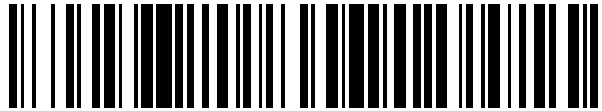


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 569 491**

51 Int. Cl.:

**H04N 19/70**

(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **09.02.2000 E 00902884 (6)**

97 Fecha y número de publicación de la concesión europea: **13.04.2016 EP 1069779**

54 Título: **Sistema de codificación y método asociado**

30 Prioridad:

**09.02.1999 JP 3194499**

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

**11.05.2016**

73 Titular/es:

**SONY CORPORATION (100.0%)  
6-7-35 KITASHINAGAWA SHINAGAWA-KU  
TOKYO 141-0001, JP**

72 Inventor/es:

**KITAMURA, TAKUYA**

74 Agente/Representante:

**LEHMANN NOVO, María Isabel**

**ES 2 569 491 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

Sistema de codificación y método asociado

## 5 CAMPO TÉCNICO

La presente invención se refiere a un sistema de codificación y un método asociado, adecuado para su uso en un transcodificador para realizar un proceso de recodificación en un flujo codificado que ha sido codificado en conformidad con la norma MPEG, para generar un flujo recodificado que tiene una diferente estructura GOP (Group of Pictures - Grupo de Imágenes) o tasa binaria.

## ANTECEDENTES DE LA INVENCION

15 En las estaciones emisoras para producir y difundir programas de televisión, la técnica de MPEG (Moving Picture Experts Group – Grupo de Expertos de Imágenes en Movimiento) se suele utilizar para comprimir/codificar datos de vídeo. En particular, en los campos de registro de datos de vídeo sobre un material de soporte de registro aleatoriamente accesible tal como una cinta y transmisión de datos de vídeo por intermedio de cable o satélite, la técnica MPEG se ha convertido en una norma *de facto*.

20 Un ejemplo de procesamiento realizado en una estación emisora hasta que un programa de vídeo allí creado se transmitido a cada domicilio se describirá de forma concisa. En primer lugar, un codificador provisto en una videocámara (*camcoder*) que comprende una cámara de vídeo y una VTR (Grabadora de Cinta de Vídeo) integradas juntas se utiliza para codificar y registrar datos de vídeo origen en cinta magnética. En este caso, el codificador de la videocámara codifica los datos de vídeo origen con el fin de que sean adecuados para un formato de registro para una cinta VTR. A modo de ejemplo, un flujo de bits MPEG registrado en la cinta magnética tiene una estructura GOP en donde se configura un grupo de imágenes GOP mediante dos tramas (a modo de ejemplo, I, B, I, B, I, B, ...). Además, los flujos binarios de MPEG registrados en la cinta magnética tienen una tasa binaria de 18 Mbps.

30 A continuación, una estación emisora principal realiza un proceso de edición para editar el flujo de bits de vídeo registrado en la cinta magnética. Para conseguirlo, la estructura GOP del flujo de bits de vídeo registrado en la cinta magnética se convierte en una estructura adecuada para el proceso de edición. En la estructura GOP adecuada para el proceso de edición, un GOP se configura por una trama y todas las imágenes son del tipo I. Esto es así porque las imágenes I, que no tienen correlaciones con otras imágenes son más adecuadas para la edición en tramas. En una operación real, el flujo de vídeo registrado en la cinta magnética se decodifica en datos de vídeo de banda base. Esta señal de vídeo de banda base se recodifica de modo que todas las imágenes sean convertidas en el tipo I. Realizando el proceso de decodificación y de recodificación de esta manera, se puede generar un flujo de bits que tenga la estructura GOP adecuada para el proceso de edición.

40 A continuación, para transmitir el programa de vídeo editado generado por el proceso de edición anteriormente descrito, desde la estación emisora principal a las estaciones emisoras locales, el flujo de bits del programa de vídeo editado se convierte en una estructura GOP y un flujo de bits adecuado para un proceso de transmisión. A modo de ejemplo, en la estructura GOP adecuada para la transmisión entre estaciones emisoras, se configura un GOP mediante 15 tramas (a modo de ejemplo, I, B, B, P, B, B, P, ...). Además, la tasa binaria adecuada para transmisiones entre estaciones emisoras es preferiblemente alta, es decir, 50 Mbps o superior puesto que una línea exclusiva constituida por fibras ópticas y que tiene una alta capacidad de transmisión se suele proporcionar entre las estaciones emisoras. Más concretamente, el flujo de bits del programa de vídeo editado se decodifica de nuevo en los datos de vídeo de banda base. A continuación, los datos de vídeo de banda base se recodifican para tener la estructura GOP y la tasa binaria adecuada para transmisiones entre estaciones emisoras.

50 En una estación emisora local, un proceso de edición se realiza para insertar películas comerciales específicas para la localidad en el programa de vídeo transmitido desde la estación emisora principal. Es decir, como en el proceso de edición anteriormente descrito, el flujo de vídeo transmitido desde el flujo principal se decodifica de nuevo en los datos de vídeo de banda base. Al recodificar la señal de vídeo de banda base de modo que todas las imágenes se conviertan en el tipo I, se puede generar un flujo de bits que tenga la estructura GOP adecuada para el proceso de edición.

60 Posteriormente, para transmitir el programa de vídeo editado en la estación emisora local para cada unidad residencial por intermedio de cable o satélite, la conversión en la estructura GOP y tasa binaria adecuada para el proceso de transmisión se realiza en esta instancia operativa. A modo de ejemplo, en la estructura GOP adecuada para transmisiones a cada vivienda, se configura una GOP mediante 15 tramas (a modo de ejemplo, I, B, B, P, B, B, P, ...) y la tasa binaria adecuada para transmisiones a cada vivienda es baja, es decir, aproximadamente 5 Mbps. Más concretamente, el flujo de bits del programa de vídeo editado se decodifica de nuevo en los datos de vídeo de banda base. A continuación, los datos de vídeo de banda base se recodifican en la estructura GOP y la tasa binaria adecuada para las transmisiones.

65 Queda entendido a partir de la descripción anterior que la pluralidad de procesos de decodificación y codificación se

repite mientras el programa de vídeo se está transmitiendo desde la estación emisora a cada vivienda. De hecho, la estación emisora requiere varios procesos de señalización distintos a los anteriormente descritos, de modo que los procesos de decodificación y codificación deben repetirse para cada proceso de señalización.

5 Como es bien conocido, sin embargo, los procesos de decodificación y de codificación basados en la norma MPEG no son 100 % reversibles. Es decir, los datos de vídeo de banda base antes de la codificación no son perfectamente los mismos que los datos de vídeo decodificados y tienen su calidad de imagen degradada debido a los procesos de decodificación y codificación. Lamentablemente, las repeticiones de los procesos de decodificación y codificación pueden degradar progresivamente la calidad de imagen, según se describió con anterioridad. Dicho de otro modo, la repetición de los procesos de decodificación/codificación puede acumular una degradación de la calidad de imagen.

10 La presente invención se da a conocer con miras a estas circunstancias, y es un objetivo de la presente invención dar a conocer un sistema de transcodificación que pueda evitar la degradación de la calidad de imagen aun cuando los procesos de decodificación y codificación se repitan para cambiar la estructura del GOP (Grupo de Imágenes) en el flujo de bits codificado sobre la base de la norma MPEG.

15 El documento EP 0 942 605 describe un dispositivo de la técnica actual que registra un flujo codificado origen. Este dispositivo tiene el medio de decodificación, el medio de codificación y el medio de recepción según la reivindicación 1.

20 La presente invención se refiere a un transcodificador para recodificar un flujo codificado generado sobre la base de la norma MPEG para generar un flujo recodificado que tenga un diferente GOP (Grupo de Imágenes) y una diferente tasa binaria.

25 Más concretamente, un dispositivo de decodificación de un transcodificador 106 decodifica un flujo codificado origen para generar datos de vídeo decodificados y extrae parámetros de codificación anteriores superpuestos en el flujo codificado bajo la forma de un flujo denominado `history_stream()`. En este caso, el dispositivo de decodificación extrae los parámetros de codificación anteriores sobre la base de la información superpuesta en el flujo codificado bajo la forma del flujo denominado `re_coding_stream_info()`.

30 Un dispositivo de codificación recibe los datos de vídeo decodificados y los parámetros de codificación anteriores y utiliza estos últimos para codificar los datos de vídeo decodificados de tal manera que se impida que el proceso de recodificación genere una degradación de la calidad de la imagen, con lo que se genera un flujo recodificado. Además, el dispositivo de codificación selecciona unos parámetros de entre los parámetros de codificación anteriores que sean óptimos para una aplicación localizada después del dispositivo de codificación y conectada a dicho dispositivo y realiza la escritura solamente de los parámetros de codificación anteriores seleccionados en el flujo codificado bajo la forma del flujo denominado `history_stream()`. El dispositivo de codificación superpone la información que indica los parámetros de codificación anteriores seleccionados. En el flujo codificado bajo la forma de `re_coding_stream_info()` de modo que la aplicación siguiente pueda extraer de forma adecuada a partir del flujo recodificado, los parámetros de codificación bajo la forma de `history_stream()`.

35 El transcodificador según las formas de realización de la presente invención puede realizar un sistema de codificación que pueda utilizar un número mínimo de parámetros de codificación adaptados para que la siguiente aplicación minimice la degradación de la calidad de imagen aun cuando se recodifiquen repetidamente datos de vídeo.

40 El transcodificador según las formas de realización de la presente invención comprende un medio de decodificación para decodificar un flujo codificado origen para generar datos de vídeo y para extraer, a partir de flujo codificado origen, parámetros de codificación anteriores generados por un proceso de codificación anterior, un medio de codificación para recodificar los datos de servicio para generar un flujo de vídeo recodificado y un medio de control para recibir los parámetros de codificación anteriores para controlar el proceso de recodificación por el medio de codificación sobre la base de los parámetros de codificación anteriores y escribir, de forma selectiva, los parámetros de codificación anteriores en un flujo recodificado.

45 El dispositivo de codificación del transcodificador según las formas de realización de la presente invención, selecciona unos parámetros de entre los parámetros de codificación anteriores que sean requeridos por la aplicación localizada después del medio de codificación y conectada a dicho medio y escriba los parámetros de codificación anteriores seleccionados en el flujo recodificado.

50 El dispositivo de codificación del transcodificador según las formas de realización de la presente invención, escribe, de forma selectiva, los parámetros de codificación anteriores en el flujo recodificado y escribe en el flujo recodificado un flag y/o indicador que indica un conjunto de datos de los parámetros de codificación anteriores escritos en el flujo recodificado.

55 El dispositivo de codificación del transcodificador según las formas de realización de la presente invención escribe información sobre los parámetros de codificación anteriores en el flujo codificado bajo la forma de `history_stream()` y

escribe información sobre el flujo recodificado en el flujo recodificado bajo la forma de `re_coding_stream_info()`.

5 El dispositivo de codificación del transcodificador según las formas de realización de la presente invención escribe, de forma selectiva, los parámetros de codificación anteriores en el flujo recodificado bajo la forma de `history_stream()` y escribe en el flujo recodificado bajo la forma de `re_coding_stream_info()`, información sobre un conjunto de datos de los parámetros de codificación anteriores escritos en el flujo recodificado.

10 El dispositivo de codificación según las formas de realización de la presente invención recibe parámetros de codificación anteriores sobre un proceso de codificación anterior realizado sobre los datos de vídeo, escribe, de forma selectiva, los parámetros de codificación anteriores en un flujo recodificado y escribe en el flujo recodificado, información que indica un conjunto de datos de los parámetros de codificación anteriores escritos en el flujo recodificado.

15 El dispositivo de decodificación según las formas de realización de la presente invención extrae, desde el flujo codificado, la información sobre el conjunto de datos de los parámetros de codificación anteriores superpuestos en el flujo codificado y extrae los parámetros de codificación anteriores desde el flujo codificado sobre la base de la información sobre el conjunto de datos.

20 El dispositivo de decodificación, según las formas de realización de la presente invención, extrae desde el flujo codificado el señalizador y/o indicador escrito en el flujo codificado bajo la forma `re_coding_stream_info()` y extrae los parámetros de codificación anteriores desde el flujo codificado sobre la base del señalizador y/o indicador.

#### BREVE DESCRIPCIÓN DE LOS DIBUJOS

25 La Figura 1 es un diagrama para describir el principio de una codificación muy eficiente.

La Figura 2 es un diagrama para describir cómo se procesan tipos de imagen en la compresión de datos de imágenes.

30 La Figura 3 es un diagrama para describir cómo los tipos de imágenes se procesan en la compresión de datos de imágenes.

La Figura 4 es un diagrama para describir el principio de la codificación de señales de imágenes en movimiento.

35 La Figura 5 es un diagrama de bloques que ilustra las configuraciones de dispositivos para codificar y decodificar señales de imágenes en movimiento.

La Figura 6 es un diagrama para describir la configuración de datos de imágenes.

40 La Figura 7 es un diagrama de bloques que ilustra la configuración de un codificador 18 en la Figura 5.

La Figura 8 es un diagrama para describir la operación de un circuito de conmutación de modos de predicción 52 en la Figura 7.

45 La Figura 9 es un diagrama para describir la operación del circuito de conmutación de modos de predicción 52 en la Figura 7.

50 La Figura 10 es un diagrama para describir la operación del circuito de conmutación de modos de predicción 52 en la Figura 7.

La Figura 11 es un diagrama para describir la operación del circuito de conmutación de modos de predicción 52 en la Figura 7.

55 La Figura 12 es un diagrama de bloques que ilustra la configuración de un decodificador 31 en la Figura 5.

La Figura 13 es un diagrama para describir el control SNR correspondiente al tipo de imagen.

60 La Figura 14 es un diagrama de bloques que ilustra la configuración de un transcodificador 101 al que se aplica la presente invención.

La Figura 15 es un diagrama de bloques que ilustra la configuración del transcodificador 101 en la Figura 14 en detalle.

65 La Figura 16 es un diagrama de bloques que ilustra la configuración de un dispositivo de decodificación 102 proporcionado en un aparato de decodificación 102 en la Figura 14.

La Figura 17 es un diagrama para describir los elementos de imagen pixels en un macrobloque.

La Figura 18 es un diagrama para describir una zona en la que se registran parámetros de codificación.

5 La Figura 19 es un diagrama de bloques que ilustra la configuración de un dispositivo de codificación 106 proporcionado en un dispositivo de codificación 106 en la Figura 14.

La Figura 20 es un diagrama de bloques que ilustra una realización ejemplo de la configuración del registro histórico VLC211 en la Figura 15.

10 La Figura 21 es un diagrama de bloques que ilustra una realización ejemplo de la configuración del registro histórico VLD203 en la Figura 15.

15 La Figura 22 es un diagrama de bloques que ilustra una realización ejemplo de la configuración de un convertidor 212 representado en la Figura 15.

La Figura 23 es un diagrama de bloques que ilustra una realización ejemplo de la configuración de un circuito de personal 323 representado en la Figura 22.

20 La Figura 24 es un cronograma para describir el funcionamiento del convertidor 212 representado en la Figura 22.

La Figura 25 es un diagrama de bloques que ilustra una realización ejemplo de la configuración de un convertidor 202 representado en la Figura 15.

25 La Figura 26 es un diagrama de bloques que ilustra una realización ejemplo de la configuración de un circuito de supresión 343 en la Figura 25.

La Figura 27 es un diagrama de bloques que ilustra otra realización ejemplo de la configuración del convertidor 212 en la Figura 15.

30 La Figura 28 es un diagrama de bloques que ilustra otra realización ejemplo de la configuración del convertidor 202 representado en la Figura 15.

35 La Figura 29 es un diagrama de bloques que ilustra una realización ejemplo de la configuración de un dispositivo formateador de datos de usuario 213 representado en la Figura 15.

La Figura 30 es un diagrama que ilustra cómo se utiliza realmente el transcodificador 101 representado en la Figura 14.

40 La Figura 31 es un diagrama para describir una zona en la que se registran parámetros de codificación.

La Figura 32 es un diagrama de flujo para describir un proceso de determinación de tipo de imagen modificable que se realiza por el dispositivo de codificación 106 representado en la Figura 14.

45 La Figura 33 es un diagrama que ilustra una realización ejemplo de un cambio en el tipo de imagen.

La Figura 34 es un diagrama que ilustra otra realización ejemplo de un cambio en el tipo de imagen.

50 La Figura 35 es un diagrama para describir un proceso de control de cuantización realizado por el dispositivo de codificación 106 representado en la Figura 14.

La Figura 36 es un diagrama de flujo para describir el proceso de control de cuantización realizado por el dispositivo de codificación 106 representado en la Figura 14.

55 La Figura 37 es un diagrama de bloques que ilustra la configuración de un transcodificador de alto grado de acoplamiento 101.

La Figura 38 es un diagrama para describir una sintaxis para un flujo de una secuencia de vídeo.

60 La Figura 39 es un diagrama para describir la configuración de la sintaxis en la Figura 38.

La Figura 40 es un diagrama para describir una sintaxis para el flujo denominado `history_stream()` que tiene una información del registro histórico de una longitud fija allí registrada.

65 La Figura 41 es un diagrama para describir la sintaxis para el flujo denominado `history_stream()` que tiene la información del registro histórico de la longitud fija allí registrada.

- La Figura 42 es un diagrama para describir la sintaxis para el flujo denominado `history_stream()` que tiene la información de registro histórico de la longitud fija allí registrada.
- 5 La Figura 43 es un diagrama para describir la sintaxis para el flujo denominado `history_stream()` que tiene la información de registro histórico de la longitud fija allí registrada.
- La Figura 44 es un diagrama para describir la sintaxis para el flujo denominado `history_stream()` que tiene la información de registro histórico de la longitud fija allí registrada.
- 10 La Figura 45 es un diagrama para describir la sintaxis para el flujo denominado `history_stream()` que tiene la información de registro histórico de la longitud fija allí registrada.
- La Figura 46 es un diagrama para describir la sintaxis para el flujo denominado `history_stream()` que tiene la información de registro histórico de la longitud fija allí registrada.
- 15 La Figura 47 es un diagrama para describir la sintaxis para el flujo denominado `history_stream()` que tiene la información de registro histórico de la longitud variable allí registrada.
- 20 La Figura 48 es un diagrama para describir una sintaxis para la cabecera de secuencia `sequence_header()`.
- La Figura 49 es un diagrama para describir una sintaxis para la extensión de secuencia `sequence_extension()`.
- La Figura 50 es un diagrama para describir una sintaxis para la extensión y datos de usuarios bajo la forma `extension_and_user_data()`.
- 25 La Figura 51 es un diagrama para describir una sintaxis para datos de usuarios bajo la forma `user_data()`.
- La Figura 52 es un diagrama para describir una sintaxis para el grupo de cabeceras de imágenes bajo la forma `group_of_pictures_header()`.
- 30 La Figura 53 es un diagrama para describir una sintaxis para la cabecera de imagen bajo la forma `picture_header()`.
- La Figura 54 es un diagrama para describir una sintaxis para la extensión de codificación de imagen bajo la forma `picture_coding_extension()`.
- 35 La Figura 55 es un diagrama para describir una sintaxis para datos de extensión bajo la forma `extension_data()`.
- La Figura 56 es un diagrama para describir una sintaxis para la extensión de matriz de cuantización bajo la forma `quant_matrix_extension()`.
- 40 La Figura 57 es un diagrama para describir una sintaxis para la extensión de derechos de autor bajo la forma `copyright_extension()`.
- La Figura 58 es un diagrama para describir una sintaxis para la extensión de visualización de imagen bajo la forma `picture_display_extension()`.
- 45 La Figura 59 es un diagrama para describir una sintaxis para datos de imagen bajo la forma `picture_data ()`.
- 50 La Figura 60 es un diagrama para describir una sintaxis para `slice()`.
- La Figura 61 es un diagrama para describir una sintaxis para `macroblock()`.
- La Figura 62 es un diagrama para describir una sintaxis para modos de macrobloques bajo la forma `macroblock_modes()`.
- 55 La Figura 63 es un diagrama para describir una sintaxis para vectores de movimiento bajo la forma `motion_vectors(s)`.
- La Figura 64 es un diagrama para describir una sintaxis para vectores de movimiento bajo la formas `motion_vector(r, s)`.
- 60 La Figura 65 es un diagrama para describir códigos de longitud variable del tipo de macrobloque para imágenes I.
- 65 La Figura 66 es un diagrama para describir códigos de longitud variable del tipo de macrobloque para imágenes P.

La Figura 67 es un diagrama para describir códigos de longitud variable del tipo de macrobloque para imágenes B.

La Figura 68 es un diagrama para describir una sintaxis para información de flujo de recodificación bajo la forma `re_coding_stream_info()`.

5 La Figura 69 es un diagrama para describir los indicadores bajo las formas `red_bw_flag` y `red_bw_indicator`.

La Figura 70 es un diagrama para describir un conjunto de datos de parámetros de codificación de información de registro histórico.

10 La Figura 71 es un diagrama para describir la formación de macrobloques bajo la forma de `Re_coding Information Bus`.

La Figura 72 es un diagrama para describir los elementos de tasas de imágenes `Picture rate`.

15 La Figura 73 es un diagrama para describir los elementos de tasas de imágenes `Picture rate`.

La Figura 74 es un diagrama para describir los elementos de tasas de imágenes `Picture rate`.

20 La Figura 75 es un diagrama una zona en la que se registra `Re_Coding Information Bus`.

La Figura 76 es un diagrama de bloques que representa la configuración, a modo de ejemplo, de un sistema de registro de grabadora de cinta de vídeo.

25 La Figura 77 es un diagrama de bloques que representa una realización, a modo de ejemplo, de la configuración de un sistema de reproducción de grabadora de cinta de vídeo.

La Figura 78 es un diagrama de bloques que representa otra realización, a modo de ejemplo, de la configuración del sistema de registro de grabadora de cinta de vídeo.

30 La Figura 79 es un diagrama de bloques que representa una realización, a modo de ejemplo, de la configuración del sistema de reproducción de grabadora de cinta de vídeo.

35 La Figura 80 es un diagrama para describir las posiciones de registro de un flujo de vídeo y flujo de registro histórico bajo la forma `history_stream`.

#### FORMAS DE REALIZACIÓN PREFERIDAS DE LA INVENCION

40 Antes de un transcodificador al que se aplica la presente invención que se describe a continuación, se explicará la codificación global de señales de imágenes en movimiento. Los términos utilizados para un sistema se refieren, en este caso, al sistema completo configurado mediante medios y dispositivos plurales.

45 A modo de ejemplo, los sistemas telefónicos o de conferencias de televisión que transmiten señales de imágenes en movimiento a lugares distantes utilizan una correlación de líneas o intertramas entre las señales de vídeo y señales de imagen de código compresivo con el fin de utilizar eficientemente las rutas de transmisión.

El uso de la correlación de líneas permite la compresión de señales de imágenes, a modo de ejemplo, por medio de un proceso de DCT (Transformación Cosenoidal Discreta).

50 Además, el uso de la correlación intertramas permite que las señales de imágenes se compriman todavía más para la codificación. Según se ilustra, extremo, en la Figura 1, si imágenes de tramas PC1 a PC3 se producen durante puntos temporales t1 a t3, la diferencia en la señal de imagen entre las imágenes de tramas PC1 y PC2 se calculan para generar PC12 y la diferencia entre las imágenes de tramas PC2 y PC3 se calcula para generar PC23. Las imágenes en tramas temporalmente adyacentes no suelen tener variaciones de gran magnitud, por lo que el cálculo de la diferencia entre ellas da lugar a una señal diferencial de un pequeño valor. En consecuencia, esta señal diferencial puede utilizarse para codificación para comprimir la magnitud de codificación.

60 Sin embargo, la transmisión de solamente señales diferenciales no permite el restablecimiento de las imágenes originales. De este modo, la imagen en cada trama se codifica en uno de tres tipos de imagen que incluyen imágenes I, P y B en orden para la codificación compresiva de las señales de imágenes.

65 Es decir, según se ilustra, a modo de ejemplo, en la Figura 2, las señales de imágenes para 17 tramas, que incluyen las tramas F1 a F17 se agrupan en un grupo de imágenes (GOP), que es una unidad de procesamiento. A continuación, la señal de imagen en la trama F1 se codifica en la imagen I, la segunda trama F2 se codifica en la imagen B y la tercera trama F3 se codifica en la imagen P. Las cuarta y posteriores tramas F4 a F17 se codifican, de forma alternativa, en imágenes B o P.

Para la señal de imagen I, la señal de imagen para la trama correspondiente se transmite tal como está. Por el contrario, para la señal de imagen P, básicamente, la señal diferencial entre esta señal de imagen y la imagen I o P temporalmente precedente se transmite según se ilustra en la Figura 2. Además, para la señal de imagen P, básicamente, la señal diferencial entre esta señal de imagen y la media de las tramas temporalmente precedentes y siguientes se codifica según se ilustra en la Figura 3.

La Figura 4 ilustra el principio de un método de codificación de señales de imágenes en movimiento. Según se ilustra en esta Figura, puesto que la primera trama F1 se procesa como la imagen I, se transmite a una ruta de transmisión como F1X (codificación intra-imagen) de datos transmitidos. Por el contrario, puesto que la segunda trama F2 se procesa como la imagen B, la señal diferencial entre esta trama y la media de la trama temporalmente precedente F1 y la trama temporalmente siguiente F3 se calcula y transmite como F2X de datos transmitidos.

De hecho, existen cuatro tipos de procesamiento para las imágenes B. Un primer tipo de procesamiento transmite los datos en la trama original F2 como la F2X de datos transmitidos sin ningún cálculo (SP1) (intracodificación); este procesamiento es similar al de las imágenes I. Un segundo tipo de procesamiento calcula y transmite la señal diferencial (SP2) entre la trama original F2 y la trama temporalmente siguiente F3 (codificación predictiva inversa). Un tercer tipo de procesamiento transmite la señal diferencial (SP3) entre la trama original F2 y la trama temporalmente precedente F1 (codificación predictiva directa). Además, un cuarto tipo de procesamiento genera la señal diferencial (SP4) entre la trama original F2 y la media de la trama temporalmente precedente F1 y la trama temporalmente siguiente F3 y la transmite como la F2X (codificación predictiva bidireccional) de datos transmitidos.

Uno de los cuatro métodos anteriormente descritos que transmiten la más pequeña cantidad de datos se utiliza realmente.

En la transmisión de los datos diferenciales, un vector en movimiento  $x_1$  (entre las tramas F1 y F2) (en el caso de predicción directa) o  $x_2$  (entre las tramas F3 y F2) (en el caso de predicción inversa) o ambas (en el caso de predicción bidireccional) entre la trama original y la imagen (imagen de referencia) en la trama para la que la señal diferencial con la trama original ha de calcularse y es objeto también de transmisión.

Además, para la trama F3 de imagen P, la trama temporalmente precedente F1 se utiliza como la imagen de referencia para calcular la señal diferencial (SP3) entre las tramas F3 y F1 así como el vector de movimiento  $x_3$ , que se transmiten ambas como F3X (codificación predictiva directa) de datos transmitidos. Como alternativa, los datos en la trama original F3 se transmiten como la F3X (SP1) (intracodificación) de los datos. Como en el caso de las imágenes B, se selecciona uno de estos métodos que transmite la menor cantidad de datos.

La Figura 5 ilustra una realización, a modo de ejemplo, de la configuración de los dispositivos para codificar y transmitir señales de imágenes en movimiento y decodificarlas sobre la base del principio anteriormente descrito. Un dispositivo de codificación 1 está adaptado para codificar y transmitir señales de vídeo de entrada a un soporte de registro 3 que actúa como una ruta de transmisión. Un dispositivo de decodificación 2 está adaptado para reproducir las señales registradas en el soporte de registro 3 y decodificar y proporcionarlas a la salida.

En un dispositivo de codificación 1, señales de vídeo de entrada se introducen en un circuito de preproceso 11, en donde las imágenes son separadas cada una en una señal de brillo y una señal de color (en esta forma de realización, una señal de diferencia de color). Las señales de diferencia de brillo y color, que son analógicas, se convierten en señales digitales por los convertidores A/D 12, 13, respectivamente. Las señales digitales en las que ha sido convertida la señal de vídeo por los convertidores A/D 12,13 se suministran a una memoria de tramas 14 para su almacenamiento. La memoria de tramas 14 memoriza la señal de vídeo en una memoria de tramas de señales de brillo 15, mientras que memoriza la señal de diferencia de color en una memoria de tramas de señales de diferencia de color 16.

Un circuito convertidor de formatos 17 convierte las señales memorizadas en la memoria de tramas 14 en un formato de trama, hacia un formato de bloque. Es decir, según se ilustra en la Figura 6, las señales de vídeo memorizadas en la memoria de tramas 14 son del formato de trama ilustrado en la Figura 6 (A), que está configurado por líneas V cada una constituida por H puntos. El circuito convertidor de formatos 17 efectúa la partición de la señal para una sola trama en M segmentos configurando cada uno por 16 líneas según se ilustra en la Figura 6 (B). Cada segmento se divide en M macrobloques. Según se ilustra en la Figura 6 (C), el macrobloque está configurado por una señal de brillo correspondiente a  $16 \times 16$  pixels (puntos), y esta señal de brillo es objeto, además, de partición en bloques Y[1] a Y[4] estando cada una configurada por  $8 \times 8$  puntos. La señal de brillo de los  $16 \times 16$  puntos corresponde a una señal Cb de  $8 \times 8$  puntos y una señal Cr de  $8 \times 8$  puntos.

De esta manera los datos convertidos en el formato de bloques se suministran por el circuito convertidor de formatos 17 a un codificador 18, en donde se realiza la codificación correspondiente. Los detalles se describirán haciendo referencia a la Figura 7.

Una señal obtenida mediante codificación por el codificador 18 se proporciona a la salida a una ruta de transmisión



como un flujo de bits. A modo de ejemplo, la señal se suministra a un circuito de registro 19 y se registra en el soporte de registro 3 como una señal digital.

Los datos reproducidos desde el soporte de registro 3 por un circuito de reproducción 30 del dispositivo de decodificación 2 se suministran a un decodificador 31 para su decodificación. Los detalles del decodificador 31 se describirán a continuación haciendo referencia a la Figura 12.

Los datos obtenidos mediante la decodificación por el decodificador 31 se aplican a la entrada de un circuito convertidor de formatos 32, en donde el formato de bloques se convierte en el formato de trama. A continuación, la señal de brillo en el formato de trama se suministra a una memoria de tramas de señales de brillo 34 de una memoria de tramas 33 para su almacenamiento y la señal de diferencia de color se suministra a una memoria de tramas de diferencia de color 35 para su almacenamiento. Las señales de diferencia de brillo y color son objeto de lectura desde la memoria de tramas de señales de brillo 34 y la memoria de tramas de señales de diferencia de color 35, respectivamente y posteriormente, se convierten por los convertidores D/A 36, 37 en señales analógicas, que se suministran luego a un circuito de postproceso 38. El circuito de postproceso 38 sintetiza las señales de diferencia de brillo y de color antes de la salida.

A continuación, la configuración del codificador 18 se describirá haciendo referencia a la Figura 7. Los datos de imágenes a codificarse se aplican a la entrada de un circuito de detección de vectores de movimiento 50 en términos de macrobloques. El circuito de detección de vectores de movimiento 50 procesa los datos de imágenes en cada trama como la imagen I, P, o B en conformidad con una secuencia predeterminada preestablecida. Si la imagen de entrada secuencial en cada trama se procesa como la imagen I, P o B se determina de antemano (a modo de ejemplo, el Grupo de imágenes configuradas por las tramas F1 a F17 se procesan como imágenes I, B, P, B, P, ..., B, P según se ilustra en las Figuras 2 y 3).

Los datos de imágenes en una trama a procesarse como la imagen I (a modo de ejemplo, la trama F1) se transfieren desde el circuito de detección de vectores de movimiento 50 a una sección de imagen original directa 51a de la memoria de tramas 51 para su almacenamiento. Los datos de imágenes en una trama a procesarse como la imagen B (a modo de ejemplo, la trama F2) se transfieren a una sección de imagen original 51b para su almacenamiento. Los datos de imágenes en una trama a procesarse como la imagen P (a modo de ejemplo, la trama F3) se transfieren a una sección de imagen original inversa 51c para su almacenamiento.

Además, cuando una imagen a una trama a procesarse como la imagen B (la trama F4) o la imagen P (la trama F5) se aplica a la entrada en conformidad con la siguiente temporización, los datos de imagen en la primera imagen P (la trama F3) memorizada en la sección de imagen original inversa 51c se transfieren a la sección de imagen original directa 51a; los datos de imágenes en la siguiente imagen B (la trama F4) se memorizan en la sección de imagen original de referencia 51b (los datos son objeto de sobreescritura en la sección 51b); y los datos de imágenes en la siguiente imagen P (la trama F5) se memorizan en la sección de imagen original inversa 51c (los datos son objeto de sobreescritura en la sección 51c). Dichas operaciones se repiten de forma secuencial.

Las señales para cada imagen memorizada en la memoria de tramas 51 son objeto de lectura y transferidas a un circuito de conmutación de modos de predicción 52, en donde se realiza un proceso de modo de predicción de trama o un proceso de modo de predicción de campo de imagen.

Además, bajo el control de un circuito de determinación de predicción 54, un dispositivo calculador 53 realiza un cálculo para una predicción intra-imagen, una predicción directa, una predicción inversa o una predicción bidireccional. Qué proceso se realiza dependerá de una señal de error de predicción (la señal diferencial entre una imagen de referencia a procesarse y una imagen de predicción). De este modo el circuito de detección de vectores de movimiento 50 genera la suma absoluta (o suma cuadrática) de señales de error de predicción que se utilizan para esta determinación.

A continuación, se describirá un modo de predicción de trama y un modo de predicción de campo de imagen en el circuito de conmutación de modos de predicción 52.

Si se establece el modo de predicción de trama, el circuito de conmutación de modo de predicción 52 proporciona, a la salida, los cuatro bloques de brillo Y[1] a Y[4] suministrados por el circuito de detección de vectores de movimiento 50, al siguiente dispositivo calculador 53 tal como están. Es decir, en este caso, los datos en las líneas de campo de imagen impares y los datos en las líneas de campos de imagen pares se mezclan en cada bloque de brillo según se ilustra en la Figura 8. En el modo de predicción de tramas, las predicciones se realizan utilizando cuatro bloques de brillo (macrobloques) en un momento determinado y los cuatro bloques de brillo corresponden a un vector de movimiento.

Por el contrario, en el modo de predicción de campo de imagen, el circuito de conmutación de modo de predicción 52 configura, antes de la salida al dispositivo calculador 53, una entrada de señal procedente del circuito de detección de vectores de movimiento 50 y que se configuran según se ilustra en la Figura 8 de modo que se compriman los cuatro bloques de brillo, los bloques de brillo Y[1] e Y[2], a modo de ejemplo, de los puntos en las líneas de campos impares, mientras que los otros dos bloques de brillo Y[3] e Y[4], se configuran por los puntos en

las líneas de campo pares según se ilustra en la Figura 9. En este caso, los dos bloques de brillo Y[1] e Y[2] corresponden a un solo vector de movimiento, mientras que los otros dos bloques de brillo Y[3] e Y[4] corresponden a otro vector de movimiento.

5 El circuito de detección de vectores de movimiento 50 proporciona, a la salida, las sumas absolutas de errores de predicción en los modos de predicción de trama y de campo de imagen al circuito de conmutación de modos de predicción 52. El circuito etapa de daos conmutación de modos de predicción 52 comparas las sumas absolutas de errores de predicción en los modos de predicción de trama y de campo de imágenes juntos y luego, realiza el procesamiento correspondiente al modo de predicción con el más pequeño valor antes de proporcionar, a la salida  
10 datos al dispositivo calculador 53.

En realidad, dicho procesamiento se realiza por el circuito de detección de vectores de movimiento 50. Es decir, el circuito de detección de vectores de movimiento 50 proporciona, a la salida, una señal configurada en correspondencia con el modo determinado para el circuito de conmutación de modos de predicción 52, mientras que  
15 el circuito de conmutación de modos de predicción 52 proporciona, a la salida, esta señal al siguiente dispositivo calculador 53 tal como está.

En el modo de predicción de trama, la señal de diferencia de color con los datos en las líneas de campos impares y los datos en las líneas de campo pares se mezcla según se ilustra en la Figura 8 y se entregan al dispositivo  
20 calculador 53. Además, en el modo de predicción de campo, la mitad superior (cuatro líneas) de cada bloque de diferencia de color Cb, Cr constituye una señal de diferencia de color para los campos impares correspondientes a los bloques de brillo Y[1], Y[2], mientras que la mitad inferior (cuatro líneas) constituye una señal de diferencia de color para los campos pares correspondientes a los bloques de brillo Y[3], Y[4], según se ilustra en la Figura 9.

25 Además, el circuito de detección de vectores de movimiento 50 utiliza el circuito de determinación de predicción 54 para generar la suma absoluta de errores de predicción requeridos para seleccionar una de entre la predicción intra-imagen, la predicción directa, la predicción inversa y la predicción bidireccional que han de realizarse, según se ilustra a continuación.

30 Como la suma absoluta de los errores de predicción intra-imagen, se determina la diferencia entre el valor absoluto  $|\Sigma A_{ij}|$  de la suma  $\Sigma A_{ij}$  de las señales de macrobloques Aij para la imagen de referencia y la suma  $\Sigma |A_{ij}|$  de los valores absolutos  $|A_{ij}|$  de las señales de macrobloques Aij. Como la suma absoluta de los errores de predicción directa, se determina la suma  $\Sigma |A_{ij} - B_{ij}|$  de los valores absolutos  $|A_{ij} - B_{ij}|$  de las diferencias Aij - Bij entre las señales de macrobloques Aij para las señales de referencia y las señales de macrobloques Bij para una imagen de predicción.  
35 Las sumas absolutas de los errores de predicción inversa y los errores de predicción bidireccional se determinan también como en la predicción directa (la imagen de predicción se cambia a una imagen diferente de la obtenida en la predicción directa).

Estas sumas absolutas se suministran al circuito de determinación de predicción 54. El circuito de determinación de predicción 54 selecciona una de las sumas absolutas de los errores de predicción directa, inversa y bidireccional como la suma absoluta de errores de interpredicción. El circuito de determinación de predicción 54 compara, además, la suma absoluta de los errores de interpredicción con la suma absoluta de los errores de predicción intra-imagen, selecciona el más pequeño de estos dos valores y selecciona un modo correspondiente a la suma absoluta seleccionada, como un modo de predicción. Es decir, si el valor absoluto de los errores de intrapredicción es más  
45 pequeño, se establece el modo de predicción intra-imagen. Si el valor absoluto de los errores de interpredicción es más pequeño, se establece uno de los modos de predicción directa, inversa y bidireccional que tiene la más pequeña suma absoluta correspondiente.

De esta manera, el circuito de detección de vectores de movimiento 50 suministra las señales de macrobloques para la imagen de referencia al dispositivo calculador 53 por intermedio del circuito de conmutación de modos de predicción 52 en una manera tal que las señales se configuren en correspondencia con el modo seleccionado por el  
50 circuito de conmutación de modos de predicción 52 en el modo de predicción de trama o de campo de imagen. El circuito de detección de vectores de movimiento 50 detecta luego un vector de movimiento entre la imagen de predicción y la imagen de referencia que corresponde a uno de los cuatro modos de predicción que se selecciona por el circuito de determinación de predicción 54 y proporciona, a la salida, el vector a un circuito de codificación de longitud variable 58 y circuito de compensación de movimiento 64. Según se describió con anterioridad, este vector de movimiento se selecciona para tener la más pequeña suma absoluta correspondiente de errores de predicción.

Mientras el circuito de detección de vectores de movimiento 50 está realizando la lectura de los datos de imágenes de tipo I a partir de la sección de imagen original directa 51a, el circuito de determinación de predicción 54 establece un modo de predicción intratrama o intracampo (imagen) (un modo sin compensaciones de movimiento) y conmuta un conmutador 53d del dispositivo calculador 53 a un contacto g. De este modo, los datos de imágenes del tipo I se aplican a la entrada de un circuito de conmutación de modo de DCT 55.

65 El circuito de conmutación de modo DCT 55 configura los datos en los cuatro bloques de brillo de modo que las líneas de campos impares y pares se mezclen juntas (un modo DCT de trama) o estos dos tipos de líneas de campo

se separan entre sí (un modo DCT de campo de imagen), según se ilustra en la Figura 10 o 11 y luego, proporciona a la salida los datos configurados a un circuito DCT 56.

5 Es decir, el circuito de conmutación de modo DCT 55 compara una codificación eficientemente realizada por un proceso DCT con las instalación de campo impares y pares mezcladas juntas, con una eficiencia de codificación conseguida por un proceso DCT con las líneas de campo impares y pares separadas entre sí, para seleccionar un modo con una más alta eficiencia de codificación.

10 A modo de ejemplo, la señal de entrada se configura de modo que las líneas de campo impares y pares se mezclen juntas según se ilustra en la Figura 10 y la diferencia entre las señales para cada par de las líneas de campo impares y pares verticalmente adyacentes se calcula y se determina la suma (o suma cuadrática) de los valores absolutos de estas diferencias.

15 Además, la señal de entrada está configurada de modo que las líneas de campo impares estén separadas de las líneas de campo pares según se ilustra en la Figura 11 y la diferencia entre las señales para cada par de las líneas de campo impares verticalmente adyacentes y la diferencia entre las señales para cada par de líneas de campo pares verticalmente adyacentes se calcula y la suma (o suma cuadrática) de los valores absolutos de estas diferencias es objeto de determinación.

20 Además, ambos valores absolutos se comparan para establecer un modo DCT correspondiente al más pequeño valor. Es decir, si el anterior es más pequeño, se establece el modo de DCT de trama mientras que si este último es más pequeño, se establece el modo DCT de campo de imagen.

25 Datos configurados en correspondencia con el modo DCT seleccionado se proporcionan al circuito DCT 56 y un indicador DCT que indica el modo DCT seleccionado se proporciona al circuito de codificación de longitud variable 58 y al circuito de compensación de movimiento 64.

30 Como resulta evidente de la comparación entre el modo de predicción (Figuras 8 y 9) en el circuito de conmutación de modos de predicción 52 y el modo DCT (Figuras 10 y 11) en el circuito de conmutación de modo DCT 55, la estructura de datos de los bloques de brillo en ambos modos son prácticamente idénticas.

35 Si el circuito de conmutación de modos de predicción 52 selecciona el modo de predicción de trama (las líneas impares y pares se mezclan juntas), en tal caso, el circuito de conmutación de modo DCT 55 es probable que seleccione el modo DCT de trama (las líneas impares y pares se mezclan juntas). Además, si el circuito de conmutación de modos de predicción 52 selecciona el modo de predicción de campo de imagen (los datos de campos impares y pares se separan entre sí), en tal caso, el circuito de conmutación de modo DCT 55 es probable que seleccione el modo DCT de campo (los datos de campos impares y pares están separados entre sí).

