

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 758 503**

51 Int. Cl.:

**H04N 19/70** (2014.01)

**H04N 19/149** (2014.01)

**H04N 19/152** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **18.09.2013 PCT/US2013/060412**

87 Fecha y número de publicación internacional: **27.03.2014 WO14047183**

96 Fecha de presentación y número de la solicitud europea: **18.09.2013 E 13767240 (8)**

97 Fecha y número de publicación de la concesión europea: **21.08.2019 EP 2898680**

54 Título: **Parámetros de descodificador de referencia hipotético en codificación de vídeo**

30 Prioridad:

**24.09.2012 US 201261705102 P**  
**14.06.2013 US 201313918041**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**05.05.2020**

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)**  
**5775 Morehouse Drive**  
**San Diego, CA 92121-1714, US**

72 Inventor/es:

**WANG, YE-KUI**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

**ES 2 758 503 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

Parámetros de descodificador de referencia hipotético en codificación de vídeo

5 **CAMPO TÉCNICO**

**[0001]** Esta divulgación se refiere a la codificación y descodificación de vídeo (es decir, a la codificación y/o descodificación de datos de vídeo).

10 **ANTECEDENTES**

**[0002]** Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas electrónicas, 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 digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, codificación de vídeo avanzada (AVC), la norma de codificación de vídeo de alta eficacia (HEVC) actualmente en desarrollo y las ampliaciones de dichas normas. 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 predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un segmento de vídeo (es decir, una trama de vídeo o una parte de una trama de vídeo) se puede dividir en bloques de vídeo. Los bloques de vídeo de un segmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a unas muestras de referencia de bloques vecinos de la misma imagen. Los bloques de vídeo de un segmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a unas muestras de referencia de bloques vecinos de la misma imagen, o predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes se pueden denominar tramas y las imágenes de referencia se pueden denominar tramas de referencia.

**[0004]** La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va 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 intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales se pueden transformar desde el dominio del píxel hasta un dominio de transformada, dando como resultado coeficientes residuales, que a continuación se pueden cuantificar. Los coeficientes cuantificados, dispuestos inicialmente en una matriz bidimensional, se pueden explorar a fin de generar un vector unidimensional de coeficientes, y se puede aplicar codificación de entropía para lograr incluso más compresión.

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

55 **SUMARIO**

**[0006]** En general, esta divulgación describe la señalización y la selección de parámetros de descodificador de referencia hipotético (HRD) en la codificación de vídeo. Más específicamente, un dispositivo informático puede seleccionar, de entre un conjunto de parámetros de descodificador de referencia hipotético (HRD) en un conjunto de parámetros de vídeo (VPS) y un conjunto de parámetros HRD en un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros HRD aplicables a un punto de funcionamiento particular de un flujo de bits. El dispositivo informático realiza, basándose al menos en parte en el conjunto de parámetros HRD aplicables al punto de funcionamiento particular, una operación HRD en un subconjunto de flujo de bits asociado al punto de funcionamiento particular.

**[0007]** La presente invención está definida por las reivindicaciones adjuntas. Las reivindicaciones dependientes definen modos de realización ventajosos de la misma. Cualquier ejemplo y modo de realización de la descripción

que no esté dentro del alcance de las reivindicaciones no forma parte de la invención y solo se proporciona con propósitos ilustrativos.

**BREVE DESCRIPCIÓN DE LOS DIBUJOS**

- 5 **[0008]**  
 La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación de vídeo que puede utilizar las técnicas descritas en esta divulgación.
- 10 La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar las técnicas descritas en esta divulgación.
- 15 La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo que puede implementar las técnicas descritas en esta divulgación.
- La FIG. 4 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un dispositivo, de acuerdo con una o más técnicas de esta divulgación.
- 20 La FIG. 5 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un dispositivo, de acuerdo con una o más técnicas de esta divulgación.
- La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de descodificador de referencia hipotético (HRD), de acuerdo con una o más técnicas de esta divulgación.

**DESCRIPCIÓN DETALLADA**

- 30 **[0009]** Un codificador de vídeo puede generar un flujo de bits que incluye datos de vídeo codificados. El flujo de bits puede comprender una serie de unidades de capa de abstracción de red (NAL). Las unidades NAL del flujo de bits pueden incluir unidades NAL de capa de codificación de vídeo (VCL) y unidades NAL no VCL. Las unidades NAL VCL pueden incluir segmentos codificados de imágenes. Una unidad NAL no VCL puede incluir un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), información de mejora suplementaria (SEI) u otros tipos de datos. Un VPS es una estructura sintáctica que puede contener elementos sintácticos que se aplican a cero o más secuencias de vídeo codificadas completas. Un SPS es una estructura sintáctica que puede contener elementos sintácticos que se aplican a cero o más secuencias de vídeo codificadas completas. Un solo VPS puede aplicarse a múltiples SPS. Un PPS es una estructura sintáctica que puede contener elementos sintácticos que se aplican a cero o más imágenes codificadas completas. Un solo SPS puede aplicarse a múltiples PPS. Diversos aspectos del VPS, SPS y PPS se pueden formar, en general, según lo definido por la norma HEVC.
- 40 **[0010]** Un dispositivo, tal como un dispositivo de red de distribución de contenido (CDN), un elemento de red sensible a los medios (MANE), o un descodificador de vídeo, puede extraer un subflujo de bits del flujo de bits. El dispositivo puede realizar el proceso de extracción de subflujo de bits eliminando determinadas unidades NAL del flujo de bits. El subflujo de bits resultante incluye las unidades NAL restantes, no eliminadas, del flujo de bits. Como ejemplos, los datos de vídeo descodificados del subflujo de bits pueden tener una velocidad de trama menor y/o pueden representar menos vistas que el flujo de bits original.
- 45 **[0011]** Las normas de codificación de vídeo pueden incluir diversas características para admitir el proceso de extracción de subflujo de bits. Por ejemplo, los datos de vídeo del flujo de bits se pueden dividir en un conjunto de capas. Para cada una de las capas, los datos en una capa inferior pueden descodificarse sin referencia a los datos en cualquier capa superior. Una unidad NAL individual solo encapsula datos de una única capa. Por tanto, las unidades NAL que encapsulan datos de la capa restante más alta del flujo de bits pueden eliminarse del flujo de bits sin afectar a la capacidad de descodificación de los datos en las capas inferiores restantes del flujo de bits. En la codificación de vídeo escalable (SVC), las capas superiores pueden incluir datos de mejora que mejoran la calidad de las imágenes en capas inferiores (escalabilidad de calidad), amplían el formato espacial de las imágenes en capas inferiores (escalabilidad espacial) o aumentan la velocidad temporal de las imágenes en capas inferiores (escalabilidad temporal). En la codificación de múltiples vistas (MVC) y la codificación de vídeo tridimensional (3DV), las capas superiores pueden incluir vistas adicionales.
- 50 **[0012]** Las unidades NAL pueden incluir cabeceras y cargas. Las cabeceras de las unidades NAL incluyen elementos sintácticos nuh\_reserved\_zero\_6bits. El elemento sintáctico nuh\_reserved\_zero\_6bits de una unidad NAL es igual a 0 si la unidad NAL está relacionada con una capa base en la codificación de múltiples vistas, la codificación 3DV o SVC. Los datos en una capa de 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 está relacionada con una capa base en la codificación de múltiples vistas, 3DV o SVC, el elemento sintáctico nuh\_reserved\_zero\_6bits puede tener un valor distinto de cero. Específicamente, si una unidad NAL no está relacionada con una capa base en la codificación de
- 55
- 60
- 65

múltiples vistas, 3DV o SVC, el elemento sintáctico `nuh_reserved_zero_6bits` de la unidad NAL especifica un identificador de capa de la unidad NAL.

**[0013]** Además, algunas imágenes dentro de una capa se pueden descodificar sin referencia a otras imágenes dentro de la misma capa. Por lo tanto, las unidades NAL que encapsulan datos de ciertas imágenes de una capa pueden eliminarse del flujo de bits sin afectar a la capacidad de descodificación de otras imágenes en la capa. Por ejemplo, las imágenes con valores pares de recuento de orden de imágenes (POC) pueden descodificarse sin referencia a imágenes con valores POC impares. La eliminación de unidades NAL que encapsulan datos de dichas imágenes puede reducir la velocidad de trama del flujo de bits. Un subconjunto de imágenes dentro de una capa que puede descodificarse sin referencia a otras imágenes dentro de la capa se puede denominar en el presente documento una subcapa.

**[0014]** Las unidades NAL pueden incluir elementos sintácticos `temporal_id`. El elemento sintáctico `temporal_id` de una unidad NAL especifica un identificador temporal de la unidad NAL. 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.

**[0015]** Cada punto de funcionamiento de un flujo de bits está asociado a un conjunto de identificadores de capa (es decir, un conjunto de valores `nuh_reserved_zero_6bits`) y un identificador temporal. El conjunto de identificadores de capa se puede denotar como `OpLayerIdSet` y el identificador temporal se puede denotar como `TemporalID`. Si el identificador de capa de una unidad NAL se encuentra en el conjunto de identificadores de capa de un punto de funcionamiento y el identificador temporal de la unidad NAL es menor que o igual al identificador temporal del punto de funcionamiento, la unidad NAL está asociada al punto de funcionamiento. Una representación de punto de funcionamiento es un subconjunto de flujo de bits que está asociado a un punto de funcionamiento. La representación de punto de funcionamiento puede incluir cada unidad NAL que esté asociada al punto de funcionamiento. La representación de punto de funcionamiento no incluye las unidades NAL VCL que no están asociadas al punto de funcionamiento.

**[0016]** Una fuente externa puede especificar un conjunto de identificadores de capa objetivo para un punto de funcionamiento. Por ejemplo, un dispositivo, tal como un dispositivo CDN o un MANE, puede especificar el conjunto de identificadores de capa objetivo. En este ejemplo, el dispositivo puede usar el conjunto de identificadores de capa objetivo para identificar un punto de funcionamiento. El dispositivo puede entonces extraer la representación de punto de funcionamiento para el punto de funcionamiento y reenviar la representación de punto de funcionamiento, en lugar del flujo de bits original, a un dispositivo cliente. La extracción y reenvío de la representación de punto de funcionamiento al dispositivo cliente puede reducir la velocidad de transferencia de bits del flujo de bits.

**[0017]** Además, las normas de codificación de vídeo especifican modelos de almacenamiento de vídeo en memoria temporal. Un modelo de almacenamiento de vídeo en memoria temporal también se puede denominar "descodificador de referencia hipotético" o "HRD". 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 su provisión. Por ejemplo, el HRD describe el funcionamiento de una memoria intermedia de imágenes codificadas ("CPB") y de una memoria intermedia de imágenes descodificadas ("DPB") en un descodificador de vídeo. La CPB es una memoria intermedia de primero en entrar primero en salir que contiene unidades de acceso en el orden de descodificación especificado por el HRD. La DPB es una memoria intermedia que contiene imágenes descodificadas para referencia, reordenamiento de salida o retardo de salida especificado por el HRD.

**[0018]** Un codificador de vídeo puede señalar un conjunto de parámetros HRD. Los parámetros HRD controlan diversos aspectos del HRD. Los parámetros 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. Estos parámetros HRD se codifican en una estructura sintáctica `hrd_parameters()` especificada en un VPS y/o un SPS. Los parámetros HRD también se pueden especificar en un mensaje de información de mejora suplementaria (SEI) de periodo de almacenamiento en memoria intermedia o un mensaje SEI de temporización de imágenes.

**[0019]** Como se explicó anteriormente, una representación de punto de funcionamiento puede tener una velocidad de trama y/o velocidad de transferencia de bits diferente que un flujo de bits original. Esto se debe a que la representación de 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 eliminase datos de la CPB y/o de la DPB a una velocidad particular al procesar el flujo de bits original y si el descodificador de vídeo eliminara datos de la CPB y/o de la DPB a la misma velocidad al procesar una representación de punto de funcionamiento, el descodificador de vídeo puede eliminar demasiados o muy pocos datos de la CPB y/o de la DPB. En consecuencia, el codificador de vídeo puede señalar diferentes conjuntos de parámetros HRD para diferentes puntos de funcionamiento. En la norma emergente de codificación de vídeo de alta eficiencia (HEVC), el codificador de vídeo puede señalar conjuntos de parámetros HRD en un VPS o el codificador de vídeo puede señalar conjuntos de parámetros HRD en un SPS. Un borrador de la inminente norma HEVC, conocido como "*HEVC Working Draft 8*" se describe en el documento de Bross *et al.*, "*High Efficiency Video Coding (HEVC) text specification draft 8*", Equipo Colaborativo Conjunto en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, décima conferencia, Estocolmo,

Suecia, julio de 2012, que, a partir del 28 de julio de 2012 está disponible en [http://phenix.intevry.fr/jct/doc\\_end\\_user/documents/10\\_Stockholm/wg11/JCTVC-J1003-v8.zip](http://phenix.intevry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip).

5 **[0020]** En algunas versiones de HEVC, solo se seleccionan los conjuntos de parámetros HRD en el VPS para operaciones HRD. Es decir, aunque los parámetros HRD se pueden proporcionar en los SPS, los descodificadores de vídeo HEVC no seleccionan los conjuntos de parámetros HRD en los SPS para operaciones HRD. Los descodificadores de vídeo siempre analizan sintácticamente y descodifican el VPS de un flujo de bits. Por lo tanto, los descodificadores de vídeo siempre analizan sintácticamente y descodifican los conjuntos de parámetros HRD del VPS. Esto es cierto independientemente de si el flujo de bits incluye unidades NAL que no son de capa base. Por lo tanto, si el flujo de bits incluye unidades NAL que no son de capa base, analizar sintácticamente y manejar los conjuntos de parámetros HRD en los SPS puede suponer un malgasto de recursos computacionales. Además, si los conjuntos de parámetros HRD están presentes en el VPS, los conjuntos de parámetros HRD en los SPS pueden ser bits desperdiciados.

15 **[0021]** De acuerdo con las técnicas de esta divulgación, un codificador de vídeo puede generar un flujo de bits que incluye un SPS que puede aplicarse a una secuencia de imágenes. El SPS incluye un conjunto de parámetros HRD. El conjunto de parámetros HRD es aplicable a cada punto de funcionamiento del flujo de bits que tiene un conjunto de identificadores de capa que coinciden con un conjunto de identificadores de capa objetivo. Por tanto, los conjuntos de parámetros HRD en los SPS no se desperdician, sino que se pueden usar para operaciones HRD. Por ejemplo, un dispositivo puede seleccionar, de entre un conjunto de parámetros HRD en un VPS y un conjunto de parámetros HRD en un SPS, un conjunto de parámetros HRD aplicables a un punto de funcionamiento particular. El dispositivo puede realizar, basándose al menos en parte en el conjunto de parámetros HRD aplicables al punto de funcionamiento particular, una prueba de conformidad de flujo de bits que prueba si un subconjunto de flujo de bits asociado al punto de funcionamiento particular se ajusta a una norma de codificación de vídeo.

25 **[0022]** Un dispositivo, tal como un codificador de vídeo, un descodificador de vídeo u otro tipo de dispositivo, tal como un dispositivo CDN o MANE, puede realizar una prueba de conformidad de flujo de bits en una representación de punto de funcionamiento para un punto de funcionamiento. La prueba de conformidad de flujo de bits puede verificar que la representación de punto de funcionamiento se ajusta a una norma de codificación de vídeo, tal como HEVC. Como se mencionó anteriormente, se puede usar un conjunto de identificadores de capa objetivo y un identificador temporal para identificar el punto de funcionamiento. El conjunto de identificadores de capa objetivo se puede denotar como "TargetDecLayerIdSet". El identificador temporal se puede denotar como "TargetDecHighestTid". Problemáticamente, el borrador de trabajo 8 de HEVC no especifica cómo se establecen TargetDecLayerIdSet o TargetDecHighestTid cuando se realiza una prueba de conformidad de flujo de bits.

35 **[0023]** De acuerdo con una o más técnicas de esta divulgación, un dispositivo puede realizar un proceso de descodificación como parte de la realización de una prueba de conformidad de flujo de bits. La realización del proceso de descodificación comprende realizar un proceso de extracción de flujo de bits para descodificar, desde un flujo de bits, una representación de punto de funcionamiento de un punto de funcionamiento definido por un conjunto objetivo de identificadores de capa y el identificador temporal objetivo más alto. El conjunto objetivo de identificadores de capa (es decir, TargetDecLayerIdSet) contiene valores de elementos sintácticos de identificador de capa (por ejemplo, elementos sintácticos nuh\_reserved\_zero\_6bits) presentes en la representación de punto de funcionamiento. El conjunto objetivo de identificadores de capa es un subconjunto de valores de elementos sintácticos de identificador de capa del flujo de bits. El identificador temporal objetivo más alto (es decir, TargetDecHighestTid) es igual al mayor identificador temporal presente en la representación de punto de funcionamiento. El identificador temporal objetivo más alto es menor que o igual al mayor identificador temporal presente en el flujo de bits. La realización del proceso de descodificación también puede comprender descodificar unidades NAL de la representación de punto de funcionamiento.

40 **[0024]** En HEVC, un SPS puede incluir una matriz de elementos sintácticos denominados `sps_max_dec_pic_buffering[i]`, donde  $i$  varía de 0 al número máximo de capas temporales en el flujo de bits. `sps_max_dec_pic_buffering[i]` indica el tamaño máximo requerido de la DPB cuando el identificador temporal más alto (HighestTid) es igual a  $i$ . `sps_max_dec_pic_buffering[i]` indica el tamaño requerido en términos de unidades de memorias intermedias de almacenamiento de imágenes.

55 **[0025]** Además, en HEVC, un SPS puede incluir una matriz de elementos sintácticos denominados `sps_max_num_reorder_pics[i]`, donde  $i$  varía de 0 al número máximo de capas temporales en el flujo de bits. `sps_max_num_reorder_pics[i]` indica un número máximo permitido de imágenes que preceden a cualquier imagen en orden de descodificación y que suceden a esa imagen en orden de salida cuando el identificador temporal más alto (HighestTid) es igual a  $i$ .

60 **[0026]** En HEVC, un conjunto de parámetros HRD puede incluir una matriz de elementos sintácticos denominados `cpb_cnt_minus1[i]`, donde  $i$  varía de 0 al número máximo de capas temporales en el flujo de bits. `cpb_cnt_minus1[i]` especifica el número de especificaciones de CPB alternativas en el flujo de bits de la secuencia de vídeo codificada cuando el identificador temporal más alto (HighestTid) es igual a  $i$ , donde una especificación de CPB alternativa se refiere a una operación CPB particular con un conjunto particular de parámetros CPB.

**[0027]** En el borrador de trabajo 8 de HEVC, `sps_max_dec_pic_buffering[i]`, `sps_max_num_reorder_pics[i]` y `cpb_cnt_minus1[i]` no se seleccionan apropiadamente en las operaciones HRD, las operaciones de conformidad de flujo de bits y las restricciones de nivel. Esto se debe, al menos en parte, a que el borrador de trabajo 8 de HEVC no especifica qué se entiende por identificador temporal más alto (HighestTid).

**[0028]** De acuerdo con una o más técnicas de esta divulgación, un dispositivo, tal como un codificador de vídeo, un descodificador de vídeo u otro dispositivo, puede determinar el identificador temporal más alto de un subconjunto de flujo de bits asociado a un punto de funcionamiento seleccionado de un flujo de bits. Además, el dispositivo puede determinar, basándose en el identificador temporal más alto, un elemento sintáctico particular de entre una matriz de elementos sintácticos (por ejemplo, `sps_max_dec_pic_buffering[ ]`, `sps_max_num_reorder_pics[ ]` o `cpb_cnt_minus1[ ]`). El dispositivo puede realizar una operación que usa el elemento sintáctico particular para determinar la conformidad del flujo de bits a una norma de codificación de vídeo o para determinar la conformidad de un descodificador de vídeo a la norma de codificación de vídeo.

**[0029]** La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación de vídeo 10 que puede utilizar las técnicas de esta divulgación. Tal como se usa 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.

**[0030]** Como se muestra en la FIG. 1, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 y un dispositivo de destino 14. El dispositivo de origen 12 genera datos de vídeo codificados. En consecuencia, el dispositivo de origen 12 se puede denominar 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 se puede denominar 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.

**[0031]** El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia gama de dispositivos, incluyendo ordenadores de escritorio, dispositivos informáticos móviles, ordenadores tipo *notebook* (por ejemplo, portátiles), tabletas electrónicas, descodificadores multimedia, aparatos telefónicos tales como los denominados teléfonos "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores para vehículos o similares.

**[0032]** El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 por medio de un canal 16. El canal 16 puede comprender uno o más medios o dispositivos capaces de transferir los datos de vídeo codificados desde el dispositivo de origen 12 hasta el 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 una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo de destino 14. El uno o más medios de comunicación pueden incluir medios de comunicación inalámbrica y/o alámbrica, tales como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El 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). El uno o más medios de comunicación pueden incluir encaminadores, conmutadores, estaciones base u otros equipos que faciliten la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

**[0033]** 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, por medio de acceso a disco o acceso a tarjeta. El medio de almacenamiento puede incluir una variedad de medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, DVD, CD-ROM, memoria flash u otros medios de almacenamiento digital adecuados para almacenar datos de vídeo codificados.

**[0034]** En otro ejemplo, el canal 16 puede incluir un servidor de archivos 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 archivos o en otro dispositivo de almacenamiento intermedio por medio de transmisión continua o descarga. El servidor de archivos 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. Ejemplos de servidores de archivos incluyen servidores web (por ejemplo, para una página web), servidores de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados en red (NAS) y unidades de disco local. En el ejemplo de la FIG. 1, el canal 16 incluye un dispositivo adicional 21. En algunos ejemplos, el dispositivo adicional 21 es un dispositivo CDN, un MANE u otro tipo de dispositivo.

**[0035]** 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. Ejemplos de tipos de conexiones de datos pueden incluir canales inalámbricos (por ejemplo, conexiones Wi-Fi), conexiones cableadas (por ejemplo, línea de abonado digital (DSL), módem por cable, etc.), o combinaciones de ambos que sean adecuadas para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

**[0036]** Las técnicas de esta divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como soporte de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, por medio de 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 estar configurado para admitir transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

**[0037]** La FIG. 1 es simplemente un ejemplo y las técnicas de esta divulgación se pueden aplicar 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 se recuperan desde una memoria local, se transmiten en continuo por una red o similares. Un dispositivo de codificación de vídeo puede codificar y almacenar datos en una memoria, y/o un dispositivo de descodificación de vídeo puede recuperar y descodificar datos de una memoria. En muchos ejemplos, la codificación y la descodificación se realizan mediante dispositivos que no se comunican entre sí, sino que simplemente codifican datos en una memoria y/o recuperan y descodifican datos de una memoria.

**[0038]** 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. La fuente de vídeo 18 puede incluir un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contenga 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 dichas fuentes de datos de vídeo.

**[0039]** El codificador de vídeo 20 puede codificar datos de vídeo procedentes de la fuente de vídeo 18. En algunos ejemplos, el dispositivo de origen 12 transmite directamente los datos de vídeo codificados al dispositivo de destino 14 por medio de la interfaz de salida 22. En otros ejemplos, los datos de vídeo codificados también pueden almacenarse en un medio de almacenamiento o en un servidor de archivos para un acceso posterior mediante el dispositivo de destino 14 para su descodificación y/o su reproducción.

**[0040]** 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 a través del canal 16. El dispositivo de visualización 32 puede estar integrado en, o ser externo a, el dispositivo de destino 14. En general, el dispositivo de visualización 32 muestra datos de vídeo descodificados. El dispositivo de visualización 32 puede comprender una variedad de 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.

**[0041]** El codificador de vídeo 20 y el descodificador de vídeo 30 pueden, cada uno, implementarse como cualquiera de una variedad de circuitos adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (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 de almacenamiento adecuado 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 lo mencionado anteriormente (incluyendo hardware, software, una combinación de hardware y software, etc.) puede considerarse uno o más procesadores. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado formando parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

**[0042]** Esta divulgación puede referirse, en general, al codificador de vídeo 20 que "señaliza" determinada información a otro dispositivo, tal como el descodificador de vídeo 30 o un dispositivo adicional 21. El término "señalizar" 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 periodo de tiempo, tal como podría producirse 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 pueden ser recuperados posteriormente por un dispositivo de descodificación en cualquier momento tras haberse almacenado en este medio.

5 **[0043]** En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 funcionan de acuerdo con una norma de compresión de vídeo, tal como ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluida su ampliación de codificación de vídeo escalable (SVC), su ampliación de codificación de vídeo de múltiples vistas (MVC) y su ampliación de 3DV basado en MVC. En algunos casos, cualquier flujo de bits que se ajuste a 3DV basado en MVC siempre contiene un subflujo de bits que es compatible con un perfil MVC, por ejemplo, un alto perfil estéreo. Además, actualmente se están emprendiendo unas iniciativas para generar una ampliación de codificación de vídeo tridimensional (3DV) para H.264/AVC, en concreto, 3DV basado en AVC. En otros ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con la ITU-T H.261, la ISO/IEC MPEG-1 Visual, la ITU-T H.262 o ISO/IEC MPEG-2 Visual, la ITU-T H.263, la ISO/IEC MPEG-4 Visual y la ITU-T H.264, ISO/IEC Visual.

10 **[0044]** En otros ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con la norma de codificación de vídeo de alta eficiencia (HEVC) elaborada actualmente 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 de la inminente norma HEVC, conocido como "Borrador de Trabajo 9 de HEVC" se describe en el documento de Bross *et al.*, "*High Efficiency Video Coding (HEVC) text specification draft 9*", Equipo Colaborativo Conjunto en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, décimo primera conferencia, Shangái, China, octubre de 2012, que, a partir del 8 de mayo de 2013 está disponible en [http://phenix.intevry.fr/jct/doc\\_end\\_user/documents/11\\_Shanghai/wg11/JCTVC-K1003-v13.zip](http://phenix.intevry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v13.zip). Además, actualmente se están emprendiendo unas iniciativas para la elaboración de ampliaciones de la SVC, la codificación de múltiples vistas y 3DV para HEVC. La ampliación de 3DV de HEVC puede denominarse 3DV basado en HEVC o 3D-HEVC.

20 **[0045]** En la HEVC y en otras normas 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 matrices de muestras, denotadas como  $S_L$ ,  $S_{Cb}$  y  $S_{Cr}$ .  $S_L$  es una matriz bidimensional (es decir, un bloque) de muestras de luma.  $S_{Cb}$  es una matriz bidimensional de muestras de crominancia Cb.  $S_{Cr}$  es una matriz bidimensional de muestras de crominancia Cr. Las muestras de crominancia también se pueden denominar en el presente documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede incluir solo una matriz de muestras de luma.

25 **[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 ser 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. Un bloque de árbol de codificación puede ser un bloque NxN de muestras. Una CTU también puede denominarse "bloque de árbol" o "unidad de codificación de máximo tamaño" (LCU). Las CTU de HEVC pueden ser análogas en términos generales a los macrobloques de otras normas, tales como H.264/AVC. Sin embargo, una CTU no está necesariamente limitada a un tamaño particular y puede incluir una o más unidades de codificación (CU). Un segmento puede incluir un número entero de CTU ordenadas consecutivamente en un orden de exploración de barrido.

30 **[0047]** Para generar una CTU codificada, el codificador de vídeo 20 puede realizar de forma recursiva una división 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 ser un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tenga 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. El codificador de vídeo 20 puede dividir bloques de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción puede ser 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 ser un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma de una imagen y estructuras sintácticas usadas para predecir las muestras de bloques de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luma, Cb y Cr para bloques de predicción de luma, Cb y Cr de cada PU de la CU.

35 **[0048]** El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 usa intrapredicción 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 asociada a la PU.



- 5 [0049] Si el codificador de vídeo 20 usa interpredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU basándose en unas muestras descodificadas de una o más imágenes distintas a la imagen asociada a la PU. El codificador de vídeo 20 puede usar predicción unidireccional o predicción bidireccional para generar los bloques predictivos de una PU. Cuando el codificador de vídeo 20 usa la predicción unidireccional para generar los bloques predictivos para una PU, la PU puede tener un solo vector de movimiento. Cuando el codificador de vídeo 20 usa predicción bidireccional para generar los bloques predictivos para una PU, la PU puede tener dos vectores de movimiento.
- 10 [0050] Después de que el codificador de vídeo 20 genere bloques predictivos de luma, Cb y Cr para una o más PU de una CU, 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 de predicción 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 predictivos de Cr de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.
- 20 [0051] Además, el codificador de vídeo 20 puede usar la división de árbol cuaternario para descomponer los bloques residuales de luma, Cb y Cr de una CU en uno o más bloques de transformada de luma, Cb y Cr. Un bloque de transformada puede ser un bloque rectangular de muestras en el que se aplique la misma transformada. Una unidad de transformada (TU) de una CU puede ser un bloque de transformada de muestras de luma, dos bloques de transformada correspondientes de muestras de croma y estructuras sintácticas usadas para transformar las muestras de bloques de transformada. Por tanto, cada TU de una CU puede estar asociada a un bloque de transformada de luma, un bloque de transformada de Cb y un bloque de transformada de Cr. El bloque de transformada de luma asociado a la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformada de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformada de Cr puede ser un subbloque del bloque residual de Cr de la CU.
- 25 [0052] El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de luma de una TU para generar un bloque de coeficientes de luma para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada 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 transformadas a un bloque de transformada de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.
- 35 [0053] 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 transformada se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformada, proporcionando más compresión. Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes, el codificador de vídeo 20 puede realizar una codificación de entropía de unos elementos sintácticos que indican los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar una codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos sintácticos que indican los coeficientes de transformada cuantificados. El codificador de vídeo 20 puede proporcionar los elementos sintácticos codificados por entropía en un flujo de bits.
- 40 [0054] El codificador de vídeo 20 puede proporcionar un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades de capa de abstracción de red (NAL). Una unidad NAL puede ser una estructura sintáctica que contiene una indicación del tipo de datos a seguir y octetos que contienen dichos datos en forma de una carga útil de secuencia de octetos sin procesar (RBSP) entremezclados según sea necesario con octetos de prevención de emulación. Es decir, cada una de las unidades NAL puede incluir una cabecera de unidad NAL y encapsular una RBSP. La cabecera de 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 octetos que están encapsulados dentro de una unidad NAL. En algunos casos, una RBSP incluye bits cero.
- 50 [0055] Diferentes tipos de unidades NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad de NAL puede encapsular una RBSP para un conjunto de parámetros de imagen (PPS), un segundo tipo de unidad NAL puede encapsular una RBSP para un segmento codificado, un tercer tipo de unidad NAL puede encapsular una RBSP para SEI, etc. Las unidades NAL que encapsulan las RBSP para datos de codificación de
- 60
- 65

vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes SEI) pueden denominarse unidades NAL de capa de codificación de vídeo (VCL).

5 **[0056]** 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 sintácticamente el flujo de bits para descodificar 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 descodificados 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 de 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 transformada asociados a las TU de la CU actual. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes de transformada para reconstruir los bloques de transformada asociados a 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 unas PU de la CU actual a unas muestras correspondientes de los bloques de transformada de las TU de la CU actual. Reconstruyendo los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.

20 **[0057]** En la codificación de múltiples vistas, puede haber múltiples vistas de la misma escena desde diferentes puntos de visión. El término "unidad de acceso" se usa para hacer referencia al conjunto de imágenes que corresponden a la misma instancia de tiempo. Por tanto, los datos de vídeo pueden conceptualizarse como una serie de unidades de acceso que se producen a lo largo del tiempo. Una "componente de vista" puede ser una representación codificada de una vista en una única unidad de acceso. En esta divulgación, una "vista" puede referirse a una secuencia de componentes de vista asociadas al mismo identificador de vista.

25 **[0058]** La codificación de múltiples vistas admite la predicción entre vistas. La predicción entre vistas es similar a la interpredicción usada en H.264/AVC y HEVC y puede usar los mismos elementos sintácticos. Sin embargo, cuando un codificador de vídeo realiza una predicción entre vistas en una unidad de vídeo actual (tal como una PU), el codificador de vídeo 20 puede usar, como una imagen de referencia, una imagen que está en la misma unidad de acceso que la unidad de vídeo actual, pero en una vista diferente. Por el contrario, la interpredicción convencional solo usa imágenes en diferentes unidades de acceso como imágenes de referencia.

