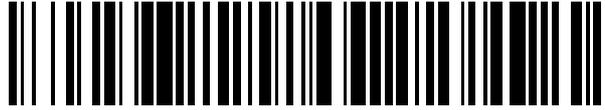


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 724 925**

51 Int. Cl.:

|                    |           |
|--------------------|-----------|
| <b>H04N 19/597</b> | (2014.01) |
| <b>H04N 19/70</b>  | (2014.01) |
| <b>H04N 19/30</b>  | (2014.01) |
| <b>H04N 19/573</b> | (2014.01) |
| <b>H04N 19/58</b>  | (2014.01) |

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **03.06.2014 PCT/US2014/040611**
- 87 Fecha y número de publicación internacional: **11.12.2014 WO14197428**
- 96 Fecha de presentación y número de la solicitud europea: **03.06.2014 E 14735758 (6)**
- 97 Fecha y número de publicación de la concesión europea: **06.02.2019 EP 3005705**

54 Título: **Modelo hipotético de descodificador de referencia y conformidad para imágenes omitidas de acceso aleatorio de capa cruzada**

30 Prioridad:

**03.06.2013 US 201361830512 P**  
**02.06.2014 US 201414293829**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**17.09.2019**

73 Titular/es:

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

72 Inventor/es:

**RAMASUBRAMONIAN, ADARSH KRISHNAN;**  
**WANG, YE-KUI;**  
**CHEN, YING y**  
**RAPAKA, KRISHNAKANTH**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

ES 2 724 925 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

Modelo hipotético de descodificador de referencia y conformidad para imágenes omitidas de acceso aleatorio de capa cruzada

5 [0001] Esta solicitud reivindica el beneficio de la solicitud provisional de patente de Estados Unidos n.º 61/830,512, presentada el 3 de junio de 2013.

## CAMPO TÉCNICO

10 [0002] Esta divulgación se refiere a la codificación y descodificación de vídeo.

## ANTECEDENTES

15 [0003] Las capacidades del vídeo digital pueden incorporarse a una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión digital directa, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo 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) y las extensiones de tales normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital con más eficiencia implementando dichas técnicas de compresión de vídeo.

20 [0004] Las técnicas de compresión de vídeo llevan a cabo la predicción espacial (dentro de las imágenes) y/o la predicción temporal (entre imágenes) para eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (es decir, una trama o una parte de vídeo de una trama de vídeo) puede dividirse en bloques de vídeo. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o la predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia.

30 [0005] 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 vaya 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 con los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxel en un dominio de transformada, dando como resultado coeficientes residuales, que pueden cuantificarse entonces. Los coeficientes cuantificados, inicialmente dispuestos en una matriz bidimensional, pueden escanearse con el fin de producir un vector unidimensional de coeficientes, y puede aplicarse la codificación por entropía para lograr incluso más compresión.

40 [0006] Un flujo de bits de codificación de múltiples visualizaciones puede generarse codificando visualizaciones, por ejemplo, desde múltiples perspectivas. Se han desarrollado algunas normas de vídeo tridimensional (3D) que hacen uso de aspectos de codificación de múltiples visualizaciones. Por ejemplo, diferentes visualizaciones pueden transmitir visualizaciones de ojo izquierdo y derecho para soportar vídeo 3D. De forma alternativa, algunos procesos de codificación de vídeo 3D pueden aplicar la denominada codificación de múltiples visualizaciones con profundidad. En la codificación de múltiples visualizaciones más profundidad, un flujo de bits de vídeo 3D puede contener no solo componentes de visualización de textura, sino también componentes de visualización de profundidad. Por ejemplo, cada visualización puede comprender un componente de visualización de textura y un componente de visualización de profundidad.

55 [0007] Hannuksela, "MV-HEVC/SHVC HLS: Layer-wise startup of the decoding process [MV-HEVC/SHVC HLS: Arranque por capas del proceso de descodificación]", 104. REUNIÓN DE MPEG; 22-4-2013 - 26-4-2013, INCHEON, (GRUPO DE EXPERTOS DE IMAGEN EN MOVIMIENTO O ISO/IEC JTC1/SC29/WG11), proporciona un procedimiento para permitir el inicio del proceso de descodificación desde una imagen CRA (con nuh\_layer\_id igual a 0 y un valor POC particular), cuando algunas de las imágenes con el mismo valor POC y con nuh\_layer\_id mayor que 0 son imágenes que no son IRAP.

65

**SUMARIO**

5 [0008] En general, una o más técnicas de esta divulgación definen, para un proceso de codificación de vídeo, conformidad adicional cuando las imágenes de capa de mejora que no son correctamente descodificables no están presentes en el flujo de bits. Además, una o más técnicas de esta divulgación definen un proceso completo de descodificación para imágenes de referencia no disponibles cuando se descodifica una imagen omitida de acceso aleatorio de capa cruzada (CL-RAS) en una unidad de acceso (AU) de punto de acceso intra aleatorio (IRAP) y cuando se indica que las imágenes CL-RAS no deben emitirse, y se proporcionan parámetros de descodificador de referencia hipotético (HRD) para ambos casos cuando las imágenes CL-RAS están presentes o ausentes.

10 [0009] En un ejemplo, esta divulgación describe un procedimiento de descodificación de datos de vídeo, comprendiendo el procedimiento: determinar, basándose en un valor, si todas las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de bits de datos de vídeo; y reconstrucción de imágenes de datos de vídeo basadas, al menos en parte, en elementos sintácticos descodificados del flujo de bits de datos de vídeo.

15 [0010] En otro ejemplo, esta divulgación describe un procedimiento de procesamiento de datos de vídeo, comprendiendo el procedimiento: generar un mensaje SEI que establece una variable que especifica si las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de bits que incluye una secuencia de bits que forman una representación de imágenes codificadas y datos asociados; e incluyendo el mensaje SEI en el flujo de bits.

20 [0011] En otro ejemplo, esta divulgación describe un dispositivo que comprende: un medio de almacenamiento de datos configurado para almacenar datos de vídeo, y uno o más procesadores configurados para: determinar, basándose en un valor, si todas las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de datos de vídeo de datos; y reconstruir imágenes de los datos de vídeo basadas, al menos en parte, en elementos sintácticos descodificados del flujo de bits de datos de vídeo.

25 [0012] En otro ejemplo, esta divulgación describe un dispositivo que comprende: un medio de almacenamiento de datos configurado para almacenar datos de vídeo, y uno o más procesadores configurados para: generar un mensaje SEI que establece una variable que especifica si las imágenes CL-RAS de un IRAP la unidad de acceso están presentes en un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados; e incluir el mensaje SEI en el flujo de bits.

30 [0013] En otro ejemplo, esta divulgación describe un dispositivo que comprende: medios para determinar, basándose en un valor, si todas las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de bits de datos de vídeo; y medios para reconstruir imágenes de los datos de vídeo basándose, al menos en parte, en elementos sintácticos descodificados del flujo de bits de datos de vídeo.

35 [0014] En otro ejemplo, esta divulgación describe un dispositivo que comprende: medios para generar un mensaje SEI que establece una variable que especifica si las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados; y medios para incluir el mensaje SEI en el flujo de bits.

40 [0015] En otro ejemplo, esta divulgación describe un medio de almacenamiento de datos legible por ordenador que tiene almacenado en el mismo instrucciones que, al ejecutarse, hacen que un dispositivo: determine, basándose en un valor, si todas las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de datos de vídeo; y reconstruya imágenes de los datos de vídeo basadas, al menos en parte, en elementos sintácticos descodificados del flujo de bits de datos de vídeo.

45 [0016] En otro ejemplo, esta divulgación describe un medio de almacenamiento de datos legible por ordenador que tiene almacenado en el mismo instrucciones que, al ejecutarse, hacen que un dispositivo: genere un mensaje SEI que establece una variable que especifica si las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados; e incluya el mensaje SEI en el flujo de bits.

50 [0017] La información de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y la descripción siguiente. Otras características, objetivos y ventajas resultarán evidentes a partir de la descripción, de los dibujos y de las reivindicaciones.

**BREVE DESCRIPCIÓN DE LOS DIBUJOS**

60 [0018]

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

65 La FIG. 2 es un diagrama conceptual que ilustra una imagen de acceso aleatorio limpio ("CRA") y las imágenes principales.

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

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

10 La FIG. C-1 es un diagrama conceptual que ilustra una estructura de ejemplo de flujos de bytes y flujos de unidades de Capa de Abstracción de Red ("NAL") para comprobaciones de conformidad del Descodificador de Referencia Hipotético ("HRD").

La FIG. C-2 es un diagrama conceptual que ilustra un modelo de memoria intermedia HRD de ejemplo.

15 La FIG. 5 es un diagrama conceptual que ilustra imágenes CL-RAS, de acuerdo con una o más técnicas de esta divulgación.

La FIG. 6A es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo, de acuerdo con una o más técnicas de esta divulgación.

20 La FIG. 6B es un diagrama de flujo que ilustra un funcionamiento de ejemplo de un dispositivo, de acuerdo con una o más técnicas de esta divulgación.

25 La FIG. 6C es un diagrama de flujo que ilustra un funcionamiento de ejemplo de un descodificador de vídeo, de acuerdo con una o más técnicas de esta divulgación.

## DESCRIPCIÓN DETALLADA

30 **[0019]** En general, esta divulgación describe técnicas para codificación de vídeo de múltiples capas. Como se describe en el presente documento, un flujo de bits de datos de vídeo puede comprender una secuencia de bits que forme una representación de imágenes codificadas y datos asociados. Las imágenes pueden pertenecer a diferentes capas y unidades de acceso. Las imágenes que pertenecen a la misma unidad de acceso pueden tener el mismo tiempo de salida. Las imágenes que pertenecen a diferentes unidades de acceso pueden tener diferentes tiempos de salida. En la codificación de múltiples visualizaciones, las imágenes que pertenecen a capas diferentes pueden pertenecer a visualizaciones diferentes. En la codificación de vídeo escalable, las imágenes que pertenecen a capas superiores pueden proporcionar una mejor calidad visual.

35 **[0020]** Algunas de las imágenes son imágenes de punto de acceso intra-aleatorias (IRAP). Un codificador de vídeo puede descodificar una imagen IRAP sin hacer referencia a ninguna imagen que ocurra antes de la imagen IRAP en orden de descodificación. Además, las imágenes pueden incluir imágenes omitidas de acceso aleatorio de capa cruzada (CL-RAS). En algunos ejemplos, una imagen CL-RAS es una imagen que cumple uno de los siguientes criterios. En primer lugar, la imagen no es una imagen IRAP sino que está en una unidad de acceso IRAP. En segundo lugar, la imagen no es una imagen IRAP, la imagen está en una unidad de acceso que sigue (en orden de descodificación) a una unidad de acceso IRAP, la imagen está en una unidad de acceso que precede a una unidad de acceso IRAP de cambio de capa, y la imagen es en una capa igual a una capa que contiene una imagen IRAP en la unidad de acceso IRAP de cambio de capa. En algunos casos, una imagen CL-RAS no se puede descodificar correctamente porque la imagen CL-RAS depende de que las imágenes se produzcan antes de la unidad de acceso IRAP anterior (o actual). En tales ejemplos, una unidad de acceso IRAP puede ser una unidad de acceso que contiene una o más imágenes IRAP y una imagen en una capa base de la unidad de acceso es una imagen IRAP. Además, en tales ejemplos, una unidad de acceso IRAP de cambio de capa puede ser una unidad de acceso que no es una unidad de acceso IRAP, pero que contiene al menos una imagen IRAP. Las imágenes CL-RAS pueden surgir debido al cambio del flujo de bits, el acceso aleatorio dentro de un flujo de bits y/o en otras circunstancias.

40 **[0021]** En algunos casos, un dispositivo (por ejemplo, un elemento de red consciente de medios (MANE)) puede eliminar imágenes CL-RAS de un flujo de bits. El dispositivo puede tener la libertad de eliminar las imágenes CL-RAS de un flujo de bits porque un descodificador de vídeo no podría descodificar las imágenes CL-RAS correctamente. Ventajosamente, la eliminación de imágenes CL-RAS del flujo de bits puede reducir la cantidad de datos en el flujo de bits. Sin embargo, la eliminación de imágenes CL-RAS del flujo de bits puede dar como resultado que un descodificador de vídeo (por ejemplo, un descodificador de referencia hipotético (HRD)) elimine las imágenes codificadas de una memoria intermedia de imágenes codificadas del descodificador de vídeo en momentos inadecuados. Por ejemplo, el descodificador de vídeo puede eliminar imágenes codificadas de la memoria intermedia de imágenes codificadas a una velocidad particular. En este ejemplo, el descodificador de vídeo puede descodificar imágenes codificadas eliminadas de la memoria intermedia de imágenes codificadas. Además, en este ejemplo, si el flujo de bits no incluye imágenes CL-RAS para una unidad de acceso IRAP, la memoria intermedia de imágenes codificadas puede incluir menos imágenes codificadas de lo que espera el descodificador de vídeo. En consecuencia, en este ejemplo, la velocidad a la que el descodificador de vídeo elimina imágenes codificadas de la memoria intermedia de imágenes codificadas puede ser grande, lo cual hace que el flujo de bits no sea conforme.

**[0022]** Por lo tanto, de acuerdo con una o más técnicas de esta divulgación, un descodificador de vídeo puede determinar, basándose en un valor, si todas las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de bits. En algunos ejemplos, el descodificador de vídeo puede establecer el valor basado al menos en parte en los datos en un mensaje de Información de Mejora Suplementaria (SEI) del flujo de bits. En otros ejemplos, un mecanismo fuera de un proceso de descodificación puede establecer el valor. Además, en algunos ejemplos, el descodificador de vídeo puede usar diferentes conjuntos de parámetros HRD, según sea apropiado, dependiendo de si todas las imágenes CL-RAS de la unidad de acceso IRAP están presentes en el flujo de bits. Los parámetros de HRD pueden incluir parámetros que controlan cómo el descodificador de vídeo elimina imágenes de la memoria intermedia de imágenes codificadas. De esta manera, los descodificadores de vídeo que cumplen con una norma de codificación de vídeo pueden descodificar flujos de bits que incluyen todas las imágenes CL-RAS de una unidad de acceso IRAP y flujos de bits que no incluyen todas las imágenes CL-RAS de una unidad de acceso IRAP.

**[0023]** La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo 10 de ejemplo que puede utilizar las técnicas de esta divulgación. 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 a la descodificación de vídeo.

**[0024]** 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. Por consiguiente, el dispositivo de origen 12 puede denominarse dispositivo de codificación de vídeo o aparato de codificación de vídeo. El dispositivo de destino 14 puede descodificar datos de vídeo codificados, generados por el dispositivo de origen 12. Por consiguiente, el dispositivo de destino 14 puede denominarse dispositivo de descodificación de vídeo o aparato de descodificación de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 pueden ser ejemplos de dispositivos de codificación de vídeo o aparatos de codificación de vídeo.

**[0025]** El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia variedad de dispositivos, incluyendo ordenadores de sobremesa, dispositivos informáticos móviles, notebooks (es decir, portátiles), ordenadores tipo tablet, descodificadores, equipos telefónicos portátiles tales como los denominados teléfonos "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores de coche o similares.

**[0026]** El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 a través de un canal 16. El canal 16 puede comprender uno o más medios o dispositivos capaces de mover los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender uno o más medios de comunicación que permitan 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. Los uno o más medios de comunicación pueden incluir medios de comunicación inalámbricos y/o alámbricos, tales como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. Los uno o más medios de comunicación pueden formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global (por ejemplo, Internet). Los uno o más medios de comunicación pueden incluir routers, conmutadores, estaciones base u otros equipos que faciliten la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

**[0027]** En otro ejemplo, el canal 16 puede incluir un medio de almacenamiento que almacena los datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder al medio de almacenamiento, por ejemplo, mediante el acceso a disco o el acceso a tarjeta. El medio de almacenamiento puede incluir varios medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, DVD, CD-ROM, memoria flash u otros medios adecuados de almacenamiento digital para almacenar datos de vídeo codificados.

**[0028]** En un ejemplo adicional, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento en memoria intermedia 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 en memoria intermedia (por ejemplo, mediante 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. Entre los ejemplos de servidores de archivos se incluyen servidores de red (por ejemplo, para una página web), servidores del protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados a red (NAS) y unidades de disco local.

**[0029]** El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de una conexión de datos estándar, como una conexión a Internet. Entre los ejemplos de tipos de conexiones de datos pueden incluirse canales inalámbricos (por ejemplo, conexiones WiFi), conexiones cableadas (por ejemplo, DSL, módem de cable, etc.) o combinaciones de ambos que sean adecuadas para acceder a datos de vídeo codificados almacenados en un servidor

de ficheros. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

**[0030]** Las técnicas de esta divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo en soporte de varias aplicaciones multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, mediante Internet, codificación de datos de vídeo para su almacenamiento en un medio de almacenamiento de datos, descodificación de datos de vídeo almacenados en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo puede configurarse para soportar la transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

**[0031]** La FIG. 1 es meramente un ejemplo y las técnicas de esta divulgación pueden aplicarse a ajustes de codificación de vídeo (por ejemplo, codificación de vídeo o descodificación de vídeo) que no incluyen necesariamente ninguna comunicación de datos entre los dispositivos de codificación y descodificación. En otros ejemplos, los datos se recuperan de una memoria local, se transmiten a través de una red o similar. Un dispositivo de codificación de vídeo puede codificar y almacenar datos (por ejemplo, datos de vídeo) en la memoria, y/o un dispositivo de descodificación de vídeo puede recuperar y descodificar datos (por ejemplo datos de vídeo) desde la memoria. En muchos ejemplos, la codificación y la descodificación se realiza mediante dispositivos que no se comunican entre sí, sino que simplemente codifican datos en la memoria y/o recuperan y descodifican datos desde la memoria.

**[0032]** 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.

**[0033]** El codificador de vídeo 20 puede codificar datos de vídeo 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 mediante 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.

**[0034]** En el ejemplo de la FIG. 1, el dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos ejemplos, la interfaz de entrada 28 incluye un receptor y/o un módem. La interfaz de entrada 28 puede recibir los datos de vídeo codificados por el canal 16. El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En general, el dispositivo de visualización 32 muestra los datos de vídeo descodificados. El dispositivo de visualización 32 puede comprender varios dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

**[0035]** El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse como cualquiera entre una variedad de circuitos adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), conjuntos de puertas programables sobre el terreno (FPGA), lógica discreta, hardware o cualquier combinación de los mismos. Si las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado de almacenamiento no transitorio legible por ordenador, y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cualquiera de los anteriores (incluyendo hardware, software, una combinación de hardware y software, etc.) puede considerarse como uno o más procesadores. Cada uno entre el codificador de vídeo 20 y el descodificador de vídeo 30 se puede incluir en uno o más codificadores o descodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

**[0036]** Esta divulgación puede referirse en general al codificador de vídeo 20 que "señala" cierta información a otro dispositivo, tal como el descodificador de vídeo 30. El término "señalización" puede referirse en general a la comunicación de elementos sintácticos y/o a otros datos usados para descodificar los datos de vídeo comprimidos. Dicha comunicación puede producirse en tiempo real o casi real. De forma alternativa, dicha comunicación puede producirse durante un tramo de tiempo, tal como podría 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 entonces pueden ser recuperado por un dispositivo de descodificación en cualquier momento tras haber sido almacenado en este medio.

**[0037]** 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 conocido como ISO/IEC MPEG-4 AVC), incluida su extensión de codificación de vídeo escalable (SVC), la extensión de codificación de vídeo

de múltiples visualizaciones (MVC) y la extensión 3DV basada en MVC. En algunos casos, cualquier flujo de bits que se ajuste a 3DV basada en MVC siempre contiene un sub-flujo 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 basada en AVC. En otros ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con la Recomendación ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual e ITU-T H.264, ISO/IEC Visual.

**[0038]** En otros ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con la norma Codificación de Vídeo de Alta Eficiencia (HEVC) desarrollada 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 norma HEVC, denominado "HEVC Working Draft 10 [Borrador de trabajo de HEVC 10]" o "HEVC WD10" se describe en Bross et al., "Editors' proposed corrections to HEVC version 1 [Correcciones propuestas por los editores para la HEVC versión 1]", Equipo de colaboración conjunta en codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 13.ª reunión, Incheon, Corea, abril de 2013. Al menos desde el 30 de mayo de 2014, el Borrador de trabajo de HEVC 10 está disponible en [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/13\\_Incheon/wg11/JCTVC-M0432-v3.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0432-v3.zip).

**[0039]** Además, actualmente se están dedicando esfuerzos continuos a la elaboración de codificación de vídeo escalable, codificación de múltiples visualizaciones y extensiones 3DV para HEVC. La extensión 3DV de HEVC puede denominarse 3DV basada en HEVC, HEVC-3DV o 3D-HEVC. La extensión de múltiples visualizaciones de HEVC, a saber, MV-HEVC, también está siendo desarrollada por el Equipo de Colaboración Conjunta sobre Desarrollo de Extensión de Codificación de Vídeo 3D (JCT-3V). Un reciente borrador de trabajo (WD) de MV-HEVC, denominado en adelante MV-HEVC WD4, se describe en Tech et al., "MV-HEVC Draft Text 4 [Texto de borrador de MV-HEVC 4]", Equipo de colaboración conjunto en codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 4.ª reunión, Incheon, Corea, abril de 2013, documento n.º JCT3V-D1004\_d0 (en adelante, "JCT3V-D1004\_d0"). Al menos desde el 30 de mayo de 2014, JCT3V-D1004\_d0 está disponible en [http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/documents/4\\_Incheon/wg11/JCT3V-D1004-v1.zip](http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/4_Incheon/wg11/JCT3V-D1004-v1.zip).

**[0040]** La extensión escalable para HEVC, llamada SHVC, también está siendo desarrollada por el JCT-VC. Un Borrador de Trabajo (WD) reciente de SHVC se denomina SHVC WD2 en adelante y se describe en Chen et al, "SHVC Working Draft 2 [Borrador de Trabajo SHVC 2]", Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 13.ª reunión, Incheon, Corea, 18-26 de abril de 2013, documento n.º JCTVC-M1008\_v1 (en adelante, "JCTVC-M1008\_v1"). Al menos desde el 30 de mayo de 2014, JCTVC-M1008\_v1 está disponible en [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/13\\_Incheon/wg11/JCTVC-M1008-v1.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M1008-v1.zip).

**[0041]** En HEVC y 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, indicadas con  $S_L$ ,  $S_{Cb}$  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 croma Cb.  $S_{Cr}$  es una matriz bidimensional de muestras de croma 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 solo incluir una matriz de muestras de luminancia.

**[0042]** Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades de árbol de codificación (CTU). Cada una de las CTU puede comprender (por ejemplo, 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. En una imagen monocromática o en una imagen que tenga tres planos de color separados, una CTU puede comprender un único bloque de árbol de codificación y estructuras sintácticas usadas para codificar las muestras del bloque de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque de muestras de tamaño  $N \times N$ . Una CTU también puede denominarse "bloque arbolado" o "máxima unidad de codificación" (LCU). Las CTU de HEVC pueden ser análogas en gran medida a los macrobloques de otras normas, tales como H.264/AVC. Sin embargo, una CTU no está limitada necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). Un fragmento puede incluir un número entero de CTU ordenadas consecutivamente en un orden de escaneado (por ejemplo, el orden de escaneado de raster).

**[0043]** Esta divulgación puede usar el término "unidad de vídeo", "bloque de vídeo" o "bloque" para referirse a uno o más bloques de muestras y estructuras sintácticas usadas para codificar muestras del uno o más bloques de muestras. Los tipos de ejemplo de unidades de vídeo pueden incluir CTU, CU, PU, unidades de transformada (TU), macrobloques, particiones de macrobloque, etc. En algunos contextos, el análisis de CTU, CU, PU, etc. puede intercambiarse con el análisis de macrobloques o de particiones de macrobloques.

**[0044]** Para generar una CTU codificada, el codificador de vídeo 20 puede llevar a cabo de forma recursiva una partición de árbol cuádruple 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 puede ser un bloque de muestras de tamaño NxN. Una CU puede comprender (por ejemplo, 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. En imágenes monocromáticas o imágenes que tengan tres planos de color independientes, una CU puede comprender un único bloque de codificación y estructuras sintácticas usadas para codificar las muestras del bloque de codificación.

**[0045]** El codificador de vídeo 20 puede dividir un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción puede ser un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las que se aplique la misma predicción. Una unidad de predicción (PU) de una CU puede comprender (por ejemplo, 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. En una imagen monocromática o una imagen que tenga tres planos de color independientes, una PU puede comprender un único bloque de predicción y estructuras sintácticas usadas para predecir el bloque predictivo.

**[0046]** El codificador de vídeo 20 puede usar predicción intra o predicción inter para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 usa la predicción intra 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 con la PU (es decir, la imagen que contiene el bloque de predicción de la PU).

**[0047]** Si el codificador de vídeo 20 utiliza predicción inter 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 una o más imágenes distintas a la imagen asociada con la PU. El codificador de vídeo 20 puede usar unipredicción o bipredicción para generar los bloques predictivos de una PU. Cuando el codificador de vídeo 20 usa la predicción unidimensional 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.

**[0048]** Para realizar la unipredicción o la bipredicción, el codificador de vídeo 20 puede generar una primera lista de imágenes de referencia (RefPicListO) y una segunda lista de imágenes de referencia (RefPicList1) para un fragmento actual. Cada una de las listas de imágenes de referencia puede incluir una o más imágenes de referencia. Cuando se utiliza unipredicción, el codificador de vídeo 20 puede buscar las imágenes de referencia en RefPicListO y RefPicList1 para determinar una ubicación de referencia dentro de una imagen de referencia. Además, cuando se usa la unipredicción, el codificador de vídeo 20 puede generar, basándose al menos en parte en las muestras correspondientes a la ubicación de referencia, los bloques predictivos (es decir, bloques de muestras predictivos) para la PU. Además, cuando se usa la unipredicción, el codificador de vídeo 20 puede generar un único vector de movimiento que indique un desplazamiento espacial entre un bloque de predicción de la PU y la ubicación de referencia. Para indicar el desplazamiento espacial entre un bloque de predicción de la PU y la ubicación de referencia, un vector de movimiento puede incluir un componente horizontal que especifique un desplazamiento horizontal entre el bloque de predicción de la PU y la ubicación de referencia y puede incluir un componente vertical que especifique un desplazamiento vertical entre el bloque predictivo de la PU y la ubicación de referencia.

**[0049]** Cuando se usa la bipredicción para codificar una PU, el codificador de vídeo 20 puede determinar una primera ubicación de referencia en una imagen de referencia en RefPicListO y una segunda ubicación de referencia en una imagen de referencia en RefPicList1. El codificador de vídeo 20 puede generar entonces, basándose al menos en parte en muestras correspondientes a las primera y segunda ubicaciones de referencia, los bloques predictivos para la PU. Además, cuando se usa la bipredicción para codificar la PU, el codificador de vídeo 20 puede generar un primer vector de movimiento que indique un desplazamiento espacial entre un bloque de predicción de la PU y la primera ubicación de referencia y un segundo movimiento que indique un desplazamiento espacial entre el bloque de predicción de la PU y la segunda ubicación de referencia.

**[0050]** Después de que el codificador de vídeo 20 genera bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para una o más PU de una CU, el codificador de vídeo 20 puede generar uno o más bloques residuales para la CU. Por ejemplo, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra en el bloque residual de luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques predictivos de luma de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual de Cb para la CU. Cada muestra en el bloque residual de Cb de la CU puede indicar una diferencia entre una muestra de Cb en uno de los bloques predictivos de Cb de la CU y una muestra correspondiente en el bloque de codificación de Cb original de la CU. El codificador de vídeo 20 también puede generar un bloque residual de Cr para la CU. Cada muestra en el bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques predictivos de Cr de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.

**[0051]** Además, el codificador de vídeo 20 puede usar la partición en árbol cuádruple para descomponer los bloques residuales (por ejemplo, los bloques residuales de luma, Cb y Cr) de una CU en uno o más bloques de transformada

(por ejemplo, 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 comprender (por ejemplo, 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. De este modo, cada TU de una CU puede tener (es decir, estar asociada con) 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 de (es decir, asociado con) 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. En una imagen monocromática o una imagen que tenga tres planos de color independientes, una TU puede comprender un único bloque de transformada de muestras y estructuras sintácticas usadas para transformar las muestras de bloque de transformada.

**[0052]** El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de una TU para generar un bloque de coeficientes 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. Por ejemplo, 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. 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.

**[0053]** Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes luma, un bloque de coeficientes Cb o un bloque de coeficientes 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 una compresión adicional. Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos sintácticos que indican los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar la Codificación Aritmética Binaria Adaptativa al Contexto (CABAC) en los elementos sintácticos que indiquen los coeficientes de transformada cuantificados. El codificador de vídeo 20 puede emitir los elementos sintácticos codificados por entropía en un flujo de bits.

**[0054]** El codificador de vídeo 20 puede emitir un flujo de bits que incluye los elementos sintácticos codificados por entropía. El flujo de bits puede incluir una secuencia de bits que forme una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades de capas de abstracción de red (NAL). Cada una de las unidades de NAL puede incluir una cabecera de unidad de NAL y puede encapsular una carga útil de secuencias de bytes sin procesar (RBSP). La cabecera de la unidad NAL puede incluir un elemento sintáctico que indique 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 contenga un número entero de bytes que se encapsule dentro de una unidad de NAL. En algunos casos, una RBSP incluye cero bits.

**[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 de NAL puede encapsular una RBSP para un fragmento codificado, un tercer tipo de unidad de NAL puede encapsular una RBSP para información de mejora complementaria (SEI), y así sucesivamente. Las unidades de NAL que encapsulan las RBSP para datos de codificación de vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes de SEI) pueden denominarse unidades de NAL de la capa de codificación de vídeo (VCL). En al menos algunos ejemplos, la SEI contiene información que no es necesaria para descodificar las muestras de imágenes codificadas de las unidades VCL NAL. Además, en al menos algunos ejemplos, un SEI RBSP puede contener uno o más mensajes SEI. Un mensaje SEI puede ser un mensaje que contiene SEI.

**[0056]** HEVC y otras normas de codificación de vídeo proporcionan varios tipos de conjuntos de parámetros. Por ejemplo, un conjunto de parámetros de vídeo (VPS) es una estructura sintáctica que comprende elementos sintácticos que se aplican a cero o más secuencias de vídeo codificadas completas (CVS). Un conjunto de parámetros de secuencia (SPS) puede contener información que se aplica a todos los fragmentos de una CVS. Un SPS puede incluir un elemento sintáctico que identifica un VPS que está activo cuando el SPS está activo. Por lo tanto, los elementos sintácticos de un VPS pueden ser más en general aplicables que los elementos sintácticos de un SPS. Un PPS es una estructura sintáctica que comprende elementos sintácticos que se aplican a cero o más imágenes codificadas. Un PPS puede incluir un elemento sintáctico que identifica un SPS que está activo cuando el PPS está activo. Una cabecera de fragmento de un fragmento puede incluir un elemento sintáctico que indica un PPS que está activo cuando el fragmento está siendo codificado.

**[0057]** El descodificador de vídeo 30 puede recibir el flujo de bits generado por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede analizar el flujo de bits para obtener (por ejemplo, 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 obtenidos (por ejemplo, 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 en movimiento de las PU para determinar bloques predictivos para las PU de una CU actual. Además, el descodificador de vídeo 30 puede cuantificar inversamente bloques de coeficientes de transformada, asociados a las TU de la CU actual. El descodificador de vídeo 5 30 puede realizar transformadas inversas en los bloques de coeficientes de transformada para reconstruir los bloques de transformada de (es decir, asociados con) las TU de la CU actual. El descodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para las PU de la CU actual a las muestras correspondientes de los bloques de transformada de las TU de la CU actual. Mediante la reconstrucción de los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede 10 reconstruir la imagen.

**[0058]** Cuando un codificador de vídeo (por ejemplo, el codificador de vídeo 20 o el descodificador de vídeo 30) comienza el procesamiento de una imagen actual, el codificador de vídeo puede determinar cinco subconjuntos del conjunto de imágenes de referencia (RPS) para la imagen actual: RefPicSetStCurrBefore, RefPicSetStCurrAfter, 15 RefPicSetFoll, RefPicSetLtCurr, y RefPicSetLtFoll. La unión de estos cinco subconjuntos de RPS para la imagen actual se puede denominar en el presente documento el RPS para la imagen actual. RefPicSetStCurrBefore puede incluir imágenes de referencia a corto plazo que aparecen antes de la imagen actual en el orden de salida y pueden ser usadas como referencia por la imagen actual. RefPicStCurrAfter puede incluir imágenes de referencia a corto plazo que se producen después de la imagen actual en el orden de salida y pueden ser usadas como referencia por la 20 imagen actual. RefPicSetStFoll puede incluir imágenes de referencia a corto plazo que no son usadas como referencia por la imagen actual. RefPicSetLtCurr puede incluir imágenes de referencia a largo plazo que pueden ser usadas como referencia por la imagen actual. RefPicSetLtFoll puede incluir imágenes de referencia a largo plazo que no son usadas como referencia por la imagen actual.

**[0059]** HEVC y otras normas de codificación de vídeo proporcionan varios tipos de fragmentos diferentes. Estos tipos de fragmento incluyen fragmentos I, fragmentos P y fragmentos B. En los fragmentos I, solo se permite la predicción intra. En los fragmentos P, se permite la predicción intra y la predicción inter unidireccional. En los fragmentos B, se permite la predicción intra, la predicción inter unidireccional y la predicción inter bidireccional. Cuando el codificador de vídeo comienza a codificar un fragmento de una imagen, el codificador de vídeo puede inicializar una primera lista 30 de imágenes de referencia (es decir, Lista 0). Además, si el fragmento actual es un fragmento B, el codificador de vídeo puede inicializar una segunda lista de imágenes de referencia (es decir, la Lista 1). Esta divulgación puede referirse a la Lista 0 como "RefPicListO" y puede referirse a la Lista 1 como "RefPicList1".

**[0060]** Para generar RefPicListO, un codificador de vídeo (por ejemplo, un codificador de vídeo o un descodificador de vídeo) puede generar una versión inicial de RefPicListO. En la versión inicial de RefPicListO, las imágenes de referencia en RefPicSetStCurrBefore se enumeran primero, seguidas de las imágenes de referencia en RefPicSetStCurrAfter, seguidas de las imágenes de referencia en RefPicSetLtCurr, si están disponibles. En la versión inicial de RefPicListO, las imágenes a corto plazo con un orden de salida anterior (es decir, las imágenes de referencia en RefPicSetStCurrBefore) se insertan primero en RefPicListO en orden ascendente de la distancia de recuento de 35 orden de imágenes (POC) hasta la imagen actual, y a continuación las imágenes a corto plazo con el orden de salida posterior (es decir, las imágenes de referencia en RefPicSetStCurrAfter) se insertan en RefPicListO en orden ascendente de la distancia POC hasta la imagen actual, y finalmente las imágenes a largo plazo (es decir, las imágenes de referencia en RefPicSetLtCurr) se insertan al final. Una distancia POC es la diferencia entre los valores POC para las imágenes. Un POC es una variable que está asociada con cada imagen que indica la posición de la imagen asociada en orden de salida en relación con las posiciones de orden de salida de las otras imágenes en la misma secuencia de vídeo codificada. 45

**[0061]** De manera similar, para generar RefPicList1, el codificador de vídeo puede generar una versión inicial de RefPicList1. En la versión inicial de RefPicList1, las imágenes de referencia en RefPicSetStCurrAfter se enumeran primero, seguidas por las imágenes de referencia en RefPicSetStCurrBefore, seguidas por las imágenes de referencia en RefPicSetLtCurr. En la versión inicial de RefPicList1, las imágenes a corto plazo con el orden de salida posterior (es decir, las imágenes de referencia en RefPicSetStCurrAfter) se insertan primero en RefPicList1 en orden ascendente de la distancia POC hasta la imagen actual, y a continuación las imágenes a corto plazo con el orden de salida anterior (es decir, las imágenes de referencia en RefPicSetStCurrBefore) se insertan en RefPicList1 en orden ascendente de la distancia POC hasta la imagen actual, y finalmente las imágenes a largo plazo (es decir, las imágenes de referencia en RefPicSetLtCurr) se insertan al final. De esta manera, la inicialización de la lista de imágenes de referencia crea la Lista 0 y la Lista 1 predeterminadas (si el sector es un sector B) basándose en tres subconjuntos RPS: RefPicSetStCurrBefore, RefPicSetStCurrAfter y RefPicSetLtCurr. 55

**[0062]** Después de que el codificador de vídeo genere la versión inicial de una lista de imágenes de referencia (por ejemplo, RefPicListO o RefPicList1), el codificador de vídeo puede modificar el orden de las imágenes de referencia en la lista de imágenes de referencia. Es decir, el codificador de vídeo puede realizar un proceso de modificación de la lista de imágenes de referencia (RPLM) para modificar el orden de las imágenes de referencia en la lista de imágenes de referencia. 60

65

- 5 **[0063]** En la codificación de múltiples visualizaciones, puede haber varias visualizaciones de la misma escena desde diferentes puntos de visualización. El término "unidad de acceso" puede usarse para referirse 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. Un "componente de visualización" puede ser una representación codificada de una visualización en una única unidad de acceso. En esta divulgación, una "visualización" puede referirse a una secuencia de componentes de visualizaciones asociadas al mismo identificador de visualización.
- 10 **[0064]** La codificación de múltiples visualizaciones soporta predicción entre visualizaciones. La predicción entre visualizaciones es similar a la predicción inter utilizada en la HEVC y puede usar los mismos elementos sintácticos. Sin embargo, cuando un codificador de vídeo realiza la predicción entre visualizaciones 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 visualización diferente. Por lo contrario, la predicción inter convencional solo usa imágenes en diferentes unidades de acceso como imágenes de referencia.
- 15 **[0065]** En la codificación de múltiples visualizaciones, una visualización puede denominarse "visualización básica" si un descodificador de vídeo (por ejemplo, descodificador de vídeo 30) puede descodificar imágenes de la visualización sin referencia a imágenes en ninguna otra visualización. Cuando se codifica una imagen en una de las visualizaciones no de base, un codificador de vídeo (tal como el codificador de vídeo 20 o el descodificador de vídeo 30) puede agregar una imagen a una lista de imágenes de referencia si la imagen está en una visualización diferente pero dentro de la misma instancia de tiempo (es decir, unidad de acceso) que la imagen que el codificador de vídeo está codificando actualmente. Al igual que otras imágenes de referencia de predicción inter, el codificador de vídeo puede insertar una imagen de referencia de predicción entre visualizaciones en cualquier posición de una lista de imágenes de referencia.
- 20 **[0066]** Las unidades NAL puede incluir cabeceras y cargas útiles. Las cabeceras de las unidades NAL pueden incluir elementos sintácticos `nuh_reserved_zero_6bits`. En algunos ejemplos, el elemento sintáctico `nuh_reserved_zero_6bits` de una unidad NAL es igual a 0 si la unidad NAL se relaciona con una capa base en codificación de múltiples visualizaciones, codificación 3DV o codificación de vídeo escalable. 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 se relaciona con una capa base en codificación de múltiples visualizaciones, 3DV o codificación de vídeo escalable, el elemento sintáctico `nuh_reserved_zero_6bits` puede tener un valor distinto de cero. Específicamente, si una unidad NAL no se relaciona con una capa base en codificación de múltiples visualizaciones, 3DV o codificación de vídeo escalable, el elemento sintáctico `nuh_reserved_zero_6bits` de la unidad NAL especifica un identificador de capa que identifica una capa de la unidad NAL.
- 25 **[0067]** 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 de NAL que encapsulan datos de ciertas imágenes de una capa pueden eliminarse del flujo de bits sin afectar la capacidad de descodificación de otras imágenes en la capa. Por ejemplo, las imágenes con valores de recuento de orden de imágenes (POC) par pueden ser descodificables sin referencia a imágenes con valores POC impares. La eliminación de unidades de NAL que encapsulan datos de tales imágenes puede reducir la velocidad de tramas 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.
- 30 **[0068]** Las unidades NAL pueden incluir elementos sintácticos `temporal_id`. El elemento sintáctico `id_temporal` de una unidad de NAL especifica un identificador temporal de la unidad de NAL. Si el identificador temporal de una primera unidad de NAL es menor que el identificador temporal de una segunda unidad de NAL en la misma capa, los datos encapsulados por la primera unidad de NAL pueden descodificarse sin referencia a los datos encapsulados por la segunda unidad de NAL.
- 35 **[0069]** Cada uno de los puntos de funcionamiento de un flujo de bits está asociado con un conjunto de identificadores de capa (es decir, un conjunto de valores `nuh_layer_id`) y un identificador temporal. El conjunto de identificadores de capa se puede indicar como `ConjuntoldCapaOp` y el identificador temporal se puede indicar como `IDTemporal`. Si el identificador de capa de una unidad de NAL se encuentra en el conjunto de identificadores de capa de un punto de funcionamiento y el identificador temporal de la unidad de NAL es menor o igual que el identificador temporal del punto de funcionamiento, la unidad de NAL pertenece al punto de funcionamiento. Un punto de funcionamiento es un subconjunto de flujo de bits que está asociado con un `OpLayerIdSet` y un identificador temporal. El punto de funcionamiento puede incluir cada unidad de NAL que esté asociada al punto de funcionamiento. El punto de funcionamiento no incluye las unidades VCL NAL que no están asociadas con el punto de funcionamiento.
- 40 **[0070]** Al menos algunas normas de codificación de vídeo especifican modelos de memoria intermedia de vídeo. En H.264/AVC y HEVC, un modelo de almacenamiento en memoria intermedia se denomina "descodificador de referencia hipotético" o "HRD". En el borrador de trabajo de HEVC 10 se describe el HRD en el anexo C. El HRD describe cómo se almacenan en memoria intermedia los datos para la descodificación y cómo se almacenan en memoria intermedia los datos descodificados para la emisión. Por ejemplo, el HRD describe el funcionamiento de una memoria intermedia de imágenes codificadas ("CPB"), una memoria intermedia de imágenes descodificadas ("DBP") y un proceso de descodificación de vídeo. La CPB es una memoria intermedia de primero en entrar, primero en salir, que contiene
- 45
- 50
- 55
- 60
- 65

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 sincronización, los tamaños de la memoria intermedia y las velocidades de transferencia de bits. Además, el HRD puede imponer restricciones indirectamente a varias características y estadísticas de flujo de bits.