40 El modo, sin embargo, no se selecciona siempre de esta manera, sino que el circuito de conmutación de modos de predicción 52 determina el modo con el fin de minimizar la suma absoluta de errores de predicción mientras que el circuito de conmutación de modo DCT 55 determina el modo con el fin de conseguir una alta eficiencia de codificación.

45 La salida de datos de imagen de tipo I proporcionados por el circuito de conmutación de modo DCT 55 se aplica a la entrada del circuito DCT 56, en donde se realiza el proceso DCT para convertir los datos en un coeficiente DCT. Este coeficiente DCT se aplica a la entrada de un circuito de cuantización 57, en donde se cuantifica utilizando una escala de cuantización correspondiente a la cantidad de datos memorizados en una memoria intermedia de transmisión 59 (la cantidad de datos memorizados en la memoria intermedia). Los datos cuantificados se aplican luego a la entrada de un circuito de codificación de longitud variable 58.

50 El circuito de codificación de longitud variable 58 convierte los datos de imágenes (en este caso, los datos de imágenes I) suministrados por el circuito de cuantización 57 en códigos de longitud variables tales como códigos Huffman en una forma correspondiente a la escala de cuantización (scale) suministrada por el circuito de cuantización 57 antes de la salida a la memoria intermedia de transmisión 59.

55 El circuito de codificación de longitud variable 58 tiene también una entrada a la escala de cuantización (scale) desde el circuito de cuantización 57, el modo de predicción (que indica una de entre la predicción intra-imagen, la predicción directa, la predicción inversa y la predicción bidireccional que ha de establecerse), desde el circuito de determinación de predicción 54, el vector de movimiento procedente del circuito de detección de vectores de movimiento 50 y el indicador de predicción (que indica si se establece el modo de predicción de trama o de campo de imagen) procedente del circuito de conmutación de modos de predicción 52, el indicador DCT (que indica si se establece el modo DCT de trama o de campo de imagen) se proporciona por el circuito de conmutación de modo DCT 55, que se codifican en longitud variable en su totalidad.

65 La memoria intermedia de transmisión 59 memoriza temporalmente los datos de entrada y proporciona, a la salida, datos correspondientes a la cantidad de datos memorizados al circuito de cuantización 57. Cuando la cantidad de

- datos restantes aumenta hasta un valor límite superior admisible, la memoria intermedia de transmisión 59 utiliza una señal de control de cuantización para aumentar la escala de cuantización para el circuito de cuantización 57 con el fin de reducir la cantidad de datos objeto de cuantización. Por el contrario, cuando la cantidad de datos restantes disminuye hasta un valor límite inferior admisible, la memoria intermedia de transmisión 59 utiliza la señal de control de cuantización para disminuir la escala de cuantización para el circuito de cuantización 57 con el fin de aumentar la cantidad de datos objeto de cuantización. De esta forma, se impide que la memoria intermedia de transmisión 59 tenga un flujo excesivo o un flujo insuficiente.
- Los datos memorizados en la memoria intermedia de transmisión 59 son objeto de lectura en conformidad con las temporizaciones predeterminadas y se proporcionan, a la salida, a la ruta de transmisión. Los datos se registran luego en el soporte de registro 3 por intermedio, a modo de ejemplo, del circuito de registro 19.
- Por otro lado, los datos de imagen I se proporcionan por el circuito de cuantización 57 como una entrada a un circuito de cuantización inversa 60, en donde son objeto de cuantización inversa en una forma correspondiente a la escala de cuantización suministrada por el circuito de cuantización 57. Una salida desde el circuito de cuantización inversa 60 se aplica a la entrada de un circuito IDCT 61 (Transformación Cosenoidal Discreta Inversa) para la transformación cosenoidal discreta inversa y el resultado se suministra a una sección de imagen de predicción directa 63a de una memoria de tramas 63 por intermedio de un dispositivo calculador 62 para su almacenamiento.
- Al procesar los datos de imágenes en tramas de entrada de forma secuencial, a modo de ejemplo, las imágenes I, B, P, B, P, B, ..., el circuito de detección de vectores de movimiento 50 procesa los datos de imágenes en la primera trama de entrada como la imagen I y antes de procesar la imagen en la siguiente trama de entrada como la imagen B procesa los datos de imágenes en la trama de entrada siguiente como la imagen P. Esto es así porque la imagen B implica la predicción inversa, de modo que la imagen P, la imagen de predicción inversas, debe proporcionarse antes de que pueda decodificarse la imagen B.
- De este modo, después del procesamiento de la imagen I, el circuito de detección de vectores de movimiento 50 inicia el procesamiento de los datos de imágenes de tipo P memorizados en la sección de imagen original 51c. A continuación, de forma similar al caso anteriormente descrito, la suma absoluta de los diferenciales de intertramas (errores de predicción) en términos de los macrobloques se suministra por el circuito de detección de vectores de movimiento 50 al circuito de conmutación de modos de predicción 52 y al circuito de determinación de predicción 54. El circuito de conmutación de modos de predicción 52 y el circuito de determinación de predicción 54 establecen el modo de predicción de trama/campo de imagen, o el modo de predicción intra-imagen, directa, inversa o bidireccional dependiendo de la suma absoluta de los errores de predicción en los macrobloques de imagen P.
- Cuando se establece el modo de predicción intra-imagen, el dispositivo calculador 53 conmuta conmutador 53d al contacto a según se describió con anterioridad. De este modo, estos datos, de forma similar a los datos de imagen I, se transmiten a la ruta de transmisión por intermedio del circuito de conmutación de modo DCT 55, el circuito DCT 56, el circuito de cuantización 57, el circuito de codificación de longitud variable 58, y al memoria intermedia de transmisión 59. Además, estos datos se entregan a una sección de imagen de predicción inversa 63b de la memoria de tramas 63 por intermedio del circuito de cuantización inversa 60, el circuito IDCT 61, y el dispositivo calculador 62 para su almacenamiento.
- Además, si se establece el modo de predicción directa, el conmutador 53d se conmuta a un contacto b y los datos de imágenes (en este caso, la imagen de tipo I) memorizados en la sección de imagen de predicción directa 63a de la memoria de tramas 63 es objeto de lectura y se somete a una compensación de movimiento por el circuito de compensación de movimiento 64 en una forma correspondiente al vector de movimiento proporcionado por el circuito de detección de vectores de movimiento 50. Es decir, cuando el circuito de determinación de predicción 54 controla el establecimiento del modo de predicción directa, el circuito de compensación de movimiento 64 efectúa la lectura de datos desplazando una dirección objeto de lectura a la sección de imagen de predicción directa 63a, desde una posición correspondiente al macrobloque actualmente proporcionado por el circuito de detección de vectores de movimiento 50, en una magnitud correspondiente al vector de movimiento. El circuito de compensación de movimiento 64 genera, de este modo, datos de imágenes objeto de predicción.
- Los datos de imágenes objeto de predicción proporcionados por el circuito de compensación de movimiento 64 se suministran a un dispositivo calculador 53a. El dispositivo calculador 53a sustrae de los datos de macrobloques en la imagen de referencia suministrada por el circuito de conmutación de modos de predicción 52, los datos de imágenes de predicción correspondientes a este macrobloque y suministrados por el circuito de compensación de movimiento 64, para proporcionar los datos diferenciales (error de predicción). Estos datos diferenciales se transmiten a la ruta de transmisión por intermedio del circuito de conmutación de modo DCT 55, el circuito DCT 56, el circuito de cuantización 57, el circuito de codificación de longitud variable 58 y la memoria intermedia de transmisión 59. Además, estos datos diferenciales se decodifican, a nivel local, por el circuito de cuantización inversa 60 y el circuito IDCT 61 antes de aplicarse a la entrada del dispositivo calculador 62.
- Los mismos datos de imágenes de predicción que se suministran al dispositivo calculador 53a se proporcionan también al dispositivo calculador 62. El dispositivo calculador 62 añade los datos de imágenes de predicción

proporcionados por el circuito de compensación de movimiento 64 a los datos diferenciales proporcionados por el circuito IDCT 61. En consecuencia, se obtiene los datos de imágenes en la imagen P original (decodificada). Los datos de imágenes de tipo P se suministran a la sección de imagen de predicción inversa 63b de la memoria de tramas 63 para su almacenamiento.

5 Una vez que los datos de imágenes I y P han sido memorizados en la sección de imagen de predicción directa e inversa 63a y 63b, respectivamente, el circuito de detección de vectores de movimiento 50 procesa la imagen B. El circuito de conmutación de modos de predicción 52 y el circuito de determinación de predicción 54 establecen el modo de trama/campo de imagen dependiendo de la magnitud de la suma absoluta de las Figuras intertramas en términos de los macrobloques y establecen uno de entre el modo de predicción intra-imagen, el modo de predicción directa, el modo de predicción inversa y el modo de predicción bidireccional.

10 Según se describió con anterioridad, en el modo de predicción intra-imagen o directa, el conmutador 53d se conmuta al contacto a o b, respectivamente. En este punto, se realiza un procesamiento similar al de la imagen P para transmitir los datos.

15 Por el contrario, cuando se establece el modo de predicción inversa o bidireccional, el conmutador 53d se conmuta a contacto c o d, respectivamente.

20 En el modo de predicción inversa con el conmutador 53d conmutado al contacto c, los datos de imágenes (en este caso, la imagen de tipo P) memorizados en la sección de imagen de predicción inversa 63b son objeto de lectura y sometidos a una compensación de movimiento por el circuito de compensación de movimiento 64 en una forma correspondiente al vector de movimiento proporcionado por el circuito de detección de vectores de movimiento 50. Es decir, cuando el circuito de determinación de predicción 54 controla el modo de predicción directa a establecer, el

25 el circuito de compensación de movimiento 64 efectúa una lectura de los datos desplazando una dirección de lectura hacia la sección de imagen de predicción inversa 63b, desde una posición correspondiente al macrobloque actualmente proporciona por el circuito de detección de vectores de movimiento 50, en una magnitud correspondiente al vector de movimiento. El circuito de compensación de movimiento 64 genera, de este modo, datos de imagen de predicción.

30 Los datos de imagen de predicción proporcionados por el circuito de compensación de movimiento 64 se suministran a un dispositivo calculador 53b. El dispositivo calculador 53b sustrae de los datos de macrobloques en la imagen de referencia suministrada por el circuito de conmutación de modos de predicción 52, los datos de imágenes de predicción suministrados por el circuito de compensación de movimiento 64, para proporcionar a la salida los datos diferenciales. Estos datos diferenciales se transmiten a la ruta de transmisión por intermedio del circuito de conmutación de modo DCT 55, el circuito DCT 56, el circuito de cuantización 57, el circuito de codificación de longitud variable 58 y la memoria intermedia de transmisión 59.

40 En el modo de predicción bidireccional con el conmutador 53d conmutado al contacto d, los datos de imágenes (en este caso, la imagen de tipo I) memorizados en la sección de imagen de predicción directa 63a y los datos de imágenes (en este caso, la imagen de tipo P) memorizados en la sección de imagen de predicción inversa 63b son objeto de lectura y se someten a una compensación de movimiento por el circuito de compensación de movimiento 64 en una forma correspondiente al vector de movimiento proporcionado por el circuito de detección de vectores de movimiento 50.

45 Es decir, cuando el circuito de determinación de predicción 54 controla el modo de predicción bidireccional a establecer, el circuito de compensación de movimiento 64 efectúa la lectura de datos desplazando las direcciones de lectura en las secciones de imagen de predicción directa e inversa 63a y 63b, desde una posición correspondiente al macrobloque actualmente proporciona por el circuito de detección de vectores de movimiento 50, en una magnitud correspondiente al vector de movimiento (en este caso, los vectores de movimiento se utilizan para las predicciones directa transcodificador inversa). El circuito de compensación de movimiento 64 genera, de este modo, datos de imágenes de predicción.

50 Los datos de imágenes de predicción proporcionados por el circuito de compensación de movimiento 64 se suministran a un dispositivo calculador 53c. El dispositivo calculador 53c sustrae del macrobloque unos datos sobre la imagen de referencia suministrada por el circuito de detección de vectores de movimiento 50, el valor medio de los datos de imágenes de predicción suministrados por el circuito de compensación de movimiento 64, para proporcionar los datos diferenciales. Estos datos diferenciales se transmiten a la ruta de transmisión por intermedio del circuito de conmutación de modo DCT 55, el circuito DCT 56, el circuito de cuantización 57, el circuito de codificación de longitud variable 58 y la memoria intermedia de transmisión 59.

60 Puesto que la imagen de tipo B no se utiliza como una imagen de predicción para otras imágenes, no se memoriza en la memoria de tramas 63.

65 En la memoria de tramas 63, las secciones de imagen de predicción directa e inversa 63a y 63b pueden realizar una conmutación de bancos cuando se requiera para la conmutación a los datos memorizados en una o la otra de estas

secciones con el fin de proporcionarla a una imagen de referencia predeterminada como la imagen de predicción directa o inversa.

5 Aunque la anterior descripción se centra sobre los bloques de brillo, los bloques de diferencia de color se procesan y transmiten, de forma similar, en términos de los macrobloques ilustrados en las Figuras 8 a 11. El vector de movimiento utilizado para procesar los bloques de diferencia de color se obtiene reduciendo el vector de movimiento para los bloques de brillo correspondientes a la mitad en las direcciones vertical y horizontal.

10 La Figura 12 es un diagrama de bloques que ilustra la configuración del decodificador 31 en La Figura 5. Los datos de imágenes codificados, transmitidos por intermedio de la ruta de transmisión (el soporte de registro 3) se recibe por un circuito de recepción (no ilustrado) o se reproduce por un dispositivo de reproducción y se memorizan temporalmente en una memoria intermedia de recepción 81 para su suministro a un circuito de decodificación de longitud variable 82 de un circuito de decodificación 90. El circuito de decodificación de longitud variable 82 somete los datos suministrados por la memoria intermedia de recepción 81 a una decodificación de longitud variable y proporciona, a la salida, el vector de movimiento obtenido, el modo de predicción, el indicador de predicción y el indicador DCT a un circuito de compensación de movimiento 87. El circuito de decodificación de longitud variable 82 proporciona, a la salida, la escala de cuantización obtenida a un circuito de cuantización inversa 83 y proporciona los datos de imágenes decodificados a un circuito de cuantización inversa 83.

20 El circuito de cuantización inversa 83 realiza una cuantización inversa de los datos de imágenes suministrados por el circuito de decodificación de longitud variable 82 en una forma correspondiente a la escala de cuantización suministrada por el circuito de decodificación de longitud variable 82, para proporcionar los datos en cuantización inversa a un circuito IDCT 84. Los datos (coeficiente DCT) proporcionados por el circuito de cuantización inversa 83 se someten a una transformación cosenoidal discreta inversa por el circuito IDCT 84 y los datos procesados se entregan a un dispositivo calculador 85.

30 Si los datos de imágenes entregados al dispositivo calculador 85 por el circuito IDCT 84 es para la imagen I, se proporciona a la salida por el dispositivo calculador 85 y se suministran a una sección de imagen de predicción directa 86a de una memoria de tramas 86 para su almacenamiento con el fin de generar datos de imágenes de predicción para los datos de imágenes (los datos de imágenes P o B) aplicados más adelante a la entrada del dispositivo calculador 85. Además, estos datos se proporcionan al circuito de conversión de formatos 32 (Figura 5).

35 Si los datos de imágenes entregados por el circuito IDCT 84 son para la imagen P, que utiliza los datos de imágenes en la trama precedente, como los datos de imágenes de predicción y está en el modo de predicción directa, en tal caso, los datos de imágenes (los datos de imagen I) en la trama precedente memorizados en la sección de imagen de predicción directa 86a de la memoria de tramas 86 es objeto de lectura y se somete a una compensación de movimiento por el circuito de compensación de movimiento 87 en una formas correspondiente al vector de movimiento proporcionado por el circuito de decodificación de longitud variable 82. Los datos resultantes se añaden por el dispositivo calculador 85 a los datos de imágenes (datos diferenciales) suministrados por el circuito IDCT 84 y los datos así obtenidos son objeto de salida. Los datos resultantes de la adicional, es decir, los datos de imágenes P decodificados se suministra a una sección de imagen de predicción inversa 86b de la memoria de tramas 86 para su almacenamiento con el fin de generar datos de imágenes de predicción para los datos de imágenes (los datos de imagen B o P) aplicados a la entrada del dispositivo calculador 85 más adelante.

45 De forma similar a los datos de imagen I, los datos en el modo de predicción intra-imagen se memorizan en la sección de imagen de predicción inversa 86b sin procesarse por el dispositivo calculador 85, aun cuando sí lo es para la imagen P.

50 Esta imagen P ha de visualizarse después de la siguiente imagen B, de modo que no se proporcione al circuito de conversión de formatos 32 en este punto (según se describió con anterioridad, la imagen P se introduce después de que se procese la imagen B y se transmite antes de la imagen B).

55 Si los datos de imágenes entregados por el circuito IDCT 84 es para la imagen B, los datos de imágenes de tipo I memorizados en la sección de imagen de predicción directa 86a de la memoria de tramas 86 (en el caso del modo de predicción directa), los datos de imágenes P memorizados en la sección de imagen de predicción inversa 86b de la memoria de tramas 86 (en el caso del modo de predicción inversa) o ambos datos de imágenes (en el caso de modo de predicción bidireccional) son objeto de lectura dependiendo del modo de predicción suministrado por el circuito de decodificación de longitud variable 82 y luego, se someten a una compensación de movimiento por el circuito de compensación de movimiento 87 en una forma correspondiente al vector de movimiento suministrado por el circuito de decodificación de longitud variable 82. De este modo, se genera una imagen de predicción. Sin embargo, no se genera ninguna imagen de predicción si no se requiere ninguna compensación de movimiento (en el caso del modo de predicción intra-imagen).

65 Los datos sometidos a la compensación de movimiento por el circuito de compensación de movimiento 87 en la manera anteriormente descrita se añaden por el dispositivo calculador 85 a la salida desde el circuito IDCT 84. Esta salida adicional se proporciona para el circuito de conversión de formatos 32.

Esta salida adicional, sin embargo, comprende los datos de imágenes B y no se utiliza para generar una imagen de predicción para otras imágenes. De este modo, no se memoriza en la memoria de tramas 86.

5 Después del suministro de la imagen B, los datos de imágenes P memorizados en la sección de imagen de predicción inversa 86b son objeto de lectura y se suministran al dispositivo calculador 85 por intermedio del circuito de compensación de movimiento 87. Sin embargo, ninguna compensación de movimiento se proporciona en este punto operativo.

10 Aunque el decodificador ilustrado 31 no tiene ningún circuito correspondiente al circuito de conmutación de modos de predicción 52 y el circuito de conmutación de modo DCT 55 en el codificador 18 en la Figura 5, el circuito de compensación de movimiento 87 realiza el procesamiento correspondiente a estos circuitos, es decir, uno para la recuperación de la configuración en donde señales de líneas de campos impares están separadas de las señales de líneas de campo pares, hacia la configuración original.

15 Además, aunque la descripción anterior es para el procesamiento de las señales de brillo, las señales de diferencia de color se procesan de modo similar. En este caso, sin embargo, el vector de movimiento se obtiene reduciendo el vector de movimiento para las señales de brillo a la mitad en ambas direcciones vertical y horizontal.

20 La Figura 13 ilustra la calidad de las imágenes codificadas. La calidad (SNR: Relación de Señal a Ruido) de las imágenes se controla en correspondencia con el tipo de imagen en tal manera que las imágenes I y P sean ambas de una alta calidad, mientras que la imagen B tiene una calidad menor que las imágenes I y P. Este método utiliza características visuales humanas; es decir, se obtiene una más alta calidad visual haciendo vibrar la calidad de cada imagen que nivelando la calidad de todas las imágenes. El control de la calidad de la imagen, que depende del tipo de imagen, se realiza por el circuito de cuantización 57 en la Figura 7.

25 Las Figuras 14 y 15 ilustran la configuración de un transcodificador 101 al que se aplica la presente invención; la Figura 15 ilustra la configuración detallada que se representa en la Figura 14. El transcodificador 101 convierte la estructura GOP y la tasa binaria de una entrada de flujo de video codificado a un dispositivo de decodificación 102 en la estructura deseada por un operador. Para explicar las funciones del transcodificador 101, se supone que tres transcodificadores que tiene funciones similares a las del transcodificador 101 esta localizados, de forma conectiva, antes del transcodificador 101, aunque no se ilustra en la Figura 15. Es decir, con el fin de variar la estructura GOP y la tasa binaria de un flujo de bits, los primero, segundo y tercero transcodificadores están conectados en serie en este orden y un cuarto transcodificador, ilustrado en la Figura 15, está situado, de forma conectiva, después del  
35 tercer transcodificador.

En la siguiente descripción de la presente invención, un proceso de codificación realizado en el primer transcodificador se define como proceso de codificación de la primera generación, uno en el segundo transcodificador, que está situado, de forma conectiva, después del primer transcodificador, se define como un  
40 proceso de codificación de la segunda generación, uno en el tercer transcodificador, que está situado, de forma conectiva, después del segundo transcodificador, se define como un proceso de codificación de la tercera generación y uno en el cuarto transcodificador (el transcodificador 101 ilustrado en la Figura 15), que está situado, de forma conectiva, después del tercer transcodificador, se define como un proceso de codificación de la cuarta generación o proceso de codificación en curso.

45 Además, los parámetros de codificación generados durante el proceso de codificación de la primera generación se denominan "parámetros de codificación de la primera generación", los parámetros de codificación generados durante el proceso de codificación de la segunda generación se denominan "parámetros de codificación de la segunda generación", los parámetros de codificación generados durante el proceso de codificación de la tercera generación se denominan "parámetros de codificación de la tercera generación" y los parámetros de codificación generados durante el proceso de codificación de la cuarta generación se denominan "parámetros de codificación de la cuarta  
50 generación".

En primer lugar, un flujo de vídeo codificado ST(3rd) suministrado al transcodificador 101 ilustrado en la Figura 15 será descrito a continuación. El flujo ST(3rd) representa un flujo codificado de la tercera generación generado durante el proceso de codificación de la tercera generación realizado por el tercer transcodificador proporcionado antes del transcodificador 101. En el flujo de vídeo codificado ST(3rd) generado durante el proceso de codificación de la tercera generación, los parámetros de codificación de la tercera generación generados durante el proceso de codificación de la tercera generación se describen en una capa de secuencias, una capa de GOP, una capa de imágenes, una capa de segmentos y una capa de macrobloques de este flujo de vídeo codificado ST (3rd) como una  
60 función de cabecera de secuencia denominada `sequence_header()`, una función de extensión de secuencia denominada `sequence_extension()`, una función de cabeceras de grupo de imágenes denominada `group_of_pictures_header()`, una función de cabecera de imagen denominada `picture_header()`, una función de extensión de codificación de imagen denominada `picture_coding_extension()`, una función de datos de imagen denominada `picture_data()`, una función de segmentos denominada `slice ()` y una función de macrobloques denominada `macroblock()`. La descripción de los terceros parámetros de codificación utilizados durante el tercer  
65

proceso de codificación en el tercer flujo codificado generado durante el tercer proceso de codificación se define en la norma MPEG2 y no constituye ninguna novedad inventiva.

5 El transcodificador 101 según la presente invención es único por cuanto que el tercer flujo codificado ST (3rd) ha descrito los terceros parámetros de codificación así como los parámetros de codificación de la primera y de la segunda generación generados durante los primeros y segundos procesos de codificación, respectivamente.

10 Más concretamente, los parámetros de codificación de la primera y de la segunda generación se describen en una zona de datos de usuario de la capa de imágenes del flujo de vídeo codificado de la tercera generación ST(3rd) como un flujo de registro histórico denominado `history_stream()`. En la presente invención, el flujo de registro histórico descrito en la zona de datos de usuario de la capa de imágenes del flujo de vídeo codificado de la tercera generación ST(3rd) se denomina "información de registro histórico" y los parámetros de codificación descritos en el flujo de registro histórico se denominan "history parameters".

15 Como alternativa, si los parámetros de codificación de la tercera generación descritos en el flujo codificado de la tercera generación ST(3rd) se denominan los "parámetros de codificación en curso", puesto que los procesos de codificación de la primera y de la segunda generación se realizan antes del proceso de codificación de la tercera generación, los parámetros de codificación descritos como un flujo de registro histórico en la zona de datos de usuario de la capa de imágenes del flujo codificado de la tercera generación ST(3rd) se denominan también "parámetros de codificación anteriores".

20 La razón por la que el flujo codificado de la tercera generación ST(3rd) ha descrito los terceros parámetros de codificación así como los parámetros de codificación de la primera y de la segunda generación generados durante los primeros y segundos procesos de codificación, respectivamente, según se describió con anterioridad es que puede impedirse que la calidad de la imagen sea objeto de degradación incluso con cambios repetidos en la estructura GOP o la tasa binaria del flujo codificado mediante un proceso de transcodificación.

25 A modo de ejemplo, se considera que una imagen puede codificarse en el tipo P durante el proceso de codificación de la primera generación, que esta imagen P puede codificarse en el tipo B durante el proceso de codificación de la segunda generación con el fin de cambiar la estructura GOP del flujo codificado de la primera generación y que esta imagen B puede codificarse de nuevo en el tipo P durante el proceso de codificación de la tercera generación con el fin de cambiar todavía más la estructura GOP del flujo codificado de la segunda generación. Es conocido en esta técnica que habida cuenta que los procesos de codificación y decodificación basados en la norma MPEG no son 100 % reversibles, la calidad de la imagen puede degradarse con la repetición de estos procesos.

30 En este caso los parámetros de codificación tales como la escala de cuantización, el vector de movimiento y el modo de predicción que han sido generados durante el proceso de codificación de la primera generación se reutilizan durante el proceso de codificación de la tercera generación en lugar de calcular estos parámetros de codificación de nuevo durante el proceso anterior. Los parámetros de codificación tales como la escala de cuantización, el vector de movimiento y el modo de predicción que se han generado por el proceso de codificación de la primera generación son evidentemente más exactos que estos parámetros recientemente generados por el proceso de codificación de la tercera generación, de modo que la reutilización de los parámetros de la primera generación puede producir una menor degradación de la calidad de la imagen a pesar de la repetición de los procesos de codificación y decodificación.

35 El procesamiento anteriormente descrito según la presente invención se describirá con mayor detalle tomando, a modo de ejemplo, un procesamiento realizado por el transcodificador de la cuarta generación 101 que se ilustra en la Figura 15.

40 Un dispositivo de decodificación 102 utiliza los parámetros de codificación de la tercera generación para decodificar vídeos codificados procedentes del flujo de bits codificado de la tercera generación ST(3rd) y genera datos de vídeo digitales para las bandas base decodificadas. Además, el dispositivo de decodificación 102 decodifica también los primeros y segundos parámetros de codificación descritos como un flujo de registro histórico en la zona de datos de usuario de la capa de imágenes del flujo de bits codificado de la tercera generación ST(3rd).

45 Más concretamente, según se ilustra en la Figura 16, el dispositivo de decodificación 102 se configura básicamente en la misma manera que el decodificador 31 (Figura 12) del dispositivo de decodificación 2 ilustrado en la Figura 5; comprende una memoria intermedia de recepción 81 para la memorización de un flujo de bits suministrado, un circuito de decodificación de longitud variable 112 para someter el flujo de bits codificado a una decodificación de longitud variable, un circuito de cuantización inversa 83 para realizar una cuantización inversa de los datos decodificados de longitud variable en conformidad con la escala de cuantización suministrada desde el circuito de decodificación de longitud variable 112, un circuito IDCT 84 para someter el coeficiente DCT de cuantización inversa a una transformación cosenoidal discreta inversa y un dispositivo calculador 85, una memoria de tramas 86 y un circuito de compensación de movimiento 87 para realizar un proceso de compensación de movimiento.

50 Para decodificar el flujo de bits codificado de la tercera generación ST (3rd), el circuito de decodificación de longitud



variable 112 extrae los parámetros de codificación de la tercera generación descritos en las capas de imágenes, de segmentos y de macrobloques del flujo de bits codificado de la tercera generación ST (3rd). A modo de ejemplo, los parámetros de codificación de la tercera generación extraídos por el circuito de decodificación de longitud variable 112 incluyen el tipo de codificación de imagen denominado `picture_coding_type` que indica el tipo de imagen, el código de escala del dispositivo de cuantización denominado `quantiser_scale_code` que indica una magnitud de la etapa de escala de cuantización, `macroblock_type` que indica el modo de predicción, `motion_vector` que indica el vector de movimiento, `frame/field_motion_type` que indica el modo de predicción de trama o el modo de predicción de campo de imagen, `dct_type` que indica el modo DCT de trama o el modo DCT de campo de imagen, etc. El código `quantiser_scale_code` extraído por el circuito de decodificación de longitud variable 112 se suministra al circuito de decodificación de cuantización inversa 83 y los parámetros tales como los denominados `picture_coding_type`, `quantiser_scale_code`, `macroblock_type`, `motion_vector`, `frame/field_motion_type`, y `dct_type` se entregan al circuito de compensación de movimiento 87.

El circuito de decodificación de longitud variable 112 extrae desde las capas de secuencia, GOP, imagen, segmento y macrobloques del flujo de bits codificado de la tercera generación ST(3rd) no solamente los parámetros de codificación anteriores requeridos para decodificar el flujo de bits codificado de la tercera generación ST (3rd) sino también los parámetros de codificación a transmitirse al siguiente transcodificador de la quinta generación como información de registro histórico de la tercera generación. Por supuesto, los parámetros de codificación de la tercera generación tales como los parámetros de los tipos denominados `picture_coding_type`, `quantiser_scale_code`, `macroblock_type`, `motion_vector`, `frame/field_motion_type`, y `dct_type` que se utilizan para el proceso de decodificación de la tercera generación están incluidos en la información de registro histórico de la tercera generación. Los parámetros de codificación a extraerse como la información de registro histórico son preestablecidos por el operador o un ordenador central dependiendo de la capacidad de transmisión.

Además, el circuito de decodificación de longitud variable 112 extrae datos de usuarios descritos en la zona de datos de usuario en la capa de imágenes del flujo de bits codificado de la tercera generación ST(3rd), para suministrar estos datos a un dispositivo de decodificación de registro histórico 104.

El dispositivo de decodificación de registro histórico 104 es un circuito para extraer los parámetros de codificación de la primera y de la segunda generación descritos como información de registro histórico (parámetros de codificación que preceden a los parámetros de codificación de la generación precedente) a partir de los datos de usuarios descritos en la capa de imágenes del flujo de bits codificado de la tercera generación ST (3rd). Más concretamente, el dispositivo de decodificación de registro histórico 104 puede analizar la sintaxis de los datos de usuarios recibidos para detectar un identificador único denominado `History_Data_Id` descritos en los datos de usuarios, de modo que se extraiga el flujo denominado `converted_history_stream()`. Además, el dispositivo de decodificación de registro histórico 104 puede obtener un flujo de registro histórico denominado `history_stream()` eliminando los bits de marcador de un bit (`marker_bit`) insertados en el flujo denominado `converted_history_stream()` en intervalos penados y puede analizar la sintaxis del flujo de registro histórico denominado `history_stream()` para obtener los parámetros de codificación de la primera y de la segunda generación descritos en el flujo de registro histórico `history_stream()`. La descripción detallada del dispositivo de decodificación de registro histórico 104 se describirá más adelante.

Un dispositivo de multiplexación de información de registro histórico 103 es un circuito para multiplexar los parámetros de codificación de la primera, segunda y tercer generación sobre los datos de vídeo de banda base decodificados por el dispositivo de decodificación 102 con el fin de suministrar estos parámetros a un dispositivo de codificación 106 que realiza el proceso de codificación de la cuarta generación. Más concretamente, el dispositivo de multiplexación de información de registro histórico 103 recibe los datos de vídeo de banda base procedentes del dispositivo calculador 85 del dispositivo de decodificación 102, los parámetros de codificación de la tercera generación procedentes del dispositivo de decodificación de longitud variable 112 del dispositivo de decodificación 102 y los parámetros de codificación de la primera y de la segunda generación procedentes del dispositivo de decodificación de registro histórico 104, para multiplexar estos parámetros de codificación de la primera, segunda y tercera generación sobre los datos de vídeo de banda base. Los datos de vídeo de banda base multiplexados con los parámetros de codificación de la primera, segunda y tercera generación se suministran a un dispositivo de separación de información de registro histórico 105 por intermedio de un cable de transmisión.

A continuación, el método para multiplexar los parámetros de codificación de la primera, segunda y tercera generación sobre los datos de vídeo de banda base se describirán haciendo referencia a las Figuras 17 y 18. La Figura 17 ilustra un macrobloque constituido por 16 x 16 pixels según se define en la norma MPEG. El macrobloque de 16 x 16 pixels está configurado por cuatro subbloques (Y[0], [1], [2] e Y[3]) para las señales de brillo que está constituido cada uno por 8 x 8 pixels y cuatro subbloques (Cr[0], r[1], b[0], y Cb[1]) para las señales de diferencia de color que está constituida cada una por 8 x 8 pixels.

La Figura 18 ilustra un determinado formato de datos de vídeo, este formato se define en la recomendación RDT 601 de la ITU y representa lo que se denomina el "formato D1", que se utiliza en el sector de la difusión. El formato D1 está normalizado para la transmisión de datos de vídeo de 10 bits y de este modo, permite que un pixel de datos de vídeo sea expresado con 10 bits.

Puesto que los datos de vídeo de banda base decodificados en con formato con la norma MPEG se configuran por 8 bits, el transcodificador en conformidad con la presente invención utiliza los ocho bits superiores de los 10 bits en el formato D1 (D9 a D2) según se ilustra en la Figura 18, para transmitir los datos de vídeo de banda base decodificados sobre la base de la norma MPEG. Cuando los datos de vídeo de 8 bits así decodificados son objeto de escritura en el formato D1, los dos bits inferiores (D1 y D0) no son asignados. El transcodificador, en conformidad con la presente invención, utiliza esta zona no asignada para transmitir la información de registro histórico.

Puesto que el bloque de datos descrito en la Figura 18 se utiliza para transmitir un pixel en cada subbloque (Y[0], Y[1], Y[2], Y[3], Cr [0], Cr [1], Cb[0], Cb[1]), 64 bloques de datos tales como los ilustrados en la Figura 18 se transmiten con el fin de transmitir un macrobloque de datos. El uso de dos bits inferiores (D1 y D0) permite que se transmitan 1024 (= 16 x 64) bits de información de registro histórico para un solo macrobloque de datos de vídeo. En consecuencia, puesto que la información de registro histórico para una generación está configurada por 256 bits, la información de registro histórico para las cuatro generaciones anteriores (= 1024/256) pueden superponerse en un solo macrobloque de datos de vídeo. En la realización, a modo de ejemplo, ilustrada en la Figura 18, la información de registro histórico de la primera, segunda y tercera generación se superpone entre sí.

El dispositivo de separación de información de registro histórico 105 es un circuito para extraer los datos de vídeo de banda base desde los ocho bits superiores de los datos transmitidos en el formato D1 mientras se extrae la información de registro histórico desde los dos bits inferiores. En la realización, a modo de ejemplo, ilustrada en la Figura 15, el dispositivo de separación de información de registro histórico 105 extrae los datos de vídeo de banda base a partir de los datos transmitidos, suministra estos datos a un dispositivo de codificación 106, realiza la información de registro histórico de la primera, segunda y tercera generación a partir de los datos transmitidos y suministra la información al dispositivo de codificación 106 y a un dispositivo de codificación de registro histórico 107.

El dispositivo de codificación 106 codifica los datos de vídeo de banda base suministrados desde el dispositivo de separación de información de registro histórico 105 en un flujo de bits que tiene una estructura GOP y una tasa binaria especificada por el operador o el ordenador central. Cambiar la estructura GOP significa cambiar el número de imágenes contenidas en el GOP, el número de imágenes P presentes entre las imágenes I y el número de imágenes B presentes entre las imágenes I y las imágenes P (o imágenes I).

En la realización ilustrada, a modo de ejemplo, en la Figura 15, puesto que los datos de vídeo de banda base suministrados tienen la información de registro histórico de la primera, segunda y tercera generación superpuestas, el dispositivo de codificación 106 reutiliza, de forma selectiva, la información de registro histórico para un proceso de codificación para la cuarta generación con el fin de disminuir la posible degradación de la calidad de la imagen que se deriva del proceso de recodificación.

La Figura 19 es un diagrama que muestra una configuración específica de un dispositivo de codificación 106 proporcionado en el dispositivo de codificación 106. El dispositivo de codificación 106 está configurado básicamente de forma similar al codificador 18 ilustrado en la Figura 7; comprende el circuito de detección de vectores de movimiento 50, el circuito de conmutación de modos de predicción de trama/campo de imagen 52, el dispositivo calculador 53, el circuito de conmutación de modo DCT 55, el circuito DCT 56, el circuito de cuantización 57, el circuito de codificación de longitud variable 58, la memoria intermedia de transmisión 59, el circuito de cuantización inversa 60, el circuito DCT inverso 61, el dispositivo calculador 62, la memoria de tramas 63, y el circuito de compensación de movimiento 64. Puesto que las funciones de cada uno de los circuitos son casi las mismas que las del codificador 18 que se describe en la Figura 7, se omite aquí su descripción. La siguiente descripción se referirá a los dispositivos entre el dispositivo de codificación 106 y el codificador 18 descrito en la Figura 7.

El dispositivo de codificación 106 tiene un controlador 70 para controlar las operaciones y funciones de cada uno de los circuitos anteriormente descritos. El controlador 70 recibe una instrucción sobre la estructura GOP desde el operador o el ordenador central para determinar el tipo de imagen de cada imagen en una forma correspondiente a la estructura GOP especificada. El controlador 70 recibe también información sobre una tasa binaria objetivo desde el operador o desde el ordenador central para controlar el circuito de cuantización 57 de modo que una tasa binaria procedente del dispositivo de codificación 106 sea igual a la que fue especificada.

Además, el controlador 70 recibe información del registro histórico para una pluralidad de generaciones que se proporciona desde el dispositivo de separación de información de registro histórico 105 para reutilizar la información con el fin de codificar la información de referencia. Los detalles correspondientes se describirán a continuación.

En primer lugar, el controlador 70 determina si el tipo de imagen de la imagen de referencia determinada a partir de la estructura GOP especificada por el operador es el mismo que el tipo de imagen contenido en la información de registro histórico. Es decir, se determina si la imagen de referencia se ha codificado, o no, en el tipo de imagen que es el mismo que el que fue especificado.

Para un mejor conocimiento, se hará referencia a la realización ejemplo ilustrada en la Figura 15. El controlador 70 determina si el tipo de imagen asignado a la imagen de referencia para el proceso de codificación de la cuarta

generación es el mismo o no que el tipo de imagen de la imagen de referencia para el proceso de codificación de la primera, segunda o tercera generación.

5 Si el tipo de imagen especificado para la imagen de referencia para el proceso de codificación de la cuarta generación es diferente de todos los tipos de imagen para los procesos de codificación anteriores, el controlador 70 realiza un denominado "proceso de codificación normal". Es decir, en este caso, esta imagen de referencia no ha sido codificada, durante el proceso de codificación de la primera, segunda o tercera generación, en el tipo de imagen asignado para el proceso de codificación de la cuarta generación. Por otro lado, si el tipo de imagen especificado para la imagen de referencia para el proceso de codificación de la cuarta generación es el mismo que el de los tipos de imagen para los procesos de codificación anteriores, en tal caso, el controlador 70 realiza un denominado "proceso de codificación de parámetros reutilizados". Es decir, en este caso, esta imagen de referencia ha sido codificada, durante el proceso de codificación de la primera, segunda o tercera generación, en el tipo de imagen asignado para el proceso de codificación de la cuarta generación.

15 En primer lugar, el proceso de codificación normal del controlador 70 será descrito a continuación.

Con el fin de determinar si seleccionar o no, el modo de predicción de trama o de campo de imagen, el circuito de detección de vectores de movimiento 50 detecta un error de predicción tanto en el modo de predicción de trama como en el modo de predicción de campo de imagen y suministra la diferencia entre los errores de predicción al controlador 70. El controlador 70 compara los valores de errores de predicción juntos para seleccionar el modo de predicción con el más pequeño valor. El circuito de conmutación de modos de predicción 52 procesa señales dependiendo del módulo de predicción seleccionado por el controlador 70 para suministrar las señales procesadas al dispositivo calculador 53.

25 Más concretamente, si se ha seleccionado el modo de predicción de trama, el circuito de conmutación de modos de predicción 52 procesa una señal de brillo de entrada de modo que la señal se proporcione al dispositivo calculador 53 tal como está, mientras que se procesa una señal de diferencia de color de entrada de modo que la señal tenga líneas de campos impares y pares mezcladas juntas, según se describe haciendo referencia a la Figura 8. Por otro lado, si se ha seleccionado el modo de predicción de campo de imagen, el circuito de conmutación de modos de predicción 52 procesa la señal de brillo en tal manera que los bloques de brillo Y[1] y Y[2] sean configurados por líneas de campo impares, mientras que los bloques de brillo Y[3] e Y [4] sean configurados mediante líneas de campo pares, según se describe haciendo referencia a la Figura 9. El circuito de conmutación de modos de predicción 52 procesa también la señal de diferencia de color de tal manera que las cuatro líneas superiores sean configuradas por líneas de campo impares, mientras que las cuatro líneas inferiores sean configuradas por líneas de campos pares, también según se describe haciendo referencia a la Figura 9.

Además, para seleccionar uno de entre el modo de predicción intra-imagen, el modo de predicción directa, el modo de predicción inversa y el modo de predicción bidireccional, el circuito de detección de vectores de movimiento 50 genera y suministra errores de predicción en estos modos de predicción al dispositivo controlador 70. El dispositivo controlador 70 selecciona los más pequeños de los errores de predicción directa, inversa y bidireccional como un error de interpredicción. Además, compara este error de interpredicción con el error de predicción de intra-imagen para seleccionar el más pequeño valor con el fin de seleccionar como un modo de predicción a un modo correspondiente al error de predicción seleccionado. Es decir, si el error de predicción intra-imagen es más pequeño, se establece el modo de predicción intra-imagen. Si el error de interpredicción es más pequeño, se establece uno de los modos de predicción directa, inversa y bidireccional que tenga el más pequeño error de predicción. El dispositivo controlador 70 controla el dispositivo calculador 53 y el circuito de compensación de movimiento 64 en una forma correspondiente al modo de predicción seleccionado.

50 Para seleccionar el modo DCT de trama o de campo de imagen, el circuito de conmutación de modo DCT 55 convierte los datos en los cuatro bloques de brillo en una forma de señal con líneas de campos impares y pares mezcladas juntas (el modo DCT de trama), mientras se convierten los mismos datos en una forma de señal con estos dos tipos de líneas de campos separadas entre sí (el modo DCT de campo de imagen). El circuito de conmutación de modo DCT 55 entrega, entonces, estas señales al circuito DCT 56. El circuito DCT 56 calcula la eficiencia de codificación de un proceso DCT con líneas de campos impares y pares mezcladas juntas y la eficiencia de codificación de un proceso DCT con estos dos tipos de campos separados entre sí y entrega el resultado al dispositivo controlador 70. El dispositivo controlador 70 compara las eficiencias de codificación entregadas desde el circuito DCT 56 para seleccionar uno de los modos DCT que tiene la más alta eficiencia de codificación con el fin de controlar el circuito de conmutación de modo DCT 55 para introducir el modo DCT seleccionado.

