

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 671 972**

51 Int. Cl.:

H04N 19/176 (2014.01)

H04N 19/70 (2014.01)

H04N 19/96 (2014.01)

H04N 19/119 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **11.12.2013 PCT/JP2013/007276**

87 Fecha y número de publicación internacional: **10.07.2014 WO14106878**

96 Fecha de presentación y número de la solicitud europea: **11.12.2013 E 13870017 (4)**

97 Fecha y número de publicación de la concesión europea: **21.03.2018 EP 2942958**

54 Título: **Señalización de división de unidad de codificación para bloques codificados PCM**

30 Prioridad:

07.01.2013 JP 2013000290

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

11.06.2018

73 Titular/es:

**NEC CORPORATION (100.0%)
7-1, Shiba 5-chome, Minato-ku
Tokyo 108-8001, JP**

72 Inventor/es:

CHONO, KEIICHI

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 671 972 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

Señalización de división de unidad de codificación para bloques codificados PCM

5 Sector técnico

La presente invención se refiere a un dispositivo de codificación de video y un dispositivo de decodificación de video para dividir automáticamente una unidad de árbol de codificación.

10 Antecedentes de la técnica

15 Un sistema de codificación de video basado en un sistema descrito en la bibliografía no de patentes (NPL) 1 divide cada cuadro de video digitalizado en unidades de árbol de codificación (CTUs), y cada CTU es codificada en un orden de barrido de trama. Cada CTU es dividida en unidades de codificación (CU) en una estructura de árbol cuaternario y codificada. Cada CU es dividida en unidades de predicción (PU) y predicha. Además, un error de predicción de cada CU es dividido en unidades de transformada (TU) en una estructura de árbol cuaternario y transformada.

20 La CU es una unidad de codificación de intra-predicción/predicción inter-cuadros. La intra-predicción y la predicción inter-cuadros se describirán a continuación.

25 La intra-predicción es una predicción a partir de una imagen reconstruida de un cuadro a codificar. En el sistema descrito en la NPL 1, un píxel reconstruido en torno a un bloque a codificar es extrapolado para generar una señal de intra-predicción. A continuación, una CU que utiliza intra-predicción se denomina una intra-CU. Se debe observar que el valor de sintaxis `pred_mode_flag` de la intra-CU es 1 en NPL 1.

30 Además, una intra-CU que no utiliza intra-predicción se denomina una CU I_PCM (modulación por impulsos intra codificados). En la CU I_PCM, se transmite intacta una imagen de la CU intacta en lugar de transmitirse un error de predicción de la CU. Se debe observar que el valor de sintaxis `pcm_flag` de la CU I_PCM es 1 en NPL 1.

35 La predicción inter-cuadros es una predicción basada en una imagen de un cuadro reconstruido (imagen de referencia) de diferente tiempo de visualización que un cuadro a codificar. En lo que sigue, la predicción inter-cuadros se denomina asimismo inter-predicción. La figura 10 es un diagrama explicativo que representa un ejemplo de predicción inter-cuadros. Un vector de movimiento $MV = (mv_x, mv_y)$ indica la magnitud de la traslación de un bloque de imagen reconstruido de una imagen de referencia con respecto a un bloque a codificar. En inter-predicción, se genera una señal de inter-predicción en base a un bloque de imagen reconstruido de una imagen de referencia (utilizando interpolación de píxeles si es necesario). En lo que sigue, una CU que utiliza inter-predicción se denomina una inter-CU. Se debe observar que el valor de sintaxis `pred_mode_flag` de la inter-CU es 0 en NPL 1.

40 Una inter-CU que utiliza predicción inter-cuadros para no transmitir información de diferencia de vector de movimiento y un error de predicción CU se denomina una CU de salto. En NPL 1, el valor de sintaxis `skip_flag` de la CU de salto es 1.

45 Un cuadro codificado solamente con las intra-CU mencionadas anteriormente se denomina un cuadro I (o una imagen I). Un cuadro codificado incluyendo tanto inter-CU como intra-CU se denomina un cuadro P (o una imagen P). Un cuadro codificado que incluye inter-CU para las que se utiliza no solamente una imagen de referencia sino simultáneamente dos imágenes de referencia para inter-predicción de un bloque se denomina un cuadro B (o una imagen B).

50 La intra-predicción y la predicción inter-cuadros son tal como se ha descrito anteriormente.

Haciendo referencia a la figura 11, se describirá la configuración y el funcionamiento de un dispositivo de codificación de video convencional que recibe cada CU de cada cuadro de video digitalizado como una imagen de entrada y entrega un flujo de bits.

55 El dispositivo de codificación de video representado en la figura 11 incluye un transformador/cuantificador 102, un codificador entrópico 103, un transformador inverso/cuantificador inverso 104, una memoria tampón 105, un predictor 106, un codificador PCM 107, un decodificador PCM 108, un selector de datos multiplexados 109, un multiplexor 110, un conmutador 121 y un conmutador 122.

60 Tal como se representa en la figura 12, un cuadro se compone de LCU (unidades máximas de codificación). Una LCU se compone de CU (unidades de codificación). La figura 12 es un diagrama explicativo que representa un ejemplo de división CTU de un cuadro *t* y un ejemplo de división CU de CTU8 en el cuadro *t* cuando la resolución espacial del cuadro es CIF (formato intermedio común) y el tamaño de la CTU es de 64. Una estructura de árbol cuaternario de CTU8 se puede representar mediante `cu_split_flag=1` a `CUDepth=0` que indica que una zona de 64x64 está dividida, tres `cu_split_flag=0` a `CUDepth=1` que indican que las primeras tres CU de 32x32 (CU0, CU1 y

CU2) no están divididas, $cu_split_flag=1$ a $CUDepth=1$ que indica que la última CU de 32×32 está dividida, tres $cu_split_flag=0$ a $CUDepth=2$ que indican que las primeras tres CU de 16×16 (CU3, CU4 y CU5) no están divididas, $cu_split_flag=1$ a $CUDepth=2$ que indica que la última CU de 16×16 está dividida y cuatro $cu_split_flag = 0$ a $CUDepth = 3$ que indican que ninguna de las CU de 8×8 (CU6, CU7, CU8 y CU9) está dividida.

5 El dispositivo de codificación de video representado en la figura 11 codifica las LCU en orden de barrido de trama, y codifica las CU constituyendo cada LCU en orden de barrido en z. El tamaño de una CU es cualquiera de 64×64 , 32×32 , 16×16 y 8×8 . La CU mínima se denomina una unidad de codificación mínima (SCU).

10 El transformador/cuantificador 102 transforma en frecuencia una imagen (imagen de error de predicción) de la que se resta una señal de predicción para obtener un coeficiente de transformada de frecuencia de una imagen de error de predicción.

15 El transformador/cuantificador 102 cuantifica adicionalmente el coeficiente de transformada de frecuencia con un tamaño de paso de cuantificación Q_s predeterminado. En adelante, el coeficiente de transformada de frecuencia cuantificado se denomina un valor de cuantificación de coeficiente o un valor de nivel de cuantificación.

20 El codificador entrópico 103 realiza codificación entrópica de un parámetro de predicción y el valor del nivel de cuantificación. El parámetro de predicción es información relacionada con información del tipo de predicción (intra-predicción o inter-predicción) de una CU mencionada anteriormente y de PU (unidades de predicción) incluidas en la CU.

25 El transformador inverso/cuantificador inverso 104 invierte-cuantifica el valor del nivel de cuantificación con el tamaño del paso de cuantificación Q_s . El transformador inverso/cuantificador inverso 104 realiza una transformada inversa en frecuencia del coeficiente de transformada de frecuencia obtenido por la cuantificación inversa. La señal de predicción se añade a una imagen de error de predicción reconstruida obtenida por la transformada inversa, y la imagen de error de predicción reconstruida es suministrada al conmutador 122.

30 El selector de datos multiplexados 109 monitoriza la cantidad de datos de entrada del codificador entrópico 103 correspondientes una CU a codificar. Cuando el codificador entrópico 103 puede realizar codificación entrópica de los datos de entrada dentro del tiempo de procesamiento de la CU, el selector de datos multiplexados 109 selecciona los datos de salida del codificador entrópico 103, y suministra los datos de salida al multiplexor 110 a través del conmutador 121. El selector de datos multiplexados 109 selecciona además datos de salida del transformador inverso/cuantificador inverso 104, y suministra los datos de salida a la memoria tampón 105 a través del conmutador 122.

35 Cuando el codificador entrópico 103 no puede realizar codificación entrópica de los datos de entrada dentro del tiempo de procesamiento de la CU, el selector de datos multiplexados 109 selecciona datos de salida del codificador PCM 107 y suministra los datos de salida al multiplexor 110 a través del conmutador 121. El selector de datos multiplexados 109 selecciona además datos de salida obtenidos por el decodificador PCM 108 que decodifica PCM los datos de salida del codificador PCM 107, y suministra los datos de salida a la memoria tampón 105 a través del conmutador 122.

40 La memoria tampón 105 almacena una imagen reconstruida suministrada a través del conmutador 122. Una imagen reconstruida para un cuadro se denomina una imagen reconstruida.

45 El multiplexor 110 multiplexa datos de salida del codificador entrópico 103 y del codificador PCM 107, y emite los datos de salida multiplexados.

50 En base al funcionamiento mencionado anteriormente, el multiplexor 110 del dispositivo de codificación de video genera un flujo de bits.

Lista de referencias

55 Bibliografía no de patentes

60 NPL 1: Benjamin Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan y Thomas Wiegand, "High Efficiency Video Coding (HEVC) text specification draft 9", JCTVC-K1003_v10, equipo colaborativo conjunto sobre codificación de video (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11 sexto congreso: Shanghai, CN, 10 a 19 de octubre de 2012.

Compendio de la invención

Problema técnico

65

En el dispositivo de codificación de video representado en la figura 11, la cabecera de bloque de una CU de codificación entrópica. En otras palabras, cuando se va transmitir la cabecera de bloque (cabecera de modo I_PCM) de una CU I_PCM son necesarios dos procesos, es decir, un proceso de emisión de símbolos no emitidos de un motor de codificación del codificador entrópico 103 (después de que pcm_flag se haya sometido a codificación entrópica), y un proceso de reseteo del motor de codificación. Tal como es evidente por la sección 7.3.9.4 Sintaxis de árbol cuaternario de codificación, y por la sección 7.3.9.5 Sintaxis de unidad de codificación, en NPL 1 la cabecera de modo I_PCM está constituida generalmente por información de división de CU (sintaxis cu_split_flag), el tipo de predicción de CU (sintaxis skip_flag, sintaxis pred_mode_flag y sintaxis part_mode) y un indicador PCM (sintaxis pcm_flag) para una PU.