**[0071]** En H.264/AVC y HEVC, la conformidad del flujo de bits y la conformidad del descodificador se especifican como partes de la especificación del HRD. En otras palabras, el modelo de HRD especifica pruebas para determinar si un flujo de bits cumple con la norma y pruebas para determinar si un descodificador cumple con 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 típicamente no necesitan el HRD.

**[0072]** H.264/AVC y HEVC especifican dos tipos de conformidad de flujo de bits o de HRD; 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 VCL NAL y la unidad NAL de datos de relleno para todas las unidades de acceso en el flujo de bits. En al menos algunos ejemplos, un flujo de unidades NAL es una secuencia de unidades NAL. Un flujo de bits de Tipo II es un flujo de unidad NAL que contiene, además de las unidades VCL NAL y unidades NAL de datos de relleno para todas las unidades de acceso en el flujo de bits, al menos uno de los siguientes: unidades NAL no VCL adicionales distintas de unidades NAL de datos de relleno; y todos los elementos sintácticos `leading_zero_8bits`, `zero_byte`, `start_coded_prefix_one_3bytes` y `trailing_zero_8bits` que forman una secuencia de bytes desde la secuencia de la unidad NAL.

**[0073]** Cuando un dispositivo realiza una prueba de conformidad de flujo de bits que determina si un flujo de bits cumple con 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 de HRD aplicables al punto de funcionamiento seleccionado. El dispositivo puede usar el conjunto de parámetros de HRD aplicable al punto de funcionamiento seleccionado para configurar el comportamiento del HRD. Más particularmente, el dispositivo puede usar el conjunto aplicable de parámetros de HRD para configurar los comportamientos de componentes particulares del HRD, tales como un programador de flujo hipotético (HSS), la CPB, un proceso de descodificación, la DPB, etc. Posteriormente, el HSS puede inyectar datos de vídeo codificados del flujo de bits en la CPB del HRD de acuerdo con un programa particular. Además, el dispositivo puede invocar un proceso de descodificación que descodifica los datos de vídeo codificados en la CPB. El proceso de descodificación puede enviar imágenes descodificadas a la DPB. A medida que el dispositivo mueve datos a través del HRD, el dispositivo puede determinar si sigue cumpliéndose un conjunto particular de restricciones. Por ejemplo, el dispositivo puede determinar si se produce una condición de sobreflujo o sub-flujo en la CPB o DPB mientras el HRD está descodificando el punto de funcionamiento del punto de funcionamiento seleccionado. El dispositivo puede seleccionar y procesar cada punto de funcionamiento del flujo de bits de esta manera. Si ningún punto de funcionamiento del flujo de bits hace que se violen las restricciones, el dispositivo puede determinar que el flujo de bits cumple con la norma de codificación de vídeo.

**[0074]** H.264/AVC y HEVC especifican dos tipos de conformidad de descodificador; esto es, la conformidad de descodificador de sincronización de salida y la conformidad de descodificador del orden de salida. Un descodificador que alega conformidad con un perfil, grado y nivel específico, es capaz de descodificar con éxito todos los flujos de bits que cumplen con los requisitos de conformidad de flujo de bits de una norma de codificación de vídeo, como HEVC. En algunos ejemplos, un "perfil" puede referirse a un subconjunto de la sintaxis del flujo de bits. Se pueden especificar "fases" y "niveles" dentro de cada perfil. Un nivel de un grado puede ser un conjunto de restricciones especificadas impuestas en los valores de los elementos sintácticos en el flujo de bits. Estas restricciones pueden ser simples limitaciones de valores. De forma alternativa, las restricciones pueden tomar la forma de restricciones sobre combinaciones aritméticas de valores (por ejemplo, el ancho de imagen multiplicado por la altura de imagen multiplicada por el número de imágenes descodificadas por segundo). Típicamente, un nivel especificado para una fase inferior es más limitado que un nivel especificado para una fase superior.

**[0075]** Cuando un dispositivo realiza una prueba de conformidad de descodificador para determinar si un descodificador bajo prueba (DUT) cumple con una norma de codificación de vídeo, el dispositivo puede proporcionar, tanto al HRD y el DUT, un flujo de bits que cumple con 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 emitidas por el DUT coincide con el orden de las imágenes descodificadas emitidas por el HRD. Además, el dispositivo puede determinar que el DUT cumple con la norma de codificación de vídeo si el tiempo con el cual el DUT envía imágenes descodificadas coincide con el tiempo con el cual el HRD emite las imágenes descodificadas.

**[0076]** En los modelos H.264/AVC y HEVC HRD, la descodificación o eliminación de CPB pueden basarse en unidad de acceso. Es decir, se supone que el HRD descodifica unidades de acceso completas de una vez y elimina unidades de acceso completas de la CPB. Además, en los modelos H.264/AVC y HEVC HRD, se supone que la descodificación de imágenes es instantánea. El codificador de vídeo 20 puede indicar, en los mensajes SEI de sincronización de imágenes, los tiempos de descodificación para comenzar la descodificación de las unidades de acceso. En aplicaciones prácticas, si un descodificador de vídeo en conformidad sigue estrictamente los tiempos de

descodificación indicados para iniciar la descodificación de unidades de acceso, el primer momento posible para emitir una imagen descodificada específica es igual al momento de descodificación de esa imagen particular, más el tiempo necesario para descodificar esa imagen particular. Sin embargo, en el mundo real, el tiempo necesario para descodificar una imagen no puede ser igual a cero.

5  
 [0077] Los parámetros HRD pueden controlar diversos aspectos del HRD. En otras palabras, el HRD puede confiar en los parámetros de HRD. Los parámetros de HRD pueden incluir un retardo inicial de eliminación de la CPB, un tamaño de CPB, una velocidad de transferencia de bits, un retardo inicial de salida de la DPB y un tamaño de DPB. El codificador de vídeo 20 puede indicar 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 indicar parámetros de HRD en mensajes SEI de período de almacenamiento en memoria intermedia o mensajes SEI de sincronización de imágenes.

15  
 [0078] Como se explicó anteriormente, un punto de funcionamiento de un flujo de bits está asociado con un conjunto de identificadores de capa (es decir, un conjunto de valores `nuh_reserved_zero_6bits`) y un identificador temporal. Un punto de funcionamiento puede incluir cada unidad NAL que está asociada con el punto de funcionamiento. Un punto de funcionamiento puede tener una velocidad de tramas y/o velocidad de transferencia de bits diferente a la de un flujo de bits original. Esto se debe a que el punto de funcionamiento puede no incluir algunas imágenes y/o algunos de los datos del flujo de bits original. Por lo tanto, si el descodificador de vídeo 30 fuera a eliminar datos de la CPB, y/o la DPB a una velocidad particular cuando procesa el flujo de bits original y si el descodificador de vídeo 30 fuera a eliminar datos de la CPB y/o la DPB a la misma velocidad cuando procesa un punto de funcionamiento, el descodificador de vídeo 30 puede eliminar demasiados o muy pocos datos de la CPB y/o la DPB. En consecuencia, el codificador de vídeo 20 puede indicar diferentes conjuntos de parámetros de HRD para diferentes puntos de funcionamiento. Por ejemplo, el codificador de vídeo 20 puede incluir, en un VPS, múltiples estructuras sintácticas `hrd_parameters()` que incluyen parámetros HRD para diferentes puntos de funcionamiento.

30  
 [0079] En el Borrador de Trabajo de HEVC 10 (también denominado WD10), el conjunto de parámetros HRD incluye opcionalmente un conjunto de información que es común para todas las subcapas temporales. 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 puede ser una capa temporal escalable de un flujo de bits temporal escalable que consta de unidades VCL NAL 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 representaciones de subcapas temporales individuales. Por ejemplo, la estructura sintáctica `hrd_parameters()` puede incluir opcionalmente un conjunto de información que es común para todas las representaciones de subcapa 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 indicar el conjunto de información común en múltiples conjuntos de parámetros HRD. Por el contrario, en el Borrador de Trabajo de HEVC 10, 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 con un primer índice de punto de funcionamiento. Por ejemplo, el Borrador de Trabajo de HEVC 10 (WD10) soporta 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 con un primer índice de punto de funcionamiento.

[0080] La Tabla 1, a continuación, es una estructura sintáctica de ejemplo para una estructura sintáctica `hrd_parameters()` en HEVC.

50

**TABLA 1 - Parámetros HRD**

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

5 **[0081]** En el ejemplo de la Tabla 1 anterior y en otras tablas sintácticas de esta divulgación, los elementos sintácticos con descriptor de tipo ue(v) pueden ser números enteros de longitud variable sin signo codificados utilizando codificación Golomb exponencial (Exp-golomb) de orden 0 con el bit izquierdo primero. En el ejemplo de la Tabla 1 y las siguientes tablas, los elementos sintácticos que tienen descriptores de la forma u(n), donde n es un entero no negativo, son valores sin signo de longitud n.

10 **[0082]** En el ejemplo de sintaxis de la Tabla 1, el elemento sintáctico nal\_hrd\_parameters\_present\_flag igual a 1 especifica que los parámetros HRD NAL (perteneciente a la conformidad de flujo de bits de Tipo II) están presentes en la estructura sintáctica hrd\_parameters(). nal\_hrd\_parameters\_present\_flag igual a 0 especifica que los parámetros NAL HRD no están presentes en la estructura sintáctica hrd\_parameters(). Por lo tanto, los parámetros NAL HRD son parámetros HRD que pertenecen a la conformidad de flujo de bits del Tipo II. El elemento sintáctico nal\_hrd\_parameters\_present\_flag igual a 1 especifica que los parámetros VCL HRD (pertenecientes a toda la conformidad del flujo de bits) están presentes en la estructura sintáctica hrd\_parameters(). vcl\_hrd\_parameters\_present\_flag igual a 0 especifica que los parámetros de VCL HRD no están presentes en la estructura sintáctica hrd\_parameters(). Por lo tanto, los parámetros VCL HRD son parámetros HRD que pertenecen a toda la conformidad del flujo de bits.

20 **[0083]** En el ejemplo de sintaxis de la Tabla 1, los elementos sintácticos en el bloque "if (commonInfPresentFlag) {... }" "bloque 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\_minus 1, cpb\_removal\_delay\_length\_minus 1, y dpb\_output\_delay\_length\_minus1.

5 **[0084]** 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 representación de 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 representación de subcapa particular del flujo de bits.

15 **[0085]** 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 una 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 de HEVC 10. El elemento sintáctico cpb\_cnt\_minus1[i] puede especificar el número de especificaciones de CPB alternativa en el flujo de bits de la secuencia de vídeo codificada cuando HighestTid es igual a i, en el que una especificación de CPB alternativa se refiere a una operación de CPB particular con un conjunto particular de parámetros de CPB.

25 **[0086]** 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 fines diversos. Por ejemplo, el descodificador de vídeo 30 u otro dispositivo puede usar los metadatos en los mensajes SEI para la sincronización de salida de la imagen, visualización de la imagen, detección de pérdida y ocultación de errores.

30 **[0087]** El codificador de vídeo 20 puede incluir una o más unidades SEI NAL en una unidad de acceso. En otras palabras, cualquier número de unidades SEI NAL puede estar asociada con una unidad de acceso. Además, cada unidad SEI NAL 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 el 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 Tabla 2, a continuación, enumera los mensajes SEI especificados en HEVC y describe brevemente sus propósitos.

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

| Mensaje SEI  | Finalidad   |
|--|---|
| Período de almacenamiento temporal                             | Retardos iniciales para la operación del descodificador de referencia hipotético (HRD)  |
| Sincronización de imágenes                                     | Tiempo de salida de la imagen y tiempo de eliminación de imagen/sub-imagen para el funcionamiento del HRD   |
| Rectángulo de escaneado panorámico                             | Exhibición 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 y actualización de descodificación gradual.  |
| información de escena  | Información sobre cambios de escena y transiciones  |
| Instantánea de la imagen                                       | Indicación para etiquetar la imagen descodificada asociada como una instantánea de imagen fija del contenido de vídeo                             |
| Segmento de refinamiento progresivo                            | Indica que ciertas imágenes consecutivas representan un refinamiento progresivo de la calidad de una imagen, en lugar de una escena en movimiento |

| Mensaje SEI   | Finalidad  |
|---|--|
| Características de textura de película                    | Permite a los decodificadores sintetizar la textura de la película                                       |
| Sugerencia de posfiltro                                   | Proporciona coeficientes de posfiltro sugeridos o información de correlación para el diseño de posfiltro |
| Información de correlación de tonos                       | Re-correlación con otro espacio de colores distinto al usado o adoptado en la codificación               |
| Disposición de empaquetamiento de tramas                  | Empaquetamiento de vídeo estereoscópico en un flujo de bits de HEVC                                      |
| Orientación de la pantalla                                | Especifica el vuelco y/o la rotación que deberían aplicarse a las imágenes de salida cuando se exhiben   |
| Información no válida de imagen decodificada              | Suma de comprobación de la imagen decodificada, que puede ser utilizada para la detección de errores     |
| Sincronización de información de unidad de decodificación | Tiempo de eliminación de sub-imágenes 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 predicción inter del flujo de bits                                  |

5 **[0088]** En HEVC, cada unidad NAL incluye un elemento sintáctico (por ejemplo, `nal_unit_type`) que indica un tipo de unidad NAL de la unidad NAL. Además, en HEVC, el decodificador de vídeo 30 puede identificar, basándose en el tipo de unidad NAL de una unidad NAL, la unidad NAL como asociada con uno de una pluralidad de tipos de imágenes. Estos tipos de imágenes pueden incluir imágenes de actualización de decodificación instantánea (IDR), imágenes de acceso aleatorio limpio (CRA), imágenes de acceso de subcapa temporal (TSA), imágenes de acceso de enlace roto (BLA) e imágenes codificadas que no son imágenes IDR, CRA o TSA.

10 **[0089]** La definición de una imagen IDR en HEVC puede ser similar a la definición de una imagen IDR en H.264/AVC. De manera similar, la definición de una imagen codificada en HEVC puede ser similar a la definición de una imagen codificada en H.264/AVC. Por ejemplo, la IDR y las imágenes codificadas pueden ser tipos de imagen heredados de la especificación H.264/AVC. Una imagen IDR puede hacer que un proceso de decodificación marque todas las imágenes de referencia como "no utilizadas como referencia". Debido a que las imágenes de referencia marcadas como "no utilizadas como referencia" pueden eliminarse de una memoria intermedia de imágenes decodificadas (DPB) que almacena las imágenes de referencia, una imagen IDR puede limpiar la DPB. Todas las imágenes  
15 codificadas que siguen a la imagen IDR en orden de decodificación pueden decodificarse sin predicción inter a partir de cualquier imagen que precede a la imagen IDR en orden de decodificación. La primera imagen de cada secuencia de vídeo codificada en orden de decodificación es una imagen IDR o una imagen BLA, o una imagen CRA que también es la primera imagen del flujo de bits. Cuando una imagen codificada de una unidad de acceso es una imagen  
20 IDR, la unidad de acceso puede denominarse una unidad de acceso IDR. En algunos ejemplos, una secuencia de vídeo codificada es una secuencia de unidades de acceso que consta, en orden de decodificación, de una unidad de acceso IDR seguida de cero o más unidades de acceso que no son unidades de acceso IRAP con `NoRaslOutputFlag` igual a 1, incluidas todas las unidades de acceso subsiguientes hasta, pero sin incluir, cualquier unidad de acceso IDR subsiguiente.

25 **[0090]** Los tipos de imágenes CRA y TSA son nuevos en HEVC y no están disponibles en la especificación H.264/AVC. El tipo de imagen CRA facilita la decodificación que comienza desde cualquier punto de acceso aleatorio (RAP) en medio de una secuencia de vídeo. Insertar imágenes CRA en una secuencia de vídeo puede ser más eficiente que insertar imágenes IDR en la misma secuencia de vídeo. Acceso aleatorio es el acto de iniciar un proceso de  
30 decodificación para un flujo de bits en un punto diferente al comienzo del flujo de bits. En HEVC, un flujo de bits que comienza con una imagen CRA puede ser un flujo de bits conforme a la norma. Es decir, la parte de un flujo de bits que comienza con una imagen CRA puede ajustarse a la especificación HEVC. Se puede utilizar una imagen TSA para indicar un punto de cambio de subcapa temporal válido.

35 **[0091]** En las aplicaciones de vídeo, tales como la radiodifusión y transmisión continua, los usuarios pueden cambiar entre diferentes canales y saltar a partes específicas de un vídeo. Idealmente, el cambio y el salto de canal de esta manera se deben realizar con un mínimo de retardo. El cambio y el salto de canal se pueden habilitar incluyendo imágenes de acceso aleatorio a intervalos regulares en flujos de bits de vídeo. Las imágenes IDR se pueden usar tanto en H.264/AVC como en HEVC como imágenes de acceso aleatorio. En otras palabras, la imagen IDR, especificada tanto en H.264/AVC como en HEVC, puede usarse para acceso aleatorio. Sin embargo, dado que una  
40 imagen IDR comienza una secuencia de vídeo codificada y puede siempre limpiar la DPB, las imágenes que siguen a

la imagen IDR en orden de decodificación no pueden usar imágenes decodificadas antes, en orden de decodificación, que la imagen IDR como imágenes de referencia. Por consiguiente, los flujos de bits que dependen de imágenes IDR para el acceso aleatorio pueden tener una eficiencia de codificación significativamente menor (por ejemplo, eficiencia de codificación un 6 % menor). Para mejorar la eficiencia de codificación, una imagen CRA en HEVC puede permitir a las imágenes que sigan a una imagen CRA en orden de decodificación, pero precedan a la imagen CRA en orden de salida, usar imágenes decodificadas antes de la CRA como referencia.

**[0092]** La FIG. 2 es un diagrama conceptual que ilustra una imagen CRA y las imágenes principales. Es decir, una estructura de predicción típica alrededor de una imagen CRA se muestra en el ejemplo de la FIG. 2. En el ejemplo de la FIG. 2, cada paralelogramo representa una imagen. Los números dentro de cada paralelogramo respectivo indican valores POC de ejemplo de la imagen respectiva representada por el paralelogramo respectivo. La imagen CRA (es decir, la imagen en el ejemplo de la FIG. 2 con valor POC 24) pertenece a un Grupo de imágenes (GOP) que contiene otras imágenes (es decir, imágenes con valores POC 17-23) después de la imagen CRA en orden de decodificación, pero antes de la imagen CRA en orden de salida. Estas imágenes (es decir, las imágenes que siguen a la imagen CRA en orden de decodificación, pero que preceden a la imagen CRA en orden de salida) se pueden denominar "imágenes iniciales" de la imagen CRA. El decodificador de vídeo 30 puede decodificar correctamente las imágenes iniciales de una imagen CRA actual si la decodificación comienza a partir de una imagen IDR o una imagen CRA que se produce en orden de decodificación antes de la imagen CRA actual. Sin embargo, es posible que el decodificador de vídeo 30 no pueda decodificar correctamente las imágenes iniciales de una imagen de CRA actual cuando se produce un acceso aleatorio desde la imagen de CRA actual. Por lo tanto, el decodificador de vídeo 30 puede descartar las imágenes iniciales de una imagen de CRA actual durante la decodificación de acceso aleatorio de la imagen de CRA actual.

**[0093]** La propagación de error se puede producir cuando decodificador de vídeo 30 utiliza una parte incorrectamente decodificada de una imagen de referencia cuando se decodifica una imagen actual. Para evitar la propagación de errores a partir de imágenes de referencia que pueden no estar disponibles dependiendo del lugar donde se inicie la decodificación, ninguna imagen en el GOP que sigue a una imagen CRA puede usar como referencia ninguna imagen que preceda a la imagen CRA en orden de decodificación u orden de salida (incluidas las imágenes principales de la imagen de la CRA). Así, en el ejemplo de la FIG. 2, todas las imágenes del próximo GOP que siguen a la imagen CRA, tanto en orden de decodificación como en orden de salida, no utilizarán ninguna imagen que preceda a la imagen CRA en orden de decodificación o en orden de salida como referencia.

**[0094]** Los mensajes SEI de punto de recuperación se pueden usar en H.264/AVC para proporcionar acceso aleatorio de una manera similar a las imágenes de CRA en HEVC. En otras palabras, H.264/AVC soporta funcionalidades de acceso aleatorio similares con un mensaje SEI de punto de recuperación. Una implementación de decodificador H.264/AVC puede o no soportar la funcionalidad de mensaje SEI de punto de recuperación. En HEVC, un flujo de bits que comienza con una imagen CRA se considera un flujo de bits conforme a la norma. Cuando un flujo de bits comienza con una imagen CRA, las imágenes iniciales de la imagen CRA pueden referirse a imágenes de referencia no disponibles y, por lo tanto, no pueden decodificarse correctamente. Sin embargo, la norma HEVC especifica que las imágenes iniciales de una imagen CRA inicial no se facilitan, de ahí el nombre "acceso aleatorio limpio". Para el establecimiento de un requisito de conformidad con el flujo de bits, HEVC puede especificar un proceso de decodificación para generar imágenes de referencia no disponibles para la decodificación de las imágenes principales sin salida. Sin embargo, las implementaciones de decodificador conformes a la norma no tienen que seguir ese proceso de decodificación, siempre que las implementaciones del decodificador puedan generar un resultado idéntico a cuando el proceso de decodificación se realiza desde el comienzo del flujo de bits. En HEVC, un flujo de bits conforme no puede incluir imágenes IDR.

**[0095]** Wang et al, "AHG9: On cross-layer alignments in HEVC 3DV and scalable extensions [AHG9.: En las alineaciones de capa cruzada en HEVC 3DV y las extensiones escalables]", el Equipo Conjunto de Colaboración sobre Codificación de Vídeo (JCT-VC) del SG-16 de ITU-T y el JTC 1/SC 29/WG 11 de ISO-IEC, 13.ª reunión, Incheon, Corea, abril de 2013, documento JCTVC-M0266\_v2 (en adelante, "JCTVC-M0266\_v2"), proporcionó un diseño para la definición del proceso de decodificación para las imágenes omitidas de acceso aleatorio de capa cruzada (CL-RAS). Al menos desde el 30 de mayo de 2014, JCTVC-M0266\_v2 está disponible en [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/13\\_Incheon/wg11/JCTVC-M0266-v2.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0266-v2.zip).

**[0096]** El diseño en JCTVC-M0266\_v2 define unidades de acceso IRAP (AU) para incluir todas las Au que tienen una imagen IRAP con `nuh_layer_id` igual a 0. El valor `nuh_layer_id` de una imagen puede indicar una capa a la que pertenece la imagen. El diseño en JCTVC-M0266\_v2 también define imágenes CL-RAS e imágenes IDR de enlace roto (BL-IDR) que se utilizan para detallar el proceso de decodificación de un flujo de bits a partir de una unidad de acceso IRAP de este tipo. Algunas de las definiciones relevantes se reproducen a continuación. Las partes subrayadas de las definiciones siguientes indican partes que difieren de las definiciones correspondientes en el Borrador de Trabajo de HEVC 10.

**unidad de acceso:** Un conjunto de *unidades NAL* que están asociadas entre sí de acuerdo con una regla de clasificación especificada, son consecutivas en *orden de decodificación* y contienen todas las imágenes codificadas asociadas con el mismo tiempo de salida.

NOTA 1 - Además de contener las unidades VCL NAL de las imágenes codificadas, una unidad de acceso también puede contener unidades NAL no VCL.

5 **secuencia de vídeo codificada (CVS):** Una secuencia de *unidades de acceso* que consta, en orden de descodificación, de una *unidad de acceso IRAP inicial*, seguida de cero o más *unidades de acceso* que no son *unidades de acceso IRAP iniciales*, incluidas todas las *unidades de acceso* subsiguientes hasta, pero sin incluir ninguna *unidad de acceso* subsiguiente que sea una *unidad de acceso IRAP inicial*.

10 **imagen omitida de acceso aleatorio de capa cruzada (CL-RAS):** Una *imagen codificada* para la cual cada *unidad VCL NAL* tiene *nal\_unit\_type* igual a CL\_RAS\_R o CL\_RAS\_N.

15 NOTA 4 - Cuando NoCirasOutputFlag es igual a 1, la imagen CL-RAS no se emite y es posible que no se pueda descodificar correctamente, ya que la imagen CL-RAS puede contener referencias a imágenes que no están presentes en el flujo de bits. Las imágenes CL-RAS no se usan como imágenes de referencia para el proceso de descodificación de imágenes que no sean CL-RASL.

20 **unidad de acceso al punto de acceso aleatorio intra (IRAP) inicial:** Una *unidad de acceso IRAP* en la que la *imagen codificada* con *nuh\_layer\_id* igual a 0 tiene NoRasOutputFlag igual a 1.

**imagen de actualización de descodificación instantánea (IDR):** Una *imagen IRAP* para la cual cada *unidad VCL NAL* tiene *nal\_unit\_type* igual a IDR\_W\_RADL, IDR\_N\_LP, BL IDR W RADL o BL IDR N LP.

25 **unidad de acceso al punto de acceso intra aleatorio (IRAP):** Una *unidad de acceso* en la que la *imagen codificada* con un ID de capa *nuh* igual a 0 es una *imagen IRAP*.

30 **[0097]** En el borrador de trabajo de HEVC 10, una imagen inicial omitida de acceso aleatorio (RASL) es una imagen codificada para la cual cada unidad VCL NAL tiene *nal\_unit\_type* igual a RASL\_R o RASL\_N. Todas las imágenes RASL son imágenes principales de una imagen BLA o CRA asociada. Cuando la imagen IRAP asociada tiene NoRasOutputFlag igual a 1, la imagen RASL no se emite y es posible que no se pueda descodificar correctamente, ya que la imagen RASL puede contener referencias a imágenes que no estén presentes en el flujo de bits. Las imágenes RASL no se usan como imágenes de referencia para el proceso de descodificación de imágenes que no son RASL. Cuando están presentes, todas las imágenes RASL preceden, en orden de descodificación, a todas las imágenes posteriores de la misma imagen RAP asociada.

35 **[0098]** Es decir, en el borrador de trabajo de HEVC 10, la variable NoRasOutputFlag se define para cada imagen IRAP. Cuando NoRasOutputFlag es igual a 1, la imagen IRAP inicia una nueva secuencia de vídeo codificada, y las imágenes RASL asociadas, cuando están presentes, no se emiten. En JCTVC-M0266v2, la definición de NoRasOutputFlag se cambia de la forma siguiente:

40 Cuando la imagen actual es una imagen IRAP, se aplica lo siguiente: [Tenga en cuenta que esta definición de NoRasOutputFlag es la misma que en JCT3V-C1004v4.zip, pero se movió para ser aplicable también a las imágenes IRAP con *nuh\_layer\_id* mayor que 0.]

45 - Si la imagen actual con un *nuh\_layer\_id* particular es una imagen IDR, una imagen BLA, la primera imagen con esa *nuh\_layer\_id* particular en el flujo de bits en orden de descodificación, o la primera imagen con esa ID de capa *nuh* particular que sigue a una unidad NAL de final de secuencia orden de descodificación, la variable NoRasOutputFlag se establece igual a 1.

50 - De lo contrario, si algunos medios externos no especificados en esta especificación están disponibles para establecer la variable HandleCraAsBlaFlag en un valor para la imagen actual, la variable HandleCraAsBlaFlag se establece igual al valor proporcionado por los medios externos y la variable NoRasOutputFlag se establece igual a HandleCraAsBlaFlag.

55 - De lo contrario, la variable HandleCraAsBlaFlag se establece igual a 0 y la variable NoRasOutputFlag se establece igual a 0.

60 **[0099]** Del mismo modo, con el fin de indicar al proceso de descodificación si las imágenes CL-RAS asociadas, cuando están presentes, se emiten otra variable, NoCirasOutputFlag, puede definirse de la forma siguiente:

60 Cuando la unidad de acceso que contiene la imagen actual es una unidad de acceso IRAP inicial, se aplica lo siguiente:

65 - Si la imagen actual es una imagen BLA, una imagen CRA con NoRasOutputFlag igual a 1, o una imagen BL-IDR, NoCirasOutputFlag es igual a 1.

- De lo contrario, si algunos medios externos no especificados en esta memoria descriptiva están disponibles para establecer la variable NoClrasOutputFlag en un valor, la variable NoClrasOutputFlag se establece en el valor proporcionado por los medios externos.

5 - De lo contrario, NoClrasOutputFlag es igual a 0.

[0100] El proceso de decodificación puede definirse de tal manera que imágenes de referencia no disponibles se generan (similar a la generación de imágenes de referencia no disponibles en HEVC WD10) cuando se decodifica una imagen CL-RAS de una unidad de acceso inicial IRAP (AU) y cuando NoClrasOutputFlag es igual a 1. En algunos ejemplos, un flujo de bits no incluye imágenes CL-RAS codificadas. Sin embargo, algunas operaciones HRD pueden depender del flujo de bits, incluidas las imágenes CL-RAS codificadas. Por ejemplo, en las operaciones de DRH tal como se especifica en el Anexo C del Borrador de Trabajo de HEVC 10, las unidades de acceso que contienen imágenes RASL pueden ser tomadas en cuenta para la obtención de los tiempos de llegada y eliminación de CPB. Por consiguiente, un descodificador de vídeo puede realizar un proceso para generar tales imágenes de referencia no disponibles. El codificador de vídeo puede establecer cada valor de muestra de cada píxel en una imagen de referencia no disponible generada a un valor máximo. Por ejemplo, como se describe en la sección 8.3.3.2 del Borrador de Trabajo de HEVC 10:

Cuando se invoca este proceso, se genera una imagen no disponible de la forma siguiente:

- El valor de cada elemento en el conjunto de muestras  $S_L$  se establece en  $1 \ll (\text{BitDepth}_L - 1)$ .
- El valor de cada elemento en las matrices de muestra  $S_{Cb}$  y  $S_{Cr}$  para la imagen se establece en  $1 \ll (\text{BitDepth}_C - 1)$ .
- El modo de predicción  $\text{CuPredMode}[x][y]$  se establece igual a  $\text{MODE\_INTRA}$  para  $x = 0..pic\_width\_in\_luma\_samples - 1$ ,  $y = 0..pic\_height\_in\_luma\_samples - 1$ .

[0101] La Solicitud de Patente provisional de EE. UU. 61/812,225, presentada el 16 de abril de 2013, se refiere a una o más técnicas de esta divulgación. En flujos de bits con más de una capa basada en extensiones HEVC, los puntos de acceso aleatorio se definen como aquellas unidades de acceso donde todas las imágenes en la unidad de acceso son imágenes IRAP. Esto impone restricciones en la capacidad de acceso aleatorio, y puede ser deseable proporcionar también capacidad de acceso aleatorio en aquellas unidades de acceso donde las imágenes con  $nuh\_layer\_id$  igual a 0 son imágenes IRAP. La Solicitud de Patente Provisional de EE. UU. 61/812,225 definió las nociones de imágenes CL-RAS asociadas con unidades de acceso IRAP que contienen una imagen IRAP con una identificación de capa  $nuh$  igual a 0. Para soportar capacidades de acceso aleatorio, sería necesario definir el comportamiento del descodificador cuando la decodificación se inicie desde una unidad de acceso en la que la imagen con  $nuh\_layer\_id$  igual a 0 es una imagen IRAP y no todas las imágenes en la AU son imágenes IRAP. La solicitud de patente provisional de EE. UU. 61/812,225 y JCTVC-M0266\_v2 proporcionan algunos de los procesos relacionados con esto.

[0102] Sin embargo, los diseños existentes, incluidos los proporcionados en la Solicitud de Patente Provisional de EE. UU. 61/812,225 y JCTVC-M0266\_v2, están asociados con uno o más problemas. Por ejemplo, los servidores de transmisión continua y los elementos de red conscientes de los medios (MANE) pueden decidir no enviar imágenes CL-RAS a los receptores cuando dichas imágenes CL-RAS no se pueden decodificar correctamente (es decir, cuando NoClrasOutputFlag es igual a 1 para la IRAP AU inicial). Para permitir que los descodificadores conformes puedan manejar ambos casos en los que los flujos de bits contengan o no tales imágenes CL-RAS, los flujos de bits de ambos casos deben definirse como flujos de bits conformes y, por lo tanto, se necesitan dos conjuntos de parámetros HRD. De lo contrario, solo un caso puede definirse como flujos de bits conformes y, por lo tanto, se necesitan mejoras en los descodificadores conformes para manejar el otro caso para el cual los flujos de bits no son flujos de bits conformes.

[0103] En otro ejemplo de las deficiencias asociadas con la Solicitud de Patente Provisional de EE. UU. 61/812,225 y JCTVC-M0266\_v2, para la generación de imágenes de referencia no disponibles cuando se decodifica una imagen CL-RAS de una IRAP AU inicial y cuando NoClrasOutputFlag es igual a 1, no solo se deben generar los subconjuntos RPS  $\text{RefPicSetStFoll}$  y  $\text{RefPicSetLtFoll}$ , sino que también se deben generar los subconjuntos RPS  $\text{RefPicSetStCurrBefore}$ ,  $\text{RefPicSetStCurrAfter}$  y  $\text{RefPicSetLtCurr}$ . Si tal proceso no está definido para los subconjuntos  $\text{RefPicSetStCurrBefore}$ ,  $\text{RefPicSetStCurrAfter}$  y  $\text{RefPicSetLtCurr}$ , entonces el proceso de decodificación de las imágenes CL-RAS no se puede realizar porque no hay imagen de referencia (ni la imagen de referencia original, ni las imágenes de referencia generadas) en esos subconjuntos utilizados para la predicción inter.

[0104] Esta divulgación define un NoClrasPicPresentFlag para especificar si las imágenes CL-RAS asociadas con una IRAP AU inicial están presentes. Por lo tanto, un dispositivo (por ejemplo, un descodificador de vídeo u otro dispositivo) puede determinar, basándose en un valor (por ejemplo, NoClrasPicPresentFlag) si todas las imágenes CL-RAS de (es decir, asociadas con) una unidad de acceso IRAP están presentes en un flujo de bits de datos de vídeo. El NoClrasPicPresentFlag es una variable que se puede establecer, por ejemplo, por medios externos o en un mensaje SEI. Por lo tanto, en algunos ejemplos, el dispositivo puede establecer el valor (por ejemplo, NoClrasPicPresentFlag)

basándose al menos en parte en los datos de un mensaje SEI del flujo de bits de datos de vídeo. Además, en algunos ejemplos, el valor (por ejemplo, NoCIRasPicPresentFlag) puede establecerse mediante un mecanismo (es decir, un medio externo) fuera del proceso de descodificación de los datos de vídeo. Como se describe en otra parte de esta divulgación, el dispositivo puede realizar, basándose al menos en parte en el valor (por ejemplo, NoCIRasPicPresentFlag), una operación de CPB.

**[0105]** En algunos ejemplos, el codificador de vídeo 20 puede generar un flujo de bits que incluya una secuencia de bits que forme una representación de imágenes codificadas y datos asociados. El flujo de bits puede incluir elementos sintácticos a partir de los cuales un dispositivo que recibe el flujo de bits puede obtener un valor (por ejemplo, NoCIRasPicPresentFlag) que especifica si las imágenes omitidas de acceso aleatorio de capa cruzada (CL-RAS) de una unidad de acceso de punto de acceso aleatorio intra (IRAP) están presentes en el flujo de bits. El codificador de vídeo 20 puede emitir el flujo de bits. En algunos ejemplos, el codificador de vídeo 20 (u otro dispositivo, como MANE) puede generar un mensaje SEI que establece una variable que especifica si las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados. En algunos ejemplos, el dispositivo puede incluir el mensaje SEI en el flujo de bits.

**[0106]** Además, algunas técnicas de esta divulgación definen la conformidad de los flujos de bits en los que las imágenes CL-RAS se asocian con una unidad de acceso IRAP inicial que tiene una imagen IRAP con un ID de capa nuh igual a 0 y NoRasOutputFlag igual a 1. Se señalizan los elementos sintácticos en un mensaje SEI de período de almacenamiento en memoria intermedia que corresponden a desviaciones de retardo de CPB y DPB cuando las imágenes CL-RAS no están presentes en el flujo de bits de conformidad. Además, algunas técnicas de esta divulgación definen elementos sintácticos en un mensaje SEI de período de almacenamiento en memoria intermedia que corresponden a un retardo de eliminación inicial de CPB alternativa y valores de desviación para tener en cuenta la eliminación de imágenes CL-RAS. Además, algunas técnicas de esta divulgación definen los comportamientos de CPB y DPB utilizando los elementos sintácticos recién definidos cuando las imágenes CL-RAS no están presentes en el flujo de bits correspondiente.

**[0107]** Adicionalmente, algunas técnicas de esta divulgación definen un proceso completo de generación para imágenes de referencia no disponibles cuando se descodifica una imagen CL-RAS en un IRAP AU inicial y cuando NoCIRasOutputFlag es igual a 1. Por ejemplo, cuando las imágenes CL-RAS no se emiten (por ejemplo, cuando NoCIRasOutputFlag es igual a 1) y la imagen actual es una imagen CL-RAS en una unidad de acceso IRAP inicial, un descodificador de vídeo (por ejemplo, descodificador de vídeo 30) puede generar imágenes de referencia no disponibles para la imagen CL-RAS. En algunos ejemplos, una unidad de acceso IRAP es una unidad de acceso donde la imagen con nuh\_layer\_id igual a 0 es una imagen IRAP. Una unidad de acceso IRAP inicial puede ser una unidad de acceso IRAP donde la imagen IRAP con nuh\_layer\_id igual a 0 tiene NoRasOutputFlag igual a 1.

**[0108]** Por otra parte, algunas técnicas de esta divulgación definen un nuevo mensaje SEI de parámetros HRD de capa cruzada que se utiliza para señalar los parámetros de velocidad de transmisión de bits y tamaño de CPB alternativa para unidades de acceso IRAP. Estos parámetros se pueden utilizar cuando la unidad de acceso IRAP correspondiente es una unidad de acceso IRAP inicial con imágenes CL-RAS asociadas no presentes.

**[0109]** La FIG. 3 es un diagrama de bloques que ilustra un codificador de vídeo 20 de ejemplo que puede implementar las técnicas de esta divulgación. La FIG. 3 se proporciona para propósitos de explicación y no debería considerarse limitadora de las técnicas tales como las ampliamente ejemplificadas y descritas en esta divulgación. Para los propósitos de explicación, esta divulgación describe el codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

**[0110]** En el ejemplo de la FIG. 3, 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 por entropía 118. La unidad de procesamiento de predicción 100 incluye una unidad de procesamiento de predicción inter 120 y una unidad de procesamiento de predicción intra 126. La unidad de procesamiento de predicción inter 120 incluye una unidad de estimación de movimiento 122 y una unidad de compensación de movimiento 124. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales.

**[0111]** El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede codificar cada CTU en un fragmento de una imagen de los datos de vídeo. Cada una de las CTU puede asociarse a bloques de árbol de codificación (CTB) de luma de igual tamaño y a los CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 100 puede realizar la partición de árbol cuádruple para dividir los CTB de la CTU en bloques progresivamente más pequeños. Los bloques más pequeños pueden ser bloques de codificación de CU. Por ejemplo, la unidad de procesamiento de predicción 100 puede dividir un CTB asociado a una CTU en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, etc.

5 **[0112]** El codificador de vídeo 20 puede codificar CU de un CTB 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 lo tanto, cada PU puede asociarse a un bloque de predicción de luma y a los correspondientes bloques de predicción de croma. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden dar soporte a las PU de diversos tamaños. El tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU, y el tamaño de una PU puede referirse al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular sea de  $2N \times 2N$ , el codificador de vídeo 20 y el descodificador de vídeo 30 pueden dar soporte a tamaños de PU de  $2N \times 2N$  o  $N \times N$  para la predicción intra, y a tamaños de PU simétricas de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ ,  $N \times N$  o similares para la predicción inter. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden dar soporte a la partición asimétrica para tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$ , para la predicción inter.