35 **[0059]** En la codificación de múltiples vistas, una vista puede denominarse "vista base" si un descodificador de vídeo (por ejemplo, el descodificador de vídeo 30) puede descodificar imágenes de la vista sin referencia a imágenes de ninguna otra vista. Cuando se codifica una imagen en una de las vistas no base, un codificador de vídeo (tal como el codificador de vídeo 20 o el descodificador de vídeo 30) puede añadir una imagen a una lista de imágenes de referencia si la imagen está en una vista diferente pero en la misma instancia de tiempo (es decir, una unidad de acceso) que la imagen que el codificador de vídeo esté codificando actualmente. Al igual que otras imágenes de referencia de interpredicción, el codificador de vídeo puede insertar una imagen de referencia de predicción entre vistas en cualquier posición de una lista de imágenes de referencia.

40 **[0060]** Las normas de codificación de vídeo especifican modelos de almacenamiento de vídeo en memoria intermedia. En H.264/AVC y HEVC, el modelo de almacenamiento en memoria intermedia se denomina "descodificador de referencia hipotético" o "HRD". En el borrador de trabajo 8 de HEVC, el HRD se describe en el Anexo C.

45 **[0061]** 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 su provisión. Por ejemplo, el HRD describe el funcionamiento de una memoria intermedia de imágenes codificadas ("CPB"), una memoria intermedia de imágenes descodificadas ("DPB") y un decodificador 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 la temporización, los tamaños de memoria intermedia y las velocidades de transferencia de bits. Además, el HRD puede imponer restricciones indirectamente a diversas características y estadísticas de flujo de bits.

55 **[0062]** En H.264/AVC y HEVC, la conformidad de flujo de bits y la conformidad de descodificador se especifican como partes de la especificación de HRD. En otras palabras, el modelo de HRD especifica pruebas para determinar si un flujo de bits se ajusta a una norma y pruebas para determinar si un descodificador se ajusta a la norma. 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 no necesitan típicamente el HRD.

60 **[0063]** 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 unidades 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. Un flujo de bits de Tipo II es un flujo de

unidades 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 lo siguiente: 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 un flujo de octetos a partir del flujo de unidades NAL.

**[0064]** Cuando un dispositivo realiza una prueba de conformidad de flujo de bits que determina si un flujo de bits se ajusta a una norma 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 HRD aplicables al punto de funcionamiento seleccionado. El dispositivo puede usar el conjunto de parámetros HRD aplicables al punto de funcionamiento seleccionado para configurar el comportamiento del HRD. Más en particular, el dispositivo puede usar el conjunto aplicable de parámetros HRD para configurar los comportamientos de componentes particulares del HRD, tales como un planificador 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 una planificación particular.

**[0065]** Además, el dispositivo puede invocar un proceso de descodificación que descodifica los datos de vídeo codificados en la CPB. El proceso de descodificación puede proporcionar imágenes descodificadas a la DPB. A medida que el dispositivo transfiere 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 desbordamiento o subdesbordamiento en la CPB o DPB mientras el HRD está descodificando la representación de 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 cumple con la norma de codificación de vídeo.

**[0066]** H.264/AVC y HEVC especifican dos tipos de conformidad de descodificador; a saber, la conformidad de descodificador de temporización de salida y la conformidad de descodificador de orden de salida. Un descodificador que alega conformidad con un perfil, grado y nivel específicos es capaz de descodificar con éxito todos los flujos de bits que cumplen con los requisitos de conformidad de flujo de bits de una norma de codificación de vídeo, tal como HEVC. En esta divulgación, un "perfil" puede referirse a un subconjunto de la sintaxis de flujo de bits. Dentro de cada perfil se pueden especificar "grados" y "niveles". Un nivel de un grado puede ser un conjunto especificado de restricciones impuestas en valores de los elementos sintácticos en el flujo de bits. Estas restricciones pueden ser simples límites en los valores. De forma alternativa, pueden adoptar la forma de restricciones en combinaciones aritméticas de valores (por ejemplo, el ancho de imagen multiplicado por la altura de imagen multiplicado por el número de imágenes descodificadas por segundo). Un nivel especificado para un grado inferior está más restringido que un nivel especificado para un grado superior.

**[0067]** Cuando un dispositivo realiza una prueba de conformidad de descodificador para determinar si un descodificador bajo prueba (DUT) se ajusta a una norma de codificación de vídeo, el dispositivo puede proporcionar, tanto al HRD como al DUT, un flujo de bits que se ajusta a la norma 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 cumple con la norma de codificación de vídeo si el orden de las imágenes descodificadas proporcionadas por el DUT coincide con el orden de las imágenes descodificadas proporcionadas por el HRD. Además, el dispositivo puede determinar que el DUT cumple con la norma de codificación de vídeo si la temporización con la que el DUT proporciona imágenes descodificadas coincide con la temporización con la que el HRD proporciona las imágenes descodificadas.

**[0068]** Además de las pruebas de conformidad de flujo de bits y las pruebas de conformidad de descodificador, los dispositivos pueden usar parámetros HRD para otros propósitos. Por ejemplo, los retardos iniciales en la eliminación de CPB pueden usarse para guiar a un sistema para establecer un retardo inicial apropiado de extremo a extremo y los tiempos de salida de DPB se pueden usar para obtener las marcas de tiempo de protocolo de tiempo real (RTP) cuando el flujo de bits de datos de vídeo se transporta a través de RTP.

**[0069]** En los modelos H.264/AVC y HRD HEVC, 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 HRD HEVC, se supone que la descodificación de imágenes es instantánea. El codificador de vídeo 20 puede señalar, en los mensajes SEI de temporización de imágenes, los tiempos de descodificación para comenzar la descodificación de unidades de acceso. En aplicaciones prácticas, si un descodificador de vídeo en conformidad sigue estrictamente los tiempos de descodificación señalizados para iniciar la descodificación de unidades de acceso, el momento más temprano posible para proporcionar una imagen descodificada particular 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.

**[0070]** Los parámetros HRD pueden controlar diversos aspectos del HRD. En otras palabras, el HRD puede basarse en los parámetros HRD. Los parámetros HRD pueden incluir un retardo inicial de eliminación de CPB, un tamaño de CPB, una velocidad de transferencia de bits, un retardo inicial de salida de DPB y un tamaño de DPB. El codificador de vídeo 20 puede señalar estos parámetros 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 HRD. En algunos ejemplos, el codificador de vídeo 20 puede señalar parámetros HRD en mensajes SEI de periodo de almacenamiento en memoria intermedia o mensajes SEI de temporización de imágenes.

**[0071]** Como se explicó anteriormente, un punto de funcionamiento de un flujo de bits está asociado a un conjunto de identificadores de capa (es decir, un conjunto de valores `nuh_reserved_zero_6bits`) y un identificador temporal. Una representación de punto de funcionamiento puede incluir cada unidad NAL que está asociada al punto de funcionamiento. Una representación de punto de funcionamiento puede tener una velocidad de trama y/o velocidad de transferencia de bits diferente a la de un flujo de bits original. Esto se debe a que la representación de punto de funcionamiento puede no incluir algunas imágenes y/o algunos de los datos del flujo de bits original. Por lo tanto, si el decodificador de vídeo 30 fuera a eliminar datos de la CPB y/o de la DPB a una velocidad particular cuando procesa el flujo de bits original y si el decodificador de vídeo 30 fuera a eliminar datos de la CPB y/o de la DPB a la misma velocidad cuando procesa una representación de punto de funcionamiento, el decodificador 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 señalar diferentes conjuntos de parámetros 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 HRD para diferentes puntos de funcionamiento.

**[0072]** En el borrador de trabajo 8 de HEVC, el conjunto de parámetros HRD incluye opcionalmente un conjunto de información que es común para todas las subcapas. En otras palabras, el conjunto de parámetros HRD puede incluir opcionalmente un conjunto de elementos sintácticos comunes que son aplicables a puntos de funcionamiento que incluyen cualquier subcapa temporal. Una subcapa temporal es una capa escalable temporal de un flujo de bits escalable temporal que consiste en unidades NAL VCL con un valor particular de `TemporalId` y las unidades NAL no VCL asociadas. Además del conjunto de información común, los conjuntos de parámetros HRD pueden incluir un conjunto de elementos sintácticos que son específicos de subcapas temporales individuales. Por ejemplo, la estructura sintáctica `hrd_parameters( )` puede incluir opcionalmente un conjunto de información que es común para todas las subcapas y siempre incluye información específica de subcapa. Debido a que el conjunto de información común es común a múltiples conjuntos de parámetros HRD, puede ser innecesario señalar el conjunto de información común en múltiples conjuntos de parámetros HRD. Por el contrario, en el borrador de trabajo 8 de HEVC, la información común puede estar presente en un conjunto de parámetros HRD cuando el conjunto de parámetros HRD es el primer conjunto de parámetros HRD en un VPS o la información común puede estar presente en un conjunto de parámetros HRD cuando el conjunto de parámetros HRD está asociado a un primer índice de punto de funcionamiento. Por ejemplo, el borrador de trabajo 8 de HEVC admite la presencia de información común cuando la estructura sintáctica `hrd_parameters( )` es la primera estructura sintáctica `hrd_parameters( )` en el VPS o cuando la estructura sintáctica `hrd_parameters( )` está asociada a un primer índice de punto de funcionamiento.

**[0073]** La siguiente Tabla 1 es una estructura sintáctica de ejemplo de una estructura sintáctica `hrd_parameters( )` en HEVC.

**TABLA 1- Parámetros HRD**

<code>hrd_parameters (commonInfPresentFlag, MaxNumSubLayersMinus1) {</code>	<b>Descriptor</b>
<code>if(commonInfPresentFlag) {</code>	
<b><code>timing_info_present_flag</code></b>	<code>u(1)</code>
<code>if(timing_info_present_flag) {</code>	
<b><code>num_units_in_tick</code></b>	<code>u(32)</code>
<b><code>time_scale</code></b>	<code>u(32)</code>
<code>}</code>	
<b><code>nal_hrd_parameters_present_flag</code></b>	<code>u(1)</code>
<b><code>vcl_hrd_parameters_present_flag</code></b>	<code>u(1)</code>
<code>if(nal_hrd_parameters_present_flag    vcl_hrd_parameters_present_flag) {</code>	
<b><code>sub_pic_cpb_params_present_flag</code></b>	<code>u(1)</code>
<code>if(sub_pic_cpb_params_present_flag) {</code>	

hrd_parameters (commonInfPresentFlag, MaxNumSubLayersMinus1) {	Descriptor
<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_minus[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)	
}	
}	

[0074] 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 sin signo de longitud variable codificados usando codificación de Golomb exponencial (Exp-Golomb) de orden 0 con 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.

[0075] En el ejemplo de sintaxis de la Tabla 1, los elementos sintácticos en el bloque "if (commonInfPresentFlag) {...}" son la información común de los conjuntos de parámetros HRD. En otras palabras, la información común del conjunto de parámetros HRD puede incluir los elementos sintácticos timing\_info\_present\_flag, num\_units\_in\_tick, time\_scale, nal\_hrd\_parameters\_present\_flag, vcl\_hrd\_parameters\_present\_flag, sub\_pic\_cpb\_params\_present\_flag, tick\_divisor\_minus2, du\_cpb\_removal\_delay\_length\_minus1, bit\_rate\_scale, cpb\_size\_scale, initial\_cpb\_removal\_delay\_length\_minus1, cpb\_removal\_delay\_length\_minus1 y dpb\_output\_delay\_length\_minus1.

[0076] Además, en el ejemplo de la Tabla 1, los elementos sintácticos fixed\_pic\_rate\_flag[i], pic\_duration\_in\_tc\_minus1[i], low\_delay\_hrd\_flag[i] y cpb\_cnt\_minus1[i] pueden ser un conjunto de parámetros HRD específicos de subcapa. En otras palabras, estos elementos sintácticos de la estructura sintáctica hrd\_parameter() solo pueden aplicarse a los puntos de funcionamiento que incluyen una subcapa específica. Por lo tanto, los parámetros HRD de una estructura sintáctica hrd\_parameters() pueden incluir, además de la información común incluida opcionalmente, un conjunto de parámetros HRD específicos de subcapa que son específicos de una subcapa particular del flujo de bits.

[0077] El elemento sintáctico fixed\_pic\_rate\_flag[i] puede indicar que, cuando HighestTid es igual a i, la distancia temporal entre los tiempos de salida de HRD de cualesquiera dos imágenes consecutivas en orden de salida está limitado de manera específica. HighestTid puede ser una variable que identifica una subcapa temporal más alta (por ejemplo, de un punto de funcionamiento). El elemento sintáctico pic\_duration\_in\_tc\_minus1[i] puede especificar, cuando HighestTid es igual a i, la distancia temporal, en pulsos de reloj, entre los tiempos de salida de HRD de cualquier imagen consecutiva en orden de salida en la secuencia de vídeo codificada. El elemento sintáctico low\_delay\_hrd\_flag[i] puede especificar el modo de funcionamiento de HRD, cuando HighestTid es igual a i, como se especifica en el Anexo C del borrador de trabajo 8 de HEVC. El elemento sintáctico cpb\_cnt\_minus1[i] puede

especificar el número de especificaciones de CPB alternativas en el flujo de bits de la secuencia de vídeo codificada cuando HighestTid es igual a i.

5 **[0078]** El codificador de vídeo 20 puede usar mensajes SEI para incluir, en el flujo de bits, metadatos que no son necesarios para la descodificación correcta de los valores de muestra de las imágenes. Sin embargo, el descodificador de vídeo 30 u otros dispositivos pueden usar los metadatos incluidos en los mensajes SEI para otros propósitos diversos. Por ejemplo, el descodificador de vídeo 30 puede usar los metadatos en los mensajes SEI para la temporización de salida de imágenes, la visualización de imágenes, la detección de pérdidas y la ocultación de errores.

10 **[0079]** El codificador de vídeo 20 puede incluir una o más unidades NAL SEI en una unidad de acceso. En otras palabras, cualquier número de unidades NAL SEI puede estar asociada a una unidad de acceso. Además, cada unidad NAL SEI puede contener uno o más mensajes SEI. La norma HEVC describe la sintaxis y la semántica para varios tipos de mensajes SEI. Sin embargo, la norma HEVC no describe el manejo de los mensajes SEI porque los mensajes SEI no afectan al proceso de descodificación normativa. Una razón para tener mensajes SEI en la norma HEVC es permitir que los datos suplementarios se interpreten de manera idéntica en diferentes sistemas usando HEVC. Las especificaciones y sistemas que usan HEVC pueden requerir codificadores de vídeo para generar ciertos mensajes SEI o pueden definir el manejo específico de tipos particulares de mensajes SEI recibidos. La siguiente Tabla 2 enumera los mensajes SEI especificados en HEVC y describe brevemente sus propósitos.

20 **TABLA 2 - Visión general de mensajes SEI**

Mensaje SEI	Finalidad
Periodo de almacenamiento en memoria intermedia	Retardos iniciales para el funcionamiento de descodificador de referencia hipotético (HRD)
Temporización de imágenes	Tiempo de salida de imagen y tiempo de eliminación de imagen/subimagen para el funcionamiento de HRD
Rectángulo de barrido panorámico	Visualización con una relación de aspecto de imagen (PAR) diferente a la PAR de las imágenes de salida
Carga útil de relleno	Ajuste de la velocidad de transferencia de bits para satisfacer limitaciones específicas
Datos de usuario registrados - Datos de usuario no registrados	Mensajes SEI a especificar por entidades externas
Punto de recuperación	Información adicional para acceso aleatorio limpio. Actualización gradual de descodificación.
Información de escena	Información sobre cambios de escena y transiciones
Instantánea a pantalla completa	Indicación para etiquetar la imagen descodificada asociada como una instantánea de imagen fija del contenido de vídeo
Segmento de refinamiento progresivo	Indica que determinadas imágenes consecutivas representan un refinamiento progresivo de la calidad de una imagen, en lugar de una escena en movimiento
Características de textura de película	Permite a los descodificadores sintetizar la textura de la película
Preferencia de visualización de filtro de eliminación de bloques	Recomienda si las imágenes visualizadas deberían someterse o no al proceso del filtro de eliminación de bloques en bucle
Sugerencia de posfiltro	Proporciona coeficientes de posfiltro sugeridos o información de correlación para el diseño de posfiltro
Información de correlación de tonos	Nueva correlación con otro espacio de colores distinto al usado o adoptado en la codificación
Disposición de empaquetamiento de tramas de	Empaquetamiento de vídeo estereoscópico en un flujo de bits de HEVC
Orientación de la visualización	Especifica el vuelco y/o la rotación que deberían aplicarse a las imágenes de salida cuando se visualizan
Indicación de campo	Proporciona información relacionada con el contenido de vídeo intercalado y/o la codificación de campo; por ejemplo, indica si la

Mensaje SEI	Finalidad
	imagen es una trama progresiva, un campo o una trama que contiene dos campos intercalados
Hash de imagen descodificada	Suma de comprobación de la imagen descodificada, que puede ser usada para la detección de errores
Temporización de subimágenes	Tiempo de eliminación de subimágenes para funcionamiento de HRD
Conjuntos de parámetros activos	Proporciona información sobre VPS, SPS, etc., activos
Estructura de descripción de imágenes	Describe la estructura temporal y de interpredicción del flujo de bits

**[0080]** Existen varios problemas o desventajas con las técnicas existentes para señalar parámetros HRD y seleccionar parámetros HRD y otros parámetros. Por ejemplo, en el borrador de trabajo 8 de HEVC, solo se seleccionan los conjuntos de parámetros HRD en el VPS para operaciones de HRD. Es decir, aunque los parámetros HRD se pueden proporcionar en los SPS, los descodificadores de vídeo HEVC no seleccionan los conjuntos de parámetros HRD en los SPS para operaciones HRD. Los descodificadores de vídeo siempre analizan sintácticamente y descodifican el VPS de un flujo de bits. Por lo tanto, los descodificadores de vídeo siempre analizan sintácticamente y descodifican los conjuntos de parámetros HRD del VPS.

**[0081]** Esto es cierto independientemente de si el flujo de bits incluye unidades NAL que no son de capa base. Por ejemplo, solo la estructura sintáctica `hrd_parameters()` codificada en el VPS puede seleccionarse para operaciones HRD, y la estructura sintáctica `hrd_parameters()` posiblemente presente en el SPS puede no seleccionarse nunca. Esto puede requerir el análisis sintáctico y manejo del VPS, incluso cuando se descodifica un flujo de bits que no contiene `nuh_reserved_zero_6bits` mayor que 0 (es decir, el flujo de bits contiene solo la capa base en una ampliación de múltiples vistas, 3DV o SVC de HEVC).

**[0082]** Si el flujo de bits incluye unidades NAL que no son de capa base, analizar sintácticamente y manejar los conjuntos de parámetros HRD en los SPS puede suponer un desperdicio de recursos computacionales. Además, si los conjuntos de parámetros HRD están presentes en el VPS, los conjuntos de parámetros HRD en los SPS pueden ser bits desperdiciados. Por ejemplo, si una estructura sintáctica `hrd_parameters()` está presente en el SPS, los bits codificados para la estructura sintáctica pueden ser simplemente un desperdicio de bits.

**[0083]** De acuerdo con una o más técnicas de esta divulgación, el codificador de vídeo puede generar un flujo de bits que incluye un SPS que puede aplicarse a una secuencia de imágenes. El SPS incluye un conjunto de parámetros HRD. El conjunto de parámetros HRD es aplicable a cada punto de funcionamiento del flujo de bits que tiene un conjunto de identificadores de capa que coincide con un conjunto de identificadores de capa objetivo. Por tanto, los conjuntos de parámetros HRD en los SPS no se desperdician, sino que se pueden usar para operaciones HRD. Por ejemplo, los puntos de funcionamiento con respecto a los cuales la estructura sintáctica `hrd_parameters()` codificada en un SPS se puede especificar claramente, por ejemplo, para ser los puntos de funcionamiento para los que solo un valor de `nuh_reserved_zero_6bits` (es decir, ID de capa en una extensión de codificación de vídeo escalable, 3DV o de múltiples vistas) está presente en el flujo de bits.

**[0084]** Por ejemplo, un dispositivo, tal como un codificador de vídeo o un descodificador de vídeo, puede seleccionar, de entre un conjunto de parámetros HRD en un conjunto de parámetros de vídeo, VPS, y un conjunto de parámetros HRD en un SPS, un conjunto de parámetros HRD aplicables a un punto de funcionamiento particular. En este ejemplo, el dispositivo puede realizar, basándose al menos en parte en el conjunto de parámetros HRD aplicables al punto de funcionamiento particular, una prueba de conformidad de flujo de bits que prueba si un subconjunto de flujo de bits asociado al punto de funcionamiento particular se ajusta a una norma de codificación de vídeo. La prueba de conformidad de flujo de bits puede verificar que la representación de punto de funcionamiento se ajusta a una norma de codificación de vídeo, tal como HEVC.

**[0085]** En esta divulgación, un punto de funcionamiento puede identificarse mediante un conjunto de valores `nuh_reserved_zero_6bits`, denotados como `OpLayerIdSet`, y un valor `TemporalId`, denotado como `OpTid`. El subconjunto de flujo de bits asociado obtenido como la salida del proceso de extracción de subflujo de bits como se especifica en la subcláusula 10.1 del borrador de trabajo 8 de HEVC con `OpTid` y `OpLayerIdSet` como entradas puede descodificarse de manera independiente. La subcláusula 10.1 del borrador de trabajo 8 de HEVC describe una operación para extraer un subflujo de bits (es decir, una representación de punto de funcionamiento) del flujo de bits. Específicamente, la subcláusula 10.1 del borrador de trabajo 8 de HEVC indica que el subflujo de bits se obtiene al eliminar del flujo de bits todas las unidades NAL con identificadores temporales (por ejemplo, `TemporalID`) mayores que `tIdTarget` o identificadores de capa (por ejemplo, `nuh_reserved_zero_6bits`) que no estén entre los valores de `targetDeclayerIdSet`. `tIdTarget` y `targetDeclayerIdSet` son parámetros del proceso de extracción de flujo de bits.

**[0086]** En otro problema o inconveniente de ejemplo con las técnicas existentes para señalar parámetros HRD, un dispositivo, tal como un codificador de vídeo, un descodificador de vídeo u otro tipo de dispositivo, puede realizar una prueba de conformidad de flujo de bits en una representación de punto de funcionamiento para un punto de funcionamiento. Como se mencionó anteriormente, se puede usar un conjunto de identificadores de capa objetivo y un identificador temporal para identificar el punto de funcionamiento. El conjunto de identificadores de capa objetivo se puede denotar como "TargetDecLayerIdSet". El identificador temporal se puede denotar como "TargetDecHighestTid". Problemáticamente, el borrador de trabajo 8 de HEVC no especifica cómo se establecen TargetDecLayerIdSet o TargetDecHighestTid cuando se realiza una prueba de conformidad de flujo de bits. Por ejemplo, cuando se invoca el proceso de descodificación para una prueba de conformidad de flujo de bits, la semántica de los elementos sintácticos no se especifica claramente, ya que los valores de TargetDecLayerIdSet y TargetDecHighestTid no se establecen correctamente.

**[0087]** Una o más técnicas de esta divulgación indican cómo TargetDecLayerIdSet y TargetDecHighestTid se establecen al realizarse una prueba de conformidad de flujo de bits. Por ejemplo, el proceso de descodificación general para un flujo de bits (o representación de punto de funcionamiento) se modifica de modo que si el flujo de bits (o la representación de punto de funcionamiento) se descodifica en una prueba de conformidad del flujo de bits, TargetDecLayerIdSet se establece como se especifica en la subcláusula C.1 de la norma HEVC. De forma similar, el proceso de descodificación general para un flujo de bits (o representación de punto de funcionamiento) puede modificarse de modo que si el flujo de bits (o representación de punto de funcionamiento) se descodifica en una prueba de conformidad de flujo de bits, TargetDecHighestTid se establece como se especifica en la subcláusula C.1 del borrador de trabajo 8 de HEVC. En otras palabras, el dispositivo puede determinar un conjunto de identificadores de capa objetivo del punto de funcionamiento particular que contiene cada identificador de capa presente en el subconjunto de flujo de bits y el conjunto de identificadores de capa del punto de funcionamiento particular es un subconjunto de identificadores de capa presente en el flujo de bits. Además, el dispositivo puede determinar un identificador temporal objetivo del punto de funcionamiento particular que es igual al mayor identificador temporal presente en el subconjunto de flujo de bits y el identificador temporal objetivo del punto de funcionamiento particular es menor que o igual al mayor identificador temporal presente en el flujo de bits.

**[0088]** En la subcláusula C.1 del borrador de trabajo 8 de HEVC, TargetDecLayerIdSet se establece como targetOpLayerIdSet. targetOpLayerIdSet contiene el conjunto de valores para nuh\_reserved\_zero\_6bits presentes en la representación de punto de funcionamiento del punto de funcionamiento bajo prueba. targetOpLayerIdSet es un subconjunto de los valores para nuh\_reserved\_zero\_6bits presentes en el flujo de bits bajo prueba.

**[0089]** Además, la variable TargetDecHighestTid identifica la subcapa temporal más alta a descodificar. Una subcapa temporal es una capa escalable temporal de un flujo de bits escalable temporal que consiste en unidades NAL VCL con un valor particular de TemporalId y las unidades NAL no VCL asociadas. En la subcláusula C.1 de la norma HEVC, TargetDecHighestTid se establece como targetOpTid. targetOpTid es igual al mayor temporal\_id presente en la representación de punto de funcionamiento del punto de funcionamiento bajo prueba y es menor que o igual al mayor id temporal presente en el flujo de bits bajo prueba. Por lo tanto, cuando se invoca el proceso de descodificación para una prueba de conformidad de flujo de bits, los valores de TargetDecLayerIdSet y TargetDecHighestTid se establecen en el conjunto de valores nuh\_reserved\_zero\_6bits y el mayor valor de TemporalId presente en el subflujo de bits correspondiente al punto de funcionamiento bajo prueba para la prueba de conformidad de flujo de bits específico.

**[0090]** De esta manera, un dispositivo (tal como el codificador de vídeo 20, el descodificador de vídeo 30, el dispositivo adicional 21 u otro dispositivo) puede, de acuerdo con una o más técnicas de esta divulgación, realizar un proceso de descodificación como parte de la realización de una prueba de conformidad de flujo de bits. La realización del proceso de descodificación puede comprender realizar un proceso de extracción de flujo de bits para extraer, de un flujo de bits, una representación de punto de funcionamiento de un punto de funcionamiento definido por un conjunto objetivo de identificadores de capa y el identificador temporal objetivo más alto. El conjunto objetivo de identificadores de capa (es decir, TargetDecLayerIdSet) contiene valores de elementos sintácticos de identificador de capa (por ejemplo, elementos sintácticos nuh\_reserved\_zero\_6bits) presentes en la representación de punto de funcionamiento. El conjunto objetivo de identificadores de capa es un subconjunto de valores de elementos sintácticos de identificador de capa del flujo de bits. El identificador temporal objetivo más alto (es decir, TargetDecHighestTid) es igual al mayor identificador temporal presente en la representación de punto de funcionamiento. El identificador temporal objetivo más alto es menor que o igual al mayor identificador temporal presente en el flujo de bits. La realización del proceso de descodificación también comprende descodificar unidades NAL de la representación de punto de funcionamiento.

**[0091]** El proceso de descodificación no siempre se realiza como parte de la realización de una prueba de conformidad de flujo de bits. Más bien, el proceso de descodificación puede ser un proceso general para descodificar un flujo de bits. Cuando el proceso de descodificación no se realiza como parte de una prueba de conformidad de flujo de bits, una fuente externa puede especificar TargetDecLayerIdSet y TargetDecHighestTid para un punto de funcionamiento. La fuente externa puede ser cualquier fuente de información fuera del flujo de bits. Por ejemplo, un dispositivo CDN puede determinar y especificar programáticamente TargetDecLayerIdSet y



TargetDecHighestTid en función de la configuración de un decodificador de vídeo particular. El dispositivo que realiza el proceso de descodificación puede usar TargetDecLayerIdSet y TargetDecHighestTid especificados externamente para extraer la representación de punto de funcionamiento del flujo de bits. El dispositivo que realiza el proceso de descodificación puede entonces descodificar unidades NAL de la representación de punto de funcionamiento extraída.

**[0092]** Por lo tanto, cuando el proceso de descodificación no se realiza como parte de la prueba de conformidad de flujo de bits, el dispositivo que realiza el proceso de descodificación puede recibir, desde una fuente externa, un conjunto objetivo de identificadores de capa y el identificador temporal objetivo más alto. El conjunto objetivo de identificadores de capa contiene valores de elementos sintácticos de identificador de capa presentes en una representación de punto de funcionamiento. El identificador temporal objetivo más alto es igual al mayor identificador temporal presente en la segunda representación de punto de funcionamiento. Además, el dispositivo que realiza el proceso de descodificación puede realizar el proceso de extracción de flujo de bits para extraer, del flujo de bits, la representación de punto de funcionamiento. El dispositivo que realiza el proceso de descodificación puede entonces descodificar unidades NAL de la representación de punto de funcionamiento.

**[0093]** En otros casos, una fuente externa no especifica TargetDecLayerIdSet o TargetDecHighestTid. En tales casos, el proceso de descodificación puede realizarse en todo el flujo de bits. Por ejemplo, el dispositivo puede realizar el proceso de extracción de flujo de bits para extraer, del flujo de bits, una representación de punto de funcionamiento. En este ejemplo, 0 es el único valor de los elementos sintácticos de identificador de capa (es decir, nuh\_reserved\_zero\_6bits) presentes en la representación de punto de funcionamiento. Además, en este ejemplo, el mayor identificador temporal presente en el flujo de bits es igual al mayor identificador temporal presente en la representación de punto de funcionamiento. En este ejemplo, el dispositivo que realiza el proceso de descodificación puede descodificar unidades NAL de la representación de punto de funcionamiento.

**[0094]** Como se indicó anteriormente, un SPS puede incluir una matriz de elementos sintácticos denominados `sps_max_dec_pic_buffering[i]`, donde *i* varía de 0 al número máximo de capas temporales en el flujo de bits. `sps_max_dec_pic_buffering[i]` indica el tamaño máximo requerido de la DPB cuando el identificador temporal más alto (HighestTid) es igual a *i*. `sps_max_dec_pic_buffering[i]` indica el tamaño requerido en términos de unidades de memorias intermedias de almacenamiento de imágenes. Además, un SPS puede incluir una matriz de elementos sintácticos denotada como `sps_max_num_reorder_pics[i]`, donde *i* varía de 0 al número máximo de capas temporales en el flujo de bits. `sps_max_num_reorder_pics[i]` indica un número máximo permitido de imágenes que preceden a cualquier imagen en orden de descodificación y que suceden a esa imagen en orden de salida cuando el identificador temporal más alto (HighestTid) es igual a *i*. Además, un conjunto de parámetros HRD puede incluir una matriz de elementos sintácticos denotados como `cpb_cnt_minus1[i]`, donde *i* varía de 0 al número máximo de capas temporales en el flujo de bits. `cpb_cnt_minus1[i]` especifica el número de especificaciones de CPB alternativas en el flujo de bits de la secuencia de vídeo codificada cuando el identificador temporal más alto (HighestTid) es igual a *i*.

**[0095]** Debido a que el borrador de trabajo 8 de HEVC no especifica qué se entiende por el identificador temporal más alto (HighestTid), borrador de trabajo 8 de HEVC,

`sps_max_dec_pic_buffering[i]`, `sps_max_num_reorder_pics[i]` y `cpb_cnt_minus1[i]` no se seleccionan apropiadamente en operaciones HRD, en operaciones de conformidad de flujo de bits ni en restricciones de nivel. En otras palabras, los parámetros `sps_max_num_reorder_pics[i]`, `sps_max_dec_pic_buffering[i]` y `cpb_cnt_minus1[i]` en las operaciones HRD, los requisitos de conformidad de flujo de bits y las restricciones de nivel no se seleccionan apropiadamente.

**[0096]** De acuerdo con una o más técnicas de esta divulgación, `sps_max_dec_pic_buffering[i]` se define de modo que `sps_max_dec_pic_buffering[i]` indica el tamaño máximo requerido de la DPB cuando TargetDecHighestTid es igual a *i*. TargetDecHighestTid se determina de la manera descrita anteriormente. Esto puede contrastar con el borrador de trabajo 8 de HEVC, donde HighestTid no está definido. El valor de `sps_max_dec_pic_buffering[i]` estará en el intervalo de 0 a MaxDpbSize (como se especifica en la subcláusula A.4 del borrador de trabajo 8 de HEVC), ambos inclusive. Cuando *i* es mayor que 0, `sps_max_dec_pic_buffering[i]` será igual a o mayor que `sps_max_dec_pic_buffering[i - 1]`. El valor de `sps_max_dec_pic_buffering[i]` será menor que o igual a `vps_max_dec_pic_buffering[i]` para cada valor de *i*.