60 El controlador 70 recibe una tasa binaria objetivo suministrada por el operador o el ordenador central y una señal indicativa de la cantidad de bits memorizados en la memoria intermedia de transmisión 59, es decir, una señal indicativa de la magnitud de la memoria intermedia restante y sobre la base de la tasa binaria objetivo y la magnitud de la memoria intermedia restante, genera un código de escala denominado feedback\_q\_scale\_code para controlar una magnitud de etapa de cuantización para el circuito de cuantización 57. El código feedback\_q\_scale\_code es una señal de control generada dependiendo de la magnitud de memoria intermedia remanente en la memoria intermedia de transmisión 59, para excluir la presencia de un flujo excesivo o insuficiente. Esta señal de control controla

también la tasa binaria de un flujo de bits procedente de la memoria intermedia de transmisión 59 hacia un valor objetivo.

Más concretamente, si, a modo de ejemplo, la cantidad de bits memorizados en la memoria intermedia de transmisión 59 disminuye, la magnitud de la etapa de cuantización se reduce para aumentar la cantidad de bits generados para la siguiente imagen que se va a codificar. Por el contrario, si aumenta la cantidad de bits memorizados en la memoria intermedia de transmisión 59, se incrementa la magnitud de la etapa de cuantización para reducir la cantidad de bits generados para la siguiente imagen objeto de codificación. La magnitud de la etapa de cuantización es proporcional al código denominado `feedback_q_scale_code`, y se aumenta y decreta en consecuencia.

A continuación, se explicará un proceso de codificación de parámetros reutilizados, que es una característica del transcodificador 101. Para facilitar el entendimiento de este proceso, la imagen de referencia se supone que ha sido codificada en la imagen P durante el proceso de codificación de la primera generación, en la imagen I durante el proceso de codificación de la segunda generación y en la imagen B durante el proceso de codificación de la tercera generación, de modo que debe codificarse en la imagen P durante el proceso de codificación de la cuarta generación.

En este caso, puesto que se ha codificado la imagen de referencia, durante el proceso de codificación de la primera generación, en el mismo tipo de imagen (imagen I) como se asigna como el tipo de imagen de la cuarta generación, el controlador 70 realiza un proceso de codificación utilizando parámetros de codificación de la primera generación sin generar nuevos parámetros de codificación a partir de los datos de vídeo suministrados. Los parámetros de codificación reutilizados para el proceso de codificación de la cuarta generación suelen incluir un parámetro denominado `quantiser_scale_code` indicativo de la magnitud de la etapa de escala de cuantización, un parámetro `macroblock_type` indicativo del modo de dirección de predicción, un parámetro `motion_vector` indicativo del vector de movimiento, un parámetro `frame/field_motion` indicativo del modo de predicción de trama o de campo de imagen y un parámetro `dct_type` indicativo del modo DCT de trama o de campo de imagen.

El controlador 70 no reutiliza todos los parámetros de codificación transmitidos como la información de registro histórico sino que reutiliza los parámetros de codificación que se suponen que sean deseablemente reutilizados según se describió con anterioridad, mientras que se genera recientemente los parámetros de codificación que no se supone que sean deseablemente reutilizados.

A continuación, este proceso de codificación de parámetros reutilizados se explicará centrándose en las diferencias respecto al proceso de codificación normal anteriormente descrito.

El circuito de detección de vectores de movimiento 50 detecta el vector de movimiento de la imagen de referencia durante el proceso de codificación normal anteriormente descrito, pero durante este proceso de codificación de parámetros reutilizados, reutiliza el vector de movimiento `motion_vector` suministrado como la información de registro histórico de la primera generación sin un proceso de detección. El motivo se describirá más adelante.

Los datos de vídeo de banda base obtenidos decodificando el flujo codificado de la tercera generación se ha sometido a procesos de decodificación y de codificación al menos tres veces, de modo que tenga evidentemente una más baja calidad de imagen que los datos de vídeo originales. Un vector de movimiento exacto no puede detectarse a partir de los datos de vídeo con una calidad de imagen reducida. Es decir, el vector de movimiento suministrado como la información de registro histórico de la primera generación es claramente más exacto que el detectado durante el proceso de codificación de la cuarta generación. Es decir, reutilizando el vector de movimiento suministrado como los parámetros de codificación de la primera generación, el proceso de codificación de la cuarta generación no podrá degradar la calidad de la imagen. El controlador 70 proporciona el vector de movimiento `motion_vector` suministrado como la información de registro histórico de la primera generación, al circuito de compensación de movimiento 64 y al circuito de codificación de longitud variable 58 como información del vector de movimiento para la imagen de referencia, que se codifica durante el proceso de codificación de la cuarta generación.

Además, para determinar si seleccionar el modo de predicción de trama o de campo de imagen, el circuito de detección de vectores de movimiento 50 detecta el error de predicción tanto en el modo de predicción de trama como en el modo de predicción de campo de imagen. Durante este proceso de codificación de parámetros reutilizados, sin embargo, el circuito de detección de vectores de movimiento 50 reutiliza el parámetro `frame/field_motion_type` indicativo del modo de predicción de trama o del modo de predicción de campo de imagen y se suministra como la información de registro histórico de la primera generación, sin necesidad de detectar el error de predicción tanto en el modo de predicción de trama como en el modo de predicción de campo de imagen. Esto es así porque el error de predicción en cada modo de predicción detectado en la primera generación es más exacto que el que se detecta durante el proceso de codificación de la cuarta generación y porque la selección del modo de predicción determinado sobre la base del error de predicción más exacto permite un proceso de codificación más óptimo.

Más concretamente, el dispositivo controlador 70 suministra al circuito de conmutación de modos de predicción 52

una señal de control correspondiente al parámetro `frame/field_motion_type` entregado como la información de registro histórico de la primera generación y el circuito de conmutación de modos de predicción 52 realiza un proceso de señalización correspondiente al parámetro `frame/field_motion_type`.

5 Además, durante el proceso de codificación normal, el circuito de detección de vectores de movimiento 50 calcula el error de predicción en cada modo de dirección de predicción con el fin de determinar la selección de entre el modo de predicción intraimagen, el modo de predicción directa, el modo de predicción inversa y el modo de predicción bidireccional (este modo de predicción se refiere más adelante como un "modo de dirección de predicción"). Durante este proceso de codificación de parámetros reutilizados, sin embargo, el circuito de detección de vectores de movimiento 50 determina el modo de dirección de predicción sobre la base del tipo de macrobloque suministrado como la información de registro histórico de la primera generación sin calcular el error de predicción en cada modo de dirección de predicción. Esto es así porque el error de predicción en cada modo de dirección de predicción durante el proceso de codificación de la primera generación es más exacto que el detectado durante el proceso de codificación de la cuarta generación y porque la selección del modo de dirección de predicción determinado sobre la base del error de predicción más exacto permite un proceso de codificación más eficiente. Más concretamente, el controlador 70 selecciona el modo de dirección de predicción indicado por el tipo de macrobloque contenido en la información de registro histórico de la primera generación y luego, controla el dispositivo calculador 53 y el circuito de compensación de movimiento 64 en una forma correspondiente al modo de dirección de predicción seleccionado.

20 Durante el proceso de codificación normal, el circuito de conmutación de modo DCT 55 suministra al circuito DCT 56 ambas señales controvertidas en las formas de señales del modo DCT de trama y de campo de imagen con el fin de comparar las eficiencias de codificación en los modos DCT de trama y de campo de imagen. Durante este proceso de codificación de parámetros reutilizados, sin embargo, el circuito de conmutación de modo DCT 55 realiza un procesamiento correspondiente al modo DCT indicado por el parámetro `dct_type` contenido en la información de registro histórico de la primera generación sin generar ambas señales convertidas en la forma de señales del modo DCT de trama y de campo de imagen. Más concretamente, el dispositivo controlador 70 reutiliza el parámetro `dct_type` contenido en la información de registro histórico de la primera generación para controlar el circuito de conmutación de modo DCT 55 con el fin de realizar un proceso de señalización correspondiente al modo DCT indicado por el parámetro `dct_type`.

30 Durante el proceso de codificación normal, el dispositivo controlador 70 controla la magnitud de la etapa de cuantización para el circuito de cuantización 57 sobre la base de la tasa binaria objetivo especificada por el operador y la magnitud de la memoria intermedia de transmisión restante. Durante este proceso de codificación de parámetros reutilizados, el dispositivo controlador 70 controla la magnitud de la etapa de cuantización para el circuito de cuantización 57 sobre la base de la tasa binaria objetivo, la magnitud de la memoria intermedia de transmisión restante y la escala de cuantización anterior contenida en la información de registro histórico. En la descripción siguiente, la escala de cuantización anterior contenida en la información de registro histórico se describe como `history_q_scale_code`. En un flujo de registro histórico, descrito a continuación, esta escala de cuantización se describe también como `quantiser_scale_code`.

40 En primer lugar, el controlador 70 genera el código de escala de cuantización `feedback_q_scale_code` como en el proceso de codificación normal. El parámetro `feedback_q_scale_code` es una magnitud determinada dependiendo de la capacidad de memoria intermedia remanente en la memoria intermedia de transmisión 59 para impedir la presencia de un flujo excesivo y de un flujo deficiente. Posteriormente, el controlador 70 compara el parámetro de escala de cuantización anterior `history_q_scale_code` contenido en el flujo de registro histórico de la primera generación con el parámetro de escala de cuantización en curso `feedback_q_scale_code` para determinar el de mayor magnitud. Una escala de cuantización mayor significa una etapa de cuantización mayor. Si el código de escala de cuantización en curso `feedback_q_scale_code` es mayor que el código de escala de cuantización anterior `history_q_scale_code`, el controlador 70 suministra el código de escala de cuantización en curso `feedback_q_scale_code` al circuito de cuantización 57. Por el contrario, si el código de escala de cuantización anterior `history_q_scale_code` es mayor que el código de escala de cuantización en curso `feedback_q_scale_code`, el controlador 70 suministra el código de escala de cuantización anterior `history_q_scale_code` al circuito de cuantización 57.

55 Es decir, el dispositivo controlador 70 selecciona el mayor código de escala de cuantización de entre la pluralidad de escalas de cuantización anteriores contenidas en la información de registro histórico y la escala de cuantización en curso calculada a partir de la capacidad de memoria intermedia de transmisión remanente. Dicho de otro modo, el controlador 70 controla el circuito de cuantización 57 para realizar una cuantización utilizando la mayor de las etapas de cuantización utilizadas para los procesos de codificación anteriores (de la primera, segunda y tercera generación) y para el proceso de codificación en curso (de la cuarta generación). El motivo se describirá más adelante.

60 A modo de ejemplo, se supone que un flujo generado durante el proceso de codificación de la tercera generación tiene una tasa binaria de 4 Mbps y que una tasa binaria objetivo de 15 Mbps se establece para el dispositivo de codificación 106 que realiza este proceso de codificación de la cuarta generación. Aunque la tasa binaria objetivo sea más alta, no se obtiene un resultado adecuado simplemente reduciendo la etapa de cuantización. La calidad de imagen de una imagen codificada con una etapa de cuantización de gran magnitud durante el proceso de

codificación anterior no puede mejorarse si se codifica con una etapa de cuantización reducida durante el proceso de codificación en curso. Es decir, una codificación con una etapa de cuantización más pequeña que una etapa para un proceso de codificación anterior simplemente aumenta la cantidad de bits y deja de mejorar la calidad de la imagen. De este modo, el proceso de codificación más eficiente puede conseguirse utilizando para la cuantización la mayor de las etapas de cuantización utilizadas para los procesos de codificación anteriores (de la primera, segunda y tercera generación) y para el proceso de codificación en curso (cuarta generación).

A continuación, el dispositivo de decodificación de registro histórico 104 y el dispositivo de codificación de registro histórico 107 en la Figura 15 se describirán a continuación. Aunque en la Figura 15, el dispositivo de decodificación de registro histórico 104, se ilustra como siendo un circuito o dispositivo diferente del dispositivo de decodificación 102, simplemente se ilustra en un bloque diferente del que sirve para ilustrar el dispositivo de decodificación 102 con el fin de describir las funciones y la configuración del dispositivo de decodificación de registro histórico 104 en una manera fácil de entender. El procesamiento por el dispositivo de decodificación de registro histórico 104 se efectúa realmente mediante un circuito de decodificación de longitud variable y un circuito de control de decodificación (un controlador de decodificación) en el dispositivo de decodificación 102. De forma similar, aunque en la Figura 15 el dispositivo de codificación de registro histórico 107 se ilustra como siendo un circuito o dispositivo diferente del dispositivo de codificación 106, simplemente se muestra en un bloque distinto del ilustrado con el dispositivo de decodificación 106, con el fin de describir las funciones y la configuración del dispositivo de decodificación de registro histórico 107 en una manera fácil de entender. El procesamiento por el dispositivo de codificación de registro histórico 107 se efectúa realmente por un circuito de codificación de longitud variable y un circuito de control de codificación (un controlador de codificación) en el dispositivo de codificación 102.

Según se ilustra en la Figura 15, el dispositivo de decodificación de registro histórico 104 está configurado por un decodificador de datos de usuario 201 para decodificar los datos de usuarios suministrados por el dispositivo de decodificación 102, un convertidor 202 para convertir una salida procedente del decodificador de datos de usuario 201 y un VLD de registro histórico 203 para reproducir la información de registro histórico procedente del convertidor 202.

Además, el dispositivo de codificación de registro histórico 107 está configurado por un VLC de registro histórico 211 para dar formato a los parámetros de codificación para las tres generaciones suministradas por el dispositivo de separación de información de registro histórico 105, un convertidor 212 para convertir una salida procedente del VLC de registro histórico 211 y un dispositivo para dar formato a los datos de usuario 213 para formatear una salida procedente del convertidor 212 en un formato de datos de usuarios.

El decodificador de datos de usuario 201 decodifica los datos de usuario suministrados por el dispositivo de decodificación 102 para proporcionar los datos convertidos al convertidor 202. Los detalles serán examinados más adelante haciendo referencia Figura 51. Los datos de usuarios (`user_data()`) consisten en un código `user_data_start_code` y `user-data` y la norma MPEG prohíbe que los datos de usuario `user_data` contengan 23 bits continuos de "0" (que son idénticos a los del código de inicio `start_code`). Lo que antecede es para impedir que estos datos sean erróneamente detectados como el código de inicio `start_code`. La información de registro histórico (`history_stream()`) se describe en la zona de datos de usuario (como un tipo de `user_data` en conformidad con la norma MPEG) y puede contener dichos 23 bits continuos o más de "0," de modo que un "1" debe insertarse en esta información en conformidad con una temporización predeterminada para su conversión en el flujo de registro histórico convertido `converted_history_stream()` (véase la Figura 38 descrita más adelante), con el fin de impedir la presencia de 23 bits o más de "0". Esta conversión se realiza por el convertidor 212 del dispositivo de codificación de registro histórico 107. El convertidor 202 del dispositivo de decodificación de registro histórico 104 realiza un proceso de conversión inversa para el convertidor 212 (que elimina el "1" insertado para impedir la presencia de 23 bits o más de "0").

El VLD de registro histórico 203 genera la información de registro histórico (en este caso, los parámetros de codificación de la primera y de la segunda generación) a partir de la salida procedente desde el convertidor 202 para proporcionarlos al dispositivo de multiplexación de información de registro histórico 103.

Por otro lado, en el dispositivo de codificación de registro histórico 107, el VLC de registro histórico 211 convierte los parámetros de codificación para las tres generaciones (primera, segunda y tercera generaciones) suministrados por el dispositivo de separación de información de registro histórico 105, en un formato de información de registro histórico. Este formato incluye un tipo de longitud fija (véase las Figuras 40 a 46, descritas más adelante) y un tipo de longitud variable (véase Figura 47 y Figuras posteriores). Los detalles se describirán más adelante.

La información de registro histórico formateada por el VLC de registro histórico 211 se convierte en el flujo de registro histórico convertido `converted_history_stream()` por el convertidor 212. Lo que antecede es para impedir que el código de inicio `start_code` se erróneamente detectado desde los `user_data()` según se describió con anterioridad. Es decir, aunque la información de registro histórico contiene 23 bits continuos o más de "0", puesto que dichos 23 bits continuos o más de "0" no se pueden disponer dentro de los `user_data`, el convertidor 212 convierte los datos (inserta un "1" en los datos en conformidad con la temporización predeterminada) con el fin de no desviarse de la prohibición.

El dispositivo para dar formato a los datos de usuario 213 añade el indicador History\_Data\_ID al flujo de registro histórico convertido `converted_history_stream()` suministrado por el convertidor 212, sobre la base de la representación de la Figura 38, descrita más adelante, y añade un código de flujo de datos de usuario `user_data_stream_code` para generar `user_data` en cumplimiento con la norma MPEG de modo que estos datos puedan insertarse en un flujo de vídeo y luego, proporcionarse al dispositivo de decodificación 106.

La Figura 20 representa una realización, a modo de ejemplo, de una configuración del VLC de registro histórico 211. Su convertidor de palabras de código 301 y el convertidor de longitud de código 305 se suministran desde el dispositivo de separación de información de registro histórico 105 con parámetros de codificación (que son actualmente transmitidos como la información de registro histórico) (datos de elementos) e información (a modo de ejemplo, el nombre de una sintaxis (a modo de ejemplo, el nombre de la cabecera de secuencia `sequence_header`, descrita más adelante)) (nº de evento) que identifica un flujo en el que han de disponerse estos parámetros de codificación. El convertidor de palabras de código 301 convierte los parámetros de codificación de entrada en palabras de código correspondientes a una sintaxis indicada y proporciona las palabras de código obtenidas a un dispositivo desplazador 302. El dispositivo desplazador 302 desplaza las palabras de código de la entrada por el convertidor de palabras de código 301 en una magnitud correspondiente a una magnitud de desplazamiento suministrada por un 306, y proporciona las palabras de código desplazadas a un conmutador 303 en bytes. El conmutador 303, que se conmuta por una señal de selección de bits proporciona da por el circuito de generación de direcciones 306, es idéntico en número a los bits requeridos y entrega las palabras de código suministradas por el dispositivo desplazador 302 a una memoria RAM 304 para su almacenamiento. Las direcciones de escritura requeridas se especifican por el circuito de generación de direcciones 306. Además, cuando el circuito de generación de direcciones 306 especifica direcciones objeto de lectura, los datos correspondientes (palabras de código) memorizados en la memoria RAM 304 son objeto de lectura y se suministran al convertidor siguiente 212 y se entregan de nuevo a la memoria RAM 304 por intermedio del conmutador 303 para su almacenamiento.

El convertidor de longitud de código 305 determina la longitud de código de los parámetros de codificación de entrada desde la sintaxis de entrada y estos parámetros se proporcionan al circuito de generación de direcciones 306. El circuito de generación de direcciones 306 genera la magnitud de desplazamiento anteriormente descrita, la señal de selección de bits o las direcciones de escritura o lectura en correspondencia con la longitud del código de entrada para su suministro al dispositivo desplazador 302, el conmutador 303, o la memoria RAM 304, respectivamente.

Según se describió con anterioridad, el VLC de registro histórico 211 está configurado como lo que se denomina un codificador de longitud variable para someter los parámetros de codificación de entrada a una codificación de longitud variable antes de la salida.

La Figura 21 representa una realización, a modo de ejemplo, de una configuración del VLD de registro histórico 203 para decodificar los datos formatos en la información de registro histórico según se describió con anterioridad. En el VLD de registro histórico 203, los datos de parámetros de codificación entregados por el convertidor 202 se suministran y luego se memorizan en una memoria RAM 311. Las direcciones de escritura requeridas se suministran por un circuito de generación de direcciones 315. El circuito de generación de direcciones 315 genera también direcciones objeto de lectura en conformidad con una temporización predeterminada para su suministro a la memoria RAM 311. La memoria RAM 311 efectúa la lectura de los datos memorizados en las direcciones de lectura para proporcionarlos a un dispositivo desplazador 312. El dispositivo desplazador 312 desplaza los datos de entrada en una magnitud correspondiente a una magnitud de desplazamiento suministrada por el circuito de generación de direcciones 315 y proporciona los datos desplazados a un convertidor de longitud de código inversa 313 y un convertidor de palabras de código inversas 314.

El convertidor de longitudes de código inversa 313 se suministra también con el nombre de una sintaxis (un nº de referencia) para un flujo con los parámetros de codificación dispuestos por el convertidor 202. Sobre la base de esta sintaxis, el convertidor de longitudes de código inversa 313 determina una longitud de código a partir de los datos de entrada (palabras de código) para proporcionarlos al circuito de generación de direcciones 315.

Además, el convertidor de palabras de código inverso 314 decodifica los datos suministrados por el dispositivo desplazador 312 sobre la base de la sintaxis y proporciona los datos decodificados al dispositivo de multiplexación de información de registro histórico 103.

El convertidor de palabras de código inversas 314 extrae también información requerida para determinar qué palabras de código están contenidas en los datos (es decir, la información requerida para determinar cómo se delimitan las palabras de código) y los proporciona al circuito de generación de direcciones 315. Sobre la base de esta información y la entrada de longitud de código por el convertidor de longitud de código inverso 313, el circuito de generación de direcciones 315 genera direcciones de escritura y de lectura para proporcionarlas a la memoria RAM 311 y genera una magnitud de desplazamiento para proporcionarlo al dispositivo desplazador 312.

La Figura 22 representa una realización, a modo de ejemplo, de una configuración del convertidor 212. En esta

realización, a modo de ejemplo, los datos de 8 bits son objeto de lectura desde una memoria intermedia 320 situada entre el VLC de registro histórico 211 y el convertidor 212, en las direcciones de lectura proporcionadas por un dispositivo controlador 326, y luego, se suministran a un circuito flip-flop D (D-FF) 321 para su retención. Los datos objeto de lectura desde el circuito flip-flop D 321 se suministran a un circuito de relleno 323 y a un circuito flip flop D de 8 bits 322 para su retención. Los datos de 8 bits objeto de lectura desde el circuito flip-flop D 322 y los datos de 8 bits objeto de lectura desde el circuito flip flop D 321 se sintetizan en datos paralelos de 16 bits, que luego se proporcionan al circuito de relleno 323.

El circuito de relleno 323 inserta el código "1" en los datos en una posición indica por una señal suministrada por el dispositivo controlador 326 y que indica la posición de relleno (stuffing) y proporciona los datos rellenos a un dispositivo desplazador 324 como datos de 17 bits.

El dispositivo desplazador 324 desplaza los datos de entrada sobre la base de una señal (shift) suministrada por el controlador 326 y que indica la magnitud del desplazamiento, y extrae datos de 8 bits para proporcionarlos a un circuito flip flop D de 8 bits 325. Los datos retenidos en el circuito flip flop D 325 son luego objeto de lectura y se entregan al siguiente dispositivo para dar formato a los datos de usuario 213 a través de una memoria intermedia 327. En este punto, el dispositivo controlador 326 genera direcciones de escritura con datos de salida para proporcionarlas a la memoria intermedia 327, que está interpuesta entre el convertidor 212 y el dispositivo para dar formato a los datos de usuario 213.

La Figura 23 representa una realización, a modo de ejemplo, de una configuración del circuito de relleno 323. La entrada de datos de 16 bits por los circuitos flip flop D 322, 321 se aplica a la entrada de un contacto a de cada conmutador 331-16 a 331-1. A un contacto c de un conmutador 331-i (i = 0 a 15) se suministra datos para un conmutador adyacente en un lado de MSB (el lado superior de la Figura). A modo de ejemplo, el contacto c del conmutador 331-12 se suministra con los 13<sup>o</sup> datos desde un lado LSB suministrado al contacto a del conmutador 331-13 adyacente en el lado MSB y el contacto c del conmutador 331-13 se suministra con los 14<sup>o</sup> datos procedentes del lado LSB suministrado al contacto a del conmutador 331-14 adyacente en el lado de MSB.

Sin embargo, el contacto a del conmutador 331-0 por debajo del conmutador 331-1 correspondiente al LSB está abierto. El contacto c del conmutador 331-16 correspondiente al MSB está también abierto porque no existe ningún conmutador más alto.

Un contacto b de cada conmutador 331-0 a 331-16 se suministra con los datos "1".

En respuesta a la posición de relleno de la señal que se suministra por el dispositivo controlador 326 y que indica la posición en la que se insertan los datos "1", un decodificador 332 conmuta uno de los conmutadores 331-0 a 331-16 al lado del contacto b, con lo que se conmuta el conmutador del lado LSB al lado del contacto c, mientras que se conmuta el conmutador del lado MSB al lado del contacto a.

La Figura 23 representa una realización, a modo de ejemplo, en donde los datos "1" se insertan en el 13<sup>o</sup> conmutador desde el lado LSB. Por lo tanto, en este caso, los conmutadores 331-0 a 331-12 son cada uno conmutados al lado del contacto c, el conmutador 331-13 se conmuta al lado del contacto b y los conmutadores 331-14 a 331-16 se conmutan cada uno al lado del contacto a.

Con la configuración anterior, el convertidor 212 en la Figura 22 convierte el código de 22 bits en 23 bits para la salida.

La Figura 24 representa temporizaciones para los datos de salida desde cada sección del convertidor 212 ilustrado en la Figura 22. Cuando el controlador 326 del convertidor 212 genera direcciones de lectura (Figura 24 (A)) en sincronismo con un reloj en bytes, los datos correspondientes son objeto de lectura desde la memoria intermedia 320 en bytes y se retienen en el circuito flip flop D 321. Los datos de lectura procedentes del circuito flip flop D 321 (Figura 24(B)) se suministran al circuito de relleno 323 y al circuito flip flop D 322 para su retención. Los datos retenidos en el circuito flip flop D 322 son objeto de nueva lectura (Figura 24(C)) y se entregan al circuito de relleno 323.

En consecuencia, los primeros datos de 1 byte D0 se aplican a la entrada del circuito de relleno 323 (Figura 24(D)) Con respecto a una temporización para una dirección de lectura A1, los datos de 2 bytes constituidos por los datos de 1 byte D0 y los datos de un byte D1 se aplican a la entrada con respecto a una temporización para la siguiente dirección de lectura A2 y los datos de 2 bytes constituidos por los datos de D1 y los datos D2 se aplican a la entrada con respecto a una temporización para una dirección de lectura A3.

El circuito de relleno 323 se suministra con la posición de relleno de señal (Figura 24 (E)) por el dispositivo controlador 326, que indica la posición en la que se insertan los datos "1". El decodificador 332 del circuito de relleno 323 conmuta al contacto b uno de los conmutadores 331-16 a 331-0 lo que corresponde a esta posición de relleno de señal, con lo que se conmuta el conmutador del lado LSB al lado del contacto c mientras que se conmuta el conmutador del lado MSB al lado del contacto a. De este modo, se insertan los datos "1", de modo que el circuito de



relleno 323 proporcione datos (Figura 24(F)) que indican que los datos "1" se han insertado en la posición indicada por la posición de relleno de señal.

5 El dispositivo desplazador 324 desplaza los datos de entrada en una magnitud indicada por el desplazamiento de señal (Figura 24(G)) que se suministra por el dispositivo controlador 326 y proporciona, a la salida, los datos desplazados (Figura 24(H)). Esta salida se retiene, además, en el circuito flip flop D 325 y luego, se proporciona al circuito siguiente (Figura 24 (I)).

10 La salida de datos por el circuito flip flop D 325 tiene los datos "1" insertados después de los datos de 22 bits. De este modo, existen 22 ceros continuos entre los datos "1" y los siguientes datos "1" aun cuando todos los bits entre estos datos sean cero.

15 La Figura 25 representa una realización, a modo de ejemplo, de una configuración del convertidor 202. La configuración del convertidor 202, que está configurado por un circuito flip flop D 341 para un dispositivo controlador 346, es esencialmente la misma que la del convertidor 212, que está constituido por el flip flop D 321 para el dispositivo controlador 326, pero difiere en que un circuito de supresión 343 está interpuesto en el convertidor 202 en lugar del circuito de relleno 323 en el convertidor 212. La otra parte de la configuración es la misma que la del convertidor 212 ilustrado en la Figura 22.

20 En el convertidor 202, el circuito de supresión 343 suprime un bit (los datos "1" insertados por el circuito de relleno 323 en la Figura 22) en conformidad con una posición de supresión de señal proporcionada por el dispositivo controlador 346 y que indica la posición de este bit.

25 Las otras operaciones son las mismas que en el convertidor 212 ilustrado en la Figura 22.

30 La Figura 26 es una realización, a modo de ejemplo, de una configuración del circuito de supresión 343. En esta realización ejemplo, los 15 bits del lado LSB de los 16 bits introducidos por los circuitos flip flops D 342, 341 se suministran cada uno a un contacto a de un conmutador correspondiente de los conmutadores 351-0 a 351-14. Un contacto b de cada conmutador se suministra con un bit que está más próximo al MSB que el bits suministrado al contacto a, en un bit. Un decodificador 352 suprime el bit indicado por la posición de supresión de señal que se suministra por el controlador 346 y proporciona los 15 bits restantes.

35 La Figura 26 representa cómo se suprimen el 13° bits desde el lado LSB. Por lo tanto, en este caso, los conmutadores 351-0 a 351-11 están cada uno conmutados al lado del contacto a y los 12 bits hasta el 12° bits desde el lado LSB se selecciona y proporcionan tal como están. Además, los conmutadores 351-12 a 351-14 están cada uno conmutados al lado del contacto b, en donde los 14° a 16° datos se seleccionan y proporcionan como los datos del 13° a 15° bits.

40 Las entradas al circuito de relleno 323 en la Figura 23 y al circuito de supresión 343 en la Figura 26 contienen cada una 16 bits porque los 16 bits suministrados por los circuitos flip flops D 322, 321 son objeto de entrada al circuito de relleno 323 del convertidor 212 en la Figura 22 y porque los 16 bits suministrados por los circuitos flip flops D 342, 341 son también objeto de entrada al circuito de supresión 343 del convertidor 202 en la Figura 25. En la Figura 22, los 17 bits proporcionados por el circuito de relleno 323 son desplazados por el dispositivo desplazador 324 para seleccionar finalmente y proporcionar, a modo de ejemplo, 8 bits. De forma similar, en el convertidor 202 ilustrado en la Figura 25, los 15 bits proporcionados por el circuito de supresión 343 son desplazados por el dispositivo desplazador 344 en una magnitud predeterminada para obtener datos de 8 bits.

45 La Figura 27 representa otra realización, a modo de ejemplo, de una configuración del convertidor 212. En esta realización ejemplo de una configuración, un contador 361 realiza el conteo del número de ceros continuos en los datos de entrada y proporciona un resultado del conteo al dispositivo controlador 326. Cuando el contador 361 efectúa el conteo, a modo de ejemplo, de 22 bits 0 continuos, el dispositivo controlador 326 proporciona la posición de relleno de señal al circuito de relleno 323. Al mismo tiempo, el dispositivo controlador 326 efectúa también una reposición del contador 361 para permitirle contar de nuevo 0 bits continuos.

50 La otra parte de la configuración y las otras operaciones son las mismas que se ilustraron en la Figura 22.

55 La Figura 28 representa otra realización ejemplo de una configuración del convertidor 202. En esta realización, a modo de ejemplo, de una configuración, un contador 371 efectúa el conteo del número de ceros continuos en los datos de entrada y proporciona un resultado de conteo al dispositivo controlador 346. Cuando el valor de conteo en el contador 371 alcanza el valor 22, el dispositivo controlador 346 proporciona la posición de supresión de señal al circuito de supresión 343 y también efectúa la reposición del contador 371 para permitirle contar, de nuevo, cero bits continuos. La otra parte de la configuración es la misma que la ilustrada en la Figura 25.

60 De esta manera, en este ejemplo de una configuración, los datos "1" se insertan y suprimen como un bit marcados sobre la base de un modelo predeterminado (el número de datos continuos "ceros").

65

Las configuraciones ilustradas en las Figuras 27 y 28 permiten un procesamiento más eficiente que las que se ilustran en las Figuras 22 y 25. La longitud de código después de la conversión, sin embargo, depende de la información de registro histórico origen.

5 La Figura 29 representa, a modo de ejemplo, de una configuración del dispositivo para dar formato a los datos de usuario 213. En esta realización ejemplo, cuando un dispositivo controlador 383 proporciona direcciones de lectura a una memoria intermedia (no ilustrada) situada entre el convertidor 212 y el dispositivo para dar formato a los datos de usuario 213, los datos objeto de lectura desde la memoria intermedia se suministran a un lado de contacto a de un conmutador 382 del dispositivo para dar formato a los datos de usuario 213. Una memoria ROM 381 memoriza datos tales como el código de inicio de datos de usuario y el identificador de datos que se requieren para generar datos de usuario `user_data()`. El dispositivo controlador 313 conmuta el conmutador 382 al lado del contacto a o a un lado de contacto b con respecto a una temporización predeterminada para seleccionar y proporcionar, cuando sea adecuado, los datos memorizados en la memoria ROM 381 o los datos suministrados por el convertidor 212. De este modo, los datos en el formato de `user_data()` se proporcionan al dispositivo de codificación 106.

15 Aunque no se ilustra, el decodificador de datos de usuario 201 puede ponerse en práctica proporcionando datos de entrada por intermedio de un conmutador para suprimir datos insertados que han sido objeto de lectura desde la memoria ROM 381 en la Figura 29.

20 La Figura 30 ilustra cómo una pluralidad de transcodificadores 101-1 a 101-N están conectados en serie para su uso en, a modo de ejemplo, un estudio de edición de vídeo. Un dispositivo de multiplexación de información de registro histórico 103-i de cada transcodificador 101-i ( $i = 1$  a  $N$ ) realiza la escritura de los más recientes parámetros de codificación que ha utilizado, en esa partición de una zona para los parámetros de codificación anteriormente descritos en donde se registran los más antiguos parámetros de codificación. A continuación, los datos de imágenes de banda base se han registrado inmediatamente próximos a las cuatro generaciones de parámetros de codificación (información de registro histórico de generaciones) en correspondencia con el mismo macrobloque (Figura 18).

25 En un dispositivo de codificación 106-i de cada dispositivo de codificación 106-i (Figura 19), el circuito de codificación de longitud variable 58 codifica datos de vídeo suministrados por el circuito de cuantización 57, sobre la base de los parámetros de codificación actuales entregados por un dispositivo de separación de información de registro histórico 105-i. Un flujo de bits así generado (a modo de ejemplo, `picture_header()`) tiene los parámetros de codificación en curso multiplexados.

30 El circuito de codificación de longitud variable 58 multiplexa también datos de usuario (incluyendo la información de registro histórico de generaciones) suministrada por un dispositivo de codificación de registro histórico 107-i, en un flujo de bits para la salida (los datos se multiplexan en el flujo de bits en lugar de incorporarse según se ilustra en la Figura 18). El flujo de bits proporcionado por el dispositivo de codificación 106-i se aplica a la entrada del transcodificador siguiente 101-( $i+1$ ) por intermedio de una SDTI (Interfaz de Transferencia de Datos en Serie).

35 Los transcodificadores 101-i y 101-( $i+1$ ) están cada uno configurados según se ilustra en la Figura 15. En consecuencia, el procesamiento por estos transcodificadores es el mismo que se describe con referencia a la Figura 15.

40 Para la codificación real utilizando registros históricos de los parámetros de codificación, si la imagen I en curso ha de cambiarse al tipo P o B, los registros históricos anteriores de los parámetros de codificación se examinan para buscar la imagen P o B anterior. Si existe dicho registro histórico, parámetros tales como sus vectores de movimiento se utilizan para cambiar el tipo de imagen. Por el contrario, si no se encuentra dicho registro histórico, se abandona la intención de cambiar el tipo de imagen sin detección de movimiento. Por supuesto, aun cuando no se encuentre dicho registro histórico, el tipo de imagen se puede cambiar mediante la detección de movimiento.

45 Aunque, en el formato ilustrado en la Figura 18, cuatro generaciones de parámetros de codificación se integran en los datos, los parámetros para cada uno de los tipos de imagen I, P, y B pueden integrarse de igual manera. La Figura 31 es un ejemplo de un formato en este caso. En esta realización ejemplo, cuando el mismo macrobloque ha sido codificado mientras se cambia el tipo de imagen, una generación de parámetros de codificación se registran para cada tipo de imagen (información de registro histórico de imágenes). En consecuencia, el dispositivo de decodificación 102 ilustrado en la Figura 16 y el dispositivo de codificación 106 ilustrado en la Figura 19 tienen a la entrada y salida una generación de parámetros de codificación correspondientes a las imágenes I, P y B en lugar de parámetros de codificación para la actual (más reciente), tercera, segunda y primera generaciones.

50 Además, en este caso, las zonas vacías  $Cb[1][x]$  y  $Cr[1][x]$  no se utilizan, de modo que la presente invención es aplicable a la imagen en un formato 4:2:0 libre de las zonas  $Cb[1][x]$  y  $Cr[1][x]$ .

55 En este caso, el dispositivo de decodificación 102 obtiene los parámetros de codificación simultáneamente con la decodificación, determina el tipo de imagen, realiza la escritura de los parámetros de codificación para una parte de la señal de imagen correspondiente al tipo de imagen (multiplexación) y proporciona, a la salida, la señal al dispositivo de separación de información de registro histórico 105. El dispositivo de separación de información de

registro histórico 105 puede separar los parámetros de codificación desde la señal para realizar una codificación mientras que el cambio del tipo de imagen toma en consideración el tipo de imagen deseado para la codificación y los parámetros de codificación anteriores de entrada.

5 A continuación, un proceso para permitir a cada transcodificador 101 determinar los posibles tipos de imágenes de destino se describirá haciendo referencia al diagrama de flujo ilustrado en la Figura 32. El cambio del tipo de imagen por el transcodificador 101 debe realizarse sin detección de movimiento porque utiliza vectores de movimiento anteriores. Además, el procesamiento descrito a continuación se ejecuta por el dispositivo de separación de información de registro histórico 105.

10 En la etapa S1, una generación de parámetros de codificación (información de registro histórico de imagen) se aplican a la entrada del dispositivo de separación de información de registro histórico 105 para cada tipo de imagen.

15 En la etapa S2, el dispositivo de separación de información de registro histórico 105 determina si la información de registro histórico de imágenes contiene, o no, parámetros de codificación utilizados para un cambio a la imagen B. Si el dispositivo de separación de información de registro histórico 105 determina que la información de registro histórico de imágenes contiene parámetros de codificación utilizados para un cambio a la imagen B, el proceso prosigue con la etapa S3.

20 En la etapa S3, el dispositivo de separación de información de registro histórico 105 determina si la información de registro histórico de imágenes contiene, o no, parámetros de codificación utilizados para un cambio a la imagen P. Si dispositivo de separación de información de registro histórico 105 determina que la información de registro histórico de imágenes contiene parámetros de codificación utilizados para un cambio a la imagen P, el proceso prosigue con la etapa S4.

25 En la etapa S4, el dispositivo de separación de información de registro histórico 105 determina que los posibles tipos de imágenes de destino son las imágenes I, P, y B.

Si se determina en la etapa S3 que información de registro histórico de imágenes no contiene parámetros de codificación utilizados para un cambio a la imagen P, el proceso prosigue con la etapa S5.

30 En la etapa S5, el dispositivo de separación de información de registro histórico 105 determina que los posibles tipos de imagen de destino son las imágenes I y B. Además, el dispositivo de separación de información de registro histórico 105 determina que puede utilizarse un procesamiento especial (solamente los vectores de predicción directa contenidos en la información de registro histórico de la imagen B se utilizan sin los vectores de predicción inversa) para cambiar a la imagen P en un pseudo-modo operativo.

Si se determina en la etapa S2 que la información de registro histórico de imagen no contiene parámetros de codificación utilizados para un cambio a la imagen B, el proceso prosigue con la etapa S6.

40 En la etapa S6, el dispositivo de separación de información de registro histórico 105 determina si la información de registro histórico de imágenes contiene, o no, parámetros de codificación utilizados para un cambio a la imagen P. Si el dispositivo de separación de información de registro histórico 105 determina que la información de registro histórico de imágenes contiene parámetros de codificación utilizados para un cambio a la imagen P, el proceso prosigue con la etapa S7.

45 En la etapa S7, el dispositivo de separación de información de registro histórico 105 determina que los posibles tipos de imágenes de destino son las imágenes I y P. Además, el dispositivo de separación de información de registro histórico 105 determina que puede utilizarse un procesamiento especial (solamente los vectores de predicción directa contenidos en la información de registro histórico de imágenes P se utilizan) para cambiar a la imagen B en un pseudo-modo operativo.

50 Si se determina en la etapa S6 que la información de registro histórico de imágenes no contiene parámetros de codificación utilizados para un cambio en la P, el proceso prosigue con la etapa S8. En la etapa S8, el dispositivo de separación de información de registro histórico 105 determina que solamente el tipo de imagen de destino posible es la imagen I porque ningún vector de movimiento está presente (puesto que se trata de la imagen I, que solamente puede cambiarse a la propia imagen I).

55 Después del procesamiento en las etapas S4, S5, S7, y S8 en la etapa S9, el dispositivo de separación de información de registro histórico 105 visualiza los posibles tipos de imagen de destino en el dispositivo de presentación visual (no ilustrado) para notificación al usuario.

60 La Figura 33 ilustra un ejemplo de cambio de tipo de imagen. Para cambiar el tipo de imagen, el número de tramas que constituyen el grupo GOP es objeto de cambio. Es decir, en este caso, un grupo GOP largo de 4 Mbps (primera generación) constituido por tramas de  $N = 15$  (número de tramas GOP  $N = 15$ ) y  $M = 3$  (ciclo de ocurrencia de imágenes I o P dentro del GOP  $M = 3$ ), se convierte en un GOP corto de 50 Mbps (segunda generación) constituido por tramas de  $N = 1$  y  $M = 1$ , que se convierte de nuevo en un grupo GOP largo de 4 Mbps (tercera generación)

constituido por tramas de  $N = 15$  y  $M = 3$ . Las líneas de trazos de la Figura indican los límites en el grupo GOP.

Si el tipo de imagen se cambia desde la primera generación a la segunda generación, todas las tramas pueden tener sus tipos de imágenes cambiados a la imagen I, como es evidente a partir de la descripción anterior del posible proceso de determinación de posibles tipos de imagen de destino. Durante este cambio de tipo de imagen, todos los vectores de movimiento calculados para convertir una imagen de movimiento (0ª generación) en la primera generación se memorizan (permanecen) en la información de registro histórico de imágenes. A continuación, si una conversión en un grupo GOP largo (el tipo de imagen se cambia desde la segunda generación a la tercera generación) se ejecuta de nuevo, puesto que los vectores de movimiento para cada tipo de imagen que han sido utilizados para convertir la 0ª generación en la primera generación están memorizados, estos vectores pueden reutilizarse para ejecutar de nuevo una conversión en un grupo GOP largo al mismo tiempo que se restringe la degradación de la calidad de imagen.

