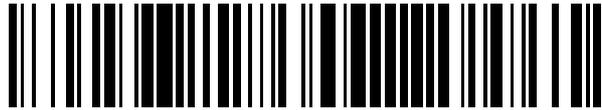


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 765 807**

51 Int. Cl.:

<b>H04N 19/13</b>	(2014.01)
<b>H04N 19/64</b>	(2014.01)
<b>H04N 19/186</b>	(2014.01)
<b>H04N 19/157</b>	(2014.01)
<b>H04N 19/91</b>	(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **18.01.2013 E 17165914 (7)**

97 Fecha y número de publicación de la concesión europea: **30.10.2019 EP 3270591**

54 Título: **Codificación modificada para un bloque omitido de transformación para CABAC en HEVC**

30 Prioridad:

19.01.2012 US 201213354272  
 27.01.2012 US 201213360615  
 02.02.2012 US 201213365215  
 11.04.2012 US 201213444710  
 26.04.2012 US 201213457272

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**11.06.2020**

73 Titular/es:

**HUAWEI TECHNOLOGIES CO., LTD. (100.0%)**  
**Huawei Administration Building, Bantian,**  
**Longgang District**  
**Shenzhen, Guangdong 518129, CN**

72 Inventor/es:

**KIM, SEUNG-HWAN;**  
**MISRA, KIRAN;**  
**KEROFSKY, LOUIS JOSEPH y**  
**SEGALL, CHRISTOPHER A.**

74 Agente/Representante:

**ELZABURU, S.L.P**

ES 2 765 807 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**

Codificación modificada para un bloque omitido de transformación para CABAC en HEVC

**Campo técnico**

5 La presente descripción se refiere de manera general a dispositivos electrónicos. Más específicamente, la presente descripción se refiere a dispositivos electrónicos que utilizan Codificación Aritmética Binaria Adaptativa al Contexto (CABAC) mejorada para codificación y/o decodificación.

**Antecedentes de la técnica**

10 Muchos decodificadores (y codificadores) reciben (y los codificadores proporcionan) datos codificados para bloques de una imagen. Típicamente, la imagen se divide en bloques y cada uno de los bloques se codifica de alguna manera, tal como usando una transformada de coseno discreta (DCT), y se proporciona al decodificador. Un bloque puede denotar una región rectangular en una imagen y consta de píxeles, por ejemplo, un bloque de 16x16 es una región de 16x píxeles de anchura y 16x píxeles de altura. El decodificador recibe los bloques codificados y decodifica cada uno de los bloques de alguna manera, tal como usando una transformada de coseno discreta inversa.

15 Los estándares de codificación de video, tales como MPEG-4 parte 10 (H.264), comprimen datos de video para la transmisión sobre un canal con ancho de banda de frecuencia limitado y/o capacidad de almacenamiento limitada. Estos estándares de codificación de video incluyen múltiples etapas de codificación, tales como intra predicción, transformación del dominio espacial al dominio de frecuencia, cuantificación, codificación de entropía, estimación de movimiento y compensación de movimiento, con el fin de codificar y decodificar cuadros de manera más efectiva.

20 El Equipo Colaborativo Conjunto sobre Codificación de Video (JCT-VC) de la Unión Internacional de Telecomunicaciones del Sector de Estandarización de Telecomunicaciones (UIT-T) Grupo de Estudio 16 (SG16) Grupo de Trabajo 3 (WP3) y Organización Internacional de Normalización/Comisión Electrotécnica Internacional (ISO/IEC) Comité Técnico Conjunto 1/Subcomité 29/Grupo de Trabajo 11 (JTC1/SC29/WG11) ha lanzado un esfuerzo de estandarización para un estándar de codificación de video llamado el estándar de Codificación de Video de Alta Eficiencia (HEVC). Similar a algunos estándares de codificación de video anteriores, HEVC es una codificación basada en bloques. Un ejemplo de un codificador HEVC conocido se muestra en la FIG. 1. Los decodificadores HEVC también son conocidos.

30 En HEVC, se usa Codificación Aritmética Binaria Adaptativa al Contexto (CABAC) para comprimir los Coeficientes Transformados y Cuantificados (TQC) sin pérdidas. Los TQC se determinan en el codificador procesando bloques de imágenes con una transformación directa para generar coeficientes de transformación que luego se cuantifican usando una operación que asigna múltiples valores de coeficientes de transformación a valores de TQC. Los valores de TQC entonces se comunican al decodificador como valores de Nivel de Coeficiente, o valores de nivel, y el valor de nivel para cada coeficiente entonces se asigna a un valor de coeficiente de transformación que es similar, pero no necesariamente idéntico, al valor del coeficiente de transformación calculado en el codificador. La técnica de codificación y/o decodificación basada en CABAC es generalmente adaptativa al contexto, lo que se refiere a (i) 35 codificar de manera adaptativa símbolos en base a los valores de símbolos anteriores codificados y/o decodificados en el pasado, y (ii) contexto, que identifica el conjunto de símbolos codificados y/o decodificados en el pasado usados para adaptación. Los símbolos pasados se pueden situar en bloques adyacentes espaciales y/o temporales. En muchos casos, el contexto se basa en valores de símbolos de bloques vecinos.

40 Como se ha mencionado anteriormente, CABAC se puede usar para comprimir los TQC sin pérdidas. A modo de antecedentes, los TQC pueden ser de diferentes tamaños de bloque según los tamaños de transformación (por ejemplo, 4x4, 8x8, 16x16, 32x32, 16x32). Los TQC bidimensionales (2D) se pueden convertir en una agrupación unidimensional (1D) antes de la codificación de entropía. En un ejemplo, los TQC agrupados en 2D en un bloque 4x4 se pueden disponer como se ilustra en la Tabla (1).

Tabla (1)

4	0	1	0
3	2	-1	...
-3	0	...	...
0	...	...	...

45 Cuando se convierten los TQC de 2D en una agrupación 1D, el bloque se puede escanear en forma de zigzag diagonal. Continuando con el ejemplo, los TQC agrupados en 2D ilustrados en la Tabla (1) se pueden convertir en los TQC de agrupación 1D [4, 0, 3, -3, 2, 1, 0, -1, 0, ...] escaneando la primera fila y la primera columna, la primera fila y la segunda columna, la segunda fila y la primera columna, la tercera fila y la primera columna, la segunda fila y la segunda

columna, la primera fila y la tercera columna, la primera fila y la cuarta columna, la segunda fila y la tercera columna, la tercera fila y la segunda columna, la cuarta fila y la primera columna y así sucesivamente.

La agrupación 1D de los TQC se representa por una secuencia de Elementos de Sintaxis (SE) en CABAC. Un ejemplo de la secuencia de SE para el ejemplo de agrupación 1D de los TQC se muestra en la FIG. 2. Los SE representan los siguientes parámetros: Última posición X/Y, Mapa de Importancia y los atributos Mayor que 1, Mayor que 2, Información de Signo y Absoluto -3. La última posición X/Y representa la posición (X/Y) del último coeficiente distinto de cero en el bloque correspondiente. El mapa de importancia representa la importancia de cada coeficiente. Mayor que 1 indica si la amplitud del coeficiente es mayor que uno para cada coeficiente distinto de cero (es decir, con bandera de importancia como 1). Mayor que 2 indica si la amplitud del coeficiente es mayor que dos para cada coeficiente con una amplitud mayor que uno (es decir, con bandera mayor que 1 como 1).

En CABAC en HEVC, los SE representativos se codifican. La FIG. 3 muestra la estructura de CABAC usada para codificar los SE. La técnica de codificación CABAC incluye símbolos de codificación que usan etapas. En la primera etapa, la CABAC usa un "convertidor a binario" para asignar símbolos de entrada a una cadena de símbolos binarios, o "contenedores". El símbolo de entrada puede ser un símbolo de valor no binario que se convierte a binario o de otro modo se convierte en una cadena de símbolos binarios (1 o 0) antes de ser codificado con bits. Los contenedores se pueden codificar con bits usando o bien un "motor de codificación por derivación" o bien un "motor de codificación habitual".

Para el motor de codificación habitual en CABAC, en la segunda etapa se selecciona un modelo de probabilidad. El modelo de probabilidad se usa para codificar aritméticamente uno o más contenedores de los símbolos de entrada convertidos a binario. Este modelo se puede seleccionar de una lista de modelos de probabilidad disponibles dependiendo del contexto, que es una función de los símbolos codificados recientemente. El modelo de probabilidad almacena la probabilidad de que un contenedor sea "1" o "0". En la tercera etapa, un codificador aritmético codifica cada contenedor según el modelo de probabilidad seleccionado. Hay dos intervalos secundarios para cada contenedor, que corresponden a un "0" y a un "1". La cuarta etapa implica actualizar el modelo de probabilidad. El modelo de probabilidad seleccionado se actualiza en base al valor de contenedor codificado real (por ejemplo, si el valor de contenedor fuera un "1", el recuento de frecuencia de los "1" aumenta). La técnica de decodificación para la decodificación CABAC invierte el proceso.

Para el motor de codificación por derivación en CABAC, la segunda etapa implica la conversión de contenedores a bits, omitiendo las etapas de actualización de estimación y probabilidad de contexto computacionalmente costosas. El motor de codificación por derivación supone una distribución de probabilidad fija para los contenedores de entrada. La técnica de decodificación para la decodificación CABAC invierte el proceso.

La CABAC codifica los símbolos conceptualmente usando dos etapas. En la primera etapa, la CABAC realiza una conversión a binario de los símbolos de entrada a los contenedores. En la segunda etapa, la CABAC realiza una conversión de los contenedores a bits usando o bien el motor de codificación por derivación o bien el motor de codificación habitual. Los valores de bits codificados resultantes se proporcionan en el flujo de bits a un decodificador.

La CABAC decodifica los símbolos conceptualmente usando dos etapas. En la primera etapa, la CABAC usa o bien el motor de decodificación por derivación o bien el motor de decodificación habitual para convertir los bits de entrada en valores de contenedor. En la segunda etapa, la CABAC realiza la desconversión de binario para recuperar el valor del símbolo transmitido para los valores de contenedor. El símbolo recuperado puede ser no binario en su naturaleza. El valor del símbolo recuperado se usa en los aspectos restantes del decodificador.

Como se ha descrito anteriormente, el proceso de codificación y/o decodificación de la CABAC incluye al menos dos modos diferentes de operación. En un primer modo, el modelo de probabilidad se actualiza en base al valor de contenedor codificado real, al que se hace referencia generalmente como "modo de codificación habitual". El modo de codificación habitual requiere varias operaciones en serie secuenciales junto con su complejidad de cálculo asociada y un tiempo significativo para completarse. En un segundo modo, el modelo de probabilidad no se actualiza en base al valor de contenedor codificado real, al que se hace referencia generalmente como "modo de codificación por derivación". En el segundo modo, no existe un modelo de probabilidad (distinto quizás una probabilidad fija) para decodificar los contenedores y, por consiguiente, no hay necesidad de actualizar el modelo de probabilidad.

Cuando se utiliza codificación CABAC en HEVC, el rendimiento de la capacidad de procesamiento puede diferir dependiendo de diferentes factores, tales como, pero no limitado a: número total de contenedores/píxeles, número de contenedores/píxeles por derivación y número de contenedores/píxeles codificados habituales (o de contexto). En términos generales, la capacidad de procesamiento para el caso de codificación de alta tasa de bits (valor del Parámetro de Cuantificación (QP) bajo) es significativamente menor que la capacidad de procesamiento en otros casos. Por lo tanto, la capacidad de procesamiento en casos de alta tasa de bits puede consumir una cantidad significativa de recursos de procesamiento y/o tardar una cantidad de tiempo significativa en codificar/decodificar. La descripción que sigue resuelve este y otros problemas.

También es sabido que CABAC se puede usar en un modo de codificación sin pérdidas para comprimir una muestra residual. En un ejemplo, una muestra residual es un valor que corresponde a una localización específica en una

imagen. Típicamente, una muestra residual corresponde a la diferencia entre un valor correspondiente a una localización específica en una imagen y un valor de predicción correspondiente a la misma localización específica en una imagen. Alternativamente, una muestra residual es un valor correspondiente a una localización específica en una imagen que no ha sido procesada con una operación de transformación, o una operación de transformación que no se usa típicamente para crear los TQC. Una muestra residual puede ser de diferentes tamaños de bloque según su tamaño de muestra (4x4, 8x8, 16x16, 32x32, 16x32, etc.) Un bloque de muestra residual de 2D se convierte primero en una agrupación 1D antes de la codificación de entropía, similar a la codificación de TQC. En un ejemplo, una muestra residual agrupada en 2D en un bloque 4x4 se puede disponer como se ilustra en la Tabla (2).

Tabla (2)

4	0	1	0
3	2	-1	...
-3	0	...	...
0	...	...	...

10 Cuando se convierte la muestra residual 2D en una agrupación 1D, el bloque se puede escanear en forma de zigzag diagonal. Continuando con el ejemplo, la muestra residual agrupada 2D ilustrada en la Tabla (2) se puede convertir en una muestra residual agrupada 1D [4, 0, 3, -3, 2, 1, 0, -1, 0, ...] escaneando la primera fila y la primera columna, la primera fila y la segunda columna, la segunda fila y la primera columna, la tercera fila y la primera columna, la segunda fila y la segunda columna, la primera fila y la tercera columna, la primera fila y la cuarta columna, la segunda fila y la tercera columna, la tercera fila y la segunda columna, la cuarta fila y la primera columna, y así sucesivamente.

15 La agrupación 1D de la muestra residual se representa por una secuencia de Elementos de Sintaxis (SE) en CABAC. Un ejemplo de una secuencia de SE para el ejemplo de agrupación 1D de la muestra residual se muestra en la FIG. 11. Los SE representan los siguientes parámetros: Última posición X/Y, Mapa de Importancia y los atributos Mayor que 1, Mayor que 2, Información de Signo y Absoluto -3.

20 En el modo de codificación sin pérdidas de CABAC en HEVC, se codifican los SE representativos. La estructura de CABAC de la FIG. 3 se puede usar para codificar los SE. La técnica de codificación CABAC incluye codificar símbolos usando etapas. En la primera etapa, la CABAC usa un “convertidor a binario” para asignar símbolos de entrada a una cadena de símbolos binarios, o “contenedores”. El símbolo de entrada puede ser un símbolo de valor no binario que se convierte a binario o se convierte de otro modo en una cadena de símbolos binarios (1 o 0) antes de ser codificado en bits. Los contenedores se pueden codificar en bits usando el “motor de codificación habitual” descrito anteriormente.

25 Para el motor de codificación habitual en el modo de codificación sin pérdidas de CABAC, en la segunda etapa se selecciona un modelo de probabilidad (también conocido como “modelo de contexto” en el modo de codificación sin pérdidas de CABAC). El modelo se usa para codificar aritméticamente uno o más contenedores de los símbolos de entrada convertidos a binario. Este modelo se puede seleccionar de una lista de modelos disponibles dependiendo del contexto, que es una función de símbolos codificados recientemente. El modelo almacena la probabilidad de que un contenedor sea “1” o “0”. En la tercera etapa, un codificador aritmético codifica cada contenedor según el modelo seleccionado. Hay dos intervalos secundarios para cada contenedor, que corresponden a un “0” y a un “1”. La cuarta etapa implica actualizar el modelo. El modelo seleccionado se actualiza en base al valor de contenedor codificado real (por ejemplo, si el valor de contenedor fuera un “1”, el recuento de frecuencia de los “1” aumenta). La técnica de decodificación para la decodificación CABAC invierte el proceso.

30 El número de modelos usados como se describe en el párrafo anterior puede ser de 184. Específicamente: 36 modelos usados para la Última posición X/Y (18 modelos para Última\_posición\_X, 18 modelos para Última\_posición\_Y); 48 modelos utilizados para Mapa de Importancia (bloque 4x4: 9 luma, 6 croma; bloque 8x8: 11 luma, 11 croma; bloque 16x16 o 32x32: 7 luma, 4 croma); y 100 modelos usados para los atributos Mayor que 1, Mayor que 2, Información de Signo y Absoluto -3 (bandera Mayor\_que\_1 de luma: 30; bandera Mayor\_que\_1 de croma 20, bandera Mayor\_que\_2 de luma: 30; y bandera Mayor\_que\_2 de croma 20).

35 Por ejemplo, Tung Nguyen et al: “Reduced-complexity entropy coding of transform coefficient levels using truncated golomb-rice codes in video compression” Procesamiento de imágenes (ICIP), 2011, 18ª Conferencia Internacional del IEEE, IEEE, se refiere a una combinación de códigos simples de longitud variable y codificación binaria adaptativa al contexto, que produce la misma eficiencia de codificación que la codificación de coeficiente de transformación H.2647AVC a un nivel de complejidad más bajo y que se ha adoptado en el modelo de prueba de HEVC.

40 Cuando se utiliza la codificación CABAC en HEVC en el modo de codificación sin pérdidas, la codificación/decodificación es computacionalmente compleja. Una razón para la complejidad de cálculo es el uso de 184 modelos, como se ha explicado anteriormente. Debido a esta complejidad de cálculo, la codificación/decodificación puede consumir una cantidad significativa de recursos de procesamiento y/o puede tardar una cantidad de tiempo significativa en completarse. La descripción que sigue resuelve este y otros problemas.

**Compendio de la invención**

En un primer aspecto, se proporciona un método para decodificar un flujo de bits asociado con Coeficientes Transformados y Cuantificados, TQC, en HEVC, el método que comprende:

- obtener el flujo de bits;
- 5     • obtener N TQC de un bloque del flujo de bits, en donde el bloque tiene un tamaño de bloque que tienen los N TQC;
- determinar un número de TQC que no es igual a cero entre los N TQC;
- determinar si el número de TQC que no es igual a cero del bloque es mayor que un umbral preestablecido, en donde el umbral preestablecido es el 50% de N;
- 10    • si el número de TQC que no es igual a cero del bloque es mayor que el umbral preestablecido:
  - decodificar los TQC no iguales a cero con una decodificación por derivación omitiendo una estimación de contexto y una actualización de probabilidad para la Codificación Aritmética Binaria Adaptativa al Contexto, CABAC
- si el número de TQC que no es igual a cero del bloque no es mayor que el umbral preestablecido:
  - 15     ○ decodificar los TQC no iguales a cero con una decodificación habitual usando la estimación de contexto y la actualización de probabilidad para la CABAC.

**Breve descripción de los dibujos**

La Figura 1 es un diagrama de bloques de un codificador HEVC.

La Figura 2 es una tabla que muestra una secuencia de elementos de sintaxis según CABAC.

- 20    La Figura 3 es un diagrama de bloques de la estructura de CABAC para una secuencia de elementos de sintaxis.

La Figura 4 es un diagrama de bloques que ilustra un ejemplo de un codificador y un decodificador.

La Figura 5 es un diagrama de flujo que ilustra una configuración de un método para el modo de conversión a binario de alta capacidad de procesamiento en un dispositivo electrónico.

- 25    La Figura 6 es un diagrama de flujo que ilustra una configuración de un procesamiento de codificador que usa el modo de conversión a binario de alta capacidad de procesamiento.

La Figura 7 es un diagrama de flujo que ilustra una configuración de un método para el modo de conversión a binario de alta capacidad de procesamiento en un dispositivo electrónico en el lado de decodificación.

La Figura 8 es un diagrama de flujo que ilustra una configuración de un procesamiento de decodificador que usa el modo de conversión a binario de alta capacidad de procesamiento.

- 30    La Figura 9 ilustra una tabla de asignación que se puede usar para determinar un valor de entrada en modo de conversión a binario de alta capacidad de procesamiento.

La Figura 10 ilustra una pluralidad de tablas de conversión a binario que se pueden usar para conversión a binario adaptativa en modo de conversión a binario de alta capacidad de procesamiento.

- 35    La Figura 11 es una tabla que muestra una secuencia de elementos de sintaxis según un modo de codificación sin pérdidas en CABAC.

La Figura 12 es un diagrama de bloques que ilustra un ejemplo de un codificador y un decodificador para una técnica de codificación sin pérdidas.

La Figura 13 es un diagrama de flujo que ilustra una configuración de un método para codificación sin pérdidas en un dispositivo electrónico.

- 40    La Figura 14 es una tabla que muestra una secuencia de elementos de sintaxis según la configuración ilustrada en la Figura 13.

La Figura 15 es un diagrama de flujo que ilustra una configuración de un método para decodificación sin pérdidas en un dispositivo electrónico en el lado de decodificación.

La Figura 16 es un diagrama de flujo que ilustra otra configuración de un método para codificación sin pérdidas en un

dispositivo electrónico.

La Figura 17 es un diagrama de flujo que ilustra otra configuración de un método para codificación sin pérdidas en un dispositivo electrónico en el lado de decodificación.

5 La Figura 18 es un diagrama de flujo que ilustra otra configuración más de un método para codificación sin pérdidas en un dispositivo electrónico.

La Figura 19 es un diagrama de flujo que ilustra otra configuración más de un método para codificación sin pérdidas en un dispositivo electrónico en el lado de decodificación.

La Figura 20A es un diagrama de flujo que ilustra configuraciones de ejemplo de un codificador o un decodificador para determinar si se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento.

10 La Figura 20B es un diagrama de flujo que ilustra configuraciones de ejemplo de un codificador o un decodificador para determinar si se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento.

La Figura 20C es un diagrama de flujo que ilustra configuraciones de ejemplo de un codificador o un decodificador para determinar si se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento.

15 La Figura 20D es un diagrama de flujo que ilustra configuraciones de ejemplo de un codificador o un decodificador para determinar si se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento.

La Figura 20E es un diagrama de flujo que ilustra configuraciones de ejemplo de un codificador o un decodificador para determinar si se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento.

La Figura 21 es un diagrama de flujo que ilustra una configuración de un método para determinar si se cumple una condición de modo de alta capacidad de procesamiento en un dispositivo electrónico en el lado de decodificación.

20 La Figura 22 es un diagrama de flujo que ilustra otra configuración de un método para determinar si se cumple una condición de modo de alta capacidad de procesamiento en un dispositivo electrónico en el lado de decodificación.

La Figura 23 es un diagrama de bloques que ilustra un ejemplo de un codificador y un decodificador.

La Figura 24 es un diagrama de flujo que ilustra una configuración de un método para decodificación de mapas de importancia de alta capacidad de procesamiento en un dispositivo electrónico en el lado de decodificación.

25 La Figura 25 es un diagrama de flujo que ilustra otra configuración de un método para la decodificación de mapas de importancia de alta capacidad de procesamiento en un dispositivo electrónico en el lado de decodificación.

La Figura 26 es un diagrama de flujo que ilustra una configuración de un método para la decodificación de mapas de importancia de alta capacidad de procesamiento con un rasgo de decodificación por derivación en un dispositivo electrónico en el lado de decodificación.

30 La Figura 27 es un diagrama de flujo que ilustra una configuración de un método para decodificación de mapas de importancia de alta capacidad de procesamiento con un rasgo de conmutación de método de decodificación en un dispositivo electrónico en el lado de decodificación.

La Figura 28 es una tabla usada para actualizar un parámetro de Rice según un modo de codificación sin pérdidas en CABAC.

35 La Figura 29 es un diagrama de bloques que ilustra un ejemplo de un codificador y un decodificador.

La Figura 30 es un diagrama de flujo que ilustra una configuración de un método para codificación sin pérdidas con una selección diferente de parámetros en un dispositivo electrónico.

La Figura 31 es un diagrama de flujo que ilustra una configuración de un método para codificación sin pérdidas con una selección diferente de parámetros en un dispositivo electrónico en el lado de decodificación.

40 La Figura 32 es un elemento de sintaxis de ejemplo generado según CABAC.

La Figura 33 es un diagrama de bloques que ilustra un ejemplo de un codificador y un decodificador.

La Figura 34 es un diagrama de flujo que ilustra una configuración de un método para codificación de alta capacidad de procesamiento para CABAC en HEVC en un dispositivo electrónico.

45 La Figura 35 es un diagrama de flujo que ilustra una configuración de un método para la codificación de alta capacidad de procesamiento para CABAC en HEVC en un dispositivo electrónico en el lado de decodificación.

La Figura 36 es un elemento de sintaxis de ejemplo generado según la configuración de la Figura 34.

La Figura 37 es una ilustración de una tabla de actualización de parámetros de Rice reducida.

La Figura 38 es un diagrama de flujo que ilustra una configuración de un método para codificación de alta capacidad de procesamiento para CABAC en HEVC en un dispositivo electrónico en el lado de decodificación.

5 La Figura 39 es un diagrama de flujo que ilustra una configuración de un método para codificación de alta capacidad de procesamiento para CABAC en HEVC en un dispositivo electrónico en el lado de decodificación.

La Figura 40 es un diagrama de flujo que ilustra una configuración de un método para la codificación de alta capacidad de procesamiento para CABAC en HEVC en un dispositivo electrónico en el lado de decodificación.

### Descripción de realizaciones

La Figura 4 es un diagrama de bloques que ilustra un ejemplo de un codificador y un decodificador.

10 El sistema 400 incluye un codificador 411 para generar bloques codificados para ser decodificados por un decodificador 412. El codificador 411 y el decodificador 412 pueden comunicarse sobre una red.

15 El codificador 411 incluye un dispositivo electrónico 421 configurado para codificar usando un modo de conversión a binario de alta capacidad de procesamiento. El dispositivo electrónico 421 puede comprender un procesador y una memoria en comunicación electrónica con el procesador, donde la memoria almacena instrucciones que son ejecutables por el procesador para realizar las operaciones mostradas en las Figuras 5 y 6.

El decodificador 412 incluye un dispositivo electrónico 422 configurado para decodificar usando el modo de conversión a binario de alta capacidad de procesamiento. El dispositivo electrónico 422 puede comprender un procesador y una memoria en comunicación electrónica con el procesador, donde la memoria almacena instrucciones que son ejecutables para realizar las operaciones mostradas en las Figuras 7 y 8.

20 La Figura 5 es un diagrama de flujo que ilustra una configuración de un método para el modo de conversión a binario de alta capacidad de procesamiento en un dispositivo electrónico.

25 En el bloque 511, el dispositivo electrónico 421 obtiene un bloque de coeficientes transformados y cuantificados (TQC). En el rombo 512, el dispositivo electrónico 421 determina si se cumple una condición de modo de conversión a binario de alta capacidad de procesamiento. Si la condición no se cumple en el rombo 512, entonces en el bloque 513 el dispositivo electrónico 421 codifica el bloque usando selectivamente un modo de codificación habitual y un modo de codificación por derivación (según los esquemas de selección de CABAC convencionales).

30 Si la condición se cumple en el rombo 512, entonces en el bloque 514 el dispositivo electrónico 421 usa el modo de conversión a binario de alta capacidad de procesamiento y el modo de codificación por derivación para codificar el bloque. El dispositivo electrónico 421 transmite el flujo de bits generado sobre una red y/o almacena el flujo de bits generado en un dispositivo de memoria en el bloque 515.

El modo HTB usa el modo de codificación por derivación para codificar los valores de nivel. A diferencia del modo de codificación habitual, la codificación por derivación omite las etapas de estimación de contexto y de actualización de probabilidad computacionalmente costosas, debido a que el modo de codificación por derivación supone una distribución de probabilidad fija para los contenedores de entrada.

35 Además de usar el modo de codificación por derivación para la codificación, a modo de contraste con la CABAC convencional, el modo HTB usa una estructura de firma simplificada para la codificación. Por ejemplo, CABAC convencional requiere cuatro partes secundarias para la codificación, incluyendo Mayor\_que\_1, Mayor\_que\_2, Información de Signo, y Absoluto -3.

40 La Figura 6 es un diagrama de flujo que ilustra una configuración de procesamiento de codificador que usa el modo de conversión a binario de alta capacidad de procesamiento.

