

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 854 274**

51 Int. Cl.:

**H04N 19/70** (2014.01)

**H04N 19/463** (2014.01)

12

## TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **19.12.2014** **PCT/US2014/071505**

87 Fecha y número de publicación internacional: **25.06.2015** **WO15095706**

96 Fecha de presentación y número de la solicitud europea: **19.12.2014** **E 14825063 (2)**

97 Fecha y número de publicación de la concesión europea: **18.11.2020** **EP 3085097**

54 Título: **Dispositivo y procedimiento para codificación escalable de información de vídeo**

30 Prioridad:

**19.12.2013 US 201361918630 P**

**18.12.2014 US 201414575788**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**21.09.2021**

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)**

**5775 Morehouse Drive**

**San Diego, CA 92121-1714, US**

72 Inventor/es:

**RAMASUBRAMONIAN, ADARSH KRISHNAN;**

**HENDRY, FNU y**

**WANG, YE-KUI**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

ES 2 854 274 T3

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

## DESCRIPCIÓN

Dispositivo y procedimiento para codificación escalable de información de vídeo

## 5 CAMPO TÉCNICO

[0001] La presente divulgación se refiere al campo de la codificación y compresión de vídeo, en particular a la codificación de vídeo escalable (SVC), la codificación de vídeo multivista (MVC) o la codificación de vídeo tridimensional (3D) (3DV).

## 10 ANTECEDENTES

[0002] Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, incluidos televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes personales digitales (PDA), ordenadores portátiles o de escritorio, cámaras digitales, dispositivos de grabación digitales, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos de radio celulares o por satélite, dispositivos de videoconferencia 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 Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC) y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente implementando dichas técnicas de codificación de vídeo.

[0003] Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca a las secuencias de vídeo. En la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo, una parte de una trama de vídeo, etc.) se puede dividir en bloques de vídeo, que también se pueden denominar bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. 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 predicción espacial con respecto a muestras de referencia de bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes se pueden denominar tramas, y las imágenes de referencia se pueden denominar tramas de referencia.

[0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales se pueden transformar desde el dominio de píxel a un dominio de transformada, dando como resultado coeficientes de transformada residuales que, a continuación, se pueden cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una matriz bidimensional, se pueden explorar para generar un vector unidimensional de coeficientes de transformada, y se puede aplicar codificación por entropía para lograr aún más compresión.

[0005] El documento de CHOI B ET AL: "MV-HEVC/SHVC HLS: Alignment of picture order counts", n.º JCT3V-F0053, 15 de octubre de 2013, divulga un esquema para mantener la alineación del valor de POC a través de capas en una unidad de acceso. En caso de restablecimiento de POC, el valor de MSB de POC de la imagen actual se establece en "0", los valores de POC de las imágenes en el búfer de imágenes descodificadas (DPB), que pertenecen a la misma capa, se cambian de acuerdo con un valor PicOrderCntOffset, para mantener las diferencias de POC con respecto a la imagen actual.

[0006] El documento de HANNUKSELA (NOKIA) MM: "MV-HEVC/SHVC HLS: on POC value derivation", n.º JCT3V-F0092, 15 de octubre de 2013, divulga una contribución a la norma SHVC, que permite imágenes IRAP no alineadas en todas las capas. Las cabeceras de fragmento contienen opcionalmente un valor de POC delta a nivel de capa, que se usa en la primera imagen de cada capa después de una imagen IDR de capa base. Se obtiene un valor de POC a nivel de capa para cada imagen de capa de mejora usando el valor de POC delta, que especifica la diferencia de la imagen de referencia anterior y la imagen actual de la misma capa.

[0007] El documento de LEE H ET AL: "MV-HEVC/SHVC HLS: On poc reset for long-term reference pictures", n.º JCT3V-F0047, 14 de octubre de 2013, divulga una contribución a la norma SHVC, proponiendo dos alternativas que permiten calcular el POC de una trama actual y una anterior, dependiendo de los valores señalizados "delta\_poc\_msb\_present\_flag" y "DeltaPocMsbCycleLt[]".

[0008] El documento de RAMASUBRAMONIAN A K ET AL: "MV-HEVC/SHVC HLS: On picture order count", n.º JCT3V-F0073, 15 de octubre de 2013, divulga una contribución a la norma SHVC, proponiendo dos banderas que

indican el reajuste de LSB de POC y de MSB de POC en la cabecera de fragmento, y un mensaje SEI de información de restablecimiento de POC adicional.

## BREVE EXPLICACIÓN

**[0009]** Cada uno de los sistemas, procedimientos y dispositivos de esta divulgación tiene varios aspectos innovadores, ninguno de los cuales es el único responsable de los atributos deseables divulgados en el presente documento.

**[0010]** De acuerdo con la invención, se proporciona un aparato configurado para descodificar información de vídeo como se expone en la reivindicación 1, así como un aparato configurado para codificar información de vídeo como se expone en la reivindicación 2.

**[0011]** Además, de acuerdo con la invención, se proporciona un procedimiento para descodificar información de vídeo como se expone en la reivindicación 7, así como un procedimiento para codificar información de vídeo como se expone en la reivindicación 8.

**[0012]** De acuerdo con la invención, también se proporciona un medio no transitorio legible por ordenador que comprende código que, cuando se ejecuta, hace que un aparato lleve a cabo una descodificación, como se expone en la reivindicación 12, o hace que un aparato lleve a cabo una codificación, como se expone en la reivindicación 13.

## BREVE DESCRIPCIÓN DE LOS DIBUJOS

### [0013]

La **FIG. 1A** es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que puede utilizar técnicas de acuerdo con aspectos descritos en esta divulgación.

La **FIG. 1B** es un diagrama de bloques que ilustra otro sistema de codificación y descodificación de vídeo de ejemplo que puede llevar a cabo técnicas de acuerdo con aspectos descritos en esta divulgación.

La **FIG. 2A** es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación.

La **FIG. 2B** es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación.

La **FIG. 3A** es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación.

La **FIG. 3B** es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación.

La **FIG. 4** es un diagrama de bloques que ilustra una configuración de ejemplo de imágenes en diferentes capas.

La **FIG. 5** es una tabla que ilustra valores de POC de imágenes en diferentes capas.

La **FIG. 6** es un diagrama de bloques que ilustra una configuración de ejemplo de imágenes en diferentes capas.

La **FIG. 7** es una tabla que ilustra valores de POC de imágenes en diferentes capas.

La **FIG. 8** es un diagrama de flujo que ilustra un procedimiento de codificación de información de vídeo, de acuerdo con un modo de realización de la presente divulgación.

## DESCRIPCIÓN DETALLADA

**[0014]** En general, esta divulgación se refiere a la gestión de POC para la codificación de vídeo multicapa en el contexto de códecs de vídeo avanzados, tal como HEVC (Codificación de Vídeo de Alta Eficacia). Más específicamente, la presente divulgación se refiere a sistemas y procedimientos para mejorar el rendimiento de la predicción entre capas en la ampliación de la codificación de vídeo escalable (SVC) de HEVC.

**[0015]** La codificación de vídeo escalable (SVC) se refiere a la codificación de vídeo en la que se usan una capa base (BL), a veces denominada capa de referencia (RL), y una o más capas de mejora (EL) escalables. En SVC, la capa base puede transportar datos de vídeo con un nivel de calidad base. Las una o más capas de mejora pueden transportar datos de vídeo adicionales para admitir, por ejemplo, niveles espaciales, temporales y/o de señal a ruido (SNR) más altos. Las capas de mejora se pueden definir con respecto a una capa codificada previamente. Por

ejemplo, una capa inferior puede servir como una BL, mientras que una capa superior puede servir como una EL. Las capas intermedias pueden servir como EL o RL, o como ambas. Por ejemplo, una capa intermedia (por ejemplo, una capa que ni es la capa más baja ni la capa más alta) puede ser una EL para las capas por debajo de la capa intermedia, tal como la capa base o cualquier capa de mejora interviniente, y al mismo tiempo servir como una RL para una o más capas de mejora por encima de la capa intermedia. De forma similar, en la ampliación multivista o 3D de la norma HEVC, puede haber múltiples vistas, y la información de una vista se puede utilizar para codificar (por ejemplo, codificar o descodificar) la información de otra vista (por ejemplo, estimación de movimiento, predicción de vector de movimiento y/u otras redundancias).

**[0016]** En algunos esquemas de codificación de vídeo se puede usar un recuento de orden de imágenes (POC) para indicar el orden en el que se deben emitir o mostrar las imágenes. Además, en algunas implementaciones, el valor del POC puede restablecerse (por ejemplo, establecerse en cero, establecerse en un valor señalado en el flujo de bits u obtenerse de la información incluida en el flujo de bits) siempre que determinados tipos de imágenes se procesen en el flujo de bits. Por ejemplo, cuando determinadas imágenes de punto de acceso aleatorio se procesan en el flujo de bits, el POC puede restablecerse.

**[0017]** Con el fin de mejorar la eficacia de la velocidad de bits, en algunas implementaciones solo se señala una parte del POC. Por ejemplo, los bits menos significativos (LSB) del POC pueden señalarse y los bits más significativos (MSB) del POC pueden calcularse (por ejemplo, determinarse) usando el tipo de unidad de capa de abstracción de red (NAL) de la imagen y los LSB y/o los MSB de imágenes que preceden a la imagen actual en orden de descodificación. En otras implementaciones, la información de MSB de POC se puede señalar en la ampliación de cabecera de fragmento de determinados tipos de imágenes, incluidas imágenes de acceso aleatorio limpio (CRA) e imágenes de acceso de enlace roto (BLA), para mejorar la resistencia a errores del proceso de obtención de POC. Sin embargo, el uso de la información de MSB de POC señalizada para calcular los valores de POC solo para imágenes CRA y BLA (con NoRasOutputFlag igual a un valor de 1) puede dar como resultado valores de POC incorrectos o dar como resultado valores de POC que no están alineados a través de múltiples capas en algunos casos. La alineación a través de múltiples capas (por ejemplo, alineación cruzada), por ejemplo, se produce cuando cada imagen de una unidad de acceso tiene el mismo valor de POC, de manera que las imágenes de la unidad de acceso se emiten simultáneamente o casi simultáneamente. Dicha falta de alineación violaría la restricción requerida por la norma de Codificación de Vídeo Escalable de Alta Eficacia (SHVC), que exige que las imágenes de una sola unidad de acceso tengan valores de POC idénticos.

**[0018]** Por tanto, se desea un procedimiento de codificación mejorado para obtener los valores de POC.

**[0019]** En la presente divulgación, se describen varias técnicas que pueden usarse para mejorar la obtención de POC para la codificación de vídeo multicapa. En algunos modos de realización de la presente divulgación, la información de MSB de POC (a veces denominada "ciclo de MSB de POC" en la presente divulgación) puede proporcionarse no solo con imágenes CRA y BLA sino también con imágenes de actualización de descodificador instantánea (IDR). Al proporcionar la información de MSB de POC en conexión con imágenes IDR, también se puede lograr resistencia a errores y una obtención de POC mejorada para imágenes IDR. En algunos modos de realización de la presente divulgación, la información de MSB de POC se puede utilizar de diferentes formas en función de si la información de MSB de POC se proporciona con una imagen de restablecimiento de POC o no. Por ejemplo, si la información de MSB de POC se proporciona con una imagen de restablecimiento de POC, la información de MSB de POC se puede usar para actualizar los valores de POC de imágenes en el búfer de imágenes descodificadas (DPB). Si la información de MSB de POC se proporciona con una imagen de no restablecimiento de POC, la información de MSB de POC se usa para calcular (por ejemplo, determinar) el POC de la imagen de no restablecimiento de POC. Al considerar si la imagen es una imagen de restablecimiento de POC, un codificador puede evitar que las imágenes en el DPB se desordenen después de un restablecimiento de POC.

**[0020]** En la siguiente descripción se describen técnicas de H.264/AVC relacionadas con determinados modos de realización, y también se analizan la norma HEVC y técnicas relacionadas. Aunque determinados modos de realización se describen en el presente documento en el contexto de las normas HEVC y/o H.264, un experto en la técnica puede apreciar que los sistemas y procedimientos divulgados en el presente documento se pueden aplicar a cualquier norma de codificación de vídeo adecuado. Por ejemplo, los modos de realización divulgados en el presente documento se pueden aplicar a una o más de las siguientes normas (por ejemplo, incluidas normas desarrolladas por el Grupo de Expertos en Codificación de Vídeo [VCEG] del Sector de Normalización de Telecomunicaciones de la Unión Internacional de Telecomunicaciones [UIT-T] por el Grupo de Expertos en Imágenes en Movimiento [MPEG] de la Organización Internacional de Normalización/Comisión Electrotécnica Internacional [ISO/IEC]): ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluidas sus ampliaciones de Codificación de Vídeo Escalable (SVC) y de Codificación de Vídeo Multivista (MVC).

**[0021]** HEVC sigue, en general, el marco de las normas de codificación de vídeo anteriores en muchos aspectos. La unidad de predicción en HEVC es diferente de las unidades de predicción (por ejemplo, macrobloques) en determinadas normas anteriores de codificación de vídeo. De hecho, el concepto de macrobloque no existe en HEVC como se entiende en determinadas normas anteriores de codificación de vídeo. Un macrobloque se

reemplaza por una estructura jerárquica basada en un esquema de árbol cuaternario, que puede proporcionar una gran flexibilidad, entre otros beneficios posibles. Por ejemplo, en el esquema de HEVC, se definen tres tipos de bloques, unidad de codificación (CU), unidad de predicción (PU) y unidad de transformada (TU). La CU se puede referir a la unidad básica de división en regiones. La CU se puede considerar análoga al concepto de macrobloque, pero HEVC no limita el tamaño máximo de las CU y puede permitir la división recursiva en cuatro CU de igual tamaño para mejorar la adaptabilidad del contenido. La PU se puede considerar la unidad básica de inter/intrapredicción, y una única PU puede contener múltiples particiones de forma arbitraria para codificar eficazmente patrones de imagen irregulares. La TU se puede considerar la unidad básica de transformada. La TU se puede definir independientemente de la PU; sin embargo, el tamaño de una TU puede estar limitado al tamaño de la CU a la que pertenece la TU. Esta separación de la estructura de bloques en tres conceptos diferentes puede permitir optimizar cada unidad de acuerdo con la función respectiva de la unidad, lo que puede dar como resultado una mejora de la eficacia de codificación.

**[0022]** Solamente con fines ilustrativos, determinados modos de realización divulgados en el presente documento se describen con ejemplos que solo incluyen dos capas (por ejemplo, una capa inferior tal como la capa base y una capa superior tal como la capa de mejora) de datos de vídeo. Una "capa" de datos de vídeo puede referirse, en general, a una secuencia de imágenes que tiene al menos una característica común, tal como una vista, una velocidad de trama, una resolución o similar. Por ejemplo, una capa puede incluir datos de vídeo asociados a una vista particular (por ejemplo, perspectiva) de datos de vídeo multivista. Como otro ejemplo, una capa puede incluir datos de vídeo asociados a una capa particular de datos de vídeo escalables. Por lo tanto, esta divulgación puede referirse indistintamente a una capa y a una vista de datos de vídeo. Es decir, una vista de datos de vídeo puede denominarse capa de datos de vídeo, y una capa de datos de vídeo puede denominarse vista de datos de vídeo. Además, un códec multicapa (también denominado codificador de vídeo multicapa o codificador-descodificador multicapa) puede referirse conjuntamente a un códec multivista o a un códec escalable (por ejemplo, un códec configurado para codificar y/o descodificar datos de vídeo usando MV-HEVC, 3D-HEVC, SHVC u otra técnica de codificación multicapa). La codificación de vídeo y la descodificación de vídeo pueden, en general, denominarse codificación de vídeo. Se debe entender que dichos ejemplos pueden ser aplicables a configuraciones que incluyen múltiples capas base y/o de mejora. Además, para facilitar la explicación, la siguiente divulgación incluye los términos "tramas" o "bloques" con referencia a determinados modos de realización. Sin embargo, estos términos no pretenden ser limitativos. Por ejemplo, las técnicas descritas a continuación se pueden usar con cualquier unidad de vídeo adecuada, tal como bloques (por ejemplo, CU, PU, TU, macrobloques, etc.), fragmentos, tramas, etc.

### **Normas de codificación de vídeo**

**[0023]** Una imagen digital, tal como una imagen de vídeo, una imagen de televisión, una imagen fija o una imagen generada por una grabadora de vídeo o un ordenador puede consistir en píxeles o muestras dispuestas en líneas horizontales y verticales. El número de píxeles en una sola imagen es, típicamente, de decenas de miles. Cada píxel contiene típicamente información de luminancia y crominancia. Sin compresión, la gran cantidad de información que se debe transmitir desde un codificador de imágenes a un descodificador de imágenes haría imposible la transmisión de imágenes en tiempo real. Para reducir la cantidad de información a transmitir, se ha desarrollado una serie de procedimientos de compresión diferentes, tales como las normas JPEG, MPEG y H.263.

**[0024]** Las normas de codificación de vídeo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluidas sus ampliaciones de Codificación de Vídeo Escalable (SVC) y de Codificación de Vídeo Multivista (MVC).

**[0025]** Además, se ha desarrollado una norma de codificación de vídeo, denominada HEVC, por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T VCEG e ISO/IEC MPEG. La cita completa del borrador 10 de HEVC es el documento JCTVC-L1003, de Bross *et al.*, titulado "High Efficiency Video Coding (HEVC) Text Specification Draft 10", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12ª conferencia: Ginebra, Suiza, del 14 de enero de 2013 al 23 de enero de 2013. La ampliación multivista para HEVC, a saber, MV-HEVC, y la ampliación escalable para HEVC, a saber, SHVC, también están siendo desarrolladas por el JCT-3V (Equipo de colaboración conjunta de ITU-T/ISO/IEC para el desarrollo de la ampliación de la codificación de vídeo 3D) y JCT-VC, respectivamente.

### **Sistema de codificación de vídeo**

**[0026]** A continuación, en el presente documento, se describe de forma más detallada diversos aspectos de los sistemas, aparatos y procedimientos novedosos, con referencia a los dibujos adjuntos. Sin embargo, esta divulgación se puede realizar de muchas formas diferentes y no se debería interpretar que está limitada a alguna estructura o función específica presentada a lo largo de esta divulgación. En cambio, estos aspectos se proporcionan de modo que esta divulgación sea exhaustiva y completa, y transmita por completo el alcance de la divulgación a los expertos en la técnica. En base a las enseñanzas del presente documento, un experto en la técnica debería apreciar que el alcance de la divulgación pretende abarcar cualquier aspecto de los sistemas, aparatos y procedimientos novedosos divulgados en el presente documento, ya sean implementados de forma independiente

de, o en combinación con, cualquier otro aspecto de la presente divulgación. Por ejemplo, un aparato se puede implementar o un procedimiento se puede llevar a la práctica usando un número cualquiera de los aspectos expuestos en el presente documento. Además, el alcance de la presente divulgación pretende abarcar dicho aparato o procedimiento que se lleva a la práctica usando otra estructura, funcionalidad o estructura y funcionalidad, además, o aparte, de los diversos aspectos de la presente divulgación expuestos en el presente documento. Se debe entender que cualquier aspecto divulgado en el presente documento se puede realizar mediante uno o más elementos de una reivindicación.

**[0027]** Aunque en el presente documento se describen aspectos particulares, muchas variantes y permutaciones de estos aspectos se hallan dentro del alcance de la divulgación. Aunque se mencionan algunos beneficios y ventajas de los aspectos preferentes, no se pretende limitar el alcance de la divulgación a unos beneficios, usos u objetivos particulares. En cambio, los aspectos de la divulgación pretenden ser ampliamente aplicables a diferentes tecnologías inalámbricas, configuraciones de sistema, redes y protocolos de transmisión, algunos de los cuales se ilustran a modo de ejemplo en las figuras y en la siguiente descripción de los aspectos preferentes. La descripción detallada y los dibujos son meramente ilustrativos de la divulgación, en lugar de limitantes, estando definido el alcance de la divulgación por las reivindicaciones adjuntas y equivalentes de las mismas.

**[0028]** Los dibujos adjuntos ilustran ejemplos. Los elementos indicados por números de referencia en los dibujos adjuntos corresponden a elementos indicados por números de referencia similares en la siguiente descripción. En esta divulgación, los elementos que tienen nombres que comienzan con palabras ordinales (por ejemplo, "primero", "segundo", "tercero", y así sucesivamente) no implican necesariamente que los elementos tienen un orden particular. En cambio, dichas palabras ordinales se usan simplemente para referirse a diferentes elementos de un mismo tipo o un tipo similar.

**[0029]** La **FIG. 1A** es un diagrama de bloques que ilustra un sistema de codificación de vídeo 10 de ejemplo que puede utilizar técnicas de acuerdo con aspectos descritos en esta divulgación. Como se usa y describe en el presente documento, el término "codificador de vídeo" se refiere, en general, tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" se pueden referir genéricamente a la codificación de vídeo y a la descodificación de vídeo. Además de los codificadores de vídeo y los descodificadores de vídeo, los aspectos descritos en la presente solicitud se pueden extender a otros dispositivos relacionados, tales como los transcodificadores (por ejemplo, dispositivos que pueden descodificar un flujo de bits y recodificar otro flujo de bits) y dispositivos intermedios (por ejemplo, dispositivos que pueden modificar, transformar y/o manipular de otro modo un flujo de bits).

**[0030]** Como se muestra en la **FIG. 1A**, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados que serán descodificados posteriormente por un dispositivo de destino 14. En el ejemplo de la **FIG. 1A**, el dispositivo de origen 12 y el dispositivo de destino 14 están en dispositivos diferentes; específicamente, el dispositivo de origen 12 es parte de un dispositivo de origen, y el dispositivo de destino 14 es parte de un dispositivo de destino. Sin embargo, cabe señalar que los módulos de origen y de destino 12, 14 pueden estar en el mismo dispositivo, o ser parte del mismo, como se muestra en el ejemplo de la **FIG. 1B**.

**[0031]** Con referencia de nuevo a la **FIG. 1A**, el dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender, respectivamente, cualquiera entre una amplia gama de dispositivos, incluidos ordenadores de escritorio, ordenadores tipo notebook (por ejemplo, ordenadores portátiles), tabletas electrónicas, descodificadores, equipos telefónicos tales como los denominados "teléfonos inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de flujo continuo de vídeo o similares. En diversos modos de realización, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

**[0032]** El dispositivo de destino 14 puede recibir, por medio de un enlace 16, los datos de vídeo codificados a descodificar. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de transferir los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En el ejemplo de la **FIG. 1A**, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden modularse de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global, tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

**[0033]** De forma alternativa, los datos codificados se pueden emitir desde una interfaz de salida 22 hasta un dispositivo de almacenamiento 31 opcional. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 31 mediante una interfaz de entrada 28, por ejemplo, del dispositivo de destino 14. El dispositivo de almacenamiento 31 puede incluir cualquiera entre una diversidad de medios de almacenamiento de

datos, de acceso distribuido o local, tales como una unidad de disco duro, memoria flash, memoria volátil o no volátil u otros medios adecuados cualesquiera de almacenamiento digital para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 31 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que pueda guardar el vídeo codificado generado mediante el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados del dispositivo de almacenamiento 31, por medio de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor que pueda almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Ejemplos de servidores de archivos incluyen un servidor web (por ejemplo, para un sitio web), un servidor de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectado a red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados mediante cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión a de red inalámbrica de área local (WLAN)), una conexión cableada (por ejemplo, una línea de abonado digital (DSL), un módem por cable, etc.), o una combinación de ambos que sea adecuada para acceder a los datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 31 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

**[0034]** Las técnicas de esta divulgación no se limitan a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, para admitir cualquiera de una diversidad de aplicaciones multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo mediante flujo continuo, por ejemplo, por medio de Internet (por ejemplo, transmisión continua adaptativa y dinámica a través del protocolo de transferencia de hipertexto (HTTP), etc.), codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo 10 se puede configurar para admitir transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

**[0035]** En el ejemplo de la **FIG. 1A**, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y la interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. En el dispositivo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo previamente capturado, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como vídeo de origen, o una combinación de dichas fuentes. Como un ejemplo, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos, tal como se ilustra en el ejemplo de la **FIG. 1B**. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o por cable.

**[0036]** El codificador de vídeo 20 puede codificar vídeo capturado, precapturado o generado por ordenador. Los datos de vídeo codificados pueden ser transmitidos directamente al dispositivo de destino 14 por medio de la interfaz de salida 22 del dispositivo de origen 12. De forma adicional o alternativa, los datos de vídeo codificados pueden almacenarse en el dispositivo de almacenamiento 31 para un acceso posterior por parte del dispositivo de destino 14 u otros dispositivos, para su descodificación y/o reproducción. El codificador de vídeo 20 ilustrado en las **FIGS. 1A** y **1B** pueden comprender el codificador de vídeo 20 ilustrado en la **FIG. 2A**, el codificador de vídeo 23 ilustrado en la **FIG. 2B**, o cualquier otro codificador de vídeo descrito en el presente documento.

**[0037]** En el ejemplo de la **FIG. 1A**, el dispositivo de destino 14 incluye la interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 puede recibir los datos de vídeo codificados a través del enlace 16 y/o desde el dispositivo de almacenamiento 31. Los datos de vídeo codificados comunicados a través del enlace 16 o proporcionados en el dispositivo de almacenamiento 31, pueden incluir una variedad de elementos de sintaxis generados por el codificador de vídeo 20, para su uso por un descodificador de vídeo, tal como el descodificador de vídeo 30, en la descodificación de los datos de vídeo. Dichos elementos de sintaxis se pueden incluir con los datos de vídeo codificados transmitidos en un medio de comunicación, almacenarse en un medio de almacenamiento o almacenarse en un servidor de archivos. El descodificador de vídeo 30 ilustrado en la **FIG. 1A** y la **FIG. 1B** puede comprender el descodificador de vídeo 30 ilustrado en la **FIG. 3A**, el descodificador de vídeo 33 ilustrado en la **FIG. 3B**, o cualquier otro descodificador de vídeo descrito en el presente documento.

**[0038]** El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interactuar con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 muestra los datos de vídeo descodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización,

tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0039] En aspectos relacionados, la **FIG. 1B** muestra un sistema de codificación y decodificación de vídeo 10' de ejemplo en el que los módulos de origen y de destino 12, 14 están en, o forman parte de, un dispositivo 11. El dispositivo 11 puede ser un equipo telefónico de mano, tal como un teléfono "inteligente" o similar. El dispositivo 11 puede incluir un dispositivo controlador/procesador 13 opcional en comunicación operativa con los módulos de origen y de destino 12, 14. El sistema 10' de la **FIG. 1B** puede incluir, además, una unidad de procesamiento de vídeo 21 entre el codificador de vídeo 20 y la interfaz de salida 22. En algunas implementaciones, la unidad de procesamiento de vídeo 21 es una unidad individual, como se ilustra en la **FIG. 1B**; sin embargo, en otras implementaciones, la unidad de procesamiento de vídeo 21 puede implementarse como una parte del codificador de vídeo 20 y/o del dispositivo procesador/controlador 13. El sistema 10' también puede incluir un rastreador 29 opcional, que puede rastrear un objeto de interés en una secuencia de vídeo. El objeto de interés a rastrear se puede segmentar mediante una técnica descrita en relación con uno o más aspectos de la presente divulgación. En aspectos relacionados, el rastreo se puede realizar mediante el dispositivo de visualización 32, por sí solo o junto con el rastreador 29. El sistema 10' de la **FIG. 1B** y sus componentes son, por lo demás, similares al sistema 10 de la **FIG. 1A** y sus componentes.

[0040] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma HEVC, y se pueden ajustar a un modelo de pruebas (HM) de HEVC. De forma alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con otras normas propietarias o de la industria, tales como la norma ITU-T H.264, denominada de forma alternativa MPEG-4, Parte 10, AVC, o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no se limitan a ninguna norma de codificación particular. Otros ejemplos de normas de compresión de vídeo incluyen MPEG-2 e ITU-T H.263.

[0041] Aunque no se muestra en los ejemplos de las **FIGS. 1A y 1B**, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar integrados, cada uno, con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o flujos de datos individuales. Si procede, en algunos ejemplos, las unidades MUX-DEMUX se pueden ajustar al protocolo de multiplexación ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0042] Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 se pueden implementar como cualquiera de una variedad de circuitos de codificador adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables por campo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 se pueden incluir en uno o más codificadores o decodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

#### **Proceso de codificación de vídeo**

[0043] Como se ha mencionado anteriormente de manera breve, el codificador de vídeo 20 codifica datos de vídeo. Los datos de vídeo pueden comprender una o más imágenes. Cada una de las imágenes es una imagen fija que forma parte de un vídeo. En algunos casos, una imagen se puede denominar "trama" de vídeo. Cuando el codificador de vídeo 20 codifica los datos de vídeo, el codificador de vídeo 20 puede generar un flujo de bits. El flujo de bits puede incluir una secuencia de bits que forma una representación codificada de los datos de vídeo. El flujo de bits puede incluir imágenes codificadas y datos asociados. Una imagen codificada es una representación codificada de una imagen.

[0044] Para generar el flujo de bits, el codificador de vídeo 20 puede realizar operaciones de codificación en cada imagen de los datos de vídeo. Cuando el codificador de vídeo 20 realiza operaciones de codificación en las imágenes, el codificador de vídeo 20 puede generar una serie de imágenes codificadas y datos asociados. Los datos asociados pueden incluir conjuntos de parámetros de vídeo (VPS), conjuntos de parámetros de secuencia (SPS), conjuntos de parámetros de imagen (PPS), conjuntos de parámetros de adaptación (APS) y otras estructuras sintácticas. Un SPS puede contener parámetros aplicables a cero o más secuencias de imágenes. Un PPS puede contener parámetros aplicables a cero o más imágenes. Un APS puede contener parámetros aplicables a cero o más imágenes. Los parámetros de un APS pueden ser parámetros que tienen más probabilidades de cambiar que los parámetros de un PPS.

[0045] Para generar una imagen codificada, el codificador de vídeo 20 puede dividir una imagen en bloques de vídeo de igual tamaño. Un bloque de vídeo puede ser una matriz bidimensional de muestras. Cada uno de los



bloques de vídeo está asociado a un bloque de árbol. En algunos casos, un bloque de árbol también se puede denominar unidad de codificación más grande (LCU). Los bloques de árbol de HEVC pueden ser, en términos generales, análogos a los macrobloques de normas anteriores, tales como la H.264/AVC. Sin embargo, un bloque de árbol no se limita necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). El codificador de vídeo 20 puede usar una división en árbol cuaternario para dividir los bloques de vídeo de bloques de árbol en bloques de vídeo asociados a las CU, de ahí el nombre "bloques de árbol".

**[0046]** En algunos ejemplos, el codificador de vídeo 20 puede dividir una imagen en una pluralidad de fragmentos. Cada uno de los fragmentos puede incluir un número entero de CU. En algunos casos, un fragmento comprende un número entero de bloques de árbol. En otros casos, un límite de un fragmento puede estar dentro de un bloque de árbol.

**[0047]** Como parte de realizar una operación de codificación en una imagen, el codificador de vídeo 20 puede realizar operaciones de codificación en cada fragmento de la imagen. Cuando el codificador de vídeo 20 realiza una operación de codificación en un fragmento, el codificador de vídeo 20 puede generar datos codificados asociados al fragmento. Los datos codificados asociados al fragmento se pueden denominar "fragmento codificado".

**[0048]** Para generar un fragmento codificado, el codificador de vídeo 20 puede realizar operaciones de codificación en cada bloque de árbol de un fragmento. Cuando el codificador de vídeo 20 realiza una operación de codificación en un bloque de árbol, el codificador de vídeo 20 puede generar un bloque de árbol codificado. El bloque de árbol codificado puede comprender datos que representan una versión codificada del bloque de árbol.

**[0049]** Cuando el codificador de vídeo 20 genera un fragmento codificado, el codificador de vídeo 20 puede realizar operaciones de codificación (es decir, codificar) en los bloques de árbol del fragmento de acuerdo con un orden de exploración de tramas. Por ejemplo, el codificador de vídeo 20 puede codificar los bloques de árbol del fragmento en un orden dirigido de izquierda a derecha en la fila más alta de los bloques de árbol del fragmento, a continuación de izquierda a derecha en la siguiente fila inferior de los bloques de árbol y así sucesivamente hasta que el codificador de vídeo 20 ha codificado cada uno de los bloques de árbol del fragmento.

**[0050]** Como resultado de codificar los bloques de árbol de acuerdo con el orden de exploración de tramas, los bloques de árbol por encima y a la izquierda de un bloque de árbol dado pueden haberse codificado, pero los bloques de árbol por debajo y a la derecha del bloque de árbol dado aún no se han codificado. En consecuencia, el codificador de vídeo 20 puede ser capaz de acceder a la información generada por la codificación de los bloques de árbol por encima y a la izquierda del bloque de árbol dado cuando se codifica el bloque de árbol dado. Sin embargo, el codificador de vídeo 20 puede no ser capaz de acceder a la información generada por la codificación de los bloques de árbol por debajo y a la derecha del bloque de árbol dado cuando se codifica el bloque de árbol dado.