Haciendo referencia a la figura 13, se describirá la codificación de las CU I_PCM sucesivas en orden de codificación en un CUT. Tal como se representa en (A) de la figura 13, se supone que, en una CUT a codificar, las CU (CU(0), CU(1), CU(2) y CU(3)) de 0, 1, 2 y 3 sucesivas en orden de codificación en la CTU son las CU I_PCM. Un flujo de bits de CTU correspondiente se constituye entrelazando un flujo de bits de video de la cabecera de modo I_PCM de cada CU, datos de alineamiento (pcm_alignment_zero_bit) y datos PCM (pcm_sample_luma/pcm_sample_chroma). En otras palabras, el flujo de bits de CTU representado en (A) de la figura 13 se compone de: cu_split_flag=1, cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit y pcm_sample_luma/pcm_sample_chroma de CU(0); cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit y pcm_sample_luma/pcm_sample_chroma de CU(1); cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit y pcm_sample_luma/pcm_sample_chroma de CU(2); y cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit y pcm_sample_luma/pcm_sample_chroma de CU(3). Tal como resulta evidente a partir de la sección 7.3.9.5 Sintaxis de unidad de codificación, y de la sección 7.3.9.7 Sintaxis de muestra PCM en NPL 1, pcm_alignment_zero_bit, pcm_sample_luma y pcm_sample_chroma son sintaxis no para codificación entrópica.

Dado que la cabecera de modo I_PCM de cada CU(0), CU(1), CU(2) y CU(3) de I_PCM sucesiva en orden de codificación en la CTU es de codificación entrópica, se repite el proceso de entrega símbolos no entregados del motor de codificación y el proceso de resetear el motor de codificación mencionados anteriormente. Por lo tanto, aunque las CU(0), CU(1), CU(2) y CU(3) son I_PCM sucesivas en orden de codificación en la CTU, las CU tienen que esperar hasta la finalización del proceso de entrega de símbolos no entregados del motor de codificación y del proceso de reseteo del motor de codificación mencionados anteriormente, ejecutados en cada una de las cabeceras de modo I_PCM, antes de que sean multiplexados los datos PCM en cada CU. En la tecnología convencional, existe el problema de que los datos PCM en las CU de I_PCM sucesivas en el orden de codificación en la CTU no se pueden multiplexar de manera eficiente en un flujo de bits, es decir, los datos PCM en las CU de I_PCM sucesivas en orden de codificación en la CTU no se pueden transmitir eficientemente.

De manera similar, tal como se representa en (B) de la figura 13, existe también el mismo problema con las CU de I_PCM sucesivas en orden de codificación en una CTU incompleta en el límite derecho de la imagen. En (B) de la figura 13, "Inferir" indica que el valor es determinado por el descodificador sin transmitir cu_split_flag en base a la posición (x0 y y0) y al tamaño de bloque (log2CbSize) de una CU a procesar, al tamaño de un cuadro de imagen (la anchura pic_width_in_luma_samples y la altura pic_height_in_luma_samples) y al tamaño de CU mínimo (Log2MinCbSizeY) (ver la sección 7.3.9.5 Sintaxis de unidad de codificación y la sección 7.4.9.4 Semántica del árbol cuaternario de codificación en NPL 1 para más detalle). Además, en (B) de la figura 13, "N" indica que no existe ninguna CU en la posición afectada y, por lo tanto, no se transmite cu_split_flag redundante.

El flujo de bits de CTU representado en (B) de la figura 13 se compone de: cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit y pcm_sample_luma/pcm_sample_chroma de CU(0); cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit y pcm_sample_luma/pcm_sample_chroma de CU(1); cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit y pcm_sample_luma/pcm_sample_chroma de CU(2); cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit y pcm_sample_luma/pcm_sample_chroma de CU(3); cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit y pcm_sample_luma/pcm_sample_chroma de CU(4); y cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit y pcm_sample_luma/pcm_sample_chroma de CU(5).

Un objetivo de la presente invención es dar a conocer un dispositivo de codificación de video, un dispositivo de descodificación de video, un procedimiento de codificación de video, un procedimiento de descodificación de video, un programa de codificación de video y un programa de descodificación de video que puedan transmitir datos PCM en CU I_PCM de manera eficiente incluso cuando las CU de I_PCM se generan sucesivamente en orden de codificación en una CTU.

Solución al problema

5 Un dispositivo de codificación de video según la presente invención incluye: medios de transformada para transformar un bloque de imagen; medios de codificación entrópica para someter a codificación entrópica datos transformados, en el bloque de imagen transformado por los medios de transformada; medios de codificación PCM para codificar PCM el bloque de imagen; medios de selección de datos multiplexados para seleccionar, para cada bloque de imagen, datos de salida de los medios de codificación entrópica o bien datos de salida de los medios de codificación PCM; y medios de multiplexación para multiplexar sintaxis `ctu_pcm_flag` que indica si todos los bloques de imagen pertenecientes a una unidad de árbol de codificación como un conjunto de bloques de imagen seleccionados por los medios de selección de datos multiplexados son bloques PCM, en la parte superior de la unidad de árbol de codificación, donde cuando todos los bloques de imagen que pertenecen a la unidad de árbol de codificación son bloques PCM, los medios de multiplexación no multiplexan por lo menos una sintaxis `cu_split_flag` indicativa de la estructura de división de la unidad de árbol de codificación.

15 Un dispositivo de descodificación de video según la presente invención incluye: medios de desmultiplexación para desmultiplexar un flujo de bits que incluye una sintaxis `ctu_pcm_flag` que indica si todos los bloques de imagen que pertenecen a la unidad de árbol de codificación a descodificar son bloques PCM; medios de descodificación PCM para descodificar PCM datos PCM en un bloque de imagen incluido en el flujo de bits; medios de descodificación entrópica para someter a descodificación entrópica datos transformados, en el bloque de imagen incluido en el flujo de bits; y medios de control de descodificación que, cuando el valor de sintaxis `ctu_pcm_flag` de la unidad de árbol de codificación a descodificar es 1, determinan la sintaxis `cu_split_flag` indicativa de la estructura de división de la unidad de árbol de codificación en base a la posición del bloque de imagen en la unidad de árbol de codificación, al tamaño de un cuadro de imagen a descodificar y al tamaño máximo de bloque PCM.

25 Un procedimiento de codificación de video según la presente invención incluye: transformar un bloque de imagen; someter a codificación entrópica datos transformados, en el bloque de imagen transformado; codificar PCM bloques de imagen; seleccionar, para cada bloque de imagen, datos sometidos a realizar codificación entrópica o bien datos codificados PCM; y multiplexar sintaxis `ctu_pcm_flag` que indica si todos los bloques de imagen que pertenecen a una unidad de árbol de codificación como un conjunto de bloques de imagen son bloques PCM, en la parte superior de la unidad de árbol de codificación, donde cuando todos los bloques de imagen que pertenecen a la unidad de árbol de codificación son bloques PCM, por lo menos la sintaxis `cu_split_flag` indicativa de la estructura de división de la unidad de árbol de codificación no es multiplexada.

35 Un procedimiento de descodificación de video según la presente invención incluye: desmultiplexar un flujo de bits que incluye una sintaxis `ctu_pcm_flag` que indica si todos los bloques de imagen que pertenecen a una unidad de árbol de codificación a descodificar son bloques PCM; descodificar PCM datos PCM en un bloque de imagen incluido en el flujo de bits; someter a descodificación entrópica datos transformados, en el bloque de imagen incluido en el flujo de bits; y cuando el valor de la sintaxis `ctu_pcm_flag` de la unidad de árbol de codificación a descodificar es 1, determinar la sintaxis `cu_split_flag` indicativa de la estructura de división de la unidad de árbol de codificación en base a la posición del bloque de imagen en la unidad de árbol de codificación, al tamaño de un cuadro de imagen a descodificar y al tamaño máximo de bloque PCM.

45 Un programa de codificación de video según la presente invención hace que un ordenador ejecute: un proceso de transformación de un bloque de imagen; un proceso de codificación entrópica de datos transformados en el bloque de imagen transformado; un proceso de codificación PCM del bloque de imagen; un proceso de selección, para cada bloque de imagen, de datos de codificación entrópica o bien datos codificados PCM; y un proceso de multiplexación de sintaxis `ctu_pcm_flag` que indica si todos los bloques de imagen que pertenecen a una unidad de árbol de codificación como un conjunto de bloques de imagen son bloques PCM, en la parte superior de la unidad de árbol de codificación, en el que cuando todos los bloques de imagen que pertenecen a la unidad de árbol de codificación son bloques PCM, se hace que el ordenador no multiplexe por lo menos la sintaxis `cu_split_flag` indicativa de la estructura de división de la unidad de árbol de codificación.

55 Un programa de descodificación de video según la presente invención hace que un ordenador ejecute: un proceso de desmultiplexación de un flujo de bits que incluye una sintaxis `ctu_pcm_flag` que indica si todos los bloques de imagen que pertenecen a una unidad de árbol de codificación a descodificar son bloques PCM; un proceso de descodificación PCM de datos PCM en un bloque de imagen incluido en el flujo de bits; un proceso de descodificación entrópica de datos transformados, en el bloque de imagen incluido en el flujo de bits; y cuando el valor de la sintaxis `ctu_pcm_flag` de la unidad de árbol de codificación a descodificar es 1, un proceso de determinar la sintaxis `cu_split_flag` indicativa de la estructura de división de la unidad de árbol de codificación en base a la posición del bloque de imagen en la unidad de árbol de codificación, al tamaño de un cuadro de imagen a descodificar y al tamaño máximo de bloque PCM.

Resultados ventajosos de la invención

65 De acuerdo con la presente invención, cuando todas las CU de una CTU son I_PCM, estos elementos de datos PCM se pueden transmitir de manera eficiente.

Breve descripción de los dibujos

- 5 [Figura 1] Representa un diagrama de bloques que describe un dispositivo de codificación de video de una primera realización a modo de ejemplo.
- [Figura 2] Representa un diagrama explicativo de codificación de CU I_PCM.
- [Figura 3] Representa un diagrama de flujo que describe el funcionamiento del dispositivo de codificación de video de la primera realización a modo de ejemplo.
- 10 [Figura 4] Representa un diagrama de bloques que describe un dispositivo de descodificación de video de una segunda realización a modo de ejemplo.
- [Figura 5] Representa un diagrama de flujo que describe el funcionamiento del dispositivo de descodificación de video de la segunda realización a modo de ejemplo.
- [Figura 6] Representa un diagrama explicativo de una lista que representa a sintaxis de unidad de árbol de codificación.
- 15 [Figura 7] Representa un diagrama explicativo de una lista que representa sintaxis de árbol cuaternario PCM.
- [Figura 8] Representa un diagrama explicativo de una lista que representa sintaxis de muestra PCM.
- [Figura 9] Representa un diagrama de bloques que describe un sistema de procesamiento de información que utiliza un programa.
- 20 [Figura 10] Representa un diagrama explicativo que describe un ejemplo de predicción inter-cuadros.
- [Figura 11] Representa un diagrama de bloques que describe un dispositivo de codificación de video convencional.
- [Figura 12] Representa un diagrama explicativo que describe un ejemplo de división CTU de un cuadro t, y un ejemplo de división CU de CTU8 en el cuadro t.
- 25 [Figura 13] Representa un diagrama explicativo de codificación de CU I_PCM.