La Figura 34 ilustra otra realización ejemplo del cambio de tipos de imagen. En este caso, un grupo GOP largo de 4 Mbps (primera generación) de  $N = 14$  y  $M = 2$  se convierte en un grupo GOP corto de 18 Mbps (segunda generación) de  $N = 2$  y  $M = 2$ , que se convierte en un grupo GOP corto de 50 Mbps (tercera generación) de  $N = 1$  y  $M = 1$ , es decir, que tiene una sola trama, que se convierte, además, en un grupo GOP aleatorio de 1 Mbps (cuarta generación) que tiene  $N$  tramas.

En este caso, los vectores de movimiento para cada tipo de imagen que se utiliza para la conversión desde la 0ª generación a la primera generación son memorizados hasta la conversión desde la tercera generación a la cuarta generación. A continuación, según se ilustra en la Figura 34, a pesar de un cambio complicado en el tipo de imagen, la degradación de la calidad de la imagen se puede minimizar reutilizando los parámetros de codificación memorizados. Además, utilizando efectivamente la escala de cuantización contenida en los parámetros de codificación memorizados, la codificación puede realizarse mientras se restringe la degradación de la calidad de la imagen.

La reutilización de la escala de cuantización se describirá haciendo referencia a la Figura 35. La Figura 35 ilustra que una trama predeterminada se convierte constantemente en la imagen I desde la primera generación a la cuarta generación, mientras que solamente la tasa binaria se cambia a 4, 18, o 50 Mbps.

A modo de ejemplo, al convertir la primera generación (4 Mbps) en la segunda generación (18 Mbps), la calidad de imagen no se mejora incluso utilizando una escala de cuantización más fina para la codificación en respuesta al aumento en la tasa binaria. Esto es así porque los datos anteriormente objeto de cuantización con una escala de cuantización menos fina no se restablecen. En consecuencia, utilizando una escala de cuantización más fina para la codificación en respuesta al aumento en la tasa binaria que se ilustra en la Figura 35 simplemente da lugar a un aumento en la información pero no mejora la calidad de la imagen. En consecuencia, se permite una codificación más eficiente proporcionando dicho control que mantiene la escala de cuantización anteriormente menos fina (mayor).

Al cambiar la tercera generación a la cuarta generación, la tasa binaria disminuye desde 50 a 4 Mbps pero, en este caso, la escala de cuantización anteriormente menos fina (mayor) se mantiene también a este respecto.

Según se describió con anterioridad, al cambiar la tasa binaria, es muy efectivo utilizar un registro histórico anterior de la escala de cuantización para la codificación.

Este proceso de control de cuantización se explicará haciendo referencia al diagrama de flujo ilustrado en la Figura 36. En la etapa S11, el dispositivo de separación de información de registro histórico 105 determina si la información de registro histórico de imágenes de la entrada contiene, o no, parámetros de codificación para un tipo de imagen a convertirse. Si el dispositivo de separación de información de registro histórico 105 determina que la información de registro histórico de imágenes de entrada contiene parámetros de codificación para el tipo de imagen objeto de conversión, el proceso prosigue con la etapa S12.

En la etapa S12, el dispositivo de separación de información de registro histórico 105 extrae el código `history_q_scale_code` a partir de los parámetros de codificación pertinentes en la información de registro histórico de imágenes.

En la etapa S13, el dispositivo de separación de información de registro histórico 105 calcula el código `feedback_q_scale_code` sobre la base de la capacidad de memoria intermedia remanente que se realimenta desde la memoria intermedia de transmisión 59 al circuito de cuantización 57.

En la etapa S14, el dispositivo de separación de información de registro histórico 105 determina si, o no, el código `history_q_scale_code` es mayor (menos fino) que el código `feedback_q_scale_code`. Si el dispositivo de separación de información de registro histórico 105 determina que el código `history_q_scale_code` es mayor que el código `feedback_q_scale_code`, el proceso prosigue con la etapa S15.

En la etapa S15, el dispositivo de separación de información de registro histórico 105 proporciona el código `history_q_scale_code` al circuito de cuantización 57 como una escala de cuantización. El circuito de cuantización 57 utiliza el código `history_q_scale_code` para realizar la cuantización.

5 En la etapa S16, se determina si todos los macrobloques contenidos en la trama han sido objeto de cuantización. Si se determina que no todos los macrobloques han sido todavía objeto de cuantización, el proceso retorna a la etapa S12 para repetir el procesamiento desde la etapa S12 a la etapa S16 hasta que todos los macrobloques hayan sido objeto de cuantización.

10 Si se determina en la etapa S14 que el código `history_q_scale_code` no es mayor (más fino) que el código `feedback_q_scale_code`, el proceso prosigue con la etapa S17.

15 En la etapa S17, el dispositivo de separación de información de registro histórico 105 proporciona el código `feedback_q_scale_code` al circuito de cuantización 57 como una escala de cuantización. El circuito de cuantización 57 utiliza el código `feedback_q_scale_code` para realizar la cuantización.

Si se determina en la etapa S11 que la información de registro histórico no contiene parámetros de codificación para el tipo de imagen convertido, el proceso prosigue con la etapa S18.

20 En la etapa S18, el dispositivo de separación de información de registro histórico 105 calcula el código `feedback_q_scale_code` sobre la base de la cantidad de memorización restante realimentada desde la memoria intermedia de transmisión 59 al circuito de cuantización 57.

25 En la etapa S19, el circuito de cuantización 57 utiliza el código `feedback_q_scale_code` para realizar la cuantización.

30 En la etapa S20, se determina si, o no, todos los macrobloques contenidos en la trama han sido objeto de cuantización. Si se determina que no todos los macrobloques contenidos en la trama han sido objeto de cuantización, el proceso retorna a la etapa S18 para repetir el procesamiento de la etapa S18 a la etapa S20 hasta que todos los macrobloques hayan sido objeto de cuantización.

35 En el interior del transcodificador 101 en conformidad con esta forma de realización, los lados de decodificación y de codificación tienen un acoplamiento débil juntos y los parámetros de codificación se multiplexan en los datos de imagen antes de la transmisión, según se describió con anterioridad. Según se ilustra en la Figura 37, sin embargo, el dispositivo de decodificación 102 y el dispositivo de codificación 106 pueden estar directamente conectados juntos (en fuerte acoplamiento).

40 El transcodificador 101 según se describe en la Figura 15 multiplexa los parámetros de codificación anteriores en los datos de vídeo de banda base antes de la transmisión con el fin de suministrar al dispositivo de codificación 106 los parámetros de codificación anteriores para la primera, segunda y tercera generaciones. Sin embargo, la multiplexación de los parámetros de codificación anteriores en los datos de vídeo de banda base no es esencial para la presente invención, pero una ruta de transmisión (a modo de ejemplo, el bus de transferencia de datos) diferente de la que tienen los datos de vídeo de banda base puede utilizarse para transmitir los parámetros de codificación anteriores, según se ilustra en la Figura 37.

45 Es decir, el dispositivo de registro histórico 102, el dispositivo de decodificación de registro histórico 104, el dispositivo de codificación 106 y el dispositivo de codificación de registro histórico 107 que se ilustran en la Figura 37, tienen exactamente las mismas funciones y configuraciones que el dispositivo de decodificación 102, el dispositivo de decodificación de registro histórico 104, el dispositivo de codificación 106 y el dispositivo de codificación de registro histórico 107 descritos en la Figura 15.

50 El circuito de decodificación de longitud variable 112 del dispositivo de decodificación 102 extrae los parámetros de codificación de la tercera generación desde la capa de secuencias, capa de GOP, capa de imágenes, capa de segmentos y capa de macrobloques del flujo codificado de la tercera generación ST(3rd) y suministra estos parámetros al dispositivo controlador 70 para el dispositivo de codificación de registro histórico 107 y el dispositivo de codificación 106. El dispositivo de codificación de registro histórico 107 convierte los parámetros de codificación recibidos en el flujo denominado `converted_history_stream()` de modo que los parámetros puedan describirse en la zona de datos de usuario de la capa de imágenes y suministra el flujo `converted_history_stream()` al circuito de codificación de longitud variable 58 del dispositivo de codificación 106 como datos de usuarios.

60 Además, el dispositivo de decodificación de longitud variable 112 extrae los datos de usuarios `user_data` que contiene los parámetros de codificación de la primera y de la segunda generación, desde la zona de datos de usuario de la capa de imagen del flujo codificado de la tercera generación y suministra estos datos al dispositivo de decodificación de registro histórico 104 y al dispositivo de codificación de longitud variable 58 del dispositivo de codificación 106. El dispositivo de decodificación de registro histórico 104 extrae los parámetros de codificación de la primera y de la segunda generación desde el flujo de registro histórico descrito en la zona de datos de usuario como `converted_history_stream()`, y suministra estos parámetros al dispositivo controlador para el dispositivo de

codificación 106.

5 El dispositivo controlador 70 para el dispositivo de codificación 106 controla el proceso de codificación mediante el dispositivo de codificación 106 sobre la base de los parámetros de codificación de la primera y de la segunda generación recibidos desde el dispositivo de decodificación de registro histórico 104 y los parámetros de codificación de la tercera generación recibidos desde el dispositivo de codificación 102.

10 El circuito de codificación de longitud variable 58 del dispositivo de codificación 106 recibe desde el dispositivo de decodificación 102 los datos de usuarios `user_data` que contienen los parámetros de codificación de la primera y de la segunda generación, mientras que se recibe desde el dispositivo de codificación de registro histórico 107 los datos de usuarios `user_data` que contienen los parámetros de codificación de la tercera generación y describe estos datos de usuarios en la zona de datos de usuario de la capa de imágenes del flujo codificado de la cuarta generación como información de registro histórico.

15 La Figura 38 representa una sintaxis para uso en la decodificación de flujos de vídeo de MPEG. El decodificador decodifica un flujo de bits de MPEG en conformidad con esta sintaxis para extraer una pluralidad de elementos de datos significativos desde el flujo de bits. En la sintaxis ilustrada, que se describirá a continuación, las funciones y las declaraciones condicionales se representan por caracteres delgados, mientras que los elementos de datos se representan por caracteres gruesos. Los elementos de datos se describen cada uno en forma nemónica que indica su nombre, longitud de bits, tipo y orden de transmisión.

En primer lugar, las funciones utilizadas en la sintaxis que se ilustran en la Figura 38 serán objeto de explicación.

25 Una función de `next_start_code()` busca un código de inicio descrito en flujo de bits. En la sintaxis ilustrada en la Figura 38, una función de cabecera de secuencia `sequence_header()` y una función de extensión de secuencia `sequence_extension()` se disponen de forma secuencial después de la función del código de inicio siguiente `next_start_code()`, de modo que se describe en este flujo de bits a elementos de datos definidos por las funciones de `sequence_header()` y funciones `sequence_extension()`. En consecuencia, durante la decodificación del flujo de bits, la función `next_start_code()` encuentra códigos de inicio (una clase de elementos de datos) en el flujo de bits, que se describen en las cabeceras de las funciones `sequence_header()` y `sequence_extension()` y utiliza estos códigos de inicio como referencias para ilustrar las funciones de `sequence_header()` y `sequence_extension()`. La función `next_start_code()` decodifica luego los elementos de datos definidos por las funciones de `sequence_header()` y `sequence_extension()`.

35 La función de `sequence_header` define datos de cabeceras para la capa de secuencias del flujo de bits MPEG y la función `sequence_extension()` define datos de extensión para la capa de secuencias del flujo de bits MPEG.

40 Una sintaxis de `do{}while` después de la función `sequence_extension()` extrae del flujo de datos los elementos de datos descritos sobre la base de una función en las `{ }` de la `do` statement mientras que una condición definida por la `while` statement es verdadera. Es decir, la `do{}while` statement ejecuta un proceso de decodificación para la extracción desde el flujo de datos de los elementos de datos descritos sobre la base de la función en `do` statement, mientras que la condición definida por la `while` statement es verdadera.

45 Una función `nexbits()` utilizada en la declaración `while` statement compara bits o una cadena de bits que aparece en el flujo de bits con los siguientes elementos de datos a decodificar. En el ejemplo de sintaxis ilustrado en la Figura 38, la función `nexbits()` compara la cadena de bits en el flujo de bits con el código `sequence_end_code` indicativo del final de la secuencia de vídeo y cuando la cadena de bits en el flujo de bits y el código `sequence_end_code` no coinciden, la condición indicada por la declaración `while` statement es verdadera. En consecuencia, la sintaxis `do{}while` que sigue a la función de `sequence_extension()` indica que los elementos de datos definidos por la función en la declaración `do` statement se describe en el flujo de bits a no ser que aparezca el código `sequence_end_code` indicativo del final de la secuencia de vídeo.

55 En el flujo de bits, los elementos de datos definidos por la función `sequence_extension()` están seguidos por elementos de datos definidos por la función `extension_and_user_data(0)`. La función `extension_and_user_data(0)` define los datos de extensión y de usuarios para la capa de secuencias del flujo de bits MPEG.

60 Una sintaxis `do{}while` que sigue a la función `extension_and_user_data(0)` extrae desde el flujo de bits los elementos de datos descritos sobre la base de una función en las `{ }` de la declaración `do` statement, mientras que sea verdadera una condición definida por la declaración `while` statement. Una función `nexbits()` utilizada en la declaración `while` statement determina si los bits o una cadena de bits que aparece en el flujo de bits coincide con el código `picture_start_code` o el código `group_start_code`. Si los bits o la cadena de bits que aparece en el flujo de bits coinciden con el código `picture_start_code` o el código `group_start_code`, la condición definida por la declaración `while` statement es verdadera. De este modo, cuando el código `picture_start_code` o el código `group_start_code` aparece en el flujo de bits, puesto que este código de inicio describe códigos para los elementos de datos definidos por la función en la declaración `do` statement, la sintaxis `do{}while` puede buscar el código de inicio indicado por el código `picture_start_code` o el código `group_start_code` para extraer desde el flujo de bits los elementos de datos

definidos en la declaración do statement.

La declaración if statement describa al principio de la declaración do statement indica la condición de que el código group\_start\_code aparezca en el flujo de bits. Si la condición indicada por esta declaración if statement es verdadera, entonces en la descripción de este flujo de bits, el código group\_start\_code está seguido por los elementos de datos definidos por una función de group\_of\_picture\_header(1) y una función de extension\_and\_user\_data(1).

La función de group\_of\_picture\_header(1) define datos de cabeceras para la capa GOP del flujo de datos MPEG.

La función de extension\_and\_user\_data(1) define datos de extensión (extension\_data) y los datos de usuarios (user\_data) para la capa GOP del flujo de bits MPEG.

Además, en la descripción de este flujo de bits, los elementos de datos definidos por las funciones group\_of\_picture\_header(1) y extension\_and\_user\_data(1) están seguidos por los elementos de datos definidos por una función picture\_header() y una función picture\_coding\_extension(). Por supuesto, si la condición indicada por la declaración if statement anteriormente descrita no es verdadera, los elementos de datos definidos por las funciones de group\_of\_picture\_header(1) y extension\_and\_user\_data(1) no se describen, de modo que los elementos de datos definidos por las funciones extension\_and\_user\_data(0), están seguidos por los elementos de datos definidos por las funciones de picture\_header() y picture\_coding\_extension().

La función de picture\_header () define datos de cabecera para la capa de imágenes del flujo de bits MPEG.

La función picture\_coding\_extension() define los primeros datos de extensión para la capa de imágenes del flujo de bits MPEG.

La siguiente declaración while statement determina si una condición indicada por la siguiente declaración if statement es o no verdadera, mientras que una condición definida por estas declaración while statement es verdadera. Una función nextbits() utilizada en esta declaración while statement determina si una cadena de bits que aparece en el flujo de bits coincide con el código extension\_start\_code o user\_data\_start\_code. Si la cadena de bits que aparece en el flujo de bits coincide con el código extension\_start\_code o el código user\_data\_start\_code, la condición definida por esta declaración while statement es verdadera.

Una primera declaración if statement determina si, o no, la cadena de bits que aparece en el flujo de bits coincide con el código extension\_start\_code. Si la cadena de bits que aparece en el flujo de bits coincide con el código extension\_start\_code de 32 bits, entonces en la descripción del flujo de bits, el código extension\_start\_code está seguido por los elementos de datos definidos por la función extension\_data(2).

Una segunda declara if statement determina si, o no, la cadena de bits que aparece en el flujo de bits coincide con el código user\_data\_start\_code. Si la cadena de bits que aparece en el flujo de bits coincide con el código user\_data\_start\_code de 32 bits, se determina si una condición indicada por una tercera sentencia if statement es o no verdadera. El código user\_data\_start\_code es un código de inicio que indica el comienzo de la zona de datos de usuario de la capa de imágenes del flujo de bits MPEG.

La tercera declaración if statement es una sintaxis para determinar si, o no, una cadena de bits que aparece en el flujo de bits coincide con el identificador History\_Data\_ID. Si la cadena de bits que aparece en el flujo de bits coincide con el identificador History\_Data\_ID de 32 bits, entonces en la descripción de la zona de datos de usuario de la capa de imágenes de este flujo de bits MPEG, un código indicado por este History\_Data\_ID de 32 bits está seguido por elementos de datos definidos por la función converted\_history\_stream().

La función de converted\_history\_stream() describe información de registro histórico y datos para transmitir todos los parámetros de codificación utilizados para la codificación MPEG. Los detalles de elementos de datos definidos por esta función converted\_history\_stream() se examinarán a continuación haciendo referencia a las Figuras 40 a 47 como history\_stream(). Además, el identificador History\_Data\_ID es un código de inicio indicativo de la cabecera de la información de registro histórico y los datos descritos en la zona de datos de usuario de la capa de imágenes del flujo de bits MPEG.

Una declaración else statement es una sintaxis que indica que la condición indicada por la tercera declaración if statement no es verdadera. En consecuencia, si la descripción de la zona de datos de usuarios de la capa de imágenes de este flujo de bits de MPEG no contiene los elementos de datos definidos por la función converted\_history\_stream(), contiene los elementos de datos definidos por la función user\_data().

En la Figura 38, la información de registro histórico se describe en la función converted\_history\_stream() y no en la user\_data(), mientras que el converted\_history\_stream() se describe como una clase de datos de usuarios en conformidad con la norma MPEG. De este modo, aunque la especificación pueda declarar que la información de registro histórico se describe en los datos de usuarios user\_data, esto significa que la información se describe como

una clase de datos de usuarios en conformidad con la norma MPEG.

La función `picture_data ()` describe elementos de datos para las capas de segmentos y de macrobloques después de los datos de usuarios para la capa de imágenes del flujo de bits MPEG. Los elementos de datos indicados por la función `picture_data()` se suelen describir de los elementos de datos definidos por la función `converted_history_stream ()` descrita en la zona de datos de usuario de la capa de imágenes del flujo o por la función `user_data()`. Si, sin embargo, un flujo de bits que indica los elementos de datos de la capa de imágenes no contiene el código `extension_start_code` o el código `user_data_start_code`, los elementos de datos indicados por la función `picture_data ()` se describen después de los elementos de datos definidos por la función `picture_coding_extension()`.

Los elementos de datos indicados por la función `picture_data()` están seguidos por los elementos de datos definidos por las funciones `sequence_header()` y `sequence_extension ()`. Los elementos de datos definidos por las funciones `sequence_header()` y `sequence_extension()` son idénticas para los elementos de datos descritos por las funciones `sequence_header()` y `sequence_extension ()` en la cabecera de la secuencia de flujo de vídeo. Los datos idénticos se describen en el flujo para impedir el error siguiente: si un dispositivo de recepción de flujos de bits comienza a recibir la parte intermedia del flujo de datos (a modo de ejemplo, una parte del flujo de bits correspondiente a la capa de imágenes), no puede recibir los datos para la capa de secuencias, lo que da lugar a un fallo en la decodificación del flujo.

Después de los elementos de datos definidos por las funciones finales `sequence_header()` y `sequence_extension()`, es decir, al final del flujo de datos, se describe el código `sequence_end_code` de 32 bits indicativo del final de la secuencia.

Una configuración básica para la sintaxis anterior se ilustra esquemáticamente en la Figura 39.

A continuación, el flujo de registro histórico definido por la función `converted_history_stream()` será objeto de explicación.

El flujo `converted_history_stream()` es una función para insertar el flujo de registro histórico indicativo de la información de registro histórico en la zona de datos de usuario de la capa de imágenes del flujo MPEG. El término "converted" significa que se trata de un flujo que se ha sometido a un proceso de conversión para insertar un bit marcador (1 bit) en un flujo de registro histórico a intervalos de al menos 22 bits con el fin de evitar la emulación del inicio, estando constituido el flujo de registro histórico por datos de registro histórico a insertarse en la zona del usuario.

El `converted_history_stream()` se describe en la forma de un flujo de registro histórico de longitud fija (Figuras 40 a 46) o un flujo de registro histórico de longitud variable (Figura 47), que se describirá a continuación. Si el codificador selecciona el flujo de registro histórico de longitud fija, el decodificador puede utilizar ventajosamente circuitos simples y programas informáticos para decodificar cada elemento de datos desde el flujo de registro histórico. Por el contrario, si el codificador selecciona el flujo de registro histórico de longitud variable, puede seleccionar arbitrariamente la información de registro histórico (elemento de datos) que se describe en la zona de usuarios de la capa de imágenes para reducir la cantidad de datos de flujos de registro histórico y de este modo, la tasa de datos del flujo codificado global.

Los términos "flujo de registro histórico", "información de registro histórico", "datos de registro histórico" y "parámetros de registro histórico", según aquí se describen, significan los parámetros de codificación (o elementos de datos) utilizados para los procesos de codificación anteriores y no los parámetros de codificación utilizados para el proceso de codificación (final) en curso. A continuación se describirá una realización ejemplo en donde durante el proceso de codificación de la primera generación, se codifica una imagen en el tipo I antes de la transmisión, en donde durante el proceso de codificación de la segunda generación, esta imagen I se codifica en el tipo P antes de la transmisión y en donde durante el proceso de codificación de la tercera generación, esta imagen P se codifica además, en el tipo B antes de la transmisión.

Los parámetros de codificación utilizados para el proceso de codificación de la tercera generación se describen en posiciones predeterminadas en las capas de secuencias, GOP, imágenes, segmentos y macrobloques del flujo de bits codificado generado durante el proceso de codificación de la tercera generación. Por el contrario, los parámetros de codificación utilizados para los procesos de codificación de la primera generación y de la segunda generación, los procesos de codificación anteriores, no se describen en la capa de secuencias o de GOP en donde se describen los parámetros de codificación utilizados para el proceso de codificación de la tercera generación pero en la zona de datos de usuarios de la capa de imágenes como información de registro histórico para codificar parámetros en conformidad con la sintaxis anteriormente descrita.

En primer lugar, la sintaxis del flujo de registro histórico de longitud fija se describirá con referencia a las Figuras 40 a 46.

Los parámetros de codificación contenidos en la cabecera de secuencias de la capa de secuencias que se utilizan



- 5 para el proceso de codificación anterior (a modo de ejemplo, el proceso de codificación de la primera generación o de la segunda generación) se insertan en primer lugar, como un flujo de registro histórico, en la zona de datos de usuario de la capa de imágenes de un flujo de bits generado durante el proceso de codificación final (a modo de ejemplo el proceso de codificación de la tercera generación). Conviene señalar que la información de registro histórico, tal como la cabecera de secuencias de la capa de secuencias de un flujo de bits que se genera durante el proceso de codificación anterior, no se inserta en la cabecera de secuencia de la capa de secuencias del flujo de bits que se genera durante el proceso de codificación final.
- 10 Los elementos de datos contenidos en la cabecera de secuencia (`sequence_header`) que se utiliza para el proceso de codificación anterior están constituidos por los parámetros siguientes `sequence_header_code`, `sequence_header_present_flag`, `horizontal_size_value`, `marker_bit`, `vertical_size_value`, `aspect_ratio_information`, `frame_rate_code`, `bit_rate_value`, `VBV_buffer_size_value`, `constrained_parameter_flag`, `load_intra_quantiser_matrix`, `load_non_intra_quantizer_matrix`, `intra_quantiser_matrix` y `non_intra_quantiser_matrix`, etc.
- 15 El código `sequence_header_code` son datos que representan un código de sincronización de inicio para la capa de secuencias. El indicador `sequence_header_present_flag` son datos que indican si son válidos, o no, los datos en la cabecera de secuencia. El valor `horizontal_size_value` son datos que consisten en los 12 bits inferiores del número de pixels en una dirección horizontal de una imagen. El `marker_bit` son datos de bits insertados para emulación del código de inicio. El valor `vertical_size_value` son datos que consisten en los 12 bits inferiores de número de líneas verticales en una imagen. La `aspect_ratio_information` son datos que representan una relación de aspecto de imagen o una relación de aspecto de pantalla de presentación visual. El código `frame_rate_code` son datos que representan un ciclo de presentación visual de imágenes.
- 20 El `bit_rate_value` son datos de los 18 bits inferiores (en forma aproximada, utilizando 400 bsp como una unidad) de la tasa binaria para limitar la cantidad de bits generados. El valor `VBV_buffer_size_value` son datos de 10 bits inferiores de un valor que determina la capacidad de una memoria intermedia virtual (un verificador de memoria de vídeo) para controlar la cantidad de códigos generados. El indicador `constrained_parameter_flag` son datos que indican cada parámetro está por debajo del límite. La `load_intra_quantiser_matrix` son datos que indican la presencia de datos de matriz de cuantización intra-MB. La `load_non_intra_quantiser_matrix` son datos que indican la presencia de datos de matrices de cuantización non\_intra-MB. La `intra_quantiser_matrix` son datos que indican un valor para los datos de matriz de cuantización intra-MB. La `non_intra_quantiser_matrix` son datos que indican un valor para datos de matriz de cuantización no-intra-MB
- 25 Después de estos elementos de datos, se describen elementos de datos que representan una etiqueta de secuencia de la capa de secuencias que se utiliza para el proceso de codificación anterior, como un flujo de registro histórico, en la zona de datos de usuario de la capa de imágenes de un flujo de bits generado durante el proceso de codificación final.
- 30 Los elementos de datos que representan la extensión de secuencia (`sequence_extension`) utilizados para el proceso de codificación anterior son `extension_start_code`, `extension_start_code_identifier`, `sequence_extension_present_flag`, `profile_and_level_indication`, `progresive_sequence`, `chroma_format`, `horizontal_size_extension`, `vertical_size_extension`, `bit_rate_extension`, `vbv_buffer_size_extension`, `low_delay`, `frame_rate_extension_n`, `frame_rate_extension_d`, etc.
- 35 El `extension_start_code` son datos que representan un código de sincronización de inicio para los datos de extensión. El `extension_start_code_identifier` son datos que indican qué datos de extensión se envían. El indicador `sequence_extension_present_flag` son datos que indican si son válidos, o no, los datos en la extensión de secuencia. La `profile_and_level_indication` son datos que especifican el perfil y el nivel de datos de vídeo. La `progresive_sequence` son datos que indican que los datos de vídeo están escaneados de forma secuencial. El `chroma_format` son datos que especifican el formato de aberración cromática de los datos de vídeo.
- 40 La `horizontal_size_extension` son datos de 2 bits superiores añadidos a `horizontal_size_value` de la cabecera de secuencia. La `vertical_size_extension` son datos de 2 bits superiores al valor `vertical_size_value` de la cabecera de secuencia. La `bit_rate_extension` son datos de 12 bits superiores añadidos al valor de la tasa binaria `bit_rate_value` de la cabecera de secuencia. La `vbv_buffer_size_extension` son datos de 8 bits superiores añadidos al valor `vbv_buffer_size_value` de la cabecera de secuencia. El `low_delay` son datos indicativos de la ausencia de la imagen B. La `frame_rate_extension_n` son datos que se combinan con el código `frame_rate_code` de la cabecera de secuencia para obtener una tasa de tramas. La `frame_rate_extension_d` son datos que se combinan con el código `frame_rate_code` de la cabecera de secuencia para obtener la tasa de tramas.
- 45 Después de estos elementos de datos, se describen elementos de datos que representan una etiqueta de presentación visual de secuencia de la capa de secuencias que se utiliza para el proceso de codificación anterior, como un flujo de registro histórico, en la zona de usuarios de la capa de imágenes del flujo de bits.
- 50 Los elementos de datos descritos como la extensión de la presentación visual de secuencias (`sequence_display_extension`) están constituidos por el código `extension_start_code` y el identificador
- 55
- 60
- 65

extension\_start\_code\_identifier, sequence\_display\_extension\_present\_flag, video\_format, colour\_description, colour\_primaries, transfer\_characteristics, matrix\_coefficients, display\_horizontal\_size y display\_vertical\_size.

5 El código extension\_start\_code son datos que representan un código de sincronización de inicio para los datos de extensión. El indicador extension\_start\_code\_identifier es un código que indica qué datos de extensión se envían. El sequence\_display\_extension\_present\_flag son datos que indican si, o no, los elementos de datos en la extensión de presentación visual de secuencia son válidos. El video\_format son datos que representan el formato de vídeo de una señal origen. La colour\_description son datos indicativos de la presencia de datos detallados en un espacio de color. Los colour\_primaries son datos indicativos de los detalles de las características cromáticas de la señal origen. La transfer\_characteristics son datos que indican en detalle cómo se ha realizado la conversión fotoeléctrica. Los matrix\_coefficients son datos que indican en detalle cómo los colores primarios se han convertido en la señal origen. El display\_horizontal\_size son datos que representan una zona activa (magnitud horizontal) de una presentación visual prevista. display\_vertical\_size son datos que representan una zona activa (magnitud vertical) de una presentación visual prevista.

15 Después de estos elementos de datos, se describen datos de asignación de macrobloques (macroblock\_assignment\_in\_user\_data) que indican información de fase para un macrobloque generado durante el proceso de codificación anterior, como un flujo de registro histórico, en la zona de usuarios de la capa de imágenes del flujo de bits que se genera durante el proceso de codificación final.

20 Los macroblock\_assignment\_in\_user\_data que indican la información de fase para el macrobloque están constituidos por elementos de datos tales como macroblock\_assignment\_present\_flag, v\_phase, y h\_phase.

25 El indicador macroblock\_assignment\_present\_flag son datos que indican si los elementos de datos en macroblock\_assignment\_in\_user\_data son válidos o no. v\_phase son datos que indican información de fase vertical para uso en el corte de un macrobloque a partir de los datos de imágenes. La h\_phase son datos que indican información de fase horizontal para uso en el corte de un macrobloque a partir de los datos de imágenes.

30 Después de estos elementos de datos, se describen elementos de datos que representan la cabecera GOP de la capa GOP utilizada para el proceso de codificación anterior, como un flujo de registro histórico, en la zona de usuarios de la capa de imágenes del flujo de bits que se genera durante el proceso de codificación final.

35 Los elementos de datos que representan la cabecera GOP (group\_of\_picture\_header) están constituidos por group\_start\_code, group\_of\_picture\_header\_present\_flag, time\_code, closed\_gop y broken\_link.

40 El código group\_start\_code son datos que indican un código de sincronización de inicio para la capa GOP. El indicador group\_of\_picture\_header\_present\_flag son datos que indican si los elementos de datos en el grupo de cabeceras de imágenes son válidos o no. El time\_code es un código temporal que indica el tiempo desde la cabecera de la secuencia de la imagen inicial de un GOP. El closed\_gop son datos indicadores que indican que la imagen en el GOP puede reproducirse con independencias de los otros grupos GOPs. El broken\_link son datos de indicadores para indicar que la imagen B inicial en el GOP no puede reproducirse con exactitud debido a edición o circunstancia similar.

45 Después de estos elementos de datos, se describen elementos de datos que representan la cabecera de imagen de la capa de imágenes que se utiliza para el proceso de codificación anterior, como un flujo de registro histórico, en la zona de usuarios de la capa de imágenes del flujo de bits generado durante el proceso de codificación final.

50 Los elementos de datos para la cabecera de imagen (picture\_header) están constituidos por picture\_start\_code, temporal\_reference, picture\_coding\_type, vbv\_delay, full\_pel\_forward\_vector, forward\_f\_code, full\_pel\_backward\_vector y backward\_f\_code.

55 Más concretamente, el picture\_start\_code son datos que representan un código de sincronización de inicio para la capa de imágenes. La temporal\_reference son datos que indican el orden de presentación visual de una imagen y qué, es objeto de reposición en la cabecera de un grupo GOP. El picture\_coding\_type son datos que indican el tipo de imagen. El vbv\_delay son datos que indican el estado inicial de la memoria virtual durante un acceso aleatorio. El full\_pel\_forward\_vector son datos que indican si la precisión de los vectores de movimiento en sentido directo son pixels enteros o mitad. El forward\_f\_code son datos que indican una gama de recuperación de los vectores de movimiento en sentido directo. El full\_pel\_backward\_vector son datos que indican si la precisión de los vectores de movimiento en sentido inverso son pixels enteros o mitad. El backward\_f\_code son datos que indican una gama de recuperación de los vectores de movimiento en sentido inverso.

60 Después de estos elementos de datos, se describe una etiqueta de codificación de imagen de la capa de imágenes que se utiliza para el proceso de codificación anterior, como un flujo de registro histórico, en la zona de usuarios de la capa de imágenes del flujo de bits generado durante el proceso de codificación final.

65 Los elementos de datos para la extensión de codificación de imagen (picture\_coding\_extension) están constituidos

por `extension_start_code`, `extension_start_code_identifier`, `f_code[0][0]`, `f_code[0][1]`, `f_code[1][0]`, `f_code[1][1]`, `intra_dc_precision`, `picture_structure`, `top_field_first`, `frame_predictive_frame_dct`, `concealment_motion_vectors`, `q_scale_type`, `intra_vlc_format`, `alternate_scan`, `repeat_first_field`, `chroma_420_type`, `progressive_frame`, `composite_display_flag`, `v_axis`, `field_sequence`, `sub_carrier`, `burst_amplitude` y `sub_carrier_phase`.

5 El código `extension_start_code` es un código de inicio que indica el inicio de datos de extensión para la capa de imágenes. El identificador `extension_start_code_identifier` es un código que indica qué código de extensión se envía. El `f_code[0][0]` son datos que representan una gama de recuperación de vectores de movimiento horizontales en una dirección directa. El `f_code[0][1]` son datos que representan una gama de recuperación de vectores de movimiento verticales en la dirección directa. El `f_code[1][0]` son datos que representan una gama de recuperación de vectores de movimiento horizontales en una dirección inversa. El `f_code[1][1]` son datos que representan una gama de recuperación de vectores de movimiento verticales en la dirección inversa.

15 El `intra_dc_precision` son datos que representan la exactitud de un coeficiente DC. La `picture_structure` son datos que indican si la imagen tiene una estructura de tramas o de campos. Si la imagen tiene la estructura de campos, estos datos indican también si el campo es más alto o más bajo. El `top_field_first` son datos que indican que el primer campo es más alto o más bajo si la imagen tiene la estructura de tramas. El `frame_predictive_frame_dct` son datos que indican que las predicciones DCT del modo de tramas indica solamente el modo de tramas. Si la imagen tiene la estructura de tramas. Lo `concealment_motion_vectors` son datos que indican que un intra-macrobloque tiene un vector de movimiento para ocultar errores de transmisión. El `q_scale_type` son datos que indican si utilizar una escala de cuantización lineal o no lineal. El `intra_vlc_format` son datos que indican si se utiliza otro vlc de dos dimensiones para el intra-macrobloque. El `alternate_scan` son datos que indican si seleccionar un escaneado alternado o en zigzag. El `repeat_first_field` son datos utilizados para la extracción en la relación 2:3. El `chroma_420_type` son datos que representan el mismo valor que la siguiente trama progresiva si el formato de la siguiente trama progresiva si el formato es 4:2:0 y de no ser así, representan 0. El `progressive_frame` son datos que indican si esta imagen ha sido escaneada de forma secuencial con resultado positivo o no. El `composite_display_flag` son datos que indican si la señal origen es compuesta o no lo es.

30 El `v_axis` son datos utilizados si la señal origen es PAL. La `field_sequence` son datos utilizados si la señal origen es PAL. La `sub_carrier` son datos utilizados si la señal origen es PAL. La `burst_amplitude` son datos utilizados si la señal origen es PAL. La `sub_carrier_phase` son datos utilizados si la señal origen es PAL.

35 Después de estos elementos de datos, se describe una extensión de matriz de cuantización utilizada para el proceso de codificación anterior, como un flujo de registro histórico, en la zona de usuarios de la capa de imágenes del flujo de bits generado durante el proceso de codificación final.

40 Los elementos de datos para la extensión de la matriz de cuantización (`quant_matrix_extension`) están constituidos por `extension_start_code`, `extension_start_code_identifier`, `quant_matrix_extension_present_flag`, `load_intra_quantiser_matrix`, `intra_quantiser_matrix[64]`, `load_non_intra_quantiser_matrix`, `non_intra_quantiser_matrix[64]`, `load_chroma_intra_quantiser_matrix`, `chroma_intra_quantiser_matrix[64]`, `load_chroma_non_intra_quantiser_matrix`, y `chroma_non_intra_quantiser_matrix[64]`.

45 El código `extension_start_code` es un código de inicio indicativo del comienzo de esta extensión de matriz de cuantización. El identificador `extension_start_code_identifier` es un código que indica qué datos de extensión se envían. El indicador `quant_matrix_extension_present_flag` son datos que indican si los elementos de datos en esta extensión de matriz de cuantización son válidos o no. La matriz `load_intra_quantiser_matrix` son datos indicativos de la presencia de datos de matriz de cuantización para un intra-macrobloque. La `intra_quantiser_matrix` son datos que indican un valor para la matriz de cuantización para el intra-macrobloque. La `load_non_intra_quantiser_matrix` son datos indicativos de la presencia de datos de matriz de cuantización para un no-intra-macrobloque. La `non_intra_quantiser_matrix` son datos que representan un valor para los datos de matriz de cuantización para el no-intra-macrobloque. La `load_chroma_intra_quantiser_matrix` son datos indicativos de la presencia de datos de matriz de cuantización para el intra-macrobloque de aberración cromática. La `chroma_intra_quantiser_matrix` son datos indicativos de un valor para los datos de matriz de cuantización para el intra-macrobloque de aberración cromática. La `load_chroma_non_intra_quantiser_matrix` son datos indicativos de la presencia de datos de matriz de cuantización para el no-intra-macrobloque de aberración cromática. La `chroma_non_intra_quantiser_matrix` son datos indicativos de un valor para los datos de matriz de cuantización para el no-intra-macrobloque de aberración cromática.

60 Después de estos elementos de datos, se describe una extensión de copyright utilizada para el proceso de codificación anterior, como un flujo de registro histórico, en la zona de usuarios de la capa de imágenes del flujo de bits generado durante el proceso de codificación final.

65 Los elementos de datos para la extensión de copyright (`copyright_extension`) están constituidos por `extension_start_code`, `extension_start_code_identifier`, `copyright_extension_present_flag`, `copyright_flag`, `copyright_identifier`, `original_or_copy`, `copyright_number_1`, `copyright_number_2` y `copyright_number_3`.

5 El `extension_start_code` es un código de indicativo del comienzo de la extensión de copyright. El `extension_start_code_identifier` es un código que indica qué datos de extensión se envían. El `copyright_extension_present_flag` son datos que indican si los elementos de datos en la extensión de copyright son válidos o no. El `copyright_flag` indica si un derecho de autor copyright se adjudica a datos de vídeo codificados hasta la siguiente extensión de copyright o un final de secuencia.

10 El `copyright_identifier` son datos que identifican una organización de registro de copyright especificada por ISO/IEC JTC/SC29. El `original_or_copy` son datos que indican si los datos en el flujo de bits son originales o una copia. El `copyright_number_1` son datos que representan los bits 44 a 63 de un número de copyright. El `copyright_number_2` son datos que representan los bits 22 a 43 de un número de copyright. El `copyright_number_3` son datos que representan los bits 0 a 21 de un número de copyright.

15 Después de estos elementos de datos, se describe una extensión de presentación visual de imagen (`picture_display_extension`) utilizada para el proceso de codificación anterior, como un flujo de registro histórico, en la zona de usuarios de la capa de imágenes del flujo de bits generado durante el proceso de codificación final.

20 Los elementos de datos de esta presentación visual de imágenes están constituidos por `extension_start_code`, `extension_start_code_identifier`, `picture_display_extension_present_flag`, `frame_center_horizontal_offset_1`, `frame_center_vertical_offset_1`, `frame_center_horizontal_offset_2`, `frame_center_vertical_offset_2`, `frame_center_horizontal_offset_3`, y `frame_center_vertical_offset_3`.

25 El `extension_start_code` es un código de inicio indicativo del comienzo de la extensión de presentación visual de la imagen. El `extension_start_code_identifier` es un código que indica qué datos de extensión se envían. El indicador `picture_display_extension_present_flag` son datos que indican si los elementos de datos en la extensión de presentación visual de imágenes son válidos o no. El `frame_center_horizontal_offset` son datos que indican desplazamientos horizontales en una zona de presentación visual y que pueden definir tres valores de desplazamiento. El `frame_center_vertical_offset` son datos que indican desplazamientos verticales en la zona de presentación visual y que pueden definir tres valores de compensación.

30 La presente invención está caracterizada por cuanto que la información de registro histórico que representa esta extensión de presentación visual de imágenes está seguida por elementos de datos para la `re_coding_stream_information`. Los elementos de datos para la `re_coding_stream_information` están constituidos por `user_data_start_code`, `re_coding_stream_info_ID`, `red_bw_flag`, `red_bw_indicator`, o similares.

35 El `user_data_start_code` es un código de inicio que indica que comienzan los datos de usuario `user_data`. El `re_coding_stream_info_ID` es un número entero de 16 bits utilizados para identificar la función `re_coding_stream_infor()`. Más concretamente, el valor de este número entero es "1001 0001 1110 1100" (0x91ec).

40 El indicador `red_bw_flag` es un indicador de 1 bit que es 0 para la transmisión de parámetros de codificación para toda la información de registro histórico y que es 1 para la transmisión selectiva de parámetros de codificación para información de registro histórico. El indicador `red_bw_indicator` es un número entero de 2 bits que es un indicador para definir un conjunto de datos de parámetros de codificación.

45 La `re_coding_stream_information`, el indicador `red_bw_flag`, el `red_bw_indicato` y conjunto de datos se describirán en detalle más adelante.

50 Los datos de usuarios (`user_data`) utilizados para el proceso de codificación anterior se describen, como un flujo de registro histórico, en la zona de usuarios de la capa de imágenes del flujo de bits generado durante el proceso de codificación final.

Después de los datos de usuarios, se describe la información sobre la capa de macrobloques utilizada para el proceso de codificación anterior como un flujo de registro histórico.