**[0051]** Para generar un bloque de árbol codificado, el codificador de vídeo 20 puede realizar de forma recursiva una división en árbol cuaternario en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños se puede asociar a una CU diferente. Por ejemplo, el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, y así sucesivamente. Una CU dividida puede ser una CU cuyo bloque de vídeo está dividido en bloques de vídeo asociados a otras CU. Una CU no dividida puede ser una CU cuyo bloque de vídeo no está dividido en bloques de vídeo asociados a otras CU.

**[0052]** Uno o más elementos de sintaxis del flujo de bits pueden indicar un número máximo de veces que el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol. Un bloque de vídeo de una CU puede tener forma cuadrada. El tamaño del bloque de vídeo de una CU (es decir, el tamaño de la CU) puede variar desde 8x8 píxeles hasta el tamaño de un bloque de vídeo de un bloque de árbol (es decir, el tamaño del bloque de árbol) con un máximo de 64x64 píxeles o más.

**[0053]** El codificador de vídeo 20 puede realizar operaciones de codificación (por ejemplo, codificar) en cada CU de un bloque de árbol de acuerdo con un orden de exploración en z. En otras palabras, el codificador de vídeo 20 puede codificar una CU superior izquierda, una CU superior derecha, una CU inferior izquierda y, a continuación, una CU inferior derecha, en ese orden. Cuando el codificador de vídeo 20 realiza una operación de codificación en una CU dividida, el codificador de vídeo 20 puede codificar las CU asociadas a subbloques del bloque de vídeo de la CU dividida de acuerdo con el orden de exploración en z. En otras palabras, el codificador de vídeo 20 puede codificar una CU asociada a un subbloque superior izquierdo, una CU asociada a un subbloque superior derecho, una CU asociada a un subbloque inferior izquierdo y, a continuación, una CU asociada a un subbloque inferior derecho, en ese orden.

**[0054]** Como resultado de la codificación de las CU de un bloque de árbol de acuerdo con un orden de exploración en z, las CU por encima, por encima y a la izquierda, por encima y a la derecha, a la izquierda y por debajo y a la izquierda de una CU dada se pueden haber codificado. Las CU por debajo y a la derecha de la CU dada todavía no se han codificado. En consecuencia, el codificador de vídeo 20 puede ser capaz de acceder a información generada

por la codificación de algunas CU vecinas de la CU dada cuando se codifica la CU dada. Sin embargo, el codificador de vídeo 20 puede no ser capaz de acceder a información generada por la codificación de otras CU vecinas de la CU dada cuando se codifica la CU dada.

**[0055]** Cuando el codificador de vídeo 20 codifica una CU no dividida, el codificador de vídeo 20 puede generar una o más unidades de predicción (PU) para la CU. Cada una de las PU de la CU se puede asociar a un bloque de vídeo diferente dentro del bloque de vídeo de la CU. El codificador de vídeo 20 puede generar un bloque de vídeo predicho para cada PU de la CU. El bloque de vídeo predicho de una PU puede ser un bloque de muestras. El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar el bloque de vídeo predicho para una PU.

**[0056]** Cuando el codificador de vídeo 20 usa intrapredicción para generar el bloque de vídeo predicho de una PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU en base a unas muestras descodificadas de la imagen asociada a la PU. Si el codificador de vídeo 20 usa intrapredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU intrapredicha. Cuando el codificador de vídeo 20 usa interpredicción para generar el bloque de vídeo predicho de la PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU en base a unas muestras descodificadas de una o más imágenes distintas a la imagen asociada a la PU. Si el codificador de vídeo 20 usa interpredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU interpredicha.

**[0057]** Además, cuando el codificador de vídeo 20 usa interpredicción para generar un bloque de vídeo predicho para una PU, el codificador de vídeo 20 puede generar información de movimiento para la PU. La información de movimiento para una PU puede indicar uno o más bloques de referencia de la PU. Cada bloque de referencia de la PU puede ser un bloque de vídeo dentro de una imagen de referencia. La imagen de referencia puede ser una imagen distinta de la imagen asociada a la PU. En algunos casos, un bloque de referencia de una PU también se puede denominar "muestra de referencia" de la PU. El codificador de vídeo 20 puede generar el bloque de vídeo predicho para la PU en base a los bloques de referencia de la PU.

**[0058]** Después de que el codificador de vídeo 20 genere bloques de vídeo predichos para una o más PU de una CU, el codificador de vídeo 20 puede generar datos residuales para la CU en base a los bloques de vídeo predichos para las PU de la CU. Los datos residuales para la CU pueden indicar diferencias entre muestras en los bloques de vídeo predichos para las PU de la CU y el bloque de vídeo original de la CU.

**[0059]** Además, como parte de realizar una operación de codificación en una CU no dividida, el codificador de vídeo 20 puede realizar una división recursiva en árbol cuaternario en los datos residuales de la CU para dividir los datos residuales de la CU en uno o más bloques de datos residuales (es decir, bloques de vídeo residuales) asociados a las unidades de transformada (TU) de la CU. Cada TU de una CU puede estar asociada a un bloque de vídeo residual diferente.

**[0060]** El codificador de vídeo 20 puede aplicar una o más transformadas a los bloques de vídeo residuales asociados a las TU para generar bloques de coeficientes de transformada (por ejemplo, bloques de coeficientes de transformada) asociados a las TU. Conceptualmente, un bloque de coeficientes de transformada puede ser una matriz bidimensional (2D) de coeficientes de transformada.

**[0061]** Después de generar un bloque de coeficientes de transformada, el codificador de vídeo 20 puede realizar un proceso de cuantificación en el bloque de coeficientes de transformada. La cuantificación se refiere, en general, a un proceso en el que coeficientes de transformada se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformada, proporcionando más compresión. 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 se puede redondear a la baja hasta un coeficiente de transformada de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ .

**[0062]** El codificador de vídeo 20 puede asociar cada CU a un valor de parámetro de cuantificación (QP). El valor de QP asociado a una CU puede determinar cómo el codificador de vídeo 20 cuantifica los bloques de coeficientes de transformada asociados a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformada asociados a una CU ajustando el valor de QP asociado a la CU.

**[0063]** Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes de transformada, el codificador de vídeo 20 puede generar conjuntos de elementos de sintaxis que representan los coeficientes de transformada en el bloque de coeficientes de transformada cuantificados. El codificador de vídeo 20 puede aplicar operaciones de codificación por entropía, tales como operaciones de codificación aritmética binaria adaptable al contexto (CABAC), a algunos de estos elementos de sintaxis. También podrían usarse otras técnicas de codificación por entropía, tales como la codificación de longitud variable adaptable al contexto (CAVLC), la codificación por entropía de división en intervalos de probabilidad (PIPE) u otra codificación aritmética binaria.

**[0064]** El flujo de bits generado por el codificador de vídeo 20 puede incluir una serie de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL puede ser una estructura de sintaxis que contiene una indicación de un tipo de datos en la unidad NAL y octetos que contienen los datos. Por ejemplo, una unidad NAL puede contener datos que representan un conjunto de parámetros de vídeo, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un fragmento codificado, información de mejora complementaria (SEI), un delimitador de unidad de acceso, datos de relleno u otro tipo de datos. Los datos de una unidad NAL pueden incluir diversas estructuras de sintaxis.

**[0065]** El descodificador de vídeo 30 puede recibir el flujo de bits generado por el codificador de vídeo 20. El flujo de bits puede incluir una representación codificada de los datos de vídeo codificados por el codificador de vídeo 20. Cuando el descodificador de vídeo 30 recibe el flujo de bits, el descodificador de vídeo 30 puede realizar una operación de análisis sintáctico en el flujo de bits. Cuando el descodificador de vídeo 30 realiza la operación de análisis sintáctico, el descodificador de vídeo 30 puede extraer elementos de sintaxis del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo en base a los elementos de sintaxis extraídos del flujo de bits. El proceso para reconstruir los datos de vídeo en base a los elementos de sintaxis puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20 para generar los elementos de sintaxis.

**[0066]** Después de que el descodificador de vídeo 30 extrae los elementos de sintaxis asociados a una CU, el descodificador de vídeo 30 puede generar bloques de vídeo predichos para las PU de la CU en base a los elementos de sintaxis. Además, el descodificador de vídeo 30 puede realizar una cuantificación inversa en bloques de coeficientes de transformada asociados a las TU de la CU. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes de transformada para reconstruir bloques de vídeo residuales asociados a las TU de la CU. Después de generar los bloques de vídeo predichos y reconstruir los bloques de vídeo residuales, el descodificador de vídeo 30 puede reconstruir el bloque de vídeo de la CU en base a los bloques de vídeo predichos y los bloques de vídeo residuales. De esta manera, el descodificador de vídeo 30 puede reconstruir los bloques de vídeo de las CU en base a los elementos de sintaxis del flujo de bits.

#### **Codificador de vídeo**

**[0067]** La **FIG. 2A** es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación. El codificador de vídeo 20 se puede configurar para procesar una sola capa de una trama de vídeo, tal como para HEVC. Además, el codificador de vídeo 20 se puede configurar para realizar cualquiera o la totalidad de las técnicas de esta divulgación. Como ejemplo, la unidad de procesamiento de predicción 100 se puede configurar para realizar cualquiera o todas las técnicas descritas en la presente divulgación. En otro modo de realización, el codificador de vídeo 20 incluye una unidad de predicción entre capas 128 opcional que está configurada para realizar cualquiera o todas las técnicas descritas en esta divulgación. En otros modos de realización, la predicción entre capas se puede realizar por la unidad de procesamiento de predicción 100 (por ejemplo, la unidad de interpredicción 121 y/o la unidad de intrapredicción 126), en cuyo caso la unidad 128 de predicción entre capas se puede omitir. Sin embargo, los aspectos de esta divulgación no se limitan a esto. En algunos ejemplos, las técnicas descritas en la presente divulgación se pueden compartir entre los diversos componentes del codificador de vídeo 20. En algunos ejemplos, de forma adicional o alternativa, un procesador (no mostrado) se puede configurar para realizar cualquiera o todas las técnicas descritas en esta divulgación.

**[0068]** Con propósitos explicativos, la presente divulgación describe un 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. El ejemplo representado en la **FIG. 2A** es para un códec de una sola capa. Sin embargo, como se describirá en mayor detalle con respecto a la **FIG. 2B**, una parte o la totalidad del codificador de vídeo 20 se puede duplicar para procesar un códec multicapa.

**[0069]** El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal de un vídeo dentro de tramas o imágenes contiguas de una secuencia de vídeo. Intramodo (modo I) se puede referir a cualquiera de varios modos de codificación de base espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), se pueden referir a cualquiera de varios modos de codificación de base temporal.

**[0070]** En el ejemplo de la **FIG. 2A**, el codificador de vídeo 20 incluye una pluralidad de componentes funcionales. Los componentes funcionales del codificador de vídeo 20 incluyen una unidad de procesamiento de predicción 100, una unidad de generación de residuo 102, una unidad de procesamiento de transformada 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de transformada inversa 110, una unidad de reconstrucción 112, una unidad de filtro 113, un búfer de imágenes descodificadas 114 y una unidad de codificación por entropía 116. La unidad de procesamiento de predicción 100 incluye una unidad de interpredicción 121, una unidad de estimación de movimiento 122, una unidad de compensación de movimiento 124, una unidad de

intrapredicción 126 y una unidad de predicción entre capas 128. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales. Además, la unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden estar altamente integradas, pero están representadas en el ejemplo de la **FIG. 2A** de forma separada con fines explicativos.

**[0071]** El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede recibir los datos de vídeo desde diversas fuentes. Por ejemplo, el codificador de vídeo 20 puede recibir los datos de vídeo desde la fuente de vídeo 18 (por ejemplo, mostrada en la **FIG. 1A** o **1B**) u otra fuente. Los datos de vídeo pueden representar una serie de imágenes. Para codificar los datos de vídeo, el codificador de vídeo 20 puede realizar una operación de codificación en cada una de las imágenes. Como parte del modo de realización de la operación de codificación en una imagen, el codificador de vídeo 20 puede realizar operaciones de codificación en cada fragmento de la imagen. Como parte de realizar una operación de codificación en un fragmento, el codificador de vídeo 20 puede realizar operaciones de codificación en bloques de árbol en el fragmento.

**[0072]** Como parte de realizar una operación de codificación en un bloque de árbol, la unidad de procesamiento de predicción 100 puede realizar la división en árbol cuaternario en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños se puede asociar a una CU diferente. Por ejemplo, la unidad de procesamiento de predicción 100 puede dividir un bloque de vídeo de un bloque de árbol en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, etc.

**[0073]** Los tamaños de los bloques de vídeo asociados a las CU pueden variar desde muestras de 8x8 hasta el tamaño del bloque de árbol, con un máximo de 64x64 muestras o más. En esta divulgación, "NxN" y "N por N" se pueden usar de manera intercambiable para hacer referencia a las dimensiones de muestras de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16x16 muestras o 16 por 16 muestras. En general, un bloque de vídeo de 16x16 tiene dieciséis muestras en una dirección vertical ( $y = 16$ ) y dieciséis muestras en una dirección horizontal ( $x = 16$ ). Asimismo, un bloque de NxN tiene, en general, N muestras en una dirección vertical y N muestras en una dirección horizontal, donde N representa un valor entero no negativo.

**[0074]** Además, como parte de realizar la operación de codificación en un bloque de árbol, la unidad de procesamiento de predicción 100 puede generar una estructura de datos jerárquica en árbol cuaternario para el bloque de árbol. Por ejemplo, un bloque de árbol puede corresponder a un nodo raíz de la estructura de datos en árbol cuaternario. Si la unidad de procesamiento de predicción 100 divide el bloque de vídeo del bloque de árbol en cuatro subbloques, el nodo raíz tiene cuatro nodos hijo en la estructura de datos en árbol cuaternario. Cada uno de los nodos hijo corresponde a una CU asociada a uno de los subbloques. Si la unidad de procesamiento de predicción 100 divide uno de los subbloques en cuatro subsubbloques, el nodo correspondiente a la CU asociada al subbloque puede tener cuatro nodos hijo, cada uno de los cuales corresponde a una CU asociada a uno de los subsubbloques.

**[0075]** Cada nodo de la estructura de datos en árbol cuaternario puede contener datos de sintaxis (por ejemplo, elementos de sintaxis) para el bloque de árbol o CU correspondiente. Por ejemplo, un nodo en el árbol cuaternario puede incluir una bandera de división, que indica si el bloque de vídeo de la CU correspondiente al nodo está dividido (es decir, partido) en cuatro subbloques. Los elementos de sintaxis para una CU se pueden definir de manera recursiva y pueden depender de si el bloque de vídeo de la CU está dividido en subbloques. Una CU cuyo bloque de vídeo no está dividido puede corresponder a un nodo hoja en la estructura de datos en árbol cuaternario. Un bloque de árbol codificado puede incluir datos basados en la estructura de datos en árbol cuaternario para un bloque de árbol correspondiente.

**[0076]** El codificador de vídeo 20 puede realizar operaciones de codificación en cada CU no dividida de un bloque de árbol. Cuando el codificador de vídeo 20 realiza una operación de codificación en una CU no dividida, el codificador de vídeo 20 genera datos que representan una representación codificada de la CU no dividida.

**[0077]** Como parte de realizar una operación de codificación en una CU, la unidad de procesamiento de predicción 100 puede dividir el bloque de vídeo de la CU entre una o más PU de la CU. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir diversos tamaños de PU. Suponiendo que el tamaño de una CU en particular sea  $2N_x2N_y$ , el codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir tamaños de PU de  $2N_x2N_y$  o  $N_xN_y$  e interpredicción en tamaños de PU simétricos de  $2N_x2N_y$ ,  $2N_xN_y$ ,  $N_x2N_y$ ,  $N_xN_y$ ,  $2N_xnU$ ,  $nL_x2N_y$ ,  $nR_x2N_y$  o similares. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden admitir la división asimétrica para tamaños de PU de  $2N_xnU$ ,  $2N_xnD$ ,  $nL_x2N_y$  y  $nR_x2N_y$ . En algunos ejemplos, la unidad de procesamiento de predicción 100 puede realizar una división geométrica para dividir el bloque de vídeo de una CU entre las PU de la CU a lo largo de un límite que no coincide con los lados del bloque de vídeo de la CU en ángulos rectos.

**[0078]** La unidad de interpredicción 121 puede realizar interpredicción en cada PU de la CU. La interpredicción puede proporcionar compresión temporal. Para realizar la interpredicción en una PU, la unidad de estimación de movimiento 122 puede generar información de movimiento para la PU. La unidad de compensación de movimiento 124 puede generar un bloque de vídeo predicho para la PU en base a la información de movimiento y las muestras

descodificadas de imágenes distintas a la imagen asociada a la CU (por ejemplo, imágenes de referencia). En esta divulgación, un bloque de vídeo predicho generado por la unidad de compensación de movimiento 124 se puede denominar bloque de vídeo interpredicho.

**[0079]** Los fragmentos pueden ser fragmentos I, fragmentos P o fragmentos B. La unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden 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 I, todas las PU se intrapredicen. Por lo tanto, si la PU está en un fragmento I, la unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 no realizan la interpredicción en la PU.

**[0080]** Si la PU está en un fragmento P, la imagen que contiene la PU se asocia a una lista de imágenes de referencia denominada "lista 0". Cada una de las imágenes de referencia de la lista 0 contiene muestras que se pueden usar para la interpredicción de otras imágenes. Cuando la unidad de estimación de movimiento 122 realiza la operación de estimación de movimiento con respecto a una PU en un fragmento P, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia en la lista 0 para un bloque de referencia para la PU. El bloque de referencia de la PU puede ser un conjunto de muestras, por ejemplo, un bloque de muestras, que se corresponde más estrechamente con las muestras del bloque de vídeo de la PU. La unidad de estimación de movimiento 122 puede usar una variedad de métricas para determinar cuán estrechamente un conjunto de muestras en una imagen de referencia se corresponde con las muestras del bloque de vídeo de una PU. Por ejemplo, la unidad de estimación de movimiento 122 puede determinar cuán estrechamente un conjunto de muestras en una imagen de referencia se corresponde con las muestras en el bloque de vídeo de una PU mediante una suma de diferencias absolutas (SAD), una suma de diferencias al cuadrado (SSD) u otras métricas de diferencia.

**[0081]** Después de identificar un bloque de referencia de una PU en un fragmento P, la unidad de estimación de movimiento 122 puede generar un índice de referencia que indica la imagen de referencia de la lista 0 que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia. En diversos ejemplos, la unidad de estimación de movimiento 122 puede generar vectores de movimiento con diferentes grados de precisión. Por ejemplo, la unidad de estimación de movimiento 122 puede generar vectores de movimiento con una precisión de un cuarto de muestra, una precisión de un octavo de muestra u otra precisión de fracción de muestra. En el caso de la precisión de fracción de muestra, los valores de bloque de referencia se pueden interpolar a partir de valores enteros de posición de muestra en la imagen de referencia. La unidad de estimación de movimiento 122 puede proporcionar el índice de referencia y el vector de movimiento como información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar un bloque de vídeo predicho de la PU en base al bloque de referencia identificado por la información de movimiento de la PU.

**[0082]** Si la PU está en un fragmento B, la imagen que contiene la PU puede estar asociada a dos listas de imágenes de referencia, denominadas "lista 0" y "lista 1". En algunos ejemplos, una imagen que contiene un fragmento B puede estar asociada a una combinación de listas que es una combinación de la lista 0 y la lista 1.

**[0083]** Además, si la PU está en un fragmento B, la unidad de estimación de movimiento 122 puede llevar a cabo la predicción unidireccional o la predicción bidireccional para la PU. Cuando la unidad de estimación de movimiento 122 realiza la predicción unidireccional para la PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia de la lista 0 o la lista 1 para un bloque de referencia para la PU. La unidad de estimación de movimiento 122 puede entonces generar un índice de referencia que indica la imagen de referencia en la lista 0 o la lista 1 que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia. La unidad de estimación de movimiento 122 puede emitir el índice de referencia, un indicador de dirección de predicción y el vector de movimiento como información de movimiento de la PU. El indicador de dirección de predicción puede indicar si el índice de referencia indica una imagen de referencia en la lista 0 o la lista 1. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU en base al bloque de referencia indicado por la información de movimiento de la PU.

**[0084]** Cuando la unidad de estimación de movimiento 122 realiza la predicción bidireccional para una PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia de la lista 0 para un bloque de referencia para la PU y también puede buscar las imágenes de referencia de la lista 1 para otro bloque de referencia para la PU. La unidad de estimación de movimiento 122 puede entonces generar índices de referencia que indican las imágenes de referencia en la lista 0 y la lista 1 que contienen los bloques de referencia y los vectores de movimiento que indican desplazamientos espaciales entre los bloques de referencia y la PU. La unidad de estimación de movimiento 122 puede proporcionar los índices de referencia y los vectores de movimiento de la PU como información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU en base a los bloques de referencia indicados por la información de movimiento de la PU.

**[0085]** En algunos casos, la unidad de estimación de movimiento 122 no proporciona un conjunto completo de información de movimiento para una PU a la unidad de codificación por entropía 116. En cambio, la unidad de estimación de movimiento 122 puede señalar la información de movimiento de una PU con referencia a la información de movimiento de otra PU. Por ejemplo, la unidad de estimación de movimiento 122 puede determinar

que la información de movimiento de la PU es suficientemente similar a la información de movimiento de una PU vecina. En este ejemplo, la unidad de estimación de movimiento 122 puede indicar, en una estructura de sintaxis asociada a la PU, un valor que indica al decodificador de vídeo 30 que la PU tiene la misma información de movimiento que la PU vecina. En otro ejemplo, la unidad de estimación de movimiento 122 puede identificar, en una estructura de sintaxis asociada a la PU, una PU vecina y una diferencia de vectores de movimiento (MVD). La diferencia de vectores de movimiento indica una diferencia entre el vector de movimiento de la PU y el vector de movimiento de la PU vecina indicada. El decodificador de vídeo 30 puede usar el vector de movimiento de la PU vecina indicada y la diferencia de vectores de movimiento para determinar el vector de movimiento de la PU. Con referencia a la información de movimiento de una primera PU cuando se señala la información de movimiento de una segunda PU, el codificador de vídeo 20 puede ser capaz de señalar la información de movimiento de la segunda PU usando menos bits.

**[0086]** Como se analizará con más detalle posteriormente con referencia a la FIG. 8, la unidad de procesamiento de predicción 100 puede configurarse para codificar (por ejemplo, codificar o decodificar) la PU (o cualquier otra capa de referencia y/o bloques de capa de mejora o unidades de vídeo) realizando los procedimientos ilustrados en la FIG. 8. Por ejemplo, la unidad de interpredicción 121 (por ejemplo, a través de la unidad de estimación de movimiento 122 y/o la unidad de compensación de movimiento 124), la unidad de intrapredicción 126, o la unidad de predicción entre capas 128 pueden configurarse para realizar los procedimientos ilustrados en la FIG. 8 ya sea juntos o por separado.

**[0087]** Como parte de realizar una operación de codificación en una CU, la unidad de intrapredicción 126 puede llevar a cabo intrapredicción en las PU de la CU. La intrapredicción puede proporcionar compresión espacial. Cuando la unidad de intrapredicción 126 realiza intrapredicción en una PU, la unidad de intrapredicción 126 puede generar datos de predicción para la PU en base a muestras decodificadas de otras PU en la misma imagen. Los datos de predicción para la PU pueden incluir un bloque de vídeo predicho y diversos elementos de sintaxis. La unidad de intrapredicción 126 puede realizar intrapredicción en PU en fragmentos I, fragmentos P y fragmentos B.

**[0088]** Para realizar la intrapredicción en una PU, la unidad de intrapredicción 126 puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos de predicción para la PU. Cuando la unidad de intrapredicción 126 usa un modo de intrapredicción para generar un conjunto de datos de predicción para la PU, la unidad de intrapredicción 126 puede extender muestras de bloques de vídeo de PU vecinas a través del bloque de vídeo de la PU en una dirección y/o gradiente asociados al modo de intrapredicción. Las PU vecinas pueden estar por encima, por encima y a la derecha, por encima y a la izquierda o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha, de arriba abajo para las PU, CU y bloques de árbol. La unidad de intrapredicción 126 puede usar diversos números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccional, dependiendo del tamaño de la PU.

**[0089]** La unidad de procesamiento de predicción 100 puede seleccionar los datos de predicción para una PU de entre los datos de predicción generados por la unidad de compensación de movimiento 124 para la PU o los datos de predicción generados por la unidad de intrapredicción 126 para la PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos de predicción para la PU en base a las métricas de velocidad/distorsión de los conjuntos de datos de predicción.

**[0090]** Si la unidad de procesamiento de predicción 100 selecciona datos de predicción generados por la unidad de intrapredicción 126, la unidad de procesamiento de predicción 100 puede señalar el modo de intrapredicción que se usó para generar los datos de predicción para las PU, es decir, el modo de intrapredicción seleccionado. La unidad de procesamiento de predicción 100 puede señalar el modo de intrapredicción seleccionado de varias maneras. Por ejemplo, es probable que el modo de intrapredicción seleccionado sea el mismo que el modo de intrapredicción de una PU vecina. En otras palabras, el modo de intrapredicción de la PU vecina puede ser el modo más probable para la PU actual. De este modo, la unidad de procesamiento de predicción 100 puede generar un elemento de sintaxis para indicar que el modo de intrapredicción seleccionado es el mismo que el modo de intrapredicción de la PU vecina.

**[0091]** Como se analiza anteriormente, el codificador de vídeo 20 puede incluir la unidad de predicción entre capas 128. La unidad de predicción entre capas 128 está configurada para predecir un bloque actual (por ejemplo, un bloque actual en la EL) usando una o más capas diferentes que están disponibles en SVC (por ejemplo, una capa base o de referencia). Dicha predicción se puede denominar predicción entre capas. La unidad de predicción entre capas 128 utiliza procedimientos de predicción para reducir la redundancia entre capas, mejorando así la eficacia de codificación y reduciendo los requisitos de recursos computacionales. Algunos ejemplos de predicción entre capas incluyen la intrapredicción entre capas, la predicción de movimiento entre capas y la predicción de residuo entre capas. La intrapredicción entre capas usa la reconstrucción de bloques ubicados conjuntamente en la capa base para predecir el bloque actual en la capa de mejora. La predicción de movimiento entre capas usa información de movimiento de la capa base para predecir el movimiento en la capa de mejora. La predicción de residuo entre capas usa el residuo de la capa base para predecir el residuo de la capa de mejora. Cada uno de los esquemas de predicción entre capas se analiza a continuación con mayor detalle.

**[0092]** Después de que la unidad de procesamiento de predicción 100 seleccione los datos de predicción para las PU de una CU, la unidad de generación de residuo 102 puede generar datos residuales para la CU restando (por ejemplo, indicado por el signo menos) los bloques de vídeo predichos de las PU de la CU al bloque de vídeo de la CU. Los datos residuales de una CU pueden incluir bloques de vídeo residuales 2D que corresponden a diferentes componentes de muestra de las muestras en el bloque de vídeo de la CU. Por ejemplo, los datos residuales pueden incluir un bloque de vídeo residual que corresponde a diferencias entre componentes de luminancia de muestras en los bloques de vídeo predichos de las PU de la CU y componentes de luminancia de muestras en el bloque de vídeo original de la CU. Además, los datos residuales de la CU pueden incluir bloques de vídeo residuales que corresponden a las diferencias entre componentes de crominancia de muestras en los bloques de vídeo predichos de las PU de la CU y las componentes de crominancia de las muestras en el bloque de vídeo original de la CU.

**[0093]** La unidad de procesamiento de predicción 100 puede realizar la división en árbol cuaternario para dividir los bloques de vídeo residuales de una CU en subbloques. Cada bloque de vídeo residual no dividido puede estar asociado a una TU diferente de la CU. Los tamaños y posiciones de los bloques de vídeo residuales asociados a las TU de una CU pueden estar basados, o no, en los tamaños y posiciones de bloques de vídeo asociados a las PU de la CU. Una estructura en árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados a cada uno de los bloques de vídeo residuales. Las TU de una CU pueden corresponder a nodos hoja del RQT.

**[0094]** La unidad de procesamiento de transformada 104 puede generar uno o más bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a un bloque de vídeo residual asociado a la TU. Cada uno de los bloques de coeficientes de transformada puede ser una matriz 2D de coeficientes de transformada. La unidad de procesamiento de transformada 104 puede aplicar diversas transformadas al bloque de vídeo residual 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 vídeo residual asociado a una TU.

**[0095]** Después de que la unidad de procesamiento de transformada 104 genere un bloque de coeficientes de transformada asociado a una TU, la unidad de cuantificación 106 puede cuantificar los coeficientes de transformada en el bloque de coeficientes de transformada. La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes de transformada asociado a una TU de una CU en base a un valor de QP asociado a la CU.

**[0096]** El codificador de vídeo 20 puede asociar un valor de QP a una CU de diversas maneras. Por ejemplo, el codificador de vídeo 20 puede realizar un análisis de velocidad-distorsión en un bloque de árbol asociado a la CU. En el análisis de velocidad-distorsión, el codificador de vídeo 20 puede generar múltiples representaciones codificadas del bloque de árbol realizando una operación de codificación múltiples veces en el bloque de árbol. El codificador de vídeo 20 puede asociar diferentes valores de QP a la CU cuando el codificador de vídeo 20 genera diferentes representaciones codificadas del bloque de árbol. El codificador de vídeo 20 puede señalar que un valor de QP dado está asociado a la CU cuando el valor de QP dado está asociado a la CU en una representación codificada del bloque de árbol que tiene una velocidad de bits y una métrica de distorsión más bajas.

**[0097]** La unidad de cuantificación inversa 108 y la unidad de transformada inversa 110 pueden aplicar la cuantificación inversa y transformadas inversas al bloque de coeficientes de transformada, respectivamente, para reconstruir un bloque de vídeo residual a partir del bloque de coeficientes de transformada. La unidad de reconstrucción 112 puede añadir el bloque de vídeo residual reconstruido a muestras correspondientes de uno o más bloques de vídeo predichos generados por la unidad de procesamiento de predicción 100 para producir un bloque de vídeo reconstruido asociado a una TU. Reconstruyendo bloques de vídeo para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir el bloque de vídeo de la CU.

**[0098]** Después de que la unidad de reconstrucción 112 reconstruya el bloque de vídeo de una CU, la unidad de filtro 113 puede realizar una operación de eliminación de efecto pixelado para reducir los artefactos de efecto pixelado en el bloque de vídeo asociado a la CU. Después de realizar la una o más operaciones de eliminación de efecto pixelado, la unidad de filtro 113 puede almacenar el bloque de vídeo reconstruido de la CU en el búfer de imágenes descodificadas 114. La unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden usar una imagen de referencia que contiene el bloque de vídeo reconstruido para realizar la interpredicción en las PU de las imágenes posteriores. Además, la unidad de intrapredicción 126 puede usar bloques de vídeo reconstruidos en el búfer de imágenes descodificadas 114 para realizar la intrapredicción en otras PU en la misma imagen que la CU.

**[0099]** La unidad de codificación por entropía 116 puede recibir datos desde otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación por entropía 116 puede recibir bloques de coeficientes de transformada desde la unidad de cuantificación 106 y puede recibir elementos de sintaxis desde la unidad de procesamiento de predicción 100. Cuando la unidad de codificación por entropía 116 recibe los datos, la unidad de codificación por entropía 116 puede realizar una o más operaciones de codificación por entropía para generar datos codificados por entropía. Por ejemplo, el codificador de vídeo 20 puede realizar una operación de codificación de longitud variable adaptable al contexto (CAVLC), una operación CABAC, una operación de

codificación de longitud de variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una operación de codificación por entropía de división en intervalos de probabilidad (PIPE) u otro tipo de operación de codificación por entropía en los datos. La unidad de codificación por entropía 116 puede emitir un flujo de bits que incluye los datos codificados por entropía.

**[0100]** Como parte de realizar una operación de codificación por entropía en los datos, la unidad de codificación por entropía 116 puede seleccionar un modelo de contexto. Si la unidad de codificación por entropía 116 está realizando una operación CABAC, el modelo de contexto puede indicar estimaciones de probabilidades de que bins particulares tengan unos valores particulares. En el contexto de CABAC, el término "bin" se usa para hacer referencia a un bit de una versión binarizada de un elemento de sintaxis.

### **Codificador de vídeo multicapa**

**[0101]** La **FIG. 2B** es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo multicapa 23 (también denominado simplemente codificador de vídeo 23) que puede implementar técnicas de acuerdo con los aspectos descritos en esta divulgación. El codificador de vídeo 23 se puede configurar para procesar tramas de vídeo multicapa, tal como para SHVC y codificación multivista. Además, el codificador de vídeo 23 se puede configurar para realizar cualquiera o la totalidad de las técnicas de esta divulgación.

