits que incluye datos codificados por entropía, generados por la unidad de codificación por entropía 218.

**[0144]** De acuerdo con algunos ejemplos de esta divulgación, la unidad de codificación de entropía 218 puede incluir, en el flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición. Para cada esquema de partición respectivo del uno o más esquemas de partición, el esquema de partición respectivo especifica un conjunto respectivo de particiones disjuntas cuya unión forma un conjunto de capas de salida, en el que cada partición respectiva del conjunto respectivo de particiones disjuntas contiene una o más capas, el conjunto de capas de salida puede consistir en todas las capas de la pluralidad de capas o un subconjunto apropiado de la pluralidad de capas.

**[0145]** La FIG. 4 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que puede implementar las técnicas de esta divulgación. La FIG. 4 se proporciona con fines de explicación y no se limita a las técnicas como las ampliamente ejemplificadas y descritas en esta divulgación. Con fines de explicación, esta divulgación describe el descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros estándares o procedimientos de codificación.

**[0146]** En el ejemplo de la FIG. 4, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 250, una unidad de procesamiento de predicción 252, una unidad de cuantificación inversa 254, una unidad de procesamiento de transformación inversa 256, una unidad de reconstrucción 258, una unidad de filtro 260 y una memoria intermedia de imágenes descodificadas 262. La unidad de procesamiento de predicción 252 incluye una unidad de compensación de movimiento 264 y la unidad de procesamiento de intra-predicción 266. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

**[0147]** Una memoria intermedia de imágenes codificadas (CPB) 251 puede recibir y almacenar datos de vídeo codificados (por ejemplo, unidades NAL) de un flujo de bits. Los datos de vídeo almacenados en la CPB 151 pueden obtenerse, por ejemplo, del medio legible por ordenador 16, por ejemplo, desde una fuente de vídeo local, tal como una cámara, mediante comunicación de datos de vídeo por red alámbrica o inalámbrica, o accediendo a medios de almacenamiento de datos físicos. La CPB 151 puede formar una memoria de datos de vídeo que almacena datos de vídeo codificados a partir de un flujo de bits de vídeo codificado. La memoria intermedia de imágenes descodificadas 162 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para usar en la descodificación de datos de vídeo mediante el descodificador de vídeo 30, por ejemplo, en modos de intra-codificación o de inter-codificación. La CPB 151 y la memoria intermedia de imágenes descodificadas 162 pueden estar formadas por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La CPB 151 y la memoria intermedia de imágenes descodificadas 162 pueden proporcionarse mediante el mismo dispositivo de memoria o dispositivos de memoria independientes. En diversos ejemplos, la CPB 151 puede estar en un chip con otros componentes del descodificador de vídeo 30, o fuera de chip con respecto a esos componentes.

**[0148]** La unidad de descodificación por entropía 250 puede recibir unidades NAL de la CPB 251 y analizar las unidades NAL para obtener elementos sintácticos a partir del flujo de bits. La unidad de descodificación por entropía 250 puede descodificar por entropía elementos sintácticos codificados por entropía en las unidades NAL. La unidad de procesamiento de predicción 252, la unidad de cuantificación inversa 254, la unidad de procesamiento de transformación inversa 256, la unidad de reconstrucción 258 y la unidad de filtro 260 pueden generar datos de vídeo descodificados basándose en los elementos sintácticos obtenidos a partir del flujo de bits.

**[0149]** Por ejemplo, la unidad de descodificación de entropía 250 puede obtener, a partir del flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición. La unidad de procesamiento de predicción 252, la unidad de cuantificación inversa 254, la unidad de procesamiento de transformación inversa 256, la unidad de reconstrucción 258 y la unidad de filtro 260 pueden descodificar las particiones de un esquema de partición particular. En algunos ejemplos, para cada esquema de partición respectivo del uno o más esquemas de partición, la unidad de descodificación de entropía 250 obtiene, a partir de un mensaje SEI en el flujo de bits o VUI en un VPS en el flujo de bits, un conjunto respectivo de conjuntos de parámetros de HRD para el respectivo esquema de partición. El descodificador de vídeo 30 puede probar, basándose en los parámetros de HRD para el uno o más esquemas de partición, cada partición de cada uno de uno o más esquemas de partición en cuanto a conformidad con un estándar de codificación de vídeo.

**[0150]** Las unidades NAL del flujo de bits pueden incluir unidades NAL de fragmentos codificados. Como parte de la descodificación del flujo de bits, la unidad de descodificación por entropía 250 puede descodificar por entropía elementos sintácticos de las unidades NAL de fragmentos codificados. Cada uno de los fragmentos codificados puede incluir una cabecera de fragmento y datos de fragmento. La cabecera de fragmento puede contener elementos sintácticos pertenecientes a un fragmento.

**[0151]** Además de obtener elementos sintácticos del flujo de bits, el descodificador de vídeo 30 puede realizar una operación de descodificación en una CU. Realizando la operación de descodificación en una CU, el descodificador de vídeo 30 puede reconstruir bloques de codificación de la CU.

5 **[0152]** Como parte de la realización de una operación de descodificación en una CU, la unidad de cuantificación inversa 254 puede cuantificar de forma inversa, es decir, descuantificar, los bloques de coeficientes de las TU de la CU. La unidad de cuantificación inversa 254 puede utilizar un valor de QP para la CU de la TU para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa, para aplicar por parte de la unidad de cuantificación inversa 254. Es decir, la razón de compresión, es decir, la razón entre el número de bits utilizados para representar la secuencia original y la comprimida, puede controlarse ajustando el valor del QP utilizado al cuantificar los coeficientes de transformación. La razón de compresión también puede depender del procedimiento de codificación por entropía empleado.

15 **[0153]** Después de que la unidad de cuantificación inversa 254 cuantifica inversamente un bloque de coeficientes, la unidad de procesamiento de transformación inversa 256 puede aplicar una o más transformaciones inversas al bloque de coeficientes para generar un bloque de muestras residual para la TU. Por ejemplo, la unidad de procesamiento de transformación inversa 256 puede aplicar una DCT inversa, una transformación entera inversa, una transformación de Karhunen-Loeve (KLT) inversa, una transformación de rotación inversa, una transformación direccional inversa u otra transformación inversa al bloque de coeficientes.

20 **[0154]** Si se codifica una PU usando la intra-predicción, la unidad de procesamiento de intra-predicción 266 puede realizar la intra-predicción para generar bloques predictivos para la PU. Por ejemplo, la unidad de procesamiento de intra-predicción 266 puede realizar una intra-predicción para generar bloques predictivos de luma, Cb y Cr para la PU. La unidad de procesamiento de intra-predicción 266 puede usar un modo de intra-predicción para generar los bloques predictivos para la PU basándose en los bloques de predicción de PU espacialmente contiguas. La unidad de procesamiento de intra-predicción 266 puede determinar el modo de intra-predicción para la PU basándose en uno o más elementos sintácticos obtenidos a partir del flujo de bits.

30 **[0155]** La unidad de procesamiento de predicción 252 puede construir una primera lista de imágenes de referencia (ListalmagenRef0) y una segunda lista de imágenes de referencia (ListalmagenRef1) basándose en los elementos sintácticos extraídos del flujo de bits. Además, si se codifica una PU usando la inter-predicción, la unidad de descodificación por entropía 250 puede obtener información de movimiento para la PU. La unidad de compensación de movimiento 264 puede determinar, basándose en la información de movimiento de la PU, una o más regiones de referencia para la PU. La unidad de compensación de movimiento 264 puede generar, basándose en muestras en los uno o más bloques de referencia para la PU, bloques predictivos para la PU. Por ejemplo, la unidad de compensación de movimiento 264 puede generar, basándose en muestras de los uno o más bloques de referencia para la PU, bloques predictivos de luma, Cb y Cr para la PU.

40 **[0156]** La unidad de reconstrucción 258 puede usar los valores residuales de los bloques de transformación de las TU de una CU y los bloques predictivos de las PU de la CU para reconstruir los bloques de codificación de la CU. Por ejemplo, la unidad de reconstrucción 258 puede utilizar valores residuales de los bloques de transformación de luma, Cb y Cr de las TU de una CU y los bloques predictivos de luma, Cb y Cr de las PU de la CU para reconstruir los bloques de codificación de luma, Cb y Cr de la CU. Por ejemplo, la unidad de reconstrucción 258 puede agregar muestras de los bloques de transformación a las muestras correspondientes de los bloques de predictivos para reconstruir los bloques de codificación de la CU.

50 **[0157]** La unidad de filtro 260 puede realizar una operación de desbloqueo para reducir las distorsiones de bloqueo asociadas con los bloques de codificación de la CU. El descodificador de vídeo 30 puede almacenar los bloques de codificación de la CU en la memoria intermedia de imágenes descodificadas 262. La memoria intermedia de imágenes descodificadas 262 puede proporcionar imágenes de referencia para la posterior compensación de movimiento, la intra-predicción y la presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el descodificador de vídeo 30 puede realizar, basándose en los bloques de la memoria intermedia de imágenes descodificadas 262, operaciones de intra-predicción o de inter-predicción en las PU de otras CU. De esta forma, el descodificador de vídeo 30 puede obtener, a partir del flujo de bits, los niveles de coeficientes de transformación de un bloque de coeficientes, cuantificar inversamente los niveles de coeficientes de transformación, aplicar una transformación a los niveles de coeficientes de transformación para generar un bloque de transformación, generar, basándose al menos en parte en el bloque de transformación, un bloque de codificación, y emitir el bloque de codificación para visualizar.

60 **[0158]** Las siguientes secciones de esta divulgación describen algunos detalles de implementación de algunas de las técnicas de ejemplo de esta divulgación descritas anteriormente. El siguiente texto muestra los cambios para implementar una o más técnicas de esta divulgación. El texto subrayado indica texto agregado. El texto en cursiva incluido entre corchetes es texto eliminado. Los nombres de los elementos sintácticos se muestran en negrita.

65 **[0159]** Como se indicó anteriormente, de acuerdo con algunas técnicas de esta divulgación, el codificador de vídeo 20 puede señalar uno o más esquemas de partición. Cada uno de los esquemas de partición divide las capas en un

conjunto de capas de salida en una o más particiones. Cada capa en un conjunto de capas de salida debe pertenecer exactamente a una partición en un esquema de partición.

5 **[0160]** La Tabla 2, a continuación, indica los cambios en la sintaxis de extensión de VPS para señalar los esquemas de partición definidos en SHVC WD6 y MV-HEVC WD8.

**TABLA 2**

vps_extension( ) {	Descriptor
<b>avc_base_layer_flag</b>	u(1)
...	u(1)
for(i = 1; i < NumOutputLayerSets; i++) {	
if(i > vps_num_layer_sets_minus1 )	
<b>output_layer_set_idx_minus1[i]</b>	u(v)
if(i > vps_num_layer_sets_minus1    defaultTargetOutputLayerIdc == 2)	
for(j = 0; j < NumLayersInIdList[LayerSetIdxForOutputLayerSet[i]]; j++)	
<b>output_layer_flag[i][j]</b>	u(1)
<b>profile_level_tier_idx[i]</b>	u(v)
<b>num_partition_schemes[i]</b>	ue(v)
for(j = 0; j < num_partition_schemes[i]; j++) {	
<b>num_partitions_in_scheme_minus1[i][j]</b>	ue(v)
for(k = 0; k < num_partitions_in_scheme_minus1[i][j]; k++) {	
for(r = 0; r <= vps_max_layers_minus1; r++)	
if(layer_id_included_flag[LayerSetIdxForOutputLayerSet[i]][r])	
<b>layer_included_in_partition_flag[i][j][k][r]</b>	u(1)
<b>profile_level_tier_partition_idx[i][j][k]</b>	u(v)
}	
}	
if(NumOutputLayersInOutputLayerSet[i] == 1 && NumDirectRefLayers[OlsHighestOutputLayerId[i]] > 0)	
<b>alt_output_layer_flag[i]</b>	u(1)
}	
...	
}	

10 **[0161]** En el ejemplo de la Tabla 2, num\_partition\_schemes[i] especifica el número de esquemas de partición  
 especificados para el i-ésimo conjunto de capas de salida. El valor de num\_partition\_schemes[i] estará dentro del  
 rango de 0 y 15, inclusive. Además, num\_partitions\_in\_scheme\_minus1[i][j] más 1 especifica el número de  
 particiones en el j-ésimo esquema de partición del i-ésimo conjunto de capas de salida. El valor de  
 15 num\_partitions\_in\_scheme\_minus1[i][j] estará dentro del rango de 0 a  
 NumLayersInIdList[LayerSetIdxForOutputLayerSet[i]] - 1, inclusive.

**[0162]** Además, en el ejemplo de la Tabla 2, layer\_included\_in\_partition\_flag[i][j][k][r] igual a 1 especifica que la r-  
 ésima capa especificada en el VPS se incluye en la k-ésima partición del j-ésimo esquema de partición del i-ésimo  
 conjunto de capas de salida. layer\_included\_in\_partition\_flag[i][j][k][r] igual a 0 especifica que la r-ésima capa  
 20 especificada en el VPS no está incluida en la k-ésima partición del j-ésimo esquema de partición del i-ésimo conjunto  
 de capas de salida.

**[0163]** De este modo, para cada esquema de partición respectivo, para cada partición respectiva del esquema de  
 partición respectivo, para cada capa respectiva del conjunto de capas de salida, el decodificador de vídeo 30 puede  
 25 obtener, a partir del flujo de bits, un elemento sintáctico (es decir, layer\_included\_in\_partition\_flag) que indica si la

capa respectiva está incluida en la partición respectiva. En un ejemplo similar, para cada esquema de partición respectivo del uno o más esquemas de partición, para cada partición respectiva del esquema de partición respectivo, para cada capa respectiva del conjunto de capas de salida, el codificador de vídeo 20 puede incluir, en un flujo de bits, un elemento sintáctico (es decir, `layer_included_in_partition_flag`) que indica si la capa respectiva está incluida en la partición respectiva.

**[0164]** En el ejemplo de la Tabla 2, es un requisito de conformidad del flujo de bits que para cada capa con `nuh_layer_id` igual a `currLayerId` en el *i*-ésimo conjunto de capas de salida, exista un único valor de *k* en el rango de 0 a `num_partitions_in_scheme_minus1 [i] [j]`, inclusive, para cada esquema de partición con índice *j* del *i*-ésimo conjunto de capas de salida, tal que `layer_included_in_partition_flag[i][j][k][LayerIdxInVps[currLayerId]]` sea igual a 1.

**[0165]** Como se ha mencionado anteriormente, de acuerdo con algunos ejemplos de esta divulgación, se proporciona señalización de un perfil, un grado y un nivel para cada partición. Como se muestra en el ejemplo de la Tabla 2, `profile_level_tier_partition_idx[i][j][k]` especifica el índice, en la lista de `profile_level_tier_level( )` estructuras sintácticas en el VPS, de la `profile_tier_level( )` estructura sintáctica que se aplica a la *k*-ésima partición del *j*-ésimo esquema de partición del *i*-ésimo conjunto de capas de salida. La longitud del `profile_level_tier_partition_idx[i]` elemento sintáctico es de `Ceil(Log2(vps_num_profile_tier_level_minus1 + 1))` bits. El valor de `profile_level_tier_idx[0][0][0]` se establece para que sea igual a 0. El valor de `profile_level_tier_partition_idx[i]` estará dentro del rango de 0 a `vps_num_profile_tier_level_minus1`, inclusive.

**[0166]** En otro ejemplo, los cambios mostrados en la Tabla 3 pueden realizarse en la sintaxis de extensión de VPS para señalar esquemas de partición. Los cambios que se muestran en la Tabla 3 son relativos a la Tabla 2 anterior.