**[0097]** De manera similar, de acuerdo con una o más técnicas de esta divulgación, `sps_max_num_reorder_pics[i]` se define de modo que `sps_max_num_reorder_pics[i]` indica el número máximo permitido de imágenes que preceden a cualquier imagen en orden de descodificación y que suceden esa imagen en orden de salida cuando TargetDecHighestTid es igual a *i*. TargetDecHighestTid se determina de la manera descrita anteriormente. El valor de `sps_max_num_reorder_pics[i]` estará en el intervalo de 0 a `sps_max_dec_pic_buffering[i]`, ambos inclusive. Cuando *i* es mayor que 0, `sps_max_num_reorder_pics[i]` será igual a o mayor que `sps_max_num_reorder_pics[i - 1]`. El valor de `sps_max_num_reorder_pics[i]` será menor que o igual a `vps_max_num_reorder_pics[i]` para cada valor de *i*.

**[0098]** Además, de acuerdo con una o más técnicas de esta divulgación, `cpb_cnt_minus1[i]` puede especificar el número de especificaciones de CPB alternativas en el flujo de bits de la secuencia de vídeo codificada cuando `TargetDecHighestTid` es igual a `i`, donde `i` varía de 0 al número máximo de capas temporales en el flujo de bits. `TargetDecHighestTid` se determina de la manera descrita anteriormente. El valor de `cpb_cnt_minus1[i]` está en el intervalo de 0 a 31, ambos inclusive. Cuando `low_delay_hrd_flag[i]` es igual a 1, `cpb_cnt_minus1[i]` es igual a 0. Cuando `cpb_cnt_minus1[i]` no está presente, se deduce que `cpb_cnt_minus1[i]` es igual a 0.

**[0099]** Por lo tanto, de acuerdo con una o más técnicas de esta divulgación, un dispositivo puede determinar, basándose en un identificador temporal más alto, un elemento sintáctico particular de entre una matriz de elementos sintácticos. El identificador temporal más alto se define de modo que el identificador temporal más alto siempre identifica la capa temporal más alta a descodificar. Por lo tanto, `sps_max_num_reorder_pics[i]`, `sps_max_dec_pic_buffering[i]`, y

`cpb_cnt_minus1[i]` en las operaciones HRD, los requisitos de conformidad de flujo de bits y las restricciones de nivel se seleccionan de manera coherente con `i` igual al valor claramente especificado de `TargetDecHighestTid`.

**[0100]** De esta manera, un dispositivo (tal como un codificador de vídeo 20, un descodificador de vídeo 30, un dispositivo adicional 21 u otro dispositivo) puede realizar una operación HRD para determinar la conformidad de un flujo de bits a una norma de codificación de vídeo o para determinar la conformidad de un descodificador de vídeo a la norma de codificación de vídeo. Como parte de la realización de la operación HRD, el dispositivo puede determinar el identificador temporal más alto de un subconjunto de flujo de bits asociado a un punto de funcionamiento seleccionado del flujo de bits. Además, el dispositivo puede determinar, basándose en el identificador temporal más alto, un elemento sintáctico particular de entre una matriz de elementos sintácticos (por ejemplo, `sps_max_num_reorder_pics[i]`, `sps_max_dec_pic_buffering[i]` o `cpb_cnt_minus1[i]`). Además, el dispositivo puede usar el elemento sintáctico particular en la operación HRD.

**[0101]** Además, en el borrador de trabajo 8 de HEVC, cada una de las estructuras sintácticas `hrd_parameters()` en el VPS puede asociarse a una estructura sintáctica `operation_point_layer_ids()` en base a la cual se selecciona una estructura sintáctica `hrd_parameters()` para su uso en las operaciones HRD. En correspondencia con cada estructura sintáctica `hrd_parameters()` seleccionada, también puede ser necesario un conjunto de mensajes SEI de periodo de almacenamiento en memoria intermedia y mensajes SEI de temporización de imágenes en las operaciones HRD. Sin embargo, no hay manera de asociar un mensaje SEI de periodo de almacenamiento en memoria intermedia o un mensaje SEI de temporización de imágenes a una estructura sintáctica `hrd_parameters()` para la cual la estructura sintáctica `operation_point_layer_ids()` asociada incluye múltiples valores de `nuh_reserved_zero_6bits` (es decir, múltiples ID de capa en una extensión de codificación de vídeo escalable, 3DV o de múltiples vistas de HEVC).

**[0102]** Una solución a este problema puede ser aplicar el mensaje SEI de anidación escalable de codificación de múltiples vistas como se especifica en el anexo H de H.264/AVC o similar. Sin embargo, el mensaje SEI de anidación escalable de codificación de múltiples vistas o mensajes SEI similares pueden tener las siguientes desventajas. En primer lugar, dado que las unidades NAL SEI en H.264/AVC solo tienen una cabecera de unidad NAL de un octeto, puede que no haya forma de usar la información contenida en `nuh_reserved_zero_6bits` y `temporal_id_plus1` en la cabecera de unidad NAL de la unidad NAL SEI HEVC para la asociación de un periodo de almacenamiento en memoria intermedia o un mensaje SEI de temporización de imágenes a los puntos de funcionamiento. En segundo lugar, cada mensaje SEI anidado solo puede asociarse a un punto de funcionamiento.

**[0103]** Una o más técnicas de esta divulgación pueden proporcionar un mecanismo para especificar claramente los puntos de funcionamiento a los que se aplica un mensaje SEI de periodo de almacenamiento en memoria intermedia, un mensaje SEI de temporización de imágenes o un mensaje SEI de temporización de subimágenes, a través de la estructura sintáctica `aplicable_operation_points()` que puede transportarse en un mensaje SEI de periodo de almacenamiento en memoria intermedia, un mensaje SEI de temporización de imágenes o un mensaje SEI de temporización de subimágenes. El mecanismo puede permitir el uso de la información transportada en los elementos sintácticos `nuh_reserved_zero_6bits` y `temporal_id_plus1` en la cabecera de unidad NAL de las unidades NAL SEI, y puede permitir la compartición de la información transmitida en un mismo periodo de almacenamiento en memoria intermedia, mensaje SEI de temporización de imágenes o de temporización de subimágenes mediante múltiples puntos de funcionamiento.

**[0104]** La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas de esta divulgación. La FIG. 2 se proporciona con propósitos explicativos y no se debería considerar limitativa de las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con propósitos explicativos, esta divulgación describe un codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden aplicarse a otras normas o procedimientos de codificación.

**[0105]** En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de procesamiento de predicción 100, una unidad de generación residual 102, una unidad de procesamiento de transformada 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de procesamiento de transformada inversa

110, una unidad de reconstrucción 112, una unidad de filtro 114, una memoria intermedia de imágenes descodificadas 116 y una unidad de codificación de entropía 118. La unidad de procesamiento de predicción 100 incluye una unidad de procesamiento de interpredicción 120 y una unidad de procesamiento de intrapredicción 126. La unidad de procesamiento de interpredicción 120 incluye una unidad de estimación de movimiento 122 y una  
 5 unidad de compensación de movimiento 124. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales.

**[0106]** El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede codificar cada CTU en un segmento de una imagen de los datos de vídeo. Cada una de las CTU puede estar asociada a unos bloques de árbol de codificación (CTB) de luma de igual tamaño y a unos CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 100 puede realizar una división de árbol cuaternario para dividir los CTB de la CTU en bloques progresivamente más pequeños. Los bloques más pequeños pueden ser bloques de codificación de unas CU. Por ejemplo, la unidad de procesamiento de predicción 100 puede dividir un CTB asociado a una CTU en cuatro subbloques de igual tamaño, dividir uno o  
 10 más de los subbloques en cuatro subsubbloques de igual tamaño, y así sucesivamente.

**[0107]** 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 100 puede dividir los bloques de codificación asociados a la CU entre una o más PU de la CU. Por tanto, cada PU puede estar asociada a un bloque de predicción de luma y a unos bloques de predicción de croma correspondientes. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir PU que tienen diversos tamaños. El tamaño de una CU se puede referir al tamaño del bloque de codificación de luma de la CU, y el tamaño de una PU se puede referir al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular es de  $2N \times 2N$ , el codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir tamaños de PU de  $2N \times 2N$  o  $N \times N$  para intrapredicción, y tamaños de PU simétricos de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ ,  $N \times N$  o similares para interpredicción. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden admitir una división asimétrica para tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$  para interpredicción.  
 20  
 25

**[0108]** La unidad de procesamiento de interpredicción 120 puede generar datos predictivos para una PU realizando una interpredicción 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 interpredicción 120 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un segmento I, un segmento P o un segmento B. En un segmento I, todas las PU se intrapredicen. Por consiguiente, si la PU está en un segmento I, la unidad de procesamiento de interpredicción 120 no realiza interpredicción en la PU. Por tanto, para bloques codificados en el modo I, el bloque predictivo se forma usando predicción espacial a partir de bloques vecinos previamente codificados dentro de la misma trama.  
 30  
 35

**[0109]** Si una PU está en un segmento P, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia en una lista de imágenes de referencia (por ejemplo, "RefPicListO") 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 muestras que se corresponden más estrechamente a los bloques de muestras de la PU. La unidad de estimación de movimiento 122 puede generar un índice de referencia que indique una posición en la RefPicListO de la imagen de referencia que contiene la región de referencia para la PU. Además, la unidad de estimación de movimiento 122 puede generar un vector de movimiento que indica un desplazamiento espacial entre un bloque de predicción de la PU y una ubicación de referencia asociada a la región de referencia. Por ejemplo, el vector de movimiento puede ser un vector bidimensional que proporciona una desviación desde las coordenadas en la imagen actual a las coordenadas en una imagen de referencia. La unidad de estimación de movimiento 122 puede proporcionar el índice de referencia y el vector de movimiento como información de movimiento de la PU. La unidad de compensación de movimiento 124 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.  
 40  
 45  
 50

**[0110]** Si una PU está en un fragmento B, la unidad de estimación de movimiento 122 puede llevar a cabo la predicción unidireccional o la predicción bidireccional para la PU. Para realizar predicción unidireccional para la PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia de RefPicListO o una segunda lista de imágenes de referencia ("RefPicList1") para una región de referencia para la PU. La unidad de estimación de movimiento 122 puede proporcionar, como la información de movimiento de la PU, un índice de referencia que indica una posición en RefPicListO 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 muestra de la PU y una ubicación de referencia asociada a la región de referencia, y uno o más indicadores de dirección de predicción que indican si la imagen de referencia está en RefPicListO o RefPicList1. La unidad de compensación de movimiento 124 puede generar los bloques predictivos de la PU basándose, al menos en parte, en muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.  
 55  
 60

**[0111]** Para realizar interpredicción bidireccional para una PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia en RefPicListO 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  
 65

movimiento 122 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 122 puede generar vectores de movimiento que indican desplazamientos espaciales entre la ubicación de referencia asociada a las regiones de referencia y un bloque de muestras 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 124 puede generar los bloques predictivos de la PU basándose, al menos en parte, en muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.

**[0112]** La unidad de procesamiento de intrapredicción 126 puede generar datos predictivos para una PU realizando una intrapredicción en la PU. Los datos predictivos para la PU pueden incluir bloques predictivos para la PU y diversos elementos sintácticos. La unidad de procesamiento de intrapredicción 126 puede realizar una intrapredicción en las PU en segmentos I, segmentos P y segmentos B.

**[0113]** Para realizar una intrapredicción en una PU, la unidad de procesamiento de intrapredicción 126 puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos predictivos para la PU. Para usar un modo de intrapredicción para generar un conjunto de datos predictivos para la PU, la unidad de procesamiento de intrapredicción 126 puede ampliar muestras de bloques de muestras de PU vecinas a través de los bloques de muestras de la PU en una dirección asociada al modo de intrapredicción. Las PU vecinas pueden estar encima, encima y a la derecha, encima y a la izquierda, o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha y de arriba abajo, para las PU, CU y CTU. La unidad de procesamiento de intrapredicción 126 puede usar diversos números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccional. En algunos ejemplos, el número de modos de intrapredicción puede depender del tamaño de la región asociada a la PU.

**[0114]** La unidad de procesamiento de predicción 100 puede seleccionar los datos predictivos para las PU de una CU de entre los datos predictivos generados por la unidad de procesamiento de interpretación 120 para las PU o los datos predictivos generados por la unidad de procesamiento de intrapredicción 126 para las PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos predictivos para las PU de la CU basándose en 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.

**[0115]** La unidad de generación residual 102 puede generar, basándose en el bloque de codificación de luma, Cb y Cr de una CU y los bloques predictivos seleccionados de luma, Cb y Cr de las PU de la CU, unos bloques residuales de luma, Cb y Cr de la CU. Por ejemplo, la unidad de generación residual 102 puede generar los bloques residuales de la CU de modo 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 bloque predictivo seleccionado correspondiente de una PU de la CU.

**[0116]** La unidad de procesamiento de transformada 104 puede realizar una división de árbol cuaternario para dividir los bloques residuales asociados a una CU en bloques de transformada asociados a unas TU de la CU. Por tanto, una TU puede estar asociada a un bloque de transformada de luma y a dos bloques de transformada de croma. Los tamaños y las posiciones de los bloques de transformada de luma y croma de unas TU de una CU pueden o no estar basados en los tamaños y las posiciones de los bloques de predicción de las PU de la CU. Una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados a cada una de las regiones. Las TU de una CU pueden corresponder a nodos hoja del RQT.

**[0117]** La unidad de procesamiento de transformada 104 puede generar bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a los bloques de transformada de la TU. La unidad de procesamiento de transformada 104 puede aplicar diversas transformadas a un bloque de transformada asociado a una TU. Por ejemplo, la unidad de procesamiento de transformada 104 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar al bloque de transformada. En algunos ejemplos, la unidad de procesamiento de transformada 104 no aplica transformadas a un bloque de transformada. En dichos ejemplos, el bloque de transformada se puede tratar como un bloque de coeficientes de transformada.

**[0118]** La unidad de cuantificación 106 puede cuantificar los coeficientes de transformada en un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes de transformada. Por ejemplo, un coeficiente de transformación de  $n$  bits puede redondearse hacia abajo hasta un coeficiente de transformada de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ . La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes asociado a una TU de una CU basándose en un valor de parámetro de cuantificación (QP) asociado a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados a una CU, ajustando el valor de QP asociado a la CU. La cuantificación puede introducir pérdida de información, por lo que los coeficientes de transformada cuantificados pueden tener una precisión menor que los originales.

**[0119]** La unidad de cuantificación inversa 108 y la unidad de procesamiento de transformada inversa 110 pueden aplicar una cuantificación inversa y transformadas inversas a un bloque de coeficientes, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficientes. La unidad de reconstrucción 112 puede añadir el bloque residual reconstruido a las muestras correspondientes de uno o más bloques predictivos generados por la unidad de procesamiento de predicción 100 para producir un bloque de transformada reconstruido asociado a una TU. Reconstruyendo bloques de transformada para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la CU.

**[0120]** La unidad de filtro 114 puede realizar una o más operaciones de eliminación de bloques para reducir los artefactos de bloque en los bloques de codificación asociados a una CU. La memoria intermedia de imágenes descodificadas 116 puede almacenar los bloques de codificación reconstruidos después de que la unidad de filtro 114 realice la una o más operaciones de eliminación de bloques en los bloques de codificación reconstruidos. La unidad de interpredicción 120 puede usar una imagen de referencia que contiene los bloques de codificación reconstruidos para realizar la interpredicción en las PU de otras imágenes. Además, la unidad de procesamiento de intrapredicción 126 puede usar bloques de codificación reconstruidos en la memoria intermedia de imágenes descodificadas 116 para realizar una intrapredicción en otras PU de la misma imagen que la CU.

**[0121]** La unidad de codificación de entropía 118 puede recibir datos desde otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación de entropía 118 puede recibir bloques de coeficientes desde la unidad de cuantificación 106 y puede recibir elementos sintácticos desde la unidad de procesamiento de predicción 100. La unidad de codificación de entropía 118 puede realizar una o más operaciones de codificación de entropía en los datos para generar datos sometidos a codificación de entropía. Por ejemplo, la unidad de codificación de entropía 118 puede realizar una operación de codificación de longitud variable adaptativa al contexto (CAVLC), una operación CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), una operación de codificación de entropía de división en intervalos de probabilidad (PIPE), una operación de codificación Golomb exponencial u otro tipo de operación de codificación de entropía en los datos. El codificador de vídeo 20 puede proporcionar un flujo de bits que incluye datos sometidos a codificación de entropía, generados por la unidad de codificación de entropía 118. Por ejemplo, el flujo de bits puede incluir datos que representan un RQT para una CU.

**[0122]** Como se indica en otra parte de esta divulgación, el codificador de vídeo 20 puede señalar un VPS en el flujo de bits. En el borrador de trabajo 8 de HEVC, elementos sintácticos particulares del SPV (es decir, `vps_max_dec_pic_buffering[i]`, `vps_max_num_reorder_pics[i]` y `vps_max_latency_increase[i]`) se definen con referencia a un valor `HighestTid`, que no está definido. De acuerdo con una o más técnicas de esta divulgación, estos elementos sintácticos del VPS pueden definirse con referencia a un valor `TargetDecHighestTid`, que se define como el `TargetDecHighestTid` descrito en otra parte de esta divulgación. La siguiente Tabla 3 ilustra una sintaxis de un VPS de acuerdo con una o más técnicas de esta divulgación.

**TABLA 3 - VPS**

	<b>Descriptor</b>
<code>video_parameter_set_rbsp( ) {</code>	
<b>video_parameter_set_id</b>	u(4)
<b>vps_temporal_id_nesting_flag</b>	u(1)
<b>vps_reserved_zero_2bits</b>	u(2)
<b>vpa_reserved_zero_6bits</b>	u(6)
<b>vps_max_sub_layers_minus1</b>	u(3)
profile_tier_level (1, vps_max_sub_layers_minus1)	
<b>vps_reserved_zero_12bits</b>	u(12)
for(i = 0; i <= vps_max_sub_layers_minus1; i++) {	
<b>vps_max_dec_pic_buffering[i]</b>	ue(v)
<b>vps_max_num_reorder_pics[i]</b>	ue(v)
<b>vps_max_latency_increase[i]</b>	ue(v)
}	
<b>vps_num_hrd_parameters</b>	ue(v)
for(i = 0; i < vps_num_hrd_parameters; i++) {	
operation_point_layer_ids(i)	

	Descriptor
video_parameter_set_rbsp( ) {	
hrd_parameters (i = 0, vps_max_sub_layers_minus1)	
}	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data( ))	
<b>vps_extension_data_flag</b>	u(1)
}	
rbbsp_trailing_bits( )	
}	

**[0123]** Las partes en cursiva de la Tabla 3 y otras tablas sintácticas o descripciones semánticas a lo largo de esta divulgación pueden indicar diferencias con respecto al borrador de trabajo 8 de HEVC. De acuerdo con una o más técnicas de esta divulgación, la semántica para los siguientes elementos sintácticos del VPS puede cambiarse de la siguiente manera. La semántica de otros elementos sintácticos del VPS puede permanecer igual que en el borrador de trabajo 8 de HEVC.

**vps\_max\_dec\_pic\_buffering[i]** especifica el tamaño requerido de la memoria intermedia de imágenes descodificadas en unidades de memorias intermedias de almacenamiento de imágenes cuando *TargetDecHighestTid* es igual a *i*. El valor de *vps\_max\_dec\_pic\_buffering[i]* estará en el intervalo de 0 a *MaxDpbSize* (como se especifica en la subcláusula A.4), ambos inclusive. Cuando *i* es mayor que 0, *vps\_max\_dec\_pic\_buffering[i]* será igual a o mayor que *vps\_max\_dec\_pic\_buffering[i - 1]*.

**vps\_max\_num\_reorder\_pics[i]** indica el número máximo permitido de imágenes que preceden a cualquier imagen en orden de descodificación y que suceden a esa imagen en orden de salida cuando *TargetDecHighestTid* es igual a *i*. El valor de *vps\_max\_num\_reorder\_pics[i]* estará en el intervalo de 0 a *vps\_max\_dec\_pic\_buffering[i]*, ambos inclusive. Cuando *i* es mayor que 0, *vps\_max\_num\_reorder\_pics[i]* será igual a o mayor que *vps\_max\_num\_reorder\_pics[i - 1]*.

**vps\_max\_latency\_increase[i]** no igual a 0 se usa para calcular el valor de *MaxLatencyPictures[i]* como se especifica al establecer *MaxLatencyPictures[i]* como *vps\_max\_num\_reorder\_pics[i] + vps\_max\_latency\_increase[i]*. Cuando *vps\_max\_latency\_increase[i]* no es igual a 0, el valor de *MaxLatencyPictures[i]* especifica el número máximo de imágenes que pueden preceder a cualquier imagen en la secuencia de vídeo codificada en orden de salida y seguir esa imagen en orden de descodificación cuando *TargetDecHighestTid* es igual a *i*. Cuando *vps\_max\_latency\_increase[i]* es igual a 0, no se expresa ningún límite correspondiente. El valor de *vps\_max\_latency\_increase[i]* estará en el intervalo de 0 a  $2^{32} - 2$ , ambos inclusive.

**[0124]** Como se muestra anteriormente, la semántica de *vps\_max\_dec\_buffering[i]*, *vps\_max\_num\_reorder\_pics[i]* y *vps\_max\_latency\_increase[i]* puede definirse con referencia a *TargetDecHighestTid*. Por el contrario, el borrador de trabajo 8 de HEVC define *vps\_max\_dec\_pic\_buffering[i]*, *vps\_max\_num\_reorder\_pics[i]* y *vps\_max\_latency\_increase[i]* con referencia a *HighestTid*, donde *HighestTid* no está definido.

**[0125]** Como se muestra en la sintaxis de ejemplo de la Tabla 3, el VPS incluye pares de estructuras sintácticas *operation\_point\_layer\_ids( )* y de estructuras sintácticas *hrd\_parameters( )*. Las estructuras sintácticas *hrd\_parameters( )* incluyen elementos sintácticos que especifican conjuntos de parámetros HRD. Una estructura sintáctica *operation\_point\_layer\_ids( )* incluye elementos sintácticos que identifican un conjunto de puntos de funcionamiento. El conjunto de parámetros HRD especificados en una estructura sintáctica *hrd\_parameters( )* puede aplicarse a los puntos de funcionamiento identificados por los elementos sintácticos en la estructura sintáctica *operation\_point\_layer\_ids( )* correspondiente. La siguiente Tabla 4 muestra una sintaxis de ejemplo de una estructura sintáctica *operation\_point\_layer\_ids( )*.

**TABLA 4 - ID de capa de punto de funcionamiento**

	Descriptor
<i>operation_point_layer_ids(opIdx) {</i>	
<b>op_num_layer_id_values_minus1[opIdx]</b>	ue(v)
for(i = 0; i <= op_num_layer_id_values_minus1[opIdx]; i++)	
<b>op_layer_id[opIdx][i]</b>	u(6)

operation_point_layer_ids(oplidx) {	Descriptor
}	

[0126] La sección 7.4.4 del borrador de trabajo 8 de HEVC describe la semántica de una estructura sintáctica op\_point. De acuerdo con la una o más técnicas de esta divulgación, la sección 7.4.4 del borrador de trabajo 8 de HEVC se puede cambiar de la siguiente manera para proporcionar semántica para la estructura sintáctica operation\_point\_layer\_ids( ) de la Tabla 4.

La estructura sintáctica operation\_layer\_ids(oplidx) especifica el conjunto de valores nuh\_reserved\_zero\_6bits incluidos en OpLayerIdSet de los puntos de funcionamiento a los que se aplica la opldx-ésima estructura sintáctica hrd\_parameters( ) en el conjunto de parámetros de vídeo.

op\_num\_layer\_id\_values\_minus1[oplidx] más 1 especifica el número de valores nuh\_reserved\_zero\_6bits incluidos en OpLayerIdSet de los puntos de funcionamiento a los que se aplica la opldx-ésima estructura sintáctica hrd\_parameters( ) en el conjunto de parámetros de vídeo. op\_num\_layer\_id\_values\_minus1[oplidx] será menor que o igual a 63. En los flujos de bits que se ajustan a esta memoria descriptiva, op\_num\_layer\_id\_values\_minus1[oplidx] será igual a 0. Aunque se requiere que el valor de op\_num\_layer\_id\_values\_minus1[oplidx] sea igual a 0 en esta versión de esta memoria descriptiva, los descodificadores permitirán que aparezcan otros valores en la sintaxis op\_numlayer\_id\_values\_minus1[oplidx].

op\_layer\_id[oplidx][i] especifica el i-ésimo valor de nuh\_reserved\_zero\_6bits incluido en OpLayerIdSet de los puntos de funcionamiento a los que se aplica la opldx-ésima estructura sintáctica hrd\_parameters( ) en el conjunto de parámetros de vídeo. Ningún valor de op\_layer\_id[oplidx][i] será igual a op\_layer\_id[oplidx][j] cuando i no es igual a j y tanto i como j están en el intervalo de 0 a op\_num\_layer\_id\_values\_minus1, ambos inclusive. Se infiere que op\_layer\_id[0][0] es igual a 0.

[0127] Como se indicó anteriormente, el elemento sintáctico op\_num\_layer\_id\_values\_minus1[oplidx], más 1, especifica el número de valores nuh\_reserved\_zero\_6bits incluidos en OpLayerIdSet de los puntos de funcionamiento a los que se aplica la estructura sintáctica opldx-th hrd\_parameters( ) en el conjunto de parámetros de vídeo. Por el contrario, el borrador de trabajo 8 de HEVC establece que el elemento sintáctico op\_num\_layer\_id\_values\_minus1[oplidx], más 1, especifica el número de valores nuh\_reserved\_zero\_6bits incluidos en el punto de funcionamiento identificado por opldx. De manera similar, en el ejemplo de la Tabla 4, el elemento sintáctico op\_layer\_id[oplidx][i] especifica el i-ésimo valor de nuh\_reserved\_zero\_6bits incluido en el OpLayerIdSet de los puntos de funcionamiento a los que se aplica la estructura sintáctica opldx-th hrd\_parameters( ) en el conjunto de parámetros de vídeo. Por el contrario, el borrador de trabajo 8 de HEVC establece que el elemento sintáctico op\_layer\_id[oplidx][i] especifica el i-ésimo valor de nuh\_reserved\_zero\_6bits incluido en el punto de funcionamiento identificado por opldx.

[0128] La sección 7.4.2.2 del borrador de trabajo 8 de HEVC describe la semántica para el SPS. De acuerdo con una o más técnicas de esta divulgación, se pueden hacer los siguientes cambios en la sección 7.4.2.2 del borrador de trabajo 8 de HEVC. La semántica de otros elementos sintácticos de SPS puede ser la misma que en el borrador de trabajo 8 de HEVC:

sps\_max\_dec\_pic\_buffering[i] especifica el tamaño máximo requerido de la memoria intermedia de imágenes descodificadas en unidades de memorias intermedias de almacenamiento de imágenes cuando TargetDecHighestTid es igual a i. El valor de sps\_max\_dec\_pic\_buffering[i] estará en el intervalo de 0 a MaxDpbSize (como se especifica en la subcláusula A.4), ambos inclusive. Cuando i es mayor que 0, sps\_max\_dec\_pic\_buffering[i] será igual a o mayor que sps\_max\_dec\_pic\_buffering[i - 1]. El valor de sps\_max\_dec\_pic\_buffering[i] será menor que o igual a vps\_max\_dec\_pic\_buffering[i] para cada valor de i.

sps\_max\_num\_reorder\_pics[i] indica el número máximo permitido de imágenes que preceden a cualquier imagen en orden de descodificación y que suceden a esa imagen en orden de salida cuando TargetDecHighestTid es igual a i. El valor de sps\_max\_num\_reorder\_pics[i] estará en el intervalo de 0 a sps\_max\_dec\_pic\_buffering[i], ambos inclusive. Cuando i es mayor que 0, sps\_max\_num\_reorder\_pics[i] será igual a o mayor que sps\_max\_num\_reorder\_pics[i - 1]. El valor de sps\_max\_numreorder\_pics[i] será menor que o igual a vps\_max\_num\_reorder\_pics[i] para cada valor de i.

sps\_max\_latency\_increase[i] no igual a 0 se usa para calcular el valor de MaxLatencyPictures[i] como se especifica al establecer MaxLatencyPictures[i] igual a sps\_max\_num\_reorder\_pics[i] + sps\_max\_latency\_increase[i]. Cuando sps\_max\_latency\_increase[i] no es igual a 0, el valor de MaxLatencyPictures[i] especifica el número máximo de imágenes que pueden preceder a cualquier imagen en la secuencia de vídeo codificada en orden de salida y seguir esa imagen en orden de descodificación cuando TargetDecHighestTid es igual a i. Cuando sps\_max\_latency\_increase[i] es igual a 0, no se expresa ningún límite

correspondiente. El valor de `sps_max_latency_increase[i]` estará en el intervalo de 0 a  $2^{32} - 2$ , ambos inclusive. El valor de `sps_max_latency_increase[i]` será menor que o igual a `vps_max_latency_increase[i]` para cada valor de `i`.

5 **[0129]** Como se muestra anteriormente, la semántica de `sps_max_dec_pic_buffering[i]`, `sps_max_num_reorder_pics[i]` y `sps_max_latency_increase[i]` se definen en lo que respecta a `TargetDecHighestTid`. `TargetDecHighestTid` se determina como se describe en otra parte de esta divulgación. Por el contrario, el borrador de trabajo 8 de HEVC define la semántica de `sps_max_dec_pic_buffering[i]`, `sps_max_num_reorder_pics[i]` y `sps_max_latency_increase[i]` con referencia a `HighestTid`, que no está definido.

10 **[0130]** La sección 7.4.5.1 del borrador de trabajo 8 de HEVC describe la semántica general de la cabecera de segmento. De acuerdo con una o más técnicas de esta divulgación, se pueden hacer los siguientes cambios en la sección 7.4.5.1 del borrador de trabajo 8 de HEVC. Otras partes de la sección 7.4.5.1 del borrador de trabajo 8 de HEVC pueden seguir siendo las mismas.

15 **no\_output\_of\_prior\_pics\_flag** especifica cómo se tratan las imágenes previamente descodificadas en la memoria intermedia de imágenes descodificadas después de descodificar una imagen IDR o BLA. Consúltense el Anexo C. Cuando la imagen actual es una imagen CRA, o la imagen actual es una imagen IDR o BLA que es la primera imagen en el flujo de bits, el valor de `no_output_of_prior_pics_flag` no tiene efecto en el proceso de descodificación. Cuando la imagen actual es una imagen IDR o BLA que no es la imagen en el flujo de bits, y el valor de `pic_width_in_luma_samples` o `pic_height_in_luma_samples` o

20 `sps_max_dec_pic_buffering[TargetDecHighestTid]` obtenido del conjunto de parámetros de secuencia activo es diferente del valor de `pic_width_in_luma_samples` o `pic_height_in_luma_samples` o

25 `sps_max_dec_pic_buffering[TargetDecHighestTid]` obtenido del conjunto de parámetros de secuencia activo para la imagen anterior, `no_output_of_prior_pics_flag` igual a 1 puede (pero no debe) deducirse por el descodificador, independientemente del valor real de `no_output_of_prior_pics_flag`.

30 **[0131]** Como se muestra anteriormente, la semántica de `no_output_of_prior_pics_flag` se define con referencia a `sps_max_dec_pic_buffering[TargetDecHighestTid]`. `TargetDecHighestTid` se determina como se describe en otra parte de esta divulgación. Por el contrario, el borrador de trabajo 8 de HEVC define la semántica de

35 `no_output_of_prior_pics_flags` con referencia a

`sps_max_dec_pic_buffering[HighestTid]`, donde `HighestTid` no está definido.

40 **[0132]** La sección 8.1 del borrador de trabajo 8 de HEVC describe un proceso de descodificación general. De acuerdo con una o más técnicas de esta divulgación, el proceso de descodificación general del borrador de trabajo 8 de HEVC se puede cambiar como sigue.

*La entrada de este proceso es un flujo de bits y la salida es una lista de imágenes descodificadas.*

45 *El conjunto `TargetDecLayerIdSet`, que especifica el conjunto de valores para `nuh_reserved_zero_6bits` de unidades NAL VCL a descodificar, se especifica de la siguiente manera:*

- *Si algunos medios externos no especificados en esta memoria descriptiva están disponibles para establecer `TargetDecLayerIdList`, `TargetDecLayerIdList` se establece mediante los medios externos.*

50 - *De lo contrario, si el proceso de descodificación se invoca en una prueba de conformidad de flujo de bits como se especifica en la subcláusula C.1, `TargetDecLayerIdList` se establece como se especifica en la subcláusula C.1.*

55 - *De lo contrario, `TargetDecLayerIdSet` contiene solo un valor para `nuh_reserved_zero_6bits`, que es igual a 0.*

*La variable `TargetDecHighestTid`, que identifica la subcapa temporal más alta a descodificar, se especifica de la siguiente manera:*

60 - *Si algunos medios externos no especificados en esta memoria descriptiva están disponibles para establecer `TargetDecHighestTid`, `TargetDecHighestTid` se establece mediante los medios externos.*

- *De lo contrario, si el proceso de descodificación se invoca en una prueba de conformidad de flujo de bits como se especifica en la subcláusula C.1, `TargetDecHighestTid` se establece como se especifica en la subcláusula C.1.*

65



- De lo contrario, *TargetDecHighestTid* se establece como *sps\_max\_sub\_layers\_minus1*.

