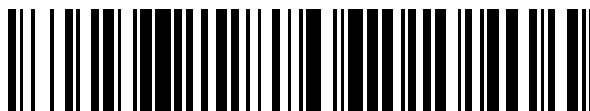


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 708 969**

51 Int. Cl.:

H04N 19/70 (2014.01)
H04N 19/30 (2014.01)
H04N 19/187 (2014.01)
H04N 19/423 (2014.01)
H04N 21/2343 (2011.01)
H04N 21/44 (2011.01)
H04N 21/845 (2011.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **04.04.2014 PCT/JP2014/001967**
- 87 Fecha y número de publicación internacional: **16.10.2014 WO14167817**
- 96 Fecha de presentación y número de la solicitud europea: **04.04.2014 E 14782853 (7)**
- 97 Fecha y número de publicación de la concesión europea: **31.10.2018 EP 2984847**

54 Título: **Cambio de señalización en conjuntos de capas de salida**

30 Prioridad:

07.04.2013 US 201313858076
09.07.2013 US 201361844272 P
11.07.2013 US 201361845309 P
19.07.2013 US 201361856575 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
12.04.2019

73 Titular/es:

DOLBY INTERNATIONAL AB (100.0%)
Apollo Building, 3E, Herikerbergweg 1-35
1101 CN Amsterdam Zuidoost, NL

72 Inventor/es:

DESHPANDE, SACHIN G.

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 708 969 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

Cambio de señalización en conjuntos de capas de salida

Campo técnico

5 La presente descripción se refiere de manera general a dispositivos electrónicos. Más específicamente, la presente descripción se refiere a dispositivos electrónicos para señalar parámetros de decodificadores de referencia hipotéticos basados en imagen secundaria, y a sistemas y métodos para operación híbrida de un almacenador temporal de imágenes decodificadas (DPB).

Antecedentes de la técnica

10 Los dispositivos electrónicos han llegado a ser más pequeños y más potentes con el fin de satisfacer las necesidades de los consumidores y mejorar la portabilidad y la comodidad. Los consumidores han llegado a ser dependientes de los dispositivos electrónicos y han llegado a esperar un aumento de funcionalidad. Algunos ejemplos de dispositivos electrónicos incluyen ordenadores de sobremesa, ordenadores portátiles, teléfonos celulares, teléfonos inteligentes, reproductores multimedia, circuitos integrados, etc.

15 Algunos dispositivos electrónicos se usan para procesar y mostrar medios digitales. Por ejemplo, los dispositivos electrónicos portátiles ahora permiten que los medios digitales se consuman casi en cualquier ubicación donde pueda estar un consumidor. Además, algunos dispositivos electrónicos pueden proporcionar descarga o difusión en forma continua de contenido multimedia digital para el uso y disfrute de un consumidor.

20 La creciente popularidad de los medios digitales ha presentado diversos problemas. Por ejemplo, representar de manera eficiente medios digitales de alta calidad para almacenamiento, transmisión y reproducción rápida presenta diversos retos. Como se puede observar a partir de esta discusión, pueden ser beneficiosos sistemas y métodos que representen medios digitales de manera eficiente con un rendimiento mejorado.

Los anteriores y otros objetivos, características y ventajas de la invención se entenderán más fácilmente tras la consideración de la siguiente descripción detallada de la invención, tomada junto con los dibujos que se acompañan.

25 J. Chen et al. han presentado "SHVC Working Draft 1" en la 12. REUNIÓN JCT-VC; 103. REUNIÓN DEL MPEG celebrada del 14 de enero de 2013 al 23 de enero de 2013 en GINEBRA; (Equipo de Colaboración Conjunto en Codificación de Vídeo de ISO/IEC JTC1/SC29/WG1 1 e ITU-T SG.1 6), recuperable en el URL: [HTTP://WFPT3.ITU.INT/AV-ARCH/JCTVC-SITE/](http://WFPT3.ITU.INT/AV-ARCH/JCTVC-SITE/), nº JCTVC-L1008.

30 G. Tech et al. han presentado "MV-HEVC Draft Text 3" en la 3. REUNIÓN JCT-VC; 103. REUNIÓN DEL MPEG celebrada del 14 de enero de 2013 al 23 de enero de 2013 en GINEBRA; (Equipo de Colaboración Conjunto sobre Codificación de Vídeo de ISO/IEC JTC1/SC29/WG1 1 e ITU-T SG.1 6), recuperable en el URL: [HTTP://PHENIX.INT-EVRY.FR/JCT2/](http://PHENIX.INT-EVRY.FR/JCT2/), nº JCT3V-C1004.

Compendio de la Invención

Solución al Problema

35 La presente invención se refiere a un método para decodificar un flujo de bits de vídeo como se establece en la reivindicación 1. La descripción de habilitación para la invención se encuentra en la realización de las Figuras 35 y 36. Las realizaciones restantes han de ser entendidas como ejemplos que no describen partes de la presente invención.

Breve descripción de los dibujos

40 La Figura 1A es un diagrama de bloques que ilustra un ejemplo de uno o más dispositivos electrónicos en los que se pueden implementar sistemas y métodos para enviar un mensaje y almacenar temporalmente un flujo de bits.

La Figura 1B es otro diagrama de bloques que ilustra un ejemplo de uno o más dispositivos en los que se pueden implementar sistemas y métodos para enviar un mensaje y almacenar temporalmente un flujo de bits.

La Figura 2 es un diagrama de flujo que ilustra una configuración de un método para enviar un mensaje.

45 La Figura 3 es un diagrama de flujo que ilustra una configuración de un método para determinar uno o más retardos de eliminación para unidades de decodificación en una unidad de acceso.

La Figura 4 es un diagrama de flujo que ilustra una configuración de un método para almacenar temporalmente un flujo de bits;

La Figura 5 es un diagrama de flujo que ilustra una configuración de un método para determinar uno o más retardos de eliminación para unidades de decodificación en una unidad de acceso.

- La Figura 6A es un diagrama de bloques que ilustra una configuración de un codificador 604 en un dispositivo electrónico.
- La Figura 6B es otro diagrama de bloques que ilustra una configuración de un codificador 604 en un dispositivo electrónico.
- 5 La Figura 7A es un diagrama de bloques que ilustra una configuración de un decodificador en un dispositivo electrónico.
- La Figura 7B es otro diagrama de bloques que ilustra una configuración de un decodificador en un dispositivo electrónico.
- La Figura 8 ilustra varios componentes que se pueden utilizar en un dispositivo electrónico de transmisión.
- 10 La Figura 9 es un diagrama de bloques que ilustra varios componentes que se pueden utilizar en un dispositivo electrónico de recepción.
- La Figura 10 es un diagrama de bloques que ilustra una configuración de un dispositivo electrónico en el que se pueden implementar sistemas y métodos para enviar un mensaje.
- 15 La Figura 11 es un diagrama de bloques que ilustra una configuración de un dispositivo electrónico en el que se pueden implementar sistemas y métodos para almacenar temporalmente un flujo de bits.
- La Figura 12 es un diagrama de bloques que ilustra una configuración de un método para la operación de un almacenador temporal de imágenes decodificadas.
- La Figura 13A ilustra diferentes sintaxis de cabecera de Unidad de NAL.
- La Figura 13B ilustra diferentes sintaxis de cabecera de Unidad de NAL.
- 20 La Figura 13C ilustra diferentes sintaxis de cabecera de Unidad de NAL.
- La Figura 14 ilustra una sintaxis general de Unidad de NAL.
- La Figura 15 ilustra un conjunto de parámetros de vídeo existente.
- La Figura 16 ilustra tipos de escalabilidad existentes.
- La Figura 17 ilustra un conjunto de parámetros de vídeo ejemplar.
- 25 La Figura 18 ilustra una sintaxis de mapa de escalabilidad ejemplar.
- La Figura 19 ilustra un conjunto de parámetros de vídeo ejemplar.
- La Figura 20 ilustra un conjunto de parámetros de vídeo existente.
- La Figura 21 ilustra un tipo de dimensión, sintaxis de id de dimensión existentes.
- La Figura 22 ilustra un conjunto de parámetros de vídeo ejemplar.
- 30 La Figura 23 ilustra una sintaxis de mapa de escalabilidad ejemplar.
- La Figura 24 ilustra un conjunto de parámetros de vídeo ejemplar.
- La Figura 25 ilustra un conjunto de parámetros de vídeo ejemplar.
- La Figura 26 ilustra un conjunto de parámetros de vídeo ejemplar.
- La Figura 27 ilustra una sintaxis de máscara de escalabilidad ejemplar.
- 35 La Figura 28 ilustra una sintaxis de extensión de conjunto de parámetros de vídeo ejemplar.
- La Figura 29 ilustra una sintaxis de extensión de conjunto de parámetros de vídeo ejemplar.
- La Figura 30 ilustra una sintaxis de extensión de conjunto de parámetros de vídeo ejemplar.
- La Figura 31 ilustra una sintaxis de extensión de conjunto de parámetros de vídeo ejemplar.
- La Figura 32 ilustra una sintaxis de extensión de conjunto de parámetros de vídeo ejemplar.
- 40 La Figura 33 ilustra una sintaxis de extensión de conjunto de parámetros de vídeo ejemplar.

- La Figura 34 ilustra una sintaxis de conjunto de parámetros de vídeo ejemplar.
- La Figura 35 ilustra una sintaxis de extensión de conjunto de parámetros de vídeo ejemplar.
- La Figura 36 ilustra una sintaxis de cambio de conjuntos de capas de salida ejemplar.
- La Figura 37 ilustra otra sintaxis de cambio de conjuntos de capas de salida ejemplar.
- 5 La Figura 38A ilustra una sintaxis de extensión de parámetros de vídeo ejemplar.
- La Figura 38B ilustra una sintaxis de extensión de parámetros de vídeo ejemplar.
- La Figura 39A ilustra una sintaxis de `op_dpb_info_parameters(j)` ejemplar.
- La Figura 39B ilustra una sintaxis de `op_dpb_info_parameters(j)` ejemplar.
- La Figura 40 ilustra otra sintaxis de extensión de parámetros de vídeo ejemplar.
- 10 La Figura 41 ilustra otra sintaxis de `oop_dpb_info_parameters(j)` ejemplar.
- La Figura 42 ilustra otra sintaxis de `oop_dpb_info_parameters(j)` ejemplar.
- La Figura 43 ilustra una sintaxis de `num_dpb_info_parameters` ejemplar.
- La Figura 44 ilustra otra sintaxis de `oop_dpb_info_parameters(j)` ejemplar.
- La Figura 45 ilustra otra sintaxis de `num_dpb_info_parameters` ejemplar.
- 15 La Figura 46 ilustra otra sintaxis de `num_dpb_info_parameters` ejemplar.
- La Figura 47 ilustra otra sintaxis de extensión de parámetros de vídeo ejemplar y `layer_dpb_info(i)`.
- La Figura 48 ilustra una sintaxis de `oop_dpb_info_parameters` y `layer_dpb_info(i)` ejemplar.
- La Figura 49A ilustra otra `vps_extension()` ejemplar.
- La Figura 49B ilustra otra `vps_extension()` ejemplar.
- 20 La Figura 50 ilustra un `oop_dpb_maxbuffering_parameters(i)` ejemplar.
- La Figura 51 ilustra un `layer_dpb_info_parameters(i)` ejemplar.
- La Figura 52 ilustra otra `vps_extension()` ejemplar.
- La Figura 53 ilustra otra `vps_extension()` ejemplar.
- La Figura 54 ilustra un `oop_dpb_maxbuffering_parameters(i,k)` ejemplar.
- 25 La Figura 55 ilustra un `oop_dpb_maxbuffering_parameters(i,k)` ejemplar.
- La Figura 56 ilustra otra `vps_extension()` ejemplar.
- La Figura 57 ilustra un `oop_dpb_maxbuffering_parameters(i,k)` ejemplar.
- La Figura 58 ilustra un `oop_dpb_maxbuffering_parameters(i,k)` ejemplar.
- La Figura 59 ilustra un `oop_dpb_maxbuffering_parameters(i,k)` ejemplar.
- 30 La Figura 60 ilustra un `oop_dpb_maxbuffering_parameters(i,k)` ejemplar.
- La Figura 61 ilustra un `oop_dpb_maxbuffering_parameters(i,k)` ejemplar.
- La Figura 62 ilustra un `seq_parameter_set_rbsp()` ejemplar.
- La Figura 63 es un diagrama de bloques que ilustra codificación de vídeo entre múltiples dispositivos electrónicos.
- 35 La Figura 64 es un diagrama de flujo de un método para operación de almacenador temporal de imágenes decodificadas (DPB) híbrida.
- La Figura 65 es un diagrama de flujo de otro método para operación de almacenador temporal de imágenes decodificadas (DPB) híbrida.

La Figura 66 es un diagrama de bloques que ilustra una configuración de un decodificador;

La Figura 67A es un diagrama de bloques que ilustra el uso tanto de una capa de mejora como de una capa base para codificación de vídeo con un almacenador temporal de imágenes decodificadas (DPB) separado y un módulo de operación de almacenador temporal de imágenes decodificadas (DPB) híbrida separada para la capa base y la capa de mejora.

La Figura 67B es un diagrama de bloques que ilustra el uso de un almacenador temporal de imágenes decodificadas (DPB) compartido y un módulo de operación de almacenador temporal de imágenes decodificadas (DPB) híbrida compartida para la capa base y la capa de mejora.

La Figura 68 es un diagrama de temporización que ilustra la operación del almacenador temporal de imágenes decodificadas (DPB) híbrida.

La Figura 69 es un diagrama de bloques que ilustra la estructura y la temporización para unidades de capas de abstracción de red (NAL) de capas para imágenes codificadas y unidades de acceso (AU) cuando la segunda capa de mejora (EL2) tiene una tasa de imagen más baja que la capa base (BL) y la primera capa de mejora (EL1).

La Figura 70 es un diagrama de bloques que ilustra la estructura y la temporización para las unidades de capas de abstracción de red (NAL) de capas para imágenes codificadas y unidades de acceso (AU) cuando la capa base (BL) tiene una tasa de imagen más baja que la primera capa de mejora (EL1) y la segunda capa de mejora (EL2).

Ejemplo 1

Se describe un dispositivo electrónico para enviar un mensaje. El dispositivo electrónico incluye un procesador e instrucciones almacenadas en memoria que está en comunicación electrónica con el procesador. El dispositivo electrónico determina, cuando un Almacenador Temporal de Imágenes Codificadas (CPB) soporta la operación a nivel de imagen secundaria, ya sea incluir un parámetro de retardo de eliminación del CPB de la unidad de decodificación común en un mensaje de Información de Mejora Suplementaria (SEI) de temporización de imagen. El dispositivo electrónico también genera, cuando el parámetro de retardo de eliminación del CPB de la unidad de decodificación común ha de ser incluido en el mensaje de SEI de temporización de imagen (o algún otro mensaje de SEI o algún otro conjunto de parámetros, por ejemplo, un conjunto de parámetros de imagen o conjunto de parámetros de secuencia o conjunto de parámetros de vídeo o conjunto de parámetros de adaptación), el parámetro de retardo de eliminación del CPB de la unidad de decodificación común, en donde el parámetro de retardo de eliminación del CPB de la unidad de decodificación común es aplicable a todas las unidades de decodificación en una unidad de acceso desde el CPB. El dispositivo electrónico también genera, cuando el parámetro de retardo de eliminación del CPB de la unidad de decodificación común no ha de ser incluido en el mensaje de SEI de temporización de imagen, un parámetro de retardo de eliminación del CPB de la unidad de decodificación separado para cada unidad de decodificación en la unidad de acceso. El dispositivo electrónico también envía el mensaje de SEI de temporización de imagen con el parámetro de retardo de eliminación del CPB de la unidad de decodificación común o los parámetros de retardo de eliminación del CPB de la unidad de decodificación.

El parámetro de retardo de eliminación del CPB de la unidad de decodificación común puede especificar una cantidad de tictacs de reloj de imagen secundaria a esperar después de la eliminación del CPB de una unidad de decodificación inmediatamente anterior antes de la eliminación del CPB de una unidad de decodificación actual en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen.

Además, cuando una unidad de decodificación es una primera unidad de decodificación en una unidad de acceso, el parámetro de retardo de eliminación del CPB de la unidad de decodificación común puede especificar una cantidad de tictacs de reloj de imagen secundaria a esperar después de la eliminación del CPB de una última unidad de decodificación en una unidad de acceso asociada con un mensaje de SEI de período de almacenamiento temporal más reciente en una unidad de acceso anterior antes de eliminar del CPB la primera unidad de decodificación en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen.

Por el contrario, cuando la unidad de decodificación no es una primera unidad de decodificación en una unidad de acceso, el parámetro de retardo de eliminación del CPB de la unidad de decodificación común puede especificar una cantidad de tictacs de reloj de imagen secundaria a esperar después de la eliminación del CPB de una unidad de decodificación anterior en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen antes de la eliminación del CPB de una unidad de decodificación actual en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen.

Los parámetros de retardo de eliminación del CPB de la unidad de decodificación pueden especificar una cantidad de tictacs de reloj de imagen secundaria a esperar después de la eliminación del CPB de última unidad de decodificación antes de la eliminación del CPB de una unidad de decodificación de orden i en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen.

El dispositivo electrónico puede calcular los parámetros de retardo de eliminación del CPB de la unidad de decodificación según el resto de un contador de módulo $2^{(cpb_removal_delay_length_minus1+1)}$ donde

cpb_removal_delay_length_minus1+1 es una longitud de un parámetro de retardo de eliminación del CPB de la unidad de decodificación común.

5 El dispositivo electrónico también puede generar, cuando el CPB soporta la operación a nivel de unidad de acceso, un mensaje de SEI de temporización de imagen que incluye un parámetro de retardo de eliminación del CPB que especifica cuantos tictacs de reloj esperar después de la eliminación del CPB de una unidad de acceso asociada con un mensaje de SEI de período de almacenamiento temporal más reciente en una unidad de acceso anterior antes de la eliminación del CPB de los datos de la unidad de acceso asociados con el mensaje de SEI de temporización de imagen.

10 El dispositivo electrónico también puede determinar si el CPB soporta operación a nivel de imagen secundaria o a nivel de unidad de acceso. Esto puede incluir determinar una bandera de temporización de imagen que indica si un Almacenador Temporal de Imágenes Codificadas (CPB) proporciona parámetros que soportan operación a nivel de imagen secundaria en base a un valor de la bandera de temporización de imagen. La bandera de temporización de imagen se puede incluir en el mensaje de SEI de temporización de imagen.

15 La determinación de si incluir un parámetro de retardo de eliminación del CPB de la unidad de decodificación común puede incluir el establecimiento de una bandera de retardo de eliminación del CPB de la unidad de decodificación común en 1 cuando el parámetro de retardo de eliminación del CPB de la unidad de decodificación común ha de ser incluido en el mensaje de SEI de temporización de imagen. También puede incluir el establecimiento de la bandera de retardo de eliminación del CPB de la unidad de decodificación común en 0 cuando el parámetro de retardo de eliminación del CPB de la unidad de decodificación común no ha de ser incluido en el mensaje de SEI de temporización de imagen. La bandera de retardo de eliminación del CPB de la unidad de decodificación común se puede incluir en el mensaje de SEI de temporización de imagen.

20 El dispositivo electrónico también puede generar, cuando el CPB soporta operación a nivel de imagen secundaria, parámetros relacionados con unidades de capa de abstracción de red (NAL) separadas que indican una cantidad, desplazada en uno, de unidades de NAL para cada unidad de decodificación en una unidad de acceso. Alternativamente, o además de, el dispositivo electrónico puede generar un parámetro de NAL común que indica una cantidad, desplazada en uno, de unidades de NAL comunes a cada unidad de decodificación en una unidad de acceso.

25 También se describe un dispositivo electrónico para almacenar temporalmente un flujo de bits. El dispositivo electrónico incluye un procesador e instrucciones almacenadas en memoria que está en comunicación electrónica con el procesador. El dispositivo electrónico determina que un CPB señala parámetros a nivel de imagen secundaria para una unidad de acceso. El dispositivo electrónico también determina, cuando un mensaje de Información de Mejora Suplementaria (SEI) de temporización de imagen recibido comprende la bandera de retardo de eliminación del Almacenador Temporal de Imágenes Codificadas (CPB) de la unidad de decodificación común, un parámetro de retardo de eliminación del CPB de la unidad de decodificación común aplicable a todas las unidades de decodificación en la unidad de acceso. El dispositivo electrónico también determina, cuando el mensaje de SEI de temporización de imagen no comprende la bandera de retardo de eliminación del CPB de la unidad de decodificación común, un parámetro de retardo de eliminación del CPB de la unidad de decodificación separado para cada unidad de decodificación en la unidad de acceso. El dispositivo electrónico también elimina las unidades de decodificación del CPB usando el parámetro de retardo de eliminación del CPB de la unidad de decodificación común o los parámetros de retardo de eliminación del CPB de la unidad de decodificación separados. El dispositivo electrónico también decodifica las unidades de decodificación en la unidad de acceso.

40 En una configuración, el dispositivo electrónico determina que una bandera de temporización de imagen se establece en el mensaje de SEI de temporización de imagen. El dispositivo electrónico también puede establecer un parámetro de retardo de eliminación del CPB, cpb_removal_delay, según

45 [Expresión matemática 1]

$$\text{cpb_removal_delay} = \frac{\left(\sum_{i=0}^{\text{num_decoding_units_minus1}} \text{du_cpb_removal_delay}[i] \right) * t_{c,sub}}{t_c}$$

donde du_cpb_removal_delay[i] son los parámetros de retardo de eliminación del CPB de la unidad de decodificación, t_c es un tictac de reloj, t_{c,sub} es un tictac de reloj de imagen secundaria, num_decoding_units_minus1 es una cantidad de unidades de decodificación en la unidad de acceso desplazada en uno, e i es un índice.

50 Alternativamente, el dispositivo electrónico puede establecer un parámetro de retardo de eliminación del CPB, cpb_removal_delay, y du_cpb_removal_delay[num_decoding_units_minus1] para satisfacer la ecuación

[Expresión matemática 2]

$$-1 \leq [\text{cpb_removal_delay} * t_c - (\sum_{i=0}^{\text{num_decoding_units_minus1}} \text{du_cpb_removal_delay}[i]) * t_{c,\text{sub}}] \leq 1$$

donde $\text{du_cpb_removal_delay}[i]$ son los parámetros de retardo de eliminación del CPB de la unidad de decodificación, t_c es un tictac de reloj, $t_{c,\text{sub}}$ es un tictac de reloj de imagen secundaria, $\text{num_decoding_units_minus1}$ es una cantidad de unidades de decodificación en la unidad de acceso desplazada en uno, e i es un índice.

5 Alternativamente, el dispositivo electrónico puede establecer un parámetro de retardo de eliminación del CPB, cpb_removal_delay , y $\text{du_cpb_removal_delay}[\text{num_decoding_units_minus1}]$ según $\text{cpb_removal_delay} * t_c = \text{du_cpb_removal_delay}[\text{num_decoding_units_minus1}] * t_{c,\text{sub}}$ donde $\text{du_cpb_removal_delay}[\text{num_decoding_units_minus1}]$ es el parámetro de retardo de eliminación del CPB de la
10 unidad de decodificación para la unidad de decodificación de orden $\text{num_decoding_units_minus1}$, t_c es un tictac de reloj, $t_{c,\text{sub}}$ es un tictac de reloj de imagen secundaria, $\text{num_decoding_units_minus1}$ es una cantidad de unidades de decodificación en la unidad de acceso desplazada en uno.

En una configuración, el dispositivo electrónico determina que una bandera de temporización de imagen se establece en el mensaje de SEI de temporización de imagen. El dispositivo electrónico también puede establecer los
15 parámetros de retardo de eliminación del CPB, cpd_removal_delay , y $\text{du_cpd_removal_delay}[\text{num_decoding_units_minus1}]$ para satisfacer la ecuación: $-1 \leq (\text{cpd_removal_delay} * t_c - \text{du_cpd_removal_delay}[\text{num_decoding_units_minus1}] * t_{c,\text{sub}}) \leq 1$ donde $\text{du_cpd_removal_delay}[\text{num_decoding_units_minus1}]$ es el parámetro de retardo de eliminación del CPB de la
20 unidad de decodificación para la unidad de decodificación de orden $\text{num_decoding_units_minus1}$, t_c es un tictac de reloj, $t_{c,\text{sub}}$ es un tictac de reloj de imagen secundaria, $\text{num_decoding_units_minus1}$ es una cantidad de unidades de decodificación en la unidad de acceso desplazada en uno.

Una variable ClockDiff se puede definir como $\text{ClockDiff} = (\text{num_units_in_tick} - (\text{num_units_in_sub_tick} * (\text{num_decoding_units_minus1} + 1))) / \text{time_scale}$ donde num_units_in_tick es el número de
25 unidades de tiempo de un reloj que opera a la frecuencia time_scale Hz que corresponde a un aumento de un contador de tictac de reloj, $\text{num_units_in_sub_tick}$ es el número de unidades de tiempo de un reloj que opera a la frecuencia time_scale Hz que corresponde a un aumento de un contador de tictac de reloj de imagen secundaria, $\text{num_decoding_units_minus1} + 1$ es una cantidad de unidades de decodificación en la unidad de acceso, y time_scale es el número de unidades de tiempo que pasan en un segundo.

30 Cuando una bandera de decodificador de referencia hipotética (HRD) de bajo retardo (por ejemplo, $\text{low_delay_hrd_flag}$) se establece en 1, $t_{r,n}(m) < t_{af}(m)$, una bandera de temporización de imagen se establece en 1, el CPB está operando en nivel de imagen secundaria y ClockDiff es mayor que cero, el tiempo de eliminación para la unidad de decodificación m , $t_r(m)$ se determina según: $t_r(m) = t_{r,n}(m) + t_{c,\text{sub}} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c,\text{sub}}) + \text{ClockDiff}$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la unidad de decodificación m , $t_{c,\text{sub}}$ es un tictac de reloj de imagen secundaria, $\text{Ceil}()$ es una función de techo y $t_{af}(m)$ es un tiempo de llegada final de la unidad de decodificación m .

35 Cuando una bandera de decodificador de referencia hipotética (HRD) de bajo retardo (por ejemplo, $\text{low_delay_hrd_flag}$) se establece en 1, $t_{r,n}(n) < t_{af}(n)$, una bandera de temporización de imagen se establece en 1, el CPB está operando en un nivel de unidad de acceso y ClockDiff es mayor que cero, el tiempo de eliminación para la unidad de acceso n , $t_r(n)$ se determina según: $t_r(n) = t_{r,n}(n) + t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c) - \text{ClockDiff}$ donde $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n , t_c es un tictac de reloj, $\text{Ceil}()$ es una función de techo y $t_{af}(n)$ es un tiempo de llegada final de la unidad de acceso n .
40

45 Cuando una bandera de decodificador de referencia hipotética (HRD) de bajo retardo (por ejemplo, $\text{low_delay_hrd_flag}$) se establece en 1, $t_{r,n}(m) < t_{af}(m)$, una bandera de temporización de imagen se establece en 1 y el CPB está operando en nivel de imagen secundaria, el tiempo de eliminación para la última unidad de decodificación m de unidad de acceso, $t_r(m)$ según: $t_r(m) = t_{r,n}(m) + \max((t_{c,\text{sub}} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c,\text{sub}})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c)))$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la última unidad de acceso m , $t_{c,\text{sub}}$ es un tictac de reloj de imagen secundaria, $\text{Ceil}()$ es una función de techo, $t_{af}(m)$ es el tiempo de llegada final de la última unidad de decodificación m , $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n , t_c es un tictac de reloj y $t_{af}(n)$ es un tiempo de llegada final de la unidad de acceso n .

50 Cuando una bandera de decodificador de referencia hipotética (HRD) de bajo retardo se establece en 1, $t_{r,n}(n) < t_{af}(n)$, una bandera de temporización de imagen se establece en 1 y el CPB está operando en un nivel de unidad de acceso, el tiempo de eliminación para la unidad de acceso n , $t_r(n)$ según: $t_r(n) = t_{r,n}(n) + \max((t_{c,\text{sub}} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c,\text{sub}})), ((t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c))))$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la última unidad de decodificación n , $t_{c,\text{sub}}$ es un tictac de reloj imagen secundaria, $\text{Ceil}()$ es una función de techo, $t_{af}(m)$ es un tiempo de llegada final de la última unidad de decodificación m , $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de
55 acceso n , t_c es la marca de reloj y $t_{af}(n)$ es un tiempo de llegada final de la unidad de acceso n .

Cuando una bandera de decodificador de referencia hipotética (HRD) de bajo retardo (por ejemplo,

$t_{af}(n)$ es un tiempo de llegada final de la unidad de acceso n , y $t_{af}(m)$ es el tiempo de llegada final de la unidad de decodificación m en la unidad de acceso n .

5 Cuando una bandera de decodificador de referencia hipotética (HRD) de bajo retardo se establece en 1, $t_{r,n}(n) < t_{af}(n)$, una bandera de temporización de imagen se establece en 1 y el CPB está operando a nivel de unidad de acceso, el tiempo de eliminación para la unidad de acceso n , $t_r(n)$ según: $t_r(n) = t_{af}(n)$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la última unidad de decodificación n , t_{c_sub} es un tictac de reloj de imagen secundaria, $Ceil()$ es una función de techo, $t_{af}(m)$ es un tiempo de llegada final de la última unidad de decodificación m , $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n , t_c es la marca de reloj y $t_{af}(n)$ es un tiempo de llegada final de la unidad de acceso n .

10 Además, en algunos casos, se puede enviar una bandera en parte del flujo de bits para señalar cuál de las ecuaciones alternativas anteriores se usa para decidir el tiempo de eliminación de las unidades de decodificación y tiempo de eliminación de la unidad de acceso. En un caso, la bandera se puede llamar `du_au_cpb_alignment_mode_flag`. Si `du_au_cpb_alignment_mode_flag` es 1, entonces se usan las ecuaciones anteriores las cuales alinean la operación del CPB que opera en modo basado en imagen secundaria con el CPB que opera en el modo de unidad de acceso. Si `du_au_cpb_alignment_mode_flag` es 0, entonces se usan las ecuaciones anteriores las cuales no alinean la operación del CPB que opera en modo basado en imagen secundaria con el CPB que opera en el modo de unidad de acceso.

20 En un caso, la bandera `du_au_cpb_alignment_mode_flag` se puede señalar en la información de usabilidad de vídeo (VUI). En otro caso, la bandera `du_au_cpb_alignment_mode_flag` se puede enviar en el mensaje de SEI de temporización de imagen. En otro caso más, la bandera `du_au_cpb_alignment_mode_flag` se puede enviar a alguna otra parte normativa del flujo de bits. Un ejemplo de la sintaxis modificada y la semántica según los sistemas y métodos descritos en la presente memoria se da en la Tabla (0) de la siguiente manera.

[Tabla 0]

<code>pic_timing(payloadSize) {</code>
<code> if(CpbDpbDelaysPresentFlag) {</code>
<code> cpb_removal_delay</code>
<code> dpb_output_delay</code>
<code> if(sub_pic_cpb_params_present_flag) {</code>
<code> num_decoding_units_minus1</code>
<code> du_au_cpb_alignment_mode_flag</code>
<code> for(i = 0; i <= num_decoding_units_minus1; i++) {</code>
<code> num_nalus_in_du_minus1[i]</code>
<code> du_cpb_removal_delay[i]</code>
<code> }</code>
<code> }</code>
<code> }</code>
<code>}</code>

25 Tabla (0)

30 Se debería observar que se pueden usar símbolos (nombres) diferentes que los usados anteriormente para diversas variables. Por ejemplo, $t_r(n)$ de la unidad de acceso n se puede llamar `CpbRemovalTime(n)`, $t_r(m)$ de la unidad de decodificación n se puede llamar `CpbRemovalTime(m)`, t_{c_sub} se puede llamar `ClockSubTick`, t_c se puede llamar `ClockTick`, $t_{af}(n)$ de la unidad de acceso m se puede llamar `FinalArrivalTime(n)` de la unidad de acceso n , $t_{af}(m)$ de la unidad de decodificación m se puede llamar `FinalArrivalTime(m)`, $t_{r,n}(n)$ se puede llamar `NominalRemovalTime(n)` de la unidad de acceso n , $t_{r,n}(m)$ se puede llamar `NominalRemovalTime(m)` de la unidad de decodificación m .

35 También se describe un método para enviar un mensaje mediante un dispositivo electrónico. El método incluye determinar, cuando un Almacenador Temporal de Imagen Codificada (CPB) soporta la operación en un nivel de imagen secundaria, ya sea para incluir un parámetro de retardo de eliminación del CPB de la unidad de decodificación común en un mensaje de Información de Mejora Suplementaria (SEI) de temporización de imagen. El método también incluye la generación, cuando el parámetro de retardo de eliminación del CPB de la unidad de decodificación común ha de ser incluido en el mensaje de SEI de temporización de imagen, el parámetro de retardo de eliminación del CPB de la unidad de decodificación común, en donde el parámetro de retardo de eliminación del CPB de la unidad de decodificación común es aplicable a todas las unidades de decodificación en una unidad de acceso desde el CPB. El método también incluye la generación, cuando el parámetro de retardo de eliminación del CPB de la unidad de decodificación común no ha de ser incluido en el mensaje de SEI de temporización de imagen, un parámetro de retardo de eliminación del CPB de la unidad de decodificación separado para cada unidad de

decodificación en la unidad de acceso. El método también incluye enviar el mensaje de SEI de temporización de imagen con el parámetro de retardo de eliminación del CPB de la unidad de decodificación común o los parámetros de retardo de eliminación del CPB de la unidad de decodificación.

5 También se describe un método para almacenar temporalmente un flujo de bits mediante un dispositivo electrónico. El método incluye determinar que un CPB señala parámetros en un nivel de imagen secundaria para una unidad de acceso. El método también incluye la determinación, cuando un mensaje de información de Mejora Suplementaria (SEI) de temporización de imagen recibido comprende la bandera de retardo de eliminación del Almacenador Temporal de Imagen Codificada (CPB) de la unidad de decodificación común, un parámetro de retardo de eliminación del CPB de la unidad de decodificación común aplicable a todas las unidades de decodificación en la
10 unidad de acceso. El método también incluye la determinación, cuando el mensaje de SEI de temporización de imagen no comprende la bandera de retardo de eliminación del CPB de la unidad de decodificación común, un parámetro de retardo de eliminación del CPB de la unidad de decodificación separado para cada unidad de decodificación en la unidad de acceso. El método también incluye eliminar unidades de decodificación del CPB usando el parámetro de retardo de eliminación del CPB de la unidad de decodificación común o los parámetros de retardo de eliminación del CPB de la unidad de decodificación separados. El método también incluye la
15 decodificación de las unidades de decodificación en la unidad de acceso.

Los sistemas y métodos descritos en la presente memoria describen dispositivos electrónicos para enviar un mensaje y almacenar temporalmente un flujo de bits. Por ejemplo, los sistemas y métodos descritos en la presente memoria describen el almacenamiento temporal para flujos de bits que comienzan con parámetros de imagen
20 secundaria. En algunas configuraciones, los sistemas y métodos descritos en la presente memoria pueden describir parámetros de Decodificador de Referencia Hipotética (HRD) basados en imagen secundaria de señalización. Por ejemplo, los sistemas y métodos descritos en la presente memoria describen la modificación de un mensaje de Información de Mejora de Suplementaria (SEI) de temporización de imagen. Los sistemas y métodos descritos en la presente memoria (por ejemplo, la modificación de HRD) pueden dar como resultado una señalización más compacta de los parámetros cuando llega cada imagen secundaria y se elimina del CPB a intervalos regulares.
25

Además, cuando están presentes los parámetros de retardo de eliminación de CPB a nivel de imagen secundaria, el Almacenador Temporal de Imagen Codificada (CPB) puede operar a nivel de unidad de acceso o nivel de imagen secundaria. Los sistemas y métodos presentes también pueden imponer una restricción de flujo de bits de modo que la operación de CPB basada en el nivel de imagen secundaria y la operación de CPB a nivel de unidad de acceso
30 dan como resultado la misma temporización de eliminación de unidad de decodificación. Específicamente, la temporización de la eliminación de la última unidad de decodificación en una unidad de acceso cuando se opera en modo de imagen secundaria y la temporización de la eliminación de la unidad de acceso cuando se opera en el modo de unidad de acceso será la misma.

Se debería observar que aunque el término “hipotético” se usa en referencia a un HRD se puede implementar físicamente. Por ejemplo, “HRD” se puede usar para describir una implementación de un decodificador real. En algunas configuraciones, se puede implementar un HRD con el fin de determinar si un flujo de bits se ajusta a las especificaciones de Codificación de Vídeo de Alta Eficiencia (HEVC). Por ejemplo, se puede usar un HRD para determinar si las secuencias de bits de Tipo I y secuencias de bits de Tipo II se ajustan a las especificaciones HEVC. Un flujo de bits de Tipo I puede contener solamente unidades de Capa de Acceso a la Red (NAL) de Capa de Codificación de Vídeo (VCL) y unidades de NAL de datos de relleno. Un flujo de bits de Tipo II puede contener otras
40 unidades de NAL y elementos de sintaxis adicionales.

El documento del Equipo Colaborativo Conjunto sobre Codificación de Vídeo (JCTVC) JCTVC-10333 incluye un HRD basado en imagen secundaria y soporta mensajes de SEI de temporización de imagen. Esta funcionalidad ha sido incorporada en el Borrador del Comité de Codificación de Vídeo de Alta Eficiencia (HEVC) (JCTVC-11003).

45 Un ejemplo de sintaxis modificada y semántica según los sistemas y métodos descritos en la presente memoria se da en la Tabla (1) de la siguiente manera.

[Tabla 1]

pic timing(payloadSize) {
if(CpbDpbDelaysPresentFlag) {
cpb_removal_delay
dpb_output_delay
if(sub_pic_cpb_params_present_flag) {
num_decoding_units_minus1
common_du_cpb_removal_delay_flag
if(common_du_cpb_removal_delay_flag) {
common_du_cpb_removal_delay
}
for(i = 0; i <= num_decoding_units_minus1; i++) {
num_nalus_in_du_minus1[i]
if(common_du_cpb_removal_delay_flag)
du_cpb_removal_delay[i]
}
}
}
}

Se proporcionan a continuación ejemplos con respecto a la semántica de mensajes de SEI de período de almacenamiento temporal según los sistemas y métodos descritos en la presente memoria. En particular, los detalles adicionales con respecto a la semántica de los elementos de sintaxis modificada se dan a continuación.

5 Cuando NalHrdBpPresentFlag o VclHrdBpPresentFlag son iguales a 1, un mensaje de SEI de período de almacenamiento temporal se puede asociar con cualquier unidad de acceso en el flujo de bits, y un mensaje de SEI de período de almacenamiento temporal se puede asociar con cada unidad de acceso IDR, con cada unidad de acceso CRA y con cada unidad de acceso asociada con un mensaje de SEI del punto de recuperación. Para algunas aplicaciones, la presencia frecuente de un mensaje de SEI de período de almacenamiento temporal puede ser deseable. Un período de almacenamiento temporal se especifica como el conjunto de unidades de acceso entre dos instancias del mensaje de SEI de período de almacenamiento temporal en orden de decodificación.

15 'seq_parameter_set_id' especifica el conjunto de parámetros de secuencia que contiene los atributos de HRD de secuencia. El valor de seq_parameter_set_id puede ser igual al valor de seq_parameter_set_id en el conjunto de parámetros de imagen referenciado por la imagen codificada primaria asociada con el mensaje de SEI de período de almacenamiento temporal. El valor de seq_parameter_set_id puede estar en el intervalo de 0 a 31, inclusive.

20 'initial_cpb_removal_delay'[SchedSelIdx] especifica el retardo de CPB de orden SchedSelIdx entre el momento de llegada al CPB del primer bit de los datos codificados asociados con la unidad de acceso asociada con el mensaje de SEI de período de almacenamiento temporal y el tiempo de eliminación del CPB de los datos codificados asociados con la misma unidad de acceso, para el primer período de almacenamiento temporal después de la inicialización de HRD. El elemento de sintaxis tiene una longitud en bits dada por initial_cpb_removal_delay_length_minus1 + 1. Está en unidades de un reloj de 90 kHz. initial_cpb_removal_delay[SchedSelIdx] puede no ser igual a 0 y no puede exceder de 90000 * (CpbSize[SchedSelIdx]/BitRate[SchedSelIdx]), el equivalente de tiempo del tamaño de CPB en unidades de reloj de 90 kHz.

25 'initial_cpb_removal_delay_offset'[SchedSelIdx] se usa para el CPB de orden SchedSelIdx en combinación con el cpb_removal_delay para especificar el tiempo de entrega inicial de las unidades de acceso codificadas al CPB. initial_cpb_removal_delay_offset[SchedSelIdx] está en unidades de un reloj de 90 kHz. El elemento de sintaxis initial_cpb_removal_delay_offset[SchedSelIdx] es un código de longitud fija cuya longitud en bits está dada por initial_cpb_removal_delay_length_minus1 + 1. Este elemento de sintaxis no se usa por los decodificadores y se necesita solamente para el programador de entrega (HSS) (por ejemplo, como se especifica en el Anexo C de JCTVC-11003).

En toda la secuencia de vídeo codificada, la suma de initial_cpb_removal_delay[SchedSelIdx] y de initial_cpb_removal_delay_offset[SchedSelIdx] puede ser constante para cada valor de SchedSelIdx.

35 'initial_du_cpb_removal_delay'[SchedSelIdx] especifica el retardo para el CPB de orden SchedSelIdx entre el tiempo de llegada al CPB del primer bit de los datos codificados asociados con la primera unidad de decodificación en la unidad de acceso asociada con el mensaje de SEI del período de almacenamiento temporal y el tiempo de eliminación del CPB de los datos codificados asociados con la misma unidad de decodificación, para el primer período de almacenamiento temporal después de la inicialización de HRD. El elemento de sintaxis tiene una longitud en bits dada por initial_cpb_removal_delay_length_minus1 + 1. Está en unidades de un reloj de 90 kHz. initial_du_cpb_removal_delay[SchedSelIdx] puede no ser igual a 0 y no puede exceder de 90000 * (CpbSize[SchedSelIdx]/BitRate[SchedSelIdx]), el equivalente de tiempo del tamaño de CPB en unidades de reloj de 90 kHz.

40 'initial_du_cpb_removal_delay_offset'[SchedSelIdx] se usa para el CPB de orden SchedSelIdx en combinación con

el `cpb_removal_delay` para especificar el tiempo de entrega inicial de las unidades de decodificación al CPB.

`initial_cpb_removal_delay_offset[SchedSelIdx]` está en unidades de un reloj de 90 kHz. El elemento de sintaxis `initial_du_cpb_removal_delay_offset[SchedSelIdx]` es un código de longitud fija cuya longitud en bits se da por `initial_cpb_removal_delay_length_minus1 + 1`. Este elemento de sintaxis no se usa por los decodificadores y se necesita solamente para el programador de entrega (HSS) (por ejemplo, como se especifica en el Anexo C de JCTVC-11003).

En toda la secuencia de vídeo codificada, la suma de `initial_du_cpb_removal_delay[SchedSelIdx]` y de `initial_du_cpb_removal_delay_offset[SchedSelIdx]` puede ser constante para cada valor de `SchedSelIdx`.

Ejemplos con respecto a la semántica del mensaje de SEI de temporización de imagen según los sistemas y métodos descritos en la presente memoria se dan de la siguiente manera. En particular, los detalles adicionales con respecto a la semántica de los elementos de sintaxis modificados se dan de la siguiente manera.

La sintaxis del mensaje de SEI de temporización de imagen es dependiente del contenido del conjunto de parámetros de secuencia que está activo para la imagen codificada asociada con el mensaje de SEI de temporización de imagen. No obstante, a menos que el mensaje de SEI de temporización de imagen de una unidad de acceso de Regeneración Instantánea de Decodificación (IDR) esté precedido por un mensaje de SEI de período de almacenamiento temporal dentro de la misma unidad de acceso, la activación del conjunto de parámetros de secuencia asociado (y, para imágenes de IDR que no son la primera imagen en el flujo de bits, la determinación de que la imagen codificada es una imagen de IDR) no ocurre hasta la decodificación de la primera unidad de Capa de Abstracción de Red (NAL) de la porción codificada de la imagen codificada. Dado que la unidad de NAL de la porción codificada de la imagen codificada sigue al mensaje de SEI de temporización de imagen en orden de unidad de NAL, puede haber casos en los que sea necesario que un decodificador almacene la carga útil de secuencia de bytes en bruto (RBSP) que contiene el mensaje de SEI de temporización de imagen hasta determinar los parámetros del parámetro de secuencia que estarán activos para la imagen codificada, y luego realizar el análisis sintáctico del mensaje de SEI de temporización de imagen.

La presencia de un mensaje de SEI de temporización de imagen en el flujo de bits se especifica de la siguiente manera. Si `CpbDpbDelaysPresentFlag` es igual a 1, un mensaje de SEI de temporización de imagen puede estar presente en cada unidad de acceso de la secuencia de vídeo codificada. De otro modo (`CpbDpbDelaysPresentFlag` es igual a 0), pueden no estar presentes mensajes SEI de temporización de imagen en ninguna unidad de acceso de la secuencia de vídeo codificada.

'`cpb_removal_delay`' especifica cuántos tictacs de reloj (véase la subcláusula E.2.1 de JCTVC-11003) esperar después de la eliminación del CPB de la unidad de acceso asociada con el mensaje de SEI de período de almacenamiento temporal más reciente en una unidad de acceso anterior antes de eliminar del almacenador temporal los datos de la unidad de acceso asociados con el mensaje de SEI de temporización de imagen. Este valor también se usa para calcular un tiempo de llegada más temprano posible de los datos de la unidad de acceso al CPB para el HSS, como se especifica en el Anexo C de JCTVC-11003. El elemento de sintaxis es un código de longitud fija cuya longitud en bits se da por `cpb_removal_delay_length_minus1 + 1`. El `cpb_removal_delay` es el resto de un contador de módulo $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$.

El valor de `cpb_removal_delay_length_minus1` que determina la longitud (en bits) del elemento de sintaxis `cpb_removal_delay` es el valor de `cpb_removal_delay_length_minus1` codificado en el conjunto de parámetros de secuencia que está activo para la imagen codificada primaria asociada con el mensaje de SEI de temporización de imagen, aunque `cpb_removal_delay` especifica un número de tictacs de reloj en relación con el tiempo de eliminación de la unidad de acceso anterior que contiene un mensaje de SEI de período de almacenamiento temporal, que puede ser una unidad de acceso de una secuencia de vídeo codificada diferente.

'`dpb_output_delay`' se usa para calcular el tiempo de salida del Almacenador Temporal de Imágenes Decodificadas (DPB) de la imagen. Especifica cuántos tictacs de reloj esperar después de la eliminación de la última unidad de decodificación en una unidad de acceso del CPB antes de que la imagen decodificada se emita desde el DPB (véase la subcláusula C.2 de JCTVC-11003).

Con respecto al DPB, una imagen no se elimina del DPB en su tiempo de salida cuando aún está marcada como "usada para referencia a corto plazo" o "usada para referencia a largo plazo". Solamente se especifica un `dpb_output_delay` para una imagen decodificada. La longitud del elemento de sintaxis `dpb_output_delay` se da en bits por `dpb_output_delay_length_minus1 + 1`. Cuando `max_dec_pic_buffering[max_temporal_layers_minus1]` es igual a 0, `dpb_output_delay` puede ser igual a 0.

El tiempo de salida derivado del `dpb_output_delay` de cualquier imagen que se emita desde un decodificador conforme a la temporización de salida como se especifica en la subcláusula C.2 de JCTVC-11003 puede preceder al tiempo de salida derivado del `dpb_output_delay` de todas las imágenes en cualquier secuencia de vídeo codificada posterior en orden de decodificación. El orden de salida de imágenes establecido por los valores de este elemento de sintaxis puede ser el mismo orden que se estableció por los valores de `PicOrderCnt()` como se especifica por la subcláusula. Para imágenes que no se emiten mediante el proceso de "sacudidas" de la subcláusula porque

preceden, en orden de decodificación, a una imagen de IDR con `no_output_of_prior_pics_flag` igual a 1 o se infiere que es igual a 1, los tiempos de salida derivados de `dpb_output_delay` se pueden aumentar con un valor creciente de `PicOrderCnt()` relativo a todas las imágenes dentro de la misma secuencia de vídeo codificada.

5 `'num_decoding_units_minus1'` más 1 especifica el número de unidades de decodificación en la unidad de acceso con el que se asocia el mensaje de SEI de temporización de imagen. El valor de `num_decoding_units_minus1` puede estar en el intervalo de 0 a $\text{PicWidthInCtbs} * \text{PicHeightInCtbs} - 1$, inclusive.

`'common_du_cpb_removal_delay_flag'` igual a 1 especifica que el elemento de sintaxis `common_du_cpb_removal_delay` está presente.

10 `'common_du_cpb_removal_delay_flag'` igual a 0 especifica que el elemento de sintaxis `common_du_cpb_removal_delay` no está presente.

15 `'common_du_cpb_removal_delay'` especifica información de la siguiente manera: Si una unidad de decodificación es la primera unidad de decodificación en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen, entonces `common_du_cpb_removal_delay` especifica cuántos tictacs de reloj de imagen secundaria (véase la subcláusula E.2.1 de JCTVC-11003) esperar después de la eliminación del CPB de la última unidad de decodificación en la unidad de acceso asociada con el mensaje de SEI de período de almacenamiento temporal más reciente en una unidad de acceso anterior antes de eliminar del CPB la primera unidad de decodificación en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen.

20 De otro modo, `common_du_cpb_removal_delay` especifica cuántos tictacs de reloj de imagen secundaria (véase la subcláusula E.2.1 de JCTVC-11003) esperar después de la eliminación del CPB de la unidad de decodificación anterior en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen antes de eliminar del CPB la unidad de decodificación actual en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen. Este valor también se usa para calcular un tiempo de llegada más temprano posible de los datos de la unidad de decodificación al CPB para el HSS, como se especifica en el Anexo C. El elemento de sintaxis es un código de longitud fija cuya longitud en bits se da por $\text{cpb_removal_delay_length_minus1} + 1$. El `common_du_cpb_removal_delay` es el resto de un contador de módulo $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$.

Una forma alternativa de especificar `'common_du_cpb_removal_delay'` es de la siguiente manera.

30 `common_du_cpb_removal_delay` especifica cuántos tictacs de reloj de imagen secundaria (véase la subcláusula E.2.1 de JCTVC-11003) esperar después de la eliminación del CPB de la última unidad de decodificación antes de eliminar del CPB la unidad de decodificación actual en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen. Este valor también se usa para calcular un tiempo de llegada más temprano posible de los datos de la unidad de decodificación al CPB para el HSS, como se especifica en el Anexo C. El elemento de sintaxis es un código de longitud fija cuya longitud en bits se da por $\text{cpb_removal_delay_length_minus1} + 1$. El `common_du_cpb_removal_delay` es el resto de un contador de módulo $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$.

35 El valor de `cpb_removal_delay_length_minus1` que determina la longitud (en bits) del elemento de sintaxis `common_du_cpb_removal_delay` es el valor de `cpb_removal_delay_length_minus1` codificado en el conjunto de parámetros de secuencia que está activo para la imagen codificada asociada con el mensaje de SEI de temporización de imagen, aunque `common_du_cpb_removal_delay` especifica un número de tictacs de reloj de imagen secundaria en relación con el tiempo de eliminación de la primera unidad de decodificación en la unidad de acceso anterior que contiene un mensaje de SEI de período de almacenamiento temporal, que puede ser una
40 unidad de acceso de una secuencia de vídeo codificada diferente

`'num_nalus_in_du_minus1[i]'` más 1 especifica el número de unidades de NAL en la unidad de decodificación de orden *i* de la unidad de acceso con la que está asociado el mensaje de SEI de temporización de imagen. El valor de `num_nalus_in_du_minus1[i]` puede estar en el intervalo de 0 a $\text{PicWidthInCtbs} * \text{PicHeightInCtbs} - 1$, inclusive.

45 La primera unidad de decodificación de la unidad de acceso consta de las primeras `num_nalus_in_du_minus1[0] + 1` unidades de NAL consecutivas en orden de decodificación en la unidad de acceso. La unidad de decodificación de orden *i* (con *i* mayor que 0) de la unidad de acceso consiste en las `num_nalus_in_du_minus1[i] + 1` unidades de NAL consecutivas inmediatamente después de la última unidad de NAL en la unidad de decodificación anterior de la unidad de acceso, en orden de decodificación. Puede haber al menos una unidad de NAL de VCL en cada unidad de decodificación. Todas las unidades de NAL no VCL asociadas con una unidad de NAL de VCL se pueden incluir en
50 la misma unidad de decodificación.

55 `'du_cpb_removal_delay[i]'` especifica cuántos tictacs de reloj de imagen secundaria (véase la subcláusula E.2.1 de JCTVC-11003) esperar después de la eliminación del CPB de la primera unidad de decodificación en la unidad de acceso asociada con el mensaje de SEI de período de almacenamiento temporal más reciente en una unidad de acceso anterior antes de retirar del CPB la unidad de decodificación de orden *i* en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen. Este valor también se usa para calcular un tiempo de llegada más temprano posible de los datos de la unidad de decodificación al CPB para el HSS (por ejemplo, como se especifica en el Anexo C de JCTVC-11003). El elemento de sintaxis es un código de longitud fija cuya longitud en bits se da

por $\text{cpb_removal_delay_length_minus1} + 1$. El $\text{du_cpb_removal_delay}[i]$ es el resto de un contador de módulo $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$.

5 El valor de $\text{cpb_removal_delay_length_minus1}$ que determina la longitud (en bits) del elemento de sintaxis $\text{du_cpb_removal_delay}[i]$ es el valor de $\text{cpb_removal_delay_length_minus1}$ codificado en el conjunto de parámetros de secuencia que está activo para la imagen codificada asociada con el mensaje de SEI de temporización de imagen, aunque $\text{du_cpb_removal_delay}[i]$ especifica un número de ticks de reloj de imagen secundaria en relación con el tiempo de eliminación de la primera unidad de decodificación en la unidad de acceso anterior que contiene un mensaje de SEI de período de almacenamiento temporal, que puede ser una unidad de acceso de una secuencia de video codificada diferente.

10 En una configuración, la temporización de la eliminación de la unidad de decodificación y la decodificación de las unidades de decodificación se pueden implementar de la siguiente manera.

15 Si SubPicCpbFlag es igual a 0, la variable $\text{CpbRemovalDelay}(m)$ se establece en el valor de cpb_removal_delay en el mensaje de SEI de temporización de imagen asociado con la unidad de acceso que es la unidad de decodificación m , y la variable T_c se establece en t_{c_sub} . De otro modo, si SubPicCpbFlag es igual a 1 y $\text{common_du_cpb_removal_delay_flag}$ es 0, la variable $\text{CpbRemovalDelay}(m)$ se establece en el valor de $\text{du_cpb_removal_delay}[i]$ para la unidad de decodificación m (con m que oscila de 0 a $\text{num_decoding_units_minus1}$) en el mensaje de SEI de temporización de imagen asociado con la unidad de acceso que contiene la unidad de decodificación m , y la variable T_c se establece en t_{c_sub} .

20 En algunos casos, de otro modo, si SubPicCpbFlag es igual a 1 y $\text{common_du_cpb_removal_delay_flag}$ es 0, la variable $\text{CpbRemovalDelay}(m)$ se establece en el valor de $(m+1) * \text{du_cpb_removal_delay}[i]$ para la unidad de decodificación m (con m que oscila de 0 a $\text{num_decoding_units_minus1}$) en el mensaje de SEI de temporización de imagen asociado con la unidad de acceso que contiene la unidad de decodificación m , y la variable T_c se establece en t_{c_sub} .

25 De otro modo, si SubPicCpbFlag es igual a 1 y $\text{common_du_cpb_removal_delay_flag}$ es 1 la variable $\text{CpbRemovalDelay}(m)$ se establece en el valor de $\text{common_du_cpb_removal_delay}$ para la unidad de decodificación m en el mensaje de SEI de temporización de imagen asociado con la unidad de acceso que contiene la unidad de decodificación m , y la variable T_c se establece en t_{c_sub} .

30 Cuando una unidad de decodificación m es la unidad de decodificación con n igual a 0 (la primera unidad de decodificación de la unidad de acceso que inicializa el HRD), el tiempo nominal de eliminación de la unidad de decodificación del CPB se especifica por $t_{r,n}(0) = \text{InitCpbRemovalDelay}[\text{SchedSelIdx}]/90000$.

Cuando una unidad de decodificación m es la primera unidad de decodificación de la primera unidad de acceso de un período de almacenamiento temporal que no inicializa el HRD, el tiempo nominal de eliminación de la unidad de decodificación del CPB se especifica por $t_{r,n}(m) = t_{r,n}(m_b) + T_c * \text{CpbRemovalDelay}(m)$, donde $t_{r,n}(m_b)$ es el tiempo nominal de eliminación de la primera unidad de decodificación del período de almacenamiento temporal anterior.

35 Cuando una unidad de decodificación m es la primera unidad de decodificación de un período de almacenamiento temporal, m_b se establece igual a m en el tiempo de eliminación $t_{r,n}(m)$ de la unidad de decodificación m . El tiempo nominal de eliminación $t_{r,n}(m)$ de una unidad de decodificación m que no es la primera unidad de decodificación de un período de almacenamiento temporal se da por $t_{r,n}(m) = t_{r,n}(m_b) + T_c * \text{CpbRemovalDelay}(m)$, donde $t_{r,n}(m_b)$ es el tiempo nominal de eliminación de la primera unidad de decodificación del período de almacenamiento temporal anterior.

40 El tiempo de eliminación de la unidad de decodificación m se especifica de la siguiente manera. Si $\text{low_delay_hrd_flag}$ es igual a 0 o $t_{r,n}(m) \geq t_{af}(m)$, el tiempo de eliminación de la unidad de decodificación m se especifica por $t_r(m) = t_{r,n}(m)$. De otro modo ($\text{low_delay_hrd_flag}$ es igual a 1 y $t_{r,n}(m) < t_{af}(m)$), el tiempo de eliminación de la unidad de decodificación m se especifica por $t_r(m) = t_{r,n}(m) + T_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/T_c)$. En este último caso ($\text{low_delay_hrd_flag}$ es igual a 1 y $t_{r,n}(m) < t_{af}(m)$) indica que el tamaño de la unidad de decodificación m , $b(m)$ es tan grande que evita la eliminación en el tiempo nominal de eliminación.

45 En otro caso, el tiempo de eliminación de la unidad de decodificación m se especifica de la siguiente manera. Si $\text{low_delay_hrd_flag}$ es igual a 0 o $t_{r,n}(m) \geq t_{af}(m)$, el tiempo de eliminación de la unidad de decodificación m se especifica por $t_r(m) = t_{r,n}(m)$. De otro modo ($\text{low_delay_hrd_flag}$ es igual a 1 y $t_{r,n}(m) < t_{af}(m)$), el tiempo de eliminación de la unidad de decodificación m que no es la última unidad de decodificación en la unidad de acceso se especifica por $t_r(m) = t_{af}(m)$, y el tiempo de eliminación de la unidad de decodificación m que es la última unidad de decodificación en la unidad de acceso $t_r(m) = t_{r,n}(m) + T_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/T_c)$. En este último caso ($\text{low_delay_hrd_flag}$ es igual a 1 y $t_{r,n}(m) < t_{af}(m)$) indica que el tamaño de la unidad de decodificación m , $b(m)$, es tan grande que evita la eliminación en el tiempo nominal de eliminación.

55 En otro caso, el tiempo de eliminación de la unidad de decodificación m se especifica de la siguiente manera. Si $\text{low_delay_hrd_flag}$ es igual a 0 o $t_{r,n}(m) \geq t_{af}(m)$, el tiempo de eliminación de la unidad de decodificación m se especifica por $t_r(m) = t_{r,n}(m)$. De otro modo ($\text{low_delay_hrd_flag}$ es igual a 1 y $t_{r,n}(m) < t_{af}(m)$), el tiempo de

eliminación de la unidad de decodificación m que no es la última unidad de decodificación en la unidad de acceso se especifica por $t_r(m) = t_{af}(m)$, y el tiempo de eliminación de la unidad de decodificación m, que es la última unidad de decodificación en la unidad de acceso $t_r(m) = t_{r,n}(m) + t_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/t_c)$. Este último caso (low_delay_hrd_flag es igual a 1 y $t_{r,n}(m) < t_{af}(m)$) indica que el tamaño de la unidad de decodificación m, b(m), es tan grande que evita la eliminación en el tiempo nominal de eliminación.

5

En otro caso, el tiempo de eliminación de la unidad de decodificación m se especifica de la siguiente manera. Si low_delay_hrd_flag es igual a 0 o $t_{r,n}(m) \geq t_{af}(m)$, el tiempo de eliminación de la unidad de decodificación m se especifica por $t_r(m) = t_{r,n}(m)$. De otro modo (low_delay_hrd_flag es igual a 1 y $t_{r,n}(m) < t_{af}(m)$), el tiempo de eliminación de la unidad de decodificación m que se especifica por $t_r(m) = t_{af}(m)$. Este último caso (low_delay_hrd_flag es igual a 1 y $t_{r,n}(m) < t_{af}(m)$) indica que el tamaño de la unidad de decodificación m, b(m), es tan grande que evita la eliminación en el tiempo nominal de eliminación.

10

Cuando SubPicCpbFlag es igual a 1, el tiempo de eliminación de CPB nominal de la unidad de acceso n $t_{r,n}(n)$ se establece en el tiempo de eliminación del CPB nominal de la última unidad de decodificación en la unidad de acceso n, el tiempo de eliminación del CPB de la unidad de acceso n $t_r(n)$ se establece en el tiempo de eliminación de CPB de la última unidad de decodificación en la unidad de acceso n.

15

Cuando SubPicCpbFlag es igual a 0, cada unidad de decodificación es una unidad de acceso, por lo tanto, el tiempo de eliminación del CPB nominal y el tiempo de eliminación del CPB de la unidad de acceso n son el tiempo de eliminación del CPB nominal y el tiempo de eliminación del CPB de la unidad de decodificación n.

En el tiempo de eliminación del CPB de la unidad de decodificación m, la unidad de decodificación se decodifica instantáneamente.

20

Otro ejemplo de sintaxis y semántica modificadas para un mensaje de SEI de temporización de imagen según los sistemas y métodos descritos en la presente memoria se da en la Tabla (2) de la siguiente manera. Las modificaciones según los sistemas y métodos descritos en la presente memoria se denotan en negrita.

[Tabla 2]

<code>pic_timing(payloadSize) {</code>
<code> if(CpbDpbDelaysPresentFlag) {</code>
<code> cpb_removal_delay</code>
<code> dpb_output_delay</code>
<code> if(sub_pic_cpb_params_present_flag) {</code>
<code> num_decoding_units_minus1</code>
<code> common_du_cpb_removal_delay_flag</code>
<code> if(common_du_cpb_removal_delay_flag) {</code>
<code> common_num_nalus_in_du_minus1</code>
<code> common_du_cpb_removal_delay</code>
<code> }</code>
<code> for(i = 0; i <= num_decoding_units_minus1; i++) {</code>
<code> num_nalus_in_du_minus1[i]</code>
<code> if(!common_du_cpb_removal_delay_flag)</code>
<code> du_cpb_removal_delay[i]</code>
<code> }</code>
<code> }</code>
<code> }</code>
<code>}</code>

25

Tabla 2

El ejemplo ilustrado en la Tabla (2) incluye un elemento de sintaxis `common_num_nalus_in_du_minus1`, que se puede usar para determinar cuántos datos se deberían eliminar del CPB cuando se elimina una unidad de decodificación. 'common_num_nalus_in_du_minus 1' más 1 especifica el número de unidades de NAL en cada unidad de decodificación de la unidad de acceso con el que está asociado el mensaje de SEI de temporización de imagen. El valor de `common_num_nalus_in_du_minus1` puede estar en el intervalo de 0 a `PicWidthInCtbs * PicHeightInCtbs - 1`, inclusive.

30

La primera unidad de decodificación de la unidad de acceso consta de las primeras `common_num_nalus_in_du_minus1 + 1` unidades de NAL consecutivas en orden de decodificación en la unidad de acceso. La unidad de decodificación de orden i (con i mayor que 0) de la unidad de acceso consiste en las `common_num_nalus_in_du_minus1 + 1` unidades de NAL consecutivas inmediatamente después de la última unidad

35

de NAL en la unidad de decodificación anterior de la unidad de acceso, en orden de decodificación. Puede haber al menos una unidad de NAL de VCL en cada unidad de decodificación. Todas las unidades de NAL no VCL asociadas con una unidad de NAL de VCL se pueden incluir en la misma unidad de decodificación.

5 Otro ejemplo de sintaxis y semántica modificadas para un mensaje de SEI de temporización de imagen según los sistemas y métodos descritos en la presente memoria, se da en la Tabla (3) de la siguiente manera. Las modificaciones según los sistemas y métodos descritos en la presente memoria se denotan en negrita.

[Tabla 3]

<code>pic_timing(payloadSize) {</code>
<code> if(CpbDpbDelaysPresentFlag) {</code>
<code> cpb_removal_delay</code>
<code> dpb_output_delay</code>
<code> if(sub_pic_cpb_params_present_flag) {</code>
<code> num_decoding_units_minus1</code>
<code> common_num_nalus_in_du_flag</code>
<code> if(common_num_nalus_in_du_flag) {</code>
<code> common_num_nalus_in_du_minus1</code>
<code> }</code>
<code> common_du_cpb_removal_delay_flag</code>
<code> if(common_du_cpb_removal_delay_flag) {</code>
<code> common_du_cpb_removal_delay</code>
<code> }</code>
<code> for(i = 0; i <= num_decoding_units_minus1; i++) {</code>
<code> if(!common_num_nalus_in_du_flag)</code>
<code> num_nalus_in_du_minus1[i]</code>
<code> if(!common_du_cpb_removal_delay_flag)</code>
<code> du_cpb_removal_delay[i]</code>
<code> }</code>
<code> }</code>
<code> }</code>
<code>}</code>

Tabla 3

10 El ejemplo ilustrado en la Tabla (3) incluye un elemento de sintaxis 'common_num_nalus_in_du_flag' que, cuando es igual a 1, especifica que está presente el elemento de sintaxis 'common_num_nalus_in_du_minus1'. 'common_num_nalus_in_du_flag' igual a 0 especifica que no está presente el elemento de sintaxis 'common_num_nalus_in_du_minus1'.

15 En otra realización más, las banderas **common_du_cpb_removal_delay_flag** y **common_num_nalus_in_du_minus1**, no se pueden enviar. En su lugar, los elementos de sintaxis **common_num_nalus_in_du_minus1** y **common_du_cpb_removal_delay** se podrían enviar cada vez. En este caso, un valor de 0 (o algún otro) para estos elementos de sintaxis se podría usar para indicar que estos elementos no están señalados.

20 Además de modificaciones a los elementos de sintaxis y semántica del mensaje de SEI de temporización de imagen, los presentes sistemas y métodos también pueden implementar una restricción de flujo de bits de modo que la operación del CPB basada en imagen secundaria y la operación del CPB a nivel de unidad de acceso den como resultado la misma temporización de eliminación de unidad de decodificación.