Descripción de realizaciones

Realización 1 a modo de ejemplo

30 La figura 1 es un diagrama de bloques que representa un dispositivo de codificación de video de una primera realización a modo de ejemplo. El dispositivo de codificación de video de la realización a modo de ejemplo incluye un multiplexor 1100 en lugar del multiplexor 110 del dispositivo de codificación de video representado en la figura 11. A diferencia del multiplexor 110 representado en la figura 11, el multiplexor 1100 de la realización a modo de ejemplo incluye un determinador PCM 1101 que determina si todas las CU en una CTU a codificar son I_PCM. En la figura 1 está representado un ejemplo en el que el determinador PCM 1101 está presente en el multiplexor 1100, pero esto no es indispensable. El determinador PCM 1101 puede estar dispuesto en una parte diferente al multiplexor 1100, o puede estar dispuesto independientemente de cada unidad representada en la figura 1.

40 Se supone que el dispositivo de codificación de video de la realización a modo de ejemplo transmite sucesivas CU I_PCM en orden de transmisión en cada CTU. En la realización a modo de ejemplo, se supone asimismo, tal como se representa en (A) de la figura 2 (igual que en (A) de la figura 13 que las CU(0), CU(1), CU(2) y CU(3) sucesivas en orden de codificación son bloques a codificar PCM.

45 El funcionamiento del dispositivo de codificación de video se describirá haciendo referencia a un diagrama de flujo de la figura 3 tomando como ejemplo las CU I_PCM representadas en (A) de la figura 2.

50 El dispositivo de codificación de video de la realización a modo de ejemplo funciona tal como se representa en el diagrama de flujo de la figura 3. En otras palabras, el determinador PCM 1101 determina en la etapa S101 si todas las CU en una CTU a codificar son I_PCM. Cuando todas las CU son I_PCM, la sintaxis `ctu_pcm_flag` se ajusta a 1. De lo contrario, la sintaxis `ctu_pcm_flag` se ajusta a 0. Cuando se utiliza el ejemplo representado en (A) de la figura 2, el valor de `ctu_pcm_flag` se hace 1. En este caso, el codificador entrópico 103 somete a codificación entrópica `ctu_pcm_flag=1` en la etapa S102. El codificador entrópico 103 entrega además símbolos que todavía no han sido entregados. Cuando el valor de `ctu_pcm_flag` es 0, aunque el proceso se complete en el diagrama de flujo de la figura 3, el dispositivo de codificación de video de la realización a modo de ejemplo codifica cada CU del mismo modo que el dispositivo de codificación de video convencional después de que el codificador entrópico 103 someta a codificación entrópica `ctu_pcm_flag=0`.

60 En base a la posición (x0, y0) y al tamaño (log2CbSize) de una CU a procesar, al tamaño (la anchura `pic_width_in_luma_samples` y la altura `pic_height_in_luma_samples`) de un cuadro de imagen y al tamaño máximo (Log2MaxIpcmCbSizeY, donde Log2MaxIpcmCbSizeY es mayor o igual que el tamaño de CU mínimo Log2MinCbSizeY) de la CU I_PCM, el multiplexor 1100 calcula en la etapa S103 `cu_split_flag` de la CU determinada para procesar por el decodificador. Cuando se satisfacen cualesquiera una o varias de las siguientes condiciones, el multiplexor 1100 ajusta a `cu_split_flag=1`. En otras palabras, la CU a procesar es dividida.

65 [Condiciones]

1. $x0 + (1 \ll \log2CbSize) > pic_width_in_luma_samples$
2. $y0 + (1 \ll \log2CbSize) > pic_height_in_luma_samples$
3. $\log2CbSize > Log2MaxIpcmCbSizeY$

5 Cuando no se cumple ninguna de las condiciones anteriores, el multiplexor 1100 establece $cu_split_flag=0$. En otras palabras, la CU a procesar no se divide.

El multiplexor 1100 determina en la etapa S104 si $cu_split_flag=0$. Cuando $cu_split_flag=0$, el procedimiento avanza a la etapa S105. De lo contrario, la CU se divide y el procedimiento vuelve a la etapa S103.

10 En la etapa S105, el multiplexor 1100 someta a codificación no entrópica $pcm_alignment_zero_bit$ de CU(n).

En la etapa S106, el codificador PCM 107 someta a codificación no entrópica datos PCM sobre CU(n).

15 En la etapa S107, el multiplexor 1100 determina si todas las CU en una CTU a codificar están codificadas. Cuando todas las CU en la CTU a codificar están codificadas, el procedimiento avanza a la etapa S108. De lo contrario, el procedimiento vuelve a la etapa S103 para codificar las CU a procesar a continuación.

20 En la etapa S108, el codificador entrópico 103 resetea del motor de codificación. A continuación, el procedimiento avanza al proceso de codificación de la siguiente CTU.

El multiplexor 1100 entrega un flujo de bits obtenido multiplexando datos codificados por el codificador PCM 107 y el codificador entrópico 103.

25 El funcionamiento del dispositivo de codificación de video de la realización a modo de ejemplo es tal como se ha descrito anteriormente. Se debe observar que, aunque el multiplexor 1100 multiplexa ctu_pcm_flag , que indica si todos los bloques de imagen pertenecientes a una unidad de árbol de codificación que corresponde a un conjunto de bloques de imagen seleccionados por el selector de datos multiplexados 109 son bloques PCM, en la parte superior de la unidad de árbol de codificación, el multiplexor 1100 no multiplexa por lo menos una sintaxis cu_split_flag indicativa de la estructura de división de la unidad de árbol de codificación cuando todos los bloques de imagen pertenecientes a la unidad de árbol de codificación son bloques PCM.

35 De acuerdo con el funcionamiento descrito anteriormente del dispositivo de codificación de video de la realización a modo de ejemplo, cuando todas las CU en una CTU a codificar son I_PCM, la cabecera de modo I_PCM para codificación entrópica no se introduce entre los datos PCM en las sucesivas CU I_PCM. En otras palabras, el flujo de bits CTU, mostrado en (A) de la figura 2, se compone de: $ctu_pcm_flag=1$, $pcm_alignment_zero_bit$ y $pcm_sample_luma/pcm_sample_chroma$ de CU(0); $pcm_sample_luma/pcm_sample_chroma$ de CU(1); $pcm_sample_luma/pcm_sample_chroma$ de CU(2) y $pcm_sample_luma/pcm_sample_chroma$ de CU(3). El número de bits de una pieza de datos PCM es forzosamente un múltiplo de 8 bits. Por lo tanto, después de que se transmita en primer lugar una CU I_PCM, la posición de inicio de la subsiguiente CU I_PCM pasa a ser forzosamente un múltiplo de 8 bits. En otras palabras, la posición de inicio de la subsiguiente CU I_PCM pasa a ser forzosamente una dirección alineada en octetos. Por lo tanto, no se transmite $pcm_alignment_zero_bit$ en CU(1), CU(2) y CU(3).

45 Análogamente, el flujo de bits de CTU representado en (B) de la figura 2 se compone de: $ctu_pcm_flag=1$; $pcm_alignment_zero_bit$ y $pcm_sample_luma/pcm_sample_chroma$ de CU(0); $pcm_sample_luma/pcm_sample_chroma$ de CU(1); $pcm_sample_luma/pcm_sample_chroma$ de CU(2); $pcm_sample_luma/pcm_sample_chroma$ de CU(3); $pcm_sample_luma/pcm_sample_chroma$ de CU(4); y $pcm_sample_luma/pcm_sample_chroma$ de CU(5). En otras palabras, cuando todas las CU en una CTU a codificar son I_PCM, la cabecera de modo I_PCM para codificación entrópica no se introduce entre los datos PCM en sucesivas CU I_PCM.

50 Por lo tanto, cuando los datos PCM en la CU I_PCM subsiguiente a la primera CU I_PCM son multiplexados en un flujo de bits, no es necesario esperar hasta la finalización del proceso de entregar símbolos no entregados del motor de codificación y del proceso de resetear el motor de codificación, mencionados anteriormente. Por lo tanto, el dispositivo de codificación de video de la realización a modo de ejemplo puede multiplexar datos PCM en sucesivas CU I_PCM en un flujo de bits de manera eficiente. En otras palabras, los datos PCM en las sucesivas CU I_PCM se pueden transmitir de manera eficiente.

60 Realización 2 a modo de ejemplo

La figura 4 es un diagrama de bloques que representa un dispositivo de descodificación de video que corresponde al dispositivo de codificación de video de la primera realización a modo de ejemplo. El dispositivo de descodificación de video de la realización a modo de ejemplo incluye un desmultiplexor 201, un controlador de descodificación 202, un descodificador PCM 203, un descodificador entrópico 204, un transformador inverso/cuantificador inverso 206, un predictor 207, una memoria tampón 208, y un conmutador 221 y un conmutador 222.

El desmultiplexor 201 incluye un determinador PCM 2011 que determina si todas las CU en una CTU a descodificar son I_PCM. En la figura 4 se representa un ejemplo en el que el determinador PCM 2011 está presente en el desmultiplexor 201, pero esto no es indispensable. El determinador PCM 2011 puede estar dispuesto en una parte diferente al desmultiplexor 201, o puede estar dispuesto independientemente de cada unidad representada en la figura 4.

El desmultiplexor 201 desmultiplexa un flujo de bits de entrada y extrae un flujo de bits de video. **YY2**

Cuando `ctu_pcm_flag` obtenido por descodificación entrópica del flujo de bits de video de una CTU a descodificar el 1 (es decir, cuando el determinador PCM 2011 determina que todas las CU en la CTU a descodificar son I_PCM), el controlador de descodificación 202 calcula la `cu_split_flag` de una CU a procesar en la CTU a descodificar. Específicamente, en base a la posición (`x0`, `y0`) y al tamaño (`log2CbSize`) de la CU a procesar, al tamaño (la anchura `pic_width_in_luma_samples` y la altura `pic_height_in_luma_samples`) de un cuadro de imagen, y al tamaño máximo (`Log2MaxIpcmCbSizeY`, donde `Log2MaxIpcmCbSizeY` es mayor o igual que el tamaño de CU mínimo `Log2MinCbSizeY`) de la CU I_PCM, cuando cualesquiera una o varias de las siguientes condiciones se cumplen, se establece `cu_split_flag=1`. En otras palabras, la CU a procesar se divide.