El proceso de extracción de subflujo de bits como se especifica en la subcláusula 10.1 se aplica con *TargetDecHighestTid* y *TargetDecLayerIdSet* como entradas, y la salida se asigna a un flujo de bits denominado *BitstreamToDecode*.

Lo siguiente se aplica a cada imagen codificada (denominada imagen actual, que se denota con la variable *CurrPic*) en *BitstreamToDecode*.

**[0133]** Dependiendo del valor de *chroma\_format\_idc*, el número de matrices de muestras de la imagen actual es la siguiente.

- Si *chroma\_format\_idc* es igual a 0, la imagen actual consiste en 1 matriz de muestras  $S_L$ .

- De lo contrario (*chroma\_format\_idc* no es igual a 0), la imagen actual consiste en 3 matrices de muestras  $S_L$ ,  $S_{Cb}$ ,  $S_{Cr}$ .

El proceso de descodificación para la imagen actual toma como entradas los elementos sintácticos y variables en mayúsculas de la cláusula 7. Cuando se interpreta la semántica de cada elemento sintáctico en cada unidad NAL y "el flujo de bits" o parte del mismo (por ejemplo, una secuencia de vídeo codificada) está involucrado, el flujo de bits o parte del mismo significa *BitstreamToDecode* o parte del mismo.

**[0134]** El proceso de descodificación se especifica de modo que todos los descodificadores producirán resultados idénticos numéricamente. Cualquier proceso de descodificación que produzca resultados idénticos al proceso aquí descrito en el presente documento se ajusta a los requisitos de proceso de descodificación de esta memoria descriptiva.

**[0135]** Cuando la imagen actual es una imagen CRA, se aplica lo siguiente:

- Si algunos medios externos no especificados en esta memoria intermedia están disponibles para establecer la variable *HandleCraAsBlaFlag* a un valor, *HandleCraAsBlaFlag* se establece en el valor proporcionado por los medios externos.

- De lo contrario, el valor de *HandleCraAsBlaFlag* se establece en 0.

**[0136]** Cuando la imagen actual es una imagen CRA y *HandleCraAsBlaFlag* es igual 1, se aplica lo siguiente durante los procesos de análisis sintáctico y descodificación para cada unidad NAL de fragmento codificado:

- El valor de *nal\_unit\_type* se establece en *BLA\_W\_LP*.

- El valor de *no\_output\_of\_prior\_pics\_flag* se establece en 1.

NOTA 1: las implementaciones de descodificador pueden elegir establecer el valor de *no\_output\_of\_prior\_pics\_flag* a 0 cuando la configuración no afecta a la descodificación de la imagen actual y de las imágenes siguientes en orden de descodificación, por ejemplo, cuando siempre hay una memoria intermedia de almacenamiento de imágenes disponible cuando es necesario.

**[0137]** Cada imagen a la que se hace referencia en esta cláusula es una imagen codificada completa.

**[0138]** Dependiendo del valor de *separate\_colour\_plane\_flag*, el proceso de descodificación se estructura como sigue.

- Si *separate\_colour\_plane\_flag* es igual a 0, el proceso de descodificación se invoca una sola vez siendo la imagen actual la salida.

- De lo contrario (*separate\_colour\_plane\_flag* es igual a 1), el proceso de descodificación se invoca tres veces. Las entradas al proceso de descodificación son todas las unidades NAL de la imagen codificada con el valor idéntico de *colour\_plane\_id*. El proceso de descodificación de unidades NAL con un valor particular de *colour\_plane\_id* se especifica como si solo una secuencia de vídeo codificada con formato de color monocromo con ese valor particular de *colour\_plane\_id* estuviera presente en el flujo de bits. La salida de cada uno de los tres procesos de descodificación se asigna a las 3 matrices de muestras de la imagen actual con las unidades NAL con *colour\_plane\_id* igual a 0 asignadas a  $S_L$ , las unidades NAL con *colour\_plane\_id* igual a 1 asignadas a  $S_{Cb}$ , y las unidades NAL con *colour\_plane\_id* igual a 2 asignadas a  $S_{Cr}$ .

NOTA 1: La variable ChromaArrayType se obtiene como 0 cuando separate\_colour\_plane\_flag es igual a 1 y chroma\_format\_idc es igual a 3. En el proceso de descodificación, el valor de esta variable se evalúa, dando como resultado operaciones idénticas a las de las imágenes monocromo con chroma\_format\_idc igual a 0.

5 **[0139]** El proceso de descodificación funciona como sigue para la imagen actual CurrPic:

1. La descodificación de unidades NAL se especifica en la subcláusula 8.2.

10 2. Los procesos de la subcláusula 8.3 especifican procesos de descodificación que usan elementos sintácticos en la capa de segmento y superiores:

- Las variables y funciones relacionadas con el recuento de orden de imágenes se obtienen en la subcláusula 8.3.1 (que solo debe invocarse para el primer segmento de una imagen).

15 - Se invoca el proceso de descodificación para el conjunto de imágenes de referencia de la subcláusula 8.3.2, en el que las imágenes de referencia pueden marcarse como "no usadas como referencia" o "usadas como referencia a largo plazo" (que solo necesita invocarse para el primer segmento de una imagen).

20 - Cuando la imagen actual es una imagen BLA o es una imagen CRA que es la primera imagen en el flujo de bits, se invoca el proceso de descodificación para generar imágenes de referencia no disponibles especificadas en la subcláusula 8.3.3 (que solo necesita invocarse para el primer segmento de una imagen).

- PicOutputFlag se establece como sigue:

25 - Si la imagen actual es una imagen TFD y la imagen RAP anterior en orden de descodificación es una imagen BLA o es una imagen CRA que es la primera imagen codificada en el flujo de bits, PicOutputFlag se establece igual a 0.

30 - De lo contrario, PicOutputFlag se establece igual a pic\_output\_flag.

- Al comienzo del proceso de descodificación para cada segmento P o B, se invoca el proceso de descodificación para la construcción de listas de imágenes de referencia especificadas en la subcláusula 8.3.4 para la obtención de la lista de imágenes de referencia 0 (RefPicListO), y cuando se descodifica un segmento B, la lista de imágenes de referencia 1 (RefPicList1).

35 - Después de que se hayan descodificado todos los segmentos de la imagen actual, la imagen descodificada se marca como "usada como referencia a corto plazo".

40 3. Los procesos en las subcláusulas 8.4, 8.5, 8.6 y 8.7 especifican procesos de descodificación que usan elementos sintácticos en la capa de unidad de árbol de codificación y superiores.

45 **[0140]** Como se indica en otra parte de esta divulgación, en el borrador de trabajo 8 de HEVC, cuando se invoca el proceso de descodificación para la prueba de conformidad de flujo de bits, la semántica de los elementos sintácticos no se especifica claramente, ya que los valores de TargetDecLayerIdSet y TargetDecHighestTid no se establecen apropiadamente. Las modificaciones mostradas anteriormente en el proceso de descodificación general pueden solucionar este problema. Como se muestra anteriormente, cuando se invoca el proceso de descodificación general para una prueba de conformidad de flujo de bits, los valores de TargetDecLayerIdSet y TargetDecHighestTid se establecen como se especifica en la subcláusula C.1. Como se describe a continuación, una versión modificada de la subcláusula C.1 puede establecer TargetDecLayerIdSet como el conjunto de valores nuh\_reserved\_zero\_6bits presentes en el subflujo de bits correspondiente al punto de funcionamiento bajo prueba. La versión modificada de la subcláusula C.1 puede establecer TargetDecHighestTid como el mayor valor de TemporalId presente en el subflujo de bits correspondiente al punto de funcionamiento bajo prueba.

55 **[0141]** De esta manera, un dispositivo, tal como el descodificador de vídeo 30, puede realizar un proceso de descodificación como parte de la realización de una prueba de conformidad de flujo de bits. La realización del proceso de descodificación puede comprender realizar un proceso de extracción de flujo de bits para extraer, de un flujo de bits, una representación de punto de funcionamiento de un punto de funcionamiento definido por un conjunto objetivo de identificadores de capa y el identificador temporal objetivo más alto. El conjunto objetivo de identificadores de capa puede contener valores de elementos sintácticos de identificador de capa presentes en la representación de punto de funcionamiento, siendo el conjunto objetivo de identificadores de capa un subconjunto de valores de elementos sintácticos de identificador de capa del flujo de bits. El identificador temporal objetivo más alto puede ser igual al mayor identificador temporal presente en la representación de punto de funcionamiento, siendo el identificador temporal objetivo más alto menor que o igual al mayor identificador temporal presente en el flujo de bits. Además, el dispositivo puede descodificar unidades NAL de la representación de punto de funcionamiento.

60

65

**[0142]** Como se indica en las modificaciones de la sección 8.1 anterior, el proceso de descodificación no se realiza necesariamente como parte de la prueba de conformidad de flujo de bits. En algunos casos en los que el proceso de descodificación no se realiza como parte de una prueba de conformidad de flujo de bits, un dispositivo puede realizar el proceso de extracción de flujo de bits para extraer, del flujo de bits, una representación de punto de funcionamiento de un punto de funcionamiento. En este caso, 0 puede ser el único valor de los elementos sintácticos de identificador de capa (por ejemplo, `nuh_reserved_zero_6bits`) presentes en la representación de punto de funcionamiento, y el mayor identificador temporal presente en el flujo de bits es igual al mayor identificador temporal presente en la representación de punto de funcionamiento del punto de funcionamiento. El dispositivo puede descodificar unidades NAL de la representación de punto de funcionamiento del segundo punto de funcionamiento.

**[0143]** De forma alternativa, el dispositivo puede recibir, desde una fuente externa, un conjunto objetivo de identificadores de capa y el identificador temporal objetivo más alto. El conjunto objetivo de identificadores de capa puede contener valores de elementos sintácticos de identificador de capa presentes en una representación de punto de funcionamiento de un punto operativo que está definido por el conjunto objetivo de identificadores de capa y el identificador temporal objetivo más alto. El identificador temporal objetivo más alto puede ser igual al mayor identificador temporal presente en la representación de punto de funcionamiento del punto de funcionamiento. Además, el dispositivo puede realizar el proceso de extracción de flujo de bits para extraer, del flujo de bits, la representación de punto de funcionamiento del punto de funcionamiento. Además, el dispositivo puede descodificar unidades NAL de la representación de punto de funcionamiento del punto de funcionamiento.

**[0144]** Además, de acuerdo con una o más técnicas de esta divulgación, el proceso de extracción de subflujo de bits descrito en la subcláusula 10.1 del borrador de trabajo 8 de HEVC se puede cambiar de la siguiente manera.

Un requisito de conformidad con el flujo de bits es que cualquier subflujo de bits que esté incluido en la salida del proceso especificado en esta subcláusula con `tIdTarget` igual a cualquier valor en el intervalo de 0 a 6, ambos inclusive, y con `layerIdSetTarget` que solo contiene el valor 0 se ajusten a esta memoria descriptiva.

NOTA: Un flujo de bits en conformidad contiene una o más unidades NAL de segmento codificado con `nuh_reserved_zero_6bits` igual a 0 y `TemporalId` igual a 0.

Las entradas a este proceso son una variable `tIdTarget` y un conjunto `layerIdSetTarget`.

La salida de este proceso es un subflujo de bits.

El subflujo de bits se obtiene eliminando del flujo de bits todas las unidades NAL con `TemporalId` mayor que `tIdTarget` o `nuh_reserved_zero_6bits` no entre los valores en `layerIdSetTarget`.

**[0145]** En la subcláusula 10.1 del borrador de trabajo 8 de HEVC, el nombre de la variable `targetDecLayerIdSet` se usa donde se ha usado `layerIdSetTarget` anteriormente. Los cambios mostrados anteriormente en la subcláusula 10.1 del borrador de trabajo de HEVC para usar `layerIdSetTarget` pueden servir para aclarar que puede haber una distinción entre el conjunto de identificadores de capa usados en el proceso de extracción de subflujo de bits y `targetDecLayerIdSet`, que, como se describe en otra parte de esta divulgación, tiene una definición particular.

**[0146]** Además, de acuerdo con una o más técnicas de esta divulgación, las especificaciones generales de grado y nivel de la sección A.4.1 del borrador de trabajo 8 de HEVC pueden modificarse de la siguiente manera. En esta divulgación, un "perfil" puede referirse a un subconjunto de la sintaxis de flujo de bits. Dentro de cada perfil se pueden especificar "grados" y "niveles". Un nivel de un grado puede ser un conjunto especificado de restricciones impuestas en valores de los elementos sintácticos en el flujo de bits. Estas restricciones pueden ser simples límites en los valores.

**[0147]** De forma alternativa, las restricciones pueden adoptar la forma de restricciones en combinaciones aritméticas de valores (por ejemplo, el ancho de imagen multiplicado por la altura de imagen multiplicado por el número de imágenes descodificadas por segundo). Un nivel especificado para un grado inferior está más restringido que un nivel especificado para un grado superior. De acuerdo con un ejemplo de esta divulgación, la sección "general level specifications" (es decir, la sección A.4.1) del borrador de trabajo 8 de HEVC se renombra como "General tier and level specifications", y el texto cambia de la siguiente manera. La Tabla A-1 puede permanecer igual que en el borrador de trabajo 8 de HEVC.

Con el fin de comparar las capacidades de grado, el grado con `general_tier_flag` igual a 0 se considerará un grado más bajo que el grado con `general_tier_flag` igual a 1.

Para propósitos de comparación de capacidades de nivel, para un grado específico, un nivel inferior tiene un valor más bajo de `general_level_idc`.

Se especifica lo siguiente para expresar las restricciones en este anexo.

- Dejar que la unidad de acceso  $n$  sea la  $n$ -ésima unidad de acceso en orden de descodificación, siendo la primera unidad de acceso la unidad de acceso 0 (es decir, la unidad de acceso 0-ésima).

- Dejar que la imagen  $n$  sea la imagen codificada o la imagen descodificada correspondiente de la unidad de acceso  $n$ .

- Dejar que la variable  $fR$  se establezca igual a  $1 \div 300$ .

Los flujos de bits que se ajustan a un perfil en un nivel especificado deberán cumplir las siguientes restricciones para cada prueba de conformidad de flujo de bits como se especifica en el Anexo C:

a) El tiempo de eliminación nominal de la unidad de acceso  $n$  (con  $n > 0$ ) de la CPB como se especifica en la subcláusula C.2.2 satisface la restricción de que  $t_{r,n}(n) - t_{r,n}(n-1)$  es igual a o mayor que  $\text{Max}(\text{PicSizeInSamplesY} \div \text{MaxLumaSR}, fR)$  para el valor de  $\text{PicSizeInSamplesY}$  de la imagen  $n-1$ , donde  $\text{MaxLumaSR}$  es el valor especificado en la Tabla A-1 que se aplica a la imagen  $n-1$ .

b) La diferencia entre tiempos de salida consecutivos de imágenes de la DPB como se especifica en la subcláusula C.3.2 satisface la restricción de que  $\Delta t_{o,dpb}(n) \geq \text{Max}(\text{PicSizeInSamplesY} \div \text{MaxLumaSR}, fR)$  para el valor de  $\text{PicSizeInSamplesY}$  de la imagen  $n$ , donde  $\text{MaxLumaSR}$  es el valor especificado en la Tabla A-1 para la imagen  $n$ , siempre que la imagen  $n$  sea una imagen que se proporcione y no sea la última imagen del flujo de bits que se proporcione.

c)  $\text{PicSizeInSamplesY} \leq \text{MaxLumaPS}$ , donde  $\text{MaxLumaPS}$  se especifica en la Tabla A-1.

d)  $\text{pic\_width\_in\_luma\_samples} \leq \text{Sqrt}(\text{MaxLumaPS} * 8)$

e)  $\text{pic\_height\_in\_luma\_samples} \leq \text{Sqrt}(\text{MaxLumaPS} * 8)$

f)  $\text{sps\_max\_dec\_pic\_buffering}[\text{TargetDecHighestTid}] \leq \text{MaxDpbSize}$ , donde  $\text{MaxDpbSize}$  se obtiene como se especifica a continuación:

if ( $\text{PicSizeInSamplesY} \leq (\text{MaxLumaPS} \gg 2)$ )

$\text{MaxDpbSize} = \text{Min}(4 * \text{MaxDpbPicBuf}, 16)$

else if ( $\text{PicSizeInSamplesY} \leq (\text{MaxLumaPS} \gg 1)$ )

$\text{MaxDpbSize} = \text{Min}(2 * \text{MaxDpbPicBuf}, 16)$

else if ( $\text{PicSizeInSamplesY} \leq (\text{MaxLumaPS} \ll 1) / 3$ )

$\text{MaxDpbSize} = \text{Min}((3 * \text{MaxDpbPicBuf}) \gg 1, 16)$

else if ( $\text{PicSizeInSamplesY} \leq ((3 * \text{MaxLumaPS}) \gg 2)$ )

$\text{MaxDpbSize} = \text{Min}(4 * \text{MaxDpbPicBuf} / 3, 16)$

else

$\text{MaxDpbSize} = \text{MaxDpbPicBuf}$

donde  $\text{MaxLumaPS}$  se especifica en la Tabla A-1 y  $\text{MaxDpbPicBuf}$  es igual a 6.

La Tabla A-1 especifica los límites para cada nivel de cada grado. El uso de la columna de parámetros  $\text{MinCR}$  de la Tabla A-1 se especifica en la subcláusula A.4.2.

Un grado y nivel al que se ajusta el flujo de bits se indicarán mediante los elementos sintácticos  $\text{general\_tier\_flag}$  y  $\text{general\_level\_idc}$  como sigue.

-  $\text{general\_tier\_flag}$  igual a 0 indica conformidad con el grado principal, y  $\text{general\_tier\_flag}$  igual a 1 indica conformidad con el grado alto, de acuerdo con las especificaciones de restricciones de grado en la Tabla A-1.  $\text{general\_tier\_flag}$  será igual a 0 para niveles inferiores al nivel 4 (correspondiente a las entradas en la Tabla A-1

marcadas con "-"). Los límites de nivel distintos de MaxBR y MaxCPB en la Tabla A-1 son comunes tanto para el grado principal como para el grado alto.

- `general_level_idc` se establecerá igual a un valor de 30 veces el número de nivel especificado en la Tabla A-1.

**[0148]** Como se indica en el elemento (f) anterior, los flujos de bits que se ajustan a un perfil en un nivel específico obedecen la restricción `sps_max_dec_pic_buffering [TargetDecHighestTid] <= MaxDpbSize`. `TargetDecHighestTid` puede definirse de la manera descrita en otra parte de esta divulgación. Por el contrario, el borrador de trabajo 8 de HEVC indica para el elemento (f) que los flujos de bits que se ajustan a un perfil en un nivel especificado obedecen la restricción `sps_max_dec_pic_buffering[sps_max_temporal_layers_minus1] <= MaxDpbSize`. Como se indica en otra parte de esta divulgación, los parámetros `sps_max_dec_pic_buffering[i]` pueden no seleccionarse apropiadamente en las restricciones de nivel. Reemplazar `sps_max_temporal_layers_minus1` por `TargetDecHighestTid` como el índice `i` de `sps_max_dec_pic_buffering[i]` puede, de acuerdo con una o más técnicas de esta divulgación, garantizar que las restricciones de nivel se seleccionen de manera coherente con `i` igual al valor claramente especificado de `TargetDecHighestTid`.

**[0149]** De esta manera, un proceso de descodificación de un HRD puede descodificar, a partir de un SPS, una matriz de elementos sintácticos (por ejemplo, `sps_max_dec_pic_buffering[ ]`) donde cada uno de los elementos sintácticos en la matriz indica un tamaño máximo requerido de una DPB del HRD. Además, cuando un dispositivo realiza una operación HRD, el dispositivo puede determinar, basándose en el identificador temporal objetivo más alto (por ejemplo, `TargetDecHighestTid`), un elemento sintáctico particular en la matriz (por ejemplo, `sps_max_dec_pic_buffering[TargetDecHighestTid]`). Además, el dispositivo puede determinar que el flujo de bits no está en conformidad con la norma de codificación de vídeo cuando un valor del elemento sintáctico particular es mayor que un tamaño máximo de DPB (por ejemplo, `MaxDpbSize`).

**[0150]** Además, de acuerdo con una o más técnicas de ejemplo de esta divulgación, la sección A.4.2 del borrador de trabajo 8 de HEVC se puede cambiar de la siguiente manera. La sección A.4.2 del borrador de trabajo 8 de HEVC describe los límites de nivel específicos de perfil para el perfil principal. La Tabla A-2 puede permanecer igual que en el borrador de trabajo 8 de HEVC. Los flujos de bits que se ajustan al perfil principal en un grado y nivel especificados obedecerán las siguientes restricciones para las pruebas de conformidad de flujo de bits como se especifica en el Anexo C:

a) El número de segmentos (*con `dependent_slice_flag` igual a 0 o 1*) en una imagen es menor que o igual a `MaxSlicesPerPicture`, donde `MaxSlicesPerPicture` se especifica en la Tabla A-1.

b) Para los parámetros HRD VCL,

$\text{BitRate}[\text{SchedSelIdx}] \leq \text{cpbBrVclFactor} * \text{MaxBR}$  y

$\text{CpbSize}[\text{SchedSelIdx}] \leq \text{cpbBrVclFactor} * \text{MaxCPB}$  para al menos un valor de `SchedSelIdx`, donde `cpbBrVclFactor` se especifica en la Tabla A-2 y `BitRate[SchedSelIdx]` y `CpbSize[SchedSelIdx]` se proporcionan de la siguiente manera.

- Si `vcl_hrd_parameters_present_flag` es igual a 1,

`BitRate[SchedSelIdx]` y `CpbSize[SchedSelIdx]` vienen dados por las ecuaciones E-45 y E-46, respectivamente, usando los elementos sintácticos que se seleccionan como se especifica en la subcláusula C.1.

- De lo contrario (`vcl_hrd_parameters_present_flag` es igual a 0), `BitRate[SchedSelIdx]` y `CpbSize[SchedSelIdx]` se infieren como se especifica en la subcláusula E.2.3 para los parámetros HRD VCL.

`MaxBR` y `MaxCPB` se especifican en la Tabla A-1 en unidades de bits/s de `cpbBrVclFactor` y de bits de `cpbBrVclFactor`, respectivamente. El flujo de bits cumplirá estas condiciones para al menos un valor de `SchedSelIdx` en el intervalo de 0 a `cpb_cnt_minus1[TargetDecHighestTid]`, ambos inclusive.

c) En cuanto a los parámetros HRD NAL,

$\text{BitRate}[\text{SchedSelIdx}] \leq \text{cpbBrNalFactor} * \text{MaxBR}$  y

$\text{CpbSize}[\text{SchedSelIdx}] \leq \text{cpbBrNalFactor} * \text{MaxCPB}$  para al menos un valor de `SchedSelIdx`, donde `cpbBrNalFactor` se especifica en la Tabla A-2 y `BitRate[SchedSelIdx]` y `CpbSize[SchedSelIdx]` se proporcionan de la siguiente manera.

- Si `nal_hrd_parameters_present_flag` es igual a 1,

BitRate[SchedSelIdx] y CpbSize[SchedSelIdx] vienen dados por las ecuaciones E-45 y E-46, respectivamente, usando los elementos sintácticos que se seleccionan como se especifica en la subcláusula C.1.

- De lo contrario (nal\_hrd\_parameters\_present\_flag es igual a 0), BitRate[SchedSelIdx] y CpbSize[SchedSelIdx] se infieren como se especifica en la subcláusula E.2.3 para los parámetros HRD NAL.

MaxBR y MaxCPB se especifican en la Tabla A-1 en unidades de bits/s de cpbBrNalFactor y de bits de cpbBrNalFactor, respectivamente. El flujo de bits cumplirá estas condiciones para al menos un valor de SchedSelIdx en el intervalo de 0 a cpb\_cnt\_minus1[TargetDecHighestTid], ambos inclusive.

d) La suma de las variables NumBytesInNALunit para la unidad de acceso 0 es menor que o igual a

$1,5 * (\text{Max}(\text{PicSizeInSamplesY}, \text{fR} * \text{MaxLumaSR}) + \text{MaxLumaSR} * (\text{tr}(0) - \text{tr}_n(0))) \div \text{MinCR}$  para el valor de PicSizeInSamplesY de la imagen 0, donde MaxLumaSR y MinCR son los valores especificados en Tabla A-1 que se aplica a la imagen 0.

e) La suma de las variables NumBytesInNALunit para la unidad de acceso n con  $n > 0$  es menor que o igual a

$1,5 * \text{MaxLumaSR} * (\text{tr}(n) - \text{tr}(n-1)) \div \text{MinCR}$ , donde MaxLumaSR y MinCR son los valores especificados en la Tabla A-1 que se aplican a la imagen n.

f) Para el nivel 5 y los niveles superiores, la variable CtbSizeY será igual a 32 o 64.

g) El valor de NumPocTotalCurr será menor que o igual a 8.

h) El valor de num\_tile\_columns\_minus1 será menor que MaxTileCols y num\_tile\_rows\_minus1 será menor que MaxTileRows, donde MaxTileCols y MaxTileRows son como se especifica en la Tabla A-1.

**[0151]** Como se indica en otra parte de esta divulgación, los parámetros cpb\_cnt\_minus1[i] pueden no seleccionarse apropiadamente en las restricciones de nivel. El borrador de trabajo 8 de HEVC especifica que "el flujo de bits cumplirá estas condiciones para al menos un valor de SchedSelIdx en el intervalo de 0 a cpb\_cnt\_minus1, ambos inclusive..." Especificar TargetDecHighestTid como el índice i de cpb\_cnt\_minus1[i] puede, de acuerdo con una o más técnicas de esta divulgación, garantizar que las restricciones de nivel se seleccionen de manera coherente con i igual al valor claramente especificado de TargetDecHighestTid.

**[0152]** Además, de acuerdo con una o más técnicas de esta divulgación, la subcláusula general C.1 en el Anexo C del borrador de trabajo 8 de HEVC puede modificarse. Las figuras C-1 y C-2 de la subcláusula C.1 del borrador de trabajo 8 de HEVC pueden seguir siendo las mismas que en el WD8 de HEVC. El texto de la subcláusula C.1 del borrador de trabajo 8 de HEVC puede modificarse de la siguiente manera.

Este anexo especifica el descodificador de referencia hipotético (HRD) y su uso para comprobar el flujo de bits y la conformidad del descodificador.

Dos tipos de flujos de bits están sujetos a la comprobación de conformidad de HRD para esta memoria descriptiva. El primer tipo de flujo de bits, denominado flujo de bits de Tipo I, es un flujo de unidades NAL que contiene solo las unidades NAL VCL y unidades NAL con nal\_unit\_type igual a FD\_NUT (unidades NAL de datos de relleno) para todas las unidades de acceso en el flujo de bits. El segundo tipo de flujo de bits, denominado flujo de bits de Tipo II, 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 lo siguiente:

- unidades NAL no VCL adicionales que no sean unidades NAL de datos de relleno,
- todos los elementos sintácticos leading\_zero\_8bits, zero\_byte, start\_code\_prefix\_one\_3bytes, y trailing\_zero\_8bits que forman un flujo de octetos a partir del flujo de unidades NAL (como se especifica en el Anexo B).

La figura C-1 muestra los tipos de puntos de conformidad de flujo de bits comprobados por el HRD.

Los elementos sintácticos de unidades NAL no VCL (o sus valores por defecto para algunos de los elementos sintácticos), requeridos para el HRD, se especifican en las subcláusulas semánticas de la cláusula 7, Anexos D y E.

Se usan dos tipos de parámetros HRD (parámetros HRD NAL y parámetros HRD VCL). Los parámetros HRD se señalizan a través de la estructura sintáctica de conjunto de parámetros de vídeo o mediante la información de usabilidad de vídeo como se especifica en las subcláusulas E.1 y E.2, que es parte de la estructura sintáctica de conjunto de parámetros de secuencia.

Se pueden necesitar múltiples pruebas para verificar la conformidad de un flujo de bits. Para cada prueba, se aplican las siguientes etapas en el orden indicado:

- 5 1. Se selecciona un punto de funcionamiento bajo prueba, denotado como TargetOp. TargetOp se identifica mediante OpLayerIdSet igual a targetOpLayerIdSet y OpTid igual a targetOpTid. targetOpLayerIdSet contiene el conjunto de valores para nuh\_reserved\_zero\_6bits presente en el subconjunto de flujo de bits asociado a TargetOp y será un subconjunto de valores para nuh\_reserved\_zero\_6bits presente en el flujo de bits bajo prueba. targetOpTid es igual al mayor TemporalId presente en el subconjunto de flujo de bits asociado a TargetOp y será menor que o igual a greatestTemporalId presente en el flujo de bits bajo prueba.
- 10 2. TargetDecLayerIdSet se establece como targetOpLayerIdSet, TargetDecHighestTid se establece como targetOpTid y BitstreamToDecode se establece en la salida del proceso de extracción de subflujo de bits como se especifica en la subcláusula 10.1 con TargetDecHighestTid y TargetDecLayerIdSet como entradas.
- 15 3. Se selecciona la estructura sintáctica hrd\_parameters( ) y la estructura sintáctica sub\_layer\_hrd\_parameters( ) aplicable a TargetOp. Si TargetDecLayerIdSet contiene solo el valor 0, se selecciona la estructura sintáctica hrd\_parameters( ) en el conjunto de parámetros de secuencia activa. De lo contrario, se selecciona la estructura sintáctica hrd\_parameters( ) que está en el conjunto de parámetros de secuencia activo (o proporcionada a través de un medio externo) y para la cual el conjunto de valores especificados por on\_layer\_id[opIdx][i] para i en el intervalo de 0 a op\_num\_layer\_id\_values\_minus1[opIdx], ambos inclusive, es idéntico a TargetDecLayerIdSet. Dentro de la estructura sintáctica hrd\_parameters( ) seleccionada, si BitstreamToDecode es un flujo de bits de Tipo I, se selecciona la estructura sintáctica sub\_layer\_hrd\_parameters(TargetDecHighestTid) que sigue inmediatamente a la condición "if (vcl\_hrd\_parameters\_present\_flag)" (en este caso, la variable NalHrdModeFlag es igual a 0), de lo contrario (BitstreamToDecode es un flujo de bits de Tipo II), se selecciona la estructura sintáctica sub\_layer\_hrd\_parameters(TargetDecHighestTid) que sigue inmediatamente 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), y todas las unidades NAL no VCL, excepto las unidades NAL de datos de relleno, se descartan de BitstreamToDecode en el caso anterior y el resultado se asigna a BitstreamToDecode.
- 20 4. Una unidad de acceso asociada a un mensaje SEI de periodo de almacenamiento en memoria intermedia aplicable a TargetOp se selecciona como el punto de inicialización de HRD y se denomina unidad de acceso 0.
- 25 5. Se seleccionan mensajes SEI que incluyen información de temporización. Se selecciona el mensaje SEI de periodo de almacenamiento en memoria intermedia que se codifica en la unidad de acceso 0 y se aplica a TargetOp, según lo indicado por la estructura sintáctica aplicable\_operation\_points( ). Para cada unidad de acceso en BitstreamToDecode que comienza desde la unidad de acceso 0, se selecciona el mensaje SEI de temporización de imágenes que está asociado a la unidad de acceso y se aplica a TargetOp, según lo indicado por la estructura sintáctica aplicable\_operation\_points( ), y cuando SubPicCpbFlag es igual a 1 y sub\_pic\_cpb\_params\_in\_pic\_timing\_sei\_flag es igual a 0, se seleccionan los mensajes SEI de temporización de subimágenes que están asociados a unidades de descodificación en la unidad de acceso y se aplican a TargetOp, según lo indicado por las estructuras sintácticas aplicable\_operation\_points( ).
- 30 6. Se selecciona un valor de SchedSelIdx. El SchedSelIdx seleccionado estará en el intervalo de 0 a cpb\_cnt\_minus1[TargetDecHighestTid], ambos inclusive, donde cpb\_cnt\_minus1[TargetDecHighestTid] se encuentra en la estructura sintáctica sub\_layer\_hrd\_parameters(TargetDecHighestTid) como se seleccionó anteriormente.
- 35 7. Se selecciona el retardo de eliminación de CPB y el desfase de retardo iniciales y las unidades de acceso TFD asociadas a la unidad de acceso 0 pueden descartarse de BitstreamToDecode. Si la imagen codificada en la unidad de acceso 0 tiene nal\_unit\_type igual a CRA\_NUT o BLA\_W\_LP, y rap\_cpb\_params\_present\_flag en el mensaje SEI seleccionado de periodo de almacenamiento en memoria intermedia es igual a 1, o bien se selecciona el retardo de eliminación de CPB y el desfase de retardo iniciales por defecto representados por initial\_cpb\_removal\_delay[SchedSelIdx] e initial\_cpb\_removal\_delay\_offset[SchedSelIdx] correspondientes a NalHrdModeFlag (en este caso, la variable DefaultInitCpbParamsFlag se establece igual a 1) o bien se selecciona el retardo de eliminación de CPB y el desfase de retardo iniciales alternativos representados por initial\_alt\_cpb\_removal\_delay[SchedSelIdx] e initial\_alt\_cpb\_removal\_delay\_offset[SchedSelIdx] correspondientes a NalHrdModeFlag (en este caso la variable DefaultInitCpbParamsFlag se establece igual a 0), y las unidades de acceso TFD asociadas a la unidad de acceso 0 se descartan de BitstreamToDecode en este último caso y el resultado se asigna a BitstreamToDecode. De lo contrario, se selecciona el retardo de eliminación de CPB y la el desfase de retardo iniciales por defecto (en este caso, la variable DefaultInitCpbParamsFlag se establece igual a 1).
- 40 45 50 55 60 65

El número de pruebas de conformidad de flujo de bits realizadas es igual a  $N1 * N2 * N3 * (N4 * 2 + N5)$ , donde los valores de  $N1$ ,  $N2$ ,  $N3$ ,  $N4$  y  $N5$  se especifican de la siguiente manera.