45 Los bloques 612-615 ilustran las operaciones realizadas en el bloque 514 con más detalle. En el bloque 612, el dispositivo electrónico 421 genera información de signo y nivel para cualquier valor distinto de cero a partir del bloque de TQC aplicando una función de valor absoluto menos uno a cada valor distinto de cero y comprobando el signo de cada valor distinto de cero. Por facilidad de explicación, consideremos los valores para el TQC agrupado en 1D de la sección de antecedentes de la presente solicitud [4, 0, 3, -3, 2, 1, 0, -1, 0, ...]. La aplicación de la función de valor absoluto menos uno a cada valor distinto de cero y la comprobación del signo de cada valor distinto de cero genera seis combinaciones de información de signo y nivel de la siguiente manera: +3, +2, -2, +1, +0 y -0.

50 En el bloque 613, el dispositivo electrónico 421 asigna un valor de entrada a cada combinación generada de información de signo y nivel usando una tabla de asignación. Un ejemplo de tabla de asignación se muestra en la FIG. 9. La FIG. 9 también muestra una ecuación para determinar un valor de entrada según los bloques 612 y 613.

En el bloque 614, el dispositivo electrónico 421 realiza una conversión a binario adaptativa de los valores de entrada usando una pluralidad de tablas de conversión a binario, por ejemplo, las tablas VLC de Codificación de Longitud

Variable Adaptativa al Contexto (CAVLC). Un ejemplo de las tablas VLC de CAVLC se muestra en la FIG. 10. La FIG. 10 también muestra una ecuación para actualizar las tablas de conversión a binario en base a la información de entrada anterior.

5 En un ejemplo, el bloque 614 puede incluir usar inicialmente valores de la columna Tabla VLC 0 (FIG. 10) para convertir a binario al menos el primer valor de entrada. El número de la tabla VLC se puede aumentar de forma monótona cuando un valor anterior es mayor que los valores umbral dados, por ejemplo, 3, 5, 13, 27. Por consiguiente, la conversión a binario adaptativa posterior después del primer aumento monótono puede usar valores de la columna Tabla VLC 1, la conversión a binario adaptativa posterior después del segundo aumento monótono puede usar valores de la columna Tabla VLC 2, etc.

10 En el bloque 615, el dispositivo electrónico 421 codifica los valores resultantes de la conversión a binario adaptativa usando el modo de codificación por derivación de CABAC.

(Condición de modo de conversión a binario de alta capacidad de procesamiento)

15 En un ejemplo, si una característica correspondiente a un bloque de datos de imagen es mayor que un umbral preestablecido, entonces se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento, por ejemplo, el dispositivo electrónico 421 puede establecer un indicador de modo de conversión a binario de alta capacidad de procesamiento, por ejemplo, una bandera de modo HTB, a un valor de 1 (que por supuesto puede incluir cambiar un valor por defecto de la bandera de modo HTB o dejar la bandera de modo HTB en un valor por defecto dependiendo de la preferencia de diseño).

20 En un ejemplo, el dispositivo electrónico 421 determina si una tasa de bits para una codificación es mayor que un umbral preestablecido. Si la tasa de bits es mayor que el umbral preestablecido, entonces se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento. En un ejemplo, el umbral de tasa de bits preestablecido corresponde a QP 16; no obstante, se puede usar un umbral preestablecido correspondiente a diferentes valores de QP.

25 En un ejemplo, la determinación (por el dispositivo electrónico 421 o el dispositivo electrónico 422) de si se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento se basa en si el nivel de la unidad de transformación (por ejemplo, pero no limitado a los valores de nivel generados por una unidad de transformación) de un bloque correspondiente de datos de imagen es mayor que un umbral preestablecido.

30 En un ejemplo, la condición de modo de conversión a binario de alta capacidad de procesamiento se puede cumplir cuando el número de valores de nivel de un bloque correspondiente de datos de imagen y con magnitud mayor que cero es mayor que un umbral preestablecido, por ejemplo, 8. En otro ejemplo, la condición de modo de conversión a binario de alta capacidad de procesamiento se cumple cuando el número de valores de nivel de un bloque correspondiente de datos de imagen y con magnitud mayor que un primer umbral preestablecido es mayor que un segundo umbral preestablecido. En un ejemplo más, la condición de modo de conversión a binario de alta capacidad de procesamiento se cumple cuando un valor de nivel de un bloque correspondiente de datos de imagen es mayor que un umbral preestablecido.

35 Las FIG. 20A-E muestran algunos ejemplos de configuraciones que se pueden usar para un codificador o decodificador en sistemas de ejemplo que operan según al menos algunos de los principios descritos en los dos párrafos inmediatamente anteriores. La FIG. 20A ilustra los procesos 1611-1616, como se muestra. La FIG. 20B ilustra los procesos 1711-1716, como se muestra. La FIG. 20C ilustra los procesos 1801-1805 y 1814-1820. La FIG. 20D ilustra los procesos 1901-1905 y 1914-1920. En la FIG. 20E, los procesos de la FIG. 20C se realizan hasta el proceso 1816, como se muestra. Si el contador es mayor que el umbral en el proceso 1816, entonces la configuración continúa como se muestra en la FIG. 20E.

45 En un ejemplo, la determinación (por el dispositivo electrónico 421 o el dispositivo electrónico 422) de si se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento se basa en si el nivel de segmento de un bloque correspondiente de datos de imagen es mayor que un umbral preestablecido.

(Indicador de modo de conversión a binario de alta capacidad de procesamiento)

50 En un ejemplo, el dispositivo electrónico 421 se configura para establecer un indicador de conversión a binario de alta capacidad de procesamiento, por ejemplo, una bandera de modo HTB, en una cabecera, por ejemplo, la cabecera de segmento. El indicador de conversión a binario de alta capacidad de procesamiento se puede usar para determinar si el proceso mostrado en la FIG. 5 se ejecuta o no para el bloque o bloques correspondientes a la cabecera de segmento.

55 En un ejemplo, establecer la bandera de modo HTB en "1" hace que el dispositivo electrónico 421 realice el proceso mostrado en el diagrama de flujo de la FIG. 5 para el bloque o bloques correspondientes a la cabecera de segmento en respuesta a la observación del valor de la bandera de modo HTB de "1". Establecer la bandera de modo HTB en "0" hace que el dispositivo electrónico 421 codifique el bloque o bloques correspondientes a la cabecera de segmento según una técnica de CABAC convencional en respuesta a la observación del valor de la bandera de modo HTB de "0".

5 El valor de la bandera de modo HTB también se puede observar por el dispositivo electrónico 422 para la decodificación. En un ejemplo, el dispositivo electrónico 422 decodifica el bloque o los bloques correspondientes a una cabecera de segmento que tiene el valor de la bandera de modo HTB de "1" según el proceso mostrado en el diagrama de flujo de la FIG. 7 para el bloque o bloques correspondientes a la cabecera de segmento en respuesta a la observación del valor de la bandera de modo HTB de "1". El dispositivo electrónico 422 decodifica el bloque o bloques correspondientes a una cabecera de segmento que tiene el valor de la bandera de modo HTB de "0" según una técnica de CABAC convencional en respuesta a la observación del valor de la bandera de modo HTB de "0".

La Figura 7 es un diagrama de flujo que ilustra una configuración de un método para el modo de conversión a binario de alta capacidad de procesamiento en un dispositivo electrónico en el lado de decodificación.

10 En el bloque 710, el dispositivo electrónico 422 obtiene un flujo de bits. En el bloque 711, el dispositivo electrónico 422 recupera un símbolo binario del flujo de bits obtenido.

15 En el rombo 712, el dispositivo electrónico 422 determina si se cumple una condición de modo de conversión a binario de alta capacidad de procesamiento. En un ejemplo, la determinación puede incluir comprobar una cabecera, tal como una cabecera de segmento, correspondiente al flujo de bits recibido. Comprobar la cabecera puede comprender además comprobar una cabecera de segmento correspondiente al flujo de bits obtenido para un valor de un indicador de modo de conversión a binario de alta capacidad de procesamiento. Si la condición no se cumple en el rombo 712, entonces en el bloque 713 el dispositivo electrónico 422 decodifica el símbolo binario usando selectivamente el modo de decodificación habitual y el modo de codificación por derivación.

20 Si la condición se cumple en el rombo 712, entonces en el bloque 714 el dispositivo electrónico 421 usa el modo de conversión a binario de alta capacidad de procesamiento y el modo de decodificación por derivación para decodificar el símbolo binario. El dispositivo electrónico 422 puede almacenar un bloque obtenido de los TQC en un dispositivo de memoria y/o puede recuperar datos de video en el bloque 715.

La Figura 8 es un diagrama de flujo que ilustra una configuración de procesamiento de decodificador que usa el modo de conversión a binario de alta capacidad de procesamiento.

25 Los bloques 812-815 ilustran operaciones realizadas en el bloque 714 con más detalle. En el bloque 812, el dispositivo electrónico 422 decodifica por derivación el símbolo binario codificado. En el bloque 813, el dispositivo electrónico 422 desconvierte de binario un resultado de la decodificación por derivación. En el bloque 814, el dispositivo electrónico 422 asigna valores de entrada recuperados de la desconversión de binario a información de signo y nivel usando una tabla de asignación. En el bloque 815, el dispositivo electrónico 422 decodifica un bloque de coeficientes transformados y cuantificados (TQC) usando la información de signo y nivel.

30 En un ejemplo, se proporciona un dispositivo electrónico que incluye un procesador y una memoria en comunicación electrónica con el procesador. Almacenadas en la memoria están las instrucciones ejecutables por el procesador para realizar operaciones.

35 En un ejemplo, una operación puede incluir la obtención de un bloque de coeficientes transformados y cuantificados (TQC). Otra operación puede incluir determinar si se cumple una condición de modo de conversión a binario de alta capacidad de procesamiento. Otra operación puede incluir generar un primer flujo de bits usando el modo de conversión a binario de alta capacidad de procesamiento en respuesta a la determinación de que se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento. Otra operación puede incluir generar un segundo flujo de bits en respuesta a la determinación de que no se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento. Otra operación puede incluir transmitir el primer o segundo flujo de bits generado a un decodificador.

40 En un ejemplo, la generación del primer flujo de bits usando el modo de conversión a binario de alta capacidad de procesamiento puede incluir operaciones adicionales. Una operación puede incluir generar información de signo y nivel para cualquier valor distinto de cero del bloque aplicando una función valor absoluto menos uno a cada valor distinto de cero y comprobar el signo de cada valor distinto de cero. Otra operación puede incluir la asignación de un valor de entrada a cada combinación generada de información de signo y nivel usando una tabla de asignación. Otra operación puede incluir realizar una conversión a binario adaptativa de los valores de entrada asignados usando una pluralidad de tablas de conversión a binario. Otra operación puede incluir codificar un resultado de la conversión a binario adaptativa.

50 En un ejemplo, la pluralidad de tablas de conversión a binario incluye tablas VLC de CAVLC. La codificación del resultado de la conversión a binario adaptativa puede incluir además la operación de utilizar un modo de codificación por derivación de CABAC.

55 En un ejemplo, la conversión a binario adaptativa de los valores de entrada asignados usando una pluralidad de tablas de conversión a binario puede incluir operaciones adicionales. Una operación puede incluir determinar si uno de los valores de entrada asignados es mayor que un umbral preestablecido. Otra operación puede incluir realizar una actualización de la tabla en respuesta a la determinación de que dicho valor de entrada asignado es mayor que el umbral preestablecido. En un ejemplo, la selección de actualización de tabla comprende la selección de una tabla a

partir de un conjunto de tablas.

5 En un ejemplo, la generación del primer flujo de bits puede incluir operaciones adicionales. Una operación puede incluir la codificación del bloque utilizando selectivamente un modo de codificación habitual y un modo de codificación por derivación según CABAC. Otra operación puede incluir generar el primer flujo de bits utilizando solamente el modo de codificación por derivación.

En un ejemplo, la determinación de si se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento se basa en si una característica correspondiente a un bloque de datos de imagen es mayor que un umbral preestablecido.

10 En un ejemplo, la determinación de si se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento se basa en si el nivel de segmento de un bloque correspondiente de datos de imagen es mayor que un umbral preestablecido.

En un ejemplo, la determinación de si se cumple la condición de modo de conversión a binario de alta capacidad de procesamiento se basa en si el nivel de la unidad de transformación de un bloque correspondiente de datos de imagen es mayor que un umbral preestablecido.

15 (Técnica de codificación sin pérdidas para CABAC en HEVC)

La Figura 12 es un diagrama de bloques que ilustra un ejemplo de un codificador y un decodificador para una técnica de codificación sin pérdidas.

El sistema 1400 incluye un codificador 1411 para generar bloques codificados para ser decodificados por un decodificador 1412. El codificador 1411 y el decodificador 1412 pueden comunicarse sobre una red.

20 El codificador 1411 incluye un dispositivo electrónico 1421 configurado para codificar usando una técnica de codificación sin pérdidas para CABAC en HEVC. El dispositivo electrónico 1421 puede comprender un procesador y una memoria en comunicación electrónica con el procesador, donde la memoria almacena instrucciones que son ejecutables por el procesador para realizar las operaciones mostradas en las Figuras 13, 16 y 18.

25 El decodificador 1412 incluye un dispositivo electrónico 1422 configurado para decodificar usando una técnica de codificación sin pérdidas para CABAC en HEVC. El dispositivo electrónico 1422 puede comprender un procesador y una memoria en comunicación electrónica con el procesador, donde la memoria almacena instrucciones que son ejecutables para realizar las operaciones mostradas en las Figuras 15, 17 y 19.

La Figura 13 es un diagrama de flujo que ilustra una configuración de un método para codificación sin pérdidas en un dispositivo electrónico.

30 En el bloque 911, el dispositivo electrónico 1421 obtiene un bloque que representa una muestra residual. En un ejemplo, la dirección de escaneado en zigzag se puede redefinir para ajustarse a la dirección de intra predicción que se usa para eliminar las redundancias espaciales entre píxeles vecinos. Hay varios modos intra predicción disponibles en el modo intra codificación sin pérdidas. En un ejemplo, en el modo de intra predicción vertical, los píxeles superiores llegan a ser el valor de predicción del valor de píxel actual y la diferencia entre el valor actual y el valor de predicción (valor de píxel superior en modo vertical) llega a ser el valor de muestra residual. La selección del modelo de contexto también puede depender de la dirección de la intra predicción y del tamaño de bloque correspondiente.

35 En el bloque 912, el dispositivo electrónico 1421 genera un mapa de importancia a ser usado en una secuencia de elementos de sintaxis. En el bloque 913, el dispositivo electrónico 1421 rellena un campo de mapa de importancia que corresponde a la última posición de escaneado del bloque con un valor correspondiente al nivel de la última posición del bloque.

40 En el bloque 914, el dispositivo electrónico 1421 genera una secuencia de elementos de sintaxis que incluye el mapa de importancia que tiene dicho valor. La generación de la secuencia de elementos de sintaxis excluye la etapa de codificación de la última posición del modo de codificación sin pérdidas de CABAC convencional.

45 La Figura 14 es una tabla que muestra una secuencia de elementos de sintaxis según la configuración ilustrada en la Figura 13.

50 Se pueden observar varias diferencias por medio de contraste de la secuencia de elementos de sintaxis mostrados en la FIG. 14 en comparación con la secuencia de elementos de sintaxis mostrada en la FIG. 11. La secuencia de elementos de sintaxis mostrada en la FIG. 11 incluye un campo Última\_posición\_X y un campo Última\_posición\_Y debido a que el modo de codificación sin pérdidas de CABAC convencional incluye una etapa de codificación de la última posición. Por el contrario, la secuencia de elementos de sintaxis mostrada en la FIG. 14 no incluye un campo Última\_posición\_X y un campo Última\_posición\_Y debido a que la configuración de la FIG. 14 omite la etapa de codificación de última posición.

Aunque ambas de las secuencias de elementos de sintaxis incluyen mapas de importancia, hay diferencias entre los

mapas de importancia. En el mapa de importancia de la secuencia de elementos de sintaxis de la FIG. 11, un campo de mapa de importancia no está relleno para corresponder con el campo de Última\_posición\_X/Última\_posición\_Y que se rellena. Por el contrario, en la FIG. 14 un campo de mapa de importancia que corresponde a la última posición de escaneado del bloque se rellena con un valor, es decir, "0" para el bloque de ejemplo, correspondiente al nivel de la última posición del bloque.

La Figura 15 es un diagrama de flujo que ilustra una configuración de un método para decodificación sin pérdidas en un dispositivo electrónico en el lado de decodificación.

En el bloque 1011, el dispositivo electrónico 1422 recupera, a partir de un flujo de bits, una secuencia de elementos de sintaxis que tienen un campo de mapa de importancia que contiene un número de valores que corresponden a una última posición de escaneado de un bloque. En el bloque 1012, el dispositivo electrónico 1422 decodifica los niveles del bloque usando el mapa de importancia y usando dicho valor del mapa de importancia. En el bloque 1013, el dispositivo electrónico 1422 almacena un bloque obtenido correspondiente a un valor residual en un dispositivo de memoria y/o recupera datos de video.

La Figura 16 es un diagrama de flujo que ilustra otra configuración de un método para codificación sin pérdidas en un dispositivo electrónico.

En el bloque 1111, el dispositivo electrónico 1421 obtiene una secuencia de elementos de sintaxis que representa información de nivel para un bloque de una muestra residual. En el bloque 1112, el dispositivo electrónico 1421 realiza una conversión a binario adaptativa sobre valores de la parte Absoluto -3 de la secuencia de elementos de sintaxis usando una pluralidad de tablas de conversión a binario, por ejemplo, las tablas VLC de CAVLC (FIG. 10), en el que los valores de la parte Absoluto -3 de la secuencia de elementos de sintaxis se usan como valores de entrada para la pluralidad de tablas de conversión a binario. Se muestra a continuación una ecuación para actualizar las tablas de conversión a binario en base a información de entrada anterior:

$$\text{si}(\text{'abs[coeficiente}(i)\text{'-3' > (Tabla[vlc])}) \text{ vlc: ++;}$$

donde Tabla[vlc] = {3,5,13,27};

'i' representa la posición de escaneado y 'vlc' representa el número de la tabla vlc actual

\*vlc se establece primero a cero (o uno para intra segmento) debido a que no hay disponible una Tabla vlc 'Absoluto -3' anterior actualizada se detiene cuando vlc es igual a 4

En un ejemplo, el bloque 1111 puede incluir inicialmente usar valores de la columna Tabla VLC 0 (FIG. 10) para el inter segmento y la columna Tabla VLC 1 para el intra segmento para convertir a binario al menos el primer valor de entrada. El número de la tabla VLC se puede aumentar de forma monótona cuando un valor anterior es mayor que los valores umbral dados, por ejemplo, 3, 5, 13, 27. Por consiguiente, la conversión a binario adaptativa posterior después del primer aumento monótono puede usar valores de la columna Tabla VLC 1, la conversión a binario adaptativa posterior después del segundo aumento monótono puede usar valores de la columna Tabla VLC 2, etc.

En el bloque 1113, el dispositivo electrónico 1421 codifica los valores resultantes de la conversión a binario adaptativa usando el modo de codificación por derivación de CABAC.

La Figura 17 es un diagrama de flujo que ilustra otra configuración de un método para la codificación sin pérdidas en un dispositivo electrónico en el lado de decodificación.

En el bloque 1211, el dispositivo electrónico 1422 recupera un símbolo binario a partir de un flujo de bits. En el bloque 1212, el dispositivo electrónico 1422 decodifica por derivación el símbolo binario. En el bloque 1213, el dispositivo electrónico 1422 desconvierte de binario de manera adaptativa un resultado de la decodificación por derivación. En el bloque 1214, el dispositivo electrónico 1422 recupera un bloque que representa información residual usando un resultado de la desconversión de binario adaptativa.

La Figura 18 es un diagrama de flujo que ilustra otra configuración más de un método para codificación sin pérdidas en un dispositivo electrónico.

En el bloque 1311, el dispositivo electrónico 1421 accede solamente a un subconjunto de los modelos de contexto de CABAC. El número de modelos de contexto de CABAC puede ser de 184. Para generar el subconjunto, estos modelos de contexto se pueden filtrar en base a las características asociadas de los modelos de contexto, por ejemplo, en base a qué modelos de contexto están asociados con un componente de frecuencia, en base a qué modelos de contexto están asociados con una posición de escaneado, en base a qué modelos de contexto están asociados con la etapa de codificación de última posición de CABAC, o similar, o cualquier combinación de los mismos. El filtrado se puede realizar por el dispositivo electrónico 1421 en un ejemplo, pero en otros ejemplos, el subconjunto se puede proporcionar al dispositivo electrónico 1421 de modo que el dispositivo electrónico 1421 pueda acceder al subconjunto proporcionado para el modo de codificación sin pérdidas. En un ejemplo, con el fin de generar el subconjunto, los modelos de contexto de CABAC se pueden clasificar en base a las características asociadas de los modelos de

contexto, por ejemplo, en base a qué modelos de contexto están asociados con un componente de frecuencia, en base a qué modelos de contexto están asociados con una posición de escaneado, en base a qué modelos de contexto están asociados con la etapa de codificación de última posición de CABAC, o similar, o cualquier combinación de los mismos. En un ejemplo, el componente de frecuencia y la posición de escaneado pueden ser iguales e intercambiables.

En un ejemplo, un subconjunto puede no contener modelos de contexto de CABAC con un componente de frecuencia no igual a un primer componente de frecuencia. En un ejemplo, el subconjunto resultante incluiría 26 modelos de contexto, es decir, dos modelos de contexto (uno es para un primer componente de frecuencia luma y el otro es para un primer componente de frecuencia croma) para codificar el mapa de importancia, y 6 modelos de contexto para codificar el primer componente de frecuencia luma de la bandera Mayor\_que\_1, codificando el primer componente de frecuencia croma de la bandera Mayor\_que\_1, codificando el primer componente de frecuencia luma de la bandera Mayor\_que\_2 de luma, y codificando el primer componente de frecuencia croma de la bandera Mayor\_que\_2, respectivamente. Por lo tanto, se usan un total de 24 modelos de contexto para Mayor\_que\_1 y Mayor\_que\_2. En un ejemplo, solamente se accede a dicho primer componente de frecuencia cuando se codifica el mapa de importancia en el bloque 1312.

Como se muestra en la Tabla (3), los 6 modelos de contexto para codificar la primera frecuencia luma de una bandera pueden depender del tipo de bloque secundario y del valor LargerT1, donde el valor LargerT1 es el número de valores de nivel de coeficiente mayores que uno en el bloque secundario anterior. En un ejemplo, el término "bloque secundario" se refiere a una partición de las muestras residuales (o bloque a los TQC). Por ejemplo, para un tamaño de bloque secundario de 4x4, la muestra residual con un tamaño de 8x8 se divide en cuatro bloques secundarios de 4x4. De manera similar, para un tamaño de bloque secundario de 8x4, las muestras residuales con un tamaño de 32x8 se dividen en ocho bloques secundarios de 8x4. Los bloques secundarios se identifican por orden de codificación, donde el bloque secundario 0 denota el primer bloque secundario codificado. En un ejemplo, el primer bloque secundario codificado es el bloque secundario situado en la parte inferior derecha del bloque. En otro ejemplo, el primer bloque secundario codificado es el bloque secundario situado en el medio del bloque.

Tabla (3)

6 modelos de contexto en base al tipo de bloque secundario y a LargerT1		
0 1 2	Bloque secundario 0	0 LargerT1 en subconjunto previo 1-3 LargerT1 en subconjunto previo >3 LargerT1 en subconjunto previo
3 4 5	Otros bloques secundarios	0 LargerT1 en subconjunto previo 1-3 LargerT1 en subconjunto previo >3 LargerT1 en subconjunto previo

En el bloque 1312, el dispositivo electrónico 1421 usa el subconjunto de los modelos de contexto para codificar el mapa de importancia.

Como se describe en el párrafo 0120, los modelos de contexto con un componente de frecuencia (o posición de escaneado) no igual a un primer componente de frecuencia (o posición de escaneado) no se pueden usar en un modo de codificación sin pérdidas. Esto tiene el beneficio de reducir la complejidad de cálculo y la memoria para el modo de codificación sin pérdidas. Se puede usar un primer subconjunto de modelos de contexto para el procesamiento de mapas de importancia. Se puede usar un segundo subconjunto de modelos de contexto para codificación de nivel, por ejemplo, codificación Mayor\_que\_1 y/o codificación Mayor\_que\_2. El primer subconjunto puede ser diferente del segundo subconjunto.

En un ejemplo que aplica al menos algunos de los principios descritos anteriormente, el primer subconjunto de modelos de contexto usados en el procesamiento de mapas de importancia puede comprender solamente un modelo de contexto. En otro ejemplo que aplica al menos algunos de los principios descritos anteriormente, el primer subconjunto de modelos de contexto usados en el procesamiento de mapas de importancia puede comprender más de un modelo de contexto, por ejemplo, dos o tres modelos de contexto, basados en la información de color (luma/croma). En otro ejemplo más que aplica al menos algunos de los principios descritos anteriormente, el primer subconjunto de modelos de contexto usados en el procesamiento de mapas de importancia puede comprender más de un modelo de contexto, por ejemplo, varios modelos de contexto, basados en el tipo de predicción, por ejemplo, el uso de predicción dentro de un cuadro o entre cuadros dentro de un bloque. En otro ejemplo que aplica al menos algunos de los principios descritos anteriormente, el primer subconjunto de modelos de contexto usados en el procesamiento de mapas de importancia puede comprender más de un modelo de contexto, por ejemplo, dos o tres modelos de contexto, basados en el tamaño de bloque. En otro ejemplo que aplica al menos algunos de los principios descritos anteriormente, el primer subconjunto de modelos de contexto usados en el procesamiento de mapas de importancia puede comprender más de un modelo de contexto, por ejemplo, dos o tres modelos de contexto, basados en el tipo de bloque secundario.

En un ejemplo que aplica al menos algunos de los principios descritos anteriormente, el segundo subconjunto de modelos de contexto usados en codificación de nivel puede comprender solamente un modelo de contexto. En otro ejemplo que aplica al menos algunos de los principios descritos anteriormente, el segundo subconjunto de modelos de contexto usados en codificación de nivel puede comprender más de un modelo de contexto, por ejemplo, dos o tres modelos de contexto, basados en información de color (luma/croma). En otro ejemplo más que aplica al menos algunos de los principios descritos anteriormente, el segundo subconjunto de modelos de contexto usados en codificación de nivel puede comprender más de un modelo de contexto, por ejemplo, varios modelos de contexto, basados en el tipo de predicción de bloque, por ejemplo, el uso de predicción dentro de un cuadro o entre cuadros dentro de un bloque. En otro ejemplo que aplica al menos algunos de los principios descritos anteriormente, el primer subconjunto de modelos de contexto usados en la codificación de nivel puede comprender más de un modelo de contexto, por ejemplo, dos o tres modelos de contexto, basados en el tamaño de bloque. En otro ejemplo que aplica al menos algunos de los principios descritos anteriormente, el primer subconjunto de modelos de contexto usados en procesamiento de codificación de nivel puede comprender más de un modelo de contexto, por ejemplo, dos o tres modelos de contexto, basados en el tipo de bloque secundario.

La Figura 19 es un diagrama de flujo que ilustra otra configuración más de un método para codificación sin pérdidas en un dispositivo electrónico en el lado de decodificación.

En el bloque 1511, el dispositivo electrónico 1422 accede solamente a un subconjunto de los modelos de contexto de CABAC. En el bloque 1512, el dispositivo electrónico 1422 recupera un símbolo binario a partir de un flujo de bits usando el subconjunto de modelos de contexto. En el bloque 1513, el dispositivo electrónico 1422 recupera datos de video usando un resultado de la decodificación.