**TABLA 3**

vps_extension( ) {	Descriptor
...	
<b>profile_level_tier_idx[i]</b>	u(v)
<b>num_partitioning_schemes_minus1[i]</b>	ue(v)
for(j = [[0]]1; j <= num_partitioning_schemes_minus1[i]; j++) {	
<b>num_partitions_in_scheme_minus[[1]]2[i][j]</b>	ue(v)
for(k = 0[[01]]; k <= N[[n]]um[[_p]]Partitions[[_i]]n[[_s]]Scheme[[_minus1]]i[j]; k++) {	
for(r = 0; r <= vps_max_layers_minus1; r++)	
if(layer_id_included_flag[LayerSetIdxForOutputLayerSet[i][r])	
<b>layer_included_in_partition_flag[i][j][k][r]</b>	u(1)
<b>profile_level_tier_partition_idx[i][j][k]</b>	u(v)
}	
}	
...	

**[0167]** En el ejemplo de la Tabla 3, se pueden realizar los siguientes cambios en la semántica de elementos sintácticos particulares de la extensión VPS descrita anteriormente con referencia a la Tabla 2.

**num\_partition ing\_schemes\_minus1[i]** plus 1 especifica el número de esquemas de partición especificados para el *i*-ésimo conjunto de capas de salida. El valor de `num_partitioning_schemes_minus1 [ i ]` estará en el rango de 0 a 15, inclusive.

**num\_partitions\_in\_scheme\_minus[[1]]2[ i ] [ j ]** más `[[1]]2` especifica el número de particiones en el *j*-ésimo esquema de partición del *i*-ésimo conjunto de capas de salida. El valor de `num_partitions_in_schcmc_minus[[1]]2[ i ] [ j ]` estará en el rango de 0 a `NumLayersInIdList[ LayerSetIdxForOutputLayerSet[ i ] ] - 1`, inclusive.

El valor de la variable `NumPartitionInScheme[ i ] [ i ]` se obtiene de la forma siguiente:

`NumPartitionsInScheme[ i ] [ i ] = ( j = 0 ) ? 1 : num_partitions_in_scheme_minus2[ i ] [ i ] + 2.`

[[Es un requisito de conformidad del flujo de bits que cuando *j* es mayor que 0, el valor de `num_partitions_in_scheme_minus1 [ i ] [ j ]` será mayor que 0.]]

**layer\_included\_in\_partition\_flag**[ i ][ j ][ k ][ r ] igual a 1 especifica que la r-ésima capa especificada en el VPS se incluye en la k-ésima partición del j-ésimo esquema de partición del i-ésimo conjunto de capas de salida.

**layer\_included\_in\_partition\_flag**[ i ][ j ][ k ][ r ] igual a 0 especifica que la r-ésima capa especificada en el VPS no se incluye en la k-ésima partición del j-ésimo esquema de partición del i-ésimo conjunto de capas de salida. Se infiere que el valor de **layer\_included\_in\_partition\_flag**[ i ][ 0 ][ 0 ][ r ] es igual al valor de **layer\_id\_included\_flag**[ **layerSetIdxForOutputLayerSet**[ i ][ j ][ r ] para r en el rango de 0 a **vps\_max\_layers\_minus1**, inclusive.

**NOTA 5** - [[Es un requisito de conformidad del flujo de bits que f]]Para cada capa con **nuh\_layer\_id** igual a **currLayerId** en el i-ésimo conjunto de capas de salida, exista un único valor de k en el rango de 0 a [[n]]**Num**[[\_p]]**Partitions**[[\_i]]**n**[[\_s]]**Scheme**[[\_minus1]][ i ][ j ] [[ ,inclusive]], para cada esquema de partición con índice j del i-ésimo conjunto de capas de salida, tal que **layer\_included\_in\_partition\_flag**[ i ][ j ][ k ][ **LayerIdxInVps**[ **currLayerId** ] sea igual a 1.

**profile\_level\_tier\_partition\_idx**[ i ][ j ][ k ] especifica el índice, en la lista de estructuras sintácticas de nivel de grado de perfil( ) en el VPS, de la estructura sintáctica **profile\_tier\_level**( ) que se aplica a la k-ésima partición del j-ésimo esquema de partición del i-ésimo conjunto de capas de salida. La longitud del elemento sintáctico **profile\_level\_tier\_partition\_idx**[ i ][ j ][ k ] es de **Ceil**( **Log2**( **vps\_num\_profile\_tier\_level\_minus1** + 1 ) ) bits. El valor de **profile\_level\_tier\_partition\_idx**[ 0 ][ 0 ][ 0 ] se infiere que es igual a 0. El valor de **profile\_level\_tier\_partition\_idx**[ i ][ 0 ][ 0 ] se establece para que sea igual a **profile\_level\_tier\_idx**[ i ] para i en el rango de 1 a **NumOutputLayerSets** - 1, inclusive. El valor de **profile\_level\_tier\_partition\_idx**[ i ][ j ][ k ] estará en el rango de 0 a **vps\_num\_profile\_tier\_level\_minus1**, inclusive.

**[0168]** En algunos ejemplos, se agrega una restricción tal que no más de un esquema de partición para cada conjunto de capas de salida contiene solo una partición. Además, en algunos ejemplos, el elemento sintáctico **profile\_tier\_level\_idx**[i] no se señala y el elemento sintáctico **profile\_tier\_tier\_partition\_idx**[i][j][k] se utiliza para señalar las indicaciones de perfil, grado y nivel del esquema de partición del conjunto de capas de salida que contiene una partición que incluye todas las capas del conjunto de capas de salida.

**[0169]** Además, de acuerdo con algunas técnicas de esta divulgación, los límites y las restricciones de nivel que definen perfiles, grados y niveles son específicos de la partición. El siguiente texto indica otros cambios a MV-HEVC WD8 para implementar límites y restricciones de nivel específicos de partición para perfiles, grados y niveles.

## G.11 Perfiles, grados y niveles

### G.11.1 Perfiles

#### G.11.1.2 Perfil estéreo principal

**[0170]** Los flujos de bits que contienen conjuntos de capas de salida con particiones de capas conforme al perfil estéreo principal deberán cumplir las siguientes restricciones en el sub-flujo de bits obtenido para cada uno de los conjuntos de capas de salida, siendo **layerSetIdx** el índice del conjunto de capas para el conjunto de capas de salida, obteniéndose el sub-flujo de bits invocando el proceso de extracción sub-flujo de bits como se especifica en la subcláusula F.10 con el flujo de bits, **tldTarget** igual a 7, y **TargetDeclayerIdList** que contiene los valores **nuh\_layer\_id** del conjunto de capas con el **index\_layerSetIdx** como entradas, obteniéndose las particiones al invocar el proceso de desmultiplexación para obtener una partición de flujo de bits como se especifica en la subcláusula C.6 con el sub-flujo de bits, con la lista de capas en la partición **listOfLayersInPartition** y el número de capas en la partición **numLayersInPartition** como entradas.

**[0171]** Los flujos de bits correspondientes a los sub-flujos de bits para los conjuntos de capas de salida con particiones de capas que se ajusten al perfil estéreo principal deberán cumplir las siguientes restricciones:

- El sub-flujo de bits resultante del proceso de extracción de sub-flujo de bits con cualquier valor de **tldTarget** y un valor de 0 en **layerIdListTarget** como entradas se ajustará al perfil principal.
- El flujo de bits debe contener una capa con **nuh\_layer\_id** igual a i para la cual **ViewScalExtLayerFlag**[ i ] es igual a 1.
- Cuando **ViewScalExtLayerFlag**[ i ] es igual a 1, **inter\_view\_mv\_vert\_constraint\_flag** será igual a 1 en la estructura sintáctica **sps\_multilayer\_extension**( ) del SPS RBSP activo para la capa con **nuh\_layer\_id** igual a i.
- Cuando **ViewScalExtLayerFlag**[ i ] es igual a 1, **ScalabilityId**[ **LayerIdxInVps**[ i ] ][ **smlDx** ] será igual a 0 para cualquier valor **smlDx** de 0 a 15, inclusive, que no sea igual a 1, para cualquier imagen codificada con **nuh\_layer\_id** igual a i.
- Cuando **ViewScalExtLayerFlag**[ i ] es igual a 1, **num\_scaled\_ref\_layer\_offsets** será igual a 0 en cada SPS activo para la capa con **nuh\_layer\_id** igual a i.

- 5 - Cuando `ViewScalExtLayerFlag[ i ]` es igual a 1, los valores de `pic_width_in_luma_samples` y `pic_height_in_luma_samples` en el SPS activo para la capa con `nuh_layer_id` igual a `i` serán iguales a los valores de `pic_width_in_luma_samples` y `pic_height_in_luma_samples`, respectivamente, en los SPS activos para todas las capas de referencia directa de esa capa.
- 10 - El flujo de bits debe contener un sub-flujo de bits que consta de dos capas con `nuh_layer_id` igual a 0 y `nuhLayerIdA` para las cuales `ScalabilityId[ LayerIdxInVps[ nuhLayerIdA ] ][ smlIdx ]` debe ser igual a 0 para cualquier `smlIdx` de 0 a 15, inclusive, que no sea igual a 1.
- 15 - Los VPS tendrán `avc_base_layer_flag` igual a 0 solamente.
- Los VPS tendrán `vps_num_rep_formats_minus1` en la gama entre 0 y 15, inclusive.
- Los SPS deben tener `sps_extension_type_flag[ i ]` igual a 0 solo para `i` igual a 0, y en el rango de 2 a 6, inclusive.
- Los PPS tendrán `pps_extension_type_flag[ i ]` igual a 0 solo para `i` en el rango de 1 a 6, inclusive.

20 **[0172]** En el resto de esta subcláusula y la subcláusula G.11.3, para un sub-flujo de bits de un conjunto de capas de salida, `general_profile_idc`, `general_profile_compatibility_flag[ i ]`, `general_tier_flag`, y `general_level_idc` se refieren a `general_profile_idc`, `general_profile_compatibility_flag[ i ]`, `general_tier_flag`, y `general_level_idc`, respectivamente, en la estructura sintáctica `profile_tier_level( )` para el conjunto de capas de salida. La conformidad de un sub-flujo de bits de un conjunto de capas de salida en el perfil estéreo principal se indica mediante `general_profile_idc` siendo igual a 5 o `general_profile_compatibility_flag[ 5 ]` siendo igual a 1.

- 25 - NOTA - Cuando `general_profile_compatibility_flag[ 5 ]` es igual a 1, `general_profile_compatibility_flag[ i ]` para `i` igual a 1 también debe ser igual a 1.

30 **[0173]** Además, para implementar límites y restricciones de nivel específicos de partición para perfiles, grados y niveles, los siguientes cambios pueden hacerse en el Párrafo G.11.2 de JCTVC-Q0145, que a su vez modifica el Párrafo G.11.2 de Chen et al., "Extensiones escalables de codificación de vídeo de alta eficiencia (HEVC), borrador 5, Equipo colaborativo conjunto en codificación de vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 16.ª reunión: San José, EE. UU., del 9 al 17 de enero de 2014, documento JCTVC-P1008\_v4.

### 35 **G.11.2 Grados y niveles**

#### **G.11.2.1 Límites generales de grados y niveles**

40 **[0174]** Con el fin de comparar las capacidades de grado, el grado con `general_tier_flag` igual a 0 se considera un grado más bajo que el grado con `general_tier_flag` igual a 1.

45 **[0175]** Con el fin de comparar las capacidades de nivel, un nivel particular de un grado específico se considera un nivel más bajo que otro nivel del mismo grado cuando el valor del `general_level_idc` del nivel particular es menor que el del otro nivel.

**[0176]** Se especifica lo siguiente para expresar las restricciones en esta subcláusula, la subcláusula G.11.2.2, y la subcláusula G.11.3:

- 50 - Dejar que la unidad de acceso `n` sea la `n`-ésima unidad de acceso en el orden de decodificación, siendo la primera unidad de acceso la unidad de acceso 0 (es decir, la unidad de acceso 0-ésima).
- Dejar que la variable `fR` se establezca igual a  $1 \div 300$ .
- 55 - Dejar que la variable `optLsIdx` sea el índice del conjunto de capas de salida de un conjunto de capas de salida contenido en un flujo de bits, y el sub-flujo de bits del conjunto de capas de salida se obtenga invocando el proceso de extracción de sub-flujos de bits como se especifica en la subcláusula F.10 con el flujo de bits, `tIdTarget` igual a 7 y `TargetDecLayerIdList` conteniendo los valores `nuh_layer_id` del conjunto de capas de salida como entradas.
- 60 - Dejar que la variable `partitionIdx` sea el índice de cualquier partición particular del conjunto de capas de salida con el índice `optLsIdx`, y la partición se obtenga invocando el proceso de desmultiplexación como se especifica en la subcláusula C.6 con el sub-flujo de bits, la lista de capas en la partición `listOfLayersInPartition`, y el número de capas en la partición `numLayersInPartition` como entradas.
- 65 - Dejar que las variables `numLayersInOptLayerSet` y `numLayersInSubBitstream` se obtengan de la forma siguiente:

```

numLayersInOptLayerSet =
NumLayersInIdList[ LayerSetIdxForOutputLayerSet[ optLsIdx ] ]
if (la capa base en el conjunto de capas de salida es proporcionada por medios externos)
5     numLayersInSubBitstream = numLayersInOptLayerSet - 1 (H-63)
    else
        numLayersInSubBitstream = numLayersInOptLayerSet

```

- Dejar que la variable partitionSizeInSamplesY se obtenga de la forma siguiente:

```

10     partitionSizeInSamplesY = 0 (H-64)

```

```

    for( i = 0; i < numLayersInPartition; i-H-)
        partitionSizeInSamplesY += PicSizeInSamplesY de la i-ésima capa en la partición

```

- 15 - Dejar que la variable scaleFactor se establezca igual a  $\text{Ceil}(\text{partitionSizeInSamplesY} / \text{MaxLumaPs})$ , donde MaxLumaPs se especifica en la Tabla A-1 para el nivel de la partición.

- Dejar que la variable maxPartitionSliceSegs se establezca igual a  $\text{scaleFactor} * \text{MaxSliceSegmentsPerPicture}$ , donde MaxSliceSegmentsPerPicture se especifica en la Tabla A-1 para el grado y el nivel de la partición.

20 **[0177]** Cada partición que se ajuste a un perfil en un grado y nivel especificado deberá cumplir las siguientes restricciones para cada prueba de conformidad de flujo de bits como se especifica en el Anexo C:

25 a) Para cada capa en la partición, PicSizeInSamplesY será menor o igual que MaxLumaPs, donde MaxLumaPs se especifica en la Tabla A-1 para el grado y el nivel de la partición.

b) Para cada capa en la partición, el valor de pic\_width\_in\_luma\_samples será menor o igual que  $\text{Sqrt}(\text{MaxLumaPs} * 8)$ .

30 c) Para cada capa en la partición, el valor de pic\_height\_in\_luma\_samples será menor o igual a  $\text{Sqrt}(\text{MaxLumaPs} * 8)$ .

35 d) Para cada capa con nuh\_layer\_id igual a currLayerId en la partición, el valor de max\_vps\_dec\_pic\_buffering\_minus1[ optLsIdx ][ LayerIdxInVps[ currLayerId ] ][ HighestTid ] será menor o igual que MaxDpbSize como se obtiene de la ecuación A-2, con PicSizeInSamplesY siendo reemplazado por partitionSizeInSamplesY, para el grado y el nivel de la partición.

e) Para el nivel 5 y niveles superiores, el valor de CtbSizeY para cada capa en la partición será igual a 32 o 64.

40 f) El valor de NumPicTotalCurr para cada imagen en la partición será menor o igual a 8.

45 g) Al descodificar cada imagen codificada en la partición, el valor de num\_tile\_columns\_minus1 será menor que MaxTileCols y num\_tile\_rows\_minus1 será menor que MaxTileRows, donde MaxTileCols y MaxTileRows se especifican en la Tabla A-1 para el grado y el nivel de la partición.