25 Cuando **sub_pic_cpb_params_present_flag** es igual a 1 especifica que los parámetros de retardo de eliminación del CPB a nivel de imagen secundaria están presentes, el CPB puede operar a nivel de unidad de acceso o a nivel de imagen secundaria. **sub_pic_cpb_params_present_flag** igual a 0 especifica que los parámetros de retardo de eliminación del CPB a nivel de imagen secundaria no están presentes y que el CPB opera a nivel de unidad de acceso. Cuando **sub_pic_cpb_params_present_flag** no está presente, se infiere que su valor es igual a 0.

30 Para soportar la operación tanto a nivel de la unidad de acceso como a nivel de imagen secundaria, se pueden usar las siguientes restricciones del flujo de bits: Si **sub_pic_cpb_params_present_flag** es 1, entonces es requisito de la conformidad del flujo de bits que se cumpla la siguiente restricción cuando se señalan los valores para **cpb_removal_delay** y **du_cpb_removal_delay[i]** para todo i:

[Expresión matemática 3]

$$\text{cpb_removal_delay} = \frac{\sum_{i=0}^{\text{num_decoding_units_minus1}} \text{du_cpb_removal_delay}[i] * t_{c,sub}}{t_c}$$

donde du_cpb_removal_delay[i] son los parámetros de retardo del CPB de la unidad de decodificación, t_c es un tictac de reloj, t_{c,sub} es un tictac de reloj de imagen secundaria, num_decoding_units_minus1 es una cantidad de unidades de decodificación en la unidad de acceso desplazada en uno, e i un índice. En algunas realizaciones, se podría añadir un parámetro de tolerancia para satisfacer la restricción anterior.

Para soportar la operación tanto a nivel de la unidad de acceso como a nivel de imagen secundaria, se pueden usar las restricciones de flujo de bits de la siguiente manera: Permitamos que la variable T_{du}(k) se defina como:

[Expresión matemática 4]

$$T_{du}(k) = T_{du}(k-1) + t_{c,sub} * \sum_{i=0}^{\text{num_decoding_units_minus1}_k} (\text{du_cpb_removal_delay_minus1}_k[i] + 1)$$

donde du_cpb_removal_delay_minus1_k[i] y num_decoding_units_minus1_k son parámetros para la unidad de decodificación de orden l de la unidad de acceso de orden k (con k=0 para la unidad de acceso que inicializó el HRD y T_{du}(k) = 0 para k<1), y donde du_cpb_removal_delay_minus1_k[i]+1 = du_cpb_removal_delay_minus1_k[i] es el parámetro de retardo de eliminación del CPB de la unidad de decodificación para la unidad de decodificación de orden l de la unidad de acceso de orden k, y num_decoding_units_minus1_k es el número de unidades de decodificación en la unidad de acceso de orden k, t_c es un tictac de reloj, t_{c,sub} es un tictac de reloj de imagen secundaria, e i y k son índices. Entonces cuando la bandera de temporización de imagen (por ejemplo, sub_pic_cpb_params_present_flag) se establece en 1, la siguiente restricción será cierta: (au_cpb_removal_delay_minus1 + 1) * t_c == T_{du}(k), donde (au_cpb_removal_delay_minus1 + 1) = cpb_removal_delay, el retardo de eliminación del CPB. De este modo, en este caso, el retardo de eliminación del CPB (au_cpb_removal_delay_minus1 + 1) se establece de manera que la operación de CPB basada en imagen secundaria y la operación del CPB basada en la unidad de acceso den como resultado la misma temporización de eliminación de la unidad de acceso y la última unidad de decodificación de la eliminación de la unidad de acceso.

Para soportar la operación tanto a nivel de la unidad de acceso como a nivel de imagen secundaria, se pueden usar las siguientes restricciones del flujo de bits: Si sub_pic_cpb_params_present_flag es 1, entonces es requisito de la conformidad del flujo de bits que se cumpla la siguiente restricción cuando se señalan los valores para cpb_removal_delay y du_cpb_removal_delay[i] para todo i:

[Expresión matemática 5]

$$-1 \leq [\text{cpb_removal_delay} * t_c - (\sum_{i=0}^{\text{num_decoding_units_minus1}} \text{du_cpb_removal_delay}[i] * t_{c,sub})] \leq 1$$

donde du_cpb_removal_delay[i] son los parámetros de retardo de eliminación del CPB de la unidad de decodificación, t_c es un tictac de reloj, t_{c,sub} es un tictac de reloj de imagen secundaria, num_decoding_units_minus1 es una cantidad de unidades de decodificación en la unidad de acceso desplazada en uno, e i es un índice.

Para soportar la operación tanto a nivel de unidad de acceso como a nivel de imagen secundaria, se pueden usar las siguientes restricciones del flujo de bits: Si sub_pic_cpb_params_present_flag es 1, entonces es un requisito de la conformidad del flujo de bits que se cumpla la siguiente restricción cuando se señalan los valores para cpb_removal_delay y du_cpb_removal_delay[num_decoding_units_minus1]: cpb_removal_delay * t_c = du_cpb_removal_delay[num_decoding_units_minus1] * t_{c,sub} donde du_cpb_removal_delay[num_decoding_units_minus1] es el parámetro de retardo de eliminación del CPB de la unidad de decodificación para la unidad de decodificación de orden num_decoding_units_minus1, t_c es un tictac de reloj, t_{c,sub} es un tictac de reloj de imagen secundaria, num_decoding_units_minus1 es una cantidad de unidades de decodificación en la unidad de acceso desplazada en uno. En algunas realizaciones, se podría añadir un parámetro de tolerancia para satisfacer la restricción anterior.

Para soportar la operación tanto a nivel de la unidad de acceso como a nivel de imagen secundaria, se pueden usar las siguientes restricciones del flujo de bits: Si sub_pic_cpb_params_present_flag es 1, entonces es un requisito de la conformidad del flujo de bits que se cumpla la siguiente restricción cuando se señalan los valores para cpb_removal_delay y du_cpb_removal_delay[i] para todo i: -1 <= (cpb_removal_delay * t_c -

5 $du_cpb_removal_delay[num_decoding_units_minus1]*t_{c,sub} \leq 1$ donde
 $du_cpb_removal_delay[num_decoding_units_minus1]$ es el parámetro de retardo de eliminación del CPB de la
 unidad de decodificación para la unidad de decodificación de orden $num_decoding_units_minus1$, t_c es un tictac de
 reloj, $t_{c,sub}$ es un tictac de reloj de imagen secundaria, $num_decoding_units_minus1$ es una cantidad de unidades de
 decodificación en la unidad de acceso desplazada en uno.

10 Además, los presentes sistemas y métodos pueden modificar la temporización de eliminación de la unidad de
 decodificación. Cuando están presentes parámetros de retardo de eliminación del CPB a nivel de imagen
 secundaria, el tiempo de eliminación de la unidad de decodificación para “imágenes grandes” (cuando
 $low_delay_hrd_flag$ es 1 y $t_{r,n}(m) < t_{af}(m)$) se puede cambiar para compensar la diferencia que pueda surgir debido al
 contador de tictac de reloj y el contador de marca de reloj de imagen secundaria.

15 Cuando $sub_pic_cpb_params_present_flag$ es igual a 1, entonces están presentes los parámetros de retardo de
 eliminación del CPB a nivel de imagen secundaria y el CPB puede operar a nivel de unidad de acceso o a nivel de
 imagen secundaria. $sub_pic_cpb_params_present_flag$ igual a 0 especifica que los parámetros de retardo de
 eliminación del CPB a nivel de imagen secundaria no están presentes y que el CPB opera a nivel de unidad de
 acceso. Cuando $sub_pic_cpb_params_present_flag$ no está presente, se infiere que su valor es igual a 0.

20 Específicamente, un ejemplo de temporización de eliminación de unidad de decodificación y decodificación de
 implementación de unidades de decodificación es de la siguiente manera. La variable $SubPicCpbPreferredFlag$ o
 bien se especifica por medios externos, o bien cuando no se especifica por medios externos, se establece en 0. La
 variable $SubPicCpbFlag$ se deriva de la siguiente manera: $SubPicCpbFlag = SubPicCpbPreferredFlag \ \&\&$
 $sub_pic_cpb_params_present_flag$. Si $SubPicCpbFlag$ es igual a 0, el CPB opera a nivel de unidad de acceso y
 cada unidad de decodificación es una unidad de acceso. De otro modo, el CPB opera a nivel de imagen secundaria
 y cada unidad de decodificación es un subconjunto de una unidad de acceso.

25 Si $SubPicCpbFlag$ es igual a 0, la variable $CpbRemovalDelay(m)$ se establece en el valor de $cpb_removal_delay$ en
 el mensaje de SEI de temporización de imagen asociado con la unidad de acceso que es la unidad de decodificación
 m , y la variable T_c se establece en t_c . De otro modo, la variable $CpbRemovalDelay(m)$ se establece en el valor de
 $du_cpb_removal_delay[i]$ para la unidad de decodificación m en el mensaje de SEI de temporización de imagen
 asociado con la unidad de acceso que contiene la unidad de decodificación m , y la variable T_c se establece en t_{c_sub} .

30 Cuando una unidad de decodificación m es la unidad de decodificación con n igual a 0 (la primera unidad de
 decodificación de la unidad de acceso que inicializa el HRD), el tiempo nominal de eliminación de la unidad de
 decodificación del CPB se especifica por $t_{r,n}(0) = InitCpbRemovalDelay[SchedSelIdx]/90000$.

35 Cuando una unidad de decodificación m es la primera unidad de decodificación de la primera unidad de acceso de
 un período de almacenamiento temporal que no inicializa el HRD, el tiempo nominal de eliminación de la unidad de
 decodificación del CPB se especifica por $t_{r,n}(m) = t_{r,n}(m_b) + T_c * CpbRemovalDelay(m)$ donde $t_{r,n}(m_b)$ es el tiempo
 nominal de eliminación de la primera unidad de decodificación del período de almacenamiento temporal anterior.

40 Cuando una unidad de decodificación m es la primera unidad de decodificación de un período de almacenamiento
 temporal, m_b se establece igual a m en el tiempo de eliminación $t_{r,n}(m)$ de la unidad de decodificación m .

45 El tiempo nominal de eliminación $t_{r,n}(m)$ de una unidad de decodificación m que no es la primera unidad de
 decodificación de un período de almacenamiento temporal se da por $t_{r,n}(m) = t_{r,n}(m_b) + T_c * CpbRemovalDelay(m)$
 donde $t_{r,n}(m_b)$ es el tiempo nominal de eliminación de la primera unidad de decodificación del período de
 almacenamiento temporal actual.

El tiempo de eliminación de la unidad de decodificación m se especifica de la siguiente manera. La variable $ClockDiff$
 se define como $ClockDiff = (num_units_in_tick - (num_units_in_sub_tick * (num_decoding_units_minus\ 1 + 1))) /$
 $time_scale$). En algún caso, puede ser un requisito de una conformidad del flujo de bits que los parámetros
 $num_units_in_tick$, $num_units_in_sub_tick$, $num_decoding_units_minus1$ se señalen de manera que se satisfaga la
 siguiente ecuación.

$$(num_units_in_tick - (num_units_in_sub_tick * (num_decoding_units_minus1 + 1))) >= 0$$

En algún otro caso, puede ser un requisito de una conformidad del flujo de bits que los parámetros
 $num_units_in_tick$, $num_units_in_sub_tick$, $num_decoding_units_minus1$ se puede señalar de manera que se
 satisfaga la siguiente ecuación.

$$(num_units_in_tick - (num_units_in_sub_tick * (num_decoding_units_minus1 + 1))) <= 0$$

50 Si $low_delay_hrd_flag$ es igual a 0 o $t_{r,n}(m) >= t_{af}(m)$, el tiempo de eliminación de la unidad de decodificación m se
 especifica por $t_r(m) = t_{r,n}(m)$.

De otro modo ($low_delay_hrd_flag$ es igual a 1 y $t_{r,n}(m) < t_{af}(m)$), y cuando $sub_pic_cpb_params_present_flag$ es
 igual a 1 y el CPB está operando a nivel de imagen secundaria, y si $ClockDiff$ es mayor que cero, el tiempo de

eliminación de la unidad de decodificación m cuando es la última unidad de decodificación de la unidad de acceso n se especifica por $t_r(m) = t_{r,n}(m) + T_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / T_c) + \text{ClockDiff}$.

5 De otro modo (low_delay_hrd_flag es igual a 1 y $t_{r,n}(m) < t_{af}(m)$), y cuando sub_pic_cpb_params_present_flag es igual a 1 y el CPB está operando a nivel de unidad de acceso y, si ClockDiff es menor que cero, el tiempo de eliminación de la unidad de acceso n se especifica por $t_r(m) = t_{r,n}(m) + t_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/t_c) - \text{ClockDiff}$.

De otro modo (low_delay_hrd_flag es igual a 1 y $t_{r,n}(m) < t_{af}(m)$), el tiempo de eliminación de la unidad de decodificación m se especifica por $t_r(m) = t_{r,n}(m) + T_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/T_c)$. Este último caso (low_delay_hrd_flag es igual a 1 y $t_{r,n}(m) < t_{af}(m)$) indica que el tamaño de la unidad de decodificación m, b(m), es tan grande que impide la eliminación en el tiempo nominal de eliminación.

10 De otro modo (low_delay_hrd_flag es igual a 1 y $t_{r,n}(m) < t_{af}(m)$), y cuando una bandera de temporización de imagen se establece en 1 y el CPB está operando a nivel de imagen secundaria, el tiempo de eliminación de la última unidad de decodificación m de unidad de acceso, $t_r(m)$ según: $t_r(m) = t_{r,n}(m) + \min((t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c)))$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la última unidad de decodificación m, t_{c_sub} es un tictac de reloj de imagen secundaria, Ceil() es una función de techo, $t_{af}(m)$ es un tiempo de llegada final de la última unidad de decodificación m, $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n, t_c es un tictac de reloj y $t_{af}(n)$ es el tiempo de llegada final de la unidad de acceso n.

20 De otro modo (low_delay_hrd_flag es igual a 1 y $t_{r,n}(n) < t_{af}(n)$) y cuando una bandera de temporización de imagen se establece en 1 y el CPB está operando a nivel de unidad de acceso, el tiempo de eliminación para la unidad de acceso n, $t_r(n)$ según: $t_r(n) = t_{r,n}(n) + \min((t_{c_sub} * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c)))$ donde $t_{r,n}(n)$ es el tiempo nominal de eliminación de la última unidad de decodificación n, t_{c_sub} es un tictac de reloj de imagen secundaria, Ceil() es una función de techo, $t_{af}(n)$ es el tiempo de llegada final de la última unidad de decodificación n, $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n, t_c es un tictac de reloj y $t_{af}(n)$ es un tiempo de llegada final de la unidad de acceso n.

25 De otro modo (low_delay_hrd_flag es igual a 1 y $t_{r,n}(m) < t_{af}(m)$) y una bandera de temporización de imagen se establece en 1 y el CPB está operando a nivel de imagen secundaria, el tiempo de eliminación para la última unidad de decodificación m de la unidad de acceso, $t_r(m)$ según: $t_r(m) = t_{r,n}(m) + (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c))$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la última unidad de decodificación m, t_{c_sub} es un tictac de reloj de imagen secundaria, Ceil() es una función de techo, $t_{af}(m)$ es un tiempo de llegada final de la última unidad de decodificación m, $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n, t_c es un tictac de reloj y $t_{af}(n)$ es el tiempo de llegada final de la unidad de acceso n.

30 De otro modo (low_delay_hrd_flag es igual a 1 y $t_{r,n}(n) < t_{af}(n)$) y una bandera de temporización de imagen se establece en 1 y el CPB está operando a nivel de unidad de acceso, el tiempo de eliminación para la unidad de acceso n, $t_r(n)$ según: $t_r(n) = t_{r,n}(n) + (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c))$ donde $t_{r,n}(n)$ es el tiempo nominal de eliminación de la última unidad de decodificación n, t_{c_sub} es un tictac de reloj de imagen secundaria, Ceil() es una función de techo, $t_{af}(n)$ es un tiempo de llegada final de la última unidad de decodificación n, $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n, t_c es un tictac de reloj y $t_{af}(n)$ es un tiempo de llegada final de la unidad de acceso n.

40 De otro modo (low_delay_hrd_flag es igual a 1 y $t_{r,n}(m) < t_{af}(m)$) y una bandera de temporización de imagen se establece en 1 y el CPB está operando a nivel de imagen secundaria, el tiempo de eliminación para la unidad de decodificación que no es la última unidad de decodificación de la unidad de acceso se establece como $t_r(m) = t_{af}(m)$, donde $t_{af}(m)$ es un tiempo de llegada final de la unidad de decodificación m. Y el tiempo de eliminación de la última unidad de decodificación m de la unidad de acceso, $t_r(m)$ se establece según: $t_r(m) = t_{r,n}(m) + (t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/t_{c_sub}))$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la última unidad de decodificación m, t_{c_sub} es un tictac de reloj de imagen secundaria, Ceil() es una función de techo, $t_{af}(m)$ es un tiempo de llegada final de la última unidad de decodificación m, $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n, t_c es un tictac de reloj y $t_{af}(n)$ es el tiempo de llegada final de la unidad de acceso n y $t_{af}(m)$ es el tiempo de llegada final de la última unidad de decodificación m en la unidad de acceso n.

50 De otro modo (low_delay_hrd_flag es igual a 1 y $t_{r,n}(m) < t_{af}(m)$) y una bandera de temporización de imagen se establece en 1 y el CPB está operando a nivel de imagen secundaria, el tiempo de eliminación para la unidad de decodificación que no es la última unidad de decodificación de la unidad de acceso se establece como $t_r(m) = t_{af}(m)$, donde $t_{af}(m)$ es un tiempo de llegada final de la unidad de decodificación m. Y el tiempo de eliminación de la última unidad de decodificación m de la unidad de acceso, $t_r(m)$ se establece según: $t_r(m) = t_{r,n}(m) + (t_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/t_c))$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la última unidad de decodificación m, t_{c_sub} es un tictac de reloj de imagen secundaria, Ceil() es una función de techo, $t_{af}(m)$ es un tiempo de llegada final de la última unidad de decodificación m, $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n, t_c es un tictac de reloj y $t_{af}(n)$ es un tiempo de llegada final de la unidad de acceso n y $t_{af}(m)$ es un tiempo de llegada final de la última unidad de decodificación m en la unidad de acceso n.

De otro modo (low_delay_hrd_flag es igual a 1 y $t_{r,n}(m) < t_{af}(m)$) y una bandera de temporización de imagen se

establece en 1 y el CPB está operando a nivel de imagen secundaria, el tiempo de eliminación para la unidad de decodificación se establece como $t_r(m) = t_{ar}(m)$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la unidad de decodificación m , t_{c_sub} es un tictac de reloj de imagen secundaria, $Ceil()$ es una función de techo, $t_{ar}(m)$ es un tiempo de llegada final de la unidad de decodificación m , $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n , t_c es un tictac de reloj y $t_{ar}(n)$ es el tiempo de llegada final de la unidad de acceso n y $t_{ar}(m)$ es un tiempo de llegada final de la unidad de decodificación m en la unidad de acceso n .

De otro modo ($low_delay_hrd_flag$ es igual a 1 y $t_{r,n}(n) < t_{ar}(n)$) y una bandera de temporización de imagen se establece en 1 y el CPB está operando a nivel de unidad de acceso, el tiempo de eliminación de la unidad de acceso n , $t_r(n)$ según: $t_r(n) = t_{ar}(n)$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la última unidad de decodificación n , t_{c_sub} es un tictac de reloj de imagen secundaria, $Ceil()$ es una función de techo, $t_{ar}(m)$ es un tiempo de llegada final de la última unidad de decodificación m , $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n , t_c es un tictac de reloj y $t_{ar}(n)$ es el tiempo de llegada final de la unidad de acceso n .

Cuando $SubPicCpbFlag$ es igual a 1, el tiempo de eliminación del CPB nominal de la unidad de acceso n $t_{r,n}(n)$ se establece en el tiempo de eliminación del CPB nominal de la última unidad de decodificación en la unidad de acceso n , el tiempo de eliminación del CPB de la unidad de acceso n $t_r(n)$ se establece en el tiempo de eliminación del CPB de la última unidad de decodificación en la unidad de acceso n .

Cuando $SubPicCpbFlag$ es igual a 0, cada unidad de decodificación es una unidad de acceso, por lo tanto, el tiempo de eliminación del CPB nominal y el tiempo de eliminación del CPB de la unidad de acceso n son el tiempo de eliminación del CPB nominal y el tiempo de eliminación del CPB de la unidad de decodificación n . En el tiempo de eliminación del CPB de la unidad de decodificación m , la unidad de decodificación se decodifica instantáneamente.

Como se ilustra por lo precedente, los sistemas y métodos descritos en la presente memoria proporcionan una sintaxis y una semántica que modifican un flujo de bits de mensaje de SEI de temporización de imagen que llevan parámetros basados en imagen secundaria. En algunas configuraciones, los sistemas y métodos descritos en la presente memoria se pueden aplicar a las especificaciones de HEVC.

Por conveniencia, se dan varias definiciones de la siguiente manera, que se pueden aplicar a los sistemas y métodos descritos en la presente memoria. Un punto de acceso aleatorio puede ser cualquier punto en un flujo de datos (por ejemplo, flujo de bits) donde la decodificación del flujo de bits no requiere acceso a ningún punto en un flujo de bits que precede al punto de acceso aleatorio para decodificar una imagen actual y todas las imágenes posteriores a dicha imagen actual en orden de salida.

Se puede especificar un período de almacenamiento temporal como un conjunto de unidades de acceso entre dos instancias del mensaje de SEI del período de almacenamiento temporal en orden de decodificación. La Información de Mejora Suplementaria (SEI) puede contener información que no es necesaria para decodificar las muestras de imágenes codificadas de las unidades de NAL de VCL. Los mensajes de SEI pueden ayudar en procedimientos relacionados con la decodificación, visualización u otros propósitos. Puede que no se requiera que los decodificadores conformes procesen esta información para orden de salida conforme a las especificaciones de HEVC (Anexo C de las especificaciones de HEVC (JCTVC-11003) incluye especificaciones de conformidad, por ejemplo). Alguna información del mensaje de SEI se puede usar para comprobar la conformidad del flujo de bits y para conformidad del decodificador de temporización de salida.

Un mensaje de SEI de período de almacenamiento temporal puede ser un mensaje de SEI relacionado con el período de almacenamiento temporal. Un mensaje de SEI de temporización de imagen puede ser un mensaje de SEI relacionado con temporización de eliminación de CPB. Estos mensajes pueden definir la sintaxis y la semántica que definen la temporización de llegada del flujo de bits y la temporización de eliminación de imágenes codificadas.

Un Almacenador Temporal de Imágenes Codificadas (CPB) puede ser un almacenador temporal de tipo primero en entrar, primero en salir que contiene unidades de acceso en el orden de decodificación especificado en un decodificador de referencia hipotético (HRD). Una unidad acceso puede ser un conjunto de unidades de Capa de Acceso a la Red (NAL) que son consecutivas en orden de decodificación y contienen exactamente una imagen codificada. Además de las unidades de NAL de porciones codificadas, la unidad de acceso también puede contener otras unidades de NAL que no contengan porciones de la imagen codificada. La decodificación de una unidad de acceso siempre da como resultado una imagen decodificada. Una unidad de NAL puede ser una estructura de sintaxis que contiene una indicación del tipo de datos a seguir y los bytes que contienen esos datos en forma de una carga útil de secuencia de bytes en bruto intercalada según sea necesario con los bytes de prevención de emulación.

Como se usa en la presente memoria, el término "común" se refiere generalmente a un elemento de sintaxis o una variable que es aplicable a más de una cosa. Por ejemplo, en el contexto de elementos de sintaxis en un mensaje de SEI de temporización de imagen, el término "común" puede significar que el elemento de sintaxis (por ejemplo, `common_du_cpb_removal_delay`) es aplicable a todas las unidades de decodificación en una unidad de acceso asociada con el mensaje de SEI de temporización de imagen. Además, las unidades de datos se describen en términos de "n" y "m" generalmente se refieren a unidades de acceso y a unidades de decodificación,

respectivamente.

Ahora se describen diversas configuraciones con referencia a las Figuras, donde números de referencia similares pueden indicar elementos funcionalmente similares.

5 La Figura 1A es un diagrama de bloques que ilustra un ejemplo de uno o más dispositivos electrónicos 102 en los que se pueden implementar sistemas y métodos para enviar un mensaje y almacenar temporalmente un flujo de bits. En este ejemplo, se ilustran el dispositivo electrónico A 102a y el dispositivo electrónico B 102b. No obstante, se debería observar que una o más de las características y funcionalidades descritas en relación con el dispositivo electrónico A 102a y el dispositivo electrónico B 102b se pueden combinar en un único dispositivo electrónico en algunas configuraciones.

10 El dispositivo electrónico A 102a incluye un codificador 104. El codificador 104 incluye un módulo de generación de mensajes 108. Cada uno de los elementos incluidos dentro del dispositivo electrónico A 102a (por ejemplo, el codificador 104 y el módulo de generación de mensajes 108) se puede implementar en hardware, software o una combinación de ambos.

15 El dispositivo electrónico A 102a puede obtener una o más imágenes de entrada 106. En algunas configuraciones, la imagen o las imágenes de entrada 106 se pueden capturar en el dispositivo electrónico A 102a usando un sensor de imagen, se puede recuperar de la memoria y/o se puede recibir desde otro dispositivo electrónico.

20 El codificador 104 puede codificar la imagen o las imágenes de entrada 106 para producir datos codificados. Por ejemplo, el codificador 104 puede codificar una serie de imágenes de entrada 106 (por ejemplo, vídeo). En una configuración, el codificador 104 puede ser un codificador de HEVC. Los datos codificados pueden ser datos digitales (por ejemplo, parte de un flujo de bits 114). El codificador 104 puede generar una señalización de sobrecarga en base a la señal de entrada.

25 El módulo de generación de mensajes 108 puede generar uno o más mensajes. Por ejemplo, el módulo de generación de mensajes 108 puede generar uno o más mensajes de SEI u otros mensajes. Para un CPB que soporta operación a nivel de imagen secundaria, el dispositivo electrónico 102 puede enviar parámetros de imagen secundaria, (por ejemplo, el parámetro de retardo de eliminación del CPB). Específicamente, el dispositivo electrónico 102 (por ejemplo, el codificador 104) puede determinar si incluir un parámetro de retardo de eliminación del CPB de la unidad de decodificación común en un mensaje de SEI de temporización de imagen. Por ejemplo, el dispositivo electrónico puede establecer una bandera (por ejemplo, `common_du_cpb_removal_delay_flag`) en uno cuando el codificador 104 está incluyendo un parámetro de retardo de eliminación del CPB de la unidad de decodificación común (por ejemplo, `common_du_cpb_removal_delay`) en el mensaje de SEI de temporización de imagen. Cuando está incluido el parámetro de retardo de eliminación del CPB de la unidad de decodificación común, el dispositivo electrónico puede generar el parámetro de retardo de eliminación del CPB de la unidad de decodificación común que es aplicable a todas las unidades de decodificación en una unidad de acceso. En otras palabras, en lugar de incluir un parámetro de retardo de eliminación del CPB de la unidad de decodificación para cada unidad de decodificación en una unidad de acceso, puede aplicarse un parámetro común a todas las unidades de decodificación en la unidad de acceso con la que está asociado el mensaje de SEI de temporización de imagen.

30 Por el contrario, cuando el parámetro de retardo de eliminación del CPB de la unidad de decodificación común no ha de ser incluido en el mensaje de SEI de temporización de imagen, el dispositivo electrónico 102 puede generar un retardo de eliminación del CPB de la unidad de decodificación separado para cada unidad de decodificación en la unidad de acceso con la que está asociado el mensaje de SEI de temporización de imagen. Un módulo de generación de mensajes 108 puede realizar uno o más de los procedimientos descritos en conexión con la Figura 2 y la Figura 3 a continuación.

35 En algunas configuraciones, el dispositivo electrónico A 102a puede enviar el mensaje al dispositivo electrónico B 102b como parte del flujo de bits 114. En algunas configuraciones, el dispositivo electrónico A 102a puede enviar el mensaje al dispositivo electrónico B 102b mediante una transmisión 110 separada. Por ejemplo, la transmisión separada puede no ser parte del flujo de bits 114. Por ejemplo, un mensaje de SEI de temporización de imagen u otro mensaje se puede enviar usando algún mecanismo fuera de banda. Se debería observar que, en algunas configuraciones, el otro mensaje puede incluir una o más de las características de un mensaje de SEI de temporización de imagen descrito anteriormente. Además, el otro mensaje, en uno o más aspectos, se puede utilizar de manera similar al mensaje de SEI descrito anteriormente.

40 El codificador 104 (y el módulo de generación de mensajes 108, por ejemplo) puede producir un flujo de bits 114. El flujo de bits 114 puede incluir datos de imágenes codificadas basadas en la imagen o las imágenes de entrada 106. En algunas configuraciones, el flujo de bits 114 también puede incluir datos de sobrecarga, tales como un mensaje de SEI de temporización de imagen u otro mensaje, cabecera o cabeceras de porciones, PPS, etc. A medida que se codifican imágenes de entrada 106 adicionales, el flujo de bits 114 puede incluir una o más imágenes codificadas. Por ejemplo, el flujo de bits 114 puede incluir una o más imágenes codificadas con datos de sobrecarga correspondientes (por ejemplo, un mensaje de SEI de temporización de imagen u otro mensaje).

45 El flujo de bits 114 se puede proporcionar a un decodificador 112. En un ejemplo, el flujo de bits 114 se puede

transmitir al dispositivo electrónico B 102b usando un enlace cableado o inalámbrico. En algunos casos, esto se puede hacer sobre una red, tal como Internet o una Red de Área Local (LAN). Como se ilustra en la Figura 1A, el decodificador 112 se puede implementar en el dispositivo electrónico B 102b por separado del codificador 104 en el dispositivo electrónico A 102a. No obstante, se debería observar que el codificador 104 y el decodificador 112 se pueden implementar en el mismo dispositivo electrónico en algunas configuraciones. En una implementación donde el codificador 104 y el decodificador 112 se implementan en el mismo dispositivo electrónico, por ejemplo, el flujo de bits 114 se puede proporcionar sobre un bus al decodificador 112 o almacenar en la memoria para su recuperación por el decodificador 112.

El decodificador 112 se puede implementar en hardware, software o una combinación de ambos. En una configuración, el decodificador 112 puede ser un decodificador de HEVC. El decodificador 112 puede recibir (por ejemplo, obtener) el flujo de bits 114. El decodificador 112 puede generar una o más imágenes decodificadas 118 basadas en el flujo de bits 114. La imagen o las imágenes decodificadas 118 se pueden mostrar, reproducir, almacenar en la memoria y/o transmitir a otro dispositivo, etc.

El decodificador 112 puede incluir un CPB 120. El CPB 120 puede almacenar temporalmente imágenes codificadas. El CPB 120 puede usar los parámetros encontrados en un mensaje de SEI de temporización de imagen para determinar cuándo eliminar los datos. Cuando el CPB 120 soporta la operación a nivel de imagen secundaria, las unidades de decodificación individuales se pueden eliminar en lugar de acceso completas al mismo tiempo. El decodificador 112 puede incluir un Almacenador Temporal de Imágenes Decodificadas (DPB) 122. Cada imagen decodificada se coloca en el DPB 122 para ser referenciada por el proceso de decodificación así como para salida y recorte. Una imagen decodificada se elimina del DPB en el último del tiempo de salida del DPB o el tiempo que ya no llega a ser necesario para referencia entre predicciones.

El decodificador 112 puede recibir un mensaje (por ejemplo, el mensaje de SEI de temporización de imagen u otro mensaje). El decodificador 112 también puede determinar si el mensaje recibido incluye un parámetro de retardo de eliminación del CPB de la unidad de decodificación común (por ejemplo, `common_du_cpb_removal_delay`). Esto puede incluir la identificación de una bandera (por ejemplo, `common_du_cpb_removal_delay_flag`) que se establece cuando el parámetro común está presente en el mensaje de SEI de temporización de imagen. Si está presente el parámetro común, el decodificador 112 puede determinar el parámetro de retardo de eliminación del CPB de la unidad de decodificación común aplicable a todas las unidades de decodificación en la unidad de acceso. Si el parámetro común no está presente, el decodificador 112 puede determinar un parámetro de retardo de eliminación del CPB de la unidad de decodificación separado para cada unidad de decodificación en la unidad de acceso. El decodificador 112 también puede eliminar las unidades de decodificación del CPB 120 usando o bien el parámetro de retardo de eliminación del CPB de la unidad de decodificación común o bien los parámetros de retardo de eliminación del CPB de la unidad de decodificación separada. El CPB 120 puede realizar uno o más de los procedimientos descritos en conexión con la Figura 4 y la Figura 5 a continuación.

El HRD descrito anteriormente puede ser un ejemplo del decodificador 112 ilustrado en la Figura 1A. De este modo, un dispositivo electrónico 102 puede operar según el HRD y el CPB 120 y el DPB 122 descritos anteriormente, en algunas configuraciones.

Se debería observar que uno o más de los elementos o partes de los mismos incluidos en el dispositivo o dispositivos electrónicos 102 se pueden implementar en hardware. Por ejemplo, uno o más de estos elementos o partes de los mismos se pueden implementar como un circuito integrado, circuitería o componentes de hardware, etc. También se debería observar que una o más de las funciones o métodos descritos en la presente memoria se pueden implementar en y/o realizar usando hardware. Por ejemplo, uno o más de los métodos descritos en la presente memoria se pueden implementar en y/o realizar usando un conjunto de circuitos integrados, un Circuito Integrado de Aplicaciones Específicas (ASIC), un circuito Integrado a Gran Escala (LSI) o un circuito integrado, etc.

La Figura 1B es un diagrama de bloques que ilustra otro ejemplo de un codificador 1908 y un decodificador 1972. En este ejemplo, se ilustran el dispositivo electrónico A 1902 y el dispositivo electrónico B 1970. No obstante, se debería observar que las características y la funcionalidad descritas en relación con el dispositivo electrónico A 1902 y el dispositivo electrónico B 1970 se pueden combinar en un único dispositivo electrónico en algunas configuraciones.

El dispositivo electrónico A 1902 incluye el codificador 1908. El codificador 1908 puede incluir un codificador de capa base 1910 y un codificador de capa de mejora 1920. El codificador de vídeo 1908 es adecuado para codificación de vídeo escalable y codificación de vídeo de vista múltiple, como se describe más adelante. El codificador 1908 se puede implementar en hardware, software o una combinación de ambos. En una configuración, el codificador 1908 puede ser un codificador de codificación de vídeo de alta eficiencia (HEVC), incluyendo escalable y/o de vista múltiple. Se pueden usar del mismo modo otros codificadores. El dispositivo electrónico A 1902 puede obtener una fuente 1906. En algunas configuraciones, la fuente 1906 se puede capturar en el dispositivo electrónico A 1902 usando un sensor de imagen, recuperada de la memoria o recibida desde otro dispositivo electrónico.

El codificador 1908 puede codificar la fuente 1906 para producir un flujo de bits de capa base 1934 y un flujo de bits de capa de mejora 1936. Por ejemplo, el codificador 1908 puede codificar una serie de imágenes (por ejemplo, vídeo) en la fuente 1906. En particular, para codificación de vídeo escalable para la escalabilidad de SNR también

conocida como escalabilidad de calidad, la misma fuente 1906 se puede proporcionar al codificador de capa base y de capa de mejora. En particular, para codificación de vídeo escalable para escalabilidad espacial, se puede usar una fuente de submuestreo para el codificador de capa base. En particular, para la codificación de vista múltiple, se puede usar una fuente de vista diferente para el codificador de capa base y el codificador de capa de mejora. El codificador 1908 puede ser similar al codificador 1782 descrito más adelante en conexión con la Figura 6B.

Los flujos de bits 1934, 1936 pueden incluir datos de imágenes codificadas basados en la fuente 1906. En algunas configuraciones, los flujos de bits 1934, 1936 también pueden incluir datos de sobrecarga, tales como información de cabecera de porción, información de PPS, etc. Como las imágenes adicionales en la fuente 1906 están codificadas, los flujos de bits 1934, 1936 pueden incluir una o más imágenes codificadas.

Los flujos de bits 1934, 1936 se pueden proporcionar al decodificador 1972. El decodificador 1972 puede incluir un decodificador de capa base 1980 y un decodificador de capa de mejora 1990. El decodificador de vídeo 1972 es adecuado para decodificación de vídeo escalable y decodificación de vídeo de vista múltiple. En un ejemplo, los flujos de bits 1934, 1936 se pueden transmitir al dispositivo electrónico B 1970 usando un enlace cableado o inalámbrico. En algunos casos, esto se puede hacer sobre una red, tal como Internet o una Red de Área Local (LAN). Como se ilustra en la Figura 1B, el decodificador 1972 se puede implementar en el dispositivo electrónico B 1970 por separado del codificador 1908 en el dispositivo electrónico A 1902. No obstante, se debería observar que el codificador 1908 y el decodificador 1972 se pueden implementar en el mismo dispositivo electrónico en algunas configuraciones. En una implementación donde el codificador 1908 y el decodificador 1972 se implementan en el mismo dispositivo electrónico, por ejemplo, los flujos de bits 1934, 1936 se pueden proporcionar sobre un bus al decodificador 1972 o almacenar en la memoria para su recuperación por el decodificador 1972. El decodificador 1972 puede proporcionar una capa base decodificada 1992 y una imagen o imágenes de capa de mejora decodificada 1994 como salida.

El decodificador 1972 se puede implementar en hardware, software o una combinación de ambos. En una configuración, el decodificador 1972 puede ser un decodificador de codificación de vídeo de alta eficiencia (HEVC), incluyendo escalable y/o de vista múltiple. Se pueden usar del mismo modo otros decodificadores. El decodificador 1972 puede ser similar al decodificador 1812 descrito más tarde en conexión con la Figura 7B. También, el codificador de capa base y/o el codificador de capa de mejora pueden incluir cada uno un módulo de generación de mensajes, tal como el descrito en relación con la Figura 1A. También, el decodificador de capa base y/o el decodificador de capa de mejora pueden incluir un almacenador temporal de imágenes codificadas y/o un almacenador temporal de imágenes decodificadas, tal como el descrito en relación con la Figura 1A. Además, los dispositivos electrónicos de la Figura 1B pueden operar según las funciones de los dispositivos electrónicos de la Figura 1A, según sea aplicable.

La Figura 2 es un diagrama de flujo que ilustra una configuración de un método 200 para enviar un mensaje. El método 200 se puede realizar por un codificador 104 o una de sus subpartes (por ejemplo, un módulo de generación de mensajes 108). El codificador 104 puede determinar 202 una bandera de temporización de imagen (por ejemplo, `sub_pic_cpb_params_present_flag`) que indica si un CPB 120 admite la operación a nivel de imagen secundaria. Por ejemplo, cuando la bandera de temporización de imagen se establece en 1, el CPB 120 puede operar a nivel de unidad de acceso o a nivel de imagen secundaria. Se debería observar que incluso cuando la bandera de temporización de imagen se establece en 1, la decisión acerca de si realmente operar a nivel de imagen secundaria se deja al decodificador 112 en sí mismo.

El codificador 104 también puede determinar 204 uno o más retardos de eliminación para las unidades de decodificación en una unidad de acceso. Por ejemplo, el codificador 104 puede determinar un único parámetro de retardo de eliminación del CPB de la unidad de decodificación común (por ejemplo, `common_du_cpb_removal_delay`) que es aplicable a todas las unidades de decodificación en la unidad de acceso desde el CPB 120. Alternativamente, el codificador 104 puede determinar un retardo de eliminación del CPB de la unidad de decodificación separado (por ejemplo, `du_cpb_removal_delay[i]`) para cada unidad de decodificación en la unidad de acceso.

El codificador 104 también puede determinar 206 uno o más parámetros de NAL que indican una cantidad, desplazada en una, de unidades de NAL en cada unidad de decodificación en el punto de acceso. Por ejemplo, el codificador 104 puede determinar un único parámetro de NAL común (por ejemplo, `common_num_nalus_in_du_minus1`) que es aplicable a todas las unidades de decodificación en la unidad de acceso del CPB 120. Alternativamente, el codificador 104 puede determinar un retardo de eliminación del CPB de la unidad de decodificación separado (por ejemplo, `num_nalus_in_du_minus1[i]`) para cada unidad de decodificación en la unidad de acceso.

El codificador 104 también puede enviar 208 un mensaje de SEI de temporización de imagen que incluye la bandera de temporización de imagen, los retardos de eliminación y los parámetros de NAL. El mensaje de SEI de temporización de imagen también puede incluir otros parámetros (por ejemplo, `cpb_removal_delay`, `dpb_output_delay`, etc.). Por ejemplo, el dispositivo electrónico 102 puede transmitir el mensaje a través de uno o más de transmisión inalámbrica, transmisión por cable, bus de dispositivo, red, etc. Por ejemplo, el dispositivo electrónico A 102a puede transmitir el mensaje al dispositivo electrónico B 102b. El mensaje puede ser parte del flujo

de bits 114, por ejemplo. En algunas configuraciones, el dispositivo electrónico A 102a puede enviar 208 el mensaje al dispositivo electrónico B 102b en una transmisión 110 separada (que no forma parte del flujo de bits 114).

Por ejemplo, el mensaje puede ser enviado usando algún mecanismo fuera de banda. En algún caso, la información indicada en 204, 206 se puede enviar en un mensaje de SEI diferente al mensaje de SEI de temporización de imagen. En otro caso más, la información indicada en 204, 206 se puede enviar en un conjunto de parámetros, por ejemplo, un conjunto de parámetros de vídeo y/o conjunto de parámetros de secuencia y/o conjunto de parámetros de imagen y/o conjunto de parámetros de adaptación y/o cabecera de porción.

La Figura 3 es un diagrama de flujo que ilustra una configuración de un método 300 para determinar uno o más retardos de eliminación para unidades de decodificación en una unidad de acceso. En otras palabras, el método 300 ilustrado en la Figura 3 puede ilustrar además el paso 204 en el método 200 ilustrado en la Figura 2. El método 300 se puede realizar por un codificador 104. El codificador 104 puede determinar 302 si incluir un parámetro de retardo de eliminación del CPB de la unidad de decodificación común (por ejemplo, `common_du_cpb_removal_delay`).

Esto puede incluir determinar si se establece una bandera de retardo de eliminación del CPB de la unidad de decodificación común (por ejemplo, `common_du_cpb_removal_delay_flag`). Un codificador 104 puede enviar este parámetro común en caso de que las unidades de decodificación se eliminan del CPB a intervalos regulares. Este puede ser el caso, por ejemplo, cuando cada unidad de decodificación corresponde a cierto número de filas de la imagen o tiene alguna otra estructura regular.

Por ejemplo, la bandera de retardo de eliminación del CPB de la unidad de decodificación común se puede establecer en 1 cuando el parámetro de retardo de eliminación del CPB de la unidad de decodificación común ha de ser incluido en el mensaje de SEI de temporización de imagen y 0 cuando no ha de ser incluido. Si es sí (por ejemplo, la bandera se establece en 1), el codificador 104 puede determinar 304 un parámetro de retardo de eliminación del CPB de la unidad de decodificación común (por ejemplo, `common_du_cpb_removal_delay`) que es aplicable a todas las unidades de decodificación en una unidad de acceso. Si es no (por ejemplo, la bandera se establece en 0), el codificador 104 puede determinar 306 parámetros de retardo de eliminación del CPB de la unidad de decodificación separados (por ejemplo, `du_cpb_removal_delay[i]`) para cada unidad de decodificación en una unidad de acceso.

Si un parámetro de retardo de eliminación del CPB de la unidad de decodificación común está presente en un mensaje de SEI de temporización de imagen, puede especificar una cantidad de tictacs de reloj de imagen secundaria a esperar después de la eliminación del CPB 120 de una unidad de decodificación inmediatamente anterior antes de eliminar del CPB 120 una unidad de decodificación actual en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen.

Por ejemplo, cuando una unidad de decodificación es una primera unidad de decodificación en una unidad de acceso, el parámetro de retardo de eliminación del CPB 120 de la unidad de decodificación común puede especificar una cantidad de tictacs de reloj de imagen secundaria a esperar después de la eliminación del CPB 120 de una última unidad de decodificación en una unidad de acceso asociada con un mensaje de SEI de período de almacenamiento temporal más reciente en una unidad de acceso anterior antes de eliminar del CPB 120 la primera unidad de decodificación en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen.

Cuando la unidad de decodificación no es una primera unidad de decodificación en una unidad de acceso, el parámetro de retardo de eliminación del CPB de la unidad de decodificación común puede especificar una cantidad de tictacs de reloj de imagen secundaria a esperar después de la eliminación del CPB 120 de una unidad de decodificación anterior en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen antes de eliminar del CPB una unidad de decodificación actual en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen.

Por el contrario, cuando un parámetro de retardo de eliminación del CPB de la unidad de decodificación común (por ejemplo, `common_du_cpb_removal_delay`) no se envía en un mensaje de SEI de temporización de imagen, se pueden incluir parámetros de retardo de eliminación del CPB de la unidad de decodificación separados (por ejemplo, `du_cpb_removal_delay[i]`) en el mensaje de SEI de temporización de imagen para cada unidad de decodificación en una unidad de acceso. Los parámetros de retardo de eliminación del CPB de la unidad de decodificación (por ejemplo, `du_cpb_removal_delay[i]`) pueden especificar una cantidad de tictacs de reloj de imagen secundaria a esperar después de la eliminación del CPB 120 de la última unidad de decodificación antes de eliminar del CPB 120 una unidad de decodificación de orden i en la unidad de acceso asociada con el mensaje de SEI de temporización de imagen. Los parámetros de retardo de eliminación del CPB de la unidad de decodificación se pueden calcular según un resto de un contador de módulo $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$ donde `cpb_removal_delay_length_minus1 + 1` es la longitud de un parámetro de retardo de eliminación del CPB de la unidad de decodificación común.

La Figura 4 es un diagrama de flujo que ilustra una configuración de un método 400 para almacenamiento temporal de un flujo de bits. El método 400 se puede realizar por un decodificador 112 en un dispositivo electrónico 102 (por ejemplo, el dispositivo electrónico B 102b), que puede recibir 402 un mensaje (por ejemplo, un mensaje de SEI de temporización de imagen u otro mensaje). Por ejemplo, el dispositivo electrónico 102 puede recibir 402 el mensaje a

5 través de uno o más de transmisión inalámbrica, transmisión por cable, bus de dispositivo, red, etc. Por ejemplo, el dispositivo electrónico B 102b puede recibir 402 el mensaje del dispositivo electrónico A 102a. El mensaje puede ser parte del flujo de bits 114, por ejemplo. En otro ejemplo, el dispositivo electrónico B 102b puede recibir el mensaje del dispositivo electrónico A 102a en una transmisión separada 110 (que no forma parte del flujo de bits 114, por ejemplo). Por ejemplo, el mensaje de SEI de temporización de imagen se puede recibir usando algún mecanismo fuera de banda. En algunas configuraciones, el mensaje puede incluir una o más de una bandera de temporización de imagen, uno o más retardos de eliminación para unidades de decodificación en una unidad de acceso y uno o más parámetros de NAL. De este modo, la recepción 402 del mensaje puede incluir recibir uno o más de una bandera de temporización de imagen, uno o más retardos de eliminación para unidades de decodificación en una unidad de acceso y uno o más parámetros de NAL.

10 El decodificador 112 puede determinar 404 si un CPB 120 opera a nivel de unidad de acceso o a nivel de imagen secundaria. Por ejemplo, un decodificador 112 puede decidir operar sobre la base de imagen secundaria si quiere lograr una baja latencia. Alternativamente, la decisión se puede basar en si el decodificador 112 tiene suficientes recursos para soportar la operación basada en imagen secundaria. Si el CPB 120 opera a nivel de imagen secundaria, el decodificador puede determinar 406 uno o más retardos de eliminación para unidades de decodificación en una unidad de acceso. Por ejemplo, el decodificador 112 puede determinar un único parámetro de retardo de eliminación del CPB de la unidad de decodificación común (por ejemplo, common_du_cpb_removal_delay) que es aplicable a todas las unidades de decodificación en la unidad de acceso. Alternativamente, el decodificador 112 puede determinar un retardo de eliminación del CPB de la unidad de decodificación separado (por ejemplo, du_cpb_removal_delay[i]) para cada unidad de decodificación en la unidad de acceso. En otras palabras, el mensaje de SEI de temporización de imagen puede incluir un parámetro común aplicable a todas las unidades de decodificación en una unidad de acceso o parámetros separados para cada unidad de decodificación.

15 El decodificador 112 también puede eliminar 408 unidades de decodificación en base a los retardos de eliminación de las unidades de decodificación, es decir, usando o bien un parámetro común aplicable a todas las unidades de decodificación en una unidad de acceso o bien parámetros separados para cada unidad de decodificación. El decodificador 112 también puede decodificar 410 las unidades de decodificación.

20 El decodificador 112 puede usar una variable ClockDiff cuando se determina un tiempo de eliminación para determinados de diversos parámetros señalados. Específicamente, ClockDiff se puede determinar según $ClockDiff = (\text{num_units_in_tick} - (\text{num_units_in_sub_tick} * (\text{num_decoding_units_minus1} + 1)) / \text{time_scale})$ donde num_units_in_tick es el número de unidades de tiempo de un reloj que opera a la frecuencia time_scale Hz que corresponde a un aumento de un contador de tictac de reloj, num_units_in_sub_tick es el número de unidades de tiempo de un reloj que opera a la frecuencia time_scale Hz que corresponde a un aumento de un contador de tictac de reloj de imagen secundaria, num_decoding_units_minus1+1 es una cantidad de unidades de decodificación en la unidad de acceso, y time_scale es el número de unidades de tiempo que pasan en un segundo.

25 Cuando una bandera de decodificador de referencia hipotético (HRD) de bajo retardo (por ejemplo, low_delay_hrd_flag) se establece en 1, $t_{r,n}(m) < t_{af}(m)$, una bandera de temporización de imagen se establece en 1, el CPB está operando a nivel de imagen secundaria y ClockDiff es mayor que cero, el tiempo de eliminación para la unidad de decodificación m, $t_r(m)$ se determina según: $t_r(m) = t_{r,n}(m) + t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub}) + \text{ClockDiff}$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la unidad de decodificación m, t_{c_sub} es un tictac de reloj de imagen secundaria, Ceil() es una función de techo y $t_{af}(m)$ es el tiempo de llegada final de la unidad de decodificación m.

30 Cuando una bandera de decodificador de referencia hipotético (HRD) de bajo retardo (por ejemplo, low_delay_hrd_flag) se establece en 1, $t_{r,n}(n) < t_{af}(n)$, una bandera de temporización de imagen se establece en 1, el CPB está operando a nivel de unidad de acceso y ClockDiff es mayor que cero, el tiempo de eliminación para la unidad de acceso n, $t_r(n)$ se determina según: $t_r(n) = t_{r,n}(n) + t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c) - \text{ClockDiff}$ donde $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n, t_c es un tictac de reloj, Ceil() es una función de techo y $t_{af}(n)$ es el tiempo de llegada final de la unidad de acceso n.

35 Cuando una bandera de decodificador de referencia hipotético (HRD) de bajo retardo (por ejemplo, low_delay_hrd_flag) se establece en 1, $t_{r,n}(m) < t_{af}(m)$, una bandera de temporización de imagen se establece en 1 y el CPB está operando a nivel de imagen secundaria, el tiempo de eliminación para la última unidad de decodificación m de la unidad de acceso, $t_r(m)$ según: $t_r(m) = t_{r,n}(m) + \max(t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub}), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c)))$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la última unidad de decodificación m, t_{c_sub} es un tictac de reloj de imagen secundaria, Ceil() es una función de techo, $t_{af}(m)$ es un tiempo de llegada final de la última unidad de decodificación m, $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n, t_c es un tictac de reloj y $t_{af}(n)$ es un tiempo de llegada final de la unidad de acceso n.

40 Cuando una bandera de decodificador de referencia hipotético (HRD) de bajo retardo se establece en 1, $t_{r,n}(n) < t_{af}(n)$, una bandera de temporización de imagen se establece en 1 y el CPB está operando a nivel de unidad de acceso, el tiempo de eliminación para la unidad de acceso n, $t_r(n)$ según: $t_r(n) = t_{r,n}(n) + \max(t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub}), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c)))$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la última unidad de

de la unidad de decodificación m, $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n, t_c es un tictac de reloj, $t_{af}(n)$ es un tiempo de llegada final de la unidad de acceso n y $t_{af}(m)$ es un tiempo de llegada final de la unidad de decodificación en la unidad de acceso n.

5 Cuando una bandera de decodificador de referencia hipotético (HRD) de bajo retardo se establece en 1, $t_{r,n}(n) < t_{af}(n)$, una bandera de temporización de imagen se establece en 1 y el CPB está operando a nivel de unidad de acceso, el tiempo de eliminación para la unidad de acceso n, $t_r(n)$ según: $t_r(n) = t_{af}(n)$ donde $t_{r,n}(m)$ es el tiempo nominal de eliminación de la última unidad de decodificación n, t_{c_sub} es un tictac de reloj de imagen secundaria, Ceil() es una función de techo, $t_{af}(m)$ es un tiempo de llegada final de la última unidad de decodificación m, $t_{r,n}(n)$ es el tiempo nominal de eliminación de la unidad de acceso n, t_c es un tictac de reloj y $t_{af}(n)$ es un tiempo de llegada final de la unidad de acceso n.

10 Si el CPB opera en un nivel de unidad de acceso, el decodificador 112 puede determinar 412 un parámetro de retardo de eliminación del CPB. Éste se puede incluir en el mensaje de SEI de temporización de imagen recibido (por ejemplo, `cpb_removal_delay`). El decodificador 112 también puede eliminar 414 una unidad de acceso en base al parámetro de retardo de eliminación de CPB y decodificar 416 la unidad de acceso. En otras palabras, el decodificador 112 puede decodificar todas las unidades de acceso a la vez, en lugar de unidades de decodificación dentro de la unidad de acceso.

15 La Figura 5 es un diagrama de flujo que ilustra una configuración de un método 500 para determinar uno o más retardos de eliminación para unidades de decodificación en una unidad de acceso. En otras palabras, el método 500 ilustrado en la Figura 5 puede ilustrar además el paso 406 en el método 400 ilustrado en la Figura 4. El método 500 se puede realizar mediante un decodificador 112. El decodificador 112 puede determinar 502 si un mensaje de SEI de temporización de imagen recibido incluye un parámetro de retardo de eliminación del CPB de la unidad de decodificación común. Esto puede incluir determinar si se establece una bandera de retardo de eliminación del CPB de la unidad de decodificación común (por ejemplo, `common_du_cpb_removal_delay_flag`). Si es así, el decodificador 112 puede determinar 504 un parámetro de retardo de eliminación del CPB de la unidad de decodificación común (por ejemplo, `common_du_cpb_removal_delay`) que es aplicable a todas las unidades de decodificación en una unidad de acceso. Si no es así, el decodificador 112 puede determinar 506 parámetros de retardo de eliminación del CPB de la unidad de decodificación separados (por ejemplo, `du_cpb_removal_delay[i]`) para cada unidad de decodificación en una unidad de acceso.

20 Además de modificar la semántica del mensaje de SEI de temporización de imagen, los presentes sistemas y métodos también pueden imponer una restricción de flujo de bits de modo que la operación de operación del CPB basada en imagen secundaria y la operación del CPB basada en unidad de acceso den como resultado la misma temporización de eliminación de unidad de decodificación. Específicamente, cuando la bandera de temporización de imagen (por ejemplo, `sub_pic_cpb_params_present_flag`) se establece en 1, el retardo de eliminación de CPB se puede establecer según

25 [Expresión matemática 6]

$$cpb_removal_delay = \frac{\sum_{i=0}^{num_decoding_units_minus1} du_cpb_removal_delay[i] * t_{c_sub}}{t_c}$$

donde `du_cpb_removal_delay[i]` son los parámetros de retardo de eliminación del CPB de la unidad de decodificación, t_c es un tictac de reloj, t_{c_sub} es un tictac de reloj de imagen secundaria, `num_decoding_units_minus1` es una cantidad de unidades de decodificación en la unidad de acceso desplazada en una, e i es un índice.

30 Alternativamente, el retardo de eliminación del CPB se puede establecer como se describe a continuación: Permitamos que la variable $T_{du}(k)$ se defina como:

35 [Expresión matemática 7]

$$T_{du}(k) = T_{du}(k-1) + t_{c_sub} * \sum_{i=0}^{num_decoding_units_minus1_k} (du_cpb_removal_delay_minus1_k[i] + 1)$$

40 donde `du_cpb_removal_delay_minus1k[i]` y `num_decoding_units_minus1k` son parámetros para la unidad de decodificación de orden l de la unidad de acceso de orden k (con $k=0$ para la unidad de acceso que inicializó el HRD y $T_{du}(k)=0$ para $k<1$, y donde `du_cpb_removal_delay_minus1k[i]+1 = du_cpb_removal_delay_minus1k[i]` es el parámetro de retardo de eliminación del CPB de la unidad de decodificación para la unidad de decodificación de orden l de la unidad de acceso de orden k , y `num_decoding_units_minus1k` es el número de unidades de decodificación en la unidad de acceso de orden k , t_c es un tictac de reloj, t_{c_sub} es un tictac de reloj de imagen secundaria, e i y k son índices. Entonces cuando la bandera de temporización de la imagen (por ejemplo, `sub_pic_cpb_params_present_flag`) se establece en 1, la siguiente condición será cierta:

($au_cpb_removal_delay_minus1 + 1$) * $t_c == T_{du}(k)$, donde ($au_cpb_removal_delay_minus1 + 1$) = $cpb_removal_delay$, el retardo de eliminación del CPB. De este modo, en este caso, el retardo de eliminación del CPB ($au_cpb_removal_delay_minus1 + 1$) se establece de manera que la operación de la operación del CPB basada en imagen secundaria y la operación del CPB basada en la unidad de acceso dan como resultado la misma temporización de eliminación de unidad de acceso y eliminación de la última unidad de decodificación de la unidad de acceso.

Alternativamente, el retardo de eliminación del CPB se puede establecer según

[Expresión matemática 8]

$$-1 \leq [cpb_removal_delay * t_c - (\sum_{i=0}^{num_decoding_units_minus1} du_cpb_removal_delay[i] * t_{c,sub})] \leq 1$$

donde $du_cpb_removal_delay[i]$ son los parámetros de retardo de eliminación del CPB de la unidad de decodificación, t_c es un tictac de reloj, $t_{c,sub}$ es un tictac de reloj de imagen secundaria, $num_decoding_units_minus1$ es una cantidad de unidades de decodificación en la unidad de acceso desplazada en una, e i es un índice.

Alternativamente, $cpb_removal_delay$ y $du_cpb_removal_delay[num_decoding_units_minus1]$ puede ser según: $cpb_removal_delay * t_c = du_cpb_removal_delay[num_decoding_units_minus1] * t_{c,sub}$ donde $du_cpb_removal_delay[num_decoding_units_minus1]$ es el parámetro de retardo de eliminación del CPB de la unidad de decodificación para la unidad de decodificación de orden $num_decoding_units_minus1$, t_c es un tictac de reloj, $t_{c,sub}$ es un tictac de reloj de imagen secundaria, $num_decoding_units_minus1$ es una cantidad de unidades de decodificación en la unidad de acceso desplazada en una.

Además de modificar la semántica del mensaje de SEI de temporización de imagen, los presentes sistemas y métodos también pueden imponer una restricción del flujo de bits para que la operación de operación del CPB basada en imagen secundaria y la operación del CPB basada en unidad de acceso den como resultado la misma temporización de eliminación de unidad de decodificación. Específicamente, cuando la bandera de temporización de imagen (por ejemplo, $sub_pic_cpb_params_present_flag$) se establece en 1, los valores de $cpb_removal_delay$ y $du_cpb_removal_delay[num_decoding_units_minus1]$ se pueden establecer para satisfacer: $-1 \leq (cpb_removal_delay * t_c - du_cpb_removal_delay[num_decoding_units_minus1] * t_{c,sub}) \leq 1$ donde $du_cpb_removal_delay[num_decoding_units_minus1]$ es el parámetro de retardo de eliminación del CPB de la unidad de decodificación para la unidad de decodificación de orden $num_decoding_units_minus1$, t_c es un tictac de reloj, $t_{c,sub}$ es un tictac de reloj de imagen secundaria, $num_decoding_units_minus1$ es una cantidad de unidades de decodificación en la unidad de acceso desplazada en una.

La Figura 6A es un diagrama de bloques que ilustra una configuración de un codificador 604 en un dispositivo electrónico 602. Se debería observar que uno o más de los elementos ilustrados como incluidos dentro del dispositivo electrónico 602 se pueden implementar en hardware, software o una combinación de ambos. Por ejemplo, el dispositivo electrónico 602 incluye un codificador 604, que se puede implementar en hardware, software o una combinación de ambos. Por ejemplo, el codificador 604 se puede implementar como un circuito, circuito integrado, circuito integrado de aplicaciones específicas (ASIC), procesador en comunicación electrónica con memoria con instrucciones ejecutables, microprograma, agrupación de puertas programables en campo (FPGA), etc., o una combinación de los mismos. En algunas configuraciones, el codificador 604 puede ser un codificador de HEVC.

El dispositivo electrónico 602 puede incluir una fuente 622. La fuente 622 puede proporcionar datos de cuadro o imagen (por ejemplo, video) como una o más imágenes de entrada 606 al codificador 604. Ejemplos de la fuente 622 pueden incluir sensores de imagen, memoria, interfaces de comunicación, interfaces de red, receptores inalámbricos, puertos, etc.

Se pueden proporcionar una o más imágenes de entrada 606 a un módulo de predicción intra imagen y a un almacenador temporal de reconstrucción 624. También se puede proporcionar una imagen de entrada 606 a un módulo de estimación de movimiento y compensación de movimiento 646 y a un módulo de sustracción 628.

El módulo de predicción intra imagen y el almacenador temporal de reconstrucción 624 pueden generar información intra modo 640 y una intra señal 626 en base a una o más imágenes de entrada 606 y datos reconstruidos 660. El módulo de estimación de movimiento y compensación de movimiento 646 puede generar información inter modo 648 y una inter señal 644 en base a una o más imágenes de entrada 606 y una imagen de referencia 678 del almacenador temporal de imágenes decodificadas 676. En algunas configuraciones, el almacenador temporal de imágenes decodificadas 676 puede incluir datos de una o más imágenes de referencia en el almacenador temporal de imágenes decodificadas 676.

El codificador 604 puede seleccionar entre la intra señal 626 y la inter señal 644 según un modo. La intra señal 626 se puede usar con el fin de explotar las características espaciales dentro de una imagen en un modo de intra

codificación. La inter señal 644 se puede usar con el fin de explotar características temporales entre imágenes en un modo de inter codificación. Mientras que está en el modo de intra codificación, la intra señal 626 se puede proporcionar al módulo de sustracción 628 y la información intra modo 640 se puede proporcionar a un módulo de codificación de entropía 642. Mientras que está en el modo de inter codificación, la inter señal 644 se puede proporcionar al módulo de sustracción 628 y la información inter modo 648 se puede proporcionar al módulo de codificación de entropía 642.

O bien la intra señal 626 o bien la inter señal 644 (dependiendo del modo) se resta de una imagen de entrada 606 en el módulo de sustracción 628 con el fin de producir un residuo de predicción 630. El residuo de predicción 630 se proporciona a un módulo de transformación 632. El módulo de transformación 632 puede comprimir el residuo de predicción 630 para producir una señal transformada 634 que se proporciona a un módulo de cuantificación 636. El módulo de cuantificación 636 cuantifica la señal transformada 634 para producir coeficientes transformados y cuantificados (TQC) 638.

Los TQC 638 se proporcionan a un módulo de codificación de entropía 642 y a un módulo de cuantificación inversa 650. El módulo de cuantificación inversa 650 realiza una cuantificación inversa sobre los TQC 638 para producir una señal cuantificada inversa 652 que se proporciona a un módulo de transformación inversa 654. El módulo de transformación inversa 654 descomprime la señal cuantificada inversa 652 para producir una señal descomprimida 656 que se proporciona a un módulo de reconstrucción 658.

El módulo de reconstrucción 658 puede producir datos reconstruidos 660 en base a la señal descomprimida 656. Por ejemplo, el módulo de reconstrucción 658 puede reconstruir imágenes (modificadas). Los datos reconstruidos 660 se pueden proporcionar a un filtro de desbloqueo 662 y al módulo de intra predicción y al almacenador temporal de reconstrucción 624. El filtro de desbloqueo 662 puede producir una señal filtrada 664 en base a los datos reconstruidos 660.

La señal filtrada 664 se puede proporcionar a un módulo de desplazamiento adaptativo de muestras (SAO) 666. El módulo de SAO 666 puede producir información de SAO 668 que se proporciona al módulo de codificación de entropía 642 y una señal de SAO 670 que se proporciona a un filtro de bucle adaptativo (ALF) 672. El ALF 672 produce una señal de ALF 674 que se proporciona al almacenador temporal de imágenes decodificadas 676. La señal de ALF 674 puede incluir datos de una o más imágenes que se pueden usar como imágenes de referencia.

El módulo de codificación de entropía 642 puede codificar los TQC 638 para producir el flujo de bits A 614a (por ejemplo, datos de imagen codificada). Por ejemplo, el módulo de codificación de entropía 642 puede codificar los TQC 638 usando Codificación de Longitud Variable Adaptativa al Contexto (CAVLC) o Codificación Aritmética Binaria Adaptativa al Contexto (CABAC). En particular, el módulo de codificación de entropía 642 puede codificar los TQC 638 en base a una o más de información intra modo 640, información inter modo 648 e información de SAO 668. El flujo de bits A 614a (por ejemplo, datos de imagen codificada) se puede proporcionar a un módulo de generación de mensajes 608. El módulo de generación de mensajes 608 se puede configurar de manera similar al módulo de generación de mensajes 108 descrito en conexión con la Figura 1. Además o alternativamente, el módulo de generación de mensajes 608 puede realizar uno o más de los procedimientos descritos en conexión con la Figura 2 y la Figura 3.

Por ejemplo, el módulo de generación de mensajes 608 puede generar un mensaje (por ejemplo, mensaje de SEI de temporización de imagen u otro mensaje) que incluye parámetros de imagen secundaria. Los parámetros de imagen secundaria pueden incluir uno o más retardos de eliminación para unidades de decodificación (por ejemplo, `common_du_cpb_removal_delay` o `du_cpb_removal_delay[i]`) y uno o más parámetros de NAL (por ejemplo, `common_num_nalus_in_du_minus1` o `num_nalus_in_du_minus1[i]`). En algunas configuraciones, el mensaje se puede insertar en el flujo de bits A 614a para producir el flujo de bits B 614b. De este modo, el mensaje se puede generar después de que se genere todo el flujo de bits A 614a (por ejemplo, después de que se genere la mayor parte del flujo de bits B 614b), por ejemplo. En otras configuraciones, el mensaje no se puede insertar en el flujo de bits A 614a (en cuyo caso el flujo de bits B 614b puede ser el mismo que el flujo de bits A 614a), pero se puede proporcionar en una transmisión separada 610.

En algunas configuraciones, el dispositivo electrónico 602 envía el flujo de bits 614 a otro dispositivo electrónico. Por ejemplo, el flujo de bits 614 se puede proporcionar a una interfaz de comunicación, interfaz de red, transmisor inalámbrico, puerto, etc. Por ejemplo, el flujo de bits 614 se puede transmitir a otro dispositivo electrónico a través de LAN, Internet, una estación base de telefonía celular, etc. El flujo de bits 614 se puede almacenar además o alternativamente en la memoria u otro componente en el dispositivo electrónico 602.