**[0102]** El codificador de vídeo 23 incluye un codificador de vídeo 20A y un codificador de vídeo 20B, cada uno de los cuales se puede configurar como el codificador de vídeo 20 y puede realizar las funciones descritas anteriormente con respecto al codificador de vídeo 20. Además, como se indica mediante la reutilización de números de referencia, los codificadores de vídeo 20A y 20B pueden incluir, al menos, algunos de los sistemas y subsistemas, como el codificador de vídeo 20. Aunque se ilustra que el codificador de vídeo 23 incluye dos codificadores de vídeo 20A y 20B, el codificador de vídeo 23 no está limitado como tal y puede incluir cualquier número de capas de codificador de vídeo 20. En algunos modos de realización, el codificador de vídeo 23 puede incluir un codificador de vídeo 20 para cada imagen o trama en una unidad de acceso. Por ejemplo, una unidad de acceso que incluye cinco imágenes se puede procesar o codificar mediante un codificador de vídeo que incluye cinco capas de codificador. En algunos modos de realización, el codificador de vídeo 23 puede incluir más capas de codificador que tramas en una unidad de acceso. En algunos de dichos casos, algunas de las capas de codificador de vídeo pueden estar inactivas al procesar algunas unidades de acceso.

**[0103]** Además de los codificadores de vídeo 20A y 20B, el codificador de vídeo 23 puede incluir una unidad de remuestreo 90. La unidad de remuestreo 90 puede, en algunos casos, realizar muestrear de manera ascendente la capa base de una trama de vídeo recibida para, por ejemplo, crear una capa de mejora. La unidad de remuestreo 90 puede muestrear de manera ascendente información particular asociada a la capa base recibida de una trama, pero no otra información. Por ejemplo, la unidad de remuestreo 90 puede muestrear de manera ascendente el tamaño espacial o el número de píxeles de la capa base, pero el número de fragmentos o el recuento de orden de imágenes puede permanecer constante. En algunos casos, la unidad de remuestreo 90 puede no procesar el vídeo recibido y/o puede ser opcional. Por ejemplo, en algunos casos, la unidad de procesamiento de predicción 100 puede realizar un muestreo ascendente. En algunos modos de realización, la unidad de remuestreo 90 está configurada para muestrear de manera ascendente una capa y reorganizar, redefinir, modificar o ajustar uno o más fragmentos para cumplir con un conjunto de reglas de límite de fragmento y/o reglas de exploración de trama. Aunque se describe principalmente como el muestreo ascendente de una capa base, o de una capa inferior en una unidad de acceso, en algunos casos, la unidad de remuestreo 90 puede muestrear una capa de manera descendente. Por ejemplo, si durante la transmisión continua de un vídeo se reduce el ancho de banda, una trama muestrearse de manera descendente en lugar de ascendente.

**[0104]** La unidad de remuestreo 90 se puede configurar para recibir una imagen o trama (o información de imagen asociada a la imagen) desde el búfer de imágenes descodificadas 114 del codificador de capa inferior (por ejemplo, el codificador de vídeo 20A) y para muestrear de manera ascendente la imagen (o la información de imagen recibida). Esta imagen muestreada de manera ascendente se puede proporcionar a continuación a la unidad de procesamiento de predicción 100 de un codificador de capa superior (por ejemplo, el codificador de vídeo 20B) configurado para codificar una imagen en la misma unidad de acceso que el codificador de capa inferior. En algunos casos, el codificador de capa superior es una capa eliminada del codificador de capa inferior. En otros casos, puede haber uno o más codificadores de capa superior entre el codificador de vídeo de capa 0 y el codificador de capa 1 de la **FIG. 2B**.

**[0105]** En algunos casos, la unidad de remuestreo 90 se puede omitir o ignorar. En dichos casos, la imagen del búfer de imágenes descodificadas 114 del codificador de vídeo 20A se puede proporcionar directamente, o al menos sin proporcionarse a la unidad de remuestreo 90, a la unidad de procesamiento de predicción 100 del codificador de vídeo 20B. Por ejemplo, si los datos de vídeo proporcionados al codificador de vídeo 20B y la imagen de referencia del búfer de imágenes descodificadas 114 del codificador de vídeo 20A son del mismo tamaño o resolución, la imagen de referencia se puede proporcionar al codificador de vídeo 20B sin ningún remuestreo.



**[0106]** En algunos modos de realización, el codificador de vídeo 23 muestrea de manera descendente los datos de vídeo que se deben proporcionar al codificador de capa inferior usando la unidad de muestreo descendente 94 antes de proporcionar los datos de vídeo al codificador de vídeo 20A. De forma alternativa, la unidad de muestreo descendente 94 puede ser una unidad de remuestreo 90 capaz de realizar un muestreo ascendente o un muestreo descendente de los datos de vídeo. Aún en otros modos de realización, la unidad de muestreo descendente 94 se puede omitir.

**[0107]** Como se ilustra en la **FIG. 2B**, el codificador de vídeo 23 puede incluir además un multiplexor 98 o mux. El mux 98 puede proporcionar un flujo de bits combinado desde el codificador de vídeo 23. El flujo de bits combinado se puede crear tomando un flujo de bits de cada uno de los codificadores de vídeo 20A y 20B y alternando qué flujo de bits se proporciona en un momento dado. Mientras que en algunos casos los bits de los dos (o más en el caso de más de dos capas de codificador de vídeo) flujos de bits se pueden alternar un bit a la vez, en muchos casos los flujos de bits se combinan de manera diferente. Por ejemplo, el flujo de bits de salida se puede crear alternando el flujo de bits seleccionado un bloque a la vez. En otro ejemplo, el flujo de bits de salida se puede crear proporcionando una relación de bloques diferente a 1:1 de cada uno de los codificadores de vídeo 20A y 20B. Por ejemplo, se pueden proporcionar dos bloques desde el codificador de vídeo 20B para cada salida de bloque del codificador de vídeo 20A. En algunos modos de realización, el flujo de salida del mux 98 puede estar preprogramado. En otros modos de realización, el mux 98 puede combinar los flujos de bits de los codificadores de vídeo 20A, 20B en base a una señal de control recibida desde un sistema externo al codificador de vídeo 23, tal como desde un procesador en un dispositivo de origen que incluye el dispositivo de origen 12. La señal de control se puede generar en base a la resolución o velocidad de bits de un vídeo de la fuente de vídeo 18, en base al ancho de banda del enlace 16, en base a una suscripción asociada a un usuario (por ejemplo, una suscripción de pago frente a una suscripción gratuita) o en base a cualquier otro factor para determinar una salida de resolución deseada del codificador de vídeo 23.

## **Descodificador de vídeo**

**[0108]** La **FIG. 3A** es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación. El descodificador de vídeo 30 se puede configurar para procesar una sola capa de una trama de vídeo, tal como para HEVC. Además, el descodificador de vídeo 30 se puede configurar para realizar una cualquiera o la totalidad de las técnicas de esta divulgación. Como ejemplo, la unidad de compensación de movimiento 162 y/o la unidad de intrapredicción 164 se pueden configurar para realizar una cualquiera o la totalidad de las técnicas descritas en esta divulgación. En un modo de realización, el descodificador de vídeo 30 puede incluir opcionalmente una unidad de predicción entre capas 166 que está configurada para realizar una cualquiera o la totalidad de las técnicas descritas en esta divulgación. En otros modos de realización, la predicción entre capas se puede realizar por la unidad de procesamiento de predicción 152 (por ejemplo, la unidad de compensación de movimiento 162 y/o la unidad de intrapredicción 164), en cuyo caso la unidad de predicción entre capas 166 se puede omitir. Sin embargo, los aspectos de esta divulgación no se limitan a esto. En algunos ejemplos, las técnicas descritas en esta divulgación se pueden compartir entre los diversos componentes del descodificador de vídeo 30. En algunos ejemplos, de forma adicional o alternativa, un procesador (no mostrado) se puede configurar para realizar cualquiera o todas las técnicas descritas en esta divulgación.

**[0109]** Con propósitos explicativos, esta divulgación describe un descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación. El ejemplo representado en la **FIG. 3A** es para un códec de una sola capa. Sin embargo, como se describirá en mayor detalle con respecto a la **FIG. 3B**, una parte o la totalidad del descodificador de vídeo 30 se puede duplicar para el procesamiento de un códec multicapa.

**[0110]** En el ejemplo de la **FIG. 3A**, el descodificador de vídeo 30 incluye una pluralidad de componentes funcionales. Los componentes funcionales del descodificador de vídeo 30 incluyen una unidad de descodificación por entropía 150, una unidad de procesamiento de predicción 152, una unidad de cuantificación inversa 154, una unidad de transformada inversa 156, una unidad de reconstrucción 158, una unidad de filtro 159 y un búfer de imágenes descodificadas 160. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 162, una unidad de intrapredicción 164 y una unidad de predicción entre capas 166. En algunos ejemplos, el descodificador de vídeo 30 puede realizar una pasada de descodificación, en general, recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la **FIG. 2A**. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

**[0111]** El descodificador de vídeo 30 puede recibir un flujo de bits que comprende datos de vídeo codificados. El flujo de bits puede incluir una pluralidad de elementos de sintaxis. Cuando el descodificador de vídeo 30 recibe el flujo de bits, la unidad de descodificación por entropía 150 puede realizar una operación de análisis sintáctico en el flujo de bits. Como resultado de realizar la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación por entropía 150 puede extraer elementos de sintaxis del flujo de bits. Como parte de realizar la operación de análisis sintáctico, la unidad de descodificación por entropía 150 puede realizar la descodificación por entropía de elementos de sintaxis codificados por entropía en el flujo de bits. La unidad de procesamiento de

predicción 152, la unidad de cuantificación inversa 154, la unidad de transformada inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 159 pueden realizar una operación de reconstrucción que genera datos de vídeo descodificados en base a los elementos de sintaxis extraídos del flujo de bits.

5 **[0112]** Como se analiza anteriormente, el flujo de bits puede comprender una serie de unidades NAL. Las unidades NAL del flujo de bits pueden incluir unidades NAL de conjunto de parámetros de vídeo, unidades NAL de conjunto de parámetros de secuencia, unidades NAL de conjunto de parámetros de imagen, unidades NAL SEI, etc. Como parte de realizar la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican por entropía conjuntos de parámetros de secuencia a partir de unidades NAL de conjuntos de parámetros de secuencia, conjuntos de parámetros de imagen a partir de unidades NAL de conjuntos de parámetros de imagen, datos SEI a partir de unidades NAL SEI, etc.

15 **[0113]** Además, las unidades NAL del flujo de bits pueden incluir unidades NAL de fragmentos codificados. Como parte de realizar la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican por entropía fragmentos codificados de las unidades NAL de fragmentos codificados. Cada uno de los fragmentos codificados puede incluir una cabecera de fragmento y datos de fragmento. La cabecera de fragmento puede contener elementos de sintaxis pertenecientes a un fragmento. Los elementos de sintaxis de la cabecera de fragmento pueden incluir un elemento de sintaxis que identifica un conjunto de parámetros de imagen asociado a una imagen que contiene el fragmento. La unidad de descodificación por entropía 150 puede realizar operaciones de descodificación por entropía, tales como operaciones de descodificación de CABAC, en elementos de sintaxis de la cabecera de fragmento codificado para recuperar la cabecera de fragmento.

25 **[0114]** Como parte de extraer los datos de fragmento a partir de unidades NAL de fragmento codificado, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen elementos de sintaxis de las CU codificadas en los datos de fragmento. Los elementos de sintaxis extraídos pueden incluir elementos de sintaxis asociados a bloques de coeficientes de transformada. La unidad de descodificación por entropía 150 puede entonces realizar operaciones de descodificación de CABAC en algunos de los elementos de sintaxis.

35 **[0115]** Después de que la unidad de descodificación por entropía 150 realice una operación de análisis sintáctico en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en la CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Al realizar la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir un bloque de vídeo residual asociado a la CU.

40 **[0116]** Como parte de realizar una operación de reconstrucción en una TU, la unidad de cuantificación inversa 154 puede cuantificar de forma inversa, es decir, descuantificar, un bloque de coeficientes de transformada asociado a la TU. La unidad de cuantificación inversa 154 puede cuantificar de forma inversa el bloque de coeficientes de transformada de una manera similar a los procesos de cuantificación inversa propuestos para HEVC o definidos por la norma de descodificación H.264. La unidad de cuantificación inversa 154 puede usar un parámetro de cuantificación QP calculado por el codificador de vídeo 20 para una CU del bloque de coeficientes de transformada para determinar un grado de cuantificación y, del mismo modo, un grado de cuantificación inversa para que la unidad de cuantificación inversa 154 lo aplique.

50 **[0117]** Después de que la unidad de cuantificación inversa 154 cuantifique de forma inversa un bloque de coeficientes de transformada, la unidad de transformada inversa 156 puede generar un bloque de vídeo residual para la TU asociada al bloque de coeficientes de transformada. La unidad de transformada inversa 156 puede aplicar una transformada inversa al bloque de coeficientes de transformada para generar el bloque de vídeo residual para la TU. Por ejemplo, la unidad de transformada inversa 156 puede aplicar una DCT inversa, una transformada de número entero 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 de transformada. En algunos ejemplos, la unidad de transformada inversa 156 puede determinar una transformada inversa que se debe aplicar al bloque de coeficientes de transformada en base a la señalización del codificador de vídeo 20. En dichos ejemplos, la unidad de transformada inversa 156 puede determinar la transformada inversa en base a una transformada señalizada en el nodo raíz de un árbol cuaternario para un bloque de árbol asociado al bloque de coeficientes de transformada. En otros ejemplos, la unidad de transformada inversa 156 puede inferir la transformada inversa a partir de una o más características de codificación, tales como tamaño de bloque, modo de codificación o similares. En algunos ejemplos, la unidad de transformada inversa 156 puede aplicar una transformada inversa en cascada.

65 **[0118]** En algunos ejemplos, la unidad de compensación de movimiento 162 puede refinar el bloque de vídeo predicho de una PU mediante interpolación en base a filtros de interpolación. Los identificadores para los filtros de interpolación que se van a usar para la compensación de movimiento con una precisión de submuestra se pueden incluir en los elementos de sintaxis. La unidad de compensación de movimiento 162 puede usar los mismos filtros de

interpolación usados por el codificador de vídeo 20 durante la generación del bloque de vídeo predicho de la PU para calcular valores interpolados para muestras fraccionarias de un bloque de referencia. La unidad de compensación de movimiento 162 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 de acuerdo con la información de sintaxis recibida y usar los filtros de interpolación para generar el bloque de vídeo predicho.

**[0119]** Como se analiza adicionalmente más adelante con referencia a la **FIG. 8**, la unidad de procesamiento de predicción 152 puede codificar (por ejemplo, codificar o decodificar) la PU (o cualquier otra capa de referencia y/o bloques de capa de mejora o unidades de vídeo) realizando los procedimientos ilustrados en la **FIG. 8**. Por ejemplo, la unidad de compensación de movimiento 162, la unidad de intrapredicción 164 o la unidad de predicción entre capas 166 pueden configurarse para realizar los procedimientos ilustrados en la **FIG. 8**, ya sea juntos o por separado.

**[0120]** Si se codifica una PU usando intrapredicción, la unidad de intrapredicción 164 puede realizar intrapredicción para generar un bloque de vídeo predicho para la PU. Por ejemplo, la unidad de intrapredicción 164 puede determinar un modo de intrapredicción para la PU en base a elementos de sintaxis del flujo de bits. El flujo de bits puede incluir elementos de sintaxis que la unidad de intrapredicción 164 puede usar para determinar el modo de intrapredicción de la PU.

**[0121]** En algunos casos, los elementos de sintaxis pueden indicar que la unidad de intrapredicción 164 debe usar el modo de intrapredicción de otra PU para determinar el modo de intrapredicción de la PU actual. Por ejemplo, puede ser probable que el modo de intrapredicción de la PU actual sea el mismo que el modo de intrapredicción de una PU vecina. En otras palabras, el modo de intrapredicción de la PU vecina puede ser el modo más probable para la PU actual. Por lo tanto, en este ejemplo, el flujo de bits puede incluir un pequeño elemento de sintaxis que indica que el modo de intrapredicción de la PU es el mismo que el modo de intrapredicción de la PU vecina. La unidad de intrapredicción 164 puede entonces usar el modo de intrapredicción para generar datos de predicción (por ejemplo, muestras predichas) para la PU en base a los bloques de vídeo de PU espacialmente vecinas.

**[0122]** Como se analiza anteriormente, el decodificador de vídeo 30 también puede incluir una unidad de predicción entre capas 166. La unidad de predicción entre capas 166 está configurada para predecir un bloque actual (por ejemplo, un bloque actual en la EL) usando una o más capas diferentes que están disponibles en SVC (por ejemplo, una capa base o de referencia). Dicha predicción se puede denominar predicción entre capas. La unidad de predicción entre capas 166 utiliza procedimientos de predicción para reducir la redundancia entre capas, mejorando así la eficacia de codificación y reduciendo los requisitos de recursos computacionales. Algunos ejemplos de predicción entre capas incluyen la intrapredicción entre capas, la predicción de movimiento entre capas y la predicción de residuo entre capas. La intrapredicción entre capas usa la reconstrucción de bloques ubicados conjuntamente en la capa base para predecir el bloque actual en la capa de mejora. La predicción de movimiento entre capas usa información de movimiento de la capa base para predecir el movimiento en la capa de mejora. La predicción de residuo entre capas usa el residuo de la capa base para predecir el residuo de la capa de mejora. Cada uno de los esquemas de predicción entre capas se analiza a continuación con mayor detalle.

**[0123]** La unidad de reconstrucción 158 puede usar los bloques de vídeo residuales asociados a las TU de una CU y los bloques de vídeo predichos de las PU de la CU, es decir, datos de intrapredicción o datos de interpredicción, según corresponda, para reconstruir el bloque de vídeo de la CU. Por tanto, el decodificador de vídeo 30 puede generar un bloque de vídeo predicho y un bloque de vídeo residual en base a elementos de sintaxis del flujo de bits y puede generar un bloque de vídeo en base al bloque de vídeo predicho y el bloque de vídeo residual.

**[0124]** Después de que la unidad de reconstrucción 158 reconstruya el bloque de vídeo de la CU, la unidad de filtro 159 puede realizar una operación de eliminación de efecto pixelado para reducir los artefactos de efecto pixelado asociados a la CU. Después de que la unidad de filtro 159 realice una operación de eliminación de efecto pixelado para reducir los artefactos de efecto pixelado asociados a la CU, el decodificador de vídeo 30 puede almacenar el bloque de vídeo de la CU en el búfer de imágenes descodificadas 160. El búfer de imágenes descodificadas 160 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intrapredicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la **FIG. 1A** o **1B**. Por ejemplo, el decodificador de vídeo 30 puede realizar, en base a los bloques de vídeo del búfer de imágenes descodificadas 160, operaciones de intrapredicción o de interpredicción en las PU de otras CU.

### **Descodificador multicapa**

**[0125]** La **FIG. 3B** es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo multicapa 33 (también denominado simplemente decodificador de vídeo 33) que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación. El decodificador de vídeo 33 se puede configurar para procesar tramas de vídeo multicapa, tal como para SHVC y la codificación multivista. Además, el decodificador de vídeo 33 se puede configurar para realizar una cualquiera o la totalidad de las técnicas de esta divulgación.

**[0126]** El descodificador de vídeo 33 incluye un descodificador de vídeo 30A y un descodificador de vídeo 30B, cada uno de los cuales se puede configurar como el descodificador de vídeo 30 y puede realizar las funciones descritas anteriormente con respecto al descodificador de vídeo 30. Además, como se indica mediante la reutilización de números de referencia, los descodificadores de vídeo 30A y 30B pueden incluir, al menos, algunos de los sistemas y subsistemas, como el descodificador de vídeo 30. Aunque se ilustra que el descodificador de vídeo 33 incluye dos descodificadores de vídeo 30A y 30B, el descodificador de vídeo 33 no está limitado como tal y puede incluir cualquier número de capas de descodificador de vídeo 30. En algunos modos de realización, el descodificador de vídeo 33 puede incluir un descodificador de vídeo 30 para cada imagen o trama en una unidad de acceso. Por ejemplo, una unidad de acceso que incluye cinco imágenes se puede procesar o descodificar por un descodificador de vídeo que incluye cinco capas de descodificador. En algunos modos de realización, el descodificador de vídeo 33 puede incluir más capas de descodificador que tramas en una unidad de acceso. En algunos de dichos casos, algunas de las capas de descodificador de vídeo pueden estar inactivas al procesar algunas unidades de acceso.

**[0127]** Además de los descodificadores de vídeo 30A y 30B, el descodificador de vídeo 33 puede incluir una unidad de muestreo ascendente 92. En algunos modos de realización, la unidad de muestreo ascendente 92 puede muestrear de manera ascendente una capa base de una trama de vídeo recibida para crear una capa mejorada que se añadirá a la lista de imágenes de referencia para la trama o la unidad de acceso. Esta capa mejorada se puede almacenar en el búfer de imágenes descodificadas 160. En algunos modos de realización, la unidad de muestreo ascendente 92 puede incluir algunos o todos los modos de realización descritos con respecto a la unidad de remuestreo 90 de la **FIG. 2A**. En algunos modos de realización, la unidad de muestreo ascendente 92 está configurada para muestrear de manera ascendente una capa y reorganizar, redefinir, modificar o ajustar uno o más fragmentos para cumplir con un conjunto de reglas de límite de fragmento y/o reglas de exploración de trama. En algunos casos, la unidad de muestreo ascendente 92 puede ser una unidad de remuestreo configurada para muestrear de manera ascendente y/o descendente una capa de una trama de vídeo recibida.

**[0128]** La unidad de muestreo ascendente 92 puede estar configurada para recibir una imagen o trama (o información de imagen asociada a la imagen) desde el búfer de imágenes descodificadas 160 del descodificador de capa inferior (por ejemplo, el descodificador de vídeo 30A) y para muestrear de manera ascendente la imagen (o la información de imagen recibida). Esta imagen muestreada de manera ascendente se puede proporcionar a continuación a la unidad de procesamiento de predicción 152 de un descodificador de capa superior (por ejemplo, el descodificador de vídeo 30B) configurado para descodificar una imagen en la misma unidad de acceso que el descodificador de capa inferior. En algunos casos, el descodificador de capa superior es una capa eliminada del descodificador de capa inferior. En otros casos, puede haber uno o más descodificadores de capa superior entre el descodificador de capa 0 y el descodificador de capa 1 de la **FIG. 3B**.

**[0129]** En algunos casos, la unidad de muestreo ascendente 92 se puede omitir o ignorar. En dichos casos, la imagen del búfer de imágenes descodificadas 160 del descodificador de vídeo 30A se puede proporcionar directamente, o al menos sin proporcionarse a la unidad de muestreo ascendente 92, a la unidad de procesamiento de predicción 152 del descodificador de vídeo 30B. Por ejemplo, si los datos de vídeo proporcionados al descodificador de vídeo 30B y la imagen de referencia del búfer de imágenes descodificadas 160 del descodificador de vídeo 30A son del mismo tamaño o resolución, la imagen de referencia se puede proporcionar al descodificador de vídeo 30B sin muestreo ascendente. Además, en algunos modos de realización, la unidad de muestreo ascendente 92 puede ser una unidad de remuestreo 90 configurada para realizar muestreo ascendente o muestreo descendente en una imagen de referencia recibida desde el búfer de imágenes descodificadas 160 del descodificador de vídeo 30A.

**[0130]** Como se ilustra en la **FIG. 3B**, el descodificador de vídeo 33 puede incluir además un desmultiplexor 99 o demux. El demux 99 puede dividir un flujo de bits de vídeo codificado en múltiples flujos de bits, proporcionándose cada flujo de bits emitido por el demux 99 a un descodificador de vídeo 30A y 30B diferente. Los múltiples flujos de bits se pueden crear recibiendo un flujo de bits y cada uno de los descodificadores de vídeo 30A y 30B recibe una parte del flujo de bits en un momento dado. Mientras que en algunos casos los bits del flujo de bits recibido en el demux 99 se pueden alternar un bit a la vez entre cada uno de los descodificadores de vídeo (por ejemplo, los descodificadores de vídeo 30A y 30B en el ejemplo de la **FIG. 3B**), en muchos casos el flujo de bits se divide de manera diferente. Por ejemplo, el flujo de bits se puede dividir alternando qué descodificador de vídeo recibe el flujo de bits un bloque a la vez. En otro ejemplo, el flujo de bits se puede dividir por una relación de bloques diferente a 1:1 para cada uno de los descodificadores de vídeo 30A y 30B. Por ejemplo, se pueden proporcionar dos bloques al descodificador de vídeo 30B para cada bloque proporcionado al descodificador de vídeo 30A. En algunos modos de realización, la división del flujo de bits por el demux 99 se puede preprogramar. En otros modos de realización, el desmultiplexor 99 puede dividir el flujo de bits en base a una señal de control recibida desde un sistema externo al descodificador de vídeo 33, tal como desde un procesador en un dispositivo de destino que incluye el dispositivo de destino 14. La señal de control se puede generar en base a la resolución o la velocidad de bits de un vídeo de la interfaz de entrada 28, en base al ancho de banda del enlace 16, en base a una suscripción asociada a un usuario (por ejemplo, una suscripción de pago frente a una suscripción gratuita) o en base a cualquier otro factor para determinar una resolución obtenible por el descodificador de vídeo 33.

#### **Imágenes de punto de acceso aleatorio interno (IRAP)**

**[0131]** Algunos esquemas de codificación de vídeo pueden proporcionar puntos de acceso aleatorios en todo el flujo de bits de modo que el flujo de bits se puede descodificar comenzando desde cualquiera de esos puntos de acceso aleatorio sin necesidad de descodificar ninguna imagen que preceda a esos puntos de acceso aleatorio. En dichos esquemas de codificación de vídeo, todas las imágenes que siguen a un punto de acceso aleatorio en orden de salida (por ejemplo, incluidas aquellas imágenes que están en la misma unidad de acceso que la imagen que proporciona el punto de acceso aleatorio) se pueden descodificar correctamente sin usar ninguna imagen que preceda al punto de acceso aleatorio. Por ejemplo, incluso si una parte del flujo de bits se pierde durante la transmisión o durante la descodificación, un descodificador puede reanudar la descodificación del flujo de bits comenzando desde el próximo punto de acceso aleatorio. La compatibilidad con el acceso aleatorio puede facilitar, por ejemplo, servicios dinámicos de transmisión continua, operaciones de búsqueda, conmutación de canales, etc.

**[0132]** En algunos esquemas de codificación, dichos puntos de acceso aleatorio se pueden proporcionar por imágenes que se denominan imágenes de punto de acceso aleatorio interno (IRAP). Por ejemplo, un punto de acceso aleatorio (por ejemplo, proporcionado por una imagen IRAP de capa de mejora) en una capa de mejora ("capaA") contenida en una unidad de acceso ("auA") puede proporcionar acceso aleatorio específico de capa de modo que para cada capa de referencia ("capaB") de la capaA (por ejemplo, donde una capa de referencia es una capa que se usa para predecir la capaA) que tiene un punto de acceso aleatorio contenido en una unidad de acceso ("auB") que está en la capaB y precede a auA en orden de descodificación (o a un punto de acceso aleatorio contenido en la auA), las imágenes en la capaA que siguen a la auB en orden de salida (incluidas las imágenes ubicadas en la auB) se pueden descodificar correctamente sin necesidad de descodificar ninguna imagen de la capaA que preceda a la auB.

**[0133]** Las imágenes IRAP pueden codificarse usando intrapredicción (por ejemplo, codificarse sin hacer referencia a otras imágenes), y pueden incluir, por ejemplo, imágenes IDR, imágenes CRA e imágenes BLA. Cuando hay una imagen IDR en el flujo de bits, todas las imágenes que preceden a la imagen IDR en orden de descodificación no se usan para la predicción mediante las imágenes que siguen a la imagen IDR en orden de descodificación. Cuando hay una imagen CRA en el flujo de bits, las imágenes que siguen a la imagen CRA pueden usar, o no, imágenes que preceden a la imagen CRA en orden de descodificación para la predicción. Las imágenes que siguen a la imagen CRA en orden de descodificación, pero que usan imágenes que preceden a la imagen CRA en orden de descodificación, se pueden denominar imágenes principales omitidas de acceso aleatorio (RASL). Otro tipo de imagen que sigue a una imagen IRAP en orden de descodificación y que precede a la imagen IRAP en orden de salida es una imagen principal descodificable de acceso aleatorio (RADL), que puede no contener referencias a ninguna imagen que preceda a la imagen IRAP en orden de descodificación. Las imágenes RASL se pueden descartar por el descodificador si las imágenes que preceden a la imagen CRA no están disponibles. Una imagen BLA indica al descodificador que las imágenes que preceden a la imagen BLA pueden no estar disponibles para el descodificador (por ejemplo, porque dos flujos de bits están unidos entre sí y la imagen BLA es la primera imagen del segundo flujo de bits en orden de descodificación). Una unidad de acceso (por ejemplo, un grupo de imágenes que consisten en todas las imágenes codificadas asociadas al mismo tiempo de salida en las múltiples capas) que contiene una imagen de capa base (por ejemplo, una imagen que tiene un valor de ID de capa de 0) que es una imagen IRAP se puede denominar unidad de acceso IRAP.

#### **Alineación de capas cruzadas de imágenes IRAP**

**[0134]** En SVC, es posible que no se requiera que las imágenes IRAP estén alineadas (por ejemplo, contenidas en la misma unidad de acceso) en diferentes capas. Por ejemplo, si se requiere que las imágenes IRAP estén alineadas, cualquier unidad de acceso que contenga al menos una imagen IRAP solo contendría imágenes IRAP. Por otro lado, si no se requiere que las imágenes IRAP estén alineadas, en una sola unidad de acceso, una imagen (por ejemplo, en una primera capa) puede ser una imagen IRAP, y otra imagen (por ejemplo, en una segunda capa) puede ser una imagen no IRAP. Tener dichas imágenes IRAP no alineadas en un flujo de bits puede proporcionar algunas ventajas. Por ejemplo, en un flujo de bits de dos capas, si hay más imágenes IRAP en la capa base que en la capa de mejora, en aplicaciones de radiodifusión y multidifusión, se puede lograr un bajo retardo de ajuste y una alta eficacia de codificación.

**[0135]** En algunos esquemas de codificación de vídeo, se puede usar un recuento de orden de imágenes (POC) para realizar un seguimiento del orden relativo en el que se muestran las imágenes descodificadas. Algunos de dichos esquemas de codificación pueden hacer que los valores de POC se restablezcan (por ejemplo, se establezcan en cero o se establezcan en algún valor señalado en el flujo de bits) siempre que determinados tipos de imágenes se procesen en el flujo de bits. Por ejemplo, los valores de POC de determinadas imágenes IRAP pueden restablecerse, lo que hace que los valores de POC de otras imágenes que preceden a esas imágenes IRAP en orden de descodificación también se restablezcan. Esto puede ser problemático cuando no se requiere que las imágenes IRAP se alineen en las diferentes capas. Por ejemplo, cuando una imagen ("picA") es una imagen IRAP y otra imagen ("picB") en la misma unidad de acceso no es una imagen IRAP, el valor de POC de una imagen ("picC"), que se restablece debido a que picA sea una imagen IRAP, en la capa que contiene picA puede ser diferente del valor de POC de una imagen ("picD"), que no se ha restablecido, en la capa que contiene picB, donde picC y picD están en la misma unidad de acceso. Esto hace que picC y picD tengan diferentes valores de POC aunque

pertenezcan a la misma unidad de acceso (por ejemplo, al mismo tiempo de salida). Por tanto, en este ejemplo, el proceso de obtención para obtener los valores de POC de picC y picD se puede modificar para producir valores de POC que sean coherentes con la definición de valores de POC y unidades de acceso.