50 h) Para los parámetros VCL HRD de la partición, CpbSize[ i ] debe ser menor o igual a  $1000 * \text{MaxCPB}$  para al menos un valor de i en el rango de 0 a cpb\_cnt\_minus1[ HighestTid ], inclusive, donde se especifica CpbSize[ i ] en la subcláusula E.3.3 basándose en los parámetros seleccionados como se especifica en la subcláusula C.1 y el MaxCPB se especifica en la Tabla A-1 para el grado y el nivel de la partición, en unidades de 1000 bits.

55 i) Para los parámetros NAL HRD de la partición, CpbSize[ i ] debe ser menor o igual a  $1100 * \text{MaxCPB}$  para al menos un valor de i en el rango de 0 a cpb\_cnt\_minus1[ HighestTid ], inclusive, donde se especifica CpbSize[ i ] en la subcláusula E.3.3 basándose en los parámetros seleccionados como se especifica en la subcláusula C.1 y MaxCPB se especifica en la Tabla A-1 para el grado y el nivel de la partición, en unidades de 1100 bits.

**[0178]** La Tabla A-1 especifica los límites para cada nivel de cada grado.

**[0179]** Un grado y nivel al que se ajusta una partición se indican mediante los elementos sintácticos general\_tier\_flag y general\_level\_idc de la forma siguiente:

60 - general\_tier\_flag igual a 0 indica conformidad con el grado principal, y general\_tier\_flag igual a 1 indica conformidad con el grado alto, de acuerdo con las restricciones de grado especificadas en la Tabla A-1. general\_tier\_flag será igual a 0 para niveles inferiores al nivel 4 (correspondiente a las entradas en la Tabla A-1 marcadas con "-").

65 - general\_level\_idc se establecerá igual a un valor de 30 veces el número de nivel especificado en la Tabla A-1.

[0180] JCTVC-Q0145 incluye un archivo adjunto (JCTVC-Q0145\_MvHvcText) que describe los cambios en Tech et al., "MV-HEVC Draft Text 7", Equipo colaborativo conjunto en codificación de vídeo 3D, Desarrollo de extensión de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 7.<sup>a</sup> reunión: San José, EE. UU., 11-17 de enero de 2014, JCT3V-G1004-v7. El siguiente texto indica cambios en JCTVC-Q0145\_MvHvcText para implementar límites y restricciones de nivel específicos de partición para perfiles, grados y niveles. Para ser concisos, el texto eliminado de JCTVC-Q0145\_MvHvcText se omite a continuación.

#### G.11.2.2 Límites de nivel específicos del perfil para los perfiles principal, principal 10 y estéreo principal

[0181] Cada partición que se ajuste al perfil estéreo principal en un grado y nivel específico deberá cumplir las siguientes restricciones para cada prueba de conformidad de flujo de bits especificada en el Anexo C, donde "unidad de acceso" se utiliza para indicar el conjunto de imágenes contenidas en la partición, junto con sus unidades NAL no-VCL asociadas, de capas contenidas en la partición y que tienen el mismo tiempo de salida y se entiende que la CPB es el BPB:

a) El momento de eliminación nominal de la unidad de acceso  $n$  (con  $n$  mayor que 0) de la CPB, como se especifica en la subcláusula C.2.3, debe cumplir la restricción de que  $AuNominalRemovalTime[n] - AuCpbRemovalTime[n - 1]$  sea mayor o igual que  $Max(\text{partitionSizeInSamplesY} \div \text{MaxLumaSr}, fR)$ , donde  $\text{partitionSizeInSamplesY}$  es el valor de  $\text{partitionSizeInSamplesY}$  para la unidad de acceso  $n - 1$  y  $\text{MaxLumaSr}$  es el valor especificado en la Tabla A-2 que se aplica a la unidad de acceso  $n - 1$  para el grado y nivel de la partición.

b) La diferencia entre los tiempos de salida consecutivos de las imágenes de la DPB, como se especifica en la subcláusula C.3.3, debe cumplir la restricción de que  $DpbOutputInterval[n]$  es mayor o igual a  $Max(\text{partitionSizeInSamplesY} \div \text{MaxLumaSr}, fR)$ , donde  $\text{partitionSizeInSamplesY}$  es el valor de  $\text{partitionSizeInSamplesY}$  de la unidad de acceso  $n$  para el grado y el nivel de la partición, siempre que la unidad de acceso  $n$  sea una unidad de acceso que tenga una imagen que se emita y no sea la última de esas unidades de acceso.

c) El momento de eliminación de la unidad de acceso 0 cumplirá la restricción de que el número de segmentos de fragmento codificados en la unidad de acceso 0 es menor o igual a  $Min( Max( \text{numLayersInPartition}, \text{MaxSliceSegmentsPerPicture} * \text{MaxLumaSr} / \text{MaxLumaPs} * (AuCpbRemovalTime[0] - AuNominalRemovalTime[0]) + \text{MaxSliceSegmentsPerPicture} * \text{partitionSizeInSamplesY} / \text{MaxLumaPs} ), \text{MaxSliceSegmentsPerPicture} )$ , donde  $\text{MaxSliceSegmentsPerPicture}$ ,  $\text{MaxLumaPs}$  y  $\text{MaxLumaSr}$  son los valores especificados en la Tabla A-1 y la Tabla A-2 para el grado y el nivel de la partición.

d) La diferencia entre los tiempos de eliminación de CPB consecutivos de las unidades de acceso  $n$  y  $n - 1$  (con  $n$  mayor que 0) debe satisfacer la restricción de que el número de segmentos de fragmento codificados en la unidad de acceso  $n$  es menor o igual a  $Min( (Max(\text{numLayersInPartition}, \text{MaxSliceSegmentsPerPicture} * \text{MaxLumaSr} / \text{MaxLumaPs} * (AuCpbRemovalTime[n] - AuCpbRemovalTime[n - 1]))), \text{MaxSliceSegmentsPerPicture} )$ , donde  $\text{numLayersInPartition}$  es el valor para la unidad de acceso  $n$ , y  $\text{MaxSliceSegmentsPerPicture}$ ,  $\text{MaxLumaPs}$  y  $\text{MaxLumaSr}$  son los valores especificados en Tabla A-1 y Tabla A-2 que se aplican a la unidad de acceso  $n$  para el grado y el nivel de la partición.

e) Para los parámetros VCL HRD para la partición,  $\text{BitRate}[i]$  debe ser menor o igual a  $\text{CpbBrVclFactor} * \text{MaxBR}$  para al menos un valor de  $i$  en el rango de 0 a  $\text{cpb\_cnt\_minus1}[\text{HighestTid}]$ , inclusive, donde se especifica  $\text{BitRate}[i]$  en la subcláusula E.3.3 basada en parámetros seleccionados como se especifica en la subcláusula C.1 y  $\text{MaxBR}$  se especifica en la Tabla A-2 en unidades de  $\text{CpbBrVclFactor}$  bits/s, donde  $\text{CpbBrVclFactor}$  se especifica en la Tabla A-3 para el grado y nivel de la partición.

f) Para los parámetros NAL HRD para la partición,  $\text{BitRate}[i]$  debe ser menor o igual que  $\text{CpbBrNalFactor} * \text{MaxBR}$  para al menos un valor de  $i$  en el rango de 0 a  $\text{cpb\_cnt\_minus1}[\text{HighestTid}]$ , inclusive, donde se especifica  $\text{BitRate}[i]$  en la subcláusula E.3.3 basándose en parámetros seleccionados como se especifica en la subcláusula C.1 y  $\text{MaxBR}$  se especifica en la Tabla A-2 en unidades de  $\text{CpbBrNalFactor}$  bits/s, donde  $\text{CpbBrNalFactor}$  se especifica en la Tabla A-3 para el grado y nivel de la partición.

g) La suma de las variables  $\text{NumBytesInNalUnit}$  para la unidad de acceso 0 será menor o igual que  $\text{FormatCapabilityFactor} * (Max(\text{partitionSizeInSamplesY}, fR * \text{MaxLumaSr}) + \text{MaxLumaSr} * (AuCpbRemovalTime[0] - AuNominalRemovalTime[0])) \div \text{MinCr}$ , donde  $\text{MaxLumaSr}$  y  $\text{MinCr}$  son los valores especificados en la Tabla A-2 y  $\text{FormatCapabilityFactor}$  es el valor especificado en la Tabla A-3 que se aplica a la imagen 0 para el grado y el nivel de la partición.

h) La suma de las variables  $\text{NumBytesInNalUnit}$  para la unidad de acceso  $n$  (con  $n$  mayor que 0) debe ser menor o igual que  $\text{FormatCapabilityFactor} * \text{MaxLumaSr} * (AuCpbRemovalTime[n] - AuCpbRemovalTime[n - 1]) \div \text{MinCr}$ , donde  $\text{MaxLumaSr}$  y  $\text{MinCr}$  son los valores especificados en la Tabla A-2 y  $\text{FormatCapabilityFactor}$  es el valor especificado en la Tabla A-3 que se aplica a la imagen  $n$  para el grado y el nivel de la partición.



i) El momento de eliminación de la unidad de acceso 0 satisfará la restricción de que el número de elementos en imágenes codificadas en la unidad de acceso 0 es menor o igual a  $\text{Min}(\text{Max}(\text{numLayersInPartition}, \text{MaxTileCols} * \text{MaxTileRows} * 120 * (\text{AuCpbRemovalTime}[0] - \text{AuNominalRemovalTime}[0]) + \text{MaxTileCols} * \text{MaxTileRows}), \text{MaxTileCols} * \text{MaxTileRows})$ , donde  $\text{MaxTileCols}$  y  $\text{MaxTileRows}$  son los valores especificados en la Tabla A-1 para el grado y el nivel de la partición.

j) La diferencia entre los tiempos de eliminación de CPB consecutivos de unidades de acceso  $n$  y  $n - 1$  (con  $n$  mayor que 0) debe satisfacer la restricción de que el número de elementos en imágenes codificadas en la unidad de acceso  $n$  es menor o igual a  $\text{Min}(\text{Max}(\text{numLayersInPartition}, \text{MaxTileCols} * \text{MaxTileRows} * 120 * (\text{AuCpbRemovalTime}[n] - \text{AuCpbRemovalTime}[n - 1])), \text{MaxTileCols} * \text{MaxTileRows})$ , donde  $\text{numLayersInPartition}$  son los valores para la unidad de acceso  $n$ , y  $\text{MaxTileCols}$  y  $\text{MaxTileRows}$  son los valores especificados en la Tabla A- 1 que se aplica a la unidad de acceso  $n$  para el grado y el nivel de la partición.

[0182] Además, de acuerdo con algunos ejemplos de esta divulgación, se pueden hacer los siguientes cambios en el Párrafo G.11.1.2 Perfil estéreo principal de MV-HEVC WD8. En el texto a continuación, se muestran los cambios relativos a la versión de ejemplo del Párrafo G.11.1.2 descrito anteriormente en esta divulgación.

### G.11.1.2 Perfil estéreo principal

[0183] Para un flujo de bits que contenga conjuntos de capas de salida con particiones que se ajusten al perfil estéreo principal, dejar que

- olsIdx sea el índice de conjunto de capas de salida de un conjunto de capas de salida.
- layerSetIdx sea el índice de conjunto de capas para el conjunto de capas de salida.
- el sub-flujo de bits del conjunto de capas de salida se obtenga invocando el proceso de extracción de sub-flujo de bits como se especifica en la subcláusula F.10 con las siguientes entradas: el flujo de bits, tldTarget igual a 6, y layerIdListTarget que contiene los valores nuh\_layer\_id de las capas que están incluidas en el conjunto de capas con el índice layerSetIdx y son capas de imágenes principales.
- el flujo de bits de la capa base se obtiene invocando el proceso de extracción de sub-flujo de bits como se especifica en la subcláusula F.10 con el flujo de bits, tldTarget igual a 6 y layerIdListTarget que contiene solo un valor de nuh\_layer\_id que es igual a 0 como entradas.

[0184] Unos [flujos de bits que contienen conjuntos de capas de salida con particiones de capas conforme a los perfiles estéreo principal cumplirán las siguientes restricciones en el sub-flujo de bits obtenido para cada uno de los conjuntos de capas de salida, siendo layerSetIdx el índice del conjunto de capas para el conjunto de capas de salida, obteniéndose el sub-flujo de bits invocando el proceso de extracción de sub-flujo de bits como se especifica en la subcláusula F.10 con el flujo de bits, tldTarget igual a 7, y con TargetDecLayerIdList que contiene los valores de nuh\_layer\_id del conjunto de capas con el índice layerSetIdx como entradas, obteniéndose las particiones invocando el proceso de demultiplexación para obtener una partición de flujo de bits como se especifica en la subcláusula C.6 con el sub-flujo de bits, la lista de capas en la partición listOfLayersInPartition, y el número de capas en la partición numLayersInPartition como entradas B]] flujos de bit[es correspondientes a sub-flujos de bits para]] que contienen conjuntos de capas de salida con particiones [de capas]] conforme al perfil estéreo principal cumplirán las siguientes restricciones:

- Todos los VPS activos deberán tener vps\_base\_layer\_internal\_flag igual a 1 solo.
- Todos los VPS activos deberán tener vps\_num\_rep\_formats\_minus1 en el rango de 0 a 15, inclusive.
- [[Los flujos de bits que contienen conjuntos de capas de salida conforme al perfil estéreo principal cumplirán las siguientes restricciones en el sub-flujo de bits obtenido para cada uno de los conjuntos de capas de salida, con el índice de conjunto de capas de salida olsIdx y layerSetIdx siendo el conjunto de capas para el conjunto de capas de salida, y obteniéndose el sub-flujo de bits invocando el proceso de extracción de sub-flujo de bits como se especifica en la subcláusula F.10 con las siguientes entradas: el flujo de bits, tldTarget igual a 6 y layerIdListTarget que contiene los valores nuh\_layer\_id de las capas que están incluidas en el conjunto de capas con el índice layerSetIdx y son capas de imagen principales.]]

[0185] [[Los flujos de bits que contienen conjuntos de capas de salida que se ajustan al perfil estéreo principal también deben cumplir las siguientes restricciones en el flujo de bits de capa base obtenido invocando el proceso de extracción de sub-flujo de bits especificado en la subcláusula F.10 con el flujo de bits, tldTarget igual a 6, y con layerIdListTarget que contiene solo un valor nuh\_layer\_id que es igual a 0 como entradas.]]

**[0186]** El flujo de bits de la capa base *[[obtenido de los flujos de bits que se ajustan a los perfiles estéreo principal]]* deberá cumplir las siguientes restricciones:

- El flujo de bits de la capa base debe cumplir todas las restricciones del perfil principal especificado en la subcláusula A.3.2.
- Todos los SPS activos *[[de]] para* la capa base *[[flujo de bits]]* tendrán `general_profile_idc` igual a 1 o `general_profile_compatibility_flag[ 1 ]` igual a 1.

**[0187]** El sub-flujo de bits *[[obtenido]] de cada [[para un]]* conjunto de capas de salida *[[que se ajuste al perfil estéreo principal]]* deberá cumplir las siguientes restricciones:

- Todos los SPS activos para el sub-flujo de bits deberá tener `chroma_format_idc` igual a 1 solo.
- `CtbLog2SizeY` obtenido de cualquier SPS activo *[[de]] para* el sub-flujo de bits estará en el rango de 4 a 6, inclusive.
- `ScalabilityId[ j ] [ smldx ]` será igual a 0 para cualquier valor de `smldx` no igual a 1 y para cualquier valor de `j` para el sub-flujo de bits.

**[0188]** En el resto de esta subcláusula y subcláusula G.11.3, para un sub-flujo de bits de un conjunto de capas de salida, `general_profile_idc`, `general_profile_compatibility_flag[ i ]`, `general_tier_flag`, y `general_level_idc` se refieren a `general_profile_idc`, `general_profile_compatibility_flag[ i ]`, `general_tier_flag`, y `general_level_idc`, respectivamente, en la estructura sintáctica `profile_tier_level( )` para la partición *[[conjunto de capas de salida]]*.