- $N1$  es el número de puntos de funcionamiento contenidos en el flujo de bits bajo prueba.
- Si *BitstreamToDecode* es un flujo de bits de Tipo I,  $N2$  es igual a 1, de lo contrario (*BitstreamToDecode* es un flujo de bits de Tipo II)  $N2$  es igual a 2.
- $N3$  es igual a  $cpb\_cnt\_minus1[TargetDecHighestTid] + 1$ .
- $N4$  es el número de unidades de acceso asociadas a mensajes SEI de periodo de almacenamiento en memoria intermedia aplicables a *TargetOp* en *BitstreamToDecode*, donde la imagen codificada en cada una de estas unidades de acceso tiene *nal\\_unit\\_type* igual a *CRA\_NUT* o *BLA\_W\_LP*, y el mensaje SEI asociado de periodo de almacenamiento en memoria intermedia aplicable a *TargetOp* tiene *rap\\_cpb\\_params\\_present\\_flag* igual a 1.
- $N5$  es el número de unidades de acceso asociadas a mensajes SEI de periodo de almacenamiento en memoria intermedia aplicables a *TargetOp* en *BitstreamToDecode*, donde la imagen codificada en cada una de estas unidades de acceso tiene un *nal\\_unit\\_type* no igual a uno de *CRA\_NUT* y *BLA\_W\_LP*, o el mensaje SEI asociado de periodo de almacenamiento en memoria intermedia aplicable a *TargetOp* tiene *rap\\_cpb\\_params\\_present\\_flag* igual a 0.

Cuando *BitstreamToDecode* es un flujo de bits de Tipo II, si se selecciona la estructura sintáctica *sub\_layer\_hrd\_parameters(TargetDecHighestTid)* que sigue inmediatamente la condición "if (*vcl\\_hrd\\_parameters\\_present\\_flag*)", la prueba se realiza en el punto de conformidad de Tipo I que se muestra en la Figura C-1, y solo VCL y las unidades NAL de datos de relleno se cuentan para la velocidad de transferencia de bits de entrada y el almacenamiento en CPB; de lo contrario, se selecciona la estructura sintáctica (*sub\_layer\_hrd\_parameters(TargetDecHighestTid)* que sigue inmediatamente la condición "if (*nal\\_hrd\\_parameters\\_present\\_flag*)", las pruebas se llevan a cabo en el punto de conformidad de Tipo II que se muestra en la Figura C-1, y todas las unidades NAL (de un flujo de unidades NAL de Tipo II) o todos los octetos (de un flujo de octetos) se cuentan para la velocidad de transferencia de bits de entrada y el almacenamiento en CPB.

NOTA 3: Los parámetros HRD NAL establecidos por un valor de *SchedSelIdx* para el punto de conformidad de Tipo II mostrado en la Figura C-1 son suficientes para establecer también la conformidad HRD VCL para el punto de conformidad de Tipo I mostrado en la Figura C-1 para los mismos valores de *InitCpbRemovalDelay[SchedSelIdx]*, *BitRate[SchedSelIdx]* y *CpbSize[SchedSelIdx]* para el caso VBR (*cbr\\_flag[SchedSelIdx]* igual a 0). Esto se debe a que el flujo de datos hacia el punto de conformidad de Tipo I es un subconjunto del flujo de datos hacia el punto de conformidad de Tipo II y porque, para el caso VBR, se permite que la CPB se vacíe y permanezca vacía hasta el momento en que está planificada la llegada de una imagen siguiente. Por ejemplo, cuando se descodifica una secuencia de vídeo codificada conforme a uno o más de los perfiles especificados en el Anexo A usando el proceso de descodificación especificado en las cláusulas 2 a 9, cuando se proporcionan parámetros HRD NAL para el punto de conformidad de Tipo II que no solo se encuentran dentro de los límites establecidos para los parámetros HRD NAL para la conformidad de perfil en el punto c) de la subcláusula A.4.2, sino que también se encuentran dentro de los límites establecidos para los parámetros HRD VCL para la conformidad de perfil en el punto b) de la subcláusula A.4.2, la conformidad del HRD VCL para el punto de conformidad de Tipo I también se garantiza que se encuentra dentro de los límites del punto b) de la subcláusula A.4.2.

Todos los conjuntos de parámetros de vídeo, los conjuntos de parámetros de secuencia y los conjuntos de parámetros de imágenes a los que se hace referencia en las unidades NAL VCL, y los correspondientes mensajes SEI de temporización de imágenes y de periodo de almacenamiento en memoria intermedia se transmitirán al HRD, de manera oportuna, ya sea en el flujo de bits (mediante unidades NAL no VCL), o mediante otros medios no especificados en esta memoria descriptiva.

En los anexos C, D y E, la especificación para la "presencia" de unidades NAL no VCL también se cumple cuando esas unidades NAL (o solo algunas de ellas) se transmiten a los descodificadores (o al DRH) por otros medios no especificados por esta memoria descriptiva. Con el fin de contar los bits, solo se cuentan los bits apropiados que están realmente presentes en el flujo de bits.

NOTA 1: Como ejemplo, la sincronización de una unidad NAL no VCL, transmitida por medios distintos a la presencia en el flujo de bits, con las unidades NAL que están presentes en el flujo de bits, se puede lograr indicando dos puntos en el flujo de bits, entre los cuales la unidad NAL no VCL hubiera estado presente en el flujo de bits, si el codificador hubiera decidido transmitirla en el flujo de bits. Cuando el contenido de una unidad NAL no VCL se transmite para la aplicación mediante algún otro medio que no sea la presencia en el flujo de



bits, no se requiere que la representación del contenido de la unidad NAL no VCL use la misma sintaxis que la especificada en esta memoria descriptiva.

5 NOTA 2: Cuando la información HRD está contenida dentro del flujo de bits, es posible verificar la conformidad de un flujo de bits con los requisitos de esta subcláusula en función de, únicamente, la información contenida en el flujo de bits. Cuando la información HRD no está presente en el flujo de bits, como es el caso de todos los flujos de bits de Tipo I "independientes", la conformidad solo puede verificarse cuando los datos HRD se suministran por algún otro medio no especificado en esta memoria descriptiva.

10 El HRD contiene una memoria intermedia de imágenes codificadas (CPB), un proceso de descodificación instantánea, una memoria intermedia de imágenes descodificadas (DPB) y un recorte de salida como se muestra en la Figura C-2.

15 Para cada prueba de conformidad de flujo de bits, el tamaño de CPB (número de bits) es  $CpbSize[SchedSelIdx]$  como se especifica en la Ecuación E-46, donde  $SchedSelIdx$  y los parámetros HRD se seleccionan como se ha especificado anteriormente en esta subcláusula. El tamaño de DPB (número de memorias intermedias de almacenamiento de imágenes) es  $ps\_max\_dec\_pic\_buffering[TargetDecHighestTid]$ .

20 La variable  $SubPicCpbPreferredFlag$  se especifica ya sea por medios externos, o cuando no se especifica por medios externos, se establece igual a 0.

La variable  $SubPicCpbFlag$  se obtiene de la siguiente forma:

25  $SubPicCpbFlag = SubPicCpbPreferredFlag \ \&\& \ sub\_pic\_cpb\_params\_present\_flag(C-1)$

Si  $SubPicCpbFlag$  es igual a 0, la CPB funciona a nivel de unidad de acceso, y cada unidad de descodificación es una unidad de acceso. De otro modo, el HRD funciona a nivel de subimagen, y cada unidad de descodificación es un subconjunto de una unidad de acceso.

30 El HRD funciona de la siguiente manera. Los datos asociados a las unidades de descodificación que fluyen en la CPB de acuerdo con una planificación de llegada especificada son entregados por el HSS. 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. Cada imagen descodificada se coloca en la DPB. Una imagen descodificada se elimina de la DPB como se especifica en la subcláusula C.3.1 o la subcláusula C.5.2.

40 El funcionamiento de la CPB para cada prueba de conformidad de flujo de bits se especifica en la subcláusula C.2. La operación de descodificador instantánea se especifica en las cláusulas 2 a 9. El funcionamiento de la DPB para cada prueba de conformidad de flujo de bits se especifica en la subcláusula C.3. El recorte de salida para cada prueba de conformidad de flujo de bits se especifica en la subcláusula C.3.2 y la subcláusula C.5.2.

45 La información de HSS y HRD referente al número de planificaciones de entrega enumeradas y sus velocidades de transferencia de bits y tamaños de memoria intermedia asociados se especifican en las subcláusulas E.1.1, E.1.2, E.2.1 y E.2.2. El HRD se inicializa como se especifica en el mensaje SEI de periodo de almacenamiento en memoria intermedia especificado en subcláusulas D.1.1 y D.2.1. La temporización de eliminación de las unidades de descodificación de la CPB y la temporización de salida de las imágenes descodificadas de la DPB se especifican en el mensaje SEI de temporización de imágenes como se especifica en las subcláusulas D.1.2 y D.2.1. Toda la información de temporización relacionada con una unidad de descodificación específica deberá llegar antes del tiempo de eliminación de CPB de la unidad de descodificación.

50 *Los requisitos para la conformidad de flujo de bits se especifican en la subcláusula C.4, y el HRD se usa para verificar la conformidad de los descodificadores como se especifica en la subcláusula C.5.*

55 NOTA 3: Si bien la conformidad está garantizada bajo el supuesto de que todas las velocidades de imagen y los relojes usados para generar el flujo de bits coinciden exactamente con los valores señalizados en el flujo de bits, en un sistema real cada uno de estos puede variar respecto al valor señalado o especificado.

60 Toda la aritmética en este anexo se realiza con valores reales, de modo que no se pueden propagar errores de redondeo. Por ejemplo, el número de bits en una CPB justo antes o después de la eliminación de una unidad de descodificación no es necesariamente un número entero.

La variable  $t_c$  se puede obtener como se indica a continuación y se denomina pulso de reloj:

65 
$$t_c = num\_units\_in\_tick \div time\_scale \ (C-1)$$

La variable  $t_{c\_sub}$  se puede obtener como se indica a continuación y se denomina pulso de reloj de subimagen:

$$t_{c\_sub} = t_c \div (\text{tick\_divisor\_minus2} + 2) \quad (C-2)$$

Se especifica lo siguiente para expresar las restricciones en este anexo:

- Dejar que la unidad de acceso  $n$  sea la  $n$ -ésima unidad de acceso en el orden de descodificación, siendo la primera unidad de acceso la unidad de acceso 0 (es decir, la unidad de acceso 0-ésima).
- Dejar que la imagen  $n$  sea la imagen codificada o la imagen descodificada de la unidad de acceso  $n$ .
- Dejar que la unidad de descodificación  $m$  sea la  $m$ -ésima unidad de descodificación en el orden de descodificación siendo la primera unidad de descodificación una unidad de descodificación 0.

**[0153]** Las modificaciones de la sección C.1 del borrador de trabajo 8 de HEVC anterior pueden aclarar las pruebas de conformidad de flujo de bits. Como se indicó anteriormente, cuando se invoca el proceso de descodificación para la prueba de conformidad de flujo de bits en el borrador de trabajo 8 de HEVC, la semántica de los elementos sintácticos no se especifica claramente, ya que los valores de TargetDecLayerIdSet y TargetDecHighestTid no se establecen apropiadamente. Las modificaciones de la sección C.1 aclaran las definiciones de TargetDecLayerIdSet y TargetDecHighestTid.

**[0154]** Como se muestra en las modificaciones anteriores de la sección C.1 del borrador de trabajo 8 de HEVC, un dispositivo puede realizar una operación HRD (como una prueba de conformidad de flujo de bits) que selecciona un punto de funcionamiento, y determinar un conjunto objetivo de identificadores de capa (TargetDecLayerIdSet) del punto de funcionamiento y el identificador temporal más alto (TargetDecHighestTid). Además, en la operación HRD, el dispositivo puede seleccionar un conjunto de parámetros HRD aplicables al punto de funcionamiento y usar el conjunto seleccionado de parámetros HRD para configurar un HRD que realiza el proceso de descodificación. El conjunto de parámetros HRD aplicables al punto de funcionamiento particular puede incluir parámetros que especifican un retardo de eliminación de CPB inicial, un tamaño de CPB, una velocidad de transferencia de bits, un retardo de salida de DPB inicial, un tamaño de DPB, etc. La operación HRD puede incluir realizar un proceso de descodificación.

**[0155]** En algunos ejemplos, el dispositivo puede seleccionar, de entre uno o más conjuntos de parámetros HRD (por ejemplo, estructuras sintácticas hrd\_parameters( )) en un VPS y un conjunto de parámetros HRD en un SPS, el conjunto de parámetros HRD aplicable al punto de funcionamiento. En algunos ejemplos, el dispositivo puede determinar que el conjunto de parámetros HRD en el SPS puede aplicarse al punto de funcionamiento particular cuando un conjunto de identificadores de capa del punto de funcionamiento contiene un conjunto de todos los identificadores de capa presentes en una secuencia de vídeo codificada asociada al SPS. Además, en algunos ejemplos, el dispositivo puede seleccionar el conjunto de parámetros HRD en el SPS en respuesta a la determinación de que el conjunto de identificadores de capa objetivo (por ejemplo, TargetDecLayerIdSet) del punto de funcionamiento contiene solo el valor 0. En algunos ejemplos, el dispositivo puede seleccionar un conjunto de parámetros HRD en el SPS en respuesta a la determinación de que un conjunto de identificadores de capa (por ejemplo, op\_layer\_id[ ][ ]) es idéntico al conjunto de identificadores de capa objetivo (por ejemplo, TargetDecLayerIdSet) del punto de funcionamiento.

**[0156]** Además, como se muestra en las modificaciones anteriores de la sección C.1 del borrador de trabajo 8 de HEVC y otras partes de esta divulgación, el dispositivo puede descodificar, a partir de un SPS, la matriz de elementos sintácticos (sps\_max\_dec\_pic\_buffering[ ]), donde cada uno indica un tamaño máximo requerido de una DPB del HRD. El dispositivo puede determinar, basándose en el identificador temporal objetivo más alto, un elemento sintáctico particular en la matriz (es decir, sps\_max\_dec\_pic\_buffering[TargetDecHighestTid]). Como se indicó anteriormente, el número de memorias intermedias de almacenamiento de imágenes en la DPB se indica mediante el elemento sintáctico particular (es decir, el tamaño de DPB (número de memorias intermedias de almacenamiento de imágenes) es

sps\_max\_dec\_pic\_buffering[TargetDecHighestTid]).

**[0157]** Además, un proceso de descodificación puede descodificar una estructura sintáctica de parámetros HRD (hrd\_parameters( )) que incluye el conjunto seleccionado de parámetros HRD. El conjunto seleccionado de parámetros HRD incluye una matriz de elementos sintácticos (cbp\_cnt\_minus1[ ]), donde cada uno indica un número de especificaciones de CPB alternativas en el flujo de bits. Las modificaciones de la sección C.1 del borrador de trabajo 8 de HEVC aclaran que cuando un dispositivo realiza una operación HRD, el dispositivo puede seleccionar, basándose en el identificador temporal objetivo más alto (TargetDecHighestTid), un elemento sintáctico particular de la matriz (cbp\_cnt\_minus1[TargetDecHighestTid]) y puede seleccionar un índice de selección de planificador (SchedSelIdx) en un intervalo de 0 a un valor del elemento sintáctico particular. El dispositivo puede determinar, basándose al menos en parte en el índice de selección de planificador, un retardo de eliminación de CPB inicial de una CPB del HRD.

**[0158]** La sección C.2.1 del borrador de trabajo 8 de HEVC se refiere a la eliminación de imágenes de la DPB para la conformidad de flujo de bits. De acuerdo con una o más técnicas de ejemplo de esta divulgación, la sección C.2.1 del borrador de trabajo 8 de HEVC se puede cambiar de la siguiente manera:

5 *Las especificaciones en esta subcláusula se aplican independientemente a cada conjunto de parámetros DPB seleccionados como se especifica en la subcláusula C.1.*

10 La eliminación de imágenes de la DPB antes de la descodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de segmento del primer segmento de la imagen actual) sucede instantáneamente en el momento de la eliminación de la CPB de la primera unidad de descodificación de la unidad de acceso n (que contiene la imagen actual) y procede como se indica a continuación.

15 Se invoca el proceso de descodificación para el conjunto de imágenes de referencia como se especifica en la subcláusula 8.3.2.

Cuando la imagen actual es una imagen IDR o BLA, se aplica lo siguiente:

20 1. Cuando la imagen IDR o BLA no es la primera imagen descodificada y el valor de `pic_width_in_luma_samples` o `pic_height_in_luma_samples` o `sps_max_dec_pic_buffering[TargetDecHighestTid]` obtenido del conjunto de parámetros de secuencia activo es diferente del valor de `pic_width_in_luma_samples` o `pic_height_in_luma_samples` o `sps_max_dec_pic_buffering[TargetDecHighestTid]` obtenido del conjunto de parámetros de secuencia que estuvo activo para la imagen anterior, respectivamente, `no_output_of_prior_pics_flag` puede inferirse que sea igual a 1 por el HRD, independientemente del valor real de `no_output_of_prior_pics_flag`.

25 **NOTA:** las implementaciones de descodificador deben intentar manejar los cambios en la imagen o el tamaño de DPB más fácilmente que el HRD con respecto a cambios en `pic_width_in_luma_samples`, `pic_height_in_luma_samples`, o `sps_max_dec_pic_buffering[TargetDecHighestTid]`.

30 2. Cuando `no_output_of_prior_pics_flag` es igual a 1 o se infiere que es igual a 1, todas las memorias intermedias de almacenamiento de imágenes en la DPB se vacían sin proporcionar las imágenes que contienen, y la completitud de DPB se ajusta a 0.

35 Todas las imágenes k de la DPB, para las que ambas condiciones siguientes son verdaderas, se eliminan de la DPB:

- la imagen k está marcada como "no usada como referencia",
- 40 - la imagen k tiene `PicOutputFlag` igual a 0 o su tiempo de salida DPB es menor que o igual al tiempo de eliminación de CPB de la primera unidad de descodificación (denotada como unidad de descodificación m) de la imagen actual n; es decir  $t_{o,dpb}(k) \leq t_r(m)$ .

45 Cuando una imagen se elimina de la DPB, la completitud de DPB se reduce en uno.

**[0159]** Como se indica en otra parte de esta divulgación, los parámetros `sps_max_dec_pic_buffering[i]` pueden no seleccionarse apropiadamente en operaciones HRD. El borrador de trabajo 8 de HEVC simplemente indica `sps_max_dec_pic_buffering[i]` en lugar de `sps_max_dec_pic_buffering[TargetDecHighestTid]`, como se muestra anteriormente. El borrador de trabajo 8 de HEVC no indica la semántica del índice i en la sección C.2.1. Especificar `TargetDecHighestTid` como el índice i de `sps_max_dec_pic_buffering[i]` puede, de acuerdo con una o más técnicas de esta divulgación, garantizar que i igual al valor claramente especificado de `TargetDecHighestTid` se use en

`sps_max_dec_pic_buffering[i]` al realizar la operación HRD de eliminar imágenes de la DPB.

55 **[0160]** Como se muestra en las modificaciones anteriores de la sección C.2.1 del borrador de trabajo 8 de HEVC, un dispositivo puede descodificar, a partir de un SPS activo para una imagen actual, una primera matriz de elementos sintácticos (`sps_max_dec_pic_buffering[ ]`), donde cada uno indica el tamaño máximo requerido de una DPB del HRD. Además, el dispositivo puede descodificar, a partir de un SPS activo para una imagen anterior, una segunda matriz de elementos sintácticos (`sps_max_dec_pic_buffering[ ]`), donde cada uno indica el tamaño máximo requerido de la DPB del HRD. El dispositivo puede determinar, basándose en el identificador temporal objetivo más alto (`TargetDecHighestTid`), un primer elemento sintáctico en la primera matriz (`sps_max_dec_pic_buffering[TargetDecHighestTid]`). Además, el dispositivo puede determinar, basándose en el identificador temporal objetivo más alto, un segundo elemento sintáctico en la segunda matriz

65 (`sps_max_dec_pic_buffering[TargetDecHighestTid]`). Cuando la imagen actual es una imagen de actualización de descodificación instantánea (IDR) o una imagen de acceso de enlace roto (BLA) y un valor del primer elemento

sintáctico es diferente de un valor del segundo elemento sintáctico, el dispositivo puede inferir un valor de un tercer elemento sintáctico (`no_output_of_prior_pics_flag`) independientemente de un valor indicado por el tercer elemento sintáctico. El tercer elemento sintáctico puede especificar cómo se tratan imágenes descodificadas previamente en la DPB después de descodificar una imagen IDR o una imagen BLA.

5 **[0161]** Una imagen IDR puede ser una imagen de punto de acceso aleatorio (RAP) para la cual cada segmento tiene un `nal_unit_type` igual a `IDR_W_LP` o `IDR_N_LP`. Una imagen IDR contiene solo segmentos I, y puede ser la primera imagen en el flujo de bits en orden de descodificación, o puede aparecer más adelante en el flujo de bits. Una imagen IDR que tiene `nal_unit_type` igual a `IDR_N_LP` no tiene imágenes principales asociadas presentes en el flujo de bits. Una imagen principal es una imagen que precede a la imagen RAP asociada en orden de salida. Una imagen IDR que tiene `nal_unit_type` igual a `IDR_W_LP` no tiene imágenes etiquetadas para su descarte (TFD) asociadas presentes en el flujo de bits, pero puede tener imágenes DLP asociadas en el flujo de bits.

10 **[0162]** Una imagen BLA es una imagen RAP para la cual cada segmento tiene un `nal_unit_type` igual a `BLA_W_TFD`, `BLA_W_DLP` o `BLA_N_LP`. Una imagen BLA que tenga un `nal_unit_type` igual a `BLA_W_TFD` puede tener imágenes TFD asociadas presentes en el flujo de bits. Una imagen BLA que tiene un `nal_unit_type` igual a `BLA_N_LP` no tiene imágenes principales asociadas presentes en el flujo de bits. Una imagen BLA que tiene un `nal_unit_type` igual a `BLA_W_DLP` no tiene imágenes TFD asociadas presentes en el flujo de bits, pero puede tener imágenes DLP asociadas en el flujo de bits.

15 **[0163]** La sección C.3 del borrador de trabajo 8 de HEVC describe operaciones de conformidad de flujo de bits. De acuerdo con una o más técnicas de ejemplo de esta divulgación, la sección C.3 del borrador de trabajo 8 de HEVC puede modificarse de la siguiente manera:

20 Un flujo de bits de datos codificados conforme a esta memoria descriptiva deberá cumplir con todos los requisitos especificados en esta subcláusula.

25 El flujo de bits se construirá de acuerdo con la sintaxis, la semántica y las restricciones especificadas en esta memoria descriptiva fuera de este anexo.

30 La primera imagen codificada en un flujo de bits será una imagen RAP, es decir, una imagen IDR, una imagen CRA o una imagen BLA.

35 Para cada imagen actual que se descodifica, permítase que las variables `maxPicOrderCnt` y `minPicOrderCnt` se establezcan iguales al máximo y al mínimo, respectivamente, de los valores `PicOrderCntVal` de las siguientes imágenes:

- La imagen actual.
- 40 - La imagen anterior en orden de descodificación que tiene `TemporalId` igual a 0.
- Las imágenes de referencia a corto plazo en el conjunto de imágenes de referencia de la imagen actual.
- Todas las imágenes `n` que tienen `PicOutputFlag` igual a 1 y  $t_r(n) < t_r(\text{currPic})$  y  $t_{o,dpb}(n) = t_r(\text{currPic})$ , donde
- 45 `currPic` es la imagen actual.

Se cumplirán todas las condiciones siguientes para cada una de las pruebas de conformidad de flujo de bits:

1. Para cada unidad de acceso `n`, donde  $n > 0$ , asociada a un mensaje SEI de periodo de almacenamiento en memoria intermedia, donde  $\Delta t_{g,90}(n)$  especificado por

$$\Delta t_{g,90}(n) = 90000 * ( t_{r,n}(n) - t_{r,n}(n-1) ) \quad (C-18)$$

55 el valor de `InitCpbRemovalDelay[SchedSelIdx]` se limitará de la siguiente manera.

- Si `cbr_flag[SchedSelIdx]` es igual a 0,
- `InitCpbRemovalDelay[SchedSelIdx] <= Ceil( $\Delta t_{g,90}(n)$ )` (C-19)
- 60 - De otro modo, (`cbr_flag[SchedSelIdx]` es igual a 1),
- `Floor( $\Delta t_{g,90}(n)$ ) <= InitCpbRemovalDelay[SchedSelIdx] <= Ceil( $\Delta t_{g,90}(n)$ )` (C-20)

65 **NOTA 4:** El número exacto de bits en la CPB en el momento de la eliminación de cada imagen puede depender de qué mensaje SEI de periodo de almacenamiento en memoria intermedia se seleccione para

inicializar el HRD. Los codificadores deben tener esto en cuenta para garantizar que se deben obedecer todas las restricciones especificadas, independientemente de qué mensaje SEI de periodo de almacenamiento en memoria intermedia se seleccione para inicializar el HRD, ya que el HRD se puede inicializar en cualquiera de los mensajes SEI de periodo de almacenamiento en memoria intermedia.

5 2. Un desbordamiento de CPB se especifica como la condición en la que el número total de bits en la CPB es mayor que el tamaño de CPB. La CPB no se desbordará nunca.

10 3. Un subdesbordamiento de CPB se especifica como la condición en la que el tiempo de eliminación de CPB nominal de la unidad de descodificación  $m$   $t_{r,n}(m)$  es menor que el tiempo de llegada de CPB final de la unidad de descodificación  $m$   $t_{af}(m)$  para al menos un valor de  $m$ . Cuando `low_delay_hrd_flag` es igual a 0, nunca se producirá un subdesbordamiento en la CPB.

15 4. Cuando `low_delay_hrd_flag` es igual a 1, puede producirse un subdesbordamiento de CPB en la unidad de descodificación  $m$ . En este caso, el tiempo de llegada de CPB final de la unidad de acceso  $n$  que contiene la unidad de descodificación  $m$   $t_{af}(n)$  será mayor que el tiempo de eliminación nominal de CPB de la unidad de acceso  $n$  que contiene la unidad de descodificación  $m$   $t_{r,n}(n)$ .

20 5. Los tiempos de eliminación nominales de las imágenes de la CPB (a partir de la segunda imagen en orden de descodificación) deberán satisfacer las restricciones en  $t_{r,n}(n)$  y  $t_r(n)$  expresadas en las subcláusulas A.4.1 a A.4.2.

25 6. *Para cada imagen actual que se descodifica, después de invocar el proceso de eliminación de imágenes de la DPB como se especifica en la subcláusula C.3.1, el número de imágenes descodificadas en la DPB, incluidas todas las imágenes  $n$  que están marcadas como "usadas como referencia" o que tienen `PicOutputFlag` igual a 1 y  $t_{o,dpb}(n) \geq t_r(\text{currPic})$ , donde `currPic` es la imagen actual, será menor que o igual a  $\text{Max}(0, \text{sps\_max\_dec\_pic\_buffering}[\text{TargetDecHighestTid}] - 1)$ .*

30 7. Todas las imágenes de referencia deberán estar presentes en la DPB cuando sean necesarias para la predicción. Cada imagen que tenga `OutputFlag` igual a 1 deberá estar presente en la DPB en su tiempo de salida de DPB a menos que se elimine de la DPB antes de su tiempo de salida por uno de los procesos especificados en la subcláusula C.3.

35 8. Para cada imagen actual que se descodifique, el valor de `maxPicOrderCnt - minPicOrderCnt` será menor que `MaxPicOrderCntLsb/2`.

40 9. El valor de  $\Delta_{t_{o,dpb}}(n)$  según la Ecuación C-17, que es la diferencia entre el tiempo de salida de una imagen y el de la primera imagen que la sigue en orden de salida y que tiene `PicOutputFlag` igual a 1 satisfará la restricción expresada en la subcláusula A.4.1 para el perfil, grado y nivel especificado en el flujo de bits usando el proceso de descodificación especificado en las cláusulas 2 a 9.

45 **[0164]** Como se indica en otra parte de esta divulgación, los parámetros `sps_max_dec_pic_buffering[i]` pueden no seleccionarse adecuadamente en operaciones de conformidad de flujo de bits. En el punto 6 de la sección C.3, el borrador de trabajo 8 de HEVC indica que "el número de imágenes descodificadas en la DPB... será menor que o igual a  $\text{Min}(0, \text{sps\_max\_dec\_pic\_buffering}[\text{TemporalId}] - 1)$ ", donde `TemporalId` no está definido. Especificar `TargetDecHighestTid` como el índice  $i$  de `sps_max_dec_pic_buffering[i]` puede, de acuerdo con una o más técnicas de esta divulgación, garantizar que  $i$  igual al valor claramente especificado de `TargetDecHighestTid` se use en

50 `sps_max_dec_pic_buffering[i]` cuando se realicen las operaciones de conformidad de flujo de bits.

**[0165]** Cuando un dispositivo realiza un proceso de descodificación como parte de una operación HRD, el dispositivo puede descodificar, a partir de un SPS, una matriz de elementos sintácticos

55 (`sps_max_dec_pic_buffering[ ]`), donde cada uno indica el tamaño máximo requerido de una DPB del HRD. Además, como parte de la realización de la operación HRD, el dispositivo puede determinar, basándose en el identificador temporal objetivo más alto (`TargetDecHighestTid`), un elemento sintáctico particular en la matriz. Además, como se muestra en las modificaciones anteriores de la sección C.3 del borrador de trabajo 8 de HEVC, el dispositivo puede determinar, basándose al menos en parte en si un número de imágenes descodificadas en la DPB es menor que o igual al máximo de 0 y un valor del elemento sintáctico particular menos 1, si el flujo de bits se ajusta a la norma de codificación de vídeo.

60 **[0166]** La sección C.4 del borrador de trabajo 8 de HEVC describe la conformidad del descodificador. De acuerdo con una o más técnicas de ejemplo de esta divulgación, la sección C.4 del borrador de trabajo 8 de HEVC se puede cambiar de la siguiente manera:

65

Un descodificador conforme a esta memoria descriptiva deberá cumplir con todos los requisitos especificados en esta subcláusula.

Un descodificador que reclame conformidad con un perfil, grado y nivel específicos podrá descodificar con éxito todos los flujos de bits que cumplan con los requisitos de conformidad de flujo de bits especificados en la subcláusula C.4, de la manera especificada en el Anexo A, siempre que todos los conjuntos de parámetros de vídeo, conjuntos de parámetros de secuencia y conjuntos de parámetros de imagen a los que se hace referencia en las unidades NAL VCL y los mensajes SEI apropiados de periodo de almacenamiento en memoria intermedia y temporización de imágenes se transmitan al descodificador, de manera oportuna, ya sea en el flujo de bits (por unidades NAL no VCL) o por medios externos no especificados por esta memoria descriptiva.