[Condiciones]

1. $x0 + (1 \ll \log2CbSize) > pic_width_in_luma_samples$
2. $y0 + (1 \ll \log2CbSize) > pic_height_in_luma_samples$
3. $\log2CbSize > Log2MaxIpcmCbSizeY$

Cuando no se cumple ninguna de las condiciones anteriores, el controlador de descodificación 202 establece `cu_split_flag=0`. En otras palabras, la CU a procesar no se divide.

Además, cuando `cu_split_flag=0`, el controlador de descodificación 202 lee, a partir del flujo de bits de video, sintaxis `pcm_alignment_zero_bit` de una CU I_PCM a descodificar, y suministra los subsiguientes datos PCM al descodificador PCM 203 sin hacer que el descodificador entrópico 204 realice la descodificación entrópica de la CU/cabecera PU de una CU a descodificar actualmente. El descodificador PCM 203 lee los datos PCM suministrados por medio del conmutador 221 y descodifica PCM los datos PCM. El controlador de descodificación 202 cambia el conmutador 222 para suministrar una imagen reconstruida suministrada desde el descodificador PCM 203 a la memoria tampón 208. En el caso de `cu_split_flag=1`, el controlador de descodificación 202 avanza al proceso de descodificar la siguiente CU a procesar en la CTU a descodificar.

Cuando `ctu_pcm_flag` obtenido mediante descodificación entrópica del flujo de bits de video de la CTU a descodificar es 0, el controlador de descodificación 202 hace que el descodificador entrópico 204 someta a descodificación entrópica la CU/cabecera PU de una CU a descodificar actualmente. El descodificador entrópico 204 somete a descodificación entrópica `pcm_flag` que tiene el valor 1, el controlador de descodificación 202 lee, a partir del flujo de bits de video, sintaxis `pcm_alignment_zero_bit` de la CU I_PCM a descodificar, y suministra los datos PCM subsiguientes al descodificador PCM 203. El descodificador PCM 203 lee los datos PCM suministrados por medio del conmutador 221 y descodifica PCM los datos PCM. El controlador de descodificación 202 cambia el conmutador 222 para suministrar una imagen reconstruida suministrada desde el descodificador PCM 203 a la memoria tampón 208. Cuando el descodificador entrópico 204 somete a descodificación entrópica `pcm_flag` que tiene el valor 0 o cuando no existe ningún `pcm_flag`, el descodificador entrópico 204 realiza además la descodificación entrópica del parámetro de predicción y el valor de nivel de cuantificación de una CU a descodificar, y los suministra al transformador inverso/cuantificador inverso 206 y al predictor 207.

El transformador inverso/cuantificador inverso 206 realiza una cuantificación por frecuencia inversa del valor del nivel de cuantificación, y además realiza una transformada en frecuencia inversa del coeficiente de transformada de frecuencia obtenido por la cuantificación inversa.

Después de la transformada inversa, el predictor 207 genera una señal de predicción utilizando la imagen de una imagen reconstruida almacenada en la memoria tampón 208, en base al parámetro de predicción sometido a descodificación entrópica. Después de que se genere la señal de predicción, la señal de predicción suministrada desde el predictor 207 se añade a una imagen de error de predicción reconstruida, transformada inversa en frecuencia mediante el transformador inverso/cuantificador inverso 206, y la imagen de error de predicción reconstruida es suministrada al conmutador 222. Después de que se añada la señal de predicción, el controlador de descodificación 202 cambia el conmutador 222 para suministrar la imagen de error de predicción reconstruida, a la que se añade a la señal de predicción, a la memoria tampón 208 como una imagen reconstruida. A continuación, el procedimiento avanza al proceso de descodificar la siguiente CU.

El funcionamiento del dispositivo de descodificación de video de la realización a modo de ejemplo se describirá haciendo referencia al diagrama de flujo de la figura 5

En la etapa S201, el determinador PCM 2011 determina si todas las CU en una CTU a descodificar son I_PCM. Cuando todas las CU en la CTU a descodificar son I_PCM, el procedimiento avanza a la etapa S202. De lo contrario, aunque el proceso finaliza en el diagrama de flujo de la figura 5, el dispositivo de descodificación de video de la realización a modo de ejemplo descodifica cada CU del mismo modo que el dispositivo de codificación de video convencional.

En la etapa S202, el controlador de descodificación 202 calcula `cu_split_flag` de CU(n) a descodificar.

En la etapa S203, el controlador de descodificación 202 determina si el `cu_split_flag` calculado es 0. Cuando es 0, el procedimiento avanza a la etapa S204. De lo contrario (cuando `cu_split_flag` es 1), la CU a descodificar se divide, y el procedimiento vuelve a la etapa S202.

En la etapa S204, el controlador de descodificación 202 lee, desde el flujo de bits de video, la sintaxis `pcm_alignment_zero_bit` de CU(n) de I_PCM a descodificar. Se debe observar que el descodificador PCM 203 puede asimismo leer sintaxis `pcm_alignment_zero_bit` para codificación no entrópica.

En la etapa S205, el controlador de descodificación 202 lee, a partir del flujo de bits de video, datos PCM sobre CU(n) de I_PCM a descodificar. El descodificador PCM 203 somete a descodificación no entrópica datos PCM sobre CU(n). Se debe observar que el descodificador PCM 203 puede asimismo leer los datos PCM para descodificación no entrópica. A continuación, tal como se ha mencionado anteriormente, se determina una imagen reconstruida de CU(n) en base a los datos PCM.

En la etapa S206, el controlador de descodificación 202 determina si todas las CU en la CTU a descodificar están descodificadas. Cuando todas las CU están descodificadas, el procedimiento avanza a la etapa S207. De lo contrario, el procedimiento vuelve a la etapa S202 para descodificar la siguiente CU.

En la etapa S207, el descodificador entrópico 204 resetea un motor de descodificación, y a continuación finaliza el proceso de descodificación de la CTU a descodificar.

El funcionamiento del dispositivo de descodificación de video de la realización a modo de ejemplo es tal como se ha descrito anteriormente.

De acuerdo con el funcionamiento del dispositivo de descodificación de video de la realización a modo de ejemplo descrita anteriormente, no es necesario esperar a la finalización del proceso de resetear el motor de descodificación para leer, desde el flujo de bits, datos PCM sobre una CU I_PCM subsiguiente a la primera CU I_PCM. Por lo tanto, el dispositivo de descodificación de video de la realización a modo de ejemplo puede leer datos PCM sobre CU I_PCM sucesivas a partir del flujo de bits, de manera eficiente. En otras palabras, los datos PCM en las sucesivas CU I_PCM se pueden recibir de manera eficiente.

La sintaxis del árbol de codificación, la sintaxis del árbol cuaternario PCM y la sintaxis de muestra PCM correspondientes a cada una de las realizaciones a modo de ejemplo mencionadas anteriormente se representan en la figura 6, la figura 7 y la figura 8 respectivamente. Haciendo referencia a la figura 6, a la figura 7 y a la figura 8, en la figura 6 se encuentra un estado de transmisión de sintaxis `ctu_pcm_flag`. En la figura 7 se encuentra un estado de determinación de sintaxis `cu_split_flag`, indicativa de la estructura de división de la unidad de árbol de codificación, en base a la posición de un bloque de imagen de la unidad de árbol de codificación, al tamaño de un cuadro de imagen a descodificar y al tamaño máximo de bloque PCM ($\text{Log2MaxIpcmCbSizeY}$). En la figura 8 se encuentra un estado de codificación de `pcm_alignment_zero_bit` y datos PCM (`pcm_sample_luma/ pcm_sample_chroma`). En otras palabras, se encuentra que el proceso de resetear el motor de descodificación no es necesario para leer datos PCM en CU I_PCM sucesivas a partir de un flujo de bits.

En cada una de las realizaciones a modo de ejemplo mencionadas anteriormente, se supone que todas las CU en una CTU son I_PCM, pero la presente invención puede asimismo aplicarse a un caso en que se saltan todas las CU en una CTU. Por ejemplo, el determinador de todas-PCM que determina si todas las CU en una CTU son I_PCM tiene solamente que ser sustituido por un determinador de saltar-todas, que determina si se saltan todas las CU en una CTU de tal modo que, cuando se saltan todas las CU B, la sintaxis `ctu_skip` será codificada/descodificada igual que la sintaxis `ctu_pcm`. Además, `merge_idx` descrito en la sección 7.3.9.6 Sintaxis de unidad de predicción, de NPL 1, tiene que ser codificado/descodificado solamente a continuación del indicador `ctu_skip`.

Cada una de las realizaciones a modo de ejemplo mencionadas anteriormente se puede configurar en hardware, pero es posible asimismo implementar la realización a modo de ejemplo mediante un programa informático.

Un sistema de procesamiento de información representado en la figura 9 incluye un procesador 1001, una memoria de programa 1002, un medio de almacenamiento 1003 para almacenar datos de video y un medio de almacenamiento 1004 para almacenar un flujo de bits. El medio de almacenamiento 1003 y el medio de almacenamiento 1004 pueden ser medios de almacenamiento independientes, o áreas de almacenamiento en el

mismo medio de almacenamiento. Un medio de almacenamiento magnético, tal como un disco duro, se puede utilizar como un medio de almacenamiento.

5 En el sistema de procesamiento de información representado en la figura 9, un programa para realizar la función de cada bloque representado respectivamente en la figura 1 y en la figura 4 está almacenado en la memoria de programa 1002. A continuación, el procesador 1001 ejecuta procesos de acuerdo con el programa almacenado en la memoria de programa 1002 para realizar las funciones del dispositivo de codificación de video, o del dispositivo de descodificación de video, representados en la figura 1 o la figura 4, respectivamente.

10 Aunque la presente invención se ha descrito haciendo referencia a las realizaciones a modo de ejemplo y los ejemplos mencionados anteriormente, la presente invención no se limita a las realizaciones a modo de ejemplo y a los ejemplos mencionados anteriormente. Dentro del alcance de la presente invención se pueden realizar diversos cambios a las configuraciones y detalles de la presente invención, que pueden comprender los expertos en la materia.