15 **[0113]** La unidad de procesamiento de predicción inter 120 puede generar datos predictivos para una PU realizando predicción inter 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 predicción inter 120 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, un fragmento P o un fragmento B. En un fragmento en I, todas las PU tiene predicción intra. Por lo tanto, si la PU está en un fragmento I, la unidad de procesamiento de predicción inter 120 no realiza predicción inter en la PU. Por tanto, para bloques codificados en el modo I, el bloque predictivo se forma usando predicción espacial a partir de bloques adyacentes previamente codificados dentro de la misma trama.

25 **[0114]** Si una PU está en un fragmento 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 contenga bloques de muestra que se correspondan más estrechamente con los bloques de muestra 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 codificación de la PU y una ubicación de referencia asociada con la región de referencia. Por ejemplo, el vector de movimiento puede ser un vector bidimensional que proporciona una desviación respecto a las coordenadas en la imagen actual a las coordenadas en una imagen de referencia. La unidad de estimación de movimiento 122 puede emitir el índice de referencia y el vector de movimiento como la información de movimiento de la PU. La unidad de compensación de movimiento 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.

35 **[0115]** Si una PU está en un fragmento B, la unidad de estimación de movimiento 122 puede llevar a cabo predicción unidireccional o predicción bidireccional para la PU. Para realizar predicción unidireccional para la PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia de la 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 emitir, 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 predicción de la PU y una ubicación de referencia asociada con la región de referencia, y uno o más indicadores de dirección de predicción que indican si la imagen de referencia está en 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.

40 **[0116]** Para realizar predicción inter 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 movimiento 122 puede generar índices de referencia que indiquen posiciones en RefPicListO 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 las ubicaciones de referencia asociadas con las regiones de referencia y un bloque de predicción de la PU. La información de movimiento de la PU puede incluir los índices de referencia y los vectores de movimiento de la PU. La unidad de compensación de movimiento 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.

50 **[0117]** La unidad de procesamiento de predicción intra 126 puede generar datos predictivos para una PU realizando predicción intra en la PU. Los datos predictivos para la PU pueden incluir bloques predictivos para la PU y varios elementos sintácticos. La unidad de procesamiento de predicción intra 126 puede realizar la predicción intra en las PU en fragmentos I, fragmentos P y fragmentos B.

60 **[0118]** Para realizar la predicción intra en una PU, la unidad de procesamiento de predicción intra 126 puede usar múltiples modos de predicción intra para generar múltiples conjuntos de datos predictivos para la PU. Las PU

adyacentes pueden estar arriba, arriba y a la derecha, arriba y a la izquierda, o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha, de arriba a abajo, para las PU, CU y CTU. La unidad de procesamiento de predicción intra 126 puede usar diversos números de modos de predicción intra, por ejemplo, 33 modos de predicción intra direccional. En algunos ejemplos, el número de modos de predicción intra puede depender del tamaño de los bloques de predicción de la PU.

**[0119]** La unidad de procesamiento de predicción 100 puede seleccionar los datos predictivos para PU de una CU entre los datos predictivos generados por la unidad de procesamiento de predicción inter 120 para las PU o los datos predictivos generados por la unidad de procesamiento de predicción intra 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 las mediciones 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.

**[0120]** La unidad de generación residual 102 puede generar, basándose en los bloques de codificación (por ejemplo, los bloques de codificación de luma, Cb y Cr) de una CU y los bloques predictivos seleccionados (por ejemplo, los bloques predictivos de luma, Cb y Cr seleccionados) de las PU de la CU, los bloques residuales (por ejemplo, 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 manera que cada muestra en los bloques residuales tenga un valor igual a una diferencia entre una muestra en un bloque de codificación de la CU y una muestra correspondiente en un correspondiente bloque predictivo seleccionado de una PU de la CU.

**[0121]** La unidad de procesamiento de transformada 104 puede realizar división de árbol cuádruple para dividir los bloques residuales de una CU en bloques de transformada de (es decir, asociados con) TU de la CU. De este modo, una TU puede tener (es decir, estar asociada con) un bloque de transformada de luma y dos bloques de transformada de croma correspondientes. Los tamaños y las posiciones de los bloques de transformada de luma y croma de las TU de una CU pueden o no basarse en los tamaños y posiciones de los bloques de predicción de las PU de la CU. Una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT) puede incluir nodos asociados con (por ejemplo, correspondientes a) cada una de las regiones. Las TU de una CU pueden corresponder a nodos de hoja del RQT.

**[0122]** 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 puede tratarse como un bloque de coeficientes de transformada.

**[0123]** 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 transformada 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 del QP asociado a la CU. La cuantificación puede introducir pérdida de información, por lo que los coeficientes de transformada cuantificados pueden tener una precisión inferior a los originales.

**[0124]** La unidad de cuantificación inversa 108 y la unidad de procesamiento de transformada inversa 110 pueden aplicar 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. Mediante la reconstrucción de 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.

**[0125]** La unidad de filtro 114 puede realizar una o más operaciones de desbloqueo para reducir las distorsiones de bloqueo en los bloques de codificación asociados con 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 las una o más operaciones de desbloqueo en los bloques de codificación reconstruidos. La unidad de procesamiento de predicción inter 120 puede usar una imagen de referencia que contenga los bloques de codificación reconstruidos para realizar la predicción inter en las PU de otras imágenes. Además, la unidad de procesamiento de predicción intra 126 puede usar bloques de codificación reconstruidos en la memoria intermedia de imágenes descodificadas 116 para realizar predicción intra en otras PU en la misma imagen que la CU.

**[0126]** La unidad de codificación por entropía 118 puede recibir datos de otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación por 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 por entropía 118 puede realizar una o más operaciones de codificación por entropía en los datos para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 118 puede realizar una operación de codificación de longitud variable adaptable al contexto (CAVLC), una operación de CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una operación de codificación por entropía de partición de intervalos de probabilidad (PIPE), una operación de codificación Exponencial-Golomb u otro tipo de operación de codificación por entropía en los datos. El codificador de vídeo 20 puede emitir un flujo de bits que incluya datos codificados por entropía, generados por la unidad de codificación por entropía 118. Por ejemplo, el flujo de bits puede incluir datos que representen un RQT para una CU.

**[0127]** En algunos ejemplos, el flujo de bits generado por el codificador de vídeo 20 de la FIG. 3 puede incluir uno o más elementos sintácticos a partir de los cuales un dispositivo que recibe el flujo de bits (por ejemplo, el descodificador de vídeo 30) puede obtener un valor que especifica si las imágenes CL-RAS de una unidad de acceso IRAP están presentes en el flujo de bits. Por ejemplo, un mensaje SEI de período de almacenamiento en memoria intermedia generado por el codificador de vídeo 20 puede incluir uno o más elementos sintácticos desde los cuales un dispositivo que recibe el flujo de bits puede obtener un valor que especifica si las imágenes CL-RAS de una unidad de acceso IRAP están presentes en el flujo de bits.

**[0128]** La FIG. 4 es un diagrama de bloques que ilustra un descodificador de vídeo 30 de ejemplo que está configurado para implementar las técnicas de esta divulgación. La FIG. 4 se proporciona para propósitos de explicación y no se limita a las técnicas como las ampliamente ejemplificadas y descritas en esta divulgación. Con fines de explicación, esta divulgación describe el descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

**[0129]** En el ejemplo de la FIG. 4, el descodificador de vídeo 30 incluye una unidad de descodificación por 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 predicción intra 166. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

**[0130]** 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 por entropía 150 puede recibir unidades NAL de la CPB 151 y analizar las unidades NAL para obtener (por ejemplo, descodificar) elementos sintácticos. La unidad de descodificación por entropía 150 puede descodificar por entropía elementos sintácticos codificados por 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.

**[0131]** Las unidades NAL del flujo de bits pueden incluir unidades NAL de fragmentos codificados. Como parte de la descodificación del flujo de bits, la unidad de descodificación por entropía 150 puede extraer y descodificar por entropía elementos sintácticos de las unidades de NAL de fragmentos codificados. Cada uno de los fragmentos codificados puede incluir una cabecera de fragmento y datos de fragmento. La cabecera de fragmento puede contener elementos sintácticos pertenecientes a un fragmento. Los elementos sintácticos en la cabecera de fragmento pueden incluir un elemento sintáctico que identifique un PPS asociado a una imagen que contenga el fragmento.

**[0132]** De acuerdo con una o más técnicas de esta divulgación, uno o más componentes del descodificador de vídeo 30 (por ejemplo, la unidad de descodificación por entropía 150) puede determinar un valor (por ejemplo, NoCIRasPicPresentFlag). Además, de acuerdo con una o más técnicas de esta divulgación, el descodificador de vídeo 30 puede determinar, basándose en el valor, si todas las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de bits de datos de vídeo. Además, el dispositivo puede descodificar el flujo de bits de datos de vídeo.

**[0133]** Además de obtener (por ejemplo, descodificar) elementos sintácticos del flujo de bits, el descodificador de vídeo 30 puede realizar una operación de descodificación en una CU. Para realizar la operación de reconstrucción en una CU, 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.

**[0134]** Como parte de la realización de una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 154 puede cuantificar de forma inversa, es decir, descuantificar, bloques de coeficientes de (por ejemplo, asociados con) la TU. La unidad de cuantificación inversa 154 puede utilizar 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 aplicar por la unidad de cuantificación inversa 154. Es decir, la relación de compresión, es decir, la relación entre el 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 por entropía empleado.

[0135] Después de que la unidad de cuantificación inversa 154 cuantifique inversamente un bloque de coeficientes, la unidad de procesamiento de transformada inversa 156 puede aplicar una o más transformadas inversas al bloque de coeficientes para generar un bloque residual asociado con la TU. Por ejemplo, la unidad de procesamiento de transformada inversa 156 puede aplicar una DCT inversa, una transformada entera inversa, una transformada de Karhunen-Loeve (KLT) inversa, una transformada de rotación inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes.

[0136] Si se codifica una PU usando predicción intra, la unidad de procesamiento de predicción intra 166 puede realizar la predicción intra para generar bloques predictivos para la PU. La unidad de procesamiento de predicción intra 166 puede usar un modo de predicción intra para generar los bloques predictivos de luma, Cb y Cr para la PU basándose en los bloques predictivos de las PU espacialmente contiguas. La unidad de procesamiento de predicción intra 166 puede determinar el modo de predicción intra para la PU basándose en uno o más elementos sintácticos descodificados a partir del flujo de bits.

[0137] 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 los elementos sintácticos extraídos del flujo de bits. Además, si se codifica una PU usando la predicción inter, la unidad de descodificación por 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 bloques de muestras en los uno o más bloques de referencia para la PU, bloques predictivos de luma, Cb y Cr para la PU.

[0138] La unidad de reconstrucción 158 puede usar los bloques de transformada (por ejemplo, los bloques de transformada de luma, Cb y Cr) asociados con las TU de una CU y los bloques predictivos (por ejemplo, los bloques predictivos de luma, Cb y Cr) de las PU de la CU, es decir, cualquiera de los datos de predicción intra o de los datos de predicción inter, según corresponda, para reconstruir los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) de la CU. Por ejemplo, la unidad de reconstrucción 158 puede agregar muestras de los bloques de transformada (por ejemplo, los bloques de transformada de luma, Cb y Cr) a las muestras correspondientes de los bloques predictivos (por ejemplo, los bloques predictivos de luma, Cb y Cr) para reconstruir los bloques de codificación (por ejemplo, bloques de codificación luma, Cb y Cr) de la CU.

[0139] La unidad de filtro 160 puede realizar una operación de desbloqueo para reducir las distorsiones de bloqueo asociadas con los bloques de codificación (por ejemplo, los bloques de codificación de luma, Cb y Cr) de la CU. El descodificador de vídeo 30 puede almacenar los bloques de codificación (por ejemplo, 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 la posterior compensación de movimiento, la predicción intra y la presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el descodificador de vídeo 30 puede realizar, basándose en los bloques (por ejemplo, los bloques de luma, Cb y Cr) en la memoria intermedia de imágenes descodificadas 162, operaciones de predicción intra o de predicción inter en PU de otras CU. De esta forma, el descodificador de vídeo 30 puede extraer, a partir del flujo de bits, los niveles de coeficientes de transformada de bloques de coeficientes, cuantificar inversamente los niveles de coeficientes de transformada, aplicar una transformada a los niveles de coeficientes de transformada para generar bloques de transformada, generar, basándose al menos en parte en bloques de transformada, uno o más bloques de codificación, y emitir el uno o más bloques de codificación para visualizar.

[0140] En un primer conjunto de ejemplos de las técnicas de esta divulgación, los cambios en la sintaxis, la semántica y procesos de descodificación necesarios para la definición de la conformidad con el uso de imágenes CL-RAS se definen en relación a los procesos particulares descritos en JCTVC-M0266v2 y la Solicitud de Patente Provisional de EE. UU. 61/812,225. En este ejemplo, los elementos sintácticos introducidos para los parámetros HRD se incluyen en el mensaje SEI de período de almacenamiento en memoria intermedia.

[0141] En la primera serie de ejemplos de las técnicas de esta divulgación, se puede proporcionar la siguiente definición de imágenes CL-RAS asociadas:

**imágenes CL-RAS asociadas:** Imágenes CL-RAS para una IRAP AU inicial donde las imágenes CL-RAS dentro de una CVS con valor de ID de capa nuh\_layer\_id contenidas en layerSetLayerIdList que suceden a la primera imagen en una IRAP AU inicial y preceden, en orden de descodificación, a una imagen IRAP con valor de ID de capa nuh\_layer\_id.

[0142] Además, en el primer conjunto de ejemplos de las técnicas de esta divulgación, los siguientes se pueden hacer cambios al proceso de descodificación general descrito en el borrador de trabajo de HEVC 10. Los borradores de trabajo de SHVC y MV-HEVC definen algunas partes de SHVC y MV-HEVC en términos de cambios relativos al Borrador de Trabajo de HEVC (por ejemplo, Borrador de Trabajo de HEVC 10). A lo largo de las siguientes secciones, los pasajes marcados con subrayado son cambios adicionales en la sintaxis, semántica y proceso de descodificación del Borrador de Trabajo de HEVC 10 introducido por JCTVC-M0266v2 para implementar SHVC y/o MV-HEVC. Los pasajes marcados con cursiva en negrita son cambios adicionales introducidos por la divulgación actual. Los pasajes en cursiva marcados con dobles corchetes se eliminan mediante la divulgación actual del Borrador de Trabajo de HEVC 10 (o los cambios introducidos en los Borradores de trabajo de SHVC o MV-HEVC al Borrador de Trabajo de HEVC 10) para implementar SHVC, MV-HEVC u otras extensiones.

### 8.1 Proceso de descodificación general

[0143] La entrada a este proceso es un flujo de bits. La salida de este proceso es una lista de imágenes descodificadas.

[0144] La lista de identificadores de capa TargetDecLayerIdList, que especifica la lista de valores nuh\_layer\_id, en orden creciente de valores nuh\_layer\_id, de las unidades NAL para ser descodificado, se especifica de la forma siguiente:

- Si algunos medios externos, no especificados en esta memoria descriptiva, están disponibles para establecer TargetDecLayerIdList, TargetDecLayerIdList se establece mediante los medios externos.
- De lo contrario, si el proceso de descodificación se invoca en una prueba de conformidad del flujo de bits como se especifica en la subcláusula C.1, TargetDecLayerIdList se establece como se especifica en la subcláusula C.1.
- De lo contrario, TargetDecLayerIdList solo contiene un valor de ID de capa nuh que es igual a 0.

[0145] La variable HighestTid, que identifica la más alta subcapa temporal a descodificar, se especifica de la forma siguiente:

- Si algunos medios externos, no especificados en esta memoria intermedia, están disponibles para establecer HighestTid, HighestTid se establece por medios externos.
- De lo contrario, si el proceso de descodificación se invoca en una prueba de conformidad del flujo de bits como se especifica en la subcláusula C.1, HighestTid se establece como se especifica en la subcláusula C.1.
- De lo contrario, HighestTid se establece igual a sps\_max\_sub\_layers\_minus1.

[0146] El proceso de extracción de sub-flujo de bits como se especifica en la cláusula 10 se aplica con el flujo de bits, HighestTid y TargetDecLayerIdList como entradas, y la salida se asigna a un flujo de bits denominado BitstreamToDecode.

[0147] Los procesos de descodificación que se especifican en el resto de esta subcláusula se aplican a cada imagen codificada, denominada la imagen actual y se denotan mediante la variable CurrPic, en BitstreamToDecode.

[0148] Dependiendo del valor de chroma\_format\_idc, el número de matrices de muestra de la imagen actual es el siguiente:

- Si chroma\_format\_idc es igual a 0, la imagen actual consta de 1 matriz de muestra S<sub>L</sub>.
- De lo contrario (chroma\_format\_idc no es igual a 0), la imagen actual consta de 3 matrices de muestra S<sub>L</sub>, S<sub>Cb</sub>, S<sub>Cr</sub>.

[0149] El proceso de descodificación para la imagen actual toma como entradas de los elementos sintácticos y variables en mayúsculas de la cláusula 7. Al interpretar la semántica de cada elemento sintáctico en cada unidad NAL, el término "el flujo de bits" (o parte del mismo, por ejemplo, una CVS del flujo de bits) se refiere a BitstreamToDecode (o parte del mismo).

[0150] El proceso de descodificación se especifica de tal manera que todos los descodificadores producirán imágenes descodificadas recortadas idénticas numéricamente. Cualquier proceso de descodificación que produzca imágenes descodificadas recortadas idénticas a las producidas por el proceso descrito en el presente documento (con el orden de salida o la sincronización de salida correcta, como se especifica) cumple con los requisitos del proceso de descodificación de esta memoria descriptiva.

[0151] Cuando la imagen actual es una imagen IRAP, se aplica lo siguiente:

- Si la imagen actual con un ID de capa nuh particular es una imagen IDR, una imagen BLA, la primera imagen con ese ID de capa nuh particular en el flujo de bits en orden de descodificación, o la primera imagen con ese nuh\_layer\_id particular que sigue a un final de unidad NAL de secuencia en orden de descodificación, la variable NoRasOutputFlag se establece igual a 1.

- De lo contrario, si algunos medios externos no especificados en esta especificación están disponibles para establecer la variable HandleCraAsBlaFlag en un valor para la imagen actual, la variable HandleCraAsBlaFlag se establece igual al valor proporcionado por los medios externos y la variable NoRasOutputFlag se establece igual a HandleCraAsBlaFlag.

- De lo contrario, la variable HandleCraAsBlaFlag se establece igual a 0 y la variable NoRasOutputFlag se establece igual a 0.

**[0152]** Cuando la unidad de acceso que contiene la imagen actual es una unidad de acceso IRAP inicial, se aplica lo siguiente:

- Si la imagen actual es una imagen BLA, una imagen CRA con NoRasOutputFlag igual a 1, o una imagen BL-IDR, NoClrasOutputFlag es igual a 1.

- De lo contrario, si algunos medios externos no especificados en esta memoria descriptiva están disponibles para establecer la variable NoClrasOutputFlag en un valor, la variable NoClrasOutputFlag se establece en el valor proporcionado por los medios externos.

- De lo contrario, NoClrasOutputFlag es igual a 0.

**[0153]** Cuando la AU actual es una IRAP AU inicial, el valor de NoClrasPicPresentFlag se especifica por medios externos. NoClrasPicPresentFlag igual a 0 especifica que todas las imágenes CL-RAS asociadas con la IRAP AU inicial no están presentes en el flujo de bits. NoClrasPicPresentFlag igual a 1 especifica que todas las imágenes CL-RAS que están asociadas con la IRAP AU inicial están presentes en el flujo de bits.

**[0154]** De forma alternativa, NoClrasPicPresentFlag se puede obtener a partir de los elementos sintácticos presentes como parte del flujo de bits, por ejemplo, mediante nuevos valores de tipo de unidad NAL, o a través de un indicador en el VPS, a través de indicadores en un mensaje SEI, por ejemplo en el mensaje SEI de período de almacenamiento en memoria intermedia.

**[0155]** Cuando la imagen actual tiene nuh\_layer\_id igual a 0, se invoca el proceso de descodificación para una imagen codificada con un ID de capa nuh igual a 0 especificado en la subcláusula 8.1.1.

**[0156]** Es un requisito de conformidad del flujo de bits que se siga la restricción siguiente:

***Si la imagen actual es una imagen IRAP con nuh\_layer\_id mayor que 0, entonces para cada capa de referencia directa de la capa actual con el valor de ID de capa directRefLayerId, existe al menos una imagen IRAP en la CVS con un valor nuh\_layer\_id igual a directRefLayerId y con un orden de descodificación que precede a la imagen actual.***

**[0157]** Por lo tanto, en al menos algunos ejemplos, un descodificador de vídeo (u otro dispositivo) puede realizar una prueba de conformidad de flujo de bits para determinar si el flujo de bits de datos de vídeo cumple con una especificación de codificación de vídeo, en el que es un requisito de la conformidad de flujo de bits que si una la imagen actual es una imagen IRAP con un identificador de capa mayor que 0, entonces, para cada capa de referencia directa de una capa actual, existe al menos una imagen IRAP en una CVS con un identificador de capa igual a un identificador de capa de la capa actual y con un orden de descodificación que precede a la imagen actual. Además, un dispositivo (por ejemplo, el codificador de vídeo 20 u otro dispositivo) puede generar un flujo de bits que incluye un VPS que incluye un elemento sintáctico (por ejemplo, un indicador) a partir del cual un descodificador de vídeo puede obtener el NoClrasPicPresentFlag.

**[0158]** Además, de acuerdo con el primer conjunto de ejemplos de las técnicas de esta divulgación, se pueden realizar los siguientes cambios en la semántica de cabecera de unidad NAL que se proporciona en el borrador de trabajo de HEVC 10.

#### **F.7.4.2.2 Semántica de cabecera de unidad NAL**

**[0159]** Las especificaciones en la subcláusula 7.4.2.2 se aplican con las siguientes modificaciones y adiciones.

**[0160]** **nal\_unit\_type** especifica el tipo de estructura de datos RBSP contenida en la unidad NAL como se especifica en la Tabla 7-1.

**[0161]** Las unidades NAL que tienen `nal_unit_type` en el rango de UNSPEC48..UNSPEC63, inclusive, para las cuales no se especifica la semántica, no afectarán el proceso de descodificación especificado en esta memoria descriptiva.

5 NOTA 1 - Los tipos de unidades NAL en el rango de UNSPEC48..UNSPEC63 se pueden usar según lo determine la aplicación. En esta memoria descriptiva no se especifica ningún proceso de descodificación para estos valores de `nal_unit_type`. Dado que diferentes aplicaciones podrían usar estos tipos de unidades NAL para diferentes propósitos, se debe tener especial cuidado en el diseño de los codificadores que generan unidades NAL con estos valores `nal_unit_type`, y en el diseño de descodificadores que interpretan el contenido de las unidades NAL con estos valores `nal_unit_type`.

10 **[0162]** Para fines distintos a la determinación de la cantidad de datos en las unidades de descodificación del flujo de bits (como se especifica en el Anexo C), los descodificadores ignorarán (eliminarán del flujo de bits y descartarán) el contenido de todas las unidades NAL que usan valores reservados de `nal_unit_type`.

15 NOTA 2 - Este requisito permite la definición futura de extensiones compatibles con esta memoria descriptiva.

**Tabla 7-1 - Códigos de tipo de unidad NAL y clases de tipo de unidad NAL**

| <code>nal_unit_type</code> | Nombre de <code>nal_unit_type</code> | Contenido de la unidad NAL y estructura sintáctica RBSP  | Clase de tipo de unidad de NAL |
|----------------------------|--------------------------------------|--|--------------------------------|
| 0                          | TRAIL_N                              | Segmento de fragmento codificado de una imagen posterior no TSA ni STSA                        | VCL                            |
| 1                          | TRAIL_R                              | <code>slice_segment_layer_rbsp( )</code>   |                                |
| 2                          | TSA_N                                | Segmento de fragmento codificado de una imagen TSA   | VCL                            |
| 3                          | TSA_R                                | <code>slice_segment_layer_rbsp( )</code>   |                                |
| 4                          | STSA_N                               | Segmento de fragmento codificado de una imagen STSA  | VCL                            |
| 5                          | STSA_R                               | <code>slice_segment_layer_rbsp( )</code>   |                                |
| 6                          | RADL_N                               | Segmento de fragmento codificado de una imagen RADL  | VCL                            |
| 7                          | RADL_R                               | <code>slice_segment_layer_rbsp( )</code>   |                                |
| 8                          | RASL_N                               | Segmento de fragmento codificado de una imagen RASL  | VCL                            |
| 9                          | RASL_R                               | <code>slice_segment_layer_rbsp( )</code>   |                                |
| 10                         | <u>CL_RAS_N</u>                      | <u>Segmento de fragmento codificado de una imagen CL-RAS</u>                                   | VCL                            |
| 11                         | <u>CL_RAS_R</u>                      | <u><code>slice_segment_layer_rbsp( )</code></u>  |                                |
| 12                         | RSV_VCL_N12                          | Sub-capa no-IRAP reservada   | VCL                            |
| 14                         | RSV_VCL_N14                          | tipos de unidades VCL NAL no de referencia   |                                |
| 13                         | RSV_VCL_R13                          | Sub-capa no-IRAP reservada   | VCL                            |
| 15                         | RSV_VCL_R15                          | tipos de unidades VCL NAL de referencia  |                                |
| 16                         | BLA_W_LP                             | Segmento de fragmento codificado de un BLA   | VCL                            |
| 17                         | BLA_W_RADL                           | imagen   |                                |
| 18                         | BLA_N_LP                             | <code>slice_segment_layer_rbsp( )</code>   |                                |
| 19                         | IDR_W_RADL                           | Segmento de fragmento codificado de una imagen IDR   | VCL                            |
| 20                         | IDR_N_LP                             |  |                                |
| <u>22</u>                  | <u>BL_IDR_W_RADL</u>                 | <code>slice_segment_layer_rbsp( )</code>   |                                |
| <u>23</u>                  | <u>BL_IDR_N_LP</u>                   |  |                                |
| 21                         | CRA_NUT                              | Segmento de fragmento codificado de una imagen CRA<br><code>slice_segment_layer_rbsp( )</code> | VCL                            |

| nal_unit_type | Nombre de nal_unit_type    | Contenido de la unidad NAL y estructura sintáctica RBSP             | Clase de tipo de unidad de NAL |
|---------------|----------------------------|---|--------------------------------|
| 24..31        | RSV_VCL24..<br>RSV_VCL31   | Tipos de unidades VCL NAL no IRAP reservadas                        | VCL                            |
| 32            | VPS_NUT                    | Conjunto de parámetros de vídeo<br>video_parameter_set_rbsp( )      | no VCL                         |
| 33            | SPS_NUT                    | Conjunto de parámetros de secuencia<br>seq_parameter_set_rbsp( )    | no VCL                         |
| 34            | PPS_NUT                    | Conjunto de parámetros de imagen<br>pic_parameter_set_rbsp( )       | no VCL                         |
| 35            | AUD_NUT                    | Delimitador de la unidad de acceso<br>access_unit_delimiter_rbsp( ) | no VCL                         |
| 36            | EOS_NUT                    | Fin de la secuencia end_of_seq_rbsp( )                              | no VCL                         |
| 37            | EOB_NUT                    | Fin del flujo de bits end_of_bitstream_rbsp( )                      | no VCL                         |
| 38            | FD_NUT                     | Datos de relleno filler_data_rbsp( )                                | no VCL                         |
| 39            | PREFIX_SEI_NUT             | Información de mejora suplementaria                                 | no VCL                         |
| 40            | SUFFIX_SEI_NUT             | sei_rbsp()  |                                |
| 41..47        | RSV_NVCL41..<br>RSV_NVCL47 | Reservado   | no VCL                         |
| 48..63        | UNSPEC48..<br>UNSPEC63     | Sin especificar   | no VCL                         |

NOTA 3 - Una imagen CRA puede tener imágenes RASL o RADL asociadas presentes en el flujo de bits.

5 NOTA 4 - Una imagen BLA que tenga nal\_unit\_type igual a BLA\_W\_LP puede tener imágenes RASL o RADL asociadas presentes en el flujo de bits. Una imagen BLA que tiene nal\_unit\_type igual a BLA\_W\_DLP no tiene imágenes RASL asociadas presentes en el flujo de bits, pero puede tener imágenes RADL asociadas en el flujo de bits. Una imagen BLA que tiene nal\_unit\_type igual a BLA\_N\_LP no tiene imágenes principales asociadas presentes en el flujo de bits.

10 NOTA 5 - Una imagen IDR que tiene nal\_unit\_type igual a IDR\_N\_LP o BL\_IDR\_N\_LP no tiene imágenes principales asociadas presentes en el flujo de bits. Una imagen IDR que tiene nal\_unit\_type igual a IDR\_W\_RADL o BL\_IDR\_W\_RADL no tiene imágenes RASL asociadas presentes en el flujo de bits, pero puede tener imágenes RADL asociadas en el flujo de bits.

15 NOTA 6 - No se incluye una imagen de subcapa sin referencia en RefPicSetStCurrBefore, RefPicSetStCurrAfter ni RefPicSetLtCurr de ninguna imagen con el mismo valor de TemporalId, y se puede descartar sin afectar la decodificación de otras imágenes con el mismo valor de TemporalId.

20 **[0163]** Todas las unidades NAL del segmento de fragmento codificado con un valor particular de nuh\_layer\_id en una unidad de acceso tendrán el mismo valor de nal\_unit\_type. También se hace referencia a una imagen que tiene un nal\_unit\_type igual al nal\_unit\_type de las unidades NAL del segmento de fragmento codificado de la imagen.

25 **[0164]** Si una imagen tiene nal\_unit\_type igual a TRAIL\_N, TSA\_N, STSA\_N, RADL\_N, RASL\_N, CL\_RAS\_N, RSV\_VCL\_N12, o RSV\_VCL\_N14, la imagen es una imagen de subcapa sin referencia. De lo contrario, la imagen es una imagen de referencia de subcapa.

**[0165]** Si la unidad de acceso actual es una unidad de acceso IRAP inicial, se aplica lo siguiente:

30 - Si cada imagen en la unidad de acceso actual con nuh\_layer\_id igual a RefLayerId[LayerIdInVps[currLayerId]][i] es una imagen IRAP, donde i está en el rango de 0 a

NumDirectRefLayers[LayerIdInVps[currLayerId]] - 1, inclusive, y currLayerId es el ID de capa nuh de la imagen actual, la imagen actual debe ser una imagen IRAP o una imagen CL-RAS con un tipo de unidad nal igual a CL\_RAS R.

35

- De lo contrario, la imagen actual será una imagen CL-RAS con nal\_unit\_type igual a CL\_RAS\_R. De lo contrario, se aplica lo siguiente:
- Si la imagen anterior en orden de decodificación en la misma capa que la imagen actual es una imagen CL-RAS, la imagen actual debe ser una imagen CL-RAS o una imagen IRAP.
- De lo contrario, la imagen actual no será una imagen CL-RAS.

[0166] Dentro de cada capa, cada imagen, aparte de la primera imagen en el flujo de bits en orden de decodificación, se considera que está asociada con la imagen IRAP anterior en orden de decodificación.

[0167] Cuando una imagen es una imagen principal, debe ser una imagen RADL o RASL.

[0168] Cuando una imagen es una imagen posterior, no debe ser una imagen RADL o RASL.

[0169] Cuando una imagen es una imagen principal, precederá, en el orden de decodificación, a todas las imágenes posteriores que están asociadas con la misma imagen IRAP.

[0170] Ninguna imagen RASL estará presente en el flujo de bits asociado con una imagen BLA que tenga nal\_unit\_type igual a BLA\_W\_RADL o BLA\_N\_LP. Ninguna imagen RASL estará presente en el flujo de bits que esté asociado con una imagen IDR.

[0171] Ninguna imagen RADL estará presente en el flujo de bits que esté asociado con una imagen BLA que tenga nal\_unit\_type igual a BLA\_N\_LP o que esté asociada con una imagen IDR que tenga nal\_unit\_type igual a IDR\_N\_LP o BL\_IDR\_N\_LP.

NOTA 7 - Es posible realizar un acceso aleatorio en la posición de una unidad de acceso IRAP descartando todas las unidades de acceso antes de la unidad de acceso IRAP (y decodificar correctamente la imagen IRAP y todas las imágenes posteriores no RASL en el orden de decodificación), siempre cada conjunto de parámetros esté disponible (ya sea en el flujo de bits o por medios externos no especificados en esta memoria descriptiva) cuando sea necesario activarlo.

[0172] Cualquier imagen que tenga PicOutputFlag igual a 1 que precede a una imagen IRAP en orden de decodificación deberá preceder a la imagen IRAP en orden de salida y deberá preceder a cualquier imagen RADL asociada con la imagen IRAP en orden de salida.

[0173] Cualquier imagen RASL asociada a una imagen CRA o BLA precederá a cualquier imagen RADL asociada a la imagen CRA o BLA en orden de salida.

[0174] Cualquier imagen RASL asociada con una imagen CRA deberá seguir, en orden de salida, a cualquier imagen IRAP que preceda a la imagen CRA en orden de decodificación.

[0175] Cuando sps\_temporal\_id\_nesting\_flag es igual a 1 y TemporalId es mayor que 0, el nal\_unit\_type será igual a TSA\_R, TSA\_N, RADL\_R, RADL\_N, RASL\_R o RASL\_N.

[0176] nuh\_layer\_id especifica el identificador de la capa.

[0177] Cuando nal\_unit\_type es igual a CL\_RAS\_R, CL\_RAS\_N, BL\_IDR\_W\_RADL o BL\_IDR\_N\_LP, el valor de nuh\_layer\_id será mayor que 0.

[0178] Además, en el primer conjunto de ejemplos de las técnicas de esta divulgación, se pueden hacer los siguientes cambios en la activación de VPS, SPS y PPS del Borrador de Trabajo de HEVC 10.

#### F.7.4.2.4.2 Orden de las RBSP de VPS, SPS y PPS y su activación

[0179] Las especificaciones en la subcláusula 7.4.2.4.2 se aplican con las siguientes adiciones. Cada RBSP de PPS se considera inicialmente no activa para cualquier capa con nuh\_layer\_id mayor que 0 al inicio de la operación del proceso de decodificación. A lo sumo, una RBSP de PPS se considera activa para cada valor nuh\_layer\_id distinto de cero en cualquier momento dado durante la operación del proceso de decodificación, y la activación de cualquier RBSP de PPS particular para un valor nuh\_layer\_id particular no cero da como resultado la desactivación de los valores la RBSP de PPS previamente activa para ese valor nuh\_layer\_id distinto de cero (si existe).

[0180] Cuando una RBSP de PPS (con un valor particular de pps\_pic\_parameter\_set\_id) no está activa para un valor de nuh\_layer\_id y se hace referencia a la misma mediante una unidad NAL de segmento de fragmento codificado (usando un valor de slice\_pic\_parameter\_set\_id igual al pps\_pic\_parameter\_set\_id), se activa para ese valor de ID de capa nuh. Esta RBSP de PPS se llama la RBSP de PPS de capa activa para ese valor nuh\_layer\_id hasta que es

desactivada por la activación de otra RBSP de PPS para la misma capa. Una RBSP de PPS, con ese valor particular de pps\_pic\_parameter\_set\_id, deberá estar disponible para el proceso de descodificación antes de su activación, incluida en al menos una unidad de acceso con TemporalId menor o igual que el TemporalId de la unidad PPS NAL o proporcionada por medios externos. El valor nuh\_layer\_id de la unidad NAL que contiene la RBSP de PPS que se activa para nuh\_layer\_id igual a nuhLayerIdA será menor o igual que nuhLayerIdA. El mismo RBSP de PPS puede ser el PPS de capa activa para más de un valor de nuh\_layer\_id.

**[0181]** Cualquier unidad PPS NAL que contenga el valor de pps\_pic\_parameter\_set\_id para la RBSP de PPS de capa activa para una imagen codificada deberá tener el mismo contenido que el de la RBSP de PPS de capa activa para la imagen codificada, a menos que siga a la última unidad VCL NAL de la imagen codificada y preceda a la primera unidad VCL NAL de otra imagen codificada.

**[0182]** Cada RBSP de SPS inicialmente se considera no activa para cualquier capa con nuh\_layer\_id mayor que 0 al inicio de la operación del proceso de descodificación. A lo sumo, una RBSP de SPS se considera activa para cada valor nuh\_layer\_id distinto de cero en cualquier momento dado durante el funcionamiento del proceso de descodificación, y la activación de cualquier RBSP de SPS particular para un valor nuh\_layer\_id particular no cero da como resultado la desactivación de la RBSP de SPS previamente activa para ese valor nuh\_layer\_id distinto de cero (si existe).

**[0183]** Cuando una RBSP de SPS (con un valor particular de sps\_seq\_parameter\_set\_id) aún no está activa para un valor de ID de capa nuh y se hace referencia a la misma mediante la activación de una RBSP de PPS para ese valor de ID de capa nuh (en el que pps\_seq\_parameter\_set\_id es igual al sps\_seq\_parameter\_set\_id value), se activa para ese valor nuh\_layer\_id. Esta RBSP de SPS se llama la RBSP de SPS de capa activa para ese valor de nuh\_layer\_id hasta que es desactivada por la activación de otra RBSP de SPS para la misma capa. Una RBSP de SPS, con ese valor particular de sps\_seq\_parameter\_set\_id, estará disponible para el proceso de descodificación antes de su activación, e incluida en al menos una unidad de acceso con TemporalId igual a 0, o proporcionada a través de medios externos. Una RBSP de SPS activada para un valor nuh\_layer\_id particular permanecerá activa para una secuencia de imágenes en orden de descodificación con ese valor nuh\_layer\_id a partir de una imagen que tiene ese valor nuh\_layer\_id, ya sea en una unidad de acceso IRAP inicial o la primera unidad de acceso después de la unidad de acceso IRAP inicial que contiene una imagen con valor nuh\_layer\_id, inclusive, hasta que la imagen siguiente tenga ese valor nuh\_layer\_id en una unidad de acceso IRAP inicial, exclusivo, o el final de la CVS, lo que ocurra primero. El valor nuh\_layer\_id de la unidad NAL que contiene la RBSP de SPS que se activa para nuh\_layer\_id igual a nuhLayerIdA será menor o igual que nuhLayerIdA. El mismo RBSP de SPS puede ser el SPS de capa activa para más de un valor de nuh\_layer\_id.

**[0184]** Cualquier unidad SPS NAL que contiene el valor de sps\_seq\_parameter\_set\_id para la RBSP de SPS de capa activa tendrá el mismo contenido que el de la RBSP de SPS de capa activa a menos que siga a la última imagen codificada para la cual se requiera que el SPS de capa activa esté activa y preceda a la primera unidad NAL que active un SPS del mismo valor de seq\_parameter\_set\_id.

**[0185]** Durante el funcionamiento del proceso de descodificación para unidades VCL NAL con un valor nuh\_layer\_id distinto de cero, los valores de los parámetros de la SPS de capa activa para ese valor nuh\_layer\_id no cero, y la PPS RBSP de capa activa para ese valor nuh\_layer\_id no cero, se consideran vigentes.

**[0186]** En la primera serie de ejemplos de las técnicas de esta divulgación, se pueden realizar los siguientes cambios en el proceso de descodificación de Borrador de trabajo de HEVC 10 para iniciar la descodificación de una imagen codificada con nuh\_layer\_id mayor que 0.

#### **F.8.1.1 Proceso de descodificación para iniciar la descodificación de una imagen codificada con nuh\_layer\_id mayor que 0**

**[0187]** Cada imagen a la que se hace referencia en esta subcláusula es una imagen completa codificada.

**[0188]** El proceso de descodificación funciona de la forma siguiente para la imagen actual CurrPic:

1. La descodificación de las unidades NAL se especifica en la subcláusula 4.
2. Los procesos en la subcláusula 8.3 especifican los siguientes procesos de descodificación utilizando elementos sintácticos en la capa de segmento de fragmento y superiores:
  - Las variables y funciones relacionadas con el recuento de orden de imágenes se obtienen en la subcláusula 8.3.1. Esto debe invocarse solo para el primer fragmento de una imagen. Es un requisito de conformidad del flujo de bits que la diferencia entre los valores PicOrderCnVal de las dos imágenes con un valor particular de nuh\_layer\_id dentro de la misma CVS sea la misma para cualquier valor de ID de capa de nuh.

- El proceso de descodificación para RPS en la subcláusula 8.3.2 se invoca para imágenes con nuh\_layer\_id igual a la de CurrPic, en el que las imágenes de referencia pueden marcarse como "no utilizadas como referencia" o "usadas para referencia a largo plazo". Esto debe invocarse solo para el primer fragmento de una imagen.