Cuando un flujo de bits contiene elementos sintácticos que tienen valores que se especifican como reservados y se especifica que los descodificadores deben ignorar los valores de los elementos sintácticos o unidades NAL que contienen los elementos sintácticos que tienen los valores reservados, y el flujo de bits se ajusta a esta memoria descriptiva, un descodificador en conformidad descodificará el flujo de bits de la misma manera que descodificaría un flujo de bits en conformidad e ignorará los valores de los elementos sintácticos o unidades NAL que contienen los elementos sintácticos que tienen los valores reservados como se especifica.

Hay dos tipos de conformidad que puede reclamar un descodificador:

conformidad de temporización de salida y conformidad de orden de salida.

Para verificar la conformidad de un descodificador, los flujos de bits de prueba que se ajustan al perfil, grado y nivel reclamados, según lo especificado en la subcláusula C.4, son entregados por un planificador de flujo hipotético (HSS) tanto al HRD como al descodificador bajo prueba (DUT). Todas las imágenes proporcionadas por el HRD también serán proporcionadas por el DUT y, para cada imagen proporcionada por el HRD, los valores de todas las muestras proporcionadas por el DUT para la imagen correspondiente serán iguales a los valores de las muestras proporcionadas por el HRD.

Para la conformidad de descodificador de temporización de salida, el HSS funciona como se describe anteriormente, con planificaciones de entrega seleccionadas solo del subconjunto de valores de SchedSelIdx para las cuales la velocidad de transferencia de bits y el tamaño de CPB están restringidos como se especifica en el Anexo A para el perfil, grado y nivel especificados, o con planificaciones de entrega "interpoladas" como se especifica a continuación para las cuales la velocidad de transferencia de bits y el tamaño de CPB están restringidos como se especifica en el Anexo A. La misma planificación de entrega se usa tanto para el HRD como para el DUT.

Cuando los parámetros HRD y los mensajes SEI de periodo de almacenamiento en memoria intermedia están presentes con  $cpb\_cnt\_minus1[TargetDecHighestTid]$  mayor que 0, el descodificador será capaz de descodificar el flujo de bits tal como se entrega desde el HSS que funciona usando una planificación de entrega "interpolada" especificada como que tiene una velocidad pico  $r$  de transferencia de bits, un tamaño de CPB  $c(r)$  y un retardo inicial de eliminación de CPB ( $f(r) \div r$ ) de la siguiente manera:

$$\alpha = (r - BitRate[ SchedSelIdx - 1 ]) \div ( BitRate[ SchedSelIdx ] - BitRate[ SchedSelIdx - 1 ] ), (C-22)$$

$$c(r) = \alpha * CpbSize[ SchedSelIdx ] + (1 - \alpha) * CpbSize[ SchedSelIdx - 1 ], (C-23)$$

$$f(r) = \alpha * InitCpbRemovalDelay[ SchedSelIdx ] * BitRate[ SchedSelIdx ] + (1 - \alpha) * InitCpbRemovalDelay[ SchedSelIdx - 1 ] * BitRate[ SchedSelIdx - 1 ] (C-24)$$

para cualquier  $SchedSelIdx > 0$  y  $r$  de modo que  $BitRate[ SchedSelIdx - 1 ] \leq r \leq BitRate[ SchedSelIdx ]$  de modo que  $r$  y  $c(r)$  estén dentro de los límites especificados en el Anexo A para la máxima velocidad de transferencia de bits y el tamaño de memoria intermedia para el perfil, grado y nivel especificados.

NOTA 1:  $InitCpbRemovalDelay[ SchedSelIdx ]$  puede ser diferente de un periodo de almacenamiento en memoria intermedia a otro y tiene que recalcularse.

En cuanto a la conformidad de descodificador de temporización de salida, se usa un HRD como el descrito anteriormente y la temporización (relativa al tiempo de entrega del primer bit) de salida de imagen es la misma tanto para el HRD como para el DUT hasta un retardo fijo.

5 En cuanto a la conformidad de descodificador de orden de salida, se aplica lo siguiente.

- El HSS entrega el flujo de bits *BitstreamToDecode* al DUT "bajo demanda" del DUT, lo que significa que el HSS entrega bits (en orden de descodificación) solo cuando el DUT requiere más bits para continuar con su procesamiento.

10 NOTA 2: Esto significa que para esta prueba, la memoria intermedia de imágenes codificadas del DUT podría ser tan pequeña como el tamaño de la unidad de descodificación más grande.

- Se usa un HRD modificado como se describe a continuación, y el HSS entrega el flujo de bits al HRD mediante una de las planificaciones especificadas en el flujo de bits *BitstreamToDecode* de modo que la velocidad de transferencia de bits y el tamaño de CPB estén restringidos como se especifica en el Anexo A. El orden de salida de imágenes será el mismo tanto para el HRD como para el DUT.

- En cuanto a la conformidad de descodificador de orden de salida, *el tamaño de CPB es  $CpbSize[SchedSellIdx]$  como se especifica en la Ecuación E-46, donde  $SchedSellIdx$  y los parámetros HRD se seleccionan como se especifica anteriormente en la subcláusula C.1. El tamaño de DPB es  $sps\_max\_dec\_pic\_buffering[TargetDecHighestTid]$ . El tiempo de eliminación de la CPB para el HRD es igual al tiempo de llegada de bit final y la descodificación es inmediata. El funcionamiento de la DPB de este HRD es como se describe en las subcláusulas C.5.1 a C.5.3.*

25 **[0167]** Como se indica en otra parte de esta divulgación, los parámetros *cpb\_cnt\_minus1[i]* y *sps\_max\_dec\_pic\_buffering[i]* pueden no seleccionarse apropiadamente en los requisitos de conformidad de descodificador. Por ejemplo, la sección C.4 del borrador de trabajo 8 de HEVC no especifica un índice para *cpb\_cnt\_minus1*. Especificar *TargetDecHighestTid* como el índice *i* de *cpb\_cnt\_minus1[i]* y *sps\_max\_dec\_pic\_buffering[i]* puede, de acuerdo con una o más técnicas de esta divulgación, garantizar que las operaciones de conformidad de descodificador se realicen de manera coherente con *i* igual al valor claramente especificado de *TargetDecHighestTid*.

35 **[0168]** Además, la sección C.4.2 del borrador de trabajo 8 de HEVC describe la eliminación de imágenes de la DPB para la conformidad de descodificador. De acuerdo con una o más técnicas de ejemplo de esta divulgación, el título de la sección C.4.2 puede cambiarse de "eliminación de imágenes de la DPB" a "provisión y eliminación de imágenes de la DPB". El texto de la sección C.4.2 del borrador de trabajo 8 de HEVC puede modificarse de la siguiente manera:

40 La *provisión* y eliminación de imágenes de la DPB antes de la descodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de segmento del primer segmento de la imagen actual) puede producirse instantáneamente cuando la primera unidad de descodificación de la unidad de acceso que contiene la imagen actual se elimina de la CPB y procede como se indica a continuación.

45 Se invoca el proceso de descodificación para el conjunto de imágenes de referencia como se especifica en la subcláusula 8.3.2.

- Cuando la imagen actual es una imagen IDR o BLA, se aplica lo siguiente.

50 1. Cuando la imagen IDR o BLA no es la primera imagen descodificada y el valor de *pic\_width\_in\_luma\_samples* o *pic\_height\_in\_luma\_samples* o *sps\_max\_dec\_pic\_buffering[TargetDecHighestTid]* obtenido del conjunto de parámetros de secuencia activo es diferente del valor de *pic\_width\_in\_luma\_samples* o *pic\_height\_in\_luma\_samples* o *sps\_max\_dec\_pic\_buffering[TargetDecHighestTid]* obtenido del conjunto de parámetros de secuencia que estuvo activo para la imagen anterior, respectivamente, *no\_output\_of\_prior\_pics\_flag* puede inferirse que sea igual a 1 por el HRD, independientemente del valor real de *no\_output\_of\_prior\_pics\_flag*.

60 NOTA: las implementaciones de descodificador deben intentar manejar los cambios en la imagen o el tamaño de DPB más fácilmente que el HRD con respecto a cambios en *pic\_width\_in\_luma\_samples*, *pic\_height\_in\_luma\_samples* o *sps\_max\_dec\_pic\_buffering[TargetDecHighestTid]*.

65 2. Cuando *no\_output\_of\_prior\_pics\_flag* es igual a 1 o se infiere que es igual a 1, todas las memorias intermedias de almacenamiento de imágenes en la DPB se vacían sin proporcionar las imágenes que contienen.

3. Cuando *no\_output\_of\_prior\_pics\_flag* no es igual a 1 y no se infiere que sea igual a 1, las memorias intermedias de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para proporcionarse" y "no usada como referencia" se vacían (sin proporcionar su contenido), y todas las memorias intermedias de almacenamiento de imágenes no vacías en la DPB se vacían invocando repetidamente el proceso de "choque" especificado en la subcláusula C.5.2.1.

- De lo contrario (la imagen actual no es una IDR o una imagen BLA), las memorias intermedias de almacenamiento de imágenes que contienen una imagen marcada como "no necesaria para su provisión" y "no usada como referencia" se vacían (sin proporcionar su contenido). Cuando una o más de las siguientes condiciones son verdaderas, el proceso de "choque" especificado en la subcláusula C.5.2.1 se invoca repetidamente hasta que haya una memoria intermedia de almacenamiento de imágenes vacía para almacenar la imagen descodificada actual.

1. El número de imágenes en la DPB que están marcadas como "necesarias para su provisión" es mayor que

$sps\_max\_num\_reorder\_pics[TargetDecHighestTid]$ .

2. El número de imágenes en la DPB es igual a

$sps\_max\_dec\_pic\_buffering[TargetDecHighestTid]$ .

#### **Proceso de "choque"**

[0169] El proceso de "choque" se invoca en los siguientes casos.

- La imagen actual es una imagen IDR o BLA y *no\_output\_of\_prior\_pics\_flag* no es igual a 1 y no se infiere que sea igual a 1, como se especifica en la subcláusula C.5.2.

- La imagen actual no es una imagen IDR o BLA, y el número de imágenes en la DPB que están marcadas como "necesarias para su provisión" es mayor que  $sps\_max\_num\_reorder\_pics[TargetDecHighestTid]$ , como se especifica en la subcláusula C.5.2.

- La imagen actual no es una imagen IDR o BLA y el número de imágenes en la DPB es igual a  $sps\_max\_dec\_pic\_buffering[TargetDecHighestTid]$ , como se especifica en la subcláusula C.5.2.

[0170] El proceso de "choque" puede consistir en las siguientes etapas ordenadas:

1. La imagen que está primero para su provisión se selecciona como la única que tiene el valor más pequeño de *PicOrderCntVal* de todas las imágenes en la DPB marcadas como "necesaria para su provisión".

2. La imagen se recorta, usando el rectángulo de recorte especificado en el conjunto de parámetros de secuencia activo para la imagen, la imagen recortada se proporciona y la imagen se marca como "no necesaria para su provisión".

3. Si la memoria intermedia de almacenamiento de imágenes que incluía la imagen que se recortó y proporcionó contiene una imagen marcada como "no usada como referencia", la memoria intermedia de almacenamiento de imágenes se vacía.

[0171] Como se indica en otra parte de esta divulgación, los parámetros  $sps\_max\_dec\_pic\_buffering[i]$  y  $sps\_max\_num\_reorder\_pics[i]$  pueden no seleccionarse apropiadamente en las operaciones HRD, tal como la eliminación de imágenes de la DPB. Especificar *TargetDecHighestTid* como el índice *i* de  $sps\_max\_dec\_pic\_buffering[i]$  y  $sps\_max\_num\_reorder\_pics[i]$  puede, de acuerdo con una o más técnicas de esta divulgación, garantizar que *i* igual al valor claramente especificado de *TargetDecHighestTid* se usa en  $sps\_max\_dec\_pic\_buffering[i]$  y  $sps\_max\_num\_reorder\_pics[i]$  cuando se realiza la operación HRD de eliminación de imágenes de la DPB.

[0172] Cuando un dispositivo realiza un proceso de descodificación durante una operación HRD, el dispositivo puede descodificar, a partir de un SPS, una matriz de elementos sintácticos

( $sps\_max\_dec\_pic\_buffering[ ]$ ), donde cada uno indica el tamaño máximo requerido de una DPB del HRD. Además, cuando el dispositivo realiza la operación HRD, el dispositivo puede determinar, basándose en el identificador temporal objetivo más alto, un elemento sintáctico particular de la matriz ( $sps\_max\_dec\_pic\_buffering[TargetDecHighestTid]$ ). Además, el dispositivo puede realizar un proceso de choque



que vacía una o más memorias intermedias de almacenamiento de imágenes de la DPB cuando una imagen actual no es una imagen IDR o una imagen BLA y el número de imágenes en la DPB marcadas como necesarias para su provisión es mayor que el valor del elemento sintáctico particular.

5 **[0173]** De manera similar, cuando un dispositivo realiza un proceso de decodificación durante una operación HRD, el dispositivo puede decodificar, a partir un SPS, una matriz de elementos sintácticos

10 (sps\_max\_dec\_pic\_buffering[ ]), donde cada uno indica el tamaño máximo requerido de una DPB del HRD. Además, cuando el dispositivo realiza la operación HRD, el dispositivo puede determinar, basándose en el identificador temporal objetivo más alto, un elemento sintáctico particular de la matriz (sps\_max\_dec\_pic\_buffering[TargetDecHighestTid]). Además, el dispositivo puede realizar un proceso de choque que vacía una o más memorias intermedias de almacenamiento de imágenes de la DPB cuando una imagen actual no es una imagen IDR o una imagen BLA y el número de imágenes en la DPB es igual al indicado por el elemento sintáctico particular.

15 **[0174]** Además, de acuerdo con una o más técnicas de esta divulgación, una estructura sintáctica applicable\_operation\_points( ) y la semántica asociada pueden añadirse al borrador de trabajo 8 de HEVC. La siguiente Tabla 5 muestra un ejemplo de sintaxis de la estructura sintáctica applicable\_operation\_points( ).

20 **TABLA 5 - puntos de funcionamiento aplicables**

applicable_operation_points( ) {	
<b>num_applicable_ops_minus1</b>	ue(v)
if(num_applicable_ops_minus1 > 0)	
<b>default_op_applicable_flag</b>	u(1)
mumOpsSignalled = default_op_applicable_flag ? num_applicable_ops_minus1 : num_applicable_ops_minus1 + 1	
for(i = 0; i < mumOpsSignalled; i++) {	
operation_point_layer_ids(i)	
<b>op_temporal_id[i]</b>	u(3)
}	
}	

25 **[0175]** La estructura sintáctica applicable\_operation\_point( ) que se muestra en la Tabla 5 especifica los puntos de funcionamiento a los que se aplica el mensaje SEI asociado a esta estructura sintáctica. El mensaje SEI asociado a una estructura sintáctica aplicable\_operation\_point( ) (también conocido como el mensaje SEI asociado) es el mensaje SEI que contiene la estructura sintáctica aplicable\_operation\_point( ). El mensaje SEI asociado a una estructura sintáctica applicable\_operation\_point( ) puede ser un mensaje SEI de periodo de almacenamiento en memoria intermedia, un mensaje SEI de temporización de imágenes o un mensaje SEI de temporización de subimágenes.

30 **[0176]** Un punto de funcionamiento por defecto puede definirse como el punto de funcionamiento identificado por OpLayerIdSet que contiene los valores 0 a nuh\_reserved\_zero\_6bits, ambos inclusive, donde nuh\_reserved\_zero\_6bits se codifica en la cabecera de unidad NAL de la unidad NAL SEI que contiene el mensaje SEI asociado, y OpTid es igual al valor TemporalId de la unidad NAL SEI que contiene el mensaje SEI asociado. De forma alternativa, el punto de funcionamiento por defecto puede definirse como el punto de funcionamiento identificado por el OpLayerIdSet que solo contiene los nuh\_reserved\_zero\_6bits en la cabecera de unidad NAL de la unidad NAL SEI que contiene el mensaje SEI asociado, y OpTid es igual al valor TemporalId de la unidad NAL SEI que contiene el mensaje SEI asociado. De forma alternativa, el punto de funcionamiento por defecto puede definirse como el punto de funcionamiento identificado por el OpLayerIdSet que solo contiene el valor 0, y OpTid es igual al valor TemporalId de la unidad NAL SEI que contiene el mensaje SEI asociado.

35 **[0177]** Si default\_op\_applicable\_flag es igual a 1, los puntos de funcionamiento a los que se aplica el mensaje SEI asociado son el punto de funcionamiento por defecto y los puntos de funcionamiento num\_applicable\_ops\_minus1 identificados por OpLayerIdSet según lo especificado por operation\_point\_layer\_ids(i) y OpTid igual a op\_temporal\_id[i], donde i está en el intervalo de 0 a num\_applicable\_ops\_minus1, ambos inclusive. De lo contrario (default\_op\_applicable\_flag es igual a 0), los puntos de funcionamiento a los que se aplica el mensaje SEI asociado pueden ser los puntos de funcionamiento num\_applicable\_ops\_minus1 + 1 identificados por OpLayerIdSet según lo especificado por operation\_point\_layer\_ids(i) y OpTid igual a op\_temporal\_id[i], donde i está en el intervalo de 0 a num\_applicable\_ops\_minus1 + 1, ambos inclusive.

5 **[0178]** Además, en el ejemplo de sintaxis de la Tabla 5, el elemento sintáctico `num_applicable_ops_minus1`, más 1, especifica el número de puntos de funcionamiento a los que se aplica el mensaje SEI asociado. El valor de `num_applicable_ops_minus1` puede estar en el intervalo de 0 a 63, ambos inclusive. En el ejemplo de la Tabla 5, el elemento sintáctico `default_op_applicable_flag` igual a 1 especifica que el mensaje SEI asociado se aplica al punto de funcionamiento por defecto. El elemento sintáctico `default_op_applicable_flag` igual a 0 especifica que los mensajes SEI no se aplican al punto de funcionamiento por defecto. El elemento sintáctico `op_temporal_id[i]` especifica el *i*-ésimo valor `OpTid` señalado explícitamente en la estructura sintáctica `applicable_operation_point( )`. El valor de `op_temporal_id[i]` puede estar en el intervalo entre 0 y 6, ambos inclusive.

10 **[0179]** Como se indica anteriormente, el borrador de trabajo 8 de HEVC no proporciona la manera de asociar un mensaje SEI de periodo de almacenamiento en memoria intermedia o un mensaje SEI de temporización de imágenes a una estructura sintáctica `hrd_parameters( )` para la cual la estructura sintáctica `operation_point_layer_ids( )` asociada incluye múltiples valores de `nuh_reserved_zero_6bits` (es decir, múltiples ID de capa en una extensión de codificación de vídeo escalable, 3DV o de múltiples vistas de HEVC). La inclusión de la estructura sintáctica `applicable_operation_point( )` puede resolver este problema al menos parcialmente. La estructura sintáctica `applicable_operation_point( )` puede especificar claramente los puntos de funcionamiento a los que se aplica un mensaje SEI de periodo de almacenamiento en memoria intermedia, un mensaje SEI de temporización de imágenes o un mensaje SEI de temporización de subimágenes. Esto puede permitir el uso de información transportada en los elementos sintácticos `nuh_reserved_zero_6bits` y `temporal_id_plus1` en la cabecera de unidad NAL de las unidades NAL SEI, y puede permitir la compartición de la información transmitida en un mismo mensaje SEI de periodo de almacenamiento en memoria intermedia, de temporización de imágenes o de temporización de subimágenes para el procesamiento de datos de vídeo asociados a múltiples puntos de funcionamiento.

25 **[0180]** La sección D.1.1 del borrador de trabajo 8 de HEVC describe la sintaxis de los mensajes SEI de periodo de almacenamiento en memoria intermedia. De acuerdo con una o más técnicas de ejemplo de esta divulgación, la sintaxis de mensaje SEI del periodo de almacenamiento en memoria intermedia puede cambiarse como se muestra en la siguiente Tabla 6. Los cambios en la sintaxis del mensaje SEI de periodo de almacenamiento en memoria intermedia pueden permitir que los mensajes SEI de periodo de almacenamiento en memoria intermedia incluyan estructuras sintácticas `applicable_operation_points( )`.

**TABLA 6 - periodo de almacenamiento en memoria intermedia**

Buffering_period(payloadSize) {	Descriptor
<b>seq_parameter_set_id</b>	ue(v)
<i>applicable_operation_points()</i>	
if(!sub_pic_cpb_params_present_flag)	
<b>rap_cpb_params_present_flag</b>	u(1)
if(NalHrdBpPresentFlag) {	
for(SchedSelIdx= 0; SchedSelIdx< CpbCnt; SchedSelIdx++)	
{	
<b>initial_cpb_removal_delay</b> [SchedSelIdx]	u(v)
<b>initial_cpb_removal_delay_offset</b> [SchedSelIdx]	u(v)
if(sub_pic_cpb_params_present_flag    rap_cpb_params_present_flag) {	
<b>initial_alt_cpb_removal_delay</b> [SchedSelIdx]	u(v)
<b>initial_alt_cpb_removal_delay_offset</b> [SchedSelIdx]	u(v)
}	
}	
}	
if(VclHrdBpPresentFlag) {	
for(SchedSelIdx= 0; SchedSelIdx< CpbCnt; SchedSelIdx++)	
{	
<b>initial_cpb_removal_delay</b> [SchedSelIdx]	u(v)
<b>initial_cpb_removal_delay_offset</b> [SchedSelIdx]	u(v)

	Descriptor
Buffering_period(payloadSize) {	
if(sub_pic_cpb_params_present_flag    rap_cpb_params_present_flag) {	
initial_alt_cpb_removal_delay[SchedSelIdx]	u(v)
initial_alt_cpb_removal_delay_offset[SchedSelIdx]	u(v)
}	
}	
}	
}	

[0181] La Sección D.2.1 del borrador de trabajo 8 de HEVC describe la semántica de los elementos sintácticos de los mensajes SEI de periodo de almacenamiento en memoria intermedia. De acuerdo con una o más técnicas de esta divulgación, la semántica de la estructura sintáctica buffering\_period(payloadSize) se puede cambiar de la siguiente manera. La semántica para aquellos elementos sintácticos no mencionados es la misma que en el borrador de trabajo 8 de HEVC.

*Un mensaje SEI de periodo de almacenamiento en memoria intermedia proporciona información sobre el retardo de eliminación de CPB y el desfase de retardo de eliminación de CPB iniciales.*

*Lo siguiente se aplica a la sintaxis y la semántica de los mensajes SEI de periodo de almacenamiento en memoria intermedia:*

- *Los elementos sintácticos initial\_cpb\_removal\_delay\_length\_minus1 y sub\_pic\_cpb\_params\_present\_flag, y las variables NalHrdBpPresentFlag, VclHrdBpPresentFlag, CpbSize[SchedSelIdx], BitRate[SchedSelIdx] y CpbCnt se encuentran en o se obtienen a partir de los elementos sintácticos encontrados en la estructura sintáctica hrd\_parameters( ) y la estructura sintáctica sub\_layer\_hrd\_parameters( ) que son aplicables a cualquiera de los puntos de funcionamiento a los que se aplica el mensaje SEI de periodo de almacenamiento en memoria intermedia.*
- *Dos puntos de funcionamiento cualesquiera a los que se aplica el mensaje SEI de periodo de almacenamiento en memoria intermedia con diferentes valores tldA y tldB de OpTid indican que los valores de cpb\_cnt\_minus1[tldA] y cpb\_cnt\_minus1[tldB] codificados en la(s) estructura(s) sintáctica(s) de hrd\_parameters() aplicable(s) a los dos puntos de funcionamiento son idénticos.*
- *Dos puntos de funcionamiento cualesquiera a los que se aplica el mensaje SEI de periodo de almacenamiento en memoria intermedia con diferentes valores layerIdSetA y layerIdSetB de OpLayerIdSet indica que los valores de nal\_hrd\_parameters\_present\_flag y vcl\_hrd\_parameters\_present\_flag, respectivamente, para las dos estructuras sintácticas hrd\_parameters( ) aplicables a los dos puntos de funcionamiento son idénticos.*
- *El flujo de bits (o una parte del mismo) se refiere al subconjunto de flujo de bits (o una parte del mismo) asociado a cualquiera de los puntos de funcionamiento a los que se aplica el mensaje SEI de periodo de almacenamiento en memoria intermedia.*

*Si NalHrdBpPresentFlag o VclHrdBpPresentFlag son iguales a 1, un mensaje SEI de periodo de almacenamiento en memoria intermedia aplicable a los puntos de funcionamiento especificados puede estar presente en cualquier unidad de acceso en la secuencia de vídeo codificada, y un mensaje SEI de periodo de almacenamiento en memoria intermedia aplicable a los puntos de funcionamiento especificados estará presente en cada unidad de acceso RAP, y en cada unidad de acceso asociada a un mensaje SEI de punto de recuperación. De lo contrario (NalHrdBpPresentFlag y VclHrdBpPresentFlag son iguales a 0), ninguna unidad de acceso en la secuencia de vídeo codificada tendrá un mensaje SEI de periodo de almacenamiento en memoria intermedia aplicable a los puntos de funcionamiento especificados.*

NOTA: en algunas aplicaciones, la presencia frecuente de un mensaje SEI de periodo de almacenamiento en memoria intermedia puede ser deseable.

Cuando una unidad NAL SEI que contiene un mensaje SEI de periodo de almacenamiento en memoria intermedia y que tiene nuh\_reserved\_zero\_6bits igual a 0 está presente, la unidad NAL SEI deberá preceder, en orden de descodificación, a la primera unidad NAL VCL en la unidad de acceso.

Un periodo de almacenamiento en memoria intermedia se especifica como el conjunto de unidades de acceso entre dos instancias del mensaje SEI de periodo de almacenamiento en memoria intermedia en orden de descodificación.

La variable *CpbCnt* se obtiene para que sea igual a  $cpb\_cnt\_minus1[tld] + 1$ , donde  $cpb\_cnt\_minus1[tld]$  se codifica en la estructura sintáctica *hrd\_parameters()* que es aplicable a cualquiera de los puntos de funcionamiento a los que se aplica el mensaje SEI de periodo de almacenamiento en memoria intermedia y que tienen *OpTid* igual a *tld*.

**seq\_parameter\_set\_id** se refiere al conjunto de parámetros de secuencia activo. El valor de *seq\_parameter\_set\_id* será igual al valor de *seq\_parameter\_set\_id* en el conjunto de parámetros de imagen al que hace referencia la imagen codificada asociada al mensaje SEI de periodo de almacenamiento en memoria intermedia. El valor de *seq\_parameter\_set\_id* estará en el intervalo de 0 y 31, ambos inclusive.

**rap\_cpb\_params\_present\_flag** igual a 1 especifica la presencia de los elementos sintácticos *initial\_alt\_cpb\_removal\_delay[SchedSelIdx]* e *initial\_alt\_cpb\_removal\_delay\_offset[SchedSelIdx]*. Cuando no está presente, se infiere que el valor de *alt\_cpb\_params\_present\_flag* es igual a 0. Cuando la imagen asociada no es una imagen CRA ni una imagen BLA, el valor de *alt\_cpb\_params\_present\_flag* será igual a 0.

**initial\_cpb\_removal\_delay[SchedSelIdx]** e

**initial\_alt\_cpb\_removal\_delay[SchedSelIdx]** especifican los retardos de eliminación de CPB iniciales *por defecto y alternativos, respectivamente*, para la *SchedSelIdx*-ésima CPB. Los elementos sintácticos tienen una longitud en bits dada por  $initial\_cpb\_removal\_delay\_length\_minus1 + 1$ , y están en unidades de un reloj de 90 kHz. Los valores de los elementos sintácticos no serán iguales a 0 y serán menores que o iguales a  $90000 * (CpbSize[SchedSelIdx] \div BitRate[SchedSelIdx])$ , el equivalente en tiempo del tamaño de CPB en unidades de reloj de 90 kHz.

**initial\_cpb\_removal\_delay\_offset [SchedSelIdx]** e

**initial\_alt\_cpb\_removal\_delay\_offset[SchedSelIdx]** especifican los desfases de eliminación de CPB iniciales *por defecto y alternativos, respectivamente, para la SchedSelIdx-ésima CPB*. Los elementos sintácticos tienen una longitud en bits dada por  $initial\_cpb\_removal\_delay\_length\_minus1 + 1$  y están en unidades de un reloj de 90 kHz. Estos elementos sintácticos no se usan por los descodificadores y pueden ser necesarios solo para el planificador de entrega (HSS) especificado en el Anexo C.

**[0182]** El mensaje SEI de periodo de almacenamiento en memoria intermedia puede incluir parámetros HRD (por ejemplo, *initial\_cpb\_removal\_delay[SchedSelIdx]*,

*initial\_cpb\_removal\_delay\_offset[SchedSelIdx]*,

*initial\_alt\_cpb\_removal\_delay[SchedSelIdx]* e

*initial\_alt\_cpb\_removal\_delay\_offset[SchedSelIdx]*). Como se indica anteriormente, el borrador de trabajo 8 de HEVC no proporciona ninguna manera de asociar un mensaje SEI de periodo de almacenamiento en memoria intermedia a una estructura sintáctica *hrd\_parameters()* en un VPS para el cual la estructura sintáctica *operation\_point\_layer\_ids()* incluye múltiples valores de *nuh\_reserved\_zero\_6bits* (es decir, múltiples ID de capa en una extensión de codificación de vídeo escalable, 3DV o de múltiples vistas de HEVC). Por lo tanto, de acuerdo con una o más técnicas de esta divulgación, el elemento sintáctico aplicable *operation\_points()* en el mensaje SEI de periodo de almacenamiento en memoria intermedia especifica los puntos de funcionamiento a los que se aplica el mensaje SEI de periodo de almacenamiento en memoria intermedia.

**[0183]** La Sección D.1.2 del borrador de trabajo 8 de HEVC indica la sintaxis de los mensajes SEI de temporización de imágenes. De acuerdo con una o más técnicas de esta divulgación, la sintaxis del mensaje SEI de temporización de imágenes puede cambiarse como se muestra en la siguiente Tabla 7. Los cambios en la sintaxis del mensaje SEI de temporización de imágenes pueden permitir que los mensajes SEI de temporización de imágenes incluyan las estructuras sintácticas *applicable\_operation\_points()*.

**TABLA 7 - Mensaje SEI de temporización de imágenes**

pic_timing(payloadSize) {	Descriptor
<i>applicable_operation_points()</i>	
<b><i>au_cpb_removed_delay_minus1</i></b>	<i>u(v)</i>
<b><i>pic_dpb_output_delay</i></b>	<i>u(v)</i>
if(sub_pic_cpb_params_present_flag) {	
<b><i>num_decoding_units_minus1</i></b>	<i>ue(v)</i>

	Descriptor
pic_timing(payloadSize) {	
<b>du_common_cpb_removal_delay_flag</b>	u(1)
if(du_common_cpb_removal_delay_flag)	
<b>du_common_cpb_removal_delay_minus1</b>	u(v)
for(i = 0; i <= num_decoditig_units_minus1; i++) {	
<b>num_nalus_in_du_minus1[i]</b>	ue(v)
if(!du_common_cpb_removal_delay_flag)	
<b>du_cpb_removal_delay_minus1[i]</b>	u(v)
}	
}	
}	

**[0184]** Además, la semántica del mensaje SEI de temporización de imágenes se puede cambiar de la siguiente manera. La semántica de aquellos elementos sintácticos de la estructura sintáctica pic\_timing (payloadSize) no mencionados a continuación puede ser la misma que la de los del borrador de trabajo 8 de HEVC.

*El mensaje SEI de temporización de imágenes proporciona información sobre el retardo de eliminación de CPB y el retardo de salida de DPB para la unidad de acceso asociada al mensaje SEI.*

*Lo siguiente se aplica a la sintaxis y la semántica de los mensajes SEI de temporización de imágenes:*

- Los elementos sintácticos *sub\_pic\_cpb\_params\_present\_flag*, *cpb\_removal\_delay\_length\_minus1*, *dpb\_output\_delay\_length\_minus1* y *du\_cpb\_removal\_delay\_length\_minus1*, y la variable *CpbDpbDelaysPresentFlag* se encuentran en o se obtienen a partir de los elementos sintácticos encontrados en la estructura sintáctica *hrd\_parameters()* y la estructura sintáctica *sub\_layer\_hrd\_parameters()* aplicables a cualquiera de los puntos de funcionamiento a los que se aplica el mensaje SEI de temporización de imágenes.

- El flujo de bits (o una parte del mismo) se refiere al subconjunto del flujo de bits (o una parte del mismo) asociado con cualquiera de los puntos de funcionamiento a los que se aplica el mensaje SEI de temporización de imágenes.