15 Lista de signos de referencia

- 102 transformador/cuantificador
- 103 codificador entrópico
- 20 104 transformador inverso/cuantificador inverso
- 105 memoria tampón
- 106 predictor
- 107 codificador PCM
- 108 descodificador PCM
- 25 109 selector de datos multiplexados
- 110, 1100 multiplexor
- 1101 determinador PCM
- 121 conmutador
- 122 conmutador
- 30 201 desmultiplexor
- 2011 determinador PCM
- 202 controlador de descodificación
- 203 descodificador PCM
- 204 descodificador entrópico
- 35 206 transformador inverso/cuantificador inverso
- 207 predictor
- 208 memoria tampón
- 221 conmutador
- 222 conmutador
- 40 1001 procesador
- 1002 memoria de programa
- 1003 medio de almacenamiento
- 1004 medio de almacenamiento

REIVINDICACIONES

1. Un dispositivo de codificación de video, que comprende:

5 medios de transformada para transformar un bloque de imagen;
 medios de codificación entrópica para realizar la codificación entrópica de datos transformados, en el
 bloque de imagen transformado mediante los medios de transformada;
 medios de codificación PCM para codificar PCM el bloque de imagen;
 medios de selección de datos multiplexados para seleccionar, para cada bloque de imagen, datos de
 10 salida de los medios de codificación entrópica o bien datos de salida de los medios de codificación
 PCM; y
 medios de multiplexación para multiplexar sintaxis `ctu_pcm_flag` que indica si todos los bloques de
 imagen que pertenecen a una unidad de árbol de codificación como un conjunto de bloques de imagen
 15 seleccionados por los medios de selección de datos multiplexados son bloques PCM, en la parte
 superior de la unidad de árbol de codificación;
 en el que cuando todos los bloques de imagen que pertenecen a la unidad de árbol de codificación son
 bloques PCM, los medios de multiplexación no multiplexan por lo menos una sintaxis `cu_split_flag`
 indicativa de una estructura de división de la unidad de árbol de codificación, y cuando no todos los
 20 bloques de imagen que pertenecen a la unidad de árbol de codificación son bloques PCM, los medios
 de multiplexación determinan la sintaxis `cu_split_flag` de acuerdo con la posición y el tamaño del bloque
 de imagen en la unidad de árbol de codificación, el tamaño de un cuadro de imagen a descodificar y el
 tamaño máximo de bloque PCM y multiplexan la sintaxis `cu_split_flag`.

2. Un dispositivo de descodificación de video, que comprende:

25 medios de desmultiplexación para desmultiplexar un flujo de bits que incluye una sintaxis `ctu_pcm_flag`
 que indica si todos los bloques de imagen que pertenecen a una unidad de árbol de codificación a
 descodificar son bloques PCM;
 medios de descodificación PCM para descodificar PCM datos PCM en un bloque de imagen incluido en
 30 el flujo de bits;
 medios de descodificación entrópica para realizar la descodificación entrópica de datos transformados,
 en el bloque de imagen incluido en el flujo de bits; y
 medios de control de descodificación que, cuando un valor de sintaxis `ctu_pcm_flag` de la unidad de
 árbol de codificación a descodificar indica que todos los bloques de imagen que pertenecen a la unidad
 35 de árbol de codificación son bloques PCM, determinar la sintaxis `cu_split_flag` indicativa de una
 estructura de división de la unidad de árbol de codificación de acuerdo con el resultado de un cálculo en
 base a la posición y al tamaño del bloque de imagen en la unidad de árbol de codificación, al tamaño de
 un cuadro de imagen a descodificar y al tamaño máximo de bloque PCM.

3. Un procedimiento de codificación de video, que comprende:

transformar un bloque de imagen;
 someter a codificación entrópica datos transformados, en el bloque de imagen transformado;
 codificar PCM el bloque de imagen;
 45 seleccionar, para cada bloque de imagen, datos de codificación entrópica o bien datos codificados PCM;
 y
 multiplexar sintaxis `ctu_pcm_flag` que indica si todos los bloques de imagen que pertenecen a una
 unidad de árbol de codificación como un conjunto de bloques de imagen son bloques PCM, en la parte
 superior de la unidad de árbol de codificación,
 50 en el que cuando todos los bloques de imagen que pertenecen a la unidad de árbol de codificación son
 bloques PCM, no se multiplexa por lo menos una sintaxis `cu_split_flag` indicativa de una estructura de
 división de la unidad de árbol de codificación, y cuando no todos los bloques de imagen que pertenecen
 a la unidad de árbol de codificación son bloques PCM, se determina la sintaxis `cu_split_flag` de acuerdo
 con la posición y el tamaño del bloque de imagen en la unidad de árbol de codificación, el tamaño de un
 55 cuadro de imagen a descodificar y el tamaño máximo de bloque PCM y se multiplexa la sintaxis
`cu_split_flag`.

4. Un procedimiento de descodificación de video, que comprende:

60 desmultiplexar un flujo de bits que incluye una sintaxis `ctu_pcm_flag` que indica si todos los bloques de
 imagen que pertenecen a una unidad de árbol de codificación a descodificar son bloques PCM;
 descodificar PCM datos PCM, en un bloque de imagen incluido en el flujo de bits;
 realizar la descodificación entrópica de datos transformados, en el bloque de imagen incluido en el flujo
 de bits; y
 65 cuando un valor de la sintaxis `ctu_pcm_flag` de la unidad de árbol de codificación a descodificar indica
 que todos los bloques de imagen que pertenecen a la unidad de árbol de codificación son bloques PCM,

determinar una sintaxis `cu_split_flag` indicativa de una estructura de división de la unidad de árbol de codificación de acuerdo con el resultado de un cálculo en base a la posición y al tamaño del bloque de imagen en la unidad de árbol de codificación, al tamaño de un cuadro de imagen a descodificar y al tamaño máximo de bloque PCM.

- 5
5. Un programa de codificación de video para hacer que un ordenador ejecute:
- 10 un proceso de transformación de un bloque de imagen;
un proceso de codificación entrópica de datos transformados, en el bloque de imagen transformado;
un proceso de codificación PCM del bloque de imagen;
un proceso de seleccionar, para cada bloque de imagen, datos de codificación entrópica o bien datos codificados PCM; y
un proceso de multiplexar sintaxis `ctu_pcm_flag` que indica si todos los bloques de imagen que pertenecen a una unidad de árbol de codificación como un conjunto de bloques de imagen son bloques PCM, en la parte superior de la unidad de árbol de codificación,
15 en el que cuando todos los bloques de imagen que pertenecen a una unidad de árbol de codificación son bloques PCM, se hace que el ordenador no multiplexe por lo menos una sintaxis `cu_split_flag` indicativa de una estructura de división de la unidad de árbol de codificación, y cuando no todos los bloques de imagen que pertenecen a la unidad de árbol de codificación son bloques PCM, la sintaxis `cu_split_flag` se determina de acuerdo con la posición y el tamaño del bloque de imagen en la unidad de árbol de codificación, el tamaño de un cuadro de imagen a descodificar y un tamaño máximo de bloque PCM, y se multiplexa la sintaxis `cu_split_flag`.
- 20

- 25 6. Un programa de descodificación de video para hacer que un ordenador ejecute:
- un proceso de desmultiplexación de un flujo de bits que incluye una sintaxis `ctu_pcm_flag` que indica si todos los bloques de imagen que pertenecen a una unidad de árbol de codificación a descodificar son bloques PCM;
un proceso de descodificación PCM de datos PCM, en un bloque de imagen incluido en el flujo de bits;
30 un proceso de descodificación entrópica de datos transformados, en el bloque de imagen incluido en el flujo de bits; y
cuando un valor de la sintaxis `ctu_pcm_flag` de la unidad de árbol de codificación a descodificar indica que todos los bloques de imagen que pertenecen a la unidad de árbol de codificación son bloques PCM,
un proceso de determinar una sintaxis `cu_split_flag` indicativa de una estructura de división de la unidad de árbol de codificación de acuerdo con el resultado de un cálculo en base a la posición y al tamaño del
35 bloque de imagen en la unidad de árbol de codificación, al tamaño de un cuadro de imagen a descodificar y al tamaño máximo de bloque PCM.

40

FIG. 1

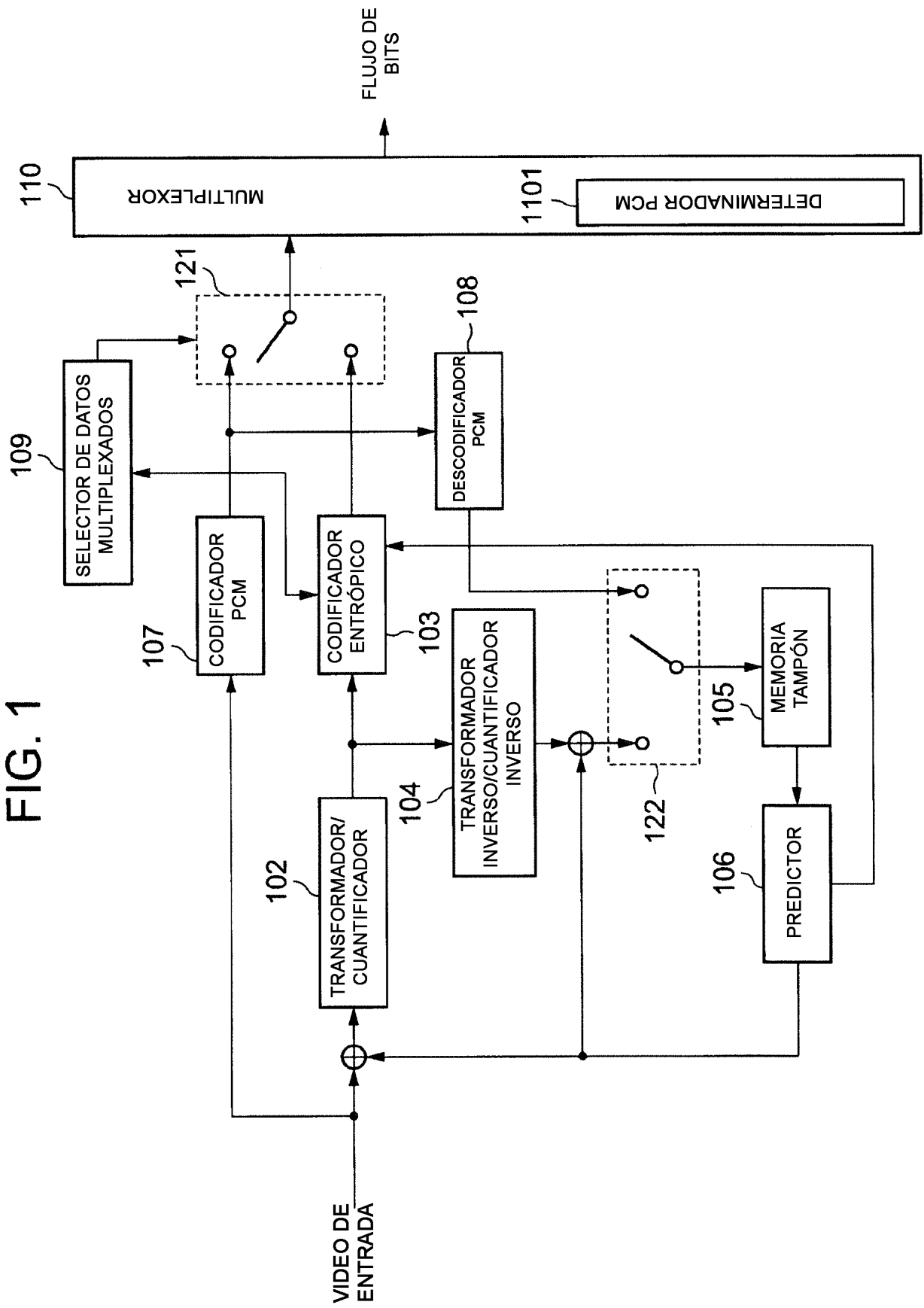
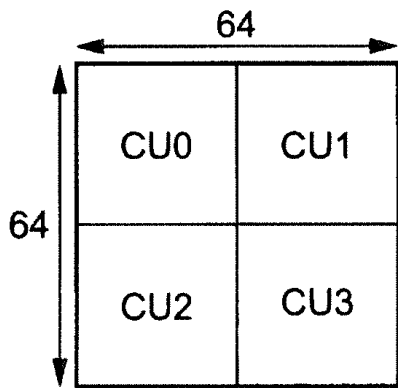
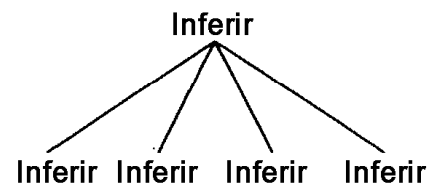