- Cuando NoCInrasOutputFlag es igual a 1 y la unidad de acceso que contiene CurrPic es una unidad de acceso IRAP inicial, se invoca el proceso de descodificación para generar imágenes de referencia no disponibles especificadas en la **Sección 5.7**, que solo debe invocarse para el primer segmento de fragmento de una imagen.

[0189] Además, en el primer conjunto de ejemplos de las técnicas de esta divulgación, se pueden hacer los siguientes cambios en el proceso de descodificación del Borrador de trabajo de HEVC 10 para poner fin a la descodificación de una imagen codificada con nuh\_layer\_id mayor que 0.

#### 15 **F.8.1.1 Proceso de descodificación para finalizar la descodificación de una imagen codificada con nuh\_layer\_id mayor que 0**

[0190] PicOutputFlag se establece de la forma siguiente:

- Si la imagen actual es una imagen RASL y NoRasIOutputFlag de la imagen IRAP asociada es igual a 1, PicOutputFlag se establece igual a 0.
- De lo contrario, si la imagen actual es una imagen CL-RAS y NoCInrasOutputFlag es igual a 1, PicOutputFlag se establece igual a 0.
- De lo contrario, PicOutputFlag se establece igual a pic\_output\_flag.

[0191] Se aplica lo siguiente:

- La imagen descodificada se marca como "utilizada para referencia a corto plazo".
- Cuando TemporalId es igual a HighestTid, el proceso de marcado para las imágenes de subcapa no de referencia no es necesario para la predicción entre capas especificada en la subcláusula F.8.1.2.1 con latestDeclayerId igual a la ID de capa nuh como entrada.

[0192] Además, en el primer conjunto de ejemplos de las técnicas de esta divulgación, se pueden realizar los siguientes cambios en la generación de imágenes de referencia no disponibles en el borrador de trabajo de HEVC 10.

#### 40 **8.3.3.1 Proceso de descodificación general para generar imágenes de referencia no disponibles**

[0193] Este proceso se invoca una vez por imagen codificada cuando la imagen actual es una imagen BLA o es una imagen CRA con NoRasIOutputFlag igual a 1.

NOTA - Este proceso se especifica principalmente solo para la especificación de restricciones de sintaxis para imágenes RASL. La especificación completa del proceso de descodificación para imágenes RASL asociadas con una imagen IRAP que tiene NoRasIOutputFlag igual a 1 se incluye en el presente documento solo con el propósito de especificar restricciones en el contenido sintáctico permitido de dichas imágenes RASL. Durante el proceso de descodificación, cualquier imagen RASL asociada con una imagen IRAP que tenga NoRasIOutputFlag igual a 1 puede ignorarse, ya que estas imágenes no están especificadas para la salida y no tienen ningún efecto en el proceso de descodificación de ninguna otra imagen que esté especificada para la salida. Sin embargo, en las operaciones de HRD, tal como se especifica en el Anexo C, es posible que se deba tener en cuenta al obtener los tiempos de llegada y eliminación de CPB.

[0194] Cuando se invoca este proceso, se aplica lo siguiente:

- Para cada RefPicSetStFoll[i], con i en el rango de 0 a NumPocStFoll - 1, inclusive, que es igual a "sin imagen de referencia", se genera una imagen como se especifica en la subcláusula 8.3.3.2, y se aplica lo siguiente:
  - El valor de PicOrderCntVal para la imagen generada se establece igual a PocStFoll[i].
  - El valor de PicOutputFlag para la imagen generada se establece igual a 0.
  - La imagen generada está marcada como "utilizada para referencia a corto plazo".
- RefPicSetStFoll[i] se establece para ser la imagen de referencia generada.

- El valor de ID de nuh\_layer para la imagen generada se establece igual a nuh\_layer\_id.
- Para cada RefPicSetLtFoll[i], con i en el rango de 0 a NumPocLtFoll - 1, inclusive, que es igual a "sin imagen de referencia", se genera una imagen como se especifica en la subcláusula 8.3.3.2, y se aplica lo siguiente:
  - El valor de PicOrderCntVal para la imagen generada se establece igual a PocLtFoll[i].
  - Se infiere que el valor de slice\_pic\_order\_cnt\_lsb para la imagen generada es igual a (PocLtFoll[i] & (MaxPicOrderCntLsb - 1)).
  - El valor de PicOutputFlag para la imagen generada se establece igual a 0.
  - La imagen generada se marca como "utilizada para referencia a largo plazo".
  - RefPicSetLtFoll[i] se establece para ser la imagen de referencia generada.
  - El valor de nuh\_layer\_id para la imagen generada se establece igual a nuh\_layer\_id.

**[0195]** Además, en el primer conjunto de ejemplos de las técnicas de esta divulgación, la generación de imágenes de referencia no disponibles para imágenes CL-RAS en el Borrador de Trabajo de HEVC 10 se puede cambiar de la forma siguiente.

**[0196]** Este proceso se invoca una vez por imagen codificada cuando NoClrasOutputFlag es igual a 1 y la imagen actual es una imagen CL-RAS en una unidad de acceso IRAP inicial.

NOTA - Este proceso se especifica principalmente solo para la especificación de restricciones de sintaxis para imágenes CL-RAS. La especificación completa del proceso de descodificación para imágenes CL-RAS cuando NoClrasOutputFlag es igual a 1 se incluye en el presente documento solo con el propósito de especificar restricciones en el contenido sintáctico permitido de dichas imágenes CL-RAS. Durante el proceso de descodificación, cuando NoClrasOutputFlag es igual a 1, cualquier imagen CL-RAS puede ignorarse, ya que estas imágenes no se especifican para la salida y no tienen efecto en el proceso de descodificación de ninguna otra imagen que se especifique para la salida. Sin embargo, en las operaciones de DRH que se especifican en el Anexo C, es posible que las imágenes CL-RAS deban tomarse en consideración al obtener los tiempos de llegada y eliminación de CPB.

Cuando se invoca este proceso, se aplica lo siguiente:

- ***Para cada RefPicSetStCurrBefore[i], con i en el rango de 0 a NumPocStCurrBefore -1, inclusive, que es igual a "imagen sin referencia", se genera una imagen como se especifica en la subcláusula 8.3.3.2, y se aplica lo siguiente:***
  - ***El valor de PicOrderCntVal para la imagen generada se establece igual a PocStCurrBefore[i].***
  - ***El valor de PicOutputFlag para la imagen generada se establece igual a 0.***
  - ***La imagen generada está marcada como "utilizada para referencia a corto plazo".***
  - ***RefPicSetStCurrBefore[i] se establece para que sea la imagen de referencia generada.***
  - ***El valor de nuh\_layer\_id para la imagen generada se establece igual a nuh\_layer\_id.***
- ***Para cada RefPicSetStCurrAfter[i], con i en el rango de 0 a NumPocStCurrAfter -1, inclusive, que es igual a "imagen sin referencia", se genera una imagen como se especifica en la subcláusula 8.3.3.2, y se aplica lo siguiente:***
  - ***El valor de PicOrderCntVal para la imagen generada se establece igual a PocStCurrAfter[i].***
  - ***El valor de PicOutputFlag para la imagen generada se establece igual a 0.***
  - ***La imagen generada está marcada como "utilizada para referencia a corto plazo".***
  - ***RefPicSetStCurrAfter[i] se establece para que sea la imagen de referencia generada.***
  - ***El valor de nuh\_layer\_id para la imagen generada se establece igual a nuh\_layer\_id.***
- Para cada RefPicSetStFoll[i], con i en el rango de 0 a NumPocStFoll - 1, inclusive, que es igual a "sin imagen de referencia", se genera una imagen como se especifica en la subcláusula 8.3.3.2, y se aplica lo siguiente:

- El valor de PicOrderCntVal para la imagen generada se establece igual a PocStFoll[i].
- El valor de PicOutputFlag para la imagen generada se establece igual a 0.
- La imagen generada está marcada como "utilizada para referencia a corto plazo".
- RefPicSetStFoll[i] se establece para ser la imagen de referencia generada.
- El valor de ID de capa nuh para la imagen generada se establece igual a nuh\_layer\_id.
- **Para cada RefPicSetLtCurr[i], con i en el rango de 0 a NumPocLtCurr - 1, inclusive, que es igual a "imagen sin referencia", se genera una imagen como se especifica en la subcláusula 8.3.3.2, y se aplica lo siguiente:**
  - **El valor de PicOrderCntVal para la imagen generada se establece igual a PocLtCurr[i].**
  - **Se infiere que el valor de slice\_pic\_order\_cnt\_lsb para la imagen generada es igual a (PocLtCurr[i] & (MaxPicOrderCntLsb - 1)).**
  - **El valor de PicOutputFlag para la imagen generada se establece igual a 0.**
  - **La imagen generada se marca como "utilizada para referencia a largo plazo".**
  - **RefPicSetLtCurr[i] se establece para que sea la imagen de referencia generada.**
  - **El valor de nuh\_layer\_id para la imagen generada se establece igual a nuh\_layer\_id.**
- Para cada RefPicSetLtFoll[i], con i en el rango de 0 a NumPocLtFoll - 1, inclusive, que es igual a "sin imagen de referencia", se genera una imagen como se especifica en la subcláusula 8.3.3.2, y se aplica lo siguiente:
  - El valor de PicOrderCntVal para la imagen generada se establece igual a PocLtFoll[i].
  - Se infiere que el valor de slice\_pic\_order\_cnt\_lsb para la imagen generada es igual a (PocLtFoll[i] & (MaxPicOrderCntLsb - 1)).
  - El valor de PicOutputFlag para la imagen generada se establece igual a 0.
  - La imagen generada se marca como "utilizada para referencia a largo plazo".
  - RefPicSetLtFoll[i] se establece para ser la imagen de referencia generada.
  - El valor de nuh\_layer\_id para la imagen generada se establece igual a nuh\_layer\_id.

[0197] Por lo tanto, en algunos ejemplos, el descodificador de vídeo 30 puede generar imágenes de referencia no disponibles durante la descodificación de las imágenes CL-RAS para un primer, segundo, tercero, cuarto, y quinto subconjunto de imágenes de referencia. Cada imagen de referencia en el primer subconjunto de imágenes de referencia (por ejemplo, RefPicSetStCurrBefore) es una imagen de referencia a corto plazo que aparece antes de la imagen actual en orden de salida, cada imagen de referencia en el segundo subconjunto de imágenes de referencia (por ejemplo, RefPicSetStCurrAfter) es una imagen de referencia a corto plazo que aparece después de la imagen actual en orden de salida, cada imagen de referencia en el tercer subconjunto de imágenes de referencia (por ejemplo, RefPicSetStFoll) es una imagen de referencia a corto plazo que no se usa para la predicción inter de la imagen actual, cada imagen de referencia en el cuarto subconjunto de imágenes de referencia (por ejemplo, RefPicSetLtCurr) es una imagen de referencia a largo plazo que se puede usar para la predicción inter de la imagen actual, y cada imagen de referencia en el quinto subconjunto de imágenes de referencia (por ejemplo, RefPicSetLtFoll) es una imagen de referencia a largo plazo que no se usa para la predicción. de la imagen actual.

[0198] Además, en algunos ejemplos, descodificador de vídeo 30 puede, para cada posición respectiva en una matriz para el primer subconjunto de imágenes de referencia (por ejemplo, RefPicSetStBefore[i]), si una entrada en la posición respectiva es igual a "ninguna imagen de referencia", generar una imagen de referencia para la cual se aplica lo siguiente: un valor de recuento de orden de imágenes (POC) para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un primer conjunto de valores POC (por ejemplo, PocStCurrBefore), un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0, la imagen de referencia generada se marca como utilizada para referencia a corto plazo, la posición respectiva en la matriz para el subconjunto de imágenes de referencia se establece en la imagen de referencia generada y el identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen actual (por ejemplo, ID de capa nuh). Además, para cada posición respectiva en una matriz para el segundo subconjunto de imágenes de referencia (por

ejemplo, RefPicSetStCurrAfter[i]), si una entrada en la posición respectiva es igual a "ninguna imagen de referencia", el codificador de vídeo puede generar una imagen de referencia para la cual se aplica lo siguiente: un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un segundo conjunto de valores POC (p. ej., PocStCurrAfter), un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0, la imagen de referencia generada se marca como utilizada para referencia a corto plazo, la posición respectiva en la matriz para el segundo subconjunto de imágenes de referencia se establece en la imagen de referencia generada y el identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen actual. Además, para cada posición respectiva en una matriz para el tercer subconjunto de imágenes de referencia (por ejemplo, RefPicSetStFoll[i]), si una entrada en la posición respectiva es igual a "ninguna imagen de referencia", el descodificador de vídeo puede generar una imagen de referencia para la cual se aplica lo siguiente: un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un tercer conjunto de valores POC (por ejemplo, PocStFoll), un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0, la imagen de referencia generada se marca como utilizada para referencia a corto plazo, la posición respectiva en la matriz para el tercer subconjunto de imágenes de referencia se establece en el identificador de capa de la imagen actual. Además, para cada posición respectiva en una matriz para el cuarto subconjunto de imágenes de referencia (por ejemplo, RefPicSetLtCurr[i]), si una entrada en la posición respectiva es igual a "ninguna imagen de referencia", el descodificador de vídeo puede generar una imagen de referencia para la cual se aplica lo siguiente: un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un cuarto conjunto de valores POC (p. ej., PocLtCurr), un valor de un elemento sintáctico que especifica los bits menos significativos de un valor POC (por ejemplo, slice\_pic\_order\_cnt\_lsb) para la imagen generada se infiere que es igual al resultado de una operación "y" a nivel de bits del valor POC en la posición respectiva en el cuarto conjunto de valores POC y un valor resultante de una variable que indica el valor máximo que los bits menos significativos de POC pueden tener (por ejemplo, MaxPicOrderCntLsb) menos 1, un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0, la imagen de referencia generada se marca como utilizada para referencia a largo plazo, la posición respectiva en la matriz para el cuarto subconjunto de imágenes de referencia se establece en la imagen de referencia generada, y el identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen actual. Además, para cada posición respectiva en una matriz para el quinto subconjunto de imágenes de referencia (por ejemplo, PocSetLtFoll[i]), si una entrada en la posición respectiva es igual a "ninguna imagen de referencia", el descodificador de vídeo puede generar una imagen de referencia para la cual se aplica lo siguiente: un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un quinto conjunto de valores POC (por ejemplo, PocLtFoll), un valor de un elemento sintáctico que especifica los bits menos significativos de un valor POC (por ejemplo, slice\_pic\_order\_cnt\_lsb) para la imagen generada se infiere como igual a un resultado de una operación "y" a nivel de bits del valor POC en la posición respectiva en el quinto conjunto de valores POC y un valor resultante de una variable que indica el valor máximo que los bits menos significativos del valor POC pueden tener (por ejemplo, MaxPicOrderCntLsb) menos 1, un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0, la imagen de referencia generada se marca como utilizada para referencia a largo plazo, la posición respectiva en la matriz para el quinto subconjunto de imágenes de referencia se establece en la imagen de referencia generada, y el identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen actual.

**[0199]** Además, en el primer conjunto de ejemplos de las técnicas de esta divulgación, se pueden realizar los siguientes cambios en la sintaxis del mensaje SEI de período de memoria intermedia del Borrador de trabajo de HEVC 10.

45

|   | Descriptor  |
|---|-------------|
| buffering_period( payloadSize ) {                                       |             |
| <b>bp_seq_parameter_set_id</b>  | u(v)        |
| if( !sub_pic_hrd_params_present_flag ) {                                |             |
| <b>irap_cpb_params_present_flag</b>                                     | u(1)        |
| <i>irap_cross_layer_cpb_params_present_flag</i>                         | <i>u(1)</i> |
| }   |             |
| if( irap_cpb_params_present_flag ) {                                    |             |
| <b>cpb_delay_offset</b>   | u(v)        |
| <b>dpb_delay_offset</b>   | u(v)        |
| }   |             |
| <i>if( irap_cross_layer_cpb_params_present_flag ) {</i>                 |             |
| <i>cpb_cross_layer_delay_offset</i>                                     | <i>u(v)</i> |
| <i>dpb_cross_layer_delay_offset</i>                                     | <i>u(v)</i> |
| }   |             |
| <b>concatenation_flag</b>   | u(1)        |
| <b>au_cpb_removal_delay_delta_minus1</b>                                | u(v)        |
| if( NalHrdBpPresentFlag ) {   |             |
| for( i = 0; i <= CpbCnt; i++ ) {  |             |
| <b>nal_initial_cpb_removal_delay[ i ]</b>                               | u(v)        |
| <b>nal_initial_cpb_removal_offset[ i ]</b>                              | u(v)        |
| if( sub_pic_hrd_params_present_flag    irap_cpb_params_present_flag ) { |             |
| <b>nal_initial_alt_cpb_removal_delay[ i ]</b>                           | u(v)        |
| <b>nal_initial_alt_cpb_removal_offset[ i ]</b>                          | u(v)        |
| }   |             |
| <i>if( irap_cross_layer_cpb_params_present_flag ) {</i>                 |             |
| <i>nal_initial_alt_cross_layer_cpb_removal_delay[ i ]</i>               | <i>u(v)</i> |
| <i>nal_initial_alt_cross_layer_cpb_removal_offset[ i ]</i>              | <i>u(v)</i> |
| }   |             |
| }   |             |
| }   |             |
| if( VclHrdBpPresentFlag ) {   |             |

|  |      |
|--|------|
| for( i = 0; i <= CpbCnt; i++) {  |      |
| vcl_initial_cpb_removal_delay[ i ]                                     | u(v) |
| vcl_initial_cpb_removal_offset[ i ]                                    | u(v) |
| if( sub_pic_hrd_params_present_flag    irap_cpb_params_present_flag) { |      |
| vcl_initial_alt_cpb_removal_delay[ i ]                                 | u(v) |
| vcl_initial_alt_cpb_removal_offset[ i ]                                | u(v) |
| }  |      |
| if( irap_cross_layer_cpb_params_present_flag ) {                       |      |
| vcl_initial_alt_cross_layer_cpb_removal_delay[ i ]                     | u(v) |
| vcl_initial_alt_cross_layer_cpb_removal_offset[ i ]                    | u(v) |
| }  |      |
| }  |      |
| }  |      |
| }  |      |
| }  |      |

**[0200]** De forma alternativa, en algunos ejemplos, la señalización de irap\_cross\_layer\_cpb\_params\_present\_flag no está condicionada por el valor de sub\_pic\_hrd\_params\_present\_flag y se señala de forma independiente en el mensaje SEI de período de almacenamiento en memoria intermedia.

**[0201]** Además, de acuerdo con el primer conjunto de ejemplos de las técnicas de esta divulgación, se pueden realizar los siguientes cambios en la semántica de mensajes SEI de período de almacenamiento en memoria intermedia.

**irap\_cross\_layer\_cpb\_params\_present\_flag igual a 1 especifica la presencia de los elementos sintácticos initial\_alt\_cross\_layer\_cpb\_removal\_delay[i] e initial\_alt\_**

**cross\_layer\_cpb\_removal\_offset[i]. Cuando no está presente, se infiere que el valor de rap\_cross\_layer\_cpb\_params\_present\_flag es igual a 0. Cuando la imagen asociada no es una imagen CRA ni una imagen BLA, el valor de irap\_cross\_layer\_cpb\_params\_present\_flag será igual a 0.**

**NOTA 2 - No más de uno de los valores de sub\_pic\_hrd\_params\_present\_flag, irap\_cpb\_params\_present\_flag y irap\_cross\_layer\_cpb\_params\_present\_flag puede ser igual a 1.**

**La desviación cpb\_cross\_layer\_delay especifica una desviación a utilizar en la obtención de los tiempos de eliminación de la CPB nominales de las unidades de acceso que siguen, en orden de decodificación, a la unidad de acceso IRAP asociada con el mensaje SEI de período de almacenamiento en memoria intermedia cuando las unidades de imágenes CL-RAS asociadas con la unidad de acceso IRAP no estén presentes. El elemento sintáctico tiene una longitud en bits dada por au\_cpb\_removal\_delay\_length\_minus1 + 1. Cuando no está presente, se infiere que el valor de cpb\_cross\_layer\_delay\_offset es igual a 0.**

**dpb\_cross\_layer\_delay\_offset especifica una desviación que se utilizará en la obtención de los tiempos de salida de DPB de la unidad de acceso IRAP asociada con el mensaje SEI de período de almacenamiento en memoria intermedia cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP no están presentes. El elemento sintáctico tiene una longitud en bits dada por dpb\_output\_delay\_length\_minus1 + 1. Cuando no está presente, se infiere que el valor de dpb\_cross\_layer\_delay\_offset es igual a 0. Cuando la imagen actual no es la primera imagen en el flujo de bits en orden de decodificación, deje que prevNonDiscardablePic sea la imagen anterior en orden de decodificación con TemporalId igual a 0 que no es una imagen sin referencia de RASL, RADL o subcapa.**

**nal\_initial\_cpb\_removal\_delay[i] y nal\_initial\_alt\_cpb\_removal\_delay[i] especifican los retardos de eliminación de CPB iniciales alternativos y predeterminados, respectivamente, para la i-ésima CPB cuando se utilizan los parámetros NAL HRD. 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 o iguales a  $90000 * (CpbSize[i] \div BitRate[i])$ , el equivalente en tiempo del tamaño de la CPB en unidades de reloj de 90 kHz.

***nal\_initial\_alt\_cross\_layer\_cpb\_removal\_delay[i]*** especifica un retardo de eliminación de CPB inicial alternativa para la *i*-ésima CPB cuando los parámetros NAL HRD están en uso. El elemento sintáctico tiene una longitud en bits dada por ***initial\_cpb\_removal\_delay\_length\_minus1 + 1***, y está en unidades de un reloj de 90 kHz. El valor del elemento sintáctico no será igual a 0 y será inferior o igual a  $90000 * (CpbSize[i] \div BitRate[i])$ , el equivalente en tiempo del tamaño de la CPB en unidades de reloj de 90 kHz.

**nal\_initial\_cpb\_removal\_offset[i]** y **nal\_initial\_alt\_cpb\_removal\_offset[i]** especifican las desviaciones de eliminación de CPB inicial predeterminadas y alternativas, respectivamente, para la *i*-ésima CPB cuando se utilizan los parámetros NAL HRD. 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.

***nal\_initial\_alt\_cross\_layer\_cpb\_removal\_offset[i]*** especifica una desviación de eliminación de CPB inicial alternativa para la *i*-ésima CPB cuando los parámetros NAL HRD están en uso. El elemento sintáctico tiene una longitud en bits dada por ***initial\_cpb\_removal\_delay\_length\_minus1 + 1*** y está en unidades de un reloj de 90 kHz.

Durante toda la CVS, la suma de **nal\_initial\_cpb\_removal\_delay[i]** y **nal\_initial\_cpb\_removal\_offset[i]** será constante para cada valor de *i*, la suma de **nal\_initial\_alt\_cpb\_removal\_delay[i]** y **nal\_initial\_alt\_cpb\_removal\_offset[i]** será constante para cada valor de *i*, y la suma de ***nal\_initial\_alt\_cross\_layer\_cpb\_removal\_delay[i]*** y ***nal\_initial\_alt\_cross\_layer\_cpb\_removal\_offset[i]*** será constante para cada valor de *i*.

**vcl\_initial\_cpb\_removal\_delay[i]** y **vcl\_initial\_alt\_cpb\_removal\_delay[i]** especifican los retardos de eliminación de CPB iniciales predeterminados y alternativos, respectivamente, para la *i*-ésima CPB cuando los parámetros VCL HRD están en uso. 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 o iguales a  $90000 * (CpbSize[i] \div BitRate[i])$ , el equivalente en tiempo del tamaño de la CPB en unidades de reloj de 90 kHz.

***vcl\_initial\_alt\_cross\_layer\_cpb\_removal\_delay[i]*** especifica un retardo de eliminación de CPB inicial alternativa, respectivamente, para la *i*-ésima CPB cuando los parámetros VCL HRD están en uso. El elemento sintáctico tiene una longitud en bits dada por ***initial\_cpb\_removal\_delay\_length\_minus1 + 1***, y está en unidades de un reloj de 90 kHz. El valor del elemento sintáctico no será igual a 0 y será inferior o igual a  $90000 * (CpbSize[i] \div BitRate[i])$ , el equivalente en tiempo del tamaño de la CPB en unidades de reloj de 90 kHz.

**vcl\_initial\_cpb\_removal\_offset [i]** y **vcl\_initial\_alt\_cpb\_removal\_offset[i]** especifican las desviaciones de eliminación de CPB inicial predeterminadas y alternativas, respectivamente, para la *i*-ésima CPB cuando se utilizan los parámetros VCL HRD. 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.

***vcl\_initial\_alt\_cross\_layer\_cpb\_removal\_offset[i]*** especifica una desviación de eliminación de CPB inicial alternativa, respectivamente, para la *i*-ésima CPB cuando los parámetros VCL HRD están en uso. El elemento sintáctico tiene una longitud en bits dada por ***initial\_cpb\_removal\_delay\_length\_minus1 + 1*** y está en unidades de un reloj de 90 kHz.

Durante toda la CVS, la suma de **vcl\_initial\_cpb\_removal\_delay[i]** y **vcl\_initial\_cpb\_removal\_offset[i]** será constante para cada valor de *i*, la suma de **vcl\_initial\_alt\_cpb\_removal\_delay[i]** y **vcl\_initial\_alt\_cpb\_removal\_offset[i]** será constante para cada valor de *i*, y la suma de ***vcl\_initial\_alt\_cross\_layer\_cpb\_removal\_delay[i]*** y ***vcl\_initial\_alt\_cross\_layer\_cpb\_removal\_offset[i]*** será constante para cada valor de *i*.

NOTA 4 - Se recomienda que los codificadores no incluyan **irap\_cpb\_params\_present\_flag** igual a 1 en los mensajes SEI de período de almacenamiento en memoria intermedia asociados con una imagen CRA o BLA para la cual al menos una de sus imágenes RASL asociadas sigue una o más de sus imágenes RADL asociadas en orden de decodificación.

La semántica del resto de los elementos sintácticos en el mensaje SEI de período de almacenamiento en memoria intermedia sigue siendo la misma.

De forma alternativa, se señalan elementos sintácticos adicionales en la estructura sintáctica **hrd\_parameters ( )** que especifican las longitudes (en bits) de los elementos sintácticos ***cpb\_cross\_layer\_delay\_offset***, ***dpb\_cross\_layer\_delay\_offset***, ***nal\_initial\_alt\_cross\_layer\_cpb\_removal\_delay[i]***, ***nal\_initial***

*alt\_cross\_layer\_cpb\_removal\_offset[i], vcl\_initial alt\_cross\_layer\_cpb\_removal\_delay[i], e  
vcl\_initial\_alt\_cross\_layer\_cpb\_removal\_offset[i].*

5 **[0202]** Por lo tanto, en algunos ejemplos, el descodificador de vídeo 30 puede obtener, a partir de un mensaje SEI del  
flujo de bits de vídeo codificado, un primer elemento sintáctico (por ejemplo, irap\_cross\_layer\_cpb\_params\_present\_flag),  
que especifica si un segundo elemento sintáctico de retardo de eliminación de CPB alternativa (por ejemplo, nal\_initial\_alt\_cross\_layer\_cpb\_removal\_delay) y un tercer elemento  
10 sintáctico de desviación de eliminación de CPB alternativa (por ejemplo, nal\_initial\_alt\_cross\_layer\_cpb\_removal\_offset) están presentes en el mensaje SEI. El segundo elemento sintáctico  
de retardo de eliminación de CPB alternativa puede especificar un retardo de eliminación de CPB inicial alternativa  
para una CPB cuando los parámetros NAL HRD están en uso. El tercer elemento sintáctico de desviación de  
eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para la CPB cuando  
15 los parámetros NAL HRD están en uso. Además, el descodificador de vídeo 30 puede obtener, a partir del mensaje  
SEI, un elemento sintáctico de desviación de retardo de capa cruzada de CPB que especifica una desviación a usar  
en una obtención de los tiempos de eliminación de CPB nominales de las unidades de acceso que siguen, en orden  
de descodificación, a una unidad de acceso IRAP asociada con el mensaje SEI cuando las imágenes CL-RAS  
asociadas con la unidad de acceso IRAP no están presentes en el flujo de bits de datos de vídeo. El descodificador  
de vídeo 30 también puede obtener, a partir del mensaje SEI, un elemento sintáctico de retardo de retardo de capa  
20 cruzada de memoria intermedia de imágenes descodificadas (DPB) que especifica una desviación que se utilizará en  
una obtención de los tiempos de salida de DPB de las unidades de acceso IRAP asociadas con el mensaje SEI cuando  
las imágenes CL-RAS asociadas con la unidad de acceso IRAP no están presentes en el flujo de datos de vídeo.

25 **[0203]** Del mismo modo, en algunos ejemplos, un dispositivo (por ejemplo, el codificador de vídeo 20) genera un  
mensaje SEI que incluye un primer elemento sintáctico (por ejemplo, irap\_cross\_layer\_cpb\_params\_present\_flag) que  
especifica si un segundo elemento sintáctico de retardo de eliminación de CPB alternativa (por ejemplo,  
nal\_initial\_alt\_cross\_layer\_cpb\_removal\_delay) y un tercer elemento sintáctico de desviación de eliminación de CPB  
30 alternativa (por ejemplo, nal\_initial\_alt\_cross\_layer\_cpb\_removal\_offset) están presentes en el mensaje SEI. El  
segundo elemento sintáctico de retardo de eliminación de CPB alternativa especifica un retardo de eliminación de CPB  
inicial alternativa para una CPB cuando los parámetros NAL HRD están en uso. El tercer elemento sintáctico de  
desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para  
la CPB cuando los parámetros NAL HRD están en uso. El mensaje SEI incluye un elemento sintáctico de desviación  
de retardo de capa cruzada de CPB que especifica una desviación que se utilizará en una obtención de los tiempos  
de eliminación de CPB nominales de las unidades de acceso que siguen, en orden de descodificación, a una unidad  
de acceso IRAP asociada con el mensaje SEI cuando las imágenes CL-RAS asociados con la unidad de acceso IRAP  
35 no están presentes en el flujo de bits de datos de vídeo. El mensaje SEI incluye un elemento sintáctico de desviación  
de retardo de capa cruzada DPB que especifica una desviación que se utilizará en una obtención de los tiempos  
de salida DPB de la unidad de acceso IRAP asociada con el mensaje SEI cuando las imágenes CL-RAS asociadas con  
la unidad de acceso IRAP no están presentes en el flujo de bits de datos de vídeo.

40 **[0204]** Además, en algunos ejemplos, descodificador de vídeo 30 puede obtener, a partir de un mensaje SEI del flujo  
de bits de vídeo codificados, un primer elemento sintáctico (por ejemplo, irap\_cross\_layer\_cpb\_params\_present\_flag)  
que especifica si un segundo elemento sintáctico de retardo de eliminación de CPB alternativa (por ejemplo,  
nal\_initial\_alt\_cross\_layer\_cpb\_removal\_delay) y un tercer elemento sintáctico de desviación de eliminación de CPB  
45 alternativa (por ejemplo, nal\_initial\_alt\_cross\_layer\_cpb\_removal\_offset) están presentes en el mensaje SEI. El  
segundo elemento sintáctico de retardo de eliminación de CPB alternativa puede especificar un retardo de eliminación  
de CPB inicial alternativa para una CPB cuando los parámetros VCL HRD están en uso. El tercer elemento sintáctico  
de desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa  
para la CPB cuando los parámetros VCL HRD están en uso. Además, el descodificador de vídeo 30 puede obtener, a  
partir del mensaje SEI, un elemento sintáctico de desviación de retardo de capa cruzada de CPB que especifica una  
50 desviación a usar en una obtención de los tiempos de eliminación de CPB nominales de las unidades de acceso que  
siguen, en orden de descodificación, a una unidad de acceso IRAP asociada con el mensaje SEI cuando las imágenes  
CL-RAS asociadas con la unidad de acceso IRAP no están presentes en el flujo de bits de datos de vídeo. El  
descodificador de vídeo 30 también puede obtener, a partir del mensaje SEI, un elemento sintáctico de desviación de  
retardo de capa cruzada DPB que especifica una desviación que se usará en una obtención de los tiempos de salida  
55 DPB de las unidades de acceso IRAP asociadas con el mensaje SEI cuando las imágenes CLRAS asociadas con la  
unidad de acceso IRAP no está presentes en el flujo de bits de datos de vídeo.

60 **[0205]** Del mismo modo, en algunos ejemplos, un dispositivo (por ejemplo, el codificador de vídeo 20) genera un  
mensaje SEI que incluye un primer elemento sintáctico (por ejemplo, irap\_cross\_layer\_cpb\_params\_present\_flag) que  
especifica si un segundo elemento sintáctico de retardo de eliminación de CPB alternativa (por ejemplo,  
nal\_initial\_alt\_cross\_layer\_cpb\_removal\_delay) y un tercer elemento sintáctico de desviación de eliminación de CPB  
65 alternativa (por ejemplo, nal\_initial\_alt\_cross\_layer\_cpb\_removal\_offset) están presentes en el mensaje SEI. El  
segundo elemento sintáctico de retardo de eliminación de CPB alternativa especifica un retardo de eliminación de CPB  
inicial alternativa para una CPB cuando los parámetros VCL HRD están en uso. El tercer elemento sintáctico de  
desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para  
la CPB cuando los parámetros VCL HRD están en uso. El mensaje SEI incluye un elemento sintáctico de desviación

de retardo de capa cruzada de CPB que especifica una desviación que se utilizará en una obtención de los tiempos de eliminación de CPB nominales de las unidades de acceso que siguen, en orden de descodificación, a una unidad de acceso IRAP asociada con el mensaje SEI cuando las imágenes CL-RAS asociados con la unidad de acceso IRAP no están presentes en el flujo de bits de datos de vídeo. El mensaje SEI incluye un elemento sintáctico de desviación de retardo de capa cruzada DPB que especifica una desviación que se utilizará en una obtención de los tiempos de salida DPB de la unidad de acceso IRAP asociada con el mensaje SEI cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP no están presentes en el flujo de bits de datos de vídeo.

**[0206]** Además, en el primer conjunto de ejemplos de las técnicas de esta divulgación, se pueden realizar los siguientes cambios en el Anexo C del borrador de trabajo de HEVC 10.

**[0207]** Este anexo especifica el descodificador de referencia hipotético (HRD) y su uso para comprobar el flujo de bits y la conformidad del descodificador.

**[0208]** Dos tipos de flujos de bits o subconjuntos de flujos de bits están sujetos a la comprobación de conformidad de HRD para esta memoria descriptiva. El primer tipo, llamado un flujo de bits de Tipo I, es un flujo de unidades NAL que contiene solo las unidades VCL NAL y las 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, denominado flujo de bits Tipo II, contiene, además de las unidades VCL NAL y las unidades NAL de datos de relleno para todas las unidades de acceso en el flujo de bits, al menos uno de los siguientes:

- unidades NAL no VCL adicionales 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 bytes del flujo de la unidad NAL (como se especifica en el Anexo B).

**[0209]** La figura C-1 muestra los tipos de puntos de conformidad del flujo de bits comprobados por el HRD.

**[0210]** Los elementos sintácticos de unidades no VCL NAL (o sus valores predeterminados para algunos de los elementos sintácticos), requeridos para el HRD, se especifican en las subcláusulas semánticas de la cláusula 7, Anexos D y E.

**[0211]** Se usan dos tipos de conjuntos de parámetros de HRD (parámetros NAL HRD y parámetros VCL HRD). Los conjuntos de parámetros de HRD se señalizan a través de la estructura sintáctica `hrd_parameters()`, que puede ser parte de la estructura sintáctica de SPS o de la estructura sintáctica de VPS.

**[0212]** Se pueden necesitar múltiples pruebas para comprobar la conformidad de un flujo de bits, que se conoce como el flujo de bits bajo prueba. Para cada prueba, se aplican los siguientes pasos en el orden indicado:

1. Se selecciona un punto de funcionamiento bajo prueba, indicado como `TargetOp`. La lista de identificadores de capa `OpLayerIdList` de `TargetOp` consiste en la lista de los valores de `nuh_layer_id`, en orden creciente de los valores de `nuh_layer_id`, presentes en el subconjunto del flujo de bits asociado con `TargetOp`, que es un subconjunto de los valores de ID de la capa de `nuh` presentes en el flujo de bits bajo prueba. El `OpTid` de `TargetOp` es igual al `TemporalId` más alto presente en el subconjunto del flujo de bits asociado con `TargetOp`.

2. `TargetDecLayerIdList` se establece igual a `OpLayerIdList` de `TargetOp`, `HighestTid` se establece igual a `OpTid` de `TargetOp`, y el proceso de extracción del sub-flujo de bits especificado en la cláusula 10 se invoca con el flujo de bits sometido a prueba, `HighestTid` y `TargetDecLayerIdList` como entradas, y la salida se asigna a `BitstreamToDecode`.

3. La estructura sintáctica `hrd_parameters()` y la estructura sintáctica `sub_layer_hrd_parameters()` aplicable a `TargetOp` se seleccionan. Si `TargetDecLayerIdList` contiene todos los valores `nuh_layer_id` presentes en el flujo de bits bajo prueba, se selecciona la estructura sintáctica `hrd_parameters()` en el SPS activo (o proporcionado a través de medios externos no especificados en esta memoria descriptiva). De lo contrario, se selecciona la estructura sintáctica `hrd_parameters()` en el VPS activo (o proporcionado a través de medios externos no especificados en esta memoria descriptiva) que se aplica a `TargetOp`. 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(HighestTid)` que sigue inmediatamente a la condición "`if(vcl_hrd_parameters_present_flag)`" y la variable `NalHrdModeFlag` se establece igual a 0; de lo contrario (`BitstreamToDecode` es un flujo de bits de tipo II), se selecciona la estructura sintáctica `sub_layer_hrd_parameters(HighestTid)` que sigue inmediatamente a la condición "`if(vcl_hrd_parameters_present_flag)`" (en este caso, la variable `NalHrdModeFlag` se establece igual a 0) o la condición "`if(nal_hrd_parameters_present_flag)`" (en este caso la variable `NalHrdModeFlag` se establece igual a 1). Cuando `BitstreamToDecode` es un flujo de bits Tipo II y `NalHrdModeFlag` es igual a 0, todas las unidades NAL no VCL excepto las unidades NAL de datos de relleno y todos los elementos sintácticos `leading_zero_8bits`, `zero_byte`, `start_code_prefix_one_3bytes`, y `trailing_zero_8bits`

que forman un flujo de bytes a partir el flujo de la unidad NAL (como se especifica en el Anexo B), cuando están presentes, se descartan de BitstreamToDecode, y el flujo de bits restante se asigna a BitstreamToDecode.

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

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

20 6. Se selecciona un valor de SchedSelIdx. El SchedSelIdx seleccionado estará en el rango de 0 a cpb\_cnt\_minus1[HighestTid], inclusive, donde cpb\_cnt\_minus1[HighestTid] se encuentra en la estructura sintáctica sub\_layer\_hrd\_parameters(HighestTid) como se seleccionó anteriormente.

25 7. Cuando la imagen codificada en la unidad de acceso 0 tiene nal\_unit\_type igual a CRA\_NUT o BLA\_W\_LP, e irap\_cpb\_params\_present\_flag en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado es igual a 1, cualquiera de las siguientes opciones se aplica para la selección de la desviación de retardo y el retardo de eliminación de CPB:

30 - Si NalHrdModeFlag es igual a 1, la desviación de retardo y el retardo de eliminación de CPB inicial predeterminados representados por nal\_initial\_cpb\_removal\_delay [SchedSelIdx] y nal\_initial\_cpb\_removal\_offset [SchedSelIdx], respectivamente, se seleccionan en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado. De lo contrario, se selecciona la desviación de retardo y el retardo de eliminación de CPB inicial predeterminados representados por vcl\_initial\_cpb\_removal\_delay[SchedSelIdx] y vcl\_initial\_cpb\_removal\_offset[SchedSelIdx], respectivamente, en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado. La variable DefaultInitCpbParamsFlag se establece igual a 1.

40 - Si NalHrdModeFlag es igual a 1, se selecciona la desviación de retardo y el retardo de eliminación de CPB inicial alternativa representados por nal\_initial\_alt\_cpb\_removal\_delay[SchedSelIdx] y nal\_initial\_alt\_cpb\_removal\_offset[SchedSelIdx], respectivamente, en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado. De lo contrario, se selecciona la desviación de retardo y el retardo de eliminación de CPB inicial alternativa representados por vcl\_initial\_alt\_cpb\_removal\_delay[SchedSelIdx] y vcl\_initial\_alt\_cpb\_removal\_offset[SchedSelIdx], respectivamente, en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado. La variable DefaultInitCpbParamsFlag se establece igual a 0, y las unidades de acceso RASL asociadas con la unidad de acceso 0 se descartan de BitstreamToDecode y el flujo de bits restante se asigna a BitstreamToDecode.