## 5 **Recuento de orden de imágenes (POC)**

**[0136]** Como se analizó anteriormente, el valor de un recuento de orden de imágenes (POC) (por ejemplo, PicOrderCntVal en HEVC) para una imagen codificada particular denota el orden relativo de la imagen codificada particular en el proceso de salida de imágenes con respecto a otras imágenes en la misma secuencia de vídeo codificado. En algunos modos de realización, el POC comprende bits menos significativos (LSB) y bits más significativos (MSB), y el POC se puede obtener concatenando los MSB y los LSB. En otros modos de realización, el POC se puede obtener sumando el valor de MSB y el valor de LSB. Los LSB pueden señalizarse en la cabecera de fragmento, y los MSB pueden calcularse por el codificador o el descodificador en base al tipo de unidad NAL de la imagen actual y los MSB y los LSB de una o más imágenes previas en orden de descodificación que (1) son imágenes no RASL o RADL, (2) son no desechables (por ejemplo, imágenes marcadas como "desechables", lo que indica que ninguna otra imagen depende de ellas, lo que permite eliminarlas para satisfacer las restricciones de ancho de banda), (3) no son imágenes de subcapa que no son de referencia (por ejemplo, imágenes que no se usan como referencia en otras imágenes en la misma subcapa temporal o en la misma capa), (4) tienen un ID temporal (por ejemplo, un ID de subcapa temporal) igual a un valor de 0. Dichas imágenes descritas en (1)-(4) pueden denominarse en el presente documento imágenes de anclaje POC. De manera similar, imágenes que tienen un valor de ID temporal mayor que un valor de 0, las imágenes RASL o RADL, las imágenes desechables o las imágenes de subcapa que no son de referencia pueden denominarse imágenes de no anclaje POC. Las imágenes de anclaje POC pueden incluir además imágenes que un codificador y/o un descodificador no pueden elegir eliminar del flujo de bits (por ejemplo, para satisfacer una restricción de ancho de banda). Las imágenes de anclaje POC pueden incluir además cualquier imagen distinta de los tipos de imágenes que un codificador y/o un descodificador pueden estar configurados para eliminar del flujo de bits (por ejemplo, para satisfacer una restricción de ancho de banda). Las imágenes de no anclaje POC pueden incluir cualquier imagen que no sea una imagen de anclaje POC.

**[0137]** Cuando la imagen actual es (1) una imagen IRAP con NoRaslOutputFlag (por ejemplo, una bandera que indica que las imágenes RASL no se deben proporcionar si está establecido en un valor de 1 e indica que las imágenes RASL se deben proporcionar si está establecido en un valor de 0) igual a un valor de 1 o (2) una imagen CRA que es la primera imagen del flujo de bits, se infiere que el valor de MSB de POC es igual a un valor de 0. Tal como se ha descrito anteriormente, en un flujo de bits multicapa (por ejemplo, un flujo de bits SHVC o MV-HEVC con más de una capa), puede haber unidades de acceso (AU) donde una o más imágenes son imágenes IRAP y otra u otras imágenes son imágenes no IRAP, y dichas AU pueden denominarse "AU de IRAP no alineadas". Al descodificar flujos de bits que contienen AU de IRAP no alineadas, es posible (y probable) que los POC obtenidos en base a los valores de LSB de POC señalizados en el flujo de bits incumplan el requisito de conformidad de flujo de bits de que todas las imágenes en una unidad de acceso deberían tener el mismo valor de POC.

**[0138]** En algunos modos de realización, una bandera de restablecimiento de POC (por ejemplo, poc\_reset\_flag) se puede usar para restablecer el POC de las imágenes de tal manera que incluso cuando haya AU de IRAP no alineadas en el flujo de bits, el valor de POC de la imagen actual y de las imágenes en el DPB se ajuste de tal manera que el POC de todas las imágenes en una AU sea el mismo.

**[0139]** En algunos modos de realización, en lugar de una sola bandera de restablecimiento de POC, se pueden usar dos banderas: una bandera de restablecimiento de MSB de POC (por ejemplo, poc\_msb\_reset\_flag) y una bandera de restablecimiento de LSB de POC (por ejemplo, poc\_lsb\_reset\_flag). La primera (es decir, poc\_msb\_reset\_flag) restablece los MSB del POC, y la segunda (es decir, poc\_lsb\_reset\_flag) restablece los LSB del POC. Ambas banderas se señalizan en la cabecera de fragmento.

**[0140]** Por ejemplo, si una imagen en particular tiene un valor de POC de 233, y los MSB y los LSB del POC constituyen 1 bit y 7 bits, respectivamente, los MSB serían "1" (por ejemplo, con un valor de 128) y los LSB serían "1101001" (por ejemplo, con un valor de 105). Por lo tanto, si solo se restablecen los MSB del POC (por ejemplo, en respuesta al procesamiento de poc\_msb\_reset\_flag que tiene un valor de 1), el valor de POC pasa a ser 105, y si solo se restablecen los LSB (por ejemplo, en respuesta al procesamiento de poc\_lsb\_reset\_flag que tiene un valor de 1), el valor POC pasa a ser 128. Si tanto los MSB como los LSB se restablecen (por ejemplo, en respuesta al procesamiento de poc\_msb\_reset\_flag y poc\_lsb\_reset\_flag, cada uno con un valor de 1), el valor de POC pasa a ser 0.

**[0141]** En algunos modos de realización, las dos banderas (por ejemplo, poc\_msb\_reset\_flag y poc\_lsb\_reset\_flag) pueden reemplazarse con un indicador de restablecimiento de POC de dos bits. Por ejemplo, el indicador de restablecimiento de POC de dos bits puede tener 4 valores posibles (por ejemplo, 0-3), donde 0 indica que no hay restablecimiento, 1 indica un restablecimiento de MSB, 2 indica un restablecimiento de MSB y LSB, y 3 se usa para la resistencia a errores (por ejemplo, indica algo sobre otro restablecimiento de POC, tal como un ID de restablecimiento de POC, un ID de período de restablecimiento de POC, etc.).

**Restablecimiento de valores de POC**

[0142] Con referencia a las **FIGS. 4-7**, se describirá el motivo de restablecer los valores de POC (por ejemplo, los LSB y los MSB) en las AU de IRAP no alineadas. Como se describió anteriormente, en algunos esquemas de codificación, determinadas restricciones de conformidad pueden especificar que el POC de todas las imágenes codificadas en una única AU debe ser el mismo. Sin los restablecimientos adecuados de los valores de POC, las AU de IRAP no alineadas en el flujo de bits pueden producir valores de POC que violan tales restricciones de conformidad.

[0143] La **FIG. 4** muestra un flujo de bits multicapa 400 que incluye una capa de mejora (EL) 410 y una capa base (BL) 420. La EL 410 incluye imágenes EL 412-418, y la BL incluye imágenes BL 422-428. El flujo de bits multicapa 400 incluye además unidades de acceso (AU) 430-460. La AU 430 incluye la imagen EL 412 y la imagen BL 422, la AU 440 incluye la imagen EL 414 y la imagen BL 424, la AU 450 incluye la imagen EL 416 y la imagen BL 426, y la AU 460 incluye la imagen EL 418 y la imagen BL 428. En el ejemplo de la **FIG. 4**, la imagen EL 414 es una imagen IDR, y la imagen BL 424 correspondiente en la AU 440 es una imagen final (por ejemplo, una imagen no IRAP) y, en consecuencia, la AU 440 es una AU de IRAP no alineadas. En algunos modos de realización, un restablecimiento de MSB se realiza en una imagen dada si la imagen es una imagen IDR que no está en la capa base. Dicha imagen IDR puede tener un valor de LSB de POC distinto de cero.

[0144] La **FIG. 5** muestra una tabla 500 que ilustra los valores de POC que pueden señalizarse u obtenerse en relación con el flujo de bits multicapa 400 de la **FIG. 4**. Como se muestra en la **FIG. 5**, los MSB del POC en la EL 410 se restablecen en la imagen EL 414, mientras que los MSB del POC en la BL 420 no se restablecen. Por lo tanto, si no se realiza un restablecimiento en la BL 420 en la imagen BL 424 en la AU de IRAP no alineadas 440, los valores de POC de las imágenes BL y las imágenes EL en las AU 440-460 no coincidirán (es decir, serían equivalentes) según lo especificado por las restricciones de conformidad. Las diferencias en los valores de POC con y sin restablecimiento se resaltan en negrita en la **FIG. 5**.

[0145] La **FIG. 6** muestra un flujo de bits multicapa 600 que incluye una capa de mejora (EL) 610 y una capa base (BL) 620. La EL 610 incluye imágenes EL 612-618, y la BL incluye imágenes BL 622-628. El flujo de bits multicapa 600 incluye además unidades de acceso (AU) 630-660. La AU 630 incluye la imagen EL 612 y la imagen BL 622, la AU 640 incluye la imagen EL 614 y la imagen BL 624, la AU 650 incluye la imagen EL 616 y la imagen BL 626, y la AU 660 incluye la imagen EL 618 y la imagen BL 628. En el ejemplo de la **FIG. 6**, la imagen BL 624 es una imagen IDR, y la imagen EL 614 correspondiente en la AU 640 es una imagen final (por ejemplo, una imagen no IRAP) y, en consecuencia, la AU 640 es una AU de IRAP no alineada. En algunos modos de realización, un restablecimiento de MSB y un restablecimiento de LSB se realizan para una imagen dada si la imagen es una imagen IDR que se encuentra en la capa base. Por ejemplo, el flujo de bits puede incluir una indicación de que los MSB de POC y los LSB de POC de una imagen IDR de BL de este tipo deben restablecerse. De forma alternativa, el descodificador puede realizar el restablecimiento de los MSB de POC y de los LSB de POC de una imagen IDR de BL sin ninguna indicación en el flujo de bits de que se debe realizar un restablecimiento de POC.

[0146] La **FIG. 7** muestra una tabla 700 que ilustra los valores de POC que pueden señalizarse u obtenerse en relación con el flujo de bits multicapa 600 de la **FIG. 6**. Como se muestra en la **FIG. 7**, los MSB y los LSB del POC en la BL 620 se restablecen en la imagen BL 624, mientras que ni los MSB ni los LSB del POC en la EL 610 se restablecen. Por lo tanto, si no se realiza un restablecimiento de los MSB y los LSB del POC en la EL 610 en la imagen EL 614 en la AU de IRAP no alineadas 640, los valores de POC de las imágenes BL y las imágenes EL en las AU 640-660 no coincidirán según lo especificado por las restricciones de conformidad. Las diferencias en los valores de POC con y sin restablecimiento se resaltan en negrita en la **FIG. 7**.

[0147] Los modos de realización descritos en el presente documento no están limitados a las configuraciones de flujo de bits de ejemplo ilustradas en las **FIGS. 4 y 6**, y las técnicas descritas en el presente documento pueden extenderse a cualquier flujo de bits multicapa que tenga cualquier número de capas, unidades de acceso e imágenes. Además, en los ejemplos ilustrados en las **FIGS. 4-7**, los LSB del POC se representan usando siete bits. Sin embargo, las técnicas descritas en el presente documento pueden extenderse a escenarios que tengan cualquier forma de representación de valor de POC.

**Restablecimiento de imágenes anteriores y pérdida de imágenes restablecidas**

[0148] Cuando se realiza un restablecimiento de MSB o un restablecimiento de LSB en una imagen en particular, otras imágenes en la misma capa que preceden a la imagen particular en orden de descodificación también se restablecen en base al restablecimiento realizado en la imagen particular. Por ejemplo, en el ejemplo de la **FIG. 6**, la imagen EL 614 tiene un valor de POC de 241 (por ejemplo, LSB de "1110001" + MSB de "1", que es 113+128). Cuando los restablecimientos de MSB y LSB se realizan en la imagen EL 614, el valor de POC de la imagen EL 614 pasa a ser 0, y la imagen EL 612 en la EL 610 que precede a la imagen EL 614 en orden de descodificación también se restablece en base al valor de POC original de 241 de la imagen EL 614. Por ejemplo, el nuevo valor de POC de la imagen EL 612 se calcula restando el valor de POC preestablecido de la imagen EL 614 (que es un valor de 241) al valor de POC preestablecido de la imagen EL 612, que es 240 (por ejemplo, LSB de "1110000" + MSB de "1", que

es 112+128). Por lo tanto, después del restablecimiento, el valor de POC de la imagen EL 612 pasa a ser -1, de acuerdo con el hecho de que la imagen EL 612 debe proporcionarse antes que la imagen EL 614, donde un valor de POC menor denota una posición anterior en orden de salida. Como se muestra en la **FIG. 7**, los valores de LSB señalizados para las AU posteriores 650 y 660 se ajustan en consecuencia (por ejemplo, a un valor de 1 y un valor de 2, respectivamente), suponiendo que el restablecimiento se realiza en la imagen EL 614.

**[0149]** Sin embargo, incluso si un restablecimiento de POC apropiado de los MSB y/o los LSB descritos anteriormente se señala en el flujo de bits (por ejemplo, en la cabecera de fragmento) de modo que el descodificador pueda procesar la señal y realizar el restablecimiento de POC en consecuencia, si la señalización de imagen de tal restablecimiento de POC se pierde durante la transmisión del flujo de bits o se elimina del flujo de bits para satisfacer las restricciones de ancho de banda, el restablecimiento de POC que se pretende realizar en la imagen en particular puede no realizarse correctamente.

**[0150]** Por ejemplo, en el ejemplo de la **FIG. 6**, si la imagen EL 614 no está disponible para el descodificador, el descodificador no sabría (es decir, no determinaría) que hay que restablecer los MSB y los LSB del POC en la EL 610 en la AU 640. En consecuencia, los valores de POC de cualquier imagen que precede a la imagen EL 614 no disponible en orden de decodificación seguirían teniendo sus valores de POC preestablecidos originales, ya que el restablecimiento en la imagen EL 614 nunca tuvo lugar (es decir, no se realizó la operación de restablecimiento). Por otro lado, los valores de POC de las imágenes que siguen a la imagen EL 614 no disponible en orden de decodificación se habrían determinado o señalado como si realmente se hubiera realizado un restablecimiento (es decir, se realizó la operación de restablecimiento). Así, en el ejemplo de la **FIG. 7**, las imágenes EL 612, 616 y 618 tendrían valores de POC de 240, 1 y 2, respectivamente, lo que sería incorrecto dado que la imagen EL 612 precede a las imágenes EL 616 y 618 en orden de salida. Por lo tanto, se desea un procedimiento de codificación que dé como resultado valores de POC correctos, incluso cuando la imagen que señala el restablecimiento de POC no esté disponible.

#### **ID de período de restablecimiento de POC**

**[0151]** En algunos modos de realización se usa un ID de período de restablecimiento de POC para identificar cada período de restablecimiento de POC (por ejemplo, un período que comienza con un restablecimiento de POC y termina inmediatamente antes del siguiente restablecimiento de POC). Una restricción de conformidad del flujo de bits puede exigir que los ID de período de restablecimiento de POC durante dos períodos de restablecimiento de POC temporalmente consecutivos (por ejemplo, en orden de decodificación) sean diferentes. Cada imagen en un período de restablecimiento de POC puede asociarse al ID de período de restablecimiento de POC del período de restablecimiento de POC. Usando los ID de período de restablecimiento de POC, el codificador puede asegurar que los valores de POC estén alineados en múltiples capas incluso si se pierden una o más imágenes en la AU de restablecimiento de POC.

#### **Uso de valores de POC como información de anclaje**

**[0152]** En algunos modos de realización, los valores de POC asignados a una imagen descodificada pueden utilizarse como información de anclaje. Por ejemplo, el valor de POC asignado a una imagen particular puede usarse para calcular al menos una parte del valor de POC (por ejemplo, MSB, LSB o ambos) de otra imagen que sigue a la imagen particular en orden de decodificación. En tal caso, la parte del valor de POC calculada usando información previamente señalizada no necesita ser señalizada explícitamente, dando como resultado ahorros de bits.

**[0153]** Sin embargo, el valor de POC que se asigna a una imagen descodificada puede no ser estático y puede actualizarse cuando se invoca un proceso de restablecimiento de POC. Esta naturaleza mutable de los valores de POC puede hacer que los valores de POC sean menos útiles para su uso como información de anclaje para otros procesadores u otras imágenes en el flujo de bits. Por tanto, si un flujo de bits invoca procesos, tal como un restablecimiento de POC, que pueden alterar los valores de POC asociados a las imágenes descodificadas, se desea un esquema de codificación mejorado que tenga en cuenta dichos procesos cuando se usen los valores de POC.

#### **Provisión de imágenes del DPB en el restablecimiento de POC**

**[0154]** En algunas implementaciones, una restricción de flujo de bits puede exigir que para cualquier período de restablecimiento de POC, todas las imágenes que preceden al período de restablecimiento de POC deben proporcionarse antes de la imagen de restablecimiento de POC (por ejemplo, una imagen que tiene una indicación para restablecer el POC asociado a la misma) asociada al período de restablecimiento de POC o cualquier imagen que siga a la imagen de restablecimiento de POC en orden de decodificación. Para satisfacer dicha restricción de flujo de bits, cuando se invoca un restablecimiento de POC (por ejemplo, indicado por un indicador de restablecimiento de POC asociado a la imagen de restablecimiento de POC), el codificador puede proporcionar todas las imágenes del DPB que se descodificaron antes de la imagen de restablecimiento de POC.



[0155] Sin embargo, la provisión de todas las imágenes anteriores en el DPB cada vez que se invoca un restablecimiento de POC (por ejemplo, cuando el codificador procesa una imagen de restablecimiento de POC) daría como resultado un orden de salida incorrecto de imágenes en descodificadores que siguen el orden de salida, ya que no hay garantía de que las imágenes que siguen a la imagen de restablecimiento de POC en orden de descodificación sigan necesariamente, en orden de salida, a todas las imágenes que preceden a la imagen de restablecimiento de POC en orden de descodificación. Por ejemplo, una imagen que sigue a la imagen de restablecimiento de POC en orden de descodificación puede preceder a una de las imágenes que se descodifican antes de la imagen de restablecimiento de POC en orden de salida.

[0156] Por ejemplo, si la imagen A es una imagen de restablecimiento de POC, la imagen B es una imagen que precede a la imagen A en orden de descodificación y la imagen C es una imagen que sigue a la imagen A en orden de descodificación, el orden de descodificación adecuado sería la imagen B, la imagen A e Imagen C. Si la Imagen C se va a proporcionar antes que la Imagen A, que se va a proporcionar antes que la Imagen B, el orden de salida apropiado sería Imagen C, Imagen A e Imagen B. Si todas las imágenes del DPB van a proporcionarse de acuerdo con el modo de realización anterior, el codificador descodificaría y almacenaría la Imagen B en el DPB, descodificaría la Imagen A y proporcionaría la Imagen B del DPB al darse cuenta de que la Imagen A es una imagen de restablecimiento de POC. Sin embargo, la provisión de la Imagen B antes que la Imagen C sería incorrecta de acuerdo con el orden de salida apropiado descrito anteriormente.

## **Mensajes SEI y de restablecimiento de POC**

[0157] En algunos esquemas de codificación existentes, la semántica de varios mensajes SEI no sería compatible con el proceso de restablecimiento de POC propuesto para la codificación de vídeo multicapa y daría lugar a ambigüedad en la aplicación del mensaje SEI al flujo de bits. Por ejemplo, en la versión 1 de la especificación HEVC, muchos de los mensajes SEI se definieron de tal manera que dependían del POC. Cuando la implementación se extiende a casos multicapa, se introdujo el concepto de restablecimiento del POC. Como resultado, los mensajes SEI existentes se volvieron incorrectos o ambiguos en muchos casos, especialmente en la manera en que los mensajes SEI definían el POC. Por lo tanto, se desea un esquema de codificación mejorado que incluya semántica actualizada para los mensajes SEI que sean compatibles con el concepto de restablecimientos de POC.

## **Información de POC en mensajes de retroalimentación**

[0158] En algunas implementaciones, el uso de la información POC para identificar una imagen en mensajes de retroalimentación no está claro. Por ejemplo, muchos sistemas usan mensajes de retroalimentación que identifican imágenes. Por ejemplo, en una sesión entre una fuente y un receptor, si se pierden algunas imágenes, los POC de las imágenes se usan para identificar las imágenes que se han perdido. Esta información de POC puede transmitirse en mensajes de retroalimentación. Con la posibilidad de restablecimientos de POC, la información de POC incluida en los mensajes de retroalimentación puede necesitar incluir, además del POC de la imagen que se está identificando, información acerca del período de restablecimiento de POC al que pertenece la imagen. Por ejemplo, múltiples imágenes pueden tener el mismo POC, y es posible que el sistema no pueda identificar correctamente esas imágenes a menos que el sistema reciba la información acerca de los períodos de restablecimiento de POC a los que pertenecen las imágenes. Por lo tanto, se desea un sistema mejorado que indique el período de restablecimiento de POC de la imagen que se está identificando en un mensaje de retroalimentación.

## **Señalización de ciclo de MSB de POC**

[0159] En algunas implementaciones, los LSB de POC se señalizan y los MSB de POC se calculan usando el tipo de unidad NAL de la imagen actual y los LSB y/o MSB de las imágenes que preceden a la imagen actual en orden de descodificación. En otras implementaciones, la información de MSB de POC se señala en la ampliación de cabecera de segmento de imágenes CRA y BLA para mejorar la resistencia a errores de la derivación de POC. Sin embargo, el uso de los MSB de POC señalizados para calcular los valores de POC solo para imágenes CRA y BLA (con NoRaslOutputFlag igual a un valor de 1) daría como resultado valores de POC que no están alineados a través de múltiples capas en algunos casos. Por tanto, se desea un esquema de codificación mejorado que mejore la alineación entre capas de valores de POC.

## **Ejemplos e implementaciones**

[0160] A continuación se describirán diversos procedimientos que se pueden usar para abordar determinados problemas descritos anteriormente. Algunos de estos procedimientos se pueden aplicar de forma independiente y algunos de ellos se pueden aplicar de forma combinada. Además, la sintaxis y la semántica de ejemplo que se pueden usar para implementar uno o más de los procedimientos descritos en el presente documento también se proporcionan a continuación. Cuando determinadas partes de la especificación de HEVC se reproducen para ilustrar las adiciones y eliminaciones que se pueden incorporar para implementar uno o más de los procedimientos descritos en el presente documento, dichas modificaciones se muestran en  *cursiva*.

## **Señalización de valores para la obtención de POC**

**[0161]** En algunos modos de realización de la presente divulgación, un mensaje SEI que contiene información para la obtención del POC correcto se señala para una o más imágenes que siguen a la imagen para la cual se restablecerá los MSB de POC y/o los LSB de POC. Por ejemplo, el mensaje SEI puede asociarse a una imagen, picA, que sigue a otra imagen, picB, para la cual se restablecerá los MSB de POC, los LSB de POC o ambas cosas. Por lo tanto, incluso cuando picB se ha perdido por completo, el mensaje SEI asociado a picA se puede usar para obtener los valores de POC correctos para otras imágenes en la misma capa.

**[0162]** En algunos modos de realización de la presente divulgación, la información para la obtención del POC correcto se señala en la cabecera de fragmento de una o más imágenes que siguen a la imagen para la cual se restablecerá los MSB de POC y/o los LSB de POC. Por ejemplo, la información puede incluirse en la cabecera de fragmento de una imagen picA que sigue a otra imagen picB para la cual se restablecerán los MSB de POC, los LSB de POC o ambas cosas. Por lo tanto, incluso cuando PicB se ha perdido por completo, la información incluida en la cabecera de fragmento de picA se puede usar para obtener los valores de POC correctos para otras imágenes en la misma capa.

**[0163]** En algunos modos de realización de la presente divulgación, la información (que puede denominarse en el presente documento información de obtención de POC) señalizada en la cabecera de fragmento de la imagen y/o señalizada como un mensaje SEI asociado a la imagen puede incluir: un tipo de restablecimiento de POC que indica si el valor de POC de la imagen de restablecimiento de POC precedente (por ejemplo, una imagen en la que se realizará un restablecimiento de POC) en la misma capa se restablecerá mediante el restablecimiento de los bits más significativos (MSB) y los bits menos significativos (LSB) del valor de POC o mediante el restablecimiento solo de los MSB del valor de POC; un valor de restablecimiento de POC que indica el valor de POC de la imagen que se ha perdido o eliminado que también precede a la imagen a la que está asociada la información de obtención de POC; y un ID de restablecimiento de POC que identifica el restablecimiento de POC para el cual se proporciona la información de obtención de POC. Por ejemplo, el decodificador puede omitir un restablecimiento de POC señalado en conexión con una imagen particular si el restablecimiento de POC señalado tiene un valor de ID de restablecimiento de POC de 1 y ya se ha realizado otro restablecimiento de POC con un ID de restablecimiento de POC de 1.

#### **Restricción del orden de salida con respecto a una imagen de restablecimiento de POC**

**[0164]** En algunos modos de realización de la presente divulgación, se añade una restricción de flujo de bits para que una imagen que sigue, en orden de decodificación, a una imagen de restablecimiento de POC no preceda, en orden de salida, a otra imagen que precede, en orden de decodificación, a la imagen de restablecimiento de POC. En dichos modos de realización, el codificador puede determinar que una restricción de flujo de bits de este tipo sea aplicable y se adhiera a la restricción de flujo de bits de modo que el flujo de bits codificado se ajuste a la restricción de flujo de bits.

**[0165]** Por ejemplo, antes de que se señale un restablecimiento de POC en relación con una imagen actual, el codificador puede determinar si cualquier imagen que sigue a la imagen actual en orden de decodificación debe proporcionarse antes que cualquier otra imagen que preceda a la imagen actual en orden de decodificación. Si el codificador determina que cualquier imagen que sigue a la imagen actual en orden de decodificación va a proporcionarse antes que cualquier otra imagen que preceda a la imagen actual en orden de decodificación, el codificador puede abstenerse de señalar el restablecimiento de POC en asociación con la imagen actual para garantizar que el flujo de bits codificado se ajuste a la restricción de flujo de bits.

**[0166]** De forma alternativa, el codificador puede, después de determinar que se va a señalar un restablecimiento de POC en relación con una imagen actual, garantizar que ninguna imagen que siga a la imagen actual en orden de decodificación se proporcione antes que cualquier imagen que preceda a la imagen actual en orden de decodificación. Por ejemplo, el codificador puede abstenerse de codificar cualquier imagen (o hacer que se proporcione una imagen) en un flujo de bits después de la imagen actual si la imagen se va a proporcionar antes que cualquier imagen que preceda a la imagen actual en orden de decodificación, para garantizar que el flujo de bits codificado se ajuste a la restricción de flujo de bits.

#### **Ciclo de MSB de POC**

**[0167]** En algunos modos de realización de la presente divulgación, el codificador puede señalar un valor de desplazamiento que puede denominarse ciclo de MSB de POC. En un ejemplo, el ciclo de MSB de POC puede indicar la diferencia entre el MSB de POC de una imagen codificada previamente y el MSB de POC de la imagen actual. En un ejemplo de este tipo, el codificador puede usar el ciclo de MSB de POC para actualizar los valores de POC de imágenes que preceden a la imagen a la que está asociado el ciclo de MSB de POC. En otro ejemplo, el ciclo de MSB de POC puede ser idéntico al MSB de POC de la imagen a la que está asociado el ciclo de MSB de POC. Por tanto, cuando el ciclo de MSB de POC se señala en asociación con una imagen particular, el MSB de POC de la imagen particular se puede calcular en base al ciclo de MSB de POC señalado.

**Señalización del ciclo de MSB de POC con imágenes CRA y BLA**

[0168] En algunos modos de realización de la presente divulgación, en lo que respecta a imágenes CRA y BLA, el valor del ciclo de MSB de POC señalado se determina en relación con la imagen de restablecimiento de POC anterior en la misma capa o la imagen IDR previa en la misma capa, la que esté más cerca. Si ninguna está presente (por ejemplo, lo que significa que la imagen CRA/BLA es la primera imagen CRA/BLA en la capa), el ciclo de MSB de POC de la imagen CRA/BLA puede ser cualquier valor en el intervalo permitido de los valores de ciclo de MSB de POC.

**Ciclo de MSB de POC en imágenes de restablecimiento de POC**

[0169] En algunos modos de realización de la presente divulgación, cuando una imagen CRA o BLA en una capa particular es también una imagen de restablecimiento de POC, entonces el valor señalado del ciclo de MSB de POC se usa para actualizar los valores de POC de todas las imágenes en el DPB en la capa particular. Si la imagen CRA o BLA también es una imagen de restablecimiento de POC, el MSB de la imagen CRA o BLA puede ser cero. Cuando la imagen CRA o BLA en la capa particular no es una imagen de restablecimiento de POC, el valor señalado del ciclo de MSB de POC se usa para calcular el POC de la imagen CRA o BLA.

[0170] En algunos modos de realización de la presente divulgación, el ciclo de MSB de POC puede señalizarse para imágenes IDR. Cuando una imagen IDR en una capa particular es también una imagen de restablecimiento de POC, entonces el valor señalado del ciclo de MSB de POC se usa para actualizar los valores de POC de todas las imágenes en el DPB en la capa particular. Cuando la imagen IDR en la capa particular no es una imagen de restablecimiento de POC, el valor señalado del ciclo de MSB de POC se usa para calcular el POC de la imagen IDR.

[0171] La FIG. 8 es un diagrama de flujo que ilustra un procedimiento 800 para codificar información de vídeo, de acuerdo con un modo de realización de la presente divulgación. Las etapas ilustradas en la FIG. 8 se pueden realizar mediante un codificador (por ejemplo, el codificador de vídeo mostrado en la FIG. 2A o la FIG. 2B), un descodificador (por ejemplo, el descodificador de vídeo como se muestra en la FIG. 3A o la FIG. 3B) o cualquier otro componente. Por conveniencia, el procedimiento 800 se describe como realizado por un codificador, que puede ser el codificador, el descodificador u otro componente.

[0172] El procedimiento 800 comienza en el bloque 801. En el bloque 805, el codificador determina si una imagen actual en una capa de vídeo es una imagen de restablecimiento de POC. Si el codificador determina que la imagen actual no es una imagen de restablecimiento de POC, el procedimiento 800 avanza hasta el bloque 810. Por otro lado, si el codificador determina que la imagen actual es una imagen de restablecimiento de POC, el procedimiento 800 avanza hasta el bloque 815.

[0173] En el bloque 810, el codificador determina el POC de la imagen actual basándose en un desplazamiento asociado a la imagen actual. El desplazamiento puede indicar la diferencia entre los MSB de POC de la imagen actual y los MSB de POC de otra imagen previamente descodificada en la misma capa. En el bloque 815, el codificador actualiza el POC de todas las imágenes del DPB que están en la misma capa que la imagen actual basándose en el desplazamiento asociado a la imagen actual. Por ejemplo, el codificador puede actualizar el POC de todas las imágenes del DPB restando un valor al POC de cada imagen del DPB. En un ejemplo, el valor restado se puede determinar en base al desplazamiento asociado a la imagen actual. En otro ejemplo, el valor restado puede ser lo que hubiera valido el POC de la imagen actual sin el restablecimiento de POC. El procedimiento 800 termina en 820.

[0174] Como se analizó anteriormente, uno o más componentes del codificador de vídeo 20 de la FIG. 2A, del codificador de vídeo 23 de la FIG. 2B, del descodificador de vídeo 30 de la FIG. 3A o del descodificador de vídeo 33 de la FIG. 3B (por ejemplo, la unidad de predicción entre capas 128 y/o la unidad de predicción entre capas 166) se pueden usar para implementar cualquiera de las técnicas analizadas en la presente divulgación, tal como determinar si la imagen actual es una imagen de restablecimiento de POC, determinar el POC de la imagen actual en base al desplazamiento asociado a la imagen actual y actualizar el POC de todas las imágenes del DPB en base al desfase asociado a la imagen actual.

[0175] En el procedimiento 800, uno o más de los bloques mostrados en la FIG. 8 se pueden eliminar (por ejemplo, no realizar), modificar, y/o el orden en el que se realiza el procedimiento se puede cambiar. Por ejemplo, aunque el bloque 810 se muestra en la FIG. 8, el bloque 810 puede eliminarse, y el procedimiento 800 puede finalizar sin realizar ninguna operación adicional si el codificador determina que la imagen actual no es un restablecimiento de POC. De forma alternativa, el bloque 815 puede eliminarse, y el procedimiento 800 puede finalizar sin realizar ninguna operación adicional si el codificador determina que la imagen es una imagen de restablecimiento de POC. Por tanto, los modos de realización de la presente divulgación no están limitados a, o por, el ejemplo mostrado en la FIG. 8, y otras variaciones pueden implementarse.

**Señalización de presencia de ciclo de MSB de POC**

[0176] En algunos modos de realización de la presente divulgación, se puede señalar una bandera o elemento de sintaxis en el VPS que especifica si el ciclo de MSB de POC está señalado (por ejemplo, en la cabecera de fragmento) para las imágenes CRA y BLA. La señalización del ciclo de MSB de POC puede basarse en la bandera o elemento de sintaxis incluido en el VPS. La determinación de si señalar el ciclo de MSB de POC se puede realizar en base a la bandera o elemento de sintaxis incluido en el VPS.

#### **Ciclo de MSB de POC en imágenes IRAP**

[0177] En algunas implementaciones, el codificador puede realizar un restablecimiento de POC para alinear los POC en una AU no alineada (por ejemplo, una imagen IRAP y una imagen no IRAP en la misma AU). En tal caso, poc\_reset\_idc puede establecerse en un valor mayor que un valor de 0, para indicar que se va a realizar un restablecimiento de POC. Sin embargo, es posible que no se necesite tal restablecimiento de POC si el ciclo de MSB de POC se señala junto con una imagen CRA o BLA, como se describe anteriormente.