En lo anterior, las configuraciones que se pueden implementar mediante el dispositivo electrónico 1421 se ilustran en las FIG. 13, 16 y 18. La configuración de un codificador con todas estas configuraciones mejora el rendimiento de la codificación en comparación con el modo de codificación sin pérdidas de CABAC conocido. Sin embargo, la configuración de un codificador con menos de todas estas configuraciones en cualquier combinación, por ejemplo, una de estas configuraciones o dos cualesquiera de estas configuraciones, también es posible y práctico, y también mejora el rendimiento de codificación en comparación con el modo de codificación sin pérdidas de CABAC conocido.

En lo anterior, las configuraciones que se pueden implementar mediante el dispositivo electrónico 1422 se ilustran en las FIG. 14, 17 y 19. La configuración de un decodificador con todas estas configuraciones mejora el rendimiento de codificación en comparación con el modo de codificación sin pérdidas de CABAC conocido. Sin embargo, la configuración de un decodificador con menos de todas estas configuraciones en cualquier combinación, por ejemplo, una de estas configuraciones o dos cualesquiera de estas configuraciones, también es posible y práctico, y también mejora el rendimiento de codificación en comparación con el modo de codificación sin pérdidas de CABAC conocido.

En un ejemplo, se proporciona un dispositivo electrónico que incluye un procesador y una memoria en comunicación electrónica con el procesador. Almacenadas en la memoria están las instrucciones ejecutables por el procesador para realizar operaciones.

En un ejemplo, una operación puede incluir la obtención de un bloque que representa una muestra residual para codificación sin pérdidas. Otra operación puede incluir generar un mapa de importancia, en donde la generación incluye rellenar un campo de mapa de importancia que corresponde a la última posición de escaneo del bloque con un valor correspondiente a un nivel de la última posición de escaneo del bloque. Otra operación puede incluir generar una secuencia de elementos de sintaxis que incluya el mapa de importancia que tiene el valor. Otra operación puede incluir transmitir un flujo de bits que representa la secuencia generada de elementos de sintaxis a un decodificador.

En un ejemplo, la secuencia de elementos de sintaxis se genera sin realizar la etapa de codificación de última posición de la Codificación Aritmética Binaria Adaptativa al Contexto (CABAC).

En un ejemplo, otra operación puede incluir realizar una conversión a binario adaptativa usando una pluralidad de tablas de conversión a binario, en donde los valores de una parte Absoluto -3 de la secuencia de elementos de sintaxis se usan como valores de entrada para la pluralidad de tablas de conversión a binario. Otra operación puede incluir codificar un resultado de la conversión a binario adaptativa. La pluralidad de tablas de conversión a binario pueden ser tablas VLC de CAVLC.

En un ejemplo, codificar el resultado de la conversión a binario adaptativa puede incluir operaciones adicionales. Una operación adicional puede incluir utilizar un modo de codificación por derivación de CABAC.

En un ejemplo, la conversión a binario adaptativa de los valores de entrada usando la pluralidad de tablas de conversión a binario puede incluir operaciones adicionales. Una operación adicional puede incluir determinar si uno de los valores de entrada es mayor que un umbral preestablecido. Una operación adicional puede incluir realizar una actualización de tabla en respuesta a determinar que dicho valor de entrada es mayor que el umbral preestablecido.

En un ejemplo, otra operación puede incluir acceder solamente a un subconjunto de los modelos de contexto de CABAC. Otra operación puede incluir usar al subconjunto de los modelos de contexto para codificar el mapa de importancia. El subconjunto puede comprender los modelos de contexto de CABAC con un componente de frecuencia

no igual a una primera frecuencia.

En un ejemplo, se proporciona un dispositivo electrónico que incluye un procesador y una memoria en comunicación electrónica con el procesador. Almacenadas en la memoria están las instrucciones ejecutables por el procesador para realizar operaciones.

- 5 En un ejemplo, una operación puede incluir obtener un bloque que representa una muestra residual para codificación sin pérdidas. Otra operación puede incluir generar una secuencia de elementos de sintaxis para representar el bloque. Otra operación puede incluir realizar una conversión a binario adaptativa usando una pluralidad de tablas de conversión a binario, en donde los valores de una parte Absoluto -3 de la secuencia de elementos de sintaxis se usan como valores de entrada para la pluralidad de tablas de conversión a binario. Otra operación puede incluir codificar un resultado de la conversión a binario adaptativa. Otra operación puede incluir la transmisión de la codificación a un decodificador.
- 10

En un ejemplo, la pluralidad de tablas de conversión a binario son tablas VLC de CAVLC.

- 15 En un ejemplo, codificar el resultado de la conversión a binario adaptativa puede incluir operaciones adicionales. Una operación adicional puede incluir utilizar un modo de codificación por derivación de Codificación Aritmética Binaria Adaptativa al Contexto (CABAC).

En un ejemplo, la conversión a binario adaptativa de los valores de entrada usando la pluralidad de tablas de conversión a binario puede incluir operaciones adicionales. Una operación adicional puede incluir determinar si uno de los valores de entrada es mayor que un umbral preestablecido. Una operación adicional puede incluir realizar una actualización de tabla en respuesta a determinar que dicho valor de entrada es mayor que el umbral preestablecido.

- 20 En un ejemplo, otra operación puede incluir generar un mapa de importancia, en donde la generación incluye rellenar un campo de mapa de importancia que corresponde a la última posición de escaneado del bloque con un valor correspondiente a un nivel de la última posición de escaneado del bloque. Otra operación puede incluir generar la secuencia de elementos de sintaxis usando el mapa de importancia generado.

- 25 En un ejemplo, la secuencia de elementos de sintaxis se genera sin realizar la etapa de codificación de última posición de CABAC.

- 30 En un ejemplo, se proporciona un método. El método se puede realizar usando un decodificador. Una operación del método puede incluir filtrar los modelos de contexto de la Codificación Aritmética Binaria Adaptativa al Contexto (CABAC) en base a qué modelos de contexto están asociados con el componente de frecuencia. Otra operación del método puede incluir obtener un flujo de bits. Otra operación del método puede incluir recuperar un símbolo binario del flujo de bits. Otra operación del método puede incluir decodificar el símbolo binario usando los modelos de contexto filtrados. Otra operación del método puede incluir recuperar datos de video usando un resultado de la decodificación.

- 35 En un ejemplo, otra operación puede incluir recuperar, a partir del flujo de bits, una secuencia de elementos de sintaxis que tienen un mapa de importancia rellenado con un valor correspondiente a una última posición de escaneado de un bloque que representa una muestra residual. Otra operación puede incluir decodificar los niveles del bloque usando el mapa de importancia y usando dicho valor del mapa de importancia.

En un ejemplo, la decodificación de los niveles del bloque se puede realizar sin realizar la etapa de decodificación de última posición de CABAC.

- 40 En un ejemplo, otra operación puede incluir decodificar por derivación del símbolo binario recuperado. Otra operación puede incluir desconvertir de binario adaptativamente un resultado de la decodificación por derivación. Otra operación puede incluir recuperar un bloque que representa información residual usando un resultado de la desconversión de binario.

En un ejemplo, otra operación puede incluir usar una pluralidad de tablas VLC de CAVLC para la desconversión de binario adaptativa.

En un ejemplo, la decodificación por derivación puede incluir utilizar un modo de decodificación por derivación CABAC.

- 45 La Figura 21 es un diagrama de flujo que ilustra una configuración de un método para determinar si se cumple una condición de modo de alta capacidad de procesamiento en un dispositivo electrónico en el lado de decodificación.

En el bloque 2611, el dispositivo electrónico 422 obtiene un flujo de bits. En el bloque 2612, el dispositivo electrónico 422 obtiene un bloque de valores de nivel. En un ejemplo, el bloque comprende un bloque de TQC.

- 50 En el bloque 2613, el dispositivo electrónico 422 determina el número de valores de nivel que no son iguales a cero. En el rombo 2614, el dispositivo electrónico 422 determina si el número es mayor que un umbral preestablecido. En un ejemplo, el umbral preestablecido puede ser 8, que es la mitad del número de valores de un bloque 4x4. En ejemplos con un tamaño de bloque que tiene N valores de nivel, el umbral puede corresponder al 50% de N. En un ejemplo, el dispositivo electrónico 422 recibe señalización del dispositivo electrónico 421. La señalización transmitida por el

dispositivo electrónico 421 puede especificar el umbral preestablecido o incluir información que se puede usar por el dispositivo electrónico 421 para determinar el umbral preestablecido.

5 Si el número no es mayor que el umbral preestablecido en el rombo 2614, entonces en el bloque 2615 el dispositivo electrónico 422 decodifica valores de nivel no iguales a cero con un primer método de conversión a binario. Si el número es mayor que el umbral preestablecido en el rombo 2614, entonces en el bloque 2616 el dispositivo electrónico 422 decodifica valores de nivel que no son iguales a cero con un segundo método de conversión a binario que es diferente al primer método de conversión a binario. En un ejemplo, el segundo método de conversión a binario puede comprender un modo de desconversión de binario de alta capacidad de procesamiento, tal como el modo HTB descrito anteriormente. En un ejemplo, el primer método de conversión a binario puede comprender la conversión a binario de CABAC convencional.

La Figura 22 es un diagrama de flujo que ilustra otra configuración de un método para determinar si se cumple una condición de modo de alta capacidad de procesamiento en un dispositivo electrónico en el lado de decodificación.

En el bloque 2711, el dispositivo electrónico 422 obtiene un flujo de bits. En el bloque 2712, el dispositivo electrónico 422 obtiene un bloque de valores de nivel. En un ejemplo, el bloque comprende un bloque de TQC.

15 En el bloque 2713, el dispositivo electrónico 422 determina el número de valores de nivel con un valor absoluto mayor que un primer umbral preestablecido. En un ejemplo, el primer umbral preestablecido puede ser o bien 1 o bien 2, aunque se pueden usar otros primeros umbrales preestablecidos en otros ejemplos. En el rombo 2714, el dispositivo electrónico 422 determina si el número es mayor que un segundo umbral preestablecido. En un ejemplo, el segundo umbral preestablecido puede ser 8, que es la mitad del número de valores de un bloque 4x4. En ejemplos con un tamaño de bloque que tiene N valores de nivel, el segundo umbral preestablecido puede corresponder al 50% de N.

20 En un ejemplo, el dispositivo electrónico 422 recibe señalización del dispositivo electrónico 421. La señalización transmitida por el dispositivo electrónico 421 puede especificar el primer umbral preestablecido y/o el segundo umbral preestablecido, o incluir información que se puede usar por el dispositivo electrónico 421 para determinar el primer umbral preestablecido y/o el segundo umbral preestablecido.

25 Si el número no es mayor que el segundo umbral preestablecido en el rombo 2714, entonces en el bloque 2715 el dispositivo electrónico 422 decodifica valores de nivel no iguales a cero con un primer método de conversión a binario. Si el número es mayor que el umbral preestablecido en el rombo 2714, entonces en el bloque 2716 el dispositivo electrónico 422 decodifica valores de nivel no iguales a cero con un segundo método de conversión a binario que es diferente del primer método de conversión a binario. En un ejemplo, el segundo método de conversión a binario puede comprender un modo de desconversión de binario de alta capacidad de procesamiento, tal como el modo HTB descrito anteriormente. En un ejemplo, el primer método de conversión a binario puede comprender la conversión a binario de la decodificación de CABAC conocida.

La Figura 23 es un diagrama de bloques que ilustra un ejemplo de un codificador y un decodificador.

35 El sistema 2400 incluye un codificador 2411 para generar bloques codificados para ser decodificados por un decodificador 2412. El codificador 2411 y el decodificador 2412 pueden comunicarse sobre una red.

El decodificador 2412 incluye un dispositivo electrónico 2422 configurado para decodificar usando el procesamiento de mapa de importancia de alta capacidad de procesamiento. El dispositivo electrónico 2422 puede comprender un procesador y una memoria en comunicación electrónica con el procesador, donde la memoria almacena instrucciones que son ejecutables para realizar las operaciones mostradas en las Figuras 24-27.

40 El codificador 2411 incluye un dispositivo electrónico 2421, que puede comprender un procesador y una memoria en comunicación electrónica con el procesador, donde la memoria almacena instrucciones que son ejecutables por el procesador para realizar operaciones que serán entendidas por un experto en la técnica a partir de la descripción de las configuraciones mostradas en las Figuras 24-27 y la descripción correspondiente de las mismas.

45 La Figura 24 es un diagrama de flujo que ilustra una configuración de un método para decodificación de mapas de importancia de alta capacidad de procesamiento en un dispositivo electrónico en el lado de decodificación.

En el bloque 2801, el dispositivo electrónico 2422 obtiene un flujo de bits. En el bloque 2802, el dispositivo electrónico 2422 obtiene un bloque de valores de nivel. En un ejemplo, el bloque comprende un bloque de TQC. En el bloque 2803, el dispositivo electrónico 2422 obtiene un valor de nivel del bloque, por ejemplo, el primer valor del nivel del bloque o un siguiente valor de nivel del bloque.

50 En el rombo 2804, el dispositivo electrónico 2422 determina si el valor de nivel obtenido es el último valor de nivel del bloque. Si el valor de nivel obtenido no es el último valor de nivel en el rombo 2804, entonces el dispositivo electrónico 2422 pasa al rombo 2814. Si el valor de nivel obtenido es el último valor de nivel en el rombo 2804, entonces en el bloque 2805 el dispositivo electrónico 2422 decodifica magnitudes de los valores de nivel (que pueden incluir determinar tanto la información de signo como la magnitud absoluta para cada valor de nivel).

Con referencia de nuevo al rombo 2814, el dispositivo electrónico 2422 determina si el valor de nivel obtenido no es cero usando un primer método de decodificación. Si el valor de nivel obtenido no es cero en el rombo 2814, el dispositivo electrónico 2422 pasa al bloque 2815; de otro modo, el dispositivo electrónico 2422 vuelve al bloque 2803. En el bloque 2815, el dispositivo electrónico 2422 aumenta un contador.

- 5 En el rombo 2816, el dispositivo electrónico 2422 determina si un recuento actual del contador es mayor que un umbral preestablecido. En un ejemplo, el umbral preestablecido puede comprender el umbral preestablecido descrito con referencia a la FIG. 21. Si el recuento actual del contador es mayor que el umbral preestablecido en el rombo 2816, el dispositivo electrónico 2422 pasa al bloque 2817; de otro modo, el dispositivo electrónico 2422 vuelve al bloque 2803.

- 10 En el bloque 2817, el dispositivo electrónico 2422 obtiene el siguiente valor de nivel del bloque. En el rombo 2818, el dispositivo electrónico 2422 determina si el valor de nivel obtenido es el último valor de nivel del bloque. Si el valor de nivel obtenido no es el último valor de nivel en el rombo 2818, entonces el dispositivo electrónico 2422 pasa al bloque 2819; de otro modo, en el bloque 2820 el dispositivo electrónico 2422 decodifica las magnitudes de los valores de nivel.

- 15 En el bloque 2819, el dispositivo electrónico 2422 determina si el valor de nivel obtenido no es cero usando un segundo método de decodificación que es diferente del primer método de decodificación. En un ejemplo, el segundo método de decodificación comprende un método de decodificación de alta capacidad de procesamiento, un método de decodificación por derivación o similar. En un ejemplo, el primer método de decodificación comprende el modo de decodificación habitual de CABAC.

- 20 Según lo anterior, un mapa de importancia se puede decodificar elemento por elemento, por ejemplo, campo de mapa de importancia por campo de mapa de importancia. Cuando se alcanza el umbral preestablecido, el dispositivo electrónico 2422 puede cambiar la decodificación de la parte de mapa de importancia restante. Se puede usar un modo de decodificación de mapa de importancia de alta capacidad de procesamiento o de derivación para la parte de mapa de importancia restante. Por lo tanto, el rendimiento de decodificación se puede mejorar sobre la decodificación de mapa de importancia de CABAC convencional.

- 25 La Figura 25 es un diagrama de flujo que ilustra otra configuración de un método para la decodificación de mapas de importancia de alta capacidad de procesamiento en un dispositivo electrónico en el lado de decodificación.

- 30 En el método mostrado en la FIG. 25, los procesos 2901-2905 se pueden realizar como se muestra, similar a los procesos 2801-2805 (FIG. 24). En el rombo 2914, el dispositivo electrónico 2422 determina si el valor absoluto del valor de nivel obtenido es mayor que un primer umbral usando un primer método de decodificación. En un ejemplo, el primer umbral puede ser 1 o 2, aunque se pueden usar otros primeros umbrales en otros ejemplos. Si el valor absoluto del valor de nivel obtenido es mayor que el primer umbral en el rombo 2914, el dispositivo electrónico 2422 pasa al bloque 2915; de otro modo, el dispositivo electrónico 2422 vuelve al bloque 2903. En el bloque 2915, el dispositivo electrónico 2422 aumenta un contador.

- 35 En el rombo 2916, el dispositivo electrónico 2422 determina si un recuento actual del contador es mayor que un segundo umbral preestablecido. En un ejemplo, el segundo umbral preestablecido puede ser 8, que es la mitad del número de valores de un bloque 4x4. En ejemplos con un tamaño de bloque que tiene N valores de nivel, el segundo umbral preestablecido puede corresponder al 50% de N. Si el recuento actual del contador es mayor que el segundo umbral en el rombo 2916, el dispositivo electrónico 2422 pasa al bloque 2917; de otro modo, el dispositivo electrónico 2422 vuelve al bloque 2903.

- 40 En el bloque 2917, el dispositivo electrónico 2422 obtiene el siguiente valor de nivel del bloque. En el rombo 2918, el dispositivo electrónico 2422 determina si el valor de nivel obtenido es el último valor de nivel del bloque. Si el valor de nivel obtenido no es el último valor de nivel en el rombo 2918, entonces el dispositivo electrónico 2422 pasa al bloque 2919; de otro modo, en el bloque 2920 el dispositivo electrónico 2422 decodifica las magnitudes de los valores de nivel.

- 45 En el bloque 2919, el dispositivo electrónico 2422 determina si el valor absoluto del valor de nivel obtenido es mayor que el primer umbral usando un segundo método de decodificación que es diferente del primer método de decodificación. En un ejemplo, el segundo método de decodificación comprende un método de decodificación de alta capacidad de procesamiento, un método de decodificación por derivación o similar. En un ejemplo, el primer método de decodificación comprende el modo de decodificación habitual de CABAC.

- 50 Según lo anterior, un mapa de importancia se puede decodificar elemento por elemento, por ejemplo, campo de mapa de importancia por campo de mapa de importancia. Cuando se alcanza el umbral preestablecido, el dispositivo electrónico 2422 puede cambiar la decodificación de la parte de mapa de importancia restante. Se puede usar un modo de decodificación de mapa de importancia de alta capacidad de procesamiento o de derivación para la parte de mapa de importancia restante. Por lo tanto, el rendimiento de decodificación se puede mejorar sobre la decodificación de mapa de importancia de CABAC convencional.
- 55

La Figura 26 es un diagrama de flujo que ilustra una configuración de un método para la decodificación de mapas de importancia de alta capacidad de procesamiento con un rasgo de decodificación por derivación en un dispositivo

electrónico en el lado de decodificación.

En el método mostrado en la FIG. 26, los procesos 3001-3004 y 3014-3016 se pueden realizar como se muestra, similar a los procesos 2801-2804 y 2814-2816 (FIG. 24). En el bloque 3005, el dispositivo electrónico 2422 recupera las magnitudes de los valores de nivel usando un tercer método de decodificación, por ejemplo, un método de conversión a binario. En el bloque 3020, el dispositivo electrónico 2422 recupera las magnitudes de una primera parte de los valores de nivel usando el tercer método de decodificación, y recupera las magnitudes de una segunda parte de los valores de nivel usando un cuarto método de decodificación, por ejemplo, un método de conversión a binario diferente.

En un ejemplo, la primera parte de los valores de nivel comprende los valores de nivel procesados con el primer método de decodificación. La segunda parte de los valores de nivel comprende los valores de nivel no procesados con el primer método de decodificación.

Debería ser evidente que otras configuraciones de un método para la decodificación de mapas de importancia de alta capacidad de procesamiento con un rasgo de decodificación por derivación en un dispositivo electrónico en el lado de decodificación similar a la configuración mostrada en la Figura 26 pueden ser posibles y prácticas. Por ejemplo, en otra configuración, el dispositivo electrónico 2422 determina si el valor absoluto del valor de nivel obtenido es mayor que un primer umbral preestablecido usando un primer método de decodificación, similar al rombo 2914 (FIG. 25). También, el dispositivo electrónico 2422 determina si el contador es mayor que el segundo umbral preestablecido, similar al rombo 2916 (FIG. 25).

Según lo anterior, un mapa de importancia se puede decodificar elemento por elemento, por ejemplo, campo de mapa de importancia por campo de mapa de importancia. Cuando se alcanza el umbral preestablecido, el dispositivo electrónico 2422 puede parar de decodificar el mapa de importancia (los elementos restantes del mapa de importancia no se decodifican). A partir de entonces, los valores de nivel que corresponden a los elementos decodificados se procesan usando un método de conversión a binario (por ejemplo, un método de conversión a binario que puede enviar el valor cero), al tiempo que los elementos restantes se procesan usando un método de conversión a binario diferente (por ejemplo, un método de conversión a binario que no puede enviar el valor cero). Por lo tanto, el rendimiento de decodificación se puede mejorar sobre la decodificación de mapa de importancia de CABAC convencional.

La Figura 27 es un diagrama de flujo que ilustra una configuración de un método para la decodificación de mapas de importancia de alta capacidad de procesamiento con un rasgo de conmutación del método de decodificación en un dispositivo electrónico en el lado de decodificación.

En el método mostrado en la FIG. 27, los procesos 3801-3804 y 3814-3819 se pueden realizar como se muestra, similar a los procesos 2801-2804 y 2814-2819 (FIG. 24). En el bloque 3805, el dispositivo electrónico 2422 recupera las magnitudes de los valores de nivel usando el tercer método de decodificación (el tercer método de decodificación de la FIG. 26). En el bloque 3820, el dispositivo electrónico 2422 recupera magnitudes de una primera parte de los valores de nivel usando el tercer método de decodificación, y recupera magnitudes de una segunda parte de los valores de nivel usando el cuarto método de decodificación (el cuarto método de decodificación de la FIG. 26). En un ejemplo, la primera parte de los valores de nivel comprende los valores de nivel obtenidos en el bloque 3803, mientras que la segunda parte de los valores de nivel comprende los valores de nivel obtenidos en el bloque 3817.

Debería ser evidente que otras configuraciones de un método para la decodificación de mapas de importancia de alta capacidad de procesamiento con un rasgo de decodificación por derivación en un dispositivo electrónico en el lado de decodificación similar a la configuración mostrada en la Figura 27 pueden ser posibles y prácticas. Por ejemplo, en otra configuración, el dispositivo electrónico 2422 determina si el valor absoluto del valor de nivel obtenido es mayor que un primer umbral preestablecido usando un primer método de decodificación, similar al rombo 2914 (FIG. 25). También, el dispositivo electrónico 2422 determina si el contador es mayor que el segundo umbral preestablecido, similar al rombo 2916 (FIG. 25).

En un ejemplo, se proporciona un primer dispositivo electrónico que incluye un procesador y una memoria en comunicación electrónica con el procesador. Almacenadas en la memoria están las instrucciones ejecutables por el procesador para realizar operaciones.

En un ejemplo, una operación puede incluir recibir un flujo de bits. Otra operación puede incluir obtener un bloque de valores de nivel en base al flujo de bits recibido. Otra operación puede incluir identificar una parte de los valores de nivel según un umbral. Otra operación puede incluir, después de identificar la parte, procesar cualesquiera restantes de los valores de nivel usando un modo de procesamiento de mapa de importancia de alta capacidad de procesamiento. Otra operación puede incluir recuperar datos de video en base al procesamiento.

En un ejemplo, se proporciona un segundo dispositivo electrónico que incluye un procesador y una memoria en comunicación electrónica con el procesador. Almacenadas en la memoria están las instrucciones ejecutables por el procesador para realizar operaciones. Una operación puede incluir transmitir señalización al primer dispositivo electrónico, en donde la señalización identifica el umbral.

Según lo anterior, un mapa de importancia se puede decodificar elemento por elemento, por ejemplo, campo de mapa de importancia por campo de mapa de importancia. Cuando se alcanza el umbral preestablecido, el dispositivo electrónico 2422 puede parar de decodificar el mapa de importancia (los elementos restantes del mapa de importancia no se decodifican). A partir de entonces, los valores de nivel que corresponden a los elementos decodificados se procesan usando un método de conversión a binario (por ejemplo, un método de conversión a binario que puede enviar el valor cero), al tiempo que los elementos restantes se procesan usando un método de conversión a binario diferente (por ejemplo, un método de conversión a binario que no puede enviar el valor cero). Por lo tanto, el rendimiento de decodificación se puede mejorar sobre la decodificación de mapa de importancia de CABAC convencional.

5 (Codificación sin pérdidas con técnica de selección de parámetros diferente para CABAC en HEVC).

Cuando se utiliza la codificación CABAC en HEVC en el modo de codificación sin pérdidas, la codificación/decodificación es computacionalmente compleja. Una razón para la complejidad de cálculo es la codificación del elemento de sintaxis "Absoluto -3". En la codificación CABAC conocida, el método de codificación Golomb-Rice exponencial se usa para codificar el elemento de sintaxis.

15 A modo de antecedentes, el método de codificación Golomb-Rice (G-R) exponencial utiliza la tabla de actualización de parámetros de Rice que se muestra en la FIG. 28. El método de codificación G-R se aplica al elemento de sintaxis de código "Absoluto -3" (es decir, la última línea de la tabla de la FIG. 2) en el modo de codificación sin pérdidas de CABAC conocido, como se explica con más detalle en el siguiente párrafo.

20 El parámetro de Rice controla la conversión de símbolos a contenedores. Para ilustrar a modo de ejemplo, consideremos usar la tabla de la FIG. 28 y la codificación G-R para convertir los símbolos 0, 11, 4 ..., donde "0" (el primer símbolo) es el símbolo inicial en un bloque secundario. El parámetro de Rice se inicializa en cero para el primer símbolo, debido a que el primer símbolo es el símbolo inicial en el bloque secundario. El primer símbolo "0" se codifica usando el parámetro de Rice actual de cero. En un ejemplo, el proceso de codificar un símbolo con un parámetro de Rice de RP consiste en calcular el valor Cociente = suelo( (símbolo-1)/RP) y generar una salida que contiene una cadena de contenedores de Cociente igual a 1 seguida de un contenedor igual a 0. En este caso, Cociente es un entero y suelo() es una operación que asigna un valor que contiene un número entero y un componente fraccional al componente entero. Para ilustrar, codificar un símbolo de "5" con el parámetro de Rice 3 daría como resultado un valor de Cociente de 1 y los contenedores de salida de "01". De manera similar, la codificación de un símbolo de "100" con el parámetro de Rice 33 daría como resultado un valor de Cociente de 3 y los contenedores de salida de "0001". En un ejemplo alternativo, el proceso de codificación de un símbolo con un parámetro de Rice de RP consiste en calcular el valor Cociente = suelo( (símbolo-1)/RP) y generar una salida que contiene una cadena de contenedores de Cociente igual a 0 seguido de un contenedor igual a 1. En otro ejemplo más, el proceso de codificación de un símbolo con el parámetro de Rice de RP consiste en seleccionar una tabla de búsqueda de orden RP que defina una asignación entre símbolos y una secuencia de contenedores de un conjunto de tablas de búsqueda. Dado que el resultado de la búsqueda es cero según la tabla de la FIG. 28, el parámetro de Rice no se actualiza para el siguiente símbolo. El segundo símbolo "11" se codifica por lo tanto usando el parámetro de Rice actual de cero. Dado que el resultado de la búsqueda ("2") para el segundo símbolo "11" y el parámetro de Rice "0" es diferente del valor del parámetro de Rice actual (es decir, cero), el parámetro de Rice se actualiza de cero a dos. El tercer símbolo "4" se codifica entonces con el parámetro de Rice actual de dos. Dado que el resultado de la búsqueda no es un valor diferente del parámetro de Rice actual, el parámetro de Rice de dos se usa para el siguiente símbolo.