**[0189]** La conformidad de un sub-flujo de bits de una partición *[[conjunto de capas de salida]]* con el perfil estéreo principal se indica con `general_profile_idc` siendo igual a 6 *[[5]]* o `general_profile_compatibility_flag [ 6 ]` *[[5]]* siendo igual a 1.

- NOTA - Cuando `general_profile_compatibility_flag[ 6 ]` *[[5]]* es igual a 1, `general_profile_compatibility_flag[ 1 ]` *[[i]]* *[[para i igual a 1]]* también debería ser igual a 1.

**[0190]** Además, en algunos ejemplos de esta divulgación, se pueden hacer los siguientes cambios en el Párrafo G.11.2.1 de MV-HEVC WD8. El texto a continuación indica los cambios de una parte de la versión de ejemplo del Párrafo G.11.2.1 descrito anteriormente en esta divulgación.

#### G. 11.2.1 Límites generales de grado y nivel

**[0191]** La condición (a) se cambia de la forma siguiente:

- j) *[[Para cada capa en el]]* El valor de la partición *[[,]]* `PicSizeInSamplesY` será menor o igual que `MaxLumaPs`, donde `MaxLumaPs` se especifica en la Tabla A-1 para el grado y el nivel de la partición.

**[0192]** Además, en algunos ejemplos de esta divulgación, se hacen los siguientes cambios en el Párrafo G.11.2.2 de MV-HEVC WD8. El texto a continuación indica los cambios de una parte de la versión de ejemplo del Párrafo G.11.2.1 descrito anteriormente en esta divulgación.

#### G.11.2.2 Límites de nivel específicos del perfil para los perfiles principal, principal 10 y estéreo principal

**[0193]** Cada partición que se ajuste al perfil estéreo principal en un grado y nivel especificado cumplirá las siguientes restricciones para *[[cada flujo de bits]]* la prueba de conformidad especificada en el Anexo C, donde "unidad de acceso" se utiliza para designar el conjunto de imágenes contenidas en la partición, junto con sus unidades no VCL NAL asociadas, de capas contenidas en la partición y que tienen el mismo tiempo de salida y se entiende que la CPB es el BPB:

**[0194]** Además, como se describe en otra parte de acuerdo con una o más técnicas de esta divulgación, para cada esquema de partición respectivo de cada conjunto de capas de salida, un mensaje VPS VUI o SEI de un flujo de bits puede identificar un conjunto de parámetros de HRD para el esquema de partición respectivo. Los cambios que se muestran en la Tabla 4 pueden realizarse en la estructura sintáctica de los parámetros de HRD de la partición de flujo de bits VUI de VPS definida en el Párrafo F.7.3.2.1.6 de MV-HEVC WD8 y SHVC WD6 para implementar dichos ejemplos.

#### TABLA 4

vps_vui_bsp_hrd_parameters( ) {	Descriptor
<b>vps_num_bsp_hrd_parameters_minus1</b>	ue(v)
for( i = 0; i <= vps_num_bsp_hrd_parameters_minus1; i++) {	
if(i>0)	
<b>bsp_cprms_present_flag[ i ]</b>	u(1)
hrd_parameters(bsp_cprms_present_flag[ i ],	
vps_max_sub_layers_minus1)	
}	
for(h = 1; h <= NumOutputLayerSets[ <i>vps_num_layer_sets_minus1</i> ]; h++){	
[[ <b>num_bitstream_partitions[h]</b> ]]	[[ue(v)]]
[[ for(i = 0; i < num_bitstream_partitions[ h ]; i++) ]]	
[[for(j = 0; j < NumLayersInIdList[ h ];j++) ]]	
[[ <b>layer_in_bsp_flag[ h ][ i ][ j ]</b> ]]	[[u(1)]]
for( i = 1; i <= num_partitioning_schemes_minus1[ h ]; i++) {	
[[ if(num_bitstream_partitions[ h ] > 0) { ]]	
<b>num_bsp_sched_combinations_minus1[ h ][ i ]</b>	ue(v)
for( [[i]] = 0; [[i]] <= num_bsp_sched_combinations_minus1[ h ][ i ]; [[i]]++)	
for([[j]] = 0; [[j]] < [[n]]Num[[p]]Partitions[[i]]N[[s]]Scheme[[minus1	
num_bitstream_partitions][ h ][ i ]; [[j]]++) {	
<b>bsp_comb_hrd_idx[ h ][ i ][ j ][ k ]</b>	ue(v)
<b>bsp_comb_sched_idx[ h ][ i ][ j ][ k ]</b>	ue(v)
}	
}	
}	
}	
}	

**[0195]** De acuerdo con el ejemplo de la Tabla 4, se pueden realizar los siguientes cambios en la semántica de los parámetros de HRD de la partición de flujo de bits VUI de VPS, como se define en los párrafos F.7.4.3.1.6 de MV-HEVC WD8 y SHVC WD6.

5            [[**num\_bitstream\_partitions**[ h ] especifica el número de particiones de flujo de bits para las que se especifican los parámetros de HRD para el conjunto de capas con el índice h.

10           **layer\_in\_bsp\_flag** [ h ][ i ][ j ] especifica que la capa con el índice j es una parte de la partición de flujo de bits con el índice i dentro del conjunto de capas con el índice h.]]

15           **num\_bsp\_sched\_combinations\_minus1** [ h ][ i ] más 1 especifica el número de combinaciones de programas de entregas y hrd\_parameters( ) especificados para las particiones de flujo de bits para el i-ésimo esquema de partición del h-ésimo conjunto de capas de salida [[con índice h]].

La variable SchedCombCnt[ h ][ i ] se establece igual a num\_bsp\_sched\_combinations\_minus1[ h ][ i ] + 1.

20           **bsp\_comb\_hrd\_idx** [ h ][ i ][ j ][ k ] especifica el índice de hrd\_parameters( ) dentro de la estructura sintáctica vps\_vui\_bsp\_hrd\_parameters( ) utilizada en la [[i]]-ésima combinación de un programa de entregas y hrd\_parameters( ) especificados para la k-ésima partición [[flujo de bits]] [[con índice jy]] del i-ésimo esquema de partición para el h-ésimo conjunto de capas de salida [[con índice h]].

25           **bsp combinas sched\_idx** [ h ][ i ][ j ][ k ] especifica el índice de un programa de entregas dentro de la estructura sintáctica hrd\_parameters( ) con el índice bsp\_comb\_hrd\_idx [ h ][ i ][ j ][ k ] que se utiliza en la [[i]] j-ésima combinación de un programa de entregas y hrd\_parameters( ) especificados para la k-ésima partición [[flujo de bits]] [[con índice jy]] del i-ésimo esquema de partición [[y]] para el h-ésimo conjunto de capas de salida [[con índice h]].

**[0196]** Además, de acuerdo con una o más técnicas de esta divulgación, los cambios mostrados en la Tabla 5, a continuación, pueden realizarse en la sintaxis de mensaje SEI de los parámetros de HRD de la partición de flujo de bits, como se define en el Párrafo F.14.1.5 de MV-HEVC WD8 y SHVC WD6.

5

**TABLA 5**

	Descriptor
bsp_hrd(payloadSize) {	
<b>sei_num_bsp_hrd_parameters_minus1</b>	ue(v)
for(i = 0; i <= sei_num_bsp_hrd_parameters_minus1; i++) {	
if(i > 0)	
<b>sei_bsp_cprms_present_flag[ i ]</b>	u(1)
hrd_parameters(sei_bsp_cprms_present_flag[ i ],	
nesting_max_temporal_id_plus1[ 0 ] - 1)	
}	
for(h = 0; h <= nesting_num_ops_minus1; h++) {	
olsIdx = nesting_op_idx[h]	
<b>num_sei_partitioning_schemes_minus1[olsIdx]</b>	ue(v)
for(i = 1; i <= num_sei_partitioning_schemes_minus1[olsIdx]; i++) {	
<b>__num_sei_bitstream_partitions_minus1[1]/2[olsIdx][i]</b>	ue(v)
__for( [[i]] j = 0; [[i]] j <= [[n]]Num[[s]]Sei[[b]]Bitstream[[p]]Partitions[[minus1]][olsIdx][i]; ij++)	
__for( [[i]] k = 0; [[i]] k <= vps_max_layers_minus1; [[i]] k++)	
__if(layer_id_included_flag[LayerSetIdxForOutputLayerSet[olsIdx]][[layer_id_in_nuh][j][k]])	
<b>__sei_layer_in_bsp_flag[olsIdx][i][j][k]</b>	u(1)
<b>__sei_num_bsp_sched_combinations_minus1[olsIdx][i]</b>	ue(v)
__for( [[i]] k = 0; [[i]] k <= sei_num_bsp_sched_combinations_minus1[olsIdx][i]; [[i]] k++)	
__for( [[i]] k = 0; [[i]] k <= vps_max_layers_minus1; [[i]] k++)	
<b>__sei_bsp_comb_hrd_idx[olsIdx][i][j][k]</b>	ue(v)
<b>__sei_bsp_comb_sched_idx[olsIdx][i][j][k]</b>	ue(v)
__}	
__}	
}	
}	

}	
for( h = 0; h <= nesting_num_ops_minus1; h++) {	
olsIdx = nesting_op_idx[ h ]	
<b>num_sei_partitioning_schemes_minus1[ olsIdx ]</b>	ue(v)
for( i = 1; i <= num sei partitioning schemes minus 1[ olsIdx ]; i++) {	
<b>__num_sei_bitstream_partitions_minus1[1]/2[ olsIdx ][ i ]</b>	ue(v)
__for( [[i]] j = 0; [[i]] j <= [[n]]Num[[s]]Sei[[b]]Bitstream[[p]]Partitions[[minus1]][ olsIdx ][ i ]; ij++)	
__for( [[i]] k = 0; [[i]] k <= vps_max_layers_minus1; [[i]] k++)	
__if(layer_in_included_flag[ LayerSetIdxForOutputLayerSet[ olsIdx ] ][ [[layer_id_in_nuh[ j ] ] k	

[[i]] )	
<b>_sei_layer_in_bsp_flag[ olslidx ][ i ][ j ][ k ]</b>	u(1)
<b>_sei_num_bsp_sched_combinations_minus1[ olslidx ][ i ]</b>	ue(v)
<b>_for( [[i]] j = 0; [[i]] j &lt;= sei_num_bsp_sched_combinations_minus1[ olslidx ][ i ]; [[i]]++)</b>	
<b>_for( [[i]] k = 0; [[i]] k &lt;= [[n]]Num[[_s]]Sei[[_b]]Bitstream[[_p]]Partitions[[_minus1]][[ olslidx ][ i ]; [[i]]k++) {</b>	
<b>_sei_bsp_comb_hrd_idx[ olslidx ][ i ][ j ][ k ]</b>	ue(v)
<b>_sei_bsp_comb_sched_idx[ olslidx ][ i ][ j ][ k ]</b>	ue(v)
<b>}_</b>	
<b>}_</b>	
<b>}</b>	
<b>}</b>	

[0197] Además, en el ejemplo de la Tabla 5, el mensaje SEI de agrupamiento de partición de flujo de bits proporciona un mecanismo para asociar mensajes SEI con una partición de flujo de bits de un conjunto de capas de salida. En el ejemplo de la Tabla 5, pueden hacerse los siguientes cambios en la semántica de mensaje SEI de los parámetros de HRD de la partición de flujo de bits en el Párrafo F.14.2.5 de MV-HEVC WD8 y SHVC WD6.

**num sei partitioning schemes minus1 [ olslidx ]** más 1 especifica el número f de esquemas de partición especificados para el conjunto de capas de salida con el índice olslidx. El valor de **num sei partitioning schemes minus1 [ olslidx ]** estará en el rango de 0 a 15, inclusive.

**num sei bitstream partitions minus2[[i]] [ olslidx ][ i ]** más  $\frac{[[i]]}{2}$  especifica el número de particiones de flujo de bits para las cuales se especifican los parámetros de HRD para el esquema de partición i-ésimo del conjunto de capas de salida con el índice olslidx. El valor de **num sei bitstream partitions minus1 [ olslidx ][ i ]** estará en el rango de 0 a **NumLayersInIdList[ LayerSetIdxForOutputLayerSet[ olslidx ] ] - 1**, inclusive. El valor de **num sei bitstream partitions minus1 [ olslidx ][ 0 ]** se infiere que es igual a 0.

El valor de la variable **NumPartitionInScheme[ i ][ j ]** se obtiene de la forma siguiente:

$$\text{NumSeiBitstreamPartitions [ olslidx ][ i ]} = (i = 0) ? 1 : \text{num sei bitstream partitions minus2[ i ][ j ]} + 2.$$

**sei\_layer\_in\_bsp\_flag[ olslidx ][ i ][ j ][ k ]** especifica que la capa con índice  $[[j]] k$  es una parte de la partición de flujo de bits con índice  $[[i]] j$  dentro del esquema de partición con índice i del conjunto de capas con índice olslidx. El valor de **layer included in partition flag[ olslidx ][ 0 ][ 0 ][ k ]** se infiere que es igual al valor de **layer id included flag[ LayerSetIdxForOutputLayerSet[ olslidx ] ][ k ]** para k en el rango de 0 a **vps max layer minus1**, inclusive.

[0198] Es un requisito de conformidad de flujo de bits que la partición de flujo de bits con índice  $[[i]] n$  no incluya capas de referencia directas o indirectas de ninguna capa en la partición de flujo de bits con índice  $[[i]] m$  para ningún valor de  $[[i]] m$  y  $[[i]] n$  en el rango de 0 a **Num[[\_s]]Sei[[\_b]]Bitstream[[\_p]]Partitions [[\_minus1]][[ olslidx ][ i ]]**, inclusive], tal que  $[[i]] m$  es menor que  $[[i]] n$ .

**sei\_num\_bsp\_sched\_combinations\_minus1 [ olslidx ][ i ]** más 1 especifica el número de combinaciones de programas de entrega y **hrd\_parameters( )** especificados para las particiones de flujo de bits para el esquema de partición con el índice i del conjunto de capas de salida con el índice olslidx.

La variable **SchedCombCnt[ olslidx ][ i ]** se establece igual a **sei\_num\_bsp\_sched\_combinations\_minus1 [ olslidx ][ i ] + 1**.

**sei\_bsp\_comb\_hrd\_idx [ olslidx ][ i ][ j ][ k ]** especifica el índice de **hrd\_parameters( )** dentro de este mensaje SEI utilizado en la  $[[i]] j$ -ésima combinación de un programa de entregas y **hrd\_parameters( )** especificado para la partición de flujo de bits con el índice  $[[i]] k$  del esquema de partición con el índice i  $[[i]] y$  para el conjunto de capas de salida con el índice olslidx.

**sei\_bsp\_comb\_sched\_idx [ olslidx ][ i ][ j ][ k ]** especifica el índice de un programa de entrega dentro de la estructura sintáctica de **hrd\_parameters( )** con el índice **sei\_bsp\_comb\_hrd\_idx[ olslidx ][ i ][ j ][ k ]** que se utiliza en la  $[[i]] j$ -ésima combinación de un programa de entregas y **hrd\_parameters( )** especificada para la partición de flujo de bits con índice  $[[i]] k$  del esquema de partición con índice i  $[[i]] y$  para el conjunto de capas de salida con índice olslidx.

[0199] El Anexo C en MV-HEVC WD8 y SHVC WD6 describe el HRD. Más particularmente, el párrafo C.1 de MV-HEVC WD8 y SHVC WD6 indica que pueden ser necesarias múltiples pruebas para comprobar la conformidad de un flujo de bits, lo cual se conoce como el flujo de bits bajo prueba. Además, el párrafo C.1 de MV-HEVC WD8 y SHVC WD6 indica una serie de pasos a seguir para comprobar la conformidad de un flujo de bits.