55 La información sobre la capa de macrobloques está constituida por elementos de datos para las posiciones de macrobloques (`macroblock`) tal como `macroblock_addr_h`, `macroblock_addr_v`, `slice_header_present_flag` y `skipped_macroblock_flag`, con elementos de datos para un modo de macrobloques (`macroblock_modes[]`) tal como `macroblock_quant`, `macroblock_motion_forward`, `macroblock_motion_backward`, `macroblock_pattern`, `macroblock_intra`, `spatial_temporal_weight_code_flag`, `frame_motion_type`, y `dct_type`, elementos de datos para el control de la etapa de cuantización tal como `quantiser_scale_code`, elementos de datos para la compensación de movimiento tales como `PMV[0][0][0]`, `PMV[0][0][1]`, `motion_vertical_field_select[0][0]`, `PMV[0][1][0]`, `PMV[0][1][1]`, `motion_vertical_field_select[0][1]`, `PMV[1][0][0]`, `PMV[1][0][1]`, `motion_vertical_field_select[1][0]`, `PMV[1][1][0]`, `PMV[1][1][1]`, y `motion_vertical_field_select[1][1]`, elementos de datos para un modelo de macrobloques tal como `coded_block_pattern`, y elementos de datos para la cantidad de códigos generados tales como `num_mv_bits`, `num_coef_bits` y `num_other_bits`.

65 Los elementos de datos para la capa de macrobloques se describirán a continuación en detalle.

5 El macroblock\_adres\_h son datos para definir la posición absoluta horizontal del macrobloque en curso. La macroblock\_adres\_v son datos para definir la posición absoluta vertical del macrobloque en curso. El slice\_header\_present\_flag son datos que indican si este macrobloque está, o no, en la cabecera de la capa de segmentos e incluye una cabecera de segmentos. El skipped\_macroblock\_flag son datos que indican si saltar operativamente o no, este macrobloque durante un proceso de decodificación.

10 El macroblock\_quant son datos derivados de un tipo de macrobloque (macroblock\_type), ilustrado en las Figuras 63 y 64, descrito a continuación, e indica si el código quantiser\_scale\_code aparece, o no, en el flujo de bits. El macroblock\_motion\_forward son datos derivados del tipo de macrobloque ilustrado en las Figuras 63 y 64 y se utiliza para el proceso de decodificación. El macroblock\_motion\_backward son datos derivados del tipo de macrobloque ilustrado en las Figuras 63 y 64 y se utiliza para el proceso de decodificación. El macroblock\_pattern son datos derivados del tipo de macrobloque ilustrado en las Figuras 63 y 64 e indica si el coded\_block\_pattern aparece, o no, en el flujo de bits.

15 El macroblock\_intra son datos derivados del tipo de macrobloque ilustrado en las Figuras 63 y 64 y se utiliza para el proceso de decodificación. El indicador spatial\_temporal\_weight\_code\_flag son datos derivados del tipo de macrobloque ilustrado en las Figuras 63 y 64 e indica si el flujo de bits contiene, o no, el código spatial\_temporal\_weight\_code indicativo de un método para el muestreo ascendente de imágenes de capas inferiores utilizando la escalabilidad temporal.

20 El frame\_motion\_type es un código de 2 bits que indica un tipo de predicción de un macrobloque en una trama. Este código es "00" si dos vectores de predicción están presentes y son de un tipo de predicción de base de campos, "01" si un vector de predicción está presente y es del tipo de predicción de base de campos, "10" si un vector de predicción está presente y es de un tipo de predicción de base de tramas y "11" si un vector de predicción está presente y es de un tipo de predicción principal dual. El field\_motion\_type es un código de 2 bits que indica una predicción de movimiento para un macrobloque en un campo. Este código es "01" si un vector de predicción está presente y es del tipo de predicción de base de campos, "10", si dos vectores de predicción están presentes y son de un tipo de predicción de base de macrobloque 18 x 8 y "11" si un vector de predicción está presente y es de un tipo de predicción principal dual. El dct\_type son datos que indican si el DCT está en un modo DCT de tramas o de campos de imagen. El quantiser\_scale\_code son datos que indican la magnitud de la etapa de cuantización de un macrobloque.

25 A continuación, se describirán elementos de datos para vectores de movimiento. Un vector de movimiento se codifica en una forma diferencial desde el vector codificado precedente con el fin de reducir el número de vectores de movimiento requeridos para la decodificación. El decodificador debe mantener cuatro valores de predicción de vectores de movimiento (cada uno incluyendo componentes horizontales y verticales) para decodificar vectores de movimiento. Estos vectores de movimiento objeto de predicción están representados como PMV[r] [s] [v]. El [r] es un indicador que indica un primero o un segundo vector de movimiento en un macrobloque y es "0" para el primer vector de movimiento y "1" para el segundo vector de movimiento. El [s] es un indicador que indica si un vector de movimiento en un macrobloque es un vector en sentido directo o inverso y es "0" si este vector de movimiento es un vector en sentido directo y "1" si es un vector en sentido inverso. El [v] es un indicador que indica si existe un componente de vector horizontal o vertical en un macrobloque y es "0" para un componente de vector horizontal y "1" para un componente de vector vertical.

35 De este modo, el PMV[0][0][0] representa datos para el componente horizontal del vector de movimiento en sentido directo para el primer vector. El PMV[0] [0] [1] representa datos para el componente vertical del vector de movimiento en sentido directo para el primer vector. El PMV[0] [1] [0] representa datos para el componente horizontal del vector de movimiento en sentido inverso para el primer vector. El PMV[0][1][1] representa datos para el componente vertical del vector de movimiento en sentido inverso para el primer vector.

40 El PMV[1] [0] [0] representa datos para el componente horizontal del vector de movimiento directo del segundo vector. El PMV[1][0][1] representa datos para el componente vertical del vector de movimiento directo del segundo vector.

45 El PMV[1] [1] [0] representa datos para el componente horizontal del vector de movimiento en sentido inverso del segundo vector. El PMV[1][1][1] representa datos para el componente vertical del vector de movimiento en sentido inverso del segundo vector.

50 El motion\_vertical\_field\_select[r][s] son datos que indican qué campo de referencia se utiliza para una forma de predicción. Si el motion\_vector\_field\_select[r][s] es "0", se utiliza un campo de referencia superior. Si es "1", se utiliza un campo de referencia inferior.

55 En consecuencia, motion\_vertical\_field\_select[0][0] indica un campo de referencia para uso en la generación del vector de movimiento en sentido directo del primer vector. motion\_vertical\_field\_select[0][1] indica un campo de referencia para uso en la generación del vector de movimiento en sentido inverso del primer vector. motion\_vertical\_field\_select[1][0] indica un campo de referencia para uso en la generación del vector de movimiento

en sentido directo del segundo vector. `motion_vertical_field_select[1][1]` indica un campo de referencia para uso en la generación del vector de movimiento en sentido inverso del segundo vector.

5 El `coded_block_pattern` son datos de una longitud variable que indica cuáles de bloques DCT plurales que memorizan coeficientes DCT tiene un coeficiente significativo (un coeficiente no nulo). El `num_mv_bits` son datos indicativos de la cantidad de códigos en un vector de movimiento en un macrobloque. El `num_coef_bits` son datos indicativos de la cantidad de códigos en un coeficiente DCT en un macrobloque. El `num_other_bits` son datos que indican la cantidad de códigos en un macrobloque que no sean los vectores de movimiento y los coeficientes DCT.

10 A continuación, se explicará una sintaxis para la decodificación de cada elemento de datos desde un flujo de registro histórico de una longitud variable haciendo referencia a las Figuras 47 a 67.

15 Este flujo de registro histórico de longitud variable está constituido por la función `next_start_code()`, la función `sequence_header()`, la función `sequence_extension()`, la función `extension_and_user_data(0)`, la función `group_of_picture_header()`, la función `extension_and_user_data(1)`, la función `picture_header()`, la función `picture_coding_extension()`, la función `re_coding_stream_info()`, la función `extension_and_user_data(2)` y la función `picture_data()`.

20 Puesto que la función `next_start_code()` busca un código de inicio presente en el flujo de bits, se describe en la cabecera del flujo de registro histórico, con los elementos de datos utilizados para el proceso de codificación anterior y definido por la función `sequence_header()` según se ilustra en la Figura 48.

25 Los elementos de datos definidos por la función `sequence_header()` es el `sequence_header_code`, el `sequence_header_present_flag`, el `horizontal_size_value`, el `vertical_size_value`, la `aspect_ratio_information`, el `frame_rate_code`, el `bit_rate_value`, el `marker_bit`, el valor `VBV_buffer_size_value`, el indicador `constrained_parameter_flag`, la matriz `load_intra_quantiser_matrix`, la `intra_quantiser_matrix`, la `load_non_intra_quantiser_matrix`, la `non_intra_quantiser_matrix`, etc.

30 El `sequence_header_code` son datos que representan un código de sincronización de inicio para la capa de secuencias., El `sequence_header_present_flag` indica si los datos en la cabecera de secuencia son válidos o no. El `sequence_header` `horizontal_size_value` son datos que están constituidos por los 12 bits inferiores del número de pixels en una dirección horizontal de una imagen. El `vertical_size_value` son datos que consisten en los 12 bits inferiores el número de líneas verticales en una imagen. La `aspect_ratio_information` son datos que representan una relación de aspecto del pixel o una relación de aspecto de pantalla de presentación visual. El `frame_rate_code` son datos que representan un ciclo de presentación visual de imagen. El `bit_rate_value` son datos de 18 bits inferiores (con utilización de 400 bsp como una unidad) de una tasa binaria para limitar la cantidad de bits generados.

35 El `marker_bit` son datos de bits insertados para impedir la emulación del código de inicio. El `VBV_buffer_size_value` son los datos de 10 bits inferiores de un valor que determina la magnitud de una memoria intermedia virtual (un verificador de memoria de vídeo) para controlar la cantidad de códigos generados. El `constrained_parameter_flag` son datos que indican que cada parámetro está por debajo del límite. La `load_intra_quantiser_matrix` son datos que indican la presencia de datos de matriz de cuantización intra-MB. La `intra_quantiser_matrix` son datos que indican un valor para los datos de matriz de cuantización intra-MB. La `load_non_intra_quantiser_matrix` son datos que indican la presencia de datos de matriz de cuantización no\_intra-MB. La `intra_quantiser_matrix` son datos que indican un valor para los datos de matriz de cuantización no-intra-MB.

40 Después de los elementos de datos definidos por la función `sequence_header()`, los elementos de datos definidos por las función `sequence_extension()` tales como los ilustrados en la Figura 49 se describen como un flujo de registro histórico.

45 Los elementos de datos definidos por la función `sequence_extension()` son los `extension_start_code`, `extension_start_code_identifier`, `sequence_extension_present_flag`, `profile_and_level_indication`, `progressive_sequence`, `chroma_format`, `horizontal_size_extension`, `vertical_size_extension`, `bit_rate_extension`, `vbv_buffer_size_extension`, `low_delay`, `frame_rate_extension_n`, `frame_rate_extension_d`, etc.

50 El `extension_start_code` son datos que representan un código de sincronización de inicio para datos de extensión. El `extension_start_code_identifier` son datos que indican qué datos de extensión se envían. El `sequence_extension_present_flag` son datos que indican si los datos en la extensión de secuencia son válidos o no. La `profile_and_level_indication` son datos que especifican el perfil y el nivel de datos de vídeo. El `progressive_sequence` son datos que indican que los datos de vídeo están escaneados de forma secuencia. El `chroma_format` son datos que especifican el formato de aberración cromática de los datos de vídeo. La `horizontal_size_extension` son los datos de 2 bits superiores añadidos al valor `horizontal_size_value` de la cabecera de secuencia. La `vertical_size_extension` son los datos de 2 bits superiores añadidos al valor de `vertical_size_value` de la cabecera de secuencia. La `bit_rate_extension` son los datos de 12 bits superiores añadidos al valor `bit_rate_value` de la cabecera de secuencia. La `vbv_buffer_size_extension` son los datos de 8 bits superiores añadidos al valor `vbv_buffer_size_value` de la cabecera de secuencia.

5 El `low_delay` son datos indicativos de la ausencia de la imagen B. La `frame_rate_extension_n` son datos que se combinan con el `frame_rate_code` de la cabecera de secuencia para obtener una tasa de tramas. La `frame_rate_extension_d` son datos que se combinan con el código `frame_rate_code` de la cabecera de secuencia para obtener la tasa de tramas.

10 Después de los elementos de datos definidos por la función `sequence_extension()`, elementos de datos definidos por la función `extension_and_user_data(0)` tales como los ilustrados en la Figura 50 se describen como un flujo de registro histórico. Cuando "i" no es 1, la función `extension_and_user_data(i)` no describe los elementos de datos definidos por la función `extension_data()` sino solamente los elementos de datos definidos por la función `user_data()` como un flujo de registro histórico.

15 La función `extension_and_user_data(0)` describe solamente los elementos de datos definidos por la función `user_data()`, como un flujo de registro histórico.

La función `user_data()` describe los datos de usuarios como un flujo de registro histórico basado en dicha sintaxis según se ilustra en la Figura 51.

20 Después de los elementos de datos definidos por la función `extension_and_user_data(0)`, los elementos de datos definidos por la función `group_of_picture_header()` y la función `extension_and_user_data(1)`, como un flujo de registro histórico, se ilustran en la Figura 52. Sin embargo, los elementos de datos definidos por la función `group_of_picture_header()` y la función `extension_and_user_data(1)` se describen solamente si el `group_start_code` indicativo del código de inicio para la capa GOP se describe en el flujo de registro histórico.

25 Los elementos de datos definidos por la función `group_of_picture_header()` están constituidos por el `group_start_code`, `group_of_picture_header_present_flag`, `time_code`, `closed_gop`, y `broken_link`.

30 El `group_start_code` son datos que indican un código de sincronización de inicio para la capa GOP. El `group_of_picture_header_present_flag` son datos que indican si los elementos de datos en `group_of_picture_header` son válidos o no. El `time_code` es un código temporal que indica el tiempo desde la cabecera de la secuencia de la imagen inicial de un GOP. El `closed_gop` son datos que indican que la imagen en el grupo GOP puede reproducirse con independencia de los otros grupos GOPs. El `broken_link` son datos que indican que la imagen B inicial en el grupo GOP no puede reproducirse con exactitud debido a edición o circunstancia similar.

35 Como la función `extension_and_user_data(0)`, la función `extension_and_user_data(1)` describe solamente los elementos de datos definidos por la función `user_data()`, como un flujo de registro histórico.

40 Si el flujo de registro histórico no contiene el `group_start_code` indicativo del código de inicio para la capa GOP, el flujo de registro histórico descrito no tiene elementos de datos definidos por la función `group_of_picture_header()` y la función `extension_and_user_data(1)`. En este caso, después de los elementos de datos definidos por la función `extension_and_user_data(0)`, se describen elementos de datos definidos por la función `picture_header()` como un flujo de registro histórico.

45 Los elementos de datos definidos por la función `picture_header()` son los `picture_start_code`, `temporal_reference`, `picture_coding_type`, `vbv_delay`, `full_pel_forward_vector`, `forward_f_code`, `full_pel_backward_vector`, `backward_f_code`, `extra_bit_picture`, y `extra_information_picture`, según se ilustra en la Figura 53.

50 Más concretamente, el `picture_start_code` son datos que representan un código de sincronización de inicio para la capa de imágenes. La `temporal_reference` son datos que indican el orden de presentación visual de una imagen y lo que es objeto de reposición en la cabecera de un grupo GOP. El `picture_coding_type` son datos que indican el tipo de imagen. El `vbv_delay` son datos que indican el estado inicial de la memoria virtual durante un acceso aleatorio. El `full_pel_forward_vector` son datos que indican si la precisión de los vectores de movimiento en sentido directo es números enteros de pixels o pixels mitad. El `forward_f_code` son datos que indican una gama de recuperación de los vectores de movimiento en sentido directo. El `full_pel_backward_vector` son datos que indican si la precisión de los vectores de movimiento en sentido inverso son números enteros de pixels o mitad. El `backward_f_code` son datos que indican una gama de recuperación de los vectores de movimiento en sentido inverso. El `extra_bit_picture` es un indicador de la presencia de información adicional posterior. Si el `extra_bit_picture` es "1", sigue el `extra_information_picture`, mientras si es "0", no siguen datos algunos. La `extra_information_picture` es información reservada para la norma.

60 Después de los elementos de datos definidos por la función `picture_header()`, se describen los elementos de datos definidos por la función `picture_coding_extension()` tal como los ilustrados en las Figura 54 como flujo de registro histórico.

65 Los elementos de datos definidos por la función `picture_coding_extension()` están constituidos por `extension_start_code`, `extension_start_code_idenfier`, `f_code[0][0]`, `f_code[0][1]`, `f_code[1][0]`, `f_code[1][1]`, `intra_dc_precision`, `picture_structure`, `top_field_first`, `frame_predictive_frame_dct`, `concealment_motion_vectors`,

q\_scale\_type, intra\_vlc\_format, alternate\_scan, repeat\_firt\_field, chroma\_420\_type, progresive\_frame, composite\_display\_flag, v\_axis, field\_sequence, sub\_carrier, burst\_amplitude y sub\_carrier\_phase.

5 El extension\_start\_code es un código de inicio que indica el comienzo de los datos de extensión para la capa de imágenes. El extension\_start\_code\_idenfier es un código que indica qué código de extensión se envía. El f\_code[0][0] son datos que representan una gama de recuperación de vectores de movimiento horizontales en una dirección hacia delante. El f\_code[0][1] son datos que representan una gama de recuperación de vectores de movimiento verticales en la dirección directa hacia delante. El f\_code[1][0] son datos que representan una gama de recuperación de vectores de movimiento horizontales en una dirección inversa. El f\_code[1][1] son datos que representan una gama de recuperación de vectores de movimiento verticales en una dirección inversa. La intra\_dc\_precision son datos que representan la exactitud de un coeficiente DC.

15 El picture\_structure son datos que indican si la imagen tiene una estructura de tramas o de campos. Si la imagen tiene la estructura de campos, estos datos indican también si el campo es más alto o más bajo. El top\_field\_first son datos que indican si el primer campo es más alto o más bajo si la imagen tiene la estructura de tramas. La frame\_predictive\_frame\_dct son datos que indican que las predicciones DCT del modo de trama significa solamente el modo de tramas, si la imagen tiene la estructura de tramas. Los concealment\_motion\_vectors son datos que indican que un intra-macrobloque tiene un vector de movimiento para ocultar errores de transmisión. El q\_scale\_type son datos que indican si utilizar una escala de cuantización lineal o no lineal. El intra\_vlc\_format son datos que indican si se utiliza otro VLC bidimensional para el intra-macrobloque.

25 El alternate\_scan son datos que indican si se seleccionar un escaneado alternado o en zigzag. El repeat\_firt\_field son datos utilizados para la relación 2:3 de presentación desplegable. El chroma\_420\_type son datos que representan el mismo valor que la siguiente trama progresiva progresive\_frame si el formato de señal es 4:2:0 y de o ser así, representa 0. La progresive\_frame son datos que indican si esta imagen ha sido, o no, secuencialmente escaneada con resultado positivo. El composite\_display\_flag son datos que indican si la señal origen es compuesta o no lo es. El v\_axis son datos utilizados si la señal de origen es PAL. La field\_sequence son datos utilizados si la señal de origen es PAL. La sub\_carrier son datos utilizados si la señal de origen es PAL. La burst\_amplitude son datos utilizados si la señal de origen es PAL. La sub\_carrier\_phase son datos utilizados si la señal origen es PAL.

30 Después de los elementos de datos definidos por la función picture\_coding\_extension(), se describen los elementos de datos definidos por la función re\_coding\_stream\_info() como un flujo de registro histórico. La función re\_coding\_stream\_info() es característica de la presente invención y se utiliza principalmente para describir una combinación de información de registro histórico. Los detalles se examinarán más adelante haciendo referencia a la Figura 68.

40 Después de los elementos de datos definidos por la función re\_coding\_stream\_info(), se describen los elementos de datos definidos por la función extensions\_and\_user\_data(2) como un flujo de registro histórico. Según se ilustra en la Figura 50, si el código de inicio de extensión (extension\_start\_code) está presente en el flujo de bits se contienen los elementos de datos definidos por la función extension\_data() que se describen en la extension\_and\_user\_data(2). Después de estos elementos de datos, se describen los elementos de datos definidos por la función user\_data() si el código de inicio de datos de usuario (user\_data\_start\_code) está presente en el flujo de bits. Si el código de inicio de extensión y el código de inicio de datos de usuario están ausentes del flujo de bits, los elementos de datos definidos por las funciones extension\_data() y user\_data() no se describen en el flujo de bits.

45 La función extension\_data() describe en el flujo de bits. Como un flujo de registro histórico, los elementos de datos que indican el extension\_start\_code y elementos de datos definidos por las funciones quant\_matrix\_extension(), copyright\_extension() y picture\_display\_extension(), según se ilustra en la Figura 55.

50 Los elementos de datos definidos por la función quant\_matrix\_extension() son los denominados extension\_start\_code, extension\_start\_code\_idenfier, quant\_matrix\_extension\_present\_flag, load\_intra\_quantiser\_matrix, intra\_quantiser\_matrix[64], load\_non\_intra\_quantiser\_matrix, non\_intra\_quantiser\_matrix[64], load\_chroma\_intra\_quantiser\_matrix, ckroma\_intra\_quantiser\_matrix[64], load\_chroma\_non\_intra\_quantiser\_matrix y chroma\_non\_intra\_quantiser\_matrix[64], según se ilustra en la Figura 56.

60 El extension\_start\_code es un código de inicio que indica el comienzo de la extensión de matriz de canal de cuantización. El extension\_start\_code\_idenfier es un código que indica qué datos de extensión se envían. El quant\_matrix\_extension\_present\_flag son datos que indican si los elementos de datos en esta extensión de matriz de cuantización son válidos o no. La load\_intra\_quantiser\_matrix son datos indicativos de la presencia de datos de matriz de cuantización para un intra-macrobloque. La intra\_quantiser\_matrix son datos que indican un valor para la matriz de cuantización para el intra-macrobloque. La load\_non\_intra\_quantiser\_matrix son datos indicativos de la presencia de datos de matriz de cuantización para un no-intra-macrobloque. La no\_intra\_quantiser\_matrix son datos que representan un valor para los datos de matriz de cuantización para el no-intra. La load\_chroma\_intra\_quantiser\_matrix son datos indicativos de la presencia de datos de matriz de cuantización para intra-macrobloque de aberración cromática. La chroma\_intra\_quantiser\_matrix son datos indicativos de un valor para



los datos de matriz de cuantización para el intra-macrobloque de aberración cromática. La `load_chroma_non_intra_quantiser_matrix` son datos indicativos de la presencia de datos de matriz de cuantización para no-intra-macrobloque de aberración cromática. La `chroma_non_intra_quantiser_matrix` son datos indicativos de un valor para los datos de matriz de cuantización para el no-intra-macrobloque de aberración cromática.

5 Los elementos de datos definidos por la función `copyright_extension()` están constituidos por `extension_start_code`, `extension_start_code_idenfier`, `copyright_extension_present_flag`, `copyright_flag`, `copyright_idenfier`, `original_or_copy`, `copyright_number_1`, `copyright_number_2`, y `copyright_number_3`, según se ilustra en la Figura 57.

10 El `extension_start_code` es un código de inicio indicativo del comienzo de la extensión de copyright. El `extension_start_code_idenfier` es un código que indica qué datos de extensión se envían. El `copyright_extension_present_flag` son datos que indican si los elementos de datos en la extensión de copyright son válidos o no.

15 El `copyright_flag` indica si se adjudica, o no, un copyright a los datos de vídeo codificados hasta la siguiente extensión de copyright o un final de secuencia. El `copyright_idenfier` son datos que identifican una organización de registro de copyright especificada por ISO/IEC JTC/SC29. El `original_or_copy` son datos que indican si los datos en el flujo de bits son originales o una copia. El `copyright_number_1` son datos que representan los bits 44 a 63 de un número de copyright. El `copyright_number_2` son datos que representan los bits 22 a 43 de un número de copyright.  
20 El `copyright_number_3` son datos que representan los bits 0 a 21 de un número de copyright.

Los elementos de datos definidos por la función `picture_display_extension()` son los denominados `extension_start_code_idenfier`, `frame_center_horizontal_offset`, `frame_center_vertical_offset`, y similares, según se ilustra en la Figura 58.

25 El `extension_start_code_idenfier` es un código que indica qué datos de extensión se envían. El `frame_center_horizontal_offset` son datos que indican las compensaciones horizontales en una zona de presentación visual y que pueden definir un número definido de valores de desplazamiento de compensación utilizando `number_of_frame_center_offset`. La `frame_center_vertical_offset` son datos que indican desplazamientos de compensación verticales en la zona de presentación visual y que definir un número definido de compensación utilizando `number_of_frame_center_offsets`.  
30

Haciendo referencia de nuevo a la Figura 47, después de los elementos de datos definidos por la función `extension_and_user_data(2)`, se describen los elementos de datos definidos por la función `picture_header()` como un flujo de registro histórico. La función `picture_data()`, sin embargo, está presente si el indicador `red_bw_flag` no es 1 o si el indicador `red_bw_indicator` es 2 o menor. El `red_bw_flag` y el `red_bw_indicator` se describen en la función `re_coding_stream_info()` y se describirán más adelante haciendo referencia a las Figuras 68 y 69.  
35

Los elementos de datos definidos por la función `picture_data()` son los definidos por la función `slice()` según se ilustra en la Figura 59. Al menos uno de estos elementos de datos definidos por la función `slice()` se describen en el flujo de bits.  
40

La función `slice()` describe un flujo de registro histórico de elementos de datos tales como `slice_start_code`, `slice_quantiser_scale_code`, `intra_slice_flag`, `intra_slice`, `reserved_bits`, `extra_bit_slice`, `extra_information_slice` y `extra_bit_slice` y elementos de datos definidos por la función `macroblock()`.  
45

El `slice_start_code` es un código de inicio indicativo del comienzo de los elementos de datos definidos por la función `slice()`. El `slide_quantiser_scale_code` son datos que indican una magnitud de la etapa de cuantización establecida para un macrobloque presente en esta capa de segmentos. Si, sin embargo, el `quantiser_scale_code` se establece para cada macrobloque, los datos en `macroblock_quantiser_scale_code` establecidos para cada macrobloque se utilizan de forma preferente.  
50

El `intra_slice_flag` es un indicador de si los `intra_slice` y los `reserved_bits` están presentes, o no, en el flujo de bits. Los `intra_slice` son datos que indican si un no-intra-macrobloque está presente, o no, en la capa de segmentos. Si cualquiera de los macrobloques en la capa de segmentos es un no-intra-macrobloque, el `intra_slice` es "0". Si todos los macrobloques en la capa de segmentos son no-intra-macrobloques, el `intra_slice` es "1". Los `reserved_bits` son datos de 7 bits y son "0". El `extra_bit_slice` es un indicador de la presencia de información adicional como un flujo de registro histórico. Si sigue el `extra_information_slice`, este indicador se establece a "1", mientras que sino sigue ninguna información adicional, se establece a "0."  
55  
60

Después de estos elementos de datos de datos, se describen los elementos de datos definidos por la función de `macroblock()` como un flujo de registro histórico.

La función de `macroblock()` describe elementos de datos tales como `macroblock_escape`, `macroblock_addres_increment`, `macroblock_quantiser_scale_code`, y `marker_bit` y elementos de datos definidos por una función `macroblock_modes`, una función `motion_vectors(s)` y una función `code_block_pattern()`.  
65

- 5 El `macroblock_escape` es una cadena de bits fija que indica si la diferencia entre un macrobloque de referencia y el macrobloque precedente es 34 o más, si la diferencia entre el macrobloque de referencia y el macrobloque precedente es 34 o más, se añade 33 al valor del `macroblock_adres_increment`. El `macroblock_adres_increment` son datos que indican una diferencia en la dirección horizontal entre el macrobloque de referencia y el macrobloque precedente. Si un `macroblock_escape` está presente antes `macroblock_adres_increment`, los datos indicativos de los datos diferenciales reales en la dirección horizontal entre el macrobloque de referencia y el macrobloque precedente se obtienen añadiendo 33 al valor de `macroblock_adres_increment`.
- 10 El `macroblock_quantiser_scale_code` es una magnitud de la etapa de cuantización establecida para cada macrobloque y está presente solamente cuando el `macroblock_quant` es "1". Cada capa de segmentos tiene el `slice_quantiser_scale_code` establecido y que indica la magnitud de la etapa de cuantización y esta magnitud de la etapa de cuantización se selecciona si el `macroblock_quantiser_scale_code` se establece para el macrobloque de referencia.
- 15 Después del `macroblock_adres_increment`, se describen elementos de datos definidos en la función `macroblock_modes()`. La función `macroblock_modes()` describe los elementos de datos tales como `macroblock_type`, `frame_motion_type`, `field_motion_type`, y `dct_type` como un flujo de registro histórico, según se ilustra en la Figura 62.
- 20 El `macroblock_type` son datos indicativos del tipo de codificación de un macrobloque. Los detalles se examinarán más adelante haciendo referencia a las Figuras 65 a 67.
- 25 Si el `macroblock_motion_forward` o el `macroblock_motion_backward` es "1", la estructura de imagen es la trama y el `frame_pred_frame_dct` es "0", representando entonces los elementos de datos el `frame_motion_type` que se describe después de los elementos de datos que representan el `macroblock_type`. El `frame_pred_frame_dct` es un indicador de si el tipo de movimiento de campo `field_motion_type` está presente en el flujo de bits.
- 30 El `frame_motion_type` es un código de 2 bits que indica el tipo de predicción de un macrobloque en una trama. Este código es "00", si dos vectores de predicción están presentes y son del tipo de predicción de base de campo, "01" si un vector de predicción está presente y es del tipo de predicción de base de campo, "10" si un vector de predicción está presente y si es del tipo de predicción de base de trama y "11" si un vector de predicción está presente y es de un tipo de predicción principal dual.
- 35 Si no se cumplen las condiciones para la descripción del tipo `frame_motion_type`, los elementos de datos que representan el tipo de movimiento de campo `field_motion_type` se describen después de los elementos de datos que representan el tipo de macrobloque `macroblock_type`.
- 40 El `field_motion_type` es un código de 2 bits que indica una predicción de movimiento para un macrobloque en un campo. Este código es "01" si un vector de predicción está presente y es del tipo de predicción de base de campo, "10" si dos vectores de predicción están presentes y son del tipo de predicción de base de macrobloque 18 x 8 y "11" si un vector de predicción está presente y es de un tipo de predicción principal dual.
- 45 Si la estructura de imagen es la trama, la `frame_pred_frame_dct` indica que el `frame_motion_type` y el tipo DCT están presentes en el flujo de bits, los elementos de datos que representan el `dct_type` se describen después de los elementos de datos que representan el tipo `macroblock_type`. El `dct_type` son datos que indican si el DCT está en el modo DCT de trama o de campo.
- 50 Haciendo referencia de nuevo a la Figura 61, si el macrobloque de referencia es del tipo de predicción en sentido directo o es un intra-macrobloque con un macrobloque de ocultación, se describen los elementos de datos definidos por los `motion_vectors(0)`. Además, si el macrobloque de referencia es del tipo de predicción inversa, se describen los elementos de datos definidos por la función `motion_vectors(1)`.
- 55 La función `motion_vectors(0)` describe elementos de datos para un primer vector de movimiento y la función `motion_vectors(1)` describes elementos de datos para un segundo vector de movimiento.
- La función `motion_vectors(s)` describe los elementos de datos para el vector de movimiento según se ilustra en la Figura 63.
- 60 Si un vector de movimiento está presente y no se utiliza el modo de predicción principal dual, se describen elementos de datos definidos por `motion_vertical_field_select[0][s]` y la función `motion_vector[0, s]`.
- 65 El `motion_vertical_field_select[r][s]` es un indicador de si el primer vector de movimiento (que puede ser un vector en sentido directo o inverso) se obtiene haciendo referencia a un campo superior o a un campo inferior. El indicador "r" indica si es el primero o segundo vector y el indicador "s" indica si la dirección de predicción es directa o inversa.

La función `motion_vector[r, s]` describe una cadena de datos para `motion_code [r] [s] [t]`, una cadena de datos para `motion_residual [r] [s] [t]`, y datos que representan `dmvector[t]`, según se ilustra en la Figura 64.

5 El `motion_code [r] [s] [t]` son datos de una longitud variable que representa la magnitud de un vector de movimiento entre -16 y +16. De este modo, los valores del `motion_code[r][s][t]` y del `motion_residual[r][s][t]` pueden describir vectores de movimiento detallados. El `dmvector[t]` son datos que operan en el modo de predicción principal dual para generar vectores de movimiento en uno de los campos (a modo de ejemplo, el campo se refiere como "uno de los campos" en comparación con el campo inferior) mediante el escalado de vectores de movimiento existentes dependiendo de las distancias temporales mientras que se relacionan correcciones con el fin de reflejar los desplazamientos verticales entre líneas que aparecen entre el campo superior y el campo inferior. El indicador "r" indica si es el primero o el segundo vector y el indicador "s" indica si la dirección de predicción es directa o inversa.

15 La función `motion_vector[r,s]` describe una cadena de datos que representa a `motion_code[r][s][0]` para la dirección horizontal, como un flujo de registro histórico, según se ilustra en la Figura 64. Puesto que el número de bits en ambas funciones `motion_residual[0][s][t]` y `motion_residual[1][s][t]` se indica por `f_code[s] [t]`, el `f_code [s] [t]` de un valor distinto de 1 indica la presencia de la función `motion_residual[r][s][t]` en el flujo de bits. Cuando la función `motion_residual[r][s][0]` para una componente horizontal no es "1" y la función `motion_coder[r] [s] [0]` para una componente horizontal no es "0", esto significa que los elementos de datos que representan la función `motion_residual [r] [s] [0]` están presentes en el flujo de bits y el vector de movimiento tiene una componente horizontal. En este caso, se describen elementos de datos que representan la función `motion_residual[r][s][0]` para una componente horizontal.

25 Después de estos elementos de datos se describe una cadena de datos que representa la función `motion_coder[r][s][1]` para la dirección vertical como un flujo de registro histórico. Según se describió con anterioridad, puesto que el número de bits en ambas funciones `motion_residual[0] [s] [t]` y `motion_residual[1] [s] [t]` se indica por el `f_code[s] [t]`, el `f_code [s] [t]` de un valor distinto de 1 indica la presencia de la función `motion_residual[r][s][t]` en el flujo de bits. Cuando la función `motion_residual[r] [s] [1]` no es "1" y la función `motion_coder[r] [s] [1]` no es "0," esto significa que elementos de datos que representan la función `motion_residual [r] [s] [1]` están presentes en el flujo de bits y el vector de movimiento tiene una componente vertical. En este caso, se describen elementos de datos que representan la función `motion_residual [r] [s] [1]` para una componente vertical.

35 A continuación, se describirá `macroblock_type` haciendo referencia a las Figuras 65 a 67. El `macroblock_type` son datos de una longitud variable generados por indicadores tales como `macroblock_quant`, `dct_type_flag`, `macroblock_motion_forward` y `macroblock_motion_backward`. El `macroblock_quant` es un indicador de si se ha establecido o no, el `macroblock_quantiser_scale_code`, que establece una magnitud de la etapa de cuantización para un macrobloque. El `macroblock_quant` es "1" si el `macroblock_quantiser_scale_code` está presente en el flujo de bits.

40 El `dct_type_flag` es un indicador de si está presente, o no, el `dct_type`, lo que indica si el macrobloque de referencia se ha codificado, o no, utilizando el DCT de trama o de campo (dicho de otro modo, indica si el macrobloque de referencia ha sido sometido al DCT). Si el `dct_type` está presente en el flujo de bits, el `dct_type_flag` es "1". El `macroblock_motion_forward` es un indicador de si el macrobloque de referencia ha sido objeto de predicción directa o no. Este indicador es 1 si el macrobloque de referencia ha sido objeto de predicción directa. El `macroblock_motion_backward` es un indicador de si el macrobloque de referencia ha sido objeto de predicción inversa. Este indicador es 1 si el macrobloque de referencia ha sido objeto de predicción inversa. Con el formato de longitud variable, la información de registro histórico puede reducirse para disminuir la tasa binaria para transmisiones.

50 Es decir, si se transfieren las funciones `macroblock_type` y `motion_vectors()` y no el `quantiser_scale_code`, la tasa binaria puede disminuirse estableciendo el código `slice_quantiser_scale_code` a "00000".

Además, si solamente se transfiere el `macroblock_type` y no la función `motion_vectors()`, el `quantiser_scale_code` y el `dct_type`, la tasa binaria puede disminuirse utilizando "no codificado" como el tipo de macrobloque.

55 Además, si solamente se transfiere `picture_coding_type` y no `slice()` e información posterior, la tasa binaria puede reducirse utilizando los `picture_data()` que no tienen el código `slice_start_code`.

60 En la descripción anterior, para evitar 23 bits continuos de "0" que aparezcan en los datos de usuarios, `user_data`, se inserta "1" en los datos a los de 22 bits, pero el intervalo no puede ser de 22 bits. Además, en lugar de contar el número de 0 continuos, se puede examinar `Byte_align` para la inserción.

Además, aunque el MPEG prohíbe la presencia de 23 bits continuos de "0" realmente se trata de 23 bits continuos de 0 comenzando en la cabecera del byte y no con dichos bits comenzando en la parte intermedia del byte. De este modo, se inserta "1" en posiciones distintas a LSBs, a modo de ejemplo, en intervalos de 24 bits.

65 Además, en la descripción anterior, la información de registro histórico está en una forma similar a flujos elementales de vídeo, pero puede estar prácticamente en la forma de flujos elementales en paquetes o flujos de transporte.

Además, aunque el `user_data` en el flujo elemental precede a `picture_data`, se puede localizar en cualquier otro lugar.

El transcodificador 101 descrito en la Figura 15 proporciona el proceso posterior con parámetros de codificación para una pluralidad de generaciones como información de registro histórico, pero no se requiere toda la información de registro histórico anteriormente descrita. A modo de ejemplo, si este transcodificador está seguido por un sistema de registro y reproducción que incluye un soporte de registro de gran capacidad que esté relativamente libre de limitaciones sobre la capacidad de almacenamiento, no se produce ningún problema si toda la información anteriormente descrita se describe en los parámetros de codificación. Si, sin embargo, el transcodificador es seguido por un sistema de registro y reproducción que incluye un soporte de registro de una capacidad relativamente pequeña, solamente la información de registro histórico requerida, se describe, de forma deseable, en los parámetros de codificación en lugar de toda la información de registro histórico, con el fin de reducir más o menos la tasa de datos de flujos codificados. A modo de otro ejemplo, si este transcodificador es seguido por una ruta de transmisión que incluye un soporte de registro capacidad de transmisión grande que esté relativamente libre de limitaciones sobre la capacidad de transmisión, no se produce ningún problema si toda la información anteriormente descrita se describe en los parámetros de codificación. Si, sin embargo, el transcodificador está seguido por una ruta de transmisión que tiene una capacidad de transmisión relativamente pequeña, solamente la información de registro histórico es deseablemente descrita en los parámetros de codificación en lugar de toda la información de registro histórico, con el fin de reducir más o menos la tasa de datos de flujos codificados.

La presente invención está caracterizada por cuanto que la información de registro histórico requerida para cada una de las diversas aplicaciones después del dispositivo de transmisión se describe en el flujo codificado de forma adaptativa y selectiva, dependiendo de la aplicación. Para conseguirlo, esta forma de realización describe la información de `re-coding_stream_info` en el flujo codificado.

Elementos de datos y de sintaxis para la `re-coding_stream_info` se describirán a continuación en detalle haciendo referencia a la Figura 68.

Según se ilustra en la Figura 68, la función `re_coding_stream_info()` está constituida por `user_data_start_code`, `re_coding_stream_info_ID`, `red_bw_flag`, `red_bw_indicator`, `marker_bit`, `num_other_bits`, `num_mv_bits`, `num_coef_bits`, etc.

El `user_data_start_code` es un código de indicativo del comienzo de los datos de usuarios `user_data`. El `re_coding_stream_info_ID` es un número entero de 16 bits utilizado para identificar la función `re_coding_stream_info()`. Más concretamente, este número entero tiene un valor de "1001 0001 1110 1100" (0x91ec).

El `red_bw_flag` 0 es un indicador de 1 bit que es 0 si se transmiten los parámetros de codificación para toda la información de registro histórico y que es 1 si se transmiten, de forma selectiva, los parámetros de codificación para la información de registro histórico. Más concretamente, según se ilustra en la Figura 69, si el indicador `red_bw_flag` es 1, la comprobación de `red_bw_indicator` después de este indicador permite la determinación de cuál de los cinco conjuntos de datos se utiliza para transmitir los parámetros de codificación correspondientes para la información de registro histórico. Este conjunto de datos contiene información para determinar una combinación de parámetros de codificación a transmitirse con el flujo codificado recodificado. En consecuencia, los parámetros de codificación a describirse en el flujo codificado se seleccionan en conformidad con este conjunto de datos.

El `red_bw_indicator` es un número entero de 2 bits que actúa como un indicador para definir un conjunto de datos para los parámetros de codificación. Más concretamente, el indicador `red_bw_indicator` son datos que indican cuál de los conjuntos de datos 2 a 5 se representan según se ilustra en la Figura 69.

De este modo, haciendo referencia a `red_bw_flag` y a `red_bw_indicator`, que se describen en el flujo codificado, puede determinarse cuál de los cinco conjuntos de datos se utiliza para transmitir los parámetros de codificación para la información de registro histórico.

A continuación, los parámetros de codificación para la información de registro histórico que se transmiten en cada conjunto de datos se explicarán haciendo referencia a la Figura 70.

La información del registro histórico puede clasificarse, de forma aproximada, en información en imágenes e información en macrobloques. La información en segmentos se obtiene recogiendo información de macrobloques contenida en dicha información. La información en GOPs se obtiene recogiendo la información en imágenes contenidas en dicha información.

Puesto que la información en imágenes se transmite solamente una vez por trama, su tasa binaria no es tan alta en comparación con la información de registro histórico insertada en el flujo codificado. Por el contrario, puesto que la información en macrobloques se transmite para cada macrobloque, si, a modo de ejemplo, en un sistema de vídeo con 525 líneas de barrido para una trama y con una tasa de campos de 60/segundos, 720 x 480 pixels están

contenidos en una sola trama, entonces la información en macrobloques debe transmitirse 1,350  $(=(720/16) \times (480/16))$  veces por trama. En consecuencia, una parte relativamente grande de la información de registro histórico está ocupada por la información en macrobloques.

5 En consecuencia, en esta forma de realización, puesto que la información de registro histórico insertada en el flujo codificado, al menos la información en imágenes está constantemente transmitida pero la información en macrobloques se transmite, de forma selectiva, dependiendo de la aplicación, con ello se reduce la información transmitida.

10 Según se ilustra en la Figura 70, la información en macrobloques transmitida como la información de registro histórico incluye, a modo de ejemplo, los elementos num\_coef\_bits, num\_mv\_bits, num\_other\_bits, q\_scale\_code, q\_scale\_type, motion\_type, mv\_vert\_field\_sel [], mv[] [], mb\_mfwd, mb\_mbwd, mb\_pattern, coded\_block\_pattern, mb\_intra, slice\_start, dct\_type, mb\_quant, skipped\_mb, etc. Esta información se representa utilizando el elemento de macroblock\_rate\_information definido en SMPTE-327M.

15 Los num\_coef\_bits representan la cantidad de códigos de macrobloques que se requiere para los coeficientes DCT. Los num\_mv\_bits representan la cantidad de códigos de macrobloques requerida para vectores de movimiento. Los num\_other\_bits representan la cantidad de códigos de macrobloques distintos de los num\_coef\_bits y los num\_mv\_bits.