*NOTA 1: La sintaxis del mensaje SEI de temporización de imágenes depende del contenido de las estructuras sintácticas hrd\_parameters() aplicables a los puntos de funcionamiento a los que se aplica el mensaje SEI de temporización de imágenes. Estas estructuras sintácticas hrd\_parameters() están en el conjunto de parámetros de vídeo y/o en el conjunto de parámetros de secuencia que están activos para la imagen codificada asociada al mensaje SEI de temporización de imágenes. Cuando el mensaje SEI de temporización de imágenes está asociado a una unidad de acceso CRA que es la primera unidad de acceso en el flujo de bits, una unidad de acceso IDR o una unidad de acceso BLA, a menos que esté precedida por un mensaje SEI de periodo de almacenamiento en memoria intermedia dentro de la misma unidad de acceso, la activación del conjunto de parámetros de vídeo y del conjunto de parámetros de secuencia (y, para imágenes IDR o BLA que no son la primera imagen en el flujo de bits, la determinación de que la imagen codificada es una imagen IDR o una imagen BLA) no se produce hasta la descodificación de la primera unidad NAL de segmento codificado de la imagen codificada. Dado que la unidad NAL de segmento codificado de la imagen codificada sigue al mensaje SEI de temporización de imágenes en el orden de unidades NAL, puede haber casos en los que sea necesario que un descodificador almacene la RBSP que contiene el mensaje SEI de temporización de imágenes hasta la determinación del conjunto de parámetros de vídeo activo y/o del conjunto de parámetros de secuencia activo, y después realice el análisis sintáctico del mensaje SEI de temporización de imágenes.*

La presencia de un mensaje SEI de temporización de imágenes en el flujo de bits se especifica de la siguiente manera.

- Si *CpbDpbDelaysPresentFlag* es igual a 1, un mensaje SEI de temporización de imágenes aplicable a los puntos de funcionamiento especificados estará presente en cada unidad de acceso de la secuencia de vídeo codificada.

- De otro modo, (*CpbDpbDelaysPresentFlag* es igual a 0), ningún mensaje SEI de temporización de imágenes aplicable a los puntos de funcionamiento especificados estará presente en ninguna unidad de acceso de la secuencia de vídeo codificada.

Cuando una unidad NAL SEI que contiene un mensaje SEI de temporización de imágenes y que tiene `nuh_reserved_zero_6bits` igual a 0 está presente, la unidad NAL SEI precederá, en orden de descodificación, a la primera unidad NAL VCL en la unidad de acceso.

5 **au\_cpb\_removal\_delay\_minus1** más 1 especifica, cuando el HRD funciona a nivel de unidad de acceso, cuántos pulsos de reloj esperar después de la eliminación de la CPB de la unidad de acceso asociada al mensaje SEI de periodo de almacenamiento en memoria intermedia más reciente en una unidad de acceso anterior antes de eliminarse de la memoria intermedia los datos de unidad de acceso asociados al mensaje SEI de temporización de imágenes. Este valor también se usa para calcular un tiempo lo más temprano posible de llegada de datos de unidad de acceso a la CPB para el HSS. El elemento sintáctico es un código de longitud fija cuya longitud en bits viene dada por `cpb_removal_delay_length_minus1 + 1`.

15 NOTA 2: El valor de `cpb_removal_delay_length_minus1` que determina la longitud (en bits) del elemento sintáctico `au_cpb_removal_delay_minus1` es el valor de `cpb_removal_delay_length_minus1` codificado en el conjunto de parámetros de vídeo o el conjunto de parámetros de secuencia que está activo para la imagen codificada asociada al mensaje SEI de temporización de imágenes, aunque `au_cpb_removal_delay_minus1` más 1 especifica el número de pulsos de reloj relativos al momento de eliminación de la unidad de acceso anterior que contiene un mensaje SEI de periodo de almacenamiento en memoria intermedia, que puede ser una unidad de acceso de una secuencia de vídeo codificada diferente.

20 **pic\_dpb\_output\_delay** se usa para calcular el tiempo de salida de DPB de la imagen. Especifica cuántos pulsos de reloj esperar después de la eliminación de la última unidad de descodificación en una unidad de acceso de la CPB antes de que la imagen descodificada se proporcione desde la DPB.

25 NOTA 3 - Una imagen no se elimina de la DPB en su tiempo de salida cuando todavía está marcada como "usada como referencia a corto plazo" o "usada como referencia a largo plazo".

30 NOTA 4: Solo se especifica un `pic_dpb_output_delay` para una imagen descodificada.

La longitud del elemento sintáctico `pic_dpb_output_delay` viene dada en bits por `dpb_output_delay_length_minus1 + 1`. Cuando

35 `sps_max_dec_pic_buffering[minTid]` es igual a 1, donde `minTid` es el mínimo de los valores `OpTid` de todos los puntos de funcionamiento a los que se aplica el mensaje SEI de sincronización de imágenes, `pic_dpb_output_delay` será igual a 0.

40 El tiempo de salida obtenido a partir de `pic_dpb_output_delay` de cualquier imagen que se proporciona desde una temporización de salida conforme al descodificador precederá al tiempo de salida obtenido a partir de `pic_dpb_output_delay` de todas las imágenes en cualquier secuencia de vídeo codificada posterior en orden de descodificación.

45 El orden de salida de imagen establecido por los valores de este elemento sintáctico será el mismo orden que el establecido por los valores de `PicOrderCntVal`.

50 En cuanto a las imágenes que no se proporcionan por el proceso de "choque" porque preceden, en orden de descodificación, a una imagen IDR o BLA con `no_output_of_prior_pics_flag` igual a 1 o que se infiera como igual a 1, los tiempos de salida obtenidos de `dpb_output_delay` aumentarán con el valor en aumento de `PicOrderCntVal` con respecto a todas las imágenes dentro de la misma secuencia de vídeo codificada.

**du\_common\_cpb\_removal\_delay\_flag** igual a 1 especifica que el elemento sintáctico `du_common_cpb_removal_delay_minus1` está presente. `du_common_cpb_removal_delay_flag` igual a 0 especifica que el elemento sintáctico `du_common_cpb_removal_delay_minus1` no está presente.

55 **du\_common\_cpb\_removal\_delay\_minus1** más 1 especifica cuántos pulsos de reloj de subimagen (véase subcláusula E.2.1) esperar antes de la eliminación de la CPB de cada unidad de descodificación en la unidad de acceso asociada al mensaje SEI de temporización de imágenes, después de la eliminación de la CPB de la anterior unidad de descodificación en orden de descodificación. Este valor también se usa para calcular el tiempo de llegada más temprano posible de los datos de unidad de descodificación a la CPB para el HSS, como se especifica en Anexo C. El elemento sintáctico es un código de longitud fija cuya longitud en bits viene dada por `du_cpb_removal_delay_length_minus1 + 1`.

65 **[0185]** Como se indica anteriormente, el borrador de trabajo 8 de HEVC no proporciona ninguna manera de asociar un mensaje SEI de temporización de imágenes a una estructura sintáctica `hrd_parameters( )` en un VPS para el cual la estructura sintáctica `operation_point_layer_ids( )` asociada incluye múltiples valores de `nuh_reserved_zero_6bits` (es decir, (es decir, múltiples ID de capa en una extensión de codificación de vídeo

escalable, 3DV o de múltiples vistas de HEVC).). Por lo tanto, de acuerdo con una o más técnicas de esta divulgación, el elemento sintáctico `applicable_operation_points()` en el mensaje SEI de temporización de imágenes especifica los puntos de funcionamiento a los que se aplica el mensaje SEI de periodo de almacenamiento en memoria intermedia.

5 **[0186]** Además, de acuerdo con una o más técnicas de esta divulgación, la sintaxis del mensaje SEI de temporización de subimágenes puede cambiarse como se muestra en la siguiente Tabla 8. Los cambios en la sintaxis del mensaje SEI de temporización de subimágenes pueden permitir que los mensajes SEI de temporización de subimágenes incluyan las estructuras sintácticas `applicable_operation_points()`. En el borrador de trabajo 8 de HEVC, el mensaje SEI de temporización de subimágenes no incluye la estructura sintáctica `applicable_operation_points()`.

**TABLA 8 - Mensaje SEI de temporización de subimágenes**

<code>sub_pic_timing(payloadSize) {</code>	<b>Descriptor</b>
<code>  applicable_operation_points()</code>	
<code>  <b>du_spt_cpb_removal_delay_minus1</b></code>	u(v)
<code>}</code>	

15 **[0187]** La sección D.2.2.2 del borrador de trabajo 8 de HEVC describe la semántica de los mensajes SEI de temporización de subimágenes. De acuerdo con una o más técnicas de esta divulgación, la sección D.2.2.2 del borrador de trabajo 8 de HEVC puede modificarse de la siguiente manera:

20 El mensaje SEI de temporización de subimágenes proporciona información de retardo de eliminación de CPB para la unidad de descodificación asociada al mensaje SEI.

*Lo siguiente se aplica a la sintaxis y la semántica de mensajes SEI de temporización de subimágenes:*

25 - Los elementos sintácticos `sub_pic_cpb_params_present_flag` y `cpb_removal_delay_length_minus1`, y la variable `CpbDpbDelaysPresentFlag` se encuentran en o se obtienen a partir de los elementos sintácticos encontrados en la estructura sintáctica `hrd_parameters()` y la estructura sintáctica `sub_layer_hrd_parameters()` aplicables a cualquiera de los puntos de funcionamiento a los que se aplica el mensaje SEI de temporización de subimágenes.

30 - El flujo de bits (o una parte del mismo) se refiere al subconjunto del flujo de bits (o una parte del mismo) asociado con cualquiera de los puntos de funcionamiento a los que se aplica el mensaje SEI de temporización de subimágenes.

35 La presencia del mensaje SEI de temporización de subimágenes en el flujo de bits se especifica de la siguiente manera.

40 - Si `CpbDpbDelaysPresentFlag` es igual a 1 y `sub_pic_cpb_params_present_flag` es igual a 1, un mensaje SEI de temporización de subimágenes aplicable a los puntos de funcionamiento especificados puede estar presente en cada unidad de descodificación en la secuencia de vídeo codificada.

45 - De lo contrario (`CpbDpbDelaysPresentFlag` es igual a 0 o `sub_pic_cpb_params_present_flag` es igual a 0), no habrá mensajes SEI de temporización de subimágenes aplicables a los puntos de funcionamiento especificados en la secuencia de vídeo codificada.

50 La unidad de descodificación asociada a un mensaje SEI de temporización de subimágenes consiste, en orden de descodificación, en la unidad NAL SEI que contiene el mensaje SEI de temporización de subimágenes, seguida de una o más unidades NAL que no contienen un mensaje SEI de temporización de subimágenes, incluidas todas las unidades NAL subsiguientes en la unidad de acceso hasta, pero sin incluir ninguna, unidad NAL SEI subsiguiente que contenga un mensaje SEI de temporización de su-imágenes. Al menos habrá una unidad NAL VCL en cada unidad de descodificación. Todas las unidades NAL no VCL asociadas a una unidad NAL VCL se incluirán en la misma unidad de descodificación.

55 **du\_spt\_cpb\_removal\_delay\_minus1** más 1 especifica el número de ciclos de reloj de subimagen a esperar después de la eliminación de la CPB de la última unidad de descodificación en la unidad de acceso asociada al mensaje SEI de periodo de almacenamiento en memoria intermedia más reciente en una unidad de acceso anterior antes de la eliminación de la CPB de la unidad de descodificación asociada al mensaje SEI de temporización de subimágenes. Este valor también se usa para calcular el tiempo de llegada más temprano

posible de los datos de unidad de descodificación a la CPB para el HSS, como se especifica en Anexo C. El elemento sintáctico se representa mediante un código de longitud fija cuya longitud en bits viene dada por

$cpb\_removal\_delay\_length\_minus1 + 1$ .

NOTA: El valor de  $cpb\_removal\_delay\_length\_minus1$  que determina la longitud (en bits) del elemento sintáctico  $du\_spt\_cpb\_removal\_delay\_minus1$  es el valor de  $cpb\_removal\_delay\_length\_minus1$  codificado en el conjunto de parámetros de vídeo o el conjunto de parámetros de secuencia que está activo para la unidad de acceso que contiene la unidad de descodificación asociada al mensaje SEI de temporización de subimágenes, aunque  $du\_spt\_cpb\_removal\_delay\_minus1$  más 1 especifica un número de pulsos de reloj de subimagen relativos al momento de eliminación de la última unidad de descodificación en la unidad de acceso anterior que contiene un mensaje SEI de periodo de almacenamiento en memoria intermedia, que puede ser una unidad de acceso de una secuencia de vídeo codificada diferente.

**[0188]** La sección E.2.2 del borrador de trabajo 8 de HEVC describe la semántica de parámetros HRD. De acuerdo con una o más técnicas de esta divulgación, la sección E.2.2 del borrador de trabajo 8 de HEVC puede modificarse de la siguiente manera. La semántica para los elementos sintácticos de los parámetros HRD no mencionados a continuación puede ser la misma que los del borrador de trabajo 8 de HEVC.

*La estructura sintáctica  $hrd\_parameters()$  proporciona los parámetros HRD usados en las operaciones HRD. Cuando la estructura sintáctica  $hrd\_parameters()$  se incluye en un conjunto de parámetros de vídeo, el conjunto de número de valores  $nuh\_reserved\_zero\_6bits$  incluidos en  $OpLayerIdSet$  de los puntos de funcionamiento a los que se aplica la estructura sintáctica se especifica mediante la estructura sintáctica  $operation\_point\_layer\_ids()$  correspondiente en el conjunto de parámetros de vídeo o se obtiene implícitamente, como se especifica en la subcláusula 7.4.4. Cuando la estructura sintáctica  $hrd\_parameters()$  se incluye en un conjunto de parámetros de secuencia, los puntos de funcionamiento aplicables son todos los puntos de funcionamiento con  $OpLayerIdSet$  que contiene solo el valor 0. De forma alternativa, cuando la estructura sintáctica  $hrd\_parameters()$  está incluida en un conjunto de parámetros de secuencia, los puntos de funcionamiento aplicables son todos los puntos de funcionamiento con  $OpLayerIdSet$  idénticos a  $TargetDecLayerIdSet$ .*

*Es un requisito de conformidad de flujo de bits que para toda la estructura sintáctica  $hrd\_parameters()$  en la secuencia de vídeo codificada (ya sea en el conjunto de parámetros de vídeo o en el conjunto de parámetros de secuencia), no haya más de uno que se aplique al mismo punto de funcionamiento. De forma alternativa, se requiere que no haya más de una estructura sintáctica  $hrd\_parameters()$  en un conjunto de parámetros de vídeo que se aplique al mismo punto de funcionamiento. De forma alternativa, se requiere que un conjunto de parámetros de vídeo no incluya una estructura sintáctica  $hrd\_parameters()$  que se aplique a los puntos de funcionamiento con  $OpLayerIdSet$  que contiene solo el valor 0.*

$du\_cpb\_removal\_delay\_length\_minus1$  más 1 especifica la longitud, en bits, de los elementos sintácticos  $du\_cpb\_removal\_delay\_minus1[i]$  y

$du\_common\_cpb\_removal\_delay\_minus1$  del mensaje SEI de temporización de imágenes.

$cpb\_removal\_delay\_length\_minus1$  más 1 especifica la longitud, en bits, del elemento sintáctico  $au\_cpb\_removal\_delay\_minus1$  en el mensaje SEI de temporización de imágenes y del elemento sintáctico  $du\_spt\_cpb\_removal\_delay\_minus1$  en el mensaje SEI de temporización de subimágenes. Cuando el elemento sintáctico  $cpb\_removal\_delay\_length\_minus1$  no está presente, se infiere que es igual a 23,  $dpb\_output\_delay\_length\_minus1$  más 1 especifica la longitud, en bits, del elemento sintáctico  $pic\_dpb\_output\_delay$  en el mensaje SEI de temporización de imágenes. Cuando el elemento sintáctico  $dpb\_output\_delay\_length\_minus1$  no está presente, se infiere que es igual a 23.

$fixed\_pic\_rate\_flag[i]$  igual a 1 indica que, cuando  $TargetDecHighestTid$  es igual a  $i$ , la distancia temporal entre los tiempos de salida de HRD de dos imágenes consecutivas cualesquiera en el orden de salida se limita de la siguiente manera.  $fixed\_pic\_rate\_flag[i]$  igual a 0 indica que no se aplican tales restricciones a la distancia temporal entre los tiempos de salida HRD de dos imágenes consecutivas cualesquiera en orden de salida.

Cuando  $fixed\_pic\_rate\_flag[i]$  no está presente, se infiere que es igual a 0. Cuando  $TargetDecHighestTid$  es igual a  $i$  y  $fixed\_pic\_rate\_flag[i]$  es igual a 1 para una secuencia de vídeo codificada que contiene la imagen  $n$ , el valor calculado para  $\Delta t_{o,dpb}(n)$  como se especifica en la Ecuación C-17 será igual a  $t_c * (pic\_duration\_in\_tcs\_minus1[i] + 1)$ , donde  $t_c$  es como se especifica en la Ecuación C-1 (usando el valor de  $t_c$  para la secuencia de vídeo codificada que contiene la imagen  $n$ ) cuando una o más de las siguientes condiciones son verdaderas para la siguiente imagen  $nn$  que se especifica para su uso en la Ecuación C-17:

- La imagen  $nn$  está en la misma secuencia de vídeo codificada que la imagen  $n$ .



- La imagen nn está en una secuencia de vídeo codificada diferente y `fixed_pic_rate_flag[i]` es igual a 1 en la secuencia de vídeo codificada que contiene la imagen nn, el valor de  $\text{num\_units\_in\_tick} \div \text{time\_scale}$  es el mismo para ambas secuencias de vídeo codificadas y el valor de `pic_duration_in_tc_minus1[i]` es el mismo para ambas secuencias de vídeo codificadas.

5 **pic\_duration\_in\_tc\_minus1[i]** más 1 especifica, cuando *TargetDecHighestTid* es igual a i, la distancia temporal, en pulsos de reloj, entre los tiempos de salida de HRD de dos imágenes consecutivas cualesquiera en orden de salida en la secuencia de vídeo codificada. El valor de `pic_duration_in_tc_minus1[i]` estará en el intervalo de 0 to 2047, ambos inclusive.

10 **low\_delay\_hrd\_flag[i]** especifica el modo operativo HRD, cuando *TargetDecHighestTid* es igual a i, como se especifica en el anexo C. Cuando `fixed_pic_rate_flag[i]` es igual a 1, `low_delay_hrd_flag[i]` será igual a 0.

15 NOTA 3: Cuando `low_delay_hrd_flag[i]` es igual a 1, se permiten "imágenes grandes" que violen los tiempos de eliminación de CPB nominales debido al número de bits usados por una unidad de acceso. Se espera, pero no es obligatorio, que dichas "imágenes grandes" ocurran solo ocasionalmente.

20 **cpb\_cnt\_minus1[i]** más 1 especifica el número de especificaciones CPB alternativas en el flujo de bits de la secuencia de vídeo codificada cuando *TargetDecHighestTid* es igual a i. El valor de `cpb_cnt_minus1[i]` estará en el intervalo de 0 a 31, ambos inclusive. Cuando `low_delay_hrd_flag[i]` es igual a 1, `cpb_cnt_minus1[i]` será igual a 0. Cuando `cpb_cnt_minus1[i]` no está presente, se infiere que es igual a 0.

25 **[0189]** Como se describe en otra parte de esta divulgación, en el borrador de trabajo 8 de HEVC, solo las estructuras sintácticas `hrd_parameters( )` en un VPS pueden seleccionarse para operaciones HRD, mientras que las estructuras sintácticas `hrd_parameters( )` en un SPS nunca se seleccionan. Los cambios mostrados anteriormente en la semántica de la estructura sintáctica `hrd_parameters( )` aclaran que cuando la estructura sintáctica `hrd_parameters( )` está incluida en un SPS, los puntos de funcionamiento a los que se aplica la estructura sintáctica `hrd_parameters( )` pueden ser todos los puntos de funcionamiento con `OpLayerIdSet` idénticos a `TargetDecLayerIdSet`. Como se indicó anteriormente en el proceso de descodificación general modificado, si un medio externo está disponible para establecer `TargetDecLayerIdSet`, `TargetDecLayerIdSet` puede especificarse por un medio externo. De lo contrario, si se invoca el proceso de descodificación en una prueba de conformidad de flujo de bits, `TargetDecLayerIdSet` puede ser el conjunto de identificadores de capa de un punto de funcionamiento bajo prueba. De lo contrario, `TargetDecLayerIdSet` puede contener solo un identificador de capa (es decir, solo un valor de `nuh_reserved_zero_6bits`), que es igual a 0. En un ejemplo, los medios externos pueden ser una API que es parte de una implementación de terminal y que proporciona una función para establecer el valor de `TargetDecLayerIdSet`. En este ejemplo, la implementación de terminal puede comprender una implementación de descodificador y ciertas funciones que no son partes de la implementación de descodificador.

30 **[0190]** De esta manera, un dispositivo (tal como un codificador de vídeo 20, un descodificador de vídeo 30, un dispositivo adicional 21 u otro dispositivo) puede seleccionar, de entre un conjunto de parámetros HRD en un conjunto de parámetros de vídeo y un conjunto de parámetros HRD en un SPS, un conjunto de parámetros HRD aplicables a un punto de funcionamiento particular. Además, el dispositivo puede realizar, basándose al menos en parte en el conjunto de parámetros HRD aplicables al punto de funcionamiento particular, una prueba de conformidad de flujo de bits que prueba si un subconjunto de flujo de bits asociado al punto de funcionamiento particular se ajusta a una norma de codificación de vídeo.

35 **[0191]** Como se indicó anteriormente, la sección E.2.2 del borrador de trabajo 8 de HEVC puede modificarse para indicar que cuando la estructura sintáctica `hrd_parameters( )` está incluida en un conjunto de parámetros de secuencia, los puntos de funcionamiento aplicables son todos los puntos de funcionamiento con `OpLayerIdSet` idéntico a `TargetDecLayerIdSet`. Además, como se describió anteriormente, `TargetDecLayerIdSet` se establece como `targetOpLayerIdSet`, que contiene el conjunto de valores de `nuh_reserved_zero_6bits` presentes en el subconjunto de flujo de bits asociado a `TargetOp`. `TargetOp` es el punto de funcionamiento bajo prueba en una operación HRD. Además, las operaciones HRD (por ejemplo, una prueba de conformidad de flujo de bits y una prueba de conformidad de descodificador) pueden invocar el proceso de descodificación general.

40 **[0192]** Como se explicó anteriormente, la sección 8.1 del borrador de trabajo 8 de HEVC puede modificarse para permitir que el proceso de extracción de subflujo de bits como se especifica en la subcláusula 10.1 se aplique con `TargetDecHighestTid` y `TargetDecLayerIdSet` como entradas y que la salida se asigne a un flujo de bits denominado `BitstreamToDecode`. Por lo tanto, los únicos valores de `nuh_reserved_zero_6bits` presentes en `BitstreamToDecode` son los valores de `nuh_reserved_zero_6bits` en `TestDecLayerIdSet` (es decir, el conjunto de valores de `nuh_reserved_zero_6bits` presentes en el subconjunto de flujo de bits asociado a `TargetOp`). La sección 8.1 explica además que cuando se interpreta la semántica de cada elemento sintáctico en cada unidad NAL y "el flujo de bits" o parte del mismo (por ejemplo, una secuencia de vídeo codificada) está involucrado, el flujo de bits o parte del mismo significa `BitstreamToDecode` o parte del mismo.

65

**[0193]** Por lo tanto, al interpretar la sección que describe la semántica de los parámetros HRD (por ejemplo, la sección E.2.2 del borrador de trabajo 8 de HEVC), el término "secuencia de vídeo codificada" significa una parte del BitstreamToDecode. TargetDecLayerIdSet es equivalente al conjunto de todos los valores de nuh\_reserved\_zero\_6bits presentes en el BitstreamToDecode. De ello se desprende que la frase en la sección que describe la semántica de los parámetros HRD "cuando la estructura sintáctica hrd\_parameters( ) está incluida en un conjunto de parámetros de secuencia, los puntos de funcionamiento aplicables son todos los puntos de funcionamiento con OpLayerIdSet idéntico a TargetDecLayerIdSet" es equivalente a "cuando la estructura sintáctica hrd\_parameters( ) está incluida en un conjunto de parámetros de secuencia, los puntos de funcionamiento aplicables son todos los puntos de funcionamiento con OpLayerIdSet idénticos al conjunto de valores de nuh\_reserved\_zero\_6bits presentes en BitstreamToDecode".

**[0194]** Puesto que una "secuencia de vídeo codificada" es parte del BitstreamToDecode, el conjunto de nuh\_reserved\_zero\_6bits presente en la secuencia de vídeo codificada es un subconjunto del conjunto de nuh\_reserved\_zero\_6bits presente en el BitstreamToDecode. Por lo tanto, la frase "cuando la estructura sintáctica hrd\_parameters( ) se incluye en un conjunto de parámetros de secuencia, los puntos de funcionamiento aplicables son todos los puntos de funcionamiento con OpLayerIdSet idénticos al conjunto de valores de nuh\_reserved\_zero\_6bits presentes en BitstreamToDecode" necesariamente implica que "cuando la estructura sintáctica hrd\_parameters( ) está incluida en un conjunto de parámetros de secuencia, los puntos de funcionamiento aplicables son todos los puntos de funcionamiento con OpLayerIdSet que contiene todos los valores de nuh\_reserved\_zero\_6bits presentes en la secuencia de vídeo codificada". En otras palabras, si el conjunto de nuh\_reserved\_zero\_6bits de un punto de funcionamiento es idéntico al conjunto de nuh\_reserved\_zero\_6bits presente en BitstreamToDecode, entonces el conjunto de nuh\_reserved\_zero\_6bits del punto de funcionamiento contiene necesariamente todos los valores nuh\_reserved\_zero\_6bits presentes en una secuencia de vídeo codificada de BitstreamToDecode. En esta frase, "la secuencia de vídeo codificada" puede referirse a una secuencia de vídeo codificada asociada al SPS particular.

**[0195]** Al realizarse una operación HRD, el dispositivo puede determinar, a partir de las estructuras sintácticas hrd\_parameters( ) indicadas en un VPS y una estructura sintáctica hrd\_parameters( ) indicada en un SPS, una estructura sintáctica hrd\_parameters( ) aplicable a TargetOp. Una estructura sintáctica hrd\_parameters( ) particular en el VPS puede aplicarse a TargetOp si el conjunto de id de capa de TargetOp coincide con un conjunto de identificadores de capa especificados en el VPS para la estructura sintáctica hrd\_parameters( ) particular. La estructura sintáctica hrd\_parameters( ) en el SPS puede aplicarse a TargetOp si el conjunto de id de capa de TargetOp (es decir, TargetDecHighestTid) (es decir, el conjunto de nuh\_reserved\_zero\_6bits presente en BitstreamToDecode) contiene todos los nuh\_reserved\_zero\_6bits presentes en la secuencia de vídeo codificada del SPS (que es un subconjunto del conjunto de nuh\_reserved\_zero\_6bits en BitstreamToDecode). Debido a que el conjunto de nuh\_reserved\_zero\_6bits de TargetOp puede contener necesariamente todos los valores de nuh\_reserved\_zero\_6bits presentes en la secuencia de vídeo codificada asociada al SPS, la estructura sintáctica hrd\_parameters( ) en el SPS siempre puede aplicarse a TargetOp. Sin embargo, no todos los SPS tienen estructuras sintácticas hrd\_parameters( ). Si un SPS tiene una estructura sintáctica hrd\_parameters( ) y el conjunto de nuh\_reserved\_zero\_6bits presente en BitstreamToDecode contiene todos los nuh\_reserved\_zero\_6bits presentes en la secuencia de vídeo codificada del SPS, entonces se debe usar la estructura sintáctica hrd\_parameters( ) del SPS. Debido a que no todos los SPS tienen estructuras sintácticas hrd\_parameters( ), aún se puede seleccionar el VPS.

**[0196]** Además, como se muestra anteriormente en las modificaciones de la sección E.2.2 del borrador de trabajo 8 de HEVC, cuando un dispositivo realiza una prueba de conformidad de flujo de bits, el descodificador de vídeo puede determinar que el flujo de bits no se ajusta a la norma de codificación de vídeo cuando, para todos los conjuntos de parámetros HRD en una secuencia de vídeo codificada, más de un conjunto de parámetros HRD se aplica al mismo punto de funcionamiento. Además, cuando el dispositivo realiza una prueba de conformidad de flujo de bits, el descodificador de vídeo puede determinar que el flujo de bits no se ajusta a la norma de codificación de vídeo cuando más de un conjunto de parámetros HRD en el VPS se aplican al mismo punto de funcionamiento. Además, cuando el dispositivo realiza la prueba de descodificación de flujo de bits, el dispositivo puede determinar que el flujo de bits no se ajusta a la norma de codificación de vídeo cuando el VPS incluye un conjunto de parámetros HRD que se aplican a los puntos de funcionamiento que tienen conjuntos de id de capa que solo contienen el valor 0.

**[0197]** La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que está configurado para implementar las técnicas de esta divulgación. La FIG. 3 se proporciona con propósitos explicativos y no se limita a las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con propósitos explicativos, esta divulgación describe un descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden aplicarse a otras normas o procedimientos de codificación.

**[0198]** En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación de entropía 150, una unidad de procesamiento de predicción 152, una unidad de cuantificación inversa 154, una unidad de procesamiento de transformada inversa 156, una unidad de reconstrucción 158, una unidad de filtro 160 y una memoria intermedia de imágenes descodificadas 162. La unidad de procesamiento de predicción 152 incluye una

unidad de compensación de movimiento 164 y una unidad de procesamiento de intrapredicción 166. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

**[0199]** Una memoria intermedia de imágenes codificadas (CPB) 151 puede recibir y almacenar datos de vídeo codificados (por ejemplo, unidades NAL) de un flujo de bits. La unidad de descodificación de entropía 150 puede recibir unidades NAL desde la CPB 151 y analizar sintácticamente las unidades NAL para descodificar elementos sintácticos. La unidad de descodificación de entropía 150 puede realizar la descodificación de entropía de elementos sintácticos sometidos a codificación de entropía en las unidades NAL. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de procesamiento de transformada inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 160 pueden generar datos de vídeo descodificados basándose en los elementos sintácticos extraídos del flujo de bits.

**[0200]** Las unidades NAL del flujo de bits pueden incluir unidades NAL de segmentos codificados. Como parte de la descodificación del flujo de bits, la unidad de descodificación de entropía 150 puede extraer y realizar la descodificación de entropía de elementos sintácticos de las unidades NAL de segmentos codificados. Cada uno de los segmentos codificados puede incluir una cabecera de segmento y datos de segmento. La cabecera de segmento puede contener elementos sintácticos pertenecientes a un segmento. Los elementos sintácticos de la cabecera de segmento pueden incluir un elemento sintáctico que identifica un PPS asociado a una imagen que contiene el segmento.

**[0201]** Además de descodificar los elementos sintácticos del flujo de bits, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en una CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Realizando la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir bloques residuales de la CU.

**[0202]** Como parte de la realización de una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 154 puede realizar la cuantificación inversa, es decir, la descuantificación, de los bloques de coeficientes asociados a la TU. La unidad de cuantificación inversa 154 puede usar un valor de QP asociado a la CU de la TU para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para que la unidad de cuantificación inversa 154 lo aplique. Es decir, la relación de compresión, es decir, la relación del número de bits usados para representar la secuencia original y la comprimida, puede controlarse ajustando el valor del QP usado al cuantificar los coeficientes de transformada. La relación de compresión también puede depender del procedimiento de codificación de entropía empleado.

**[0203]** Después de que la unidad de cuantificación inversa 154 haya realizado la cuantificación inversa de un bloque de coeficientes, la unidad de procesamiento de transformada inversa 156 puede aplicar una o más transformadas inversas al bloque de coeficientes a fin de generar un bloque residual asociado a la TU. Por ejemplo, la unidad de procesamiento de transformada inversa 156 puede aplicar una DCT inversa, una transformada entera inversa, una transformada inversa de Karhunen-Loeve (KLT), una transformada de rotación inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes.

**[0204]** Si se codifica una PU usando intrapredicción, la unidad de procesamiento de intrapredicción 166 puede realizar una intrapredicción para generar bloques predictivos para la PU. La unidad de procesamiento de intrapredicción 166 puede usar un modo de intrapredicción para generar los bloques predictivos de luma, Cb y Cr para la PU basándose en los bloques de predicción de unas PU vecinas en el espacio. La unidad de procesamiento de intrapredicción 166 puede determinar el modo de intrapredicción para la PU basándose en uno o más elementos sintácticos descodificados del flujo de bits.