50 - **Si NalHrdModeFlag es igual a 1, se selecciona una desviación de retardo y un retardo de eliminación de CPB inicial alternativa representados por nal\_initial\_alt\_cross\_layer\_cpb\_removal\_delay[SchedSelIdx] y nal\_initial\_alt\_cross\_layer\_cpb\_removal\_offset[SchedSelIdx], respectivamente, en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado. De lo contrario, se selecciona la desviación de retardo y el retardo de eliminación de CPB inicial alternativa representados por vcl\_initial\_alt\_cross\_layer\_cpb\_removal\_delay[SchedSelIdx] y vcl\_initial\_alt\_cross\_layer\_cpb\_removal\_offset[SchedSelIdx], respectivamente, en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado. La variable DefaultInitCpbParamsFlag se establece igual a 0, y las imágenes CL-RAS asociadas con la unidad de acceso 0 se descartan de BitstreamToDecode y el flujo de bits restante se asigna a BitstreamToDecode.**

60 8. Cuando sub\_pic\_hrd\_params\_present\_flag en la estructura sintáctica hrd\_parameters ( ) seleccionada es igual a 1, la CPB se programa para funcionar a nivel de la unidad de acceso (en cuyo caso la variable SubPicHrdFlag se establece en 0) o a nivel de sub-imagen (en cuyo caso la variable SubPicHrdFlag se establece igual a 1).

65 **[0213]** Para cada punto de funcionamiento bajo prueba, el número de pruebas de conformidad del flujo de bits que se realizará es  $n_0 * n_1 * (n_2 * 2 + n_3) * n_4$ , donde los valores de  $n_0$ ,  $n_1$ ,  $n_2$ ,  $n_3$  y  $n_4$  se especifican de la forma siguiente:

- n0 se obtiene de la forma siguiente:
  - Si BitstreamToDecode es un flujo de bits de Tipo I, n0 es igual a 1.
- 5 - De lo contrario (BitstreamToDecode es un flujo de bits de Tipo II), n0 es igual a 2.
- n1 es igual a  $cpb\_cnt\_minus1[HighestTid] + 1$ .
- n2 es el número de unidades de acceso en BitstreamToDecode que cada una está asociada con un mensaje SEI de período de almacenamiento en memoria intermedia aplicable a TargetOp y para cada una de las cuales se cumplen las dos condiciones siguientes:
  - nal\_unit\_type es igual a CRA\_NUT o BLA\_W\_LP para las unidades VCL NAL;
- 15 - El mensaje SEI de período de almacenamiento en memoria intermedia asociado aplicable a TargetOp tiene irap\_cpb\_params\_present\_flag igual a 1.
- n3 es el número de unidades de acceso en BitstreamToDecode que cada una está asociada con un mensaje SEI de período de almacenamiento en memoria intermedia aplicable a TargetOp y para cada una de las cuales una o las dos condiciones siguientes son verdaderas:
  - nal\_unit\_type no es igual a CRA\_NUT ni a BLA\_W\_LP para las unidades VCL NAL;
  - El mensaje SEI de período de almacenamiento en memoria intermedia asociado aplicable a TargetOp tiene irap\_cpb\_params\_present\_flag igual a 0.
- 25 - n4 se obtiene de la forma siguiente:
  - Si sub\_pic\_hrd\_params\_present\_flag en la estructura sintáctica hrd\_parameters( ) seleccionada es igual a 0, n4 es igual a 1;
  - De lo contrario, n4 es igual a 2.

35 **[0214]** Cuando BitstreamToDecode es un flujo de bits de Tipo II, se aplica lo siguiente:

- Si se selecciona la estructura sintáctica sub\_layer\_hrd\_parameters(HighestTid) que sigue inmediatamente a 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 las unidades NAL de datos de relleno y VCL se cuentan para la velocidad de transmisión de bits de entrada y el almacenamiento de CPB.
- 40 - De lo contrario (se selecciona la estructura sintáctica sub\_layer\_hrd\_parameters(HighestTid) que sigue inmediatamente a la condición

45 "if(nal\_hrd\_parameters\_present\_flag)", la prueba se realiza en el punto de conformidad de Tipo II que se muestra en la Figura C-1, y se cuentan todos los bytes del flujo de bits Tipo II, que puede ser un flujo unitario NAL o un flujo de bytes. para la velocidad de transmisión de bits de entrada y el almacenamiento de CPB.

50 NOTA 1 - Los parámetros NAL HRD 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 VCL HRD 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 se programe la siguiente imagen para comenzar a llegar. Por ejemplo, cuando se descodifica una CVS conforme a uno o más de los perfiles especificados en el Anexo A utilizando el proceso de descodificación especificado en las cláusulas 2 a 10, cuando se proporcionan parámetros NAL HRD para el punto de conformidad de Tipo II que no solo se encuentran dentro de los límites establecidos para los parámetros NAL HRD para la conformidad del perfil en el punto f de la subcláusula A.4.2, sino que también se encuentran dentro de los límites establecidos para los parámetros VCL HRD para la conformidad del perfil en el punto e) de la subcláusula A.4.2, la conformidad del VCL HRD para el punto de conformidad de Tipo I también se garantiza que se encuentra dentro de los límites del elemento e) de la subcláusula A.4.2.

65 **[0215]** Todos los VPS, SPS y PPS a los que se hace referencia en las unidades VCL NAL, y los correspondientes mensajes SEI de período de almacenamiento en memoria intermedia, información de la unidad de descodificación y sincronización de imágenes 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 intermedia.

**[0216]** En los Anexos C, D y E, la especificación de "presencia" de unidades NAL que no son VCL que contienen mensajes SEI de período de almacenamiento en memoria intermedia, VPS, SPS, PPS, mensajes SEI de sincronización de imágenes o mensajes SEI de información de unidad de descodificación también se satisface cuando esas unidades NAL (o solo algunas de ellas) se transmiten a los descodificadores (o al HRD) por otros medios no especificados en 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 2 - Como ejemplo, la sincronización de tal 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.

**[0217]** Cuando el contenido de tal 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.

NOTA 3 - 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 basándose únicamente en la información contenida en el flujo de bits. Cuando la información del HRD no está presente en el flujo de bits, como es el caso de todos los flujos de bits tipo I "independientes", la conformidad solo puede verificarse cuando los datos del HRD se suministran por algún otro medio no especificado en esta memoria descriptiva. 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.

**[0218]** Para cada prueba de conformidad del flujo de bits, el tamaño de CPB (número de bits) es  $CpbSize[SchedSelIdx]$  como se especifica en la subcláusula E.2.3, donde  $SchedSelIdx$  y los parámetros HRD se especifican anteriormente en esta subcláusula. El tamaño de DPB (número de memorias intermedias de almacenamiento de imágenes) es

**[0219]**  $sps\_max\_dec\_pic\_buffering\_minus1[HighestTid] + 1$ .

**[0220]** La variable  $El\_SubPicHrdPreferredFlag$  se especifica ya sea por medios externos, o cuando no se especifica por medios externos, se establece igual a 0.

**[0221]** Cuando el valor de la variable  $SubPicHrdFlag$  no se ha establecido en el paso 8 anterior en esta subcláusula, se obtiene de la forma siguiente:

$$SubPicHrdFlag = SubPicHrdPreferredFlag \ \&\& \\ sub\_pic\_hrd\_params\_present\_flag(C-1)$$

**[0222]** Si  $SubPicHrdFlag$  es igual a 0, el HRD 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 sub-imagen, y cada unidad de descodificación es un subconjunto de una unidad de acceso.

NOTA 4 - Si el HRD funciona a nivel de la unidad de acceso, cada vez se elimina de la CPB una unidad de descodificación que es una unidad de acceso completa. De lo contrario (el HRD funciona a nivel de sub-imagen), cada vez se elimina de la CPB una unidad de descodificación que es un subconjunto de una unidad de acceso. En ambos casos, cada vez se emite una imagen descodificada completa desde la DPB, aunque el tiempo de salida de la imagen se obtiene basándose en los tiempos de eliminación de CPB obtenidos de forma diferente y los retardos de salida de DPB señalizados de forma diferente.

**[0223]** Se especifica lo siguiente para expresar las restricciones en este anexo:

- Cada unidad de acceso se denomina unidad de acceso  $n$ , donde el número  $n$  identifica la unidad de acceso particular. La unidad de acceso 0 se selecciona según el paso 4 anterior. El valor de  $n$  puede aumentarse en 1 para cada unidad de acceso posterior en el orden de descodificación.
- Cada unidad de descodificación puede denominarse unidad de descodificación  $m$ , donde el número  $m$  identifica la unidad de descodificación particular. La primera unidad de descodificación en orden de descodificación en la unidad de acceso 0 se denomina unidad de descodificación 0. El valor de  $m$  se aumenta en 1 para cada unidad de descodificación posterior en orden de descodificación.

NOTA 5 - La numeración de las unidades de descodificación es relativa a la primera unidad de descodificación en la unidad de acceso 0.

- La imagen n se refiere a la imagen codificada o la imagen descodificada de la unidad de acceso n.

**[0224]** El HRD funciona de la forma siguiente:

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

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

- 10 - Los datos asociados con las unidades de descodificación que fluyen en la CPB de acuerdo con una programación de llegada especificada son entregados por el HSS.
- 15 - Los datos asociados a cada unidad de descodificación se eliminan y descodifican instantáneamente mediante el proceso de descodificación instantánea en el momento de eliminación de CPB de la unidad de descodificación.
- Cada imagen descodificada se coloca en la DPB.
- 20 - Se elimina una imagen descodificada de la DPB cuando ya no es necesaria para la referencia de predicción inter y ya no es necesaria para la salida.

**[0225]** Para cada prueba de conformidad de flujo de bits, la operación de la CPB se especifica en la subcláusula C.2, la operación de descodificador instantánea se especifica en las cláusulas 2 a 10, se especifica el funcionamiento de la DPB en la subcláusula C.3, y el recorte de salida se especifica en la subcláusula C.3.3 y en la subcláusula C.5.2.2.

25 **[0226]** La información de HSS y HRD concerniente al número de programas de entrega enumerados y sus velocidad de transmisión de bits y tamaños de memorias intermedias asociados se especifican en las subcláusulas E.1.2 y E.2.2. El HRD se inicializa como se especifica en el mensaje SEI período de almacenamiento en memoria intermedia especificado en subcláusulas D.2.2 y D.3.2. El momento eliminación de unidades de descodificación de la CPB y la sincronización de salida de imágenes descodificadas desde la DPB se especifica mediante información en los mensajes SEI de sincronización de imágenes (especificados en subcláusulas D.2.3 y D.3.3) o en los mensajes SEI de información de la unidad de descodificación (especificados en las subcláusulas D.2.21 y D.3.21). Toda la información de sincronización relacionada con una unidad de descodificación específica deberá llegar antes del tiempo de eliminación de CPB de la unidad de descodificación.

35 **[0227]** Los requisitos para la conformidad de flujo de bits se especifican en la subcláusula C.4, y el HRD se utiliza para comprobar la conformidad de los flujos de bits como se especifica anteriormente en esta subcláusula y para comprobar la conformidad de los descodificadores como se especifica en la subcláusula C.5.

40 **NOTA 7** - Si bien la conformidad está garantizada bajo el supuesto de que todas las velocidades de imagen y los relojes utilizados para generar el flujo de bits coinciden exactamente con los valores señalados en el flujo de bits, en un sistema real, cada uno de estos puede variar respecto al valor señalado o especificado.

45 **[0228]** 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.

**[0229]** La variable ClockTick se obtiene de la forma siguiente y se denomina tic de reloj:

$$50 \quad \text{ClockTick} = \text{vui\_num\_units\_in\_tick} \div \text{vui\_time\_scale} \text{ (C-2)}$$

**[0230]** La variable ClockSubTick se obtiene de la forma siguiente y se llama sub-tic del reloj:

$$55 \quad \text{ClockSubTick} = \text{ClockTick} \div (\text{tick\_divisor\_minus2} + 2) \text{ (C-3)}$$

## **C.2 Funcionamiento de la memoria intermedia de imágenes descodificadas (DPB)**

### **C.2.1 General**

60 **[0231]** Las especificaciones en esta subcláusula se aplican independientemente a cada conjunto de parámetros de CPB que está presente y a los puntos de conformidad de Tipo I y Tipo II mostrados en la Figura C-1, y el conjunto de parámetros de CPB se selecciona como se especifica en la subcláusula C.1.

### **C.2.2 Sincronización de llegada de la unidad de descodificación**

65

**[0232]** Si SubPicHrdFlag es igual a 0, la variable subPicParamsFlag se establece en 0, y el proceso especificado en el resto de esta subcláusula se invoca con una unidad de descodificación considerada como una unidad de acceso, para la obtención de los tiempos de llegada de CPB inicial y final para la unidad de acceso n.

5 **[0233]** De lo contrario (SubPicHrdFlag es igual a 1), el proceso especificado en el resto de esta subcláusula se invoca primero con la variable subPicParamsFlag establecida igual a 0 y una unidad de descodificación se considera como una unidad de acceso, para la obtención de los tiempos de llegada de CPB inicial y final para la unidad de acceso n, y luego se invocan con subPicParamsFlag igual a 1 y una unidad de descodificación se considera como un subconjunto de una unidad de acceso, para la obtención de los tiempos de llegada de CPB inicial y final para las unidades de descodificación en la unidad de acceso n.

**[0234]** Las variables InitCpbRemovalDelay[SchedSelIdx] e InitCpbRemovalDelayOffset[SchedSelIdx] se obtienen de la forma siguiente:

15 - Si una o más de las siguientes condiciones son verdaderas, InitCpbRemovalDelay[SchedSelIdx] e InitCpbRemovalDelayOffset[SchedSelIdx] se establecen iguales a los valores de los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria intermedia **nal\_initial\_alt\_cpb\_removal\_delay**[SchedSelIdx] y **nal\_initial\_alt\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 1 o **vcl\_initial\_alt\_cpb\_removal\_delay**[SchedSelIdx] y **vcl\_initial\_alt\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 0, donde los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria intermedia se seleccionan como especificados en la subcláusula C.1:

20 - La unidad de acceso 0 es una unidad de acceso BLA para la cual la imagen codificada tiene nal\_unit\_type igual a BLA\_W\_DLP o BLA\_N\_LP, y el valor de irap\_cpb\_params\_present\_flag del mensaje SEI de período de almacenamiento en memoria intermedia es igual a 1.

25 - La unidad de acceso 0 es una unidad de acceso BLA para la cual la imagen codificada tiene nal\_unit\_type igual a BLA\_W\_LP o es una unidad de acceso CRA, y el valor de irap\_cpb\_params\_present\_flag del mensaje SEI de período de almacenamiento en memoria intermedia es igual a 1, y una más de las siguientes condiciones se cumplen:

30 - UseAltCpbParamsFlag para la unidad de acceso 0 es igual a 1.

35 - DefaultInitCpbParamsFlag es igual a 0.

- El valor de subPicParamsFlag es igual a 1.

40 - **De lo contrario, si la unidad de acceso 0 es una unidad de acceso IRAP inicial, el valor de NoCIRasPicPresentFlag es igual a 1, y el valor de irap\_cross\_layer\_cpb\_params\_present\_flag del mensaje SEI de período de almacenamiento en memoria intermedia es igual a 1, InitCpbRemovalDelay[SchedSelIdx] e InitCpbRemovalDelayOffset[SchedSelIdx] se establecen iguales a los valores de los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria intermedia **nal\_initial\_alt\_cross\_layer\_cpb\_removal\_delay**[SchedSelIdx] y **nal\_initial\_alt\_cross\_layer\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 1, o **vcl\_initial\_alt\_cross\_layer\_cpb\_removal\_delay**[SchedSelIdx] y **vcl\_initial\_alt\_cross\_laye\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 0, donde el mensaje SEI de período de almacenamiento en memoria intermedia que contiene los elementos sintácticos se selecciona como se especifica en la subcláusula C.1.**

50 - De lo contrario, InitCpbRemovalDelay[SchedSelIdx] e InitCpbRemovalDelayOffset[SchedSelIdx] se establecen igual a los valores de los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria intermedia **al\_initial\_cpb\_removal\_delay**[SchedSelIdx] y **nal\_initial\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 1, o **vcl\_initial\_cpb\_removal\_delay**[SchedSelIdx] y **vcl\_initial\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 0, donde los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria intermedia se seleccionan como se especifica en la subcláusula C.1.

60 **[0235]** El momento en el que el primer bit de la unidad de descodificación m comienza a entrar en la CPB puede denominarse el tiempo de llegada inicial initArrivalTime[m].

**[0236]** El tiempo de llegada inicial de la unidad de descodificación m se obtiene de la forma siguiente:

65 - Si la unidad de descodificación es la unidad descodificación 0 (es decir, m = 0), initArrivalTime[0] = 0.

- De otro modo (la unidad de descodificación es la unidad de descodificación m con m > 0), se aplica lo siguiente:

- Si  $cbr\_flag[SchedSelIdx]$  es igual a 1, el tiempo de llegada inicial para la unidad de descodificación  $m$ , es igual al tiempo de llegada final (que se obtiene a continuación) de la unidad de descodificación  $m - 1$ , es decir,

5             $if(!subPicParamsFlag)$

$initArrivalTime[m] =$   
 $AuFinalArrivalTime[m - 1]$  (C-4)

de lo contrario

10             $initArrivalTime[m] =$   
 $DuFinalArrivalTime[m - 1]$

- De otro modo ( $cbr\_flag[SchedSelIdx]$  es igual a 0), el tiempo de llegada inicial para la unidad de descodificación  $m$  se obtiene de la forma siguiente:

15             $if(!subPicParamsFlag)$

$initArrivalTime[m] = Max($   
 $AuFinalArrivalTime[m - 1], initArrivalEarliestTime[m])$  (C-5)

20            de lo contrario

$initArrivalTime[m] = Max($   
 $DuFinalArrivalTime[m - 1], initArrivalEarliestTime[m])$

donde  $initArrivalEarliestTime[m]$  se obtiene de la forma siguiente:

- La variable  $tmpNominalRemovalTime$  se obtiene de la forma siguiente:

$if(!subPicParamsFlag)$

$tmpNominalRemovalTime =$   
 $AuNominalRemovalTime[m]$  (C-6)

de lo contrario

$tmpNominalRemovalTime =$   
 $DuNominalRemovalTime[m]$

35            donde  $AuNominalRemovalTime[m]$  y

$DuNominalRemovalTime[m]$  es el tiempo de eliminación de CPB nominal de la unidad de acceso  $m$  y la unidad de descodificación  $m$ , respectivamente, como se especifica en la subcláusula C.2.3.

- Si la unidad de descodificación  $m$  no es la primera unidad de descodificación de un período de almacenamiento en memoria intermedia posterior,  $initArrivalEarliestTime[m]$  se obtiene de la forma siguiente:

$initArrivalEarliestTime[m] = tmpNominalRemovalTime -$   
 $( InitCpbRemovalDelay[ SchedSelIdx ]$   
 $+ InitCpbRemovalDelayOffset[ SchedSelIdx ] ) \div$   
45            90000      (C-7)

- De otro modo (la unidad de descodificación  $m$  es la primera unidad de descodificación de un período de almacenamiento en memoria intermedia posterior),  $\text{initArrivalEarliestTime}[m]$  se obtiene de la forma siguiente:

$$\text{initArrivalEarliestTime}[m] = \text{tmpNominalRemovalTime} - (\text{InitCpbRemovalDelay}[\text{SchedSelIdx}] \div 90000)$$

5 (C-8)

**[0237]** El tiempo de llegada final para la unidad de descodificación  $m$  se obtiene de la forma siguiente:

10  $\text{if}(\text{!subPicParamsFlag})$

$$\text{AuFinalArrivalTime}[m] = \text{initArrivalTime}[m] + \text{sizeInbits}[m] \div \text{BitRate}[\text{SchedSelIdx}] \quad (\text{C-9})$$

de lo contrario

$$\text{DuFinalArrivalTime}[m] = \text{initArrivalTime}[m] + \text{sizeInbits}[m] \div \text{BitRate}[\text{SchedSelIdx}]$$

15

donde  $\text{sizeInbits}[m]$  es el tamaño en bits de la unidad de descodificación  $m$ , contando los bits de las unidades VCL NAL y las unidades NAL de datos de relleno para el punto de conformidad de Tipo I o todos los bits del flujo de bits de Tipo II para el punto de la conformidad de Tipo II, donde los puntos de conformidad de Tipo I y Tipo II son los que se muestran en la figura C-1.

20

**[0238]** Además, los valores de  $\text{SchedSelIdx}$ ,  $\text{BitRate}[\text{SchedSelIdx}]$ , y  $\text{CpbSize}[\text{SchedSelIdx}]$  pueden limitarse de la forma siguiente:

25

- Si el contenido de las estructuras sintácticas  $\text{hrd\_parameters}()$  seleccionadas para la unidad de acceso que contiene la unidad de descodificación  $m$  y la unidad de acceso anterior difieren, el HSS selecciona un valor  $\text{SchedSelIdx1}$  de  $\text{SchedSelIdx}$  de entre los valores de  $\text{SchedSelIdx}$  proporcionados en las estructuras sintácticas  $\text{hrd\_parameters}()$  seleccionadas para la unidad de acceso que contiene la unidad de descodificación  $m$  que se traduce en  $\text{BitRate}[\text{SchedSelIdx1}]$  o  $\text{CpbSize}[\text{SchedSelIdx1}]$  para la unidad de acceso que contiene la unidad de descodificación  $m$ . El valor de  $\text{BitRate}[\text{SchedSelIdx1}]$  o  $\text{CpbSize}[\text{SchedSelIdx1}]$  puede diferir del valor de  $\text{BitRate}[\text{SchedSelIdx0}]$  o  $\text{CpbSize}[\text{SchedSelIdx0}]$  para el valor de  $\text{SchedSelIdx0}$  de  $\text{SchedSelIdx}$  que estaba en uso para la unidad de acceso anterior.

30

35

- De otro modo, el HSS continúa funcionando con los valores anteriores de  $\text{SchedSelIdx}$ ,  $\text{BitRate}[\text{SchedSelIdx}]$  y  $\text{CpbSize}[\text{SchedSelIdx}]$ .

**[0239]** Cuando el HSS selecciona los valores de  $\text{BitRate}[\text{SchedSelIdx}]$  o  $\text{CpbSize}[\text{SchedSelIdx}]$  que difieren de los de la unidad de acceso anterior, se aplica lo siguiente:

40

- La variable  $\text{BitRate}[\text{SchedSelIdx}]$  entra en vigencia en el tiempo llegada de CPB inicial de la unidad de acceso actual.

- La variable  $\text{CpbSize}[\text{SchedSelIdx}]$  entra en vigor de la forma siguiente:

45

- Si el nuevo valor de  $\text{CpbSize}[\text{SchedSelIdx}]$  es mayor que el tamaño de CPB anterior, entra en vigencia en el momento de llegada de CPB inicial de la unidad de acceso actual.
- De otro modo, el nuevo valor de  $\text{CpbSize}[\text{SchedSelIdx}]$  entra en vigor en el momento de la eliminación de la CPB de la unidad de acceso actual.

50

### C.2.3 Sincronización de la eliminación de la unidad de descodificación y la descodificación de la unidad de descodificación

55

**[0240]** Las variables  $\text{InitCpbRemovalDelay}[\text{SchedSelIdx}]$ ,  $\text{InitCpbRemovalDelayOffset}[\text{SchedSelIdx}]$ ,  $\text{CpbDelayOffset}$  y  $\text{DpbDelayOffset}$  se obtienen de la forma siguiente:

- Si una o más de las siguientes condiciones son verdaderas,  $\text{CpbDelayOffset}$  se establece igual al valor del elemento sintáctico de mensaje SEI de período de almacenamiento en memoria intermedia  $\text{cpb\_delay\_offset}$ ,

DpbDelayOffset se establece igual al valor del elemento sintáctico de mensaje SEI de período de almacenamiento en memoria intermedia e InitCpbRemovalDelay[SchedSelIdx] e InitCpbRemovalDelayOffset[SchedSelIdx] se establecen iguales a los valores de los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria intermedia **nal\_initial\_alt\_cpb\_removal\_delay**[SchedSelIdx] y **nal\_initial\_alt\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 1, o **vcl\_initial\_alt\_cpb\_removal\_delay**[SchedSelIdx] y **vcl\_initial\_alt\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 0, donde el mensaje SEI de período de almacenamiento en memoria intermedia que contiene los elementos sintácticos se selecciona como se especifica en la subcláusula C.1:

- La unidad de acceso 0 es una unidad de acceso BLA para la cual la imagen codificada tiene nal\_unit\_type igual a BLA\_W\_DLP o BLA\_N\_LP, y el valor de irap\_cpb\_params\_present\_flag del mensaje SEI de período de almacenamiento en memoria intermedia es igual a 1.

- La unidad de acceso 0 es una unidad de acceso BLA para la cual la imagen codificada tiene nal\_unit\_type igual a BLA\_W\_LP o es una unidad de acceso CRA, y el valor de irap\_cpb\_params\_present\_flag del mensaje SEI de período de almacenamiento en memoria intermedia es igual a 1, y una más de las siguientes condiciones se cumplen:

- UseAltCpbParamsFlag para la unidad de acceso 0 es igual a 1.

- DefaultInitCpbParamsFlag es igual a 0.

*De lo contrario, si la unidad de acceso 0 es una unidad de acceso IRAP inicial, el valor de NoCIRasPicPresentFlag es igual a 1, y el valor de irap\_cross\_layer\_cpb\_params\_present\_flag del mensaje SEI de período de almacenamiento en memoria intermedia es igual a 1, CpbDelayOffset se establece igual al valor del elemento sintáctico de mensaje SEI de período de almacenamiento en memoria intermedia **cpb\_cross\_layer\_delay\_offset**, DpbDelayOffset se establece igual al valor del elemento sintáctico de mensaje SEI de período de almacenamiento en memoria intermedia **dpb\_cross\_layer\_delay\_offset**, e InitCpbRemovalDelay[SchedSelIdx] e InitCpbRemovalDelayOffset[SchedSelIdx] se establecen iguales a los valores de los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria intermedia **nal\_initial\_alt\_cross\_layer\_cpb\_removal\_delay**[SchedSelIdx] y **nal\_initial\_alt\_cross\_layer\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 1, o **vcl\_initial\_alt\_cross\_layer\_cpb\_removal\_delay**[SchedSelIdx] y **vcl\_initial\_alt\_cross\_laye\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 0, donde el mensaje SEI de período de almacenamiento en memoria intermedia que contiene los elementos sintácticos se selecciona como se especifica en la subcláusula C.1.*

- De lo contrario, InitCpbRemovalDelay[SchedSelIdx] e InitCpbRemovalDelayOffset[SchedSelIdx] se establecen iguales a los valores de los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria intermedia **al\_initial\_cpb\_removal\_delay**[SchedSelIdx] y **nal\_initial\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 1, o **vcl\_initial\_cpb\_removal\_delay**[SchedSelIdx] y **vcl\_initial\_cpb\_removal\_offset**[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 0, donde el mensaje SEI de período de almacenamiento en memoria intermedia que contiene los elementos sintácticos se selecciona como se especifica en la subcláusula C.1, CpbDelayOffset y DpbDelayOffset son ambos establecidos iguales a 0.

El tiempo nominal de eliminación de la unidad de acceso n de la CPB se especifica de la forma siguiente:

- Si la unidad de acceso n es la unidad de acceso con n igual a 0 (la unidad de acceso que inicializa el HRD), el tiempo de eliminación nominal de la unidad de acceso de la CPB se especifica mediante:

$$AuNominalRemovalTime[ 0 ] = \text{InitCpbRemovalDelay}[ \text{SchedSelIdx} ] \div 90000 \text{ (C-10)}$$

- De lo contrario, se aplica lo siguiente:

- Cuando la unidad de acceso n es la primera unidad de acceso de un período de almacenamiento en memoria intermedia que no inicializa el HRD, se aplica lo siguiente:

El tiempo de eliminación nominal de la unidad de acceso n de la CPB se especifica mediante:

```
if(!concatenationflag) {
```

```

baseTime =
AuNominalRemovalTime[ firstPicInPrevBuffPeriod ]
tmpCpbRemovalDelay = AuCpbRemovalDelayVal
5      } de lo contrario {
baseTime = AuNominalRemovalTime[ prevNonDiscardablePic ]
tmpCpbRemovalDelay =
Max( ( auCpbRemovalDelayDeltaMinus1 + 1 ), (C-11)
Ceil( ( InitCpbRemovalDelay[ SchedSelIdx ] ÷ 90000 +
AuFinalArrivalTime[ n - 1 ] - AuNominalRemovalTime[ n - 1 ]
10      ) ÷ ClockTick )
}
AuNominalRemovalTime( n ) = baseTime + ClockTick *
( tmpCpbRemovalDelay - CpbDelayOffset )

```

15 donde  $AuNominalRemovalTime[firstPicInPrevBuffPeriod]$  es el tiempo de eliminación nominal de la primera unidad de acceso del período de almacenamiento en memoria intermedia anterior,  $AuNominalRemovalTime[prevNonDiscardablePic]$  es el tiempo de eliminación nominal de la imagen anterior en orden de descodificación con  $TemporalId$  igual a 0 que no es imagen sin referencia de RASL, RADL o sub-capa,  $AuCpbRemovalDelayVal$  es el valor de  $AuCpbRemovalDelayVal$  obtenido de acuerdo con  $au\_cpb\_removal\_delay\_delta\_minus1$  en el mensaje SEI de sincronización, seleccionado como se especifica en la subcláusula C.1, asociado con la unidad de acceso  $n$ , y  $concatenationFlag$  y  $auCpbRemovalDelayDeltaMinus1$  son los valores de los elementos sintácticos  $concatenation\_flag$  y  $au\_cpb\_removal\_delay\_delta\_minus1$ , respectivamente, en el mensaje SEI de período de almacenamiento en memoria intermedia, seleccionados como se especifica en la subcláusula C.1 asociada con la unidad de acceso  $n$ .

Después de la obtención del tiempo de eliminación de CPB nominal y antes de la obtención del tiempo de salida de DPB de la unidad de acceso  $n$ , los valores de  $CpbDelayOffset$  y  $DpbDelayOffset$  se actualizan de la forma siguiente:

- 30 - Si una o más de las siguientes condiciones son verdaderas,  $CpbDelayOffset$  se establece igual al valor del elemento sintáctico de mensaje SEI de período de almacenamiento en memoria intermedia  $cpb\_delay\_offset$ , y  $DpbDelayOffset$  se establece igual al valor del elemento sintáctico de mensaje SEI de período de almacenamiento en memoria intermedia  $dpb\_delay\_offset$ , donde el mensaje SEI de período de almacenamiento en memoria intermedia que contiene los elementos sintácticos se selecciona como se especifica en la subcláusula C.1:
  - 35 - La unidad de acceso  $n$  es una unidad de acceso BLA para la cual la imagen codificada tiene  $nal\_unit\_type$  igual a  $BLA\_W\_DLP$  o  $BLA\_N\_LP$ , y el valor de  $rap\_cpb\_params\_present\_flag$  del mensaje SEI de período de almacenamiento en memoria intermedia es igual a 1.
  - 40 - La unidad de acceso  $n$  es una unidad de acceso BLA para la cual la imagen codificada tiene  $nal\_unit\_type$  igual a  $BLA\_W\_LP$  o es una unidad de acceso CRA, y el valor de  $irap\_cpb\_params\_present\_flag$  del mensaje SEI de período de almacenamiento en memoria intermedia es igual a 1, y  $UseAltCpbParamsFlag$  para la unidad de acceso  $n$  es igual a 1.
  - 45 - **De lo contrario, si la unidad de acceso  $n$  es una unidad de acceso IRAP inicial, el valor de  $NoCirasPicPresentFlag$  es igual a 1, y el valor de  $irap\_cross\_layer\_cpb\_params\_present\_flag$  del mensaje SEI de período de almacenamiento en memoria intermedia es igual a 1,  $CpbDelayOffset$  se establece igual al valor del elemento sintáctico de mensaje SEI de período de almacenamiento en memoria intermedia  $cpb\_cross\_layer\_delay\_offset$ , y  $DpbDelayOffset$  se establece igual al valor del elemento sintáctico de mensaje SEI de período de almacenamiento en memoria intermedia  $dpb\_cross\_layer\_delay\_offset$ , donde el mensaje SEI de período de almacenamiento en memoria**
- 50

**intermedia que contiene los elementos sintácticos se selecciona como se especifica en la subcláusula C.1.**

- De lo contrario, CpbDelayOffset y DpbDelayOffset se establecen iguales a 0.
- Cuando la unidad de acceso n no es la primera unidad de acceso de un período de almacenamiento en memoria intermedia, el tiempo nominal de eliminación de la unidad de acceso n de la CPB se especifica mediante:

$$\begin{aligned} \text{AuNominalRemovalTime}[n] = & \\ & \text{AuNominalRemovalTime}[\text{firstPicInCurrBuffPeriod}] + \\ & \text{ClockTick} * (\text{AuCpbRemovalDelayVal} - \text{CpbDelayOffset}) \end{aligned}$$

(C-12)

donde AuNominalRemovalTime[firstPicInPrevBuffPeriod] es el tiempo de eliminación nominal de la primera unidad de acceso del período de almacenamiento en memoria intermedia actual, y AuCpbRemovalDelayVal es el valor de AuCpbRemovalDelayVal obtenido de acuerdo con au\_cpb\_removal\_delay\_minus1 en el mensaje SEI de sincronización de imágenes, seleccionado como se especifica en la subcláusula C.1, asociada con la unidad de acceso n.

**[0241]** Cuando SubPicHrdFlag es igual a 1, se aplica lo siguiente:

- La variable duCpbRemovalDelayInc se obtiene de la forma siguiente:
  - Si sub\_pic\_cpb\_params\_in\_pic\_timing\_sei\_flag es igual a 0, duCpbRemovalDelayInc se establece igual al valor de du\_spt\_cpb\_removal\_delay\_increment en el mensaje SEI de información de la unidad de descodificación, seleccionado como se especifica en la subcláusula C.1, asociada a la unidad de descodificación m.
  - De lo contrario, si du\_common\_cpb\_removal\_delay\_flag es igual a 0, duCpbRemovalDelayInc se establece igual al valor de du\_cpb\_removal\_delay\_increment\_minus1[i] + 1 para la unidad de descodificación m en el mensaje SEI de sincronización de imágenes, seleccionado de la forma especificada en la subcláusula C.1, asociada con la unidad de acceso n, donde el valor de i es 0 para las primeras num\_nalus\_in\_du\_minus1[0] + 1 unidades NAL consecutivas en la unidad de acceso que contienen la unidad de descodificación m, 1 para las num\_nalus\_in\_du\_minus1 [1] + 1 unidades NAL posteriores en la misma unidad de acceso, 2 para las num\_nalus\_in\_du\_minus1 [2] + 1 unidades NAL posteriores en la misma unidad de acceso, etc.
  - De lo contrario, duCpbRemovalDelayInc se establece igual al valor de du\_common\_cpb\_removal\_delay\_increment\_minus1 + 1 en el mensaje SEI de sincronización de imágenes, seleccionado como se especifica en la subcláusula C.1, asociada a la unidad de acceso n.
- El tiempo de eliminación nominal de la unidad de descodificación m de la CPB se especifica de la forma siguiente, donde AuNominalRemovalTime[n] es el tiempo de eliminación nominal de la unidad de acceso n:
  - Si la unidad de descodificación m es la última unidad de descodificación en la unidad de acceso n, el tiempo de eliminación nominal de la unidad de descodificación m DuNominalRemovalTime[m] se establece igual a AuNominalRemovalTime[n].
  - De lo contrario, (la unidad de descodificación m no es la última unidad de descodificación en la unidad de acceso n), el tiempo de eliminación nominal de la unidad de descodificación m NominalRemovalTime[m] se obtiene de la forma siguiente:

$$\begin{aligned} & \text{if}(\text{sub\_pic\_cpb\_params\_in\_pic\_timing\_sei\_flag}) \\ & \text{DuNominalRemovalTime}[m] = \text{DuNominalRemovalTime}[m + 1] \\ & - \\ & \text{ClockSubTick} * \text{duCpbRemovalDelayInc} \end{aligned}$$

(C-13)

de lo contrario

$$\begin{aligned} \text{DuNominalRemovalTime}[m] = & \text{AuNominalRemovalTime}(n) - \\ & \text{ClockSubTick} * \text{duCpbRemovalDelayInc} \end{aligned}$$

**[0242]** Si SubPicHrdFlag es igual a 0, el tiempo de eliminación de la unidad de acceso n de la CPB se especifica de la forma siguiente, donde AuFinalArrivalTime[n] y AuNominalRemovalTime[n] son el tiempo de llegada de CPB final y el tiempo de eliminación de CPB nominal, respectivamente, de unidad de acceso n:

5  

$$\text{if}(\text{!low\_delay\_hrd\_flag}[\text{HighestTid}] \parallel \text{AuNominalRemovalTime}[n] \geq \text{AuFinalArrivalTime}[n])$$

$$\text{AuCpbRemovalTime}[n] = \text{AuNominalRemovalTime}[n]$$

10 de lo contrario (C-14)

$$\text{AuCpbRemovalTime}[n] = \text{AuNominalRemovalTime}[n] + \text{ClockTick} * \text{Ceil}((\text{AuFinalArrivalTime}[n] - \text{AuNominalRemovalTime}[n]) \div \text{ClockTick})$$

15 **NOTA 1** - Cuando low\_delay\_hrd\_flag[HighestTid] es igual a 1 y AuNominalRemovalTime[n] es menor que AuFinalArrivalTime[n], el tamaño de la unidad de acceso n es tan grande que impide la eliminación en el momento de eliminación nominal.

**[0243]** De lo contrario (SubPicHrdFlag es igual a 1), el tiempo de eliminación de la unidad de descodificación m de la CPB se especifica de la forma siguiente:

20  

$$\text{if}(\text{!low\_delay\_hrd\_flag}[\text{HighestTid}] \parallel \text{DuNominalRemovalTime}[m] \geq \text{DuFinalArrivalTime}[m])$$

$$\text{DuCpbRemovalTime}[m] = \text{DuNominalRemovalTime}[m]$$

25 de lo contrario (C-15)

$$\text{DuCpbRemovalTime}[m] = \text{DuFinalArrivalTime}[m]$$

30 **NOTA 2** - Cuando low\_delay\_hrd\_flag[HighestTid] es igual a 1 y DuNominalRemovalTime[m] es menor que DuFinalArrivalTime[m], el tamaño de la unidad de descodificación m es tan grande que impide la eliminación en el momento de eliminación nominal.

**[0244]** Si SubPicHrdFlag es igual a 0, en el momento de eliminación de CPB de la unidad de acceso n, la unidad de acceso se descodifica instantáneamente.

35 **[0245]** De lo contrario (SubPicHrdFlag es igual a 1), en el momento de eliminación de CPB de la unidad de descodificación m, la unidad de descodificación se descodifica instantáneamente, y cuando la unidad de descodificación m es la última unidad de descodificación de la unidad de acceso n, se aplica lo siguiente:

- 40 - La imagen n se considera como descodificada.  
 - El momento de llegada de CPB final de la unidad de acceso n, es decir,

45 AuFinalArrivalTime[n], se establece igual al tiempo de llegada de CPB final de la última unidad de descodificación en la unidad de acceso n, es decir, DuFinalArrivalTime[m].

- El tiempo de eliminación de CPB nominal de la unidad de acceso n, es decir, AuNominalRemovalTime[n], se establece igual al tiempo de eliminación de CPB nominal de la última unidad de descodificación en la unidad de acceso n, es decir, DuNominalRemovalTime [m].

- 50 - El tiempo de eliminación de CPB de la unidad de acceso n, es decir, AuCpbRemovalTime[m], se establece igual al tiempo de eliminación de CPB de la última unidad de descodificación en la unidad de acceso n, es decir, DuCpbRemovalTime[m].

55 **[0246]** Como se ha descrito con respecto a los cambios en la sección C.2.2, anteriormente, un dispositivo puede llevar a cabo, basándose en parte en un valor (por ejemplo, NoCIRasPicPresentFlag), una operación de CPB. Además, como se describe con respecto a los cambios a la sección C.2.2, anteriormente, cuando el valor indica que no todas las imágenes CL-RAS asociadas con la unidad de acceso IRAP están presentes en el flujo de bits de datos de vídeo y un mensaje SEI de período de almacenamiento en memoria intermedia incluye un primer elemento sintáctico (p. ej., irap\_cross\_layer\_cpb\_params\_present\_flag) que especifica la presencia en el mensaje SEI de período de almacenamiento en memoria intermedia de un segundo elemento sintáctico (por ejemplo, nal\_initial\_alt\_cross\_layer\_cpb\_removal\_delay) y un tercer elemento sintáctico (por ejemplo,

60

nal\_initial\_alt\_cross\_layer\_cpb\_removal\_offset), el dispositivo puede establecer un retardo de eliminación de CPB en un retardo de eliminación alternativo especificado por el segundo elemento sintáctico. Además, el dispositivo puede establecer una desviación de eliminación de CPB en una desviación de eliminación alternativa especificada por el tercer elemento sintáctico.