La Figura 6B es un diagrama de bloques que ilustra una configuración de un codificador de vídeo 1782 en un dispositivo electrónico 1702. El codificador de vídeo 1782 puede incluir un codificador de capa de mejora 1706, un codificador de capa base 1709, un bloque de ampliación de escala de resolución 1770 y una interfaz de salida 1780. El codificador de vídeo de la Figura 6B, por ejemplo, es adecuado para codificación de vídeo escalable y codificación de vídeo de vista múltiple, como se describe en la presente memoria.

El codificador de capa de mejora 1706 puede incluir una entrada de vídeo 1781 que recibe una imagen de entrada

1704. La salida de la entrada de vídeo 1781 se puede proporcionar a un sumador/restador 1783 que recibe una salida de una selección de predicción 1750. La salida del sumador/restador 1783 se puede proporcionar a un bloque de transformación y cuantificación 1752. La salida del bloque de transformación y cuantificación 1752 se puede proporcionar a un bloque de codificación de entropía 1748 y un bloque de transformación inversa y de escalado 1772. Después de que se realiza la codificación de entropía 1748, la salida del bloque de codificación de entropía 1748 se puede proporcionar a la interfaz de salida 1780. La interfaz de salida 1780 puede emitir tanto el flujo de bits de vídeo de capa base codificado 1707 como el flujo de bits de vídeo de capa de mejora codificado 1710.

La salida del bloque de transformación inversa y escalado 1772 se puede proporcionar a un sumador 1779. El sumador 1779 también puede recibir la salida de la selección de predicción 1750. La salida del sumador 1779 se puede proporcionar a un bloque de desbloqueo 1751. La salida del bloque de desbloqueo 1751 se puede proporcionar a un almacenador temporal de referencia 1794. Se puede proporcionar una salida del almacenador temporal de referencia 1794 a un bloque de compensación de movimiento 1754. La salida del bloque de compensación de movimiento 1754 se puede proporcionar a la selección de predicción 1750. Una salida del almacenador temporal de referencia 1794 también se puede proporcionar a un intra predictor 1756. La salida del intra predictor 1756 se puede proporcionar a la selección de predicción 1750. La selección de predicción 1750 también puede recibir una salida del bloque de ampliación de escala de resolución 1770.

El codificador de capa base 1709 puede incluir una entrada de vídeo 1762 que recibe una imagen de entrada muestreada descendientemente, u otro contenido de imagen adecuado para combinar con otra imagen, o una imagen de entrada de vista alternativa o la misma imagen de entrada 1703 (es decir, la misma que la imagen de entrada 1704 recibida por el codificador de capa de mejora 1706). La salida de la entrada de vídeo 1762 se puede proporcionar a un bucle de predicción de codificación 1764. La codificación de entropía 1766 se puede proporcionar en la salida del bucle de predicción de codificación 1764. La salida del bucle de predicción de codificación 1764 también se puede proporcionar a un almacenador temporal de referencia 1768. El almacenador temporal de referencia 1768 puede proporcionar realimentación al bucle de predicción de codificación 1764. La salida del almacenador temporal de referencia 1768 también se puede proporcionar al bloque de ampliación de escala de resolución 1770. Una vez que se ha realizado la codificación de entropía 1766, se puede proporcionar la salida a la interfaz de salida 1780. El flujo de bits de vídeo de capa base codificado 1707 y/o el flujo de bits de vídeo de capa de mejora codificado 1710 se puede proporcionar a uno o más módulos de generación de mensajes, según se desee.

La Figura 7A es un diagrama de bloques que ilustra una configuración de un decodificador 712 en un dispositivo electrónico 702. El decodificador 712 se puede incluir en un dispositivo electrónico 702. Por ejemplo, el decodificador 712 puede ser un decodificador de HEVC. El decodificador 712 y uno o más de los elementos ilustrados que se incluyen en el decodificador 712 se pueden implementar en hardware, software o una combinación de ambos. El decodificador 712 puede recibir un flujo de bits 714 (por ejemplo, una o más imágenes codificadas y datos de sobrecarga incluidos en el flujo de bits 714) para su decodificación. En algunas configuraciones, el flujo de bits 714 recibido puede incluir datos de sobrecarga recibidos, tales como un mensaje (por ejemplo, un mensaje de SEI de temporización de imagen u otro mensaje), cabecera de porción, PPS, etc. En algunas configuraciones, el decodificador 712 puede recibir adicionalmente una transmisión separada 710. La transmisión separada 710 puede incluir un mensaje (por ejemplo, un mensaje de SEI de temporización de imagen u otro mensaje). Por ejemplo, un mensaje de SEI de temporización de imagen u otro mensaje se puede recibir en una transmisión separada 710 en lugar de en el flujo de bits 714. No obstante, se debería observar que la transmisión separada 710 puede ser opcional y puede no ser utilizada en algunas configuraciones.

El decodificador 712 incluye un CPB 720. El CPB 720 se puede configurar de manera similar al CPB 120 descrito en conexión con la Figura 1 anterior. Además, o alternativamente, el decodificador 712 puede realizar uno o más de los procedimientos descritos en conexión con la Figura 4 y la Figura 5. Por ejemplo, el decodificador 712 puede recibir un mensaje (por ejemplo, mensaje de SEI de temporización de imagen u otro mensaje) con parámetros de imagen secundaria y eliminar y decodificar unidades de decodificación en una unidad de acceso en base a los parámetros de imagen secundaria. Se debería observar que una o más unidades de acceso se pueden incluir en el flujo de bits y pueden incluir uno o más de datos de imágenes codificadas y datos de sobrecarga.

El Almacenador Temporal de Imágenes Codificadas (CPB) 720 puede proporcionar datos de imagen codificados a un módulo de decodificación de entropía 701. Los datos de imagen codificados pueden ser decodificados por entropía por un módulo de decodificación de entropía 701, produciendo por ello una señal de información de movimiento 703 y coeficientes cuantificados, escalados y/o transformados 705.

La señal de información de movimiento 703 se puede combinar con una parte de una señal de cuadro de referencia 798 de un almacenador temporal de imágenes decodificadas 709 en un módulo de compensación de movimiento 780, que puede producir una señal de predicción inter cuadro 782. Los coeficientes cuantificados, desescalados y/o transformados 705 pueden ser cuantificados inversamente, escalados y transformados inversamente por un módulo inverso 707, produciendo por ello una señal residual decodificada 784. La señal residual decodificada 784 se puede añadir a una señal de predicción 792 para producir una señal combinada 786. La señal de predicción 792 puede ser una señal seleccionada de o bien la señal de predicción inter cuadro 782 producida por el módulo de compensación de movimiento 780 o bien una señal de predicción intra cuadro 790 producida por un módulo de predicción intra cuadro 788. En algunas configuraciones, esta selección de señal se puede basar en (por ejemplo, controlar por) el

flujo de bits 714.

La señal de predicción intra cuadro 790 se puede predecir a partir de información decodificada previamente a partir de la señal combinada 786 (en el cuadro actual, por ejemplo). La señal combinada 786 también se puede filtrar mediante un filtro de desbloqueo 794. La señal filtrada resultante 796 se puede escribir en el almacenador temporal de imágenes decodificadas 709. La señal filtrada resultante 796 puede incluir una imagen decodificada. El almacenador temporal de imágenes decodificadas 709 puede proporcionar una imagen decodificada que se puede emitir 718. En algunos casos, 709 se puede considerar como memoria de cuadros.

La Figura 7B es un diagrama de bloques que ilustra una configuración de un decodificador de vídeo 1812 en un dispositivo electrónico 1802. El decodificador de vídeo 1812 puede incluir un decodificador de capa de mejora 1815 y un decodificador de capa base 1813. El decodificador de vídeo 812 también puede incluir una interfaz 1889 y ampliación de escala de resolución 1870. El decodificador de vídeo de la Figura 7B, por ejemplo, es adecuado para decodificación de vídeo escalable y decodificación de vídeo de vista múltiple, como se describe en la presente memoria.

La interfaz 1889 puede recibir un flujo de vídeo codificado 1885. El flujo de vídeo codificado 1885 puede consistir en un flujo de vídeo codificado de capa base y un flujo de vídeo codificado de capa de mejora. Estos dos flujos se pueden enviar por separado o juntos. La interfaz 1889 puede proporcionar algo de o todo el flujo de vídeo codificado 1885 a un bloque de decodificación de entropía 1886 en el decodificador de capa base 1813. La salida del bloque de decodificación de entropía 1886 se puede proporcionar a un bucle de predicción de decodificación 1887. La salida del bucle de predicción de decodificación 1887 se puede proporcionar a un almacenador temporal de referencia 1888. El almacenador temporal de referencia puede proporcionar realimentación al bucle de predicción de decodificación 1887. El almacenador temporal de referencia 1888 también puede emitir el flujo de vídeo de capa base decodificado 1884.

La interfaz 1889 también puede proporcionar algo de o todo el flujo de vídeo codificado 1885 a un bloque de decodificación de entropía 1890 en el decodificador de capa de mejora 1815. La salida del bloque de decodificación de entropía 1890 se puede proporcionar a un bloque de cuantificación inversa 1891. La salida del bloque de cuantificación inversa 1891 se puede proporcionar a un sumador 1892. El sumador 1892 puede añadir la salida del bloque de cuantificación inversa 1891 y la salida de un bloque de selección de predicción 1895. La salida del sumador 1892 se puede proporcionar a un bloque de desbloqueo 1893. La salida del bloque de desbloqueo 1893 se puede proporcionar a un almacenador temporal de referencia 1894. El almacenador temporal de referencia 1894 puede emitir el flujo de vídeo de capa de mejora decodificado 1882. La salida del almacenador temporal de referencia 1894 también se puede proporcionar a un intra predictor 1897. El decodificador de capa de mejora 1815 puede incluir compensación de movimiento 1896. La compensación de movimiento 1896 se puede realizar después de la ampliación de escala de resolución 1870. El bloque de selección de predicción 1895 puede recibir la salida del intra predictor 1897 y la salida de la compensación de movimiento 1896. También, el decodificador puede incluir uno o más almacenadores temporales de imágenes codificadas, según se desee, tal como junto con la interfaz 1889.

La Figura 8 ilustra diversos componentes que se pueden utilizar en un dispositivo electrónico de transmisión 802. Uno o más de los dispositivos electrónicos 102, 602, 702 descritos en la presente memoria se pueden implementar según el dispositivo electrónico de transmisión 802 ilustrado en la Figura 8.

El dispositivo electrónico de transmisión 802 incluye un procesador 817 que controla la operación del dispositivo electrónico 802. También se puede hacer referencia al procesador 817 como CPU. La memoria 811, que puede incluir ambas memoria de sólo lectura (ROM), memoria de acceso aleatorio (RAM) o cualquier tipo de dispositivo que pueda almacenar información, proporciona instrucciones 813a (por ejemplo, instrucciones ejecutables) y datos 815a al procesador 817. Una parte de la memoria 811 también puede incluir una memoria de acceso aleatorio no volátil (NVRAM). La memoria 811 puede estar en comunicación electrónica con el procesador 817.

Las instrucciones 813b y los datos 815b también pueden residir en el procesador 817. Las instrucciones 813b y/o los datos 815b cargados en el procesador 817 también pueden incluir las instrucciones 813a y/o los datos 815a de la memoria 811 que se cargaron para su ejecución o procesamiento por el procesador 817. Las instrucciones 813b se pueden ejecutar por el procesador 817 para implementar los sistemas y métodos descritos en la presente memoria. Por ejemplo, las instrucciones 813b pueden ser ejecutables para realizar uno o más de los métodos 200, 300, 400, 500 descritos anteriormente.

El dispositivo electrónico de transmisión 802 puede incluir una o más interfaces de comunicación 819 para comunicarse con otros dispositivos electrónicos (por ejemplo, dispositivo electrónico de recepción). Las interfaces de comunicación 819 se pueden basar en tecnología de comunicación por cable, tecnología de comunicación inalámbrica o en ambas. Ejemplos de una interfaz de comunicación 819 incluyen un puerto serie, un puerto paralelo, un Bus Serie Universal (USB), un adaptador Ethernet, una interfaz de bus IEEE 1394, una interfaz de bus de sistema de ordenador (SCSI), un puerto de comunicación de infrarrojos (IR), un adaptador de comunicación inalámbrico Bluetooth, un transceptor inalámbrico según las especificaciones del Proyecto de Cooperación de 3ª Generación (3GPP), etc.

5 El dispositivo electrónico de transmisión 802 puede incluir uno o más dispositivos de salida 823 y uno o más dispositivos de entrada 821. Ejemplos de dispositivos de salida 823 incluyen un altavoz, impresora, etc. Un tipo de dispositivo de salida que se puede incluir en un dispositivo electrónico 802 es un dispositivo de visualización 825. Los dispositivos de visualización 825 usados con las configuraciones descritas en la presente memoria pueden utilizar cualquier tecnología de proyección de imágenes adecuada, tal como un tubo de rayos catódicos (CRT), visualizador de cristal líquido (LCD), diodo emisor de luz (LED), plasma de gas, electroluminiscencia o similares. Se puede proporcionar un controlador de visualización 827 para convertir datos almacenados en la memoria 811 en texto, gráficos y/o imágenes en movimiento (según sea apropiado) mostrados en el visualizador 825. Ejemplos de dispositivos de entrada 821 incluyen un teclado, ratón, micrófono, dispositivo de control remoto, botón, palanca de mando, bola de desplazamiento, almohadilla táctil, pantalla táctil, lápiz óptico, etc.

15 Los diversos componentes del dispositivo electrónico de transmisión 802 están acoplados entre sí por un sistema de bus 829, que puede incluir un bus de potencia, un bus de señal de control y un bus de señal de estado, además de un bus de datos. No obstante, por el bien de la claridad, los diversos buses se ilustran en la Figura 8 como el sistema de bus 829. El dispositivo electrónico de transmisión 802 ilustrado en la Figura 8 es un diagrama de bloques funcional en lugar de una lista de componentes específicos.

La Figura 9 es un diagrama de bloques que ilustra diversos componentes que se pueden utilizar en un dispositivo electrónico de recepción 902. Uno o más de los dispositivos electrónicos 102, 602, 702 descritos en la presente memoria se pueden implementar según el dispositivo electrónico de recepción 902 ilustrado en Figura 9.

20 El dispositivo electrónico de recepción 902 incluye un procesador 917 que controla la operación del dispositivo electrónico 902. También se puede hacer referencia al procesador 917 como CPU. La memoria 911, que puede incluir ambas memoria de sólo lectura (ROM), memoria de acceso aleatorio (RAM) o cualquier tipo de dispositivo que pueda almacenar información, proporciona instrucciones 913a (por ejemplo, instrucciones ejecutables) y datos 915a al procesador 917. Una parte de la memoria 911 también puede incluir una memoria de acceso aleatorio no volátil (NVRAM). La memoria 911 puede estar en comunicación electrónica con el procesador 917.

25 Las instrucciones 913b y los datos 915b también pueden residir en el procesador 917. Las instrucciones 913b y/o los datos 915b cargados en el procesador 917 también pueden incluir instrucciones 913a y/o datos 915a de la memoria 911 que fueron cargados para su ejecución o procesamiento por el procesador 917. Las instrucciones 913b se pueden ejecutar por el procesador 917 para implementar los sistemas y métodos descritos en la presente memoria. Por ejemplo, las instrucciones 913b pueden ser ejecutables para realizar uno o más de los métodos 200, 300, 400, 500 descritos anteriormente.

35 El dispositivo electrónico de recepción 902 puede incluir una o más interfaces de comunicación 919 para comunicarse con otros dispositivos electrónicos (por ejemplo, un dispositivo electrónico de transmisión). La interfaz de comunicación 919 se puede basar en tecnología de comunicación por cable, tecnología de comunicación inalámbrica o en ambas. Ejemplos de una interfaz de comunicación 919 incluyen un puerto serie, un puerto paralelo, un Bus Serie Universal (USB), un adaptador Ethernet, una interfaz de bus IEEE 1394, una interfaz de bus de interfaz reducida de sistema de ordenador (SCSI), un puerto de comunicación de infrarrojos (IR), un adaptador de comunicación inalámbrico Bluetooth, un transceptor inalámbrico según las especificaciones del Proyecto de Cooperación de 3ª Generación (3GPP), etc.

40 El dispositivo electrónico de recepción 902 puede incluir uno o más dispositivos de salida 923 y uno o más dispositivos de entrada 921. Ejemplos de dispositivos de salida 923 incluyen un altavoz, una impresora, etc. Un tipo de dispositivo de salida que se puede incluir en un dispositivo electrónico 902 es un dispositivo de visualización 925. Los dispositivos de visualización 925 usados con las configuraciones descritas en la presente memoria pueden utilizar cualquier tecnología de proyección de imágenes adecuada, tal como un tubo de rayos catódicos (CRT), visualizador de cristal líquido (LCD), diodo emisor de luz (LED), plasma de gas, electroluminiscencia o similares. Se puede proporcionar un controlador de visualización 927 para convertir los datos almacenados en la memoria 911 en texto, gráficos y/o imágenes en movimiento (según sea apropiado) mostrados en el visualizador 925. Ejemplos de dispositivos de entrada 921 incluyen un teclado, ratón, micrófono, dispositivo de control remoto, botón, palanca de mando, bola de desplazamiento, almohadilla táctil, pantalla táctil, lápiz óptico, etc.

45 Los diversos componentes del dispositivo electrónico de recepción 902 están acoplados entre sí mediante un sistema de bus 929, que puede incluir un bus de potencia, un bus de señal de control y un bus de señal de estado, además de un bus de datos. No obstante, por el bien de la claridad, los diversos buses se ilustran en la Figura 9 como el sistema de bus 929. El dispositivo electrónico de recepción 902 ilustrado en la Figura 9 es un diagrama de bloques funcional en lugar de una lista de componentes específicos.

55 La Figura 10 es un diagrama de bloques que ilustra una configuración de un dispositivo electrónico 1002 en el que se pueden implementar sistemas y métodos para enviar un mensaje. El dispositivo electrónico 1002 incluye medios de codificación 1031 y medios de transmisión 1033. Los medios de codificación 1031 y los medios de transmisión 1033 se pueden configurar para realizar una o más de las funciones descritas en conexión con una o más de la Figura 1, Figura 2, Figura 3, Figura 6 y Figura 8 anteriores. Por ejemplo, los medios de codificación 1031 y los medios de transmisión 1033 pueden generar un flujo de bits 1014. La Figura 8 anterior ilustra un ejemplo de una

estructura de aparato concreto de la Figura 10. Se pueden implementar otras diversas estructuras para realizar una o más de las funciones de la Figura 1, Figura 2, Figura 3, Figura 6 y Figura 8. Por ejemplo, un DSP se puede realizar mediante software.

La Figura 11 es un diagrama de bloques que ilustra una configuración de un dispositivo electrónico 1102 en el que se pueden implementar sistemas y métodos para almacenar temporalmente un flujo de bits 1114. El dispositivo electrónico 1102 puede incluir medios de recepción 1135 y medios de decodificación 1137. Los medios de recepción 1135 y los medios de decodificación 1137 se pueden configurar para realizar una o más de las funciones descritas en conexión con una o más de la Figura 1, Figura 4, Figura 5, Figura 7 y Figura 9 anteriores. Por ejemplo, los medios de recepción 1135 y los medios de decodificación 1137 pueden recibir un flujo de bits 1114. La Figura 9 anterior ilustra un ejemplo de una estructura de aparato concreta de la Figura 11. Se pueden implementar otras diversas estructuras para realizar una o más funciones de la Figura 1, Figura 4, Figura 5, Figura 7 y Figura 9. Por ejemplo, un DSP se puede realizar mediante software.

La Figura 12 es un diagrama de flujo que ilustra una configuración de un método 1200 para operación de un almacenador temporal de imágenes decodificadas (DPB). El método 1200 se puede realizar por un codificador 104 o una de sus subpartes (por ejemplo, un módulo de almacenamiento de imágenes decodificadas 676). El método 1200 se puede realizar por un decodificador 112 en un dispositivo electrónico 102 (por ejemplo, el dispositivo electrónico B 102b). Además o alternativamente, el método 1200 se puede realizar por un decodificador 712 o una de sus subpartes (por ejemplo, un módulo de almacenador temporal de imágenes decodificadas 709). El decodificador puede analizar sintácticamente la primera cabecera de porción de una imagen 1202. La salida y eliminación de imágenes del DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de porción de la primera porción de la imagen actual) ocurre instantáneamente cuando la primera unidad de decodificación de la unidad de acceso que contiene la imagen actual se elimina del CPB y procede de la siguiente manera.

- Se invoca el proceso de decodificación para un conjunto de imágenes de referencia (RPS). El conjunto de imágenes de referencia es un conjunto de imágenes de referencia asociadas con una imagen, que consiste en todas las imágenes de referencia que son anteriores a la imagen asociada en orden de decodificación, que se pueden usar para la inter predicción de la imagen asociada o cualquier imagen que siga a la imagen asociada en orden de decodificación.

- El flujo de bits del vídeo puede incluir una estructura de sintaxis que se coloca en paquetes de datos lógicos generalmente a los que se hace referencia de manera general como unidades de Capa de Abstracción de Red (NAL). Cada unidad de NAL incluye una cabecera de unidad de NAL, tal como una cabecera de unidad de NAL de dos bytes (por ejemplo, 16 bits), para identificar el propósito de la carga útil de datos asociada. Por ejemplo, cada porción (y/o imagen) codificada se puede codificar en una o más unidades de NAL de porción (y/o imagen). Se pueden incluir otras unidades de NAL para otras categorías de datos, tales como, por ejemplo, información de mejora suplementaria, porción codificada de imagen de acceso de subcapa temporal (TSA), porción codificada de imagen de acceso de subcapa temporal escalonada (STSA), porción codificada de imagen posterior no TSA y no STSA, porción codificada de imagen de acceso de enlace roto, porción codificada de imagen de regeneración decodificada instantánea, porción codificada de imagen de acceso aleatorio limpio, porción codificada de imagen principal decodificable, porción codificada de etiquetado para descartar imagen, conjunto de parámetros de vídeo, conjunto de parámetros de secuencia, conjunto de parámetros de imagen, delimitador de unidad de acceso, final de secuencia, final de flujo de bits, datos de relleno y/o mensaje de información de mejora de secuencia. La Tabla (4) ilustra un ejemplo de códigos de unidad de NAL y clases de tipo de unidad de NAL. Se pueden incluir otros tipos de unidades de NAL, según se desee. También se debería entender que los valores de tipo de unidad de NAL para las unidades de NAL mostradas en la Tabla (4) se pueden reorganizar y reasignar. También se pueden añadir tipos de unidades de NAL adicionales. También se pueden eliminar algunos tipos de unidades de NAL.

Una imagen de intra punto de acceso aleatorio (IRAP) es una imagen codificada para la cual cada unidad de NAL de capa de codificación de vídeo tiene nal_unit_type en el intervalo de BLA_W_LP a RSV_IRAP_VCL23, inclusive como se muestra en la Tabla (4). Una imagen de IRAP contiene solamente porciones intra codificadas (I). Una imagen de regeneración instantánea de decodificación (IDR) es una imagen de IRAP para la cual cada unidad de NAL de capa de codificación de vídeo tiene nal_unit_type igual a IDR_W_RADL o IDR_N_LP como se muestra en la Tabla (4). Una imagen de regeneración instantánea de decodificación (IDR) contiene solamente I porciones, y puede ser la primera imagen en el flujo de bits en orden de decodificación, o puede aparecer más tarde en el flujo de bits. Cada imagen de IDR es la primera imagen de una secuencia de vídeo codificada (CVS) en orden de decodificación. Una imagen de acceso de enlace roto (BLA) es una imagen de IRAP para la cual cada unidad de NAL de capa de codificación de vídeo tiene nal_unit_type igual a BLA_W_LP, BLA_W_RADL o BLA_N_LP como se muestra en la Tabla (4). Una imagen de BLA contiene solamente I porciones, y puede ser la primera imagen en el flujo de bits en orden de decodificación, o puede aparecer más tarde en el flujo de bits. Cada imagen de BLA comienza una nueva secuencia de vídeo codificada, y tiene el mismo efecto en el proceso de decodificación que una imagen de IDR. No obstante, una imagen de BLA contiene elementos de sintaxis que especifican un conjunto de imágenes de referencia no vacío.

[Tabla 4]

nal_unit_type	Nombre de nal_unit_type	Contenido de unidad de NAL y estructura de sintaxis de carga útil de secuencia de bytes en bruto (RBSP)	Clase de tipo de unidad de NAL
0 1	TRAIL_N TRAIL_R	Segmento de porción codificada de una imagen posterior no TSA, no STSA slice_segment_layer_rbsp()	Capa de Codificación de Vídeo (VCL)
2 3	TSA_N TSA_R	Segmento de porción codificada de una imagen de acceso de subcapa temporal (TSA) slice_segment_layer_rbsp()	VCL
4 5	STSA_N STSA_R	Segmento de porción codificada de una imagen de acceso de subcapa temporal escalonada (STSA) slice_segment_layer_rbsp()	VCL
6 7	RADL_N RADL_R	Segmento de porción codificada de una imagen principal de acceso aleatorio (RADL) slice_segment_layer_rbsp()	VCL
8 9	RASL_N RASL_R	Segmento de porción codificada de una imagen principal de salto de acceso aleatorio (RASL) slice_segment_layer_rbsp()	VCL
10 12 14	RSV_VCL_N10 RSV_VCL_N12 RSV_VCL_N14	Tipos de unidades de NAL de VCL no de referencia de subcapas no IRAP reservados	VCL
11 13 14	RSV_VCL_R11 RSV_VCL_R13 RSV_VCL_R14	Tipos de unidades de NAL de VCL de referencia de subcapas no IRAP reservados	VCL
16 17 18	BLA_W_LP BLA_W_RADL BLA_N_LP	Segmento de porción codificada de una imagen de acceso de enlace roto (BLA) slice_segment_layer_rbsp()	VCL
19 20	IDR_W_RADL IDR_N_LP	Segmento de porción codificada de una imagen de regeneración instantánea de decodificación (IDR) slice_segment_layer_rbsp()	VCL
21	CRA_NUT	Segmento de porción codificada de una imagen de acceso aleatorio limpio (CRA) slice_segment_layer_rbsp()	VCL
22 23	RSV_IRAP_VCL22 RSV_IRAP_VCL23	Tipos de unidades de NAL de VCL de IRAP reservados	VCL
24..31	RSV_VCL24.. RSV_VCL31	Tipos de unidades de NAL de VCL de IRAP reservados	VCL
32	VPS_NUT	Conjunto de parámetros de vídeo video_parameter_set_rbsp()	Capa no de codificación de vídeo (no VCL)
33	SPS_NUT	Conjunto de parámetros de secuencia	no VCL

		seq_parameter_set_rbsp()	
34	PPS_NUT	Conjunto de parámetros de imagen pic_parameter_set_rbsp()	no VCL
35	AUD_NUT	Delimitador de unidad de acceso access_unit_delimiter_rbsp()	no VCL
36	EOS_NUT	Final de secuencia end_of_seq_rbsp()	no VCL
37	EOB_NUT	Final de flujo de bits end_of_bitstream_rbsp()	no VCL
38	FD_NUT	Datos de relleno filler_data_rbsp()	no VCL
39	PREFIX_SEI_NUT	Información de mejora suplementaria sei_rbsp()	no VCL
40	SUFFIX_SEI_NUT		
41..47	RSV_NVCL41.. RSV_NVCL47	Reservado	no VCL
48..63	UNSPEC48.. UNSPEC63	Sin especificar	no VCL

Tabla (4)

Con referencia a la Tabla (5), la sintaxis de cabecera de unidad de NAL puede incluir dos bytes de datos, esto es, 16 bits. El primer bit es un "forbidden_zero_bit" que siempre se establece en cero al inicio de una unidad de NAL. Los siguientes seis bits son un "nal_unit_type" que especifica el tipo de estructura de datos de carga útil de secuencia de bytes en bruto ("Rbsp") contenida en la unidad de NAL como se muestra en la Tabla (4). Los siguientes 6 bits son un "nuh_layer_id" que especifica el identificador de la capa. En algunos casos, estos seis bits se pueden especificar como "nuh_reserved_zero_6bits" en su lugar. Los nuh_reserved_zero_6bits pueden ser iguales a 0 en la especificación base del estándar. En una codificación de vídeo escalable y/o extensiones de sintaxis, nuh_layer_id puede especificar que esta unidad de NAL particular pertenece a la capa identificada por el valor de estos 6 bits. El siguiente elemento de sintaxis es "nuh_temporal_id_plus1". El nuh_temporal_id_plus1 menos 1 puede especificar un identificador temporal para la unidad de NAL. El identificador temporal variable TemporalId se puede especificar como TemporalId = nuh_temporal_id_plus1 - 1. El identificador temporal TemporalId se usa para identificar una subcapa temporal. La variable HighestTid identifica la subcapa temporal más alta a ser decodificada.

[Tabla 5]

nal_unit_header() {	Descriptor
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
nuh_layer_id	u(6)
nuh_temporal_id_plus1	u(3)
}	

Tabla (5)

La Tabla (6) muestra una estructura de sintaxis de conjunto de parámetros de secuencia (SPS) ejemplar.

pic_width_in_luma_samples especifica el ancho de cada imagen decodificada en unidades de muestras de luma. pic_width_in_luma_samples no será igual a 0.

pic_height_in_luma_samples especifica la altura de cada imagen decodificada en unidades de muestras de luma. pic_height_in_luma_samples no será igual a 0.

5 sps_max_sub_layers_minus1 más 1 especifica el número máximo de subcapas temporales que pueden estar presentes en cada CVS en referencia al SPS. El valor de sps_max_sub_layers_minus1 estará en el intervalo de 0 a 6, inclusive.

10 La bandera sps_sub_layer_ordering_info_present_flag igual a 1 especifica que están presentes los elementos de sintaxis sps_max_dec_pic_buffering_minus1[i], sps_max_num_reorder_pics[i], y sps_max_latency_increase_plus1[i] para las subcapas sps_max_sub_layers_minus1 + 1. sps_sub_layer_ordering_info_present_flag igual a 0 especifica que los valores de sps_max_dec_pic_buffering_minus1[sps_max_sub_layers_minus1], sps_max_num_reorder_pics[sps_max_sub_layers_minus1], y sps_max_latency_increase_plus1[sps_max_sub_layers_minus1] se aplican a todas las subcapas.

15 sps_max_dec_pic_buffering_minus1[i] más 1 especifica el tamaño máximo requerido del almacenador temporal de imágenes decodificadas para la CVS en unidades de almacenadores temporales de almacenamiento de imágenes cuando HighestTid es igual a i. El valor de sps_max_dec_pic_buffering_minus1[i] estará en el intervalo de 0 a MaxDpbSize - 1, inclusive donde MaxDpbSize especifica el tamaño máximo de almacenador temporal de imágenes decodificadas en unidades de almacenadores temporales de almacenamiento de imágenes. Cuando i es mayor que 0, sps_max_dec_pic_buffering_minus1[i] será mayor o igual que sps_max_dec_pic_buffering_minus1[i - 1]. Cuando sps_max_dec_pic_buffering_minus1[i] no está presente para i en el intervalo de 0 a sps_max_sub_layers_minus1 - 1, inclusive, debido a que sps_sub_layer_ordering_info_present_flag es igual a 0, se infiere que es igual a sps_max_dec_pic_buffering_minus1[sps_max_sub_layers_minus1].

25 sps_max_num_reorder_pics [i] indica el número máximo permitido de imágenes que pueden preceder a cualquier imagen en la CVS en orden de decodificación y seguir a esa imagen en orden de salida cuando HighestTid es igual a i. El valor de sps_max_num_reorder_pics [i] estará en el intervalo de 0 a sps_max_dec_pic_buffering_minus1[i], inclusive. Cuando i es mayor que 0, sps_max_num_reorder_pics[i] será mayor o igual que sps_max_num_reorder_pics[i - 1]. Cuando sps_max_num_reorder_pics[i] no está presente para i en el intervalo de 0 a sps_max_sub_layers_minus1 - 1, inclusive, debido a que sps_sub_layer_ordering_info_present_flag es igual a 0, se infiere que es igual a sps_max_num_reorder_pics[sps_max_sub_layers_minus1].

30 sps_max_latency_increase_plus1[i] no igual a 0 se usa para calcular el valor de SpsMaxLatencyPictures[i], que especifica el número máximo de imágenes que pueden preceder a cualquier imagen en la CVS en orden de salida y seguir a esa imagen en orden de decodificación cuando HighestTid es igual a i.

Cuando sps_max_latency_increase_plus1[i] no es igual a 0, el valor de SpsMaxLatencyPictures[i] se especifica de la siguiente manera:

$$\text{SpsMaxLatencyPictures}[i] = \text{sps_max_num_reorder_pics}[i] + \text{sps_max_latency_increase_plus1}[i] - 1$$

35 Cuando sps_max_latency_increase_plus1[i] es igual a 0, no se expresa ningún límite correspondiente.

El valor de sps_max_latency_increase_plus1[i] estará en el intervalo de 0 a $2^{32} - 2$, inclusive. Cuando sps_max_latency_increase_plus1[i] no está presente para i en el intervalo de 0 a sps_max_sub_layers_minus1 - 1, inclusive, debido a que sps_sub_layer_ordering_info_present_flag es igual a 0, se infiere que es igual a sps_max_latency_increase_plus1[sps_max_sub_layers_minus1].

40

[Tabla 6]

seq_parameter_set_rbsp() {
...
sps_max_sub_layers_minus1
...
pic_width_in_luma_samples
pic_height_in_luma_samples
...
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1);
i <= sps_max_sub_layers_minus1; i++) {
sps_max_dec_pic_buffering_minus1[i]
sps_max_num_reorder_pics[i]
sps_max_latency_increase_plus1[i]
}
...
}

Tabla (6)

Cuando la imagen actual es una imagen de IRAP, se aplica lo siguiente:

5 - Si la imagen actual es una imagen de IDR, una imagen de BLA, la primera imagen en el flujo de bits en orden de decodificación, o la primera imagen que sigue a un final de unidad de NAL de secuencia en orden de decodificación, una variable NoRasIOutputFlag se establece igual a 1.

- De otro modo, si algún medio externo está disponible para establecer una variable HandleCraAsBlaFlag en un valor para la imagen actual, la variable HandleCraAsBlaFlag se establece igual al valor proporcionado por ese medio externo y la variable NoRasIOutputFlag se establece igual a HandleCraAsBlaFlag.

10 - De otro modo, la variable HandleCraAsBlaFlag se establece igual a 0 y la variable NoRasIOutputFlag se establece igual a 0.

Si la imagen actual es una imagen de IRAP con NoRasIOutputFlag igual a 1 que no es una imagen 0, se aplican los siguientes pasos ordenados:

1. La variable NoOutputOfPriorPicsFlag se deriva para el decodificador bajo prueba de la siguiente manera:

15 - Si la imagen actual es una imagen de CRA, NoOutputOfPriorPicsFlag se establece igual a 1 (independientemente del valor de no_output_of_prior_pics_flag).

20 - De otro modo, si el valor de pic_width_in_luma_samples, pic_height_in_luma_samples, o sps_max_dec_pic_buffering_minus1[HighestTid] derivado del SPS activo es diferente del valor de pic_width_in_luma_samples, pic_height_in_luma_samples, o sps_max_dec_pic_buffering_minus1[HighestTid], respectivamente, derivado del SPS activo para la imagen anterior. NoOutputOfPriorPicsFlag puede (pero no debería) ser establecido en 1 por el decodificador bajo prueba, independientemente del valor de no_output_of_prior_pics_flag.

- De otro modo, NoOutputOfPriorPicsFlag se establece igual a no_output_of_prior_pics_flag.

25 2. El valor de NoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica para el HRD de la siguiente manera:

- Si NoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes en el DPB se vacían sin emisión de las imágenes que contienen, y la amplitud de DPB se establece igual a 0.

30 - De otro modo (NoOutputOfPriorPicsFlag es igual a 0), todos los almacenadores temporales de almacenamiento de imágenes que contienen una imagen que está marcada como “no se necesita para la salida” y “no usada para referencia” se vacían (sin salida), y todos los almacenadores temporales de almacenamiento de imágenes que no están vacíos en el DPB se vacían invocando repetidamente el proceso de “sacudidas” 1204, y la amplitud de DPB se establece igual a 0.

35 - De otro modo (la imagen actual no es una imagen de IRAP con NoRasIOutputFlag igual a 1), todos los almacenadores temporales de almacenamiento de imágenes que contienen una imagen que están marcados como “no necesarios para la salida” y “no usados para referencia” se vacían (sin salida). Por cada almacenador temporal de almacenamiento de imágenes que se vacía, la amplitud de DPB se disminuye en uno. Cuando una o más de las siguientes condiciones son ciertas, se invoca repetidamente el proceso de “sacudidas” 1204 mientras se reduce además la amplitud de DPB en uno por cada almacenador temporal de almacenamiento de imágenes adicional que

se vacía, hasta que ninguna de las siguientes condiciones sea cierta:

- 5 1. El número de imágenes con ese valor nuh_layer_id particular en el DPB que están marcadas como “necesarias para la salida” es mayor que sps_max_num_reorder_pics[HighestTid] del conjunto de parámetros de secuencia activo (cuando ese valor nuh_layer_id particular es igual a 0) o del conjunto de parámetros de secuencia de capas activo para ese valor nuh_layer_id particular.
 - 10 2. Si sps_max_latency_increase_plus1[HighestTid] del conjunto de parámetros de secuencia activo (cuando ese valor nuh_layer_id particular es igual a 0) o del conjunto de parámetros de secuencia de capas activo para ese valor nuh_layer_id particular no es igual a 0 y hay al menos una imagen con ese valor nuh_layer_id particular en el DPB que está marcada como “necesaria para la salida” para el cual la variable asociada PicLatencyCount es mayor o igual que SpsMaxLatencyPictures[HighestTid] para ese valor nuh_layer_id particular.
 - 15 3. El número de imágenes con ese valor nuh_layer_id particular en el DPB es mayor o igual que sps_max_dec_pic_buffering[HighestTid] + 1 del conjunto de parámetros de secuencia activo (cuando ese valor nuh_layer_id particular es igual a 0) o del conjunto de parámetros de secuencia de capas activo para ese valor nuh_layer_id particular.
- 15 El proceso de decodificación de imágenes en el bloque 1206 (decodificación y marcado de imágenes) ocurre instantáneamente cuando la última unidad de decodificación de la unidad de acceso que contiene la imagen actual se elimina del CPB.
- 20 Para cada imagen con un valor nuh_layer_id igual al valor nuh_layer_id de la imagen actual en el DPB que está marcada como “necesaria para la salida”, la variable asociada PicLatencyCount se establece igual a PicLatencyCount + 1.
- La imagen actual se considera como decodificada después de que se decodifique la última unidad de decodificación de la imagen. La imagen decodificada actual se almacena en un almacenador temporal de almacenamiento de imágenes vacío en el DPB, y se aplica lo siguiente:
- 25 - Si la imagen decodificada actual tiene PicOutputFlag igual a 1, se marca como “necesaria para la salida” y su variable asociada PicLatencyCount se establece igual a 0.
 - De otro modo (la imagen decodificada actual tiene PicOutputFlag igual a 0), se marca como “no necesaria para la salida”.
- La imagen decodificada actual se marca como “usada para referencia a corto plazo”.
- 30 Cuando una o más de las siguientes condiciones son ciertas, se invoca repetidamente el proceso de “sacudidas” 1208 hasta que ninguna de las siguientes condiciones sea cierta:
- 35 - El número de imágenes con valor nuh_layer_id igual al valor nuh_layer_id de la imagen actual en el DPB que están marcadas como “necesarias para la salida” es mayor que sps_max_num_reorder_pics[HighestTid] del conjunto de parámetros de secuencia activo (cuando el valor nuh_layer_id de la imagen actual es igual a 0) o del conjunto de parámetros de secuencia de capas activo para el valor nuh_layer_id de la imagen actual.
 - sps_max_latency_increase_plus1[HighestTid] del conjunto de parámetros de secuencia activo (cuando el valor nuh_layer_id de imagen actual es igual a 0) o del conjunto de parámetros de secuencia de capas activo para el valor nuh_layer_id de la imagen actual no es igual a 0 y hay al menos una imagen con ese valor nuh_layer_id particular en el DPB que está marcada como “necesaria para la salida” para el cual la variable asociada PicLatencyCount es mayor o igual que SpsMaxLatencyPictures[HighestTid] para ese valor nuh_layer_id particular.
- 40 El proceso de “sacudidas” 1204 y el proceso de sacudidas adicional 1208 son idénticos en términos de los pasos y consisten en los siguientes pasos ordenados: Las imágenes que son las primeras para la salida se seleccionan como las que tienen el valor más pequeño de recuento de orden de imágenes (PicOrderCntVal) de todas las imágenes en el DPB marcadas como “necesarias para la salida”. Un recuento de orden de imágenes es una variable que está asociada con cada imagen, identifica de manera única la imagen asociada entre todas las imágenes en la
- 45 CVS y, cuando la imagen asociada ha de ser emitida desde el almacenador temporal de imágenes decodificadas, indica la posición de la imagen asociada en orden de salida en relación con las posiciones de orden de salida de las otras imágenes en la misma CVS que ha de ser emitida desde el almacenador temporal de imágenes decodificadas.
- 50 - Estas imágenes se recortan, usando la ventana de recorte de conformidad especificada en el conjunto de parámetros de secuencia activo para la imagen con nuh_layer_id igual a 0 o en el conjunto de parámetros de secuencia de capas activo para un valor nuh_layer_id igual al de la imagen, las imágenes recortadas se emiten en orden ascendente de nuh_layer_id, y las imágenes se marcan como “no necesarias para la salida”.
 - Cada almacenador temporal de almacenamiento de imágenes que contiene una imagen marcada como “no usada para referencia” y que incluía una de las imágenes que se recortó y emitió se vacía.

Con referencia a la Figura 13A, como se ha descrito anteriormente, la sintaxis de la cabecera de la unidad de NAL puede incluir dos bytes de datos, esto es, 16 bits. El primer bit es un "forbidden_zero_bit" que siempre se establece en cero al inicio de una unidad de NAL. Los siguientes seis bits son un "nal_unit_type" que especifica el tipo de estructura de datos de cargas útiles de secuencias de bytes en bruto ("RBSP") contenida en la unidad de NAL. Los siguientes 6 bits son un "nuh_reserved_zero_6bits". Los nuh_reserved_zero_6bits pueden ser iguales a 0 en la especificación base del estándar. Se pueden especificar otros valores de nuh_reserved_zero_6bits como se desee. Los decodificadores pueden ignorar (es decir, eliminar el flujo de bits y descartar) todas las unidades de NAL con valores de nuh_reserved_zero_6bits no iguales a 0 cuando se maneja un flujo basado en la especificación base del estándar. En una extensión escalable u otra nuh_reserved_zero_6bits puede especificar otros valores, para señalar codificación de vídeo escalable y/o extensiones de sintaxis. En algunos casos, el elemento de sintaxis nuh_reserved_zero_6bits se puede llamar reserved_zero_6bits. En algunos casos, el elemento de sintaxis nuh_reserved_zero_6bits se puede llamar layer_id_plus1 o layer_id, como se ilustra en la Figura 13B y la Figura 13C. En este caso, el elemento layer_id será layer_id_plus1 menos 1. En este caso, se puede usar para señalar información relacionada con la capa de vídeo codificado escalable. El siguiente elemento de sintaxis es "nuh_temporal_id_plus1". nuh_temporal_id_plus1 menos 1 puede especificar un identificador temporal para la unidad de NAL. El identificador temporal variable TemporalId se puede especificar como TemporalId = nuh_temporal_id_plus1 - 1.

Con referencia a la Figura 14, se ilustra una estructura de sintaxis de unidad de NAL general. La sintaxis de dos bytes de cabecera de la unidad de NAL de la Figura 13 se incluye en la referencia a nal_unit_header() de la Figura 14. El resto de la sintaxis de la unidad de NAL se relaciona principalmente con la RBSP.

Una técnica existente para usar el "nuh_reserved_zero_6bits" es señalar información de codificación de vídeo escalable mediante la partición de los 6 bits de nuh_reserved_zero_6bits en distintos campos de bits, esto es, uno o más de un ID de dependencia, un ID de calidad, un ID de vista y una bandera de profundidad, cada uno de los cuales se refiere a la identificación de una capa diferente del vídeo codificado escalable. Por consiguiente, los 6 bits indican a qué capa de la técnica de codificación escalable pertenece esta unidad de NAL particular. Entonces, en una carga útil de datos, tal como una sintaxis de extensión del conjunto de parámetros de vídeo ("VPS") ("scalability_type") como se ilustra en la Figura 15, se define la información acerca de la capa. La sintaxis de extensión de VPS de la Figura 15 incluye 4 bits para el tipo de escalabilidad (elemento de sintaxis scalability_type) que especifica los tipos de escalabilidad en uso en la secuencia de vídeo codificada y las dimensiones señaladas a través de layer_id_plus1 (o layer_id) en la cabecera de la unidad de NAL. Cuando el tipo de escalabilidad es igual a 0, la secuencia de vídeo codificada se ajusta a la especificación base, de este modo layer_id_plus1 de todas las unidades de NAL es igual a 0 y no hay unidades de NAL que pertenezcan a una capa o vista de mejora. Los valores más altos del tipo de escalabilidad se interpretan como se ilustra en la Figura 16.

La layer_id_dim_len[i] especifica la longitud, en bits, del ID de dimensión de escalabilidad de orden i. La suma de los valores layer_id_dim_len[i] para todos los valores i en el intervalo de 0 a 7 es menor o igual a 6. El vps_extension_byte_alignment_reserved_zero_bit es cero. El vps_layer_id[i] especifica el valor de layer_id de la capa de orden i a la que se aplica la siguiente información de dependencia de capa. El num_direct_ref_layers[i] especifica el número de capas de las que depende directamente la capa de orden i. El ref_layer_id[i][j] identifica la capa de orden j de la que depende la capa de orden i.

De esta manera, la técnica existente señala los identificadores de escalabilidad en la unidad de NAL y en el conjunto de parámetros de vídeo para asignar los bits entre los tipos de escalabilidad enumerados en la Figura 16. Entonces, para cada tipo de escalabilidad, la Figura 16 define cuántas dimensiones se soportan. Por ejemplo, el tipo de escalabilidad 1 tiene 2 dimensiones (es decir, espacial y de calidad). Por cada una de las dimensiones, la layer_id_dim_len[i] define el número de bits asignados a cada una de estas dos dimensiones, donde la suma total de todos los valores de layer_id_dim_len[i] es menor o igual a 6, que es el número de bits en el nuh_reserved_zero_6bits de la cabecera de la unidad de NAL. De este modo, en combinación, la técnica identifica qué tipos de escalabilidad están en uso y cómo los 6 bits de la cabecera de la unidad de NAL se asignan entre la escalabilidad.

Mientras que tal combinación fija de diferentes dimensiones de escalabilidad, como se ilustra en la Figura 16, es adecuada para muchas aplicaciones, hay combinaciones deseables que no están incluidas. Con referencia a la Figura 17, una sintaxis de extensión de conjunto de parámetros de vídeo modificada especifica un tipo de escalabilidad para cada bit en el elemento de sintaxis nuh_reserved_zero_6bits. El vps_extension_byte_alignment_reserved_zero_bit se establece en 0. El max_num_layers_minus1_bits indica el número total de bits usados para el elemento de sintaxis en los primeros dos bytes de la cabecera de la unidad de NAL en la Figura 13 a la que se hace referencia como layer_id_plus1 o nuh_reserved_zero_6bits. El scalability_map[i] especifica el tipo de escalabilidad para cada bit en el elemento de sintaxis layer_id_plus1. En algunos casos, el elemento de sintaxis layer_id_plus1 se puede llamar en su lugar elemento de sintaxis nuh_reserved_zero_6bits o reserved_zero_6bits. El mapa de escalabilidad para todos los bits del elemento de sintaxis layer_id_plus1 juntos especifica la escalabilidad en uso en la secuencia de vídeo codificada. El valor real del identificador para cada uno de los tipos de escalabilidad se señala a través de los bits correspondientes en el campo layer_id_plus1 (nuh_reserved_zero_6bits) en la cabecera de la unidad de NAL. Cuando scalability_map[i] es igual a 0 para todos los valores de i, la secuencia de vídeo codificada se ajusta a la especificación base, de este modo el

valor de `layer_id_plus1` de las unidades de NAL es igual a 0 y no hay unidades de NAL que pertenezcan a una capa o vista de mejora. El `vps_layer_id [i]` especifica el valor de `layer_id` de la capa de orden *i* a la que se aplica la siguiente información de dependencia de capa. El `num_direct_ref_layers[i]` especifica el número de capas del que depende directamente la capa de orden *i*. El `ref_layer_id[i][j]` identifica la capa de orden *j* de la que depende la capa de orden *i*.

Los valores más altos de `scalability_map[i]` se interpretan como se muestra en la Figura 18. El `scalability_map[i]` incluye las dimensiones de escalabilidad de (0) ninguna; (1) espacial; (2) calidad; (3) profundidad; (4) vista múltiple; (5) no especificada; (6) reservada; y (7) reservada.

Por lo tanto, cada bit en la cabecera de la unidad de NAL se interpreta en base a los 3 bits en el conjunto de parámetros de vídeo de lo que es la dimensión de escalabilidad (por ejemplo, ninguna, espacial, calidad, profundidad, vista múltiple, no especificada, reservada). Por ejemplo, para señalar que todos los bits en `layer_id_plus1` corresponden a escalabilidad espacial, los valores de `scalability_map` en el VPS se pueden codificar como 001 001 001 001 001 001 para los 6 bits de la cabecera de la unidad de NAL. También, por ejemplo, para señalar que 3 bits en `layer_id_plus1` corresponden a escalabilidad espacial y 3 bits corresponden a escalabilidad de calidad, los valores de `scalability_map` en el VPS se pueden codificar como 001 001 001 010 010 010 para los 6 bits de la cabecera de la unidad de NAL.

Con referencia a la Figura 19, otra realización incluye el conjunto de parámetros de vídeo que señala el número de dimensiones de escalabilidad en los 6 bits de la cabecera de la unidad de NAL usando el `num_scalability_dimensions_minus1`. El `num_scalability_dimensions_minus1` más 1 indica el número de las dimensiones de escalabilidad señaladas a través de los elementos de sintaxis `layer_id_plus1`; `nuh_reserved_zero_6bits`; y/o `reserved_zero_6bits`. El `scalability_map[i]` tiene la misma semántica que se ha descrito anteriormente en relación con la Figura 17. El `num_bits_for_scalability_map[i]` especifica la longitud en bits para la dimensión de escalabilidad de orden *i*. La suma de todos los `num_bits_for_scalability_map[i]` para $i=0, \dots, \text{num_scalability_dimensions_minus1}$ es igual a seis (o de otro modo igual al número de bits usados para los elementos de sintaxis `layer_id_plus1`; `vps_reserved_zero_6bits`; `max_num_layers_minus1`; `reserved_zero_6bits`; `nuh_reserved_zero_6bits`).

Con respecto a la Figura 17 y la Figura 19, se pueden usar otras variaciones, si se desea. En una realización, por ejemplo, el `scalability_map[i]` se puede señalar con `u(4)` (o `u(n)` con $n>3$ o $n<3$). En este caso, los valores más altos de `scalability_map[i]` se pueden especificar como reservados para flujos de bits que se ajusten a un perfil particular de la técnica de vídeo. Por ejemplo, los valores del mapa de escalabilidad 6..15 se pueden especificar como 'reservados' cuando se señala `scalability_map[i]` con `u(4)`. En otra realización, por ejemplo, `scalability_map[i]` se puede señalar con `ue(v)` o algún otro esquema de codificación. En otra realización, por ejemplo, se puede especificar una restricción de manera que los valores de `scalability_map[i]` estén dispuestos en orden monótono no decreciente (o no creciente). Esto da como resultado que varios campos de dimensión de escalabilidad en el campo `layer_id_plus1` en la cabecera de la unidad de NAL sean contiguos.

Otra técnica existente para señalar la codificación de vídeo escalable usando el elemento de sintaxis "layer_id_plus1" o "nuh_reserved_zero_6bits" es correlacionar el `layer_id_plus1` en la cabecera de la unidad de NAL con una identificación de capa mediante la señalización de una tabla de búsqueda general en el conjunto de parámetros de vídeo. Con referencia a la Figura 20, la técnica existente incluye un conjunto de parámetros de vídeo que especifica el número de tipos de dimensión y las identificaciones de dimensión para la capa de orden *i* de la tabla de búsqueda. En particular, el `vps_extension_byte_alignment_reserved_zero_bit` es cero. El `num_dimensions_minus1[i]` más 1 especifica el número de tipos de dimensión (`dimension_type[i][j]`) y los identificadores de dimensión (`dimension_id[i][j]`) para la capa de orden *i*. El `dimension_type[i][j]` especifica el tipo de dimensión de escalabilidad de orden *j* de la capa de orden *i*, que tiene `layer_id` o `layer_id_plus1` igual a *i*, como se especifica en la Figura 31. Como se ilustra en la Figura 21, las dimensiones que se identifican incluyen de (0) orden de vista *idx*; (1) bandera de profundidad; (2) ID de dependencia; (3) ID de calidad; (4)-(15) reservados. El `dimension_id[i][j]` especifica el identificador del tipo de dimensión de escalabilidad de orden *j* de la capa de orden *i*, que cuando no está presente se infiere que es 0. El `num_direct_ref_layers[i]` especifica el número de capas del que depende directamente la capa de orden *i*. El `ref_layer_id[i][j]` identifica la capa de orden *j* de la que depende directamente la capa de orden *i*. Desafortunadamente, la realización propuesta ilustrada en la Figura 20 da como resultado una tabla de búsqueda grande difícil de manejar.

Con referencia a la Figura 22, una extensión del conjunto de parámetros de vídeo modificado incluye una máscara de escalabilidad que se usa en combinación con una dimensión de escalabilidad. La `scalability_mask` señala un patrón de bits 0 y 1 con cada bit correspondiente a una dimensión de escalabilidad como se indica por la sintaxis del mapa de escalabilidad de la Figura 23. Un valor de 1 para una dimensión de escalabilidad particular indica que esta dimensión de escalabilidad está presente en esta capa (capa de orden *l*). Un valor de 0 para una dimensión de escalabilidad particular indica que esta dimensión de escalabilidad no está presente en esta capa (capa de orden *l*). Por ejemplo, un conjunto de bits de 00100000 se refiere a escalabilidad de calidad. El valor del identificador real de la dimensión de escalabilidad particular que está presente se indica por el valor de `scalability_id[j]` señalado. Los valores de `num_scalability_types[i]` es igual a la suma del número de bits en la `scalability_mask` que tiene un valor de 1. De este modo

[Expresión Matemática 9]

$$\text{num_scalability_types}[i] = \sum_{k=0}^7 \text{scalability_mask}[i](k)$$

El scalability_id[j] indica el valor de identificador de dimensión de escalabilidad de orden j para el tipo de valores de escalabilidad que están señalados por el valor de scalability_mask.

5 Con referencia a la Figura 24, una modificación de la Figura 22, incluye la máscara de escalabilidad que se señala fuera del bucle. Esto da como resultado una máscara común para cada identificación de capa. Con referencia a la Figura 25, en esta modificación un conjunto de parámetros de vídeo ejemplar correspondiente puede incluir la identificación escalable con la máscara de escalabilidad que no se incluye. En este caso, el elemento de sintaxis scalable_id[j] tiene la misma interpretación que el elemento de sintaxis scalability_id[j] en la Figura 22.

10 Con referencia a la Figura 26, una modificación de la Figura 22 incluye la máscara de escalabilidad (scalability_mask) que se señala fuera del bucle. Esto da como resultado una máscara común para cada identificación de capa. La scalability_mask señala un patrón de bits 0 y 1 con cada bit correspondiente a una dimensión de escalabilidad como se indica por la sintaxis del mapa de escalabilidad de la Figura 27. Un valor de 1 para una dimensión de escalabilidad particular indica que esta dimensión de escalabilidad está presente en esta capa (capa de orden i). Un valor de 0 para una dimensión de escalabilidad particular indica que esta dimensión de escalabilidad no está presente en esta capa (capa de orden i). Por ejemplo, un conjunto de bits de 00100000 se refiere a la escalabilidad de calidad. El valor de identificador real de la dimensión de escalabilidad particular que está presente se indica por el valor de scalability_id[j] señalado. Los valores de num_scalability_types [i] es igual a la suma del número de bits en la scalability_mask que tiene el valor de 1. De este modo

20 [Expresión matemática 10]

$$\text{NumScalabilityTypes}[i] = \sum_{k=0}^{15} \text{scalability_mask}(k)$$

En este caso, la variable scalability_id[j] se puede llamar en su lugar variable dimension_id[i][j]. dimension_id[i][j] especifica el identificador de escalabilidad de la dimensión de escalabilidad de orden j de la capa de orden i. Entonces, una variable ScalabilityId[i][j] se deriva de la siguiente manera.

25 [Tabla 7]

for(i = 1; i <= vps_max_layers_minus1; i++) {
for(k=0, j=0; k<=15; k++) {
if(scalability_mask(k)==1)
ScalabilityId [i][k]=dimension_id[i][j++]
else
ScalabilityId [i][k]=0;
}
}

Donde el ScalabilityId[i][k] señala el ID de dimensión para el tipo de escalabilidad correspondiente de la siguiente manera.

[Tabla 8]

k	ScalabilityId[i][k]
0	DependencyId[i][k]
1	QualityId[i][k]
2	depthFlag[i][k]
3	ViewId[i][k]

4-15	Reservado
------	-----------

5 Donde DependencyId[i][1] es el ID de dependencia para la dimensión de escalabilidad espacial para la capa de orden i, QualityId[i][2] es el ID de calidad para la dimensión de escalabilidad de calidad para la capa de orden i, depthFlag[i][3] es la bandera de profundidad/ID de profundidad para la dimensión de escalabilidad de profundidad para la capa de orden i, y ViewId[i][4] es el ID de vista para la dimensión de escalabilidad de vista múltiple para la capa de orden i.

También en la Figura 26 avc_base_codec_flag igual a 1 especifica que la capa base se ajusta a la Rec. ITU-T H.264 | ISO/IEC 14496-10, y avc_base_codec_flag igual a 1 especifica a HEVC. vps_nuh_layer_id_present_flag indica si está señalada la variable layer_id_in_nuh[i] que señala el valor de layer_id en la cabecera de la unidad de NAL.

10 En otra realización, uno o más de los elementos de sintaxis scalability_mask[i], scalability_mask, scalability_id[j] se puede señalar usando un número diferente de bits que u(8). Por ejemplo, se podrían señalar con u(16) (o u(n) con n>8 o n<8). En otra realización, uno o más de estos elementos de sintaxis se podrían señalar con ue(v). En otra realización, la scalability_mask se puede señalar en la cabecera de la unidad de NAL en los elementos de sintaxis layer_id_plus1; vps_reserved_zero_6bits; max_num_layers_minus1; reserved_zero_6bits; y/o
 15 nuh_reserved_zero_6bits. En algunas realizaciones, el sistema puede hacer esto solamente para unidades de NAL de VPS, o solamente para unidades de NAL no VPS, o para todas las unidades de NAL. En otra realización más, scalability_mask se puede señalar por imagen en cualquier lugar en el flujo de bits. Por ejemplo, se puede señalar en la cabecera de porción, conjunto de parámetros de imagen, conjunto de parámetros de vídeo, o cualquier otro conjunto de parámetros o cualquier otra parte normativa del flujo de bits.

20 Se debería observar que las FIG. 13, 15, 18, 20, 21, 22, 23 y la descripción correspondiente se refieren a 6 bits, dado que el elemento de sintaxis nuh_reserved_zero_6bits o layer_id_plus1 en la cabecera de la unidad de NAL de la Figura 13 tiene 6 bits. No obstante, toda la descripción anterior se puede modificar adecuadamente si ese elemento de sintaxis usó un número de bits diferente de 6 bits. Por ejemplo, si ese elemento de sintaxis (nuh_reserved_zero_6bits o layer_id_plus1) en su lugar usó 9 bits, entonces en la Figura 17 el valor de
 25 max_num_layer_minus1 bits será 9 y el scalability_map[i] se señalará para cada uno de los 9 bits en lugar de los 6 bits.

Con referencia a la FIG. 24, una modificación de la FIG. 22 proporciona sintaxis para señalar información de dependencia de capa. Se define un nuevo elemento de sintaxis layer_dependency_information_pattern.

30 layer_dependency_information_pattern señala un patrón de bits 0 y 1 con la longitud igual a vps_max_layers_minus1. Un valor de 0 para el bit de orden i indica que la capa con layer_id (i+1) es una capa independiente. Un valor de 1 para el bit de orden i indica que la capa con layer_id (i+1) es una capa dependiente que depende de una o más de las otras capas.

Los valores de NumDepLayers son iguales a la suma del número de bits en el layer_dependency_information_pattern que tiene un valor de 1. De este modo

35 [Expresión matemática 11]

$$\text{NumDepLayers} = \sum_{k=0}^{\text{vps_max_layer_minus1}-1} \text{layer_dependency_information_pattern}(k)$$

40 Con referencia a la Figura 29, una modificación de la Figura 26 proporciona una sintaxis para señalar información de dependencia de capa. Se define un nuevo elemento de sintaxis layer_dependency_flag[i]. layer_dependency_flag[i] señala si una capa depende de otras capas. Un valor de 0 para la bandera indica que la capa con layer_id i es una capa independiente. Un valor de 1 para el bit de orden i indica que la capa con layer_id i es una capa dependiente.

45 Con referencia a la Figura 30, una modificación de la Figura 26 proporciona una sintaxis para señalar información de dependencia de capa. Se define un nuevo elemento de sintaxis layer_dependency_map[i]. layer_dependency_map[i] señala un patrón de bits 0 y 1 con la longitud igual a vps_max_layers_minus1. Un valor de 0 para el bit de orden k de layer_dependency_map[i] indica que la capa i no depende de la capa con layer_id (k+1). Un valor de 1 para el bit de capa k de layer_dependency_map[i] indica que la capa i depende de la capa con layer_id (k+1).

Con referencia a la Figura 31, una modificación de la Figura 26 proporciona una sintaxis para señalar información de dependencia de capa. Se define un nuevo elemento de sintaxis layer_dependency_information_pattern.

50 layer_dependency_information_pattern señala un patrón de bits 0 y 1 con la longitud igual a vps_max_layers_minus1. Un valor de 0 para el bit de orden i indica que la capa con layer_id (i+1) es una capa independiente. Un valor de 1 para el bit de orden i indica que la capa con layer_id (i+1) es una capa dependiente que depende de una o más de las otras capas. Los valores de NumDepLayers son iguales a la suma del número de bits

en el layer_dependency_information_pattern que tiene un valor de 1. De este modo

[Expresión matemática 12]

$$\text{NumDepLayers} = \sum_{k=0}^{\text{vps_max_layer_minus1}-1} \text{layer_dependency_information_pattern}(k)$$

5 layer_dependency_map[i] señala un patrón de bits 0 y 1 con la longitud igual a vps_max_layers_minus1. Un valor de 0 para el bit de orden k de layer_dependency_map[i] indica que la capa i no depende de la capa con layer_id (k+1). Un valor de 1 para el bit de orden k de layer_dependency_map[i] indica que la capa i depende de la capa con layer_id (k+1).

10 Con referencia a la Figura 32, una modificación de la Figura 26 proporciona una sintaxis para señalar información de dependencia de capa. La FIG. 28 es una variante de sintaxis basada en la sintaxis de la FIG. 27. Se define un nuevo elemento de sintaxis layer_dependency_information_pattern.

layer_dependency_information_pattern señala un patrón de bits 0 y 1 con la longitud igual a vps_max_layers_minus1. Un valor de 0 para el bit de orden i indica que la capa con layer_id (i+1) es una capa independiente. Un valor de 1 para el bit de orden i indica que la capa con layer_id (i+1) es una capa dependiente que depende de una o más de las otras capas.

15 Los valores de NumDepLayers son iguales a la suma del número de bits en el layer_dependency_information_pattern que tiene un valor de 1. De este modo

[Expresión matemática 13]

$$\text{NumDepLayers} = \sum_{k=0}^{\text{vps_max_layer_minus1}-1} \text{layer_dependency_information_pattern}(k)$$

20 Los elementos de sintaxis num_direct_ref_layers[i] y ref_layer_id[i][j] se señalan solamente cuando layer_dependency_information_pattern(i) tiene un valor de 1. Donde layer_dependency_information_pattern(i) es el bit de orden i del elemento de sintaxis layer_dependency_information_pattern.

Con referencia a la Figura 33, una modificación de la Figura 26 proporciona una sintaxis para señalar información de dependencia de capa. La FIG. 29 es una variante de sintaxis basada en la sintaxis en la Figura 31. Se define un nuevo elemento de sintaxis layer_dependency_information_pattern.

25 layer_dependency_information_pattern señala un patrón de bits 0 y 1 con la longitud igual a vps_max_layers_minus1. Un valor de 0 para el bit de orden i indica que la capa con layer_id (i+1) es una capa independiente. Un valor de 1 para el bit de orden i indica que la capa con layer_id (i+1) es una capa dependiente que depende de una o más de las otras capas.

30 Los valores de NumDepLayers son iguales a la suma del número de bits en el layer_dependency_information_pattern que tiene un valor de 1. De este modo

[Expresión matemática 14]

$$\text{NumDepLayers} = \sum_{k=0}^{\text{vps_max_layer_minus1}-1} \text{layer_dependency_information_pattern}(k)$$

35 layer_dependency_map[i] señala un patrón de bits 0 y 1 con la longitud igual a vps_max_layers_minus1. Un valor de 0 para el bit de orden k de layer_dependency_map[i] indica que la capa i no depende de la capa con layer_id (k+1). Un valor de 1 para el bit de orden k de layer_dependency_map[i] indica que la capa i depende de la capa con layer_id (k+1). Los elementos de sintaxis layer_dependency_map[i] se señalan solamente cuando layer_dependency_information_pattern(i) tiene un valor de 1. Donde layer_dependency_information_pattern(i) es el bit de orden i del elemento de sintaxis layer_dependency_information_pattern.

40 En otra realización, el elemento de sintaxis de layer_dependency_information_pattern se puede señalar como un conjunto de valores de bandera de 1 bit. En este caso, un total de vps_max_layers_minus1 valores de 1 bit se señalarán como:

[Tabla 9]

```

for( i = 1; i <= vps_max_layers_minus1 ; i++ )
{
    layer_dependency_information_pattern_flags[i];
}
    
```

En otra realización, el elemento de sintaxis de `layer_dependency_map[i]` se puede señalar como un conjunto de valores de bandera de 1 bit. En este caso, un total de `vps_max_layers_minus1` valores de 1 bit se señalarán como:

[Tabla 10]

```

for( j = 1; j <= vps_max_layers_minus1 ; j++ )
{
    layer_dependency_map_values[i][j];
}
    
```

5

En otra realización uno o más de los elementos de sintaxis `layer_dependency_information_pattern`, `layer_dependency_map` se puede señalar usando un número fijo conocido de bits en lugar de `u(v)`. Por ejemplo se podrían señalar usando `u(64)`.

10

En otra realización, uno o más de o más de los elementos de sintaxis `layer_dependency_information_pattern`, `layer_dependency_map` se pueden señalar con `ue(v)` o algún otro esquema de codificación.

En otra realización, los nombres de varios elementos de sintaxis y su semántica se puede alterar añadiendo un más 1 o más 2 o restando un menos 1 o un menos 2 en comparación con la sintaxis y la semántica descritas.