[0178] En algunos modos de realización de la presente divulgación, cuando el ciclo de MSB de POC se señala para una imagen IRAP que tiene un valor nal\_unit\_type particular (por ejemplo, IDR, CRA, BLA, etc.), y una AU contiene una imagen que no es IRAP con nuh\_layer\_id igual a un valor de 0 y al menos un IRAP con el mismo valor de nal\_unit\_type que la imagen IRAP, el poc\_reset\_idc asociado a la AU (por ejemplo, asociado a la imagen que no es IRAP o la imagen IRAP) puede establecerse igual a un valor de 0 o establecerse para que sea mayor que un valor de 0, eliminando así la restricción de flujo de bits descrita anteriormente.

#### **POC estático a nivel de capa**

[0179] En algunos modos de realización de la presente divulgación, se ha invocado un POC adicional que especifica el valor de POC de una imagen descodificada antes de que se haya invocado cualquier proceso de restablecimiento de POC. Este POC puede denominarse POC a nivel de capa. El valor de este POC a nivel de capa puede calcularse a partir de la información señalizada en la imagen, la cabecera de segmento u otras imágenes de la misma capa sin usar información adicional de imágenes de otras capas. El POC a nivel de capa podría usarse para cualquier proceso que pueda desear un valor de POC estático que no se vea afectado por otros procedimientos (por ejemplo, un restablecimiento de POC) para la identificación de imágenes. Por ejemplo, dichos procesos pueden usar el POC a nivel de capa para la identificación de imágenes en la descodificación de conjuntos de imágenes de referencia, para la identificación de imágenes en la descodificación de mensajes SEI y para cualquier posprocesamiento en la descodificación de imágenes. Cuando se usa un POC a nivel de capa, no es necesario disminuir los valores de POC de las imágenes anteriores del DPB cuando se realiza un restablecimiento de POC.

[0180] En algunos modos de realización de la presente divulgación, para implementar el POC a nivel de capa, los bits menos significativos (LSB) del POC a nivel de capa se señalizan en la imagen de restablecimiento de POC o siempre que el valor de poc\_reset\_idc no sea igual a un valor de 0. Esta información de LSB se puede usar para obtener tanto el POC como el POC a nivel de capa. Los bits más significativos del POC pueden señalizarse en la ampliación de la cabecera de fragmento de la imagen IRAP. Esta información de MSB se puede usar para obtener tanto el POC como el POC de capa.

[0181] En algunos modos de realización de la presente divulgación, para cualquier proceso que pueda usar información relacionada con POC, se usa el POC (por ejemplo, para mantener la compatibilidad con versiones anteriores de la versión de una sola capa de HEVC) para la capa base, y el POC a nivel de capa se usa para otras capas.

#### **ID de período de restablecimiento de POC en mensajes de retroalimentación**

[0182] En algunos modos de realización de la presente divulgación, durante el funcionamiento en el contexto de un perfil SHVC o MV-HEVC, además del valor de POC y el ID de capa, el ID de período de restablecimiento de POC de la última imagen descodificada también se señala en un mensaje de retroalimentación para identificar la última imagen descodificada. A continuación, el codificador puede identificar de forma única la imagen codificada previamente. Por ejemplo, una imagen descodificada actual puede tener una imagen de referencia en su conjunto de imágenes de referencia (RPS), pero el descodificador puede no tener acceso a la imagen de referencia (por ejemplo, debido a una pérdida durante la transmisión). En este ejemplo, el descodificador puede enviar un mensaje de retroalimentación al codificador, solicitando que el codificador reenvíe la imagen de referencia. El mensaje de retroalimentación puede incluir un valor de POC asociado a la imagen de referencia y un período de restablecimiento de POC asociado a la imagen de referencia, y el codificador puede usar el valor de POC y el período de restablecimiento de POC para identificar la imagen de referencia. Por ejemplo, al recibir un mensaje de retroalimentación con un valor de POC y un ID de período de restablecimiento de POC, si la última imagen descodificada correspondiente al valor de POC señalado se encuentra en un período de restablecimiento de POC diferente al de la última imagen codificada, el ID de período de restablecimiento de POC señalado se usaría para rastrear el período correcto de restablecimiento de POC, añadiendo un valor delta de POC para cada período de restablecimiento de POC presente entre los dos períodos de restablecimiento de POC.

**Implementación de ejemplo n.º 1**

[0183] A continuación se proporcionan implementaciones de ejemplo de uno o más de los modos de realización descritos anteriormente. Los cambios en la sintaxis, la semántica y el proceso de decodificación que se indican a continuación en *cursiva* son con respecto a los proporcionados en la especificación MV-HEVC WD6. Estos cambios implementan diversos modos de realización descritos en la presente solicitud. Los números de referencia que se utilizan a continuación (por ejemplo, subcláusula 5.8, sección 8.1, etc.) se refieren a las secciones de la especificación MV-HEVC WD6.

**Adición de una relación matemática**

[0184] En un modo de realización, la siguiente relación se incorpora en la subcláusula 5.8:

$$\text{GetCurrMsb}(cl, pl, pm, ml) = \begin{cases} pm + ml & ; \quad pl - cl \geq ml/2 \\ pm - ml & ; \quad cl - pl > ml/2 \\ pm & ; \quad \text{en otros casos} \end{cases}$$

**Cambios en el proceso de decodificación de fragmentos**

[0185] En un modo de realización, se realizan los siguientes cambios en la Sección 8.1 "Proceso de decodificación general".

- Cuando NoClasOutputFlag es igual a 1, la variable LayerInitializedFlag[i] se establece igual a 0 para todos los valores de i desde 0 hasta vps\_max\_layer\_id, ambos inclusive, y la variable FirstPicInLayerDecodedFlag [i] se establece igual a 0 para todos los valores de i desde 0 hasta vps\_max\_layer\_id, ambos inclusive.

**Cambios en 8.3.1. Proceso de decodificación para el recuento de orden de imágenes****8.3.1 Proceso de decodificación para el recuento de orden de imágenes**

[0186] La salida de este proceso es PicOrderCntVal, el recuento de orden de imágenes de la imagen actual.

[0187] Los recuentos de orden de imágenes se usan para identificar imágenes, para obtener parámetros de movimiento en modo de combinación y predicción de vectores de movimiento, y para la verificación de conformidad de decodificador (véase la subcláusula 11).

[0188] Cada imagen codificada está asociada a una variable de recuento de orden de imágenes, denominada PicOrderCntVal.

[0189] Cuando la imagen actual no es una imagen IRAP con NoRasOutputFlag igual a 1, las variables prevPicOrderCntLsb y prevPicOrderCntMsb se obtienen de la siguiente manera:

- Supóngase que prevTidOPic es la imagen anterior en orden de decodificación que tiene TemporalId igual a 0 y no es una imagen RASL, una imagen RADL o una imagen de subcapa que no es de referencia, y que *PrevPicOrderCnt[nuh\_layer\_id]* es *PicOrderCntVal* de *prevTidOPic*.

- La variable prevPicOrderCntLsb se establece igual a *PrevPicOrderCnt[nuh\_layer\_id] & (MaxPicOrderCntLsb - 1)*.

- La variable prevPicOrderCntMsb se establece igual a *PrevPicOrderCnt[nuh\_layer\_id] - prevPicOrderCntLsb*.

[0190] La variable PicOrderCntMsb de la imagen actual se obtiene de la siguiente manera:

- Si la imagen actual es una imagen IRAP con NoRasOutputFlag igual a 1, PicOrderCntMsb se establece igual a 0.

- De lo contrario, PicOrderCntMsb se obtiene de la siguiente manera:

```
if((slice_pic_order_cnt_lsb < prevPicOrderCntLsb) &&
    ((prevPicOrderCntLsb - slice_pic_order_cnt_lsb) >= (MaxPicOrderCntLsb /
2))
)
```

```

PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb
else if((slice_pic_order_cnt_lsb > prevPicOrderCntLsb) &&
      ((slice_pic_order_cnt_lsb - prevPicOrderCntLsb) > (MaxPicOrderCntLsb / 2)
      ))
5   PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb
    else
      PicOrderCntMsb = prevPicOrderCntMsb

```

**[0191]** PicOrderCntVal se obtiene de la siguiente manera:

PicOrderCntVal = PicOrderCntMsb + slice\_pic\_order\_cnt\_lsb

NOTA 1- Todas las imágenes IDR tendrán PicOrderCntVal igual a 0, ya que se infiere que slice\_pic\_order\_cnt\_lsb es 0 para las imágenes IDR y prevPicOrderCntLsb y prevPicOrderCntMsb se han establecido ambos a 0.

**[0192]** El valor de PicOrderCntVal estará en el intervalo de  $-2^{31}$  a  $2^{31}-1$ , ambos inclusive. En una CVS, los valores de PicOrderCntVal para dos imágenes codificadas cualesquiera no serán iguales.

**[0193]** La función PicOrderCnt(picX) se especifica de la siguiente manera:

PicOrderCnt(picX) = PicOrderCntVal de la imagen picX

**[0194]** La función DiffPicOrderCnt(picA, picB) se especifica de la siguiente manera:

DiffPicOrderCnt(picA, picB) = PicOrderCnt(picA) - PicOrderCnt(picB)

**[0195]** El flujo de bits no debe contener datos que den como resultado valores de DiffPicOrderCnt(picA, picB) usados en el proceso de descodificación que no estén en el intervalo de  $-2^{15}$  a  $2^{15}-1$ , ambos inclusive.

NOTA 2- Sea X la imagen actual e Y y Z otras dos imágenes en la misma CVS, se considera que Y y Z están en la misma dirección de orden de salida desde X cuando tanto DiffPicOrderCnt(X, Y) como DiffPicOrderCnt(X, Z) son positivos o ambos son negativos.

#### **Modificación de la semántica de los mensajes SEI**

**[0196]** En un modo de realización, la subcláusula D.3.4 se modifica como sigue:

**pan\_scan\_rect\_persistence\_flag** especifica la persistencia del mensaje SEI de rectángulo de barrido panorámico.

pan\_scan\_rect\_persistence\_flag igual a 0 especifica que la información de rectángulo de barrido panorámico se aplica únicamente a la imagen descodificada actual.

**[0197]** Supongamos que picA es la imagen actual. pan\_scan\_rect\_persistence\_flag igual a 1 especifica que la información de rectángulo de barrido panorámico persiste en orden de salida hasta que cualquiera de las siguientes condiciones sea verdadera:

- Comienza una nueva CVS.

- El flujo de bits termina.

- Se proporciona una imagen *picB* en una unidad de acceso que contiene un mensaje SEI de rectángulo de barrido panorámico con el mismo valor de pan\_scan\_rect\_id para la que *PicOrderCnt(picB)* es mayor que *PicOrderCnt(picA)*, donde *PicOrderCnt(picB)* y *PicOrderCnt(picA)* son los valores *PicOrderCntVal* de *picB* y *picA*, respectivamente, justo después de la invocación del proceso de descodificación para el recuento de orden de imágenes para *picB*.

**[0198]** En un modo de realización, la subcláusula D.3.8 se modifica como sigue:

Quando se realiza un acceso aleatorio para iniciar la descodificación desde la unidad de acceso asociada al mensaje SEI de punto de recuperación, el descodificador funciona como si la imagen asociada fuera la primera imagen del flujo de bits en orden de descodificación, y la variable *PrevPicOrderCnt [nuh\_layer\_id]* usada en la obtención de PicOrderCntVal se establece igual a 0.

NOTA 2 - Cuando la información de HRD está presente en el flujo de bits, un mensaje SEI de período de almacenamiento en búfer debe asociarse a la unidad de acceso asociada al mensaje SEI de punto de recuperación para establecer la inicialización del modelo de búfer de HRD después de un acceso aleatorio.

**[0199]** Cualquier RBSP de SPS o PPS al que se haga referencia mediante una imagen asociada a un mensaje SEI de punto de recuperación o mediante cualquier imagen que siga a dicha imagen en orden de decodificación estará disponible para el proceso de decodificación antes de su activación, independientemente de si el proceso de decodificación se inicia, o no, al comienzo del flujo de bits o con la unidad de acceso, en orden de decodificación, que está asociada al mensaje SEI de punto de recuperación.

**[0200] recovery\_poc\_cnt** especifica el punto de recuperación de imágenes decodificadas en orden de salida. Si hay una imagen *picB* que sigue a la imagen actual (es decir, la imagen asociada al mensaje SEI actual) *picA* en orden de decodificación en la CVS y  $PicOrderCnt(picB)$  es igual a  $PicOrderCnt(picA)$  más el valor de *recovery\_poc\_cnt*, donde  $PicOrderCnt(picA)$  y  $PicOrderCnt(picB)$  son los valores  $PicOrderCntVal$  de *picA* y *picB*, respectivamente, justo después de la invocación del proceso de decodificación para el recuento de orden de imágenes para *picB*, la imagen *picB* se denomina imagen de punto de recuperación. De lo contrario, la primera imagen *picC* en orden de salida para la cual  $PicOrderCnt(picC)$  es mayor que  $PicOrderCnt(picA)$  más el valor de *recovery\_poc\_cnt* se denomina imagen de punto de recuperación, donde  $PicOrderCnt(picA)$  y  $PicOrderCnt(picC)$  son los valores de  $PicOrderCntVal$  de *picA* y *picC*, respectivamente, justo después de la invocación del proceso de decodificación para el recuento de orden de imágenes para *picC*. La imagen de punto de recuperación no debe preceder a la imagen actual en orden de decodificación. Se indica que todas las imágenes decodificadas en orden de salida son correctas o aproximadamente correctas en contenido comenzando en la posición de orden de salida de la imagen de punto de recuperación. El valor de *recovery\_poc\_cnt* estará dentro del intervalo de  $-MaxPicOrderCntLsb/2$  a  $MaxPicOrderCntLsb/2-1$ , ambos inclusive.

**[0201]** En un modo de realización, la subcláusula D.3.11 se modifica como sigue:

El mensaje SEI de inicio de segmento de refinamiento progresivo especifica el comienzo de un conjunto de imágenes codificadas consecutivas en orden de decodificación que consiste en la imagen actual y una secuencia de una o más imágenes subsiguientes que refinan la calidad de la imagen actual, en lugar de una representación de una escena en movimiento continuo.

**[0202]** *Supóngase que picA es la imagen actual.* El conjunto etiquetado de imágenes codificadas consecutivas continúa hasta que una de las siguientes condiciones sea verdadera:

- Comienza una nueva CVS.

- El flujo de bits termina.

- *pic\_order\_cnt\_delta* es mayor que 0 y  $PicOrderCntVal$  del siguiente fragmento, que pertenece a la imagen *picB*, que se va a decodificar, es decir,  $PicOrderCnt(picB)$ , es mayor que  $PicOrderCnt(picA)$  más *pic\_order\_cnt\_delta*, donde  $PicOrderCnt(picB)$  y  $PicOrderCnt(picA)$  son los valores  $PicOrderCntVal$  de *picB* y *picA*, respectivamente, justo después de la invocación del proceso de decodificación para el recuento de orden de imágenes para *picB*.

- Se decodifica un mensaje SEI de fin de segmento de refinamiento progresivo con el mismo *progressive\_refinement\_id* que el de este mensaje SEI.

**[0203]** El orden de decodificación de las imágenes dentro del conjunto etiquetado de imágenes consecutivas debe ser el mismo que su orden de salida. **progressive\_refinement\_id** especifica un número de identificación para la operación de refinamiento progresivo. *progressive\_refinement\_id* debe estar en el intervalo de 0 a  $2^{32}-2$ , ambos inclusive.

Los valores de *progressive\_refinement\_id* en el intervalo de 0 a 255, ambos inclusive, y en el intervalo de 512 a  $2^{31}-1$ , ambos inclusive, pueden usarse según lo determine la aplicación. Los valores de *progressive\_refinement\_id* en el intervalo de 256 a 511, ambos inclusive, y en el intervalo de  $2^{31}$  a  $2^{32}-2$ , ambos inclusive, están reservados para uso futuro por ITU-T | ISO/IEC. Los decodificadores que encuentren un valor de *progressive\_refinement\_id* en el intervalo de 256 a 511, ambos inclusive, o en el intervalo de  $2^{31}$  a  $2^{32}-2$ , ambos inclusive, lo ignorarán.

**[0204] pic\_order\_cnt\_delta** especifica la última imagen del conjunto etiquetado de imágenes codificadas consecutivas en orden de decodificación como sigue:

- Si *pic\_order\_cnt\_delta* es igual a 0, la última imagen del conjunto etiquetado de imágenes codificadas consecutivas en orden de decodificación es la imagen siguiente:

- Si la CVS contiene una o más imágenes que siguen a la imagen actual en orden de decodificación y están asociadas con un mensaje SEI de fin de segmento de refinamiento progresivo con el mismo

progressive\_refinement\_id, la última imagen del conjunto etiquetado de imágenes codificadas consecutivas en orden de decodificación es la primera de estas imágenes en orden de decodificación.

- De lo contrario, la última imagen del conjunto etiquetado de imágenes codificadas consecutivas en orden de decodificación es la última imagen de la CVS en orden de decodificación.

- De lo contrario, la última imagen del conjunto etiquetado de imágenes codificadas consecutivas en orden de decodificación es la imagen siguiente:

- Si la CVS contiene una o más imágenes que siguen a la imagen actual en orden de decodificación y están asociadas a un mensaje SEI de fin de segmento de refinamiento progresivo con el mismo progressive\_refinement\_id y preceden a cualquier imagen *picC* en la CVS para la cual  $PicOrderCnt(picC)$  es mayor que  $PicOrderCnt(picA)$  más *pic\_order\_cnt\_delta*, donde  $PicOrderCnt(picC)$  y  $PicOrderCnt(picA)$  son los valores  $PicOrderCntVal$  de *picC* y *picA*, respectivamente, justo después de la invocación del proceso de decodificación para el recuento de orden de imágenes para *picC*, la última imagen del conjunto etiquetado de imágenes codificadas consecutivas en orden de decodificación es la primera de estas imágenes en orden de decodificación.

- De lo contrario, si la CVS contiene una o más imágenes *picD* que siguen a la imagen actual en orden de decodificación para las cuales  $PicOrderCnt(picD)$  es mayor que  $PicOrderCnt(picA)$  más *pic\_order\_cnt\_delta*, donde  $PicOrderCnt(picD)$  y  $PicOrderCnt(picA)$  son los valores  $PicOrderCntVal$  de *picD* y *picA*, respectivamente, justo después de la invocación del proceso de decodificación para el recuento de orden de imágenes para *picD*, la última imagen del conjunto etiquetado de imágenes codificadas consecutivas en orden de decodificación es la última imagen que precede a la primera de estas imágenes en orden de decodificación.

- De lo contrario, la última imagen del conjunto etiquetado de imágenes codificadas consecutivas en orden de decodificación es la última imagen de la CVS en orden de decodificación.

[0205] El valor de *pic\_order\_cnt\_delta* debe estar dentro del intervalo de 0 a 256, ambos inclusive.

[0206] En un modo de realización, la subcláusula D.3.13 se modifica como sigue:

**film\_grain\_characteristics\_persistence\_flag** especifica la persistencia del mensaje SEI de características de granulado de película.

*film\_grain\_characteristics\_persistence\_flag* igual a 0 especifica que el mensaje SEI de características de granulado de película se aplica solamente a la imagen decodificada actual.

[0207] Sea *picA* la imagen actual. *film\_grain\_characteristics\_persistence\_flag* igual a 1 especifica que el mensaje SEI de características de granulado de película persiste en orden de salida hasta que cualquiera de las siguientes condiciones sea verdadera:

- Comienza una nueva CVS.

- El flujo de bits termina.

- Se proporciona una imagen *picB* en una unidad de acceso que contiene un mensaje SEI de características de granulado de película para la que  $PicOrderCnt(picB)$  es mayor que  $PicOrderCnt(picA)$ , donde  $PicOrderCnt(picB)$  y  $PicOrderCnt(picA)$  son los valores  $PicOrderCntVal$  de *picB* y *picA*, respectivamente, justo después de la invocación del proceso de decodificación para el recuento de orden de imágenes para *picB*.

[0208] En un modo de realización, la subcláusula D.3.14 se modifica como sigue:

**tone\_map\_persistence\_flag** especifica la persistencia del mensaje SEI de información de correlación de tonos.

*tone\_map\_persistence\_flag* igual a 0 especifica que la información de correlación de tonos se aplica solamente a la imagen decodificada actual.

[0209] Sea *picA* la imagen actual. *tone\_map\_persistence\_flag* igual a 1 especifica que la información de correlación de tonos persiste en orden de salida hasta que cualquiera de las siguientes condiciones sea verdadera:

- Comienza una nueva CVS.



- Se proporciona una imagen *picB* en una unidad de acceso que contiene un mensaje SEI de información de correlación de tonos con el mismo valor de *tone\_map\_id* para la que *PicOrderCnt(picB)* es mayor que *PicOrderCnt(picA)*, donde *PicOrderCnt(picB)* y *PicOrderCnt(picA)* son los valores *PicOrderCntVal* de *picB* y *picA*, respectivamente, justo después de la invocación del proceso de descodificación para el recuento de orden de imágenes para *picB*.

[0210] En un modo de realización, la subcláusula D.3.16 se modifica como sigue:

**frame\_packing\_arrangement\_persistence\_flag** especifica la persistencia del mensaje SEI de disposición de empaquetado de tramas.

*frame\_packing\_arrangement\_persistence\_flag* igual a 0 especifica que el mensaje SEI de disposición de empaquetado de tramas se aplica solamente a la trama descodificada actual.

[0211] Sea *picA* la imagen actual. *frame\_packing\_arrangement\_persistence\_flag* igual a 1 especifica que el mensaje SEI de disposición de empaquetado de tramas persiste en orden de salida hasta que cualquiera de las siguientes condiciones sea verdadera:

- Comienza una nueva CVS.

- El flujo de bits termina

- Se proporciona una imagen *picB* en una unidad de acceso que contiene un mensaje SEI de disposición de empaquetado de tramas con el mismo valor de *frame\_packing\_arrangement\_id* para la que *PicOrderCnt(picB)* es mayor que *PicOrderCnt(picA)*, donde *PicOrderCnt(picB)* y *PicOrderCnt(picA)* son los valores *PicOrderCntVal* de *picB* y *picA*, respectivamente, justo después de la invocación del proceso de descodificación para el recuento de orden de imágenes para *picB*.

[0212] En un modo de realización, la subcláusula D.3.17 se modifica como sigue:

**display\_orientation\_persistence\_flag** especifica la persistencia del mensaje SEI de orientación de visualización.

*display\_orientation\_persistence\_flag* igual a 0 especifica que el mensaje SEI de orientación de visualización se aplica solamente a la imagen descodificada actual.

[0213] Sea *picA* la imagen actual. *display\_orientation\_persistence\_flag* igual a 1 especifica que el mensaje SEI de orientación de visualización persiste en orden de salida hasta que una o más de las siguientes condiciones sean verdaderas:

- Comienza una nueva CVS.

- El flujo de bits termina.

- Se proporciona una imagen *picB* en una unidad de acceso que contiene un mensaje SEI de orientación de visualización para la que *PicOrderCnt(picB)* es mayor que *PicOrderCnt(picA)*, donde *PicOrderCnt(picB)* y *PicOrderCnt(picA)* son los valores *PicOrderCntVal* de *picB* y *picA*, respectivamente, justo después de la invocación del proceso de descodificación para el recuento de orden de imágenes para *picB*.

[0214] La semántica de la estructura del mensaje SEI de información de imágenes especificado en la subcláusula D.3.18 se sustituye por lo siguiente:

La estructura del mensaje SEI de información de imágenes proporciona información para una lista de entradas, algunas de las cuales corresponden al *conjunto de imágenes de destino*, y consiste en una serie de imágenes que *comienzan desde la imagen actual hasta la última imagen* en orden de descodificación en la CVS o la *última imagen en orden de descodificación en el período actual de restablecimiento de POC, la que sea anterior*.

[0215] La primera entrada en la estructura del mensaje SEI de información de imágenes corresponde a la imagen actual. Cuando hay una imagen en el *conjunto de imágenes de destino* que tiene *PicOrderCntVal* igual a la variable *entryPicOrderCnt[i]* como se especifica a continuación, la entrada *i* corresponde a una imagen del *conjunto de imágenes de destino*. El orden de descodificación de las imágenes en el *conjunto de imágenes de destino* que corresponden a las entradas en la estructura del mensaje SEI de información de imágenes corresponde a valores crecientes de *i* en la lista de entradas.

[0216] Cualquier imagen *picB* en el *conjunto de imágenes de destino* que tenga *PicOrderCntVal* igual a *entryPicOrderCnt[i]* para cualquier *i* en el intervalo de 0 a *num\_entries\_in\_sop\_minus1*, ambos inclusive, *donde PicOrderCntVal* es el valor de *PicOrderCntVal* de *picB* justo después de la invocación del proceso de descodificación para el recuento de orden de imágenes para *picB*, corresponderá a una entrada en la lista de entradas.

[0217] La estructura del mensaje SEI de información de imágenes no estará presente en una CVS para la cual el SPS activo tiene *long\_term\_ref\_pics\_present\_flag* igual a 1 o *num\_short\_term\_ref\_pic\_sets* igual a 0.

[0218] La estructura del mensaje SEI de información de imágenes no debe estar presente en ninguna unidad de acceso que tenga *TemporalId* mayor que 0 o contenga una imagen RASL, RADL o de subcapa que no es de referencia. Cualquier imagen del *conjunto de imágenes de destino* que corresponde a una entrada diferente a la primera entrada descrita en la estructura del mensaje SEI de información de imágenes no será una imagen IRAP.

[0219] *sop\_seq\_parameter\_set\_id* indica y será igual al valor *sps\_seq\_parameter\_set\_id* del SPS activo. El valor de *sop\_seq\_parameter\_set\_id* debe estar en el intervalo de 0 a 15, ambos inclusive.

[0220] *num\_entries\_in\_sop\_minus1* más 1 especifica el número de entradas en la estructura del mensaje SEI de información de imágenes. *num\_entries\_in\_sop\_minus1* debe estar en el intervalo de 0 a 1023, ambos inclusive.

[0221] *sop\_vcl\_nut[i]*, cuando la *i*-ésima entrada corresponde a una imagen del *conjunto de imágenes de destino*, indica y será igual al valor *nal\_unit\_type* de la imagen correspondiente a la *i*-ésima entrada.

[0222] *sop\_temporal\_id[i]*, cuando la *i*-ésima entrada corresponde a una imagen del *conjunto de imágenes de destino*, indica y será igual al valor *TemporalId* de la imagen correspondiente a la *i*-ésima entrada. El valor de 7 para *sop\_temporal\_id[i]* está reservado para uso futuro por parte de ITU-T | ISO/IEC y no estará presente en flujos de bits que se ajusten a esta versión de esta memoria descriptiva. Los descodificadores ignorarán la estructura de mensajes SEI de información de imágenes que contengan el valor 7 para *sop\_temporal\_id[i]*.

[0223] *sop\_short\_term\_rps\_idx[i]*, cuando la *i*-ésima entrada corresponde a una imagen del *conjunto de imágenes de destino*, indica y será igual al índice, en la lista de RPS candidatos a corto plazo incluidos en el SPS activo, del RPS candidato a corto plazo usado por la imagen correspondiente a la *i*-ésima entrada para la obtención del conjunto de imágenes de referencia a corto plazo. *sop\_short\_term\_rps\_idx[i]* debe estar en el intervalo de 0 a *num\_short\_term\_ref\_pic\_sets* - 1, ambos inclusive.

[0224] *sop\_poc\_delta[i]* se usa para especificar el valor de la variable *entryPicOrderCnt[i]* para la *i*-ésima entrada descrita en la estructura del mensaje SEI de información de imágenes. *sop\_poc\_delta[i]* debe estar en el intervalo de  $(-\text{MaxPicOrderCntLsb})/2+1$  a  $\text{MaxPicOrderCntLsb}/2-1$ , ambos inclusive.

[0225] La variable *entryPicOrderCnt[i]* se obtiene como sigue:

```
entryPicOrderCnt[0] = PicOrderCnt(currPic)
for( i = 1; i <= num_entries_in_sop_minus1; i++)
    entryPicOrderCnt[i] = entryPicOrderCnt[i - 1] + sop_poc_delta[i]
donde currPic es la imagen actual.
```

[0226] En un modo de realización, la subcláusula D.3.24 se modifica como sigue:

El mensaje SEI de información de actualización de región indica si los segmentos de fragmento a los que se aplica el mensaje SEI actual pertenecen a una región actualizada de la imagen actual (como se define más adelante).

[0227] Una unidad de acceso que no es una unidad de acceso de IRAP y que contiene un mensaje SEI de punto de recuperación se denomina unidad de acceso de actualización de descodificación gradual (GDR), y su imagen correspondiente se denomina imagen GDR. La unidad de acceso correspondiente a la imagen de punto de recuperación indicada se denomina unidad de acceso de punto de recuperación.

[0228] Si hay una imagen *picB* que sigue a la imagen GDR *picA* en orden de descodificación en la CVS y *PicOrderCnt(picB)* es igual a *PicOrderCnt(picA)* más el valor de *recovery\_poc\_cnt* en el mensaje SEI de punto de recuperación, *donde PicOrderCnt(picA)* y *PicOrderCnt(picB)* son los valores *PicOrderCntVal* de *picA* y *picB*, respectivamente, justo después de la invocación del proceso de descodificación para el recuento de orden de imágenes para *picB*, entonces la variable *lastPicInSet* será la imagen de punto de recuperación. De lo contrario, *lastPicInSet* será la imagen que precede inmediatamente a la imagen de punto de recuperación en orden de salida. La imagen *lastPicInSet* no debe preceder a la imagen GDR en orden de descodificación.

#### Adición de una función de sintaxis

[0229] En un modo de realización, *more\_data\_in\_slice\_segment\_header\_extension()* se especifica como sigue:

- Si (la posición actual en la estructura de sintaxis slice\_segment\_header()) - (la posición que sigue inmediatamente a slice\_segment\_header\_extension\_length) es menor que (slice\_segment\_header\_extension\_length \* 8), el valor de retorno de more\_data\_in\_slice\_segment\_header\_extension() es igual a TRUE.

- De lo contrario, el valor de retorno de more\_data\_in\_slice\_segment\_header\_extension() es igual a FALSE.

#### **Adición de dos definiciones**

**[0230]** En un modo de realización, se añaden las siguientes definiciones a la subcláusula F.3:

**período de restablecimiento del recuento de orden de imágenes (POC):** Una secuencia de unidades de acceso en orden de descodificación, *que comienza con una unidad de acceso con poc\_reset\_idc igual a 1 o 2 y un valor particular de poc\_reset\_period\_id y que incluye todas las unidades de acceso que tienen el mismo valor de poc\_reset\_period\_id o tienen poc\_reset\_idc igual a 0.*

**imagen de restablecimiento del recuento de orden de imágenes (POC):** Una imagen que es la primera imagen, en orden de descodificación, de una capa de un período de restablecimiento de POC.

#### **Sintaxis y semántica de ampliación del conjunto de parámetros de vídeo**

**[0231]**

**Tabla 1. Sintaxis de ejemplo de vps\_extension()**

vps_extension() {	Descriptor
<b>avc_base_layer_flag</b>	u(1)
...	
<b>direct_dependency_type[i][j]</b>	u(v)
}	
<b>vps_poc_msb_present_flag</b>	u(1)
<b>vps_shvc_reserved_zero_flag</b>	u(1)
if (vps_vui_present_flag) {	
while(!byte_aligned( ))	
<b>vps_vui_alignment_bit_equal_to_one</b>	u(1)
vps_vui()	
}	
}	

**[0232]** **vps\_poc\_msb\_present\_flag** igual a 0 especifica que el elemento de sintaxis poc\_msb\_val no está presente en las cabeceras de segmento de fragmento del fragmento que hace referencia al VPS. **vps\_poc\_msb\_present\_flag** igual a 1 especifica que el elemento de sintaxis poc\_msb\_val puede estar presente en las cabeceras de segmento de fragmento del fragmento que hace referencia al VPS.

**[0233]** Cuando el valor de vps\_poc\_msb\_present\_flag es igual a 0, poc\_info\_present\_flag será igual a 0.

#### **Cambios en la semántica de cabecera de unidad NAL**

**[0234]** En un modo de realización, la siguiente deducción se añade a la semántica de la cabecera de unidad NAL:

La variable CraOrBlaPicFlag se obtiene como sigue:

$$\text{CraOrBlaPicFlag} = (\text{nal\_unit\_type} == \text{BLA\_W\_LP} \parallel \text{nal\_unit\_type} == \text{BLA\_N\_LP} \parallel \text{nal\_unit\_type} == \text{BLA\_W\_RADL} \parallel \text{nal\_unit\_type} == \text{CRA\_NUT})$$

#### **Sintaxis y semántica de RBSP de conjunto de parámetros de imagen**

**[0235]**

Tabla 2. Sintaxis de ejemplo de pic\_parameter\_set\_rbsp()

pic_parameter_set_rbsp() {	Descriptor
...	
<b>pps_extension_flag</b>	u(1)
if(pps_extension_flag) {	
<b>poc_reset_info_present_flag</b>	u(1)
<b>pps_extension2_flag</b>	u(1)
if(pps_extension2_flag)	
while(more_rbsp_data())	
<b>pps_extension_data_flag</b>	u(1)
}	
rbp_trailing_bits()	
}	

[0236] **pps\_extension\_flag** igual a 0 especifica que no hay ningún elemento de sintaxis pps\_extension\_data\_flag presente en la estructura de sintaxis de RBSP de PPS. Cuando slice\_segment\_header\_extension\_present\_flag es igual a 0, pps\_extension\_flag será igual a 0 en flujos de bits que se ajusten a esta versión de esta memoria descriptiva, el valor de 1 para pps\_extension\_flag está reservado para uso futuro por ITU-T | ISO/IEC, y los decodificadores ignorarán todos los datos que sigan al valor 1 para pps\_extension\_flag en una unidad NAL de PPS.