5  
10  
15  
[0247] Un segundo conjunto de ejemplos de las técnicas de esta divulgación es similar a la primera serie de ejemplos de las técnicas de esta divulgación. Sin embargo, en el segundo conjunto de ejemplos de las técnicas de esta divulgación, se utiliza una estructura sintáctica alternativa para que los parámetros HRD manejen la conformidad de las imágenes CL-RAS. En algunos casos, uno o más ejemplos del primer y segundo conjunto de ejemplos pueden usarse juntos o por separado. Las desviaciones de eliminación de CPB y DPB alternativas se introducen en el mensaje SEI de sincronización de imágenes, en lugar del mensaje SEI de período de almacenamiento en memoria intermedia. Además, se especifica un mensaje SEI adicional que define un conjunto alternativo de velocidad de transmisión de bits y tamaño de CPB para las unidades de acceso IRAP que se pueden usar cuando se eliminan las imágenes CL-RAS asociadas de una unidad de acceso IRAP inicial. Por lo tanto, en algunos ejemplos, un dispositivo (por ejemplo, el codificador de vídeo 20) puede generar un mensaje SEI que define un tamaño de CPB para la unidad de acceso IRAP que se puede utilizar cuando se eliminan las imágenes CL-RAS de una unidad de acceso IRAP inicial.

20  
[0248] En el segundo conjunto de ejemplos de las técnicas de esta divulgación, se pueden realizar los siguientes cambios en la sintaxis del mensaje SEI de sincronización de imágenes del borrador de trabajo de HEVC 10.

|   | Descriptor  |
|---|-------------|
| pic_timing( payloadSize ) {   |             |
| if( frame_field_info_present_flag ) {   |             |
| <b>pic_struct</b>   | u(4)        |
| <b>source_scan_type</b>   | u(2)        |
| <b>duplicate_flag</b>   | u(1)        |
| }   |             |
| if( CpbDpbDelaysPresentFlag ) {   |             |
| <b>irap_cross_layer_cpb_params_present_flag</b>   | <b>u(1)</b> |
| <b>au_cpb_removal_delay_minus1</b>  | u(v)        |
| <b>pic_dpb_output_delay</b>   | u(v)        |
| <b>if( irap_cross_layer_cpb_params_present_flag ) {</b>                                 |             |
| <b>au_alt_cpb_removal_delay_minus1</b>  | <b>u(v)</b> |
| <b>au_alt_pic_dpb_delay</b>   | <b>u(v)</b> |
| }   |             |
| if( sub_pic_hrd_params_present_flag )   |             |
| <b>pic_dpb_output_du_delay</b>  | u(v)        |
| if( sub_pic_hrd_params_present_flag &&<br>sub_pic_cpb_params_in_pic_timing_sei_flag ) { |             |
| <b>num_decoding_units_minus1</b>  | ue(v)       |
| <b>du_common_cpb_removal_delay_flag</b>   | u(1)        |
| if( du_common_cpb_removal_delay_flag )  |             |
| <b>du_common_cpb_removal_delay_increment_minus1</b>                                     | u(v)        |
| for( i = 0; i <= num_decoding_units_minus1; i++ ) {                                     |             |
| <b>num_nalus_in_du_minus1[ i ]</b>  | ue(v)       |
| if( !du_common_cpb_removal_delay_flag && i < num_decoding_units_minus1<br>)             |             |
| <b>du_cpb_removal_delay_increment_minus1[ i ]</b>                                       | u(v)        |
| }   |             |
| }   |             |
| }   |             |
| }   |             |
| }   |             |

[0249] En algunos ejemplos, la señalización de *irap\_cross\_layer\_cpb\_params\_present\_flag* se señala independientemente en el mensaje SEI de período de almacenamiento en memoria intermedia. Además, en algunos ejemplos, los parámetros CPB y DPB alternativos también se señalizan a nivel de unidad de decodificación (DU) cuando *sub\_pic\_hrd\_params\_present\_flag* es igual a 1, para operaciones HRD de sub-imagen para flujos de bits que contienen imágenes CL-RAS y *sub\_pic\_hrd\_params\_present\_flag* es igual a 1 para funcionar adecuadamente para ambos casos cuando las imágenes CL-RAS están presentes o ausentes. Un DU es un subconjunto de una unidad de acceso (cuando *SubPicHrdFlag* es igual a 0, un DU es la AU completa).

[0250] Además, de acuerdo con el segundo conjunto de ejemplos de las técnicas de esta divulgación, la semántica del mensaje SEI de sincronización de imágenes del borrador de trabajo de HEVC 10 se puede cambiar de la forma

siguiente. En este ejemplo, la semántica del resto de elementos sintácticos en el mensaje SEI de sincronización de imágenes puede permanecer igual.

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

10 ***NOTA 2 - No más de uno de los valores de sub\_pic\_hrd\_params\_present\_flag, irap\_cpb\_params\_present\_flag e irap\_cross\_layer\_cpb\_params\_present\_flag pueden ser iguales a 1.***

15 ***au\_cpb\_removal\_delay\_minus1 más 1 especifica el número de tics de reloj entre el tiempo de eliminación de CPB nominal de la unidad de acceso asociada con el mensaje SEI de sincronización de imágenes y la unidad de acceso anterior en orden de descodificación que contenía un mensaje SEI de período de almacenamiento en memoria intermedia. Este valor también puede usarse para calcular un tiempo lo más temprano posible de llegada de datos de la 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 au\_cpb\_removal\_delay\_length\_minus1 + 1.***

20 ***NOTA 6 - El valor de au\_cpb\_removal\_delay\_length\_minus1 que determina la longitud (en bits) del elemento sintáctico au\_cpb\_removal\_deus\_minus1 es el valor de au\_cpb\_removal\_delay\_length\_minus1 codificado en el VPS o el SPS que está activo para la imagen codificada asociada con el mensaje SEI de sincronización de imágenes, aunque au\_cpb\_removal\_delay\_minus1 especifica un número de tics de reloj relativos al tiempo de eliminación de la unidad de acceso anterior que contiene un mensaje SEI de período de almacenamiento en memoria intermedia, que puede ser una unidad de acceso de una CVS diferente.***

25 ***au\_alt\_cpb\_removal\_delay\_minus1 más 1 especifica el número alternativo de tics del reloj entre el tiempo de eliminación de CPB nominal de la unidad de acceso asociada con el mensaje SEI de sincronización de imágenes y la unidad de acceso anterior en orden de descodificación que contenía un mensaje SEI de período de almacenamiento en memoria intermedia. Este valor también puede usarse para calcular un tiempo lo más temprano posible de llegada de datos de la 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 au\_cpb\_removal\_delay\_length\_minus1 + 1.***

30 ***NOTA 6 - El valor de au\_cpb\_removal\_delay\_length\_minus1 que determina la longitud (en bits) del elemento sintáctico au\_cpb\_removal\_deus\_minus1 es el valor de au\_cpb\_removal\_delay\_length\_minus1 codificado en el VPS o el SPS que está activo para la imagen codificada asociada con el mensaje SEI de sincronización de imágenes, aunque au\_cpb\_removal\_delay\_minus1 especifica un número de tics de reloj relativos al tiempo de eliminación de la unidad de acceso anterior que contiene un mensaje SEI de período de almacenamiento en memoria intermedia, que puede ser una unidad de acceso de una CVS diferente.***

35 ***La variable UseAuCpbRemovalDelayMinus1 se obtiene de la forma siguiente:***

- 40 - ***Si cualquiera de las siguientes condiciones es verdadera, UseAuCpbRemovalDelayMinus1 se establece para que sea igual a au\_alt\_cpb\_removal\_delay\_minus1.***
- 45 - ***la unidad de acceso IRAP inicial anterior en orden de descodificación tiene NoCIRasPicPresentFlag igual a 1, irap\_cross\_layer\_cpb\_params\_present\_flag es igual a 1***
- 50 - ***si el AU actual es un IRAP AU inicial que tiene NoCIRasPicPresentFlag igual a 1 e irap\_cross\_layer\_cpb\_params\_present\_flag es igual a 1,***
- 55 - ***De lo contrario, UseAuCpbRemovalDelayMinus1 se establece igual a au\_cpb\_removal\_delay\_minus1.***

60 ***[0251] La variable AuCpbRemovalDelayMsb de la imagen actual se obtiene de la forma siguiente:***

- 65 - ***Si la imagen actual está asociada con un mensaje SEI de período de almacenamiento en memoria intermedia que es aplicable a al menos uno de los puntos de funcionamiento a los que se aplica el mensaje SEI de sincronización de imágenes, AuCpbRemovalDelayMsb se establece igual a 0.***
- 60 - ***De lo contrario, se aplica lo siguiente:***
  - ***Deje que maxCpbRemovalDelay sea igual a  $2^{au\_cpb\_removal\_delay\_length\_minus1 + 1}$ .***
  - ***Permita que prevAuCpbRemovalDelayMinus1 y prevAuCpbRemovalDelayMsb se establezcan iguales a UseAuCpbRemovalDelayMinus1[[au\_cpb\_removal\_delay\_minus1]] y AuCpbRemovalDelayMsb, respectivamente, de la imagen anterior en orden de descodificación que tiene TemporalID igual a 0, que no es***

una imagen sin referencia RASL , RADL o de subcapa, y que está dentro del mismo período de almacenamiento en memoria intermedia que la imagen actual.

- AuCpbRemovalDelayMsb se obtiene de la forma siguiente:

5           if(**UseAuCpbRemovalDelayMinus1**                    [[*au\_cpb\_removal\_delay\_minus1*]])                    <=  
          prevAuCpbRemovalDelayMinus1)

$$\text{AuCpbRemovalDelayMsb} = \text{prevAuCpbRemovalDelayMsb} + \text{maxCpbRemovalDelay} \quad (\text{D-1})$$

10           de lo contrario

$$\text{AuCpbRemovalDelayMsb} = \text{prevAuCpbRemovalDelayMsb}$$

- 15   **[0252]** La variable AuCpbRemovalDelayVal se obtiene de la forma siguiente:

$$\text{AuCpbRemovalDelayVal} = \text{AuCpbRemovalDelayMsb} + \text{UseAuCpbRemovalDelayMinus1} \text{ [[au_cpb_removal_delay_minus1]]} + 1 \quad (\text{D-2})$$

- 20   **[0253]** El valor de AuCpbRemovalDelayVal estará dentro del rango de 1 y 2<sup>32</sup>, inclusive. Dentro de un período de almacenamiento en memoria intermedia, los valores de AuCpbRemovalDelayVal para cualquiera de las dos unidades de acceso no serán los mismos.

25   **pic\_dpb\_output\_delay** se utiliza para calcular el tiempo de salida de DPB de la imagen cuando SubPicHrdFlag es igual a 0. Especifica cuántos tics de reloj esperar después de la eliminación de la última unidad de decodificación de una unidad de acceso de la CPB antes de que la imagen descodificada se emita desde la DPB.

NOTA 7 - Una imagen puede no se elimina de la DPB en su tiempo de emisión cuando todavía está marcada como "utilizada como referencia a corto plazo" o "utilizada como referencia a largo plazo".

- 30   **[0254]** La longitud del elemento sintáctico **pic\_dpb\_output\_delay** viene dada en bits por **dpb\_output\_delay\_length\_minus1 + 1**. Cuando **sps\_max\_dec\_pic\_buffering\_minus1[minTid]** es igual a 0, 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.

35   **pic\_alt\_dpb\_output\_delay** es el valor alternativo utilizado para calcular el tiempo de salida DPB de la imagen cuando **SubPicHrdFlag** es igual a 0 y las imágenes CL-RAS asociadas con la IRAP AU inicia anterior en el orden de decodificación se eliminan del flujo de bits. Especifica cuántos tics de reloj esperar después de la eliminación de la última unidad de decodificación de una unidad de acceso de la CPB antes de que la imagen descodificada se emita desde la DPB.

- 40   **[0255]** La longitud del elemento sintáctico **pic\_alt\_dpb\_output\_delay** viene dada en bits por **dpb\_output\_delay\_length\_minus1 + 1**. Cuando **sps\_max\_dec\_pic\_buffering\_minus1 [minTid]** es igual a 0, 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\_alt\_dpb\_output\_delay** será igual a 0.

- 45   **[0256]** El tiempo de salida obtenido a partir de **dpb\_output\_delay** o **pic\_alt\_dpb\_output\_delay** de cualquier imagen que se emite desde un descodificador conforme con sincronización de salida va a preceder al momento de salida obtenido a partir de **pic\_dpb\_output\_delay** de todas las imágenes en cualquier CVS posterior en orden de decodificación. El orden de salida de imagen establecido por los valores de **[[este elemento sintáctico]]**  
50   **pic\_dpb\_output\_delay** o **pic\_alt\_dpb\_output\_delay** tendrán el mismo orden que el establecido por los valores de **PicOrderCntVal**.

- 55   **[0257]** Para las imágenes que no se transmiten mediante el proceso de "abultamiento" porque preceden, en orden de decodificación, a una imagen IRAP con **NoRasOutputFlag** igual a 1 que tiene **no\_output\_of\_prior\_pics\_flag** igual a 1 o se infiere que es igual 1, los tiempos de salida obtenidos a partir de **pic\_dpb\_output\_delay** o **pic\_alt\_dpb\_output\_delay** aumentarán con el valor creciente de **PicOrderCntVal** relativo a todas las imágenes con la misma CVS.

- 60   **[0258]** Además, en el segundo conjunto de ejemplos de las técnicas de esta divulgación, una sintaxis de mensaje SEI de parámetros HRD de capa cruzada puede definirse de la forma siguiente.

|  | Descriptor   |
|--|--------------|
| <i>cross_layer_hrd_parameters(payloadSize) {</i>             |              |
| <i>cross_layer_video_parameter_set_id</i>                    | <i>ue(v)</i> |
| <i>for( i = 0; i &lt; vps_num_hrd_parameters; i++)</i>       |              |
| <i>for( j = 0; j &lt; vps_max_sub_layers_minus1; j++)</i>    |              |
| <i>for( k = 0; k &lt; CrossLayerCpbCnt[ i ][ j ]; k++) {</i> |              |
| <i>applicable_poc_cnt[ i ][ j ][ k ]</i>                     | <i>u(32)</i> |
| <i>cross_layer_bit_rate_value_minus1[ i ][ j ][ k ]</i>      | <i>ue(v)</i> |
| <i>cross_layer_cpb_size_value_minus1[ i ][ j ][ k ]</i>      | <i>ue(v)</i> |
| <i>}</i>   |              |
| <i>}</i>   |              |

5 **[0259]** De forma alternativa, en algunos ejemplos, también se puede enviar un mensaje SEI de final de HRD de capa  
cruzada excluyendo el elemento sintáctico *applicable\_poc\_cnt[i][j][k]* junto con cada unidad de acceso para denotar  
que los elementos sintácticos *cross\_layer\_bit\_rate\_value\_minus1[i][j][k]* y *cross\_layer\_cpb\_size\_value\_minus1[i][j][k]*  
son aplicables a la unidad de acceso asociada. De forma alternativa, en algunos ejemplos,  
10 *cross\_layer\_cpb\_size\_value\_minus1[i][j][k]* no se señala en el mensaje HRD SEI de capa cruzada. De forma  
alternativa, en algunos ejemplos, la velocidad de transmisión de bits de capa cruzada y el tamaño de CPB se señalizan  
en el VPS (en la parte de extensión, por ejemplo). De forma alternativa, en algunos ejemplos, el mensaje SEI de  
parámetros HRD de capa cruzada también se puede usar con el flujo de bits de la versión 1 de HEVC (es decir, un  
15 flujo de bits de conformidad con el borrador de trabajo de HEVC 10), donde los parámetros adicionales de velocidad  
de transmisión bits y tamaño de CPB son aplicables a la conformidad de flujos de bits donde las imágenes RASL y  
RADL se intercalan, y las imágenes RASL se eliminan del flujo de bits y la imagen IRAP asociada tiene el tipo de  
unidad NAL BLA\_W\_RADL.

**[0260]** De acuerdo con el segundo conjunto de ejemplos de las técnicas de esta divulgación, el mensaje SEI de  
parámetros HRD de capa cruzada puede tener la siguiente semántica.

20 **[0261]** *Este mensaje SEI puede y solo puede estar asociado con IRAP AU.*

*cross\_layer\_video\_parameter\_set\_id* indica y será igual al valor de *vps\_video\_parameter\_set\_id* del VPS al  
que hacen referencia las unidades VCL NAL de la unidad de acceso asociada con el mensaje SEI. El valor de  
25 *cross\_layer\_video\_parameter\_set\_id* deberá estar dentro del rango de 0 y 15, inclusive. La variable  
*CrossLayerCpbCnt[i][j]* se establece igual a *cpb\_cnt\_minus1[i][j] + 1* del *i*-ésimo conjunto de parámetros HRD  
señalados en el VPS.

*aplicables\_poc\_cnt[i][j][k]* especifica la unidad de acceso hasta la cual se aplican  
30 *cross\_layer\_bit\_rate\_value\_minus1[i][j][k]* y *cross\_layer\_cpb\_size\_value\_minus1[i][j][k]*. Si hay una unidad de  
acceso que sigue a la unidad de acceso actual (es decir, la IRAP AU inicial asociada con el mensaje SEI actual)  
en orden de descodificación en la CVS, y esta unidad de acceso contiene una imagen *picA* con *nuh\_layer\_id*  
igual a 0 y tiene *PicOrderCntVal* igual a *PicOrderCntVal* de la imagen en la unidad de acceso actual con  
35 *nuh\_layer\_id* igual a 0 más el valor de *applicable\_poc\_cnt*, la imagen *picA* se conoce como la unidad de acceso  
al punto de actualización de capa cruzada. De lo contrario, la primera unidad de acceso en orden de  
descodificación con imagen que tiene *PicOrderCntVal* mayor que *PicOrderCntVal* de la imagen actual más el  
valor de *recovery\_poc\_cnt* y *nuh\_layer\_id* igual a 0 se conoce como la unidad de acceso al punto de  
40 actualización de capa cruzada. La unidad de acceso al punto de actualización de capa cruzada no debe  
preceder a la unidad de acceso actual en el orden de descodificación. Todas las unidades de acceso  
descodificadas que preceden a la unidad de acceso de punto de actualización de capa cruzada en el orden de  
descodificación se especifican para usar *cross\_layer\_bit\_rate\_value\_minus1[i][j][k]* y  
*cross\_layer\_cpb\_size\_value\_minus1[i][j][k]* cuando *NoCIRasPicPresentFlag* asociado con la IRAP AU inicial  
actual es igual a 0. El valor de *applicable\_poc\_cnt* estará dentro del rango de  $-2^{31}$  a  $2^{31} - 1$ , inclusive.

*cross\_layer\_bit\_rate\_value\_minus1[i][j][k]* (junto con *bit\_rate\_scale*) especifica la velocidad de transmisión de  
45 bits de entrada máxima para el *i*-ésimo CPB cuando la CPB funciona a nivel de la unidad de acceso.  
*cross\_layer\_bit\_rate\_value\_minus1[i][j][k]* estará en el rango de 0 a  $2^{32} - 2$ , ambos inclusive. Para cualquier *k*  
> 0, *cross\_layer\_bit\_rate\_value\_minus1[i][j][k]* será mayor que *cross\_layer\_bit\_rate\_value\_minus1[i][j][k-1]*.

[0262] Cuando *SubPicHrdFlag* es igual a 0, la velocidad de transmisión de bits en bits por segundo viene dada por:

$$\text{CrossLayerBitRate}[i] = (\text{cross\_layer\_bit\_rate\_value\_minus1}[i][j][k] + 1) * 2^{(6 + \text{bit\_rate\_scale})}$$

(E-52)

[0263] Cuando *SubPicHrdFlag* es igual a 0 y el elemento sintáctico *cross\_layer\_bit\_rate\_value\_minus1[i]* no está presente, el valor de *BitRate[i]* se infiere que es igual a *CpbBrVclFactor \* MaxBR* para los parámetros VCL HRD y que es igual a *CpbBrNalFactor \* MaxBR* para los parámetros NAL HRD, donde se especifican *MaxBR*, *CpbBrVclFactor* y *CpbBrNalFactor* en la subcláusula A.4.

*cross\_layer\_cpb\_size\_value\_minus1[i][j][k]* se usa junto con *cross\_layer\_cpb\_size\_scale* para especificar el *i*-ésimo tamaño de CPB cuando la CPB funciona a nivel de la unidad de acceso. *cross\_layer\_cpb\_size\_value\_minus1[i][j][k]* estará en el rango de 0 a  $2^{32} - 2$ , inclusive. Para cualquier *k* mayor que 0, *cross\_layer\_cpb\_size\_value\_minus1[i][j][k]* será menor o igual que *cross\_layer\_cpb\_size\_value\_minus1[i][j][k-1]*.

[0264] Cuando *SubPicHrdFlag* es igual a 0, el tamaño de CPB en bits viene dado por:

$$\text{CrossLayerCpbSize}[i] = (\text{cross\_layer\_cpb\_size\_value\_minus1}[i][j][k] + 1) * 2^{(4 + \text{cpb\_size\_scale})}$$

(E-53)

[0265] Cuando *SubPicHrdFlag* es igual a 0 y el elemento sintáctico *cross\_layer\_cpb\_size\_value\_minus1[i]* no está presente, el valor de *CpbSize[i]* se infiere que es igual a *CpbBrVclFactor \* MaxCPB* para los parámetros VCL HRD y que es igual a *CpbBrNalFactor \* MaxCPB* para los parámetros NAL HRD, donde *MaxCPB*, *CpbBrVclFactor* y *CpbBrNalFactor* se especifican en la subcláusula A.4.

[0266] Por lo tanto, en algunos ejemplos, el descodificador de vídeo 30 puede obtener, a partir de un mensaje SEI de un flujo de bits de vídeo codificado, una velocidad de transmisión de bits alternativa y parámetros de tamaño de CPB para las unidades de acceso IRAP. Para cada una respectiva de las unidades de acceso IRAP, el descodificador de vídeo 30 puede usar la velocidad de transmisión de bits alternativa y los parámetros de tamaño de CPB para la unidad de acceso IRAP respectiva cuando la unidad de acceso IRAP respectiva es una unidad de acceso IRAP inicial con imágenes CL-RAS asociadas no presentes. Además, en algunos ejemplos, un dispositivo (por ejemplo, el codificador de vídeo 20) puede generar un mensaje SEI que incluye un elemento sintáctico que indica una unidad de acceso hasta la cual son aplicables un valor de velocidad de transmisión de bits de capa cruzada y un valor de tamaño de CPB de capa cruzada. Además, en algunos ejemplos, un dispositivo (por ejemplo, un codificador de vídeo 20) puede generar un mensaje SEI que incluye un elemento sintáctico que especifica una velocidad de transmisión de bits de entrada máxima de una CPB cuando la CPB funciona a un nivel de unidad de acceso.

[0267] Además, en el segundo conjunto de ejemplos de las técnicas de esta divulgación, se pueden realizar los siguientes cambios en el Anexo C, Sección C.1, del Borrador de trabajo de HEVC 10.

### C.1 General

[0268] Este anexo especifica el descodificador de referencia hipotético (HRD) y su uso para comprobar el flujo de bits y la conformidad del descodificador.

[0269] Dos tipos de flujos de bits o subconjuntos de flujos de bits están sujetos a la comprobación de conformidad de HRD para esta memoria descriptiva. El primer tipo, llamado un flujo de bits de Tipo I, es un flujo de unidades NAL que contiene solo las unidades VCL NAL y las 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, denominado flujo de bits Tipo II, contiene, además de las unidades VCL NAL y las unidades NAL de datos de relleno para todas las unidades de acceso en el flujo de bits, al menos uno de los siguientes:

- unidades NAL no VCL adicionales 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 bytes a partir del flujo de la unidad NAL (como se especifica en el Anexo B).

[0270] La figura C-1 muestra los tipos de puntos de conformidad del flujo de bits comprobados por el HRD.

**[0271]** Los elementos sintácticos de unidades no VCL NAL (o sus valores predeterminados para algunos de los elementos sintácticos), requeridos para el HRD, se especifican en las subcláusulas semánticas de la cláusula 7, Anexos D y E.

5 **[0272]** Se usan dos tipos de conjuntos de parámetros de HRD (parámetros NAL HRD y parámetros VCL HRD). Los conjuntos de parámetros de HRD se señalizan a través de la estructura sintáctica `hrd_parameters()`, que puede ser parte de la estructura sintáctica de SPS o de la estructura sintáctica de VPS.

10 **[0273]** Se pueden necesitar múltiples pruebas para comprobar la conformidad de un flujo de bits, que se conoce como el flujo de bits bajo prueba. Para cada prueba, se aplican los siguientes pasos en el orden indicado:

15 1. Se selecciona un punto de funcionamiento bajo prueba, indicado como `TargetOp`. La lista de identificadores de capa `OpLayerIdList` de `TargetOp` consta de la lista de valores de `nuh_layer_id`, en orden creciente de los valores de `nuh_layer_id`, presentes en el subconjunto del flujo de bits asociado con `TargetOp`, que es un subconjunto de los valores de `nuh_layer_id` presentes en el flujo de bits bajo prueba. El `OpTid` de `TargetOp` es igual al `TemporalId` más alto presente en el subconjunto del flujo de bits asociado con `TargetOp`.

20 2. `TargetDecLayerIdList` se establece igual a `OpLayerIdList` de `TargetOp`, `HighestTid` se establece igual a `OpTid` de `TargetOp`, y el proceso de extracción del sub-flujo de bits especificado en la cláusula 10 se invoca con el flujo de bits sometido a prueba, `HighestTid` y `TargetDecLayerIdList` como entradas, y la salida se asigna a `BitstreamToDecode`.

25 3. La estructura sintáctica `hrd_parameters()` y la estructura sintáctica `sub_layer_hrd_parameters()` aplicable a `TargetOp` se seleccionan. Si `TargetDecLayerIdList` contiene todos los valores `nuh_layer_id` presentes en el flujo de bits bajo prueba, se selecciona la estructura sintáctica `hrd_parameters()` en el SPS activo (o se proporciona a través de medios externos no especificados en esta memoria descriptiva). De lo contrario, se selecciona la estructura sintáctica `hrd_parameters()` en el VPS activo (o proporcionado a través de medios externos no especificados en esta memoria descriptiva) que se aplica a `TargetOp`. Dentro de la estructura sintáctica `hrd_parameters()` seleccionada, si `BitstreamToDecode` es un flujo de bits de Tipo I, se selecciona la estructura sintáctica que sigue inmediatamente a la condición `"if(vcl_hrd_parameters_present_flag)"` y la variable `NalHrdModeFlag` se establece igual a 0; de lo contrario, (`BitstreamToDecode` es un flujo de bits de Tipo II), se selecciona la estructura sintáctica `sub_layer_hrd_parameters(HighestTid)` que sigue inmediatamente a la condición `"if(vcl_hrd_parameters_present_flag)"` (en este caso, la variable `NalHrdModeFlag` se establece igual a 0) o la condición `"if(nal_hrd_parameters_present_flag)"` (en este caso, la variable `NalHrdModeFlag` se establece igual a 1). Cuando `BitstreamToDecode` es un flujo de bits Tipo II y `NalHrdModeFlag` es igual a 0, todas las unidades NAL no VCL excepto las unidades NAL de datos de relleno y todos los elementos sintácticos `leading_zero_8bits`, `zero_byte`, `start_code_prefix_one_3bytes`, y `trailing_zero_8bits` que forman un flujo de bytes a partir del flujo de la unidad NAL (como se especifica en el Anexo B), cuando están presentes, se descartan de `BitstreamToDecode`, y el flujo de bits restante se asigna a `BitstreamToDecode`.

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

45 5. Para cada unidad de acceso en `BitstreamToDecode` comenzando desde la unidad de acceso 0, se selecciona el mensaje SEI de período de almacenamiento en memoria intermedia (presente en `BitstreamToDecode` o disponible a través de medios externos no especificados en esta memoria descriptiva) que está asociado con la unidad de acceso y se aplica a `TargetOp`, se selecciona el mensaje SEI de sincronización de imágenes (presente en `BitstreamToDecode` o disponible a través de medios externos no especificados en esta memoria descriptiva) que está asociado con la unidad de acceso y se aplica a `TargetOp`, y cuando `SubPicHrdFlag` es igual a 1 y `sub_pic_cpb_params_in_pic_timing_sei_flag` es igual a 0, se seleccionan los mensajes SEI de información de la unidad de descodificación (presentes en `BitstreamToDecode` o disponibles a través de medios externos no especificados en esta memoria descriptiva) que están asociados con unidades de descodificación en la unidad de acceso y se aplican a `TargetOp`.

55 6. Se selecciona un valor de `SchedSelIdx`. El `SchedSelIdx` seleccionado estará en el rango de 0 a `cpb_cnt_minus1[HighestTid]`, inclusive, donde `cpb_cnt_minus1[HighestTid]` se encuentra en la estructura sintáctica `sub_layer_hrd_parameters(HighestTid)` como se seleccionó anteriormente.

60 7. Cuando la imagen codificada en la unidad de acceso 0 tiene `nal_unit_type` igual a `CRA_NUT` o `BLA_W_LP`, e `irap_cpb_params_present_flag` en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado es igual a 1, cualquiera de las siguientes opciones se aplica para la selección de la desviación de retardo y el retardo de eliminación de CPB:

65

- 5 - Si `NalHrdModeFlag` es igual a 1, la desviación de retardo y el retardo de eliminación de CPB inicial predeterminados representados por `nal_initial_cpb_removal_delay [SchedSelIdx]` y `nal_initial_cpb_removal_offset [SchedSelIdx]`, respectivamente, se seleccionan en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado. De lo contrario, se selecciona la desviación de retardo y el retardo de eliminación de CPB inicial predeterminados representados por `vcl_initial_cpb_removal_delay[SchedSelIdx]` y `vcl_initial_cpb_removal_offset[SchedSelIdx]`, respectivamente, en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado. La variable `DefaultInitCpbParamsFlag` se establece igual a 1.
- 10 - Si `NalHrdModeFlag` es igual a 1, se selecciona la desviación de retardo y el retardo de eliminación de CPB inicial alternativa representados por `nal_initial_alt_cpb_removal_delay[SchedSelIdx]` y `nal_initial_alt_cpb_removal_offset[SchedSelIdx]`, respectivamente, en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado. De lo contrario, se selecciona la desviación de retardo y el retardo de eliminación de CPB inicial alternativa representados por `vcl_initial_alt_cpb_removal_delay[SchedSelIdx]` y `vcl_initial_alt_cpb_removal_offset[SchedSelIdx]`, respectivamente, en el mensaje SEI de período de almacenamiento en memoria intermedia seleccionado. La variable `DefaultInitCpbParamsFlag` se establece igual a 0, y las unidades de acceso RASL asociadas con la unidad de acceso 0 se descartan de `BitstreamToDecode` y el flujo de bits restante se asigna a `BitstreamToDecode`.
- 20 **8. Cuando `NoCIRasPicPresentFlag` es igual a 1 y la unidad de acceso 0 está asociada con un mensaje SEI de parámetros HRD de capa cruzada, los valores de `BitRate[SchedSelIdx]` y `CpbSize[SchedSelIdx]` se establecen iguales a `CrossLayerBitRate[SchedSelIdx]` y a `CrossLayerCpbSize[SchedSelIdx]`, respectivamente.**
- 25 9. Cuando `sub_pic_hrd_params_present_flag` en la estructura sintáctica `hrd_parameters()` seleccionada es igual a 1, la CPB se programa para funcionar a nivel de la unidad de acceso (en cuyo caso la variable `SubPicHrdFlag` se establece en 0) o a nivel de sub-imagen (en cuyo caso la variable `SubPicHrdFlag` se establece igual a 1).
- 30 **[0274]** Para cada punto de funcionamiento bajo prueba, el número de pruebas de conformidad del flujo de bits que se realizará es  $n_0 * n_1 * (n_2 * 2 + n_3) * n_4$ , donde los valores de  $n_0$ ,  $n_1$ ,  $n_2$ ,  $n_3$  y  $n_4$  se especifican de la forma siguiente:
  - $n_0$  se obtiene de la forma siguiente:
    - 35 - Si `BitstreamToDecode` es un flujo de bits de Tipo I,  $n_0$  es igual a 1.
    - De lo contrario (`BitstreamToDecode` es un flujo de bits de Tipo II),  $n_0$  es igual a 2.
  - $n_1$  es igual a `cpb_cnt_minus1[HighestTid] + 1`.
  - 40 -  $n_2$  es el número de unidades de acceso en `BitstreamToDecode` que cada una está asociada con un mensaje SEI de período de almacenamiento en memoria intermedia aplicable a `TargetOp` y para cada una de las cuales se cumplen las dos condiciones siguientes:
    - 45 - `nal_unit_type` es igual a `CRA_NUT` o `BLA_W_LP` para las unidades VCL NAL;
    - El mensaje SEI de período de almacenamiento en memoria intermedia asociado aplicable a `TargetOp` tiene `irap_cpb_params_present_flag` igual a 1.
  - 50 -  $n_3$  es el número de unidades de acceso en `BitstreamToDecode` que cada una está asociada con un mensaje SEI de período de almacenamiento en memoria intermedia aplicable a `TargetOp` y para cada una de las cuales una o las dos condiciones siguientes son verdaderas:
    - 55 - `nal_unit_type` no es igual a `CRA_NUT` ni a `BLA_W_LP` para las unidades VCL NAL;
    - El mensaje SEI de período de almacenamiento en memoria intermedia asociado aplicable a `TargetOp` tiene `irap_cpb_params_present_flag` igual a 0.
  - $n_4$  se obtiene de la forma siguiente:
    - 60 - Si `sub_pic_hrd_params_present_flag` en la estructura sintáctica `hrd_parameters()` seleccionada es igual a 0,  $n_4$  es igual a 1;
    - De lo contrario,  $n_4$  es igual a 2.
- 65 **[0275]** Cuando `BitstreamToDecode` es un flujo de bits de Tipo II, se aplica lo siguiente:

5 - Si se selecciona la estructura sintáctica `sub_layer_hrd_parameters(HighestTid)` que sigue inmediatamente a 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 las unidades NAL de datos de relleno y VCL se cuentan para la velocidad de transmisión de bits de entrada y el almacenamiento de CPB.

10 - De lo contrario, (se selecciona la estructura sintáctica `sub_layer_hrd_parameters(HighestTid)` que sigue inmediatamente a la condición "`if(nal_hrd_parameters_present_flag)`"), la prueba se realiza en el punto de conformidad de Tipo II que se muestra en la Figura C-1, y todos los bytes del flujo de bits Tipo II, que puede ser un flujo de unidad NAL o un flujo de bytes, se cuentan para la velocidad de transmisión de bits de entrada y el almacenamiento de CPB.

15 NOTA 1 - Los parámetros NAL HRD 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 VCL HRD para el punto de conformidad de Tipo I mostrado en la Figura C-1 para los mismos valores de `InitCpbRemovalDelay[SchedSelIdx]`, `BitRate[SchedSelIdx]` (**o `CrossLayerBitRate [SchedSelIdx]` cuando corresponda**), y `CpbSize [SchedSelIdx]` (**o `CrossLayerCpbSize[SchedSelIdx]` cuando corresponda**) 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 se programe la siguiente imagen para comenzar a llegar. Por ejemplo, cuando se descodifica una CVS conforme a uno o más de los perfiles especificados en el Anexo A utilizando el proceso de descodificación especificado en las cláusulas 2 a 10, cuando se proporcionan parámetros NAL HRD para el punto de conformidad de Tipo II que no solo se encuentran dentro de los límites establecidos para los parámetros NAL HRD para la conformidad del perfil en el punto f) de la subcláusula A.4.2, sino que también se encuentran dentro de los límites establecidos para los parámetros VCL HRD para la conformidad del perfil en el punto e) de la subcláusula A.4.2, la conformidad del VCL HRD para el punto la conformidad de Tipo I también se garantiza que se encuentra dentro de los límites del punto e) de la subcláusula A.4.2.

30 **[0276]** Todos los VPS, SPS y PPS a los que se hace referencia en las unidades VCL NAL, y los correspondientes mensajes SEI de período de almacenamiento en memoria intermedia, información de la unidad de descodificación y sincronización de imágenes 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 intermedia.

35 **[0277]** En los Anexos C, D y E, la especificación de "presencia" de unidades NAL que no son VCL que contienen mensajes SEI de período de almacenamiento en memoria intermedia, VPS, SPS, PPS, mensajes SEI de sincronización de imágenes o mensajes SEI de información de unidad de descodificación también se satisface cuando esas unidades NAL (o solo algunas de ellas) se transmiten a los descodificadores (o al HRD) por otros medios no especificados en 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.

40 NOTA 2 - Como ejemplo, la sincronización de tal 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.

45 **[0278]** Cuando el contenido de tal 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.

50 NOTA 3 - 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 basándose únicamente en la información contenida en el flujo de bits. Cuando la información del HRD no está presente en el flujo de bits, como es el caso de todos los flujos de bits Tipo I "independientes", la conformidad solo puede verificarse cuando los datos del HRD se suministran por algún otro medio no especificado en esta memoria descriptiva.

55 **[0279]** 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.

60 **[0280]** Para cada prueba de conformidad del flujo de bits, el tamaño de DPB (número de memorias intermedias de almacenamiento de imagen) es `sps_max_dec_pic_buffering_minus1[HighestTid] + 1`.

65 **[0281]** La variable `El SubPicHrdPreferredFlag` se especifica ya sea por medios externos, o cuando no se especifica por medios externos, se establece igual a 0.

**[0282]** Cuando el valor de la variable SubPicHrdFlag no se ha establecido en el paso 8 anterior en esta subcláusula, se obtiene de la forma siguiente:

$$\text{SubPicHrdFlag} = \text{SubPicHrdPreferredFlag} \ \&\& \\ \text{sub\_pic\_hrd\_params\_present\_flag}(C-1)$$

5

**[0283]** Si SubPicHrdFlag es igual a 0, el HRD 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 sub-imagen, y cada unidad de descodificación es un subconjunto de una unidad de acceso.

10 **NOTA 4** - Si el HRD funciona a nivel de la unidad de acceso, cada vez se elimina de la CPB una unidad de descodificación que es una unidad de acceso completa. De lo contrario (el HRD funciona a nivel de sub-imagen), cada vez se elimina de la CPB una unidad de descodificación que es un subconjunto de una unidad de acceso. En ambos casos, cada vez se emite una imagen descodificada completa desde la DPB, aunque el tiempo de salida de la imagen se obtiene basándose en los tiempos de eliminación de CPB obtenidos de forma diferente y los retardos de salida de DPB señalizados de forma diferente.

15

**[0284]** Se especifica lo siguiente para expresar las restricciones en este anexo:

20 - Cada unidad de acceso se denomina unidad de acceso n, donde el número n identifica la unidad de acceso particular. La unidad de acceso 0 se selecciona según el paso 4 anterior. El valor de n puede aumentarse en 1 para cada unidad de acceso posterior en el orden de descodificación.

25 - Cada unidad de descodificación puede denominarse unidad de descodificación m, donde el número m identifica la unidad de descodificación particular. La primera unidad de descodificación en orden de descodificación en la unidad de acceso 0 se denomina unidad de descodificación 0. El valor de m se aumenta en 1 para cada unidad de descodificación posterior en orden de descodificación.

30 **NOTA 5** - La numeración de las unidades de descodificación es relativa a la primera unidad de descodificación en la unidad de acceso 0.

- La imagen n se refiere a la imagen codificada o la imagen descodificada de la unidad de acceso n. El DRH funciona de la forma siguiente:

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

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

40 - Los datos asociados con las unidades de descodificación que fluyen en la CPB de acuerdo con una programación de llegada especificada son entregados por el HSS.

45 - 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.

50 - Se elimina una imagen descodificada de la DPB cuando ya no es necesaria para la referencia de predicción inter y ya no es necesaria para la salida.

**[0285]** Para cada prueba de conformidad de flujo de bits, la operación de la CPB se especifica en la subcláusula C.2, la operación de descodificador instantánea se especifica en las cláusulas 2 a 10, se especifica el funcionamiento de la DPB en la subcláusula C.3, y el recorte de salida se especifica en la subcláusula C.3.3 y en la subcláusula C.5.2.2.

55

**[0286]** La información de HSS y HRD concerniente al número de programas de entrega enumerados y sus velocidad de transmisión de bits y tamaños de memorias intermedias asociados se especifican en las subcláusulas E.1.2 y E.2.2. El HRD se inicializa como se especifica en el mensaje SEI período de almacenamiento en memoria intermedia especificado en subcláusulas D.2.2 y D.3.2. El momento eliminación de unidades de descodificación de la CPB y la sincronización de salida de imágenes descodificadas desde la DPB se especifica mediante información en los mensajes SEI de sincronización de imágenes (especificados en subcláusulas D.2.3 y D.3.3) o en los mensajes SEI de información de la unidad de descodificación (especificados en las subcláusulas D.2.21 y D.3.21). Toda la información de sincronización relacionada con una unidad de descodificación específica deberá llegar antes del tiempo de eliminación de CPB de la unidad de descodificación.