20 El código q\_scale\_code representa un código de escala q aplicado al macrobloque. El motion\_type representa el tipo de los vectores de movimiento aplicados al macrobloque. El mv\_vert\_field\_sel[][] representa una selección de campos para los vectores de movimiento aplicados al macrobloque.

25 El mv[][][] representa los vectores de movimiento aplicados al macrobloque. El mb\_mfwd es un indicador de que el modo de predicción para el macrobloque es predicción directa. El mb\_mbwd es un indicador de que el modo de predicción para el macrobloque es predicción inversa. El mb\_pattern es un indicador de la presencia de un coeficiente DC no nulo del macrobloque.

30 El coded\_block\_pattern es un indicador de la presencia de un coeficiente DC no nulo del macrobloque para cada bloque DCT. El mb\_intra es un indicador de si el macrobloque es un intra\_macro o no lo es. El slice\_start es un indicador de si el macrobloque es, o no, la cabecera de un segmento. El dct\_type es un indicador de si el macrobloque es el dct de campo el dct de trama.

35 El mb\_quant es un indicador de si el macrobloque transmite, o no, el código quantiser\_scale\_code. El skipped\_mb es un indicador de si el macrobloque es, o no, un macrobloque con salto operativo.

40 No todos estos parámetros de codificación son constantemente requeridos, pero los parámetros de codificación requeridos varían dependiendo de la aplicación después del transcodificador. A modo de ejemplo, los parámetros de codificación tales como los num\_coef\_bits y slice\_start se requieren para aplicaciones que implican una demanda transparente que requieren un flujo de bits a recuperarse a su forma original lo más próximo posible durante la recodificación. La "demanda transparente" permite que se genere un flujo de bits de salida con la característica de evitarse la degradación de la calidad de imagen en comparación con un flujo de bits de entrada.

45 Es decir, las aplicaciones que demandan solamente un cambio en la tasa binaria para un proceso de transcodificación no requieren los parámetros de codificación tales como los num\_coef\_bits y el slice\_start. Además, si existen limitaciones muy estrictas sobre la ruta de transmisión, algunas aplicaciones requieren solamente el tipo de codificación de cada imagen.

50 Considerando estas circunstancias, esta forma de realización proporciona un conjunto tal como se ilustra en la Figura 70, como un ejemplo de un conjunto de datos para parámetros de codificación que se utilizan en la transmisión de la información de registro histórico.

55 En la Figura 70, el valor "2" correspondiente a parámetros de codificación en cada conjunto de datos significa que la información correspondiente está presente en el flujo codificado y disponible como la información de registro histórico y "0" significa que la información correspondiente está ausente del flujo codificado. "1" significa que la información correspondiente está presente para soporte para la presencia de otra información o para la sintaxis, pero no tiene ningún significado, a modo de ejemplo, ninguna relación con la información de flujo de bits origen. A modo de ejemplo, el slice\_start es "1" para el macrobloque en la cabecera de un segmento para transmitir información de registro histórico pero no tiene ningún significado como información de registro histórico si el segmento no mantiene necesariamente la misma relación locacional con el flujo de bits original.

60 En esta forma de realización, los parámetros de codificación (num\_coef\_bits, num\_mv\_bits, num\_other\_bits), (q\_scale\_code, q\_scale\_type), (motion\_type, mv\_vert\_field\_sel[], mv[][]), (mb\_mfwd, mb\_mbwd), (mb\_pattern), (coded\_block\_pattern), (mb\_intra), (slice\_start), (dct\_type), (mb\_quant), y (skipped\_mb) se describen de forma selectiva, en flujo codificado dependiendo del conjunto de datos seleccionado.

El conjunto de datos 1 está previsto para reconstruir un flujo de bits completamente transparente. El conjunto de datos 1 permite una transcodificación exacta que proporciona un flujo de bits con pequeña degradación de la calidad de imagen en comparación con un flujo de bits de entrada. El conjunto de datos 2 está también previsto para reconstruir un flujo de bits completamente transparente. El conjunto de datos 3 no puede reconstruir un flujo de bits completamente transparente pero reconstruye un flujo de bits bastante transparente desde el punto de vista visual. El conjunto de datos 4 es inferior al conjunto de datos 3 en términos de transparencia, pero permite la reconstrucción de un flujo de bits sin ningún problema visual. El conjunto de datos 5 es inferior al conjunto de datos 4 en términos de transparencia pero puede reconstruir un flujo de bits con una poca información de registro histórico a pesar de la falta de integridad de la reconstrucción.

Estos conjuntos de datos son funcionalmente de más alto nivel puesto que sus números de conjunto de datos son más pequeños, pero los conjuntos de datos de nivel funcional más alto requieren una mayor capacidad para la transmisión de información de registro histórico. En consecuencia, el conjunto de datos transmitido se determina tomando en consideración la aplicación asumida y la capacidad disponible para la información de registro histórico.

De los cinco conjuntos de datos ilustrados en la Figura 70, para el conjunto de datos 1, el indicador `red_bw_flag` es 0, mientras que para los conjuntos de datos 2 a 5, el indicador `red_bw_flag` es 1. Por el contrario, el `red_bw_indicator` es 0 para el conjunto de datos 2, 1 para el conjunto de datos 3, 2 para el conjunto de datos 4 y 3 para el conjunto de datos 5.

De este modo, el `red_bw_indicator` se especifica si el indicador `red_bw_flag` es 1 (es decir, para los conjuntos de datos 2 a 5).

Además, si el indicador `ref_bw_flag` es 0 (el conjunto de datos 1), los elementos denominados `marker_bit`, `num_other_bits`, `num_mv_bits`, y `num_coef_bits` se describen para cada macrobloque. Estos cuatro elementos de datos no se describen en el flujo codificado para los conjuntos de datos 2 a 5 (es decir, si el indicador `red_bw_flag` es 1).

Para el conjunto de datos 5, los otros elementos de sintaxis no se transmiten incluyendo la función `picture_data()` (véase la Figura 59). Es decir, ningún parámetro de codificación se transmite para una pluralidad de funciones `slice()` contenidos en la función `picture_data()`. En consecuencia, para el conjunto de datos 5, la información de registro histórico seleccionada está prevista para transmitir solamente parámetros de codificación en imágenes tales como el tipo de imagen `picture_type`.

Para los conjuntos de datos 1 a 4, parámetros de codificación están presentes para la pluralidad de funciones `slice()` contenidas en la función `picture_data()`. Sin embargo, la información de dirección para segmentos que se determina por la función `slice()` y para segmentos en el flujo de bits origen dependen del conjunto de datos seleccionado. Para el conjunto de datos 1 o 2, la información de dirección para los segmentos en el flujo de bits origen para la información de registro histórico debe ser idéntica a la información de dirección para los segmentos determinados por la función `slice()`.

Los elementos de datos de sintaxis de los elementos de la función `macroblock()` (véase la Figura 61) dependen del conjunto de datos seleccionado. Las funciones `macroblock_escape`, `macroblock_adres_increment` y `macroblock_modes()` están constantemente presentes en el flujo codificado. La eficacia de `macroblock_escape` y `macroblock_adres_increment` como información, sin embargo, se determina por el conjunto de datos seleccionado. Si los conjuntos de datos 1 o 2 se utiliza para los parámetros de codificación para la información de registro histórico, debe transmitirse una información idéntica a la `skipped_mb` en el flujo de datos bits origen.

Para el conjunto de datos 4, la función `motion_vectors()` está ausente del flujo codificado. Para los conjuntos de datos 1 a 3, el tipo de `macroblock_type` de la función `macroblock_modes()` determina la presencia de la función `motion_vectors()` en el flujo codificado. Para los conjuntos de datos 3 o 4, la función `coded_block_pattern()` está ausente del flujo codificado. Para los conjuntos de datos 1 y 2, el `macroblock_type` de la función `macroblock_modes()` determina la presencia de la función `coded_block_pattern()` en el flujo codificado.

Los elementos sintaxis de la función `macroblock_modes()` (véase la Figura 62) depende del conjunto de datos seleccionado. El `macroblock_type` existe constantemente. Para el conjunto de datos 4, los elementos `frame_motion_type`, `field_motion_type`, y `dct_type` están ausentes del flujo codificado.

La eficacia como información de un parámetro obtenido a partir del `macroblock_type` se determina por el conjunto de datos seleccionado.

Para los conjuntos de datos 1 o 2, el `macroblock_quant` debe ser el mismo que en el flujo de bits origen. Para los conjuntos de datos 3 o 4, el `macroblock_quant` representa la presencia del `quantiser_scale_code` en la función `macroblock()` y no necesita ser el mismo que en el flujo de bits origen.

Para los conjuntos de datos 1 a 3, la macroblock\_motion\_forward debe ser la misma que el flujo de bits origen. Esto no es necesario para los conjuntos de datos 4 o 5.

5 Para el conjunto de datos 1 o 2, el macroblock\_pattern debe ser el mismo que en el flujo de bits origen. Para el conjunto de datos 3, el macroblock\_pattern se utiliza para indicar la presencia del dct\_type. Las relaciones para los conjuntos de datos 1 a 3 no se establecen para el conjunto de datos 4.

10 Si los conjuntos de datos 1 a 3 se utilizan para los parámetros de codificación para la información de registro histórico, el macroblock\_intra debe ser el mismo que en el flujo de bits origen. Esto no se aplica al conjunto de datos 4.

15 A continuación, se describirá un procesamiento ejecutado por el transcodificador 101 sobre un flujo codificado que contiene información sobre los conjuntos de datos y para generar un flujo codificado sobre la base de los conjuntos de datos establecidos haciendo referencia al transcodificador 101 ilustrado en la Figura 15.

El ejemplo de un transcodificador ilustrado en la Figura 15 recibe un flujo codificado ST (3rd) generado por el proceso de codificación de la tercera generación para convertir la estructura GOP y/o su tasa binaria con el fin de generar un nuevo flujo codificado ST(4th).

20 En primer lugar, el dispositivo de decodificación 102 extrae desde el flujo codificado de la tercera generación St(3rd) parámetros de codificación de la tercera generación utilizados para codificar este flujo codificado St(3rd) y decodifica el flujo codificado ST(3rd) sobre la base de los parámetros de codificación extraídos para generar una señal de vídeo de banda base. Además, el dispositivo de decodificación 102 proporciona los parámetros de codificación de la tercera generación (3th) al dispositivo de multiplexación de información de registro histórico 103. Además, el dispositivo de decodificación 102 extrae el user\_data() desde la capa de imágenes del flujo codificado de la tercera generación ST(3rd) para suministrar los datos al dispositivo de decodificación de registro histórico 104.

30 El dispositivo de decodificación de registro histórico 104 extrae el history\_stream() desde el user\_data() suministrado por el dispositivo de decodificación 102. Puesto que el history\_stream() es un flujo constituido por elementos de datos codificados de longitud variable, el dispositivo de decodificación de registro histórico 104 realiza un proceso de decodificación de longitud variable en el flujo history\_stream(). En consecuencia, se puede generar un flujo que consiste en elementos de datos que tiene cada uno una longitud de datos predeterminada. A continuación, el dispositivo de decodificación de registro histórico 104 realiza un análisis de una sintaxis para el flujo de datos decodificados de longitud variable. El análisis sintáctico se refiere a la interpretación de la sintaxis para el flujo.

35 Durante el proceso de análisis sintáctico, el dispositivo de decodificación de registro histórico 104 hace referencia a red\_bw\_flag y red\_bw\_indicator que se describen en la re\_coding\_stream\_info() en la función history\_stream(). Haciendo referencia a los indicadores red\_bw\_flag y red\_bw\_indicator extraídos del flujo, el dispositivo de decodificación de registro histórico 104 determina cuáles de los cinco conjuntos de datos se establecen para history\_stream() recibido. De este modo, determinando el tipo del conjunto de datos a partir de los indicadores red\_bw\_flag y red\_bw\_indicator, el dispositivo de decodificación de registro histórico 104 puede determinar qué parámetros de codificación están contenidos en el history\_stream().

45 Más concretamente, si el red\_bw\_flag = 0, se establece el conjunto de datos 1. De este modo todos los parámetros de codificación se describen en el history\_stream () como la función picture\_data(), incluyendo los elementos num\_coef\_bits, num\_mv\_bits, num\_other\_bits, q\_scale\_code, q\_scale\_type, motion\_type, mv\_vert\_field\_sel [], mv [] [], mb\_mfwd, mb\_mbwd, mb\_pattern, coded\_block\_pattern, mb\_intra, slice\_start, dct\_type, mb\_quant y skipped\_mb.

50 Si el red\_bw\_flag = 1 y el red\_bw\_indicator = 0, se establece el conjunto de datos 2. En consecuencia, los parámetros de codificación se describen en el history\_stream() como la función picture\_data() incluyendo los elementos de datos q\_scale\_code, q\_scale\_type, motion\_type, mv\_vert\_field\_sel[], mv[][], mb\_mfwd, mb\_mbwd, mb\_pattern, coded\_block\_pattern, mb\_intra, slice\_start, dct\_type, mb\_quant, y skipped\_mb.

55 Si el red\_bw\_flag = 1 y el red\_bw\_indicator = 1, se establece el conjunto de datos 3. En consecuencia, los parámetros de codificación se describen en el history\_stream() como la función picture\_data(), incluyendo los elementos q\_scale\_code, q\_scale\_type, motion\_type, mv\_vert\_field\_sel[], mv[][], mb\_mfwd, mb\_mbwd, themb\_pattern, iab\_intra, slice\_start, dct\_type, mb\_quant, y skipped\_mb.

60 Si el red\_bw\_flag = 1 y el red\_bw\_indicator = 2, se establece el conjunto de datos 4. En consecuencia, los parámetros de codificación que incluyen el q\_scale\_code y el q\_scale\_type se describen en el history\_stream() como la función picture\_data().

65 Si el red\_bw\_flag = 1 y el red\_bw\_indicator = 3, se establece el conjunto de datos 5. En consecuencia, no se describe ningún parámetro de codificación en el history\_stream() como la función picture\_data().

5 El dispositivo de decodificación de registro histórico 104 hace referencia a la información en el `red_bw_flag` y `red_bw_indicator` para extraer los parámetros de codificación desde el `history_stream()`. En la forma de realización del transcodificador ilustrado en la Figura 15, el flujo codificado de entrada suministrado al transcodificador del dispositivo de decodificación de registro histórico 104 ha sido generado por el proceso de codificación de la tercera generación, de modo que la información de registro histórico de salida es los parámetros de codificación de la primera generación y de la segunda generación.

10 El dispositivo de multiplexación de información de registro histórico 103 multiplexa los parámetros de codificación de la tercera generación (3th) suministrados por el dispositivo de decodificación 102 y los parámetros de codificación anteriores (1st, 2nd) suministrados por el dispositivo de decodificación de registro histórico 104, en los datos de vídeo de banda base suministrados por el dispositivo de decodificación 102, en conformidad con dicho formato según se ilustra en las Figuras 68 a 73.

15 El dispositivo de separación de información de registro histórico 105 recibe los datos de vídeo de banda base suministrados por el dispositivo de multiplexación de información de registro histórico 103 y extrae los parámetros de codificación de las primera, segunda y tercera generación (1st, 2nd, 3rd) desde los datos de vídeo de banda base para suministrarlos al dispositivo de codificación 106.

20 El dispositivo de codificación 106 recibe los datos de vídeo de banda base y los parámetros de codificación (1st, 2nd, 3rd) desde el dispositivo de separación de información de registro histórico 105 para recodificar los datos de vídeo de banda base sobre la base de los parámetros de codificación recibidos. En este caso, el dispositivo de codificación 106 selecciona parámetros de codificación óptimos para el proceso de codificación, a partir de los parámetros de codificación (1st, 2nd, 3rd) generados durante los procesos de codificación anteriores y los parámetros de codificación recientemente generados a partir de los datos de vídeo de banda base suministrados. El dispositivo de codificación 106 realiza el "proceso de codificación normal" anteriormente descrito si utiliza, para la codificación, los parámetros de codificación recientemente generados a partir de los datos de vídeo de banda base suministrados. El dispositivo de codificación 106 realiza el "proceso de codificación de parámetros reutilizados" anteriormente descrito si utiliza cualquiera de los parámetros de codificación anteriores (1st, 2nd, 3rd).

30 El ordenador 100 proporcionado en la red controla el proceso de decodificación ejecutado por el dispositivo de decodificación 102 y el proceso de codificación ejecutado por el dispositivo de codificación 106. A modo de ejemplo, el ordenador 100 detecta la capacidad de la ruta de transmisión para transmitir una salida de flujo codificado por el dispositivo de codificación 106 y selecciona uno adecuado de entre los cinco conjuntos de datos dependiendo de la capacidad de transmisión. Asimismo, detecta la capacidad de almacenamiento de un dispositivo conectado a una salida del dispositivo de codificación 106 y selecciona un conjunto adecuado de entre los cinco conjuntos de datos dependiendo de la capacidad de almacenamiento.

40 El dispositivo de codificación 106 recibe información indicativa del conjunto de datos desde el ordenador 100 para generar el `red_bw_flg` y el `red_bw_indicator` sobre la base de esta información. Si la información proporcionada por el ordenador 100 indica el conjunto de datos 1, el `red_bw_flg` = 0. Si la información indica el conjunto de datos 2, el `red_bw_flg` = 1 y el `red_bw_indicator` = 0. Si la información indica el conjunto de datos 3, el `red_bw_flg` = 1 y el `red_bw_indicator` = 1. Si la información indica el conjunto de datos 4, el `red_bw_flg` = 1 y el `red_bw_indicator` = 2. Si la información indica el conjunto de datos 5, el `red_bw_flg` = 1 y el `red_bw_indicator` = 3.

45 Dependiendo de los valores determinados del `red_bw_flg` y del `red_bw_indicator`, el dispositivo de codificación 106 selecciona parámetros de codificación para la descripción en el flujo codificado como el `history_stream()` y describe los parámetros de codificación seleccionados en el flujo codificado como el `history_stream()`, mientras que describe el `red_bw_flg` y el `red_bw_indicator` en el flujo codificado como la `re_coding_stream_info()`. Los parámetros de codificación transmitidos como el `history_stream()` se seleccionan para cada una de las primera, segunda y tercera generación de parámetros de codificación.

50 Si el `red_bw_flg` = 0, entonces, el dispositivo de codificación 106 describe todos los parámetros de codificación en el flujo codificado como la función `picture_data()`, incluyendo los elementos `num_coef_bits`, `num_mv_bits`, `num_other_bits`, `q_scale_code`, `q_scale_type`, `motion_type`, `mv_vert_field_sel`[], `mv`[], `mb_mfwd`, `mb_mbwd`, `mb_pattern`, `encoded_block_pattern`, `mb_intra`, `slice_start`, `dct_type`, `mb_quant`, y `skipped_mb`.

60 Si el `red_bw_flg` = 1 y el `red_bw_indicator` = 0, el dispositivo de codificación 106 describe el parámetro de codificación en el `history_stream()` como la función `picture_data()`, incluyendo los elementos `q_scale_code`, `q_scale_type`, `motion_type`, `mv_vert_field_sel` [], `mv` [] [], `mb_mfwd`, `mb_mbwd`, `mb_pattern`, `encoded_block_pattern`, `mb_intra`, `slice_start`, `dct_type`, `mb_quant` y `skipped_mb`.

65 Si el `red_bw_flg` = 1 y `red_bw_indicator` = 1, el dispositivo de codificación 106 describe parámetros de codificación en el `history_stream()` como la función `picture_data()` incluyendo los elementos `q_scale_code`, `q_scale_type`, `motion_type`, `mv_vert_field_sel` [], `mv` [], `mb_mfwd`, `mb_mbwd`, `mb_pattern`, `mb_intra`, `slice_start`, `dct_type`, `mb_quant` y `skipped_mb`.



Si el `red_bw_flag` = 1 y el `red_bw_indicator` = 2, el dispositivo de codificación 106 describe parámetros de codificación que incluyen el `q_scale_code` y el `q_scale_type`, en el `history_stream()` como la función `picture_data()`.

5 Si el `red_bw_flag` = 1 y el `red_bw_indicator` = 3, el codificador 106 no describe ningún parámetro de codificación en el `history_stream()` como la función `picture_data()`.

10 Es decir, el dispositivo de codificación 106 selecciona de entre los parámetros de codificación transmitidos como el `history_stream()` sobre la base de la información que indica el conjunto de datos especificado por el ordenador 100, en lugar de transmitir todos los parámetros de codificación anteriores suministrados como el `history_stream()`. De este modo, sobre la base de la información recibida que indica el conjunto de datos, el dispositivo de codificación 105 puede generar un flujo codificado que contiene `history_stream()` con varias capacidades de datos. Además, el dispositivo de codificación 105 puede generar un flujo codificado que contiene `history_stream()` que está constituido por una cantidad de datos adecuada para la capacidad de transmisión de un soporte de transmisión después del dispositivo de codificación 105, para la capacidad de almacenamiento de un soporte de registro o para la aplicación.

15 En conformidad con el transcodificador de esta forma de realización, los parámetros de codificación transmitidos como el `history_stream()` de este modo se seleccionan en conformidad con una aplicación que sigue, de forma conectiva, el dispositivo de codificación. Por lo tanto, el registro histórico correspondiente a la aplicación puede transmitirse en una cantidad de datos óptima.

20 A continuación, un formato en el que se multiplexa la información de registro histórico en la salida de señal de vídeo de banda base por el dispositivo de multiplexación de información de registro histórico 103 se describirá haciendo referencia a las Figuras 71 a 74.

25 La Figura 71 es un diagrama que ilustra el formato "Re\_Coding information Bus macroblock format" en donde la información de registro histórico se multiplexa en la señal de vídeo de banda base para su transmisión. Este bloque de máscara está configurado por 16 x 16 (=256) bits. En la Figura 71, los 32 bits ilustrados en cada una de las tercera y cuarta filas desde la parte superior son `picrate_element`. Los elementos de tasa de imágenes, ilustrados en las Figuras 72 a 74, se describen en el `picrate_element`. El `red_bw_flag` de 1 bit se especifica en la segunda fila de la parte superior en la Figura 72 y el `red_bw_indicator` de 3 bits se especifica en la tercera fila desde la parte superior.

30 Es decir, los indicadores `red_bw_flag` y `red_bw_indicator` se transmiten como el `picrate_element` en la Figura 71.

Los otros datos en la Figura 71 se explicarán a continuación. `SRIB_sync_code` es un código que representa que la primera fila de un macrobloque en este formato está alineada como el extremo izquierdo del flujo. Más concretamente, este código se establece en "11111". Si la `picture_structure` indica la estructura de imagen de trama (es decir, su valor es "11"), `fr_fl_SRIB` se establece a 1, lo que indica que se transmiten más de 16 líneas del macrobloque Re\_Coding Information Bus. Si la `picture_structure` no indica la estructura de tramas, el `fr_fl_SRIB` se establece a 0, lo que indica que se transmiten más de 16 líneas del Re\_Coding Information Bus. Este mecanismo permite que el Re\_Coding Information Bus quede bloqueado para los pixels correspondientes en un campo o trama de vídeo que se decodifica de forma espacial y temporal.

40 `SRIB_top_field_first` se establece al mismo valor que `top_field_first` mantenido en el flujo de bits origen y representa con `repeat_first_field`, la alineación temporal del Re\_Coding Information Bus para un vídeo relacionado. `SRIB_repeat_field_first` se establece al mismo valor que `repeat_first_field` mantenido en el flujo de bits origen. El contenido del Re\_Coding Information Bus para un primer campo debe repetirse según se indica por este indicador.

45 `422_420_chroma` representa si el flujo de bits origen es del tipo 4:2:2 o 4:2:0. El `422_420_chroma` de 0 representa que un flujo de bits es de 4:2:0 y que una señal de aberración cromática ha sido muestreada en sentido ascendente de modo que se proporcione a la salida un vídeo de tipo 4:2:2. El `422_420_chroma` de 1 representa que no se ha filtrado la señal de aberración cromática.

50 `rolling_SRIB_mb_ref` representa un módulo de 16 bits 65521, cuyo valor se aumenta constantemente con el número de macrobloques. Este valor debe ser continuo a través de la tramas de la estructura de imagen de trama. De no ser así, debe ser continuo a través de campos. Se inicializa a un valor predeterminado entre 0 y 65520. Esto permite que se incorpore un indicador único del Re\_Coding Information Bus en un sistema de registro.

55 El significado de los otros datos para el macrobloque de Re\_Coding Information Bus es según se describió con anterioridad y por ello aquí se omite.

60 Según se ilustra en la Figura 75, los datos de 256 bits para el Re\_Coding Information Bus en la Figura 71 están dispuestos, en la forma de un bit cada vez, en `Cb [0] [0]`, `Cr [0] [0]`, `Cb [1] [0]` y `Cr [1] [0]`, que son LSBs de datos de aberración cromática. Puesto que el formato ilustrado en la Figura 75 permiten que se envíen datos de 4 bits, los datos de 256 bits en la Figura 71 pueden transmitirse enviando el formato en la Figura 75 sesenta y cuatro veces (=256/4).

65 En conformidad con el transcodificador de la presente invención, los parámetros de codificación generados durante los procesos de codificación anteriores se reutilizan para el proceso de codificación en curso, con lo que se impide la

degradación de la calidad de imagen a pesar de la repetición de los procesos de decodificación y codificación. Es decir, la presente invención puede restringir la acumulación de degradación de la calidad de imagen que se origina por la repetición de los procesos de decodificación y codificación.

- 5 Las Figuras 76 y 77 ilustran ejemplos de configuraciones de una grabadora de cinta de vídeo con el transcodificador de la presente invención aplicado. La Figura 76 ilustra un ejemplo de una configuración de un sistema de registro de una grabadora de cinta de vídeo 601. La Figura 77 ilustra un ejemplo de una configuración de un sistema de reproducción de la grabadora de cinta de vídeo 601.
- 10 La grabadora de cinta de vídeo 601 ilustrada en la Figura 76 está configurada por un transcodificador 101R, un dispositivo de codificación de canal 602 y un cabezal de registro 603. El transcodificador 101R está configurado básicamente en la misma manera que el transcodificador ilustrado en la Figura 37. En este ejemplo de configuración, el transcodificador 101R convierte un flujo de bits ST de un GOP largo en un grupo GOP corto.
- 15 Un flujo codificado de la cuarta generación ST procedente del dispositivo de codificación 106 del transcodificador 101R se suministra al dispositivo de codificación de canal 602. Según se describió con anterioridad, los datos de usuarios `user_data` que contienen los parámetros de codificación de la primera a la tercera generación se registran en la zona de datos de usuario de la capa de imágenes del flujo codificado de la cuarta generación ST.
- 20 El dispositivo de codificación de canal 602 añade un código de paridad para la corrección de errores al flujo codificado de la cuarta generación a la entrada que realiza posteriormente una codificación de canal utilizando, a modo de ejemplo, el método de modulación NRZI y luego, suministra el flujo codificado a un cabezal de registro 603. El cabezal de registro 603 registra el flujo codificado de entrada en una cinta magnética 604.
- 25 Según se ilustra en la Figura 77, en el sistema de reproducción, cabezal de reproducción 611 genera una señal desde la cinta magnética 604 para suministrarla a un dispositivo de decodificación de canal 612. El dispositivo de decodificación de canal 612 realiza una decodificación de canal de la señal suministrada por el cabezal de reproducción 611 y luego, corrige los errores presentes utilizando la paridad.
- 30 El flujo codificado de la cuarta generación ST proporcionado por el dispositivo de decodificación de canal 612 se aplica a la entrada de un transcodificador 101P. El transcodificador 101P tiene una configuración básica similar a la del transcodificador ilustrado en la Figura 37.
- 35 El dispositivo de decodificación 102 del transcodificador 101P extrae desde el flujo codificado de la cuarta generación, los datos de usuario `user_data` que contienen los primeros parámetros de codificación de la primera a tercera generación y suministra los datos al dispositivo de codificación de registro histórico 104 y al dispositivo de codificación 106. El dispositivo de decodificación de registro histórico 104 decodifica los datos de usuario `user_data` para entregar los parámetros de codificación de primera a tercera generación obtenidos al dispositivo de codificación 106.
- 40 El dispositivo de decodificación 102 decodifica también el flujo codificado de la cuarta generación ST para proporcionar, a la salida, la señal de vídeo de banda base y los parámetros de codificación de la cuarta generación. La señal de vídeo de banda base se suministra al dispositivo de codificación 106, mientras que los parámetros de codificación de la cuarta generación se entregan al dispositivo de codificación 106 y al dispositivo de codificación de registro histórico 107.
- 45 El dispositivo de codificación de registro histórico 107 convierte los parámetros de codificación de la cuarta generación de entrada en los datos de usuario `user_data` para suministrar los datos al dispositivo de codificación 106.
- 50 Según se describió con anterioridad, el dispositivo controlador 70 para el dispositivo de codificación 106 determina si el tipo de imagen de cada imagen determinada a partir de la estructura de grupo GOP especificada por el operador es el mismo que el tipo de imagen contenido en la información de registro histórico (datos de usuario `user_data`). Dependiendo del resultado de la determinación, el dispositivo de codificación 106 realiza el "proceso de codificación normal" o "proceso de codificación de parámetros reutilizados" anteriormente descritos. Después de este proceso, el dispositivo de codificación 106 proporciona el flujo codificado de la cuarta generación ST con el GOP largo en el que se ha convertido el grupo GOP corto. Los datos de usuario `user_data` en el flujo codificado ST tienen los parámetros de codificación de la primera a cuarta generación allí registrado como información de registro histórico.
- 55 Aunque la grabadora de cinta de vídeo 601 ilustrada en las Figuras 76 y 77 registra la información de registro histórico en los datos de usuarios `user_data` de la capa de imágenes, la información de registro histórico puede registrarse en una zona de la cinta magnética 604 que es diferente de una zona para datos de vídeo. Las Figuras 78 y 79 muestran ejemplos de configuraciones de la grabadora de cinta de vídeo 601 en este caso. La Figura 78 ilustra un ejemplo de una configuración de un sistema de registro de la grabadora de cinta de vídeo 601. La Figura 79 ilustra un ejemplo de una configuración de un sistema de reproducción de la grabadora de cinta de vídeo 601.
- 60
- 65

Según se ilustra en la Figura 78, en la grabadora de cinta de vídeo 601, los datos de usuario `user_data` proporcionados por el dispositivo de decodificación 102 del transcodificador 101R es objeto de entrada en el dispositivo de decodificación de registro histórico 104, que decodifica luego los parámetros de codificación anteriores (en este ejemplo, los parámetros de codificación de la primera y segunda generación) para su suministro al dispositivo de codificación 106. Además, este ejemplo no necesita registrar información de registro histórico en la cinta magnética 604 como datos de usuario `user_data` y por ello, emplea solamente el VLC de registro histórico 211 en lugar del dispositivo de codificación de registro histórico completo 107 ilustrado en la Figura 15. El VLC de registro histórico 211 se suministra con los parámetros de codificación (en este caso, los parámetros de codificación de la tercera generación) proporcionados por el dispositivo de decodificación 102 y con los parámetros de codificación (en este caso, los parámetros de codificación de la primera y segunda generación) a la salida después de decodificarse desde los datos de usuario `user_data` por el dispositivo de decodificación de registro histórico 104. El VLC de registro histórico 211 de longitud variable codifica los parámetros de codificación de la primera a tercera generación para generar el flujo de registro histórico ilustrado en las Figuras 40 a 46 o 47 para su suministro a un dispositivo de multiplexación 621.

El dispositivo de multiplexación 621 recibe también un flujo codificado de la cuarta generación ST procedente del dispositivo de codificación 106. El dispositivo de multiplexación 621 multiplexa el flujo codificado (flujo de bits) suministrado por el dispositivo de codificación 106 en una zona que es más segura que una zona en la que se registran los datos históricos suministrados por el VLC de registro histórico 211.

A modo de ejemplo, según se ilustra en la Figura 80, la salida de flujo de vídeo por el dispositivo de codificación 106 se registra en la cinta magnética 604 cerca de un código de sincronización, mientras que la salida del flujo `history_stream` por el VLC de registro histórico 211 se registra a más distancia desde el código de sincronización que el flujo de vídeo. Cuando se recupera el flujo de vídeo durante, a modo de ejemplo, una reproducción especial, el código de sincronización se detecta primero y se recupera el flujo de vídeo posterior utilizando el código de sincronización como una referencia. En consecuencia, localizando el flujo de vídeo más próximo al código de sincronización, los datos de vídeo pueden reproducirse de forma fiable incluso durante una reproducción rápida. El flujo `history_stream` no es requerido para una reproducción rápida. Por ello, se evitan los efectos adversos incluso cuando el `history_stream` está situado lejos del código de sincronización.

La señal multiplexada por el dispositivo de multiplexación 621 se aplica a la entrada del dispositivo de codificación de canal 602. Después de la codificación de canal, el cabezal de registro 603 registra la señal en la cinta magnética 604.

De esta manera, en este ejemplo se multiplexa el `history_stream` en la posición diferente de la que tienen los datos de vídeo. En consecuencia, aun cuando el código de inicio aparezcas en esa posición, puede distinguirse suficiente con respecto a los datos de vídeo. En consecuencia, esta realización ejemplo no requiere la inserción del bit de marca que se requiere para la conversión del flujo de registro histórico en el `converted_history_stream`.

Además, los parámetros de codificación pueden suministrarse al dispositivo de multiplexación 621 para su multiplexación en lugar de convertir en el formato del `history_stream`. En este caso, sin embargo, los datos no son comprimidos, con lo que se aumenta la cantidad para los parámetros de codificación para reducir la eficiencia operativa de la cinta magnética 604. De este modo, el VLC de registro histórico 211 se utiliza, de forma deseable, para comprimir los datos en el formato del `history_stream` antes de la multiplexación.

Según se ilustra en la Figura 79, en el sistema de reproducción de la grabadora de cinta de vídeo 601, la señal reproducida desde la cinta magnética 604 por el cabezal de registro 611 es objeto de decodificación de canal por el dispositivo de decodificación de canal 612. Un dispositivo de demultiplexación 631 es objeto de decodificación de canal por el dispositivo de decodificación de canal 612. El dispositivo de demultiplexación 631 separa el flujo codificado de la cuarta generación ST suministrado por el dispositivo de decodificación de canal 612, en el flujo de vídeo y el `history_stream` y suministra el flujo de vídeo al dispositivo de decodificación 102 mientras entrega el `history_stream` al VLD de registro histórico 203.

Es decir, en esta realización ejemplo se utiliza solamente el VLD de registro histórico 203 en lugar del dispositivo de decodificación de registro histórico 104 completo según se ilustra en la Figura 15.

El VLD de registro histórico 203 de longitud variable decodifica el `history_stream` a la salida de los parámetros de codificación de la primera a tercera generación obtenidos para el dispositivo de codificación 106.

Además, el `history_stream` a la salida del dispositivo de demultiplexación 631 es aplicado a la entrada de un convertidor 212'. El convertidor 212' y dispositivo para dar formato a los datos de usuario 213' siguiente están separados del convertidor 212 y del dispositivo para dar formato a los datos de usuario 213 (véase la Figura 15) incorporado en el dispositivo de codificación de registro histórico 107 pero proporciona las mismas funciones.

Es decir, el convertidor 212' añade el bit de marca a la entrada del flujo de registro histórico por el dispositivo de demultiplexación 631 para generar el denominado `converted_history_stream` y lo proporciona a la salida, al

dispositivo para dar formato a los datos de usuario 213'. El dispositivo para dar formato a los datos de usuario 213' convierte el flujo `converted_history_stream` de entrada en los datos de usuarios `user_data` para proporcionar los datos al dispositivo de codificación 106. Los datos de usuarios `user_data` contienen los parámetros de codificación de la primera a tercera generación.

5 El dispositivo de decodificación 102 decodifica la entrada de flujo de vídeo por el dispositivo de demultiplexación 631 para proporcionar la señal de vídeo de banda base al dispositivo de codificación 106. El dispositivo de decodificación 102 suministra también los parámetros de codificación de la cuarta generación al dispositivo de codificación 106, mientras que los proporciona al dispositivo de codificación de registro histórico 107. El dispositivo de codificación de registro histórico 107 genera los datos de usuarios `user_data` desde los parámetros de codificación de la cuarta generación para proporcionar los datos al dispositivo de codificación 106.

15 De forma similar al dispositivo de codificación 106 en la Figura 77, el dispositivo de codificación 106 ejecuta el denominado "proceso de codificación normal" o el "proceso de codificación de parámetros reutilizados" para proporcionar un flujo codificado de la quinta generación ST. Este flujo codificado de la quinta generación ST tiene los parámetros de codificación de la primera a cuarta generación registrados en los datos de usuarios `user_data` de su capa de imágenes.

20 Según se aprecia por la descripción anterior, en conformidad con el transcodificador de la presente invención, los parámetros de codificación generados durante los procesos de codificación anteriores se describen en la zona de datos de usuarios del flujo codificado generado durante el proceso de codificación en curso y el flujo de bits generado es un flujo codificado en conformidad con la norma MPEG. En consecuencia, cualquier decodificación actual puede utilizarse para el proceso de decodificación. Además, el transcodificador en conformidad con la presente invención no requiere ninguna línea dedicada o similar para transmitir los parámetros de codificación para los procesos de codificación anteriores, por lo que los entornos de transmisión de flujo de datos convencionales pueden utilizarse directamente para transmitir los parámetros de codificación anteriores.

30 En conformidad con el transcodificador de esta forma de realización, los parámetros de codificación generados durante los procesos de codificación anteriores se describen, de forma selectiva, en el flujo codificado generado durante el proceso de codificación en curso. En consecuencia, los parámetros de codificación anteriores pueden transmitirse sin la necesidad de aumentar en gran medida la tasa binaria del flujo de bits de salida.

35 En conformidad con el transcodificador de esta forma de realización, los parámetros de codificación óptimos para el proceso de codificación en curso se seleccionan a partir de los parámetros de codificación anteriores y actuales para su codificación. En consecuencia, se evita la degradación de la calidad de imagen a pesar de la repetición de los procesos de decodificación y codificación.

40 En conformidad con el transcodificador de esta forma de realización, los parámetros de codificación óptimos para el proceso de codificación en curso se seleccionan de entre los parámetros de codificación anteriores en conformidad con el tipo de imagen para la codificación. En consecuencia, se impide que niveles de deterioro en la calidad de imagen se hagan de efecto más perjudicial a pesar de la repetición de los procesos de decodificación y codificación.

45 El transcodificador en conformidad con esta forma de realización determina también si reutilizar, o no, los parámetros de codificación anteriores sobre la base de los tipos de imagen contenidos en los parámetros de codificación anteriores con lo que se activa un proceso de codificación óptimo.

Programas informáticos para los procesos anteriores pueden proporcionarse mediante su registro en un soporte de registro tal como un disco magnético, un disco óptico, un disco foto-electro-magnético o una memoria de semiconductores o transmitiéndolos por intermedio de una red tal como Internet, ATM, o un satélite digital, de modo que puedan registrarse en un soporte de registro del usuario.

50 Además, en conformidad con el transcodificador de esta forma de realización, la información para el conjunto de datos que representa una combinación de parámetros de codificación para los procesos de codificación anteriores se describe en el flujo codificado recodificado por el transcodificador. Esta configuración puede generar un flujo que contenga información de registro histórico que pueda transmitirse por intermedio de una ruta de transmisión de una pequeña capacidad.

60 En conformidad con el transcodificador de esta forma de realización, la pluralidad de parámetros de codificación utilizados para los procesos de codificación anteriores se combina, de forma selectiva, para generar información de registro histórico de codificación, que luego se superpone en el flujo codificado. En consecuencia, un flujo capaz de restringir la degradación de la imagen derivada de la recodificación puede transmitirse por intermedio de un soporte de pequeña capacidad.

65 En conformidad con el transcodificador de esta forma de realización, se extraen parámetros de codificación basados en la información que indica un conjunto de datos para generar el flujo codificado recodificado sobre la base de los parámetros de codificación extraídos. En consecuencia, un flujo capaz de restringir la degradación de la imagen derivada de la recodificación y de transmitirse a un soporte de transmisión con pequeña capacidad de transmisión

puede ser objeto de codificación.

5 El transcodificador en conformidad con esta forma de realización registra los datos históricos de codificación detectados de los procesos de codificación anteriores en el soporte de registro, con lo que se reduce la degradación de la calidad de la imagen incluso cuando se registre el flujo codificado en el soporte de registro.

10 El transcodificador en conformidad con esta forma de realización detecta el registro histórico de codificación contenido en el flujo codificado reproducido desde el soporte de registro para multiplexar los datos históricos con el flujo codificado recodificado para la salida, con lo que se limita la degradación de la calidad de la imagen aun cuando el flujo codificado sea reproducido desde el soporte de registro con una nueva transcodificación.

Aplicabilidad industrial

15 La presente invención se refiere a un sistema de codificación y su método, en particular aplicable a un transcodificador para recodificación de un flujo codificado que ha sido codificado en conformidad con la norma MPEG con el fin de generar un flujo recodificado que tenga un grupo GOP diferente (grupo de imágenes) o tasa binaria distinta.