[0200] En algunos ejemplos de esta divulgación, la serie de pasos descritos en el Párrafo C.1 de MV-HEVC WD8 y SHVC WD6 se modifica con el fin de soportar el esquema de partición descrito en otra parte en esta divulgación. Por ejemplo, se pueden hacer las siguientes modificaciones en el Párrafo C.1 de MV-HEVC WD8 y SHVC WD6.

[0201] Para cada prueba, se aplican los siguientes pasos en el orden indicado:

1. Un punto de funcionamiento bajo prueba, designado como TargetOp, se selecciona al seleccionar un valor para TargetOutputLayerSetIdx que identifica un conjunto de capas de salida objetivo y selecciona un valor más alto de TemporalId objetivo HighestTid. El valor de TargetOutputLayerSetIdx estará en el rango de 0 a NumOutputLayerSets - 1, inclusive. El valor de HighestTid estará en el rango de 0 a MaxSubLayersInLayerSetMinus1[ LayerSetIdxForOutputLayerSet [ Target OutputLayerSetIdx ]], inclusive. Las variables TargetDecLayerSetIdx, TargetOptLayerIdList y TargetDecLayerIdList se obtienen como se especifica en la ecuación 8-1. El punto de funcionamiento bajo prueba tiene OptLayerIdList igual a TargetOptLayerIdList, OpLayerIdList igual a TargetDecLayerIdList, y OpTid igual a HighestTid.

2. El proceso de extracción sub-flujo de bits como se especifica en la cláusula 10 se invoca con el flujo de bits bajo prueba, HighestTid y TargetDecLayerIdList como entradas, y la salida se asigna a BitstreamToDecode. Un esquema de partición se selecciona de la lista de esquemas de partición señalados en el VPS activo para el conjunto de capas de salida seleccionado. Después de eso, para el esquema de partición seleccionado, se selecciona una partición de la lista de particiones señaladas en el VPS para el conjunto de capas de salida seleccionado.

3. Si la partición seleccionada contiene todas las capas del conjunto de capas de salida seleccionado, se selecciona el funcionamiento de CPB específico del flujo de bits. De lo contrario, se selecciona el funcionamiento de CPB específico de partición de flujo de bits. Los pasos siguientes se aplican al sub-flujo de bits de la partición seleccionada, denominada partición de flujo de bits bajo prueba, [[Cuando la estructura sintáctica vps vui\_bsp\_hrd\_parameters() está presente en el VPS activo, se selecciona un esquema de partición, designado como TargetPartitioningScheme, seleccionando un valor para TargetPartitioningSchemeldx que está en el rango de 1 a num\_partitioning\_schemes\_minus1 [TargetOutputLayerSetIdx], inclusive. Cuando y num\_bitstream\_partitionsNum\_pPartitions\_in\_sScheme\_minus1[ TargetDecOutputLayerSetIdx][ TargetPartitioningSchemeldx] es mayor que 1 01, donde TargetPartitioningSchemeldx es el índice del TargetPartitioningScheme seleccionado que está en el rango de 1 a num\_partitioning\_schemes\_minus1[TargetOutputLayerSetIdx] o está presente un mensaje SEI de parámetros de HRD de partición de flujo de bits y el mensaje SEI contiene el elemento sintáctico num\_sei\_bitstream\_partitions\_in\_scheme bitstream\_partitions\_minus1[TargetDecOutputLayerSetIdx][TargetPatitioningSchemeldx] mayor que 0, se selecciona el funcionamiento de CPB específico del flujo de bits o el funcionamiento de CPB específico de partición de flujo de bits para una prueba de conformidad, y ambos funcionamientos de CPB se deben probar para comprobar la conformidad de un flujo de bits. Cuando se prueba el funcionamiento de CPB específico del flujo de bits, los pasos subsiguientes se aplican al flujo de bits bajo prueba. Cuando se prueba el funcionamiento de CPB específico de partición de flujo de bits, los pasos subsiguientes se aplican a cada partición de flujo de bits de los flujos de bits bajo prueba, a los que se hace referencia como partición de flujos de bits bajo prueba. Cuando se prueba el funcionamiento de CPB específico de partición de flujo de bits y la entrada al HRD es un flujo de bits, las particiones de flujo de bits se obtienen con el proceso de desmultiplexación para obtener una partición de flujo de bits en la subcláusula C.6. Cuando se selecciona el funcionamiento de CPB específico de partición de flujo de bits, se supone que las particiones de flujo de bits de otras particiones del mismo esquema de partición del conjunto de capas de salida seleccionado se descodifican mediante capacidades de descodificación distintas del HRD pero disponibles para el HRD según lo programado de acuerdo con los parámetros de HRD de esas particiones.

4. La estructura sintáctica hrd\_parameters( ) y la estructura sintáctica sub\_layer\_hrd\_parameters( ) aplicable a TargetOp se seleccionan de la forma siguiente:

- Si se prueba el funcionamiento de CPB específico del flujo de bits, se aplica lo siguiente:
  - Si TargetDecLayerIdList contiene todos los valores nuh\_layer\_id presentes en el flujo de bits bajo prueba, se selecciona la estructura sintáctica hrd\_parameters( ) en el SPS activo para la capa base (o proporcionado a través de medios externos no especificados en esta memoria descriptiva).

- 5

  - De lo contrario, se selecciona la estructura sintáctica `hrd_parameters( )` en el VPS activo (o proporcionado a través de medios externos no especificados en esta memoria descriptiva) que se aplica a `TargetOp`.
- 5

  - De lo contrario, se selecciona la estructura sintáctica `hrd_parameters( )` de la forma siguiente:
  - Se puede seleccionar cualquiera de las estructuras sintácticas `hrd_parameters( )` en las siguientes condiciones, si las dos condiciones siguientes son verdaderas:
- 10

  - La estructura sintáctica `vps_vui_bsp_hrd_parameters( )` está presente en el VPS activo (o está disponible a través de medios externos no especificados en esta memoria descriptiva) y contiene una estructura sintáctica `hrd_parameters( )` que se aplica a `TargetOp`, a `TargetPartitioningScheme` y a la partición de flujo de bits bajo prueba.
- 15

  - Está presente un mensaje SEI de parámetros de HRD de partición de flujo de bits que se incluye en un SEI de agrupamiento escalable que se aplica a `TargetOp` y contiene una estructura sintáctica `hrd_parameters( )` que se aplica a `TargetOp`, a `TargetPartitioningScheme` y a la partición de flujo de bits bajo prueba presente (o está disponible a través de medios externos no especificados en esta memoria descriptiva).
- 20

  - De lo contrario, si la estructura sintáctica `vps_vui_bsp_hrd_parameters( )` está presente en el VPS activo (o está disponible a través de medios externos no especificados en esta memoria descriptiva) y contiene una estructura sintáctica `hrd_parameters( )` que se aplica a `TargetOp`, `TargetPartitioningScheme` y la partición de flujo de bits bajo prueba, se selecciona esa estructura sintáctica `hrd_parameters( )`.
- 25

  - De lo contrario, estará presente y se seleccionará una estructura sintáctica `hrd_parameters( )` que se aplica a la partición de flujo de bits bajo prueba en el mensaje SEI de parámetros de HRD de partición de flujo de bits que se incluye en un mensaje SEI de agrupamiento escalable que se aplica a `TargetOp` y al `TargetPartitioningScheme` (o estará disponible a través de algunos medios externos no especificados en esta memoria descriptiva).

Dentro de la estructura sintáctica `hrd_parameters( )` seleccionada, si `BitstreamToDecode` es un flujo de bits de Tipo I, se selecciona la sub\_estructura sintáctica `sub_layer_hrd_parameters(HighestTid)` estructura sintáctica que sigue inmediatamente a la condición "`if(vcl_hrd_parameters_present_flag)`" y la variable `NalHrdModeFlag` se establece igual a 0; de lo contrario (`BitstreamToDecode` es un `Type` flujo de bits II), se selecciona la sub\_estructura sintáctica `sub_layer_hrd_parameters(HighestTid)` estructura sintáctica que sigue inmediatamente a la condición "`if(vcl_hrd_parameters_present_flag)`" (en este caso, la variable `NalHrdModeFlag` se establece igual a 0) o la condición "`if(nal_hrd_parameters_present_flag)`" (en este caso, la variable `NalHrdModeFlag` se establece igual a 1). Cuando `BitstreamToDecode` es un flujo de bits Tipo II y `NalHrdModeFlag` es igual a 0, todas las unidades NAL no VCL excepto las unidades NAL de datos de relleno y todos los elementos sintácticos `leading_zero_8bits`, `zero_byte`, `start_code_prefix_one_3bytes`, y `trailing_zero_8bits` que forman un flujo de bytes a partir el flujo de la unidad NAL (como se especifica en el Anexo B), cuando están presentes, se descartan de `BitstreamToDecode`, y el flujo de bits restante se asigna a `BitstreamToDecode`.

5. Una unidad de acceso asociada con un mensaje SEI de período de almacenamiento en memoria intermedia (presente en `BitstreamToDecode` o disponible a través de medios externos no especificados en esta memoria descriptiva) aplicable a `TargetOp` se selecciona como el punto de inicialización de HRD y se denomina unidad de acceso 0. Un mensaje SEI de período de almacenamiento en memoria intermedia aplicable está disponible a través de medios externos no especificados en esta memoria descriptiva o se selecciona a partir de la unidad de acceso 0 de la forma siguiente:

- 50

  - Si se prueba el funcionamiento de CPB específico del flujo de bits, se aplica lo siguiente:
  - Si `TargetDecLayerIdList` contiene todos los valores `nuh_layer_id` presentes en el flujo de bits bajo prueba, se selecciona un mensaje SEI de período de almacenamiento en memoria intermedia no agrupado.
  - De lo contrario, se selecciona un mensaje SEI de período de almacenamiento en memoria intermedia incluido en el mensaje SEI de agrupamiento escalable con `bitstream_subset_flag` igual a 1 y aplicable a `TargetOp`.
  - De lo contrario, se selecciona un mensaje SEI de período de almacenamiento en memoria intermedia incluido en el mensaje SEI de agrupamiento de partición de flujo de bits aplicable a la partición de flujo de bits bajo prueba.

La variable `MultiLayerCpbOperationFlag` se obtiene de la forma siguiente:

- Si el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado no está agrupado o está incluido en un mensaje SEI de agrupamiento escalable que se aplica solo al sub-flujo de bits que contiene solo la capa base, MultiLayerCpbOperationFlag se establece igual a 0.

5 - De lo contrario, MultiLayerCpbOperationFlag se establece igual a 1.

6. Para cada unidad de acceso en BitstreamToDecode comenzando desde la unidad de acceso 0, se selecciona el mensaje SEI de período de almacenamiento en memoria intermedia (presente en BitstreamToDecode o disponible a través de medios externos no especificados en esta memoria descriptiva) que está asociado con la unidad de acceso y se aplica a TargetOp, se selecciona el mensaje SEI de sincronización de imágenes (presente en BitstreamToDecode o disponible a través de medios externos no especificados en esta memoria descriptiva) que está asociado con la unidad de acceso y se aplica a TargetOp, y cuando SubPicHrdFlag es igual a 1 y sub\_pic\_cpb\_params\_in\_pic\_timing\_sei\_flag es igual a 0, se seleccionan los mensajes SEI de información de la unidad de descodificación (presentes en BitstreamToDecode o disponibles a través de medios externos no especificados en esta memoria descriptiva) que están asociados con unidades de descodificación en la unidad de acceso y se aplican a TargetOp, de la forma siguiente:

- Si se prueba el funcionamiento de CPB específico del flujo de bits, se aplica lo siguiente:

- Si TargetDecLayerIdList contiene todos los nuh\_layer\_id valores presentes en el flujo de bits bajo prueba, se seleccionan los mensajes SEI de período de almacenamiento en memoria intermedia no agrupado, sincronización de imágenes e información de la unidad de descodificación.

- De lo contrario, se seleccionan los mensajes SEI de período de almacenamiento en memoria intermedia, sincronización de imágenes e información de la unidad de descodificación incluidos en el mensaje SEI de agrupamiento escalable con bitstream\_subset\_flag igual a 1 y aplicable a TargetOp.

- De lo contrario, se seleccionan los mensajes SEI de período de almacenamiento en memoria intermedia, sincronización de imágenes e información de la unidad de descodificación incluidos en el mensaje SEI de agrupamiento de partición de flujo de bits y aplicables a la partición de flujo de bits bajo prueba.

7. Un valor de SchedSelIdx se selecciona de la forma siguiente:

- Si se prueba el funcionamiento de CPB específico del flujo de bits, el SchedSelIdx seleccionado estará en el rango de 0 a cpb\_cnt\_minus1[HighestTid], inclusive, donde cpb\_cnt\_minus1[HighestTid] se encuentra en la sub\_layer\_hrd\_parameters(HighestTid) estructura sintáctica como se seleccionó anteriormente.

- De lo contrario (se prueba el funcionamiento de CPB específico de partición de flujo de bits), se selecciona un SchedSelComblIdx para el flujo de bits bajo prueba y se utiliza para cada partición de flujo de bits bajo prueba. Se aplica lo siguiente:

- Si el vps\_vui\_bsp\_hrd\_parameters( ) la estructura sintáctica está presente en el VPS activo (o disponible a través de medios externos no especificados en esta memoria descriptiva) y contiene la hrd\_parameters( ) estructura sintáctica que se aplica a TargetOp y la partición de flujo de bits bajo prueba, el SchedSelComblIdx seleccionado estará en el rango de 0 a num\_bsp\_sched\_combinations\_minus1[TargetOutput[Dec]LayerSetIdx][TargetPartitioningSchemeldx], inclusive, y el SchedSelIdx seleccionado debe ser igual a bsp\_comb\_sched\_idx[TargetOutput[Dec]LayerSetIdx][TargetPartitioningSchemeldx][SchedSelComblIdx][j] donde j es el índice de la partición de flujo de bits bajo prueba, donde TargetPartitioningSchemeldx es el índice del TargetPartitioningScheme seleccionado que está en el rango de 1 a num partitioning schemes minus1[TargetOutputLayerSetIdx]

- De lo contrario, el SchedSelComblIdx seleccionado estará en el rango de 0 a sei\_num\_bsp\_sched\_combinations\_minus1[TargetOutput[Dec]LayerSetIdx][TargetPartitioningSchemeldx], inclusive, y el SchedSelIdx seleccionado será igual a sei\_bsp\_comb\_sched\_idx[TargetOutput[Dec]LayerSetIdx][TargetPartitioningSchemeldx][SchedSelComblIdx][j] del mensaje SEI de parámetros de HRD de la partición de flujo de bits aplicable a TargetOp donde j es el índice de la partición de flujo de bits bajo prueba.

El HRD funciona de la forma siguiente:

- El HRD se inicializa en la unidad de descodificación 0, con la CPB, cada sub-DPB de la DPB, y cada BPB estableciéndose para estar vacías (la plenitud de sub-DPB para cada sub-DPB se establece igual a 0).

- NOTA 6 - Después de la inicialización, el HRD no se inicializa de nuevo mediante los posteriores mensajes SEI de período de almacenamiento en memoria intermedia posteriores.



- 5       ▪ Para el funcionamiento de CPB específico de flujo de bits, los datos asociados con las unidades de descodificación que fluyen en la CPB de acuerdo con un programa de llegadas especificado pueden entregarse mediante el HSS. Para el funcionamiento de CPB específico de partición de flujo de bits, los datos asociados con las unidades de descodificación que fluyen en el BPB de acuerdo con un programa de llegadas especificado son entregados por un HBPS.
- 10       ▪ Cuando se usa el funcionamiento de CPB específico de partición de flujo de bits, cada partición de flujo de bits con índice j se procesa como se especifica en la cláusula C.2 con HSS reemplazado por HPBS y con SchedSelIdx igual a bsp\_comb\_sched\_idx[TargetOutput[[Dec]]LayerSetIdx][TargetPartitioningSchemeIdx][SchedSelComblIdx][j], si vps\_vui\_bsp\_hrd\_parameters( ) la estructura sintáctica está presente en el VPS activo o está disponible a través de medios externos no especificados en esta memoria descriptiva), o igual a sei\_bsp\_comb\_sched\_idx[TargetOutput[[Dec]]LayerSetIdx][TargetPartitioningSchemeIdx][SchedSelComblIdx][j] del mensaje SEI de parámetros de HRD de partición de flujo de bits aplicable a TargetOp, en caso contrario.
- 15       ▪ Los datos asociados a cada unidad de descodificación se eliminan y descodifican instantáneamente mediante el proceso de descodificación instantánea en el momento de eliminación de CPB de la unidad de descodificación.
- 20       ▪ Cada imagen descodificada se coloca en la DPB.
- 25       ▪ Se elimina una imagen descodificada de la DPB cuando ya no es necesaria para la referencia de interpredicción y ya no es necesaria para la salida.