60

**[0287]** Los requisitos para la conformidad de flujo de bits se especifican en la subcláusula C.4, y el HRD se utiliza para comprobar la conformidad de los flujos de bits como se especifica anteriormente en esta subcláusula y para comprobar la conformidad de los descodificadores como se especifica en la subcláusula C.5.

5  
NOTA 7 - Si bien la conformidad está garantizada bajo el supuesto de que todas las velocidades de imagen y los relojes utilizados para generar el flujo de bits coinciden exactamente con los valores señalados en el flujo de bits, en un sistema real, cada uno de estos puede variar respecto al valor señalado o especificado.

10 **[0288]** 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.

15 **[0289]** La variable ClockTick se obtiene de la forma siguiente y se denomina tic de reloj:

$$\text{ClockTick} = \text{vui\_num\_units\_in\_tick} \div \text{vui\_time\_scale} \quad (\text{C-2})$$

**[0290]** La variable ClockSubTick se obtiene de la forma siguiente y se llama sub-tic del reloj:

20 
$$\text{ClockSubTick} = \text{ClockTick} \div (\text{tick\_divisor\_minus2} + 2) \quad (\text{C-3})$$

**[0291]** Además, en el segundo conjunto de ejemplos de las técnicas de esta divulgación, se pueden realizar los siguientes cambios en el Anexo C, Sección C.2.2, del Borrador de trabajo de HEVC 10.

25 **C.2.2 Sincronización de llegada de la unidad de descodificación**

**[0292]** Si SubPicHrdFlag es igual a 0, la variable subPicParamsFlag se establece en 0, y el proceso especificado en el resto de esta subcláusula se invoca con una unidad de descodificación considerada como una unidad de acceso, para la obtención de los tiempos de llegada de CPB inicial y final para la unidad de acceso n.

30 **[0293]** De lo contrario (SubPicHrdFlag es igual a 1), el proceso especificado en el resto de esta subcláusula se invoca primero con la variable subPicParamsFlag establecida igual a 0 y una unidad de descodificación se considera como una unidad de acceso, para la obtención de los tiempos de llegada de CPB inicial y final para la unidad de acceso n, y luego se invocan con subPicParamsFlag igual a 1 y una unidad de descodificación se considera como un subconjunto de una unidad de acceso, para la obtención de los tiempos de llegada de CPB inicial y final para las unidades de descodificación en la unidad de acceso n.

35 **[0294]** Las variables InitCpbRemovalDelay[SchedSelIdx] e InitCpbRemovalDelayOffset[SchedSelIdx] se obtienen de la forma siguiente:

40 - Si una o más de las siguientes condiciones son verdaderas, InitCpbRemovalDelay[SchedSelIdx] e InitCpbRemovalDelayOffset[SchedSelIdx] se establecen iguales a los valores de los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria intermedia nal\_initial\_alt\_cpb\_removal\_delay[SchedSelIdx] y nal\_initial\_alt\_cpb\_removal\_offset[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 1, o vcl\_initial\_alt\_cpb\_removal\_delay[SchedSelIdx] y vcl\_initial\_alt\_cpb\_removal\_offset[SchedSelIdx], respectivamente, cuando NalHrdModeFlag es igual a 0, donde los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria intermedia se seleccionan como se especifica en la subcláusula C.1:

50 - La unidad de acceso 0 es una unidad de acceso BLA para la cual la imagen codificada tiene nal\_unit\_type igual a BLA\_W\_DLP o BLA\_N\_LP, y el valor de irap\_cpb\_params\_present\_flag del mensaje SEI de período de almacenamiento en memoria intermedia es igual a 1.

55 - La unidad de acceso 0 es una unidad de acceso BLA para la cual la imagen codificada tiene nal\_unit\_type igual a BLA\_W\_LP o es una unidad de acceso CRA, y el valor de irap\_cpb\_params\_present\_flag del mensaje SEI de período de almacenamiento en memoria intermedia es igual a 1, y una más de las siguientes condiciones se cumplen:

60 - UseAltCpbParamsFlag para la unidad de acceso 0 es igual a 1.

- DefaultInitCpbParamsFlag es igual a 0.

- El valor de subPicParamsFlag es igual a 1.

65 - De lo contrario, InitCpbRemovalDelay[SchedSelIdx] e InitCpbRemovalDelayOffset[SchedSelIdx] se establecen iguales a los valores de los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria

intermedia  $nal\_initial\_cpb\_removal\_delay[SchedSelIdx]$  y  $nal\_initial\_cpb\_removal\_offset[SchedSelIdx]$ , respectivamente, cuando  $NalHrdModeFlag$  es igual a 1, o  $vcl\_initial\_cpb\_removal\_delay[SchedSelIdx]$  y  $vcl\_initial\_cpb\_removal\_offset[SchedSelIdx]$ , respectivamente, cuando  $NalHrdModeFlag$  es igual a 0, donde los elementos sintácticos de mensaje SEI de período de almacenamiento en memoria intermedia se seleccionan como se especifica en la subcláusula C.1.

**[0295]** El momento en el que el primer bit de la unidad de descodificación  $m$  comienza a entrar en la CPB puede denominarse el tiempo de llegada inicial  $initArrivalTime[m]$ .

**[0296]** El tiempo de llegada inicial de la unidad de descodificación  $m$  se obtiene de la forma siguiente:

- Si la unidad de descodificación es la unidad de descodificación 0 (es decir,  $m = 0$ ),  $initArrivalTime[0] = 0$ .
- De otro modo (la unidad de descodificación es la unidad de descodificación  $m$  con  $m > 0$ ), se aplica lo siguiente:
  - Si  $cbr\_flag[SchedSelIdx]$  es igual a 1, el tiempo de llegada inicial para la unidad de descodificación  $m$ , es igual al tiempo de llegada final (que se obtiene a continuación) de la unidad de descodificación  $m - 1$ , es decir,

$$\begin{aligned} & \text{if}( !subPicParamsFlag ) \\ & \quad \text{initArrivalTime}[ m ] = \\ & \quad \text{AuFinalArrivalTime}[ m - 1 ] \end{aligned} \quad \text{(C-4)}$$

de lo contrario

$$\begin{aligned} & \quad \text{initArrivalTime}[ m ] = \\ & \quad \text{DuFinalArrivalTime}[ m - 1 ] \end{aligned}$$

- De otro modo ( $cbr\_flag[SchedSelIdx]$  es igual a 0), el tiempo de llegada inicial para la unidad de descodificación  $m$  se obtiene de la forma siguiente:

$\text{if}(!subPicParamsFlag)$

$$\begin{aligned} & \quad \text{initArrivalTime}[ m ] = \text{Max}( \\ & \quad \text{AuFinalArrivalTime}[ m - 1 ], \text{initArrivalEarliestTime}[ m ] ) \end{aligned} \quad \text{(C-5)}$$

de lo contrario

$$\begin{aligned} & \quad \text{initArrivalTime}[ m ] = \text{Max}( \\ & \quad \text{DuFinalArrivalTime}[ m - 1 ], \text{initArrivalEarliestTime}[ m ] ) \end{aligned}$$

donde  $initArrivalEarliestTime[m]$  se obtiene de la forma siguiente:

- La variable  $tmpNominalRemovalTime$  se obtiene de la forma siguiente:

$\text{if}(!subPicParamsFlag)$

$$\begin{aligned} & \quad \text{tmpNominalRemovalTime} = \\ & \quad \text{AuNominalRemovalTime}[ m ] \end{aligned} \quad \text{(C-6)}$$

de lo contrario

$$\begin{aligned} & \quad \text{tmpNominalRemovalTime} = \\ & \quad \text{DuNominalRemovalTime}[ m ] \end{aligned}$$

donde  $AuNominalRemovalTime[m]$  y

$DuNominalRemovalTime[m]$  es el tiempo de eliminación de CPB nominal de la unidad de acceso  $m$  y la unidad de descodificación  $m$ , respectivamente, como se especifica en la subcláusula C.2.3.

- Si la unidad de descodificación m no es la primera unidad de descodificación de un período de almacenamiento en memoria intermedia posterior,  $\text{initArrivalEarliestTime}[m]$  se obtiene de la forma siguiente:

$$\text{initArrivalEarliestTime}[m] = \text{tmpNominalRemovalTime} - \left( \text{InitCpbRemovalDelay}[\text{SchedSelIdx}] + \text{InitCpbRemovalDelayOffset}[\text{SchedSelIdx}] \right) \div 90000 \quad (\text{C-7})$$

- De otro modo (la unidad de descodificación m es la primera unidad de descodificación de un período de almacenamiento en memoria intermedia posterior),  $\text{initArrivalEarliestTime}[m]$  se obtiene de la forma siguiente:

$$\text{initArrivalEarliestTime}[m] = \text{tmpNominalRemovalTime} - \left( \text{InitCpbRemovalDelay}[\text{SchedSelIdx}] \div 90000 \right) \quad (\text{C-8})$$

[0297] El tiempo de llegada final para la unidad de descodificación m se obtiene de la forma siguiente:

if(!subPicParamsFlag)

$$\text{AuFinalArrivalTime}[m] = \text{initArrivalTime}[m] + \text{sizeInbits}[m] \div$$

$$\text{BitRate}[\text{SchedSelIdx}] \quad (\text{C-9})$$

de lo contrario

$$\text{DuFinalArrivalTime}[m] = \text{initArrivalTime}[m] + \text{sizeInbits}[m] \div$$

$$\text{BitRate}[\text{SchedSelIdx}]$$

donde  $\text{sizeInbits}[m]$  es el tamaño en bits de la unidad de descodificación m, contando los bits de las unidades VCL NAL y las unidades NAL de datos de relleno para el punto de conformidad de Tipo I o todos los bits del flujo de bits de Tipo II para el punto de la conformidad de Tipo II, donde los puntos de conformidad de Tipo I y Tipo II son los que se muestran en la figura C-1.

[0298] Además, los valores de  $\text{SchedSelIdx}$ ,  $\text{BitRate}[\text{SchedSelIdx}]$ , y  $\text{CpbSize}[\text{SchedSelIdx}]$  pueden limitarse de la forma siguiente:

- Si el contenido de las estructuras sintácticas  $\text{hrd\_parameters}()$  seleccionadas (**o el mensaje SEI de parámetros HRD de capa cruzada, cuando sea aplicable**) para la unidad de acceso que contiene la unidad de descodificación m y la unidad de acceso anterior difieren, el HSS selecciona un valor  $\text{SchedSelIdx1}$  de  $\text{SchedSelIdx}$  de entre los valores de  $\text{SchedSelIdx}$  proporcionados en las estructuras sintácticas  $\text{hrd\_parameters}()$  seleccionadas (**o el mensaje SEI de parámetros HRD de capa cruzada, cuando se aplicable**) para la unidad de acceso que contiene la unidad de descodificación m que se traduce en  $\text{BitRate}[\text{SchedSelIdx1}]$  o  $\text{CpbSize}[\text{SchedSelIdx1}]$  para la unidad de acceso que contiene la unidad de descodificación m. El valor de  $\text{BitRate}[\text{SchedSelIdx1}]$  o  $\text{CpbSize}[\text{SchedSelIdx1}]$  puede diferir del valor de  $\text{BitRate}[\text{SchedSelIdx0}]$  o  $\text{CpbSize}[\text{SchedSelIdx0}]$  para el valor de  $\text{SchedSelIdx0}$  de  $\text{SchedSelIdx}$  que estaba en uso para la unidad de acceso anterior.
- De otro modo, el HSS continúa funcionando con los valores anteriores de  $\text{SchedSelIdx}$ ,  $\text{BitRate}[\text{SchedSelIdx}]$  y  $\text{CpbSize}[\text{SchedSelIdx}]$ .

[0299] Cuando el HSS selecciona los valores de  $\text{BitRate}[\text{SchedSelIdx}]$  o  $\text{CpbSize}[\text{SchedSelIdx}]$  que difieren de los de la unidad de acceso anterior, se aplica lo siguiente:

- La variable  $\text{BitRate}[\text{SchedSelIdx}]$  entra en vigencia en el tiempo llegada de CPB inicial de la unidad de acceso actual.
- La variable  $\text{CpbSize}[\text{SchedSelIdx}]$  entra en vigor de la forma siguiente:
  - Si el nuevo valor de  $\text{CpbSize}[\text{SchedSelIdx}]$  es mayor que el tamaño de CPB anterior, entra en vigencia en el momento de llegada de CPB inicial de la unidad de acceso actual.

- De otro modo, el nuevo valor de CpbSize[SchedSelIdx] entra en vigor en el momento de la eliminación de la CPB de la unidad de acceso actual.

5 **[0300]** Además, en el segundo conjunto de ejemplos de las técnicas de esta divulgación, se pueden realizar los siguientes cambios en el Anexo C, Sección C.3.3, del Borrador de trabajo de HEVC 10. Los cambios pueden dar lugar a que el decodificador de vídeo 30 realice, basándose al menos en parte en un valor que indica si todas las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de datos de vídeo (por ejemplo, NoCIRasPicPresentFlag), un proceso de salida de imagen.

### 10 C.3.3 Salida de imágenes

**[0301]** Los procesos especificados en esta subcláusula ocurren instantáneamente en el momento de eliminación de CPB de la unidad de acceso n, AuCpbRemovalTime[n].

15 **[0302]** Cuando la imagen n tiene PicOutputFlag igual a 1, su tiempo de salida de DPB DpbOutputTime[n] se obtiene de la forma siguiente, donde la variable firstPicInBufferingPeriodFlag es igual a 1 si la unidad de acceso n es la primera unidad de acceso de un período de almacenamiento en memoria intermedia y 0 de lo contrario:

```

20         if(!SubPicHrdFlag) {
                DpbOutputTime[ n ] = AuCpbRemovalTime[ n ] + ClockTick *
                picDpbOutputDelay          (C-16)
        if(firstPicInBufferingPeriodFlag)
25                 DpbOutputTime[ n ] -= ClockTick * DpbDelayOffset
        } de lo contrario
                DpbOutputTime[ n ] = AuCpbRemovalTime[ n ] + ClockSubTick *
30                 picSptDpbOutputDuDelay

```

35 donde *[[picDpbOutputDelay es el valor de pic \_ dpb \_ output\_delay en el mensaje SEI de sincronización de imágenes asociado con la unidad de acceso n, y]]* picSptDpbOutputDuDelay es el valor de pic\_spt\_dpb\_output\_du\_delay, cuando está presente, en los mensajes SEI de información de la unidad de descodificación asociados con la unidad de acceso n, o el valor de pic\_dpb\_output\_du\_delay en el mensaje SEI de sincronización de imágenes asociado con la unidad de acceso n cuando no hay ningún mensaje SEI de información de unidad de descodificación asociado con la unidad de acceso n o ningún mensaje SEI de información de unidad de descodificación asociado con la unidad de acceso n tiene presente pic\_spt\_dpb\_output\_du\_de\_delay. **El valor depicDpbOutputDelay se establece de la forma siguiente:**

- 40 - **Si cualquiera de las siguientes condiciones es verdadera, picDpbOutputDelay se elige como el valor de pic\_alt\_dpb\_output\_delay en el mensaje SEI de sincronización de imágenes asociado con la unidad de acceso n.**
  - 45 - **la IRAP AU inicial anterior en orden de descodificación tiene NoCIRasPicPresentFlag igual a 1 e irap\_cross\_layer\_cpb\_params\_present\_flag es igual a 1.**
  - **la AU actual es una IRAP AU inicial que tiene NoCIRasPicPresentFlag igual a 1 e irap\_cross\_layer\_cpb\_params\_present\_flag es igual a 1.**
- 50 - **De lo contrario, picDpbOutputDelay es el valor de pic\_alt\_dpb\_output\_delay en el mensaje SEI de sincronización de imágenes asociado con la unidad de acceso n.** NOTA - Cuando el elemento sintáctico pic\_spt\_dpb\_output\_du\_delay no está presente en ningún mensaje SEI de información de unidad de descodificación asociado con la unidad de acceso n, se infiere que el valor es igual a pic\_dpb\_output\_du\_delay en el mensaje SEI de sincronización de imágenes asociado con la unidad de acceso n.

55 **[0303]** La salida de la imagen actual se especifica de la forma siguiente:

- Si PicOutputFlag es igual a 1 y DpbOutputTime[n] es igual a AuCpbRemovalTime[n], se emite la imagen actual.
- 60 - De otro modo, si PicOutputFlag es igual a 0, la imagen actual no se emite, sino que se almacenará en la DPB como se especifica en la subcláusula C.3.4.

- De otro modo (PicOutputFlag es igual a 1 y DpbOutputTime[n] es mayor que AuCpbRemovalTime[n]), la imagen actual se emite más tarde y se almacenará en la DPB (como se especifica en la subcláusula C.3.4) y se emite en el momento DpbOutputTime[n], a menos que se indique que no se emite mediante la descodificación o inferencia de no\_output\_of\_prior\_pics\_flag igual a 1 en un momento que precede a DpbOutputTime[n].

**[0304]** Cuando se emite, la imagen se recorta, utilizando la ventana de recorte de conformidad especificada en el SPS activo para la imagen.

**[0305]** Cuando la imagen n es una imagen que se emite y no es la última imagen del flujo de bits que se emite, el valor la variable DpbOutputInterval[n] se obtiene de la forma siguiente:

$$\text{DpbOutputInterval}[n] = \text{DpbOutputTime}[\text{nextPicInOutOrder}] - \text{DpbOutputTime}[n] \quad (\text{C-17})$$

donde nextPicInOutOrder indica la imagen que sigue a la imagen n en orden de salida y tiene PicOutputFlag igual a 1.

**[0306]** La FIG. 5 es un diagrama conceptual que ilustra imágenes CL-RAS, de acuerdo con una o más técnicas de esta divulgación. En el ejemplo de la FIG. 5, cada cuadrado representa una imagen diferente. Además, en el ejemplo de la FIG. 5, los cuadrados están dispuestos en las columnas 250A a 250D (colectivamente, "columnas 250") y las filas 252A a 252D (colectivamente, "filas 252"). Cada una de las columnas 250 corresponde a una unidad de acceso diferente (AU). Cada una de las filas 252 corresponde a una capa diferente. Las más altas de las filas 252 dependen de las más bajas de las filas 252. En el ejemplo de la FIG. 5, los cuadrados sombreados corresponden a imágenes que no son IRAP ni CL-RAS.

**[0307]** Como se muestra en el ejemplo de la FIG. 5, las imágenes CL-RAS pueden existir en unidades de acceso IRAP (es decir, unidades de acceso que incluyen imágenes IRAP en la capa base), como las unidades de acceso correspondientes a la columna 250A. Además, como se ilustra en el ejemplo de la FIG. 5, una imagen CL-RAS (p. ej., la imagen CL-RAS en la columna 250B y la fila 252B) puede existir en una unidad de acceso que sucede, en orden de descodificación, a una unidad de acceso IRAP y precede, en orden de descodificación, a una unidad de acceso IRAP de cambio de capa que incluye una imagen IRAP en la misma capa que la imagen CL-RAS. En el ejemplo de la FIG. 5, la columna 250A corresponde a una unidad de acceso IRAP y la columna 250C corresponde a una unidad de acceso IRAP de cambio de capa.

**[0308]** En diferentes casos, las representaciones codificadas de imágenes CL-RAS pueden o pueden no incluirse en un flujo de bits. Puede requerirse que un descodificador conforme pueda descodificar flujos de bits que incluyan imágenes CL-RAS y flujos de bits que no incluyan imágenes CL-RAS. Por lo tanto, de acuerdo con una o más técnicas de esta divulgación, un descodificador de vídeo u otro dispositivo puede determinar, basándose en un valor (por ejemplo, NoCIRasPicPresentFlag), si todas las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de bits de datos de vídeo.

**[0309]** La FIG. 6A es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo 20 de acuerdo con una o más técnicas de esta divulgación. La FIG. 6A se presenta como un ejemplo. Otras operaciones de ejemplo de un codificador de vídeo de acuerdo con las técnicas de esta divulgación pueden incluir más, menos o diferentes acciones.

**[0310]** En el ejemplo de la FIG. 6A, el codificador de vídeo 20 puede generar un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados, en el que el flujo de bits incluye elementos sintácticos a partir de los cuales un dispositivo que recibe el flujo de bits puede obtener un valor que especifica si las imágenes CL-RAS de una unidad de acceso IRAP están presentes en el flujo de bits (300). Además, el codificador de vídeo 20 puede emitir el flujo de bits (302).

**[0311]** La FIG. 6B 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. 6B se presenta como un ejemplo. Otras operaciones de ejemplo de un descodificador de vídeo de acuerdo con las técnicas de esta divulgación pueden incluir más, menos o diferentes acciones. Varios dispositivos pueden realizar la operación de la FIG. 6B, como el codificador de vídeo 20, un elemento de red compatible con medios (MANE) u otro tipo de dispositivo. El dispositivo puede o no ser el dispositivo que originalmente generó un flujo de bits. En el ejemplo de la FIG. 6B, el dispositivo puede generar un mensaje SEI que establece una variable que especifica si las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados (310). El dispositivo puede incluir el mensaje SEI en el flujo de bits (312).

**[0312]** La FIG. 6C es un diagrama de flujo que ilustra una operación de ejemplo del descodificador de vídeo 30, de acuerdo con una o más técnicas de esta divulgación. La FIG. 6C se presenta como un ejemplo. Otras operaciones de

ejemplo de un descodificador de vídeo de acuerdo con las técnicas de esta divulgación pueden incluir más, menos o diferentes acciones.

**[0313]** En el ejemplo de la FIG. 6C, el descodificador de vídeo 30 puede determinar, basándose en un valor, si todas las imágenes CL-RAS de una unidad de acceso IRAP están presentes en un flujo de bits de datos de vídeo (350). Además, 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 de datos de vídeo (352).

**[0314]** Los siguientes párrafos proporcionan ejemplos de esta divulgación.

Ejemplo 1. Un procedimiento de procesamiento de datos de vídeo, con el procedimiento que comprende determinar, basándose en un valor, si todas las imágenes omitidas de acceso aleatorio de capa cruzada (CL-RAS) asociadas con una unidad de acceso de punto de acceso aleatorio intra (IRAP) están presentes en un flujo de bits de datos de vídeo.

Ejemplo 2. El procedimiento del ejemplo 1, en el que el procedimiento comprende establecer el valor basándose al menos en parte en datos en un mensaje de información de mejora suplementaria (SEI) del flujo de bits de datos de vídeo.

Ejemplo 3. El procedimiento del ejemplo 1, en el que el valor se establece mediante un mecanismo fuera de un proceso de descodificación para los datos de vídeo.

Ejemplo 4. El procedimiento del ejemplo 3, en el que cuando una unidad de acceso actual es una unidad de acceso IRAP inicial, el valor se establece mediante el mecanismo fuera del proceso de descodificación para los datos de vídeo.

Ejemplo 5. El procedimiento del ejemplo 1, en el que el procedimiento comprende establecer el valor basado al menos en parte en datos en un conjunto de parámetros de vídeo (VPS).

Ejemplo 6. El procedimiento del ejemplo 1, que además comprende realizar una prueba de conformidad de flujo de bits para determinar si el flujo de bits de datos de vídeo cumple con una especificación de codificación de vídeo, en el que es un requisito de conformidad de flujo de bits que si una imagen actual es una imagen IRAP con un identificador de capa mayor que 0, entonces para cada capa de referencia directa de una capa actual, existe al menos una imagen IRAP en una secuencia de vídeo codificada (CVS) con un identificador de capa igual a un identificador de capa de la capa actual y con un orden de descodificación que precede a la imagen actual.

Ejemplo 7. El procedimiento del ejemplo 1, en el que una carga útil de secuencia de bytes sin procesar (RBSP) de un conjunto de parámetros de secuencia activada (SPS) para un valor de ID de capa particular permanece activo para una secuencia de imágenes en orden de descodificación con el valor de ID de capa particular, a partir de una imagen que tiene el valor de ID de capa particular en una unidad de acceso IRAP inicial o la primera unidad de acceso después de la unidad de acceso IRAP inicial que contiene una imagen con el valor `nuh_layer_id`, inclusive, hasta una imagen siguiente que tenga el valor de ID de capa particular en una unidad de acceso IRAP inicial, exclusivo, o el final de la CVS, lo que se produzca primero.

Ejemplo 8. El procedimiento del ejemplo 1, que además comprende cuando el valor indica que no todas las imágenes CL-RAS asociadas con una unidad de acceso IRAP están presentes en el flujo de bits de datos de vídeo y una imagen actual es una imagen CL-RAS en una unidad de acceso IRAP inicial, generando imágenes de referencia no disponibles para las imágenes CL-RAS.

Ejemplo 9. El procedimiento del ejemplo 1, que comprende además: obtener, a partir de un mensaje SEI del flujo de bits de vídeo codificado, la velocidad de transmisión de bits alternativa y los parámetros de tamaño de memoria intermedia de imágenes codificadas (CPB) para las unidades de acceso IRAP; y para cada una respectiva de las unidades de acceso IRAP, utilizar los parámetros alternativos de velocidad de transmisión de bits y tamaño de CPB para la unidad de acceso IRAP respectiva cuando la unidad de acceso IRAP respectiva es una unidad de acceso IRAP inicial con imágenes CL-RAS asociadas no presentes.

Ejemplo 10. El procedimiento del ejemplo 1, que comprende además: obtener, a partir de un mensaje SEI del flujo de bits de vídeo codificado, un elemento sintáctico que especifica si un elemento sintáctico de retardo de eliminación de CPB alternativa y un elemento sintáctico de desviación de eliminación de CPB alternativa están presentes en el mensaje SEI, en el que el elemento sintáctico de retardo de eliminación de CPB alternativa especifica un retardo de eliminación de CPB inicial alternativa para una CPB cuando se utilizan parámetros del Descodificador de referencia hipotético (HRD) de Capa de abstracción de red (NAL), en el que el elemento sintáctico de desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para la CPB cuando los parámetros NAL HRD están en uso; obtener, a partir del mensaje SEI, en el que los elementos sintácticos de retardo de eliminación de CPB alternativa especifican otro conjunto de retardo de eliminación de CPB inicial alternativa para una CPB cuando están en uso los parámetros del Descodificador de

Referencia Hipotético (HRD) de Capa de Codificación de Vídeo (VCL), en el que el elemento sintáctico de desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para la CPB cuando los parámetros VCL HRD están en uso; obtener, a partir del mensaje SEI, un elemento sintáctico de desviación de retardo de capa cruzada de CPB que especifica una desviación que se utilizará en una obtención de los tiempos de eliminación de CPB nominales de las unidades de acceso que siguen, en orden de descodificación, a una unidad de acceso IRAP asociada con el mensaje SEI cuando las imágenes de CL-RAS asociadas con la unidad de acceso IRAP no están presentes en el flujo de datos de vídeo; y obtener, a partir del mensaje SEI, un elemento sintáctico de desviación de retardo de capa cruzada de memoria intermedia de imágenes descodificadas (DPB) que especifica una desviación a utilizar en una obtención de los tiempos de salida de DPB de las unidades de acceso IRAP asociadas con el mensaje SEI cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP no están presentes en el flujo de bits de datos de vídeo.

Ejemplo 11. El procedimiento del ejemplo 1, que además comprende: obtener, a partir de un mensaje SEI del flujo de bits de vídeo codificado, un elemento sintáctico presente de parámetros de capa cruzada IRAP que especifica si un elemento sintáctico de retardo de eliminación de CPB alternativa y un elemento sintáctico de desviación de eliminación de CPB alternativa están presentes en el mensaje SEI, en el que el elemento sintáctico de retardo de eliminación de CPB alternativa especifica un retardo de eliminación de CPB inicial alternativa para una CPB cuando están en uso los parámetros del Descodificador de referencia hipotético (HRD) de Capa de abstracción de red (NAL), en el que el elemento sintáctico de desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para la CPB cuando los parámetros NAL HRD están en uso.

Ejemplo 12. El procedimiento del ejemplo 1, que comprende además: obtener, a partir de un mensaje SEI del flujo de bits de vídeo codificado, un elemento sintáctico que especifica si un elemento sintáctico de retardo de eliminación de CPB alternativa y un elemento sintáctico de desviación de eliminación de CPB alternativa están presentes en el mensaje SEI, en el que el elemento sintáctico de retardo de eliminación de CPB alternativa especifica un retardo de eliminación de CPB inicial alternativa para una CPB cuando los parámetros HRD de Capa de codificación de vídeo (VCL) están en uso, en el que el elemento sintáctico de desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para la CPB cuando los parámetros VCL HRD están en uso.

Ejemplo 13. El procedimiento del ejemplo 12, que además comprende realizar, basándose en parte en el elemento sintáctico de retardo de capa cruzada de CPB y el elemento sintáctico de desviación de eliminación de capa cruzada de CPB, una prueba de conformidad de flujo de bits para determinar si el flujo de bits de codificación de vídeo cumple con una norma de codificación de vídeo.

Ejemplo 14. El procedimiento del ejemplo 1, que además comprende: obtener, a partir de un mensaje SEI, un elemento sintáctico de retardo de eliminación de capa cruzada de CPB que especifica una desviación a utilizar en una obtención de los tiempos de eliminación de CPB nominales de las unidades de acceso que siguen, en orden de descodificación, a una unidad de acceso IRAP asociada con el mensaje SEI cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP no están presentes en el flujo de bits de datos de vídeo; y obtener, a partir del mensaje SEI, un elemento sintáctico de desviación de eliminación de capa cruzada de memoria intermedia de imágenes descodificadas (DPB) que especifica una desviación que se utilizará en una obtención de los tiempos de salida DPB de las unidades de acceso IRAP asociadas con el mensaje SEI cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP no están presentes en el flujo de bits de datos de vídeo.

Ejemplo 15. El procedimiento del ejemplo 14, que además comprende realizar, basándose en parte en el elemento sintáctico de retardo de capa cruzada de CPB y el elemento sintáctico de desviación de eliminación de capa cruzada de CPB, una prueba de conformidad de flujo de bits para determinar si el flujo de bits de codificación de vídeo cumple con una norma de codificación de vídeo.

Ejemplo 16. El procedimiento del ejemplo 14, que comprende además cuando el valor indica que no todas las imágenes CL-RAS asociadas con la unidad de acceso IRAP están presentes en el flujo de bits de datos de vídeo: establecer una desviación de retardo de CPB igual a un valor de elementos sintáctico de desviación de retardo de eliminación de capa cruzada de CPB; y establecer una desviación de retardo de DPB igual a un valor del elemento sintáctico de desviación de retardo de eliminación de capa cruzada de DPB.

Ejemplo 17. El procedimiento del ejemplo 1, que comprende además realizar, basado en parte en el valor, una operación de CPB.

Ejemplo 18. El procedimiento del ejemplo 9, que comprende además obtener, a partir de un mensaje SEI del flujo de bits de codificación de vídeo, un elemento sintáctico que indica una unidad de acceso hasta la cual son aplicables un valor de velocidad de transmisión de bits de capa cruzada y un valor de tamaño de CPB de capa cruzada.

Ejemplo 19. El procedimiento del ejemplo 1, que comprende además obtener, a partir de un mensaje SEI del flujo de bits de codificación de vídeo, un elemento sintáctico que especifica una velocidad de transmisión de bits de entrada máxima de una CPB cuando la CPB funciona a nivel de unidad de acceso.

5 Ejemplo 20. El procedimiento del ejemplo 1, que comprende además: obtener, a partir de un mensaje SEI del flujo de bits de codificación de vídeo, un elemento sintáctico; y determinar, basándose al menos en parte en el elemento sintáctico, un tamaño de CPB cuando la CPB funciona a nivel de unidad de acceso.

10 Ejemplo 21. El procedimiento del ejemplo 1, que además comprende realizar, basándose al menos en parte en el valor, un proceso de salida de imagen.

Ejemplo 22. Un procedimiento que comprende cualquiera de los procedimientos de los ejemplos 1-21.

15 Ejemplo 23. Un dispositivo que comprende uno o más procesadores se configura para realizar los procedimientos de cualquiera de los ejemplos 1-21.

Ejemplo 24. Un dispositivo que comprende medios para realizar los procedimientos de cualquiera de los ejemplos 1-21.

20 Ejemplo 25. Un medio de almacenamiento de datos legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan, configuran un dispositivo para llevar a cabo los procedimientos de cualquiera de los ejemplos 1-21.

25 Ejemplo 26. Un procedimiento para codificar datos de vídeo, con el procedimiento que comprende: generar un mensaje de Información de mejora suplementaria (SEI) que incluye un elemento sintáctico que especifica si un elemento sintáctico de retardo de eliminación de memoria intermedia de imágenes codificadas (CPB) alternativa y un elemento sintáctico de desviación de eliminación de CPB alternativa están presentes en el mensaje SEI, en el que el elemento sintáctico de retardo de eliminación de CPB alternativa especifica un retardo de eliminación de CPB inicial alternativa para una CPB cuando están en uso los parámetros del Descodificador de Referencia Hipotético (HRD) de Capa de abstracción de red (NAL), en el que el elemento sintáctico de desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para la CPB cuando los parámetros NAL HRD (y en algunos ejemplos los parámetros VCL HRD) están en uso; en el que el mensaje SEI incluye un elemento sintáctico de desviación de retardo de capa cruzada de CPB que especifica una desviación que se utilizará en una obtención de los tiempos de eliminación de CPB nominales de las unidades de acceso que siguen, en orden de descodificación, a una unidad de acceso IRAP asociada con el mensaje SEI cuando las imágenes omitidas de acceso aleatorio de capa cruzada (CL-RAS) asociadas con la unidad de acceso IRAP no están presentes en el flujo de bits de datos de vídeo; y en el que el mensaje SEI incluye un elemento sintáctico de desviación de retardo de capa cruzada de memoria intermedia de imágenes descodificadas (DPB) que especifica una desviación que se utilizará en una obtención de los tiempos de salida DPB de las unidades de acceso IRAP asociadas con el mensaje SEI cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP no están presentes en el flujo de bits de datos de vídeo.

45 Ejemplo 27. El procedimiento del ejemplo 26, en el que el mensaje SEI incluye un elemento sintáctico presente de parámetros de capa cruzada IRAP que especifica si un elemento sintáctico de retardo de eliminación de CPB alternativa y un elemento sintáctico de desviación de eliminación de CPB alternativa están presentes en el mensaje SEI, en el que el elemento sintáctico de retardo de eliminación de CPB alternativa especifica un retardo de eliminación de CPB inicial alternativa para una CPB cuando los parámetros del Descodificador de referencia hipotético (HRD) de la Capa de abstracción de red (NAL) están en uso, en el que el elemento sintáctico de desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para la CPB cuando los parámetros NAL HRD están en uso.

55 Ejemplo 28. El procedimiento del ejemplo 26, en el que el mensaje SEI incluye un elemento sintáctico que especifica si un elemento sintáctico de retardo de eliminación de CPB alternativa y un elemento sintáctico de desviación de eliminación de CPB alternativa están presentes en el mensaje SEI, en el que el elemento sintáctico de retardo de eliminación de CPB alternativa especifica un retardo de eliminación de CPB inicial alternativa para una CPB cuando los parámetros HRD de capa de codificación de vídeo (VCL) están en uso, en el que el elemento sintáctico de desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para la CPB cuando los parámetros VCL HRD están en uso.

60 Ejemplo 29. El procedimiento del ejemplo 26, en el que el mensaje SEI incluye: un elemento sintáctico de retardo de eliminación de capa cruzada de CPB que especifica una desviación a usar en una obtención de los tiempos de eliminación de CPB nominales de las unidades de acceso que siguen, en orden de descodificación, a una unidad de acceso IRAP asociada con el mensaje SEI cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP no están presentes en el flujo de bits de datos de vídeo; y un elemento sintáctico de desviación de eliminación de capa cruzada de memoria intermedia de imágenes descodificadas (DPB) que especifica una desviación que se utilizará en una obtención de los tiempos de salida DPB de las unidades de acceso IRAP asociadas con el mensaje

SEI cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP no están presentes en el flujo de datos de vídeo.

5 Ejemplo 30. Un procedimiento para codificar datos de vídeo, con el procedimiento que comprende generar un mensaje de Información de Mejora Suplementaria (SEI) que incluye un elemento sintáctico que indica una unidad de acceso hasta la cual un valor de velocidad de transmisión de bits de capa cruzada y un valor de tamaño de CPB de memoria intermedia de imágenes codificadas de capa cruzada son aplicables.

10 Ejemplo 31. Un procedimiento para codificar datos de vídeo, con el procedimiento que comprende generar un mensaje de Información de Mejora Suplementaria (SEI) que incluye un elemento sintáctico que especifica una velocidad de transmisión de bits de entrada máxima de una CPB cuando la CPB funciona a nivel de unidad de acceso.

15 Ejemplo 32. Un dispositivo que comprende uno o más procesadores se configura para realizar los procedimientos de cualquiera de los ejemplos 26-31.

Ejemplo 33. Un dispositivo que comprende medios para realizar los procedimientos de cualquiera de los ejemplos 26-31.

20 Ejemplo 34. Un medio de almacenamiento de datos legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan, configuran un dispositivo para llevar a cabo los procedimientos de cualquiera de los ejemplos 26-31.

25 **[0315]** 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 transmitir 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 correspondan a un medio tangible tal como medios de almacenamiento de datos o medios de comunicación que incluyan cualquier medio que facilite 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 sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los 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.

40 **[0316]** 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 un 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 debidamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una página web, un servidor u otra fuente remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, donde algunos discos normalmente emiten datos magnéticamente, mientras que otros discos emiten datos ópticamente con láseres. Las combinaciones de lo anterior también deberían incluirse dentro del alcance de los medios legibles por ordenador.

55 **[0317]** Las instrucciones se pueden ejecutar mediante 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 de puertas programables in situ (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o software dedicados, configurados para la codificación y la descodificación, o incorporados en un códec combinado. Además, las técnicas se podrían implementar totalmente en uno o más circuitos o elementos lógicos.

65 **[0318]** Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips).

5 Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec, o ser proporcionadas por un grupo de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, conjuntamente con software y/o firmware adecuados.

10 **[0319]** Se han descrito diversos ejemplos. Estos y otros ejemplos, junto con varias combinaciones de estos ejemplos, están dentro del alcance de las siguientes reivindicaciones.

**REIVINDICACIONES**

1. Un procedimiento de decodificación de datos de vídeo, comprendiendo el procedimiento:

5 generar imágenes de referencia no disponibles para el primero, segundo, tercero, cuarto y quinto subconjuntos de imagen de referencia para una imagen omitida de acceso aleatorio de capa cruzada (CL-RAS) en una unidad de acceso de punto de acceso aleatorio intra (IRAP) inicial presente en un flujo de bits de datos de vídeo,

10 la unidad de acceso IRAP inicial es una unidad de acceso que contiene una o más imágenes IRAP en las que una imagen codificada con ID de capa nuh igual a 0 tiene NoRaslOutputFlag igual a 1, donde una imagen en una capa base de la unidad de acceso IRAP inicial es una imagen IRAP y no todas las imágenes de la unidad de acceso IRAP inicial son imágenes IRAP,

15 la imagen CL-RAS es una imagen que no es una imagen IRAP pero está en la unidad de acceso IRAP inicial, en el que la imagen CL-RAS contiene referencias a imágenes que no están presentes en el flujo de bits de datos de vídeo y un identificador de capa de la imagen CL-RAS es mayor que 0,

20 en el que cada imagen de referencia en el primer subconjunto de imágenes de referencia es una imagen de referencia a corto plazo que aparece antes de la imagen CL-RAS en orden de salida, cada imagen de referencia en el segundo subconjunto de imágenes de referencia es una imagen de referencia a corto plazo que aparece después de la imagen CL-RAS en orden de salida, cada imagen de referencia en el tercer subconjunto de imágenes de referencia es una imagen de referencia a corto plazo que no se utiliza para la predicción inter de la imagen CL-RAS, cada imagen de referencia en el cuarto subconjunto de imágenes de referencia es una imagen de referencia a largo plazo que se puede utilizar para la predicción inter de la imagen CL-RAS, y cada imagen de referencia en el quinto subconjunto de imágenes de referencia es una imagen de referencia a largo plazo que no se utiliza para la predicción inter de la imagen CL-RAS; y

25 decodificar la imagen CL-RAS basándose al menos en parte en una o más de las imágenes de referencia no disponibles generadas; y

30 en el que generar las imágenes de referencia no disponibles comprende:

35 para cada posición respectiva en una matriz para el primer subconjunto de imágenes de referencia, si una entrada en la posición respectiva es igual a "sin imagen de referencia", generar una imagen de referencia para la que se aplica lo siguiente:

40 un valor de recuento de orden de imágenes (POC) para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un primer conjunto de valores POC,

un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0,

la imagen de referencia generada está marcada como utilizada para referencia a corto plazo,

45 la posición respectiva en la matriz para el primer subconjunto de imágenes de referencia se establece en la imagen de referencia generada, y

un identificador de capa para la imagen de referencia generada se establece en un identificador de capa de la imagen CL-RAS;

50 para cada posición respectiva en una matriz para el segundo subconjunto de imágenes de referencia, si una entrada en la posición respectiva es igual a "ninguna imagen de referencia", generar una imagen de referencia para la cual se aplica lo siguiente:

55 un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un segundo conjunto de valores POC,

un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0,

60 la imagen de referencia generada está marcada como utilizada para referencia a corto plazo,

la posición respectiva en la matriz para el segundo subconjunto de imágenes de referencia se establece en la imagen de referencia generada, y

65 un identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen CL-RAS;

para cada posición respectiva en una matriz para el tercer subconjunto de imágenes de referencia, si una entrada en la posición respectiva es igual a "sin imagen de referencia", generar una imagen de referencia para la que se aplica lo siguiente:

- 5 un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un tercer conjunto de valores POC,
- 10 un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0,
- la imagen de referencia generada está marcada como utilizada para referencia a corto plazo,
- la posición respectiva en la matriz para el tercer subconjunto de imágenes de referencia se establece en la imagen de referencia generada, y
- 15 un identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen CL-RAS;

para cada posición respectiva en una matriz para el cuarto subconjunto de imágenes de referencia, si una entrada en la posición respectiva es igual a "sin imagen de referencia", generar una imagen de referencia para la cual se aplica lo siguiente:

- 20 un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un cuarto conjunto de valores POC,
- 25 se infiere que un valor de un elemento sintáctico que especifica los bits menos significativos de un valor POC para la imagen generada es igual al resultado de una operación "y" a nivel de bits del valor POC en la posición respectiva en el cuarto conjunto de valores POC y a valor resultante de una variable que indica el valor máximo que los bits menos significativos del valor POC pueden tener menos 1,
- 30 un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0,
- la imagen de referencia generada se marca como utilizada para referencia a largo plazo,
- 35 la posición respectiva en la matriz para el cuarto subconjunto de imágenes de referencia se establece en la imagen de referencia generada, y
- 40 un identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen CL-RAS;

para cada posición respectiva en una quinta matriz de subconjuntos de imágenes de referencia, si una entrada en la posición respectiva es igual a "sin imagen de referencia", generar una imagen de referencia para la que se aplica lo siguiente:

- 45 un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un quinto conjunto de valores POC,
- 50 se infiere que un valor de un elemento sintáctico que especifica los bits menos significativos de un valor POC para la imagen generada sea igual al resultado de una operación a nivel de bits "y" del valor POC en la posición respectiva en el quinto conjunto de valores POC y un valor resultante de una variable que indica el valor máximo que los bits menos significativos del valor POC pueden tener menos 1,
- 55 un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0,
- la imagen de referencia generada se marca como utilizada para referencia a largo plazo,
- la posición respectiva en la quinta matriz de subconjuntos de imágenes de referencia se establece en la imagen de referencia generada, y
- 60 un identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen CL-RAS.