Debido a la complejidad de cálculo de la codificación G-R del valor "Absoluto -3" según la CABAC conocida, la codificación/decodificación puede consumir una cantidad significativa de recursos de procesamiento y/o puede tardar una cantidad de tiempo significativa en completarse. La descripción que sigue resuelve este y otros problemas.

La Figura 29 es un diagrama de bloques que ilustra un ejemplo de un codificador y un decodificador.

45 El sistema 2900 incluye un codificador 2911 para generar bloques codificados para ser decodificados por un decodificador 2912. El codificador 2911 y el decodificador 2912 pueden comunicarse sobre una red.

50 El codificador 2911 incluye un dispositivo electrónico 2921 configurado para codificar usando una codificación sin pérdidas con una selección de parámetros diferente para CABAC en HEVC. El dispositivo electrónico 2921 puede comprender un procesador y memoria en comunicación electrónica con el procesador, donde la memoria almacena instrucciones que son ejecutables por el procesador para realizar las operaciones mostradas en la Figura 30.

El decodificador 2912 incluye un dispositivo electrónico 2922 configurado para decodificar usando una codificación sin pérdidas con una selección de parámetros diferente para CABAC en HEVC. El dispositivo electrónico 2922 puede comprender un procesador y una memoria en comunicación electrónica con el procesador, donde la memoria almacena instrucciones que son ejecutables para realizar las operaciones mostradas en la Figura 31.

55 La Figura 30 es un diagrama de flujo que ilustra una configuración de un método para codificación sin pérdidas con una selección de parámetros diferente en un dispositivo electrónico.

En el bloque 3011, el dispositivo electrónico 2921 obtiene un bloque de datos para ser codificado usando un codificador

basado en aritmética, por ejemplo, un codificador basado en CABAC. En el rombo 3012, el dispositivo electrónico 2921 determina si el bloque ha de ser codificado usando codificación sin pérdidas. Si el bloque no ha de ser codificado usando codificación sin pérdidas, entonces en el bloque 3013 el dispositivo electrónico 2921 usa una primera técnica de codificación de Absoluto -3 para codificar el bloque de datos.

- 5 Si el bloque ha de ser codificado usando codificación sin pérdidas, entonces en el bloque 3014 el dispositivo electrónico 2921 se usa una segunda técnica de codificación de Absoluto -3 diferente para codificar el bloque de datos. En el bloque 3015, el dispositivo electrónico 2921 transmite el flujo de bits generado sobre una red y/o almacena el flujo de bits generado en un dispositivo de memoria.

- 10 En un ejemplo, la primera técnica de codificación de Absoluto -3 comprende una técnica de codificación R-G de codificación CABAC, es decir, el parámetro de Rice se inicializa a cero en cada etapa de codificación de bloque secundario, y se consideran los cinco parámetros de Rice de la tabla mostrada en la FIG. 28. En un ejemplo, la segunda técnica de codificación de Absoluto -3 diferente no se inicializa a cero en cada etapa de codificación de bloque secundario, es decir, se inicializa de manera diferente y/o usa una tabla de actualización de parámetros de Rice diferente, por ejemplo, una tabla de actualización de parámetros de Rice reducida.

- 15 En un ejemplo, la inicialización diferente puede comprender inicializar el parámetro de Rice a cero en cada bloque y no en cada bloque secundario. En un ejemplo, la inicialización diferente puede comprender usar el último parámetro de Rice usado en un bloque secundario anterior como el parámetro de Rice inicial de un bloque secundario actual.

- 20 En un ejemplo, la inicialización diferente puede comprender inicializar en base a estadísticas de muestras residuales. En un ejemplo, la inicialización diferente puede comprender inicializar en un valor de parámetro de Rice predefinido en base al tipo de bloque, al tamaño de bloque o a la información de color (luma/croma), o similares, o cualquier combinación de los mismos. El tipo de bloque es un valor para representar el bloque en base al tamaño del bloque, a la información de predicción (intra/inter) del bloque y a la información de color del bloque (luma/croma). En un ejemplo, la inicialización diferente puede comprender inicializar el parámetro de Rice en el valor predefinido "1" cuando el tipo de bloque actual es igual a un cierto valor o valores predefinidos, por ejemplo, "2" y/o "5".

- 25 En un ejemplo, la tabla de actualización de parámetros de Rice diferente comprende menos parámetros de Rice que la tabla de actualización de parámetros de Rice usada para la primera técnica de codificación de Absoluto -3. En un ejemplo, la tabla de actualización de parámetros de Rice diferente incluye solamente los dos primeros casos (el parámetro de Rice es igual a "0" y "1"). Una ilustración de tal tabla de actualización de parámetros de Rice se incluye en la FIG. 37).

- 30 En un ejemplo, si se usa la segunda técnica de codificación de Absoluto -3 diferente, el dispositivo electrónico 2921 puede establecer un indicador correspondiente, por ejemplo, una bandera asociada con la segunda técnica de codificación de Absoluto -3 diferente, a un valor de 1 (que por supuesto puede incluir cambiar un valor por defecto de dicha bandera o dejar dicha bandera en un valor por defecto dependiendo de la preferencia de diseño).

- 35 La Figura 31 es un diagrama de flujo que ilustra una configuración de un método para codificación sin pérdidas con una selección de parámetros diferente en un dispositivo electrónico en el lado de decodificación.

En el bloque 3110, el dispositivo electrónico 2922 obtiene un flujo de bits. En el bloque 3111, el dispositivo electrónico 2922 recupera un símbolo binario del flujo de bits obtenido.

- 40 En el rombo 3112, el dispositivo electrónico 2922 determina si el símbolo binario ha a ser decodificado usando decodificación sin pérdidas. En un ejemplo, la determinación puede incluir comprobar una cabecera, tal como una cabecera de segmento, correspondiente al flujo de bits recibido. La comprobación de la cabecera puede comprender además comprobar una cabecera de segmento correspondiente al flujo de bits obtenido para un valor de una bandera asociada con una segunda técnica de codificación de Absoluto -3 diferente. En otro ejemplo, la determinación puede incluir comprobar un símbolo decodificado previamente asociado con un bloque, tal como el tipo de bloque o el parámetro de cuantificación que controla la conversión de niveles de coeficientes a los TQC. Si la condición no se cumple en el rombo 3112, entonces en el bloque 3113 el dispositivo electrónico 2922 usa una primera técnica de codificación de Absoluto -3 para obtener un bloque de TQC.

- 45 Si se cumple la condición en el rombo 3112, entonces en el bloque 3114 el dispositivo electrónico 2921 usa una segunda técnica de codificación de Absoluto -3 diferente para obtener una muestra residual. El dispositivo electrónico 2922 puede almacenar el bloque obtenido de los TQC o la muestra residual obtenida en un dispositivo de memoria y/o puede recuperar datos de video en el bloque 3115.

(Codificación de alta capacidad de procesamiento para CABAC en HEVC)

- 55 Cuando se utiliza la codificación CABAC en HEVC, el rendimiento de capacidad de procesamiento puede diferir dependiendo de diferentes factores, tales como, pero no limitados a: número total de contenedores/píxeles, número de contenedores/píxeles por derivación y número de contenedores/píxeles codificados habituales (o contextuales). Por lo tanto, dependiendo de estos factores, la codificación puede consumir una cantidad significativa de recursos de procesamiento y/o puede tardar una cantidad de tiempo significativa. La descripción que sigue resuelve este y otros

problemas.

A modo de antecedentes, según la CABAC conocida, se codifican por contexto hasta veinticinco banderas de código de nivel de un elemento de sintaxis. Las banderas de código de nivel restantes se codifican por derivación. Un número predefinido (y fijo) de banderas Mayor\_que\_1 se codifican por contexto, esto es, ocho banderas Mayor\_que\_1. Un número predefinido (y fijo) de banderas Mayor\_que\_2 se codifican por contexto, esto es, uno. Todas de las banderas de mapa de importancia se codifican por contexto, esto es, hasta dieciséis (un elemento de sintaxis puede tener menos de dieciséis banderas de mapa de importancia dependiendo de la información de última posición del bloque. Por lo tanto, se necesitan un máximo de veinticinco contenedores codificados por contexto para un bloque de subconjunto dado (25 contenedores/16 píxeles = 1,56 contenedores/píxel). El ejemplo anterior es cuando se usa un bloque secundario 4x4.

La Figura 34 es un diagrama de flujo que ilustra una configuración de un método para la codificación de alta capacidad de procesamiento para CABAC en HEVC en un dispositivo electrónico.

En el bloque 3411, el dispositivo electrónico 3321 obtiene un bloque de datos para ser codificado usando un codificador basado en aritmética, por ejemplo, un codificador basado en CABAC. En el bloque 3412, el dispositivo electrónico 3321 codifica por contexto una primera cantidad de banderas de código de nivel de un elemento de sintaxis, por ejemplo, banderas Mayor\_que\_1 y Mayor\_que\_2 de un elemento de sintaxis de CABAC. La primera cantidad comprende un primer número predefinido, por ejemplo, nueve, esto es, ocho banderas Mayor\_que\_1 y una bandera Mayor\_que\_2.

En el bloque 3413, el dispositivo electrónico 3321 identifica un número de contenedores realmente codificados en el mapa de importancia del elemento de sintaxis. En el bloque 3414, el dispositivo electrónico 3321 determina una diferencia de un segundo número predefinido, por ejemplo, dieciséis en CABAC, y el número identificado. En el bloque 3415, el dispositivo electrónico 3321 codifica por contexto una segunda cantidad de las banderas de código de nivel, en donde la segunda cantidad comprende la diferencia determinada. En el bloque 3416, el dispositivo electrónico 3321 transmite el flujo de bits generado sobre una red y/o almacena el flujo de bits generado en un dispositivo de memoria.

En un ejemplo, si se usa la configuración mostrada en la FIG. 34, el dispositivo electrónico 3321 puede establecer un indicador correspondiente, por ejemplo, una bandera, a un valor de 1 (que por supuesto puede incluir cambiar un valor por defecto de la bandera o dejar la bandera en un valor por defecto dependiendo de la preferencia de diseño). En un ejemplo, el indicador puede identificar un número de indicadores Mayor\_que\_1 y/o Mayor\_que\_2 adicionales que se codifican por contexto.

Un ejemplo de un elemento de sintaxis generado según la configuración descrita anteriormente se muestra en la FIG. 36. En el ejemplo, un número de contenedores realmente codificados en el mapa de importancia del elemento de sintaxis del ejemplo es doce. La diferencia determinada entre dieciséis y doce es cuatro. La primera cantidad de las banderas de código de nivel que se codifican por contexto es nueve (ocho banderas Mayor\_que\_1 y una bandera Mayor\_que\_2). La segunda cantidad de banderas de código de nivel que se codifican por contexto es cuatro. En este ejemplo particular, estas cuatro son todas banderas Mayor\_que\_1, pero en otros ejemplos, estas cuatro pueden incluir una o más banderas Mayor\_que\_1 y una o más banderas Mayor\_que\_2, o cuatro banderas Mayor\_que\_2. Las banderas de código de nivel restantes se codifican por derivación.

La Figura 35 es un diagrama de flujo que ilustra una configuración de un método para codificación de alta capacidad de procesamiento para CABAC en HEVC en un dispositivo electrónico en el lado de decodificación.

En el bloque 3510, el dispositivo electrónico 3322 obtiene un flujo de bits. En el bloque 3511, el dispositivo electrónico 3322 recupera un símbolo binario del flujo de bits obtenido.

En el bloque 3512, el dispositivo electrónico 3322 decodifica por contexto una primera cantidad de banderas de código de nivel de un elemento de sintaxis, por ejemplo, banderas Mayor\_que\_1 y Mayor\_que\_2 de un elemento de sintaxis de CABAC, en donde la primera cantidad es igual a un primer número predefinido, por ejemplo, nueve, esto es, ocho banderas Mayor\_que\_1 y una bandera Mayor\_que\_2. En el rombo 3513, el dispositivo electrónico 3322 determina si las banderas de código de nivel adicionales del elemento de sintaxis se codifican por contexto. En un ejemplo, la determinación puede incluir comprobar una cabecera, tal como una cabecera de segmento, correspondiente al flujo de bits recibido. La comprobación de la cabecera puede comprender además comprobar una cabecera de segmento correspondiente al flujo de bits obtenido para un valor de un indicador, por ejemplo, una bandera. Si el dispositivo electrónico 3322 determina que las banderas de código de nivel adicionales no se codifican por contexto en el rombo 3513, entonces en el bloque 3514 el dispositivo electrónico 3322 decodifica por derivación las banderas de código de nivel restantes del elemento de sintaxis.

Si el dispositivo electrónico 3322 determina que las banderas de código de nivel adicionales se codifican por contexto en el rombo 3513, entonces en el bloque 3515 el dispositivo electrónico 3322 decodifica por contexto una segunda cantidad de las banderas de código de nivel del elemento de sintaxis. En un ejemplo, el dispositivo electrónico 3322 puede identificar un número de banderas Mayor\_que\_1 y/o Mayor\_que\_2 adicionales que se codifican por contexto en base a la información de la cabecera de segmento. En el bloque 3514, el dispositivo electrónico 3322 decodifica por derivación cualquier bandera de código de nivel restante. En el bloque 3516, el dispositivo electrónico 3322

almacena el bloque obtenido de los TQC o la muestra residual obtenida en un dispositivo de memoria y/o recupera datos de video.

La Figura 38 es un diagrama de flujo que ilustra una configuración de un método para la codificación de alta capacidad de procesamiento para CABAC en HEVC en un dispositivo electrónico en el lado de decodificación.

5 En el bloque 3850, el dispositivo electrónico 3322 obtiene un flujo de bits. En el bloque 3851, el dispositivo electrónico 3322 obtiene un bloque de valores de nivel a partir del flujo de bits obtenido.

10 En el bloque 3853, el dispositivo electrónico 3322 determina una posición de un último coeficiente significativo del bloque. En el bloque 3854, el dispositivo electrónico 3322 decodifica por contexto coeficientes significativos a partir del bloque obtenido, y determina un número máximo de símbolos para decodificar por contexto en función de la posición determinada.

15 En el bloque 3855, el dispositivo electrónico 3322 restablece un contador, por ejemplo, establece el contador a cero. En el rombo 3856, el dispositivo electrónico 3322 determina si una bandera de código de nivel queda para ser decodificada. Si no queda ninguna bandera de código de nivel en el rombo 3856, entonces en el bloque 3857 el dispositivo electrónico 3322 almacena el bloque obtenido de los TQC o la muestra residual obtenida en un dispositivo de memoria y/o recupera datos de video.

20 Si queda una bandera de código de nivel (rombo 3856), entonces en el rombo 3858 el dispositivo electrónico 3322 determina si el contador es mayor que un umbral. En un ejemplo, el umbral se puede asociar con el máximo determinado. En un ejemplo, el umbral puede corresponder a una diferencia del máximo determinado y un número de banderas de mapa de importancia del bloque. Si el contador es mayor que el umbral en el rombo 3858, entonces en el bloque 3859 el dispositivo electrónico 3322 decodifica por derivación la bandera de código de nivel. Si el contador no es mayor que el umbral en el rombo 3858, entonces en el bloque 3860 el dispositivo electrónico 3322 decodifica por contexto la bandera de código de nivel. El dispositivo electrónico 3322 aumenta el contador en el bloque 3861.

La Figura 39 es un diagrama de flujo que ilustra una configuración de un método para la codificación de alta capacidad de procesamiento para CABAC en HEVC en un dispositivo electrónico en el lado de decodificación.

25 En el bloque 3950, el dispositivo electrónico 3322 obtiene un flujo de bits. En el bloque 3951, el dispositivo electrónico 3322 obtiene un bloque de valores de nivel a partir del flujo de bits obtenido.

30 En el bloque 3953, el dispositivo electrónico 3322 determina una posición de un último coeficiente significativo. En el bloque 3954, el dispositivo electrónico 3322 decodifica por contexto coeficientes significativos del bloque obtenido, y determina un número máximo de símbolos para decodificar por contexto en función tanto de la posición determinada como de la posición del bloque secundario. Los procesos 3955-3961 pueden corresponder a los procesos 3855-3861.

La Figura 40 es un diagrama de flujo que ilustra una configuración de un método para la codificación de alta capacidad de procesamiento para CABAC en HEVC en un dispositivo electrónico en el lado de decodificación.

En el bloque 4050, el dispositivo electrónico 3322 obtiene un flujo de bits. En el bloque 4051, el dispositivo electrónico 3322 obtiene un bloque de valores de nivel a partir del flujo de bits obtenido.

35 En el bloque 4053, el dispositivo electrónico 3322 determina una posición de un último coeficiente significativo. En el bloque 4054, el dispositivo electrónico 3322 decodifica por contexto coeficientes significativos a partir del bloque obtenido, y determina un número máximo de símbolos para decodificar por contexto en función de una propiedad del bloque, por ejemplo, un número de coeficientes significativos del bloque. Los procesos 4055-4061 pueden corresponder a los procesos 3855-3861.

40 En un ejemplo, se proporciona un sistema. El sistema puede comprender un dispositivo electrónico configurado para obtener un bloque de valores de nivel a partir de un flujo de bits; decodificar por contexto una bandera de código de nivel del bloque; comprobar si hay una siguiente bandera de código de nivel del bloque; si hay una siguiente bandera de código de nivel, determinar si un recuento de banderas de código de nivel codificadas por contexto es mayor que un umbral; en respuesta a determinar que el recuento no es mayor que el umbral, decodificar por derivación la siguiente bandera de código de nivel; en respuesta a determinar que el recuento es mayor que el umbral, decodificar por contexto la siguiente bandera de código de nivel; recuperar un bloque de TQC o una muestra residual usando las banderas de código de nivel decodificadas; y almacenar el bloque recuperado en un dispositivo de memoria y/o recuperar datos de video.

50 El dispositivo electrónico se puede configurar para aumentar el recuento en respuesta a la codificación por contexto de la siguiente bandera de código de nivel. El dispositivo electrónico se puede configurar para repetir la comprobación, la determinación, la decodificación y el aumento hasta que se decodifiquen todas las banderas de código de nivel del bloque. El dispositivo electrónico se puede configurar para aumentar el recuento en respuesta a la codificación por contexto de la primera bandera de código de nivel.

El dispositivo electrónico se puede configurar para determinar una posición del último coeficiente significativo del

bloque; y determinar un número máximo de símbolos para decodificar por contexto en base, al menos en parte, a una propiedad del bloque. El dispositivo electrónico se puede configurar para establecer el umbral según un resultado de la determinación del número máximo.

5 El dispositivo electrónico se puede configurar para determinar una posición del último coeficiente significativo del bloque; y determinar un número máximo de símbolos para decodificar por contexto en base, al menos en parte, a la posición determinada. El dispositivo electrónico se puede configurar para determinar el número máximo de símbolos para decodificar por contexto en base, al menos en parte, a la posición determinada y en base, al menos en parte, a la posición del bloque secundario.

10 El dispositivo electrónico se puede configurar para decodificar por contexto una primera cantidad de banderas de código de nivel de un elemento de sintaxis asociado con el bloque, en donde la primera cantidad es igual a un primer número predefinido; identificar un número de contenedores realmente codificados en el mapa de importancia del elemento de sintaxis; determinar una diferencia de un segundo número predefinido y el número identificado; y decodificar por contexto una segunda cantidad de las banderas de código de nivel del elemento de sintaxis, en donde la segunda cantidad comprende la diferencia determinada. En un ejemplo, el primer número predefinido puede comprender nueve. En un ejemplo, el segundo número predefinido puede comprender dieciséis. En un ejemplo, las banderas de código de nivel correspondientes a la primera cantidad de banderas de código de nivel codificadas por contexto comprende ocho banderas "mayor que 1" y una bandera "mayor que 2". En un ejemplo, las banderas de código de nivel correspondientes a la segunda cantidad de banderas de código de nivel codificadas por contexto comprenden solamente banderas "mayor que 1". En un ejemplo, las banderas de código de nivel correspondientes a la segunda cantidad de banderas de código de nivel codificadas por contexto comprenden solamente banderas "mayor que 2". En un ejemplo, las banderas de código de nivel corresponden a la segunda cantidad de banderas codificadas por contexto que comprenden un tercer número predefinido de banderas "mayor que 2" y un número dinámico de banderas "mayor que 1", en donde el número dinámico comprende una diferencia de la segunda cantidad y del tercer número predefinido.

25 En un ejemplo, se proporciona un sistema. El sistema puede comprender un primer dispositivo electrónico de un codificador, el primer dispositivo electrónico configurado para: obtener un bloque de datos para ser codificado usando un codificador basado en aritmética; determinar si el bloque de datos ha de ser codificado usando codificación sin pérdidas; en respuesta a determinar que el bloque de datos no ha de ser codificado usando codificación sin pérdidas, usar una primera técnica de codificación de Absoluto -3 para codificar el bloque de datos; en respuesta a determinar que el bloque de datos ha de ser codificado usando codificación sin pérdidas, usar una segunda técnica de codificación de Absoluto -3 para codificar el bloque de datos; en donde la segunda técnica de codificación de Absoluto -3 es diferente a la primera técnica de codificación de Absoluto -3; y hacer que la codificación se almacene en un dispositivo de memoria.

35 El sistema puede comprender además un segundo dispositivo electrónico de un decodificador, el segundo dispositivo electrónico configurado para: determinar si un símbolo binario recibido ha de ser decodificado usando decodificación sin pérdidas; en respuesta a determinar que el símbolo binario no ha de ser decodificado usando decodificación sin pérdidas, usar la primera técnica de codificación de Absoluto -3 para obtener un bloque de TQC; y en respuesta a determinar que el símbolo binario ha de ser decodificado usando decodificación sin pérdidas, usar la segunda técnica de codificación de Absoluto -3 para obtener una muestra residual.

40 El primer dispositivo electrónico se puede configurar para: en respuesta a determinar que el bloque de datos no ha de ser codificado usando codificación sin pérdidas, inicializar un parámetro de Rice a cero para un valor inicial de un bloque secundario; y en respuesta a determinar que el bloque de datos ha de ser codificado usando codificación sin pérdidas, usar un parámetro de Rice de un último valor de un bloque secundario anterior para un valor inicial del bloque secundario.

45 El primer dispositivo electrónico se puede configurar para: en respuesta a determinar que el bloque de datos no ha de ser codificado usando codificación sin pérdidas, inicializar un parámetro de Rice a cero para un valor inicial de un bloque secundario; y en respuesta a determinar que el bloque de datos ha de ser codificado usando codificación sin pérdidas, evitar la inicialización del parámetro de Rice a cero para el valor inicial del bloque secundario.

50 El primer dispositivo electrónico se puede configurar para: en respuesta a determinar que el bloque de datos ha de ser codificado usando codificación sin pérdidas, inicializar el parámetro de Rice en un valor predefinido basado en al menos uno seleccionado del grupo que comprende el tipo de bloque, el tamaño del bloque y la información de color (luma/croma).

55 El primer dispositivo electrónico se puede configurar para: en respuesta a determinar que el bloque de datos ha de ser codificado usando codificación sin pérdidas, inicializar el parámetro de Rice a uno cuando el tipo de bloque actual sea igual a dos o a cinco.

El primer dispositivo electrónico se puede configurar para: en respuesta a determinar que el bloque de datos no ha de ser codificado usando codificación sin pérdidas, inicializar un parámetro de Rice a cero para un valor inicial de un bloque secundario; y en respuesta a determinar que el bloque de datos ha de ser codificado usando codificación sin

pérdidas, evitar la inicialización del parámetro de Rice a cero para el valor inicial del bloque secundario.

5 El primer dispositivo electrónico se puede configurar para: en respuesta a determinar que el símbolo binario no ha de ser decodificado usando decodificación sin pérdidas, emplear una primera tabla de actualización de parámetros de Rice; y en respuesta a determinar que el símbolo binario ha de ser decodificado usando una decodificación sin pérdidas, emplear una segunda tabla de actualización de parámetros de Rice que es diferente de la primera tabla de actualización de parámetros de Rice.

10 La segunda tabla de actualización de parámetros de Rice puede comprender una versión truncada de la primera tabla de actualización de parámetros de Rice. En un ejemplo, solamente la segunda tabla de actualización de parámetros de Rice se configura para evitar una actualización después de que un parámetro de Rice actual se actualice o inicialice a dos, tres o cuatro.

15 En un ejemplo, se proporciona un sistema. El sistema puede comprender un primer dispositivo electrónico de un codificador, el primer dispositivo electrónico configurado para: obtener un bloque de datos para ser codificado usando un codificador basado en aritmética; codificar por contexto una primera cantidad de banderas de código de nivel de un elemento de sintaxis, en donde la primera cantidad es igual a un primer número predefinido; identificar una cantidad de contenedores realmente codificados en el mapa de importancia del elemento de sintaxis; determinar una diferencia de un segundo número predefinido y el número identificado; codificar por contexto una segunda cantidad de las banderas de código de nivel del elemento de sintaxis, en donde la segunda cantidad comprende la diferencia determinada; y hacer que un flujo de bits generado por la codificación por contexto se almacene en un dispositivo de memoria.

20 La codificación basada en aritmética puede comprender un codificador de CABAC. El primer número predefinido puede comprender nueve. El segundo número predefinido puede comprender dieciséis. Las banderas de código de nivel correspondientes a la primera cantidad de banderas de código de nivel codificadas por contexto pueden comprender ocho banderas "mayor que 1" y una bandera "mayor que 2". Las banderas de código de nivel correspondientes a la segunda cantidad de banderas de código de nivel codificadas por contexto pueden comprender solamente banderas "mayor que 1". Las banderas de código de nivel correspondientes a la segunda cantidad de banderas de código de nivel codificadas por contexto pueden comprender solamente banderas "mayor que 2". Las banderas de código de nivel correspondientes a la segunda cantidad de banderas codificadas por contexto pueden comprender un tercer número predefinido de banderas "mayor que 2" y un número dinámico de banderas "mayor que 1", en donde el número dinámico comprende una diferencia de la segunda cantidad y del tercer número predefinido.

30 El sistema y el aparato descritos anteriormente pueden usar sistemas de procesamiento dedicados, microcontroladores, dispositivos lógicos programables, microprocesadores, o cualquier combinación de los mismos, para realizar algunas o todas las operaciones descritas en la presente memoria. Algunas de las operaciones descritas anteriormente se pueden implementar en software y otras operaciones se pueden implementar en hardware. Una o más de las operaciones, procesos y/o métodos descritos en la presente memoria se pueden realizar mediante un aparato, un dispositivo y/o un sistema sustancialmente similar a los descritos en la presente memoria y con referencia a las figuras ilustradas.

40 Un dispositivo de procesamiento puede ejecutar instrucciones o "código" almacenado en la memoria. La memoria también puede almacenar datos. El dispositivo de procesamiento puede incluir, pero no se limita a, un procesador analógico, un procesador digital, un microprocesador, un procesador de múltiples núcleos, una agrupación de procesadores, un procesador de red o similares. El dispositivo de procesamiento puede ser parte de un sistema de control integrado o gestor de sistema, o se puede proporcionar como un dispositivo electrónico portátil configurado para interactuar con un sistema en red o bien de manera local o bien de manera remota a través de transmisión inalámbrica.

45 La memoria del procesador se puede integrar junto con el dispositivo de procesamiento, por ejemplo, una memoria RAM o FLASH dispuesta dentro de un microprocesador de circuito integrado o similar. En otros ejemplos, la memoria puede comprender un dispositivo independiente, tal como una unidad de disco externa, una agrupación de almacenamiento, un llavero FLASH portátil o similar. La memoria y el dispositivo de procesamiento se pueden acoplar de manera operativa entre sí, o en comunicación uno con otro, por ejemplo, mediante un puerto de I/O, una conexión de red o similar, y el dispositivo de procesamiento puede leer un archivo almacenado en la memoria. La memoria asociada puede ser o no "de sólo lectura" por diseño (ROM) en virtud de la configuración de permisos. Otros ejemplos de memoria pueden incluir, pero no se limitan a, WORM, EPROM, EEPROM, FLASH o similares, que se pueden implementar en dispositivos semiconductores de estado sólido. Otras memorias pueden comprender partes móviles, tales como una unidad de disco giratoria convencional. Todas de tales memorias pueden ser "legibles por máquina" y pueden ser legibles por un dispositivo de procesamiento.