15

En otra realización más, varios elementos de sintaxis tales como `layer_dependency_information_pattern`, `layer_dependency_map`, `layer_dependency_flag[i]` etc. se puede señalar por imagen en cualquier lugar en el flujo de bits. Por ejemplo, se puede señalar en la cabecera de porción, `pps/sps/vps/aps` o cualquier otro conjunto de parámetros u otra parte normativa del flujo de bits.

20

Como se ha descrito anteriormente, la codificación de vídeo escalable es una técnica de codificación de un flujo de bits de vídeo que también contiene uno o más flujos de bits de subconjunto. Se puede derivar un flujo de bits de vídeo de subconjunto dejando caer paquetes del vídeo más grande para reducir el ancho de banda requerido para el flujo de bits de subconjunto. El flujo de bits del subconjunto puede representar una resolución espacial más baja (pantalla más pequeña), una resolución temporal más baja (tasa de cuadros más baja) o una señal de vídeo de calidad más baja. Por ejemplo, un flujo de bits de vídeo puede incluir 5 flujos de bits de subconjunto, donde cada uno de los flujos de bits de subconjunto añade contenido adicional a un flujo de bits base.

25

Como se ha descrito anteriormente, la codificación de vídeo de vista múltiple es una técnica de codificación de un flujo de bits de vídeo que también contiene uno o más de otros flujos de bits representativos de las vistas alternativas. Por ejemplo, las vistas múltiples pueden ser un par de vistas para vídeo estereoscópico. Por ejemplo, las vistas múltiples pueden representar vistas múltiples de la misma escena desde diferentes puntos de vista. Las vistas múltiples generalmente contienen una gran cantidad de dependencias estadísticas entre vistas, dado que las imágenes son de la misma escena desde diferentes puntos de vista. Por lo tanto, la predicción combinada temporal y entre vistas puede lograr una codificación de vista múltiple eficiente. Por ejemplo, un cuadro se puede predecir de manera eficiente no solamente a partir de cuadros relacionados temporalmente, sino también a partir de los cuadros de puntos de vista colindantes.

30

Chen, et al., "SHVC Draft Text 1," JCTVC-L1008, Ginebra, enero de 2013; Hannuksela, et al. "Test Model for Scalable Extensions of High Efficiency Video Coding (HEVC)," JCTVC-L0453-spec-text, Shanghai, octubre de 2012; y Hannuksela, "Draft Text for Multiview Extension of High Efficiency Video Coding (HEVC)," JCTVC-L0452-spec-text-r1, Shanghai, octubre de 2012 cada uno tiene el concepto de puntos de operación de salida. Esto se cambió más tarde al concepto de conjuntos de capas de salida en JCTVC-L1008. Para los elementos de sintaxis, `num_output_operation_points`, `output_op_point_index[]`, y `output_layer_flag[][]` se definen para los puntos de operación de salida para JCTVC-L0453 y LCTVC-L0452, y modifican a `num_output_layer_sets`, `output_layer_set_idx[]`, y `output_layer_flag[][]` en JCTVC-L1008. En cada uno de tales documentos, se define un mensaje de SEI de cambio de dependencia de capas. El mensaje de SEI permite señalar cambios de información de dependencia de capas comenzando con la unidad de acceso actual que contiene el mensaje de SEI.

35

40

45

Chen, et al., "SHVC Draft Text 1," JCTVC-L1008, Ginebra, enero de 2013; Hannuksela, et al. "Test Model for Scalable Extensions of High Efficiency Video Coding (HEVC)," JCTVC-L0453-spec-text, Shanghai, octubre de 2012; y Hannuksela, "Draft Text for Multiview Extension of High Efficiency Video Coding (HEVC)," JCTVC-L0452-spec-text-r1, Shanghai, octubre de 2012 cada uno tiene un almacenador temporal de imágenes decodificadas (DPB) de orden

de salida que opera en base al uso de los elementos de sintaxis `sps_max_num_reorder_pics[HighestTid]`, `sps_max_latency_increase_plus1[HighestTid]` y `sps_max_dec_pic_buffering[HighestTid]` para la salida y eliminación de imágenes 0 del DPB. Esta información se señala en el conjunto de parámetros de vídeo para la capa base, que proporciona información de almacenamiento temporal para el contenido de vídeo que incluye las capas de mejora, en su caso.

Con referencia a la Figura 34, se ilustra otro ejemplo de un conjunto de parámetros de vídeo. En particular, el conjunto de parámetros de vídeo de la Figura 34 incluye referencia a una extensión de conjunto de parámetros de vídeo (`vps_extension`) asociada. La `vps_extension` se puede señalar con la `vps_extension_flag`, la `vps_extension2_flag` y/o la `vps_extension_data_flag`.

Con referencia a la Figura 35, se ilustra otro ejemplo de una extensión de conjunto de parámetros de vídeo. En particular, la extensión del conjunto de parámetros de vídeo de la Figura 35 incluye una referencia a `num_output_layer_sets`, que define esas capas de un conjunto de capas de un flujo de bits que se puede emitir al espectador por el decodificador. `num_output_layer_sets` especifica el número de conjuntos de capas para las cuales las capas de salida se especifican con `output_layer_set_index[i]` y `output_layer_flag[lslidx][j]`. Cuando no está presente, se infiere que el valor de `num_output_layer_sets` es igual a 0. Un conjunto de capas que describe las capas de salida es un conjunto de capas de salida.

El `output_layer_set_idx[i]` especifica el índice `lslidx` del conjunto de capas para el que está presente `output_layer_flag[lslidx][j]`.

La `output_layer_flag[lslidx][j]` igual a 1 especifica que la capa con `nuh_layer_id` igual a `j` es una capa de salida objetivo del conjunto de capas de orden `lslidx`. Un valor de `output_layer_flag[lslidx][j]` igual a 0 especifica que la capa con `nuh_layer_id` igual a `j` no es la capa de salida objetivo del conjunto de capas de orden `lslidx`.

Con los `num_output_layer_sets` que se definen en `vps_extension` de la manera descrita, los `num_output_layer_sets` se pueden actualizar enviando un nuevo conjunto de parámetros de vídeo junto con una extensión del conjunto de parámetros de vídeo correspondiente. Desafortunadamente, el envío del nuevo conjunto de parámetros de vídeo junto con la extensión del conjunto de parámetros de vídeo correspondiente da como resultado una reducción significativa en el ancho de banda disponible. Además, un nuevo conjunto de parámetros de vídeo solamente se podría activar en ciertos tipos de imágenes, por ejemplo, en tipos de imágenes de puntos de acceso intra aleatorios.

Como ejemplo, un flujo de bits puede incluir una capa base 0 y cuatro capas de mejora, esto es, la capa de mejora 1, la capa de mejora 2, la capa de mejora 3 y la capa de mejora 4. Un primer conjunto de capas puede ser la capa base 0 y la capa de mejora 1. Un segundo conjunto de capas puede ser la capa base 0, la capa de mejora 1 y la capa de mejora 2. Un tercer conjunto de capas puede ser la capa base 0, la capa de mejora 1 y la capa de mejora 3. Un cuarto conjunto de capas puede ser la capa base 0 y la capa de mejora 4. Los conjuntos de capas de salida definen las capas particulares de un conjunto de capas que se pueden proporcionar como salida. Por ejemplo, el conjunto de capas de salida 1 puede ser la capa 0 que es una salida y la capa 1 que no es una salida. Por ejemplo, el conjunto de capas de salida 2 puede ser la capa 0 que es una salida, la capa 1 que es una salida y la capa 2 que no es una salida. Por ejemplo, el conjunto de capas de salida 3 puede ser la capa 0 que no es una salida, la capa 1 que es una salida y la capa 3 que es una salida. Por ejemplo, el conjunto de capas de salida 4 puede ser la capa 0 que es una salida y la capa 4 que no es una salida. Un número limitado de capas de salida es útil para adaptar capacidades limitadas de decodificación. En algunos casos, es deseable deshabilitar algunas dependencias entre diferentes capas para adaptarse a la decodificación de otras capas y/o capacidades del decodificador. Cuando algunas dependencias de capas cambian, del mismo modo puede ser deseable habilitar, deshabilitar y/o añadir otras capas como capas de salida. Como ejemplo, un cambio de dependencia de capa podría requerir que menos capas a ser decodificadas decodifiquen una capa objetivo particular (por ejemplo, una vista en MV-HEVC). De este modo, un decodificador podría decodificar y emitir una capa (por ejemplo, y una vista adicional) después de un cambio de dependencia de capa que anteriormente no podría debido a su hardware. Actualmente no existe ningún mecanismo para permitir señalar un cambio en los conjuntos de capas de salida sin enviar un nuevo conjunto de parámetros de vídeo.

Con referencia a la Figura 36, se puede usar una sintaxis de mensaje de SEI para facilitar los cambios de los conjuntos de capas de salida, esto es, `output_layer_sets_change(payloadSize)`. Este mensaje de SEI indica que los cambios de información de conjuntos de capas de salida que comienzan con la unidad de acceso actual que contienen el mensaje de SEI y se interpreta con respecto al conjunto de parámetros de vídeo activo. Se puede incluir más de un conjunto de parámetros de vídeo con el flujo de bits, y el conjunto de parámetros de vídeo activo se cambia, según se desee. Cuando está presente, el mensaje de SEI de cambio de conjuntos de capas de salida se aplica al conjunto de unidades de acceso objetivo que consiste en la unidad de acceso actual y todas las unidades de acceso posteriores, en orden de decodificación, hasta el siguiente mensaje de SEI de cambio de conjuntos de capas de salida o el final de la secuencia de vídeo codificado, lo que sea antes en orden de decodificación. Los mensajes de SEI de cambio de conjuntos de capas de salida, preferiblemente, no tienen un efecto acumulativo.

El `'active_vps_id'` identifica un conjunto de parámetros de vídeo activo que contiene la información de conjuntos de capas de salida a la que se aplica el cambio o la adición. El valor de `active_vps_id` es igual al valor de

video_parameter_set_id del conjunto de parámetros de vídeo activo para las unidades de NAL de VCL de la unidad de acceso que contiene el mensaje de SEI.

5 El 'num_changed_output_layer_sets' especifica el número de conjuntos de capas de salida cambiado para los cuales las capas de salida se especifican en la sección de extensión de VPS del conjunto de parámetros de vídeo activo identificado por active_vps_id. El valor de num_changed_output_layer_sets debería estar en el intervalo de 0 a num_output_layer_sets, inclusive. Cuando no está presente, el valor de num_changed_output_layer_sets se infiere que es igual a 0.

10 La 'changed_output_layer_sets_idx_entry'[i] identifica la entrada del índice en la lista de entradas del conjunto de capas de salida en la extensión de VPS identificada por active_vps_id para la cual output_layer_flag[clsIdx][j] se aplica el cambio. El valor de changed_output_layer_set_idx_entry[i] debería estar en el intervalo de 0 a num_output_layer_sets, inclusive. La change_output_layer_sets_idx_entry[i] no es igual a changed_output_layer_sets_idx_entry[j] para cualquier j que no es igual a i.

15 La 'output_layer_flag'[clsIdx][j] igual a 1 especifica que la capa con nuh_layer_id igual a j es una capa de salida objetivo del conjunto de capas de orden clsIdx. Un valor de output_layer_flag[clsIdx][j] igual a 0 especifica que la capa con nuh_layer_id igual a j no es una capa de salida objetivo del conjunto de capas de orden clsIdx.

Cuando output_layer_flag[clsIdx][j] no está presente para clsIdx en el intervalo de 0 a vps_num_layer_sets_minus1, inclusive y para j en el intervalo de 0 a 63, inclusive, output_layer_flag[clsIdx][j] se infiere que es igual a (j = LayerSetLayerIdList[clsIdx][NumLayersInIdList[clsIdx] - 1]).

20 El 'num_addl_output_layer_sets' especifica el número de conjuntos de capas adicionales para los cuales se especifican capas de salida con addl_output_layer_set_idx[i] y output_layer_flag[addlIdx][j]. Cuando no está presente, el valor de num_addl_output_layer_sets se infiere que es igual a 0. El 'addl_output_layer_sets_idx'[i] identifica el índice addlIdx del conjunto de capas para el que está presente output_layer_flag[addlIdx][j].

25 El 'output_layer_flag'[addlIdx][j] igual a 1 especifica que la capa con nuh_layer_id igual a j es una capa de salida objetivo del conjunto de capas de orden addlIdx. Un valor de output_layer_flag[addlIdx][j] igual a 0 especifica que la capa con nuh_layer_id igual a j no es la capa de salida objetivo del conjunto de capas de orden addlIdx.

Cuando output_layer_flag[addlIdx][j] no está presente para addlIdx en el intervalo de 0 a vps_num_layer_sets_minus1, inclusive y para j en el intervalo de 0 a 63, inclusive, output_layer_flag[addlIdx][j] se infiere que es igual a (j = LayerSetLayerIdList[addlIdx][NumLayersInIdList[addlIdx] - 1]).

30 Cuando output_layer_flag[addlIdx][j] no está presente para addlIdx en el intervalo de 0 a vps_num_layer_sets_minus1, inclusive y para j en el intervalo de 0 a 63, inclusive, y cuando output_layer_flag[clsIdx][j] no está presente para clsIdx en el intervalo de 0 a vps_num_layer_sets_minus1, inclusive y para j en el intervalo de 0 a 63, inclusive, output_layer_flag[addlIdx][j] se infiere que es igual a (j = LayerSetLayerIdList[addlIdx][NumLayersInIdList[addlIdx] - 1]).

35 Cuando output_layer_flag[addlIdx][j] no está presente para addlIdx en el intervalo de 0 a vps_num_layer_sets_minus1, inclusive y para j en el intervalo de 0 a 63, inclusive, y cuando output_layer_flag[clsIdx][j] no está presente para clsIdx en el intervalo de 0 a vps_num_layer_sets_minus1, inclusive y para j en el intervalo de 0 a 63, inclusive, output_layer_flag[clsIdx][j] se infiere que es igual a (j = LayerSetLayerIdList[clsIdx][NumLayersInIdList[clsIdx] - 1]).

40 Se puede derivar una lista objetivo de capas de salida, si se desea. La lista de identificadores de capas de salida objetivo, targetOpLayerIdList, especifica la lista de valores de nuh_layer_id, en orden creciente de los valores de nuh_layer_id que son capas de salida para el conjunto de capas de salida seleccionado, se puede derivar de la siguiente manera:

Para un conjunto de capas de salida seleccionado oLsIdx, una lista de identificadores de capas de salida objetivo targetOpLayerIdList se deriva como:

45 [Tabla 11]

for(k=0, numOutputLayers=0;k<=vps_max_layer_id;k++)
if(output_layer_flag[oLsIdx][k])
targetOpLayerIdList[numOutputLayers++]=layer_id_in_nuh[k]

Un conjunto de capas de salida objetivo se puede identificar por la lista de identificadores de capas de salida objetivo asociada targetOpLayerIdList.

El conjunto de capas de salida seleccionado oLsIdx está entre 0 y num_output_layer_sets si no hay mensajes de

SEI de cambio de conjuntos de capas de salida.

El conjunto de capas de salida seleccionado oLsIdx está entre 0 y num_output_layer_sets + num_addl_output_layer_sets si hay presentes mensajes de SEI de cambio de conjuntos de capas de salida.

5 Se puede derivar una lista objetivo de capas decodificadas, si se desea. La lista de identificadores de capas decodificadas objetivo, targetDLayerIdList, especifica la lista de los valores de nuh_layer_id, que necesitan ser decodificados para el conjunto de capas de salida seleccionado, se puede derivar de la siguiente manera:

Para un conjunto de capas de salida seleccionado oLsIdx, una lista de identificadores de capas de salida objetivo targetOpLayerIdList se deriva como:

[Tabla 12]

for(k=0, numOutputLayers=0;k<=vps_max_layer_id;k++)
if(output_layer_flag[oLsIdx][k])
targetOpLayerIdList [numOutputLayers++]=layer_id_in_nuh[k]

10

Una lista de identificadores de capas decodificadas objetivo targetDLayerIdList se puede derivar como:

[Tabla 13]

for(m=0, numDecodedLayers=0;m< numOutputLayers;m++) {
for(n=0;n<NumDirectRefLayers[LayerIdInVps[targetOpLayerIdList[m]]];n++) {
rLid=RefLayerId[LayerIdInVps[targetOpLayerIdList[m]]][n]
if(rLid not included in targetDLayerIdList[0,..., numDecodedLayers])
targetDLayerIdList[numDecodedLayers++]=rLid;
}
}

15

En algún caso, la lista targetDLayerIdList final se puede obtener clasificando la targetDLayerIdList en orden creciente de los valores nuh_layer_id.

Un conjunto de capas decodificadas objetivo se identifica mediante la lista de identificadores de capas decodificadas objetivo targetDLayerIdList asociada. En algún caso, la targetDLayerIdList puede ser la misma que la lista de identificadores de capas TargetDecLayerIdList de JCTVC-L1008, que especifica la lista de valores nuh_layer_id, en orden creciente de los valores nuh_layer_id, de las unidades de NAL a ser decodificadas:

20

El conjunto de capas de salida seleccionado oLsIdx está entre 0 y num_output_layer_sets si no hay mensajes de SEI de cambio de conjuntos de capas de salida.

El conjunto de capas de salida seleccionado oLsIdx está entre 0 y num_output_layer_sets + num_addl_output_layer_sets si hay presentes mensajes de SEI de cambio de conjuntos de capas de salida.

25

Con referencia a la Figura 37, se puede usar otra sintaxis de mensaje de SEI para facilitar el cambio de los conjuntos de capas de salida, esto es, output_layer_sets_change(payloadSize).

El 'num_deleted_output_layer_sets' especifica el número de conjuntos de capas de salida que se borran y así ya no están más presentes. El valor de num_deleted_output_layer_sets está en el intervalo de 0 a num_output_layer_sets, inclusive. Cuando no está presente, el valor de num_deleted_output_layer_sets se infiere que es igual a 0.

30

La 'deleted_output_layer_sets_idx_entry'[i] identifica la entrada de índice en la lista de entradas de conjunto de capas de salida que se indica que sean borradas y así ya no están más presentes. El valor de deleted_output_layer_set_idx_entry[i] debería estar en el intervalo de 0 a num_output_layer_sets, inclusive. La deleted_output_layer_sets_idx_entry[i] no es igual a deleted_output_layer_sets_idx_entry[j] para cualquier j que no sea igual a i.

El 'num_changed_output_layer_sets' especifica el número de conjuntos de capas de salida cambiados para los cuales las capas de salida se especifican en la sección de extensión de vps del conjunto de parámetros de vídeo activo identificado por active_vps_id. El valor de num_changed_output_layer_sets debería estar en el intervalo de 0 a num_output_layer_sets - num_deleted_output_layer_sets, inclusive. Cuando no está presente, el valor de num_changed_output_layer_sets se infiere que es igual a 0.

5

Se puede derivar una lista objetivo de capas de salida, si se desea. La lista de identificadores de capas de salida objetivo, targetOpLayerIdList, especifica la lista de valores de nuh_layer_id, en orden creciente de los valores nuh_layer_id que son capas de salida para el conjunto de capas de salida seleccionado, se puede derivar de la siguiente manera:

10

Para un conjunto de capas de salida seleccionado oLsIdx, una lista de identificadores de capas de salida objetivo targetOpLayerIdList se deriva como:

[Tabla 14]

for(k=0, numOutputLayers=0;k<=vps_max_layer_id;k++)
if(output_layer_flag[oLsIdx][k])
targetOpLayerIdList [numOutputLayers++]=layer_id_in_nuh[k]

15

Un conjunto de capas de salida objetivo se identifica mediante la lista de identificadores de capas de salida objetivo targetOpLayerIdList asociada.

El conjunto de capas de salida seleccionado oLsIdx está entre 0 y num_output_layer_sets si no hay mensajes de SEI de cambio de conjuntos de capas de salida.

20

El conjunto de capas de salida seleccionado oLsIdx está entre 0 y num_output_layer_sets - num_deleted_output_layer_sets + num_addl_output_layer_sets si hay presentes mensajes de SEI de cambio de conjuntos de capas de salida.

Se puede derivar una lista objetivo de capas decodificadas, si se desea.

Una lista de identificadores de capas decodificadas objetivo, TargetDLayerIdList, especifica la lista de valores nuh_layer_id, que necesitan ser decodificados para el conjunto de capas de salida seleccionado se puede derivar de la siguiente manera:

25

Para un conjunto de capas de salida seleccionado oLsIdx, una lista de identificadores de capas de salida objetivo targetOpLayerIdList se puede derivar como:

[Tabla 15]

for(k=0, numOutputLayers=0;k<=vps_max_layer_id;k++)
if(output_layer_flag[oLsIdx][k])
targetOpLayerIdList [numOutputLayers++]=layer_id_in_nuh[k]

Entonces, una lista de identificadores de capas decodificadas objetivo targetDLayerIdList se puede derivar como:

30

[Tabla 16]

for(m=0, numDecodedLayers=0;m< numOutputLayers;m++) {
for(n=0;n<NumDirectRefLayers[layerIdInVps[targetOpLayerIdList[m]]];n++) {
rLid=RefLayerId[layerIdInVps[targetOpLayerIdList[m]]][n]
if(rLid not included in targetDLayerIdList[0,..., numDecodedLayers])
targetDLayerIdList[numDecodedLayers++]=rLid;
}
}

En algún caso, la targetDLayerIdList final se puede obtener clasificando la targetDLayerIdList anterior en orden creciente de valores de nuh_layer_id.

5 Un conjunto de capas decodificadas objetivo se identifica por la lista de identificadores de capas decodificadas objetivo targetDLayerIdList asociada. En algún caso, la targetDLayerIdList puede ser la misma que la lista de identificadores de capas TargetDecLayerIdList de JCTVC-L1008, que especifica la lista de valores de nuh_layer_id, en orden creciente de los valores nuh_layer_id, de las unidades de NAL a ser decodificadas:

El conjunto de capas de salida seleccionado oLsldx está entre 0 y num_output_layer_sets si no hay mensajes de SEI de cambio de conjuntos de capas de salida.

10 El conjunto de capas de salida seleccionado oLsldx está entre 0 y num_output_layer_sets - num_deleted_output_layer_sets + num_addl_output_layer_sets si hay presentes mensajes de SEI de cambio de conjuntos de capas de salida.

15 El mensaje de SEI de cambio de conjuntos de capas de salida se define de manera que su efecto no es acumulativo. En otra realización, esta sintaxis y/o semántica del mensaje se pueden definir de manera que sean acumulativas. En este caso, el decodificador (y el codificador) pueden hacer el seguimiento de todos los valores anteriores de num_output_layer_sets (desde la extensión de vps) y/o num_deleted_output_layer_sets y/o num_changed_output_layer_sets y/o num_addl_output_layer_sets y pueden acumular estos cambios cada vez que se señala un nuevo mensaje de SEI.

En otra realización, uno o más de los elementos de sintaxis se pueden señalar usando un número fijo conocido de bits en lugar de u(v) en lugar de ue(v). Por ejemplo, se podrían señalar usando u(8) o u(16) o u(32) o u(64), etc.

20 En otra realización, uno o más de estos elementos de sintaxis se podrían señalar con ue(v) o algún otro esquema de codificación en lugar de un número fijo de bits tal como codificación u(v).

En otra realización, los nombres de diversos elementos de sintaxis y su semántica se pueden alterar añadiendo un más 1 o más 2 o restando un menos 1 o un menos 2 en comparación con la sintaxis y la semántica descritas.

25 En otra realización más, diversos elementos de sintaxis incluidos en el mensaje de SEI de conjuntos de capas de salida se pueden señalar por imagen o en otra frecuencia en cualquier lugar en el flujo de bits. Por ejemplo, se pueden señalar en la cabecera de segmento de porción, pps/sps/vps/conjunto de parámetros de adaptación o cualquier otro conjunto de parámetros u otra parte normativa del flujo de bits.

30 En otra realización más, diversos elementos de sintaxis se pueden señalar por imagen o en otra frecuencia en cualquier parte del flujo de bits. Por ejemplo, se pueden señalar en la cabecera de segmento de porción, pps/sps/vps/conjunto de parámetros de adaptación o cualquier otro conjunto de parámetros u otra parte normativa del flujo de bits.

En otra realización más, todos los conceptos definidos en esta invención relacionados con conjuntos de capas de salida se podrían aplicar a puntos de operación de salida como se define en JCTVC-L0452 y JCTVC-L0453 y/o a puntos de operación como se define en JCTVC-L1003.

35 Ejemplo 2

Se describe a continuación otro ejemplo de la presente invención. Obsérvese que, por comodidad, a los miembros que tienen funciones idénticas a las de los miembros respectivos ilustrados en el Primer Ejemplo se les dan números de referencia idénticos respectivos, y se omite aquí una descripción de esos miembros.

40 Se determinó que la señalización del almacenador temporal de imágenes decodificadas (DPB) de orden de salida basada en el uso de los elementos de sintaxis sps_max_num_reorder_pics[HighestTid], sps_max_latency_increase_plus1[HighestTid] y sps_max_dec_pic_buffering[HighestTid] para la salida y la eliminación de imágenes del DPB no considera las características del almacenador temporal que pueden resultar de la codificación de vídeo escalable, tal como cuando se usan diferentes números de capas de mejora que tienden a variar después de que el contenido se haya codificado en base a las preferencias de visualización del usuario, y las capas de mejora de vista múltiple que tienden a variar después de que el contenido se haya codificado en base a las preferencias de visualización del usuario. También se determinó que la señalización del almacenador temporal de imágenes decodificadas (DPB) de orden de salida basada en el uso de los elementos de sintaxis sps_max_num_reorder_pics[HighestTid], sps_max_latency_increase_plus1[HighestTid] y sps_max_dec_pic_buffering[HighestTid] para la salida y la eliminación de imágenes del DPB puede no ser óptima en términos del uso de memoria del DPB cuando el decodificador opera en un cierto punto de operación y/o está emitiendo el conjunto de capas de salida seleccionado. Para adaptar tales diferencias en las preferencias de visualización, el almacenador temporal de imágenes decodificadas (DPB) de orden de salida se puede basar además y/o alternativamente en que tales elementos de sintaxis sean incluidos juntos con la extensión de conjunto de parámetros de vídeo (extensión de VPS) para proporcionar elementos de sintaxis para una o más de las capas de mejora. De esta manera, los elementos de sintaxis se pueden seleccionar para ser especialmente adecuados

para el punto de operación o el conjunto de capas de salida particular, que tiende a corresponder con las preferencias de visualización del usuario.

5 Los parámetros relacionados con el almacenamiento temporal de DPB, `vps_max_dec_pic_buffering_minus1`, `vps_max_num_reorder_pics`, `vps_max_latency_increase_plus1` se pueden señalar para subcapas para la CVS para uno o más puntos de operación y/o para conjuntos de capas de salida en la extensión de VPS. De manera similar, el sistema puede definir la operación y el proceso de sacudidas para que el DPB de orden de salida use los parámetros de almacenamiento temporal de DPB señalados anteriormente de la extensión de VPS si se señalan para el punto de operación bajo prueba o para el conjunto de capas de salida seleccionado. De otro modo, se usan los parámetros de nivel de SPS correspondientes del SPS activo (cuando `currLayerId` que corresponde a `nuh_layer_id` de la imagen actual es igual a 0) o del SPS de capas activo que depende de `layer_id` de la capa actual. En algunos casos, los parámetros `vps_max_dec_pic_buffering_minus1`, `vps_max_num_reorder_pics`, `vps_max_latency_increase_plus1` se puede denominar como `max_vps_dec_pic_buffering_minus1`, `max_vps_num_reorder_pics`, `max_vps_latency_increase_plus1`.

15 En otro caso, el parámetro relacionado con el almacenamiento temporal de DPB `max_vps_dec_pic_buffering_minus1` se señala para las subcapas para la CVS para uno o más conjuntos de capas de salida en la extensión de VPS. Los conjuntos de capas de salida pueden corresponder con puntos de operación. Los parámetros relacionados con almacenamiento temporal de DPB `max_vps_num_reorder_pics`, `max_vps_latency_increase_plus1` se pueden señalar para subcapas para la CVS para cada capa en la extensión de VPS. De manera similar, el sistema puede definir la operación y el proceso de sacudidas para que el DPB de orden de salida use los parámetros de almacenamiento temporal de DPB señalados anteriormente de la extensión de VPS si se señalan para el punto de operación bajo prueba o para el conjunto de capas de salida seleccionado. De otro modo, se usan los parámetros de nivel de SPS correspondientes del SPS activo (cuando `currLayerId` que corresponde a `nuh_layer_id` de la imagen actual es igual a 0) o del SPS de capas activo dependiendo del `layer_id` de la capa actual.

25 Con referencia a la Figura 38A, se ilustra una `vps_extension` modificada ejemplar. La extensión de `vps` modificada incluye una nueva sintaxis, esto es, `num_op_dpb_info_parameters` y `operation_point_layer_set_idx[i]`. Esta extensión de `vps` modificada se puede definir en términos del punto de operación que es un flujo de bits creado a partir de otro flujo de bits mediante la operación de un proceso de extracción de subflujo de bits con el otro flujo de bits, un `TemporalId` más alto objetivo y una lista de identificadores de capa objetivo como entradas.

30 `num_output_layer_sets` especifica el número de conjuntos de capas para los cuales las capas de salida se especifican con `output_layer_set_idx[i]` y `output_layer_flag[lIdx][j]`. Cuando no está presente, el valor de `num_output_layer_sets` se infiere que es igual a 0. Un conjunto de capas que describe las capas de salida es un conjunto de capas de salida.

35 `output_layer_set_idx[i]` especifica el índice `lIdx` del conjunto de capas para el que está presente `output_layer_flag[lIdx][j]`.

`output_layer_flag[lIdx][j]` igual a 1 especifica que la capa con `nuh_layer_id` igual a `j` es una capa de salida objetivo del conjunto de capas de orden `lIdx`. Un valor de `output_layer_flag[lIdx][j]` igual a 0 especifica que la capa con `nuh_layer_id` igual a `j` no es la capa de salida objetivo del conjunto de capas de orden `lIdx`.

40 El `num_op_dpb_info_parameters` especifica el número de estructuras de sintaxis de `op_dpb_parameters()` presentes en la RBSP de extensión de VPS, definidas en términos del punto de operación. Los decodificadores de `num_op_dpb_info_parameters` están en el intervalo de 0 a `vps_num_layer_sets_minus1`, inclusive.

45 El `operation_point_layer_set_idx[i]` especifica el índice, dentro de la lista de conjuntos de capas definidos por los puntos de operación a los que se aplica la estructura de sintaxis de orden `i` `op_dpb_info_parameters()` en la extensión de VPS. El valor de `operation_point_layer_set_idx[i]` puede estar en el intervalo de 0 a `vps_num_layer_sets_minus1`, inclusive. Para la conformidad del flujo de bits, el `operation_point_layer_set_idx[i]` no es igual a `operation_point_layer_set_idx [j]` para cualquier `j` no igual a `i`.

Con referencia a la Figura 39A, el `op_dpb_info_parameters` especifica `vps_max_sub_layers_minus1[j]`, `vps_sub_layer_ordering_info_present_flag[j]`, `vps_max_dec_pic_buffering_minus1[j][k]`, `vps_max_num_reorder_pics[j][k]`, y `vps_max_latency_increase_plus1[j][k]`.

50 El `vps_max_sub_layers_minus1[j]` más 1 indica cuántas subcapas están incluidas. El `vps_max_sub_layers_minus1[j]` más 1 especifica el número máximo de subcapas temporales que pueden estar presentes en la CVS para la capa con `nuh_layer_id` igual a `j`. El valor de `vps_max_sub_layers_minus1[j]` está en el intervalo de 0 a 6, inclusive.

55 El `vps_sub_layer_ordering_info_present_flag[j]` indica si la sintaxis es para un conjunto que incluye todas las capas o para cada capa individual. El `vps_sub_layer_ordering_info_present_flag[j]` igual a 1 especifica que `vps_max_dec_pic_buffering_minus1[j][k]`, `vps_max_num_reorder_pics[j][k]`, y `vps_max_latency_increase_plus1[j][k]` están presentes para la capa con `nuh_layer_id` igual a `j` para las subcapas `vps_max_sub_layers_minus1[j] + 1`. El `vps_sub_layer_ordering_info_present_flag[j]` igual a 0 especifica que los valores de

$vps_max_dec_pic_buffering_minus1[j][vps_max_sub_layers_minus1[j]]$,
 $vps_max_num_reorder_pics[j][vps_max_sub_layers_minus1[j]]$, y
 $vps_max_latency_increase_plus1[j][vps_max_sub_layers_minus1[j]]$ se aplican para todas las subcapas para una capa con nuh_layer_id igual a j .

5 El $vps_max_dec_pic_buffering_minus1[j][k]$ más 1 especifica el máximo tamaño requerido del almacenador temporal de imágenes decodificadas para la CVS para la capa con nuh_layer_id igual a j en unidades de almacenadores temporales de almacenamiento de imágenes cuando $HighestTid$ es igual a k . El valor de $vps_max_dec_pic_buffering_minus1[j][k]$ estará en el intervalo de 0 a $MaxDpbSize - 1$ (como se especifica en la subcláusula A.4), inclusive. Cuando k es mayor que 0, $vps_max_dec_pic_buffering_minus1[j][k]$ será mayor o igual que $vps_max_dec_pic_buffering_minus1[j][k - 1]$. Cuando $vps_max_dec_pic_buffering_minus1[j][k]$ no está presente para k en el intervalo de 0 a $vps_max_sub_layers_minus1[j] - 1$, inclusive, debido a que $vps_sub_layer_ordering_info_present_flag[j]$ es igual a 0, se infiere que es igual a $vps_max_dec_pic_buffering_minus1[j][vps_max_sub_layers_minus1[j]]$.

15 El $vps_max_num_reorder_pics[j][k]$ indica el número máximo permitido de imágenes que pueden preceder a cualquier imagen en la CVS para una capa con nuh_layer_id igual a j en orden de decodificación y seguir a esa imagen en orden de salida cuando $HighestTid$ es igual a k . El valor de $vps_max_num_reorder_pics[j][k]$ estará en el intervalo de 0 a $vps_max_dec_pic_buffering_minus1[j][k]$, inclusive. Cuando k es mayor que 0, $vps_max_num_reorder_pics[j][k]$ es mayor o igual que $vps_max_num_reorder_pics[j][k-1]$. Cuando $vps_max_num_reorder_pics[j][k]$ no está presente para k en el intervalo de 0 a $vps_max_sub_layers_minus1[j] - 1$, inclusive, debido a que $vps_sub_layer_ordering_info_present_flag[j]$ es igual a 0, se infiere que es igual a $vps_max_num_reorder_pics[j][vps_max_sub_layers_minus1[j]]$.

20 El $vps_max_latency_increase_plus1[j][k]$ no igual a 0 se usa para calcular el valor de $VpsMaxLatencyPictures[j][k]$, que especifica el número máximo de imágenes que pueden preceder a cualquier imagen en la CVS para la capa con nuh_layer_id igual a j en el orden de salida y seguir a esa imagen en orden de decodificación cuando $HighestTid$ es igual a k .

25 Cuando $vps_max_latency_increase_plus1[j][k]$ no es igual a 0, el valor de $VpsMaxLatencyPictures[j][k]$ se puede especificar de la siguiente manera:

$$VpsMaxLatencyPictures[j][k] = vps_max_num_reorder_pics[j][k] + vps_max_latency_increase_plus1[j][k] - 1$$

Cuando $vps_max_latency_increase_plus1[j][k]$ es igual a 0, no se expresa ningún límite correspondiente.

30 El valor de $vps_max_latency_increase_plus1[j][k]$ está en el intervalo de 0 a $2^{32} - 2$, inclusive. Cuando $vps_max_latency_increase_plus1[j][k]$ no está presente para k en el intervalo de 0 a $vps_max_sub_layers_minus1[j] - 1$, inclusive, debido a que $vps_sub_layer_ordering_info_present_flag[j]$ es igual a 0, se infiere que es igual a $vps_max_latency_increase_plus1[j][vps_max_sub_layers_minus1[j]]$.

35 El ' $vps_max_sub_layers_minus1[id][j]$ ' más 1 especifica el número máximo de subcapas temporales que pueden estar presentes en la CVS para una capa con nuh_layer_id igual a j para el punto de operación asociado con el id del índice. El valor de $vps_max_sub_layers_minus1[id][j]$ estará en el intervalo de 0 a 6, inclusive.

40 El ' $vps_sub_layer_ordering_info_present_flag[id][j]$ ' igual a 1 especifica que $vps_max_dec_pic_buffering_minus1[id][j][k]$, $vps_max_num_reorder_pics[id][j][k]$, y $vps_max_latency_increase_plus1[id][j][k]$ están presentes para una capa con nuh_layer_id igual a j para el punto de operación asociado con el índice id para $vps_max_sub_layers_minus1[id][j] + 1$ subcapas.

45 $vps_sub_layer_ordering_info_present_flag[id][j]$ igual a 0 especifica que los valores de $vps_max_dec_pic_buffering_minus1[id][j]$, $vps_max_sub_layers_minus1[id][j]$, $vps_max_num_reorder_pics[id][j]$, $vps_max_sub_layers_minus1[id][j]$, y $vps_max_latency_increase_plus1[id][j]$ se aplican a todas las subcapas para la capa con nuh_layer_id igual a j para el punto de operación asociado con el id de índice.

50 El ' $vps_max_dec_pic_buffering_minus1[id][j][k]$ ' más 1 especifica el tamaño máximo requerido del almacenador temporal de imágenes decodificadas para la CVS para una capa con nuh_layer_id igual a j para el punto de operación asociado con el id de índice en unidades de almacenadores temporales de almacenamiento de imágenes cuando $HighestTid$ es igual a k . El valor de $vps_max_dec_pic_buffering_minus1[id][j][k]$ estará en el intervalo de 0 a $MaxDpbSize - 1$ (como se especifica en la subcláusula A.4), inclusive. Cuando k es mayor que 0, $vps_max_dec_pic_buffering_minus1[id][j][k]$ será mayor o igual que $vps_max_dec_pic_buffering_minus1[id][j][k - 1]$. Cuando $vps_max_dec_pic_buffering_minus1[id][j][k]$ no está presente para k en el intervalo de 0 a $vps_max_sub_layers_minus1[id][j] - 1$, inclusive, debido a que $vps_sub_layer_ordering_info_present_flag[id][j]$ es igual a 0, se infiere que es igual a $vps_max_dec_pic_buffering_minus1[id][j][vps_max_sub_layers_minus1[id][j]]$.

55 El ' $vps_max_num_reorder_pics[id][j][k]$ ' indica el número máximo permitido de imágenes que pueden preceder a cualquier imagen en la CVS para una capa con nuh_layer_id igual a j para el punto de operación asociado con el id

de índice en orden de decodificación y seguir a esa imagen en orden de salida cuando HighestTid es igual a k. El valor de $vps_max_num_reorder_pics[id][j][k]$ estará en el intervalo de 0 a $vps_max_dec_pic_buffering_minus1[id][j][k]$, inclusive. Cuando k es mayor que 0, $vps_max_num_reorder_pics[id][j][k]$ será mayor o igual que $vps_max_num_reorder_pics[id][j][k-1]$. Cuando $vps_max_num_reorder_pics[id][j][k]$ no está presente para k en el intervalo de 0 a $vps_max_sub_layers_minus1[id][j] - 1$, inclusive, debido a que $vps_sub_layer_ordering_info_present_flag[id][j]$ es igual a 0, se infiere que sea igual a $vps_max_num_reorder_pics[id][j][vps_max_sub_layers_minus1[id][j]]$.

El ' $vps_max_latency_increase_plus1[id][j][k]$ ' no igual a 0 se usa para calcular el valor de $VpsMaxLatencyPictures[id][j][k]$, que especifica el número máximo de imágenes que pueden preceder a cualquier imagen en la CVS para la capa con nuh_layer_id igual a j para el punto de operación asociado con el id de índice en orden de salida y seguir a esa imagen en orden de decodificación cuando HighestTid es igual a k.

Cuando $vps_max_latency_increase_plus1[id][j][k]$ no es igual a 0, el valor de $VpsMaxLatencyPictures[id][j][k]$ se especifica de la siguiente manera:

$$VpsMaxLatencyPictures[id][j][k] = vps_max_num_reorder_pics[id][j][k] + vps_max_latency_increase_plus1[id][j][k] - 1$$

Cuando $vps_max_latency_increase_plus1[id][j][k]$ es igual a 0, no se expresa ningún límite correspondiente.

El valor de $vps_max_latency_increase_plus1[id][j][k]$ estará en el intervalo de 0 a $2^{32} - 2$, inclusive. Cuando $vps_max_latency_increase_plus1[id][j][k]$ no está presente para k en el intervalo de 0 a $vps_max_sub_layers_minus1[id][j] - 1$, inclusive, debido a que $vps_sub_layer_ordering_info_present_flag[id][j]$ es igual a 0, se infiere que es igual a $vps_max_latency_increase_plus1[id][j][vps_max_sub_layers_minus1[id][j]]$.

Con referencia a la Figura 39 B, los $op_dpb_info_parameters$ se pueden modificar además como se muestra a $op_dpb_info_parameters(id, j)$. En este caso, la sintaxis de la extensión de VPS puede ser como se ilustra en la Figura 38B. El decodificador de referencia hipotético (HRD) se usa para comprobar el flujo de bits y la conformidad del decodificador. Dos tipos de flujos de bits o subconjuntos de flujos de bits se someten a la comprobación de conformidad de HRD para el Equipo Colaborativo Conjunto en codificación de vídeo (JCT-VC). El primer tipo, llamado flujo de bits de Tipo I, es un flujo de unidades de NAL que contiene solamente las unidades de NAL de VCL y las unidades de NAL con nal_unit_type igual a FD_NUT (unidades de NAL de datos de relleno) para todas las unidades de acceso en el flujo de bits. El segundo tipo, denominado flujo de bits de Tipo II, contiene, además de las unidades de NAL de VCL y las unidades de NAL de datos de relleno para todas las unidades de acceso en el flujo de bits, al menos una de (a) unidades de NAL no VCL adicionales distintas de las unidades de NAL de datos de relleno, y (b) todos los elementos de sintaxis $leading_zero_8bits$, $zero_byte$, $start_code_prefix_one_3bytes$, y $trailing_zero_8bits$ que forman un flujo de bytes a partir del flujo de unidades de NAL.

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

Se usan dos tipos de conjuntos de parámetros de HRD (parámetros de HRD de NAL y parámetros de HRD de VCL). Los conjuntos de parámetros de HRD se señalan a través de la estructura de sintaxis $hrd_parameters()$, que puede ser parte de la estructura de sintaxis de SPS o de la estructura de sintaxis de VPS.

Pueden ser necesarias múltiples pruebas para comprobar la conformidad de un flujo de bits, al que se hace referencia como el flujo de bits bajo prueba. Para cada prueba, los siguientes pasos se aplican en el orden enumerado:

(1) Se selecciona un punto de operación bajo prueba, denotado como TargetOp. La lista de identificadores de capas $OpLayerIdList$ de TargetOp consiste en la lista de valores nuh_layer_id , en orden creciente de los valores nuh_layer_id , presentes en el subconjunto del flujo de bits asociado con TargetOp, que es un subconjunto de los valores nuh_layer_id presentes en el flujo de bits bajo prueba. El OpTid de TargetOp es igual al más alto TemporalId presente en el subconjunto de flujo de bits asociado con TargetOp.

(2) TargetDeclayerIdList se establece igual a OpLayerIdList de TargetOp, HighestTid se establece igual a OpTid de TargetOp, y el proceso de extracción de flujo de bits secundario como se especifica en la cláusula 10 se invoca con el flujo de bits bajo prueba, HighestTid, y TargetDeclayerIdList como entradas, y la salida se asigna a BitstreamToDecode.

(3) Se seleccionan la estructura de sintaxis $hrd_parameters()$ y la estructura de sintaxis $sub_layer_hrd_parameters()$ aplicables a TargetOp. Si TargetDeclayerIdList contiene todos los valores nuh_layer_id presentes en el flujo de bits bajo prueba, se selecciona la estructura de sintaxis $hrd_parameters()$ en el SPS activo (o se proporciona a través de un medio externo no especificado en esta Especificación). De otro modo, se selecciona la estructura de sintaxis $hrd_parameters()$ en el VPS activo (o se proporciona a través de algún medio externo no especificados en esta Especificación) que se aplica a TargetOp. Dentro de la estructura de sintaxis $hrd_parameters()$ seleccionada, si BitstreamToDecode es un flujo de bits de Tipo I, se selecciona la estructura de sintaxis $sub_layer_hrd_parameters(HighestTid)$ que sigue inmediatamente a la condición "if

(vcl_hrd_parameters_present_flag)” y la variable NalHrdModeFlag se establece igual a 0; de otro modo (BitstreamToDecode es un flujo de bits de Tipo II), se selecciona la estructura de sintaxis sub_layer_hrd_parameters(HighestTid) que sigue inmediatamente o bien a la condición “if (vcl_hrd_parameters_present_flag)” (en este caso, la variable NalHrdModeFlag se establece igual a 0) o a la condición “if (“nal_hrd_parameters_present_flag)” (en este caso, la variable NalHrdModeFlag se establece igual a 1). Cuando BitstreamToDecode es un flujo de bits de Tipo II y NalHrdModeFlag es igual a 0, todas las unidades de NAL no VCL excepto las unidades de NAL de datos de relleno, y todos los elementos de sintaxis leading_zero_8bits_zero_byte, start_code_prefix_one_3bytes, y trailing_zero_8bits que forman el flujo de bytes a partir del flujo de las unidades de NAL (como se especifica en el Anexo B), cuando están presentes, se descartan de BitstreamToDecode, y el flujo de bits restante se asigna a BitstreamToDecode.

En otro caso, pueden ser necesarias múltiples pruebas para comprobar la conformidad de un flujo de bits, al que se hace referencia como el flujo de bits bajo prueba. Para cada prueba, los siguientes pasos se aplican en el orden enumerado:

(1) Se selecciona un conjunto de capas de salida bajo prueba, denotado como TargetOpLs. El punto de operación referido en TargetOpLs por output_layer_set_idx [] identifica el punto de operación bajo prueba. La lista de identificadores de capas de salida OpLayerIdList de TargetOpLs consiste en la lista de valores nuh_layer_id, en orden creciente de los valores nuh_layer_id, presente en el subconjunto del flujo de bits asociado con TargetOp y TargetOpLs, que es un subconjunto de los valores nuh_layer_id presentes en el flujo de bits bajo prueba. El OpTid de TargetOp es igual al TemporalId más alto presente en el subconjunto del flujo de bits asociado con TargetOp.

(2) TargetDeclayerIdList se establece igual a la lista de identificadores de capas decodificadas objetivo targetDLayerIdList para el conjunto de capas de salida seleccionado TargetOpLs, HighestTid se establece igual a OpTid de TargetOp, y el proceso de extracción del flujo de bits secundario como se especifica en la cláusula 10 se invoca con el flujo de bits bajo prueba, HighestTid, y TargetDeclayerIdList como entradas, y la salida se asigna a BitstreamToDecode.

(3) Se seleccionan la estructura de sintaxis hrd_parameters() y la estructura de sintaxis sub_layer_hrd_parameters() aplicables a TargetOp. Si TargetDeclayerIdList contiene todos los valores nuh_layer_id presentes en el flujo de bits bajo prueba, se selecciona la estructura de sintaxis hrd_parameters() en el SPS activo (o se proporciona a través de un medio externo no especificado en esta Especificación). De otro modo, se selecciona la estructura de sintaxis hrd_parameters() en el VPS activo (o se proporciona a través de algún medio externo no especificados en esta Especificación) que se aplica a TargetOp. Dentro de la estructura de sintaxis hrd_parameters() seleccionada, si BitstreamToDecode es un flujo de bits de Tipo I, se selecciona la estructura de sintaxis sub_layer_hrd_parameters(HighestTid) que sigue inmediatamente a la condición “if (vcl_hrd_parameters_present_flag)” y la variable NalHrdModeFlag se establece igual a 0; de otro modo (BitstreamToDecode es un flujo de bits de Tipo II), se selecciona la estructura de sintaxis sub_layer_hrd_parameters(HighestTid) que sigue inmediatamente o bien a la condición “if(vcl_hrd_parameters_present_flag)” (en este caso, la variable NalHrdModeFlag es igual a 0) o bien a la condición “if (nal_hrd_parameters_present_flag)” (en este caso, la variable NalHrdModeFlag se establece igual a 1). Cuando BitstreamToDecode es un flujo de bits de Tipo II y NalHrdModeFlag es igual a 0, todas las unidades de NAL de no VCL, excepto las unidades de NAL de datos de relleno, y todos los elementos de sintaxis leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes, y trailing_zero_8bits que forman un flujo de bytes a partir del flujo de unidades de NAL (como se especifica en el Anexo B), cuando están presentes, se descartan de BitstreamToDecode, y el flujo de bits restante se asigna a BitstreamToDecode.

Un decodificador conforme puede cumplir todos los requisitos especificados en esta subcláusula.

(1) Un decodificador que reivindica conformidad con un perfil, fila y nivel específicos será capaz de decodificar con éxito todos los flujos de bits que cumplan los requisitos de conformidad del flujo de bits especificados en la subcláusula C.4, de la manera especificada en el Anexo A, a condición de que todos los VPS, SPS y PPS a los que se hace referencia en las unidades de NAL de VCL, y el período de almacenamiento temporal y los mensajes de SEI de temporización de imagen apropiados se transmitan al decodificador, de una manera oportuna, o bien en el flujo de bits (por unidades de NAL de no VCL), o bien por medios externos no especificados en esta Especificación.

(2) Cuando un flujo de bits contiene elementos de sintaxis que tienen valores que se especifican como reservados y se especifica que los decodificadores ignorarán los valores de los elementos de sintaxis o las unidades de NAL que contienen los elementos de sintaxis que tienen los valores reservados, y el flujo de bits de otro modo es conforme con esta Especificación, un decodificador conforme decodificará el flujo de bits de la misma manera que decodificaría un flujo de bits conforme e ignorará los elementos de sintaxis o las unidades de NAL que contienen los elementos de sintaxis que tienen los valores reservados como se especifica.

Hay dos tipos de conformidad de un decodificador: conformidad de temporización de salida y conformidad de orden de salida.

Para comprobar la conformidad de un decodificador, los flujos de bits de prueba que se ajustan al perfil, fila y nivel

reivindicados, como se especifica en la subcláusula C.4, se entregan por un programador de flujo hipotético (HSS) tanto al HRD como al decodificador bajo prueba (DUT). Todas las imágenes decodificadas recortadas emitidas por el HRD también se emitirán por el DUT, cada imagen decodificada recortada emitida por el DUT será una imagen con PicOutputFlag igual a 1, y, para cada imagen tal decodificada recortada emitida por el DUT, los valores de todas las muestras que se emiten serán iguales a los valores de las muestras producidas por el proceso de decodificación especificado.

Para conformidad de decodificador de temporización de salida, el HSS opera como se ha descrito anteriormente, con programaciones de entrega seleccionadas solamente del subconjunto de valores de SchedSelIdx para los cuales la tasa de bits y el tamaño de CPB están restringidos como se especifica en el Anexo A para el perfil, fila y nivel especificados, o con programaciones de entrega "interpoladas" como se especifica a continuación para las cuales la tasa de bits y el tamaño de CPB están restringidos como se especifica en el Anexo A. La misma programación de entrega se usa tanto para el HRD como para el DUT.

Cuando los parámetros de HRD y los mensajes de SEI de período de almacenamiento temporal están presentes con cpb_cnt_minus1[HighestTid] mayor que 0, el decodificador será capaz de decodificar el flujo de bits como se entrega desde el HSS que opera usando una programación de entrega "interpolada" especificada como que tiene una tasa pico de bits r , un tamaño de CPB $c(r)$, y un retardo de eliminación de CPB inicial.

[Fórmula matemática 15]

$$(f(r) \div r)$$

de la manera siguiente:

[Tabla 17]

$\alpha = (r - \text{BitRate}[\text{SchedSelIdx} - 1]) \div (\text{BitRate}[\text{SchedSelIdx}] - \text{BitRate}[\text{SchedSelIdx} - 1]),$	(C-22)
$c(r) = \alpha * \text{CpbSize}[\text{SchedSelIdx}] + (1 - \alpha) * \text{CpbSize}[\text{SchedSelIdx} - 1],$	(C-23)
$f(r) = \alpha * \text{InitCpbRemovalDelay}[\text{SchedSelIdx}] * \text{BitRate}[\text{SchedSelIdx}] + (1 - \alpha) * \text{InitCpbRemovalDelay}[\text{SchedSelIdx} - 1] * \text{BitRate}[\text{SchedSelIdx} - 1]$	(C-24)

para cualquier SchedSelIdx > 0 y r de manera que $\text{BitRate}[\text{SchedSelIdx} - 1] \leq r \leq \text{BitRate}[\text{SchedSelIdx}]$ de manera que r y $c(r)$ estén dentro de los límites que se especifican en el Anexo A para la tasa de bits máxima y el tamaño del almacenador temporal para el perfil, fila y nivel especificados. El InitCpbRemovalDelay[SchedSelIdx] puede ser diferente de un período de almacenamiento temporal a otro y tener que ser vuelto a calcular.

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

Para conformidad de decodificador de orden de salida, se aplica lo siguiente:

(1) El HSS entrega el flujo de bits BitstreamToDecode al DUT "por demanda" del DUT, lo que significa que el HSS entrega bits (en orden de decodificación) solamente cuando el DUT requiere más bits para proceder con su procesamiento. Esto significa que para esta prueba, el almacenador temporal de imágenes codificadas del DUT podría ser tan pequeño como el tamaño de la unidad de decodificación más grande.

(2) Se usa un HRD modificado como se describe a continuación, y el HSS entrega el flujo de bits al HRD mediante una de las programaciones especificadas en el flujo de bits BitstreamToDecode, de manera que la tasa de bits y el tamaño de CPB están restringidos como se especifica en el Anexo A. El orden de salida de imágenes será el mismo tanto para el HRD como para el DUT.

(3) El tamaño de CPB de HRD se da por CpbSize [SchedSelIdx] como se especifica en la subcláusula E.2.3, donde SchedSelIdx y los parámetros de HRD se seleccionan como se especifica en la subcláusula C.1. El tamaño de DPB se da por $\text{sps_max_dec_pic_buffering_minus1}[\text{HighestTid}] + 1$ del SPS activo (cuando nuh_layer_id para la imagen decodificada actual es igual a 0) o del SPS de capa activo para el valor de nuh_layer_id de la imagen decodificada actual. En algunos casos, si los parámetros de información de DPB del punto de operación op_dpb_info_parameters() están presentes para el conjunto de capas de salida seleccionado, el tamaño del DPB se da por $\text{vps_max_dec_pic_buffering_minus1}[\text{HighestTid}]$ cuando currLayerId es igual a 0 o se establece a $\text{vps_max_dec_pic_buffering_minus1}[\text{CurrLayerId}][\text{HighestTid}]$ para el currLayerId para el punto de operación bajo prueba cuando currLayerId es mayor que 0, donde currLayerId es el nuh_layer_id de la imagen decodificada actual. De otro modo, si los parámetros de información de DPB del punto de operación op_dpb_info_parameters() no están presentes para el punto de operación bajo prueba, el Tamaño de DPB se da por

sps_max_dec_pic_buffering_minus1[HighestTid] + 1 del SPS activo (cuando nuh_layer_id para la imagen decodificada actual es igual a 0) o del SPS de capa activo para el valor de nuh_layer_id de la imagen decodificada actual.

5 En algunos casos, si los parámetros de información de DPB de conjuntos de capas de salida oop_dpb_info_parameters() están presentes para el conjunto de capas de salida seleccionado, el tamaño de DPB se da por vps_max_dec_pic_buffering_minus1[HighestTid] cuando currLayerId es igual a 0 o se establece en vps_max_dec_pic_buffering_minus1[CurrLayerId][HighestTid] para currLayerId para el conjunto de capas de salida seleccionado, donde currLayerId es el nuh_layer_id de la imagen decodificada actual. De otro modo, si los parámetros de información de DPB de conjuntos de capas de salida oop_dpb_info_parameters() no están presentes para el conjunto de capas de salida seleccionado, el tamaño de DPB se da por
10 sps_max_dec_pic_buffering_minus1[HighestTid] + 1 del SPS activo (cuando nuh_layer_id para la imagen decodificada actual es igual a 0) o del SPS de capa activo para el valor de nuh_layer_id de la imagen decodificada actual.

15 El tiempo de eliminación del CPB para el HRD es el tiempo de llegada de bits final y la decodificación es inmediata. La operación del DPB de este HRD es como se describe en las subcláusulas C.5.2 a C.5.2.3.

20 El almacenador temporal de imágenes decodificadas contiene almacenadores temporales de almacenamiento de imágenes. El número de almacenadores temporales de almacenamiento de imágenes para nuh_layer_id igual a 0 se deriva del SPS activo. El número de almacenadores temporales de almacenamiento de imágenes para cada valor nuh_layer_id distinto de cero se deriva del SPS de capa activo para ese valor nuh_layer_id distinto de cero. Cada uno de los almacenadores temporales de almacenamiento de imágenes contiene una imagen decodificada que está marcada como “usada para referencia” o que se mantiene para la salida futura. Se invoca el proceso para la salida y eliminación de imágenes del DPB como se especifica en la subcláusula F.13.5.2.2, seguido por la invocación del proceso para la decodificación, marcado, sacudidas adicionales y almacenamiento de imágenes como se especifica en la subcláusula F.13.5.2.3. El proceso de “sacudidas” se especifica en la subcláusula F.13.5.2.4 y se invoca como
25 se especifica en las subcláusulas F.13.5.2.2 y F.13.5.2.3.

La salida y eliminación de imágenes del DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de porción de la primera porción de la imagen actual) ocurre instantáneamente cuando la primera unidad de decodificación de la unidad de acceso que contiene la imagen actual se elimina del CPB y procede de la siguiente manera.

30 Se invoca el proceso de decodificación para el RPS como se especifica en la subcláusula 8.3.2.

(1) Si la imagen actual es una imagen de IRAP con NoRasIOutputFlag igual a 1 y con nuh_layer_id igual a 0 que no es la imagen 0, se aplican los siguientes pasos ordenados:

(A) La variable NoOutputOfPriorPicsFlag se deriva para el decodificador bajo prueba de la siguiente manera:

35 (i) Si la imagen actual es una imagen de CRA, NoOutputOfPriorPicsFlag se establece igual a 1 (independientemente del valor de no_output_of_prior_pics_flag).

40 (ii) De otro modo, si el valor de pic_width_in_luma_samples, pic_height_in_luma_samples, o sps_max_dec_pic_buffering_minus1[HighestTid] derivado del SPS activo es diferente del valor de pic_width_in_luma_samples, pic_height_in_luma_samples, o sps_max_dec_pic_buffering_minus1[HighestTid], respectivamente, derivado del SPS activo para la imagen anterior, NoOutputOfPriorPicsFlag se puede (pero no se debería) establecer en 1 por el decodificador bajo prueba, independientemente del valor de no_output_of_prior_pics_flag. Aunque se prefiere bajo estas condiciones establecer NoOutputOfPriorPicsFlag igual a no_output_of_prior_pics_flag, se permite que el decodificador bajo prueba establezca NoOutputOfPriorPicsFlag en 1 en este caso.

(iii) De otro modo, NoOutputOfPriorPicsFlag se establece igual a no_output_of_prior_pics_flag.

45 (B) El valor de NoOutputOfPriorPicsFlag derivado del decodificador bajo prueba se aplica para el HRD de la siguiente manera:

(i) Si NoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes en el DPB se vacían sin la salida de las imágenes que contienen, y la amplitud de DPB se establece igual a 0.

50 (ii) De otro modo (NoOutputOfPriorPicsFlag es igual a 0), todos los almacenadores temporales de almacenamiento de imágenes que contienen una imagen que está marcada como “no necesaria para la salida” y “no usada para referencia” se vacían (sin salida), y todos los almacenadores temporales de almacenamiento de imágenes no vacíos en el DPB se vacían invocando repetidamente el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4, y la amplitud de DPB se establece igual a 0.

(iii) De otro modo (la imagen actual no es una imagen de IRAP con NoRasIOutputFlag igual a 1 y con nuh_layer_id

igual a 0), todos los almacenadores temporales de almacenamiento de imágenes que contienen una imagen que está marcada como “no necesaria para la salida” y “no usada para referencia” se vacían (sin salida). Por cada almacenador temporal de almacenamiento de imágenes que se vacía, la amplitud de DPB se disminuye en uno. La variable currLayerId se establece igual a nuh_layer_id de la imagen decodificada actual.

5 Las variables MaxNumReorderPics[TargetOp][currLayerId][HighestTid], MaxLatencyIncreasePlus1[TargetOp][CurrLayerId][HighestTid], MaxLatencyPictures[TargetOp][currLayerId][HighestTid], MaxDecPicBufferingMinus1[TargetOp][currLayerId][HighestTid] se derivan de la siguiente manera en base al punto de operación actual bajo prueba:

10 (1) Si los parámetros de información de DPB del punto de operación op_dpb_info_parameters() están presentes para el punto de operación bajo prueba TargetOp, MaxNumReorderPics[TargetOp][currLayerId][HighestTid] se establece en vps_max_num_reorder_pics [HighestTid] cuando currLayerId es igual a 0 o se establece en vps_max_num_reorder_pics[TargetOp][CurrLayerId][HighestTid] para el currLayerId para el punto de operación bajo prueba cuando currLayerId es mayor que 0. De otro modo, si los parámetros de información de DPB del punto de operación op_dpb_info_parameters() no están presentes para el punto de operación bajo prueba MaxNumReorderPics[TargetOp][currLayerId][HighestTid] se establece en sps_max_num_reorder_pics[HighestTid] del SPS activo (cuando currLayerId es igual a 0) o del SPS de capa activo para el valor de currLayerId.

20 (2) Si los parámetros de información de DPB del punto de operación op_dpb_info_parameters() están presentes para el punto de operación bajo prueba TargetOp, MaxLatencyIncreasePlus 1[TargetOp][currLayerId][HighestTid] se establece en vps_max_latency_increase_plus1[HighestTid] cuando currLayerId es igual a 0 o se establece en vps_max_latency_increase_plus1[TargetOp][CurrLayerId][HighestTid] para el currLayerId para el punto de operación bajo prueba cuando currLayerId es mayor que 0. Si los parámetros de información de DPB del punto de operación op_dpb_info_parameters() están presentes para el punto de operación bajo prueba, MaxLatencyPictures[TargetOp][currLayerId][HighestTid] se establece en VpsMaxLatencyPictures[HighestTid] cuando currLayerId es igual a 0 y se establece en VpsMaxLatencyPictures[TargetOp][CurrLayerId][HighestTid] para el currLayerId para el punto de operación bajo prueba cuando currLayerId es mayor que 0. De otro modo, si los parámetros de información de DPB del punto de operación op_dpb_info_parameters() no están presentes para el punto de operación bajo prueba, MaxLatencyIncreasePlus1[TargetOp] [currLayerId][HighestTid] se establece en sps_max_latency_increase_plus1[HighestTid] del SPS activo (cuando currLayerId es igual a 0) o el SPS de capa activo para el valor de currLayerId y MaxLatencyPictures[TargetOp][currLayerId][HighestTid] se establece en SpsMaxLatencyPictures[HighestTid] derivado del SPS activo (cuando currLayerId es igual a 0) o del SPS de capa activo para el valor de currLayerId.

35 (3) Si los parámetros de información de DPB del punto de operación op_dpb_info_parameters() están presentes para el punto de operación bajo prueba seleccionado TargetOp, MaxDecPicBufferingMinus1[TargetOp][currLayerId][HighestTid] se establece en vps_max_dec_pic_buffering_minus1[HighestTid] cuando currLayerId es igual a 0 o se establece en vps_max_dec_pic_buffering_minus1[TargetOp][CurrLayerId][HighestTid] para el currLayerId para el punto de operación bajo prueba cuando currLayerId es mayor que 0. De otro modo, si los parámetros de información de DPB del punto de operación op_dpb_info_parameters() no están presentes para el punto de operación bajo prueba, MaxDecPicBufferingMinus1[TargetOp][currLayerId][HighestTid] se establece en sps_max_dec_pic_buffering_minus1[HighestTid] del SPS activo (cuando currLayerId es igual a 0) o del SPS de capa activo para el valor de currLayerId.

45 Cuando una o más de las siguientes condiciones son ciertas, el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4 se invoca repetidamente mientras se disminuye además la amplitud de DPB en uno para cada almacenador temporal de almacenamiento de imágenes adicional que se vacía, hasta que ninguna de las siguientes condiciones son ciertas:

(1) El número de imágenes con nuh_layer_id igual a currLayerId en el DPB que están marcadas como “necesarias para la salida” es mayor que MaxNumReorderPics[TargetOp][CurrLayerId][HighestTid].

50 (2) Si MaxLatencyIncreasePlus1[TargetOp][CurrLayerId][HighestTid] no es igual a 0 y hay al menos una imagen con nuh_layer_id igual a currLayerId en el DPB que está marcada como “necesaria para la salida” para la cual la variable asociada PicLatencyCount[currLayerId] es mayor o igual que MaxLatencyPictures[TargetOp][CurrLayerId][HighestTid].

(3) El número de imágenes con nuh_layer_id igual a currLayerId en el DPB es mayor o igual que MaxDecPicBuffering[TargetOp][CurrLayerId][HighestTid].

55 El proceso especificado en esta subcláusula ocurre instantáneamente cuando la última unidad de decodificación de la unidad de acceso n que contiene la imagen actual se elimina del CPB.

La variable currLayerId se establece igual a nuh_layer_id de la imagen decodificada actual.

Para cada imagen en el DPB que está marcada como “necesaria para la salida” y que tiene un valor `nuh_layer_id` igual a `currLayerId`, la variable asociada `PicLatencyCount[currLayerId]` se establece igual a `PicLatencyCount[currLayerId] + 1`.

5 La imagen actual se considera como decodificada después de que se decodifica la última unidad de decodificación de la imagen. La imagen decodificada actual se almacena en un almacenador temporal de almacenamiento de imágenes vacío en el DPB, y se aplica lo siguiente:

(A) Si la imagen decodificada actual tiene `PicOutputFlag` igual a 1, se marca como “necesaria para la salida” y su variable asociada `PicLatencyCount[currLayerId]` se establece igual a 0.

10 (B) De otro modo (la imagen decodificada actual tiene `PicOutputFlag` igual a 0), se marca como “no necesaria para la salida”.

La imagen decodificada actual se marca como “usada para referencia a corto plazo”.

Cuando una o más de las siguientes condiciones son ciertas, el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4 se invoca repetidamente hasta que ninguna de las siguientes condiciones es cierta:

15 (A) El número de imágenes con `nuh_layer_id` igual a `currLayerId` en el DPB que están marcadas como “necesarias para la salida” es mayor que `MaxNumReorderPics[TargetOp][CurrLayerId][HighestTid]`.

(B) `MaxLatencyIncreasePlus1[TargetOp][CurrLayerId][HighestTid]` no es igual a 0 y hay al menos una imagen con `nuh_layer_id` igual a `currLayerId` en el DPB que está marcada como “necesaria para la salida” para la cual la variable asociada `PicLatencyCount[currLayerId]` es mayor o igual que `MaxLatencyPictures[TargetOp][CurrLayerId][HighestTid]`.