30       **[0202]** Como se muestra arriba, cuando un HRD prueba cada partición de cada esquema de partición, basándose en una partición seleccionada de un esquema de partición particular del uno o más esquemas de partición que no contienen todas las capas del conjunto de capas de salida, se puede aplicar el HRD, un funcionamiento de memoria intermedia de imágenes codificadas (CPB) específico de partición. Por ejemplo, el HRD puede obtener, a partir de una estructura sintáctica de parámetros de programación de flujo de bits (BSP) dentro de la VUI, uno o más conjuntos de parámetros de HRD. En un ejemplo similar, el HRD puede obtener, a partir de una estructura sintáctica BSP dentro del mensaje SEI, uno o más conjuntos de parámetros de HRD. Para cada esquema de partición respectivo del uno o más esquemas de partición del conjunto de capas de salida, para cada combinación respectiva en un conjunto de una o más combinaciones de programas para el esquema de partición respectivo, para cada partición respectiva del esquema de partición respectivo, el HRD puede obtener, a partir de la estructura sintáctica BSP, un elemento sintáctico respectivo (por ejemplo, bsp\_comb\_hrd\_idx) especificando un conjunto de parámetros de HRD para la partición respectiva en la estructura sintáctica BSP y un elemento sintáctico respectivo que especifica un programa de entregas (por ejemplo, sei\_bsp\_comb\_sched\_idx).

40       **[0203]** En ejemplos correspondientes, el codificador de vídeo 20 incluye, en una estructura sintáctica BSP dentro de la VUI, uno o más conjuntos de parámetros de HRD. Para cada combinación respectiva en un conjunto de una o más combinaciones de programas para un esquema de partición respectivo, para cada partición respectiva del esquema de partición respectivo, el codificador de vídeo 20 puede incluir, en la estructura sintáctica BSP, un elemento sintáctico respectivo que especifique un conjunto de parámetros de HRD para la partición respectiva en la estructura sintáctica BSP y un elemento sintáctico respectivo que especifica un programa de entregas. En otro ejemplo, el codificador 20 de vídeo incluye, en una estructura sintáctica BSP dentro del mensaje SEI, uno o más conjuntos de parámetros de HRD. Para cada combinación respectiva en un conjunto de una o más combinaciones de programas para un esquema de partición respectivo, para cada partición respectiva del esquema de partición respectivo, el codificador de vídeo 20 incluye, en la estructura sintáctica BSP, un elemento sintáctico respectivo que especifica un conjunto de parámetros de HRD para la partición respectiva en la estructura sintáctica BSP y un elemento sintáctico respectivo que especifica un programa de entregas.

50       **[0204]** Además, en algunos ejemplos, el párrafo C.4 de MV-HEVC WD8 y SHVC WD6 se modifica de la forma siguiente:

55       La siguiente frase se elimina:  
El flujo de bits es probado por el HRD en cuanto a conformidad como se especifica en la subcláusula C.1.

60       Se agrega la siguiente frase:  
El HRD prueba el sub-flujo de bits de cada partición de cada esquema de partición en cuanto a conformidad como se especifica en la subcláusula C.1.

65       **[0205]** Como se indicó anteriormente, un dispositivo de procesamiento de vídeo puede ser capaz de determinar si un descodificador de vídeo de múltiples capas puede descodificar un flujo de bits basándose en una lista de tripletes

de perfil, grado y nivel especificada para el descodificador de múltiples capas. El siguiente texto es un ejemplo de cómo esto puede implementarse.

**G.11.3 Capacidades de descodificadores**

**[0206]** Los descodificadores que se ajusten a un conjunto de tripletes de perfil, grado y nivel indicados por  $(P_i, T_i, L_i)$  para  $i$  en el rango de 0 a  $q - 1$ , inclusive, donde  $q$  es un entero positivo, deberán ser capaces de descodificar todos los sub-flujos de bits de los conjuntos de capas de salida con al menos un esquema de partición con particiones  $p$  para las cuales se aplican todas las condiciones siguientes:

- Dejar la Partición variable  $[j]$  para  $j$  en el rango de 0 a  $p - 1$ , inclusive, donde  $p$  es menor o igual a  $q$ , denote la  $j$ -ésima partición en el esquema de partición.
- La unión de las capas que están contenidas en las particiones de la Partición  $[0]$  a la Partición  $[p-1]$ , inclusive, representa todas las capas en el conjunto de capas de salida.
- Dejar que PartitionAssignment variable  $[j]$  para  $j$  en el rango de 0 a  $p - 1$ , inclusive, se utilice para indicar la asignación de descodificación de tripletes de perfil, grado y nivel  $(P_{\text{PartitionAssignment}[j]}, T_{\text{PartitionAssignment}[j]}, L_{\text{PartitionAssignment}[j]})$  a Partición  $[j]$ . El valor de PartitionAssignment $[j]$  se obtiene de la forma siguiente:
- Dejar que UnassignedProfileTierLevelTriples sea una matriz de tripletes de perfil, grado y nivel con UnassignedProfileTierLevelTriples $[j]$  es igual  $(P_i, T_i, L_i)$  para  $i$  en el rango de 0 a  $q - 1$ , inclusive.
- Se aplica lo siguiente:

```

for( j = 0; i < p; i++)
    PartitionAssignment[ j ] = - 1
    si(hay un no vacío UnassignedProfileTierLevelTriples[ x ] tal que
        general_profile_idc de la Partición[ j ] es igual a  $P_x$  o
        general_profile_compatibility_flag[  $P_x$  ]
        es igual a 1 para Partición[ j ] y
        general_tier_flag de la Partición [ j ] representa un grado inferior
        que o igual a  $T_x$  y
        general_level_idc de la Partición[ j ] representa un nivel
        superior o igual a  $L_x$  ) {
        PartitionAssignment[ j ] = x
        Empty UnassignedProfileTierLevelTriples[ x ]
    }

```

- No habrá ninguna entrada en PartitionAssignment $[j]$  que sea igual a  $- 1$   $j$  en el rango de 0 a  $p - 1$ , inclusive.

**[0207]** La FIG. 5A es un diagrama de flujo que ilustra un ejemplo de operación del codificador de vídeo 20 de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 5A, el codificador de vídeo 20 genera un flujo de bits que comprende una representación codificada de los datos de vídeo (300). El flujo de bits comprende una pluralidad de capas. Como parte de generar el flujo de bits, el codificador de vídeo 20 incluye, en el flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición (302). Para cada esquema de partición respectivo del uno o más esquemas de partición, el esquema de partición respectivo especifica un conjunto respectivo de particiones disjuntas cuya unión forma un conjunto de capas de salida, en el que cada partición respectiva del conjunto respectivo de particiones disjuntas contiene una o más capas, el conjunto de capas de salida que consiste en todas las capas de la pluralidad de capas o un subconjunto adecuado de la pluralidad de capas.

**[0208]** En algunos ejemplos, como parte de generar el flujo de bits, para cada partición respectiva de cada esquema de partición respectivo del uno o más esquemas de partición, el codificador de vídeo 20 puede indicar, en el flujo de bits, un elemento sintáctico respectivo que indica un perfil, un grado y un nivel para la partición respectiva.

**[0209]** Además, en algunos ejemplos, el codificador de vídeo 20 puede incluir, en el flujo de bits, elementos sintácticos que indican una pluralidad de conjuntos de capas de salida, con la pluralidad de conjuntos de capas de salida que incluye el primer conjunto de capas de salida. Para cada conjunto de capas de salida respectivo de la pluralidad de conjuntos de capas de salida, el codificador de vídeo 20 puede incluir, en el flujo de bits, elementos sintácticos que indican uno o más esquemas de partición para el conjunto de capas de salida respectivo.

**[0210]** En algunos ejemplos, como parte de señalar el uno o más elementos sintácticos que indican el uno o más esquemas de partición, para cada esquema de partición respectivo del uno o más esquemas de partición, para cada partición respectiva del esquema de partición respectivo, para cada capa respectiva del conjunto de capas de salida,

el codificador de vídeo 20 puede incluir, en el flujo de bits, un elemento sintáctico que indica si la capa respectiva está incluida en la partición respectiva.

5 **[0211]** Además, en el ejemplo de la FIG. 5A, el codificador de vídeo 20 emite el flujo de bits (304). Por ejemplo, el codificador de vídeo 20 puede emitir el flujo de bits a una interfaz de salida, tal como la interfaz de salida 22. En algunos ejemplos, el codificador de vídeo 20 puede emitir el flujo de bits a uno o más medios de almacenamiento de datos legibles por ordenador para su almacenamiento.

10 **[0212]** La FIG. 5B es un diagrama de flujo que ilustra un ejemplo de funcionamiento del descodificador de vídeo 30, de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 5B, el descodificador de vídeo 30 descodifica un flujo de bits que comprende una representación codificada de los datos de vídeo, comprendiendo el flujo de bits una pluralidad de capas (320). Como parte de la descodificación del flujo de bits, el descodificador de vídeo 30 obtiene, a partir del flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición (322). Para cada esquema de partición respectivo del uno o más esquemas de partición, el esquema de partición respectivo especifica un conjunto respectivo de particiones disjuntas cuya unión forma un conjunto de capas de salida. Cada partición respectiva del conjunto respectivo de particiones disjuntas contiene una o más de las capas. El conjunto de capas de salida puede consistir en todas las capas de la pluralidad de capas o un subconjunto adecuado de la pluralidad de capas. En algunos ejemplos, como parte de obtener el uno o más elementos sintácticos, para cada esquema de partición respectivo del uno o más esquemas de partición, para cada partición respectiva del esquema de partición respectivo, para cada capa respectiva del conjunto de capas de salida, el descodificador de vídeo 30 puede obtener, a partir del flujo de bits, un elemento sintáctico que indica si la capa respectiva está incluida en la partición respectiva.

25 **[0213]** Además, como parte de descodificar el flujo de bits, el descodificador de vídeo 30 descodifica cada una de las particiones de un esquema de partición particular usando diferentes núcleos de procesamiento en una pluralidad de núcleos de hardware, siendo el esquema de partición particular uno de los uno o más esquemas de partición (324). En algunos ejemplos, como parte de descodificar el flujo de bits, el descodificador de vídeo 30 puede, para cada partición respectiva de cada esquema de partición respectivo del uno o más esquemas de partición, obtener, a partir del flujo de bits, un elemento sintáctico respectivo que indique la información de perfil, grado y nivel (PTL) respectiva para la partición respectiva.

35 **[0214]** En algunos ejemplos, el descodificador de vídeo 30 puede determinar, basándose en la información de PTL para particiones de los esquemas de partición, el esquema de partición particular del uno o más esquemas de partición. Además, en dichos ejemplos, el descodificador de vídeo 30 puede descodificar cada partición respectiva del esquema de partición particular usando uno de los respectivos núcleos de hardware, en el que cada uno de los respectivos núcleos de hardware implementa un descodificador de capa única respectivo. En algunos ejemplos, al menos un primer núcleo de hardware de la pluralidad de núcleos de hardware soporta un perfil, grado o nivel diferente de un segundo núcleo de hardware de la pluralidad de núcleos de hardware. En otros ejemplos, al menos un primer núcleo de hardware de la pluralidad de núcleos de hardware y un segundo núcleo de hardware diferente de la pluralidad de núcleos de hardware soportan el mismo conjunto de perfiles, grados y niveles.

45 **[0215]** En algunos ejemplos, cuando el descodificador de vídeo 30 descodifica el flujo de bits, el descodificador de vídeo 30 obtiene, a partir del flujo de bits, elementos sintácticos que indican una pluralidad de conjuntos de capas de salida, con la pluralidad de conjuntos de capas de salida que incluye el primer conjunto de capas de salida. En dichos ejemplos, para cada conjunto de capas de salida respectivo de la pluralidad de conjuntos de capas de salida, el descodificador de vídeo 30 puede obtener, a partir del flujo de bits, elementos sintácticos que indican uno o más esquemas de partición para el conjunto de capas de salida respectivo.

50 **[0216]** La FIG. 6A es un diagrama de flujo que ilustra un ejemplo de operación del codificador de vídeo 20 de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 6A, el codificador de vídeo 20 genera un flujo de bits que comprende una representación codificada de los datos de vídeo (350). El flujo de bits comprende una pluralidad de capas. Como parte de generar el flujo de bits, el codificador de vídeo 20 incluye, en el flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición para un conjunto de capas de salida (352). En el ejemplo de la FIG. 6A, el conjunto de capas de salida consiste en todas las capas de la pluralidad de capas o un subconjunto adecuado de la pluralidad de capas. Además, para cada esquema de partición respectivo del uno o más esquemas de partición, el esquema de partición respectivo especifica una división del conjunto de capas de salida en una o más particiones, en el que cada capa en el conjunto de capas de salida pertenece exactamente a una partición en el respectivo esquema de partición. Además, como parte de generar el flujo de bits, para cada esquema de partición respectivo del uno o más esquemas de partición, el codificador de vídeo 20 incluye, en un mensaje SEI en el flujo de bits o VUI en un VPS en el flujo de bits, un conjunto respectivo de elementos sintácticos (p.ej, bsp\_comb\_hrd\_idx elementos sintácticos) que indican conjuntos de parámetros de HRD para el esquema de partición respectivo (354). El codificador de vídeo 20 emite el flujo de bits (356). Por ejemplo, el codificador de vídeo 20 puede emitir el flujo de bits a una interfaz de salida, tal como la interfaz de salida 22. En algunos ejemplos, el codificador de vídeo 20 puede emitir el flujo de bits a uno o más medios de almacenamiento de datos legibles por ordenador para su almacenamiento.

[0217] En algunos ejemplos, como parte de incluir el conjunto respectivo de elementos sintácticos que indican los conjuntos de parámetros de HRD para cada uno de los uno o más esquemas de partición, el codificador de vídeo 20 puede incluir, en una estructura sintáctica de parámetros de programación de flujo de bits (BSP) dentro de la VUI, uno o más conjuntos de parámetros de HRD. Además, en dichos ejemplos, para cada combinación respectiva en un conjunto de una o más combinaciones de programas para el esquema de partición respectivo, para cada partición respectiva del esquema de partición respectivo, el codificador de vídeo 20 puede incluir, en la estructura sintáctica BSP, un elemento sintáctico respectivo que especifica un conjunto de parámetros de HRD para la partición respectiva en la estructura sintáctica BSP y un elemento sintáctico respectivo que especifica un programa de entregas.

[0218] En algunos ejemplos, como parte de incluir el conjunto respectivo de elementos sintácticos que indican los conjuntos de parámetros de HRD para cada uno de los uno o más esquemas de partición, el codificador de vídeo 20 puede incluir, en una estructura sintáctica de parámetros de programación de flujo de bits (BSP) dentro del mensaje SEI, uno o más conjuntos de parámetros de HRD. Además, en dichos ejemplos, para cada combinación respectiva en un conjunto de una o más combinaciones de programas para el esquema de partición respectivo, para cada partición respectiva del esquema de partición respectivo, el codificador de vídeo 20 puede incluir, en la estructura sintáctica BSP, un elemento sintáctico respectivo que especifica un conjunto de parámetros de HRD para la partición respectiva en la estructura sintáctica BSP y un elemento sintáctico respectivo que especifica un programa de entregas.