55 Las instrucciones o comandos de operación se pueden implementar o incorporar en formas tangibles de software almacenado en ordenador (también conocido como "programa de ordenador" o "código"). Los programas o código se pueden almacenar en una memoria digital y se pueden leer por el dispositivo de procesamiento. El "medio de almacenamiento legible por ordenador" (o alternativamente, el "medio de almacenamiento legible por máquina") puede incluir todos los tipos de memoria anteriores, así como nuevas tecnologías del futuro, siempre que la memoria pueda

5 ser capaz de almacenar información digital en la naturaleza de un programa de ordenador u otros datos, al menos temporalmente, y siempre que la información almacenada se pueda “leer” por un dispositivo de procesamiento apropiado. El término “legible por ordenador” no se puede limitar al uso histórico de “ordenador” para implicar un ordenador central, un miniordenador, un ordenador de sobremesa o incluso un ordenador portátil. Más bien, “legible por ordenador” puede comprender un medio de almacenamiento que puede ser legible por un procesador, un dispositivo de procesamiento o cualquier sistema informático. Tales medios pueden ser cualquier medio disponible que pueda ser accesible de manera local y/o de manera remota por un ordenador o un procesador, y puede incluir medios volátiles y no volátiles, y medios extraíbles y no extraíbles, o cualquier combinación de los mismos.

10 Un programa almacenado en un medio de almacenamiento legible por ordenador puede comprender un producto de programa de ordenador. Por ejemplo, un medio de almacenamiento se puede usar como un medio conveniente para almacenar o transportar un programa de ordenador. Por bien de la conveniencia, las operaciones se pueden describir como varios bloques o diagramas funcionales interconectados o acoplados. No obstante, puede haber casos en los que estos bloques o diagramas funcionales se puedan agregar de manera equivalente en un único dispositivo lógico, programa u operación con límites confusos.

15 Un experto en la técnica reconocerá que los conceptos enseñados en la presente memoria se pueden adaptar a una aplicación particular de muchas otras formas. La invención se expone en la reivindicación adjunta.

Aunque la especificación puede referirse a “un”, “otro” o “algunos” ejemplos en varias ubicaciones, esto no significa necesariamente que cada referencia tal sea al mismo ejemplo o ejemplos, o que el rasgo solamente se aplique a un único ejemplo.

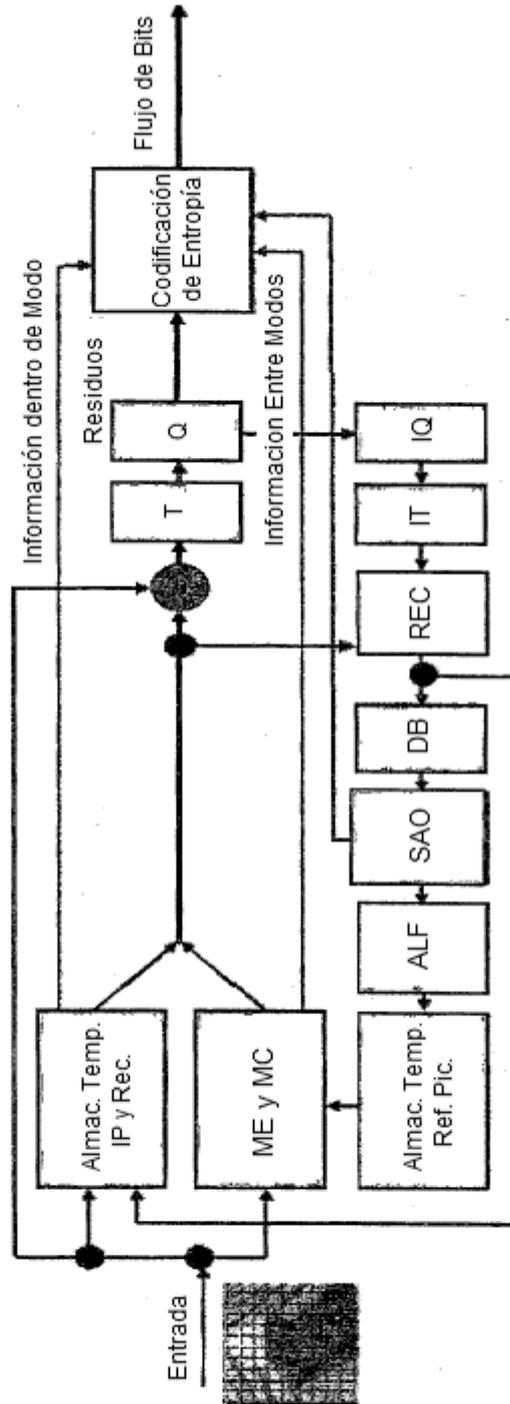
20

**REIVINDICACIONES**

1. Un método para decodificar un flujo de bits asociado con Coeficientes Transformados y Cuantificados, TQC, en HEVC, el método que comprende:

- obtener el flujo de bits (etapa 2611);
- 5     • obtener N TQC de un bloque de flujo de bits, en donde el bloque tiene un tamaño de bloque que tiene los N TQC (etapa 2612);
- determinar un número de TQC que no son iguales a cero entre los N TQC (etapa 2613);
- determinar si el número de TQC que no son iguales a cero del bloque es mayor que un umbral preestablecido, en el que el umbral preestablecido es el 50% de N (etapa 2614);
- 10    • si el número de TQC que no son iguales a cero del bloque es mayor que el umbral preestablecido:
  - decodificar los valores de nivel de los TQC no iguales a cero con una decodificación por derivación omitiendo una estimación de contexto y una actualización de probabilidad para la Codificación Aritmética Binaria Adaptativa al Contexto, CABAC (etapa 2616),
- si el número de TQC que no son iguales a cero del bloque no es mayor que el umbral preestablecido:
- 15    ○ decodificar los valores de nivel de los TQC no iguales a cero con una decodificación habitual usando la estimación de contexto y la actualización de probabilidad para la CABAC (etapa 2615).

[Fig. 1]

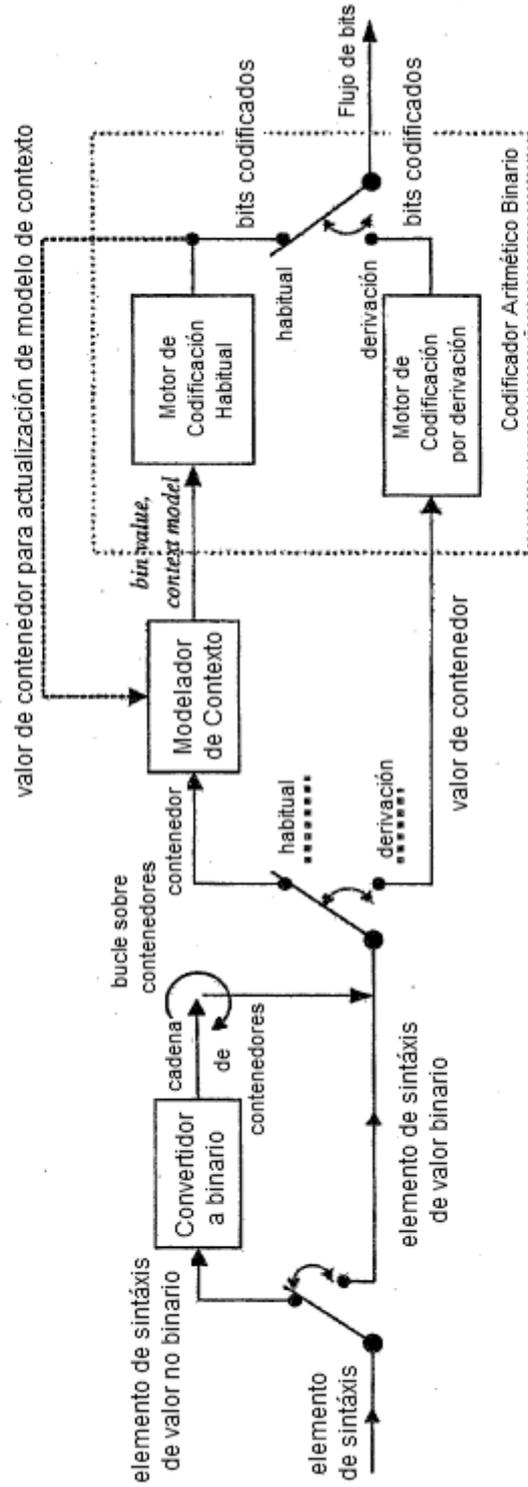


[Fig. 2]

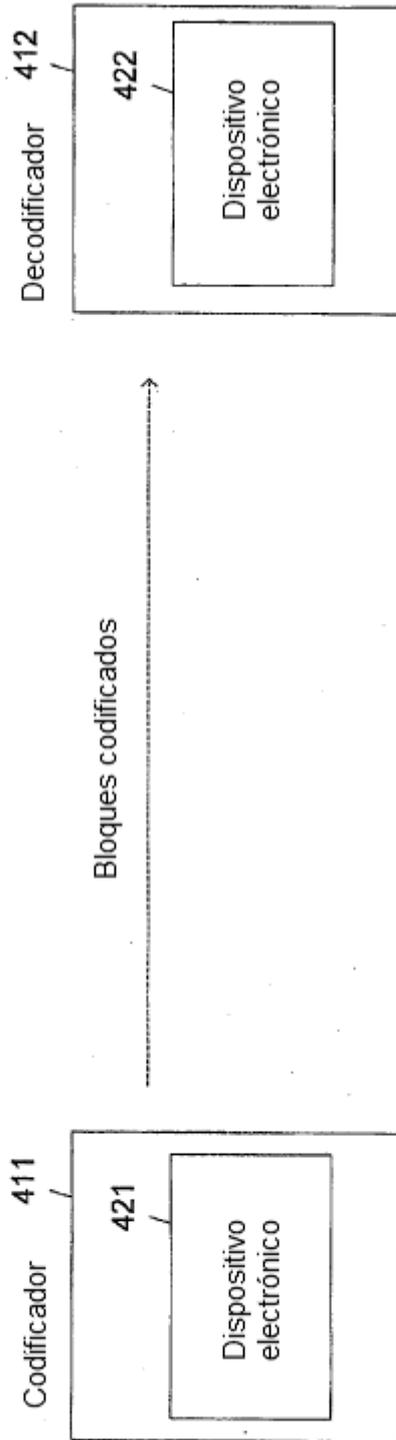
Posición de escaneado	0	1	2	3	4	5	6	7	...
Nivel de Coeficiente	4	0	3	-3	2	1	0	-1	...
1. Last_position_X Last_position_Y								2	1
2. Mapa de Importancia	1	0	1	1	1	1	0		
3. Greater_than_1	1		1	1	1	0		0	
Greater_than_2	1		1	1	0				
Información de Signo	+		+	-	+	+		-	
Absoluta -3	1		0	0					

\* Los coeficientes se codifican en orden de escaneado inverso

[Fig. 3]

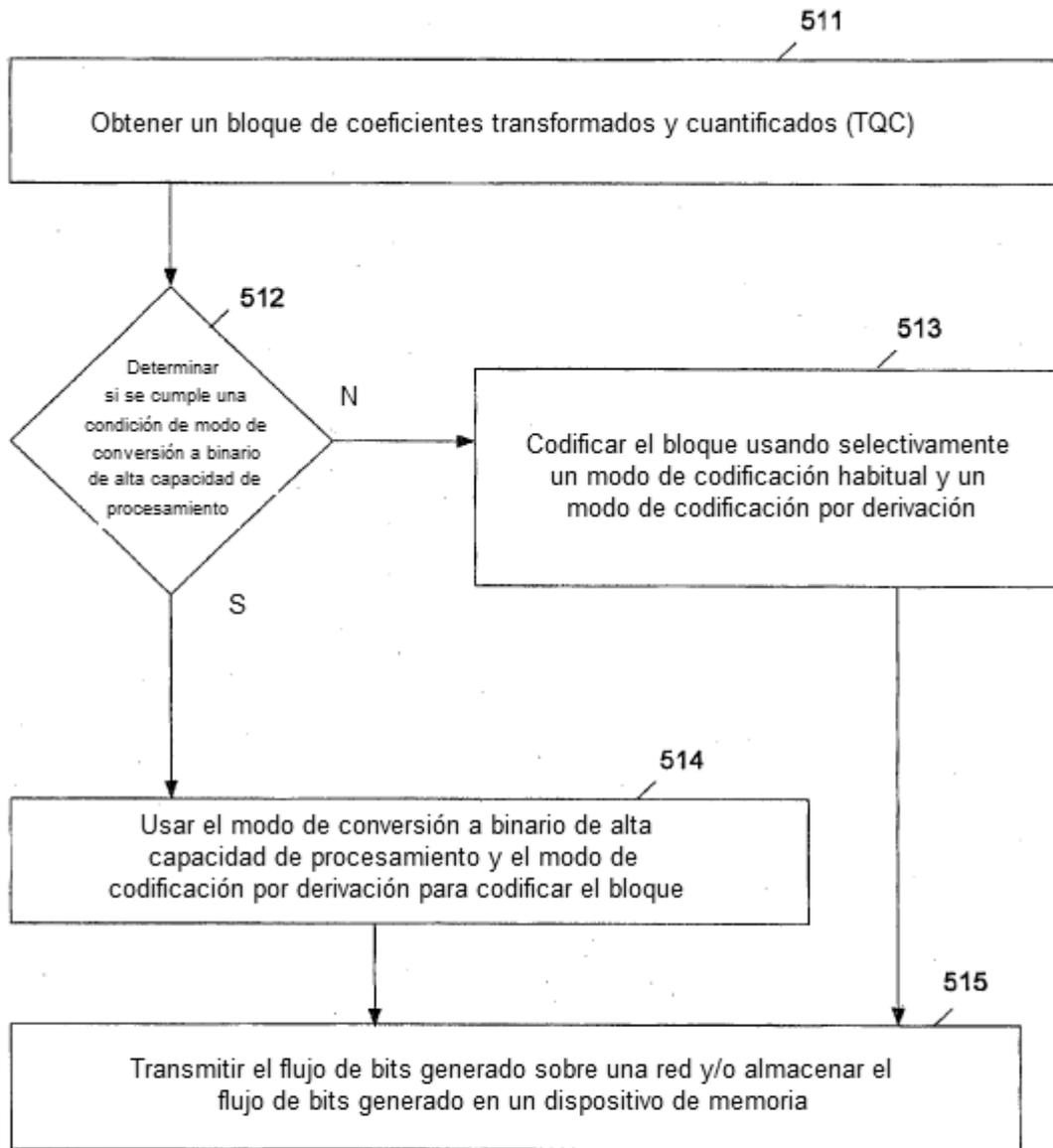


[Fig. 4]

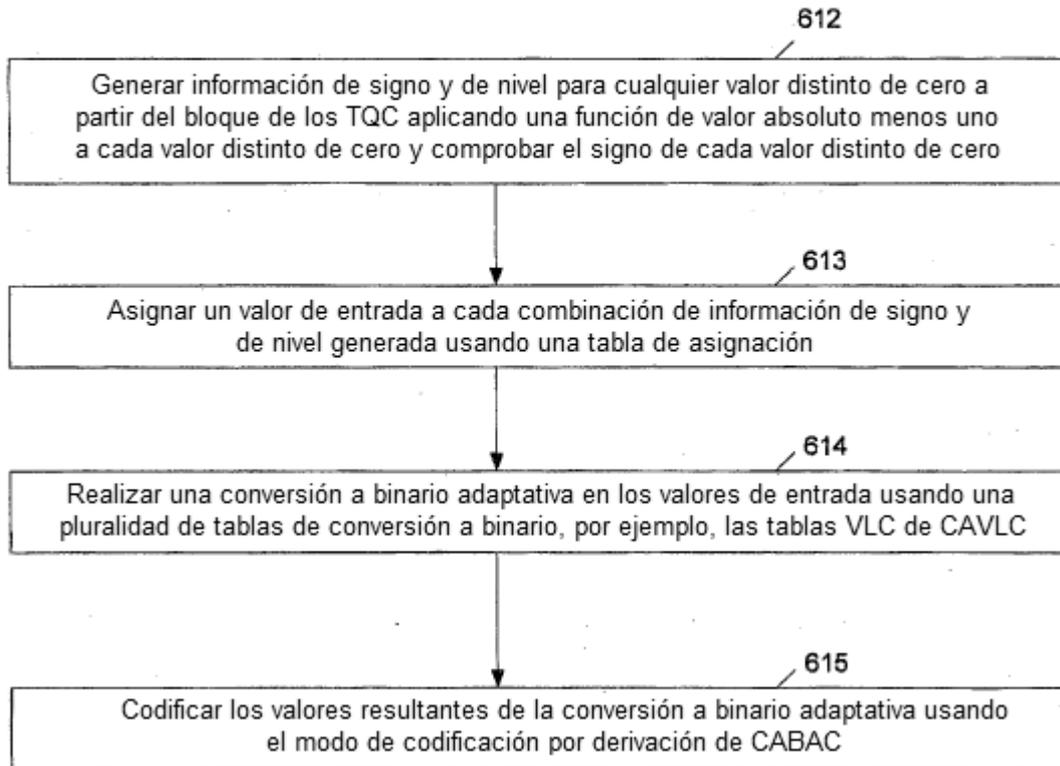


400

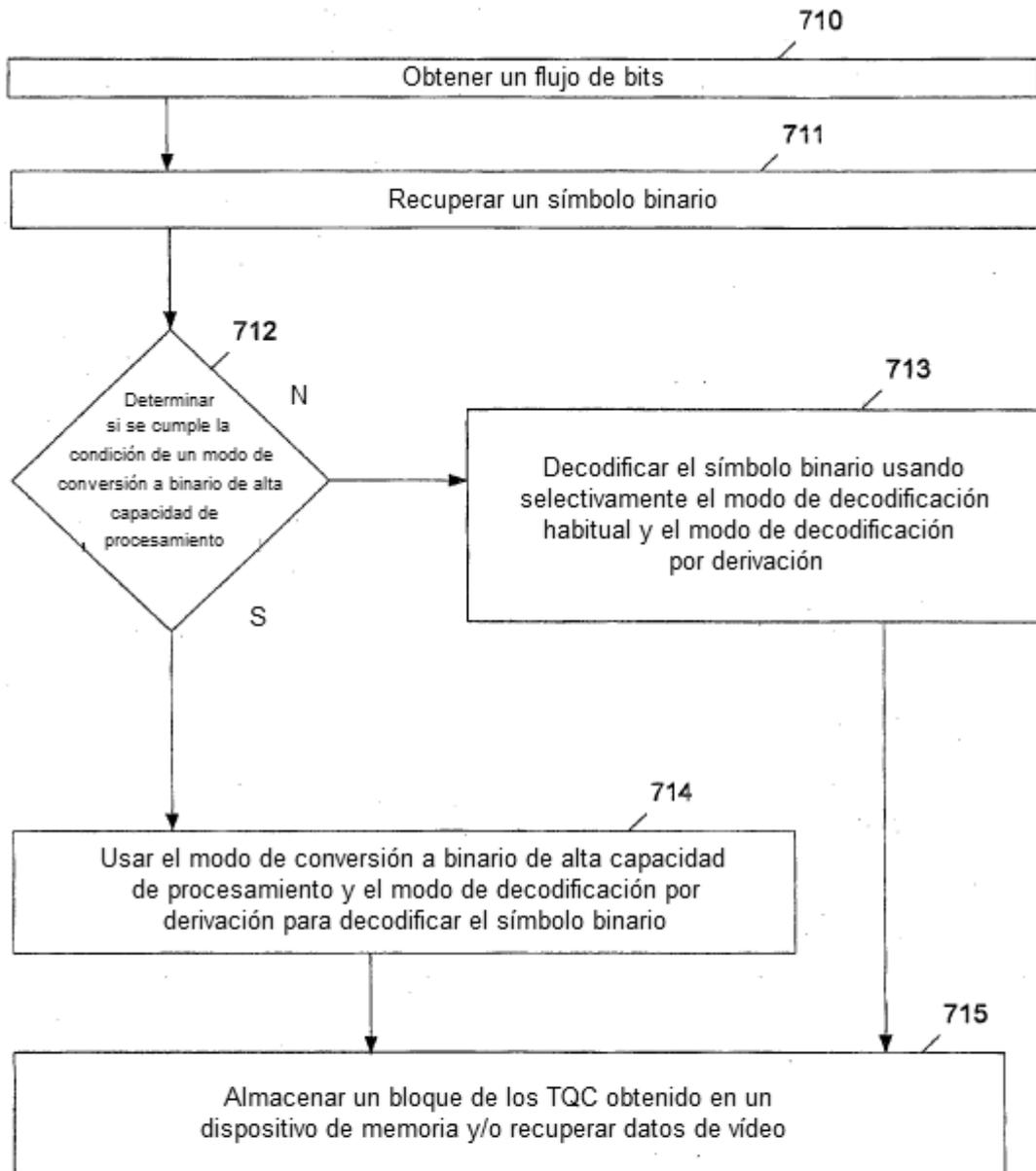
[Fig. 5]



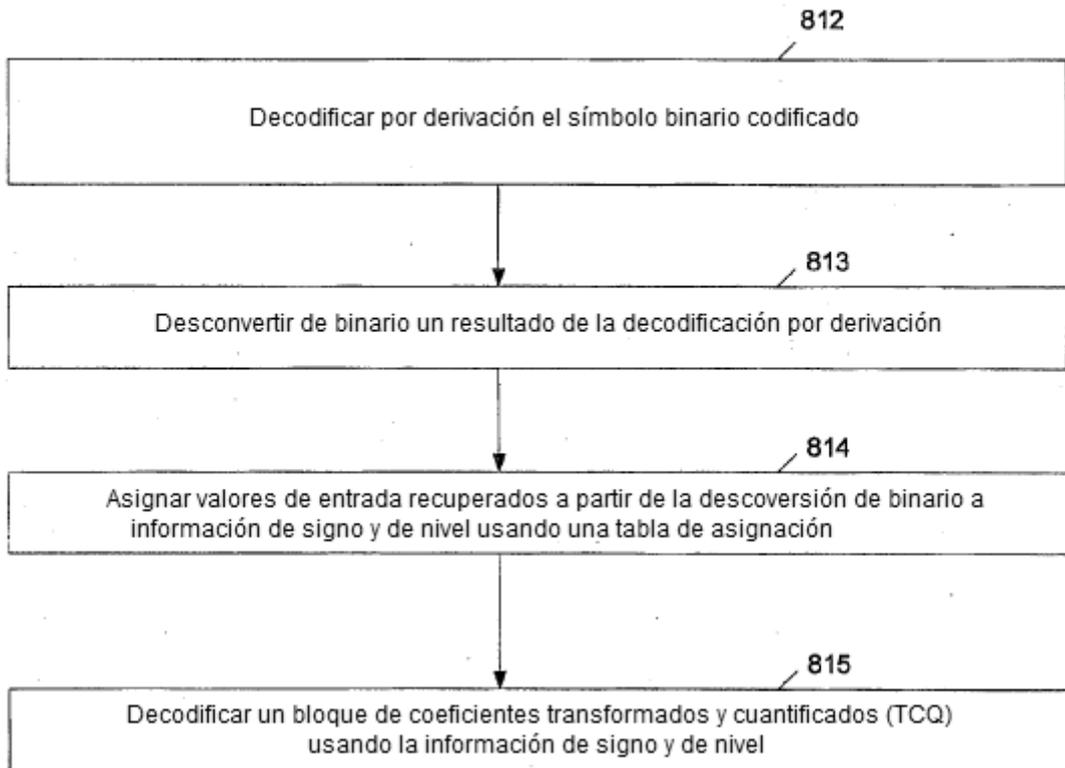
[Fig. 6]



[Fig. 7]



[Fig. 8]



[Fig. 9]

signo	Abs(coeficiente[i])	Entrada
+	1	0
-	1	1
+	2	2
-	2	3
+	3	4
-	3	5
+	4	6
-	4	7
...	...	...