20 En otro caso, las variables `MaxNumReorderPics[currLayerId][HighestTid]`, `MaxLatencyIncreasePlus1[currLayerId][HighestTid]`, `MaxLatencyPictures[currLayerId][HighestTid]`, `MaxDecPicBufferingMinus1[currLayerId][HighestTid]` se puede derivar de la siguiente manera:

25 (1) Si los parámetros de información de DPB del punto de operación `op_dpb_info_parameters()` están presentes para el punto de operación bajo prueba, `MaxNumReorderPics[currLayerId][HighestTid]` se establece en `vps_max_num_reorder_pics[HighestTid]` cuando `currLayerId` es igual a 0 o se establece en `vps_max_num_reorder_pics[CurrLayerId][HighestTid]` para el `currLayerId` para el punto de operación bajo prueba cuando `currLayerId` es mayor que 0. De otro modo, si los parámetros de información de DPB del punto de operación `op_dpb_info_parameters()` no están presentes para el punto de operación bajo prueba `MaxNumReorderPics[currLayerId][HighestTid]` se establece en `sps_max_num_reorder_pics[HighestTid]` del SPS activo (cuando `currLayerId` es igual a 0) o del SPS de capa activo para el valor de `currLayerId`.

30 (2) Si los parámetros de información de DPB del punto de operación `op_dpb_info_parameters()` están presentes para el punto de operación bajo prueba, `MaxLatencyIncreasePlus1[CurrLayerId][HighestTid]` se establece en `vps_max_latency_increase_plus1[HighestTid]` cuando `currLayerId` es igual a 0 o se establece en `vps_max_latency_increase_plus1[CurrLayerId][HighestTid]` para el `currLayerId` para el punto de operación bajo prueba cuando `currLayerId` es mayor que 0. Si los parámetros de información de DPB del punto de operación `op_dpb_info_parameters()` están presentes para el punto de operación bajo prueba, `MaxLatencyPictures[currLayerId][HighestTid]` se establece en `VpsMaxLatencyPictures[HighestTid]` cuando `currLayerId` es igual a 0 o se establece en `VpsMaxLatencyPictures[CurrLayerId][HighestTid]` para `currLayerId` para el punto de operación bajo prueba cuando `currLayerId` es mayor que 0. De otro modo, si los parámetros de información de DPB del punto de información `op_dpb_info_parameters()` no están presentes para el punto de operación bajo prueba, `MaxLatencyIncreasePlus1[currLayerId][HighestTid]` se establece en `sps_max_latency_increase_plus1[HighestTid]` del SPS activo (cuando `currLayerId` es igual a 0) o el SPS de capa activo para el valor de `currLayerId` y `MaxLatencyPictures[currLayerId][HighestTid]` se establece en `SpsMaxLatencyPictures[HighestTid]` derivado del SPS activo (cuando `currLayerId` es igual a 0) o del SPS de capa activo para el valor de `currLayerId`.

35 (3) Si los parámetros de información de DPB del punto de operación `op_dpb_info_parameters()` están presentes para el punto de operación bajo prueba seleccionado, `MaxDecPicBufferingMinus1[currLayerId][HighestTid]` se establece en `vps_max_dec_pic_buffering_minus1[HighestTid]` cuando `currLayerId` es igual a 0 o se establece en `vps_max_dec_pic_buffering_minus1[CurrLayerId][HighestTid]` para el `currLayerId` para el punto de operación bajo prueba cuando `currLayerId` es mayor que 0. De otro modo, si los parámetros de información de DPB del punto de operación `op_dpb_info_parameters()` no están presentes para el punto de operación bajo prueba, `MaxDecPicBufferingMinus1[currLayerId][HighestTid]` se establece en `sps_max_dec_pic_buffering_minus1[HighestTid]` del SPS activo (cuando `currLayerId` es igual a 0) o del SPS de capa activo para el valor de `currLayerId`.

55 Cuando una o más de las siguientes condiciones son ciertas, el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4 se invoca repetidamente mientras que se disminuye además la amplitud de DPB en uno para

cada almacenador temporal de almacenamiento de imágenes adicional que se vacía, hasta que ninguna de las siguientes condiciones son ciertas:

(1) El número de imágenes con `nuh_layer_id` igual a `currLayerId` en el DPB que están marcadas como “necesarias para la salida” es mayor que `MaxNumReorderPics[CurrLayerId][HighestTid]`.

5 (2) Si `MaxLatencyIncreasePlus1[CurrLayerId][HighestTid]` no es igual a 0 y hay al menos una imagen con `nuh_layer_id` igual a `currLayerId` en el DPB que está marcada como “necesaria para la salida” para la cual la variable asociada `PicLatencyCount[currLayerId]` es mayor o igual que `MaxLatencyPictures[CurrLayerId][HighestTid]`.

(3) El número de imágenes con `nuh_layer_id` igual a `currLayerId` en el DPB es mayor o igual que `MaxDecPicBuffering[CurrLayerId][HighestTid]`.

10 Los procesos especificados en esta subcláusula ocurren instantáneamente cuando la última unidad de decodificación de la unidad de acceso `n` que contiene la imagen actual se elimina del CPB.

La variable `currLayerId` se establece igual a `nuh_layer_id` de la imagen decodificada actual.

15 Para cada imagen en el DPB que está marcada como “necesaria para la salida” y que tiene un valor `nuh_layer_id` igual a `currLayerId`, la variable asociada `PicLatencyCount[currLayerId]` se establece igual a `PicLatencyCount[currLayerId] + 1`.

La imagen actual se considera como decodificada después de que se decodifica la última unidad de decodificación de la imagen. La imagen decodificada actual se almacena en un almacenador temporal de almacenamiento de imágenes vacío en el DPB, y se aplica lo siguiente:

20 (A) Si la imagen decodificada actual tiene `PicOutputFlag` igual a 1, se marca como “necesaria para la salida” y su variable asociada `PicLatencyCount[currLayerId]` se establece igual a 0.

(B) De otro modo (la imagen decodificada actual tiene `PicOutputFlag` igual a 0), se marca como “no es necesaria para la salida”.

La imagen decodificada actual se marca como “usada para referencia a corto plazo”.

25 Cuando una o más de las siguientes condiciones son ciertas, el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4 se invoca repetidamente hasta que ninguno de las siguientes condiciones sea cierto.

(A) El número de imágenes con `nuh_layer_id` igual a `currLayerId` en el DPB que están marcadas como “necesarias para la salida” es mayor que `MaxNumReorderPics[CurrLayerId][HighestTid]`.

30 (B) `MaxLatencyIncreasePlus1[CurrLayerId][HighestTid]` no es igual a 0 y hay al menos una imagen con `nuh_layer_id` igual a `currLayerId` en el DPB que está marcada como “necesaria para la salida” para la cual la variable asociada `PicLatencyCount[currLayerId]` es mayor o igual que `MaxLatencyPictures[CurrLayerId][HighestTid]`.

El proceso de “sacudidas” consiste en los siguientes pasos ordenados:

(A) Las imágenes que son las primeras para la salida se seleccionan como las que tienen el valor más pequeño de `PicOrderCntVal` de todas las imágenes en el DPB marcadas como “necesarias para la salida”.

35 (B) Estas imágenes se recortan, usando la ventana de recorte de conformidad especificada en el SPS activo para la imagen con `nuh_layer_id` igual a 0 o en el SPS de capa activo para un valor de `nuh_layer_id` igual al de la imagen, las imágenes recortadas se emiten en orden ascendente de `nuh_layer_id`, y las imágenes se marcan como “no necesarias para la salida”.

(C) Cada almacenador temporal de almacenamiento de imágenes que contiene una imagen marcada como “no usada para referencia” y que incluyó una de las imágenes que se recortó y se emitió se vacía.

40 La extensión de VPS puede tener modificaciones adicionales, si se desea.

Con referencia a la Figura 40, una modificación adicional puede incluir los parámetros de DPB que se envían en la extensión de VPS para los conjuntos de capas de salida en lugar de para los puntos de operación, donde los `oops_dpb_info_parameters(j)` se ilustran en la Figura 41.

45 El `num_dpb_info_parameters` especifica el número de estructuras de sintaxis `oop_dpb_parameters()` presentes en la RBSP de extensión de VPS. `num_dpb_info_parameters` decodificadores estarán en el intervalo de 0 a `num_output_layer_sets`, inclusive.

El `output_point_layer_set_idx[i]` especifica el índice, en la lista de conjuntos de capas de salida objetivo a los que se aplica la estructura de sintaxis `oop_dpb_info_parameters()` de orden `i` en la extensión de VPS.

El valor de `output_point_layer_set_idx[i]` debería estar en el intervalo de 0 a `num_output_layer_sets`, inclusive. Es requisito de la conformidad del flujo de bits que `output_point_layer_set_idx[i]` no sea igual a `output_point_layer_set_idx [j]` para cualquier `j` no igual a `i`.

5 Con referencia a la Figura 42, los `oop_dpb_info_parameters(c)` se pueden modificar además, donde la sintaxis en la extensión de VPS puede ser como se ilustra en la Figura 43.

Con referencia a la Figura 44, los `oop_dpb_info_parameters(c)` se puede modificar además, donde la sintaxis en la extensión de VPS puede ser como se ilustra en la Figura 45 o en la Figura 46.

Una alternativa ejemplar para la sintaxis en la extensión de VPS es que

[Tabla 18]

```
10 for( j = 0; j <= vps_max_layer_id; j++ )
    oop_dpb_info_parameters(j)
```

se puede cambiar a

[Tabla 19]

```
for( j = 0; j <= vps_max_layers_minus1; j++ )
    oop_dpb_info_parameters(j)
```

15 El `vps_max_layer_id` especifica el valor máximo permitido de `nuh_layer_id` de todas las unidades de NAL en la CVS. El `vps_max_layers_minus1`, especifica el número máximo de capas que pueden estar presentes en la CVS, en donde una capa puede, por ejemplo, ser una capa escalable espacial, una capa escalable de calidad, una vista de textura o una vista de profundidad.

Otra alternativa ejemplar para la sintaxis en la extensión de VPS es que

[Tabla 20]

```
20 for( j = 0; j <= vps_max_layer_id; j++ )
    oop_dpb_info_parameters(j)
```

se puede cambiar a

[Tabla 21]

```
for( j = 0; j < numOutputLayers; j++ )
    oop_dpb_info_parameters(j)
```

donde `numOutputLayers` para el índice de conjunto de capas de salida seleccionado `oplsIdx` se deriva como:

25 [Tabla 22]

```
for(k=0, numOutputLayers=0; k<=vps_max_layer_id; k++)
    if(output_layer_flag[opLsIdx][k])
        targetOpLayerIdList [numOutputLayers++] = layer_id_in_nuh[k]
```

Otra alternativa ejemplar para la sintaxis en la extensión de VPS es que

[Tabla 23]

```
30 for( j = 0; j <= vps_max_layer_id; j++ )
    oop_dpb_info_parameters(j)
```

se puede cambiar a

[Tabla 24]

for(j = 0; j < numDecodedLayers; j++)
oop_dpb_info_parameters(j)

donde numOutputLayers para el opslIdx seleccionado se deriva como:

[Tabla 25]

for(k=0, numOutputLayers=0;k<=vps_max_layer_id;k++)
if(output_layer_flag[opLsIdx][k])
targetOpLayerIdList [numOutputLayers++]=layer_id_in_nuh[k]

- 5 Entonces, una lista de identificadores de capas decodificadas objetivo targetDLayerIdList y numDecodedLayers para el opslIdx seleccionado se deriva como:

[Tabla 26]

for(m=0, numDecodedLayers=0;m< numOutputLayers;m++) {
for(n=0;n<NumDirectRefLayers[LayerIdInVps[targetOpLayerIdList[m]]];n++) {
rLid=RefLayerId[LayerIdInVps[targetOpLayerIdList[m]]][n]
if(rLid not included in targetDLayerIdList[0,..., numDecodedLayers])
targetDLayerIdList[numDecodedLayers++]=rLid;
}
}

- 10 En una realización, una bandera adicional se puede señalar para indicar si oop_dpb_information_parameters están señalados para la capa particular de la siguiente manera:

[Tabla 27]

for(j = 0; j <= vps_max_layer_id; j++) {	
vps_layer_info_present_flag[j]	u(1)
if(vps_layer_info_present_flag)	
oop_dpb_info_parameters(j)	
}	

- 15 La vps_layer_info_present_flag[j] igual a 1 especifica que oop_dpb_info_parameters están presentes para la capa de orden j para el conjunto de capas de salida particular. vps_layer_info_present_flag[j] igual a 0 especifica que oop_dpb_info_parameters no están presentes para la capa de orden j para el conjunto de capas de salida particular.

En otra realización, num_dpb_info_parameters decodificadores estarán en el intervalo de 0 a 1024, inclusive. En otra realización más, se podría usar un número fijo diferente en lugar de 1024.

En una realización alternativa, output_point_layer_set_idx[i] está en el intervalo de 0 a 1023, inclusive.

Con referencia a la Figura 47, se pueden usar otra extensión de VPS modificada y layer_dpb_info (i) si los parámetros de DPB se envían en la extensión de VPS para cada capa independientemente de los conjuntos de capas de salida y los puntos de operación.

5 Con referencia a la Figura 48, se puede usar una layer_dpb_info (i) modificada donde el elemento de sintaxis vps_max_sub_layer_minus1 señalado desde el VPS se usa para todas las capas y no se señala por separado en oop_dpb_info_parameters(id)/op_dpb_info_parameters(id).

10 Con referencia a la Figura 49A y 49B, se ilustra una vps_extension modificada ejemplar. La extensión de vps modificada incluye una nueva sintaxis, esto es, max_sub_layers_vps_predict_flag[i], max_sub_layers_vps_minus1[i] num_dpb_info_parameters, output_point_layer_set_idx[i], oop_dpb_maxbuffering_parameters(i), y layer_dpb_info_parameters(i). num_output_layer_sets especifica el número de conjuntos de capas para los cuales las capas de salida se especifican con output_layer_set_index[i] y output_layer_flag[lsIdx][j]. Cuando no está presente, se infiere que el valor de num_output_layer_sets es igual a 0. Un conjunto de capas que describe las capas de salida es un conjunto de capas de salida.

15 'max_sub_layers_vps_predict_flag'[i] igual a 1 especifica que se infiere que max_sub_layers_vps_minus1[i] es igual a max_sub_layers_vps_minus1[i - 1].

max_sub_layers_vps_predict_flag[i] igual a 0 especifica que max_sub_layers_vps_minus1[i] se señala explícitamente. El valor de max_sub_layers_vps_predict_flag[0] se infiere que es igual a 0.

20 'max_sub_layers_vps_minus1'[i] más 1 especifica el número máximo de subcapas temporales que pueden estar presentes en la CVS para la capa con nuh_layer_id igual a i. El valor de max_vps_sub_layers_vps_minus1[i] estará en el intervalo de 0 a 6, inclusive. En algunos casos, max_sub_layers_vps_minus1[i] se usa para la inferencia del elemento de sintaxis SPS sps_max_sub_layers_minus1. Cuando max_sub_layers_vps_predict_flag[i] es igual a 1, max_sub_layers_vps_minus1[i] se infiere que es igual a max_sub_layers_vps_minus1[i - 1]. El valor de max_sub_layers_vps_minus1[0] se infiere que es igual a vps_max_sub_layers_minus1.

25 La variable MaxSubLayers[setId] para setId en el intervalo de 0 a num_dpb_info_parameters- 1, inclusive, se deriva de la siguiente manera:

[Tabla 28]

for(setId = 0; setId < num_dpb_info_parameters; setId++) {
lsIdx =
output_layer_set_idx_minus1[output_point_layer_set_idx[setId]] + 1
highestLayerId =
layerSetLayerIdList[lsIdx][numLayersInIdList[lsIdx]-1]
MaxSubLayers[setId] =
(max_sub_layers_vps_minus1[highestLayerId]+1)
}

30 'num_dpb_info_parameters' especifica el número de estructuras de sintaxis oop_dpb_maxbuffering_parameters(i) presentes en la RBSP de extensión de VPS. Los decodificadores num_dpb_info_parameters estarán en el intervalo de 0 a numOutputLayerSets, inclusive.

'output_point_layer_set_idx'[i] especifica el índice, en la lista de conjuntos de capas de salida a los que se aplica la estructura de sintaxis de orden i oop_dpb_maxbuffering_parameters(i) en la extensión de VPS.

35 El valor de output_point_layer_set_idx[i] debería estar en el intervalo de 0 a numOutputLayerSets, inclusive. Es requisito de la conformidad del flujo de bits que output_point_layer_set_idx[i] no sea igual a output_point_layer_set_idx [j] para cualquier j no igual a i.

Con referencia a la Figura 50, el oop_dpb_maxbuffering_parameters especifica 'sub_layer_vps_buf_info_present_flag'[i], 'max_vps_dec_pic_buffering_minus1'[i][j].

'sub_layer_vps_buf_info_present_flag'[i] igual a 1 especifica que max_vps_dec_pic_buffering_minus1[i][j] están presentes para las subcapas MaxSubLayers [i]. sub_layer_vps_buf_info_present_flag[i] igual a 0 especifica que los

valores de $\text{max_vps_dec_pic_buffering_minus1}[i]$ [$\text{MaxSubLayers}[i] - 1$] se aplican a todas las subcapas.

5 'max_vps_dec_pic_buffering_minus1'[i][j] más 1 especifica el tamaño máximo requerido del almacenador temporal de imágenes decodificadas para la CVS para la capa con nuh_layer_id igual a highestLayerId en el conjunto de capas de salida asociado con el índice i en unidades de almacenadores temporales de almacenamiento de imágenes cuando HighestTid es igual a j . El valor de $\text{max_vps_dec_pic_buffering_minus1}[i][j]$ estará en el intervalo de 0 a $\text{MaxDpbSize} - 1$ (como se especifica en la subcláusula A.4), inclusive. Cuando j es mayor que 0, $\text{max_vps_dec_pic_buffering_minus1}[i][j]$ será mayor o igual que $\text{max_vps_dec_pic_buffering_minus1}[i][j - 1]$. En algunos casos $\text{max_vps_dec_pic_buffering_minus1}[i][j]$ se usa para inferencia de los valores de los elementos de sintaxis de SPS $\text{sps_max_dec_pic_buffering_minus1}[j]$. Cuando $\text{max_vps_dec_pic_buffering_minus1}[i][j]$ no está presente para i en el intervalo de 0 a $\text{MaxSubLayers}[i] - 2$, inclusive, debido a que $\text{sub_layer_vps_buf_info_present_flag}[i]$ es igual a 0, se infiere que es igual a $\text{max_vps_dec_pic_buffering_minus1}[i]$ [$\text{MaxSubLayers}[i] - 1$].

El valor de $\text{max_vps_dec_pic_buffering_minus1}[0][j]$ para cada valor de j se infiere que es igual a $\text{vps_max_dec_pic_buffering_minus1}[j]$.

15 Con referencia a la Figura 51, el $\text{layer_dpb_info_parameters}$ especifica 'sub_layer_vps_ordering_info_present_flag'[i], $\text{max_vps_num_reorder_pics}$ [i][j], 'max_vps_latency_increase_plus1'[i][j].

'sub_layer_vps_ordering_info_present_flag'[i] igual a 1 especifica que $\text{max_vps_num_reorder_pics}$ [i][j] y $\text{max_vps_latency_increase_plus1}[i][j]$ están presentes para $\text{max_sub_layers_vps_minus1} + 1$ subcapas.

20 $\text{sub_layer_vps_ordering_info_present_flag}[i]$ igual a 0 especifica que los valores de $\text{max_vps_num_reorder_pics}[i]$ [$\text{vps_max_sub_layers_minus1}$] y $\text{max_vps_latency_increase_plus1}[i]$ [$\text{max_sub_layers_vps_minus1}$] se aplican a todas las subcapas.

25 'max_vps_num_reorder_pics'[i][j] indica el número máximo permitido de imágenes que pueden preceder a cualquier imagen en la CVS para la capa con nuh_layer_id igual a i en orden de decodificación y siguen a esa imagen en orden de salida cuando HighestTid es igual a j . El valor de $\text{max_vps_num_reorder_pics}[i][j]$ estará en el intervalo de 0 a $\text{max_vps_dec_pic_buffering_minus1}[i][j]$, inclusive. Cuando j es mayor que 0, $\text{max_vps_num_reorder_pics}[i][j]$ será mayor o igual que $\text{max_vps_num_reorder_pics}[i][j - 1]$. En algunos casos $\text{max_vps_num_reorder_pics}[i][j]$ se usa para inferencia de los valores del elemento de sintaxis de SPS $\text{sps_max_num_reorder_pics}[j]$. Cuando $\text{max_vps_num_reorder_pics}[i][j]$ no está presente para i en el intervalo de 0 a $\text{max_sub_layers_vps_minus1}[i] - 1$, inclusive, debido a que $\text{sub_layer_vps_ordering_info_present_flag}[i]$ es igual a 0, se infiere que es igual a $\text{max_vps_num_reorder_pics}[i]$ [$\text{max_sub_layers_vps_minus1}[i]$].

30 $\text{max_vps_latency_increase_plus1}[i][j]$ no igual a 0 se usa para calcular el valor de $\text{VpsMaxLatencyPictures}$ [i][j], que especifica el número máximo de imágenes que pueden preceder a cualquier imagen en la CVS para la capa con nuh_layer_id igual a i en orden de salida y seguir a esa imagen en orden de decodificación cuando HighestTid es igual a j .

35 Cuando $\text{max_vps_latency_increase_plus1}[i][j]$ no es igual a 0, el valor de $\text{VpsMaxLatencyPictures}[i][j]$ se especifica de la siguiente manera:

$$\text{VpsMaxLatencyPictures}[i][j] = \text{max_vps_num_reorder_pics}[i][j] + \text{max_vps_latency_increase_plus1}[i][j] - 1$$

Cuando $\text{max_vps_latency_increase_plus1}[i][j]$ es igual a 0, no se expresa ningún límite correspondiente.

40 El valor de $\text{max_vps_latency_increase_plus1}[j][k]$ estará en el intervalo de 0 a $2^{32} - 2$, inclusive. En algunos casos, $\text{max_vps_latency_increase_plus1}[i][j]$ se usa para inferencia de los valores de los elementos de sintaxis de SPS $\text{sps_max_latency_increase_plus1}[j]$. Cuando $\text{max_vps_latency_increase_plus1}[i][j]$ no está presente para i en el intervalo de 0 a $\text{max_sub_layers_vps_minus1}[i] - 1$, inclusive, debido a que $\text{sub_layer_vps_ordering_info_present_flag}[i]$ es igual a 0, se infiere que es igual a $\text{max_vps_latency_increase_plus1}[i]$ [$\text{max_sub_layers_vps_minus1}[i]$].

45 Con referencia a la Figura 52, se ilustra una vps_extension modificada ejemplar. La extensión de vps modificada incluye una modificación adicional de la sintaxis en la Figura 49 con nueva sintaxis, esto es, $\text{sub_layer_vps_buf_info_predict_flag}[i]$, $\text{sub_layer_vps_ordering_info_predict_flag}[i]$ que se señalan condicionalmente. $\text{sub_layer_vps_buf_info_predict_flag}[i]$ igual a 1 especifica que $\text{max_vps_dec_pic_buffering_minus1}[i][j]$ se infiere que es igual a $\text{max_vps_dec_pic_buffering_minus1}[i - 1][j]$ para cada valor de j .

50 $\text{sub_layer_vps_buf_info_predict_flag}[i]$ igual a 0 especifica que $\text{max_vps_dec_pic_buffering_minus1}[i][j]$ para al menos un valor de j se señala explícitamente. $\text{sub_layer_vps_buf_info_predict_flag}[0]$ se infiere que es igual a 0. Cuando no está presente, $\text{sub_layer_vps_buf_info_predict_flag}[i]$ se infiere que es igual a 0.

sub_layer_vps_ordering_info_predict_flag[i] igual a 1 especifica que los elementos de sintaxis sub_layer_vps_ordering_info_present_flag[i], max_vps_num_reorder_pics [i][j], y max_vps_latency_increase_plus1 [i][j] se infieren que son iguales a sub_layer_vps_ordering_info_present_flag[i - 1], max_vps_num_reorder_pics [i - 1][j], y max_vps_latency_increase_plus1[i - 1][j], respectivamente.

5 sub_layer_vps_ordering_info_predict_flag[i] igual a 0 indica que los elementos de sintaxis sub_layer_vps_ordering_info_present_flag[i], max_vps_num_reorder_pics[i][j], y max_vps_latency_increase_plus1 [i][j] se señalan explícitamente. Cuando no está presente, el valor de sub_layer_vps_ordering_info_predict_flag[i] se establece igual a 0.

Otros elementos de sintaxis y sus significados semánticos para la Figura 52 son los mismos que los de la Figura 49.

10 Con referencia a la Figura 53, se ilustra un ejemplo de vps_extension modificada. La extensión de vps modificada incluye una modificación adicional de la sintaxis en la Figura 49. En la Figura 53 oop_dpb_maxbuffering_parameters(i, j) se señalan para cada capa j para el conjunto de capas de salida particular i en comparación con la Figura 49 que señala un único conjunto de parámetros oop_dpb_maxbuffering_parameters(i) para el id de capa más alto en el conjunto de capas de salida i.

15 Con referencia a la Figura 54, el oop_dpb_maxbuffering_parameters especifica sub_layer_vps_buf_info_present_flag[i][k], max_vps_dec_pic_buffering_minus1[i][k][j].

sub_layer_vps_buf_info_present_flag[i][k] igual a 1 especifica que max_vps_dec_pic_buffering_minus1[i][k][j] están presentes para las subcapas max_sub_layers_vps_minus1[k]. sub_layer_vps_buf_info_present_flag[i] igual a 0 especifica que los valores de max_vps_dec_pic_buffering_minus1[i][k][max_sub_layers_vps_minus1[k]] se aplican a todas las subcapas.

20 max_vps_dec_pic_buffering_minus1[i][k][j] más 1 especifica el máximo tamaño requerido del almacenador temporal de imágenes decodificadas para la CVS para la capa con nuh_layer_id igual a k para el conjunto de capas de salida asociado con el índice i en unidades de almacenadores temporales de almacenamiento de imágenes cuando HighestTid es igual a j. El valor de max_vps_dec_pic_buffering_minus1[i][k][j] estará en el intervalo de 0 a MaxDpbSize - 1 (como se especifica en la subcláusula A.4), inclusive.

25 Cuando j es mayor que 0, max_vps_dec_pic_buffering_minus1[i][k][j] será mayor o igual que max_vps_dec_pic_buffering_minus1[i][k][j - 1] se usan para inferencia de los valores de elementos de sintaxis de SPS sps_max_dec_pic_buffering_minus1[j].

30 Cuando max_vps_dec_pic_buffering_minus1[i][j] no está presente para i en el intervalo de 0 a max_sub_layers_vps_minus1[k] - 1, inclusive, debido a que sub_layer_vps_buf_info_present_flag[i][k] es igual a 0, se infiere que es igual a max_vps_dec_pic_buffering_minus1[i][k] [max_sub_layers_vps_minus1[k]].

El valor de max_vps_dec_pic_buffering_minus1[i][0][j] para cada valor de i y j se infiere que es igual a vps_max_dec_pic_buffering_minus1[j].

35 Con referencia a la Figura 55, el oop_dpb_maxbuffering_parameters especifica sub_layer_vps_buf_info_present_flag[i][k], max_vps_dec_pic_buffering_minus1[i][k][j]. La Figura 55 es una variante de sintaxis para oop_dpb_maxbuffering_parameters en comparación con la sintaxis en la Figura 54 para oop_dpb_maxbuffering_parameters.

La variable MaxSubLayers[setId][k] para setId en el intervalo de 0 a num_dpb_info_parameters - 1, inclusive, se deriva de la siguiente manera:

40 [Tabla 29]

for(setId = 0; setId < num_dpb_info_parameters; setId++) {
for(k = 1; k <= vps_max_layers_minus1; i++) {
MaxSubLayers[setId][k] =(max_sub_layers_vps_minus1[k] +1)
}
}

En este caso, los parámetros oop_dpb_maxbuffering_parameters(i, k) se definirán como en la Figura 55.

Con referencia a la Figura 56, se ilustra una vps_extension modificada ejemplar. La extensión de vps modificada incluye una modificación adicional de la sintaxis en la Figura 52 con una nueva sintaxis. En la Figura 53 en esta

variante `oop_dpb_maxbuffering_parameters(i, j)` se señala para cada capa `j` para el conjunto de capas de salida particular `i` en comparación con la Figura 52, que señala un único conjunto de parámetros `oop_dpb_maxbuffering_parameters(i)` para el id de capa más alto en el conjunto de capas de salida `i`.

Los `oop_dpb_maxbuffering_parameters(l, k)` son como se muestra en la Figura 57.

5 Con referencia a la Figura 58, el `oop_dpb_maxbuffering_parameters` especifica `sub_layer_vps_buf_info_present_flag[i][k]`, `max_vps_dec_pic_buffering_minus1[i][k][j]`. La Figura 58 es una variante de sintaxis para `oop_dpb_maxbuffering_parameters` en comparación con la sintaxis en la Figura 57 para `oop_dpb_maxbuffering_parameters`.

10 La variable `MaxSubLayers[setId][k]` para `setId` en el intervalo de 0 a `num_dpb_info_parameters - 1`, inclusive, se deriva de la siguiente manera:

[Tabla 30]

<code>for(setId = 0; setId < num_dpb_info_parameters; setId++) {</code>
<code> for(k = 1; k <= vps_max_layers_minus1; i++) {</code>
<code> MaxSubLayers[setId][k] = (max_sub_layers_vps_minus1[k]+1)</code>
<code> }</code>
<code>}</code>

En este caso, los parámetros `oop_dpb_maxbuffering_parameters(i, k)` se definirán como en la Figura 58.

15 Con referencia a la Figura 59, se ilustra una `vps_extension` modificada ejemplar. La extensión de `vps` modificada incluye una modificación adicional de la sintaxis en la Figura 56 con una nueva sintaxis, esto es, una bandera adicional `layer_dpb_info_parameters_presence_flag` que se señala condicionalmente. `layer_dpb_info_parameters_presence_flag` hace que la señalización de `layer_dpb_info_parameters(i)` en la extensión de `VPS` sea opcional.

20 `layer_dpb_info_parameters_presence_flag[i]` igual a 1 especifica que los elementos de sintaxis `sub_layer_vps_ordering_info_predict_flag[i]` y `layer_dpb_info_parameters(i)` están presentes para las capas `vps_max_num_layers_minus1`. `layer_dpb_info_parameters_presence_flag[i]` igual a 0 especifica que los elementos de sintaxis `sub_layer_vps_ordering_info_predict_flag[i]` y `layer_dpb_info_parameters(i)` no están presentes para las capas `vps_max_num_layers_minus1`.

Los `oop_dpb_maxbuffering_parameters(i, k)` son como se muestra en la Figura 60.

25 Con referencia a la Figura 61, el `oop_dpb_maxbuffering_parameters` especifica `sub_layer_vps_buf_info_present_flag[i][k]`, `max_vps_dec_pic_buffering_minus1[i][k][j]`. La Figura 61 es una variante de sintaxis para `oop_dpb_maxbuffering_parameters` en comparación con la sintaxis en la Figura 60 para `oop_dpb_maxbuffering_parameters`.

30 La variable `MaxSubLayers [setId][k]` para `setId` en el intervalo de 0 a `num_dpb_info_parameters - 1`, inclusive, se deriva de la siguiente manera:

[Tabla 31]

<code>for(setId = 0; setId < num_dpb_info_parameters; setId++) {</code>
<code> for(k = 1; k <= vps_max_layers_minus1; i++) {</code>
<code> MaxSubLayers[setId][k] =(max_sub_layers_vps_minus1[k]+1)</code>
<code> }</code>
<code>}</code>

En este caso, los parámetros `oop_dpb_maxbuffering_parameters(i, k)` se definirán como en la Figura 61.

Una alternativa ejemplar para la sintaxis en la extensión de VPS es que

[Tabla 32]

for(k = 1; j <= vps_max_layers_minus1; k++)
oop_dpb_maxbuffering_parameters(k)

se puede cambiar a

5 [Tabla 33]

for(k = 0; k <= vps_max_layers_minus1; k++)
oop_dpb_maxbuffering_parameters(k)

De este modo, el índice k podría comenzar en 0 en lugar de en 1.

Una alternativa ejemplar para la sintaxis en la extensión de VPS es que

[Tabla 34]

10

for(k = 0; k <= vps_max_layers_minus1; k++)
oop_dpb_maxbuffering_parameters(k)

se puede cambiar a

[Tabla 35]

for(k = 0; k <= vps_max_layer_id; k++)
oop_dpb_maxbuffering_parameters(k)

15

El vps_max_layer_id especifica el valor máximo permitido de nuh_layer_id de todas las unidades de NAL en la CVS. El vps_max_layers_minus1, especifica el número máximo de capas que pueden estar presentes en la CVS, en donde una capa puede, por ejemplo, ser una capa escalable espacial, una capa escalable de calidad, una vista de textura o una vista de profundidad.

Otra alternativa ejemplar para la sintaxis en la extensión de VPS es que

[Tabla 36]

20

for(k = 1; k <= vps_max_layers_minus1; k++)
oop_dpb_maxbuffering_parameters(I,k)

se puede cambiar a

[Tabla 37]

for(k = 1; k < numOutputLayers; k++)
oop_dpb_maxbuffering_parameters(I,k)

donde numOutputLayers se deriva como

25

[Tabla 38]

```

for( setId = 0; setId < num_dpb_info_parameters; setId++) {
    lsIdx =
output_layer_set_idx_minus1[ output_point_layer_set_idx[ setId ] ] + 1
    numOutputLayers = [ numLayersInIdList[ lsIdx ] - 1 ]
}
    
```

o se podría cambiar a

[Tabla 39]

```

for( k = 1; k <=
numLayersInIdList[output_layer_set_idx_minus1[output_
point_layer_set_idx[ i ] ]]-1; i++) {
    oop_dpb_maxbuffering_parameters( i, k )
}
    
```

- 5 En una realización, se puede señalar una bandera adicional para indicar si oop_dpb_information_parameters se señalan para la capa particular de la siguiente manera:

[Tabla 40]

for(k = 1; k <= vps_max_layers_minus1; i++) {	
vps_layer_info_present_flag[k]	u(1)
oop_dpb_maxbuffering_parameters(i, k)	
}	

- 10 vps_layer_info_present_flag[k] igual a 1 especifica que oop_dpb_maxbuffering_parameters están presentes para la capa de orden k para el conjunto de capas de salida particular. vps_layer_info_present_flag[k] igual a 0 especifica que oop_dpb_maxbuffering_parameters no están presentes para la capa de orden k para el conjunto de capas de salida particular.

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

- 15 Se usan dos tipos de conjuntos de parámetros de HRD (parámetros de HRD de NAL y parámetros de HRD de VCL). Los conjuntos de parámetros de HRD se señalan a través de la estructura de sintaxis hrd_parameters(), que puede ser parte de la estructura de sintaxis de SPS o la estructura de sintaxis de VPS.

- 20 Pueden ser necesarias múltiples pruebas para comprobar la conformidad de un flujo de bits, al que se hace referencia como el flujo de bits bajo prueba. Para cada prueba, los siguientes pasos se aplican en el orden enumerado:

- 25 (1) Se selecciona un conjunto de capas de salida bajo prueba, denotado como TargetOpLs. El punto de operación referido en TargetOpLs por output_layer_set_idx[] identifica el punto de operación bajo prueba. La lista de identificadores de capas de salida OpLayerIdList de TargetOpLs consiste en la lista de valores nuh_layer_id, en orden creciente de los valores de nuh_layer_id, presente en el subconjunto del flujo de bits asociado con TargetOp y TargetOpLs, que es un subconjunto de los valores de nuh_layer_id presentes en el flujo de bits bajo prueba. El OpTid de TargetOp es igual al TemporalId más alto presente en el subconjunto del flujo de bits asociado con TargetOp.

- 30 (2) TargetDecLayerIdList se establece igual a la lista de identificadores de capas decodificadas objetivo targetDLayerIdList para el conjunto de capas de salida seleccionado TargetOpLs, HighestTid se establece igual a OpTid de TargetOp, y el proceso de extracción del flujo de bits secundario como se especifica en la cláusula 10 se invoca con el flujo de bits bajo prueba, HighestTid y TargetDecLayerIdList como entradas, y la salida se asigna a BitstreamToDecode.

- 35 (3) Se seleccionan la estructura de sintaxis hrd_parameters() y la estructura de sintaxis sub_layer_hrd_parameters() aplicables a TargetOp. Si TargetDecLayerIdList contiene todos los valores de nuh_layer_id presentes en el flujo de bits bajo prueba, la estructura de sintaxis hrd_parameters() en el SPS activo (o se proporciona a través de un medio externo no especificado en esta Especificación). De otro modo, se selecciona la estructura de sintaxis

hrd_parameters() en el VPS activo (o se proporciona a través de algún medio externo no especificado en esta Especificación) que se aplica a TargetOp. Dentro de la estructura de sintaxis de hrd_parameters() seleccionada, si BitstreamToDecode es un flujo de bits de Tipo I, se selecciona la estructura de sintaxis sub_layer_hrd_parameters(HighestTid) que sigue inmediatamente a la condición "if(vcl_hrd_parameters_present_flag)" y la variable NalHrdModeFlag se establece igual a 0; de otro modo (BitstreamToDecode es un flujo de bits de Tipo II), se selecciona la estructura de sintaxis sub_layer_hrd_parameters(HighestTid) que sigue inmediatamente o bien a la condición "if(vcl_hrd_parameters_present_flag)" (en este caso, la variable NalHrdModeFlag se establece igual a 0) o bien la condición "if(nal_hrd_parameters_present_flag)" (en este caso, la variable NalHrdModeFlag se establece igual a 1). Cuando BitstreamToDecode es un flujo de bits de Tipo II y NalHrdModeFlag es igual a 0, todas las unidades de NAL no VCL, excepto las unidades de NAL de datos de relleno, y todos los elementos de sintaxis leading_zero_8bits, zero_byte, start_code_prefix_one_3bits, y trailing_zero_8bits que forman un flujo de bytes a partir del flujo de unidades de NAL (como se especifica en el Anexo B), cuando están presentes, se descartan de BitstreamToDecode, y el flujo de bits restante se asigna a BitstreamToDecode.

Un decodificador conforme puede cumplir todos los requisitos especificados en esta subcláusula.

(1) Un decodificador que reivindica conformidad con un perfil, fila y nivel específicos será capaz de decodificar con éxito todos los flujos de bits que se ajusten a los requisitos de conformidad del flujo de bits especificados en la subcláusula C.4, de la manera especificada en el Anexo A, a condición de que todos los VPS, SPS y PPS a los que se hace referencia en las unidades de NAL de VCL, y el período de almacenamiento temporal y los mensajes de SEI de temporización de imagen apropiados se transmiten al decodificador, de una manera oportuna, o bien en el flujo de bits (por unidades de NAL no VCL), o bien por medios externos no especificados en esta Especificación.

(2) Cuando un flujo de bits contiene elementos de sintaxis que tienen valores que se especifican como reservados y se especifica que los decodificadores ignorarán valores de los elementos de sintaxis o unidades de NAL que contienen los elementos de sintaxis que tienen los valores reservados, y el flujo de bits, de otro modo, es conforme con esta especificación, un decodificador conforme decodificará el flujo de bits de la misma manera que decodificaría un flujo de bits conforme e ignorará los elementos de sintaxis o las unidades de NAL que contienen elementos de sintaxis que tienen valores reservados como se especifica.

Hay dos tipos de conformidad de un decodificador: conformidad de temporización de salida y conformidad de orden de salida.

Para comprobar la conformidad de un decodificador, los flujos de bits de prueba conformes con el perfil, fila y nivel reivindicados, como se especifica en la subcláusula C.4, se entregan por un programador de flujo hipotético (HSS) tanto al HRD como al decodificador bajo prueba (DUT). Todas las imágenes decodificadas recortadas emitidas por el HRD también se emitirán por el DUT, cada imagen decodificada recortada emitida por el DUT será una imagen con PicOutputFlag igual a 1, y, para cada imagen decodificada recortada emitida por el DUT, los valores de todas las muestras que se emiten serán iguales a los valores de las muestras producidas por el proceso de decodificación especificado.

Para conformidad del decodificador de temporización de salida, el HSS funciona como se ha descrito anteriormente, con programaciones de entrega seleccionadas solamente del subconjunto de valores de SchedSelIdx para los cuales la tasa de bits y el tamaño de CPB están restringidos como se especifica en el Anexo A para el perfil, fila y nivel especificados, o con programaciones de entrega "interpoladas" como se especifica a continuación para las cuales la tasa de bits y el tamaño de CPB están restringidos como se especifica en el Anexo A. La misma programación de entrega se usa tanto para el HDR como para el DUT.

Cuando los parámetros de HRD y los mensajes de SEI de período de almacenamiento temporal están presentes con cpb_cnt_minus1[HighestTid] mayor que 0, el decodificador será capaz de decodificar el flujo de bits como se entrega desde el HSS operando usando una programación de entrega "interpolada" especificada como que tiene una tasa pico de bits r , tamaño de CPB $c(r)$ y retardo de eliminación de CPB inicial

[Expresión matemática 16]

$$(f(r) \div r)$$

de la siguiente manera:

[Tabla 41]

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

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

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

para cualquier SchedSelIdx > 0 y r de manera que BitRate[SchedSelIdx - 1] <= r <= BitRate[SchedSelIdx] de manera que r y c(r) estén dentro de los límites como se especifica en el Anexo A para la tasa de bits máxima y el tamaño del almacenador temporal para el perfil, fila y nivel especificados. El InitCpbRemovalDelay[SchedSelIdx] puede ser diferente de un período de almacenamiento temporal a otro y tener que ser vuelto a calcular.

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

Para conformidad del decodificador de orden de salida, se aplica lo siguiente:

(1) El HSS entrega el flujo de bits BitstreamToDecode al DUT “por demanda” del DUT, lo que significa que el HSS entrega bits (en orden de decodificación) solamente cuando el DUT requiere más bits para seguir con su procesamiento. Esto significa que para esta prueba, el almacenador temporal de imágenes codificadas del DUT podría ser tan pequeño como el tamaño de la unidad de decodificación más grande.

(2) Se usa un HRD modificado como se describe a continuación, y el HSS entrega el flujo de bits al HRD mediante una de las programaciones especificadas en el flujo de bits BitstreamToDecode, de manera que la tasa de bits y el tamaño de CPB se restringen como se especifica en el Anexo A. El orden de salida de las imágenes será el mismo tanto para el HRD como para el DUT.

(3) El tamaño de CPB de HRD se da por CpbSize[SchedSelIdx] como se especifica en la subcláusula E.2.3, donde SchedSelIdx y los parámetros de HRD se seleccionan como se especifica en la subcláusula C.1. El tamaño de DPB se da por sps_max_dec_pic_buffering_minus1[HighestTid] + 1 del SPS activo (cuando nuh_layer_id para la imagen decodificada actual es igual a 0) o del SPS de capa activo para el valor de nuh_layer_id de la imagen decodificada actual.

En algunos casos, si los parámetros de información de DPB de conjuntos de capas de salida oop_dpb_maxbuffering_parameters() están presentes para el conjunto de capas de salida seleccionado, el tamaño de DPB se da por max_vps_dec_pic_buffering_minus1[CurrLayerId][HighestTid] donde currLayerId es el nuh_layer_id de la imagen decodificada actual. De otro modo, si los parámetros de información de DPB de conjuntos de capas de salida oop_dpb_maxbuffering_parameters() no están presentes para el conjunto de capas de salida seleccionado, el tamaño de DPB se da por sps_max_dec_pic_buffering_minus1[HighestTid] + 1 del SPS activo (cuando nuh_layer_id para la imagen decodificada actual es igual a 0) o del SPS de capa activo para el valor de nuh_layer_id de la imagen decodificada actual).

El tiempo de eliminación del CPB para el HRD es el tiempo de llegada de bits final y la decodificación es inmediata. La operación del DPB de este HRD es como se describe en las subcláusulas C.5.2 a C.5.2.3.

El almacenador temporal de imágenes decodificadas contiene almacenadores temporales de almacenamiento de imágenes para nuh_layer_id igual a 0 se deriva del SPS activo. El número de almacenadores temporales de almacenamiento de imágenes para cada valor nuh_layer_id distinto de cero se deriva del SPS de capa activo para ese valor nuh_layer_id distinto de cero. Cada uno de los almacenadores temporales de almacenamiento de imágenes contiene una imagen decodificada que está marcada como “usada para referencia” o se mantiene para salida futura. El proceso de salida y eliminación de imágenes del DPB se invoca como se especifica en la subcláusula F.13.5.2.2, seguido por la invocación del proceso para la decodificación, marcado, sacudida adicional y almacenamiento de imágenes como se especifica en la subcláusula F.13.5.2.3. El proceso de “sacudidas” se especifica en la subcláusula F.13.5.2.4 y se invoca como se especifica en las subcláusulas F.13.5.2.2 y F.13.5.2.3.

La salida y eliminación de imágenes del DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de porción de la primera porción de la imagen actual) ocurre instantáneamente cuando la primera unidad de decodificación de la unidad de acceso que contiene la imagen actual se elimina del CPB y procede de la siguiente manera.

El proceso de decodificación para RPS se invoca como se especifica en la subcláusula 8.3.2.

(1) Si la imagen actual es una imagen de IRAP con NoRaslOutputFlag igual a 1 y con nuh_layer_id igual a 0 que no es la imagen 0, se aplican los siguientes pasos ordenados:

(A) La variable NoOutputOfPriorPicsFlag se deriva para el decodificador bajo prueba de la siguiente manera:

(i) Si la imagen actual es una imagen de CRA, NoOutputOfPriorPicsFlag se establece igual a 1 (independientemente del valor de no_output_of_prior_pics_flag).

5 (ii) De otro modo, si el valor de pic_width_in_luma_samples, pic_height_in_luma_samples, o
sps_max_dec_pic_buffering_minus1[HighestTid] derivado del SPS activo es diferente del valor de
pic_width_in_luma_samples, pic_height_in_luma_samples, o sps_max_dec_pic_buffering_minus1[HighestTid],
respectivamente, derivado del SPS activo para la imagen anterior, NoOutputOfPriorPicsFlag se puede (pero no se
debería) establecer en 1 por el decodificador bajo prueba, independientemente del valor de
10 no_output_of_prior_pics_flag. Aunque el ajuste de NoOutputOfPriorPicsFlag igual a no_output_of_prior_pics_flag se
prefiere bajo estas condiciones, no se permite al decodificador bajo prueba que establezca NoOutputOfPriorPicsFlag
en 1 en este caso.

(iii) De otro modo, NoOutputOfPriorPicsFlag se establece igual a no_output_of_prior_pics_flag.

(B) El valor de NoOutputOfPriorPicsFlag derivado del decodificador bajo prueba se aplica para el HRD de la siguiente manera:

15 (i) Si NoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes
en el DPB se vacían sin la salida de las imágenes que contienen, y la amplitud de DPB se establece igual a 0.

(ii) De otro modo (NoOutputOfPriorPicsFlag es igual a 0), todos los almacenadores temporales de almacenamiento
de imágenes que contienen una imagen que está marcada como “no necesaria para la salida” y “no usada para
referencia” se vacían (sin salida), y todos los almacenadores temporales de almacenamiento de imágenes no vacíos
20 en el DPB se vacían invocando repetidamente el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4,
y la amplitud de DPB se establece igual a 0.

(iii) De otro modo (la imagen actual no es una imagen de IRAP con NoRasIOutputFlag igual a 1 y con nuh_layer_id
igual a 0), todos los almacenadores temporales de almacenamiento de imágenes que contienen una imagen que
están marcados como “no necesarios para la salida” y “no usados para referencia” se vacían (sin salida). Para cada
almacenador temporal de almacenamiento de imágenes que se vacía, la amplitud de DPB se disminuye en uno. La
variable currLayerId se establece igual a nuh_layer_id de la imagen decodificada actual.

Las variables MaxNumReorderPics[currLayerId][HighestTid], MaxLatencyIncreasePlus1[currLayerId][HighestTid],
MaxLatencyPictures[currLayerId][HighestTid], MaxDecPicBufferingMinus1[currLayerId][HighestTid] se derivan de la
siguiente manera:

30 Si los parámetros de información de DPB de capas layer_dpb_info_parameters() están presentes en VPS,
MaxNumReorderPics[currLayerId][HighestTid] se establece en vps_max_num_reorder_pics[HighestTid] cuando
currLayerId es igual a 0 o se establece en max_vps_num_reorder_pics[currLayerId][HighestTid] para el currLayerId
cuando currLayerId es mayor que 0. De otro modo, si los parámetros de información de DPB de capas
layer_dpb_info_parameters() no están presentes MaxNumReorderPics[currLayerId][HighestTid] se establece en
35 sps_max_num_reorder_pics[HighestTid] del SPS activo (cuando currLayerId es igual a 0) o del SPS de capa activo
por el valor de currLayerId.

Si los parámetros de información de DPB de capa layer_dpb_info_parameters() están presentes en VPS,
MaxLatencyIncreasePlus1[currLayerId][HighestTid] se establece en vps_max_latency_increase_plus1[HighestTid]
cuando currLayerId es igual a 0 o se establece en max_vps_latency_increase_plus1[currLayerId][HighestTid] para el
currLayerId cuando currLayerId es mayor que 0. Si los parámetros de información de DPB de capa
layer_dpb_info_parameters() están presentes en VPS, MaxLatencyPictures[currLayerId][HighestTid] se establece en
SpsMaxLatencyPictures[HighestTid] cuando currLayerId es igual a 0 o se establece en
VpsMaxLatencyPictures[currLayerId][HighestTid] para el currLayerId cuando currLayerId es mayor que 0. De otro
modo, si los parámetros de información de DPB de capa layer_dpb_info_parameters() no están presentes para el
45 punto de operación bajo prueba, MaxLatencyIncreasePlus1[currLayerId][HighestTid] se establece en
sps_max_latency_increase_plus1[HighestTid] del SPS activo (cuando currLayerId es igual a 0) o el SPS de capa
activo para el valor de currLayerId y MaxLatencyPictures[currLayerId][HighestTid] se establece en
SpsMaxLatencyPictures[currLayerId][HighestTid] derivado del SPS activo (cuando currLayerId es igual a 0) o del
SPS de capa activo para el valor de currLayerId.

50 Si los parámetros de información de DPB del punto de operación oop_dpb_maxbuffering_parameters() están
presentes para el conjunto de capas de salida seleccionado, MaxDecPicBufferingMinus1[currLayerId][HighestTid] se
establece en vps_max_dec_pic_buffering_minus1[HighestTid] cuando currLayerId es igual a 0 o se establece en
max_vps_dec_pic_buffering_minus1[currLayerId][HighestTid] para el currLayerId para el punto de operación bajo
prueba cuando currLayerId es mayor que 0. De otro modo, si los parámetros de información de DPB del punto de
operación oop_dpb_maxbuffering_parameters() no están presentes para el punto de operación bajo prueba,
55 MaxDecPicBufferingMinus1[currLayerId][HighestTid] se establece en
sps_max_dec_pic_buffering_minus1[HighestTid] del SPS activo (cuando currLayerId es igual a 0) o del SPS de capa

activo para el valor de currLayerId.

5 Cuando una o más de las siguientes condiciones son ciertas, el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4 se invoca repetidamente mientras que se reduce aún más la amplitud de DPB en uno por cada almacenador temporal de almacenamiento de imágenes adicional que se vacía, hasta que ninguna de las siguientes condiciones son ciertas:

(1) El número de imágenes con nuh_layer_id igual a currLayerId en el DPB que están marcadas como “necesarias para la salida” es mayor que MaxNumReorderPics[CurrLayerId][HighestTid].

10 (2) Si MaxLatencyIncreasePlus1[CurrLayerId][HighestTid] no es igual a 0 y hay al menos una imagen con nuh_layer_id igual a currLayerId en el DPB que está marcada como “necesaria para la salida” para la cual la variable asociada PicLatencyCount[currLayerId] es mayor o igual que MaxLatencyPictures[CurrLayerId][HighestTid].

(3) El número de imágenes con nuh_layer_id igual a currLayerId en el DPB es mayor o igual que MaxDecPicBuffering[CurrLayerId][HighestTid].

Los procesos especificados en esta subcláusula ocurren instantáneamente cuando la última unidad de decodificación de la unidad de acceso n que contiene la imagen actual se elimina del CPB.

15 La variable currLayerId se establece igual a nuh_layer_id de la imagen decodificada actual.

Para cada imagen en el DPB que está marcada como “necesaria para la salida” y que tiene un valor nuh_layer_id igual a currLayerId, la variable asociada PicLatencyCount[currLayerId] se establece igual a PicLatencyCount[currLayerId] + 1.

20 La imagen actual se considera como decodificada después de que se decodifica la última unidad de decodificación de la imagen. La imagen decodificada actual se almacena en un almacenador temporal de almacenamiento de imágenes vacío en el DPB, y se aplica lo siguiente:

(A) Si la imagen decodificada actual tiene PicOutputFlag igual a 1, se marca como “necesaria para la salida” y su variable asociada PicLatencyCount[currLayerId] se establece igual a 0.

25 (B) De otro modo (la imagen decodificada actual tiene PicOutputFlag igual a 0), se marca como “no necesaria para la salida”.

La imagen decodificada actual se marca como “usada para referencia a corto plazo”.

Cuando una o más de las siguientes condiciones son ciertas, el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4 se invoca repetidamente hasta que ninguna de las siguientes condiciones es cierta.

30 (A) El número de imágenes con nuh_layer_id igual a currLayerId en el DPB que están marcadas como “necesarias para la salida” es mayor que MaxNumReorderPics[CurrLayerId][HighestTid].

(B) MaxLatencyIncreasePlus1[CurrLayerId][HighestTid] no es igual a 0 y hay al menos una imagen con nuh_layer_id igual a currLayerId en el DPB que está marcada como “necesaria para la salida” para la cual la variable asociada PicLatencyCount[currLayerId] es mayor o igual que MaxLatencyPictures[CurrLayerId][HighestTid].

El proceso de “sacudidas” consiste en los siguientes pasos ordenados:

35 (A) Las imágenes que son las primeras para su salida se seleccionan como las que tienen el valor más pequeño de PicOrderCntVal de todas las imágenes en el DPB marcadas como “necesarias para la salida”.

40 (B) Estas imágenes se recortan, usando la ventana de recorte de conformidad especificada en el SPS activo para la imagen con nuh_layer_id igual a 0 o en el SPS de capa activo para un valor nuh_layer_id igual al de la imagen, las imágenes recortadas se emiten en orden ascendente de nuh_layer_id, y las imágenes se marcan como “no necesarias para la salida”.

(C) Se vacía cada almacenador temporal de almacenamiento de imágenes que contiene una imagen marcada como “no usada para referencia” y que incluía una de las imágenes que se recortó y emitió.

45 Con referencia a la FIG. 62, se ilustra una sintaxis del conjunto de parámetros de secuencia (sps) modificado seq_parameter_set_rbsp. El sps modificado incluye una sps_dpb_params_present_flag. En una realización basada en el valor de esta bandera y basada en el valor de nuh_layer_id, pueden no ser señalados algunos de los elementos de sintaxis (por ejemplo, sps_sub_layer_ordering_info_present_flag, sps_max_dec_pic_buffering_minus1[i], sps_max_num_reorder_pics[i], sps_max_latency_increase_plus1[i]).

50 sps_dpb_params_present_flag igual a 0 especifica que, para un SPS con nuh_layer_id > 0 los elementos de sintaxis sps_sub_layer_ordering_info_present_flag, sps_max_dec_pic_buffering_minus1[i], sps_max_num_reorder_pics [i], sps_max_latency_increase_plus1[i] no están presentes en este SPS. El valor de estos parámetros se establece

5 igual al valor de los parámetros `sub_layer_vps_buf_info_present_flag[i]`, `max_vps_num_reorder_pics[i][j]` y `max_vps_latency_increase_plus1[i][j]` señalados en la extensión de VPS activa. `sps_dpb_params_present_flag` igual a 1 especifica que, para SPS con `nuh_layer_id > 0` los elementos de sintaxis `sps_sub_layer_ordering_info_present_flag`, `sps_max_dec_pic_buffering_minus1[i]`, `sps_max_num_reorder_pics[i]`, `sps_max_latency_increase_plus1[i]` están presentes en este SPS.

En otra realización, uno o más de los elementos de sintaxis se pueden señalar usando un número fijo de bits conocidos en lugar de `u(v)` en lugar de `ue(v)`. Por ejemplo, se podrían señalar usando `u(8)` o `u(16)` o `u(32)` o `u(64)`, etc.

10 En otra realización, uno o más de estos elementos de sintaxis se podrían señalar con `ue(v)` o algún otro esquema de codificación en lugar de un número fijo de bits tal como codificación `u(v)`.

En otra realización, los nombres de varios elementos de sintaxis y su semántica se pueden alterar sumando un más 1 o más 2 o restando un menos 1 o un menos 2 en comparación con la sintaxis y la semántica descritas.

15 En otra realización más, se pueden señalar diversos elementos de sintaxis por imagen en cualquier lugar del flujo de bits. Por ejemplo, se pueden señalar en la cabecera de segmento de porción, `pps/sps/vps/o` cualquier otro conjunto de parámetros u otra parte normativa del flujo de bits.

En otra realización más, todos los conceptos definidos en esta invención relacionados con conjuntos de capas de salida se podrían aplicar a puntos de operación de salida [2,3] y/o a puntos de operación [1].

Ejemplo 3

20 Se describe un método para codificación de vídeo. El método incluye comenzar a analizar sintácticamente una primera cabecera de porción de una imagen actual. Se determina qué pasos realizados por un almacenador temporal de imágenes decodificadas (DPB) se basarán en imágenes y qué pasos se basarán en unidad de acceso (AU). Se realiza una eliminación del DPB. Se realiza una salida de imagen desde el DPB. Se realizan una decodificación y un almacenamiento de una imagen decodificada actual en el DPB. Se marca la imagen decodificada actual en el DPB. También se realiza una salida de imagen adicional desde el DPB.

25 En algunas configuraciones, la eliminación y la salida del DPB se pueden basar en al menos una bandera de salida de AU, tal como una bandera de salida de AU, una bandera de salida de AU no rasl y/o bandera no de salida de AU de imágenes anteriores. La bandera de salida de AU se puede derivar en base a los elementos de sintaxis señalados en el flujo de bits y otras condiciones. Las banderas de AU representan banderas derivadas y aplicadas a nivel de unidad de acceso (AU). En algunos casos, éstas serán banderas diferentes de banderas similares que se señalan o se derivan a nivel de imagen, tales como una bandera de salida de imagen (por ejemplo, `pic_output_flag`), una bandera de salida no rasl de imagen (por ejemplo, `NoRaslOutputFlag`) y/o una bandera no de salida de imagen de imágenes anteriores (por ejemplo, `no_output_of_prior_pics_flag`).

30 En una configuración, una primera bandera de salida de AU se puede derivar y usar por el DPB, y una segunda bandera de salida de AU se deriva y usa por el DPB. En otra configuración, todos los valores del elemento de sintaxis de la bandera de salida de imagen se pueden restringir al mismo valor para todas las imágenes codificadas en una AU cuando una bandera de salida de imagen está presente en la primera cabecera de porción. En algunas configuraciones, todos los valores de elementos de sintaxis de bandera de salida de imagen se pueden restringir al mismo valor para todas las imágenes codificadas en una AU cuando no está presente una bandera de salida de imagen en la primera cabecera de porción.

35 En una configuración, la eliminación se puede basar en imágenes, la salida de imagen se puede basar en AU (unidad de acceso), el almacenamiento y la decodificación se pueden basar en imágenes, el marcado se puede basar en imágenes, y la salida de imagen adicional se puede basar en AU. La eliminación del DPB puede eliminar una o más imágenes del DPB antes de la decodificación de la imagen actual.

40 En otra configuración, la eliminación se puede basar en imágenes, y la salida de imagen, la decodificación y el almacenamiento, el marcado y la salida de imagen adicional se pueden basar en AU. En otra configuración, la eliminación, la salida de imagen, la decodificación y el almacenamiento, el marcado y la salida de imagen adicional se pueden basar en AU. En otra configuración, la eliminación, la decodificación y el almacenamiento y el marcado se pueden basar en imágenes, y la salida de imagen y la salida de imagen adicional se pueden basar en AU. En otra configuración más, la eliminación, la salida de imagen, la decodificación y el almacenamiento, el marcado y la salida de imagen adicional se pueden basar en imagen.

45 El marcado de imagen decodificada actual en el DPB puede incluir un paso de marcado de referencia y un paso de marcado de salida. El paso de marcado de referencia se puede basar en imágenes y el paso de marcado de salida se puede basar en AU. Una amplitud de DPB se puede aumentar en uno cuando una imagen decodificada se almacena en el DPB en un almacenador temporal de almacenamiento vacío. La amplitud de DPB puede disminuir en uno cuando se emite una imagen desde el DPB. La amplitud del DPB se puede rastrear por capa. La amplitud de DPB también se puede rastrear para un conjunto de capas de salida.

El DPB puede incluir almacenadores temporales de imágenes identificadas y gestionadas por separado para imágenes decodificadas que tienen una o más de diferentes resoluciones, diferentes profundidades de bits y diferente cromaticidad de color. El DPB puede incluir una agrupación común de almacenadores temporales de almacenamiento de imágenes. Una imagen decodificada se puede almacenar en los almacenadores temporales de almacenamiento de imágenes en base a al menos uno de tamaño, resolución y profundidad de bits. En una configuración, una imagen decodificada se puede almacenar en una ranura de almacenador temporal de imágenes de los almacenadores temporales de almacenamiento de imágenes.

El método se puede realizar por un decodificador dentro de un dispositivo electrónico que se ajuste a un estándar de codificación de vídeo de alta eficiencia escalable (SHVC). El método también se puede realizar por un decodificador dentro de un dispositivo electrónico que se ajuste a un estándar de codificación de vídeo de alta eficiencia de vista múltiple (MV-HEVC).

También se describe un dispositivo electrónico configurado para codificación de vídeo. El dispositivo electrónico incluye un procesador y una memoria en comunicación electrónica con el procesador. Las instrucciones en la memoria son ejecutables para comenzar a analizar sintácticamente una primera cabecera de porción de una imagen actual. Las instrucciones en la memoria también son ejecutables para determinar qué pasos realizados por un almacenador temporal de imágenes decodificadas (DPB) se basarán en imágenes y qué pasos se basarán en unidades de acceso (AU). Las instrucciones en la memoria son, además, ejecutables para realizar una eliminación del DPB. Las instrucciones en la memoria también son ejecutables para realizar una salida de imagen desde el DPB. Las instrucciones en la memoria son ejecutables además para realizar una decodificación y un almacenamiento de una imagen decodificada actual en el DPB. Las instrucciones en la memoria también son ejecutables para marcar la imagen decodificada actual en el DPB. Las instrucciones en la memoria son ejecutables además para realizar una salida de imagen adicional desde el DPB.

Ahora se describen diversas configuraciones con referencia a las Figuras, donde números de referencia iguales pueden indicar elementos funcionalmente similares.

La Figura 63 es un diagrama de bloques que ilustra codificación de vídeo entre múltiples dispositivos electrónicos 2102a-b. Se ilustran un primer dispositivo electrónico 2102a y un segundo dispositivo electrónico 2102b. No obstante, se debería observar que una o más de las características y funcionalidad descritas en relación con el primer dispositivo electrónico 2102a y el segundo dispositivo electrónico 2102b se pueden combinar en un único dispositivo electrónico 102 en algunas configuraciones. Cada dispositivo electrónico 102 se puede configurar para codificar vídeo y/o decodificar vídeo. Los dispositivos electrónicos 102 se pueden configurar para usar una operación de almacenador temporal de imágenes decodificadas (DPB) híbrida. La operación de almacenador temporal de imágenes decodificadas (DPB) híbrida se refiere a escenarios donde los diversos pasos de eliminación, salida (sacudidas), almacenamiento, marcado y salida (sacudidas) adicional realizados en un almacenador temporal de imágenes decodificadas (DPB) 116 ocurre o bien sobre una base de imágenes o bien sobre una base de unidades de acceso (AU). Aunque se hace referencia a combinaciones específicas de estos pasos como que se realizan sobre una base de imágenes o sobre una base de unidades de acceso (AU), se soportan todas las combinaciones posibles de realizar cada uno de estos pasos individualmente o bien sobre una base de imágenes o bien sobre una base de unidades de acceso (AU).

Como se usa en la presente memoria, unidad de acceso (AU) se refiere a un conjunto de unidades de capas de abstracción de red (NAL) que están asociadas unas con otras según una regla de clasificación especificada, que son consecutivas en orden de decodificación, y que incluyen las unidades de NAL de capa de codificación de vídeo (VCL) de todas las imágenes codificadas asociadas con el mismo tiempo de salida y sus unidades de NAL de no VCL asociadas. La capa base es una capa en la que todas las unidades de NAL de VCL tienen un nuh_layer_id igual a 0. Una imagen codificada es una representación codificada de una imagen que incluye unidades de NAL de VCL con un valor particular de nuh_layer_id y que incluye todas las unidades del árbol de codificación de la imagen. En algunos casos, una imagen codificada se puede denominar componente de capa. Detalles adicionales acerca de los pasos que se basan en imágenes o se basan en unidades de acceso (AU) se dan en relación con las Figuras 69 y 70 a continuación.

En una configuración, cada uno de los dispositivos electrónicos 102 puede cumplir con el estándar de Codificación de Vídeo de Alta Eficiencia (HEVC), el estándar de Codificación de Vídeo de Alta Eficiencia Escalable (SHVC) o el estándar de Codificación de Vídeo de Alta Eficiencia de Vista Múltiple (MV-HEVC). El estándar de HEVC es un estándar de compresión de vídeo que actúa como un sucesor de H.264/MPEG-4 AVC (Codificación de Vídeo Avanzada) y que proporciona una calidad de vídeo mejorada y un aumento en las relaciones de compresión de datos. Como se usa en la presente memoria, una imagen es una agrupación de muestras luma en formato monocromo o una agrupación de muestras luma y dos agrupaciones correspondientes de muestras croma en formato de color 4:2:0, 4:2:2 y 4:4:4 o algún otro formato de color. La operación de un decodificador de referencia hipotético (HRD) y la operación del almacenador temporal de imágenes decodificadas (DPB) de orden de salida 116 se describen para SHVC y MV-HEVC en JCTVC-M1008, JCTVC-L1008, JCTVC-D1004, JCT3V-C1004, JCTVC-L0453 y JCTVC-L0452.

El primer dispositivo electrónico 2102a puede incluir un codificador 2108 y un módulo de señalización de sobrecarga

2112. El primer dispositivo electrónico 2102a puede obtener una imagen de entrada 2106. En algunas configuraciones, la imagen de entrada 2106 se puede capturar en el primer dispositivo electrónico 2102a usando un sensor de imagen, recuperar de la memoria y/o recibir de otro dispositivo electrónico 102. El codificador 2108 puede codificar la imagen de entrada 2106 para producir datos codificados 2110. Por ejemplo, el codificador 2108 puede codificar una serie de imágenes de entrada 2106 (por ejemplo, vídeo). Los datos codificados 2110 pueden ser datos digitales (por ejemplo, un flujo de bits).

El módulo de señalización de sobrecarga 2112 puede generar una señalización de sobrecarga en base a los datos codificados 2110. Por ejemplo, el módulo de señalización de sobrecarga 2112 puede añadir datos de sobrecarga a los datos codificados 2110 tales como información de cabecera de porción, información de conjunto de parámetros de vídeo (VPS), información de conjunto de parámetros de secuencia (SPS), información de conjunto de parámetros de imagen (PPS), recuento de orden de imágenes (POC), designación de imagen de referencia, etc. En algunas configuraciones, el módulo de señalización de sobrecarga 2112 puede producir un indicador de envoltura que indica una transición entre dos conjuntos de imágenes.