[0219] La FIG. 6B es un diagrama de flujo que ilustra un ejemplo de funcionamiento del descodificador de vídeo 30, de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 6B, el descodificador de vídeo 30 obtiene, a partir de un flujo de bits que comprende una representación codificada de datos de vídeo, uno o más elementos sintácticos que indican uno o más esquemas de partición para un conjunto de capas de salida (370). En el ejemplo de la FIG. 6B, el conjunto de capas de salida consiste en todas las capas de la pluralidad de capas o un subconjunto adecuado de la pluralidad de capas. Para cada esquema de partición respectivo del uno o más esquemas de partición, el esquema de partición respectivo especifica una división del conjunto de capas de salida en una o más particiones, en el que cada capa en el conjunto de capas de salida pertenece exactamente a una partición en el esquema de partición respectivo.

[0220] Además, en el ejemplo de la FIG. 6B, para cada esquema de partición respectivo del uno o más esquemas de partición, el descodificador de vídeo 30 obtiene, a partir de un mensaje SEI en el flujo de bits o VUI en un VPS en el flujo de bits, un conjunto respectivo de elementos sintácticos (por ejemplo, `bsp_comb_hrd_idx` elementos sintácticos) que indican conjuntos de parámetros de HRD para el esquema de partición respectivo (372).

[0221] En algunos ejemplos, como parte de obtener un conjunto respectivo de elementos sintácticos que indican conjuntos de parámetros de HRD para el esquema de partición respectivo, el descodificador de vídeo 30 obtiene, a partir de una estructura sintáctica de parámetros de programación de flujo de bits (BSP) dentro de la VUI, uno o más conjuntos de parámetros de HRD. En dichos ejemplos, para cada combinación respectiva en un conjunto de una o más combinaciones de programas para el esquema de partición respectivo, para cada partición respectiva del esquema de partición respectivo, el descodificador de vídeo 30 puede obtener, a partir de la estructura sintáctica BSP, un elemento sintáctico respectivo que especifica un conjunto de parámetros de HRD para la partición respectiva en la estructura sintáctica BSP y un elemento sintáctico respectivo que especifica un programa de entregas.

[0222] En algunos ejemplos, como parte de obtener un conjunto respectivo de elementos sintácticos que indican conjuntos de parámetros de HRD para el esquema de partición respectivo, el descodificador de vídeo 30 obtiene, a partir de una estructura sintáctica de parámetros de programación de flujo de bits (BSP) dentro del mensaje SEI, uno o más conjuntos de parámetros de HRD. En dichos ejemplos, para cada combinación respectiva en un conjunto de una o más combinaciones de programas para el esquema de partición respectivo, para cada partición respectiva del esquema de partición respectivo, el descodificador de vídeo 30 puede obtener, a partir de la estructura sintáctica BSP, un elemento sintáctico respectivo que especifica un conjunto de parámetros de HRD para la partición respectiva en la estructura sintáctica BSP y un elemento sintáctico respectivo que especifica un programa de entregas.

[0223] Para cada partición respectiva del esquema de partición respectivo, el descodificador de vídeo 30 prueba, basándose en los parámetros de HRD para el esquema de partición respectivo, la partición respectiva en cuanto a conformidad (374). En algunos ejemplos, como parte de probar los sub-flujos de bits, el descodificador de vídeo 30 aplica, basándose en una partición seleccionada de un esquema de partición particular del uno o más esquemas de partición que no contienen todas las capas del conjunto de capas de salida, una CPB específica de la partición operación. En algunos ejemplos, probar la conformidad de una partición particular supone que las imágenes descodificadas y otra información necesaria de las capas de referencia directa e indirecta de las capas en la partición, y que no están contenidas en la partición particular, están disponibles en el momento de descodificar las imágenes de las capas en la partición en particular.

**[0224]** La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un descodificador de vídeo 30, de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 7, el descodificador de vídeo 30 determina, basándose en una lista de entradas triples, si el descodificador de vídeo 30 (es decir, un descodificador de vídeo de múltiples capas) es capaz de descodificar un flujo de bits que comprende una representación codificada de datos de vídeo de múltiples capas (400). En algunos ejemplos, el número de entradas triples en la lista es menor o igual que una cantidad de descodificadores de capa única en los datos de vídeo de múltiples capas. En algunos ejemplos, un único núcleo de hardware del descodificador de vídeo 30 puede descodificar múltiples capas. Cada entrada triple respectiva en la lista de entradas triples indica un perfil, un grado y un nivel para un descodificador de capa única respectivo en el descodificador de vídeo de múltiples capas. Basándose en el descodificador de vídeo de múltiples capas que es capaz de descodificar el flujo de bits, el descodificador de vídeo 30 descodifica el flujo de bits (402).

**[0225]** En algunos ejemplos, el descodificador de vídeo 30 obtiene, a partir del flujo de bits, uno o más elementos sintácticos que indican uno o más esquemas de partición. Para cada esquema de partición respectivo del uno o más esquemas de partición, el esquema de partición respectivo especifica un conjunto respectivo de particiones disjuntas cuya unión forma un conjunto de capas de salida. Cada partición respectiva del conjunto respectivo de particiones disjuntas contiene una o más de las capas, consistiendo el conjunto de capas de salida en todas las capas de la pluralidad de capas o un subconjunto adecuado de la pluralidad de capas. Como parte de determinar si el descodificador de vídeo de múltiples capas es capaz de descodificar el flujo de bits, el descodificador de vídeo 30 puede determinar, basándose en que se satisfaga una condición, que el descodificador de vídeo de múltiples capas es capaz de descodificar el flujo de bits. En dichos ejemplos, la condición se satisface basándose en que exista un esquema de partición adecuado en el uno o más esquemas de partición. Además, en dichos ejemplos, para cada partición respectiva del esquema de partición adecuado, se puede asignar uno de los descodificadores de capa única para descodificar la partición respectiva. En algunos ejemplos, como parte de determinar que el descodificador de vídeo de múltiples capas es capaz de descodificar el flujo de bits, para cada partición respectiva del esquema de partición adecuado, basándose en que se cumplan las siguientes condiciones, el descodificador de vídeo de múltiples capas puede determinar que la partición respectiva se puede asignar a uno de los descodificadores de capa única: un nivel asociado con el descodificador de capa única es mayor o igual que un nivel señalado para la partición respectiva, un grado asociado con el descodificador de capa única es mayor o igual que un grado señalado para la partición respectiva, y un perfil asociado con la partición respectiva es tal que el descodificador de capa única soporta la descodificación del perfil.

**[0226]** En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementan en software, las funciones pueden almacenarse en o transmitirse a través de, como una o más instrucciones o código, en un medio legible por ordenador o ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos o unos medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) unos medios de almacenamiento tangibles legibles por ordenador que son no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible a los que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

**[0227]** A modo de ejemplo, y no de manera limitativa, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe debidamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, tal como se utiliza en el presente documento, incluye un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, donde algunos discos habitualmente emiten datos magnéticamente, mientras que otros discos emiten datos ópticamente con láseres. Las combinaciones de los anteriores deberían incluirse también dentro del alcance de los medios legibles por ordenador.

**[0228]** Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables por campo (FPGA) u otros circuitos lógicos integrados o discretos

5 equivalentes. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

10 **[0229]** Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en la presente divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por medio de un grupo de unidades de hardware interoperativas, incluidos uno o más procesadores como los descritos anteriormente, conjuntamente con software y/o firmware adecuados.

15 **[0230]** Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

**REIVINDICACIONES**

1. Un procedimiento de descodificación de datos de vídeo de múltiples capas, comprendiendo el procedimiento:

5 descodificar un flujo de bits que comprende una representación codificada de los datos de vídeo de múltiples capas, en el que descodificar el flujo de bits comprende obtener, a partir del flujo de bits, elementos sintácticos que indican una pluralidad de esquemas de partición diferentes, en el que:

10 los datos de vídeo de múltiples capas comprenden una pluralidad de capas; las unidades de capa de acceso a la red (NAL) del flujo de bits que tienen diferentes identificadores de capa pertenecen a diferentes capas de la pluralidad de capas, y

15 para cada esquema de partición respectivo de la pluralidad de esquemas de partición, el esquema de partición respectivo especifica un conjunto respectivo de particiones disjuntas cuya unión forma un conjunto de capas de salida, en el que cada partición respectiva del conjunto respectivo de particiones disjuntas contiene una o más de las capas;

20 determinar si un descodificador de vídeo de múltiples capas es capaz de descodificar el conjunto de capas de salida, en el que el descodificador de vídeo de múltiples capas se ajusta a una lista de entradas triples, cada entrada triple respectiva en la lista de entradas triples indica un perfil, un grado y un nivel para un descodificador de capa única respectivo del descodificador de vídeo de múltiples capas, y se requieren descodificadores que se ajusten a la lista de entradas triples para poder descodificar el conjunto de capas de salida si la pluralidad de esquemas de partición incluye un esquema de partición adecuado en el que, para cada partición respectiva del esquema de partición, la información de perfil, grado y nivel señalada para dicha partición respectiva indica que un descodificador de capa única del descodificador de vídeo de múltiples capas puede asignarse exclusivamente para descodificar la partición respectiva; y

25 basándose en el descodificador de vídeo de múltiples capas que es capaz de descodificar el flujo de bits, descodificar, mediante el descodificador de vídeo de múltiples capas, el flujo de bits.

30 2. El procedimiento según la reivindicación 1, en el que el número de entradas triples en la lista es menor o igual a un número de descodificadores de capa única en el descodificador de vídeo de múltiples capas.

35 3. El procedimiento según la reivindicación 1, en el que determinar que el descodificador de vídeo de múltiples capas es capaz de descodificar el flujo de bits comprende:

para cada partición respectiva del esquema de partición adecuado:

40 basándose en que se cumplan las siguientes condiciones, determinar que la partición respectiva se puede asignar a un descodificador de capa única del descodificador de vídeo de múltiples capas:

- 45 - un nivel asociado con el descodificador de capa única es mayor o igual que un nivel señalado para la partición respectiva,
- un grado asociado con el descodificador de capa única es mayor o igual a un grado señalado para la partición respectiva, y
- 50 - un perfil asociado con la partición respectiva es tal que el descodificador de capa única soporta la descodificación del perfil.

4. Un dispositivo para descodificar datos de vídeo de múltiples capas, comprendiendo el dispositivo:

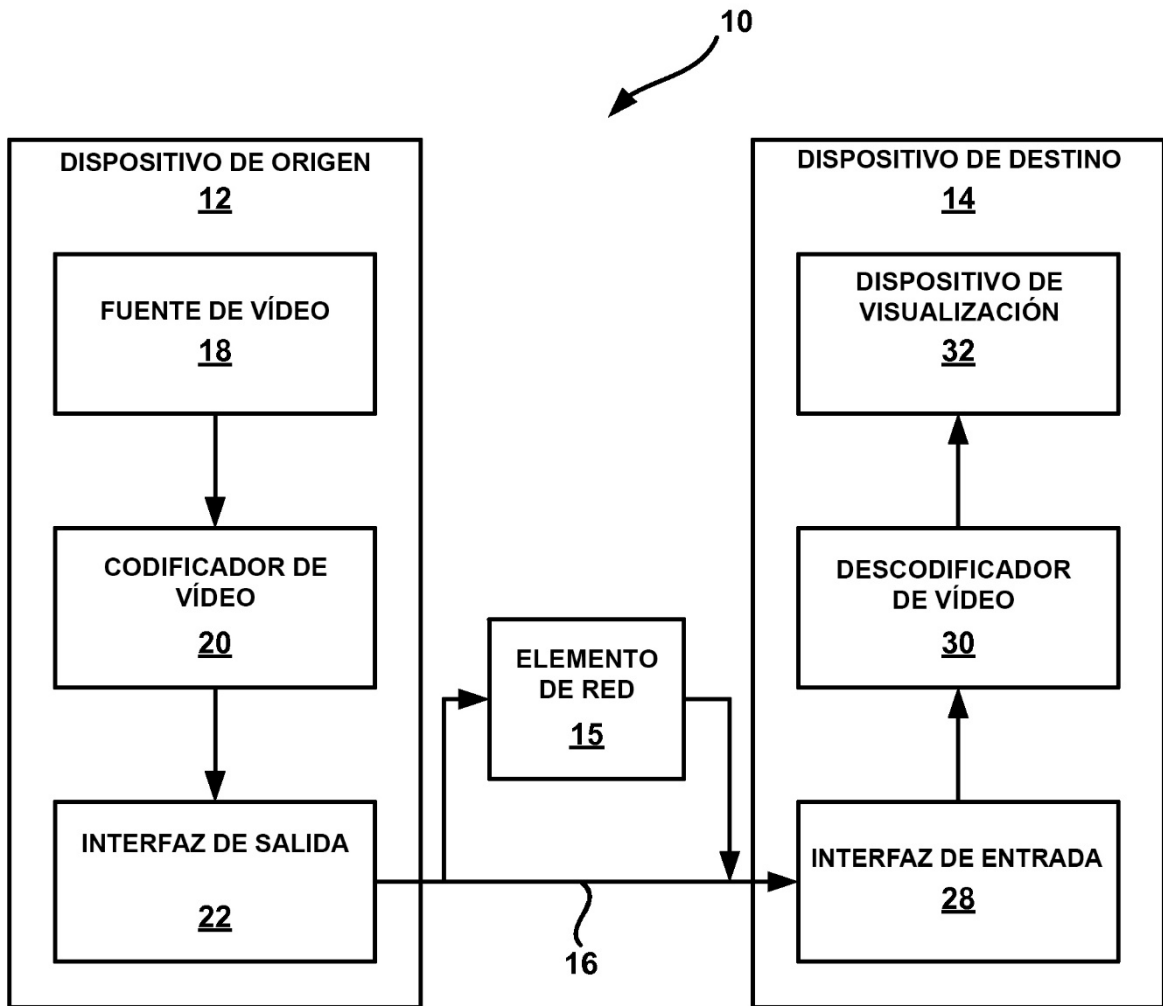
55 medios para descodificar un flujo de bits que comprende una representación codificada de datos de vídeo de múltiples capas, con los medios para descodificar el flujo de bits que comprenden medios para obtener, a partir del flujo de bits, elementos sintácticos que indican una pluralidad de esquemas de partición diferentes, en el que:

60 los datos de vídeo de múltiples capas comprenden una pluralidad de capas; las unidades de capa de acceso a la red (NAL) del flujo de bits que tienen diferentes identificadores de capa pertenecen a diferentes capas de la pluralidad de capas, y

65 para cada esquema de partición respectivo de la pluralidad de esquemas de partición, el esquema de partición respectivo especifica un conjunto respectivo de particiones disjuntas cuya unión forma un conjunto de capas de salida, en el que cada partición respectiva del conjunto respectivo de particiones disjuntas contiene una o más de las capas;

- 5 medios para determinar, basándose en una lista de entradas triples, si un descodificador de vídeo de múltiples capas es capaz de descodificar el conjunto de capas de salida, en el que el descodificador de vídeo de múltiples capas se ajusta a una lista de entradas triples, un flujo de bits que comprende una representación codificada de los datos de vídeo de múltiples capas, en el que cada entrada triple respectiva en la lista de entradas triples indica un perfil, un grado y un nivel para un descodificador de capa única respectivo del descodificador de vídeo de múltiples capas, y se requieren descodificadores que se ajusten a la lista de entradas triples para poder descodificar el conjunto de capas de salida si la pluralidad de esquemas de partición incluye un esquema de partición adecuado en el que, para cada partición respectiva del esquema de partición, la información de perfil, grado y nivel señalada para dicha partición respectiva indica que se puede asignar un descodificador de capa única del descodificador de vídeo de múltiples capas exclusivamente para descodificar la partición respectiva; y
- 10 medios para descodificar, basándose en el descodificador de vídeo de múltiples capas del dispositivo que es capaz de descodificar el flujo de bits, el flujo de bits.
- 15
5. El dispositivo de la reivindicación 4, en el que el número de entradas triples en la lista es menor o igual que un número de descodificadores de capa única en el dispositivo descodificador de vídeo de múltiples capas.
- 20
6. El dispositivo de la reivindicación 4, en el que los medios para determinar si el descodificador de vídeo de múltiples capas es capaz de descodificar el flujo de bits comprende:
- para cada partición respectiva del esquema de partición adecuado:
- 25 medios para determinar, basándose en que se cumplan las siguientes condiciones, que la partición respectiva se pueda asignar a un descodificador de capa única del descodificador de vídeo de múltiples capas:
- 30 - un nivel asociado con el descodificador de capa única es mayor o igual que un nivel señalado para la partición respectiva,
  - un grado asociado con el descodificador de capa única es mayor o igual a un grado señalado para la partición respectiva, y
  - un perfil asociado con la partición respectiva es tal que el descodificador de capa única soporta la descodificación del perfil.
- 35
7. Un medio de almacenamiento de datos legible por ordenador que comprende instrucciones almacenadas en el mismo que, al ejecutarse, hacen que un descodificador de vídeo de múltiples capa emprenda el procedimiento según cualquiera de las reivindicaciones 1 a 3.