Tabla de Asignación para Valor de Entrada

$$\text{entrada} = (\text{abs}(\text{coeficiente}[i] - 1) \ll 1 + \text{signo})$$

(signo +:0, -:1)

[Fig. 10]

Valor de entrada	Palabra de código (Tabla VLC 0)	Palabra de código (Tabla VLC 1)	Palabra de código (Tabla VLC 2)	Palabra de código (Tabla VLC 3)	Palabra de código (Tabla VLC 4)
0	1	10	100	1000	10000
1	01	11	101	1001	10001
2	001	010	110	1010	10010
3	0001	011	111	1011	10011
4	00001	0010	0100	1100	10100
5	000001	0011	0101	1101	10101
6	0000001	00010	0110	1110	10110
7	000000010	00011	0111	1111	10111
8	000000011	000010	00100	01000	11000
9	0000000100	000011	00101	01001	11001
10	0000000101	0000010	00110	01010	11010
11	0000000110	0000011	00111	01011	11011
12	0000000111	00000010	000100	01100	11100
...	...	...	...	...	...

si(entrada > (Tabla[vlc])) vlc ++;  
 donde Tabla [VLC] = {3,5,13,27}

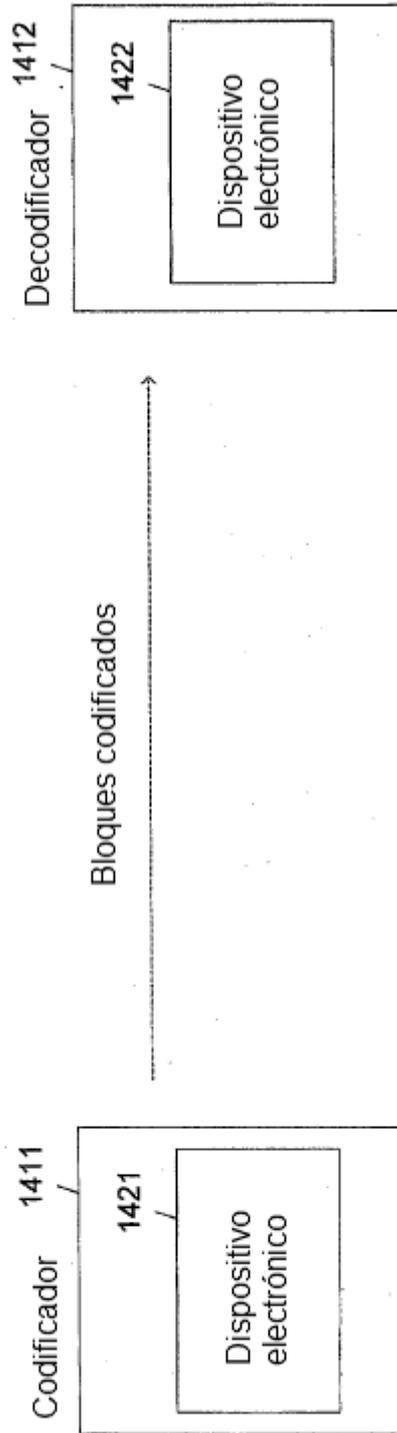
'vlc' representa el número de tabla vlc actual y  
 detener la actualización de la tabla 'vlc' cuando vlc es igual a 4

[Fig. 11]

Posición de Escaneado	0	1	2	3	4	5	6	7	...
Nivel de Coeficiente	4	0	3	-3	2	1	0	-1	...
1. Última_posición_X Última_posición_Y								2	
								1	
2. Mapa de Importancia	1	0	1	1	1	1	0		
3. Mayor_que_1	1		1	1	1	0		0	
Mayor_que_2	1		1	1	0				
Información de Signo	+		+	-	+	+		-	
Absoluto-3	1		0	0					

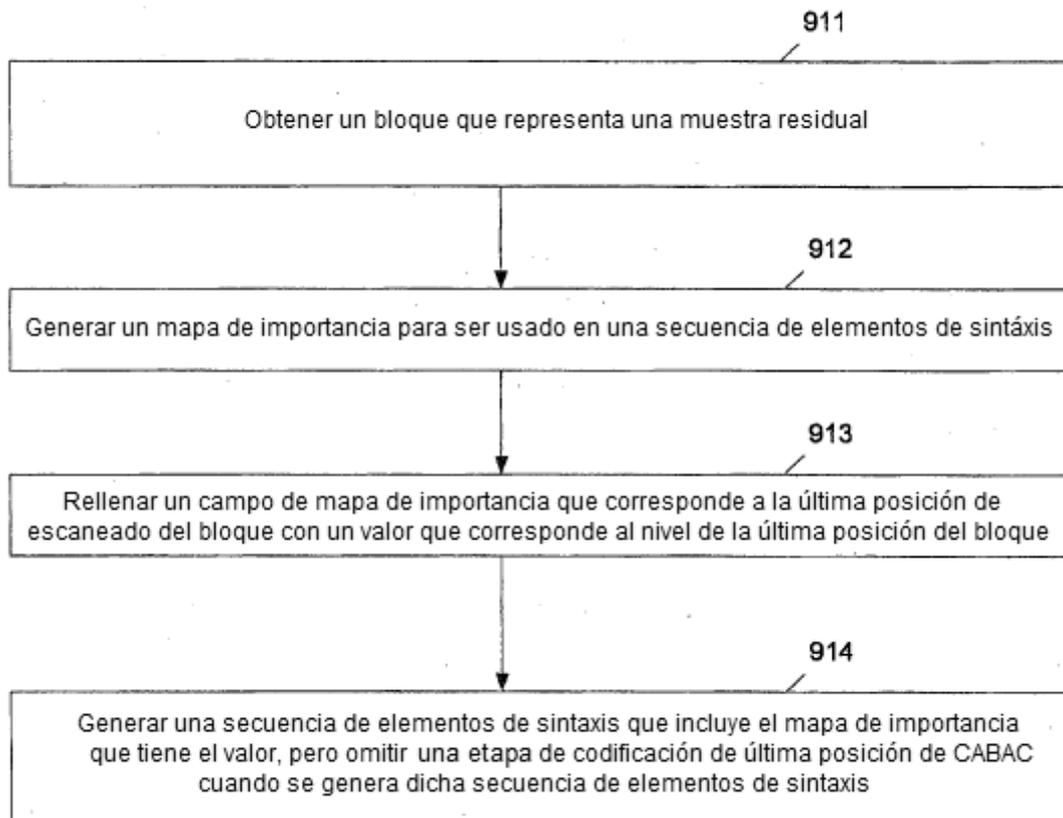
\* Los coeficientes se codifican en orden de escaneado inverso

[Fig. 12]



1400

[Fig. 13]



[Fig. 14]

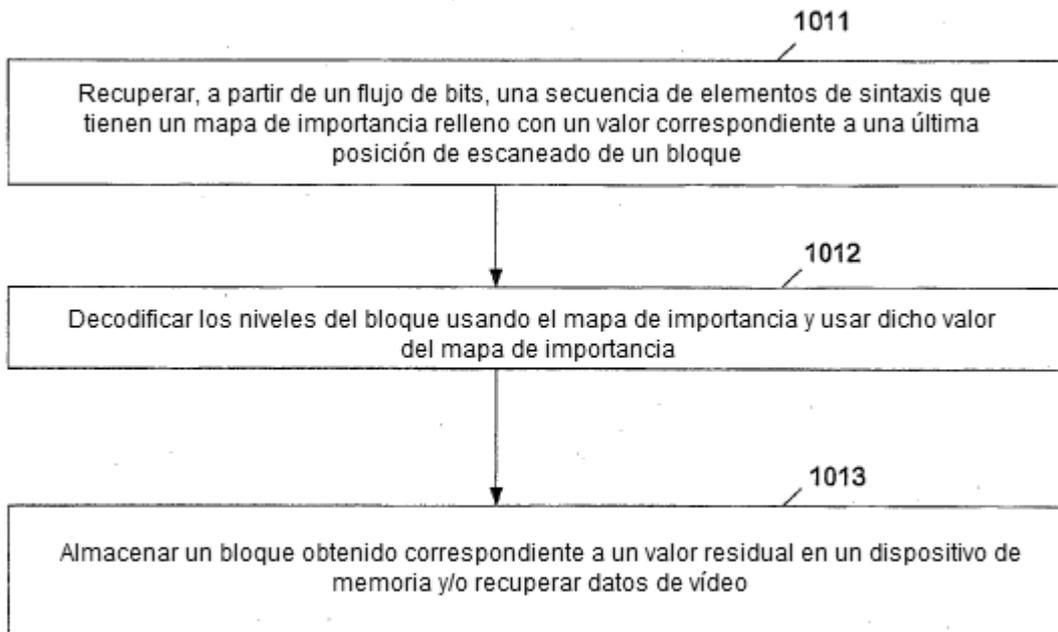
Posición de Escaneado	0	1	2	3	4	5	6	7	...	15
Nivel de Coeficiente	4	0	3	-3	2	1	0	-1	...	0
1. Mapa de Importancia	1	0	1	1	1	1	0	1	...	0

•  
•  
•

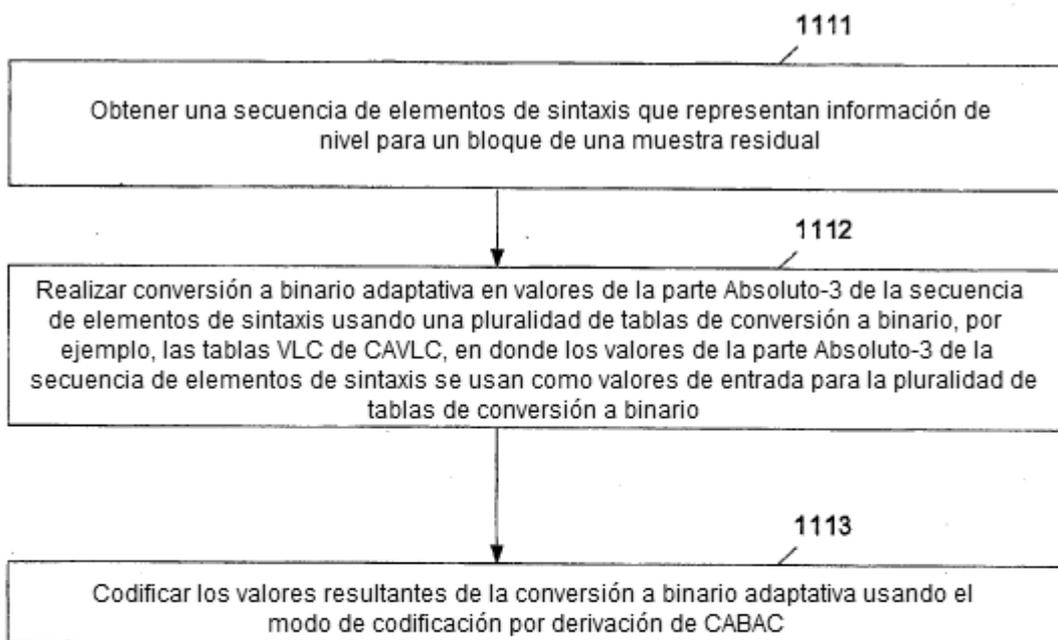
↓

\* Los coeficientes se codifican en orden de escaneado inverso

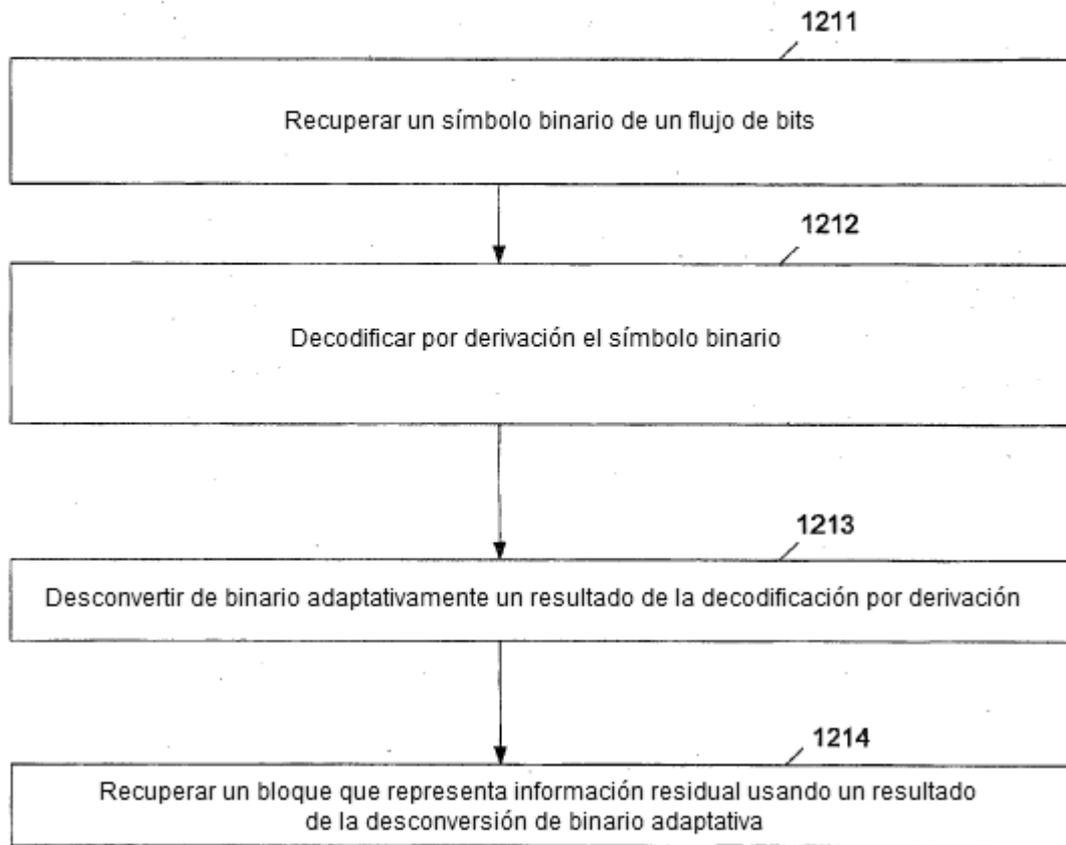
[Fig. 15]



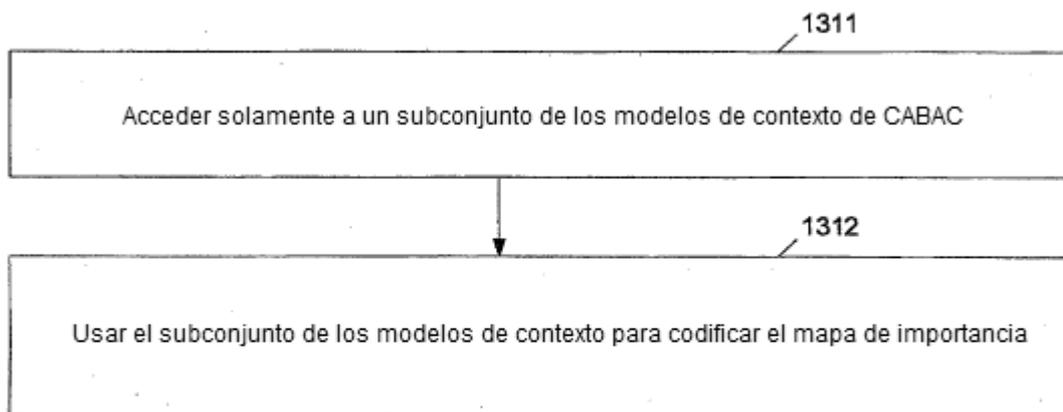
[Fig. 16]



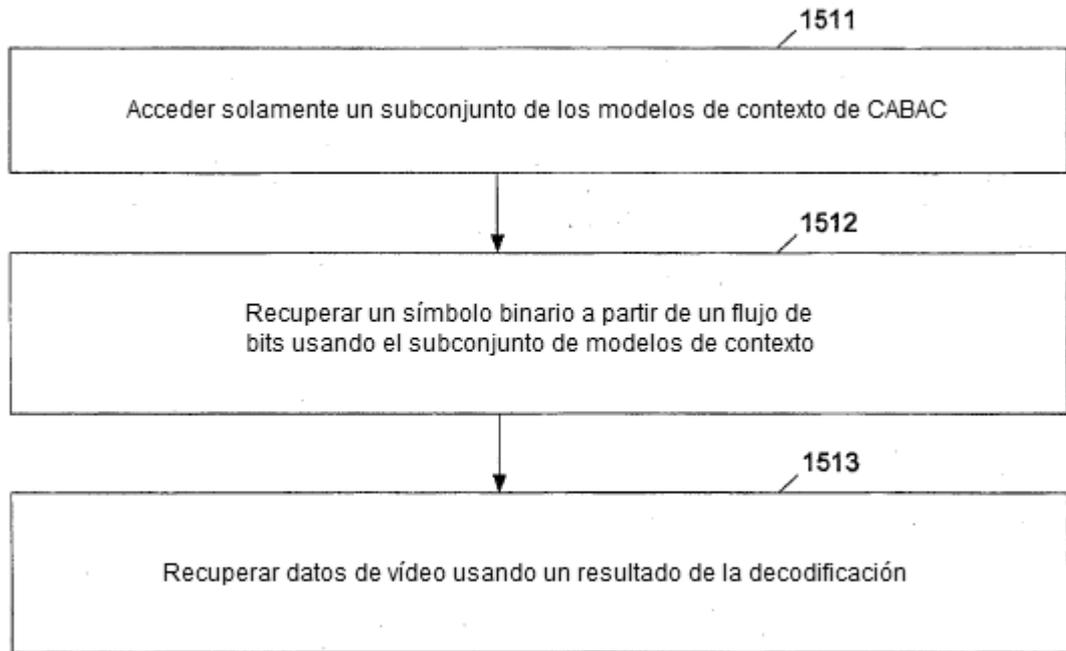
[Fig. 17]



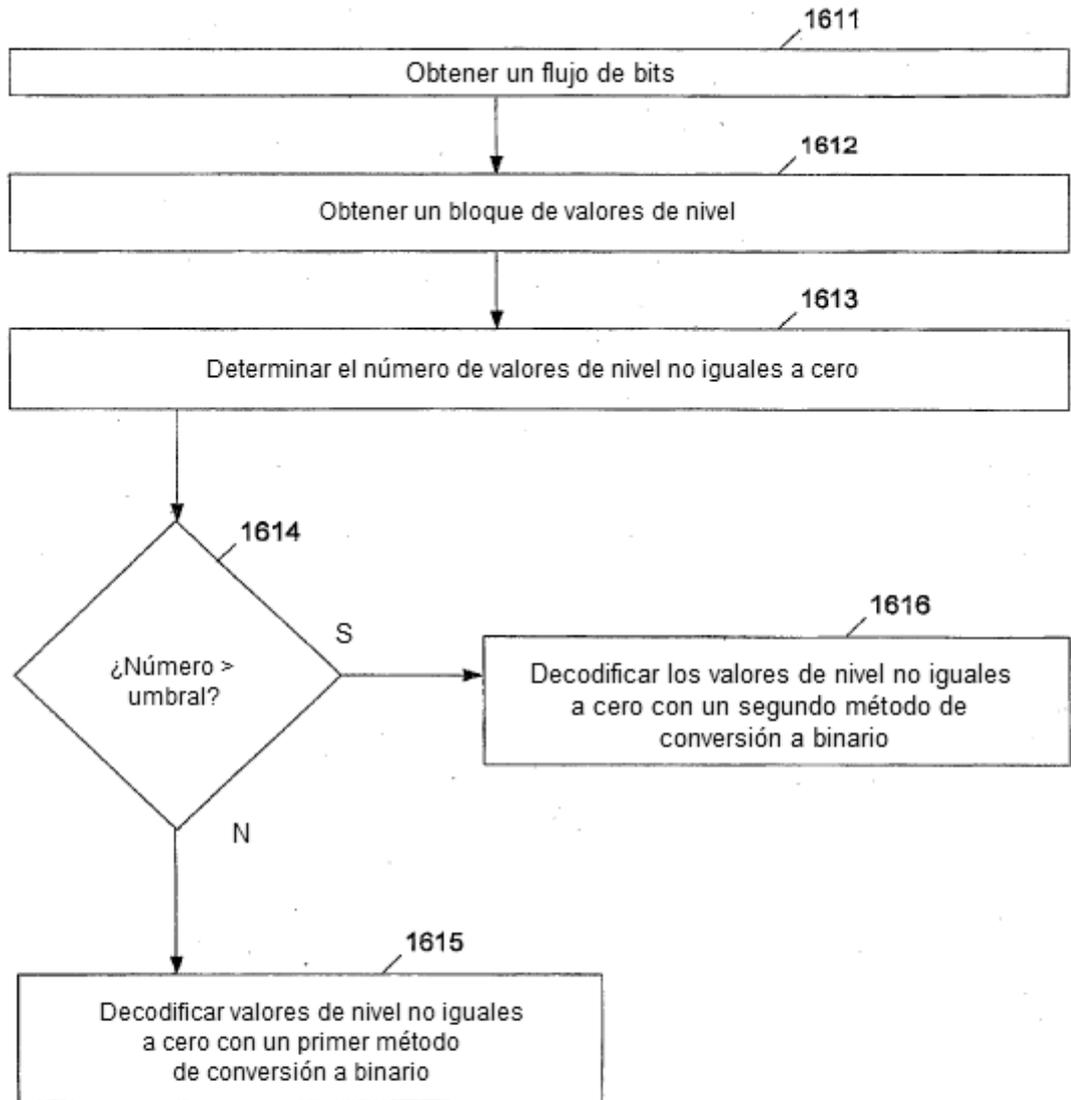
[Fig. 18]



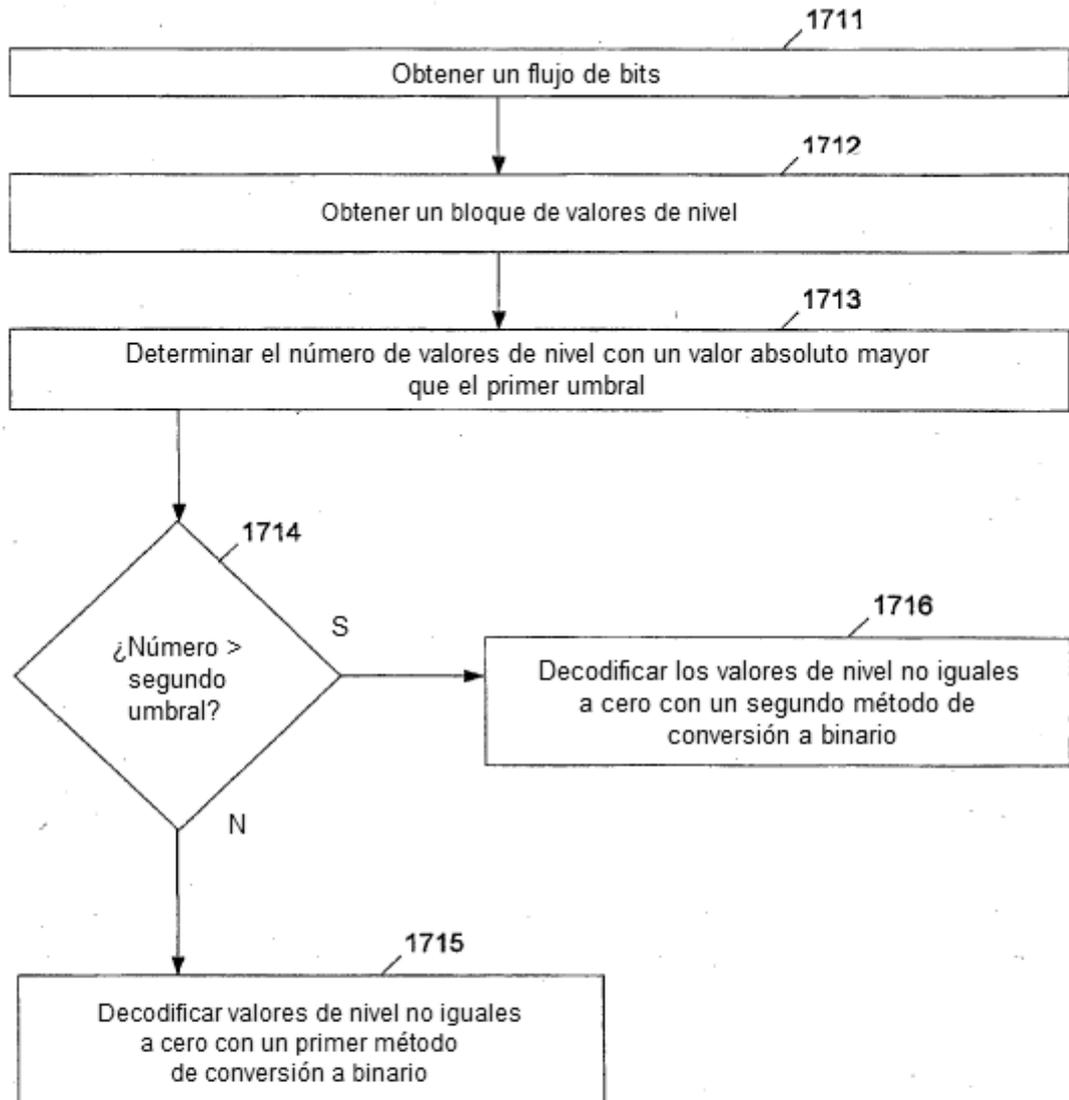
[Fig. 19]



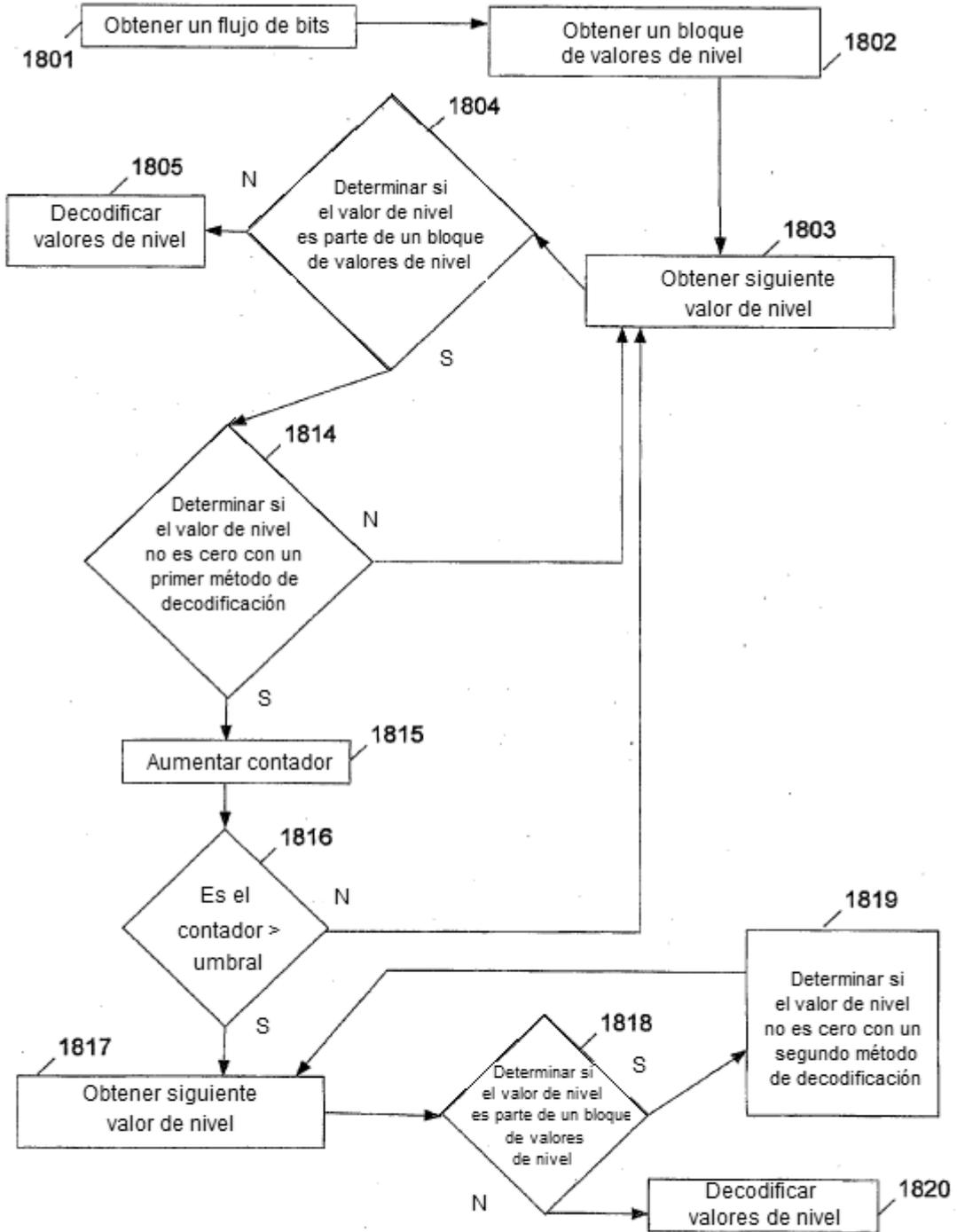
[Fig. 20A]



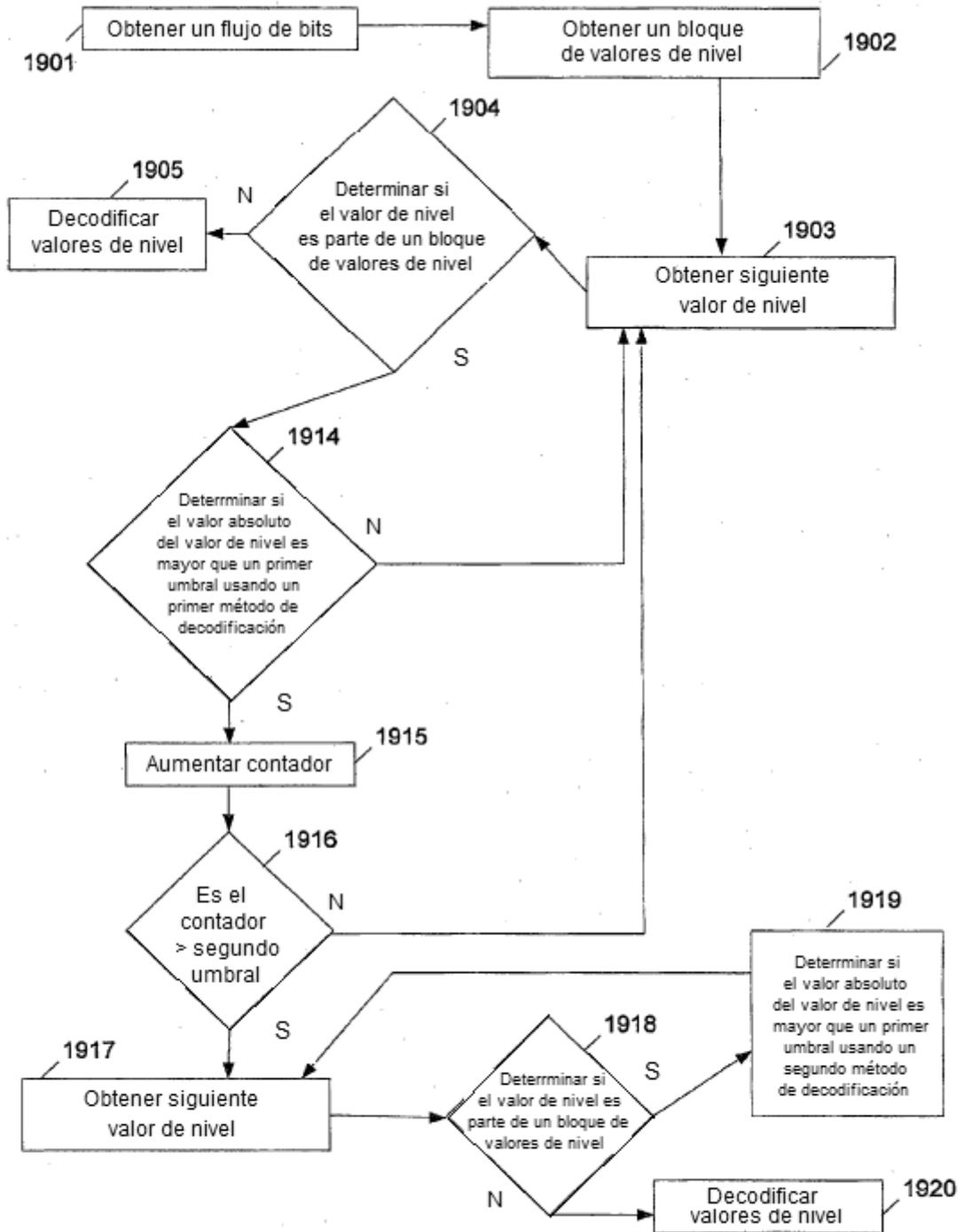
[Fig. 20B]