**[0205]** La unidad de procesamiento de predicción 152 puede construir una primera lista de imágenes de referencia (RefPicList0) y una segunda lista de imágenes de referencia (RefPicList1) basándose en elementos sintácticos extraídos del flujo de bits. Además, si se codifica una PU usando interpredicción, la unidad de descodificación de entropía 150 puede extraer información de movimiento para la PU. La unidad de compensación de movimiento 164 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 164 puede generar, basándose en unos bloques de muestras del uno o más bloques de referencia para la PU, bloques predictivos de luma, Cb y Cr para la PU.

**[0206]** La unidad de reconstrucción 158 puede usar los bloques de transformada de luma, Cb y Cr, asociados a las TU de una CU y los bloques predictivos de luma, Cb y Cr de las PU de la CU, es decir, datos de intrapredicción o datos de interpredicción, según corresponda, para reconstruir los bloques de codificación de luma, Cb y Cr de la CU. Por ejemplo, la unidad de reconstrucción 158 puede añadir muestras de los bloques de transformada de luma, Cb y Cr a muestras correspondientes de los bloques predictivos de luma, Cb y Cr para reconstruir los bloques de codificación de luma, Cb y Cr de la CU.

**[0207]** La unidad de filtro 160 puede realizar una operación de eliminación de bloques para reducir los artefactos de bloque asociados a los bloques de codificación de luma, Cb y Cr de la CU. El descodificador de vídeo 30 puede

almacenar los bloques de codificación de luma, Cb y Cr de la CU en la memoria intermedia de imágenes descodificadas 162. La memoria intermedia de imágenes descodificadas 162 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intrapredicción y 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 luma, Cb y Cr de la memoria intermedia de imágenes descodificadas 162, operaciones de intrapredicción o de interpredicción en las PU de otras CU. De esta manera, el descodificador de vídeo 30 puede descodificar, a partir del flujo de bits, niveles de coeficientes de transformada del bloque de coeficientes de luma significativo, realizar la cuantificación inversa de los niveles de coeficientes de transformada, aplicar una transformada a los niveles de coeficientes de transformada para generar un bloque de transformada, generar, en función de, al menos en parte, el bloque de transformada, un bloque de codificación, y proporcionar el bloque de codificación para su visualización.

**[0208]** La FIG. 4 es un diagrama de flujo que ilustra un ejemplo de funcionamiento 200 de un dispositivo, de acuerdo con una o más técnicas de esta divulgación. La operación 200 puede realizarse mediante el codificador de vídeo 20, el descodificador de vídeo 30, un dispositivo adicional 21 u otro dispositivo. Como se ilustra en el ejemplo de la FIG. 4, el dispositivo puede seleccionar, de entre un conjunto de parámetros de HRD hipotético (por ejemplo, estructuras sintácticas `hrd_parameters`) en un VPS y un conjunto de parámetros HRD en un SPS, un conjunto de parámetros HRD aplicables a un punto de funcionamiento particular de un flujo de bits (202). Además, el dispositivo puede realizar, basándose al menos en parte en el conjunto de parámetros HRD aplicables al punto de funcionamiento particular, una operación HRD en un subconjunto de flujo de bits asociado al punto de funcionamiento particular (204). Por ejemplo, el dispositivo puede realizar una prueba de conformidad de flujo de bits o una prueba de conformidad de descodificador.

**[0209]** La FIG. 5 es un diagrama de flujo que ilustra un ejemplo de funcionamiento 250 de un dispositivo, de acuerdo con una o más técnicas de esta divulgación. La operación 200 puede realizarse mediante el codificador de vídeo 20, el descodificador de vídeo 30, un dispositivo adicional 21 u otro dispositivo. Como se ilustra en el ejemplo de la FIG. 5, el dispositivo puede realizar una prueba de conformidad de flujo de bits que determina si un flujo de bits se ajusta a una norma de codificación de vídeo (252). El dispositivo puede realizar un proceso de descodificación como parte de la realización de una prueba de conformidad de flujo de bits (254).

**[0210]** Como se ilustra en el ejemplo de la FIG. 5, al realizarse el proceso de descodificación, el dispositivo puede realizar un proceso de extracción de flujo de bits para extraer, del flujo de bits, una representación de punto de funcionamiento de un punto de funcionamiento definido por un conjunto objetivo de identificadores de capa y un identificador temporal objetivo más alto (256). El conjunto objetivo de identificadores de capa puede contener valores de elementos sintácticos de identificador de capa presentes en la representación de punto de funcionamiento. El conjunto objetivo de identificadores de capa puede ser un subconjunto de valores de elementos sintácticos de identificador de capa del flujo de bits. El identificador temporal objetivo más alto puede ser igual al mayor identificador temporal presente en la representación de punto de funcionamiento, siendo el identificador temporal objetivo más alto menor que o igual al mayor identificador temporal presente en el flujo de bits. Además, el dispositivo puede descodificar unidades NAL de la representación de punto de funcionamiento (258).

**[0211]** La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de funcionamiento HRD 300 de un dispositivo, de acuerdo con una o más técnicas de esta divulgación. La operación HRD 300 se puede realizar mediante el codificador de vídeo 20, el descodificador de vídeo 30, un dispositivo adicional 21 u otro dispositivo. Otros dispositivos pueden incluir un verificador de flujo de bits en conformidad que toma un flujo de bits como entrada y proporciona una indicación de si el flujo de bits de entrada es un flujo de bits en conformidad o no. En algunos ejemplos, la operación HRD 300 puede determinar la conformidad de un flujo de bits con una norma de codificación de vídeo. En otros ejemplos, la operación HRD 300 puede determinar la conformidad de un descodificador con una norma de codificación de vídeo. Como parte de la realización de la operación HRD 300, el dispositivo puede determinar el identificador temporal más alto de un subconjunto de flujo de bits asociado a un punto de funcionamiento seleccionado de un flujo de bits (302). Además, el dispositivo puede determinar, basándose en el identificador temporal más alto, un elemento sintáctico particular de entre una matriz de elementos sintácticos (por ejemplo, `sps_max_num_reorder_pics[i]`, `sps_max_dec_pic_buffering[i]` y `cpb_cnt_minus1[i]`) (304). El dispositivo puede usar el elemento sintáctico particular en la operación HRD (306).

**[0212]** En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en o transmitirse a través de un medio legible por ordenador como una o más instrucciones o código, y 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 medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa informático de 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) 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 tangible al que se pueda 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 esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

5 **[0213]** A modo de ejemplo, y no de limitación, dichos 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 se pueda usar para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión recibe adecuadamente 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 unas 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 están incluidos en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se refieren a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, donde algunos discos reproducen normalmente datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior se deben incluir también dentro del alcance de los medios legibles por ordenador.

20 **[0214]** 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 *in situ* (FPGA) u otros circuitos lógicos integrados o discretos equivalentes pueden ejecutar las instrucciones. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquier estructura anterior 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 se puede proporcionar en módulos de hardware y/o de software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

30 **[0215]** Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores, como se describe anteriormente, junto con software y/o firmware adecuados.

**REIVINDICACIONES**

1. Un procedimiento de procesamiento de datos de vídeo de acuerdo con la norma de codificación de vídeo de alta eficiencia (HEVC), comprendiendo el procedimiento:

5 seleccionar un conjunto de parámetros de descodificador de referencia hipotético (HRD) de entre uno o más conjuntos de parámetros de descodificador de referencia hipotético (HRD) en un conjunto de parámetros de vídeo (VPS) en un flujo de bits y un conjunto de parámetros HRD en un conjunto de parámetros de secuencia (SPS) en el flujo de bits como un conjunto de parámetros HRD aplicables a un punto de funcionamiento particular de un flujo de bits que comprende datos de vídeo codificados, en el que  
 10 seleccionar el conjunto de parámetros HRD aplicables al punto de funcionamiento particular comprende determinar que el conjunto de parámetros HRD en el SPS es aplicable al punto de funcionamiento particular cuando un conjunto de identificadores de capa del punto de funcionamiento particular contiene un conjunto de todos los identificadores de capa presentes en un subconjunto de flujo de bits que se va a descodificar o que se está probando, y, de otro modo, determinar que uno de los uno o más conjuntos de parámetros HRD en el VPS puede aplicarse al punto de funcionamiento particular; y

realizar, en función de, al menos en parte, el conjunto seleccionado de parámetros HRD aplicables al punto de funcionamiento particular, una operación HRD en el subconjunto de flujo de bits.

2. El procedimiento según la reivindicación 1, en el que el conjunto de parámetros HRD aplicables al punto de funcionamiento particular incluye parámetros que especifican un retardo de eliminación inicial de memoria intermedia de imágenes de codificación (CPB), un tamaño de CPB, una velocidad de transferencia de bits, un retardo de salida inicial de memoria intermedia de imágenes descodificadas (DPB) y un tamaño de DPB.

3. El procedimiento según la reivindicación 1, que comprende además:

determinar un conjunto de identificadores de capa objetivo del punto de funcionamiento particular que contiene cada identificador de capa presente en el subconjunto de flujo de bits, y en el que el conjunto de identificadores de capa objetivo del punto de funcionamiento particular es un subconjunto de identificadores de capa presente en el flujo de bits, y

determinar un identificador temporal objetivo del punto de funcionamiento particular que es igual al mayor identificador temporal presente en el subconjunto de flujo de bits, en el que el identificador temporal objetivo del punto de funcionamiento particular es menor que o igual al mayor identificador temporal presente en el flujo de bits.

4. El procedimiento según la reivindicación 3, en el que seleccionar el conjunto de parámetros HRD comprende seleccionar el conjunto de parámetros HRD en el SPS en respuesta a la determinación de que un conjunto de identificadores de capa especificados en el SPS es idéntico al conjunto de identificadores de capa objetivo del punto de funcionamiento particular.

5. El procedimiento según la reivindicación 1, en el que realizar la operación HRD comprende realizar una prueba de conformidad de flujo de bits que determina si el flujo de bits se ajusta a una norma de codificación de vídeo.

6. El procedimiento según la reivindicación 5, en el que realizar la prueba de conformidad de flujo de bits comprende determinar que el flujo de bits no se ajusta a la norma de codificación de vídeo cuando, para todos los conjuntos de parámetros HRD en una secuencia de vídeo codificada, se aplica más de un conjunto de parámetros HRD al mismo punto de funcionamiento.

7. El procedimiento según la reivindicación 5, en el que realizar la prueba de conformidad de flujo de bits comprende determinar que el flujo de bits no se ajusta a la norma de codificación de vídeo cuando más de un conjunto de parámetros HRD en el VPS se aplica al mismo punto de funcionamiento.

8. El procedimiento según la reivindicación 5, en el que realizar la prueba de conformidad de flujo de bits comprende determinar que el flujo de bits no se ajusta a la norma de codificación de vídeo cuando el VPS incluye un conjunto de parámetros HRD que se aplica a los puntos de funcionamiento que tienen conjuntos de identificadores de capa que solo contienen el valor 0.

9. Un dispositivo para procesar datos de vídeo según la norma de codificación de vídeo de alta eficiencia (HEVC) que comprende:

medios para seleccionar un conjunto de parámetros de descodificador de referencia hipotético (HRD) de entre uno o más conjuntos de parámetros de descodificador de referencia hipotético (HRD) en un conjunto de parámetros de vídeo (VPS) en un flujo de bits y un conjunto de parámetros HRD en un conjunto de parámetros de secuencia (SPS) en el flujo de bits como un conjunto de parámetros HRD

5 aplicables a un punto de funcionamiento particular de un flujo de bits que comprende datos de vídeo codificados, en el que los medios de selección comprenden medios para determinar que el conjunto de parámetros HRD en el SPS puede aplicarse al punto de funcionamiento particular cuando un conjunto de identificadores de capa del punto de funcionamiento particular contiene un conjunto de todos los identificadores de capa presentes en un subconjunto de flujo de bits que se va a descodificar o que se está probando y, de lo contrario, determinar que uno de los uno o más conjuntos de parámetros HRD en el VPS puede aplicarse al punto de funcionamiento particular; y

10 medios para realizar, en función de, al menos en parte, el conjunto seleccionado de parámetros HRD aplicables al punto de funcionamiento particular, una operación HRD en un subconjunto de flujo de bits.

15 **10.** El dispositivo según la reivindicación 9, en el que el conjunto de parámetros HRD aplicables al punto de funcionamiento particular incluye parámetros que especifican un retardo de eliminación inicial de memoria intermedia de imágenes de codificación (CPB), un tamaño de CPB, una velocidad de transferencia de bits, un retardo de salida inicial de memoria intermedia de imágenes descodificadas (DPB) y un tamaño de DPB.

**11.** El dispositivo según la reivindicación 9, que comprende además:

20 medios para determinar un conjunto de identificadores de capa objetivo del punto de funcionamiento particular que contiene cada identificador de capa presente en el subconjunto de flujo de bits, y en el que el conjunto de identificadores de capa objetivo del punto de funcionamiento particular es un subconjunto de identificadores de capa presente en el flujo de bits, y

25 medios para determinar un identificador temporal objetivo del punto de funcionamiento particular que es igual al mayor identificador temporal presente en el subconjunto de flujo de bits, donde el identificador temporal objetivo del punto de funcionamiento particular es menor que o igual al mayor identificador temporal presente en el flujo de bits.

30 **12.** El dispositivo según la reivindicación 11, que comprende además medios para seleccionar el conjunto de parámetros HRD en el SPS en respuesta a la determinación de que un conjunto de identificadores de capa especificados en el SPS es idéntico al conjunto de identificadores de capa objetivo del punto de funcionamiento particular.

35 **13.** El dispositivo según la reivindicación 9, que comprende además medios para realizar una prueba de conformidad de flujo de bits que determina si el flujo de bits se ajusta a una norma de codificación de vídeo.

**14.** El dispositivo según la reivindicación 13, que comprende además medios para determinar que el flujo de bits no se ajusta a la norma de codificación de vídeo cuando:

40 para todos los conjuntos de parámetros HRD en una secuencia de vídeo codificada, se aplica más de un conjunto de parámetros HRD al mismo punto de funcionamiento; y/o

más de un conjunto de parámetros HRD en el VPS se aplica al mismo punto de funcionamiento; y/o

45 el VPS incluye un conjunto de parámetros HRD que se aplica a los puntos de funcionamiento que tienen conjuntos de identificadores de capa que solo contienen el valor 0.

50 **15.** Un medio de almacenamiento de datos legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando son ejecutadas por uno o más procesadores de un dispositivo, configuran el dispositivo para que realice el procedimiento según cualquiera de las reivindicaciones 1 a 8.

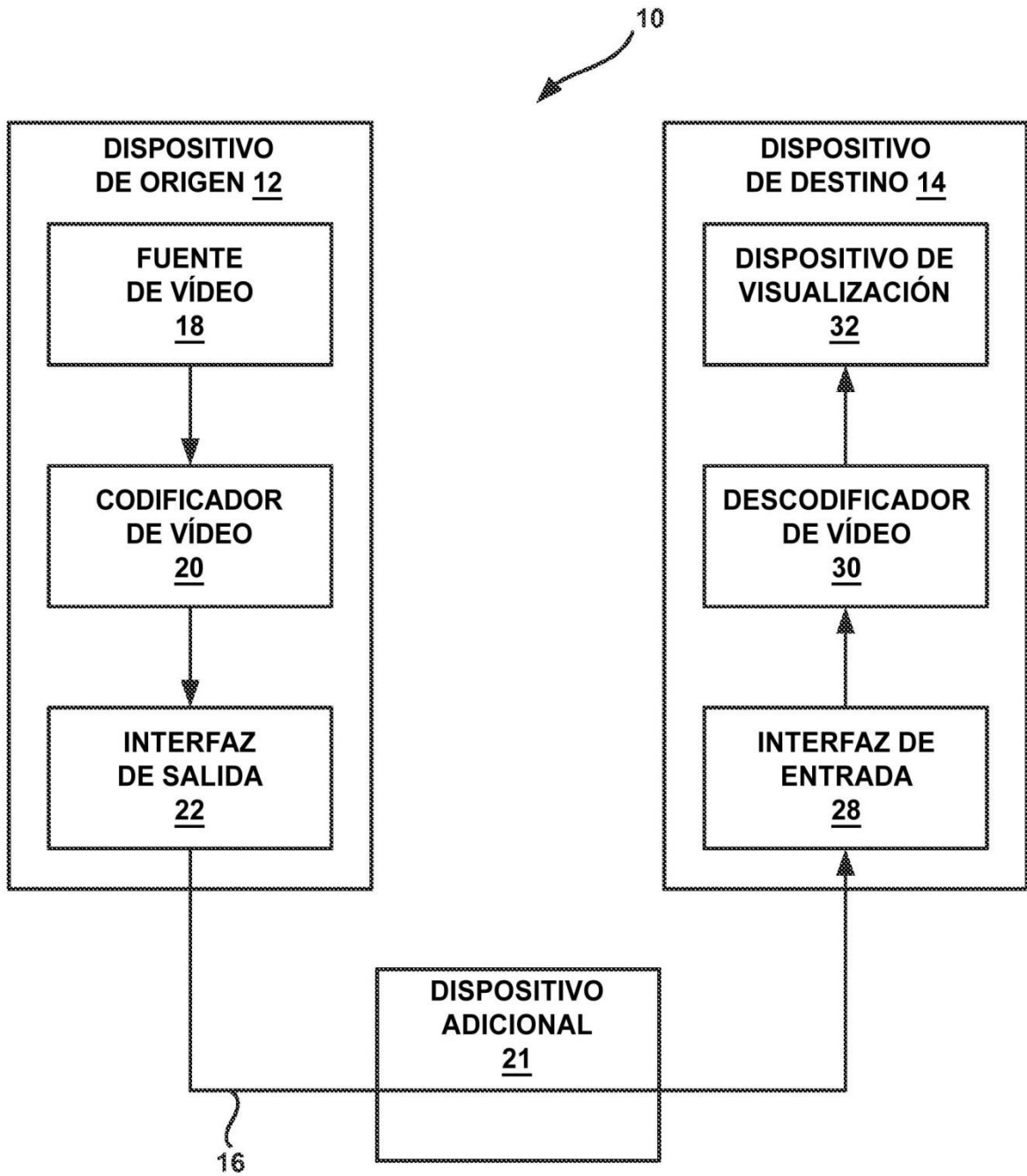


FIG. 1



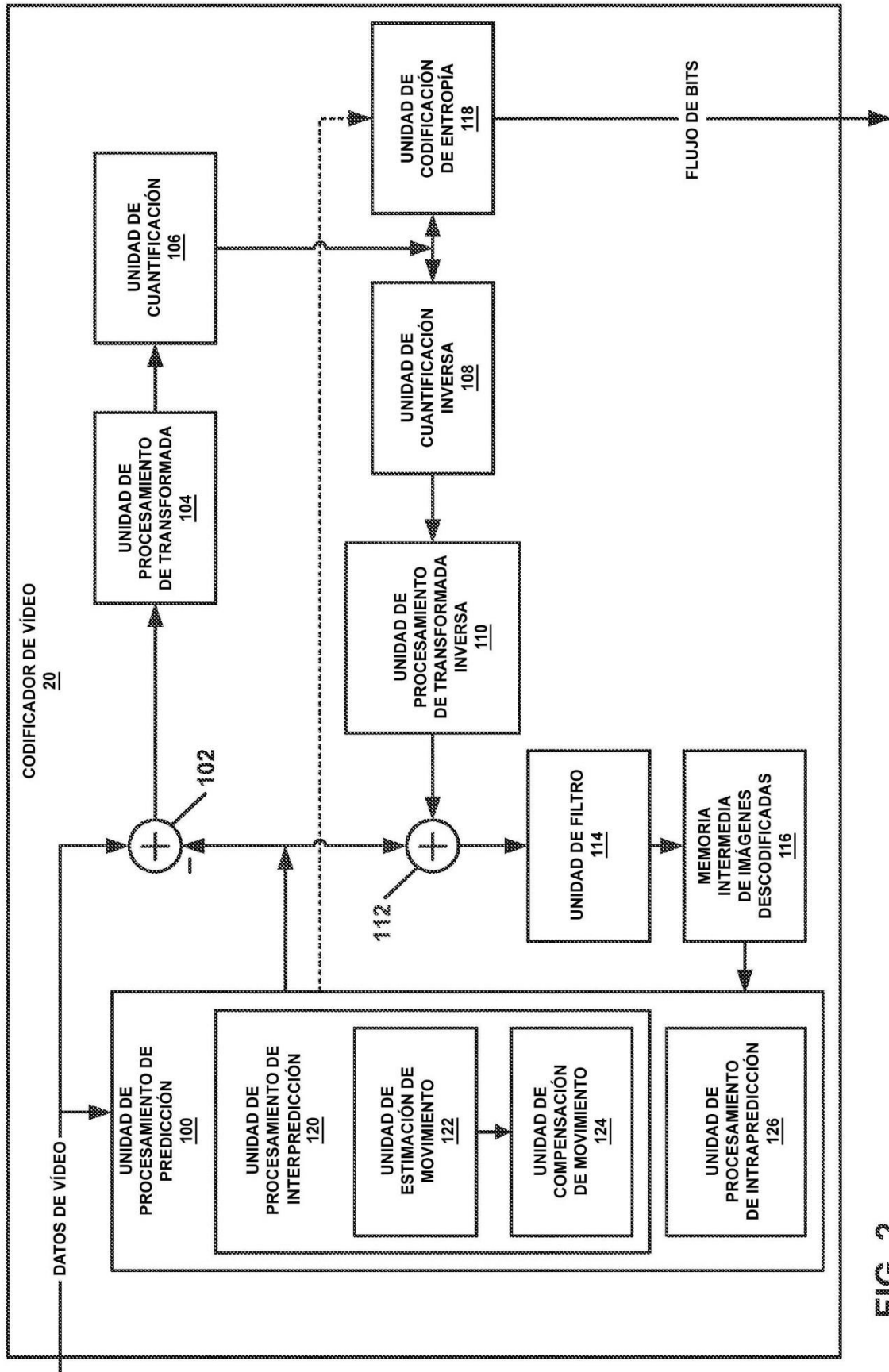


FIG. 2

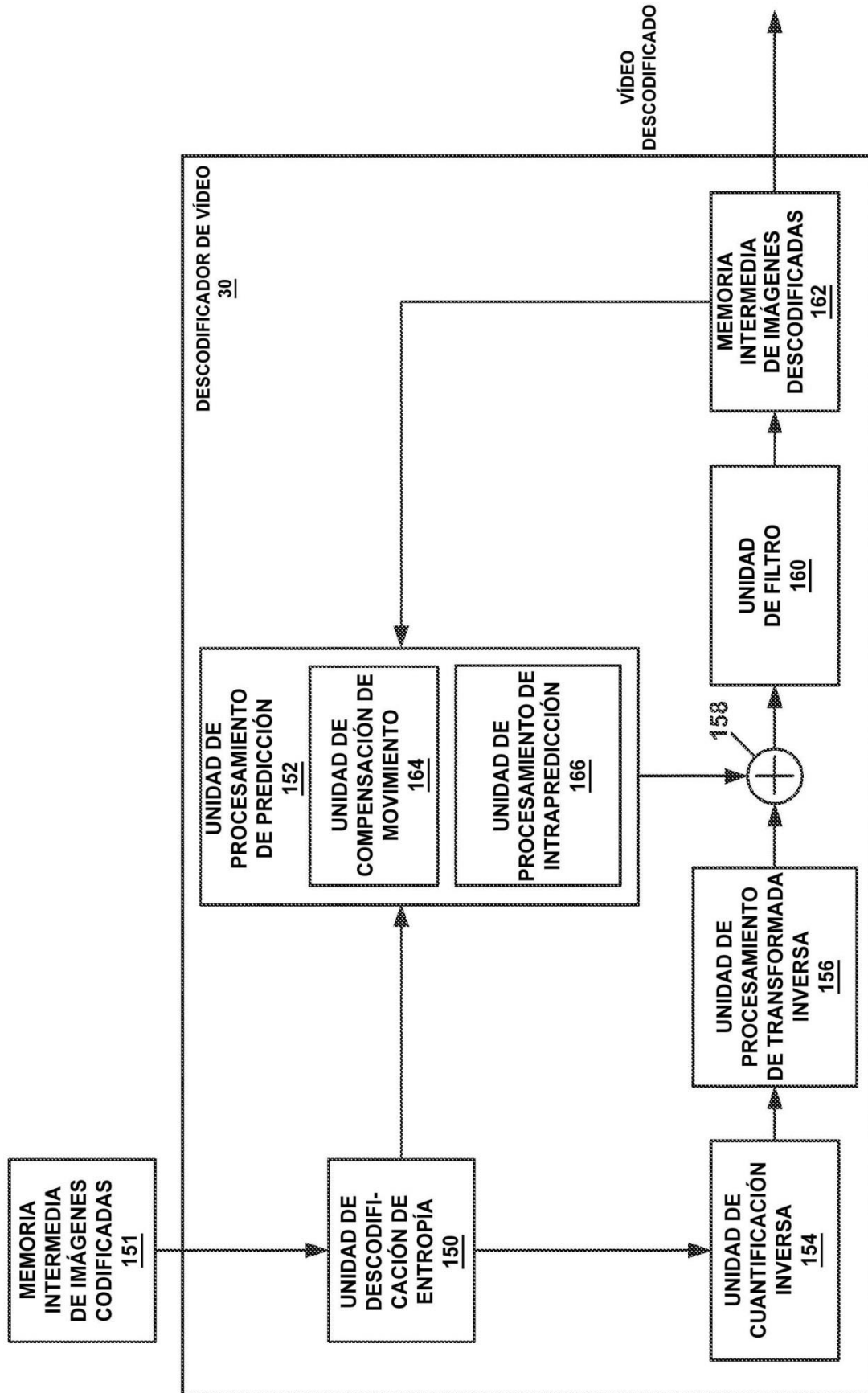


FIG. 3

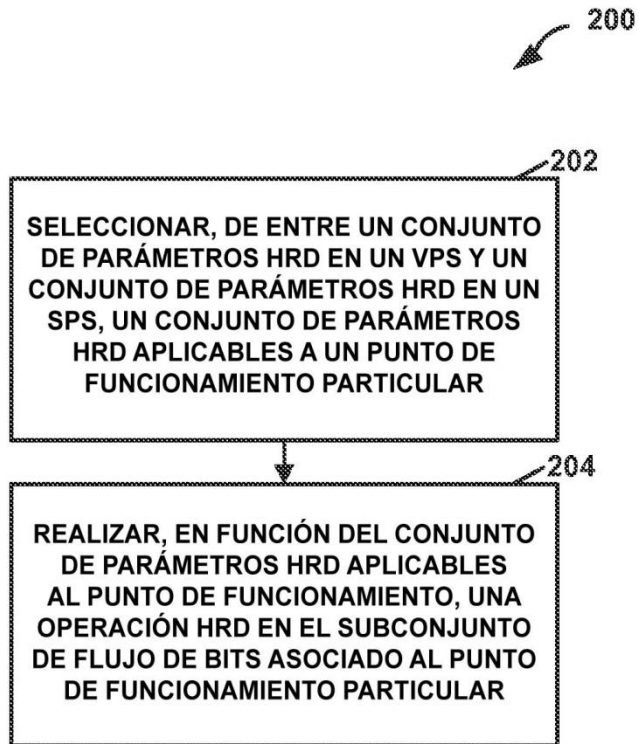


FIG. 4

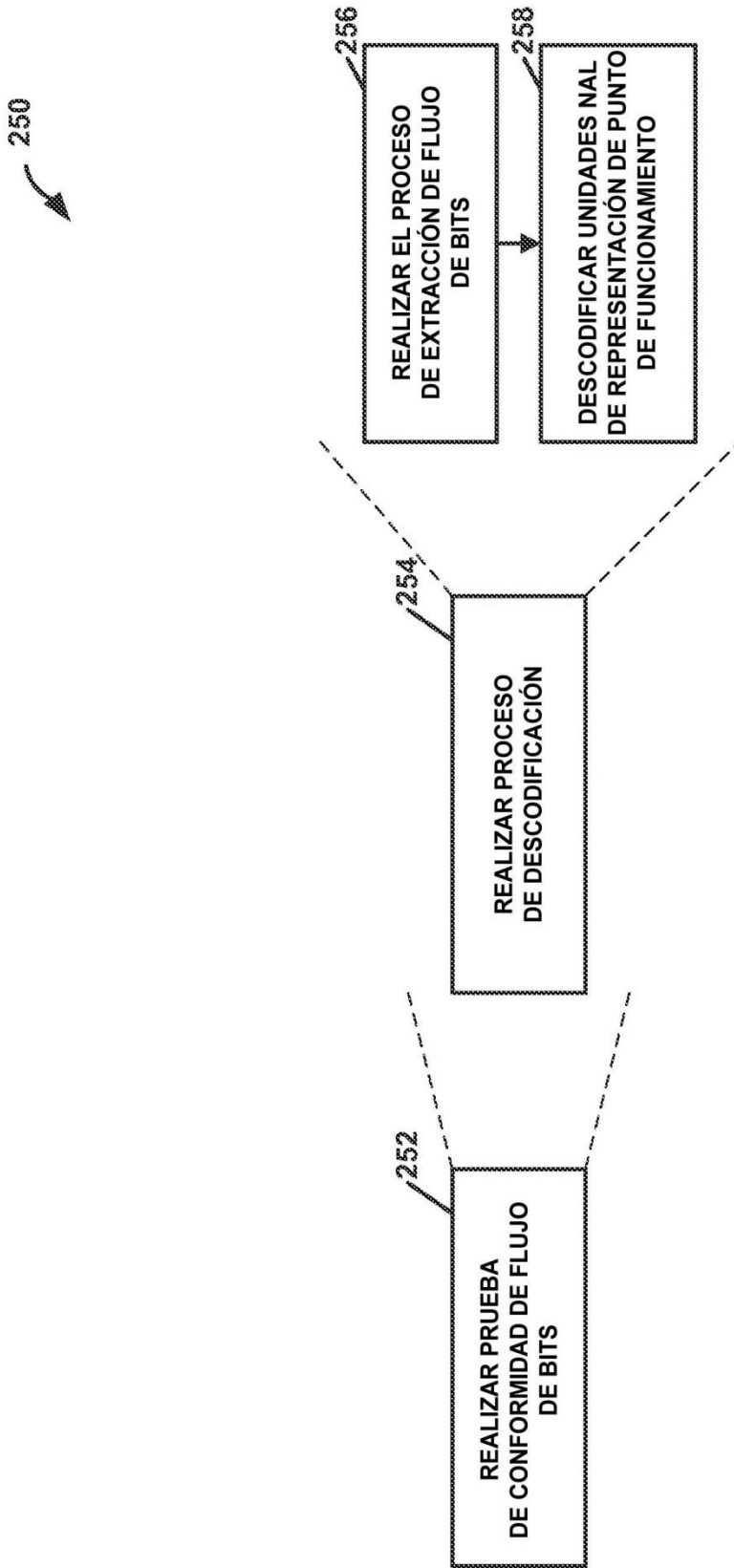


FIG. 5

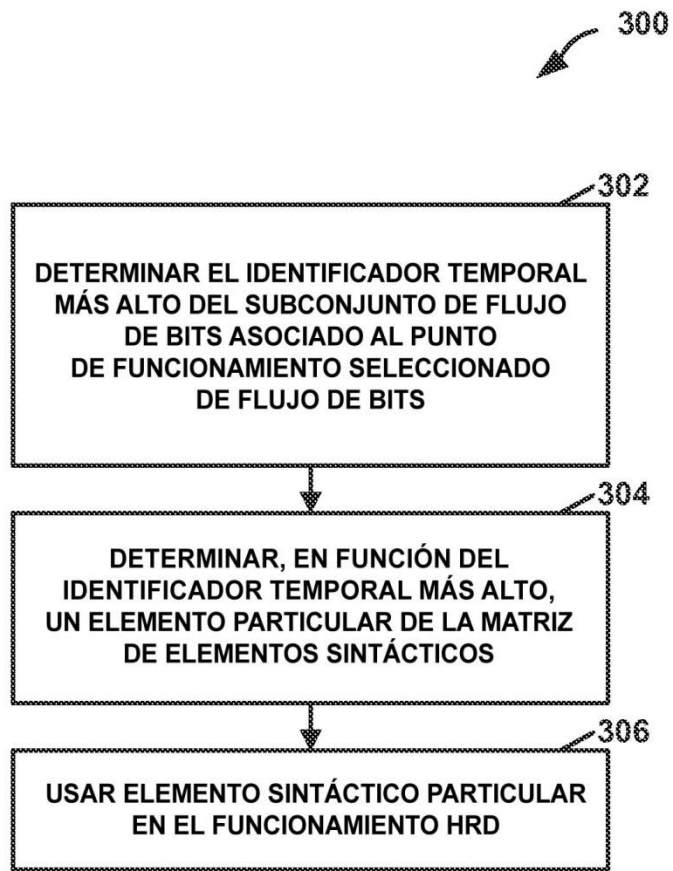


FIG. 6