El codificador 2108 (y el módulo de señalización de sobrecarga 2112, por ejemplo) puede producir un flujo de bits 2114. El flujo de bits 2114 puede incluir datos de imágenes codificadas en base a la imagen de entrada 2106. En algunas configuraciones, el flujo de bits 2114 también puede incluir datos de sobrecarga, tales como información de cabecera de porción, información de VPS, información de SPS, información de PPS, etc. A medida que se codifican imágenes de entrada 2106 adicionales, el flujo de bits 2114 puede incluir una o más imágenes codificadas. Por ejemplo, el flujo de bits 2114 puede incluir una o más imágenes de referencia codificadas y/u otras imágenes.

El flujo de bits 2114 se puede proporcionar a un decodificador 2104. En un ejemplo, el flujo de bits 2114 se puede transmitir al segundo dispositivo electrónico 2102b usando un enlace cableado o inalámbrico. En algunos casos, esto se puede hacer sobre una red, tal como Internet o una Red de Área Local (LAN). Como se ilustra en la Figura 63, el decodificador 2104 se puede implementar en el segundo dispositivo electrónico 2102b por separado del codificador 2108 en el primer dispositivo electrónico 2102a. No obstante, se debería observar que el codificador 2108 y el decodificador 2104 se pueden implementar en el mismo dispositivo electrónico 102 en algunas configuraciones. Cuando el codificador 2108 y el decodificador 2104 se implementan en el mismo dispositivo electrónico, por ejemplo, el flujo de bits 2114 se puede proporcionar sobre un bus al decodificador 2104 o almacenar en memoria para su recuperación por el decodificador 2104.

El decodificador 2104 puede recibir (por ejemplo, obtener) el flujo de bits 2114. El decodificador 2104 puede generar una imagen decodificada 2118 (por ejemplo, una o más imágenes decodificadas 2118) en base al flujo de bits 2114. La imagen decodificada 2118 se puede mostrar, reproducir, almacenar en memoria y/o transmitir a otro dispositivo, etc.

El decodificador 2104 puede incluir un almacenador temporal de imágenes decodificadas (DPB) 116. El almacenador temporal de imágenes decodificadas (DPB) 116 puede ser un almacenador temporal que contiene imágenes decodificadas para referencia, reordenación de salida o retardo de salida especificado para un decodificador de referencia hipotético (HRD). En un dispositivo electrónico 102, se puede usar un almacenador temporal de imágenes decodificadas (DPB) 116 para almacenar imágenes reconstruidas (por ejemplo, decodificadas) en el decodificador 2104. Estas imágenes almacenadas se pueden usar entonces, por ejemplo, en un mecanismo de inter-predicción. Cuando las imágenes se decodifican fuera de orden, las imágenes se pueden almacenar en el almacenador temporal de imágenes decodificadas (DPB) 116 de modo que se puedan mostrar más tarde en orden.

En el estándar H.264 o codificación de vídeo avanzada (AVC), la gestión del almacenador temporal de imágenes decodificadas (DPB) 116 (por ejemplo, borrado, adición de imágenes, reordenación de imágenes, etc.) se lleva a cabo usando las operaciones de control de gestión de memoria (MMCO). Están bajo consideración muchos planteamientos diferentes de gestión de almacenador temporal de imágenes decodificadas (DPB) 116.

El decodificador 2104 puede incluir un módulo de operación 2120 de almacenador temporal de imágenes decodificadas (DPB) híbrida. El módulo de operación 2120 de almacenador temporal de imágenes decodificadas (DPB) híbrida puede permitir planteamientos de gestión de almacenador temporal de imágenes decodificadas (DPB) 116 con pasos basados en imagen 2122 y/o planteamientos de gestión de almacenador temporal de imágenes decodificadas (DPB) 116 con pasos basados en unidad de acceso (AU) 2124. Por ejemplo, una ventaja del uso de los pasos basados en imagen 2124 para la eliminación, el almacenamiento y el marcado de referencia es que la memoria del almacenador temporal de imágenes decodificadas (DPB) 116 óptima se usará por varias capas. De este modo, la memoria total requerida cuando se usan pasos basados en imagen puede ser menor. Una ventaja del uso de pasos basados en unidad de acceso (AU) 2124 para la salida (incluida la salida, el marcado de salida y la salida adicional) es que se puede simplificar el proceso de salida.

La Figura 64 es un diagrama de flujo de un método 2200 para operación de un almacenador temporal de imágenes decodificadas (DPB) 116 híbrida. El método 2200 se puede realizar mediante un almacenador temporal de imágenes decodificadas (DPB) 116 como parte de un decodificador 2104 en un dispositivo electrónico 102. En una configuración, el método 2200 se puede realizar mediante un módulo de operación 2120 de almacenador temporal

de imágenes decodificadas (DPB) híbrida. El almacenador temporal de imágenes decodificadas (DPB) 116 puede comenzar analizando sintácticamente 2202 una primera cabecera de porción de una imagen actual. El almacenador temporal de imágenes decodificadas (DPB) 116 puede determinar 2204 qué pasos de la operación del almacenador temporal de imágenes decodificadas (DPB) híbrida estarán basados en imagen y qué pasos estarán basados en unidad de acceso (AU).

El almacenador temporal de imágenes decodificadas (DPB) 116 puede realizar 2206 una eliminación (sin salida) del almacenador temporal de imágenes decodificadas (DPB) 116. La eliminación puede eliminar imágenes del almacenador temporal de imágenes decodificadas (DPB) 116 antes de la decodificación de la imagen actual. El almacenador temporal de imágenes decodificadas (DPB) 116 puede realizar 2208 una salida de imagen (sacudida) desde el almacenador temporal de imágenes decodificadas (DPB) 116. Una salida de imagen (sacudida) puede referirse a la salida de imágenes del almacenador temporal de imágenes decodificadas (DPB) 116 en el tiempo de eliminación del almacenador temporal de imágenes decodificadas (CPB). En algunas configuraciones, el término sacudidas se puede usar para indicar la salida de una o más imágenes del almacenador temporal de imágenes decodificadas (DPB) 116. De este modo, los términos sacudidas y salida se pueden usar indistintamente.

El almacenador temporal de imágenes decodificadas (DPB) 116 puede decodificar 2210 y almacenar la imagen actual en el almacenador temporal de imágenes decodificadas (DPB) 116. El almacenador temporal de imágenes decodificadas (DPB) 116 puede marcar 2212 la imagen decodificada actual almacenada en el almacenador temporal de imágenes decodificadas (DPB) 116. Por ejemplo, el almacenador temporal de imágenes decodificadas (DPB) 116 puede marcar 2212 la imagen decodificada actual como “no usada para referencia”, “usada para referencia”, “necesaria para la salida” o “no necesaria para la salida”. El almacenador temporal de imágenes decodificadas (DPB) 116 también puede realizar 2214 otra salida de imagen (sacudidas adicionales) desde el almacenador temporal de imágenes decodificadas (DPB) 116. En algunas configuraciones, una eliminación/sacudida repetida de imágenes del almacenador temporal de imágenes decodificadas (DPB) 116 puede ocurrir hasta que se satisfagan ciertas condiciones.

La Figura 65 es un diagrama de flujo de otro método 2300 para operación del almacenador temporal de imágenes decodificadas (DPB) 116 híbrida. Por ejemplo, el método 2300 de la Figura 65 puede ser un método preferido para operación del almacenador temporal de imágenes decodificadas (DPB) 116 híbrida. El método 2300 se puede realizar mediante un almacenador temporal de imágenes decodificadas (DPB) 116 como parte de un decodificador 2104 en un dispositivo electrónico 102. En una configuración, el método 2300 se puede realizar mediante un módulo de operación de almacenador temporal de imágenes decodificadas (DPB) 2120 híbrido. El almacenador temporal de imágenes decodificadas (DPB) 116 puede comenzar analizando sintácticamente 2302 una primera cabecera de porción de una imagen actual. El término híbrida se refiere al hecho de que algunos de los pasos de operación del almacenador temporal de imágenes decodificadas (DPB) 116 se realizan sobre la base de una imagen y algunos de los pasos de operación del almacenador temporal de imágenes decodificadas (DPB) 116 se realizan sobre la base de una unidad de acceso (AU). El almacenador temporal de imágenes decodificadas (DPB) 116 puede realizar 2304 una eliminación basada en imagen (sin salida) del almacenador temporal de imágenes decodificadas (DPB) 116. El almacenador temporal de imágenes decodificadas (DPB) 116 puede realizar 2306 una salida (sacudida) de imagen basada en una unidad de acceso (AU) desde el almacenador temporal de imágenes decodificadas (DPB) 116. El almacenador temporal de imágenes decodificadas (DPB) 116 puede realizar 2308 decodificación y almacenamiento basados en imagen de la imagen actual en el almacenador temporal de imágenes decodificadas (DPB).

El almacenador temporal de imágenes decodificadas (DPB) 116 puede realizar 310 marcado basado en imagen de la imagen decodificada actual en el almacenador temporal de imágenes decodificadas (DPB). El paso de marcado realizado por el almacenador temporal de imágenes decodificadas (DPB) 116 se puede subdividir además para incluir tanto un paso de marcado de referencia como un paso de marcado de salida. Como se usa en la presente memoria, se hace referencia a las imágenes de referencia como “no usadas como referencia” o “usadas para referencia” como el paso de marcado de referencia. Una imagen decodificada en el almacenador temporal de imágenes decodificadas (DPB) 116 se puede marcar como solamente una de “no usada para referencia”, “usada para referencia a corto plazo” o “usada para referencia a largo plazo” en cualquier momento dado durante la operación del proceso de decodificación. Asignar una de estas marcas a una imagen, elimina automáticamente otra de las marcas que se asigna a la imagen. Cuando se hace referencia a una imagen como que está marcada como “usada para referencia”, esto se refiere colectivamente a la imagen que está marcada como o bien “usada para referencia a corto plazo” o bien “usada para referencia a largo plazo”, pero nunca ambas. Como se usa en la presente memoria, se hace referencia a las imágenes de marcado como “necesarias para la salida” o “no necesarias para la salida” como el paso de marcado de salida.

El almacenador temporal de imágenes decodificadas (DPB) 116 puede operar de tal manera que el paso de marcado de referencia y el paso de marcado de salida podrían ocurrir o bien sobre una base de imagen o bien sobre una base de unidad de acceso (AU). En general, se soportan todas las posibles combinaciones (típicamente cuatro combinaciones) de hacer estos dos pasos de marcado de forma individual sobre una base de imagen o sobre una base de unidad de acceso (AU). No obstante, puede ser preferible que el paso de marcado de referencia esté basado en imagen y que el paso de marcado de salida esté basado en unidad de acceso (AU).

El almacenador temporal de imágenes decodificadas (DPB) 116 también puede realizar 2312 un marcado de salida

basado en la unidad de acceso (AU) de la imagen decodificada actual. El almacenador temporal de imágenes decodificadas (DPB) 116 puede realizar 2314 otra salida (sacudida adicional) de imagen basada en unidad de acceso (AU) desde el almacenador temporal de imágenes decodificadas (DPB) 116. En algunas configuraciones, una eliminación/sacudida repetida de imágenes del almacenador temporal de imágenes decodificadas (DPB) 116 puede ocurrir hasta que se satisfagan ciertas condiciones.

En algunos planteamientos, las restricciones de conformidad de flujo de bits se pueden requerir para banderas en la cabecera de segmento de porción. En algunos casos, las restricciones se pueden aplicar para banderas a través de las imágenes que pertenecen a la misma unidad de acceso (AU). Por ejemplo, las banderas `pic_output_flag` y/o `no_output_of_prior_pics_flag` se pueden requerir para seguir las restricciones de conformidad del flujo de bits. Por ejemplo, JCTVC-L1003, JCTVC-M1008 y JCT3V-D1004 describen cada uno la señalización en la cabecera de segmento de porción usando `pic_output_flag` y `no_output_of_prior_pics_flag`. Además, las banderas tales como `PicOutputFlag`, `NoRaslOutputFlag` y `NoOutputOfPriorPicsFlag` son derivables en base a los elementos de sintaxis y a los tipos de unidades de NAL.

JCTVC-L1003, JCTVC-M1008 y JCT3V-D1004 también incluyen descripciones de DPB para HEVC, SHVC y MV-HEVC. JCTVC-M1008 SHVC Draft Text 1 proporciona un borrador de texto para una extensión escalable de HEVC. El JCT3V-D1004 MV-HEVC Draft Text 4 describe un borrador de texto para extensión de vista múltiple de HEVC.

El flujo de bits codificado de vídeo, según JCTVC-L1003, JCTVC-M1008 y/o JCT3V-D1004, puede incluir una estructura de sintaxis que se coloca en paquetes de datos lógicos a los que se hace referencia generalmente como unidades de Capa de Abstracción de Red (NAL). Cada unidad de NAL incluye una cabecera de unidad de NAL, tal como una cabecera de unidad de NAL de dos bytes (por ejemplo, 16 bits), para identificar el propósito de la carga útil de datos asociada. Por ejemplo, cada porción (y/o imagen) codificada se puede codificar en una o más unidades de NAL de porción (y/o imagen). Se pueden incluir otras unidades de NAL para otras categorías de datos, tales como por ejemplo, información de mejora suplementaria, una porción codificada de imagen de acceso de subcapa temporal (TSA), una porción codificada de imagen de acceso de subcapa temporal paso a paso (STSA), una porción codificada de una imagen posterior no TSA, no STSA, una porción codificada de imagen de acceso de enlace roto, una porción codificada de imagen de regeneración decodificada instantánea, una porción codificada de imagen de acceso aleatorio limpio, una porción codificada de imagen principal decodificable, una porción codificada de etiquetado para imagen de descarte, un conjunto de parámetros de vídeo, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un delimitador de unidad de acceso, un final de secuencia, un final de flujo de bits, datos de relleno y/o un mensaje de información de mejora de secuencia. La Tabla (7) a continuación ilustra un ejemplo de códigos de unidad de NAL y clases de tipo de unidad de NAL. También se pueden incluir otros tipos de unidades de NAL, según se desee.

También se debería entender que los valores de tipo de unidad de NAL para las unidades de NAL mostradas en la Tabla (7) se pueden reorganizar y reasignar. También, se pueden añadir o eliminar tipos de unidades de NAL adicionales.

Una imagen de intra punto de acceso intra aleatorio (IRAP) es una imagen codificada para la cual cada unidad de NAL de capa de codificación de vídeo tiene `nal_unit_type` en el intervalo de `BLA_W_LP` a `RSV_IRAP_VCL23`, inclusive, como se muestra en la Tabla (7) a continuación. Una imagen de IRAP incluye solamente porciones intra codificadas (I).

Una imagen de regeneración de decodificación instantánea (IDR) es una imagen de IRAP para la que cada unidad de NAL de capa de codificación de vídeo tiene `nal_unit_type` igual a `IDR_W_RADL` o `IDR_N_LP`, como se muestra en la Tabla (7). Una imagen de regeneración de decodificación instantánea (IDR) incluye solamente I porciones y puede ser la primera imagen en el flujo de bits en orden de decodificación, o puede aparecer más tarde en el flujo de bits.

Cada imagen de IDR es la primera imagen de una secuencia de vídeo codificada (CVS) en orden de decodificación. Una imagen de acceso de enlace roto (BLA) es una imagen de IRAP para la que cada unidad de NAL de capa de codificación de vídeo tiene `nal_unit_type` igual a `BLA_W_LP`, `BLA_W_RADL` o `BLA_N_LP`, como se muestra en la Tabla (7).

Una imagen de BLA incluye solamente I porciones, y puede ser la primera imagen en el flujo de bits en orden de decodificación, o puede aparecer más tarde en el flujo de bits. Cada imagen de BLA comienza una nueva secuencia de vídeo codificada, y tiene el mismo efecto en el proceso de decodificación que una imagen de IDR. No obstante, una imagen de BLA incluye elementos de sintaxis que especifican un conjunto de imágenes de referencia no vacío.

[Tabla 42]

<code>nal_unit_type</code>	Nombre de <code>nal_unit_type</code>	Contenido de unidad de NAL y estructura de sintaxis de carga útil de secuencia de bytes en bruto (RBSP)	Clase de tipo de unidad de NAL
0	TRAIL_N	Segmento de porción codificada de una imagen posterior	Capa de Codificación de

ES 2 708 969 T3

1	TRAIL_R	no TSA, no STSA slice_segment_layer_rbsp()	Vídeo (VCL)
2	TSA_N	Segmento de porción codificada de una imagen de acceso de subcapa temporal (TSA) slice_segment_layer_rbsp()	VCL
3	TSA_R		
4	STSA_N	Segmento de porción codificada de una imagen de acceso de subcapa temporal escalonada (STSA) slice_segment_layer_rbsp()	VCL
5	STSA_R		
6	RADL_N	Segmento de porción codificada de una imagen principal de acceso aleatorio (RADL) slice_segment_layer_rbsp()	VCL
7	RADL_R		
8	RASL_N	Segmento de porción codificada de una imagen principal de salto de acceso aleatorio (RASL) slice_segment_layer_rbsp()	VCL
9	RASL_R		
10	RSV_VCL_N10	Tipos de unidades de NAL de VCL de no referencia de subcapas no IRAP reservados	VCL
12	RSV_VCL_N12		
14	RSV_VCL_N14		
11	RSV_VCL_R11	Tipos de unidades de NAL de VCL de referencia de subcapas no IRAP reservados	VCL
13	RSV_VCL_R13		
15	RSV_VCL_R14		
16	BLA_W_LP	Segmento de porción codificada de una imagen de acceso de enlace roto (BLA) slice_segment_layer_rbsp()	VCL
17	BLA_W_RADL		
18	BLA_N_LP		
19	IDR_W_RADL	Segmento de porción codificada de una imagen de regeneración instantánea de decodificación (IDR) slice_segment_layer_rbsp()	VCL
20	IDR_N_LP		
21	CRA_NUT	Segmento de porción codificada de una imagen de acceso aleatorio limpio (CRA) slice_segment_layer_rbsp()	VCL
22	RSV_IRAP_VCL22	Tipos de unidades de NAL de VCL de IRAP reservados	VCL
23	RSV_IRAP_VCL23		
24..31	RSV_VCL24.. RSV_VCL31	Tipos de unidades de NAL de VCL de IRAP reservados	VCL
32	VPS_NUT	Conjunto de parámetros de vídeo video_parameter_set_rbsp()	Capa no de codificación de vídeo (no VCL)
33	SPS_NUT	Conjunto de parámetros de secuencia seq_parameter_set_rbsp()	no VCL
34	PPS_NUT	Conjunto de parámetros de imagen	no VCL

		pic_parameter_set_rbsp()	
35	AUD_NUT	Delimitador de unidad de acceso access_unit_delimiter_rbsp()	no VCL
36	EOS_NUT	Final de secuencia end_of_seq_rbsp()	no VCL
37	EOB_NUT	Final de flujo de bits end_of_bitstream_rbsp()	no VCL
38	FD_NUT	Datos de relleno filler_data_rbsp()	no VCL
39	PREFIX_SEI_NUT	Información de mejora suplementaria sei_rbsp()	no VCL
40	SUFFIX_SEI_NUT		
41..47	RSV_NVCL41.. RSV_NVL47	Reservado	no VCL
48..63	UNSPEC48.. UNSPEC63	Sin especificar	no VCL

Tabla (7)

5 Con referencia a la Tabla (8) a continuación, la sintaxis de cabecera de unidad de NAL puede incluir dos bytes de
 10 datos, esto es, 16 bits. El primer bit es un "forbidden_zero_bit" que siempre se establece en cero al inicio de una
 unidad de NAL. Los siguientes seis bits pueden ser un "nal_unit_type" que especifica el tipo de estructura de datos
 de cargas útiles de secuencias de bytes en bruto ("RBSP") incluidas en la unidad de NAL como se muestra en la
 Tabla (7) anterior. Los siguientes 6 bits pueden ser un "nuh_layer_id" que especifica el identificador de la capa. En
 algunos casos, estos seis bits se pueden especificar como "nuh_reserved_zero_6bits" en su lugar. Los
 nuh_reserved_zero_6bits pueden ser iguales a 0 en la especificación base del estándar. En una codificación de
 vídeo escalable y/o extensiones de sintaxis, nuh_layer_id puede especificar que esta unidad de NAL particular
 pertenece a la capa identificada por el valor de estos 6 bits.

El siguiente elemento de sintaxis puede ser "nuh_temporal_id_plus 1". El nuh_temporal_id_plus1 menos 1 puede
 especificar un identificador temporal para la unidad de NAL. El identificador temporal variable TemporalId se puede
 especificar como TemporalId = nuh_temporal_id_plus1 - 1. El identificador temporal TemporalId se usa para
 identificar una subcapa temporal. La variable HighestTid identifica la subcapa temporal más alta a ser decodificada.

15 [Tabla 43]

nal_unit_header() {	Descriptor
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
nuh_layer_id	u(6)
nuh_temporal_id_plus1	u(3)
}	

Tabla (8)

20 La Tabla (9) a continuación muestra una estructura de sintaxis de conjunto de parámetros de secuencia (SPS)
 ejemplar. pic_width_in_luma_samples especifica el ancho de cada imagen decodificada en unidades de muestras de
 luma. pic_width_in_luma_samples no será igual a 0. pic_height_in_luma_samples especifica la altura de cada
 imagen decodificada en unidades de muestras de luma. pic_height_in_luma_samples no será igual a 0.

sps_max_sub_layers_minus1 más 1 especifica el número máximo de subcapas temporales que pueden estar
 presentes en cada CVS en referencia al SPS. El valor de sps_max_sub_layers_minus1 oscila de 0 a 6, inclusive.

La bandera `sps_sub_layer_ordering_info_present_flag` igual a 1 especifica que están presentes los elementos de sintaxis `sps_max_dec_pic_buffering_minus1[i]`, `sps_max_num_reorder_pics[i]`, y `sps_max_latency_increase_plus1[i]` para las subcapas `sps_max_sub_layers_minus1 + 1`.
 5 `sps_sub_layer_ordering_info_present_flag` igual a 0 especifica que los valores de `sps_max_dec_pic_buffering_minus1[sps_max_sub_layers_minus1]`, `sps_max_num_reorder_pics[sps_max_sub_layers_minus1]`, y `sps_max_latency_increase_plus1[sps_max_sub_layers_minus1]` se aplican a todas las subcapas.

`sps_max_dec_pic_buffering_minus1[i]` más 1 especifica el tamaño máximo requerido del almacenador temporal de imágenes decodificadas para la CVS en unidades de almacenadores temporales de almacenamiento de imágenes cuando `HighestTid` es igual a `i`. El valor de `sps_max_dec_pic_buffering_minus1[i]` oscila de 0 a `MaxDpbSize - 1`, inclusive donde `MaxDpbSize` especifica el tamaño máximo de almacenador temporal de imágenes decodificadas en unidades de almacenadores temporales de almacenamiento de imágenes. Cuando `i` es mayor que 0, `sps_max_dec_pic_buffering_minus1[i]` será mayor o igual que `sps_max_dec_pic_buffering_minus1[i - 1]`. Cuando `sps_max_dec_pic_buffering_minus1[i]` no está presente para `i` en el intervalo de 0 a `sps_max_sub_layers_minus1 - 1`, inclusive, debido a que `sps_sub_layer_ordering_info_present_flag` es igual a 0, se infiere que es igual a `sps_max_dec_pic_buffering_minus1[sps_max_sub_layers_minus1]`.
 10
 15

`sps_max_num_reorder_pics [i]` indica el número máximo permitido de imágenes que pueden preceder a cualquier imagen en la CVS en orden de decodificación y seguir a esa imagen en orden de salida cuando `HighestTid` es igual a `i`. El valor de `sps_max_num_reorder_pics [i]` oscila de 0 a `sps_max_dec_pic_buffering_minus1[i]`, inclusive. Cuando `i` es mayor que 0, `sps_max_num_reorder_pics[i]` será mayor o igual que `sps_max_num_reorder_pics[i - 1]`. Cuando `sps_max_num_reorder_pics[i]` no está presente para `i` en el intervalo de 0 a `sps_max_sub_layers_minus1 - 1`, inclusive, debido a que `sps_sub_layer_ordering_info_present_flag` es igual a 0, se infiere que es igual a `sps_max_num_reorder_pics[sps_max_sub_layers_minus1]`.
 20

`sps_max_latency_increase_plus1[i]` no igual a 0 se puede usar para calcular el valor de `SpsMaxLatencyPictures[i]`, que especifica el número máximo de imágenes que pueden preceder a cualquier imagen en la CVS en orden de salida y seguir a esa imagen en orden de decodificación cuando `HighestTid` es igual a `i`. Cuando `sps_max_latency_increase_plus1[i]` no es igual a 0, el valor de `SpsMaxLatencyPictures[i]` se especifica como `SpsMaxLatencyPictures[i] = sps_max_num_reorder_pics[i] + sps_max_latency_increase_plus1[i] - 1`. Cuando `sps_max_latency_increase_plus1[i]` es igual a 0, no se expresa ningún límite correspondiente.
 25

El valor de `sps_max_latency_increase_plus1[i]` estará en el intervalo de 0 a $2^{32} - 2$, inclusive. Cuando `sps_max_latency_increase_plus1[i]` no está presente para `i` en el intervalo de 0 a `sps_max_sub_layers_minus1 - 1`, inclusive, debido a que `sps_sub_layer_ordering_info_present_flag` es igual a 0, se infiere que es igual a `sps_max_latency_increase_plus1[sps_max_sub_layers_minus1]`.
 30

[Tabla 44]

<code>seq_parameter_set_rbsp() {</code>
<code>...</code>
<code>sps_max_sub_layers_minus1</code>
<code>...</code>
<code>pic_width_in_luma_samples</code>
<code>pic_height_in_luma_samples</code>
<code>...</code>
<code>for(i = (sps_sub_layer_ordering_info_present_flag ? 0 :</code>
<code> sps_max_sub_layers_minus1);</code>
<code> I <= sps_max_sub_layers_minus1; i++) {</code>
<code> sps_max_dec_pic_buffering_minus1[i]</code>
<code> sps_max_num_reorder_pics[i]</code>
<code> sps_max_latency_increase_plus1[i]</code>
<code> }</code>
<code>...</code>
<code>}</code>

35

Tabla (9)

Además, JCTVC-L1003 describe el estándar de HEVC. Por ejemplo, se proporcionan detalles con respecto a `pic_out_flag` y `no_output_of_prior_pics_flag` en la Tabla (10) a continuación:

[Tabla 45]

SINTAXIS GENERAL DE SEGMENTO DE PORCIÓN

```

slice_segment_header() {
    first_slice_segment_in_pic_flag
    if( nal_unit_type>=BLA_W_LP&&nal_unit_type<=RSV_IRAP_VCL23 )
        no_output_of_prior_pics_flag
    slice_pic_parameter_set_id
    if( !first_slice_segment_in_pic_flag ) {
        if( dependent_slice_segments_enabled_flag )
            dependent_slice_segment_flag
        slice_segment_address
    }
    if( !dependent_slice_segment_flag ) {
        for( i = 0; i < num_extra_slice_header_bits; i++ )
            slice_reserved_flag[i]
        slice_type
        if( output_flag_present_flag )
            pic_output_flag
        ...
        ...
    }
}

```

Tabla (10)

En la Tabla (10), no_output_of_prior_pics_flag afecta a la salida de imágenes decodificadas previamente en el almacenador temporal de imágenes decodificadas (DPB) después de la decodificación de una imagen de IDR o un a de BLA que no es la primera imagen en el flujo de bits como se especifica en el Anexo C de JCTVC-L1003.

- 5 output_flag_present_flag igual a 1 indica que el elemento de sintaxis pic_output_flag está presente en las cabeceras de porción asociadas. output_flag_present_flag igual a 0 indica que el elemento de sintaxis pic_output_flag no está presente en las cabeceras de porción asociadas. pic_output_flag afecta los procesos de salida y eliminación de imágenes decodificadas como se especifica en el Anexo C de JCTVC-L1003. Cuando pic_output_flag no está presente, se infiere que es igual a 1.
- 10 Para el proceso de decodificación general (como se encuentra en 8.1 de JCTVC-L1003), PicOutputFlag se establece de la siguiente manera:
- Si la imagen actual es una imagen de RASL y NoRasIOutputFlag de la imagen de IRAP asociada es igual a 1, PicOutputFlag se puede establecer igual a 0.
 - De otro modo, PicOutputFlag se puede establecer igual a pic_output_flag.
- 15 Además, durante el proceso de decodificación general para generar imágenes de referencia no disponibles (como se encuentra en la sección 8.3.3.1 de JCTVC-L1003), el valor de PicOutputFlag para la imagen generada se puede establecer igual a 0 bajo ciertas condiciones.
- Cuando la imagen actual es una imagen de IRAP, se aplica lo siguiente:
- 20 - Si la imagen actual es una imagen de IDR, una imagen de BLA, la primera imagen en el flujo de bits en orden de decodificación, o la primera imagen que sigue a un final de unidad de NAL de secuencia en orden de decodificación, una variable NoRasIOutputFlag se puede establecer igual a 1.
- 25 - De otro modo, si algún medio externo no especificado en JCTVC-L1003 está disponible para establecer la variable HandleCraAsBlaFlag en un valor para la imagen actual, la variable HandleCraAsBlaFlag se puede establecer igual al valor proporcionado por el medio externo y la variable NoRasIOutputFlag se puede establecer igual a HandleCraAsBlaFlag.
- De otro modo, la variable HandleCraAsBlaFlag se puede establecer igual a 0 y la variable NoRasIOutputFlag se puede establecer igual a 0.
- 30 Como se ha descrito anteriormente, la unidad de acceso (AU) se refiere a un conjunto de unidades de capa de abstracción de red (NAL) que están asociadas entre sí según una regla de clasificación especificada, que son consecutivas en el orden de decodificación, y que incluyen las unidades de NAL de capa de codificación de vídeo (VCL) de todas las imágenes codificadas asociadas con el mismo tiempo de salida y sus unidades de NAL de no

VCL asociadas. La capa base es una capa en la que todas las unidades de NAL de VCL tienen un `nuh_layer_id` igual a 0. Una imagen codificada es una representación codificada de una imagen que incluye unidades de NAL de VCL con un valor particular de `nuh_layer_id` y que incluye todas las unidades del árbol de codificación de la imagen. En algunos casos, una imagen codificada se puede denominar componente de capa. Detalles adicionales acerca de los pasos que se basan en imágenes o en unidades de acceso (AU) se dan en relación con las Figuras 69 y 70 a continuación.

En algunas configuraciones, las restricciones de conformidad del flujo de bits para `pic_output_flag` y/o `no_output_of_prior_pics_flag` se pueden usar para imágenes codificadas en una unidad de acceso (AU). Además, tres nuevas banderas de salida de la unidad de acceso, bandera de salida de AU (por ejemplo, `AuOutputFlag`), la bandera de salida no de RASL de AU (por ejemplo, `AuNoRaslOutputFlag`) y la bandera no de salida de AU de imágenes anteriores (por ejemplo, `AuNoOutputOfPriorPicsFlag`), se pueden derivar para una AU en base al valor de varios elementos de sintaxis y tipos de unidades de NAL para las imágenes codificadas en la AU. En algunas configuraciones, la salida y la eliminación de las imágenes se pueden basar en estas tres banderas (por ejemplo, `AuOutputFlag`, `AuNoRaslOutputFlag` y `AuNoOutputOfPriorPicsFlag`) para SHVC y HEVC de vistas múltiples.

Por ejemplo, las restricciones de conformidad del flujo de bits para las extensiones de HEVC se pueden seguir como se describe en los sistemas y métodos en la presente memoria. En particular, las restricciones de conformidad del flujo de bits se pueden aplicar al flujo de bits de SHVC. Además, se pueden aplicar restricciones de conformidad del flujo de bits para el flujo de bits de MV-HEVC.

En una configuración, cuando está presente, se requiere que el valor de los elementos de sintaxis de cabecera de segmento de porción `pic_output_flag` se pueda requerir que sea el mismo en todas las cabeceras de segmento de porción de imágenes codificadas en una unidad de acceso (AU). En otra configuración, el valor de los elementos de sintaxis de cabecera de segmento de porción `pic_output_flag`, cuando está presente, puede ser el mismo en todas las cabeceras de segmento de porción de imágenes codificadas en una unidad de acceso (AU) cuando las imágenes codificadas tienen el mismo tipo de unidad de NAL.

En una configuración, cuando está presente, el valor de los elementos de sintaxis de cabecera de segmento de porción `pic_output_flag` para los segmentos de porción con `nuh_layer_id` igual al valor `nuh_layer_id` de una capa objetivo, puede ser el mismo en todas las cabeceras de segmento de porción de tales imágenes codificadas en una unidad de acceso (AU). En otra configuración, cuando está presente, el valor de los elementos de sintaxis de cabecera de segmento de porción `pic_output_flag` para los segmentos de porción con `nuh_layer_id` no igual al valor `nuh_layer_id` de una capa objetivo puede ser 0 en todas las cabeceras de segmento de porción de tales imágenes codificadas en una unidad de acceso (AU).

En una configuración, una capa objetivo puede ser una capa que pertenece a un conjunto de capas o un conjunto de capas objetivo o un conjunto de capas de salida como se define en JCTVC-L1003, JCTVC-M1008 o JCT3V-D1004. En otra configuración, una capa objetivo puede ser una capa que se pretende que sea decodificada. En otra configuración más, una capa objetivo puede ser una capa que se pretende que sea decodificada y emitir (mostrar o enviar de otro modo para su salida).

En algunas configuraciones, cuando está presente, el valor de los elementos de sintaxis de cabecera de segmento de porción `no_output_of_prior_pics_flag` puede ser el mismo en todas las cabeceras de segmento de porción de imágenes codificadas en una unidad de acceso (AU). En otras configuraciones, cuando está presente, el valor de los elementos de sintaxis de cabecera de segmento de porción `no_output_of_prior_pics_flag` puede ser el mismo en todas las cabeceras de segmento de porción de imágenes codificadas en una unidad de acceso (AU) cuando las imágenes codificadas tienen el mismo tipo de unidades de NAL. En una configuración, cuando está presente, el valor de los elementos de sintaxis de cabecera de segmento de porción `no_output_of_prior_pics_flag` para los segmentos de porción con `nuh_layer_id` igual al valor de `nuh_layer_id` de una capa objetivo, puede ser el mismo en todas las cabeceras de segmento de porción de tales imágenes codificadas en una unidad de acceso (AU).

En alguna configuración, los elementos de sintaxis `pic_output_flag` y/o `no_output_of_prior_pics_flag` pueden no ser señalados cuando `nuh_layer_id > 0`. En este caso, los valores para capas con `nuh_layer_id > 0` se pueden inferir para ser iguales a sus valores señalados para `nuh_layer_id` igual a 0.

En algunas configuraciones, se pueden emplear banderas adicionales, tales como `AuOutputFlag` y `AuNoRaslOutputFlag`. Las banderas `AuOutputFlag` y `AuNoRaslOutputFlag` se pueden derivar según un número de planteamientos. En un planteamiento o configuración, las dos banderas, `AuOutputFlag` y `AuNoRaslOutputFlag` se pueden derivar y usar para operación de DPB. `AuOutputFlag` se puede establecer igual a 1 si `PicOutputFlag` es igual a 1 para todas las imágenes en la AU. De otro modo, `AuOutputFlag` puede ser igual a 0.

En otro planteamiento, `AuOutputFlag` se puede establecer igual a 1 si `PicOutputFlag` es igual a 1 para al menos una imagen en la UA. De otro modo, `AuOutputFlag` se puede establecer igual a 0. De esta manera, en este caso, la `AuOutputFlag` se puede establecer igual a 0 si `PicOutputFlag` es igual a 0 para todas las imágenes en una AU.

En otro planteamiento más, `AuOutputFlag` se puede establecer igual a 1 si `PicOutputFlag` es igual a 1 para las imágenes que pertenecen a todas las capas de salida objetivo en la AU. De otro modo, `AuOutputFlag` se puede

establecer igual a 0.

Aún en otro planteamiento más, AuOutputFlag se puede establecer igual a 1 si PicOutputFlag es igual a 1 para la imagen que pertenece a al menos una capa de salida objetivo en la AU. De otro modo, AuOutputFlag se puede establecer igual a 0.

5 En un planteamiento, AuNoRasOutputFlag se puede establecer igual a 1 si NoRasOutputFlag es igual a 1 para todas las imágenes en la AU. De otro modo, AuNoRasOutputFlag se puede establecer igual a 0.

10 En otro planteamiento, AuNoRasOutputFlag se puede establecer igual a 1 si NoRasOutputFlag es igual a 1 para al menos una imagen en la AU. De otro modo, AuNoRasOutputFlag se puede establecer igual a 0. De este modo, en este caso, el AuNoRasOutputFlag se puede establecer igual a 0 si NoRasOutputFlag es igual a 0 para todas las imágenes en una AU.

En otro planteamiento más, AuNoRasOutputFlag se puede establecer igual a 1 si NoRasOutputFlag es igual a 1 para las imágenes que pertenecen a todas las capas de salida objetivo en la AU. De otro modo, AuNoRasOutputFlag puede ser igual a 0.

15 Aún en otro planteamiento más, AuNoRasOutputFlag se puede establecer igual a 1 si NoRasOutputFlag es igual a 1 para la imagen que pertenece a al menos una capa de salida objetivo en la AU. De otro modo, AuNoRasOutputFlag se puede establecer igual a 0.

En algunos de los planteamientos y configuraciones anteriores, la operación del DPB puede usar la AuOutputFlag en lugar del PicOutputFlag. Además, la operación de DPB puede usar AuNoRasOutputFlag en lugar de la NoRasOutputFlag.

20 Ejemplos que muestran el uso de banderas de salida de AU, tales como AuOutputFlag y AuNoRasOutputFlag, según los presentes sistemas y métodos, se proporcionan a continuación en el Listado (1A) y el Listado (2A) a continuación. Además, como se describe a continuación en el Listado (1), Listado (1A), Listado (2) y Listado (2A), la bandera de salida de AU AuNoOutputOfPriorPicsFlag se puede derivar y usar para la operación de DPB.

25 Los presentes sistemas y métodos se pueden implementar mediante cambios a los documentos de los estándares. El Listado (1) a continuación proporciona las secciones de JCTVC-L1003 que se cambiarían para adaptarse a los presentes sistemas y métodos.

Listado 1

C.3 Operación del almacenador temporal de imágenes decodificadas (DPB)

C.3.1 General

30 Las especificaciones en esta subcláusula se aplican independientemente a cada conjunto de parámetros de DPB seleccionados como se especifica en la subcláusula C.1. El DPB opera por separado o independientemente para cada capa. De este modo, los siguientes pasos tienen lugar por separado para cada imagen decodificada con un valor particular de nuh_layer_id.

35 El almacenador temporal de imágenes decodificadas contiene almacenadores temporales de almacenamiento de imágenes. Cada capa consiste en su propio conjunto de almacenadores temporales de almacenamiento de imágenes. De este modo, los almacenadores temporales de almacenamiento de imágenes de cada capa se asocian con el valor nuh_layer_id de la capa. Cada uno de los almacenadores temporales de almacenamiento de imágenes puede contener una imagen decodificada que está marcada como "usada para referencia" o que se mantiene para su salida futura. Los procesos especificados en las subcláusulas C.3.2, C.3.3 y C.3.4 se aplican secuencialmente como se especifica a continuación.

C.3.2 Eliminación de imágenes del DPB.

45 La eliminación de imágenes del DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de porción de la primera porción de la imagen actual) ocurre instantáneamente en el momento de eliminación de CPB de la primera unidad de decodificación de la imagen actual que pertenece a la unidad de acceso n (que contiene la imagen actual) y procede de la siguiente manera:

- Se invoca el proceso de decodificación para RPS como se especifica en la subcláusula 8.3.2.

- Cuando la imagen actual es una imagen de IRAP con NoRasOutputFlag igual a 1 que no es la imagen 0, se aplican los siguientes pasos ordenados:

1. La variable NoOutputOfPriorPicsFlag se deriva para el decodificador bajo prueba de la siguiente manera:

50 - Si la imagen actual es una imagen de CRA, NoOutputOfPriorPicsFlag se establece en 1 (independientemente del

valor de no_output_of_prior_pics_flag).

- De otro modo, si el valor de pic_width_in_luma_samples, pic_height_in_luma_samples, o sps_max_dec_pic_buffering_minus1[HighestTid] derivado del SPS activo correspondiente al valor nuh_layer_id de la imagen 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, derivado del SPS activo para la imagen precedente con el valor nuh_layer_id igual al valor nuh_layer_id de la imagen actual, NoOutputOfPriorPicsFlag se puede (pero no se debería) establecer en 1 por el decodificador bajo prueba, independientemente del valor de no_output_of_prior_pics_flag.

NOTA - Aunque se prefiere bajo estas condiciones establecer NoOutputOfPriorPicsFlag igual a no_output_of_prior_pics_flag, se permite que decodificador bajo prueba establezca NoOutputOfPriorPicsFlag en 1 en este caso.

- De otro modo, NoOutputOfPriorPicsFlag se establece igual a no_output_of_prior_pics_flag.

2. El valor de NoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica para el HRD, de manera que cuando el valor de NoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes correspondientes al valor nuh_layer_id de la imagen actual en el DPB se vacían sin salida de las imágenes que contienen, y la amplitud de DPB para el valor nuh_layer_id de la imagen actual se establece igual a 0.

En una realización, el valor de NoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica para el HRD, de manera que cuando el valor de NoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes correspondientes a todos los valores de nuh_layer_id en el DPB se vacían sin la salida de las imágenes que contienen, y la amplitud de DPB para todos los valores de nuh_layer_id se establece igual a 0.

En una realización, el valor de NoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica para el HRD, de manera que cuando el valor de NoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes PSB[currLayerId] con currLayerId igual al valor nuh_layer_id de la imagen actual en el DPB se vacían sin salida de las imágenes que contienen, y la amplitud de DPB DPBFullness[currLayerId] para el valor nuh_layer_id currLayerId de la imagen actual se establece igual a 0.

En una realización, el valor de NoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica para el HRD, de manera que cuando el valor de NoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes PSB[nuh_layer_id] para todos los valores de nuh_layer_id en el DPB se vacían sin salida de las imágenes que contienen, y la amplitud de DPB DPBFullness [nuh_layer_id] para todos los valores de nuh_layer_id se establece igual a 0.

- Cuando ambas de las siguientes condiciones son ciertas para cualquier imagen k en el almacenador temporal de almacenamiento de imágenes correspondiente al valor nuh_layer_id de la imagen actual en el DPB, todas de tales imágenes k en el DPB se eliminan del DPB:

- la imagen k se marca como “no usada para referencia”

- la imagen k tiene un PicOutputFlag igual a 0 o su tiempo de salida de DPB es menor o igual al tiempo de eliminación de CPB de la primera unidad de decodificación (denotada como unidad de decodificación m) de la imagen actual n; es decir, DpbOutputTime[k] es menor o igual que CpbRemovalTime (m)

- Para cada imagen que se elimina del DPB, la amplitud de DPB se disminuye en uno.

- En una realización para cada imagen k que tiene un valor de nuh_layer_id nuhLayerIdk que se elimina del DPB, la amplitud de DPB DPBFullness [nuhLayerIdk] se disminuye en uno.

C.3.3 Salida de imagen

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

Cuando la imagen n tiene PicOutputFlag igual a 1, su tiempo de salida de DPB DpbOutputTime [n] se deriva de la siguiente manera, donde la variable firstPicInBufferingPeriodFlag es igual a 1 si la unidad de acceso n es la primera unidad de acceso de un período de almacenamiento temporal y 0 de otro modo:

```

if( !SubPicHrdFlag ) {
    DpbOutputTime[ n ] = AuCpbRemovalTime[ n ] + ClockTick * picDpbOutputDelay (C-16)
    if( firstPicInBufferingPeriodFlag )
        DpbOutputTime[ n ] -= ClockTick * DpbDelayOffset
} else
    DpbOutputTime[ n ] = AuCpbRemovalTime[ n ] + ClockSubTick *
picSptDpbOutputDuDelay

```

5 donde picDpbOutputDelay es el valor de pic_dpb_output_delay en el mensaje de SEI de temporización de imagen asociado con la unidad de acceso n, y picSptDpbOutputDuDelay es el valor de pic_spt_dpb_output_du_delay, cuando está presente, en los mensaje de SEI de información de unidad de decodificación asociados con la unidad de acceso n, o el valor de pic_dpb_output_du_delay en el mensaje de SEI de temporización de imagen asociado con la unidad de acceso n cuando no hay ningún mensaje de SEI de información de la unidad de decodificación asociado con la unidad de acceso n o ningún mensaje de SEI de información de unidad de decodificación asociado con la unidad de acceso n tiene pic_spt_dpb_output_du_delay presente.

10 NOTA - Cuando el elemento de sintaxis pic_spt_dpb_output_du_delay no está presente en ningún mensaje de SEI de información de unidad de decodificación asociado con la unidad de acceso n, se infiere que el valor es igual a pic_dpb_output_du_delay en el mensaje de SEI de temporización de imagen asociado con una unidad de acceso n.

La salida de la imagen actual si su nuh_layer_id pertenece a la capa en la TargetDecLayerIdList está especificada de la siguiente manera:

- Si PicOutputFlag es igual a 1 y DpbOutputTime [n] es igual a AuCpbRemovalTime [n], se emite la imagen actual.

15 - De otro modo, si PicOutputFlag es igual a 0, la imagen actual no se emite, sino que se almacenará en el almacenador temporal de almacenamiento de imágenes correspondiente al valor nuh_layer_id de la imagen actual en el DPB como se especifica en la subcláusula C.3.4.

20 En una realización: De otro modo, si PicOutputFlag es igual a 0, la imagen actual no se emite, sino que se almacenará en el almacenador temporal de almacenamiento de imágenes PSB[currLayerId] correspondiente al valor nuh_layer_id currLayerId de la imagen actual en el DPB como se especifica en la subcláusula C.3.4.

25 - De otro modo (PicOutputFlag es igual a 1 y DpbOutputTime [n] es mayor que AuCpbRemovalTime [n]), la imagen actual se emite más tarde y se almacenará en el almacenador temporal de almacenamiento de imágenes correspondiente al valor nuh_layer_id de la imagen actual en el DPB (como se especifica en la subcláusula C.3.4) y se emite en el tiempo DpbOutputTime[n], a menos que se indique que no se emita por la decodificación o inferencia de no_output_of_prior_pics_flag igual a 1 en un tiempo que precede a DpbOutputTime[n].

En otra realización, los pasos anteriores se especifican para:

La salida de la imagen actual si su nuh_layer_id pertenece a una capa que pertenece al conjunto de capas de salida correspondiente al punto de operación (actual).

En otra realización más, los pasos anteriores se especifican para:

30 La salida de la imagen actual (sin comprobar si pertenece a la TargetDecLayerIdList o el conjunto de capas de salida para el punto de operación actual).

Cuando se emite, la imagen se recorta, usando la ventana de recorte de conformidad especificada en el SPS activo para la imagen.

35 Cuando la imagen n es una imagen que se emite y no es la última imagen del flujo de bits que se emite, el valor de la variable DpbOutputInterval[n] se deriva de la siguiente manera:

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

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

C.3.4 Marcado y almacenamiento de imagen decodificada actual

40 El proceso especificado en esta subcláusula ocurre instantáneamente en el tiempo de eliminación del CPB de la

unidad de acceso n, CpbRemovalTime[n].

La imagen decodificada actual se almacena en el DPB en un almacenador temporal de almacenamiento de imágenes vacío correspondiente al valor nuh_layer_id de la imagen actual, la amplitud de DPB para el valor nuh_layer_id de la imagen actual se aumenta en uno, y la imagen actual se marca como “usada para referencia a corto plazo”.

En una realización:

La imagen decodificada actual con nuh_layer_id igual a currLayerId se almacena en el DPB en un almacenador temporal de almacenamiento de imágenes vacío PSB[currLayerId] con currLayerId igual al valor nuh_layer_id de la imagen actual en el DPB, la amplitud del DPB para el valor nuh_layer_id de la imagen actual DPBFullness[currLayerId] se aumenta en uno, y la imagen actual se marca como “usada para referencia a corto plazo”.

C.4 Conformidad de flujo de bits

Se aplican las especificaciones de la subcláusula C.4.

C.5 conformidad del decodificador

F.8.1.1 C.5.1 General

Se aplican las especificaciones de la subcláusula C.5.1.

Listado (1)

Como se usa en el Listado (1) anterior, PSB se refiere a un almacenador temporal de almacenamiento de imágenes. DPBFullness se refiere a una variable usada para describir la amplitud del almacenador temporal de imágenes decodificadas (DPB) 116.

El Listado (1A) a continuación proporciona un planteamiento alternativo a la sección C.3.2 del Listado (1) de acuerdo a adaptarse a los presentes sistemas y métodos. En algunas configuraciones, el Listado (1A) solamente puede representar cambios a la sección C.3.2 en JCTVC-L1003. El Listado (1A) puede usar las banderas AuNoOutputOfPriorPicsFlag y AuNoRaslOutputFlag definidos anteriormente.

Listado 1A

C.3.2 Eliminación de imágenes de DPB

La eliminación de imágenes del DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de porción de la primera porción de la imagen actual) ocurre instantáneamente en el tiempo de eliminación de la primera unidad de decodificación de la imagen actual que pertenece a la unidad de acceso n (que contiene la imagen actual) y procede de la siguiente manera:

- Se invoca el proceso de decodificación para RPS como se especifica en la subcláusula 8.3.2.

- Cuando la imagen actual es una imagen de IRAP con AuNoRaslOutputFlag igual a 1 que no es la imagen 0, se aplican los siguientes pasos ordenados:

- En otra configuración, cuando la imagen actual es una imagen de IRAP con NoRaslOutputFlag igual a 1 que no es la imagen 0, se aplican los siguientes pasos ordenados:

1. La variable AuNoOutputOfPriorPicsFlag se deriva para el decodificador bajo prueba de la siguiente manera:

- Si la imagen actual es una imagen de CRA, AuNoOutputOfPriorPicsFlag se establece igual a 1 (independientemente del valor de no_output_of_prior_pics_flag para la imagen actual u otras imágenes en la AU).

- De otro modo, si el valor de pic_width_in_luma_samples, pic_height_in_luma_samples, o sps_max_dec_pic_buffering_minus1[HighestTid] derivado del SPS activo es diferente del valor de pic_width_in_luma_samples, pic_height_in_luma_samples, o sps_max_dec_pic_buffering_minus1[HighestTid], respectivamente, derivado del SPS activo para la imagen anterior, AuNoOutputOfPriorPicsFlag se puede (pero no se debería) establecer en 1 por el decodificador bajo prueba, independientemente del valor de no_output_of_prior_pics_flag.

NOTA – Aunque es preferible bajo estas condiciones establecer AuNoOutputOfPriorPicsFlag igual a no_output_of_prior_pics_flag, se permite que el decodificador bajo prueba establezca AuNoOutputOfPriorPicsFlag en 1 en este caso.

- De otro modo, AuNoOutputOfPriorPicsFlag se establece en base al valor de no_output_of_prior_pics_flag para la

imagen actual y otras imágenes en la UA de la siguiente manera:

- 5 - AuNoOutputOfPriorPicsFlag se establece igual a 1 si no_output_of_prior_pics_flag es igual a 1 para al menos una imagen en la AU. De otro modo, AuNoOutputOfPriorPicsFlag se establece igual a 0. De esta manera, en este caso, la AuNoOutputOfPriorPicsFlag se establece igual a 0 si no_output_of_prior_pics_flag es igual a 0 para todas las imágenes en una AU.
- En otra realización, AuNoOutputOfPriorPicsFlag se establece en 1 si no_output_of_prior_pics_flag es igual a 1 para la imagen actual. De otro modo, AuNoOutputOfPriorPicsFlag se deja sin cambios.
- En otra realización, AuNoOutputOfPriorPicsFlag se establece igual a 1 si no_output_of_prior_pics_flag es igual a 1 para todas las imágenes en la AU. De otro modo, AuNoOutputOfPriorPicsFlag se establece igual a 0.
- 10 - En otra realización, AuNoOutputOfPriorPicsFlag se establece en 1 si no_output_of_prior_pics_flag es igual a 1 para todas las imágenes que pertenecen a las capas de salida objetivo en la AU. De otro modo, AuNoOutputOfPriorPicsFlag se establece igual a 0.
- En otra realización, AuNoOutputOfPriorPicsFlag se establece igual a 1 si no_output_of_prior_pics_flag es igual a 1 para al menos una imagen que pertenece a las capas de salida objetivo en la AU. De otro modo, AuNoOutputOfPriorPicsFlag se establece igual a 0.

2. El valor de AuNoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica para el HRD, de manera que cuando el valor de AuNoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes correspondientes a todos los valores de nuh_layer_id en el DPB se vacían sin salida de las imágenes que contienen, y la amplitud de DPB para todos los valores nuh_layer_id se establece igual a 0.

- 20 En otra realización, el valor de AuNoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica para el HRD, de manera que cuando el valor de AuNoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes correspondientes al valor nuh_layer_id de la imagen actual en el DPB se vacían sin salida de las imágenes que contienen, y la amplitud de DPB para el valor nuh_layer_id de la imagen actual se establece igual a 0.

25 Listado (1A)

El Listado (2) a continuación proporciona las secciones de JCTVC-L1008 que se cambiarían para adaptarse a los presentes sistemas y métodos.

Listado 2

F.13 Decodificador de referencia hipotético

30 F.13.1 General

Se aplican las especificaciones de la subcláusula C.1.

F.13.2 Operación del almacenador temporal de imagen codificada (CPB)

Se aplican las especificaciones de la subcláusula C.2.

F.13.3 Operación del almacenador temporal de imágenes decodificadas (DPB)

- 35 Las especificaciones en la subcláusula C.3 se aplican por separado para cada conjunto de imágenes decodificadas con un valor particular de nuh_layer_id.

PicOutputFlag para imágenes que no están incluidas en una capa de salida objetivo se establece igual a 0.

Las imágenes decodificadas con el mismo tiempo de salida de DPB y con PicOutputFlag igual a 1 se emiten en orden ascendente de valores nuh_layer_id de estas imágenes decodificadas.

40 F.13.5 Conformidad del decodificador

F.13.5.1 General

Se aplican las especificaciones de la subcláusula C.5.1.

F.13.5.2 Operación de DPB de orden de salida

F.13.5.2.1 General

- 45 El almacenador temporal de imágenes decodificadas contiene almacenadores temporales de almacenamiento de

imágenes. Cada capa consiste en su propio conjunto de almacenadores temporales de almacenamiento de imágenes. De este modo, los almacenadores temporales de almacenamiento de imágenes de cada capa están asociados con el valor nuh_layer_id de la capa. El número de almacenadores temporales de almacenamiento de imágenes para nuh_layer_id igual a 0 se deriva del SPS activo de la capa con nuh_layer_id igual a 0. El número de almacenadores temporales de almacenamiento de imágenes para cada valor nuh_layer_id distinto de cero se deriva del SPS de capa activo para ese valor nuh_layer_id distinto de cero. Cada uno de los almacenadores temporales de almacenamiento de imágenes contiene una imagen decodificada que está marcada como “usada para referencia” o se mantiene para su salida futura. Se invoca el proceso para la salida y eliminación de imágenes del DPB, como se especifica en la subcláusula F.13.5.2.2, seguido por la invocación del proceso para decodificación, marcado, sacudida adicional, y almacenamiento de imágenes como se especifica en la subcláusula F.13.5.2.3. El proceso de “sacudidas” se especifica en la subcláusula F.13.5.2.4 y se invoca como se especifica en las subcláusulas F.13.5.2.2 y F.13.5.2.3.

F.13.5.2.2 Salida y eliminación de imágenes del DPB

La salida y eliminación de imágenes del DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de porción de la primera porción de la imagen actual) ocurre instantáneamente cuando la primera unidad de decodificación de la imagen actual que pertenece a la unidad de acceso que contiene la imagen actual se retira del CPB y procede de la siguiente manera.

Se invoca el proceso de decodificación para RPS como se especifica en la subcláusula 8.3.2.

- Si la imagen actual es una imagen de IRAP con NoRasIOutputFlag igual a 1 y con nuh_layer_id igual a 0 que no es la imagen 0, se aplican los siguientes pasos ordenados:

1. La variable NoOutputOfPriorPicsFlag se deriva para el decodificador bajo prueba de la siguiente manera:

- Si la imagen actual es una imagen de CRA, NoOutputOfPriorPicsFlag se establece en 1 (independientemente del valor de no_output_of_prior_pics_flag).

- De otro modo, si el valor de pic_width_in_luma_samples, pic_height_in_luma_samples, o sps_max_dec_pic_buffering_minus1[HighestTid] derivado del SPS activo que corresponde al valor nuh_layer_id de la imagen 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, derivado del SPS activo para la imagen anterior con el valor nuh_layer_id igual al valor nuh_layer_id de la imagen actual, NoOutputOfPriorPicsFlag se puede (pero no se debería) establecer en 1 por el decodificador bajo prueba, independientemente del valor de no_output_of_prior_pics_flag.

NOTA – Aunque es preferible bajo estas condiciones establecer NoOutputOfPriorPicsFlag igual a no_output_of_prior_pics_flag, se permite que el decodificador bajo prueba establezca NoOutputOfPriorPicsFlag en 1 en este caso.

- De otro modo, NoOutputOfPriorPicsFlag se establece igual a no_output_of_prior_pics_flag.

2. El valor de NoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica para el HRD de la siguiente manera:

- Si NoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes correspondientes al valor nuh_layer_id de la imagen actual en el DPB se vacían sin salida de las imágenes que contienen, y la amplitud de DPB para el valor nuh_layer_id de la imagen actual se establece igual a 0.

En una realización, si NoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes correspondientes a todos los valores nuh_layer_id en el DPB se vacían sin salida de las imágenes que contienen, y la amplitud de DPB para todos los valores nuh_layer_id se establece igual a 0.

En una realización, si NoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes PSB[currLayerId] correspondientes al valor nuh_layer_id currLayerId de la imagen actual en el DPB se vacían sin la salida de las imágenes que contienen, y la amplitud de DPB DPBFullness[currLayerId] se establece igual a 0.

En una realización, si NoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes PSB[nuh_layer_id] para todos los valores nuh_layer_id en el DPB se vacían sin la salida de las imágenes que contienen, y la amplitud de DPB DPBFullness[nuh_layer_id] para todos los valores nuh_layer_id se establece igual a 0.

- De otro modo (NoOutputOfPriorPicsFlag es igual a 0), todos los almacenadores temporales de almacenamiento de imágenes que contienen una imagen que está marcada como “no necesaria para la salida” y “no usada para referencia” se vacían (sin salida), y todos los almacenadores temporales de almacenamiento de imágenes no vacíos en el DPB correspondiente al valor nuh_layer_id de la imagen actual se vacían invocando repetidamente el proceso

de “sacudidas” en la subcláusula F.13.5.2.4, y la amplitud de DPB para el valor `nuh_layer_id` de la imagen actual se establece igual a 0.

En otra realización, cuando `NoOutputOfPriorPicsFlag` es igual a 0, todos los almacenadores temporales de almacenamiento de imágenes correspondientes a todos los valores `nuh_layer_id` que contienen una imagen que está marcada como “no necesaria para la salida” y “no usada para referencia” se vacían (sin salida), y todos los almacenadores temporales de almacenamiento de imágenes no vacíos correspondientes a todos los valores `nuh_layer_id` en el DPB se vacían invocando repetidamente el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4, y la amplitud de DPB para todos los valores `nuh_layer_id` se establece igual a 0.

- En otra realización: De otro modo (`NoOutputOfPriorPicsFlag` es igual a 0), todos los almacenadores temporales de almacenamiento de imágenes que contienen una imagen que está marcada como “no necesaria para la salida” y “no usada para referencia” se vacían (sin salida), y todos los almacenadores temporales de almacenamiento de imágenes no vacíos PSB[`currLayerId`] en el DPB correspondiente al valor `nuh_layer_id currLayerId` de la imagen actual se vacían invocando repetidamente el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4, y la amplitud del DPB `DPBFullness [currLayerId]` para el valor `nuh_layer_id` de la imagen actual se establece igual a 0.

- En otra realización, cuando `NoOutputOfPriorPicsFlag` es igual a 0, todos los almacenadores temporales de almacenamiento de imágenes PSB[`nuh_layer_id`] para todos los valores `nuh_layer_id` en el DPB que contienen una imagen que está marcada como “no necesaria para la salida” y “no usada para referencia” se vacían (sin salida), y todos los almacenadores temporales de almacenamiento de imágenes no vacíos PSB[`nuh_layer_id`] correspondientes a todos los valores `nuh_layer_id` en el DPB se vacían invocando repetidamente el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4, y la amplitud de DPB `DPBFullness(nuh_layer_id)` para todos los valores `nuh_layer_id` se establece igual a 0.

- De otro modo (la imagen actual no es una imagen de IRAP con `NoRasOutputFlag` igual a 1 y con `nuh_layer_id` igual a 0), todas los almacenadores temporales de almacenamiento de imágenes correspondientes al valor `nuh_layer_id` de la imagen actual contienen una imagen que está marcada como “no necesaria para la salida” y “no usada para referencia” se vacían (sin salida). Para cada almacenador temporal de almacenamiento de imágenes que se vacía, la amplitud de DPB correspondiente al valor `nuh_layer_id` de la imagen decodificada actual se disminuye en uno. La variable `currLayerId` se establece igual a `nuh_layer_id` de la imagen decodificada actual y cuando una o más de las siguientes condiciones son ciertas, el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4 se invoca repetidamente mientras que se disminuye aún más la amplitud de DPB en uno para cada almacenador temporal de almacenamiento de imágenes adicional correspondiente al valor `nuh_layer_id` de la imagen decodificada actual que se vacía, hasta que ninguna de las siguientes condiciones es cierta:

- El número de imágenes con `nuh_layer_id` igual a `currLayerId` en el DPB que están marcadas como “necesarias para la salida” es mayor que `sps_max_num_reorder_pics[HighestTid]` del SPS activo (cuando `currLayerId` es igual a 0) o del SPS de capa activo para el valor de `currLayerId`.

- `sps_max_latency_increase_plus1[HighestTid]` del SPS activo (cuando `currLayerId` es igual a 0) o del SPS de capa activo para el valor de `currLayerId` no es igual a 0 y hay al menos una imagen con `nuh_layer_id` igual a `currLayerId` en el DPB que está marcada como “necesaria para la salida” para la cual la variable asociada `PicLatencyCount[currLayerId]` es mayor o igual que `SpsMaxLatencyPictures[HighestTid]` derivada del SPS activo (cuando `currLayerId` es igual a 0) o del SPS de capa activo para el valor de `currLayerId`.

- El número de imágenes con `nuh_layer_id` igual a `currLayerId` en el DPB es mayor o igual que `sps_max_dec_pic_buffering_minus1[HighestTid] + 1` del SPS activo (cuando `currLayerId` es igual a 0) o del SPS de capa activo para el valor de `currLayerId`.

En otra realización: De otro modo (la imagen actual no es una imagen de IRAP con `NoRasOutputFlag` igual a 1 y con `nuh_layer_id` igual a 0), todos los almacenadores temporales de almacenamiento de imágenes correspondientes a todos los valores `nuh_layer_id` que contienen una imagen que está marcada como “no necesaria para la salida” y “no usada como referencia” se vacían (sin salida). Para cada almacenador temporal de almacenamiento de imágenes que se vacía, la amplitud de DPB correspondiente al valor `nuh_layer_id` de la imagen se disminuye en uno.

En otra realización: De otro modo (la imagen actual no es una imagen de IRAP con `NoRasOutputFlag` igual a 1 y con `nuh_layer_id` igual a 0), todos los almacenadores temporales de almacenamiento de imágenes PSB[`currLayerId`] en el DPB correspondiente al valor `nuh_layer_id currLayerId` de la imagen actual que contiene una imagen que está marcada como “no necesaria para la salida” y “no usada para referencia” se vacían (sin salida). Para cada almacenador temporal de almacenamiento de imágenes que se vacía, la amplitud de DPB `DPBFullness [currLayerId]` correspondiente al valor `nuh_layer_id` de la imagen decodificada actual se disminuye en uno.

En otra realización: De otro modo (la imagen actual no es una imagen de IRAP con `NoRasOutputFlag` igual a 1 y con `nuh_layer_id` igual a 0), todos los almacenadores temporales de almacenamiento de imágenes PSB[`nuh_layer_id`] para todos los valores `nuh_layer_id` en el DPB que contienen una imagen que está marcada como “no necesaria para la salida” y “no usada para referencia” se vacían (sin salida). Para cada almacenador

temporal de almacenamiento de imágenes que se vacía, la amplitud de DPB DPBFullness[nuh_layer_id] correspondiente al valor nuh_layer_id de la imagen vaciada se disminuye en uno.

F.13.5.2.3 Decodificación, marcado, sacudida adicional y almacenamiento de imágenes

5 Los procesos especificados en esta subcláusula ocurren instantáneamente cuando la última unidad de decodificación de la unidad de acceso n que contiene la imagen actual se elimina del CPB.

La variable currLayerId se establece igual a nuh_layer_id de la imagen decodificada actual.

Para cada imagen en el DPB que está marcada como “necesaria para la salida” y que tiene un valor nuh_layer_id igual a currLayerId, la variable asociada PicLatencyCount[currLayerId] se establece igual a PicLatencyCount[currLayerId] + 1.

10 La imagen actual se considera como decodificada después de que se decodifique la última unidad de decodificación de la imagen. La imagen decodificada actual se almacena en un almacenador temporal de almacenamiento de imágenes vacío correspondiente a currLayerId (el valor nuh_layer_id de la imagen actual) en el DPB, y se aplica lo siguiente:

15 - Si la imagen decodificada actual tiene PicOutputFlag igual a 1, se marca como “necesaria para la salida” y su variable asociada PicLatencyCount[currLayerId] se establece igual a 0.

- De otro modo (la imagen decodificada actual tiene PicOutputFlag igual a 0), se marca como “no es necesaria para la salida”.

20 En una realización, la imagen decodificada actual se almacena en un almacenador temporal de almacenamiento de imágenes vacío correspondiente a currLayerId (el valor nuh_layer_id de la imagen actual) en el DPB, y se aplica lo siguiente:

- Si la imagen decodificada actual tiene PicOutputFlag igual a 1, se marca como “necesaria para la salida” y su variable asociada PicLatencyCount[currLayerId] se establece igual a 0. Todos los almacenadores temporales de almacenamiento de imágenes que contienen una imagen que tiene el mismo valor de recuento de orden de imágenes (PicOrderCntVal) que la imagen decodificada actual se marcan como “necesarios para la salida”.

25 - De otro modo (la imagen decodificada actual tiene PicOutputFlag igual a 0), se marca como “no necesaria para la salida”.

30 En una realización, la imagen decodificada actual se almacena en un almacenador temporal de almacenamiento de imágenes vacío PSB[currLayerId] correspondiente a currLayerId (el valor nuh_layer_id de la imagen actual) en el DPB, y la amplitud de DPB para el valor nuh_layer_id de la imagen actual DPBFullness[currLayerId] se aumenta en uno, y se aplica lo siguiente

La imagen decodificada actual se marca como “usada para referencia a corto plazo”.

Cuando una o más de las siguientes condiciones son ciertas, el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4 se invoca repetidamente hasta que ninguna de las siguientes condiciones es cierta.