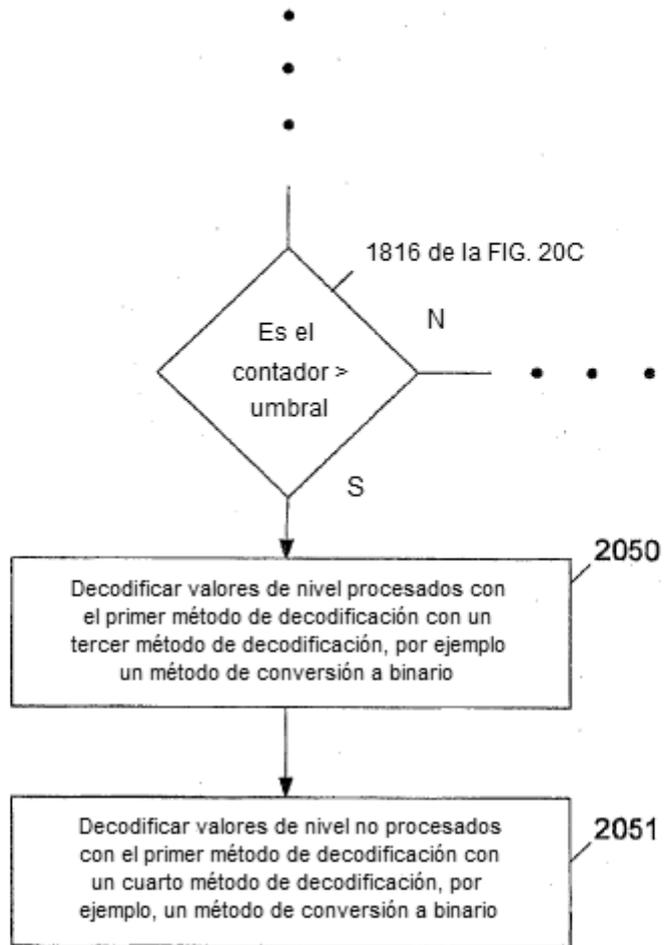
[Fig. 20C]



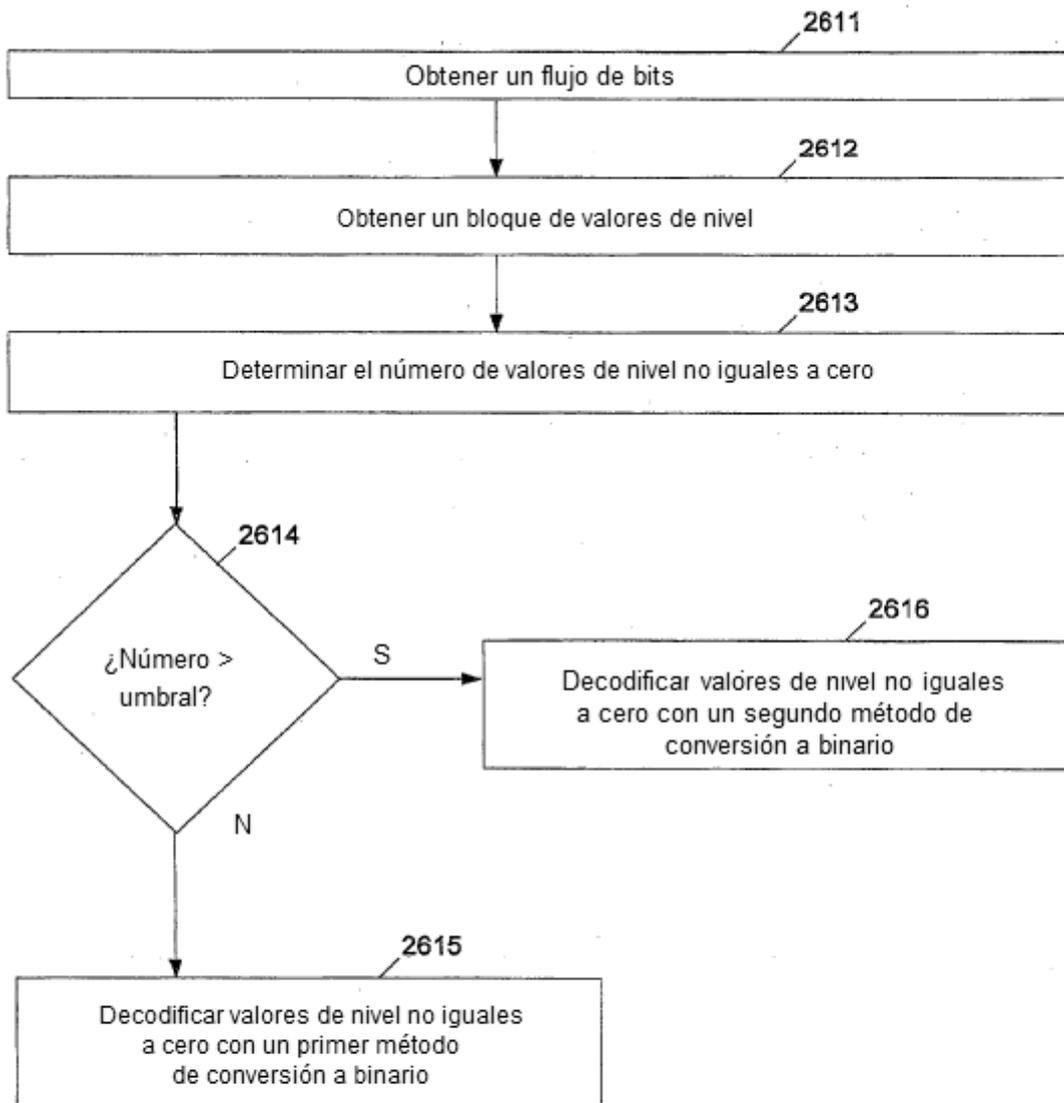
[Fig. 20D]



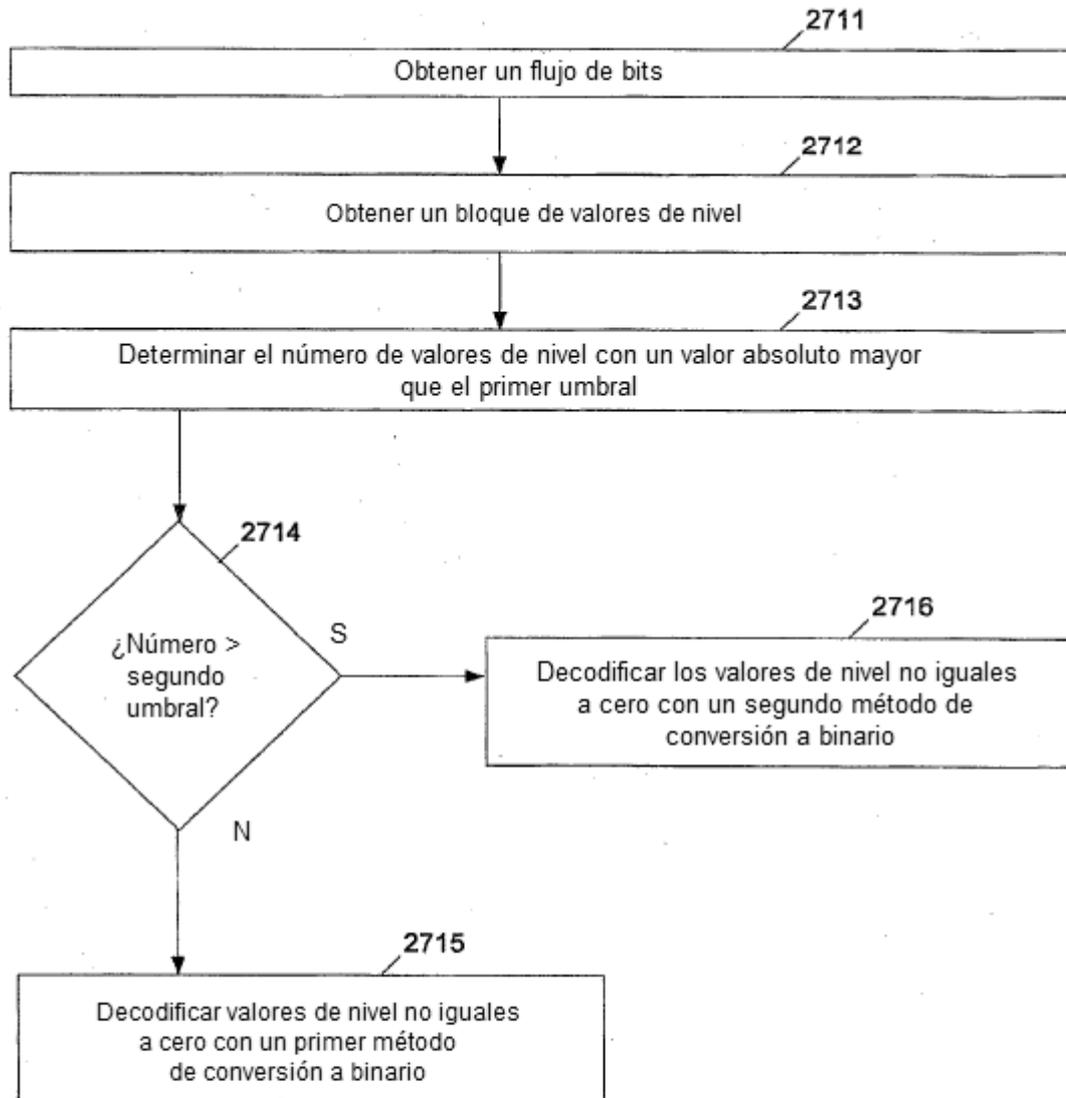
[Fig. 20E]



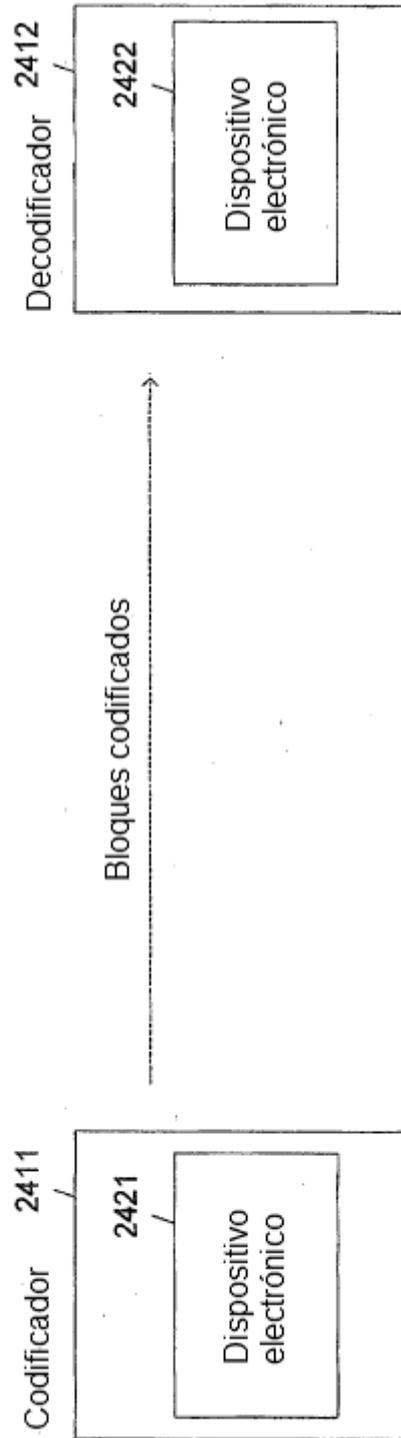
[Fig. 21]



[Fig. 22]

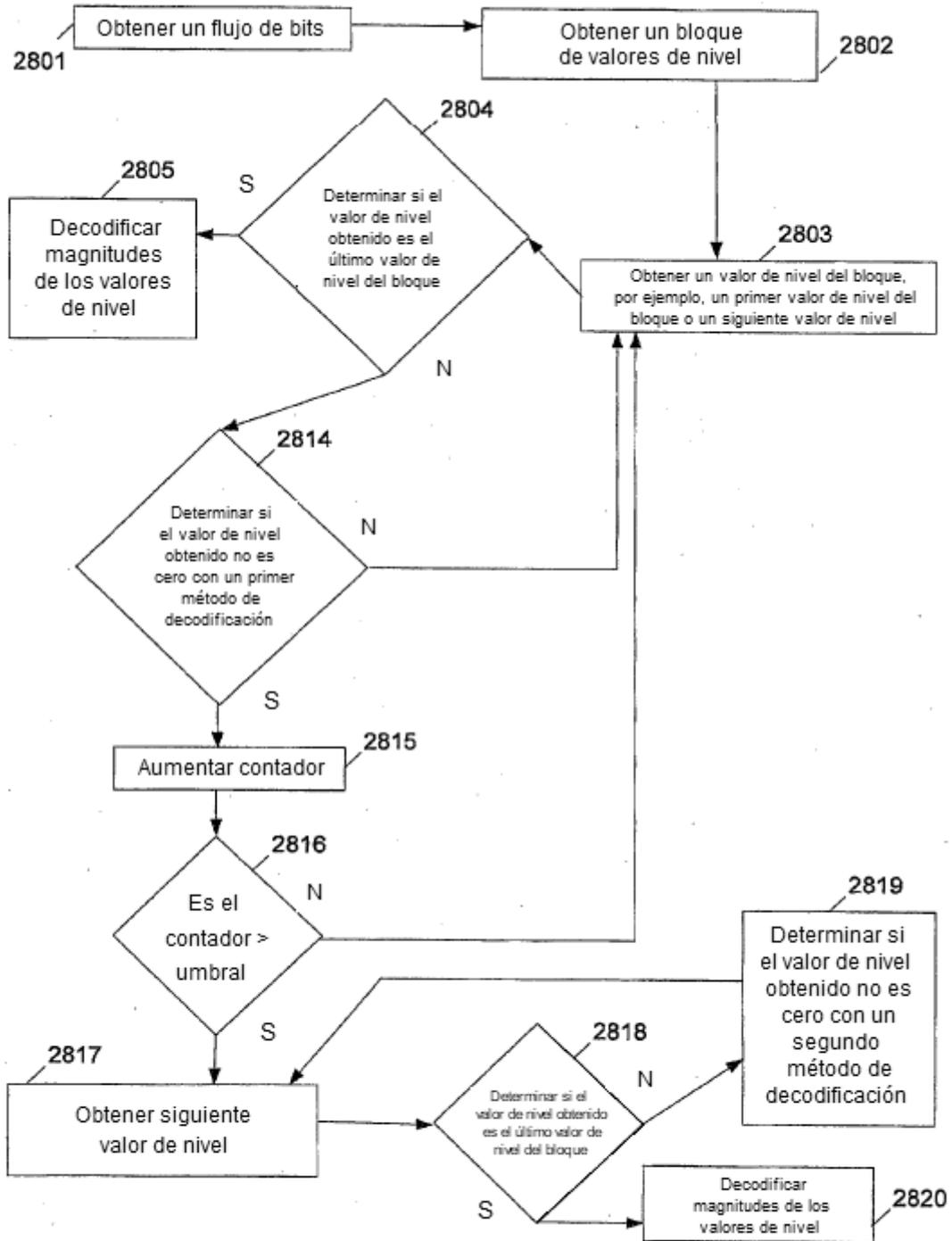


[Fig. 23]

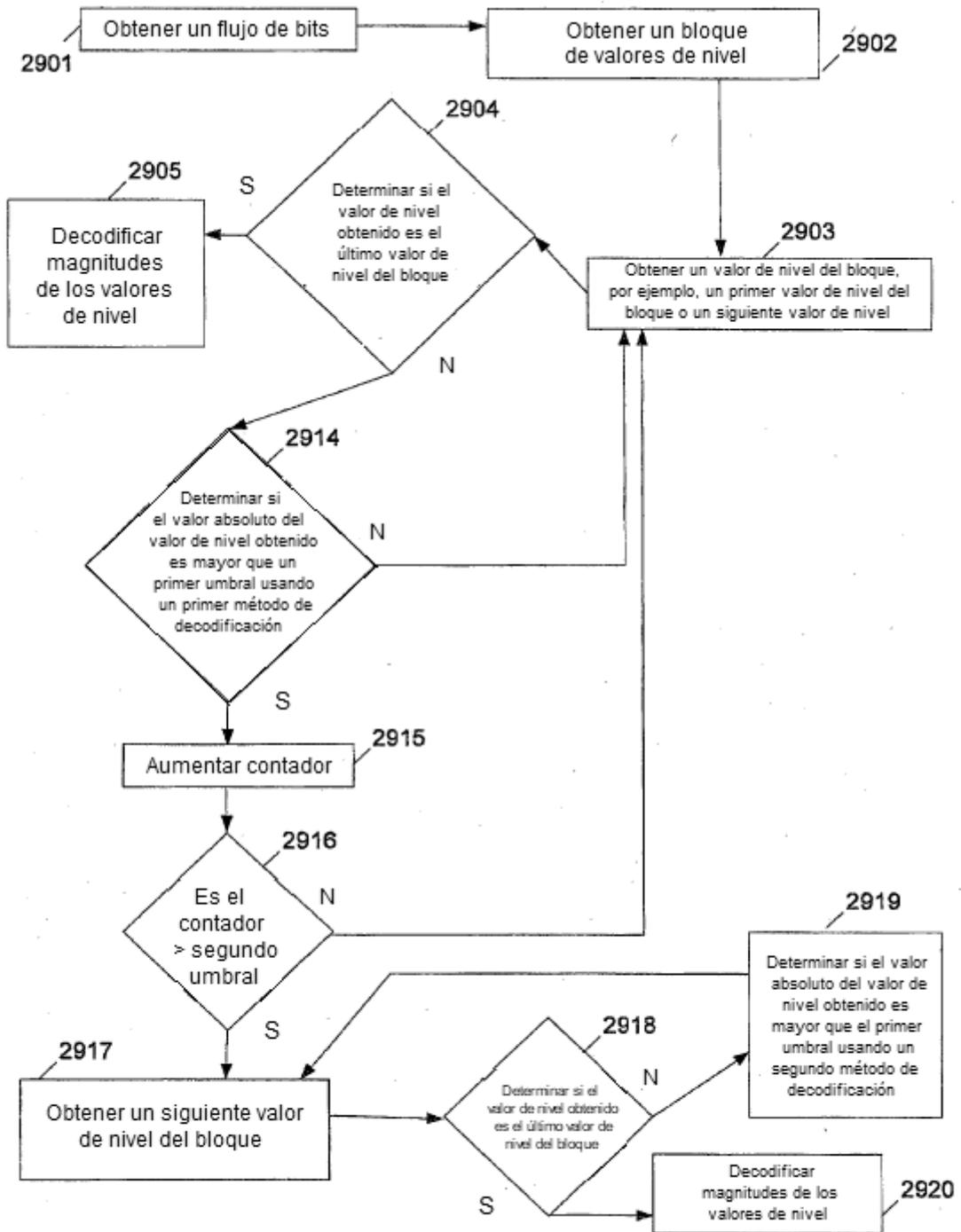


2400

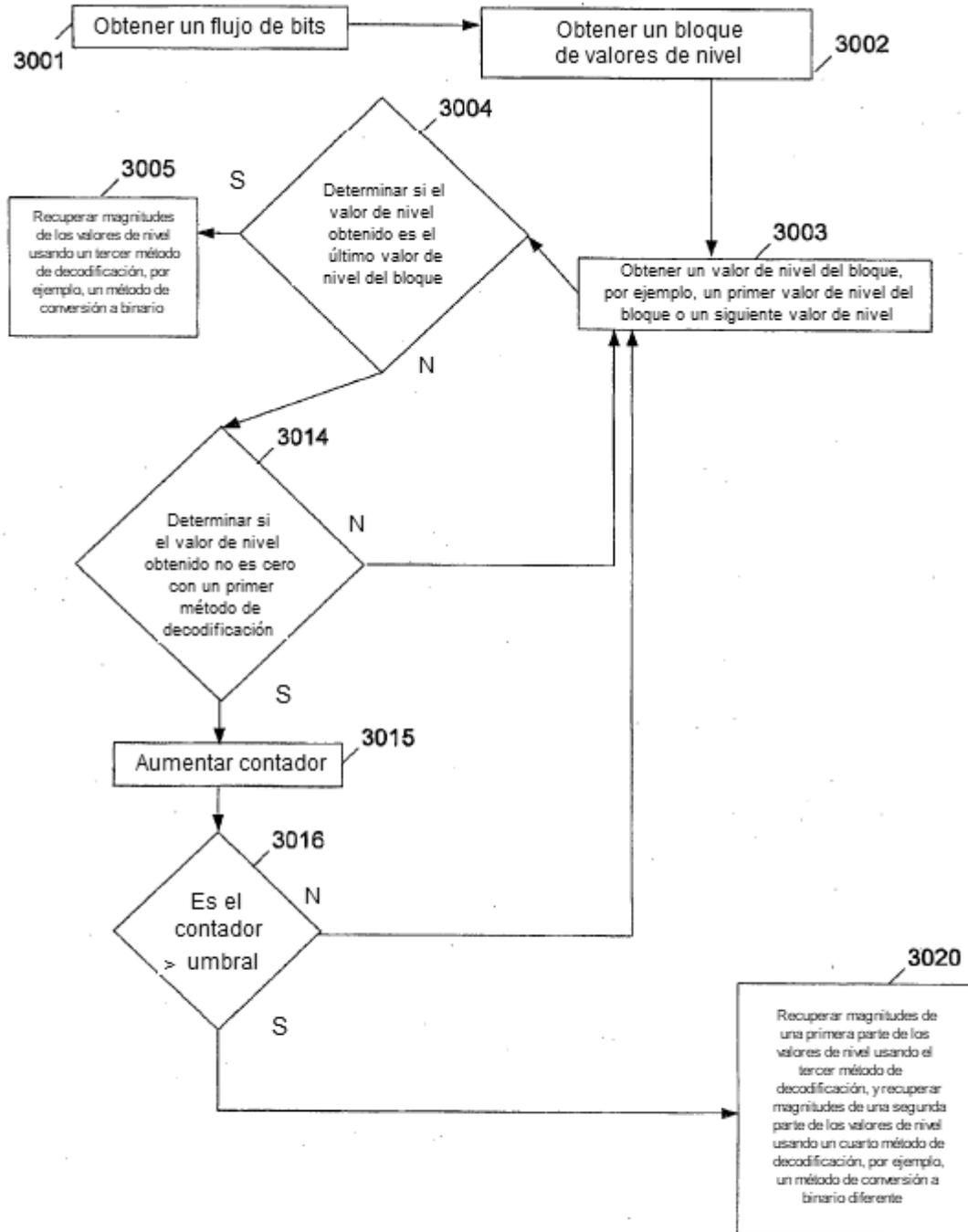
[Fig. 24]



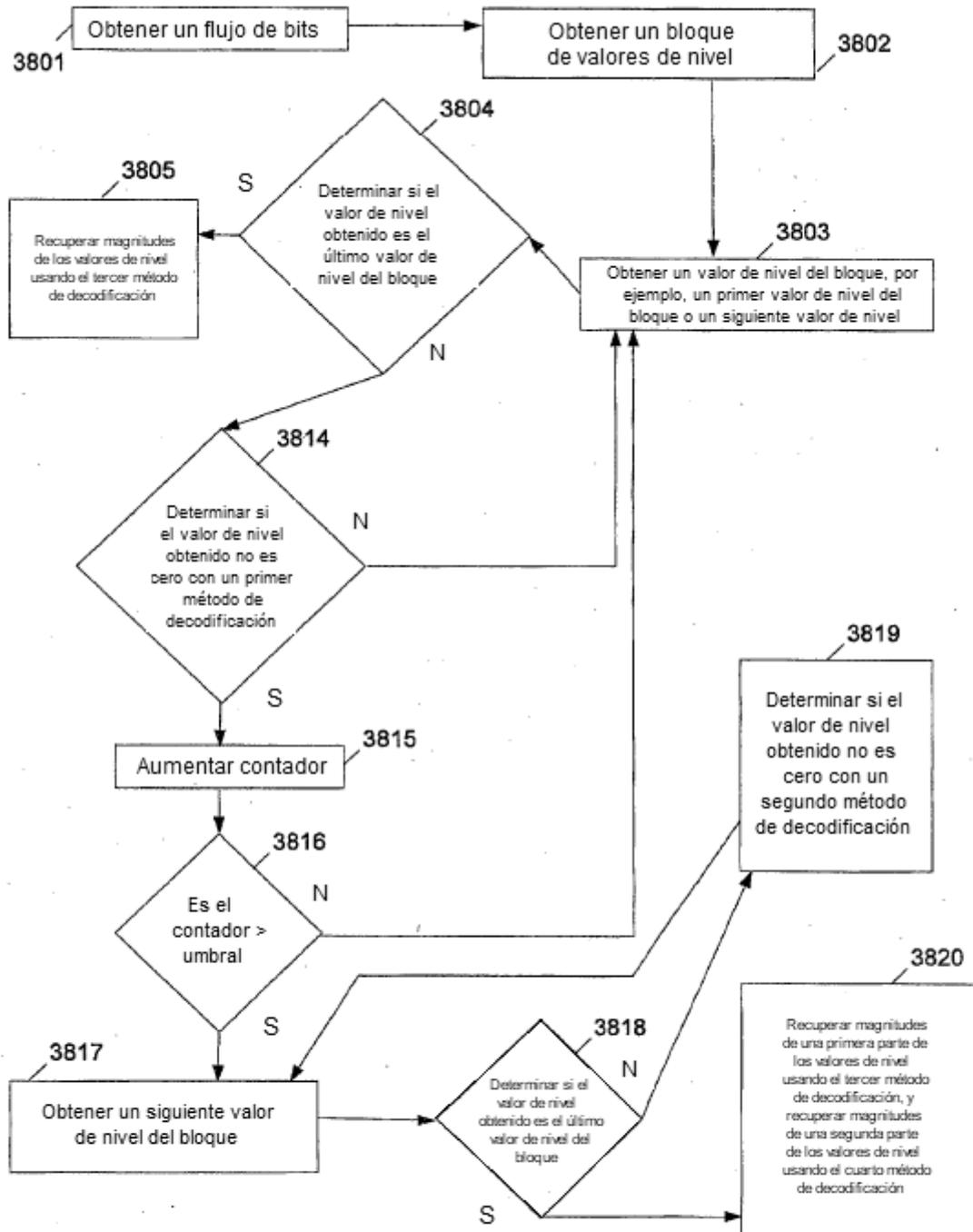
[Fig. 25]