FIG. 2

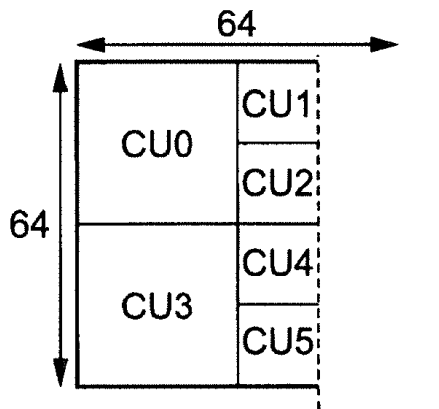


cu_split_flag
@CUDepth=0

cu_split_flag
@CUDepth=1



(A) CTU COMPLETA



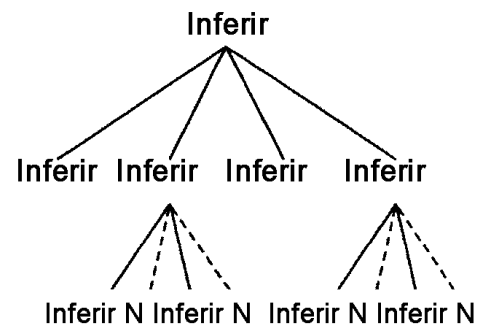
LÍMITE DE LA
IMAGEN
(BORDE DERECHO)