35 - El número de imágenes con nuh_layer_id igual a currLayerId en el DPB que están marcadas como “necesarias para la salida” es mayor que sps_max_num_reorder_pics[HighestTid] del SPS activo (cuando currLayerId es igual a 0) o del SPS de capa activo para el valor de currLayerId.

40 - sps_max_latency_increase_plus1[HighestTid] no es igual a 0 y hay al menos una imagen con nuh_layer_id igual a currLayerId en el DPB que está marcada como “necesaria para la salida” para la cual la variable asociada PicLatencyCount[currLayerId] es mayor o igual que SpsMaxLatencyPictures[HighestTid] derivado del SPS activo (cuando currLayerId es igual a 0) o del SPS de capa activo para el valor de currLayerId.

F.13.5.2.4 Proceso de “sacudidas”

El proceso de “sacudidas” consiste en los siguientes pasos ordenados:

1. Las imágenes que son las primeras para la salida se seleccionan como las que tienen el valor más pequeño de PicOrderCntVal de todas las imágenes en el DPB marcadas como “necesarias para la salida”.

45 2. Estas imágenes se recortan, usando la ventana de recorte de conformidad especificada en el SPS activo para la imagen con nuh_layer_id igual a 0 o en el SPS de capa activo para un valor nuh_layer_id igual al de la imagen, las imágenes recortadas se emiten en orden ascendente de nuh_layer_id, y las imágenes se marcan como “no necesarias para la salida”.

50 3. Cada almacenador temporal de almacenamiento de imágenes que contiene una imagen marcada como “no usada para referencia” y que incluía una de las imágenes que se recortó y emitió se vacía.

Listado (2)

El Listado (2A) a continuación proporciona un planteamiento alternativo a la sección F.13.5.2.2 del Listado (2) de acuerdo para adaptarse a los presentes sistemas y métodos. En algunas configuraciones, el Listado (2A) solamente puede representar cambios a la sección F.13.5.2.2 en JCTVC-L1008. El Listado (2A) puede usar las banderas AuNoOutputOfPriorPicsFlag y AuNoRasIOutputFlag definidas anteriormente.

Listado 2A

F.13.5.2.2 Salida y eliminación de imágenes del DPB

La salida y eliminación de las imágenes del DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de porción de la primera porción de la imagen actual) ocurre instantáneamente cuando la primera unidad de decodificación de la unidad de acceso que contiene la imagen actual se elimina del CPB y procede de la siguiente manera:

Se invoca el proceso de decodificación para RPS como se especifica en la subcláusula 8.3.2.

- Si la imagen actual es una imagen de IRAP con AuNoRasIOutputFlag igual a 1 y con nuh_layer_id igual a 0 que no es la imagen 0, se aplican los siguientes pasos ordenados:

- En otra configuración, si la imagen actual es una imagen de IRAP con NoRasIOutputFlag igual a 1 y con nuh_layer_id igual a 0 que no es la imagen 0, se aplican los siguientes pasos ordenados:

1. La variable AuNoOutputOfPriorPicsFlag se deriva para el decodificador bajo prueba de la siguiente manera.

- Si la imagen actual tiene nuh_layer_id igual a 0 AuNoOutputOfPriorPicsFlag se inicializa a 0. Entonces:

- Si la imagen actual es una imagen de CRA, AuNoOutputOfPriorPicsFlag se establece igual a 1 (independientemente del valor de no_output_of_prior_pics_flag para la imagen actual u otras imágenes en la AU).

- De otro modo, si el valor de pic_width_in_luma_samples, pic_height_in_luma_samples, o sps_max_dec_pic_buffering_minus1[HighestTid] derivado del SPS activo es diferente del valor de pic_width_in_luma_samples, pic_height_in_luma_samples, o sps_max_dec_pic_buffering_minus1[HighestTid], respectivamente, derivado del SPS activo para la imagen anterior, AuNoOutputOfPriorPicsFlag se puede (pero no se debería) establecer en 1 por el decodificador bajo prueba, independientemente del valor de no_output_of_prior_pics_flag.

NOTA – Aunque se prefiere bajo estas condiciones establecer AuNoOutputOfPriorPicsFlag igual a no_output_of_prior_pics_flag, se permite que el decodificador bajo prueba establezca AuNoOutputOfPriorPicsFlag en 1 en este caso.

- De otro modo, AuNoOutputOfPriorPicsFlag se establece en base al valor de no_output_of_prior_pics_flag para la imagen actual y otras imágenes en la UA de la siguiente manera:

- AuNoOutputOfPriorPicsFlag se establece igual a 1 si no_output_of_prior_pics_flag es igual a 1 para al menos una imagen en la AU. De otro modo, AuNoOutputOfPriorPicsFlag se establece igual a 0. De este modo, en este caso, la AuNoOutputOfPriorPicsFlag se establece igual a 0 si no_output_of_prior_pics_flag es igual a 0 para todas las imágenes en una AU.

- En otra realización, AuNoOutputOfPriorPicsFlag se establece igual a 1 si no_output_of_prior_pics_flag es igual a 1 para la imagen actual. De otro modo, AuNoOutputOfPriorPicsFlag se deja sin cambios.

- En otra realización, AuNoOutputOfPriorPicsFlag se establece igual a 1 si no_output_of_prior_pics_flag es igual a 1 para todas las imágenes en la AU. De otro modo, AuNoOutputOfPriorPicsFlag se establece igual a 0.

- En otra realización, AuNoOutputOfPriorPicsFlag se establece en 1 si no_output_of_prior_pics_flag es igual a 1 para todas las imágenes que pertenecen a las capas de salida objetivo en la AU. De otro modo, AuNoOutputOfPriorPicsFlag se establece igual a 0.

- En otra realización, AuNoOutputOfPriorPicsFlag se establece en 1 si no_output_of_prior_pics_flag es igual a 1 para al menos una imagen que pertenece a la capa de salida objetivo en la AU. De otro modo, AuNoOutputOfPriorPicsFlag se establece igual a 0.

2. El valor de AuNoOutputOfPriorPicsFlag derivado del decodificador bajo prueba se aplica para el HRD de la siguiente manera:

- Si AuNoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes en el DPB correspondientes a todos los valores nuh_layer_id se vacían sin salida de las imágenes que incluyen, y la amplitud de DPB se establece igual a 0 para todos los valores nuh_layer_id.

- En otra realización, si AuNoOutputOfPriorPicsFlag es igual a 1, todos los almacenadores temporales de almacenamiento de imágenes en el DPB correspondiente a todos los valores nuh_layer_id de la imagen actual se vacían sin la salida de las imágenes que incluyen, y la amplitud de DPB se establece igual a 0 para el valor nuh_layer_id de la imagen actual.

5 De otro modo (AuNoOutputOfPriorPicsFlag es igual a 0), todos los almacenadores temporales de almacenamiento de imágenes correspondientes a todos los valores nuh_layer_id, que incluyen una imagen que está marcada como “no necesaria para la salida” y “no usada para referencia”, se vacían (sin salida) y todos los almacenadores temporales de almacenamiento de imágenes no vacíos en el DPB se vacían invocando repetidamente el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4, y la amplitud de DPB para todos los valores nuh_layer_id se establece igual a 0.

En otra realización:

15 De otro modo (AuNoOutputOfPriorPicsFlag es igual a 0), todos los almacenadores temporales de almacenamiento de imágenes correspondientes al valor nuh_layer_id de la imagen actual, que incluye una imagen que está marcada como “no necesaria para la salida” y “no usada para referencia” se vacían (sin salida), y todos los almacenadores temporales de almacenamiento de imágenes no vacíos en el DPB se vacían invocando repetidamente el proceso de “sacudidas” especificado en la subcláusula F.13.5.2.4, y la amplitud de DPB para el valor nuh_layer_id de la imagen actual se establece igual a 0.

Listado (2A)

20 En una configuración, para el texto propuesto en el Listado (1), Listado (1A), Listado (2) y Listado (2A) anteriores, todas las apariciones de “imagen actual” se pueden sustituir por “imagen decodificada actual”.

El Listado (3) a continuación proporciona una sección adicional para decodificar para adaptarse a los presentes sistemas y métodos. El Listado (3) proporciona las secciones de JCTVC-L1003 (es decir, la especificación HEVC versión 34) que se cambiarían para adaptarse a los presentes sistemas y métodos.

Listado 3

25 8.3.2. Proceso de decodificación para conjunto de imágenes de referencia

Cuando la imagen actual es una imagen de IRAP con NoRasIOutputFlag igual a 1, todas las imágenes de referencia actualmente en el DPB correspondientes al valor nuh_layer_id de la imagen actual (en su caso) se marcan como “no usadas como referencia”.

30 En otra realización: Cuando la imagen actual es una imagen de IRAP con NoRasIOutputFlag igual a 1, todas las imágenes de referencia actualmente en el DPB correspondientes a todos los valores nuh_layer_id (en su caso) se marcan como “no usadas como referencia”.

Listado (3)

35 Se pueden usar muchas variantes diferentes para definir las operaciones del almacenador temporal de imágenes decodificadas (DPB) 116 híbridas, con varios pasos que están basados en imagen o basados en unidad de acceso. Las Tablas (11) - (15) a continuación enumeran múltiples variantes de los pasos y los ajustes basados en imagen/unidad de acceso (AU) correspondientes.

[Tabla 46]

Paso	Basado en Imagen/Basado en AU
Eliminación	Basado en imagen
Salida	Basado en AU
Almacenamiento	Basado en imagen
Marcado	Basado en AU para marcado de salida, Basado en imagen para marcado de referencia
Salida Adicional	Basado en AU

Tabla (11)

[Tabla 47]

Paso	Basado en imagen/Basado en AU
------	-------------------------------

Eliminación	Basado en imagen
Salida	Basado en AU
Almacenamiento	Basado en AU
Marcado	Basado en AU para marcado de salida, Basado en AU para marcado de referencia
Salida Adicional	Basado en AU

Tabla (12)

[Tabla 48]

Paso	Basado en imagen/Basado en AU
Eliminación	Basado en AU
Salida	Basado en AU
Almacenamiento	Basado en AU
Marcado	Basado en AU para marcado de salida, Basado en AU para marcado de referencia
Salida Adicional	Basado en AU

Tabla (13)

[Tabla 49]

Paso	Basado en imagen/Basado en AU
Eliminación	Basado en imagen
Salida	Basado en AU
Almacenamiento	Basado en imagen
Marcado	Basado en imagen para marcado de salida, Basado en imagen para marcado de referencia
Salida Adicional	Basado en AU

Tabla (14)

5

[Tabla 50]

Paso	Basado en imagen/Basado en AU
Eliminación	Basado en imagen
Salida	Basado en imagen
Almacenamiento	Basado en imagen
Marcado	Basado en imagen para marcado de salida, Basado en imagen para marcado de referencia
Salida Adicional	Basado en imagen

Tabla (15)

La Figura 66 es un diagrama de bloques que ilustra una configuración de un decodificador 2404. El decodificador 2404 se puede incluir en un dispositivo electrónico 2402. Por ejemplo, el decodificador 2404 puede ser un decodificador de codificación de vídeo de alta eficiencia (HEVC). El decodificador 2404 y/o uno o más de los elementos ilustrados que se incluyen en el decodificador 2404 se pueden implementar en hardware, software o una combinación de ambos. El decodificador 2404 puede recibir un flujo de bits 2414 (por ejemplo, una o más imágenes codificadas incluidas en el flujo de bits 2414) para su decodificación. En algunas configuraciones, el flujo de bits

10

2414 recibido puede incluir información de sobrecarga recibida, tal como una cabecera de porción recibida, PPS recibido, información de descripción de almacenador temporal recibida, etc. Las imágenes codificadas incluidas en el flujo de bits 2414 pueden incluir una o más imágenes de referencia codificadas y/o una o más de otras imágenes codificadas.

5 Los símbolos recibidos (en una o más imágenes codificadas incluidas en el flujo de bits 2414) se puede decodificar por entropía mediante un módulo de decodificación de entropía 454, produciendo por ello una señal de información de movimiento 456 y coeficientes cuantificados, escalados y/o transformados 458.

10 La señal de información de movimiento 456 se puede combinar con una parte de una señal de cuadro de referencia 484 de una memoria de cuadros 464 en un módulo de compensación de movimiento 460, que puede producir una señal de predicción inter cuadros 468. Los coeficientes cuantificados, desescalados y/o transformados 458 se pueden cuantificar inversamente, escalar y transformar inversamente por un módulo de inversión 462, produciendo por ello una señal residual decodificada 470. La señal residual decodificada 470 se puede añadir a una señal de predicción 478 para producir una señal combinada 472. La señal de predicción 478 puede ser una señal seleccionada de o bien la señal de predicción inter cuadros 468 o bien una señal de predicción intra cuadros 476 producida por un módulo de predicción intra cuadros 474. En algunas configuraciones, esta selección de señal se puede basar en (por ejemplo, controlar por) el flujo de bits 2414.

15 La señal de predicción dentro de una imagen 476 se puede predecir a partir de la formación decodificada previamente a partir de la señal combinada 472 (en el cuadro actual, por ejemplo). La señal combinada 472 también se puede filtrar mediante un filtro de desbloqueo 480. La señal filtrada resultante 482 se puede escribir en la memoria de cuadros 464. La señal filtrada resultante 482 puede incluir una imagen decodificada.

20 La memoria de cuadros 464 puede incluir un almacenador temporal de imágenes decodificadas (DPB) 2416 como se describe en la presente memoria. El almacenador temporal de imágenes decodificadas (DPB) 2416 puede ser capaz de operaciones de almacenador temporal de imágenes decodificadas (DPB) 116 híbridas. El almacenador temporal de imágenes decodificadas (DPB) 2416 puede incluir una o más imágenes decodificadas que se pueden mantener como cuadros de referencia a corto o largo plazo. La memoria de cuadros 464 también puede incluir información de sobrecarga correspondiente a las imágenes decodificadas. Por ejemplo, la memoria de cuadros 464 puede incluir cabeceras de porción, información de conjunto de parámetros de vídeo (VPS), información de conjunto de parámetros de secuencia (SPS), información de conjunto de parámetros de imagen (PPS), parámetros de ciclo, información de descripción de almacenador temporal, etc. Una o más de estas partes de información se pueden señalar desde un codificador (por ejemplo, el codificador 2108, el módulo de señalización de sobrecarga 2112).

25 La Figura 67A es un diagrama de bloques que ilustra el uso tanto de una capa de mejora como de una capa base para codificación de vídeo con almacenadores temporales de imágenes decodificadas (DPB) 516a-b separados y módulos de operación de almacenadores temporales de imágenes decodificadas (DPB) 520a-b híbrida separada para la capa base y la capa de mejora. Se ilustran un primer dispositivo electrónico 502a y un segundo dispositivo electrónico 502b. El primer dispositivo electrónico 502a puede incluir un codificador de vídeo 508 que incluye un codificador de capa de mejora 526 y un codificador de capa base 528. Cada uno de los elementos incluidos dentro del primer dispositivo electrónico 502a (es decir, el codificador de la capa de mejora 526 y el codificador de la capa base 528) se puede implementar en hardware, software o una combinación de ambos. El primer dispositivo electrónico 502a puede obtener una imagen de entrada 2506. En algunas configuraciones, la imagen de entrada 2506 se puede capturar en el primer dispositivo electrónico 502a usando un sensor de imagen, recuperado de la memoria o recibido desde otro dispositivo electrónico 502.

30 El codificador de capa de mejora 526 puede codificar la imagen de entrada 2506 para producir datos codificados. Por ejemplo, el codificador de capa de mejora 526 puede codificar una serie de imágenes de entrada 2506 (por ejemplo, vídeo). Los datos codificados se pueden incluir en un flujo de bits de vídeo de capa de mejora codificado 530. El codificador de capa de mejora 526 puede generar una señalización de sobrecarga en base a la imagen de entrada 2506.

35 El decodificador de vídeo de capa de mejora 534 puede incluir un almacenador temporal de imágenes decodificadas (DPB) 516a y un módulo de operación de almacenador temporal de imágenes decodificadas (DPB) 520a híbrida. Del mismo modo, el decodificador de capa base 536 puede incluir un almacenador temporal de imágenes decodificadas (DPB) 516b y un módulo de operación de almacenador temporal de imágenes decodificadas (DPB) 520b híbrida.

40 El codificador de capa base 528 también puede codificar la imagen de entrada 2506. En una configuración, la misma imagen de entrada 2506 usada por el codificador de capa de mejora 526 también se puede usar por el codificador de capa base 528. En otra configuración, una imagen de entrada diferente (pero similar) que la imagen de entrada 2506 usada por el codificador de capa de mejora 526 se puede usar por el codificador de la capa base 528. Por ejemplo, para la escalabilidad de la relación señal a ruido (SNR) (también conocida como escalabilidad de calidad), la misma imagen de entrada 2506 se puede usar tanto por el codificador de capa de mejora 526 como por el codificador de capa base 528. Como otro ejemplo, para escalabilidad espacial, una imagen submuestreada se puede usar por el codificador de capa base 528. En otro ejemplo más, para escalabilidad de vistas múltiples, una imagen de vista diferente se puede usar por el codificador de capa base 528. El codificador de capa base 528 puede

producir datos codificados incluidos en un flujo de bits de vídeo de capa base codificado 532.

El flujo de bits de vídeo de capa de mejora codificado 530 y el flujo de bits de vídeo de capa base codificado 532 pueden incluir cada uno datos codificados basados en la imagen de entrada 2506. En un ejemplo, el flujo de bits de vídeo de capa de mejora codificado 530 y el flujo de bits de vídeo de capa base codificado 532 pueden incluir datos de imagen codificados. En algunas configuraciones, el flujo de bits de vídeo de capa de mejora codificado 530 y/o el flujo de bits de vídeo de capa base codificado 532 también pueden incluir datos de sobrecarga, tales como información del conjunto de parámetros de secuencia (SPS), información del conjunto de parámetros de imagen (PPS), información del conjunto de parámetros de vídeo (VPS), información de cabecera de porción, etc.

El flujo de bits de vídeo de capa de mejora codificado 530 se puede proporcionar al segundo dispositivo electrónico 502b. Del mismo modo, el flujo de bits de vídeo de capa base codificado 532 se puede proporcionar al segundo dispositivo electrónico 502b. El segundo dispositivo electrónico 502b puede incluir un decodificador de vídeo 2504. El decodificador de vídeo 2504 puede incluir un decodificador de capa de mejora 534 y un decodificador de capa base 536. En una configuración, el flujo de bits de vídeo de capa base codificado 530 se decodifica mediante el decodificador de capa base 536 mientras que el flujo de bits de vídeo de capa de mejora codificado 530 se decodifica mediante el decodificador de capa de mejora 534.

En un ejemplo, el flujo de bits de vídeo de capa de mejora codificado 530 y el flujo de bits de vídeo de capa base codificado 532 se pueden transmitir al segundo dispositivo electrónico 502b usando un enlace cableado o inalámbrico. En algunos casos, esto se puede hacer sobre una red, tal como Internet, una Red de Área Local (LAN) u otro tipo de red para la comunicación entre dispositivos. Se debería observar que en algunas configuraciones, los codificadores (es decir, el codificador de capa de mejora 526 y el codificador de capa base 528) y el decodificador 2504 (por ejemplo, el decodificador de capa base 536 y el decodificador de capa de mejora 534) se pueden implementar en el mismo dispositivo electrónico 502 (es decir, el primer dispositivo electrónico 502a y el segundo dispositivo electrónico 502b pueden ser parte de un único dispositivo electrónico 502). En una implementación donde los codificadores y decodificadores se implementan en el mismo dispositivo electrónico 502, por ejemplo, el flujo de bits de vídeo de capa de mejora codificado 530 y el flujo de bits de vídeo de capa base codificado 532 pueden estar disponibles para el decodificador de vídeo 2504 en una variedad de formas. Por ejemplo, el flujo de bits de vídeo de capa de mejora codificado 530 y el flujo de bits de vídeo de capa base codificado 532 se pueden proporcionar sobre un bus al decodificador de vídeo 2504 o almacenar en memoria para su recuperación por el decodificador de vídeo 2504.

El decodificador de vídeo 2504 puede generar una o más imágenes decodificadas basadas en el flujo de bits de vídeo de capa de mejora codificado 530 y el flujo de bits de vídeo de capa base codificado 532. Una imagen decodificada 2118 (que puede incluir una imagen decodificada de capa de mejora 538 y una imagen decodificada de la capa base 540) se puede mostrar, reproducir, almacenar en memoria y/o transmitir a otro dispositivo, etc.

En un ejemplo, la imagen decodificada 2118 se puede transmitir a otro dispositivo o de vuelta al primer dispositivo electrónico 502a. La imagen decodificada 2118 también se puede almacenar o mantener de otro modo en el segundo dispositivo electrónico 502b. En otro ejemplo, el segundo dispositivo electrónico 502b puede mostrar la imagen decodificada 2118. En otras configuraciones, la imagen decodificada 2118 incluye elementos de la imagen de entrada 2506 con diferentes propiedades en base a la codificación y otras operaciones realizadas en el flujo de bits 2114. En algunas configuraciones, la imagen decodificada 2118 se puede incluir en una secuencia de imágenes con una resolución, formato, especificaciones u otro atributo diferentes de la imagen de entrada 2506.

La Figura 67B es un diagrama de bloques que ilustra el uso de un almacenador temporal de imágenes decodificadas (DPB) 516c compartido y un módulo de operación de almacenador temporal de imágenes decodificadas (DPB) 520c híbrida compartida para la capa base y la capa de mejora. La Figura 67B incluye los mismos componentes que la Figura 67A, excepto que el decodificador de vídeo de capa de mejora 534 y el decodificador de capa base 536 comparten ambos un almacenador temporal de imágenes decodificadas (DPB) 516c y un módulo de operación de almacenador temporal de imágenes decodificadas (DPB) 520c híbrida.

La Figura 68 es un diagrama de tiempo que ilustra la operación del almacenador temporal de imágenes decodificadas (DPB) 116 híbrida. Las operaciones del almacenador temporal de imágenes decodificadas (DPB) 116 híbridas de la Figura 68 muestran los pasos de una variante preferida donde la eliminación, el almacenamiento y el marcado de referencia se basan en imágenes y la salida, el marcado de salida y la salida adicional se basan en unidades de acceso (AU). Un almacenador temporal de imágenes decodificadas (DPB) 116 ideal y un decodificador de referencia hipotético (HRD) pueden operar de tal manera que los diversos pasos individuales mostrados (por ejemplo, eliminación 621a-b, salida 623, almacenamiento 625a-c, marcado 627a-c, 629 y salida adicional 631) se realizan todos instantáneamente. La secuencia de estos pasos y los desplazamientos de temporización se ilustran entre pasos individuales con propósitos de ilustración. Para conformidad del decodificador de temporización de salida, la temporización (relativa al tiempo de entrega del primer bit) de una salida de imagen es la misma tanto para el decodificador de referencia hipotético (HRD) como para el decodificador bajo prueba (DUT) hasta un retardo fijo. Por tanto, el desplazamiento de temporización ilustrado podría ocurrir para un DUT que introduce un retardo fijo en comparación con un HRD.

Se ilustran los pasos para un almacenador temporal de imágenes decodificadas (DPB) 116 de la primera capa de mejora (EL1) 2642a, un almacenador temporal de imágenes decodificadas (DPB) 116 de la segunda capa de mejora (EL2) 2642b y un almacenador temporal de imágenes decodificadas (DPB) 116 de la capa base (BL) 2644. Se ilustra un tiempo de eliminación del almacenador temporal de imágenes codificadas (CPB) 2646. Después del tiempo de eliminación del almacenador temporal de imágenes codificadas (CPB) 2646, una eliminación basada en imagen (sin salida) 621a se puede realizar por el almacenador temporal de imágenes decodificadas (DPB) 116 de la segunda capa de mejora (EL2) 2642b y una eliminación basada en imagen (sin salida) 621b se puede realizar por el almacenador temporal de imágenes decodificadas (DPB) 116 de la capa base (BL) 2644. Después de un desplazamiento de temporización, una salida (sacudida) de imagen basada en unidad de acceso (AU) 623 se puede realizar por el almacenador temporal de imágenes decodificadas (DPB) 116 de la capa base (BL) 2644, el almacenador temporal de imágenes decodificadas (DPB) 116 de la primera capa de mejora (EL1) 2642a y el almacenador temporal de imágenes decodificadas (DPB) 116 de la segunda capa de mejora (EL2) 2642b.

Después de otro desplazamiento de temporización, se ilustra el proceso relacionado con la imagen decodificada actual 613. Un paso de almacenamiento basado en imagen 625a-c se puede realizar por el almacenador temporal de imágenes decodificadas (DPB) 116 de la capa base (BL) 2644, el almacenador temporal de imágenes decodificadas (DPB) 116 de la primera capa de mejora (EL1) 2642a y el almacenador temporal de imágenes decodificadas (DPB) 116 de la segunda capa de mejora (EL2) 2642b. El paso de almacenamiento se puede realizar después de que se decodifiquen cada una de la capa base (BL) 2644, la primera capa de mejora (EL1) 2642a y la segunda capa de mejora (EL2) 2642b. El paso de almacenamiento 625 se puede subdividir aún más. En los pasos de almacenamiento 625, una imagen decodificada se almacena en el almacenador temporal de imágenes decodificadas (DPB) 116 en un almacenador temporal de almacenamiento vacío y la amplitud del almacenador temporal de imágenes decodificadas (DPB) se aumenta en uno. También, cuando se elimina una imagen (sin salida) del almacenador temporal de imágenes decodificadas (DPB) 116, la amplitud del almacenador temporal de imágenes decodificadas (DPB) se disminuye en uno. De manera similar, cuando se sacude una imagen desde el almacenador temporal de imágenes decodificadas (DPB) 116 (durante o bien una sacudida o bien una sacudida adicional), la amplitud del almacenador temporal de imágenes decodificadas (DPB) se disminuye en uno.

El almacenador temporal de imágenes decodificadas (DPB) 116 puede incluir almacenadores temporales de imágenes identificadas y gestionadas por separado para imágenes decodificadas que tienen características diferentes. Por ejemplo, el almacenador temporal de imágenes decodificadas (DPB) 116 puede incluir almacenadores temporales de imágenes identificadas y gestionadas por separado para imágenes decodificadas con diferentes resoluciones, diferentes profundidades de bits y/o diferente cromaticidad de color.

Una imagen decodificada se puede almacenar en su lugar en un grupo común de almacenadores temporales de almacenamiento de imágenes en el almacenador temporal de imágenes decodificadas (DPB) 116. Por ejemplo, se pueden usar dos subcasos adicionales para determinar las restricciones de tamaño del almacenador temporal de imágenes decodificadas (DPB) 116 que afectan al proceso de sacudidas/eliminación y a las definiciones de nivel. En una restricción del almacenador temporal de imágenes decodificadas (DPB) 116 basada en bytes, una imagen decodificada se puede almacenar en consideración del tamaño en base a la resolución y/o la profundidad de bits. Las restricciones de tamaño del almacenador temporal de imágenes decodificadas (DPB) 116 se pueden definir como un límite de bytes que considera la resolución y la profundidad de bits de cada imagen decodificada. En una restricción del almacenador temporal de imágenes decodificadas (DPB) 116 basada en unidad de imagen, se puede almacenar una imagen decodificada (y se considera que toma una ranura del almacenador temporal de imágenes). Las restricciones de tamaño del almacenador temporal de imágenes decodificadas (DPB) 116 se pueden definir entonces como un límite del número de ranuras de imagen sin considerar la resolución y la profundidad de bits de cada imagen decodificada.

En una configuración, la amplitud del almacenador temporal de imágenes decodificadas (DPB) se puede rastrear por capa. Por ejemplo, las restricciones de tamaño del almacenador temporal de imágenes decodificadas (DPB) 116 se pueden señalar, y se pueden aplicar sacudidas, por capa. Donde cada capa incluye sus propios almacenadores temporales de almacenamiento de imágenes, se podría usar una variable `DPBFullness[nuh_layer_id]` para hacer el seguimiento de la amplitud del almacenador temporal de imágenes decodificadas (DPB) de cada capa. Cuando se elimina una imagen de una capa con un valor de ID de capa igual a `nuh_layer_id`, la variable `DPBFullness[nuh_layer_id]` se puede establecer igual a `DPBFullness [nuh_layer_id] -1` (es decir, `DPBFullness[nuh_layer_id]` se puede disminuir en uno). En este caso, la imagen se eliminó de un almacenador temporal de almacenamiento de imágenes `PSB[nuh_layer_id]`.

De manera similar, cuando una imagen decodificada actualmente con un valor de ID de capa igual a `nuh_layer_id` se almacena en el almacenador temporal de imágenes decodificadas (DPB) 116, la variable `DPBFullness[nuh_layer_id]` se establece igual a `DPBFullness[nuh_layer_id] + 1` (es decir, `DPBFullness[nuh_layer_id]` se aumenta en uno). En este caso, la imagen se almacenó en un almacenador temporal de almacenamiento de imágenes `PSB[nuh_layer_id]`.

La amplitud del almacenador temporal de imágenes decodificadas (DPB) también se podría hacer un seguimiento para un conjunto de capas de salida. Las restricciones de tamaño del almacenador temporal de imágenes decodificadas (DPB) 116 se pueden señalar, y se pueden aplicar sacudidas, en base a las restricciones

especificadas para un conjunto de capas de salida. Se podría hacer un seguimiento de un valor de DPBFullness para el conjunto de capas de salida que está asociado con el punto de operación bajo prueba. De este modo, cuando se elimina una imagen de una capa que pertenece al conjunto de capas de salida, el valor de la amplitud del almacenador temporal de imágenes decodificadas (DPB) se puede disminuir en uno como $DPBFullness = DPBFullness - 1$. Del mismo modo, cuando una imagen decodificada actualmente se almacena en el almacenador temporal de imágenes decodificadas (DPB) 116, la amplitud del almacenador temporal de imágenes decodificadas (DPB) se puede disminuir en uno como $DPBFullness = DPBFullness + 1$.

Dentro del proceso relacionado con la imagen decodificada actual 613, se ilustra el proceso de marcado 611. El proceso de marcado 611 puede incluir el marcado de referencia 627a-c y el marcado de salida 629. Después de un desplazamiento de temporización de los pasos de almacenamiento 625a-c, se puede realizar un paso de marcado de referencia basado en imagen 627a-c por el almacenador temporal de imágenes decodificadas (DPB) 116 de la capa base (BL) 2644, el almacenador temporal de imágenes decodificadas (DPB) 116 de la primera capa de mejora (EL1) 2642a y el almacenador temporal de imágenes decodificadas (DPB) 116 de la segunda capa de mejora (EL2) 2642b. Después de otro desplazamiento de tiempo, se puede realizar un paso de marcado de salida basado en unidad de acceso (AU) 629 por el almacenador temporal de imágenes decodificadas (DPB) 116 de la capa base (BL) 2644, el almacenador temporal de imágenes decodificadas (DPB) 116 de la primera capa de mejora (EL1) 2642a y el almacenador temporal de imágenes decodificadas (DPB) 116 de la segunda capa de mejora (EL2) 2642b. Una vez que se completa el proceso relacionado con la imagen decodificada actual 613, se puede realizar un paso 631 de salida (sacudida adicional) de imágenes basada en unidad de acceso (AU) por el almacenador temporal de imágenes decodificadas (DPB) 116 de la capa base (BL) 2644, el almacenador temporal de imágenes decodificadas (DPB) 116 de la primera capa de mejora (EL1) 2642a y el almacenador temporal de imágenes decodificadas (DPB) 116 de la segunda capa de mejora (EL2) 2642b.

La Figura 69 es un diagrama de bloques que ilustra la estructura y temporización para unidades de capa de abstracción de red (NAL) de capas para imágenes codificadas y unidades de acceso (AU) cuando la segunda capa de mejora (EL2) 942b tiene una tasa de imágenes más baja que la capa base (BL) 944 y la primera capa de mejora (EL1) 942a. Las unidades de NAL de una imagen codificada de EL1 953a se ilustran a lo largo de la primera capa de mejora (EL1) 942a. Las unidades de NAL de una imagen codificada de EL2 953b se ilustran a lo largo de la segunda capa de mejora (EL2) 942b. Las unidades de NAL de una imagen codificada de capa base 953c se ilustran a lo largo de la capa base (BL) 944.

En el tiempo t_1 , las unidades de NAL de una imagen codificada de EL1 953a, las unidades de NAL de una imagen codificada de EL2 953b y las unidades de NAL de una imagen codificada de capa base 953c son parte de la unidad de acceso (AU) 955a. En el tiempo t_2 , las unidades de NAL de una imagen codificada de EL1 953a y las unidades de NAL de una imagen codificada de capa base 953c son parte de la unidad de acceso (AU) 955b. En el tiempo t_3 , las unidades de NAL de una imagen codificada EL1 953a, las unidades de NAL de una imagen codificada de EL2 953b y las unidades de NAL de una imagen codificada de capa base 953c son parte de la unidad de acceso (AU) 955c. En el tiempo t_4 , las unidades de NAL de una imagen codificada de EL1 953a y las unidades de NAL de una imagen codificada de capa base 953c son parte de la unidad de acceso (AU) 955d.

La Figura 70 es un diagrama de bloques que ilustra la estructura y temporización para unidades de capa de abstracción de red (NAL) de capas para imágenes codificadas y unidades de acceso (AU) cuando la capa base (BL) 1044 tiene una tasa de imagen más baja que la primera capa de mejora (EL1) 1042a y la segunda capa de mejora (EL2) 1042b. Las unidades de NAL de una imagen codificada de EL1 1053a se ilustran a lo largo de la primera capa de mejora (EL1) 1042a. Las unidades de NAL de una imagen codificada de EL2 1053b se ilustran a lo largo de la segunda capa de mejora (EL2) 1042b. Las unidades de NAL de una imagen codificada de capa base 1053c se ilustran a lo largo de la capa base (BL) 1044.

En el tiempo t_1 , las unidades de NAL de una imagen codificada de EL1 1053a, las unidades de NAL de una imagen codificada de EL2 1053b y las unidades de NAL de una imagen codificada de capa base 1053c son parte de la unidad de acceso (AU) 1055a. En el tiempo t_2 , las unidades de NAL de una imagen codificada de EL1 1053a y las unidades de NAL de una imagen codificada de EL2 1053b son parte de la unidad de acceso (AU) 1055b. En el tiempo t_3 , las unidades de NAL de una imagen codificada de EL1 1053a, las unidades de NAL de una imagen codificada de EL2 1053b y las unidades de NAL de una imagen codificada de capa base 1053c son parte de la unidad de acceso (AU) 1055c. En el tiempo t_4 , las unidades de NAL de una imagen codificada EL1 1053a y las unidades de NAL de una imagen codificada de EL1 1053b son parte de la unidad de acceso (AU) 1055d.

El término "medio legible por ordenador" se refiere a cualquier medio disponible que se puede acceder por un ordenador o un procesador. El término "medio legible por ordenador", como se usa en la presente memoria, puede denotar un medio legible por ordenador y/o por procesador que es tangible y no transitorio. A modo de ejemplo, y no de limitación, un medio legible por ordenador o legible por procesador puede comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que se puede usar para transportar o almacenar el código de programa deseado en forma de instrucciones o estructuras de datos, y se puede acceder a él por un ordenador o un procesador. Disco, como se usa en la presente memoria, incluye discos compactos (CD), discos láser, discos ópticos, discos versátiles digitales (DVD), disquetes y discos Blu-ray (marca registrada) donde los discos

normalmente reproducen datos magnéticamente, mientras que los discos reproducen datos ópticamente con láseres.

5 Se debería observar que uno o más de los métodos descritos en la presente memoria se pueden implementar en y/o realizar usando hardware. Por ejemplo, uno o más de los métodos o planteamientos descritos en la presente memoria se pueden implementar en y/o realizar usando un conjunto de chips, un ASIC, un LSI o un circuito integrado, etc.

REIVINDICACIONES

1. Un método para decodificar un flujo de bits de vídeo que comprende:
 - (a) recibir un flujo de bits base representativo de una secuencia de vídeo codificada;
 - (b) recibir una pluralidad de flujos de bits de mejora representativos de dicha secuencia de vídeo codificada;
 - 5 (c) recibir un conjunto de parámetros de vídeo que contiene elementos de sintaxis que se aplican a dicho flujo de bits base y dicha pluralidad de flujos de bits de mejora, en donde dicho conjunto de parámetros de vídeo contiene un elemento de sintaxis que señala una extensión del conjunto de parámetros de vídeo;
 - (d) recibir dicha extensión de conjunto de parámetros de vídeo que contiene elementos de sintaxis relacionados con al menos una de dichos flujos de bits de mejora,
 - 10 en donde los elementos de sintaxis comprenden un primer elemento de sintaxis que indica un número de conjuntos de capas de salida especificados para el flujo de bits de vídeo;
 - (e) recibir elementos de sintaxis que indican un cambio en al menos un conjunto de capas de salida, caracterizado por que:
 - 15 los elementos de sintaxis que indican un cambio en al menos un conjunto de capas de salida comprenden un segundo elemento de sintaxis que indica un número de conjuntos de capas adicionales para los cuales se especifican capas de salida además del número de conjuntos de capas especificados para el flujo de bits de vídeo indicado por el primer elemento de sintaxis para el cual se especifican capas de salida.
2. El método de la reivindicación 1, en donde los elementos de sintaxis que indican un cambio en al menos un conjunto de capas de salida se incluyen en la extensión del conjunto de parámetros de vídeo.
- 20 3. El método de la reivindicación 1, en donde los elementos de sintaxis que indican un cambio en al menos un conjunto de capas de salida se incluyen en un mensaje de cambio de conjunto de capas de salida.
4. El método de la reivindicación 3, en donde dicho mensaje de cambio de conjunto de capas de salida es un mensaje de información de mejora (SEI) suplementario.
- 25 5. El método de la reivindicación 3, en donde dicho mensaje de cambio de conjunto de capas de salida incluye una sintaxis que elimina uno de dichos al menos un conjunto de capas de salida.
6. El método de la reivindicación 3, en donde dicho mensaje de cambio de conjunto de capas de salida incluye una sintaxis que añade un conjunto de capas de salida a dicho al menos un conjunto de capas de salida.
7. El método de la reivindicación 3, en donde dicho mensaje de cambio de conjunto de capas de salida incluye una sintaxis que cambia un conjunto de capas de salida de dicho al menos un conjunto de capas de salida.
- 30 8. El método de la reivindicación 5, en donde dicho mensaje de cambio de conjunto de capas de salida incluye una sintaxis que añade un conjunto de capas de salida a dicho al menos un conjunto de capas de salida.
9. El método de la reivindicación 8, en donde dicho mensaje de cambio de conjunto de capas de salida incluye una sintaxis que cambia un conjunto de capas de salida de dicho al menos un conjunto de capas de salida.
- 35 10. El método de la reivindicación 6, en donde dicho mensaje de cambio de conjunto de capas de salida incluye una sintaxis que cambia un conjunto de capas de salida de dicho al menos un conjunto de capas de salida.
11. El método de la reivindicación 7, en donde dicho mensaje de cambio de conjunto de capas de salida incluye una sintaxis que elimina uno de dicho al menos un conjunto de capas de salida.
- 40 12. El método de la reivindicación 1, en donde los elementos de sintaxis que indican un cambio en al menos un conjunto de capas de salida comprenden un tercer elemento de sintaxis que identifica los conjuntos de capas adicionales y una bandera que indica si una capa en particular es una capa de salida objetivo.

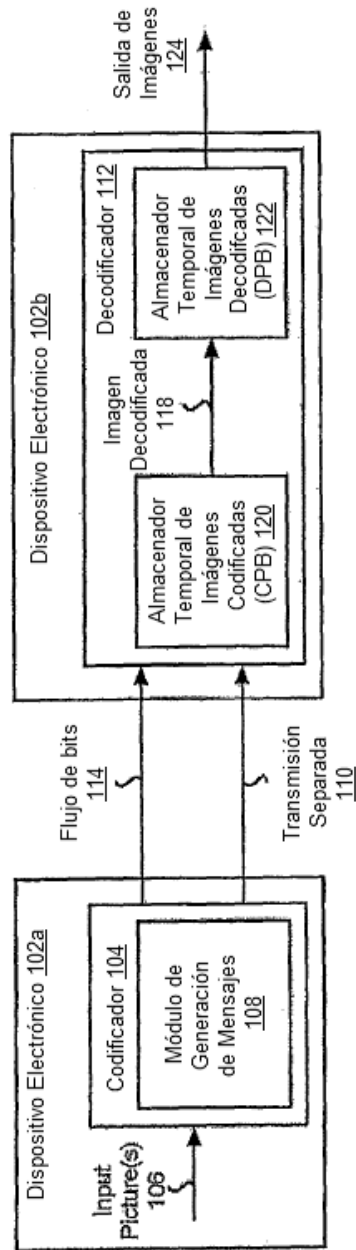


FIG. 1A

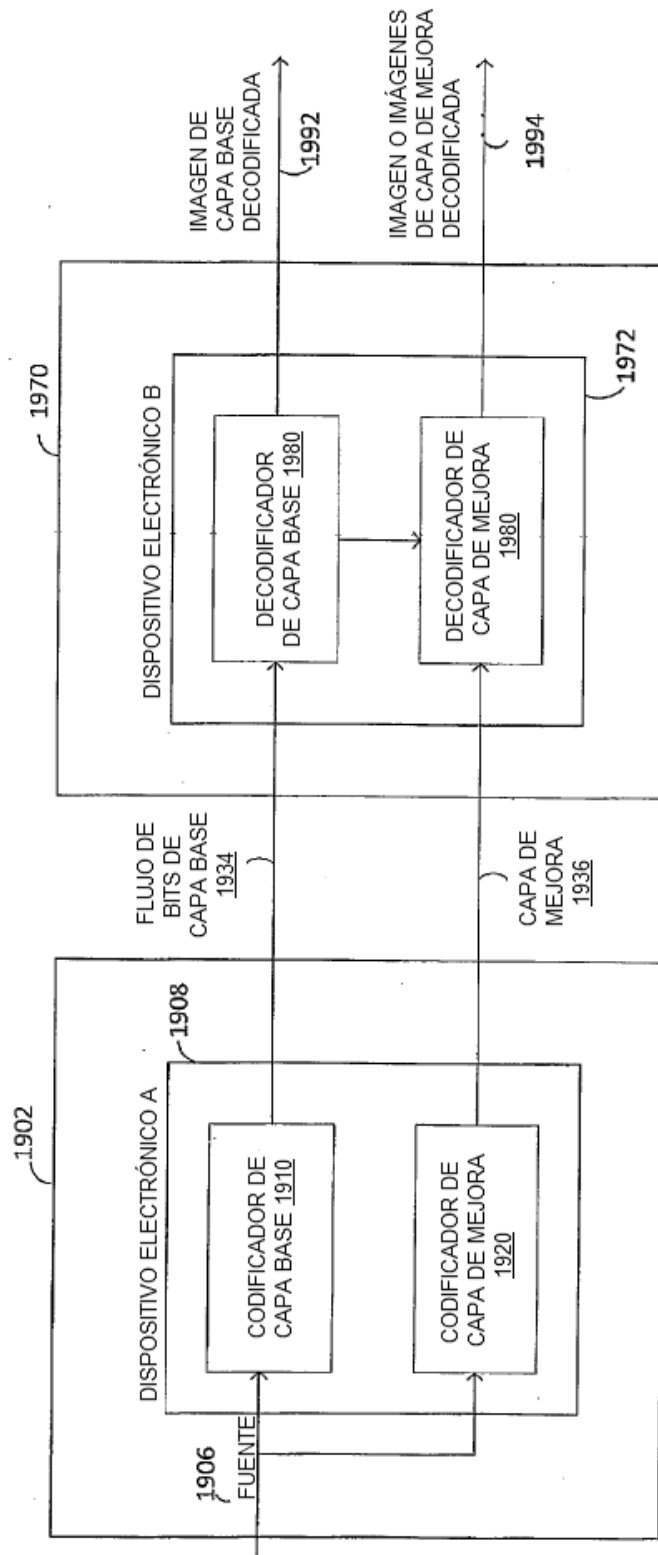


FIG. 1B

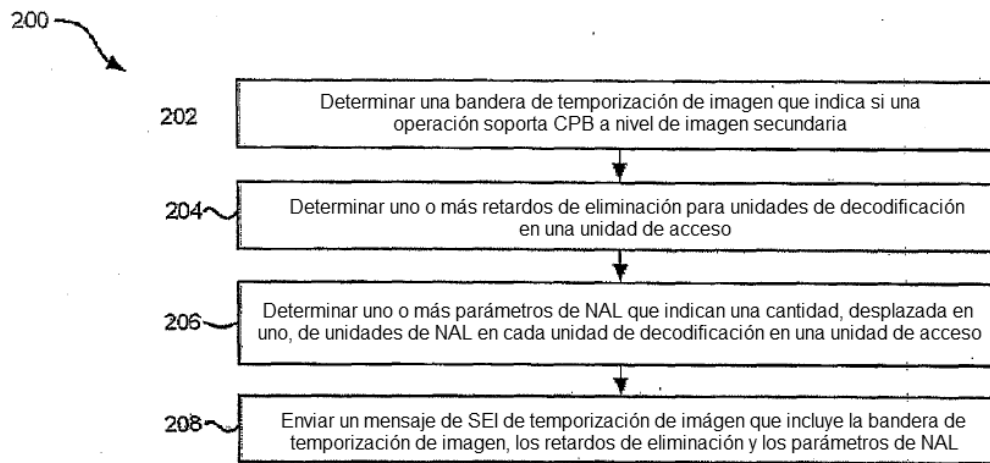


FIG. 2

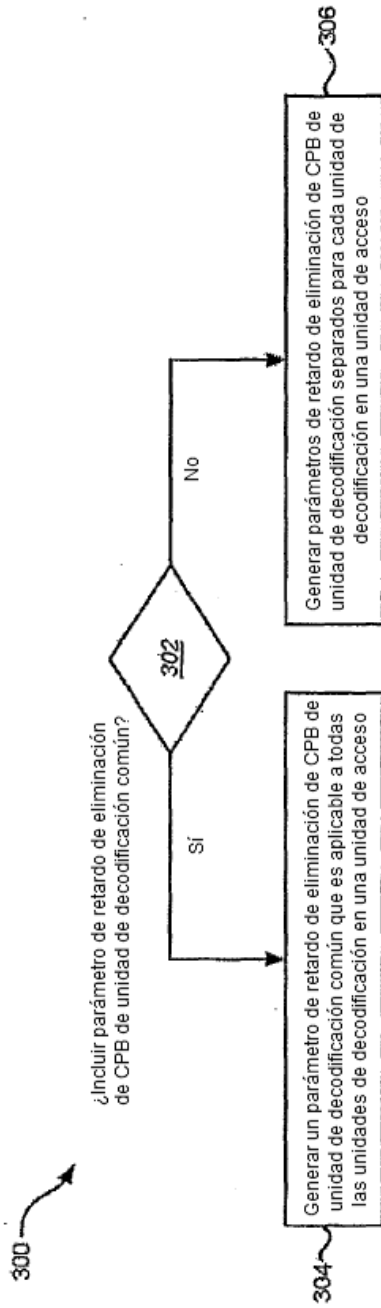


FIG. 3

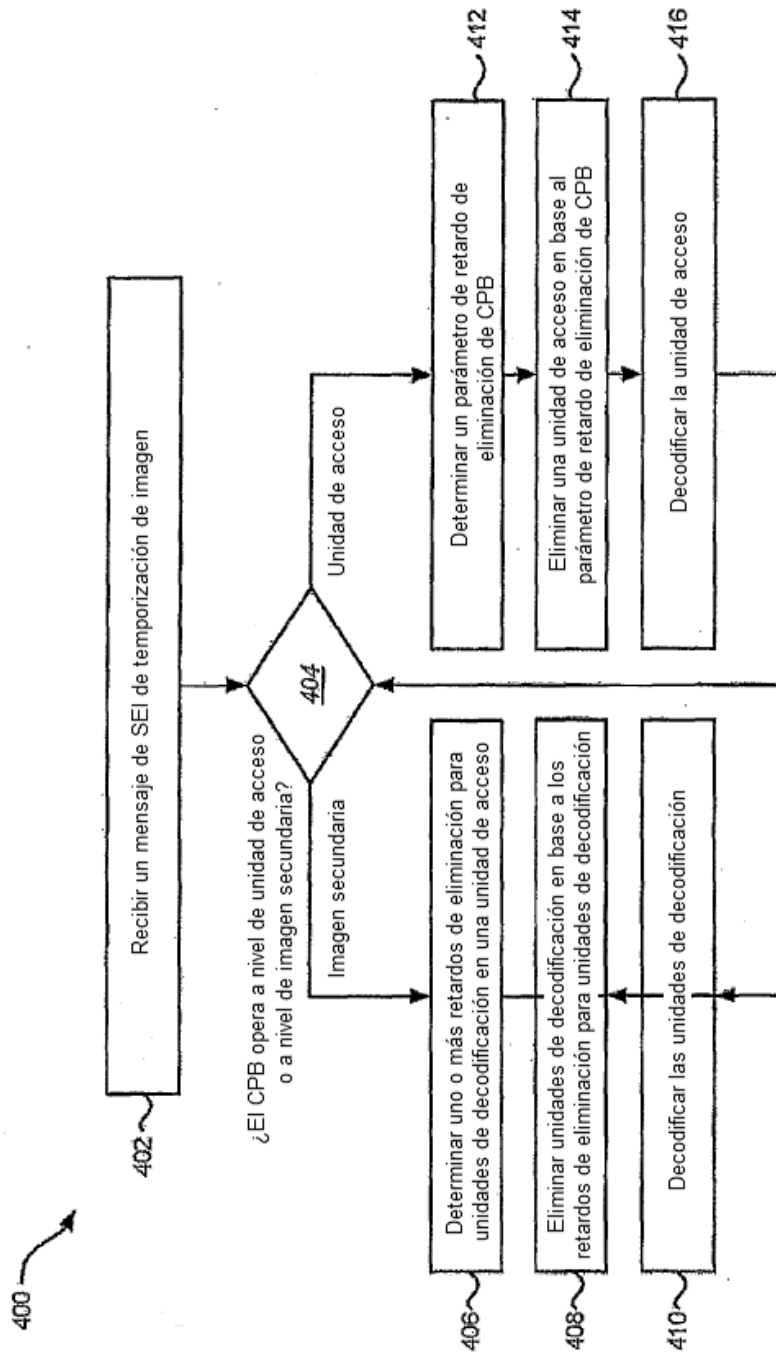


FIG. 4

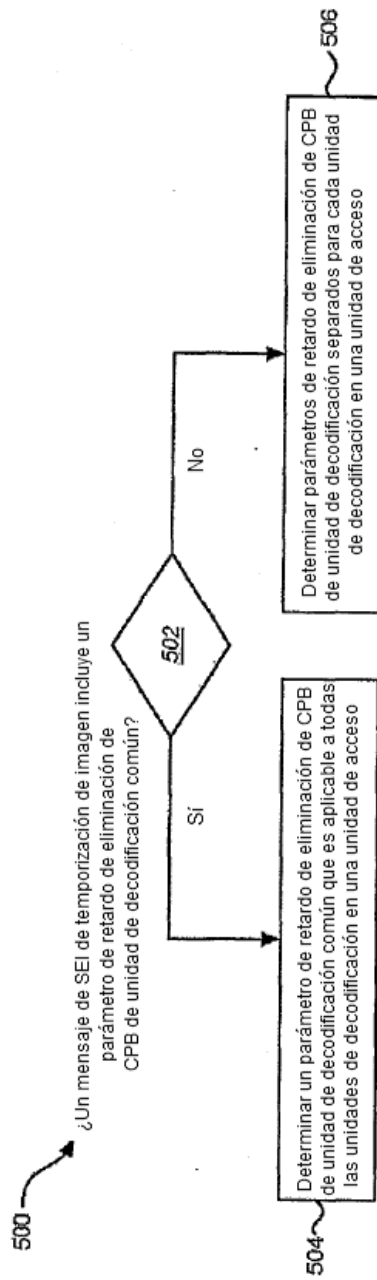


FIG. 5

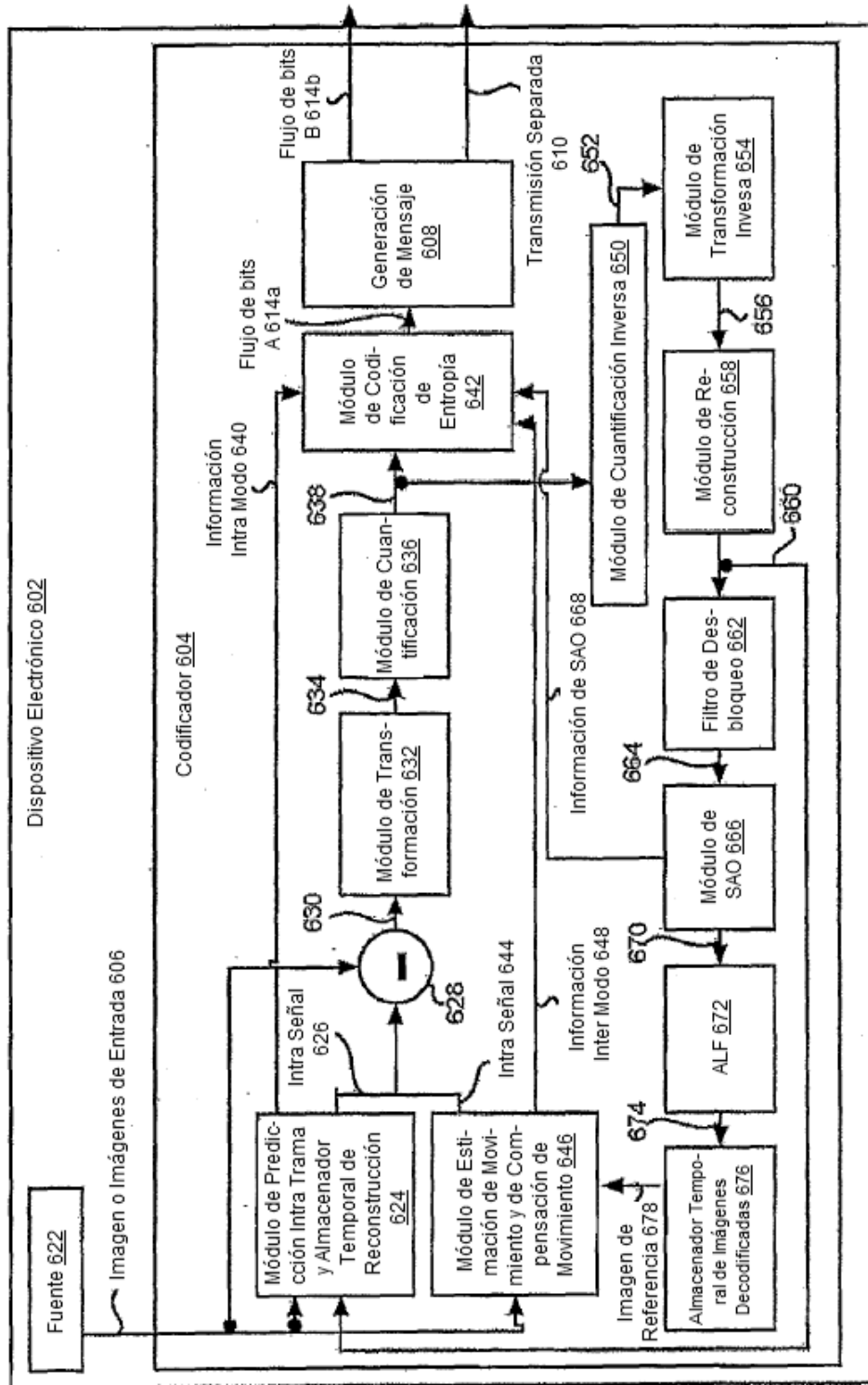
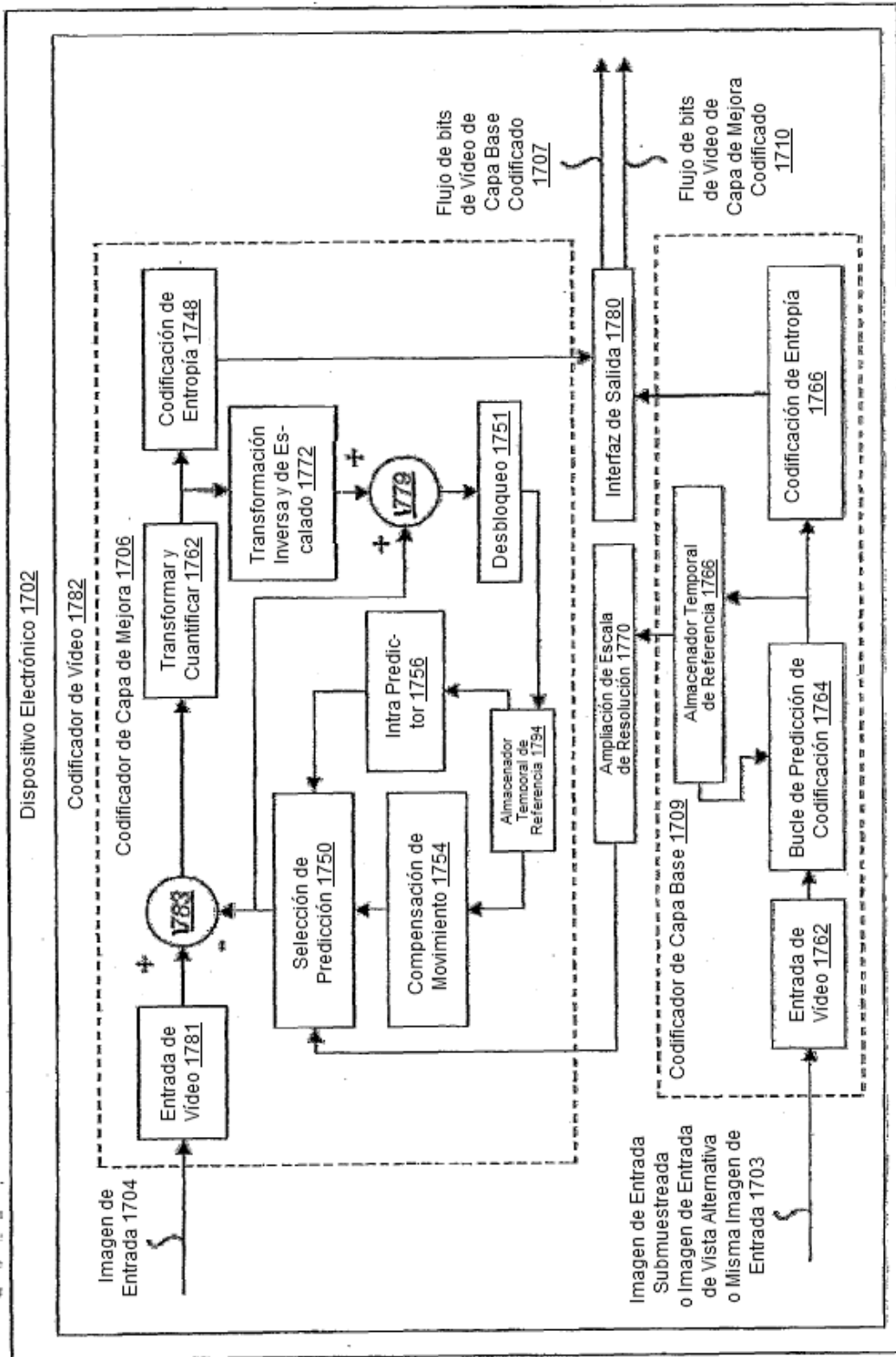


FIG. 6A

FIG. 6B



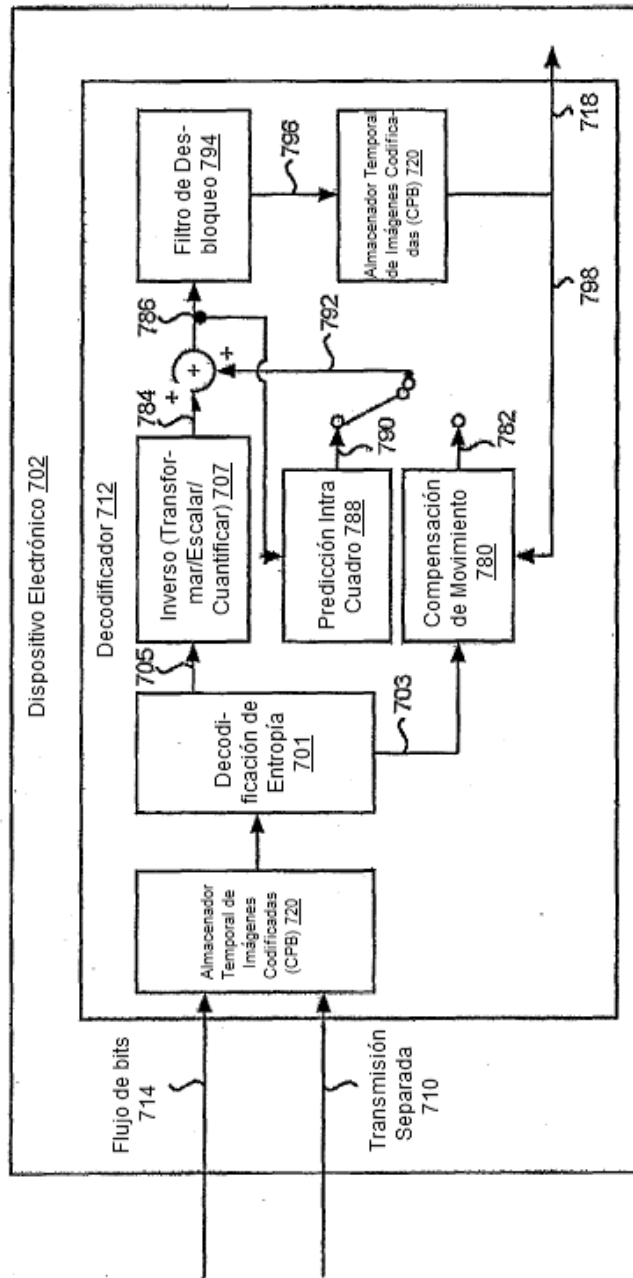
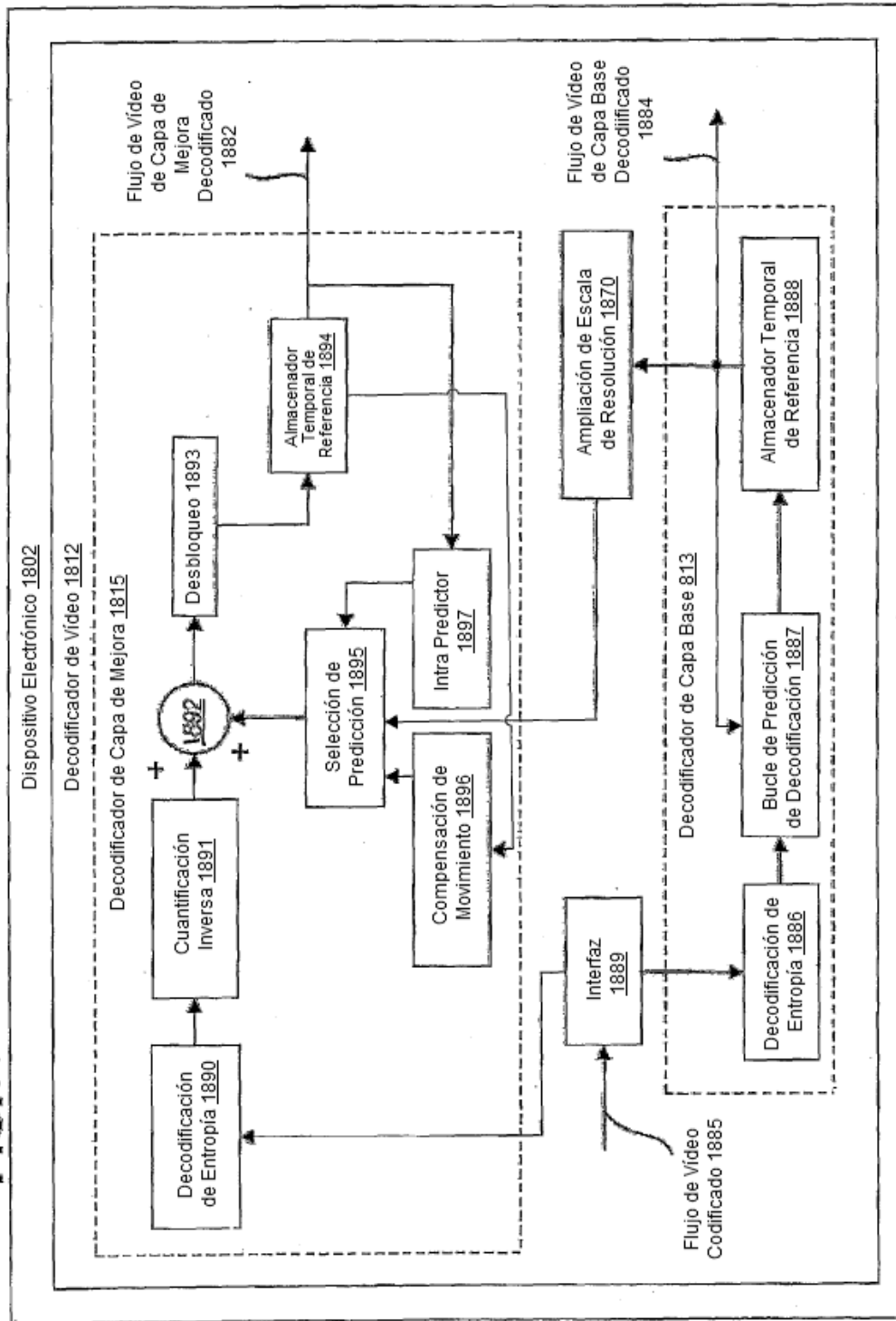