[Fig. 26]

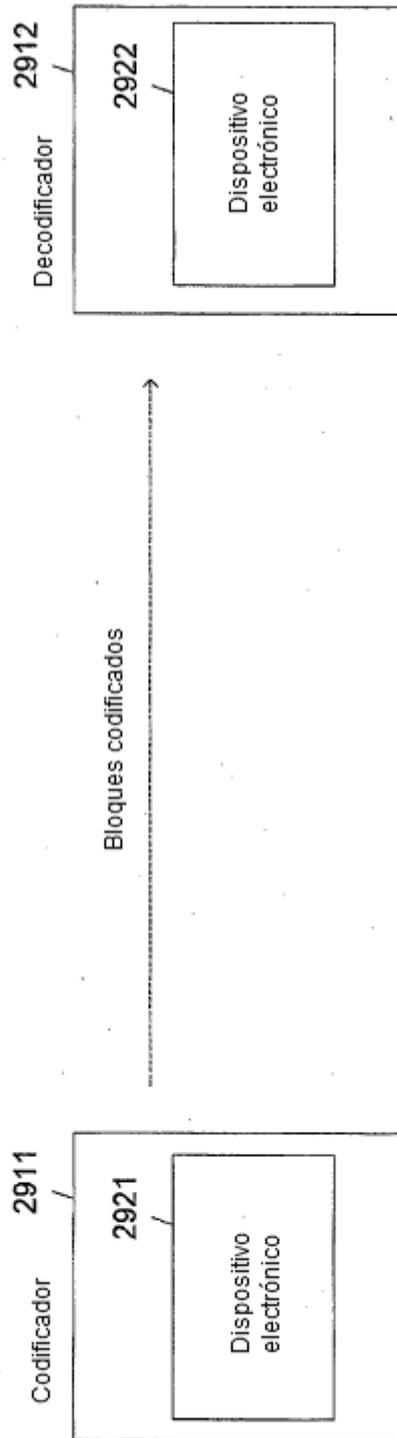


[Fig. 27]



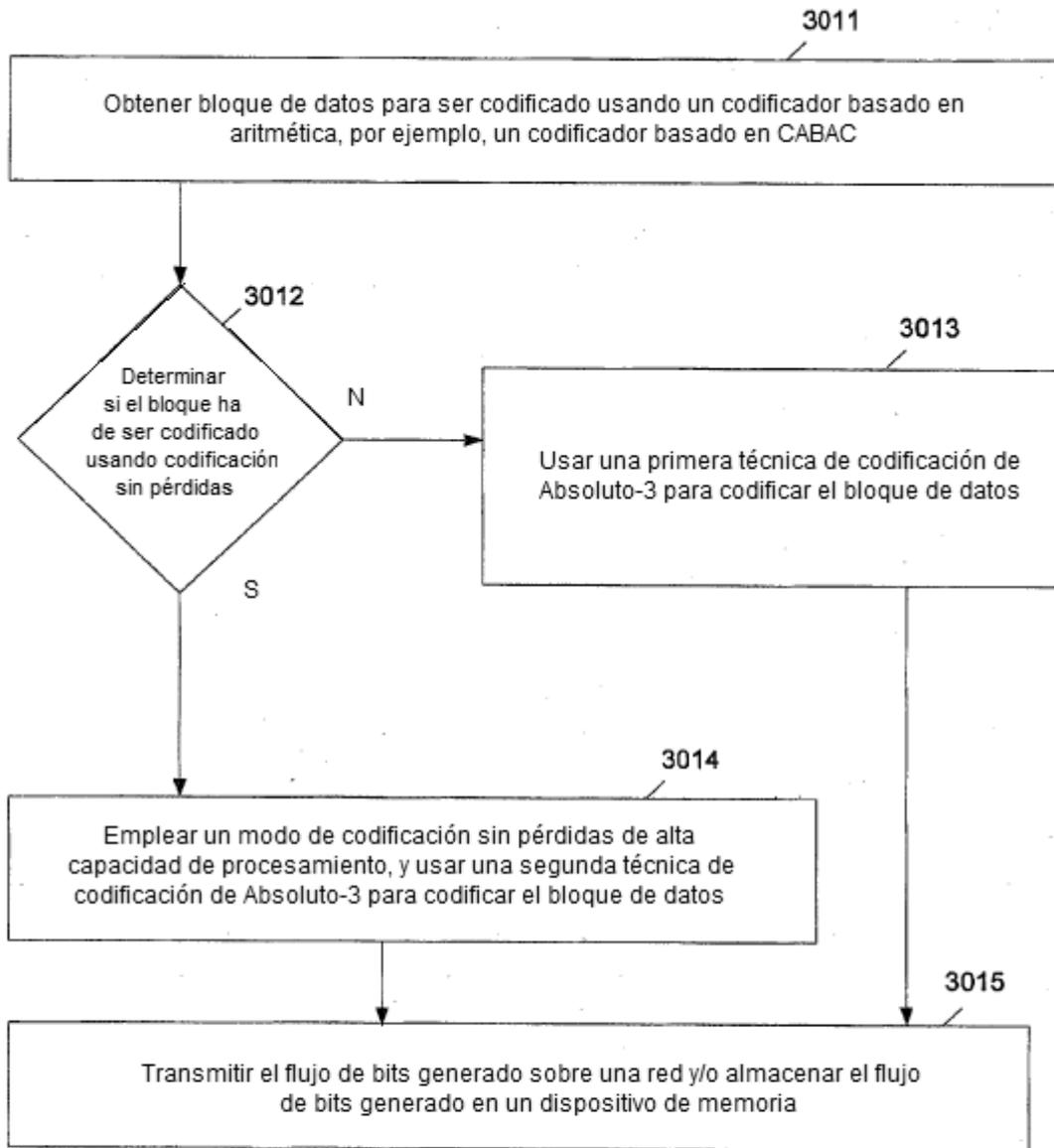


[Fig. 29]

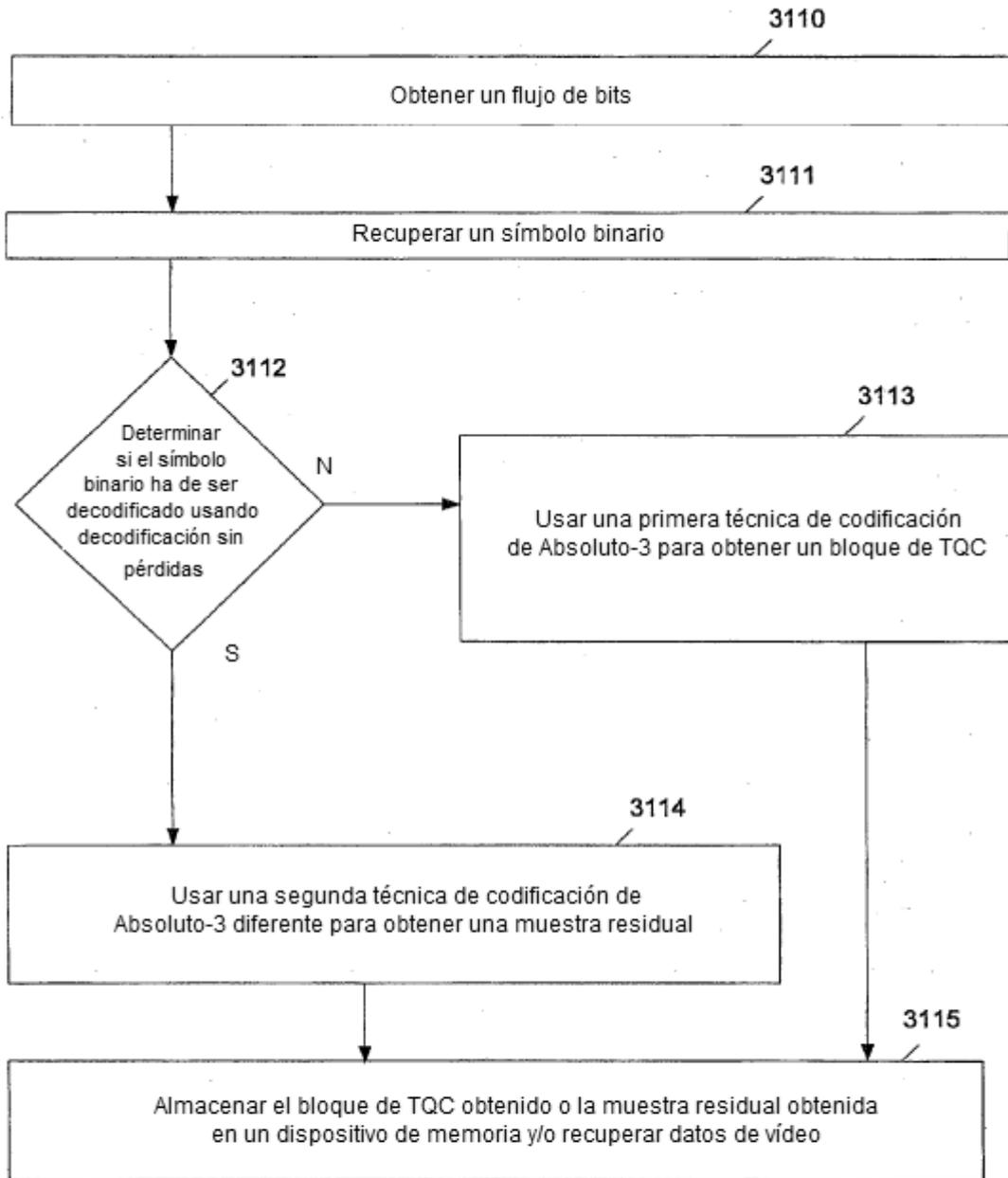


2900

[Fig. 30]



[Fig. 31]

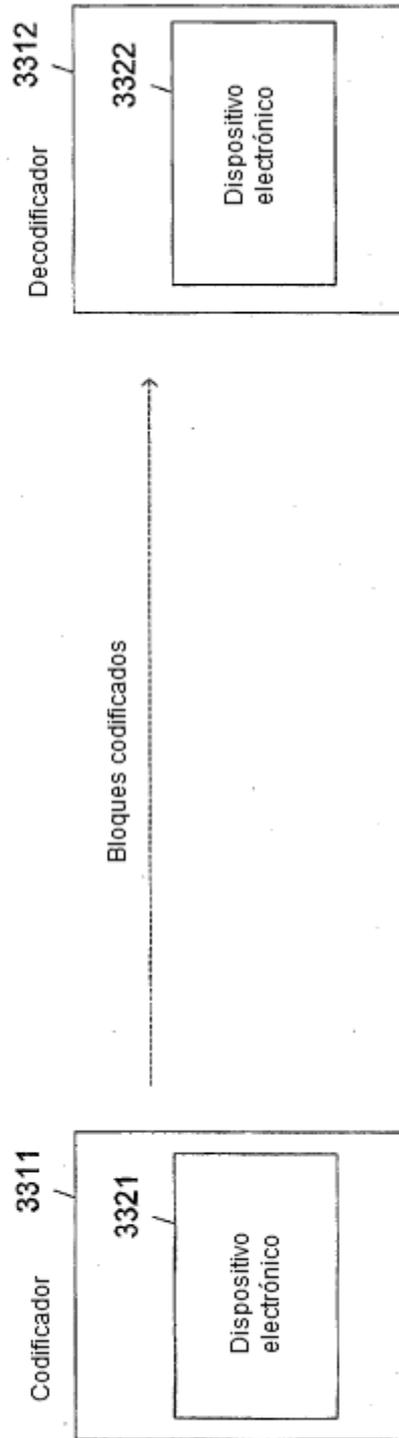


[Fig. 32]

	Codificación por derivación										Codificación basada en contexto					El último coeficiente no cero						
	0	1	2	3	4	5	6	7	8	9	10	11	1	2	3	4	5	1	2	3	4	5
Posición de escaneado	0	1	2	3	4	5	6	7	8	9	10	11	1	2	3	4	5	1	2	3	4	5
Nivel de Coeficiente	18	-9	6	7	4	-6	2	4	-4	-1	4	2						1	2	3	4	5
Mapa de Importancia	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Mayor_que_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Mayor_que_2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

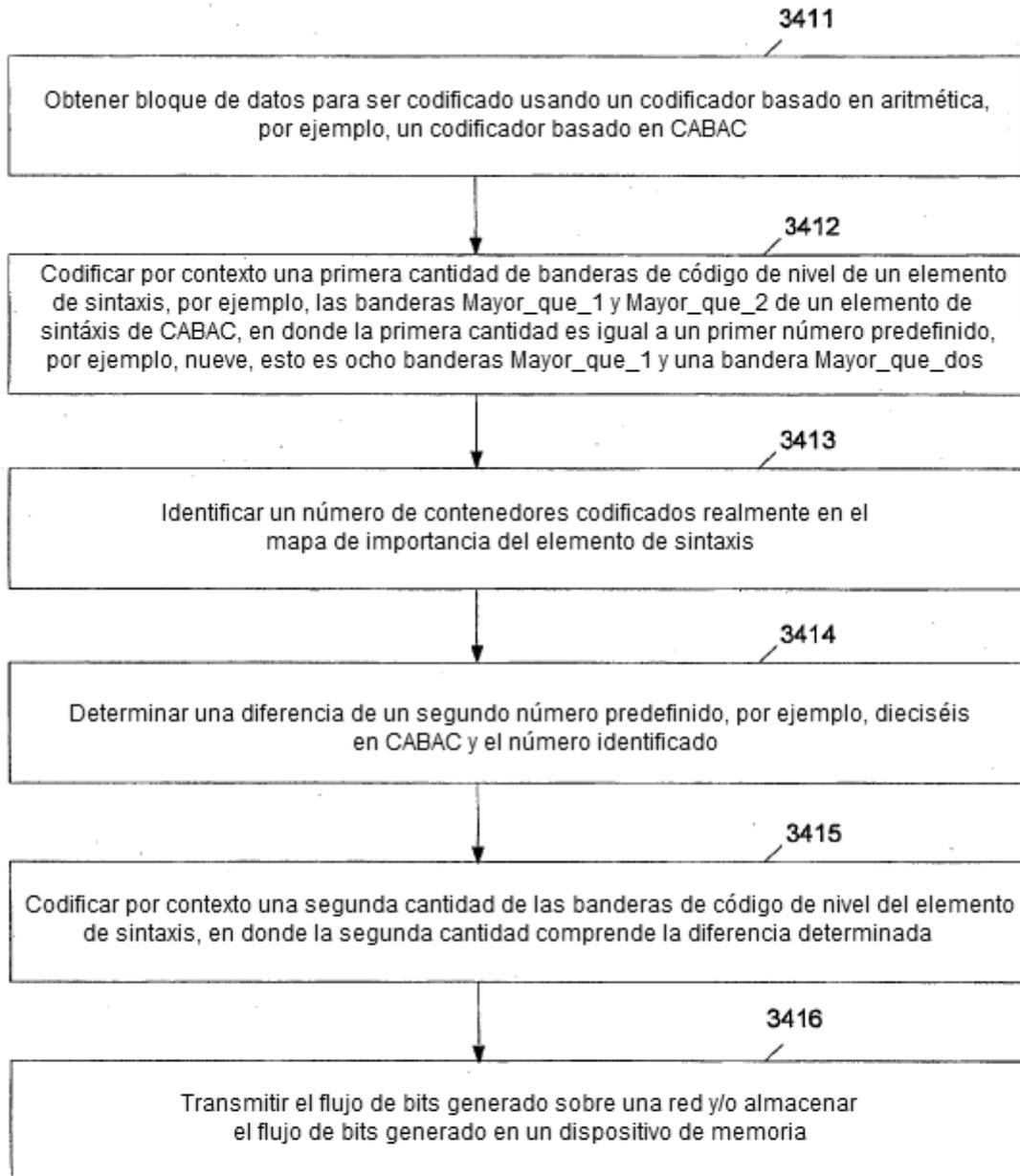
\* Los coeficientes se codifican en orden de escaneado inverso

[Fig. 33]

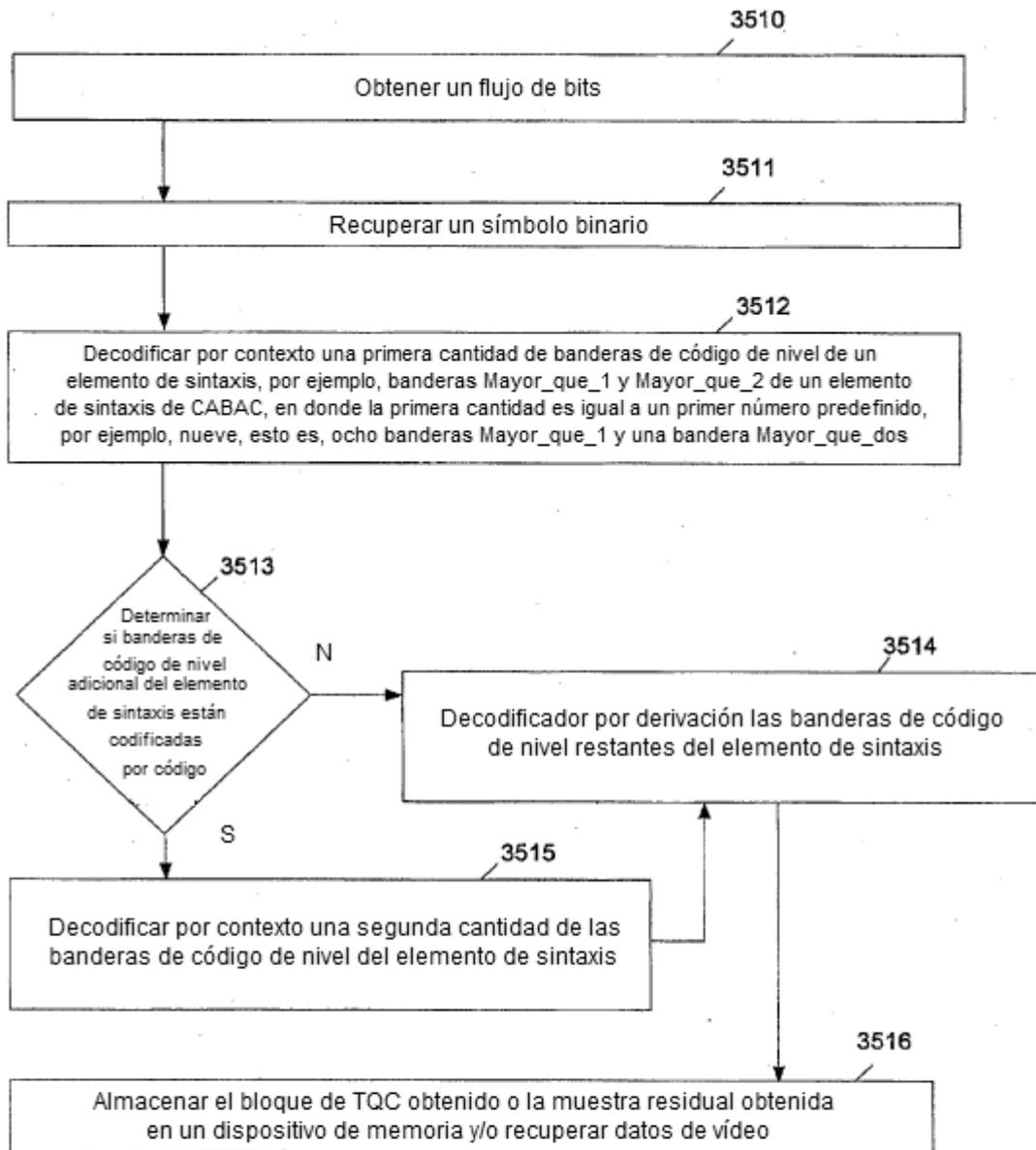


3300

[Fig. 34]



[Fig. 35]



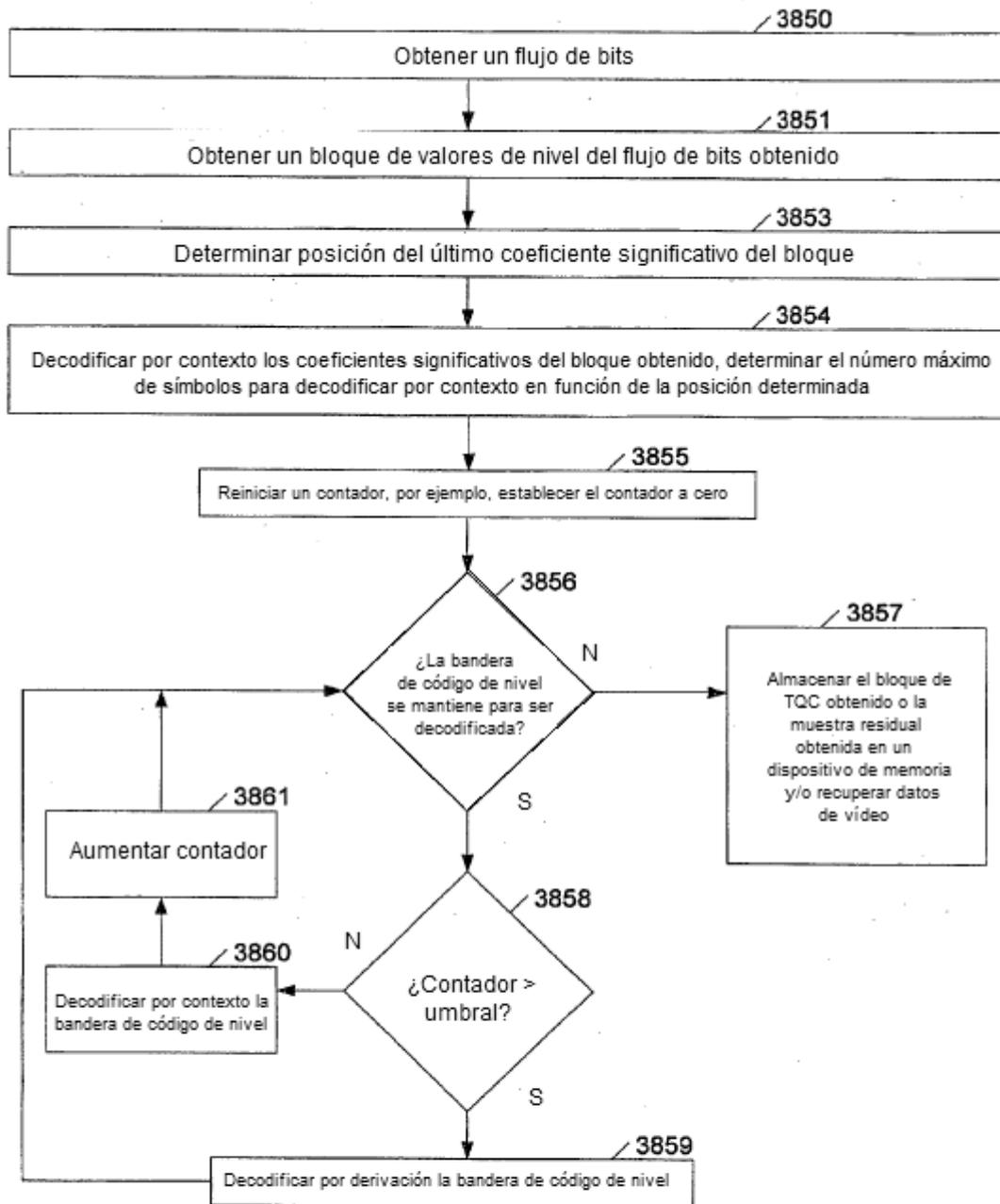
[Fig. 36]

	0	1	2	3	4	5	6	7	8	9	10	11	1	1	1	1	1
	18	-9	6	7	4	-6	2	4	-4	-1	4	2	-3	0	0	0	0
Posición de escaneado																	
Nivel de Coeficiente																	
Mapa de Importancia																	
Mayor_que_1																	
Mayor_que_2																	
Información de Signo																	
Absoluto-3																	

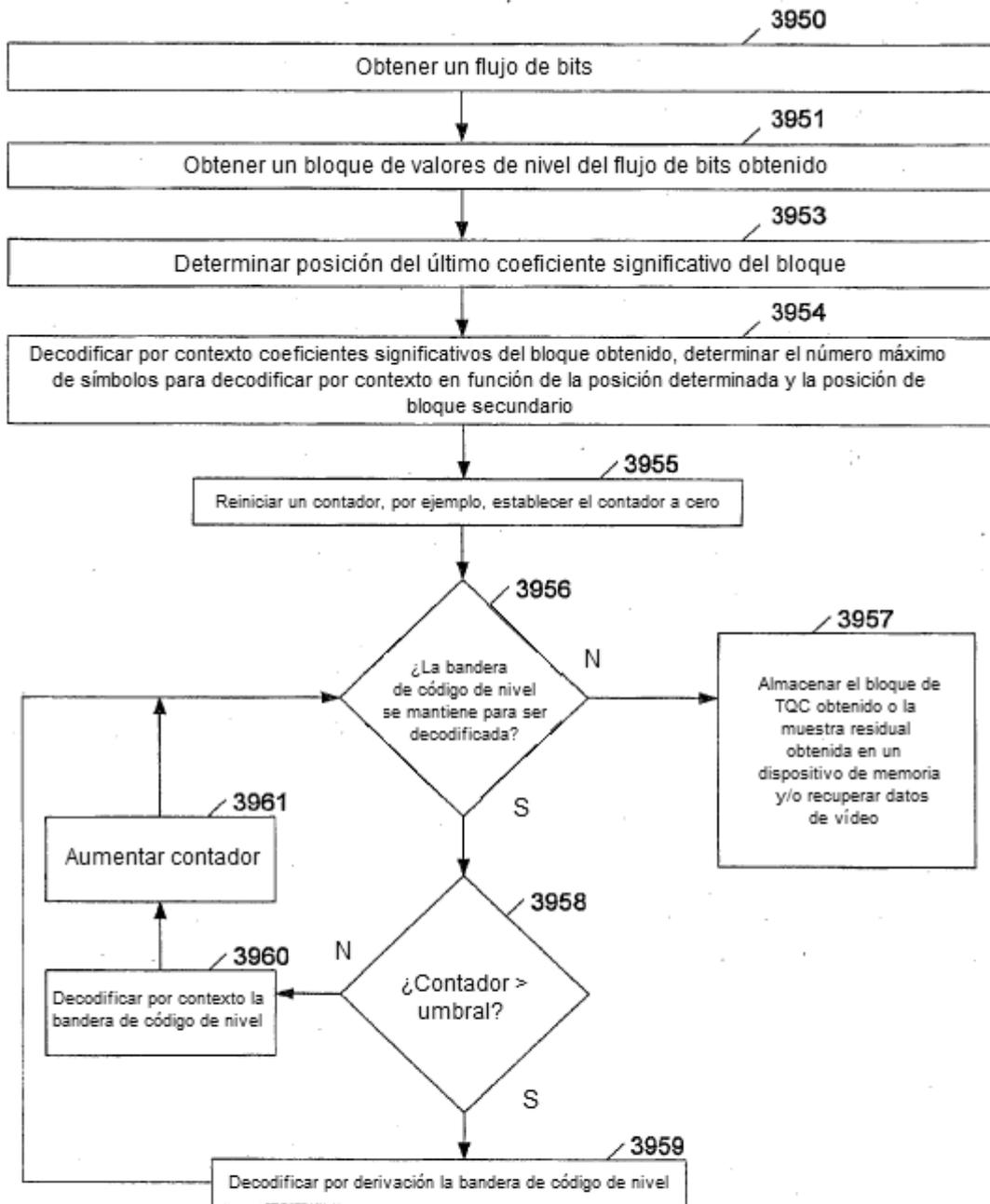
\* Los coeficientes se codifican en orden de escaneado inverso



[Fig. 38]



[Fig. 39]



[Fig. 40]