[0237] **poc\_reset\_info\_present\_flag** igual a 0 especifica que el elemento de sintaxis poc\_reset\_idc no está presente en las cabeceras de segmento de fragmento de los fragmentos que hacen referencia al PPS. poc\_reset\_info\_present\_flag igual a 1 especifica que el elemento de sintaxis poc\_reset\_idc está presente en las cabeceras de segmento de fragmento de los fragmentos que hacen referencia al PPS.

[0238] **pps\_extension2\_flag** igual a 0 especifica que no hay ningún elemento de sintaxis pps\_extension\_data\_flag presente en la estructura de sintaxis de RBSP de PPS. pps\_extension2\_flag será igual a 0 en flujos de bits que se ajusten a esta versión de esta memoria descriptiva. El valor de 1 para pps\_extension2\_flag está reservado para uso futuro por parte de ITU-T | ISO/IEC. Los decodificadores ignorarán todos los datos que sigan al valor 1 para pps\_extension2\_flag en una unidad NAL de PPS.

#### 1.1.1 Sintaxis y semántica general de cabecera de segmento de fragmento

[0239]

Tabla 3. Sintaxis de ejemplo de slice\_segment\_header() (\*\*: eliminado)

slice_segment_header() {	Descriptor
...	
if(!dependent_slice_segment_flag) {	
i = 0	
if(num_extra_slice_header_bits > i) {	
i++	
<b>discardable_flag</b>	u(1)
}	
if(num_extra_slice_header_bits > i) {	
i++	
<b>cross_layerbla_flag</b>	u(1)
}	
*** if( num_extra_slice_header_bits > i) {	
*** i++	
*** <b>poc_reset_flag</b>	*** u(1)

*** }	
...	
if( slice_segment_header_extension_present_flag) {	
<b>slice_segment_header_extension_length</b>	ue(v)
if( poc_reset_info_present_flag)	
<b>poc_reset_idc</b>	u(2)
if(poc_reset_idc > 0)	
<b>poc_reset_period_id</b>	u(6)
if(poc_reset_idc == 3) {	
<b>full_poc_reset_flag</b>	u(1)
<b>poc_lsb_val</b>	u(v)
}	
if( vps_poc_msb_present_flag && CraOrBlaPicFlag)	
<b>poc_msb_val</b>	ue(v)
while(more_data_in_slice_segment_header_extension())	
<b>slice_segment_header_extension_data_bit</b>	u(1)
}	
byte_alignment( )	
}	

[0240] De forma alternativa, poc\_reset\_period\_id se señala usando un número diferente de bits, por ejemplo, codificado como u(14).

[0241] Cuando están presentes, el valor de los elementos de sintaxis de cabecera de segmento de fragmento slice\_pic\_parameter\_set\_id, pic\_output\_flag, no\_output\_of\_prior\_pics\_flag, slice\_pic\_order\_cnt\_lsb, short\_term\_ref\_pic\_set\_sps\_flag, short\_term\_ref\_pic\_set\_idx, num\_long\_term\_sps, num\_long\_term\_pics, slice\_temporal\_mvp\_enabled\_flag, discardable\_flag, cross\_layer\_bla\_flag, inter\_layer\_pred\_enabled\_flag, num\_inter\_layer\_ref\_pics\_minus1, poc\_reset\_idc, poc\_reset\_pic\_id, full\_poc\_reset\_flag, poc\_lsb\_val y poc\_msb\_val será el mismo en todas las cabeceras de segmento de fragmento de una imagen codificada. Cuando están presentes, el valor de los elementos de sintaxis de cabecera de segmento de fragmento lt\_idx\_sps[i], poc\_lsb\_lt[i], used\_by\_curr\_pic\_lt\_flag[i], delta\_poc\_msb\_present\_flag[i], delta\_poc\_msb\_cycle\_lt[i] e inter\_layer\_pred\_layer\_idc[i] será el mismo en todas las cabeceras de segmento de fragmento de una imagen codificada para cada posible valor de i.

[0242] (\*\* ELIMINADO: poc\_reset\_flag igual a 1 especifica que el contador de orden de imágenes obtenido para la imagen actual es igual a 0. poc\_reset\_flag igual a 0 especifica que el contador de orden de imágenes obtenido para la imagen actual puede ser igual, o no, a 0. Cuando no está presente, se infiere que el valor de poc\_reset\_flag es igual a 0.

[0243] NOTA - Cuando poc\_reset\_flag es igual a 1 en una imagen de capa base, PicOrderCntVal se obtiene de manera diferente dependiendo de si se aplica el proceso de descodificación de la subcláusula 8.3.1 o la subcláusula F.8.3.1. Además, cuando una imagen de capa base con poc\_reset\_flag igual a 1 es prevTidOPic de acuerdo con la subcláusula 8.3.1 o F.8.3.1, la variable prevPicOrderCntLsb se obtiene de manera diferente en las subcláusulas 8.3.1 y F.8.3.1. Con el fin de evitar que PicOrderCntMsb se actualice incorrectamente en una de las subcláusulas 8.3.1 o F.8.3.1, cuando prevTidOPic es una imagen de capa base con poc\_reset\_flag igual a 1 y una de las dos condiciones siguientes es verdadera para prevPicOrderCntLsb obtenido con una de las subcláusulas 8.3.1 o F.8.3.1, el valor de pic\_order\_cnt\_lsb de prevTidOPic será tal que la misma condición también es verdadera para prevPicOrderCntLsb obtenido con la otra de las subcláusulas 8.3.1 o F.8.3.1:

- (slice\_pic\_order\_cnt\_lsb < prevPicOrderCntLsb) && ((prevPicOrderCntLsb - slice\_pic\_order\_cnt\_lsb) >= (MaxPicOrderCntLsb / 2))

- (slice\_pic\_order\_cnt\_lsb > prevPicOrderCntLsb) && ((slice\_pic\_order\_cnt\_lsb - prevPicOrderCntLsb) > (MaxPicOrderCntLsb / 2) ) \*\*\*

**[0244]** poc\_reset\_idc igual a 0 especifica que no se restablecen ni los bits más significativos ni los bits menos significativos del valor de recuento de orden de imágenes para la imagen actual. poc\_reset\_idc igual a 1 especifica que solo se pueden restablecer los bits más significativos del valor de recuento de orden de imágenes para la imagen actual. poc\_reset\_idc igual a 2 especifica que pueden restablecerse tanto los bits más significativos como los bits menos significativos del valor de recuento de orden de imágenes para la imagen actual. poc\_reset\_idc igual a 3 especifica que solo se pueden restablecer los bits más significativos o tanto los bits más significativos como los bits menos significativos del valor de recuento de orden de imágenes para la imagen actual y se señala información de recuento de orden de imágenes adicional. Cuando no está presente, se infiere que el valor de poc\_reset\_idc es igual a 0.

**[0245]** Es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones:

- El valor de poc\_reset\_idc no debe ser igual a 1 o 2 para una imagen RASL, una imagen RADL, una imagen de subcapa que no es de referencia, una imagen que tiene TemporalId mayor que 0 o una imagen que tiene discardable\_flag igual a 1.

- El valor de poc\_reset\_idc de todas las imágenes en una unidad de acceso será el mismo.

- Cuando la imagen en una unidad de acceso con nuh\_layer\_id igual a 0 es una imagen IRAP con un valor particular de nal\_unit\_type y hay al menos otra imagen en la misma unidad de acceso con un valor diferente de nal\_unit\_type, el valor de poc\_reset\_idc será igual a 1 o 2 para todas las imágenes en la unidad de acceso.

- Cuando hay al menos una imagen que tiene nuh\_layer\_id mayor que 0 y que es una imagen IDR con un valor particular de nal\_unit\_type en una unidad de acceso y hay al menos otra imagen en la misma unidad de acceso con un valor diferente de nal\_unit\_type, el valor de poc\_reset\_idc será igual a 1 o 2 para todas las imágenes en la unidad de acceso.

- El valor de poc\_reset\_idc de una imagen CRA o BLA será inferior a 3.

- Cuando la imagen con nuh\_layer\_id igual a 0 en una unidad de acceso es una imagen IDR y hay al menos una imagen no IDR en la misma unidad de acceso, el valor de poc\_reset\_idc será igual a 2 para todas las imágenes en la unidad de acceso.

- Cuando la imagen con nuh\_layer\_id igual a 0 en una unidad de acceso no es una imagen IDR, el valor de poc\_reset\_idc no será igual a 2 para ninguna imagen en la unidad de acceso.

**[0246]** El valor de poc\_reset\_idc de una unidad de acceso es el valor de poc\_reset\_idc de las imágenes en la unidad de acceso.

**[0247]** poc\_reset\_period\_id identifica un período de restablecimiento de POC. No habrá dos imágenes consecutivas en orden de decodificación en la misma capa que tengan el mismo valor de poc\_reset\_period\_id y poc\_reset\_idc igual a 1 o 2. Cuando no está presente, el valor de poc\_reset\_period\_id se infiere de la siguiente manera:

- Si la imagen anterior picA que tiene poc\_reset\_period\_id presente en la cabecera de segmento de fragmento está presente en la misma capa del flujo de bits que la imagen actual, se infiere que el valor de poc\_reset\_period\_id es igual al valor de poc\_reset\_period\_id de picA.

- De lo contrario, se infiere que el valor de poc\_reset\_period\_id es igual a 0. **NOTA** - No está prohibido que múltiples imágenes en una capa tengan el mismo valor de poc\_reset\_period\_id y que tengan poc\_reset\_idc igual a 1 o 2 salvo que dichas imágenes se produzcan en dos unidades de acceso consecutivas en orden de decodificación. Para minimizar la probabilidad de que dos imágenes de este tipo aparezcan en el flujo de bits debido a pérdidas de imágenes, extracción de flujo de bits, búsqueda u operaciones de empalme, los codificadores deben establecer el valor de poc\_reset\_period\_id como un valor aleatorio para cada período de restablecimiento de POC (sujeto a las restricciones especificadas anteriormente).

**[0248]** Es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones:

- Un período de restablecimiento de POC no incluirá más de una unidad de acceso con poc\_reset\_idc igual a 1 o 2.

- Una unidad de acceso con poc\_reset\_idc igual a 1 o 2 será la primera unidad de acceso en un período de restablecimiento de POC.

- Una imagen que sigue a una imagen de restablecimiento de POC en orden de decodificación no debe preceder, en orden de salida, a otra imagen que preceda a la imagen de restablecimiento de POC en orden de decodificación.

**[0249] full\_poc\_reset\_flag** igual a 1 especifica que tanto los bits más significativos como los bits menos significativos del valor de recuento de orden de imágenes para la imagen actual se restablecen cuando la imagen anterior en orden de decodificación en la misma capa no pertenece al mismo período de restablecimiento de POC. **full\_poc\_reset\_flag** igual a 0 especifica que solo los bits más significativos del valor de recuento de orden de imágenes para la imagen actual se restablecen cuando la imagen anterior en orden de decodificación en la misma capa no pertenece al mismo período de restablecimiento de POC.

**[0250] poc\_lsb\_val** especifica un valor que puede usarse para obtener el recuento de orden de imágenes de la imagen actual. La longitud del elemento de sintaxis **poc\_lsb\_val** es de  $\log_2\_max\_pic\_order\_cnt\_lsb\_minus4 + 4$  bits.

**[0251]** Es un requisito de conformidad de flujo de bits que, cuando **poc\_reset\_idc** es igual a 3, y la imagen anterior **picA** en orden de decodificación que está en la misma capa que la imagen actual, que tiene **poc\_reset\_idc** igual a 1 o 2 y pertenece al mismo período de restablecimiento de POC, está presente en el flujo de bits, **picA** será la misma imagen que la imagen anterior en orden de decodificación que está en la misma capa que la imagen actual, que no es una imagen RASL, una imagen RADL o una imagen de subcapa que no es de referencia, y que tiene **TemporalId** igual a 0 y **discardable\_flag** igual a 0, y el valor de **poc\_lsb\_val** de la imagen actual será igual al valor de **slice\_pic\_order\_cnt\_lsb** de **picA**.

**[0252] poc\_msb\_val** especifica el valor de los bits más significativos del valor de recuento de orden de imágenes de la imagen actual, que es una imagen CRA o BLA. El valor de **poc\_msb\_val** también se usa para obtener el valor usado para disminuir los valores de recuento de orden de imágenes de las imágenes decodificadas previamente en la misma capa que la imagen actual. El valor de **poc\_msb\_val** debe estar en el intervalo de 0 to  $2^{32-\log_2\_max\_pic\_order\_cnt\_lsb\_minus4}$ , ambos inclusive. El valor de **poc\_msb\_val** será igual a la diferencia entre los valores de los bits más significativos de los recuentos de orden de imágenes de la imagen actual y la imagen de restablecimiento de POC anterior en la misma capa o la imagen IDR anterior en la misma capa, la que esté más cerca, en orden de decodificación, a la imagen actual. Si ninguna imagen está presente, el valor de **poc\_msb\_val** puede ser cualquier valor en el intervalo permitido.

#### **Proceso de decodificación para una imagen codificada con nuh\_layer\_id igual a 0**

**[0253]** En un modo de realización, la subcláusula 8.1.1 de la especificación se cambia como sigue:

- Las referencias a las subcláusulas 8.2, 8.3, 8.3.1, 8.3.2, 8.3.3, 8.3.4, 8.4, 8.5, 8.6 y 8.7 se sustituyen por referencias a las subcláusulas F.8.2, F.8.3, F.8.3.1, F.8.3.2, F.8.3.3, F.8.3.4, F.8.4, F.8.5, F.8.6 y F.8.7, respectivamente.

- Al final de la subcláusula, se añade el punto 5 como se indica a continuación:

1. Cuando **FirstPicInLayerDecodedFlag[0]** es igual a 0, **FirstPicInLayerDecodedFlag[0]** es igual a 1.

#### **Proceso de decodificación para iniciar la decodificación de una imagen codificada con nuh\_layer\_id mayor que 0**

**[0254]** Cada imagen a la que se hace referencia en esta subcláusula es una imagen codificada completa.

**[0255]** El proceso de decodificación funciona de la siguiente manera para la imagen actual **CurrPic**:

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

2. Los procesos en la subcláusula F.8.3 especifican los siguientes procesos de decodificación usando elementos de sintaxis 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 F.8.3.1. Esto solo tiene que invocarse para el primer segmento de fragmento de una imagen. Es un requisito de conformidad de flujo de bits que **PicOrderCntVal** permanezca sin cambios dentro de una unidad de acceso.

- Se invoca el proceso de decodificación para RPS en la subcláusula F.8.3.2, en el que solo las imágenes de referencia con un **nuh\_layer\_id** igual al de **CurrPic** pueden marcarse como "no usadas como referencia" o "usadas como referencia a largo plazo" y cualquier imagen con un valor diferente de **nuh\_layer\_id** no está marcada. Esto solo tiene que invocarse para el primer segmento de fragmento de una imagen.

- Cuando FirstPicInLayerDecodedFlag[nuh\_layer\_id] es igual a 0, se invoca el proceso de decodificación para generar imágenes de referencia no disponibles especificadas en la subcláusula F.8.1.5, que solo tiene que invocarse para el primer segmento de fragmento de una imagen.

5 - Cuando FirstPicInLayerDecodedFlag[nuh\_layer\_id] no es igual a 0 y la imagen actual es una imagen IRAP con NoRaslOutputFlag igual a 1, se invoca el proceso de decodificación para generar imágenes de referencia no disponibles especificadas en la subcláusula F.8.3.3, que solo tiene que invocarse para el primer segmento de fragmento de una imagen.

## 10 **Proceso de decodificación para el recuento de orden de imágenes**

[0256] La salida de este proceso es PicOrderCntVal, el recuento de orden de imágenes de la imagen actual.

15 [0257] Los recuentos de orden de imágenes se usan para identificar imágenes, para obtener parámetros de movimiento en modo de combinación y predicción de vectores de movimiento, y para la verificación de conformidad de decodificador (véase la subcláusula C.5).

[0258] Cada imagen codificada está asociada a una variable de recuento de orden de imágenes, denominada PicOrderCntVal.

20 [0259] (\*\* ELIMINADO: Si FirstPicInLayerDecodedFlag[nuh\_layer\_id] es igual a 0 o la imagen actual es una imagen IRAP con NoRaslOutputFlag igual a 1, la variable PicOrderCntMsb se establece igual a 0. De lo contrario, PicOrderCntMsb se obtiene de la siguiente manera:

25 - La variable prevPicOrderCntLsb se establece igual a PrevPicOrderCnt[nuh\_layer\_id] & (MaxPicOrderCntLsb - 1).

- La variable prevPicOrderCntMsb se establece igual a PrevPicOrderCnt[nuh\_layer\_id] - prevPicOrderCntLsb.

30 - PicOrderCntMsb se obtiene de la siguiente manera:

```

35         if((slice_pic_order_cnt_lsb < prevPicOrderCntLsb) &&
            ((prevPicOrderCntLsb - slice_pic_order_cnt_lsb) >= (
            MaxPicOrderCntLsb / 2)))
            PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb
        else if((slice_pic_order_cnt_lsb > prevPicOrderCntLsb) &&
            ((slice_pic_order_cnt_lsb - prevPicOrderCntLsb) > (MaxPicOrderCntLsb /
40         2)))
            PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb
        else
            PicOrderCntMsb = prevPicOrderCntMsb

```

[0260] PicOrderCntVal se obtiene de la siguiente manera:

45 PicOrderCntVal = PicOrderCntMsb + slice\_pic\_order\_cnt\_lsb

[0261] Cuando poc\_reset\_flag es igual a 1, se aplican las siguientes etapas en el orden indicado:

50 - El PicOrderCntVal de cada imagen que está en el DPB y pertenece a la misma capa que la imagen actual se decrementa en PicOrderCntVal.

- PrevPicOrderCnt[nuh\_layer\_id] se decrementa en PicOrderCntVal.

55 - PicOrderCntVal se establece igual a 0.

[0262] Cuando la imagen actual no es una imagen RASL, una imagen RADL o una imagen de subcapa que no es de referencia y la imagen actual tiene TemporalId igual a 0, PrevPicOrderCnt[nuh\_layer\_id] se establece igual a PicOrderCntVal. \*\*\*)

60 [0263] Si FirstPicInLayerDecodedFlag[nuh\_layer\_id] es igual a 1 y la imagen actual es una *imagen de restablecimiento de POC*, se aplica lo siguiente:

- Las variables pocMsbDelta, pocLsbDelta y DeltaPocVal se obtienen de la forma siguiente:

```

65         if(poc_reset_idc == 3)
            pocLsbVal = poc_lsb_val
        else

```



```

        pocLsbVal = slice_pic_order_cnt_lsb
    if(vps_poc_msb_present_flag && CraOrBlaPicFlag)
        pocMsbDelta = poc_msb_val * MaxPicOrderCntLsb
    else {
5         prevPicOrderCntLsb =
        PrevPicOrderCnt[nuh_layer_id] & (MaxPicOrderCntLsb - 1)
        prevPicOrderCntMsb =
        PrevPicOrderCnt[nuh_layer_id] - prevPicOrderCntLsb
10        pocMsbDelta = getCurrMsb(pocLsbVal, prevPicOrderCntLsb,
        prevPicOrderCntMsb, MaxPicOrderCntLsb)
    }
    if(poc_reset_idc == 2 || (poc_reset_idc == 3 && full_poc_reset_flag))
        pocLsbDelta = pocLsbVal
15    else
        pocLsbDelta = 0
    DeltaPocVal = pocMsbDelta + pocLsbDelta

```

- El PicOrderCntVal de cada imagen que está en el DPB y pertenece a la misma capa que la imagen actual se decrementa en DeltaPocVal.

- El PicOrderCntVal de la imagen actual se obtiene de la forma siguiente:

```

    if(poc_reset_idc == 1)
        PicOrderCntVal = slice_pic_order_cnt_lsb
25    else if(poc_reset_idc == 2)
        PicOrderCntVal = 0
    else { // poc_reset_idc == 3
        PicOrderCntMsb = getCurrMsb(slice_pic_order_cnt_lsb, full_poc_reset_flag ? 0 :
        poc_lsb_val, 0, MaxPicOrderCntLsb)
30        PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
    }

```

- El valor de PrevPicOrderCnt[nuh\_layer\_id] se obtiene de la forma siguiente:

- Si la imagen actual no es una imagen RASL, una imagen RADL o una imagen de subcapa que no es de referencia, y la imagen actual tiene TemporalId igual a 0 y discardable\_flag igual a 0, PrevPicOrderCnt[nuh\_layer\_id] se establece igual a PicOrderCntVal.

- De lo contrario, cuando poc\_reset\_idc es igual a 3, PrevPicOrderCnt[nuh\_layer\_id] se establece igual a full\_poc\_reset\_flag ? 0: poc\_lsb\_val.

**[0264]** De lo contrario, se aplica lo siguiente:

- El PicOrderCntVal de la imagen actual se obtiene de la forma siguiente:

```

if( !FirstPicInLayerDecodedFlag[ nuh_layer_id ] ) {
    if( poc_reset_idc == 1 )
        PicOrderCntVal = slice_pic_order_cnt_lsb
    else if( poc_reset_idc == 2 )
        PicOrderCntVal = 0
    else if( poc_reset_idc == 3 ) {
        PicOrderCntMsb = getCurrMsb(slice_pic_order_cnt_lsb,
full_poc_reset_flag ? 0 : poc_lsb_val,
                                0, MaxPicOrderCntLsb )
        PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
    } else {
        if( vps_poc_msb_present_flag && CraOrBlaPicFlag ) {
            PicOrderCntMsb = poc_msb_val * MaxPicOrderCntLsb
            PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
        }
        else
            PicOrderCntVal = slice_pic_order_cnt_lsb
    }
} else {
    if( vps_poc_msb_present_flag && CraOrBlaPicFlag )
        PicOrderCntMsb = poc_msb_val * MaxPicOrderCntLsb

    else if( la imagen actual es una imagen IRAP con NoRaslOutputFlag igual to 1 )

        PicOrderCntMsb = 0
    else {
        prevPicOrderCntLsb =
PrevPicOrderCnt[ nuh_layer_id ] & ( MaxPicOrderCntLsb - 1 ).
        prevPicOrderCntMsb =
PrevPicOrderCnt[ nuh_layer_id ] - prevPicOrderCntLsb
        PicOrderCntMsb = getCurrMsb( slice_pic_order_cnt_lsb,
prevPicOrderCntLsb, prevPicOrderCntMsb, MaxPicOrderCntLsb)
    }
    PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
}

```

- El valor de PrevPicOrderCnt[nuh\_layer\_id] se obtiene de la forma siguiente:

- Si la imagen actual no es una imagen RASL, una imagen RADL o una imagen de subcapa que no es de referencia, y la imagen actual tiene TemporalId igual a 0 y discardable\_flag igual a 0, PrevPicOrderCnt[nuh\_layer\_id] se establece igual a PicOrderCntVal.

- De lo contrario, cuando FirstPicInLayerDecodedFlag[nuh\_layer\_id] es igual a 0 y poc\_reset\_idc es igual a 3, PrevPicOrderCnt[nuh\_layer\_id] se establece igual a full\_poc\_reset\_flag ? 0: poc\_lsb\_val.

**[0265]** El valor de PicOrderCntVal estará en el intervalo de  $-2^{31}$  a  $2^{31}-1$ , ambos inclusive. En una CVS, los valores de PicOrderCntVal para dos imágenes codificadas cualesquiera en la misma capa no serán iguales.

**[0266]** La función PicOrderCnt(picX) se especifica de la siguiente manera: PicOrderCnt(picX) = PicOrderCntVal de la imagen picX.

**[0267]** La función DiffPicOrderCnt(picA, picB) se especifica de la siguiente manera: DiffPicOrderCnt(picA, picB) = PicOrderCnt(picA) - PicOrderCnt(picB).

**[0268]** El flujo de bits no debe contener datos que den como resultado valores de DiffPicOrderCnt(picA, picB) usados en el proceso de descodificación que no estén en el intervalo de  $-2^{15}$  to  $2^{15} - 1$ , ambos inclusive.

NOTA - Sea X la imagen actual e Y y Z otras dos imágenes en la misma secuencia, se considera que Y y Z están en la misma dirección de orden de salida desde X cuando tanto DiffPicOrderCnt(X, Y) como DiffPicOrderCnt(X, Z) son positivos o ambos negativos.

#### **Funcionamiento del DPB de orden de salida**

##### **C.5.2.2 Salida y eliminación de imágenes del DPB**

**[0269]** Cuando la imagen actual no es la imagen 0 en la capa actual, la salida y la eliminación de imágenes en la capa actual de la DPB antes de la descodificación de la imagen actual, es decir, la imagen n, pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual y antes de la invocación del proceso de descodificación para el recuento de orden de imágenes, se producen inmediatamente cuando la primera unidad de descodificación de la imagen actual se elimina del CPB y se procede de la siguiente manera:

- Cuando poc\_reset\_idc de la imagen actual es mayor que 0 y la unidad de acceso actual es la primera unidad de acceso en un período de restablecimiento de POC, se proporcionan todas las imágenes del DPB que no pertenecen a la unidad de acceso actual y que están marcadas como "necesarias para transmitir", comenzando con imágenes con el valor más pequeño de PicOrderCntVal de todas las imágenes, excluidas las de la unidad de acceso actual en el DPB, en orden ascendente de los valores de PicOrderCntVal, y las imágenes con el mismo valor de PicOrderCntVal se proporcionan en orden ascendente de valores de identificación de nuh\_layer\_id. Cuando la imagen se proporciona, se recorta usando la ventana de recorte de conformidad especificada en el SPS activo para la imagen, la imagen recortada se proporciona, y la imagen se marca como "no necesaria para transmitir".

- Se invocan los procesos de descodificación para el recuento de orden de imágenes y el RPS. Cuando se descodifica una CVS conforme a uno o más de los perfiles especificados en el anexo A usando el proceso de descodificación especificado en las cláusulas 2 a 10, los procesos de descodificación para el recuento de orden de imágenes y el RPS que se invocan son los especificados en las subcláusulas 8.3.1 y 8.3. 2, respectivamente. Cuando se descodifica una CVS conforme a uno o más de los perfiles especificados en el Anexo G o H usando el proceso de descodificación especificado en el Anexo F y el Anexo G o H, los procesos de descodificación para el recuento de orden de imágenes y el RPS que se invocan son los especificados en las subcláusulas F.8.3.1 y F.8.3.2, respectivamente.

- Si la imagen actual es una imagen IRAP con NoRaslOutputFlag igual a 1, o la imagen de capa base en la unidad de acceso actual es una imagen IRAP con NoRaslOutputFlag igual a 1 y NoCilasOutputFlag igual a 1, se aplican las siguientes etapas ordenadas:

1. La variable NoOutputOfPriorPicsFlag se obtiene para el decodificador bajo prueba como sigue:

- Si la imagen actual es una imagen CRA con NoRaslOutputFlag igual a 1, NoOutputOfPriorPicsFlag se establece igual a 1 (independientemente del valor de no\_output\_of\_prior\_pics\_flag).

- De lo contrario, si la imagen actual es una imagen IRAP con NoRaslOutputFlag igual a 1 y el valor de pic\_width\_in\_luma\_samples, pic\_height\_in\_luma\_samples o

sps\_max\_dec\_pic\_buffering\_minus1[HighestTid] obtenidos del SPS activo para la capa actual es diferente del valor de pic\_width\_in\_luma\_samples, pic\_height\_in\_luma\_samples o sps\_max\_dec\_pic\_buffering\_minus1[HighestTid], respectivamente, obtenidos del SPS que estaba activo para la capa actual cuando se descodifica la imagen anterior en la capa actual, NoOutputOfPriorPicsFlag puede (pero no debe) establecerse en 1 mediante el descodificador bajo prueba, independientemente del valor de no\_output\_of\_prior\_pics\_flag. NOTA: aunque en estas condiciones se prefiere establecer NoOutputOfPriorPicsFlag igual a no\_output\_of\_prior\_pics\_flag, el descodificador bajo prueba puede establecer NoOutputOfPriorPicsFlag en 1 en este caso.

- De lo contrario, si la imagen actual es una imagen IRAP con NoRaslOutputFlag igual a 1, NoOutputOfPriorPicsFlag se establece igual a no\_output\_of\_prior\_pics\_flag.

- De lo contrario (la imagen actual no es una imagen IRAP con NoRaslOutputFlag igual a 1, la imagen de capa base en la unidad de acceso actual es una imagen IRAP con NoRaslOutputFlag igual a 1 y NoClrasOutputFlag es igual a 1), NoOutputOfPriorPicsFlag se establece igual a 1.

2. El valor de NoOutputOfPriorPicsFlag obtenido para el descodificador bajo prueba se aplica para HRD como sigue:

- Si NoOutputOfPriorPicsFlag es igual a 1, todos los búfers de almacenamiento de imágenes en el sub-DPB se vacían sin proporcionar las imágenes que contienen, y la completitud de sub-DPB se establece igual a 0.

- De lo contrario (NoOutputOfPriorPicsFlag es igual a 0), todos los búfers de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para su provisión" y "no usada como referencia" se vacían (sin proporcionarse sus datos), y todos los búfers de almacenamiento de imágenes no vacíos en el sub-DPB se vacían invocando repetidamente el proceso de "agitación" especificado en la subcláusula C.5.2.4, y la completitud de sub-DPB se establece igual a 0.

- De lo contrario, todos los búfers de almacenamiento de imágenes que contienen una imagen en la capa actual y que está marcada como "no necesaria para su provisión" y "no usada como referencia" se vacían (sin proporcionarse sus datos). Para cada búfer de almacenamiento de imágenes que se vacía, la completitud de sub-DPB se decrementa en uno. Cuando una o más de las siguientes condiciones son verdaderas, el proceso de "agitación" especificado en la subcláusula C.5.2.4 se invoca repetidamente (\*\* ELIMINADO :mientras se decrementa adicionalmente la completitud de sub-DPB en uno para cada búfer de almacenamiento de imágenes adicional que se vacía \*\*), hasta que ninguna de las siguientes condiciones sea verdadera:

- El número de unidades de acceso que contienen al menos una imagen descodificada en el DPB marcada como "necesaria para su provisión" es mayor que MaxNumReorderPics.

- MaxLatencyIncreasePlus1 no es igual a 0 y hay al menos una unidad de acceso que contiene al menos una imagen descodificada en el DPB marcada como "necesaria para su provisión" para la cual la variable asociada PicLatencyCount es mayor que o igual que MaxLatencyPictures.

- El número de imágenes en la capa actual en el sub-DPB es mayor que o igual a MaxDecPicBufferingMinus1 + 1.

### C.5.2.3 Descodificación, marcado, agitación adicional y almacenamiento de imágenes

**[0270]** Los procesos especificados en esta subcláusula se producen instantáneamente cuando la última unidad de descodificación de la imagen n se elimina del CPB.

**[0271]** PicOutputFlag se actualiza de la siguiente manera:

- Si la unidad de acceso actual no contiene una imagen en una capa de salida de destino y alt\_output\_layer\_flag es igual a 1, se aplican las siguientes etapas ordenadas:

- La lista nonOutputLayerPictures es la lista de imágenes de la unidad de acceso con PicOutputFlag igual a 1 y con valores de nuh\_layer\_id que están incluidos en TargetDecLayerIdList y que no están en las capas de salida de destino.

- La imagen con el valor de nuh\_layer\_id más alto entre la lista nonOutputLayerPictures se elimina de la lista nonOutputLayerPictures.

- PicOutputFlag para cada imagen que está incluida en la lista nonOutputLayerPictures se establece igual a 0.

- De lo contrario, PicOutputFlag para imágenes que no están incluidas en una capa de salida de destino se establece igual a 0.

**[0272]** Cuando la imagen actual tiene PicOutputFlag igual a 1, para cada imagen de la capa actual en el sub-DPB que está marcada como "necesaria para su provisión" y sigue a la imagen actual en orden de salida, la variable asociada PicLatencyCount se establece igual a PicLatencyCount + 1.

**[0273]** La imagen actual se considera descodificada después de que se descodifique la última unidad de descodificación de la imagen. La imagen descodificada actual se almacena en un búfer de almacenamiento de imágenes vacío en el sub-DPB, y se aplica lo siguiente:

- Si la imagen descodificada actual tiene PicOutputFlag igual a 1, se marca como "necesaria para su provisión" y su variable asociada PicLatencyCount se establece igual a 0.