cu_split_flag
@CUDepth=0

cu_split_flag
@CUDepth=1

cu_split_flag
@CUDepth=2

cu_split_flag
@CUDepth=3



(B) CTU INCOMPLETA

FIG. 3

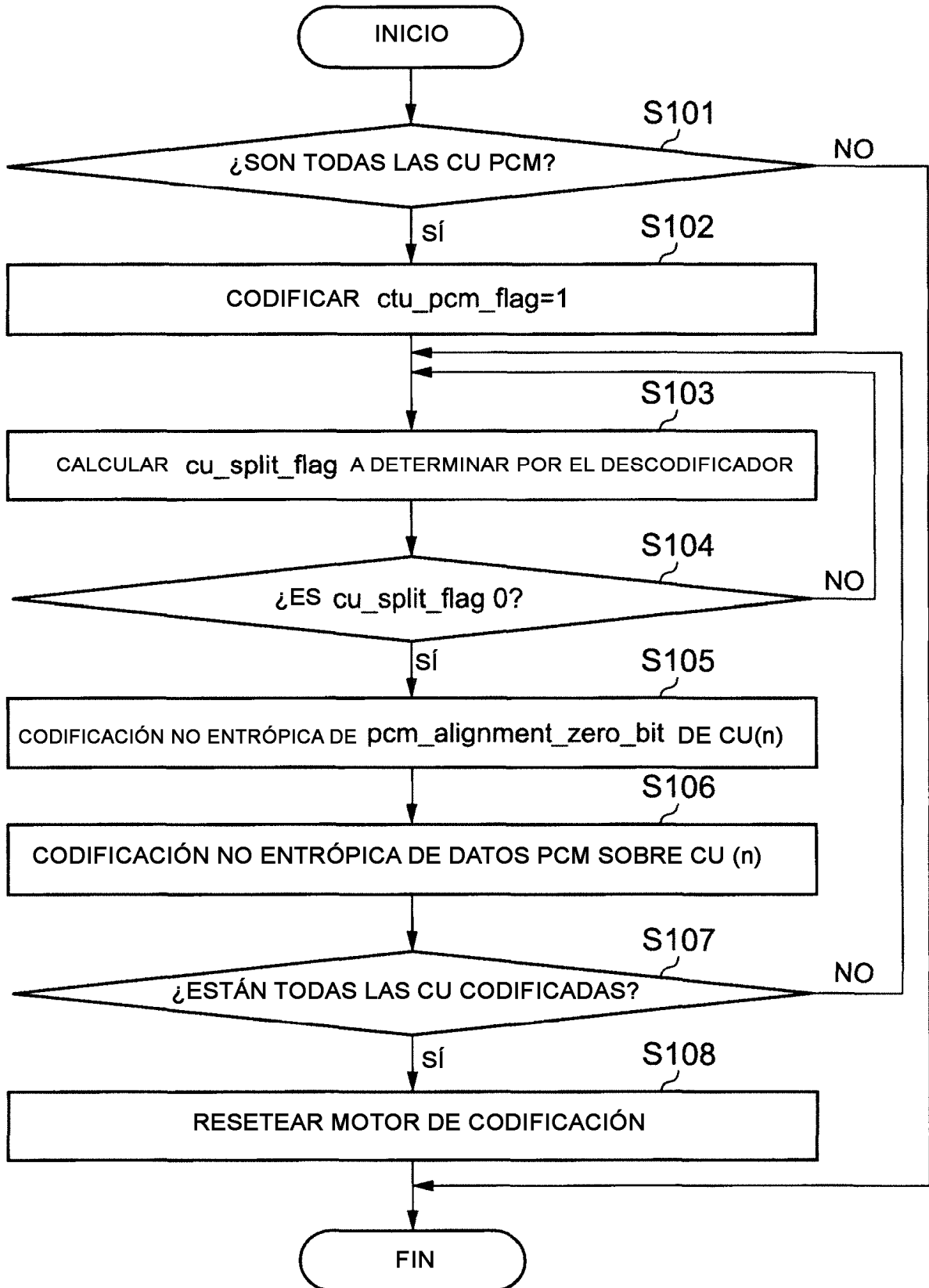


FIG. 4

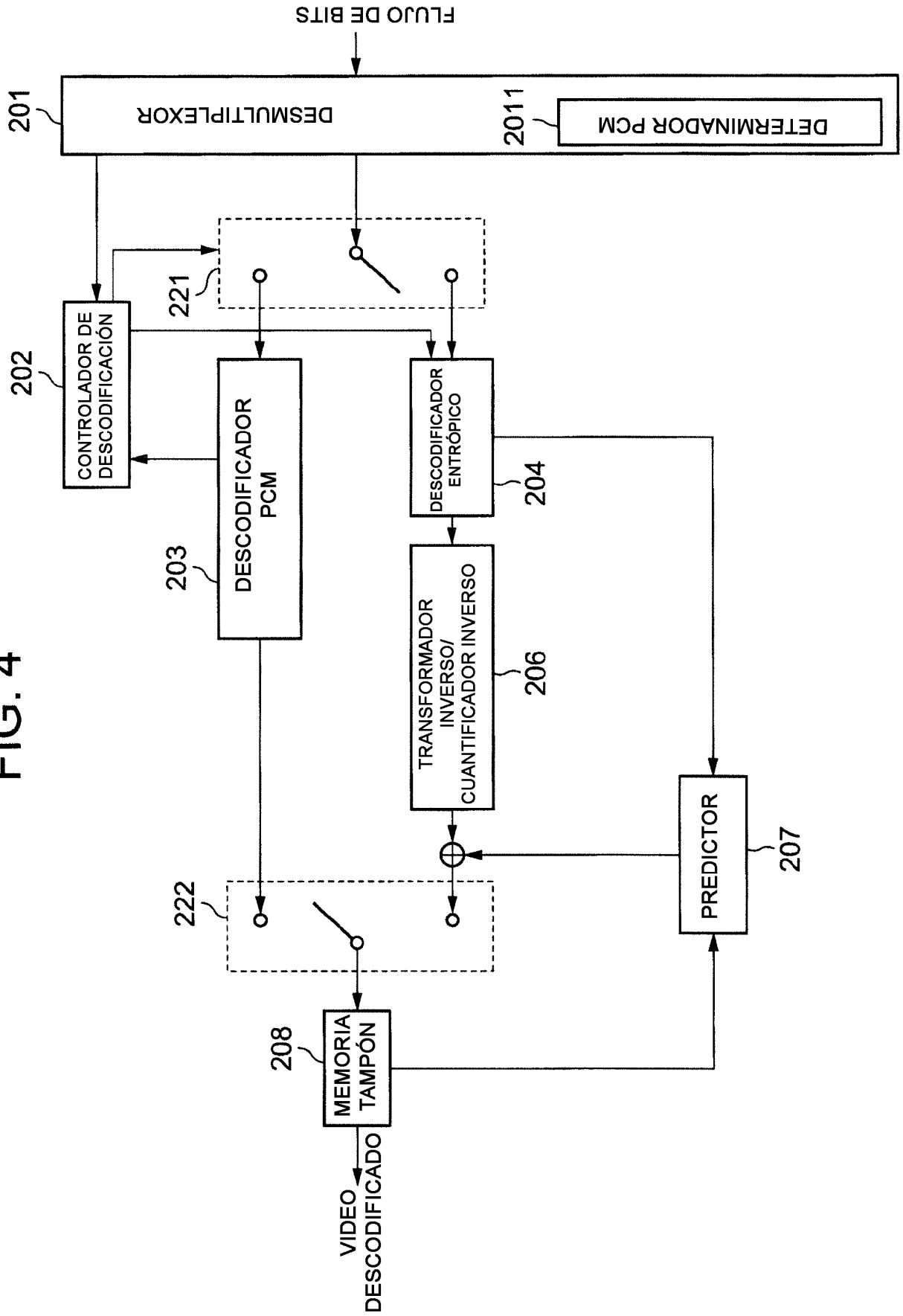


FIG. 5

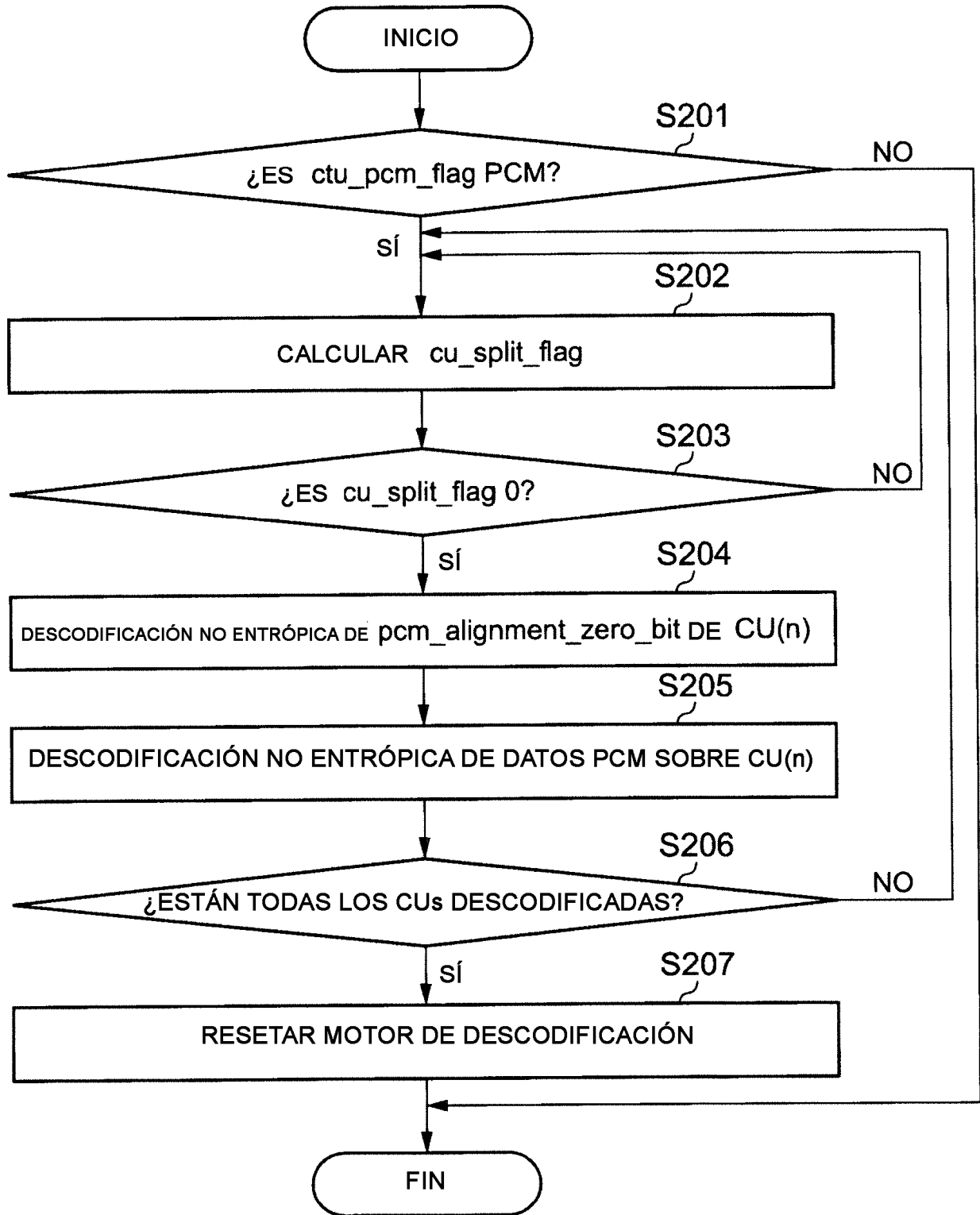


FIG. 6

(Sintaxis de unidad de árbol de codificación)

coding_tree_unit() {	Descriptor
xCtb=(CtbAddrRS % PicWidthInCtbsY) << Log2CtbSizeY	
yCtb=(CtbAddrRS / PicWidthInCtbsY) << Log2CtbSizeY	
CtbAddrInSliceSeg = CtbAddrInRS - slice_segment_address	
if(slice_sao_luma_flag slice_sao_chroma_flag)	
sao(xCtb >> Log2CtbSizeY, yCtb >> Log2CtSizeY)	
if(ctu_pcm_enabled_flag)	
ctu_pcm_flag	ae(v)
if(!ctu_pcm_flag)	
coding_quadtree(xCtb, yCtb, Log2CtbSizeY, 0)	
else	
pcm_quadtree(xCtb, yCtb, Log2CtbSizeY, 0)	
}	

FIG. 7

(Sintaxis de árbol cuaternario PCM)

pcm_quadtree(x0, y0, log2CbSize, ctDepth) {	Descriptor
if(x0 + (1 << log2CbSize) > pic_width_in_luma_samples y0 + (1 << log2CbSize) > pic_height_in_luma_samples log2CbSize > Log2MaxIpcmCbSizeY)	
cu_split_flag[x0][y0] = 1	
else	
cu_split_flag[x0][y0] = 0	
if(cu_split_flag[x0][y0]) {	
x1 = x0 + ((1 << log2CbSize) >> 1)	
y1 = y0 + ((1 << log2CbSize) >> 1)	
pcm_quadtree(x0, y0, log2CbSize - 1, ctDepth + 1)	
if(x1 < pic_width_in_luma_samples)	
pcm_quadtree(x1, y0, log2CbSize - 1, ctDepth + 1)	
if(y1 < pic_height_in_luma_samples)	
pcm_quadtree(x0, y1, log2CbSize - 1, ctDepth + 1)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
pcm_quadtree(x1, y1, log2CbSize - 1, ctDepth + 1)	
} else	
pcm_sample(x0, y0, log2CbSize)	
}	

FIG. 8

(Sintaxis de muestra PCM)

pcm_sample(x0, y0, log2CbSize) {	Descriptor
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
for(i =0; i < 1 << (log2CbSize << 1); i++)	
pcm_sample_luma[i]	u(v)
for(i =0; i < (1 << (log2CbSize << 1)) >> 1; i++)	
pcm_sample_chroma[i]	u(v)
}	