**2.** El procedimiento según la reivindicación 1, en el que el procedimiento comprende:

- 65 establecer un valor basado, al menos en parte, en datos en un mensaje de información de mejora suplementaria (SEI) del flujo de bits de datos de vídeo; y

determinar, basándose en el valor, si todas las imágenes CL-RAS asociadas con la unidad de acceso IRAP inicial están presentes en el flujo de bits de datos de vídeo.

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

determinar, basándose en un valor, si todas las imágenes CL-RAS asociadas con la unidad de acceso IRAP inicial están presentes en el flujo de bits de datos de vídeo, en el que el valor se establece mediante un mecanismo fuera de un proceso de descodificación para los datos de vídeo.

10

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

determinar, basándose en un valor, si todas las imágenes CL-RAS asociadas con la unidad de acceso IRAP inicial están presentes en el flujo de bits de datos de vídeo; y

15

realizar, basándose en parte en el valor, una operación de memoria intermedia de imágenes codificadas (CPB).

**5.** El procedimiento según la reivindicación 4, en el que la realización de la operación de CPB comprende:

20

cuando el valor indica que no todas las imágenes CL-RAS asociadas con la unidad de acceso IRAP inicial están presentes en el flujo de bits de datos de vídeo y un mensaje de información de mejora suplementaria (SEI) del período de almacenamiento en memoria intermedia incluye un primer elemento sintáctico que especifica la presencia en el mensaje SEI de período de almacenamiento en memoria intermedia de un segundo elemento sintáctico y un tercer elemento sintáctico:

25

establecer un retardo de eliminación de CPB en un retardo de eliminación alternativo especificado por el segundo elemento sintáctico; y

30

establecer una desviación de eliminación de CPB en una desviación de eliminación alternativa especificada por el tercer elemento sintáctico.

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

35

obtener, a partir de un mensaje SEI del flujo de bits de vídeo codificado, un elemento sintáctico que especifica si un elemento sintáctico de retardo de eliminación de CPB alternativa y un elemento sintáctico de desviación de eliminación de CPB alternativa están presentes en el mensaje SEI, en el que el elemento sintáctico de retardo de eliminación de CPB alternativa especifica un retardo de eliminación de CPB inicial alternativa para una CPB cuando los parámetros del Descodificador de referencia hipotético (HRD) de la Capa de abstracción de red (NAL) están en uso, y en el que el elemento sintáctico de desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para la CPB cuando los parámetros de NAL HRD están en uso;

40

obtener, a partir del mensaje SEI, un elemento sintáctico de desviación de retardo de capa cruzada de CPB que especifica una desviación que se utilizará en una obtención de los tiempos de eliminación de CPB nominales de las unidades de acceso que siguen, en orden de descodificación, a una unidad de acceso IRAP asociada con el mensaje SEI cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP asociada con el mensaje SEI no están presentes en el flujo de bits de datos de vídeo; y

45

obtener, a partir del mensaje SEI, un elemento sintáctico de desviación de retardo de capa cruzada de memoria intermedia de imágenes descodificadas (DPB) que especifica una desviación que se utilizará en una obtención de los tiempos de salida DPB de las unidades de acceso IRAP asociadas con el mensaje SEI cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP inicial no están presentes en el flujo de bits de datos de vídeo.

50

55

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

obtener, a partir de un mensaje SEI del flujo de bits de vídeo codificado, un elemento sintáctico que especifica si un elemento sintáctico de retardo de eliminación de CPB alternativa y un elemento sintáctico de desviación de eliminación de CPB alternativa están presentes en el mensaje SEI, en el que el elemento sintáctico de retardo de eliminación de CPB alternativa especifica un retardo de eliminación de CPB inicial alternativa para una CPB cuando los parámetros del Descodificador de referencia hipotético (HRD) de la capa de codificación de vídeo (VCL) están en uso, y en el que el elemento sintáctico de desviación de eliminación de CPB alternativa especifica una desviación de eliminación de CPB inicial alternativa para la CPB cuando los parámetros de VCL HRD están en uso;

60

65

5 obtener, a partir del mensaje SEI, un elemento sintáctico de desviación de retardo de capa cruzada de CPB que especifica una desviación que se utilizará en una obtención de los tiempos de eliminación de CPB nominales de las unidades de acceso que siguen, en orden de descodificación, a una unidad de acceso IRAP asociada con el mensaje SEI cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP asociada con el mensaje SEI no están presentes en el flujo de bits de datos de vídeo; y

10 obtener, a partir del mensaje SEI, un elemento sintáctico de desviación de retardo de capa cruzada de memoria intermedia de imágenes descodificadas (DPB) que especifica una desviación que se utilizará en una obtención de los tiempos de salida DPB de las unidades de acceso IRAP asociadas con el mensaje SEI cuando las imágenes CL-RAS asociadas con la unidad de acceso IRAP inicial no están presentes en el flujo de bits de datos de vídeo.

8. Un dispositivo para descodificar datos de vídeo, comprendiendo el dispositivo:

15 un medio de almacenamiento de datos configurado para almacenar datos de vídeo, y

uno o más procesadores configurados para:

20 generar imágenes de referencia no disponibles para el primer, segundo, tercero, cuarto y quinto subconjuntos de imagen de referencia para una imagen omitida de acceso aleatorio de capa cruzada (CL-RAS) en una unidad de acceso de punto de acceso aleatorio intra (IRAP) inicial presente en un flujo de bits de datos de vídeo,

25 la unidad de acceso IRAP inicial es una unidad de acceso que contiene una o más imágenes IRAP en las que una imagen codificada con ID de capa nuh igual a 0 tiene NoRaslOutputFlag igual a 1, donde una imagen en una capa base de la unidad de acceso IRAP inicial es una imagen IRAP y

30 La imagen CL-RAS es una imagen que no es una imagen IRAP pero está en la unidad de acceso IRAP inicial, en el que la imagen CL-RAS contiene referencias a imágenes que no están presentes en el flujo de bits de datos de vídeo y un identificador de capa de la imagen CL-RAS es mayor que 0,

35 en el que cada imagen de referencia en el primer subconjunto de imágenes de referencia es una imagen de referencia a corto plazo que aparece antes de la imagen CL-RAS en orden de salida, cada imagen de referencia en el segundo subconjunto de imágenes de referencia es una imagen de referencia a corto plazo que aparece después de la imagen CL-RAS en orden de salida, cada imagen de referencia en el tercer subconjunto de imágenes de referencia es una imagen de referencia a corto plazo que no se utiliza para la predicción inter de la imagen CL-RAS, cada imagen de referencia en el cuarto subconjunto de imágenes de referencia es una imagen de referencia a largo plazo que se puede utilizar para la predicción inter de la imagen CL-RAS, y cada imagen de referencia en el quinto subconjunto de imágenes de referencia es una imagen de referencia a largo plazo que no se utiliza para la predicción inter de la imagen CL-RAS; y

45 descodificar la imagen CL-RAS basándose al menos en parte en una o más de las imágenes de referencia no disponibles generadas; y

en el que uno o más procesadores están configurados de manera que:

50 para cada posición respectiva en una matriz para el primer subconjunto de imágenes de referencia, si una entrada en la posición respectiva es igual a "ninguna imagen de referencia", el uno o más procesadores generan una imagen de referencia para la cual se aplica lo siguiente:

55 un valor de recuento de orden de imágenes (POC) para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un primer conjunto de valores POC,

un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0,

la imagen de referencia generada está marcada como utilizada para referencia a corto plazo,

60 la posición respectiva en la matriz para el primer subconjunto de imágenes de referencia se establece en la imagen de referencia generada, y

un identificador de capa para la imagen de referencia generada se establece en un identificador de capa de la imagen CL-RAS;

65

para cada posición respectiva en una matriz para el segundo subconjunto de imágenes de referencia, si una entrada en la posición respectiva es igual a "ninguna imagen de referencia", el uno o más procesadores generan una imagen de referencia para la cual se aplica lo siguiente:

- 5 un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un segundo conjunto de valores POC,
- un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0,
- 10 la imagen de referencia generada está marcada como utilizada para referencia a corto plazo,
- la posición respectiva en la matriz para el segundo subconjunto de imágenes de referencia se establece en la imagen de referencia generada, y
- 15 un identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen CL-RAS;

para cada posición respectiva en una matriz para el tercer subconjunto de imágenes de referencia, si una entrada en la posición respectiva es igual a "ninguna imagen de referencia", el uno o más procesadores generan una imagen de referencia para la cual se aplica lo siguiente:

- 20 un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un tercer conjunto de valores POC,
- 25 un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0,
- la imagen de referencia generada está marcada como utilizada para referencia a corto plazo,
- 30 la posición respectiva en la matriz para el tercer subconjunto de imágenes de referencia se establece en la imagen de referencia generada, y
- un identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen CL-RAS;

para cada posición respectiva en una matriz para el cuarto subconjunto de imágenes de referencia, si una entrada en la posición respectiva es igual a "ninguna imagen de referencia", el uno o más procesadores generan una imagen de referencia para la cual se aplica lo siguiente:

- 35 un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un cuarto conjunto de valores POC,
- 40 se infiere que un valor de un elemento sintáctico que especifica un bit menos significativo de un valor POC para la imagen generada es igual al resultado de una operación "y" a nivel de bits del valor POC en la posición respectiva en el cuarto conjunto de valores POC y un valor resultante de una variable que indica el valor máximo que los bits menos significativos del valor POC puede tener menos 1,
- 45 un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0,
- 50 la imagen de referencia generada se marca como utilizada para referencia a largo plazo,
- la posición respectiva en la matriz para el cuarto subconjunto de imágenes de referencia se establece en la imagen de referencia generada, y
- 55 un identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen CL-RAS;

para cada posición respectiva en una matriz para el quinto subconjunto de imágenes de referencia, si una entrada en la posición respectiva es igual a "ninguna imagen de referencia", el uno o más procesadores generan una imagen de referencia para la cual se aplica lo siguiente:

- 60 un valor POC para la imagen de referencia generada es igual a un valor POC en la posición respectiva en un quinto conjunto de valores POC,
- 65 se infiere que un valor de un elemento sintáctico que especifica un bit menos significativo de un valor POC para la imagen generada es igual al resultado de una operación a nivel de bits "y" del valor

POC en la posición respectiva en el quinto conjunto de valores POC y un valor resultante de una variable que indica el valor máximo que los bits menos significativos del valor POC puede tener menos 1,

5 un indicador de salida de imagen para la imagen de referencia generada se establece igual a 0,

la imagen de referencia generada se marca como utilizada para referencia a largo plazo,

10 la posición respectiva en la matriz para el quinto subconjunto de imágenes de referencia se establece en la imagen de referencia generada, y

un identificador de capa para la imagen de referencia generada se establece en el identificador de capa de la imagen CL-RAS.

15 **9.** El dispositivo según la reivindicación 8, en el que el uno o más procesadores están configurados para:

establecer un valor basado, al menos en parte, en datos en un mensaje de información de mejora suplementaria (SEI) del flujo de bits de datos de vídeo; y

20 determinar, basándose en el valor, si todas las imágenes CL-RAS asociadas con la unidad de acceso IRAP inicial están presentes en el flujo de bits de datos de vídeo.

25 **10.** El dispositivo de la reivindicación 8, en el que el uno o más procesadores están configurados para determinar, basándose en un valor, si todas las imágenes CL-RAS asociadas con la unidad de acceso IRAP inicial están presentes en el flujo de bits de datos de vídeo, en el que el valor se establece mediante un mecanismo fuera de un proceso de descodificación para los datos de vídeo.

**11.** El dispositivo según la reivindicación 8, en el que el uno o más procesadores están configurados para:

30 determinar, basándose en un valor, si todas las imágenes CL-RAS asociadas con la unidad de acceso IRAP inicial están presentes en el flujo de bits de datos de vídeo; y

35 realizar, basándose en parte en el valor, una operación de memoria intermedia de imágenes codificadas (CPB).

**12.** El dispositivo de la reivindicación 11, en el que los uno o más procesadores están configurados de tal manera que, como parte de la realización de la operación de CPB, los uno o más procesadores:

40 cuando el valor indica que no todas las imágenes CL-RAS asociadas con la unidad de acceso IRAP inicial están presentes en el flujo de bits de datos de vídeo y un mensaje de información de mejora suplementaria (SEI) del período de almacenamiento en memoria intermedia incluye un primer elemento sintáctico que especifica la presencia en el mensaje SEI de período de almacenamiento en memoria intermedia de un segundo elemento sintáctico y un tercer elemento sintáctico:

45 establecen un retardo de eliminación de CPB en un retardo de eliminación alternativo especificado por el segundo elemento sintáctico; y

50 establecen una desviación de eliminación de CPB en una desviación de eliminación alternativa especificada por el tercer elemento sintáctico.

**13.** Un medio de almacenamiento de datos legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que un dispositivo lleve a cabo el procedimiento de cualquiera de las reivindicaciones 1 a 7.

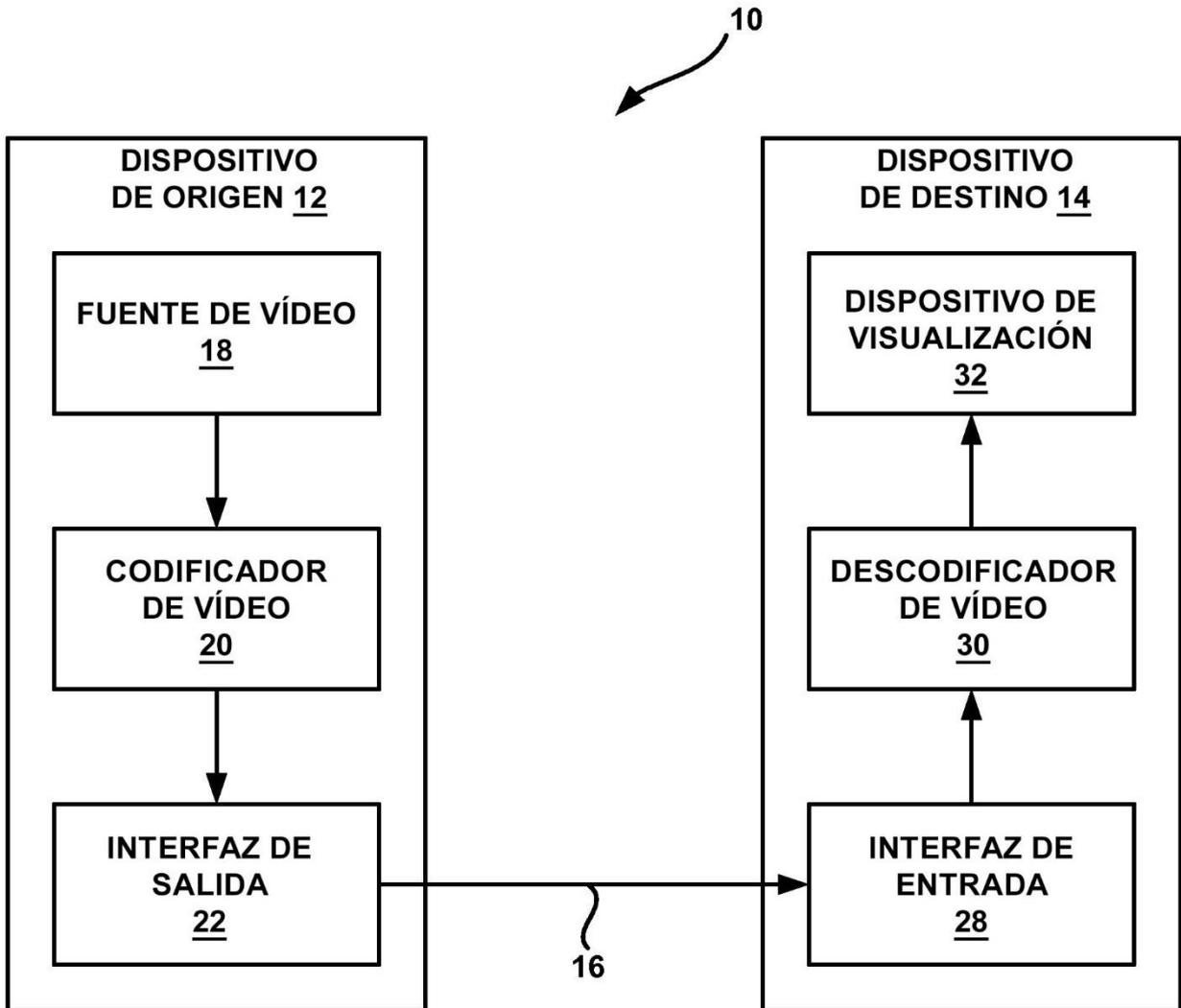
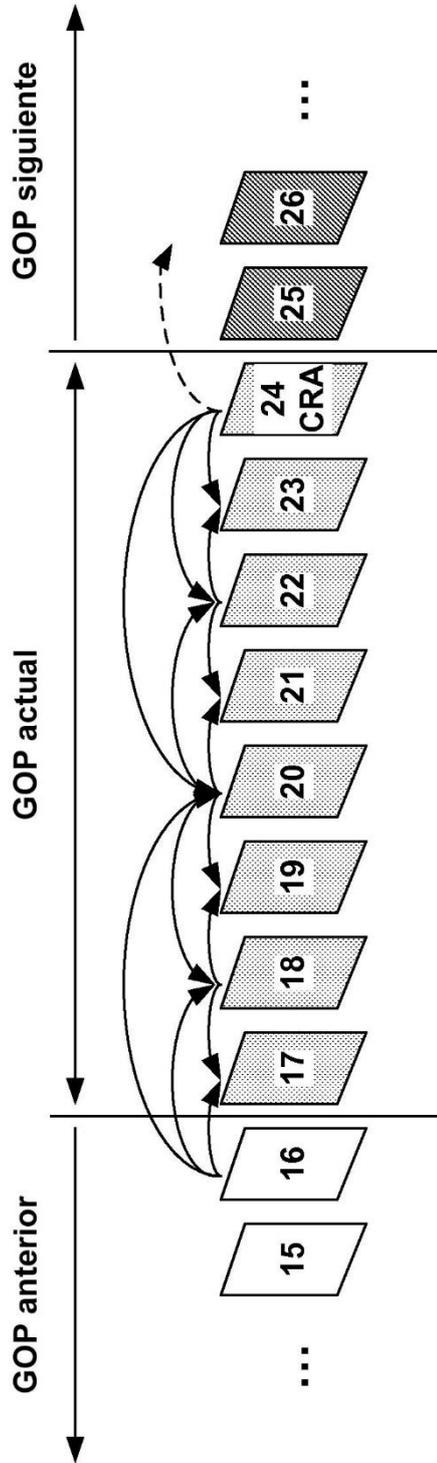


FIG. 1



**FIG. 2**

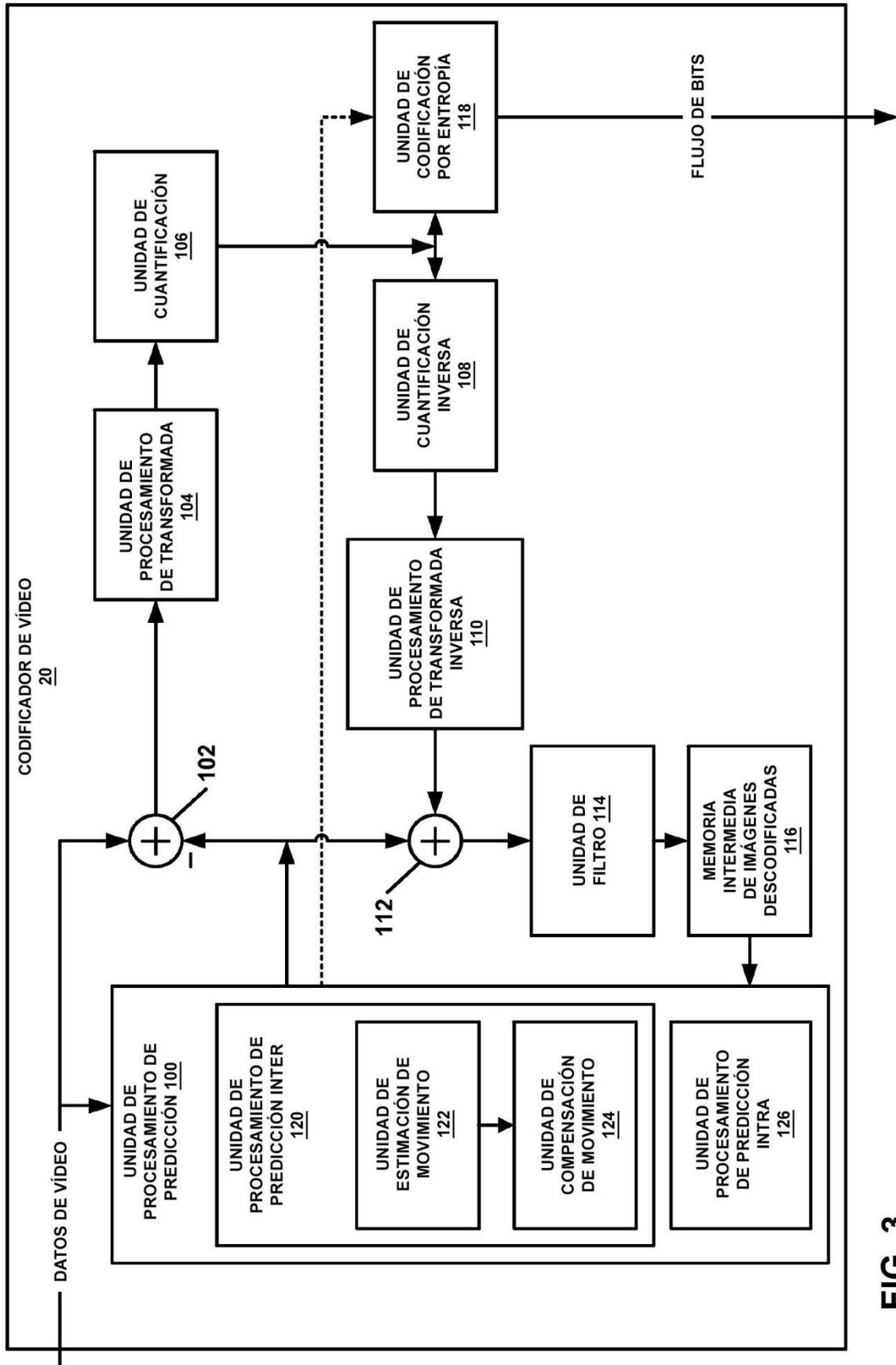


FIG. 3

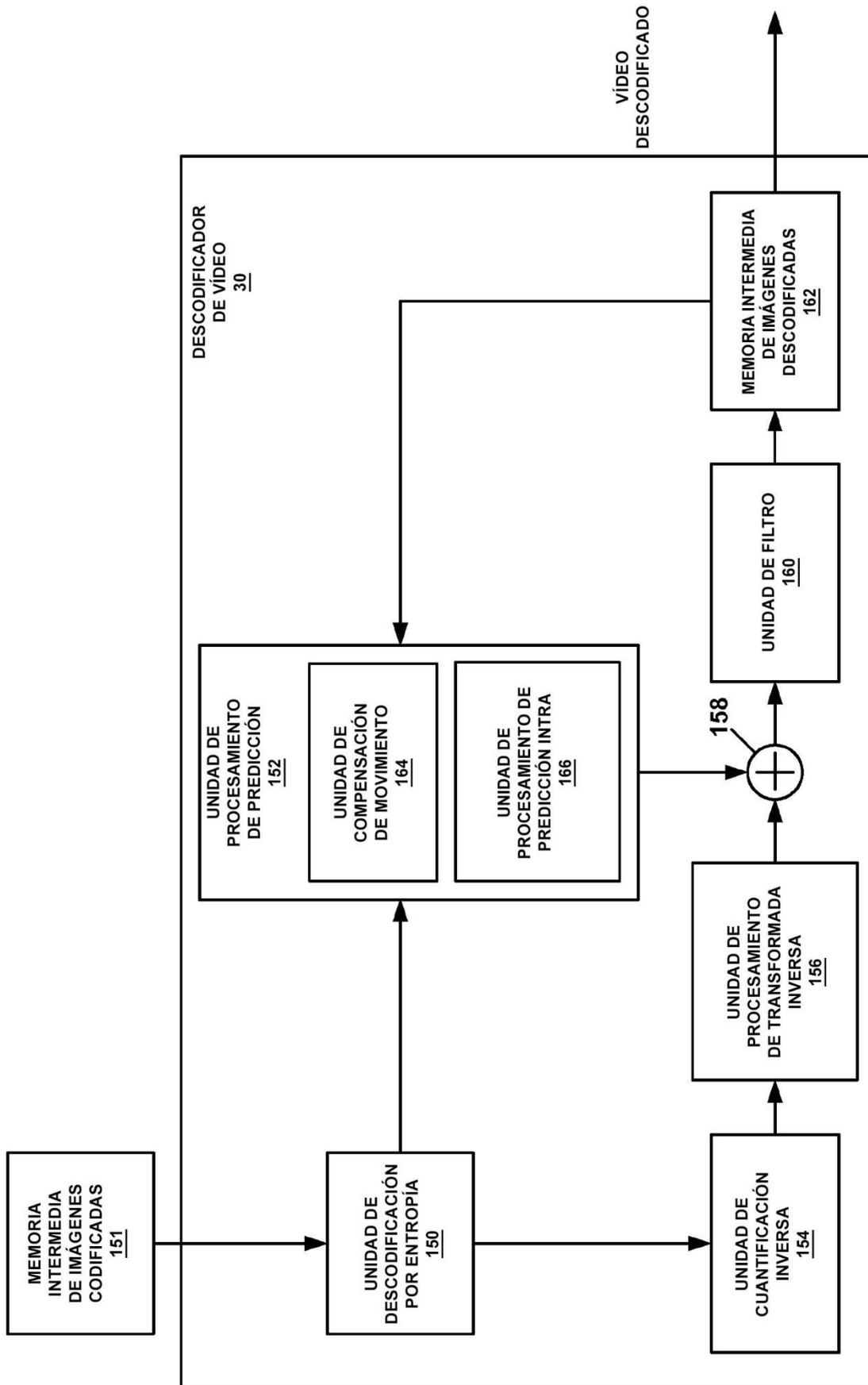


FIG. 4

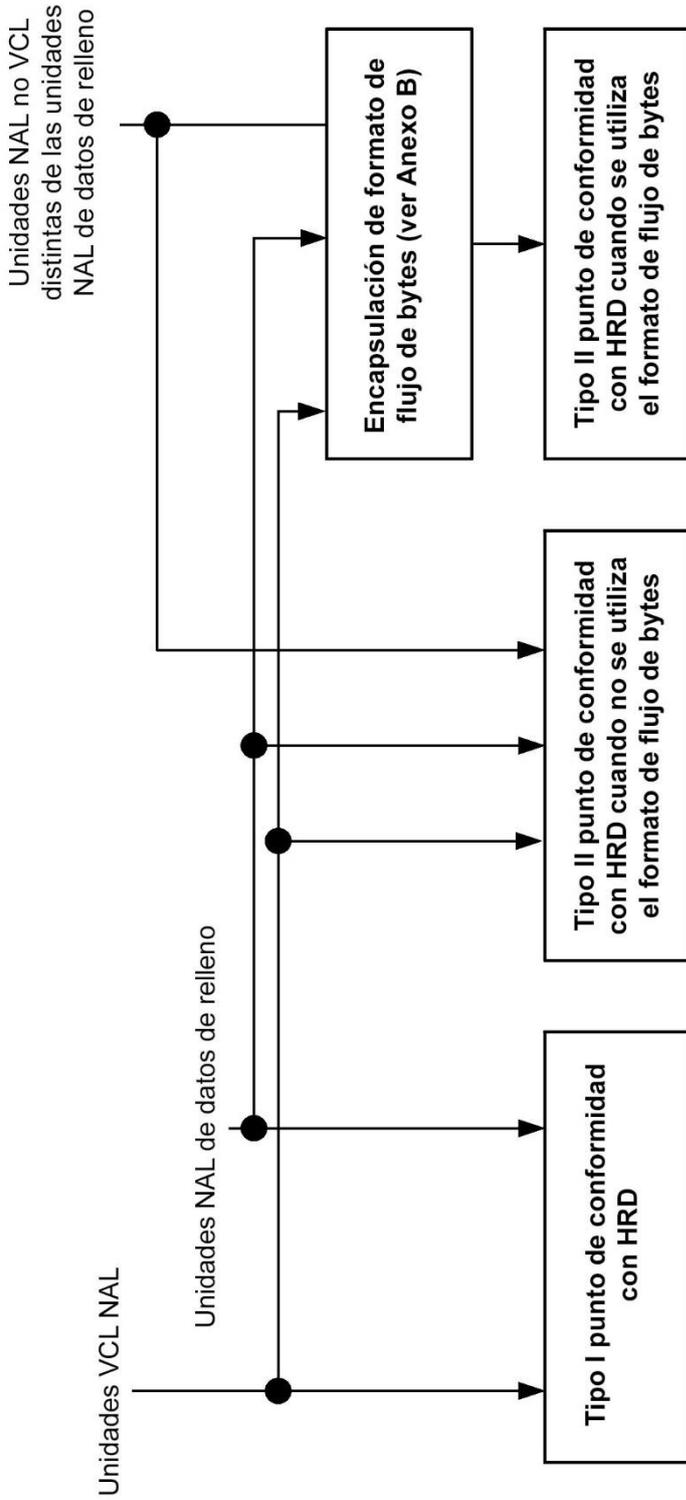
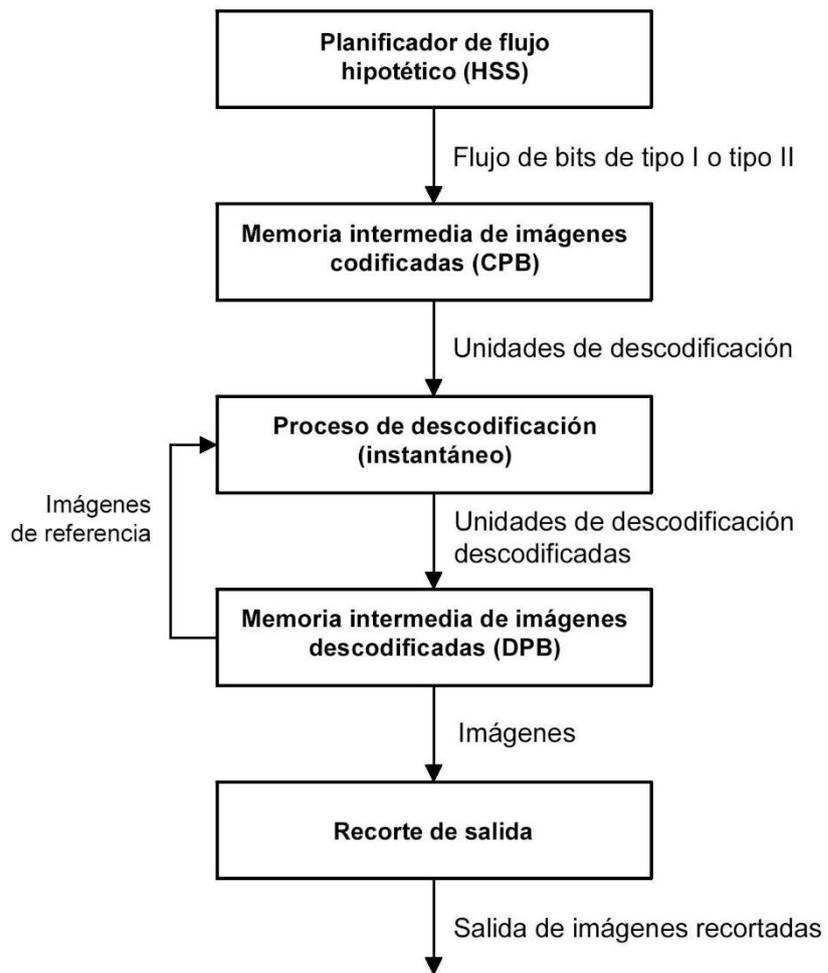


FIG. C-1



**FIG. C-2**

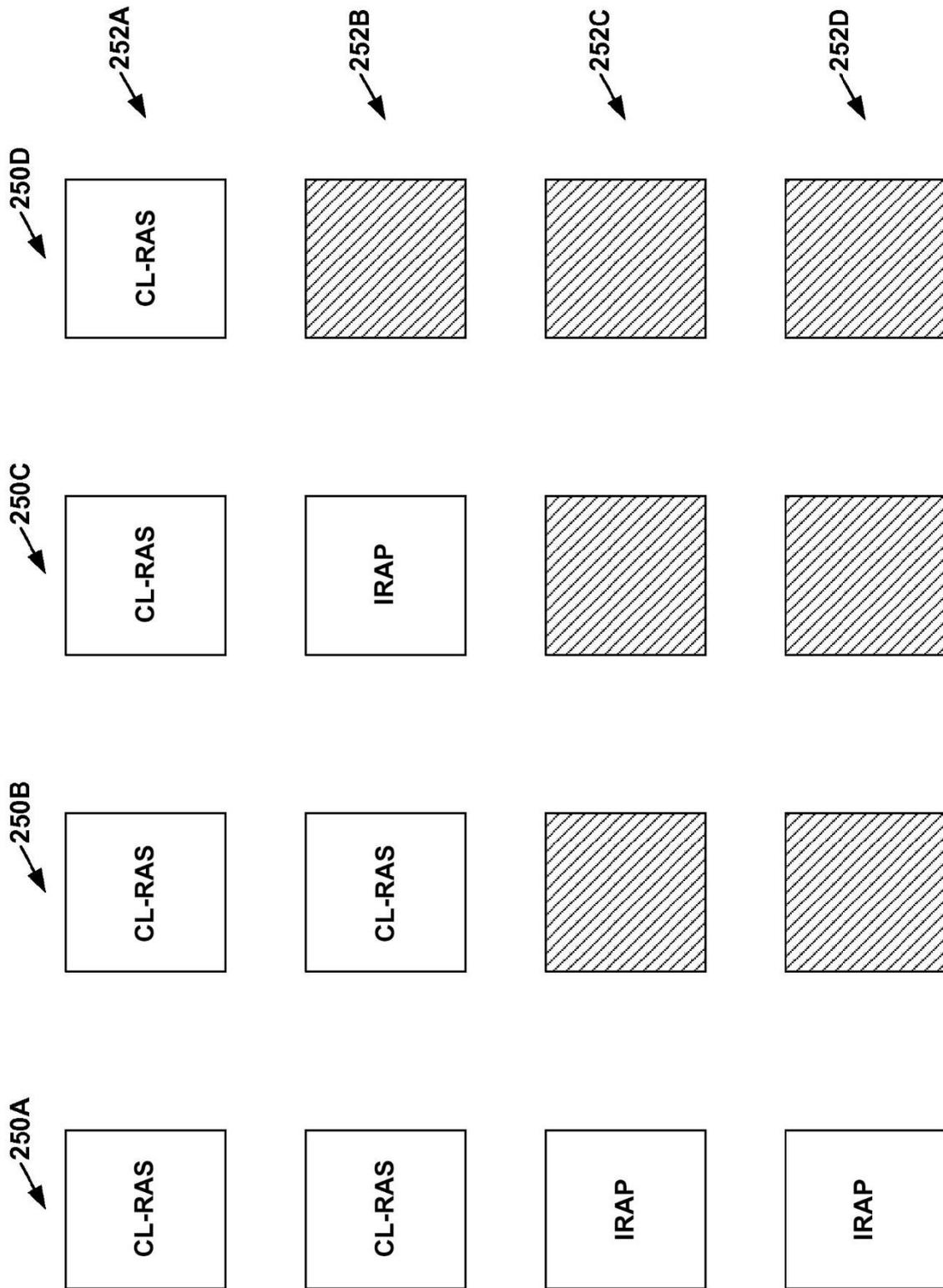


FIG. 5

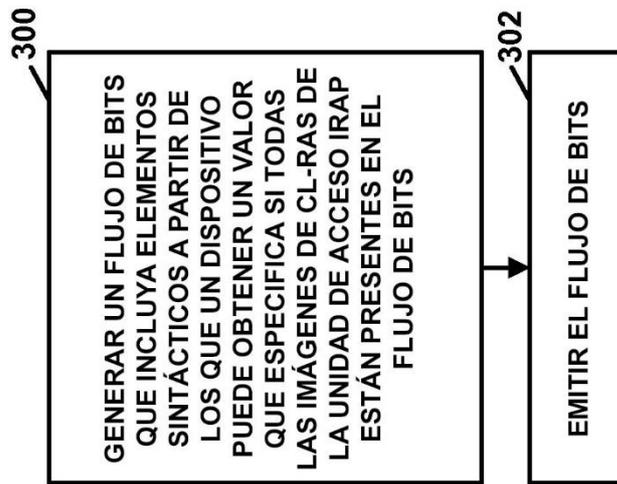


FIG. 6A

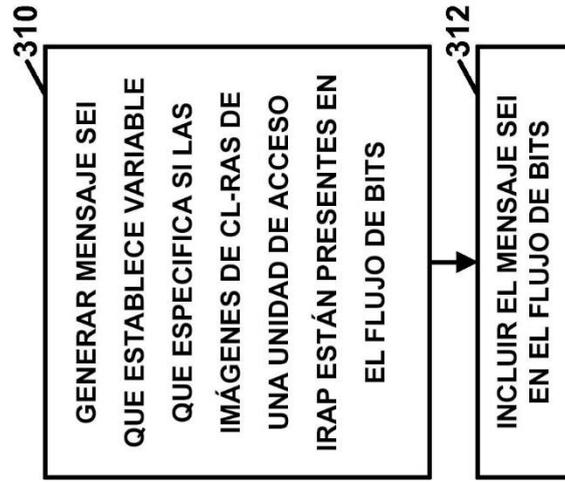


FIG. 6B

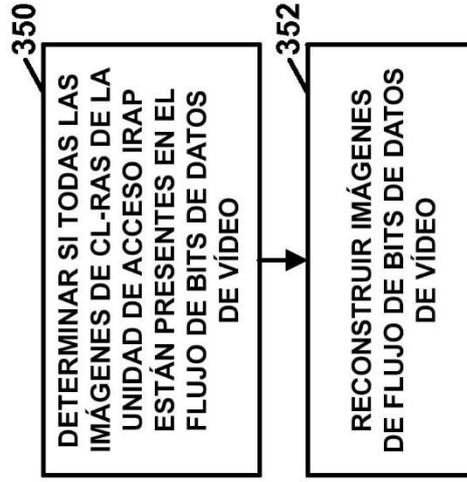


FIG. 6C