- De lo contrario, (la imagen descodificada actual tiene PicOutputFlag igual a 0), se marca como "no necesaria para transmitirse".

**[0274]** La imagen descodificada actual se marca como "usada para referencia a corto plazo".

**[0275]** Cuando una o más de las siguientes condiciones son verdaderas, el proceso de "agitación" especificado en la subcláusula C.5.2.4 se invoca repetidamente hasta que ninguna de las siguientes condiciones sea verdadera:

- El número de unidades de acceso que contienen al menos una imagen descodificada en el DPB marcada como "necesaria para su provisión" es mayor que MaxNumReorderPics.

- MaxLatencyIncreasePlus1 no es igual a 0 y hay al menos una unidad de acceso que contiene al menos una imagen descodificada en el DPB marcada como "necesaria para su provisión" para la cual la variable asociada PicLatencyCount es mayor que o igual que MaxLatencyPictures.

#### **C.5.2.4 Proceso de "agitación"**

**[0276]** El proceso de "agitación" puede consistir en las siguientes etapas ordenadas:

1. La imagen o imágenes que están primero para su provisión se seleccionan como las únicas que tienen el valor más pequeño de PicOrderCntVal de todas las imágenes en el DPB marcadas como "necesaria para su provisión".

2. Cada una de estas imágenes, en orden de nuh\_layer\_id ascendente, se recorta, usando la ventana de recorte de conformidad especificada en el SPS activo para la imagen, se proporciona la imagen recortada, y la imagen se marca como "no necesaria para su provisión".

3. Cada búfer de almacenamiento de imágenes que contiene una imagen marcada como "no usada como referencia" y que era una de las imágenes recortadas y proporcionadas se vacía, y la completitud del sub-DPB asociado se reduce en uno.

#### **Implementación de ejemplo n.º 2**

**[0277]** La implementación de ejemplo proporcionada a continuación es similar a la implementación de ejemplo n.º 1 descrita anteriormente, sin restablecimiento de POC y, además, incluye un POC a nivel de capa definido para todas las imágenes. Se pueden aplicar todas las técnicas sugeridas en el ejemplo de implementación n.º 1 anterior, excepto las que se reproducen a continuación. Los cambios propuestos en esta implementación de ejemplo están resaltados en *cursiva*.

#### **Cambios en la sintaxis y la semántica de la cabecera de segmento de fragmento**

**[0278]**

**Tabla 4. Sintaxis de ejemplo de slice\_segment\_header()**

slice_segment_header() {	<b>Descriptor</b>
<b>first_slice_segment_in_pic_flag</b>	u(1)

if(nal_unit_type >= BLA_W_LP && nal_unit_type <= RSV_IRAP_VCL23)	
<b>no_output_of_prior_pics_flag</b>	u(1)
<b>slice_pic_parameter_set_id</b>	ue(v)
if( !first_slice_segment_in_pic_flag) {	
if(dependent_slice_segments_enabled_flag)	
<b>dependent_slice_segment_flag</b>	u(1)
<b>slice_segment_address</b>	u(v)
}	
if(!dependent_slice_segment_flag) {	
i = 0	
if(num_extra_slice_header_bits > i) {	
i++	
<b>discardable_flag</b>	u(1)
}	
if(num_extra_slice_header_bits > i) {	
i++	
<b>cross_layer_bla_flag</b>	u(1)
}	
for(; i < num_extra_slice_header_bits; i++)	
<b>slice_reserved_flag[i]</b>	u(1)
<b>slice_type</b>	ue(v)
if(output_flag_present_flag)	
<b>pic_output_flag</b>	u(1)
if(separate_colour_plane_flag == 1)	
<b>colour_plane_id</b>	u(2)
if( (nuh_layer_id > 0 && !poc_lsb_not_present_flag[ LayerIdxInVPS[nuh_layer_id] ] )	
(nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP))	
<b>slice_pic_order_cnt_lsb</b>	u(v)
if(nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) {	
<b>short_term_ref_pic_set_sps_flag</b>	u(1)
if(!short_term_ref_pic_set_sps_flag)	
short_term_ref_pic_set(num_short_term_ref_pic_sets)	
else if( num_short_term_ref_pic_sets > 1)	
<b>short_term_ref_pic_set_idx</b>	u(v)
if( long_term_ref_pics_present_flag) {	
if(num_long_term_ref_pics_sps > 0)	
<b>num_long_term_sps</b>	ue(v)
<b>num_long_term_pics</b>	ue(v)
for( i = 0; i < num_long_term_sps + num_long_term_pics; i++) {	
if(i < num_long_term_sps) {	
if(num_long_term_ref_pics_sps > 1)	
<b>lt_idx_sps[i]</b>	u(v)
} else {	
<b>poc_lsb_lt[i]</b>	u(v)

<b>usado por_curr_pic_lt_flag[i]</b>	u(1)
}	
<b>delta_poc_msb_present_flag[i]</b>	u(1)
if( delta_poc_msb_present_flag[i] )	
<b>delta_poc_msb_cycle_lt[i]</b>	ue(v)
}	
}	
if( sps_temporal_mvp_enabled_flag )	
<b>slice_temporal_mvp_enabled_flag</b>	u(1)
}	
if( nuh_layer_id > 0 && ! all_ref_layers_active_flag && NumDirectRefLayers[ nuh_layer_id ] > 0 )	
{	
<b>inter_layer_pred_enabled_flag</b>	u(1)
if( inter_layer_pred_enabled_flag && NumDirectRefLayers[ nuh_layer_id ] > 1 ) {	
if( ! max_one_active_ref_layer_flag )	
<b>num_inter_layer_ref_pics_minus1</b>	u(v)
if( NumActiveRefLayerPics != NumDirectRefLayers[ nuh_layer_id ] )	
for( i = 0; i < NumActiveRefLayerPics; i++ )	
<b>inter_layer_pred_layer_idc[i]</b>	u(v)
}	
}	
if( sample_adaptive_offset_enabled_flag ) {	
<b>slice_sao_luma_flag</b>	u(1)
<b>slice_sao_chroma_flag</b>	u(1)
}	
if( slice_type == P    slice_type == B ) {	
<b>num_ref_idx_active_override_flag</b>	u(1)
if( num_ref_idx_active_override_flag ) {	
<b>num_ref_idx_10_active_minus1</b>	ue(v)
if( slice_type == B )	
<b>num_ref_idx_l1_active_minus1</b>	ue(v)
}	
if( lists_modification_present_flag && NumPicTotalCurr > 1 )	
ref_pic_lists_modification()	
if( slice_type == B )	
<b>mvd_l1_zero_flag</b>	u(l)
if( cabac_init_present_flag )	
<b>cabac_init_flag</b>	u(l)
if( slice_temporal_mvp_enabled_flag ) {	
if( slice_type == B )	
<b>collocated_from_10_flag</b>	u(1)
if( ( collocated_from_10_flag && num_ref_idx_10_active_minus1 > 0 )    ( ! collocated_from_10_flag && num_ref_idx_l1_active_minus1 > 0 ) )	
<b>collocated_ref_idx</b>	ue(v)

}	
if((weighted_pred_flag && slice_type == P)	
(weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
<b>five_minus_max_num_merge_cand</b>	ue(v)
}	
<b>slice_qp_delta</b>	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
<b>slice_cb_qp_offset</b>	se(v)
<b>slice_cr_qp_offset</b>	se(v)
}	
if( deblocking_filter_override_enabled_flag)	
<b>deblocking_filter_override_flag</b>	u(1)
if( deblocking_filter_override_flag) {	
<b>slice_deblocking_filter_disabled_flag</b>	u(1)
if( !slice_deblocking_filter_disabled_flag) {	
<b>slice_beta_offset_div2</b>	se(v)
<b>slice_tc_offset_div2</b>	se(v)
}	
}	
if( pps_loop_filter_across_slices_enabled_flag && (slice_sao_luma_flag    slice_sao_chroma_flag    !slice_deblocking_filter_disabled_flag))	
<b>slice_loop_filter_across_slices_enabled_flag</b>	u(1)
}	
if( tiles_enabled_flag    entropy_coding_sync_enabled_flag) {	
<b>num_entry_point_offsets</b>	ue(v)
if(num_entry_point_offsets > 0) {	
<b>offset_len_minus1</b>	ue(v)
for(i = 0; i < num_entry_point_offsets; i++ )	
<b>entry_point_offset_minus1[i]</b>	u(v)
}	
}	
if( slice_segment_header_extension_present_flag) {	
<b>slice_segment_header_extension_length</b>	ue(v)
if( poc_reset_info_present_flag)	
<b>poc_reiniciar_idc</b>	u(2)
if(poc_reset_idc != 0) {	
<b>poc_reset_period_id</b>	u(6)
<b>poc_lsb_val</b>	u(v)
}	
if(poc_reset_idc == 3)	
<b>full_poc_reset_flag</b>	u(1)
if( CraOrBlaPicFlag)	
<b>poc_msb_val</b>	ue(v)



while(more_data_in_slice_segment_header_extension())	
<b>slice_segment_header_extension_data_bit</b>	u(1)
}	
byte_alignment( )	
}	

**[0279]** **poc\_lsb\_val** especifica un valor que puede usarse para obtener el recuento de orden de imágenes o el recuento de orden de imágenes a nivel de capa de la imagen actual. La longitud del elemento de sintaxis poc\_lsb\_val es de  $\log_2 \text{max\_pic\_order\_cnt\_lsb\_minus4} + 4$  bits.

**[0280]** Es un requisito de conformidad de flujo de bits que, cuando poc\_reset\_idc es igual a 3, y la imagen anterior picA en orden de decodificación que está en la misma capa que la imagen actual, que tiene poc\_reset\_idc igual a 1 o 2 y pertenece al mismo período de restablecimiento de POC, está presente en el flujo de bits, picA será la misma imagen que la imagen anterior en orden de decodificación que está en la misma capa que la imagen actual, que no es una imagen RASL, una imagen RADL o una imagen de subcapa que no es de referencia, y que tiene TemporalId igual a 0 y discardable\_flag igual a 0, y el valor de poc\_lsb\_val de la imagen actual será igual al valor de slice\_pic\_order\_cnt\_lsb de picA.

#### **C.1.1.1 Cambios en el proceso de decodificación del recuento de orden de imágenes. Proceso de decodificación del recuento de orden de imágenes**

**[0281]** El resultado de este proceso es PicOrderCntVal, el recuento de orden de imágenes de la imagen actual, y LayerWisePicOrderCntVal, el recuento de orden de imágenes a nivel de capa de la imagen actual.

**[0282]** Los recuentos de orden de imágenes se usan para identificar imágenes, para obtener parámetros de movimiento en modo de combinación y predicción de vectores de movimiento, y para la verificación de conformidad de decodificador (véase la subcláusula C.5).

**[0283]** Cada imagen codificada está asociada a una variable de recuento de orden de imágenes, denotada como PicOrderCntVal, y una variable de recuento de orden de imágenes a nivel de capa, denotada como LayerWisePicOrderCntVal.

**[0284]** Si FirstPicInLayerDecodedFlag[nuh\_layer\_id] es igual a 1 y la imagen actual es una imagen de restablecimiento de POC, se aplica lo siguiente:

- Las variables pocLsbVal, lwPocMsb y lwPocLsb se obtienen de la siguiente manera:

```

if(CraOrBlaPicFlag)
    prevLwPicOrderCntLsb = PrevLwPicOrderCntReset
    [ nuh_layer_id ] & ( MaxPicOrderCntLsb - 1 )
    prevLwPicOrderCntMsb = PrevLwPicOrderCntReset
    [ nuh_layer_id ] - prevPicOrderCntLsb
    lwPocMsb = prevLwPicOrderCntMsb + ( poc_msb_val *
MaxPicOrderCntLsb )
else {
    prevLwPicOrderCntLsb =

```

```

PrevLwPicOrderCnt[ nuh_layer_id ] & ( MaxPicOrderCntLsb – 1 )
    prevLwPicOrderCntMsb =
PrevLwPicOrderCnt[ nuh_layer_id ] – prevPicOrderCntLsb
    lwpocMsbDelta = getCurrMsb( poc_lsb_val, prevLwPicOrderCntLsb,
prevPicOrderCntMsb, MaxPicOrderCntLsb)
}
LayerWisePicOrderCntVal = lwPocLsb + poc_lsb_val

```

- El PicOrderCntVal de cada imagen que está en el DPB y pertenece a la misma capa que la imagen actual se decrementa en DeltaPocVal.
- El PicOrderCntVal de la imagen actual se obtiene de la forma siguiente:

```

if( poc_reset_idc == 1 )
    PicOrderCntVal = slice_pic_order_cnt_lsb
else if( poc_reset_idc == 2 )
    PicOrderCntVal = 0
else { // poc_reset_idc == 3
    PicOrderCntMsb = getCurrMsb( slice_pic_order_cnt_lsb,
full_poc_reset_flag ? 0 : poc_lsb_val, 0, MaxPicOrderCntLsb )
    PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
    LayerWisePicOrderCntVal = LayerWisePicOrderCntVal +
PicOrderCntVal
}

```

- El valor de PrevPicOrderCnt[nuh\_layer\_id] se obtiene de la forma siguiente:

5 - Si la imagen actual no es una imagen RASL, una imagen RADL o una imagen de subcapa que no es de referencia, y la imagen actual tiene TemporalId igual a 0 y discardable\_flag igual a 0, PrevPicOrderCnt[nuh\_layer\_id] se establece igual a PicOrderCntVal.

10 - De lo contrario, cuando poc\_reset\_idc es igual a 3, PrevPicOrderCnt[nuh\_layer\_id] se establece igual a full\_poc\_reset\_flag ? 0: poc\_lsb\_val.

- El valor de PrevLwPicOrderCnt[nuh\_layer\_id] y de PrevLwPicOrderCntReset [nuh\_layer\_id] se obtienen de la siguiente manera:

15 - Si la imagen actual no es una imagen RASL, una imagen RADL o una imagen de subcapa que no es de referencia, y la imagen actual tiene TemporalId igual a 0 y discardable\_flag igual a 0, PrevLwPicOrderCnt[nuh\_layer\_id] se establece igual a

20 LayerWisePicOrderCntVal, y PrevLwPicOrderCntReset[nuh\_layer\_id] se establece igual a LayerWisePicOrderCntVal.

**[0285]** De lo contrario, se aplica lo siguiente:

25 - El PicOrderCntVal de la imagen actual se obtiene de la forma siguiente:

```

if( !FirstPicInLayerDecodedFlag[ nuh_layer_id ] ) {
    if( poc_reset_idc == 1 )
        PicOrderCntVal = slice_pic_order_cnt_lsb
    else if( poc_reset_idc == 2 )
        PicOrderCntVal = 0
    else if( poc_reset_idc == 3 ) {
        PicOrderCntMsb = getCurrMsb(slice_pic_order_cnt_lsb,
full_poc_reset_flag ? 0 : poc_lsb_val,
                                0, MaxPicOrderCntLsb )
        PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
    } else {
        if( vps_poc_msb_present_flag && CraOrBlaPicFlag ) {
            PicOrderCntMsb = poc_msb_val * MaxPicOrderCntLsb
            PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
        }
        else
            PicOrderCntVal = slice_pic_order_cnt_lsb
    }
} else {
    if( vps_poc_msb_present_flag && CraOrBlaPicFlag )
        PicOrderCntMsb = poc_msb_val * MaxPicOrderCntLsb
    else if( la_imagen_actual_es_una_imagen_IRAP_con_NoRaslOutputFlag_igual_to_1 )
        PicOrderCntMsb = 0
    else {
        prevPicOrderCntLsb =
PrevPicOrderCnt[ nuh_layer_id ] & ( MaxPicOrderCntLsb - 1 ).
        prevPicOrderCntMsb =
PrevPicOrderCnt[ nuh_layer_id ] - prevPicOrderCntLsb
        PicOrderCntMsb = getCurrMsb( slice_pic_order_cnt_lsb,

```

prevPicOrderCntLsb, prevPicOrderCntMsb, MaxPicOrderCntLsb )

}

PicOrderCntVal = PicOrderCntMsb + slice\_pic\_order\_cnt\_lsb

}

- El *LayerWisePicOrderCntVal* de la imagen actual se obtiene de la forma siguiente:

*if( !FirstPicInLayerDecodedFlag[ nuh\_layer\_id ] )*

*LayerWisePicOrderCntVal = PicOrderCntVal*

*else {*

*if (la imagen actual es una imagen CRA o una imagen BLA)*

*LayerWisePicOrderCntVal = PrevLwPicOrderCntReset[ nuh\_layer\_id ]*

*+ PicOrderCntVal*

*else if (la imagen actual es una IDR)*

*LayerWisePicOrderCntVal = PrevLwPicOrderCntReset[ nuh\_layer\_id ]*

*+ (poc\_msb\_val \* MaxPicOrderCntLsb )*

*else*

*LayerWisePicOrderCntVal = PrevLwPicOrderCnt[ nuh\_layer\_id ] + (*

*PicOrderCntVal +*

*PrevPicOrderCnt[ nuh\_layer\_id ] )*

*}*

- El valor de *PrevPicOrderCnt[nuh\_layer\_id]* se obtiene de la forma siguiente:

- 5 - Si la imagen actual no es una imagen RASL, una imagen RADL o una imagen de subcapa que no es de referencia, y la imagen actual tiene *TemporalId* igual a 0 y *discardable\_flag* igual a 0, *PrevPicOrderCnt[nuh\_layer\_id]* se establece igual a *PicOrderCntVal*.

- 10 - De lo contrario, cuando *FirstPicInLayerDecodedFlag[nuh\_layer\_id]* es igual a 0 y *poc\_reset\_idc* es igual a 3, *PrevPicOrderCnt[nuh\_layer\_id]* se establece igual a *full\_poc\_reset\_flag ? 0: poc\_lsb\_val*.

- El valor de *PrevLwPicOrderCnt[nuh\_layer\_id]* se obtiene de la forma siguiente:

- 15 - Si la imagen actual no es una imagen RASL, una imagen RADL o una imagen de subcapa que no es de referencia, y la imagen actual tiene *TemporalId* igual a 0 y *discardable\_flag* igual a 0, *PrevLwPicOrderCnt[nuh\_layer\_id]* se establece igual a *LayerWisePicOrderCntVal*.

[0286] El valor de *PicOrderCntVal* estará en el intervalo de  $-2^{31}$  a  $2^{31}-1$ , ambos inclusive. En una CVS, los valores de *PicOrderCntVal* para dos imágenes codificadas cualesquiera en la misma capa no serán iguales.

- 20 [0287] El valor de *LayerWisePicOrderCntVal* estará en el intervalo de  $-2^{31}$  a  $2^{31} - 1$ , ambos inclusive. Los valores de *LayerWisePicOrderCntVal* para dos imágenes codificadas cualesquiera no serán los mismos dentro de una secuencia de imágenes con el mismo valor *nuh\_layer\_id*, en orden de descodificación, comenzando por una unidad de acceso que contenga una imagen IRAP con *NoCllasOutputFlag* igual a 1, inclusive, hasta el final del flujo de bits o hasta la siguiente una unidad de acceso que contenga una imagen IRAP con *NoCllasOutputFlag* igual a 1, exclusive, lo que suceda primero en orden de descodificación.

- 25 [0288] La función *LayerWisePicOrderCnt(picX)* se especifica como sigue: *LayerWisePicOrderCnt(picX) = LayerWisePicOrderCntVal* de la imagen *picX*. La función *DiffPicOrderCnt(picA, picB)* se especifica como sigue:  
30 *DiffPicOrderCnt(picA, picB) = LayerWisePicOrderCnt(picA) - LayerWisePicOrderCnt(picB)*.

**[0289]** El flujo de bits no debe contener datos que den como resultado valores de DiffPicOrderCnt(picA, picB) usados en el proceso de descodificación que no estén en el intervalo de  $-2^{15}$  a  $2^{15}-1$ , ambos inclusive.

- 5      **NOTA** - Sea X la imagen actual e Y y Z otras dos imágenes en la misma secuencia, se considera que Y y Z están en la misma dirección de orden de salida desde X cuando tanto DiffPicOrderCnt(X, Y) como DiffPicOrderCnt(X, Z) son positivos o ambos son negativos.

### **Otras Consideraciones**

- 10      **[0290]** La información y las señales divulgadas en el presente documento se pueden representar usando cualquiera de una variedad de tecnologías y técnicas diferentes. Por ejemplo, los datos, instrucciones, comandos, información, señales, bits, símbolos y fragmentos de información que puedan haberse mencionado a lo largo de la descripción anterior se pueden representar mediante voltajes, corrientes, ondas electromagnéticas, campos o partículas magnéticos, campos o partículas ópticos o cualquier combinación de los mismos.

- 15      **[0291]** Los diversos bloques lógicos, módulos, circuitos y etapas de algoritmo ilustrativos descritos en relación con los modos de realización divulgados en el presente documento se pueden implementar como hardware electrónico, software informático o combinaciones de ambos. Para ilustrar claramente esta intercambiabilidad de hardware y software, anteriormente se han descrito, en general, diversos componentes, bloques, módulos, circuitos y etapas ilustrativos en lo que respecta a su funcionalidad. Que dicha funcionalidad se implemente como hardware o software depende de la aplicación particular y de las restricciones de diseño impuestas al sistema global. Los expertos en la técnica pueden implementar la funcionalidad descrita de formas distintas para cada aplicación particular, pero no debería interpretarse que dichas decisiones de implementación suponen apartarse del alcance de la presente invención.

- 20      **[0292]** Las técnicas descritas en el presente documento se pueden implementar en hardware, software, firmware o en cualquier combinación de los mismos. Dichas técnicas se pueden implementar en cualquiera entre una variedad de dispositivos tales como ordenadores de propósito general, dispositivos manuales de comunicación inalámbrica o dispositivos de circuitos integrados que tienen múltiples usos, incluida su aplicación en dispositivos manuales de comunicación inalámbrica y otros dispositivos. Todos los rasgos característicos descritos como módulos o componentes se pueden implementar juntos en un dispositivo lógico integrado o por separado, como dispositivos lógicos discretos pero interoperables. Si se implementan en software, las técnicas se pueden realizar, al menos en parte, por un medio de almacenamiento de datos legible por ordenador que comprenda código de programa que incluya instrucciones que, cuando se ejecuten, realicen uno o más de los procedimientos descritos anteriormente. El medio de almacenamiento de datos legible por ordenador puede formar parte de un producto de programa informático, que puede incluir materiales de embalaje. El medio legible por ordenador puede comprender memoria o medios de almacenamiento de datos, tales como memoria de acceso aleatorio (RAM), tal como memoria de acceso aleatorio dinámica síncrona (SDRAM), memoria de solo lectura (ROM), memoria de acceso aleatorio no volátil (NVRAM), memoria de solo lectura programable y borrrable eléctricamente (EEPROM), memoria FLASH, medios de almacenamiento de datos magnéticos u ópticos, y similares. De forma adicional o alternativa, las técnicas se pueden realizar, al menos en parte, por un medio de comunicación legible por ordenador que transporta o comunica código de programa en forma de instrucciones o estructuras de datos y al que se puede acceder, leer y/o ejecutar mediante un ordenador, tales como señales u ondas propagadas.

- 45      **[0293]** El código de programa se puede ejecutar por un procesador, que puede incluir 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 en el terreno (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. Un procesador de este tipo puede estar configurado para realizar cualquiera de las técnicas descritas en esta divulgación. Un procesador de propósito general puede ser un microprocesador pero, de forma alternativa, el procesador puede ser cualquier procesador, controlador, microcontrolador o máquina de estados convencional. Un procesador también se puede implementar como una combinación de dispositivos informáticos, por ejemplo, una combinación de un DSP y un microprocesador, una pluralidad de microprocesadores, uno o más microprocesadores junto con un núcleo de DSP o cualquier otra configuración de este tipo. Por consiguiente, el término "procesador", como se usa en el presente documento, se puede referir a cualquier estructura anterior, cualquier combinación de la estructura anterior, o cualquier otra estructura o aparato adecuados 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 software o módulos de hardware dedicados configurados para la codificación y la descodificación, o incorporados en un codificador-descodificador de vídeo combinado (CODEC). Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

- 50      **[0294]** Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización

mediante diferentes unidades de hardware. En cambio, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, incluidos uno o más procesadores como los descritos anteriormente, junto con software y/o firmware adecuados.

5

**[0295]** Se han descrito diversos modos de realización de la invención. La invención se expone en las reivindicaciones adjuntas.

**REIVINDICACIONES**

## 1. Un aparato configurado para decodificar información de vídeo, comprendiendo el aparato:

- 5 una unidad de memoria configurada para almacenar información de vídeo asociada a una primera capa de vídeo que tiene una imagen actual; y
- un procesador en comunicación con la unidad de memoria, estando configurado el procesador para:
- 10 decodificar un valor de ciclo de bit más significativo (MSB) de recuento de orden de imágenes (POC) en un flujo de bits, donde el valor de ciclo de MSB de POC está asociado a la imagen actual e indica una diferencia entre (a) los MSB de un primer POC de una imagen anterior en la primera capa de vídeo que precede a la imagen actual en orden de decodificación y (b) los MSB de un segundo POC de la imagen actual, donde la imagen anterior es una más cercana de dos imágenes con respecto a la imagen actual en orden de decodificación, siendo las dos imágenes (i) una imagen de restablecimiento de POC anterior en la primera capa de vídeo que indica un restablecimiento de POC y (ii) una imagen de actualización de decodificador instantánea (IDR) anterior en la primera capa de vídeo;
- 15 determinar si la imagen actual está asociada a un restablecimiento de POC;
- 20 en base a una determinación de que la imagen actual no está asociada a un restablecimiento de POC, determinar los MSB del segundo POC de la imagen actual en base al valor de ciclo de MSB de POC asociado a la imagen actual; o
- 25 en base a una determinación de que la imagen actual está asociada a un restablecimiento de POC, actualizar los valores de POC de todas las imágenes en un búfer de imágenes decodificadas (DPB) que están en la primera capa de vídeo en base al valor de ciclo de MSB de POC asociado a la imagen actual; y
- 30 decodificar la imagen actual en el flujo de bits.

## 2. Un aparato configurado para codificar información de vídeo, comprendiendo el aparato:

- 35 una unidad de memoria configurada para almacenar información de vídeo asociada a una primera capa de vídeo que tiene una imagen actual; y
- un procesador en comunicación con la unidad de memoria, estando configurado el procesador para:
- 40 determinar un valor de ciclo de bit más significativo (MSB) de recuento de orden de imágenes (POC), donde el valor de ciclo de MSB de POC está asociado a la imagen actual e indica una diferencia entre (a) los MSB de un primer POC de una imagen anterior en la primera capa de vídeo que precede a la imagen actual en orden de decodificación y (b) los MSB de un segundo POC de la imagen actual, donde la imagen anterior es una más cercana de dos imágenes con respecto a la imagen actual en orden de decodificación, siendo las dos imágenes (i) una imagen de restablecimiento de POC anterior en la primera capa de vídeo que indica un restablecimiento de POC y (ii) una imagen de actualización de decodificador instantánea (IDR) anterior en la primera capa de vídeo;
- 45 señalar el valor de ciclo de MSB de POC en asociación con la imagen actual en un flujo de bits;
- 50 determinar si la imagen actual está asociada a un restablecimiento de POC;
- en base a una determinación de que la imagen actual no está asociada a un restablecimiento de POC, determinar los MSB del segundo POC de la imagen actual en base al valor de ciclo de MSB de POC asociado a la imagen actual; o
- 55 en base a una determinación de que la imagen actual está asociada a un restablecimiento de POC, actualizar los valores de POC de todas las imágenes en un búfer de imágenes decodificadas (DPB) que están en la primera capa de vídeo en base al valor de ciclo de MSB de POC asociado a la imagen actual; y
- 60 codificar la imagen actual en el flujo de bits.

## 3. El aparato de la reivindicación 1 o 2, en el que la imagen actual es una imagen de acceso aleatorio limpio (CRA) o una imagen de acceso de enlace roto (BLA).

65

4. El aparato de la reivindicación 1 o 2, en el que un conjunto de parámetros de vídeo asociado a la primera capa de vídeo incluye una bandera o elemento de sintaxis que indica si el valor de ciclo de MSB de POC se señala en el flujo de bits en asociación con la imagen actual.
5. El aparato de la reivindicación 1 o 2, en el que la imagen actual es una imagen de actualización de descodificador instantánea (IDR).
6. El aparato de la reivindicación 1 o 2, donde el aparato comprende un dispositivo seleccionado de un grupo que consiste en uno o más de: un ordenador, un notebook, un ordenador portátil, una tableta electrónica, un descodificador, un equipo telefónico manual, un teléfono inteligente, un panel inteligente, un televisor, una cámara, un dispositivo de visualización, un reproductor de medios digitales, una consola de videojuegos y un ordenador para automóvil.
7. Un procedimiento de descodificación de información de vídeo, comprendiendo el procedimiento:
  - recibir un valor de ciclo de bit más significativo (MSB) de recuento de orden de imágenes (POC) en un flujo de bits, donde el valor de ciclo de MSB de POC está asociado a la imagen actual e indica una diferencia entre (a) los MSB de un primer POC de una imagen anterior en la primera capa de vídeo que precede a la imagen actual en orden de descodificación y (b) los MSB de un segundo POC de la imagen actual, donde la imagen anterior es una más cercana de dos imágenes con respecto a la imagen actual en orden de descodificación, siendo las dos imágenes (i) una imagen de restablecimiento de POC anterior en la primera capa de vídeo que indica un restablecimiento de POC y (ii) una imagen de actualización de descodificador instantánea (IDR) anterior en la primera capa de vídeo;
  - determinar si la imagen actual está asociada a un restablecimiento de POC;
  - determinar, en base a una determinación de que la imagen actual no está asociada a un restablecimiento de POC, los MSB del segundo POC de la imagen actual en base al valor de ciclo de MSB de POC asociado a la imagen actual; o
  - actualizar, en base a una determinación de que la imagen actual está asociada a un restablecimiento de POC, los valores de POC de todas las imágenes en un búfer de imágenes descodificadas (DPB) que están en la primera capa de vídeo en base al valor de ciclo de MSB de POC asociado a la imagen actual; y
  - descodificar la imagen actual en el flujo de bits.
8. Un procedimiento de codificación de información de vídeo, comprendiendo el procedimiento:
  - determinar un valor de ciclo de bit más significativo (MSB) de recuento de orden de imágenes (POC), donde el valor de ciclo de MSB de POC está asociado a la imagen actual e indica una diferencia entre (a) los MSB de un primer POC de una imagen anterior en la primera capa de vídeo que precede a la imagen actual en orden de descodificación y (b) los MSB de un segundo POC de la imagen actual, donde la imagen anterior es una más cercana de dos imágenes con respecto a la imagen actual en orden de descodificación, siendo las dos imágenes (i) una imagen de restablecimiento de POC anterior en la primera capa de vídeo que indica un restablecimiento de POC y (ii) una imagen de actualización de descodificador instantánea (IDR) anterior en la primera capa de vídeo;
  - señalar el valor de ciclo de MSB de POC en asociación con la imagen actual en un flujo de bits;
  - determinar si la imagen actual está asociada a un restablecimiento de POC;
  - determinar, en base a una determinación de que la imagen actual no está asociada a un restablecimiento de POC, los MSB del segundo POC de la imagen actual en base al valor de ciclo de MSB de POC asociado a la imagen actual; o
  - actualizar, en base a una determinación de que la imagen actual está asociada a un restablecimiento de POC, los valores de POC de todas las imágenes en un búfer de imágenes descodificadas (DPB) que están en la primera capa de vídeo en base al valor de ciclo de MSB de POC asociado a la imagen actual; y
  - codificar la imagen actual en el flujo de bits.
9. El procedimiento de la reivindicación 7 u 8, en el que la imagen actual es una imagen de acceso aleatorio limpio (CRA) o una imagen de acceso de enlace roto (BLA).



10. El procedimiento de la reivindicación 7 u 8, en el que un conjunto de parámetros de vídeo asociado a la primera capa de vídeo incluye una bandera o elemento de sintaxis que indica si el valor de ciclo de MSB de POC se señala en el flujo de bits en asociación con la imagen actual.
- 5 11. El procedimiento de la reivindicación 7 u 8, en el que la imagen actual es una imagen de actualización de descodificador instantánea (IDR).
12. Un medio no transitorio legible por ordenador que comprende código que, cuando se ejecuta, hace que un aparato realice el procedimiento de descodificación de cualquiera de las reivindicaciones 7 o 9 a 11.
- 10 13. Un medio no transitorio legible por ordenador que comprende código que, cuando se ejecuta, hace que un aparato realice el procedimiento de codificación de cualquiera de las reivindicaciones 8 a 11.