**FIG. 1**



**FIG. 2**

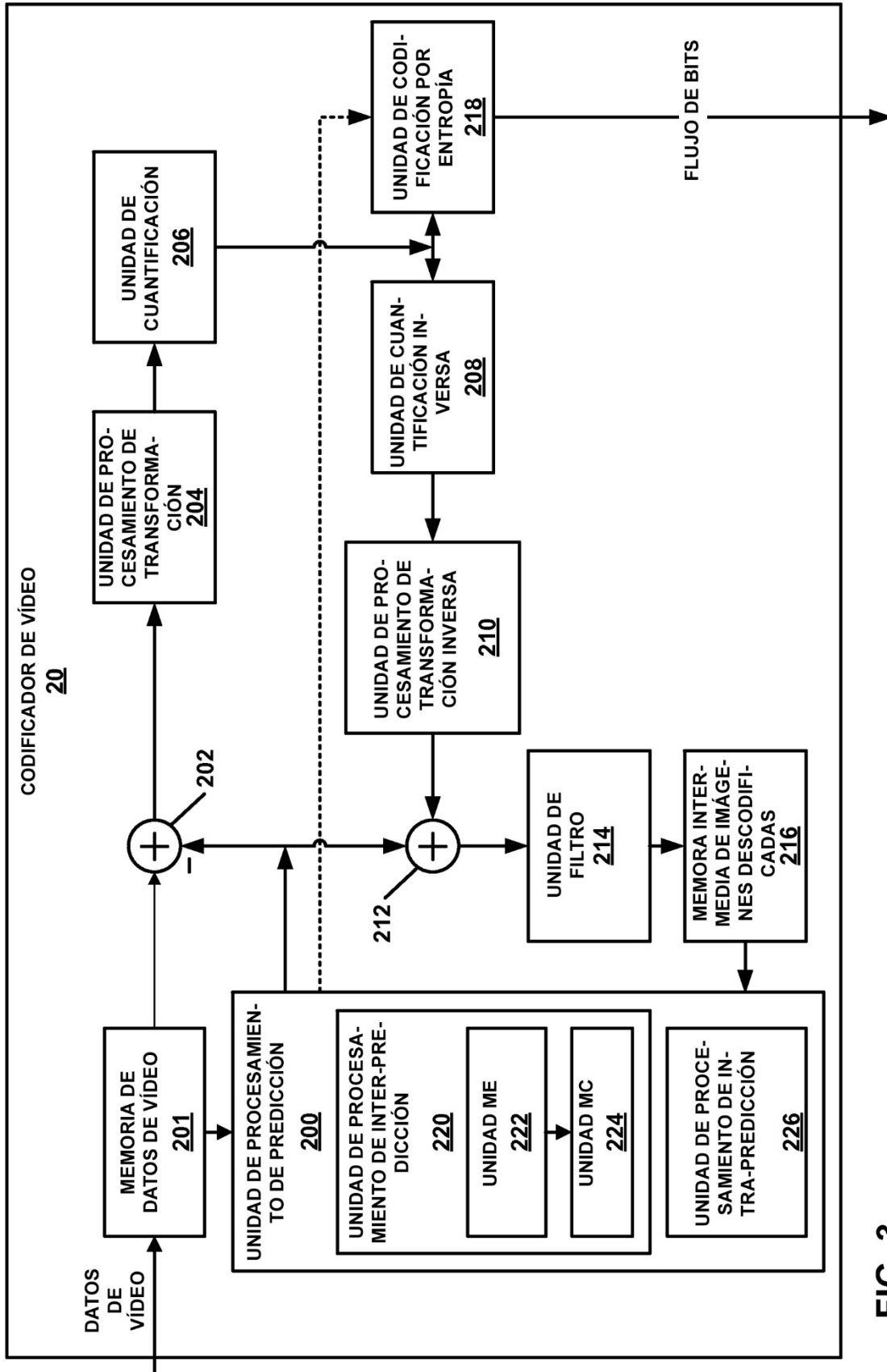


FIG. 3

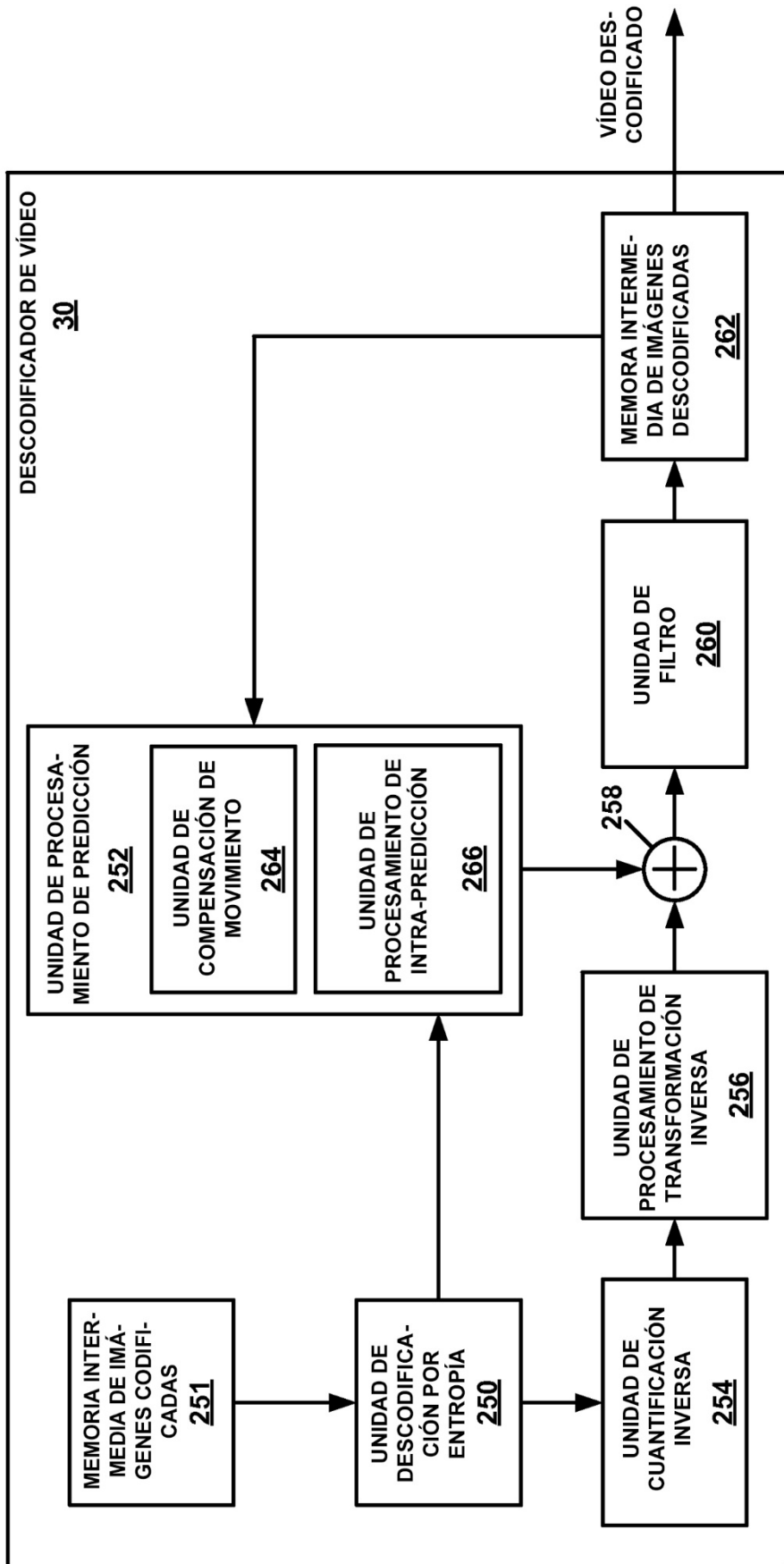
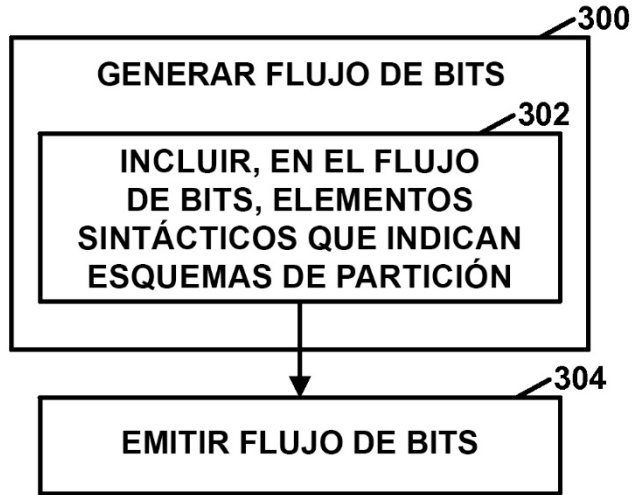
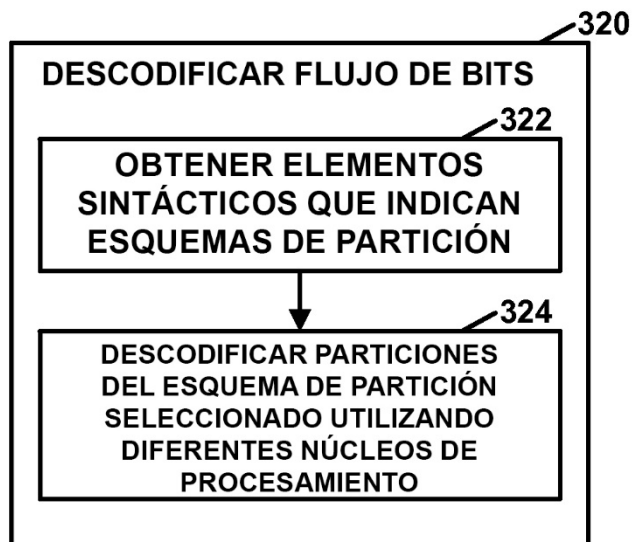


FIG. 4



**FIG. 5A**



**FIG. 5B**

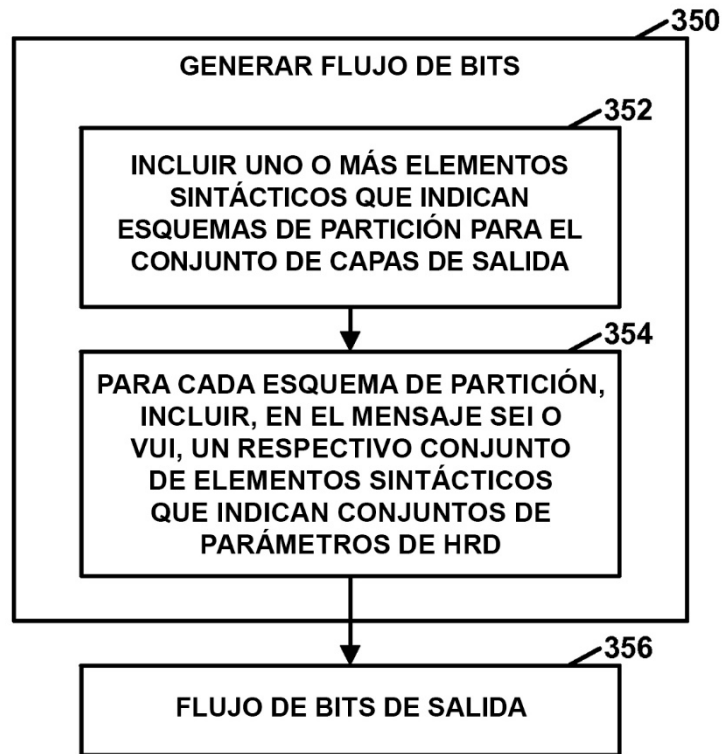


FIG. 6A

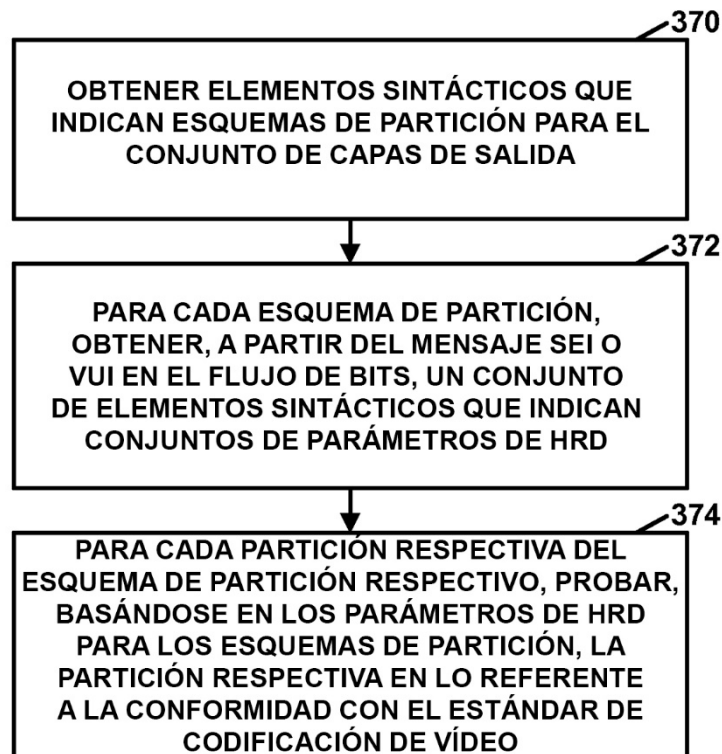
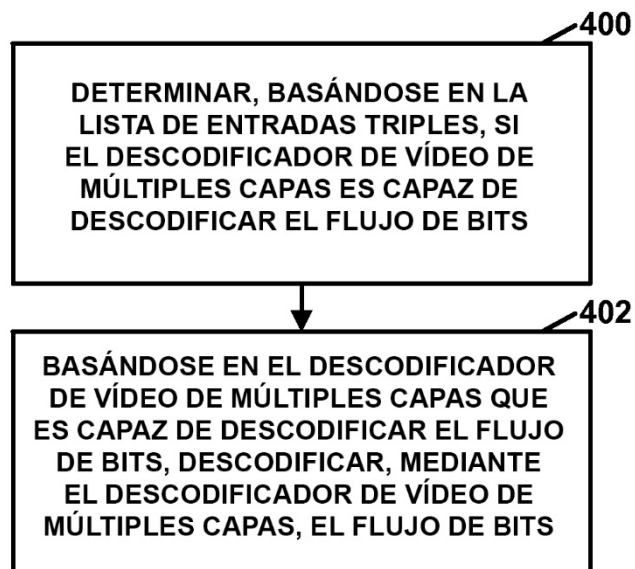


FIG. 6B



**FIG. 7**