FIG. 9

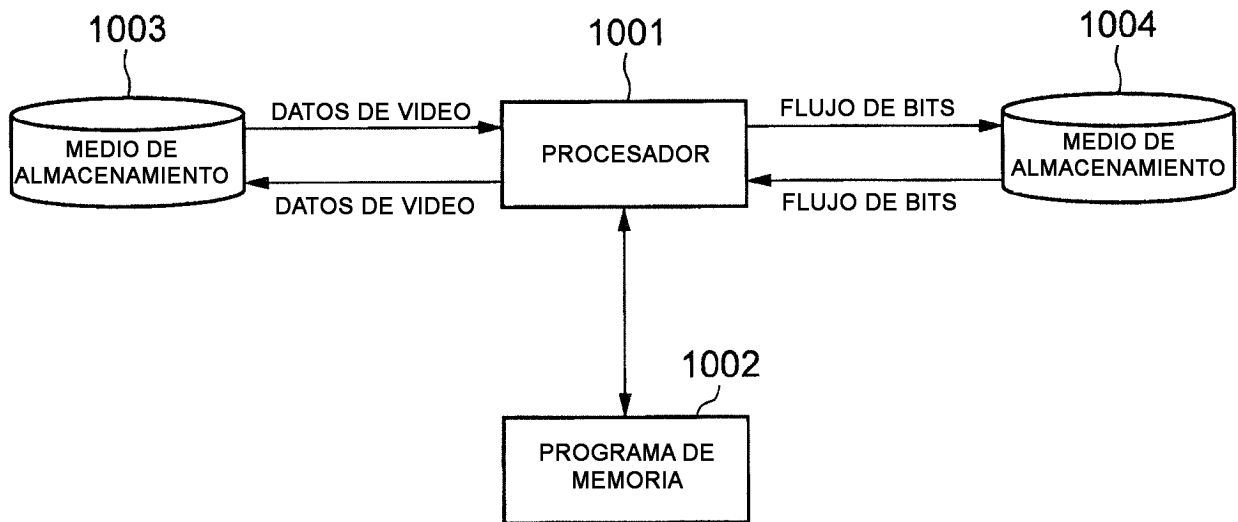


FIG. 10

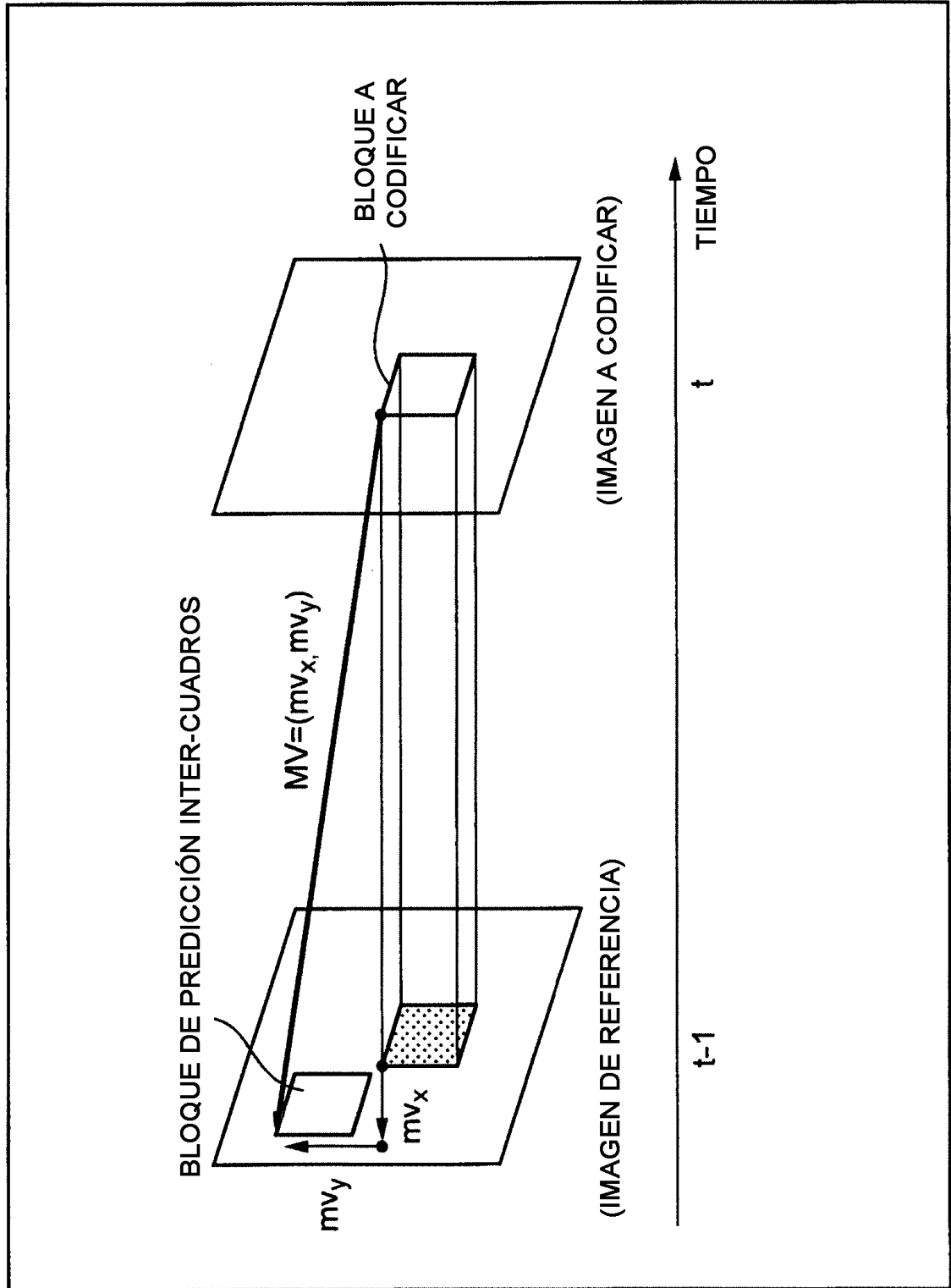


FIG. 11

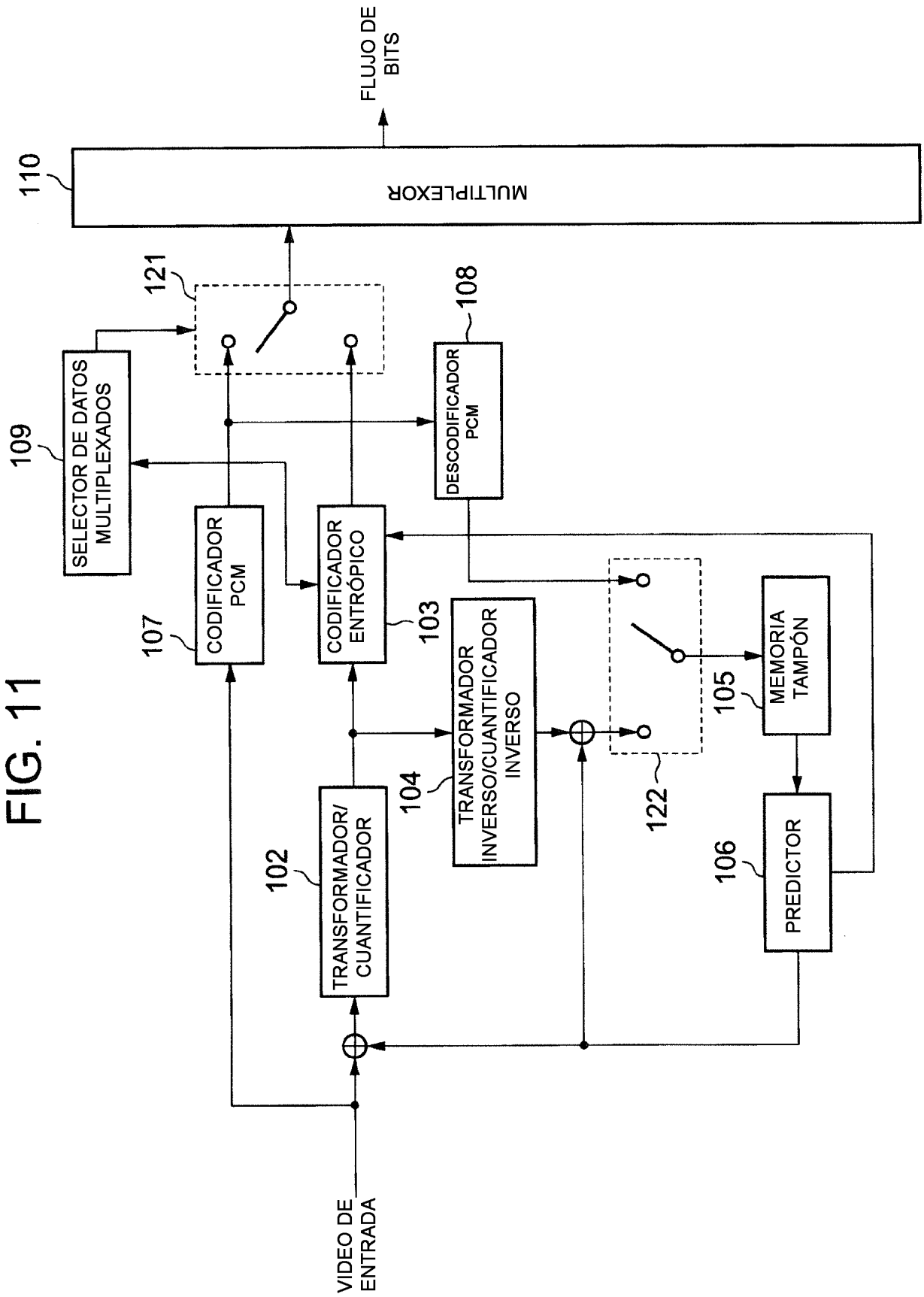


FIG. 12

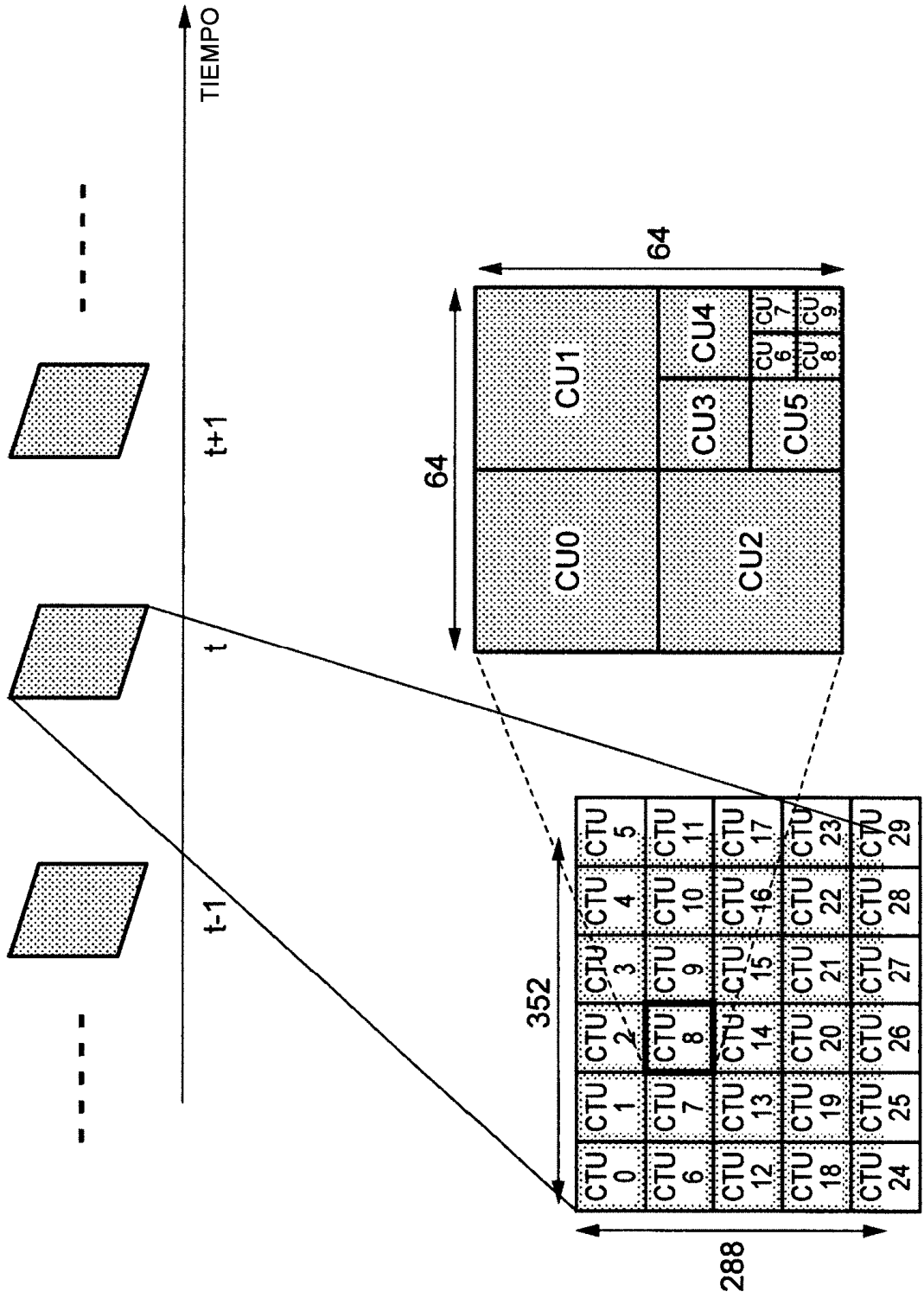
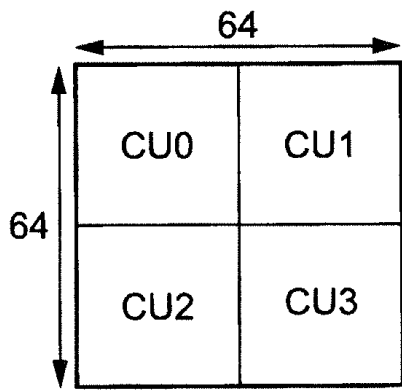
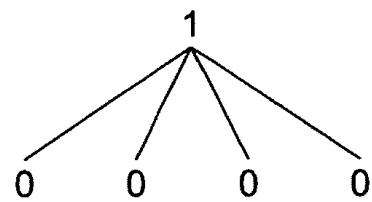


FIG. 13

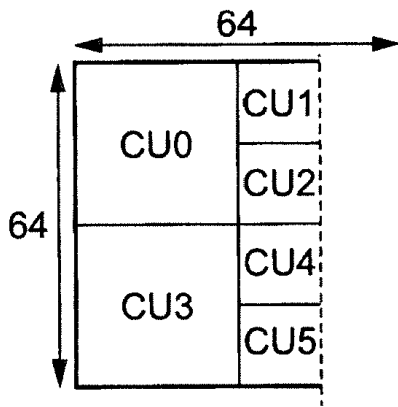


cu_split_flag
@CUDepth=0

cu_split_flag
@CUDepth=1



(A) CTU COMPLETA



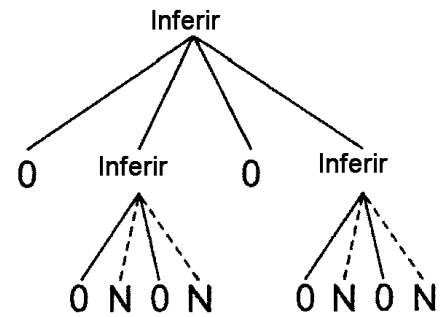
LÍMITE DE LA
IMAGEN
(BORDE DERECHO)

cu_split_flag
@CUDepth=0

cu_split_flag
@CUDepth=1

cu_split_flag
@CUDepth=2

cu_split_flag
@CUDepth=3



(B) CTU INCOMPLETA