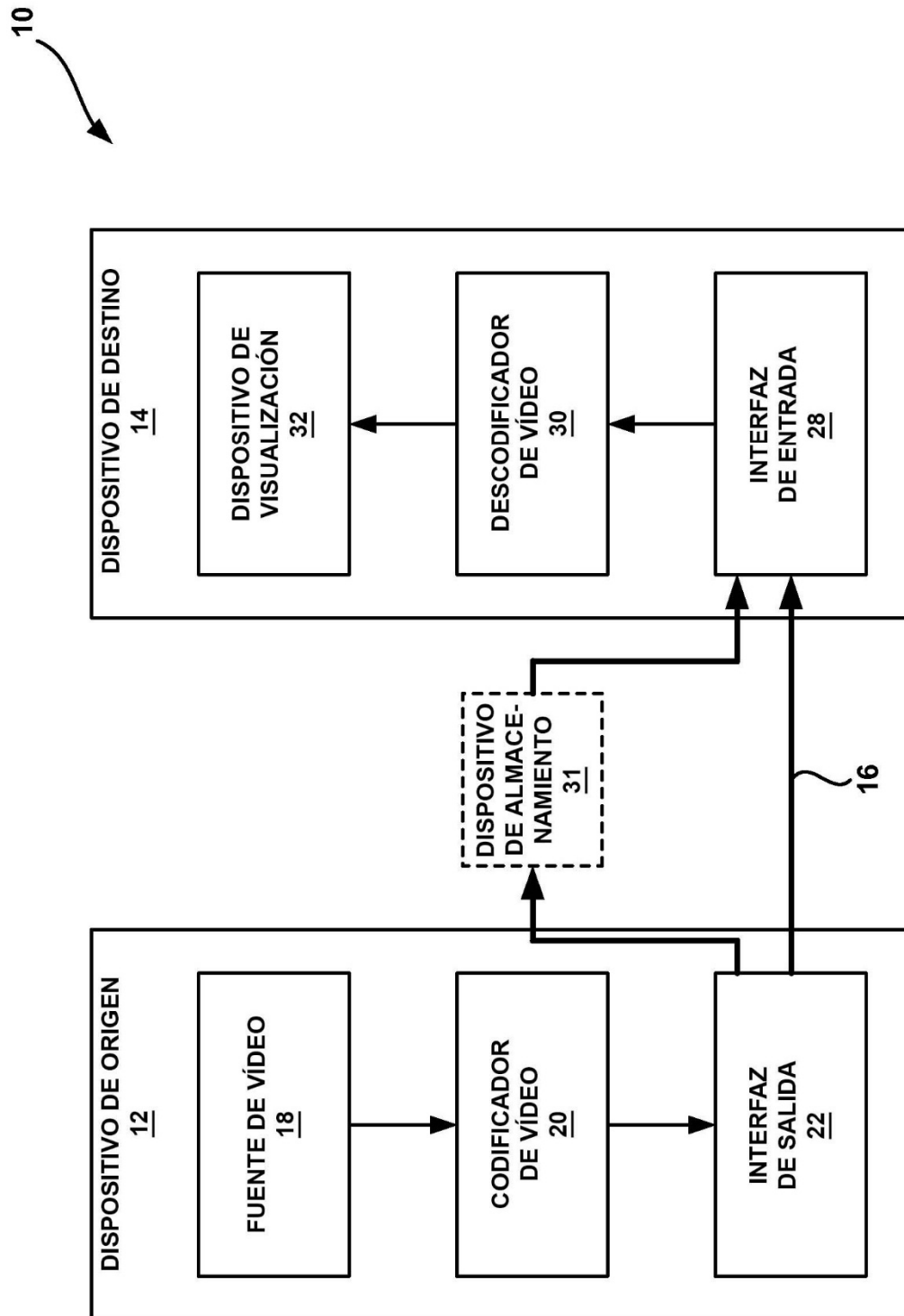


FIG. 1A

10'

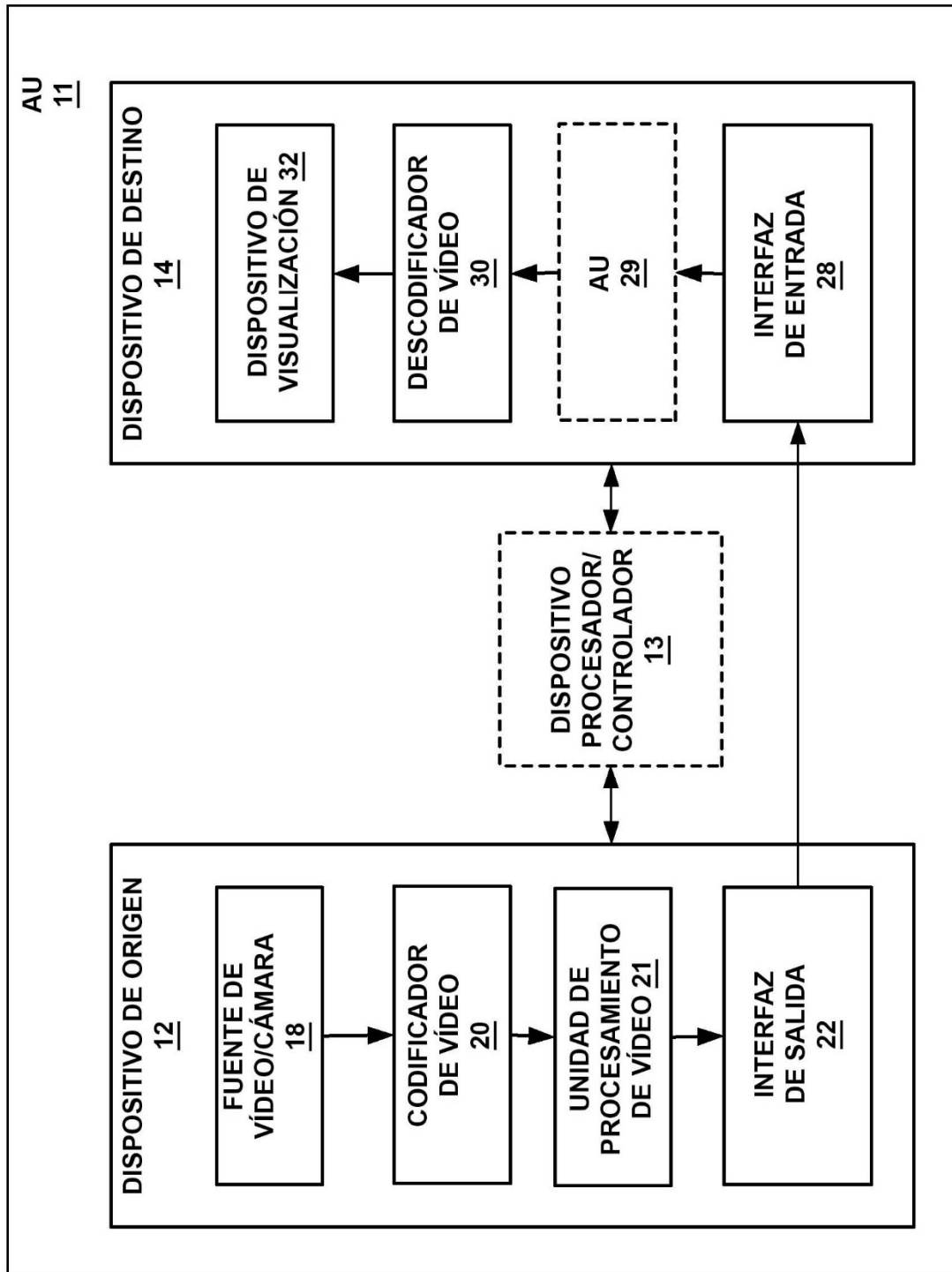


FIG. 1B

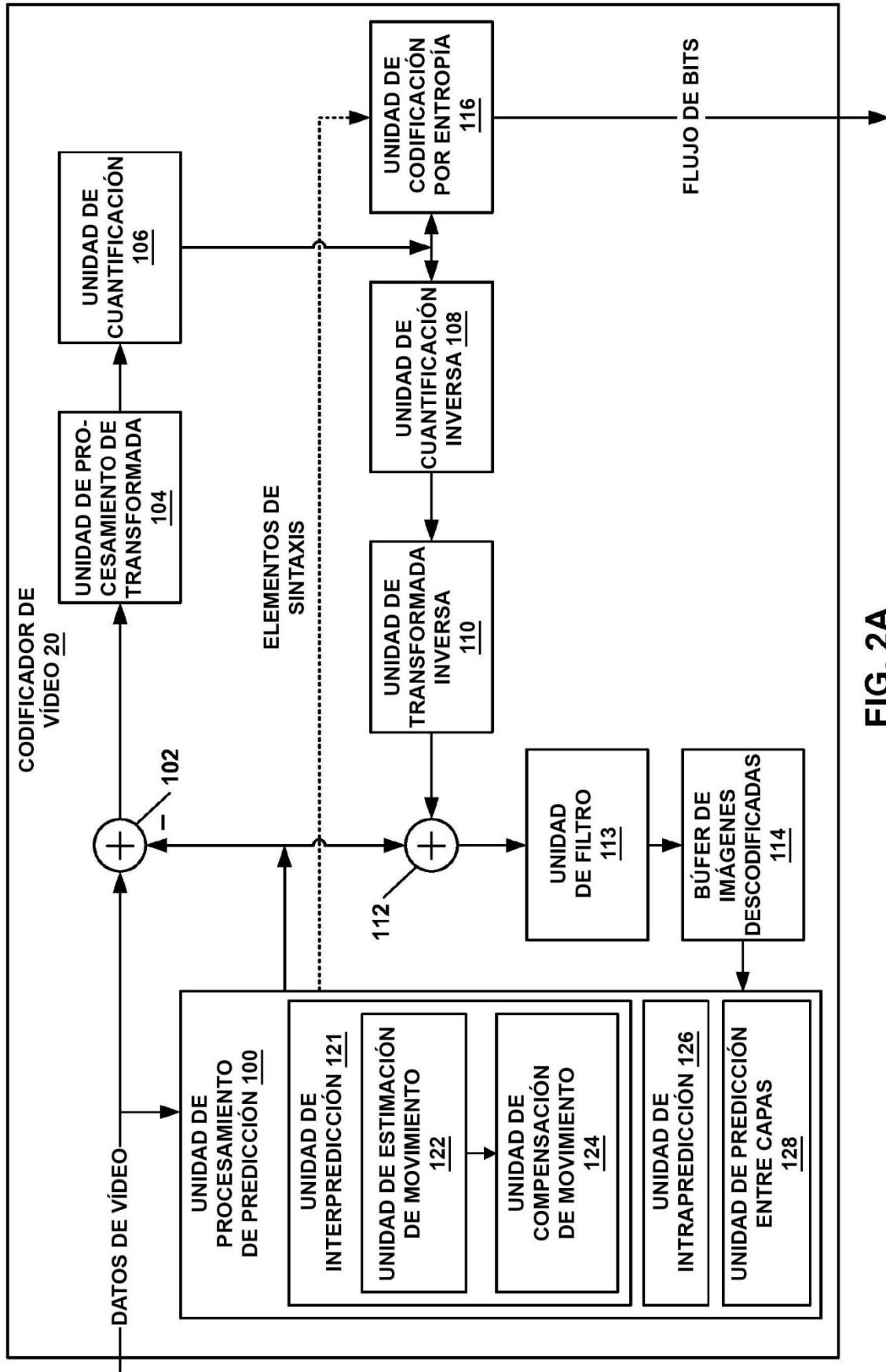


FIG. 2A

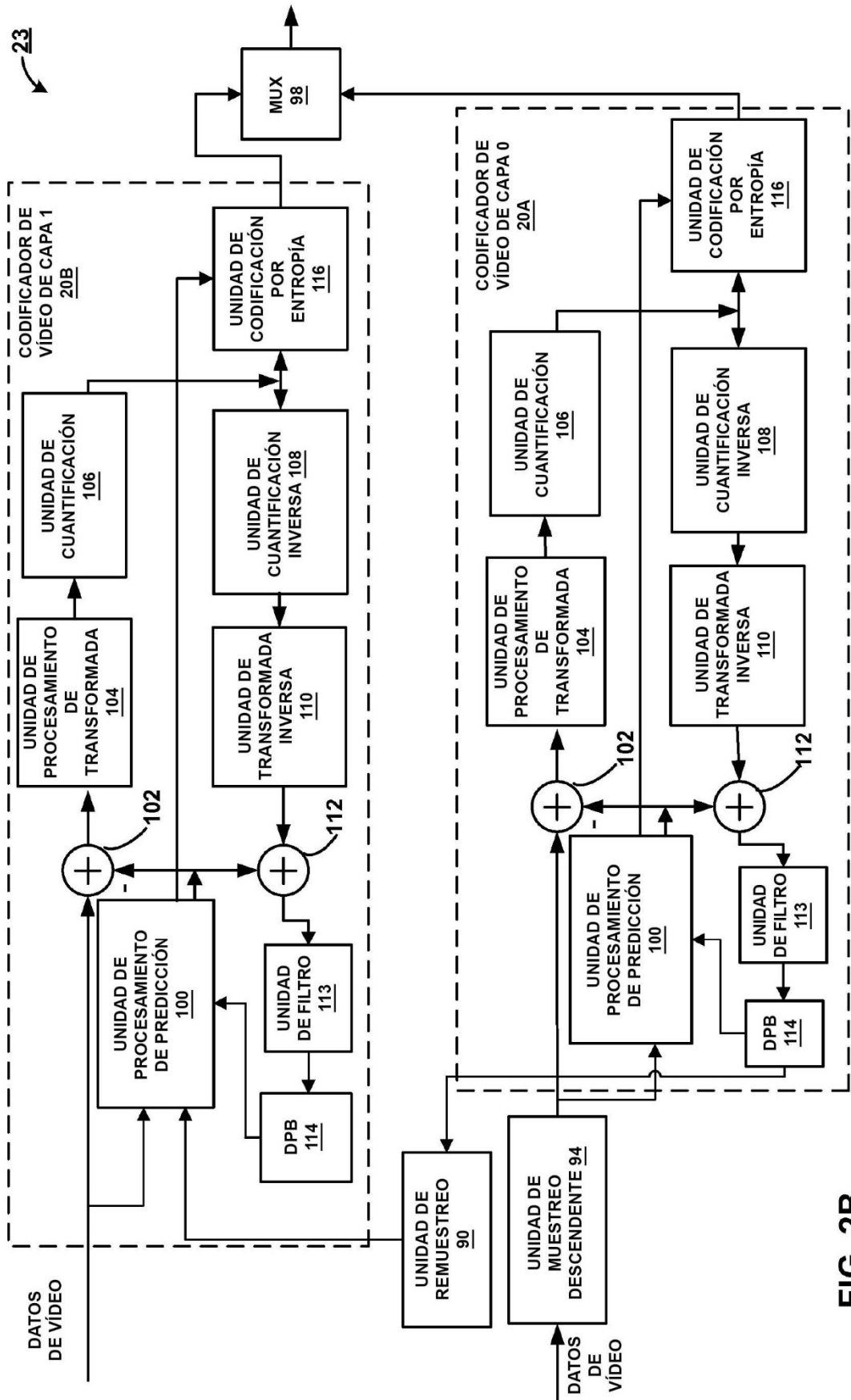


FIG. 2B

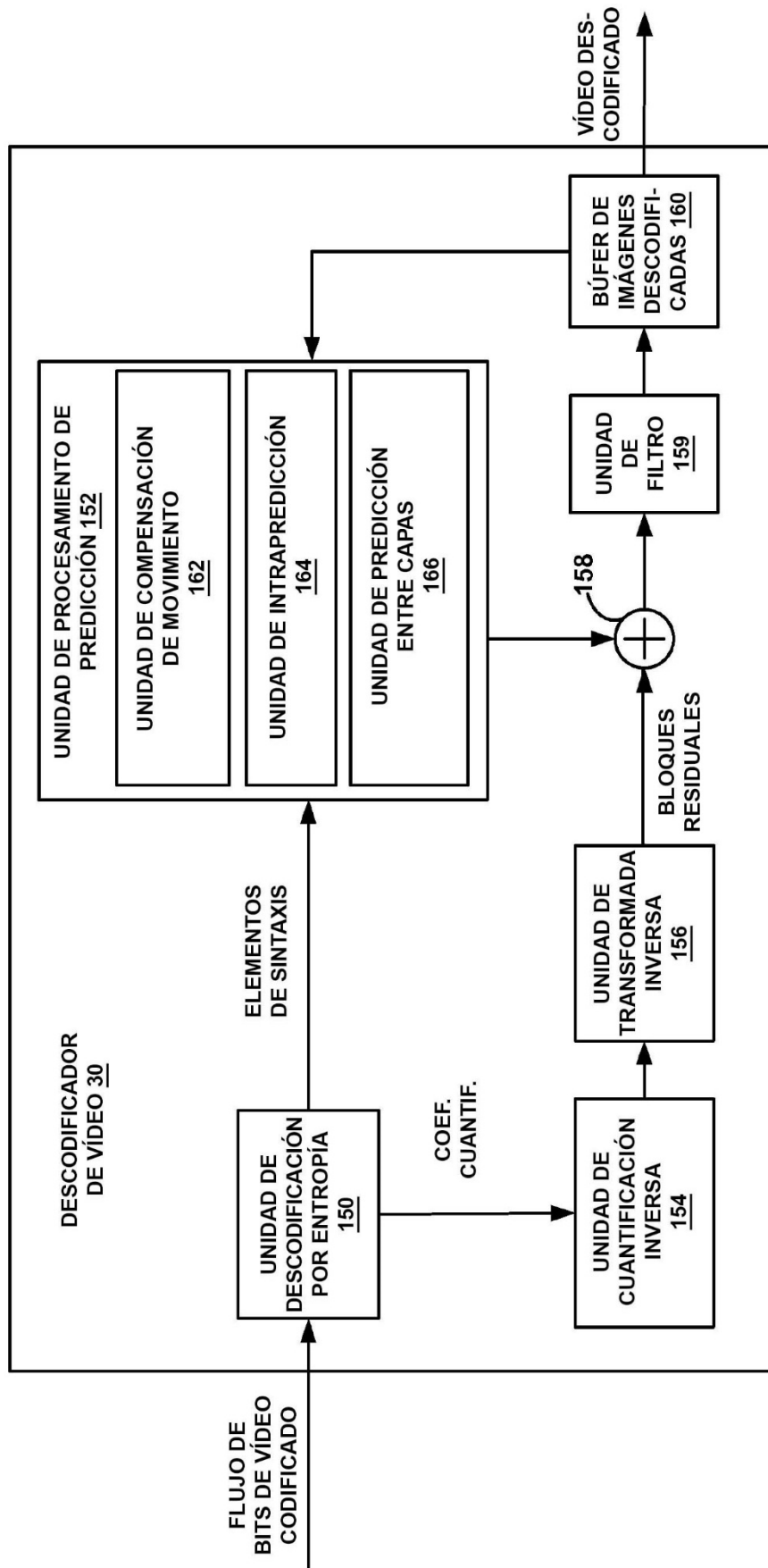
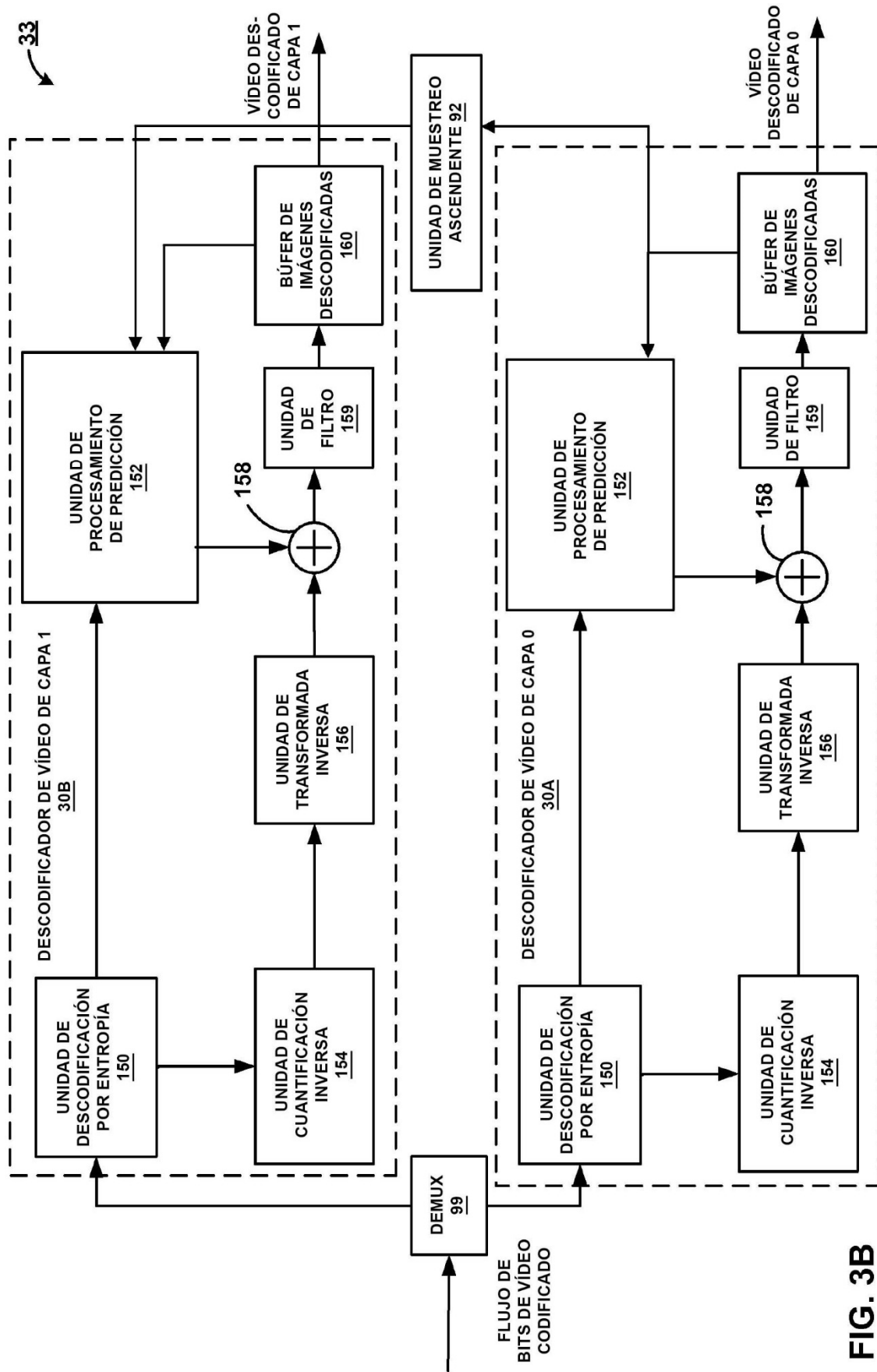


FIG. 3A



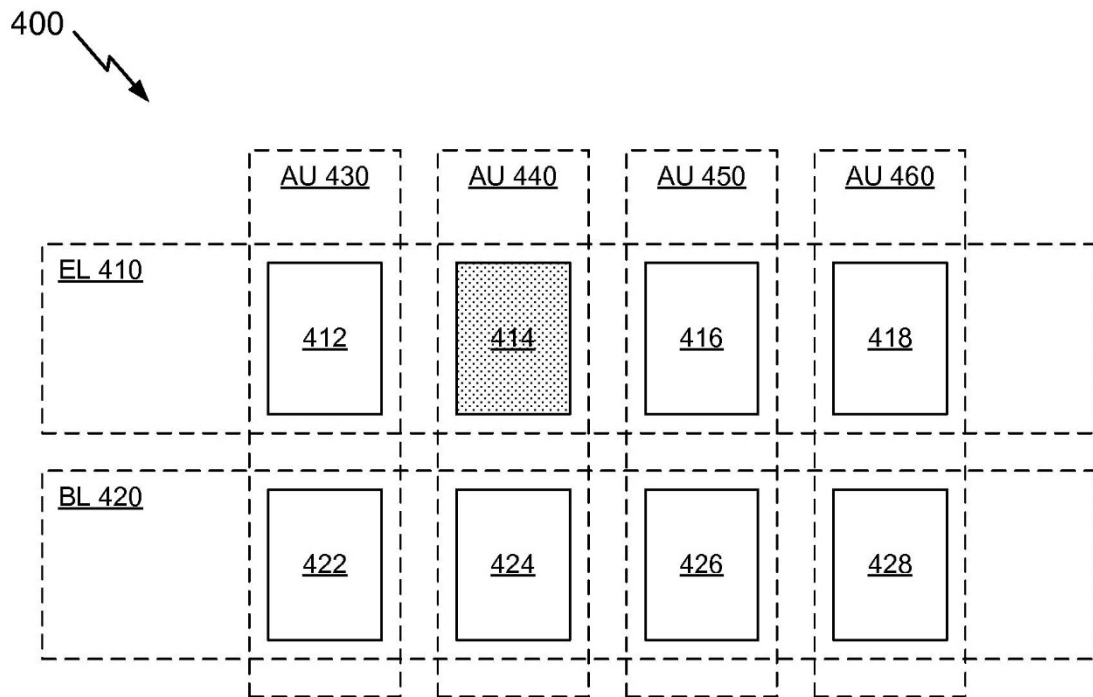


FIG. 4

500

Diagram 500 illustrates a table with data for AU 430, 440, 450, and 460. The table has two main sections: 'SIN RESTABLECIMIENTO' and 'CON RESTABLECIMIENTO'. Each section has rows for 'LSB' and 'MSB', and columns for 'EL' and 'BL'.

	AU		430	440	450	460
SIN RESTABLE- CIMIENTO	LSB	EL	112	113	114	115
		BL	112	113	114	115
	MSB	EL	1	0	0	0
		BL	1	<b>1</b>	<b>1</b>	<b>1</b>
CON RESTABLE- CIMIENTO	LSB	EL	112	113	114	115
		BL	112	113	114	115
	MSB	EL	1	0	0	0
		BL	1	<b>0</b>	<b>0</b>	<b>0</b>

FIG. 5



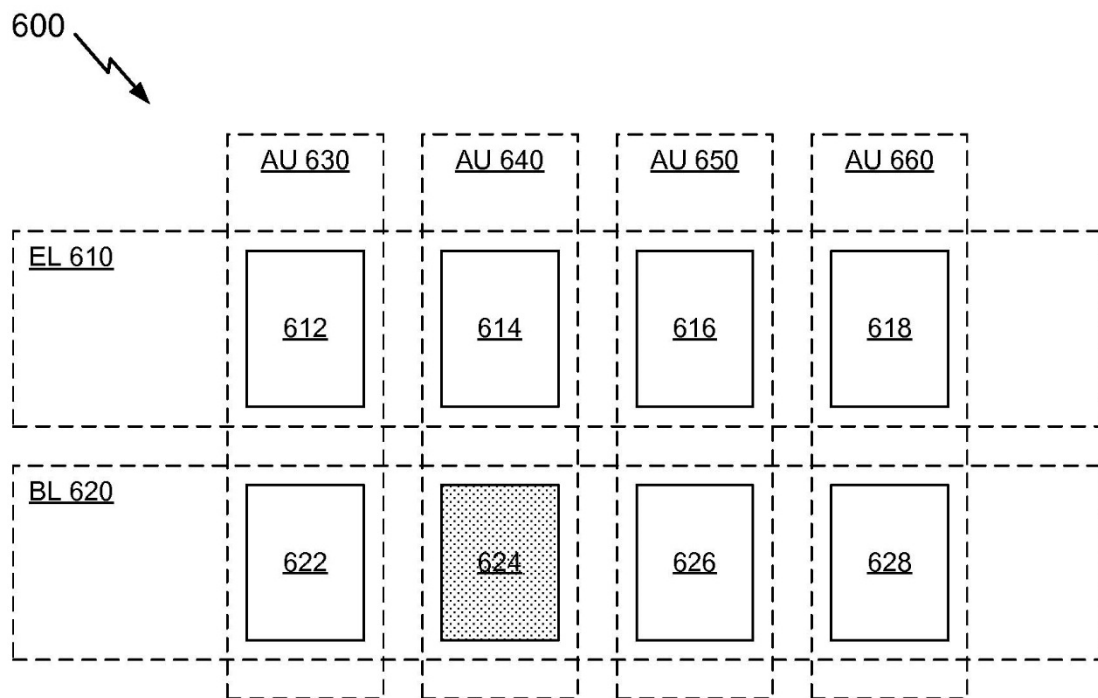


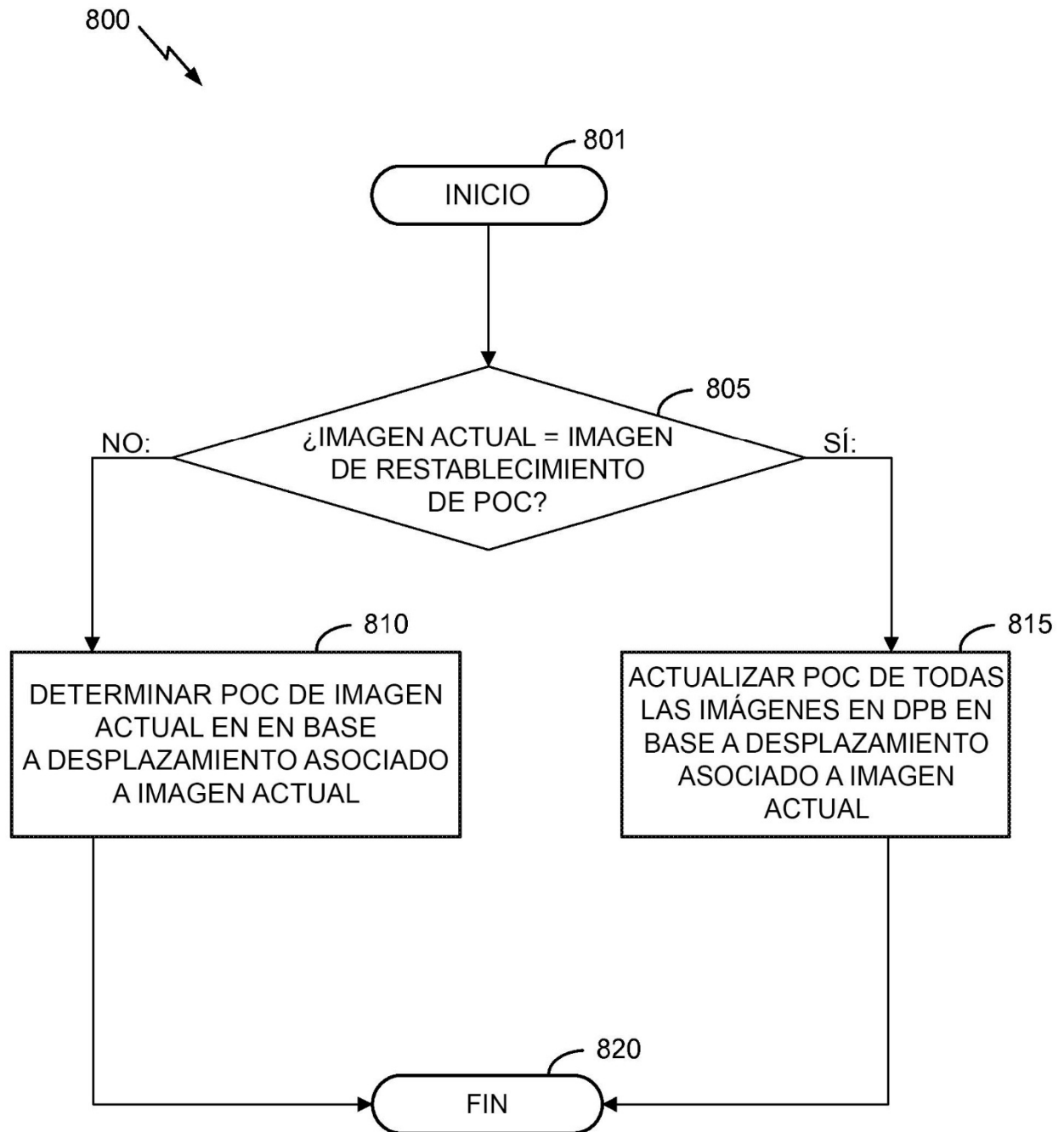
FIG. 6

700

The diagram shows a table with data for AU 630, 640, 650, and 660. The table is divided into two main sections: SIN RESTABLECIMIENTO and CON RESTABLECIMIENTO. Each section has rows for LSB and MSB, and columns for EL and BL.

	AU		630	640	650	660
SIN RESTABLE- CIMIENTO	LSB	EL	112	<b>113</b>	<b>114</b>	<b>115</b>
		BL	112	0	1	2
	MSB	EL	1	<b>1</b>	<b>1</b>	<b>1</b>
		BL	1	0	0	0
CON RESTABLE- CIMIENTO	LSB	EL	112	<b>0</b>	<b>1</b>	<b>2</b>
		BL	112	0	1	2
	MSB	EL	1	<b>0</b>	<b>0</b>	<b>0</b>
		BL	1	0	0	0

FIG. 7



**FIG. 8**