20

25

## REIVINDICACIONES

1. Un sistema de codificación para recodificar un flujo codificado origen, que comprende:
- 5 un medio de decodificación para decodificar dicho flujo codificado origen con el fin de generar datos de vídeo y para extraer desde dicho flujo codificado origen primeros y segundos parámetros de codificación anteriores respectivamente generados por primero y segundo procesos de codificación anteriores;
- 10 un medio de codificación para recodificar, utilizando terceros parámetros de codificación, dichos datos de vídeo para generar un flujo de vídeo recodificado; y
- 15 un medio de control para recibir dichos primeros y segundos parámetros de codificación anteriores y para controlar el proceso de recodificación realizado por dicho medio de codificación, sobre la base de dichos primeros y segundos parámetros de codificación anteriores; y
- 20 un medio de escritura para escribir, de forma selectiva, parámetros de codificación de los primeros y segundos parámetros de codificación anteriores en el flujo de vídeo recodificado y para escribir, en el flujo de vídeo recodificado, un indicador que indica un conjunto de datos de los primeros y segundos parámetros de codificación anteriores escrito en el flujo de vídeo recodificado.
2. El sistema de codificación según la reivindicación 1, caracterizado por cuanto que en el momento de la codificación de dichos parámetros anteriores, dicho medio de control es utilizable para seleccionar parámetros de codificación en macrobloques para describirlos en dicho flujo recodificado.
- 25 3. El sistema de codificación según la reivindicación 1, caracterizado por cuanto que en el momento de la codificación de dichos parámetros anteriores, dicho medio de control es utilizable para describir todos los parámetros de codificación en relación con una unidad de imagen, en dicho flujo recodificado, con independencia de dicha aplicación y describe, de forma selectiva, parámetros de codificación en macrobloques, en dicho flujo recodificado sobre la base de dicha aplicación.
- 30 4. El sistema de codificación según la reivindicación 1, caracterizado por cuanto que dicho medio de control es utilizable para describir dichos parámetros de codificación anteriores seleccionados en dicho flujo recodificado bajo la forma de un flujo denominado `history_stream()` y describe información que indica un conjunto de datos para dichos parámetros de codificación anteriores seleccionados, en dicho flujo recodificado, bajo la forma de información denominada `re_coding_stream_info()`.
- 35 5. El sistema de codificación según la reivindicación 4, caracterizado por cuanto que dicha información denominada `re_coding_stream_info()` contiene un indicador `red_bw_flag` y un indicador `red_bw_indicator` como información para identificar dicho conjunto de datos.
- 40 6. El sistema de codificación según la reivindicación 1, caracterizado por cuanto que dicho conjunto de datos comprende:
- 45 una pluralidad de conjuntos de datos que incluye al menos un conjunto de datos para transmitir parámetros de codificación requeridos para generar un flujo recodificado completamente transparente; y
- un conjunto de datos para transmitir parámetros de codificación requeridos para generar un flujo recodificado relativamente transparente.
- 50 7. El sistema de codificación según la reivindicación 1, caracterizado por cuanto que dicho conjunto de datos comprende: una pluralidad de conjuntos de datos que incluye al menos un conjunto de datos para transmitir todos dichos parámetros de codificación anteriores a dichos parámetros recodificados bajo la forma de un flujo denominado `history_stream()`; y
- 55 un conjunto de datos que no transmite dichos parámetros de codificación anteriores que siguen datos de imagen `picture_data()`, a dichos parámetros recodificados bajo la forma de un flujo denominado `history_stream()`.
- 60 8. El sistema de codificación según la reivindicación 1, caracterizado por cuanto que dicho conjunto de datos comprende:
- una pluralidad de conjuntos de datos que incluyen al menos un conjunto de datos para transmitir parámetros de codificación en imágenes y en macrobloques a dichos parámetros recodificados bajo la forma de un flujo denominado `history_stream()`; y
- 65 un conjunto de datos que transmite los parámetros de codificación en imágenes a dichos parámetros recodificados bajo la forma de un flujo denominado `history_stream()`, pero que no transmite los parámetros de codificación en

macrobloques a dichos parámetros recodificados bajo la forma del flujo history\_stream ().

**9.** El sistema de codificación según la reivindicación 1, caracterizado por cuanto que:

5 dicho medio de codificación es utilizable para codificar durante dicho proceso de codificación en curso, una imagen de referencia contenida en dichos datos de vídeo de entrada, utilizando el tipo de imagen en curso asignado a dicha imagen de referencia; y

10 dicho medio de control es utilizable para determinar si dicha imagen de referencia se ha codificado, o no, en el mismo tipo de imagen que el tipo asignado durante un proceso de codificación anterior y controla dicho proceso de codificación en curso sobre la base de un resultado de la determinación.

15 **10.** El sistema de codificación según la reivindicación 9, caracterizado por cuanto que, sobre la base de dicha determinación, dicho medio de control es utilizable para seleccionar parámetros óptimos de entre dichos parámetros de codificación anteriores para controlar el proceso de codificación en curso realizado por dicho medio de codificación, sobre la base de dichos parámetros de codificación óptimos seleccionados.

20 **11.** El sistema de codificación según la reivindicación 9, caracterizado por cuanto que dicho medio de control es utilizable para codificar dicha imagen de referencia utilizando uno de los parámetros de codificación anteriores generados durante dicho proceso de codificación anterior.

25 **12.** El sistema de codificación según la reivindicación 9, caracterizado por cuanto que dichos parámetros de codificación anteriores incluyen información de vectores de movimiento generada durante dicho proceso de codificación anterior; y

dicho medio de codificación es utilizable para incluir un medio de detección de vectores de movimiento para detectar información de vectores de movimiento para dicha imagen de referencia durante dicho proceso de codificación en curso.

30 **13.** El sistema de codificación según la reivindicación 12, caracterizado por cuanto que dicho medio de control es utilizable para controlar la operación de dicho medio de detección de vectores de movimiento sobre la base de dicho resultado de la determinación.

35 **14.** El sistema de codificación según la reivindicación 13, caracterizado por cuanto que dicho medio de control es utilizable para reutilizar dicha información de vectores de movimiento incluida en dichos parámetros de codificación anteriores, en lugar de permitir a dicho medio de detección de vectores de movimiento calcular una nueva información de vectores de movimiento.

40 **15.** El sistema de codificación según la reivindicación 9, caracterizado por cuanto que dicho medio de control es utilizable para seleccionar parámetros óptimos de entre dichos parámetros de codificación anteriores que corresponden a dicho proceso de codificación en curso y controla dicho proceso de codificación en curso realizado por dicho medio de codificación, sobre la base de dichos parámetros de codificación óptimos.

45 **16.** El sistema de codificación según la reivindicación 9, caracterizado por cuanto que

dicho medio de codificación es utilizable para codificar, durante dicho proceso de codificación en curso, una imagen de referencia contenida en dichos datos de vídeo de entrada, utilizando el tipo de imagen en curso que se asigna a dicha imagen de referencia, y

50 dicho medio de control es utilizable para determinar si dicha imagen de referencia ha sido codificada, o no, en el mismo tipo de imagen que se asigna durante un proceso de codificación anterior y seleccionar dichos parámetros de codificación óptimos sobre la base de un resultado de la determinación.

55 **17.** El sistema de codificación según la reivindicación 15, caracterizado por cuanto que

dichos parámetros de codificación anteriores incluyen información del modo de predicción que representa un modo de predicción de tramas o un modo de predicción de campos, y

60 dicho medio de control es utilizable para controlar dicho proceso de codificación en curso dependiendo de dicha información del modo de predicción.

**18.** El sistema de codificación según la reivindicación 15, caracterizado por cuanto que si dicha imagen de referencia ha sido codificada en el mismo tipo de imagen que dicho tipo de imagen anterior, dicho medio de control es utilizable para reutilizar dicha información de modo de predicción incluida en dichos parámetros de codificación anteriores en lugar de calcular una nueva información de modo de predicción.

65

19. El sistema de codificación según la reivindicación 15, caracterizado por cuanto que:

dichos parámetros de codificación incluyen información del tipo de predicción que indica una intra-predicción, una predicción directa, una predicción inversa o una predicción bidireccional y en donde:

5 dicho medio de control es utilizable para controlar dicho proceso de codificación en curso sobre la base de dicha información de tipo de predicción.

20. El sistema de codificación según la reivindicación 15, caracterizado por cuanto que:

10 dichos parámetros de codificación incluyen información de modo DCT que representa un modo DCT de tramas o un modo DCT de campos de imagen y en donde:

15 dicho medio de control es utilizable para controlar dicho proceso de codificación en curso sobre la base de dicha información de modo DCT.

21. El sistema de codificación según la reivindicación 9 caracterizado por cuanto que dicho medio de codificación tiene un medio de generación para generar un flujo de bits MPEG conforme a la norma de MPEG y que comprende una capa de secuencias, una capa de GOP, una capa de imágenes, una capa de segmentos y una capa de macrobloques.

20  
25 22. El sistema de codificación según la reivindicación 21, caracterizado por cuanto que dicho medio de control es utilizable para generar los parámetros de codificación en curso correspondientes a dicho proceso de codificación en curso realizado por dicho medio de codificación, y

dicho medio de control es utilizable para describir dichos parámetros de codificación en curso en dichas capas de imágenes, de segmentos y de macrobloques de dicho flujo recodificado mientras que describe, de forma selectiva, dichos parámetros de codificación anteriores en una zona de datos de usuario de la capa de imagen de dicho flujo recodificado.

30 23. El sistema de codificación según la reivindicación 9, caracterizado por cuanto que dicho medio de control es utilizable para describir dichos parámetros de codificación anteriores en dicho flujo recodificado como un flujo denominado `history_stream()`.

35 24. Un método para recodificar un flujo codificado origen, que comprende:

decodificar dicho flujo codificado origen para generar datos de vídeo y extraer desde dicho flujo codificado origen, primeros y segundos parámetros de codificación anteriores generados por primeros y segundos procesos de codificación anteriores, respectivamente;

40 recibir dichos primeros y segundos parámetros de codificación anteriores;

45 recodificar, utilizando los terceros parámetros de codificación, dichos datos de vídeo para generar un flujo de vídeo recodificado en donde la recodificación se controla sobre la base de dichos primeros segundos parámetros de codificación anteriores;

escribir, de forma selectiva, parámetros de codificación de los primeros y segundos parámetros de codificación anteriores en el flujo de vídeo recodificado; y

50 escribir, en el flujo recodificado, un indicador que indica un conjunto de datos de los primeros y segundos parámetros de codificación escritos en el flujo de vídeo recodificado.

55



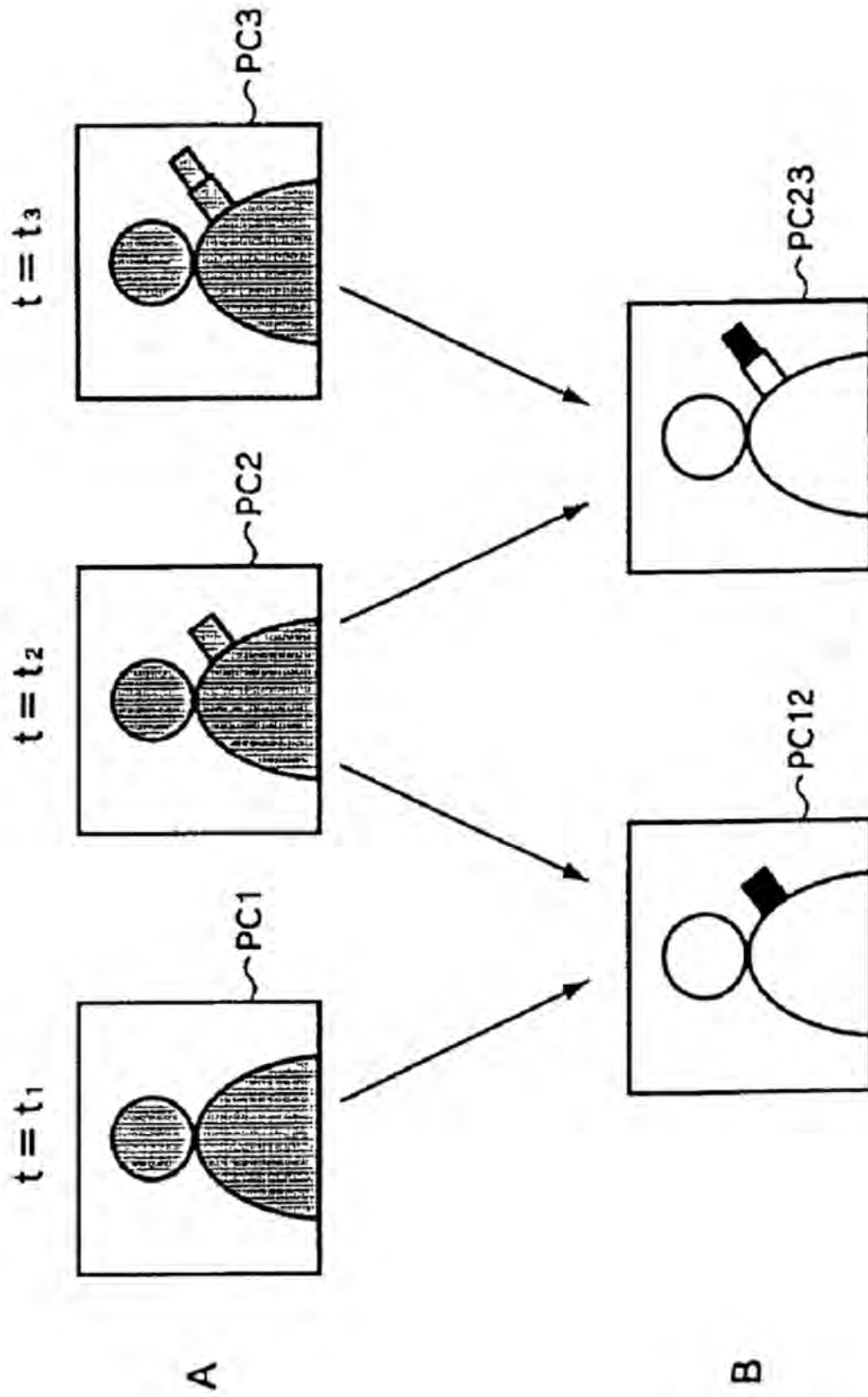


FIG. 1

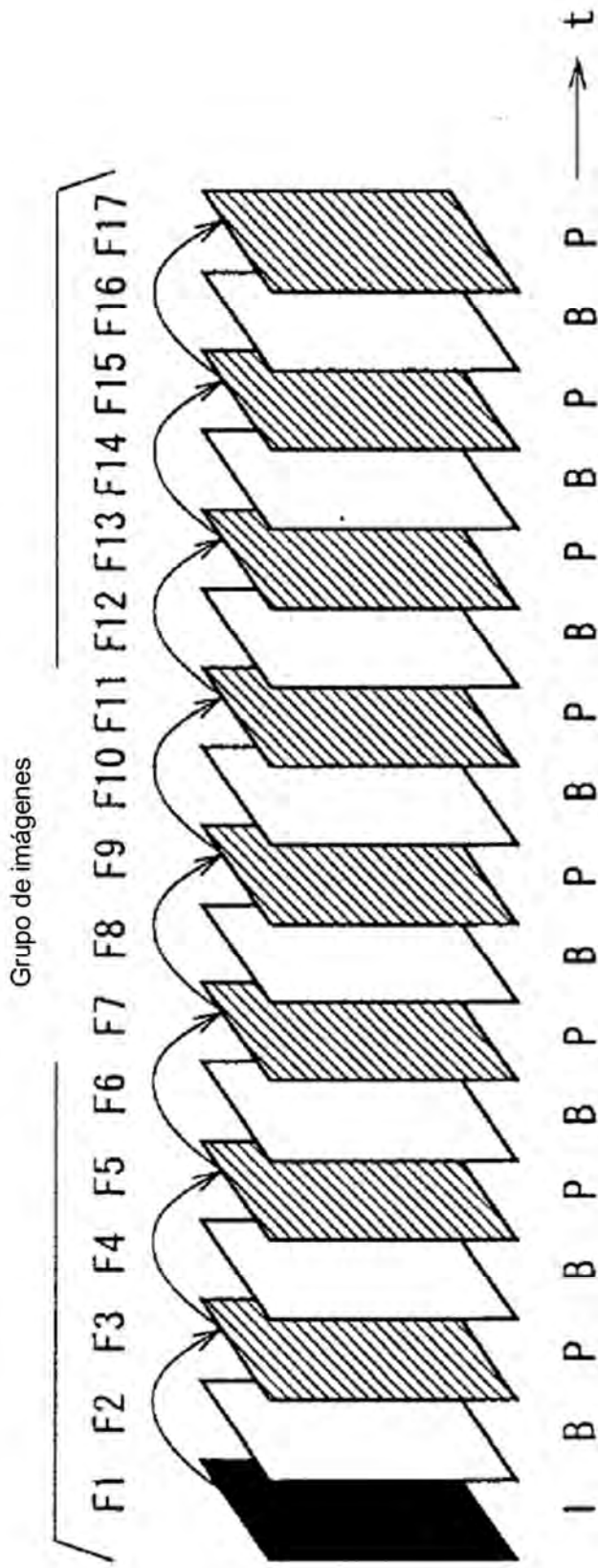


IMAGEN P (PREDICCIÓN DIRECTA)

TIPO DE IMÁGENES I, P Y B

FIG. 2

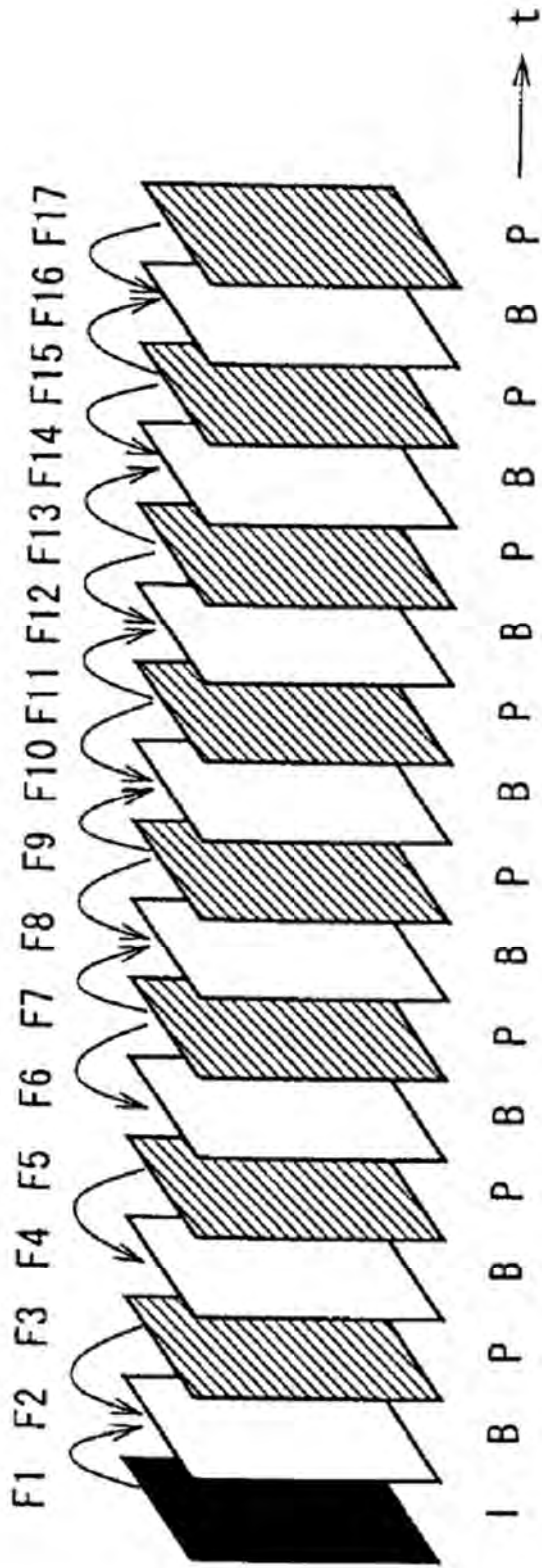
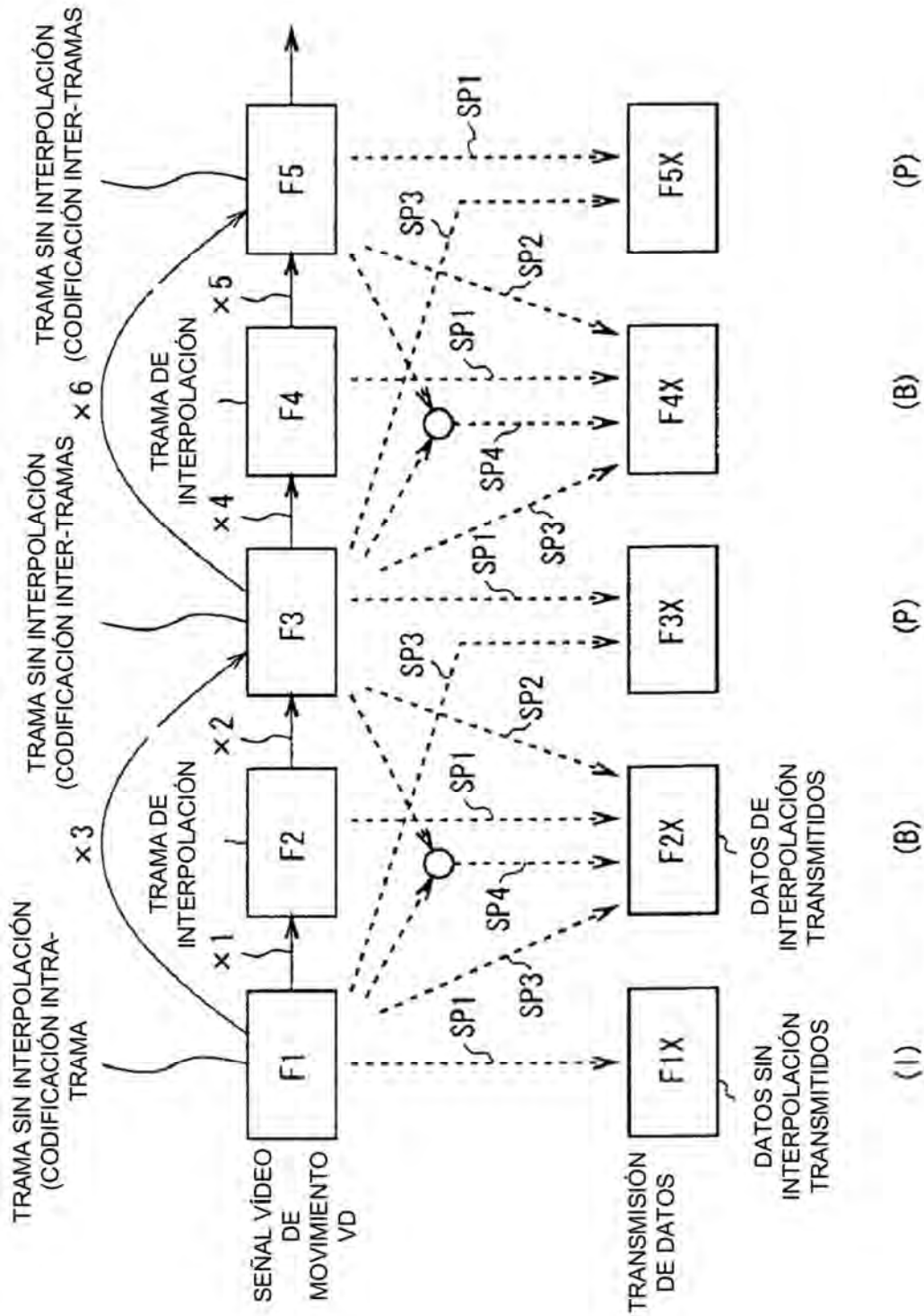


IMAGEN B (PREDICCIÓN BIDIRECCIONAL)

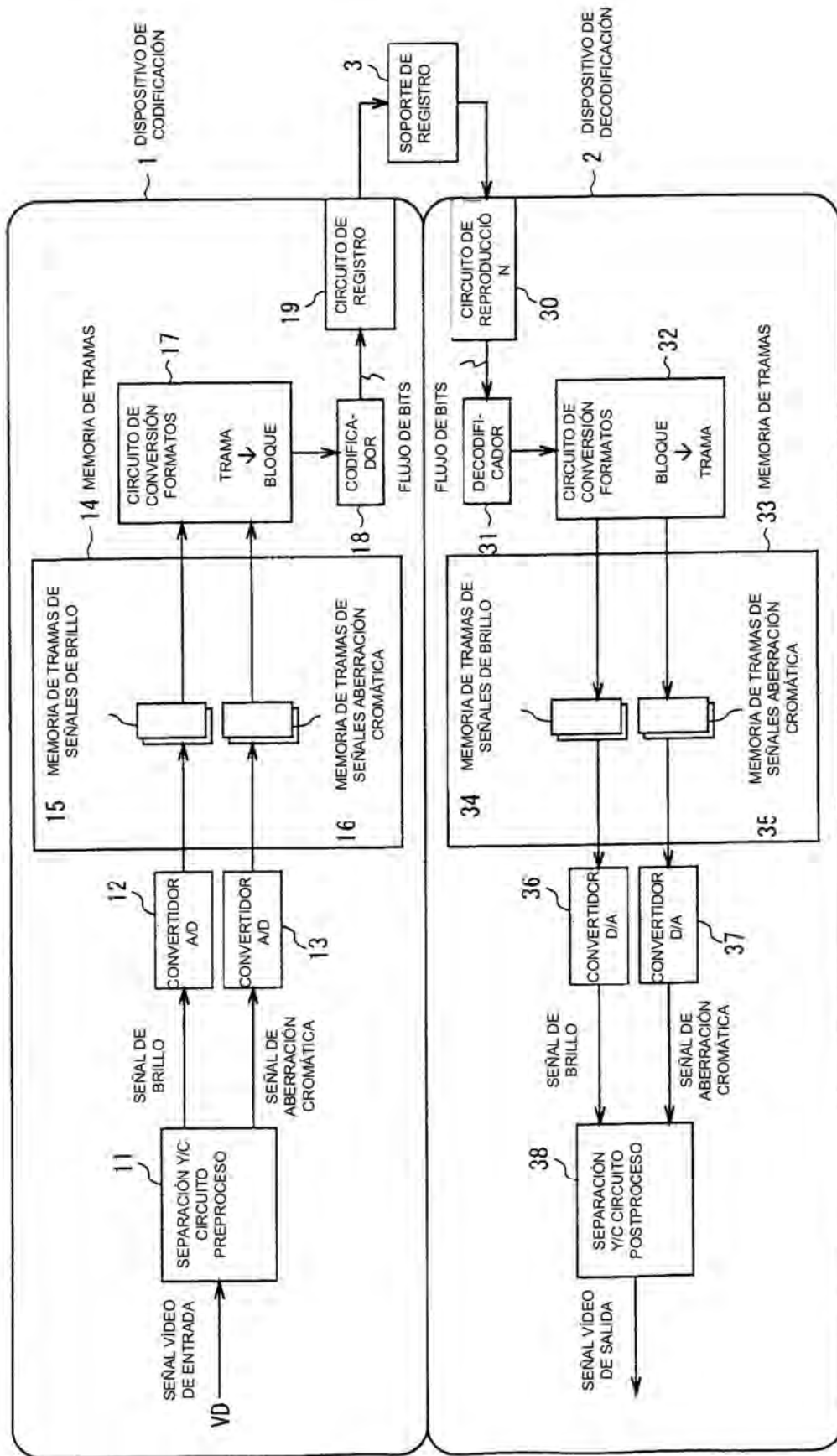
TIPO DE IMÁGENES I, P, Y B

FIG. 3



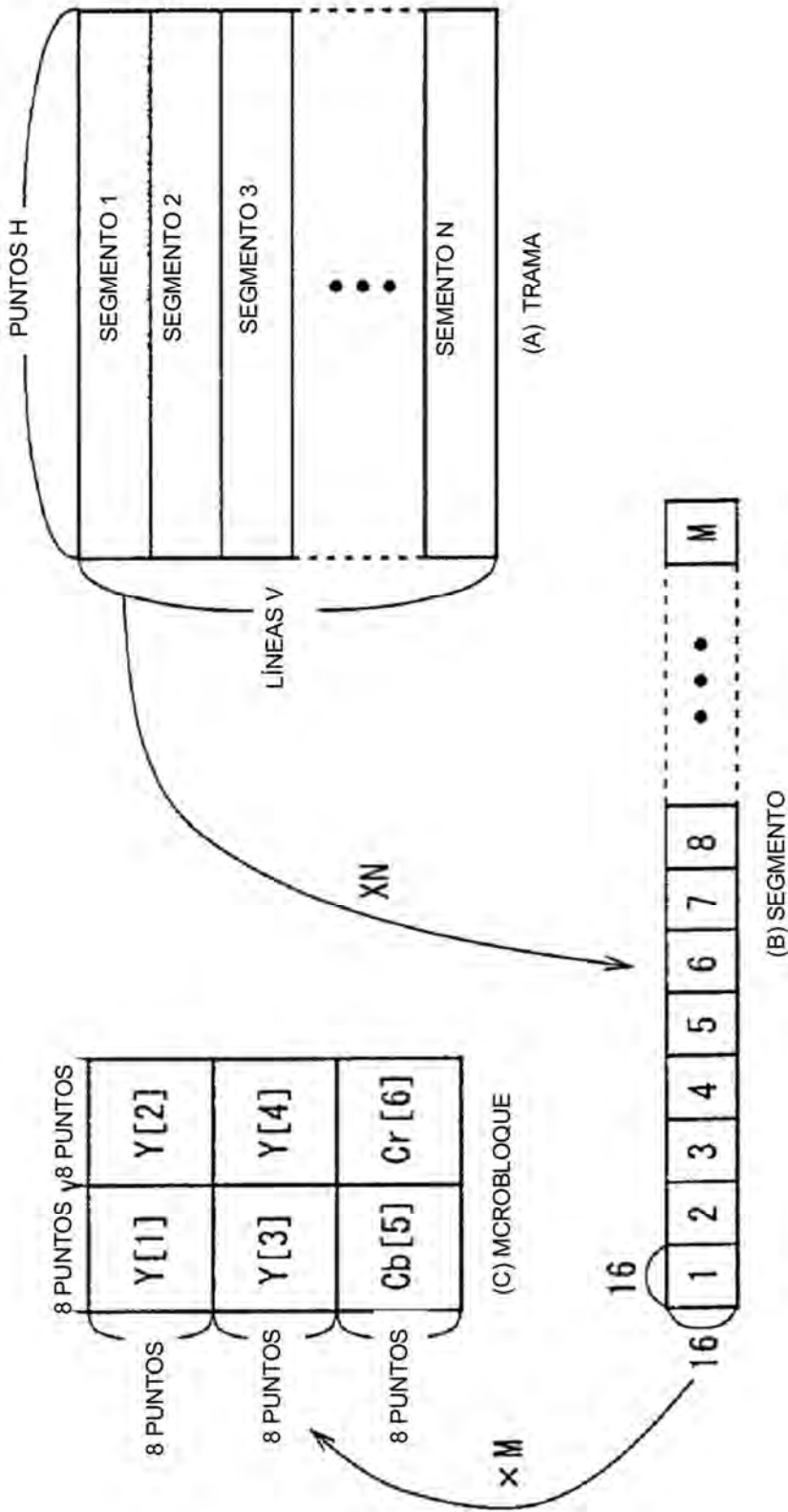
PRINCIPIO DEL MÉTODO DE CODIFICACIÓN DE SEÑALES DE MOVIMIENTO

FIG. 4



DISPOSITIVO DE CODIFICACIÓN/DECODIFICACIÓN DE IMÁGENES DE MOVIMIENTO

FIG. 5



ESTRUCTURA DE DATOS DE IMÁGENES

FIG. 6

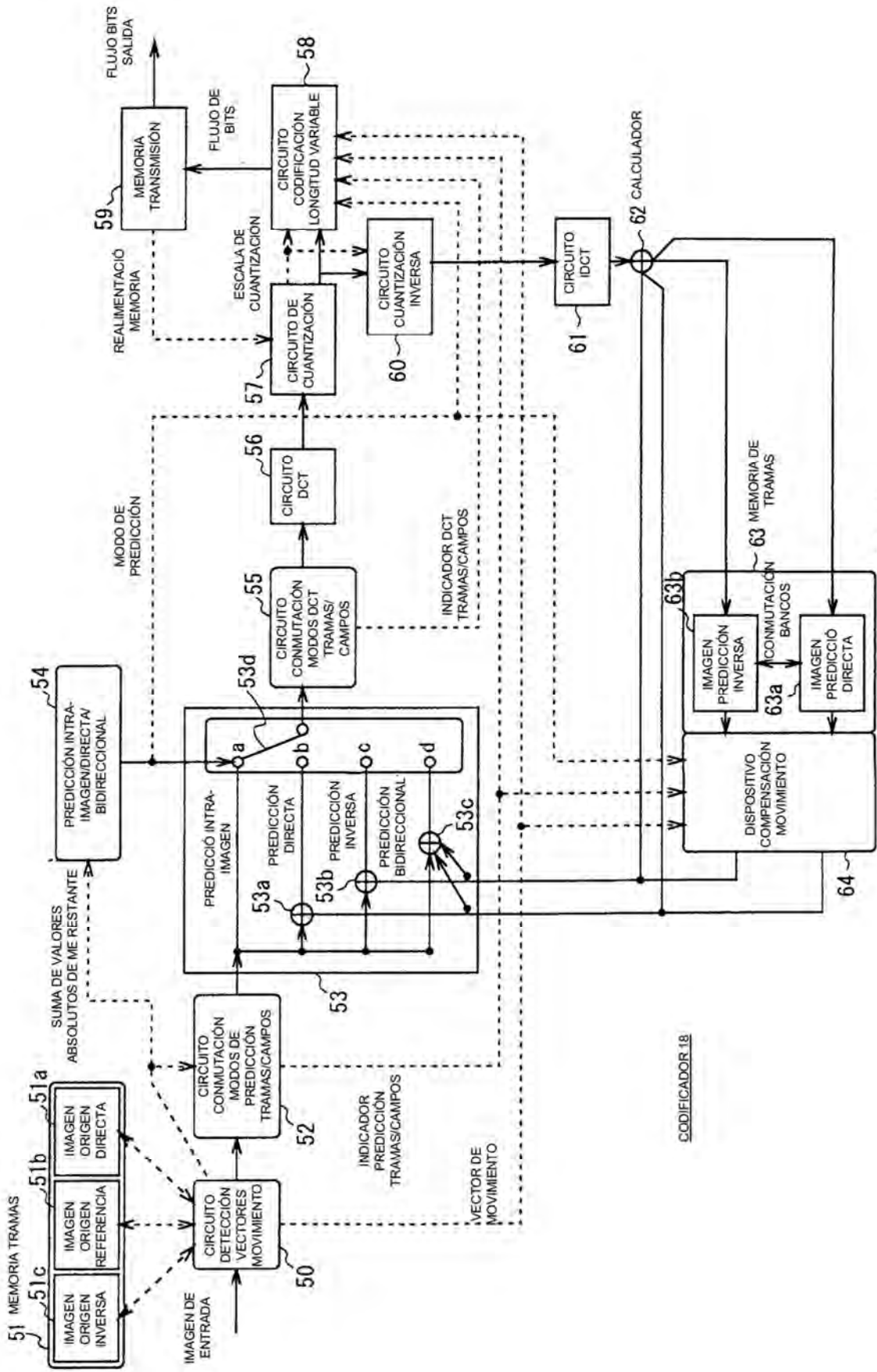


FIG. 7

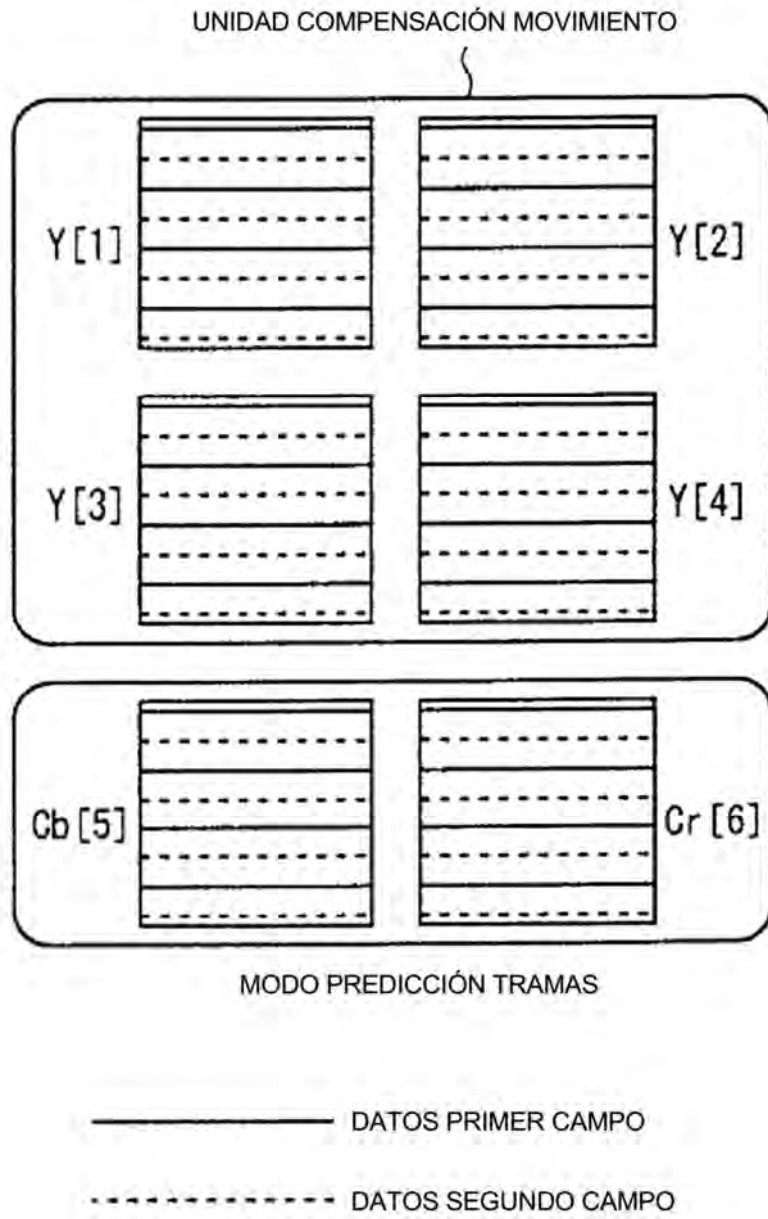
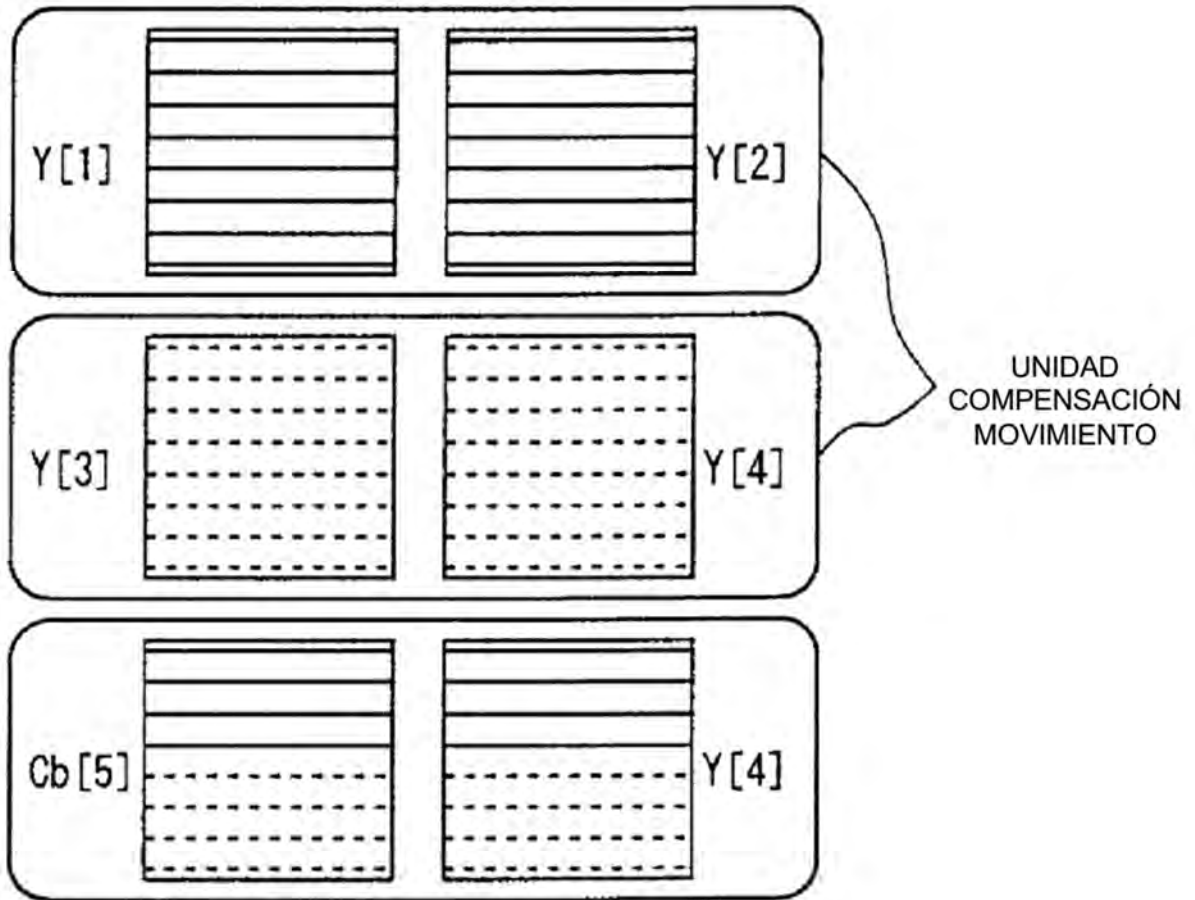