FIG. 7A

FIG. 7B



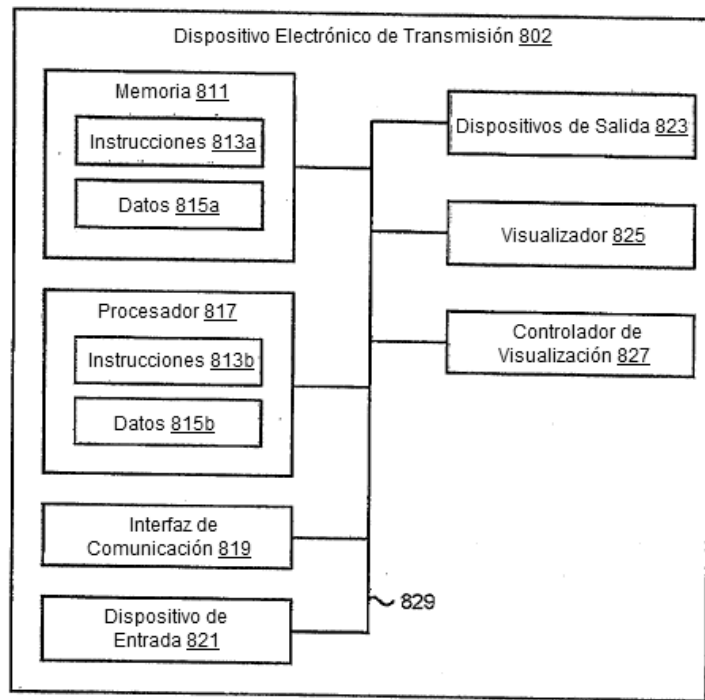


FIG. 8

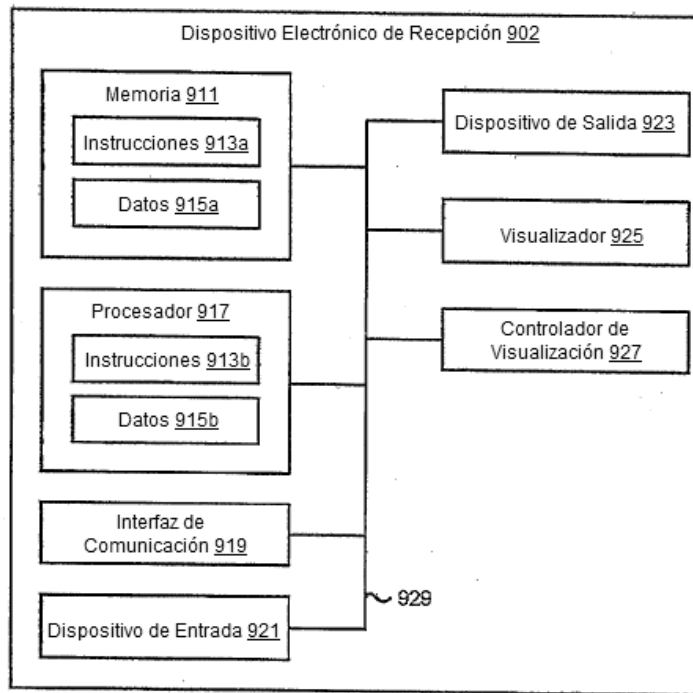


FIG. 9

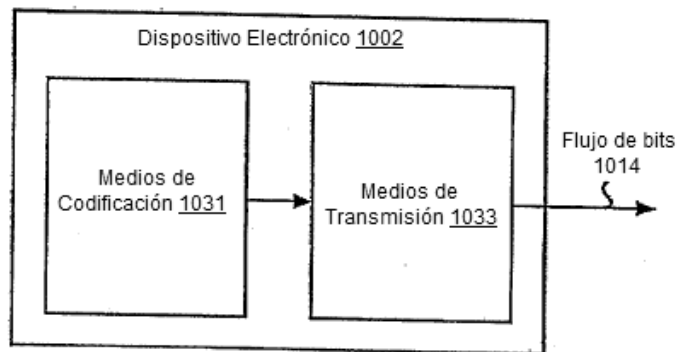


FIG. 10

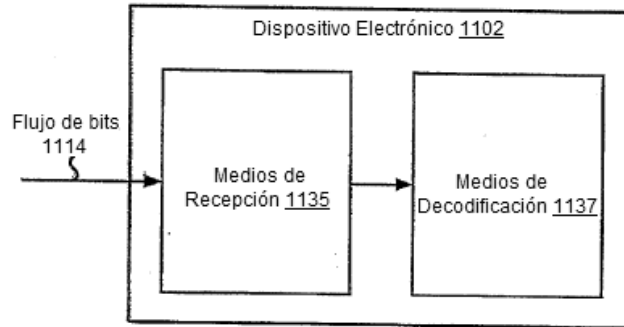


FIG. 11

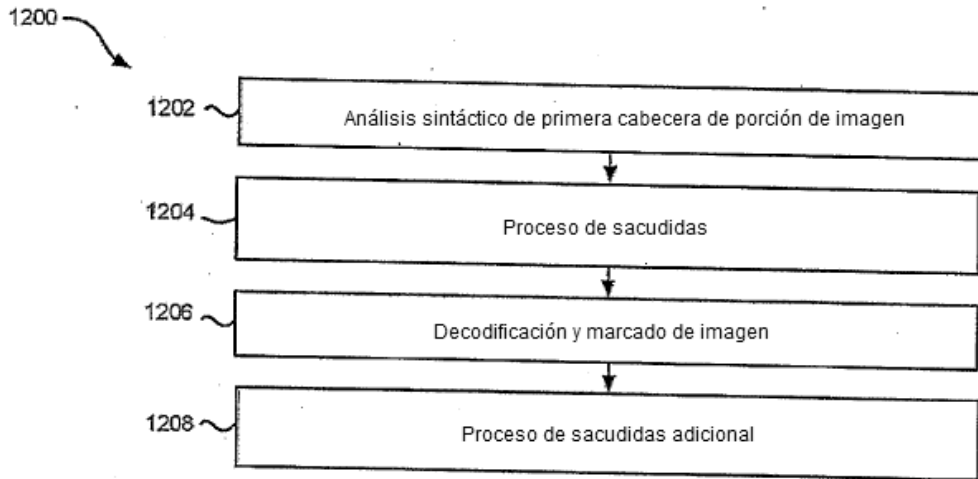


FIG. 12

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

SINTAXIS DE CABECERA DE UNIDAD DE NAL

FIG. 13A

	Descriptor
nal_unit_header() {	
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
layer_id_plus1	u(6)
nuh_temporal_id_plus1	u(3)
}	

SINTASIS DE CABECERA DE UNIDAD DE NAL

FIG. 13 b

	Descriptor
nal_unit_header() {	
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
layer_id	u(6)
nuh_temporal_id_plus1	u(3)
}	

SINTASIS DE CABECERA DE UNIDAD DE NAL

FIG. 13 c

	Descriptor
nal_unit(NumBytesInNALunit) {	
nal_unit_header() -	
NumBytesInRBSP = 0	
for(i = 2; i < NumBytesInNALunit; i++) {	
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {	
rbsp_byte[NumBytesInRBSP++]	b(8)
rbsp_byte[NumBytesInRBSP++]	b(8)
i += 2	
emulation_prevention_three_byte /* equal to 0x03 */	f(8)
} else	
rbsp_byte[NumBytesInRBSP++]	b(8)
}	
}	

SINTASIS DE UNIDAD DE NAL GENERAL

FIG. 14

vps_extension() {	Descriptor
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_zero_bit	u(1)
// scalability type and layer_id partitioning method	
scalability_type	u(4)
for(i=0; i < MaxDim(scalability_type); i++)	
layer_id_dim_len[i]	u(3)
// layer specific information	
for(i=0; i <= max_num_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
// layer dependency	
num_direct_ref_layers[i]	u(6)
for(j=0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	

Sintaxis de extensión de conjunto de parámetros de vídeo existente

FIG. 15

scalability type	MaxDim(scalability type)	Dimensiones de Escalabilidad
0	1	ninguna (HEVC base)
1	2	espacial y calidad
2	3	espacial, calidad, no especificada
3	4	espacial, calidad, no especificada, no especificada
4	2	vista múltiple y profundidad
5	3	vista múltiple, profundidad, no especificada
6	4	vista múltiple, profundidad, no especificada, no especificada
7	4	vista múltiple, espacial, calidad y profundidad
8	5	vista múltiple, espacial, calidad, profundidad, no especificada
9	6	vista múltiple, espacial, calidad, profundidad, no especificada, no especificada
10...15	Reservada	Reservada

Tipos de Escalabilidad Existentes

FIG. 16

vps_extension() {	Descriptor
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_zero_bit	u(1)
// Scalability map and layer_id_plus1 mapping method	
for (i=0; i<=max_num_layers_minus1; i++) {	
scalability_map[i]	u(3)
}	
// layer specific information	
for(i = 0; i <= max_num_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
// layer dependency	
num_direct_ref_layers[i]	n(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	

Sintaxis de extension de conjunto de parámetros de vídeo

FIG. 17

Scalability_map[i]	Dimensión de Escalabilidad
0	ninguna (HEVC)
1	espacial
2	calidad
3	profundidad
4	vista múltiple
5	no especificada
6, 7	reservada

Sintaxis de Mapa de Escalabilidad

FIG. 18

Code	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_zero_bit	u(1)
// scalability map and layer_id_plus1 mapping method	
num_scalability_dimensions_minus1	u(3)
for(i=0; i<= num_scalability_dimensions_minus1; i++) {	
scalability_map[i]	u(3)
num_bits_for_scalability_map[i]	u(3)
}	
// layer specific information	
for(i = 0; i <= max_num_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
// layer dependency	
num_direct_ref_layers[i]	u(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	

Sintaxis de extensión de conjunto de parámetros de vídeo

FIG. 19

Code	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_zero_bit	u(1)
// layer specific information	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
num_dimensions_minus1[i]	u(4)
for(j = 0; j <= num_dimensions_minus1; j++) {	
dimension_type[i][j]	u(4)
dimension_id[i][j]	u(8)
}	
// layer dependency	
num_direct_ref_layers[i]	u(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	

Sintaxis de extensión de conjunto de parámetros de vídeo existente

FIG. 20

dimension_type[i][j]	dimension_id[i][j]
0	idx de orden de vista
1	bandera de profundidad
2	ID de dependencia
3	ID de calidad
4..15	reservado

Sintaxis de ID de Dimension, Tipo de Dimensión Existente

FIG. 21

Code	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_zero_bit	u(1)
// layer specific information	
for(i = 1; i <= max_num_layers_minus1; i++) {	
// mask signalling scalability types that are present for this layer ID	
scalability_mask	u(8)
for(j = 0; j <= num_scalability_types[i]; j++) {	
scalability_id[j]	u(8)
}	
// layer dependency	
num_direct_ref_layers[i]	u(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	

Sintaxis de extensión de conjunto de parámetros de video

FIG. 22

scalability_mask_bit[k]	Dimensión de Escalabilidad
0	espacial
1	calidad
2	profundidad
3	vista múltiple
4-15	Reservada

Sintaxis de Mapa de Escalabilidad

FIG. 23

video_parameter_set_rbsp() {	Descriptor
video_parameter_set_id	u(4)
vps_temporal_id_nesting_flag	u(1)
reserved_zero_2bits	u(2)
max_num_layers_minus1 //reserved_zero_6bits	u(6)
vps_max_sub_layers_minus1	u(3)
profile_level(1, vps_max_sub_layers_minus1)	
next_essential_info_byte_offset //reserved_zero_12bits	u(12)
for(i = 0; i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering[i]	ue(v)
vps_max_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
num_hrd_parameters	ue(v)
for(i = 0; i < num_hrd_parameters; i++) {	
if(i > 0)	
op_point(i)	
hrd_parameters(i == 0, vps_max_sub_layers_minus1)	
}	
bit_equal_to_one	u(1)
scalability_mask	u(8)
for(i = 0; i <= max_num_layers_minus1; i++) {	
vps_extension_data()	
}	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	
}	
}	

Sintaxis de extensión de conjunto de parámetros de vídeo

FIG. 24

	Descriptor
vps_extension_data() {	
vps_extension_type	u(2)
layer_id	u(6)
layer_max_sub_layers_minus1	u(3)
new_profile_level_flag	u(1)
if (new_profile_level_flag == 0) {	
profile_reference_layer_id	u(6)
}	
else {	
profile_level(1, layer_max_sub_layers_minus1)	
}	
for(j = 0; j <= num_scalability_types; j++) {	
scalable_id[j]	u(8)
}	
num_reference_layers	u(6)
for(n = 0; n < num_reference_layers; n++)	
direct_coding_dependency_layer_id_plus1[n]	u(6)
}	

Sintaxis de extensión de conjunto de parámetros de vídeo

FIG. 25

vps_extension() {	Descriptor
while(byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
ava_base_codec_flag	u(1)
scalability_mask	u(16)
for(i = 0; i < NumScalabilityTypes; i++) {	
dimension_id_tent_masks[i]	u(3)
}	
vps_nuh_layer_id_present_flag	u(1)
// layer specific information	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(8)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
profile_for_level(1, vps_max_sub_layers_minus1)	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// layer dependency	
num_direct_ref_layers[i]	u(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	

Sintaxis de Extensión de Conjunto de Parámetros de Vídeo

FIG. 26

scalability_mask's k'th bit	Dimensión de Escalabilidad
0	espacial
1	calidad
2	profundidad
3	vista múltiple
4-15	Reservada

Sintaxis Scalability_mask

FIG 27

Code	Descriptor
vps_extension() {	
while(lbyte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_codec_flag	u(1)
scalability_mask	u(16)
for(i = 0; i < NumScalabilityTypes; i++) {	
dimension_id_len_minus1[i]	u(3)
}	
vps_nuh_layer_id_present_flag	u(1)
// layer specific information	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j <= num_dimensions_minus1; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
profile_tier_level(1, vps_max_sub_layers_minus1)	
layer_dependency_information_pattern	u(v)
for(l = 1; l <= NumDepLayers; l++) {	
// layer dependency information signaling	
num_direct_ref_layers[l]	u(6)
for(j = 0; j < num_direct_ref_layers[l]; j++)	
ref_layer_id[l][j]	u(6)
}	
}	
}	

Sintaxis de Extensión de Conjunto de Parámetros de Vídeo

FIG. 28

Code	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_codec_flag	u(1)
scalability_mask	u(16)
for(i = 0; i < NumScalabilityTypes; i++) {	
dimension_id_len_minus1[i]	u(3)
}	
vps_nuh_layer_id_present_flag	u(1)
// layer specific information	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j <= num_dimensions_minus1; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
profile_tier_level(i, vps_max_sub_layers_minus1)	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// layer dependency information signaling	
layer_dependency_flag[i]	u(1)
if(layer_dependency_flag[i]) {	
num_direct_ref_layers[i]	u(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	
}	
}	

Sintaxis de Extensión de Conjunto de Parámetros de Vídeo

FIG. 29

	Descriptor
<code>vps_extension() {</code>	
<code>while(!byte_aligned())</code>	
<code>vps_extension_byte_alignment_reserved_one_bit</code>	u(1)
<code>avc_base_codec_flag</code>	u(1)
<code>scalability_mask</code>	u(16)
<code>for(i = 0; i < NumScalabilityTypes; i++) {</code>	
<code>dimension_id_len_minus1[i]</code>	u(3)
<code>}</code>	
<code>vps_nuh_layer_id_present_flag</code>	u(1)
<code>// layer specific information</code>	
<code>for(i = 1; i <= vps_max_layers_minus1; i++) {</code>	
<code>// mapping of layer ID to scalability dimension IDs</code>	
<code>if(vps_nuh_layer_id_present_flag)</code>	
<code>layer_id_in_nuh[i]</code>	u(6)
<code>for(j = 0; j <= num_dimensions_minus1; j++)</code>	
<code>dimension_id[i][j]</code>	u(v)
<code>}</code>	
<code>for(i = 1; i <= vps_max_layers_minus1; i++)</code>	
<code>profile_tier_level(i, vps_max_sub_layers_minus1)</code>	
<code>for(i = 1; i <= vps_max_layers_minus1; i++) {</code>	
<code>// layer dependency information signaling</code>	
<code>layer_dependency_map[i]</code>	u(v)
<code>}</code>	
<code>}</code>	

Sintaxis de Extensión de Conjunto de Parámetros de Vídeo

FIG. 3D

	Descriptor
<code>vps_extension() {</code>	
<code>while(!byte_aligned())</code>	
<code>vps_extension_byte_alignment_reserved_one_bit</code>	u(1)
<code>avc_base_codec_flag</code>	u(1)
<code>scalability_mask</code>	u(16)
<code>for(i = 0; i < NumScalabilityTypes; i++) {</code>	
<code>dimension_id_len_minus1[i]</code>	u(3)
<code>}</code>	
<code>vps_nuh_layer_id_present_flag</code>	u(1)
<code>// layer specific information</code>	
<code>for(i = 1; i <= vps_max_layers_minus1; i++) {</code>	
<code>// mapping of layer ID to scalability dimension IDs</code>	
<code>if(vps_nuh_layer_id_present_flag)</code>	
<code>layer_id_in_nuh[i]</code>	u(6)
<code>for(j = 0; j <= num_dimensions_minus1; j++)</code>	
<code>dimension_id[i][j]</code>	u(v)
<code>}</code>	
<code>for(i = 1; i <= vps_max_layers_minus1 ; i++)</code>	
<code>profile_tier_level(i, vps_max_sub_layers_minus1)</code>	
<code>layer_dependency_information_pattern</code>	u(v)
<code>for(i = 1; i <= NumDepLayers; i++)</code>	
<code>layer_dependency_map[i]</code>	u(v)
<code>}</code>	

Sintaxis de Extensión de Conjunto de Parámetros de Vídeo

FIG. 3 \

Code	Descriptor
vps_extension() {	
while(lbyte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_codec_flag	u(1)
scalability_mask	u(16)
for(i = 0; i < NumScalabilityTypes; i++) {	
dimension_id_len_minus1[i]	u(3)
}	
vps_nuh_layer_id_present_flag	u(1)
// layer specific information	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j <= num_dimensions_minus1; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
profile_tier_level(1, vps_max_sub_layers_minus1)	
layer_dependency_information_pattern	u(v)
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(layer_dependency_information_pattern(i)) {	
// layer dependency information signalling	
num_direct_ref_layers[i]	u(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	
}	
}	

Sintaxis de Extensión de Conjunto de Parámetros de Vídeo

FIG. 32.

Code	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_codec_flag	u(1)
scalability_mask	u(16)
for(i = 0; i < NumScalabilityTypes; i++) {	
dimension_id_len_minus1[i]	u(3)
}	
vps_nuh_layer_id_present_flag	u(1)
// layer specific information	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j <= num_dimensions_minus1; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
profile_tier_level(1, vps_max_sub_layers_minus1)	
layer_dependency_information_pattern	u(v)
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(layer_dependency_information_pattern(i)) {	
layer_dependency_map[i]	u(v)
}	
}	
}	

Sintaxis de Extensión de Conjunto de Parámetros de Vídeo

FIG. 33

	Descriptor
video_parameter_set_rbsp() {	
vps_video_parameter_set_id	u(4)
vps_reserved_three_2bits	u(2)
vps_max_layers_minus1	u(6)
vps_max_sub_layers_minus1	u(3)
vps_temporal_id_nesting_flag	u(1)
vps_extension_offset //vps_reserved_0xffff_16bits	u(16)
profile_tier_level(1, vps_max_sub_layers_minus1)	
vps_sub_layer_ordering_info_present_flag	u(1)
for(i = (vps_sub_layer_ordering_info_present_flag ? 0 : vps_max_sub_layers_minus1);	
i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering_minus1[i]	ue(v)
vps_max_num_reorder_pics[i]	ue(v)
vps_max_latency_increase_plus1[i]	ue(v)
}	
vps_max_layer_id	u(6)
vps_num_layer_sets_minus1	ue(v)
for(i = 1; i <= vps_num_layer_sets_minus1; i++)	
for(j = 0; j <= vps_max_layer_id; j++)	
layer_id_included_flag[i][j]	u(1)
vps_timing_info_present_flag	u(1)
if(vps_timing_info_present_flag) {	
vps_num_units_in_tick	u(32)
vps_time_scale	u(32)
vps_poc_proportional_to_timing_flag	u(1)
if(vps_poc_proportional_to_timing_flag)	
vps_num_ticks_poc_diff_one_minus1	ue(v)
vps_num_hrd_parameters	ue(v)
for(i = 0; i < vps_num_hrd_parameters; i++) {	
hrd_layer_set_idx[i]	ue(v)
if(i > 0)	
cprms_present_flag[i]	u(1)
hrd_parameters(cprms_present_flag[i], vps_max_sub_layers_minus1)	
}	
}	
vps_extension_flag	u(1)
if(vps_extension_flag) {	
vps_extension()	
vps_extension2_flag	u(1)
if(vps_extension2_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

FIG. 34

	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0, NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
vps_nuh_layer_id_present_flag	u(1)
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(lsIdx = 1; lsIdx <= vps_num_layer_sets_minus1; lsIdx ++) {	
vps_profile_present_flag[lsIdx]	u(1)
if(!vps_profile_present_flag[lsIdx])	
profile_layer_set_ref_minus1[lsIdx]	ue(v)
profile_tier_level(vps_profile_present_flag[lsIdx], vps_max_sub_layers_minus1)	
}	
num_output_layer_sets	ue(v)
for(i = 0; i < num_output_layer_sets; i++) {	
output_layer_set_idx[i]	ue(v)
lsIdx = output_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[lsIdx][j])	
output_layer_flag[lsIdx][j]	u(1)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

FIG. 35

	Descriptor
output_layer_sets_change(payloadSize) {	
active_vps_id	u(4)
num_changed_output_layer_sets	ue(v)
for(i = 0; i < num_changed_output_layer_sets; i++) {	
changed_output_layer_set_idx_entry[i]	ue(v)
clsIdx = output_layer_set_idx[changed_output_layer_set_idx_entry[i]]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[clsIdx][j])	
output_layer_flag[clsIdx][j]	u(1)
}	
num_addl_output_layer_sets	
for(i = num_output_layer_sets; i < num_output_layer_sets + num_addl_output_layer_sets; i++) {	
addlIdx = addl_output_layer_sets_idx[i]	ue(v)
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[addlIdx][j])	
output_layer_flag[addlIdx][j]	u(1)
}	
}	

FIG. 36

	Descriptor
output_layer_sets_change(payloadSize) {	
active_vps_id	u(4)
num_deleted_output_layer_sets	ue(v)
for(i = 0; i < num_deleted_output_layer_sets; i++)	
deleted_output_layer_sets_idx_entry[i]	ue(v)
num_changed_output_layer_sets	ue(v)
for(i = 0; i < num_changed_output_layer_sets; i++) {	
changed_output_layer_set_idx_entry[i]	ue(v)
clsIdx = output_layer_set_idx[changed_output_layer_set_idx_entry[i]]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[clsIdx][j])	
output_layer_flag[clsIdx][j]	u(1)
}	
num_addl_output_layer_sets	
for(i = num_output_layer_sets - num_deleted_output_layer_sets; i < num_output_layer_sets - num_deleted_output_layer_sets + num_addl_output_layer_sets; i++) {	
addlIdx = addl_output_layer_sets_idx[i]	ue(v)
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[addlIdx][j])	
output_layer_flag[addlIdx][j]	u(1)
}	
}	

FIG. 37

	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
ave_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0; NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
...	
num_output_layer_sets	ue(v)
for(i = 0; i < num_output_layer_sets; i++) {	
output_layer_set_idx[i]	ue(v)
lsIdx = output_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[lsIdx][j])	
output_layer_flag[lsIdx][j]	u(1)
}	
num_op_dpb_info_parameters	ue(v)
for(i = 0; i < num_op_dpb_info_parameters; i++) {	
operation_point_layer_set_idx[i]	ue(v)
for(j = 0; j <= vps_max_layer_id; j++)	
op_dpb_info_parameters(j)	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

FIG. 38A

	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
ave_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0, NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
...	
num_output_layer_sets	ue(v)
for(i = 0; i < num_output_layer_sets; i++) {	
output_layer_set_idx[i]	ue(v)
lsIdx = output_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[lsIdx][j])	
output_layer_flag[lsIdx][j]	u(1)
}	
num_op_dpb_info_parameters	ue(v)
for(i = 0; i < num_op_dpb_info_parameters; i++) {	
operation_point_layer_set_idx[i]	ue(v)
for(j = 0; j <= vps_max_layer_id; j++)	
op_dpb_info_parameters(i, j)	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

FIG. 38B

op_dpb_info_parameters(j) {	
vps_max_sub_layers_minus1[j]	u(3)
vps_sub_layer_ordering_info_present_flag[j]	u(1)
for(k = (vps_sub_layer_ordering_info_present_flag[j] ? 0 : vps_max_sub_layers_minus1[j]);	
k <= vps_max_sub_layers_minus1[j]; k++) {	
vps_max_dec_pic_buffering_minus1[j][k]	ue(v)
vps_max_num_reorder_pics[j][k]	ue(v)
vps_max_latency_increase_plus1[j][k]	ue(v)
}	
}	

FIG. 39A

op dpb info parameters(id, j) {	
vps_max_sub_layers_minus1[id][j]	u(3)
vps_sub_layer_ordering_info_present_flag[id][j]	u(1)
for(k = (vps_sub_layer_ordering_info_present_flag[id][j] ? 0 : vps_max_sub_layers_minus1[id][j]); k <= vps_max_sub_layers_minus1[id][j]; k++) {	
vps_max_dec_pic_buffering_minus1[id][j][k]	ue(v)
vps_max_num_reorder_pics[id][j][k]	ue(v)
vps_max_latency_increase_plus1[id][j][k]	ue(v)
}	
}	

FIG. 39B

	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0; NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
...	
num_output_layer_sets	ue(v)
for(i = 0; i < num_output_layer_sets; i++) {	
output_layer_set_idx[i]	ue(v)
lsIdx = output_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[lsIdx][j])	
output_layer_flag[lsIdx][j]	u(1)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
output_point_layer_set_idx[i]	ue(v)
oplsIdx = output_point_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
oop_dpb_info_parameters(j)	
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

FIG. 40

oop_dpb_info_parameters(j) {	
vps_max_sub_layers_minus1[j]	u(3)
vps_sub_layer_ordering_info_present_flag[j]	u(1)
for(k = (vps_sub_layer_ordering_info_present_flag[j] ? 0 : vps_max_sub_layers_minus1[j]);	
k <= vps_max_sub_layers_minus1[j]; k++) {	
vps_max_dec_pic_buffering_minus1[j][k]	ue(v)
vps_max_num_reorder_pics[j][k]	ue(v)
vps_max_latency_increase_plus1[j][k]	ue(v)
}	
}	

FIG. 41

oop_dpb_info_parameters() {	
vps_max_sub_layers_minus1	u(3)
vps_sub_layer_ordering_info_present_flag	u(1)
for(k = (vps_sub_layer_ordering_info_present_flag ? 0 : vps_max_sub_layers_minus1);	
k <= vps_max_sub_layers_minus1; k++) {	
vps_max_dec_pic_buffering_minus1[k]	ue(v)
vps_max_num_reorder_pics[k]	ue(v)
vps_max_latency_increase_plus1[k]	ue(v)
}	
}	

FIG. 42

num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
output_point_layer_set_idx[i]	ue(v)
oplsidx = output_point_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
oop_dpb_info_parameters()	
}	

FIG. 43

oop_dpb_info_parameters(id, j) {	
vps_max_sub_layers_minus1[id][j]	u(3)
vps_sub_layer_ordering_info_present_flag[id][j]	u(1)
for(k = (vps_sub_layer_ordering_info_present_flag[id][j] ? 0 : vps_max_sub_layers_minus1[id][j]);	
k <= vps_max_sub_layers_minus1[id][j]; k++) {	
vps_max_dec_pic_buffering_minus1[id][j][k]	ue(v)
vps_max_num_reorder_pics[id][j][k]	ue(v)
vps_max_latency_increase_plus1[id][j][k]	ue(v)
}	
}	

FIG. 44

num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
output_point_layer_set_idx[i]	ue(v)
oplsIdx=output_point_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
oop_dpb_info_parameters(oplsIdx, j)	
}	

FIG. 45

num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
output_point_layer_set_idx[i]	ue(v)
oplsIdx=output_point_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
oop_dpb_info_parameters(i, j)	
}	

FIG. 46

vps_extension() {	Descriptor
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
ave_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0, NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
...	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
for(i = 1; i <= vps_max_layers_minus1; i++) {	
layer_dpb_info(i)	
}	
}	
}	

layer_dpb_info(i) {	
vps_max_sub_layers_minus1[i]	u(3)
vps_sub_layer_ordering_info_present_flag[i]	u(1)
for(k = (vps_sub_layer_ordering_info_present_flag[i] ? 0 : vps_max_sub_layers_minus1[i]);	
k <= vps_max_sub_layers_minus1[i]; k++) {	
vps_max_dec_pic_buffering_minus1[i][k]	ue(v)
vps_max_num_reorder_pics[i][k]	ue(v)
vps_max_latency_increase_plus1[i][k]	ue(v)
}	
}	

FIG. 47

oop_dpb_info_parameters(id) / op_dpb_info_parameters(id) / layer_dpb_info(id) {	
vps_sub_layer_ordering_info_present_flag[id]	u(1)
for(i = (vps_sub_layer_ordering_info_present_flag ? 0 : vps_max_sub_layers_minus1);	
i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering_minus1[id][i]	ue(v)
vps_max_num_reorder_pics[id][i]	ue(v)
vps_max_latency_increase_plus1[id][i]	ue(v)
}	
}	

FIG. 48

	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0, NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
vps_nuh_layer_id_present_flag	u(1)
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
max_sub_layers_vps_predict_flag[i]	u(1)
if(!max_sub_layers_vps_predict_flag[i])	
max_sub_layers_vps_minus1[i]	u(3)
}	
vps_number_layer_sets_minus1	u(10)
vps_num_profile_tier_level_minus1	u(6)
for(i = 1; i <= vps_num_profile_tier_level_minus1; i++) {	
vps_profile_present_flag[i]	u(1)
if(!vps_profile_present_flag[i])	
profile_ref_minus1[i]	u(6)
profile_tier_level(vps_profile_present_flag[i], vps_max_sub_layers_minus1)	
}	

FIG. 49A

multiple_output_layer_sets_in_layer_setmore_output_layer_sets_than_default_flag	u(1)
if(!multiple_output_layer_sets_in_layer_setmore_output_layer_sets_than_default_flag)	
numOutputLayerSets = vps_number_layer_sets_minus1 + 1	
else {	
num_add_output_layer_sets	u(10)
numOutputLayerSets = num_add_output_layer_sets +vps_number_layer_sets_minus1 + 1	
}	
if(numOutputLayerSets > 1)	
default_one_target_output_layer_flag	u(1)
for(i = 1; i < numOutputLayerSets; i++) {	
if(i > vps_num_layer_sets_minus1) {	
output_layer_set_idx_minus1[i]	u(v)
lsIdx = output_layer_set_idx_minus1[i] + 1	
for(j = 0 ; j < NumLayersInIdList[lsIdx] - 1; j++)	
output_layer_flag[i][j]	u(1)
}	
profile_level_tier_idx[i]	u(v)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
output_point_layer_set_idx[i]	ue(v)
oop_dpb_maxbuffering_parameters(i)	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
layer_dpb_info_parameters(i)	
}	
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

FIG. 49B

oop_dpb_maxbuffering_parameters (i) {	
sub_layer_vps_buf_info_present_flag[i]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i] ? 0 : MaxSubLayers[i] - 1); j <= MaxSubLayers[i] - 1; j++)	
max_vps_dec_pic_buffering_minus1[i][j]	ue(v)
}	

FIG. 50

layer_dpb_info_parameters(i) {	
sub_layer_vps_ordering_info_present_flag[i]	u(1)
for(j = (sub_layer_vps_ordering_info_present_flag[i] ? 0 : max_sub_layers_vps_minus1[i]); j <= max_sub_layers_vps_minus1[i]; j++) {	
max_vps_num_reorder_pics[i][j]	ue(v)
max_vps_latency_increase_plus1[i][j]	ue(v)
}	
}	

FIG. 51

vps_extension() {	Descriptor
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
...	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
max_sub_layers_vps_predict_flag[i]	u(1)
if(!max_sub_layers_vps_predict_flag[i])	
max_sub_layers_vps_minus1[i]	u(3)
}	
...	
profile_level_tier_idx[i]	u(v)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
if(i>0 && (MaxSubLayers[i] == MaxSubLayers[i-1]))	
sub_layer_vps_buf_info_predict_flag[i]	u(1)
if(!sub_layer_vps_buf_info_predict_flag[i]) {	
output_point_layer_set_idx[i]	ue(v)
oop_dpb_maxbuffering_parameters(i)	
}	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(max_sub_layers_vps_predict_flag[i])	
sub_layer_vps_ordering_info_predict_flag[i]	u(1)
if(!sub_layer_ordering_info_predict_flag[i])	
layer_dpb_info_parameters(i)	
}	
}	
...	
}	

FIG. 52

	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
...	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
max_sub_layers_vps_predict_flag[i]	u(1)
if(!max_sub_layers_vps_predict_flag[i])	
max_sub_layers_vps_minus1[i]	u(3)
}	
...	
profile_level_tier_idx[i]	u(v)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
output_point_layer_set_idx[i]	ue(v)
for(k = 1; k <= vps_max_layers_minus1; i++) {	
oop_dpb_maxbuffering_parameters(i, k)	
}	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
layer_dpb_info_parameters(i)	
}	
}	
...	
}	

FIG. 53

oop_dpb_maxbuffering_parameters (i, k) {	
sub_layer_vps_buf_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i][k] ? 0 :	
max_sub_layers_vps_minus1[k]); j <= max_sub_layers_vps_minus1[k]; j++); j++)	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 54

oop_dpb_maxbuffering_parameters (i, k) {	
sub_layer_vps_buf_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i] ? 0 : MaxSubLayers[i][k] - 1); j <= MaxSubLayers[i][k] - 1; j++)	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 55

	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
...	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
max_sub_layers_vps_predict_flag[i]	u(1)
if(!max_sub_layers_vps_predict_flag[i])	
max_sub_layers_vps_minus1[i]	u(3)
}	
...	
profile_level_tier_idx[i]	u(v)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
if(i > 0 && (MaxSubLayers[i] == MaxSubLayers[i - 1]))	
sub_layer_vps_buf_info_predict_flag[i]	u(1)
if(!sub_layer_vps_buf_info_predict_flag[i]) {	
output_point_layer_set_idx[i]	ue(v)
for(k = 1; k <= vps_max_layers_minus1; i++) {	
oop_dpb_maxbuffering_parameters(i, k)	
}	
}	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(max_sub_layers_vps_predict_flag[i])	
sub_layer_vps_ordering_info_predict_flag[i]	u(1)
if(!sub_layer_ordering_info_predict_flag[i])	
layer_dpb_info_parameters(i)	
}	
}	
...	
}	

FIG. 56

oop_dpb_maxbuffering_parameters (i, k) {	
sub_layer_vps_buf_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i][k] ? 0 : max_sub_layers_vps_minus1[k]); j <= max_sub_layers_vps_minus1[k]; j++); j++)	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 57

oop_dpb_maxbuffering_parameters (i, k) {	
sub_layer_vps_buf_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i] ? 0 : MaxSubLayers[i][k] - 1); j <= MaxSubLayers[i][k] - 1; j++)	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 58

	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
...	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
max_sub_layers_vps_predict_flag[i]	u(1)
if(!max_sub_layers_vps_predict_flag[i])	
max_sub_layers_vps_minus1[i]	u(3)
}	
...	
profile_level_tier_idx[i]	u(v)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
if(i>0 && (MaxSubLayers[i] == MaxSubLayers[i - 1]))	
sub_layer_vps_buf_info_predict_flag[i]	u(1)
if(!sub_layer_vps_buf_info_predict_flag[i]) {	
output_point_layer_set_idx[i]	ue(v)
for(k = 1; k <= vps_max_layers_minus1; i++) {	
oop_dpb_maxbuffering_parameters(i, k)	
}	
}	
}	
layer_dpb_info_parameters_presence_flag	u(1)
If(layer_dpb_info_parameters_presence_flag) {	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(max_sub_layers_vps_predict_flag[i])	
sub_layer_vps_ordering_info_predict_flag[i]	u(1)
if(!sub_layer_ordering_info_predict_flag[i])	
layer_dpb_info_parameters(i)	
}	
}	
}	
...	
}	

FIG. 59

oop_dpb_maxbuffering_parameters (i, k) {	
sub_layer_vps_maxbuffering_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_maxbuffering_info_present_flag [i][k] ? 0 : max_sub_layers_vps_minus1[k]); j <= max_sub_layers_vps_minus1[k]; j++)	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 60

oop_dpb_maxbuffering_parameters (i, k) {	
sub_layer_vps_buf_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i] ? 0 : MaxSubLayers[i][k] - 1); j <= MaxSubLayers[i][k] - 1; j++)	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 61

	Descriptor
seq_parameter_set_rbsp() {	
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
if(nuh_layer_id == 0)	
profile_tier_level(1, sps_max_sub_layers_minus1)	
sps_seq_parameter_set_id	ue(v)
...	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
if(nuh_layer_id > 0)	
sps_dpb_params_present_flag	u(1)
if((nuh_layer_id == 0) ((nuh_layer_id > 0) && sps_dpb_params_present_flag)) {	
sps_sub_layer_ordering_info_present_flag	u(1)
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1); i <= sps_max_sub_layers_minus1; i++) {	
sps_max_dec_pic_buffering_minus1[i]	ue(v)
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase_plus1[i]	ue(v)
}	
}	
log2_min_luma_coding_block_size_minus3	ue(v)
log2_diff_max_min_luma_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
...	
sps_extension_flag	u(1)
if(sps_extension_flag) {	
sps_extension()	
sps_extension2_flag	u(1)
if(sps_extension2_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

FIG. 62

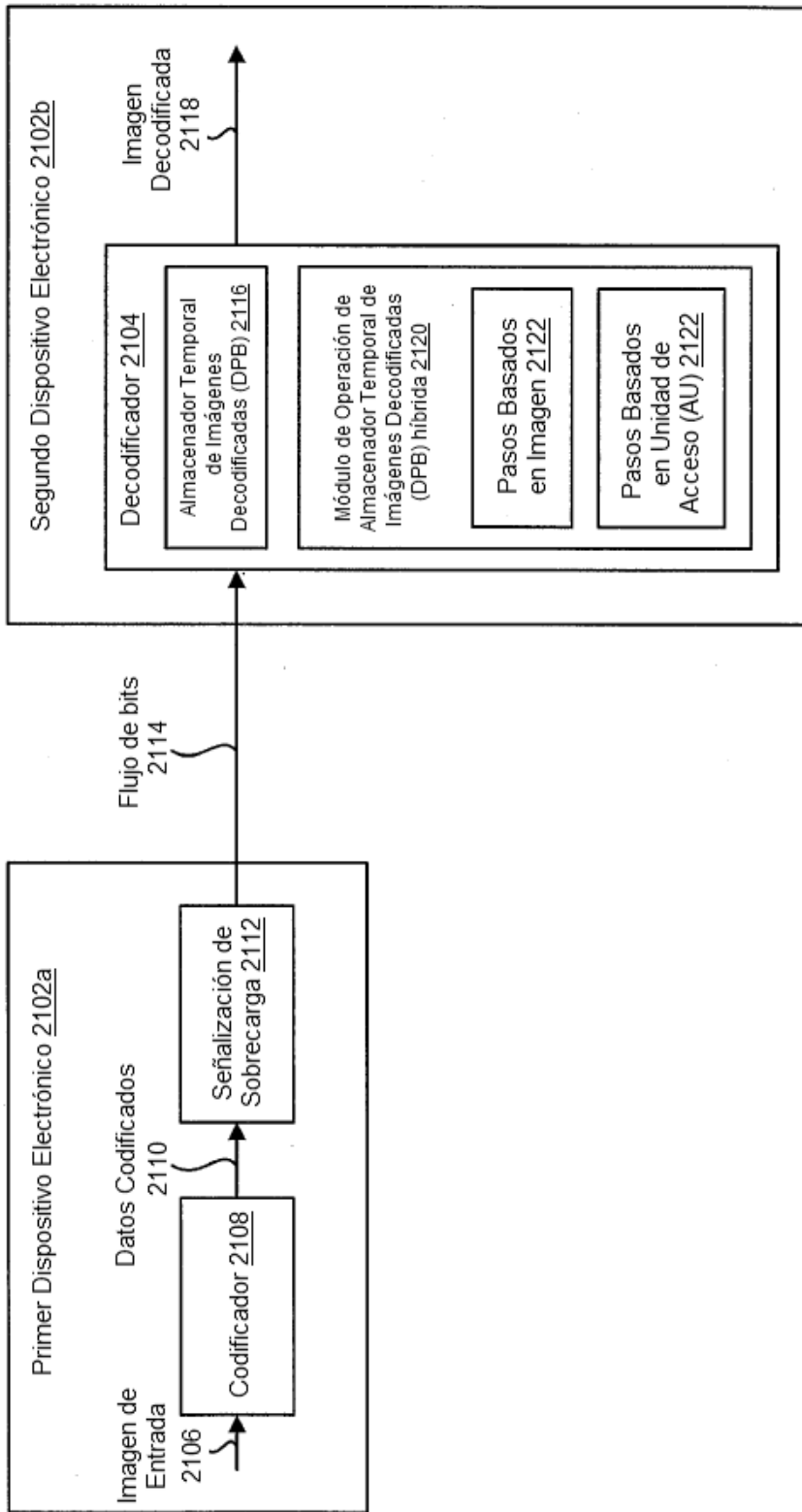


FIG. 63

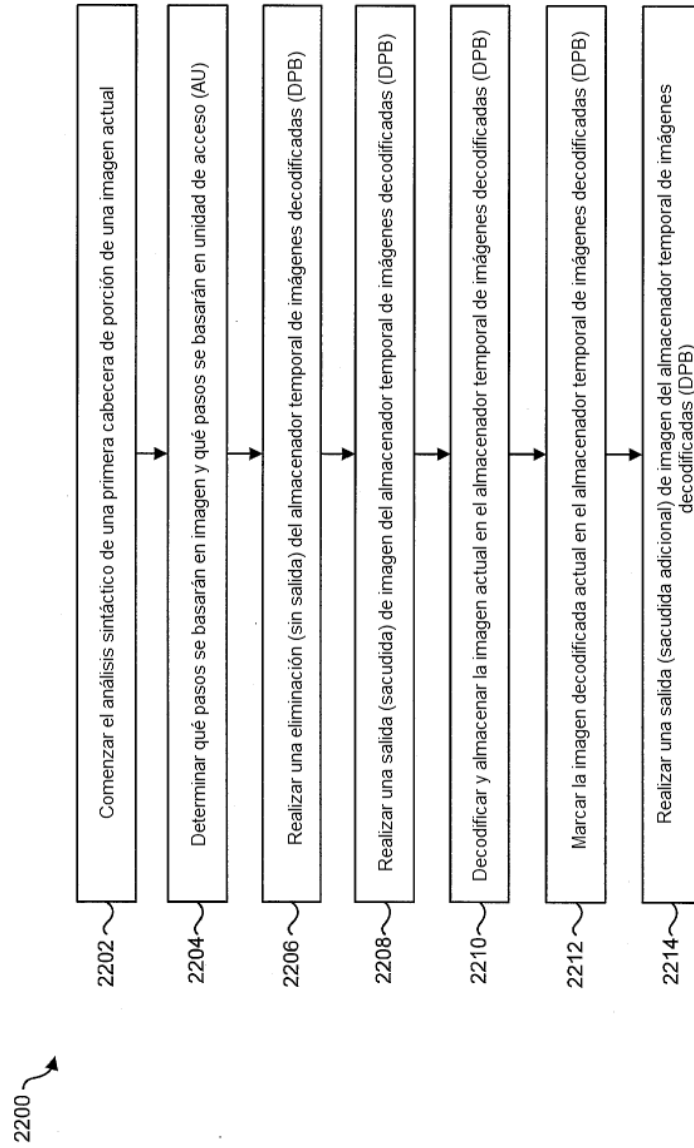


FIG. 64

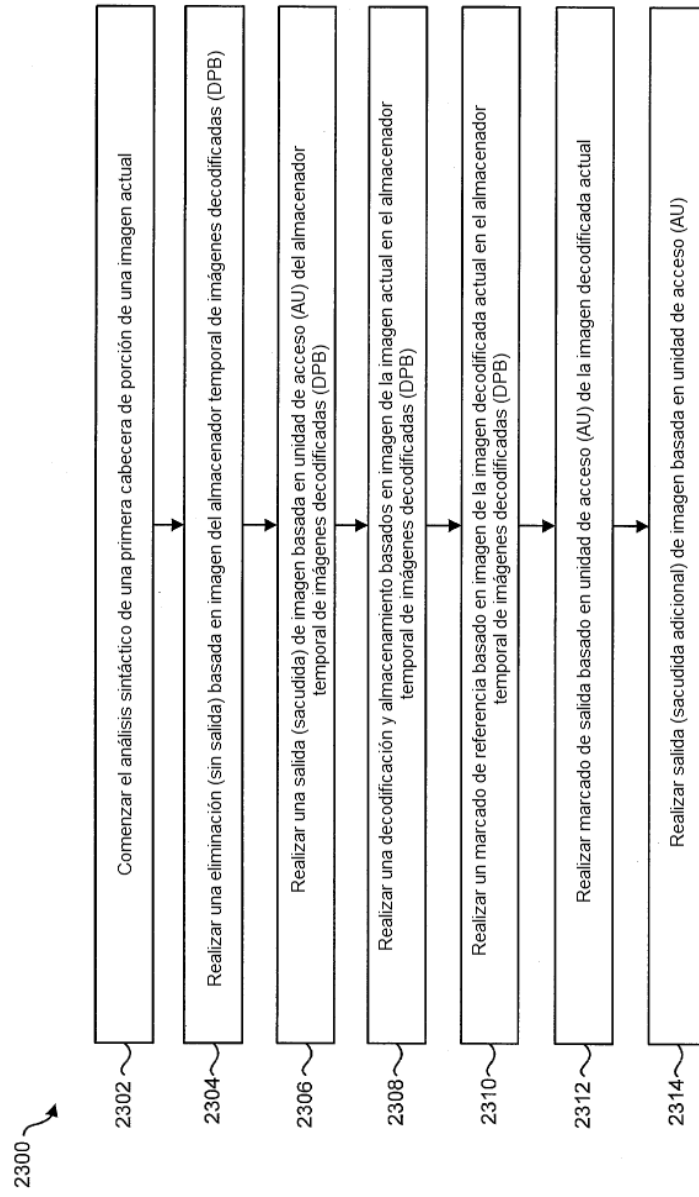


FIG. 65

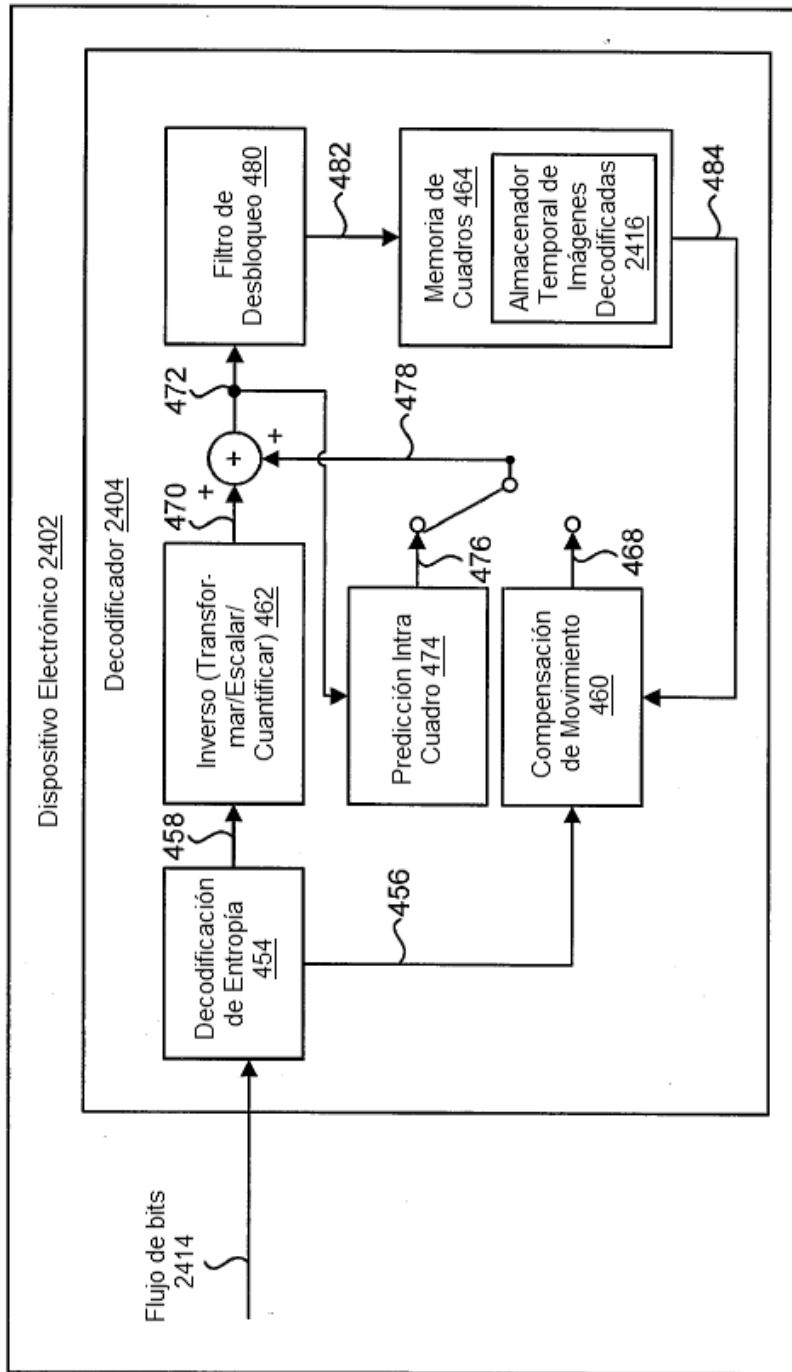


FIG. 66

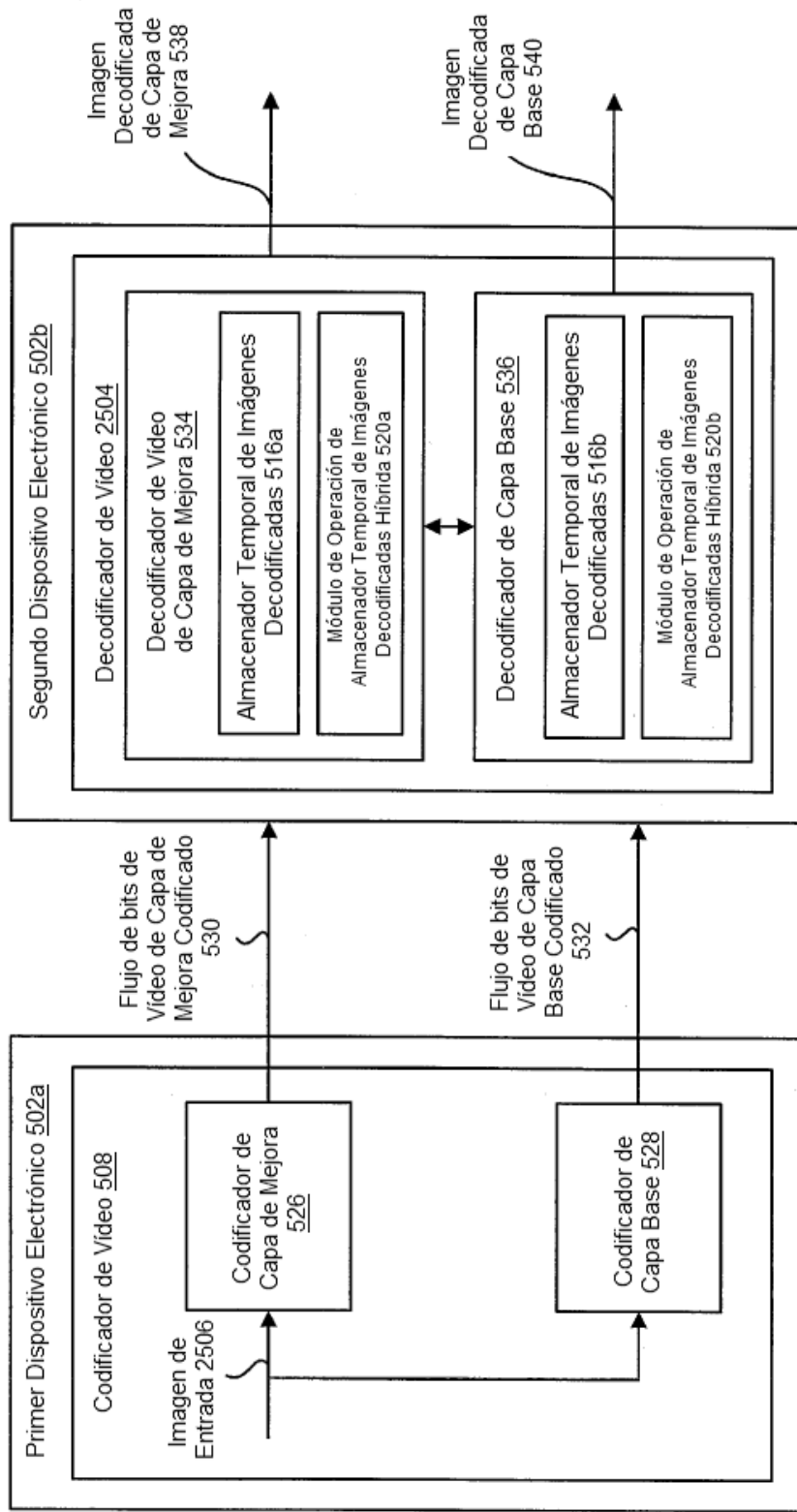


FIG. 67A

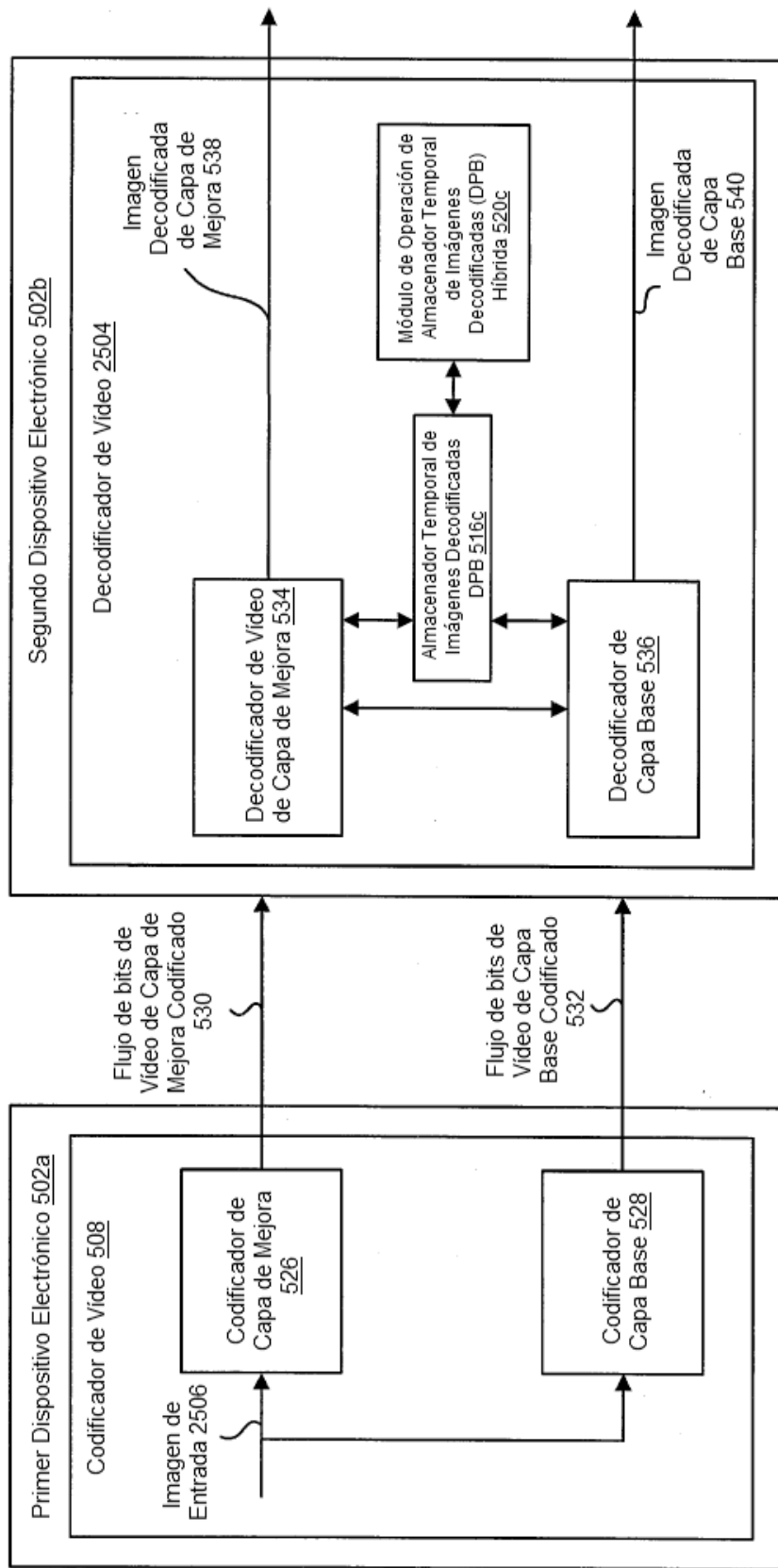


FIG. 67B

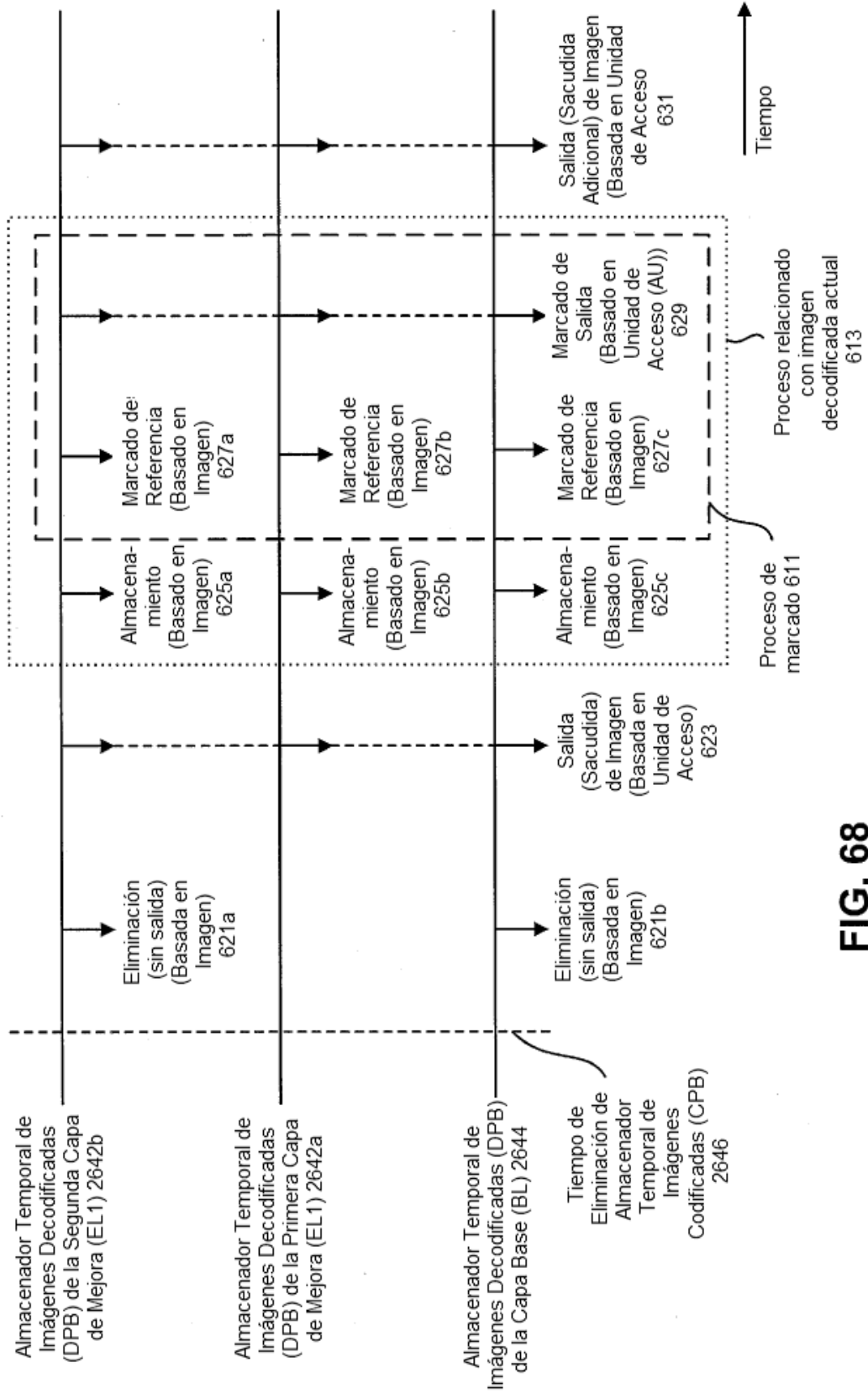


FIG. 68

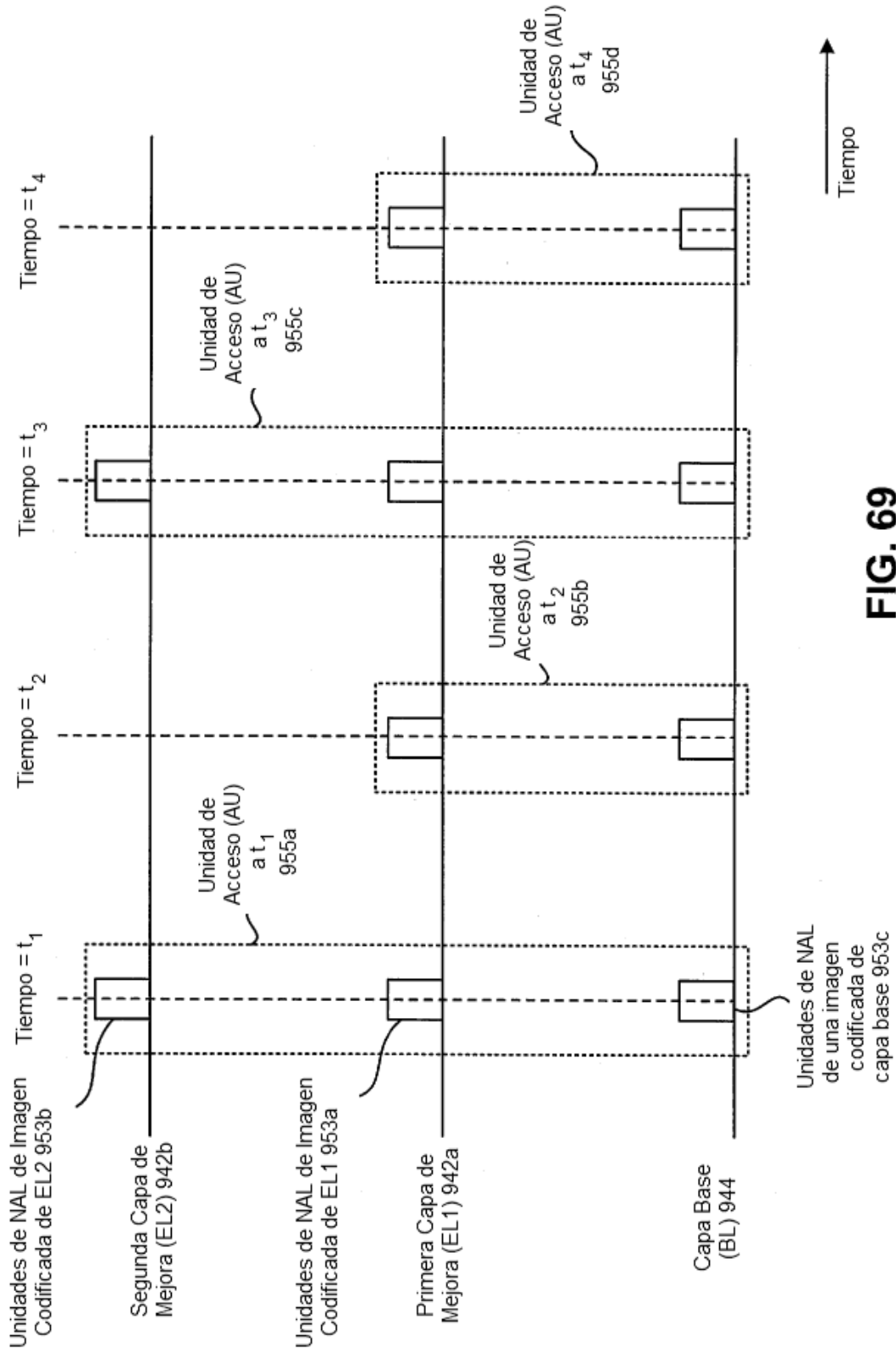


FIG. 69

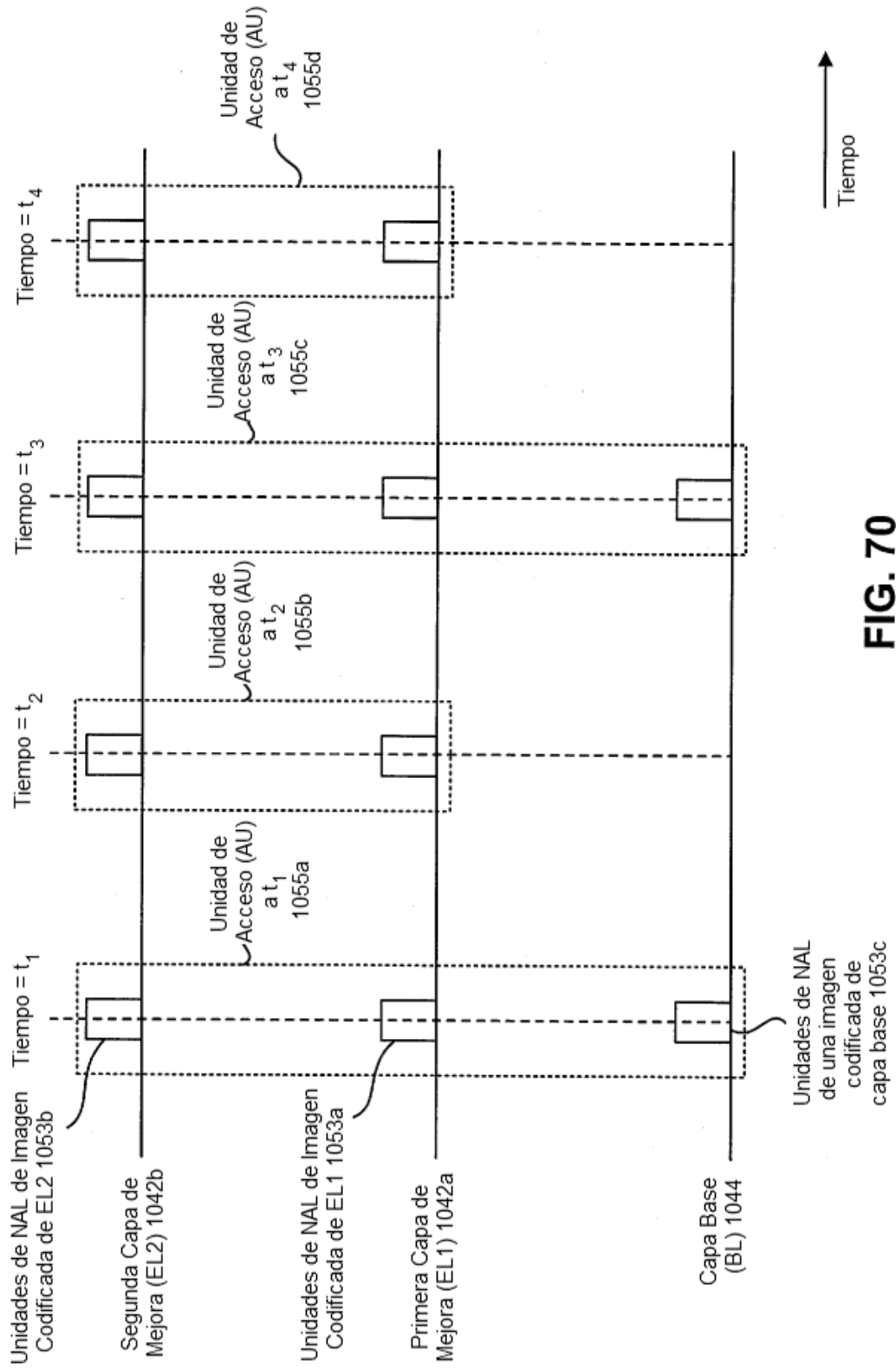


FIG. 70