FIG. 8

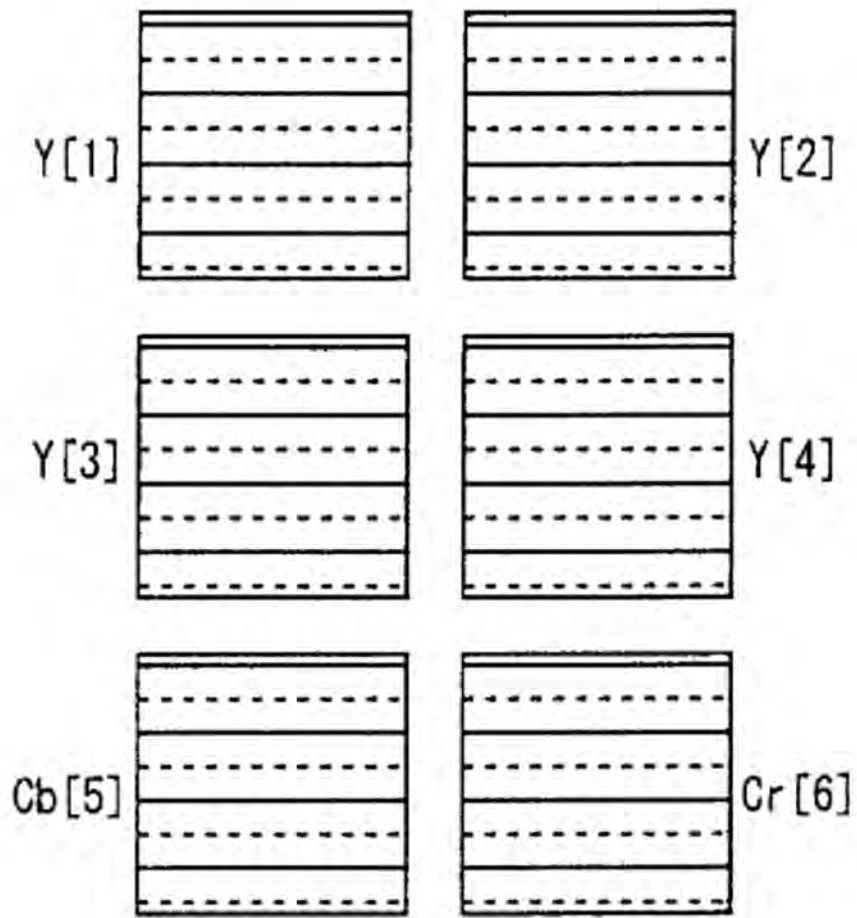




MODO PREDICCIÓN CAMPOS

———— DATOS PRIMER CAMPO  
----- DATOS SEGUNDO CAMPO

FIG. 9



MODO DCT TRAMAS

————— DATOS PRIMER CAMPO  
- - - - - DATOS SEGUNDO CAMPO

FIG. 10

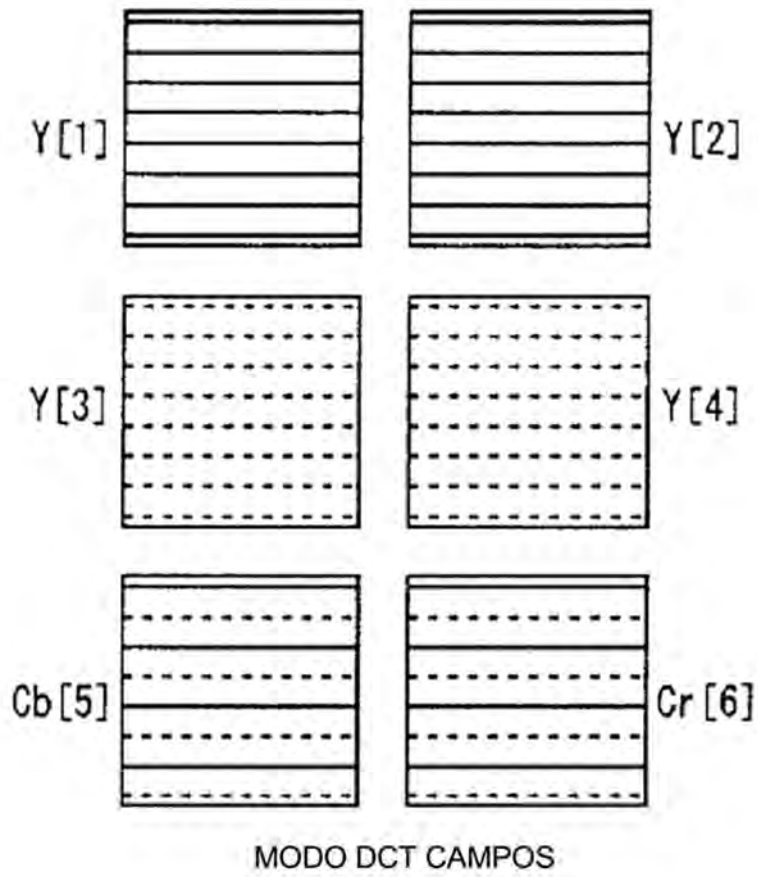


FIG. 11

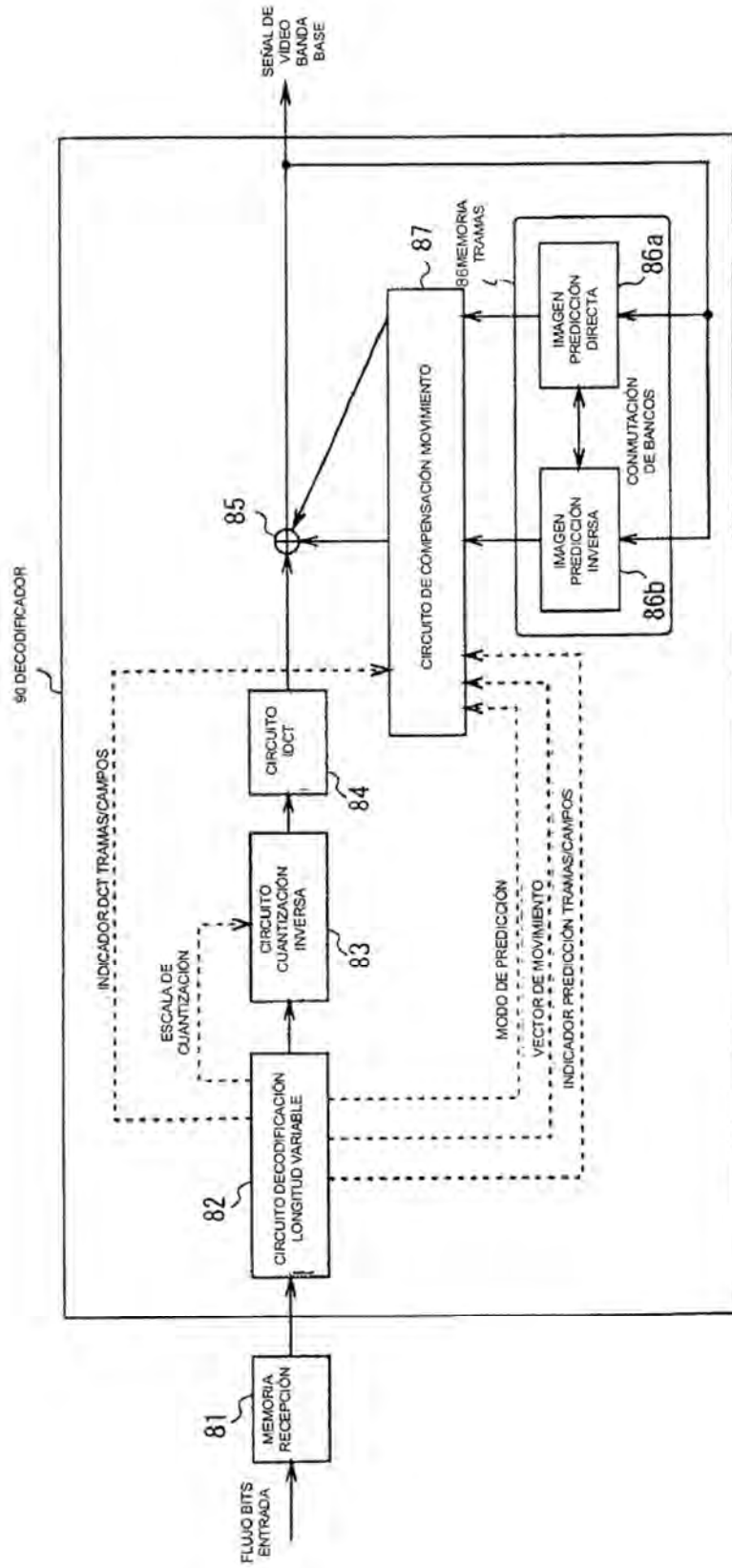


FIG. 12

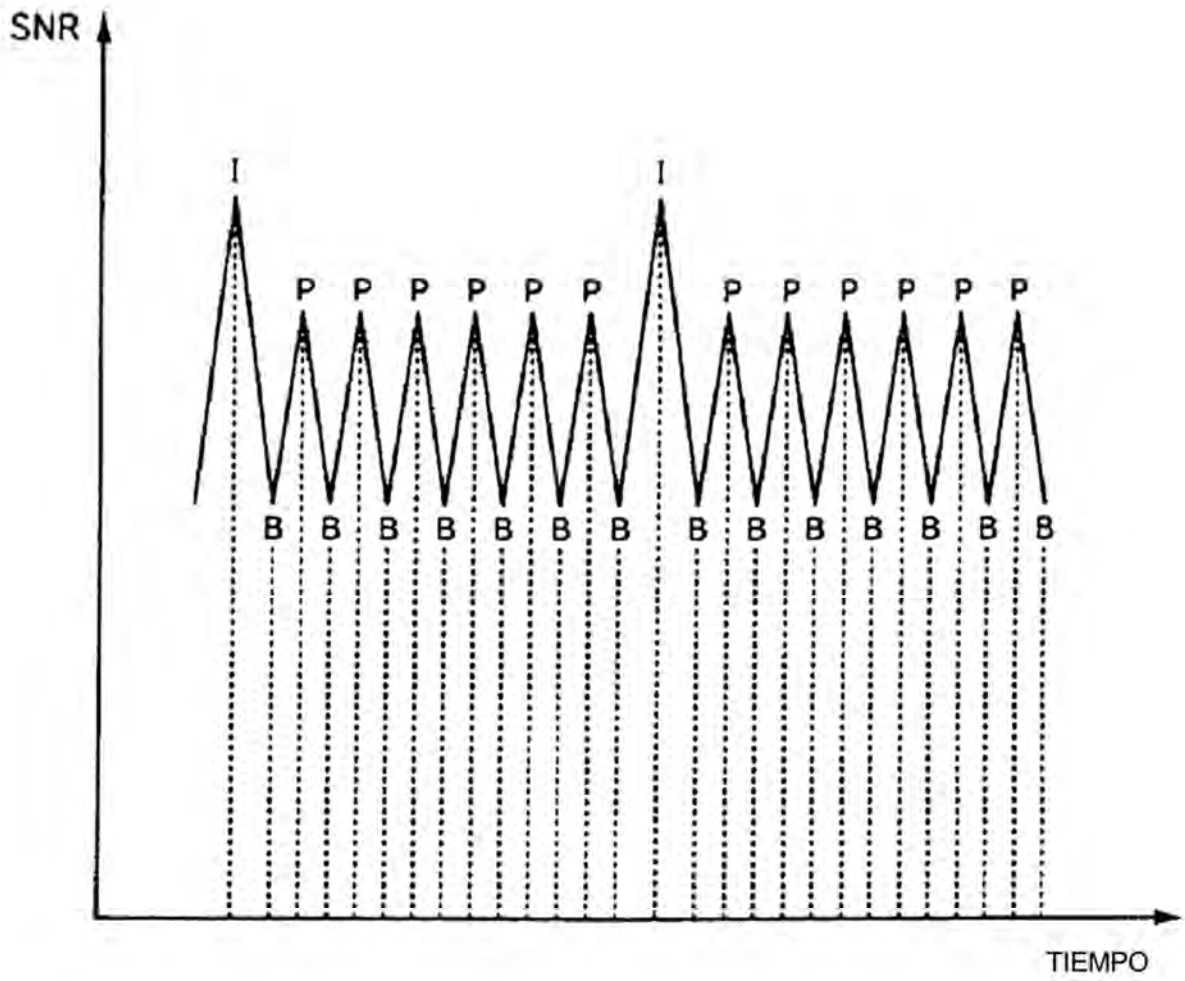
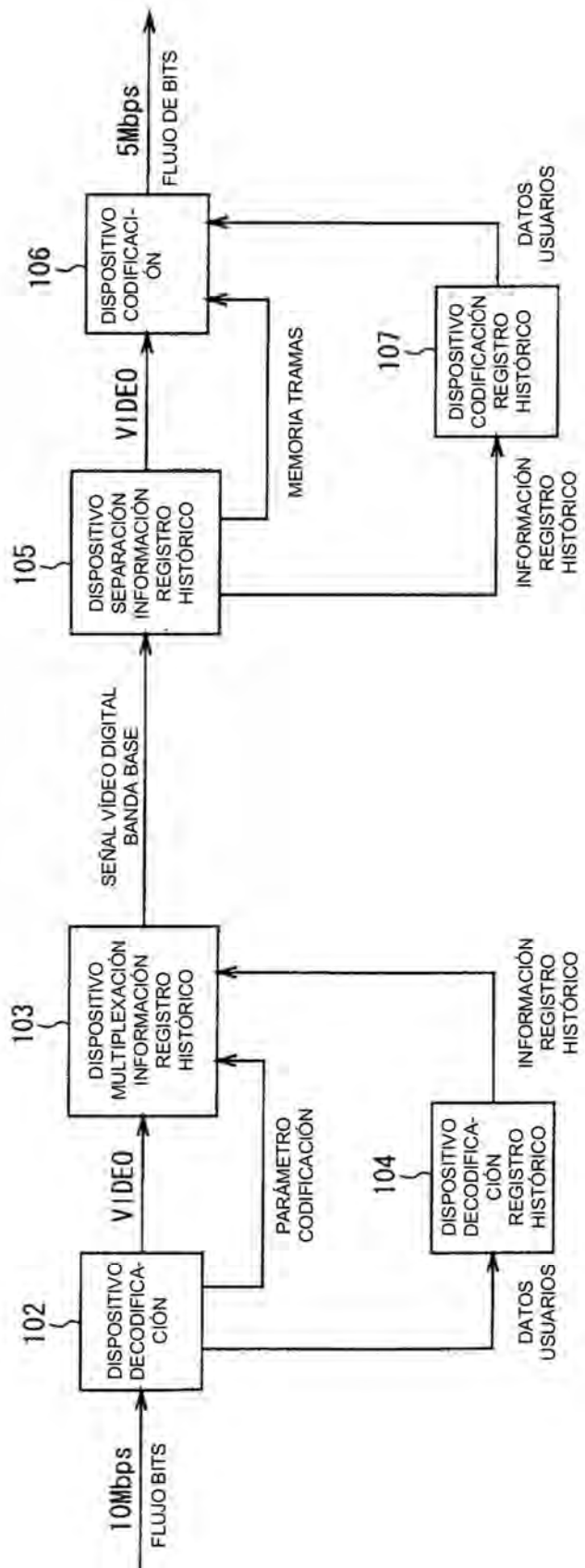


FIG. 13



TRANSCODIFICADOR 101

FIG. 14

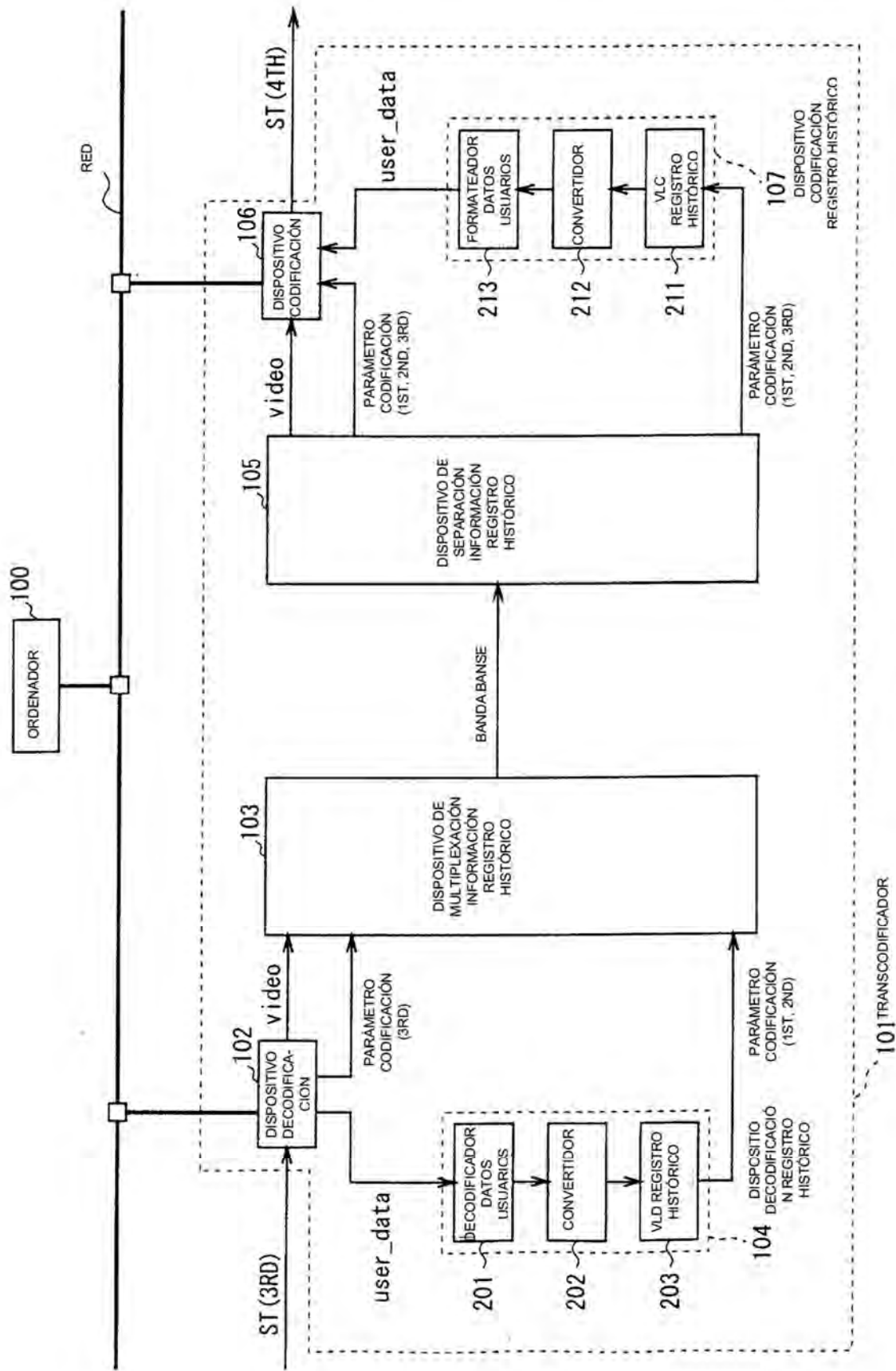


FIG. 15

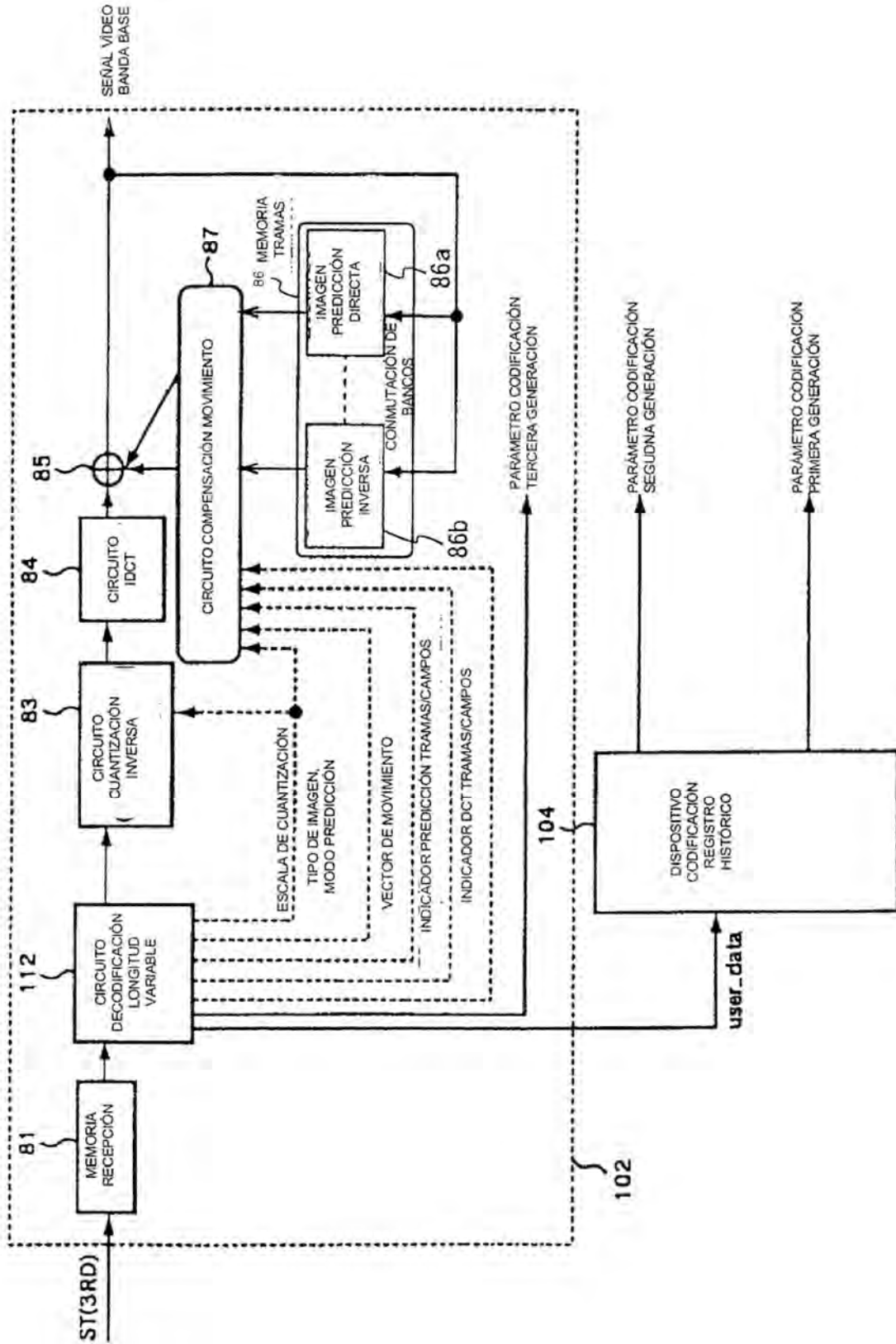


FIG. 16



0	1	2	.....	13	14	15
16	17	18	.....	29	30	31
32	33	34	.....	45	46	47
·	·	·		·	·	·
·	·	·		·	·	·
·	·	·		·	·	·
·	·	·		·	·	·
·	·	·		·	·	·
·	·	·		·	·	·
·	·	·		·	·	·
·	·	·		·	·	·
·	·	·		·	·	·
208	209	210	.....	221	222	223
224	225	226	.....	237	238	239
240	241	242	.....	251	254	255

MACROBLOQUE

FIG. 17

D9	Cb[0][9]	Y[0][9]	Cr[0][9]	Y[1][9]	Cb[1][9]	Y[2][9]	Cr[1][9]	Y[3][9]
D8	Cb[0][8]	Y[0][8]	Cr[0][8]	Y[1][8]	Cb[1][8]	Y[2][8]	Cr[1][8]	Y[3][8]
D7	Cb[0][7]	Y[0][7]	Cr[0][7]	Y[1][7]	Cb[1][7]	Y[2][7]	Cr[1][7]	Y[3][7]
D6	Cb[0][6]	Y[0][6]	Cr[0][6]	Y[1][6]	Cb[1][6]	Y[2][6]	Cr[1][6]	Y[3][6]
D5	Cb[0][5]	Y[0][5]	Cr[0][5]	Y[1][5]	Cb[1][5]	Y[2][5]	Cr[1][5]	Y[3][5]
D4	Cb[0][4]	Y[0][4]	Cr[0][4]	Y[1][4]	Cb[1][4]	Y[2][4]	Cr[1][4]	Y[3][4]
D3	Cb[0][3]	Y[0][3]	Cr[0][3]	Y[1][3]	Cb[1][3]	Y[2][3]	Cr[1][3]	Y[3][3]
D2	Cb[0][2]	Y[0][2]	Cr[0][2]	Y[1][2]	Cb[1][2]	Y[2][2]	Cr[1][2]	Y[3][2]
D1	PRIMERA GENERACIÓN		SEGUNDA GENERACIÓN		TERCERA GENERACIÓN			
D0	Cb[0][x]	Y[0][x]	Cr[0][x]	Y[1][x]	Cb[1][x]	Y[2][x]	Cr[1][x]	Y[3][x]

ÁREA DATOS  
IMÁGENES

ÁREA  
INFORMACIÓN  
REGISTRO  
HISTÓRICO

FIG. 18

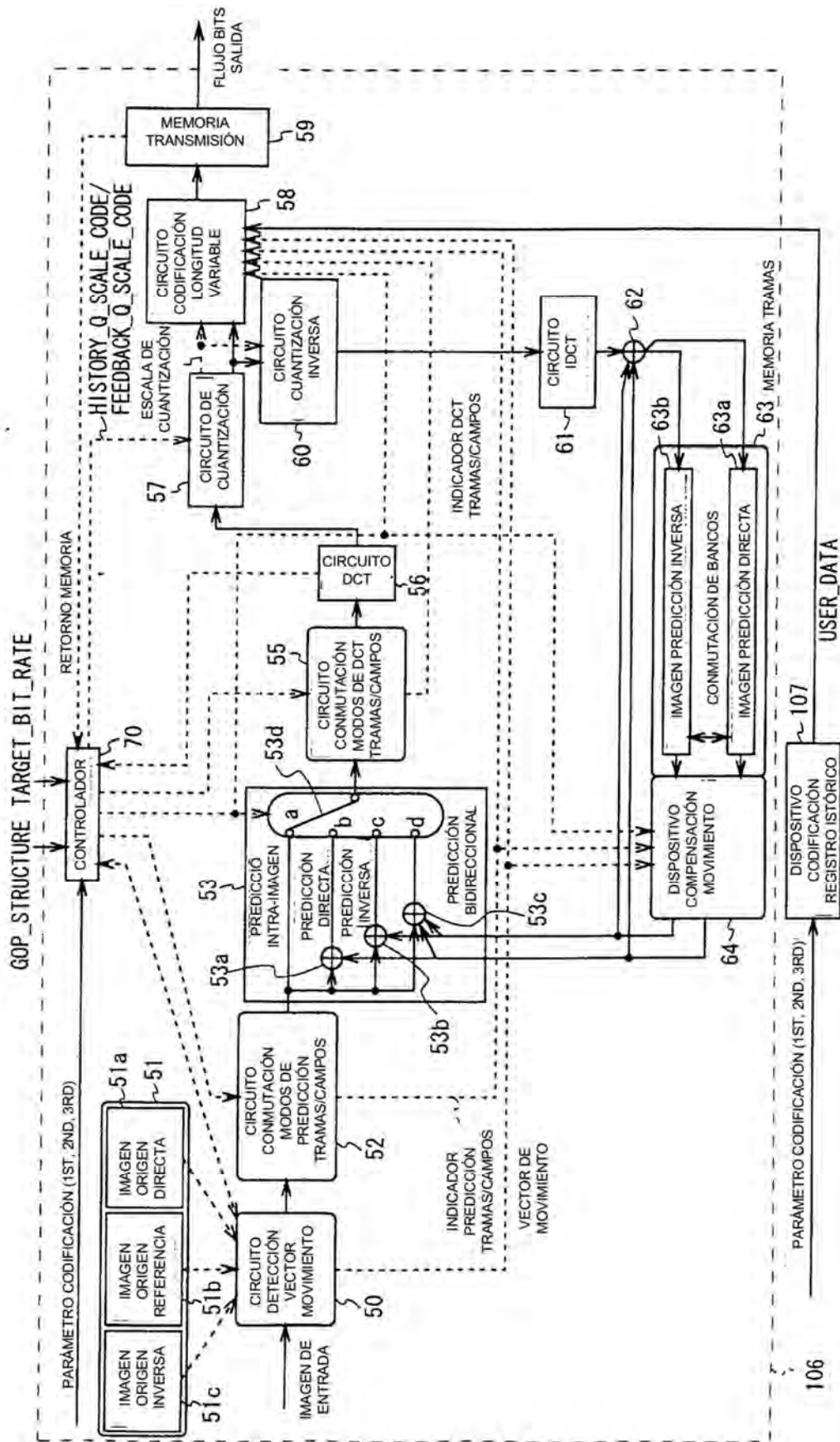
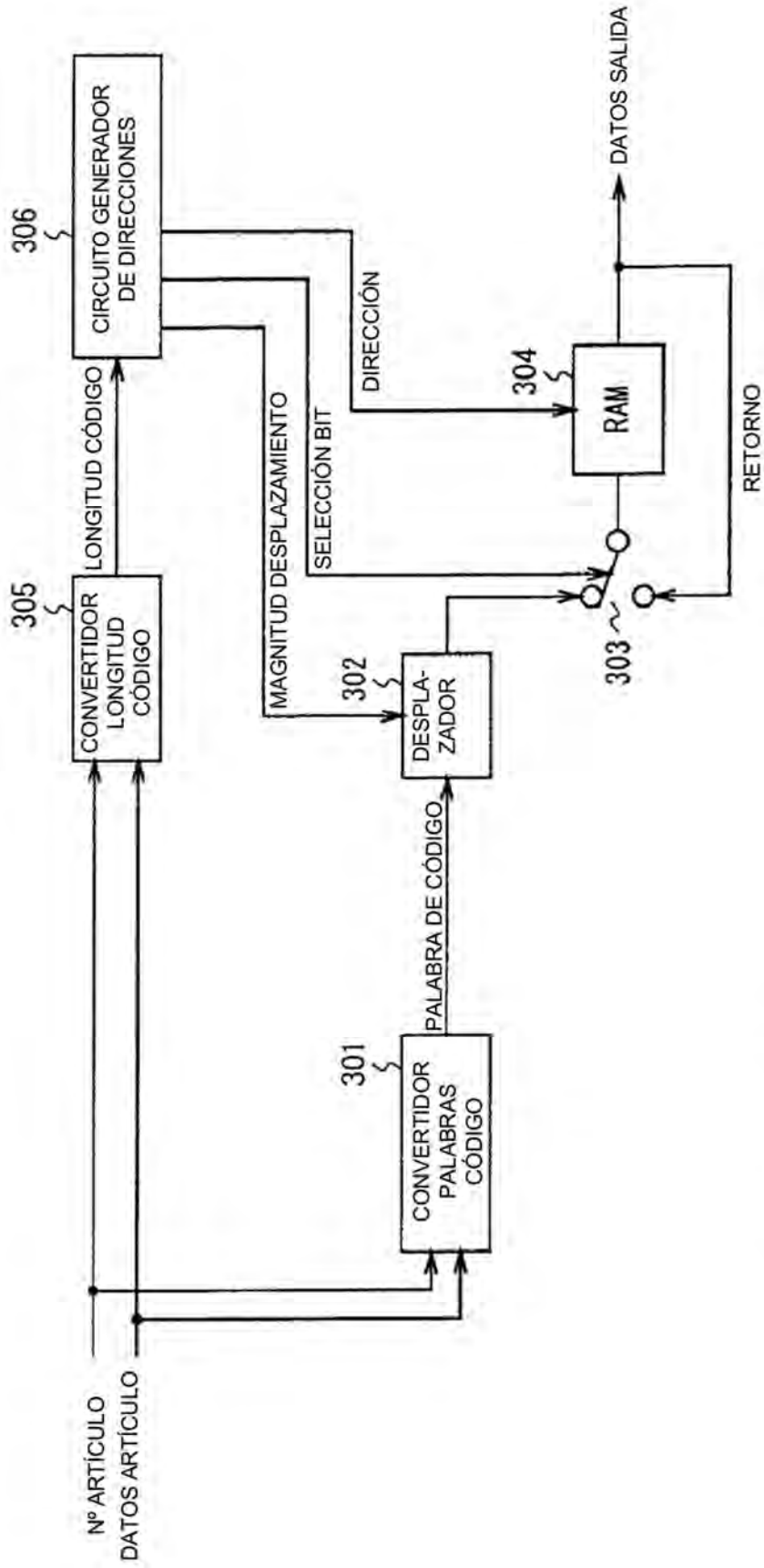
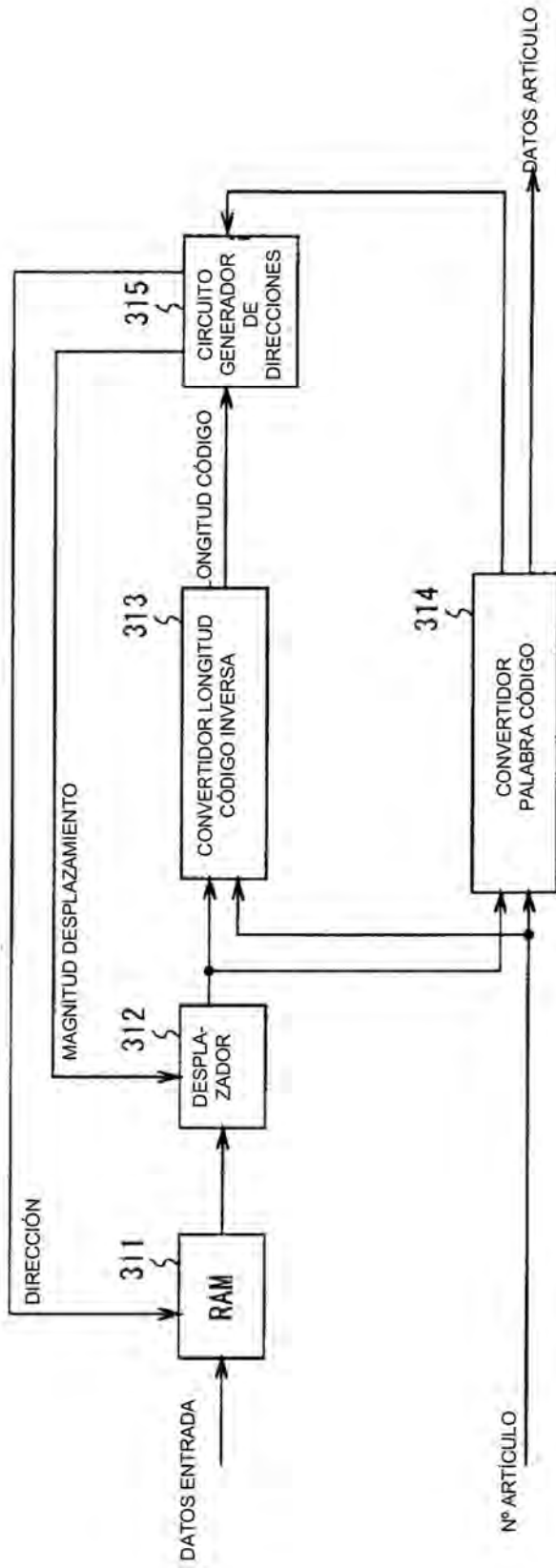


FIG. 19



VLC REGISTRO HISTÓRICO 211

FIG. 20



VLD REGISTRO HISTÓRICO 203

FIG. 21

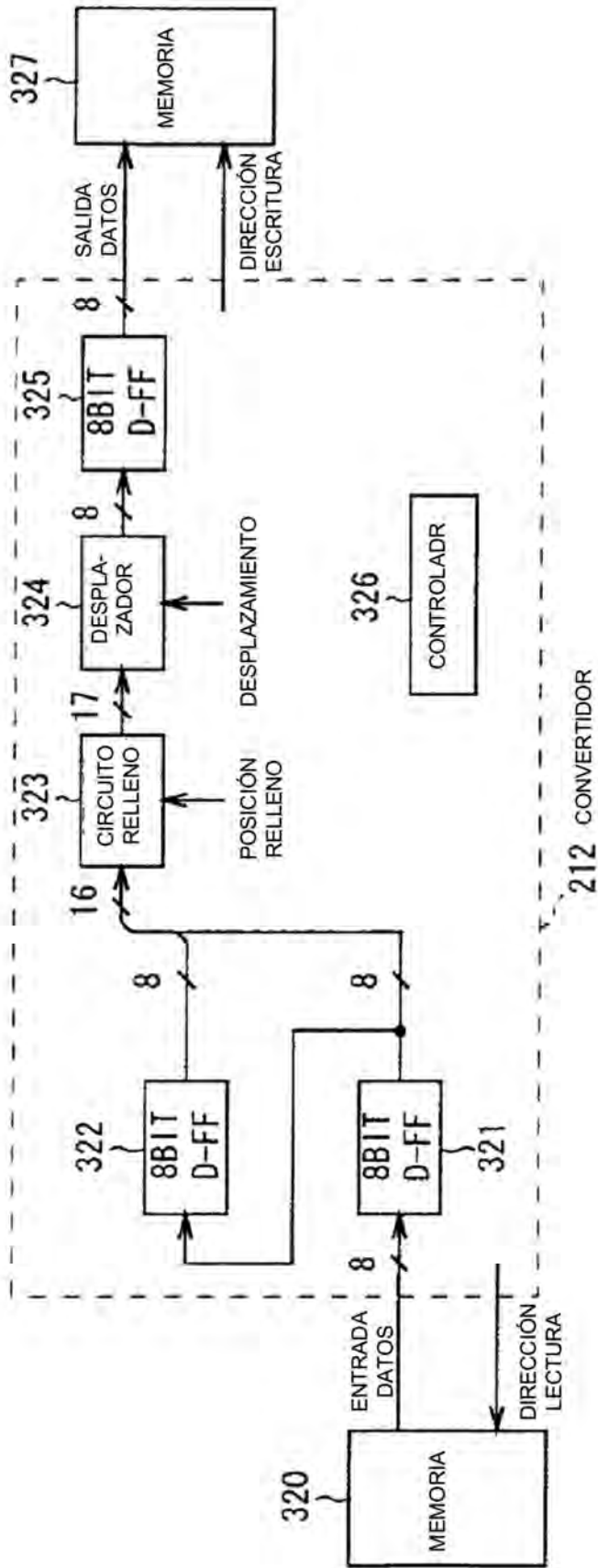


FIG. 22

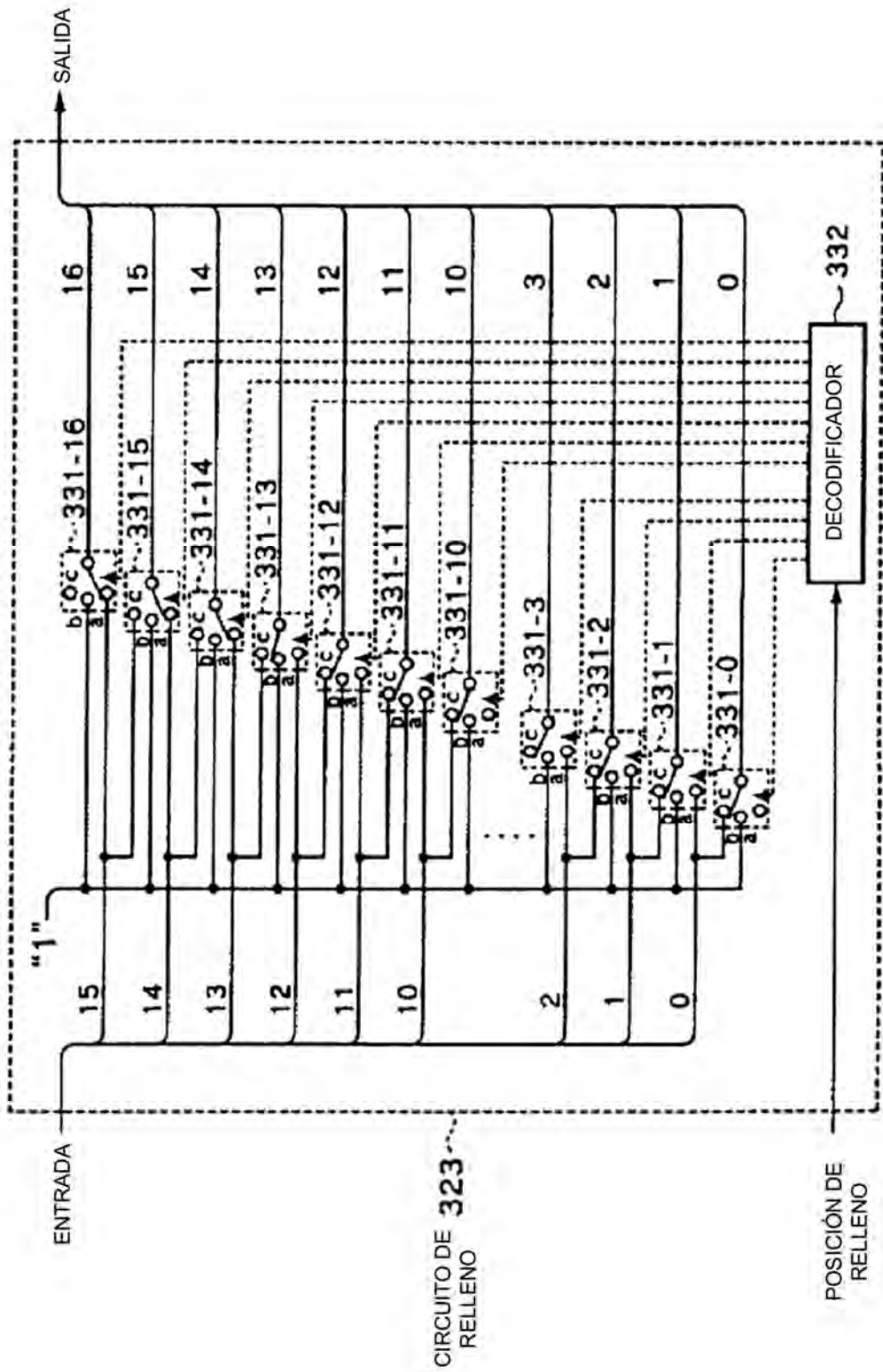


FIG. 23

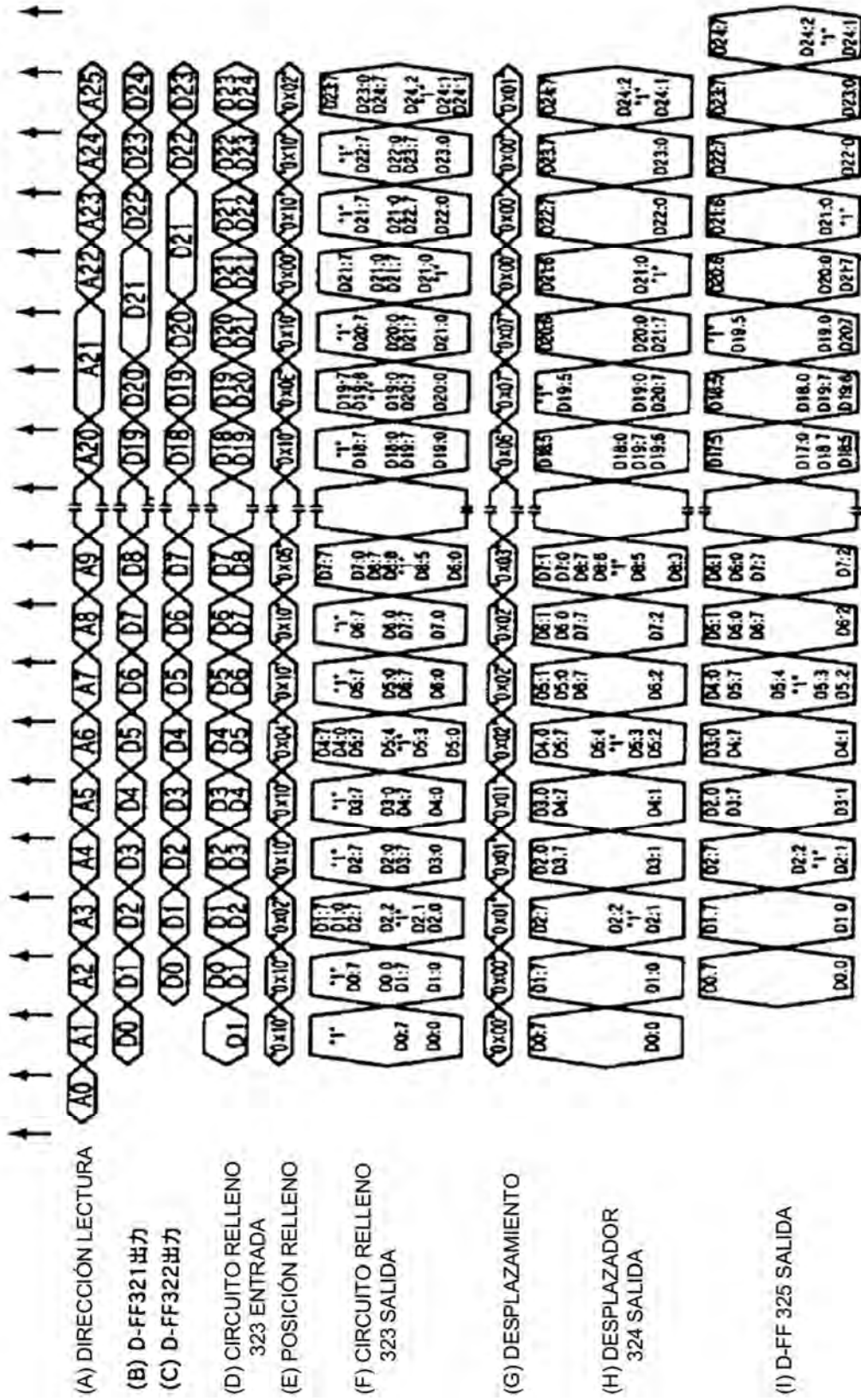


FIG. 24



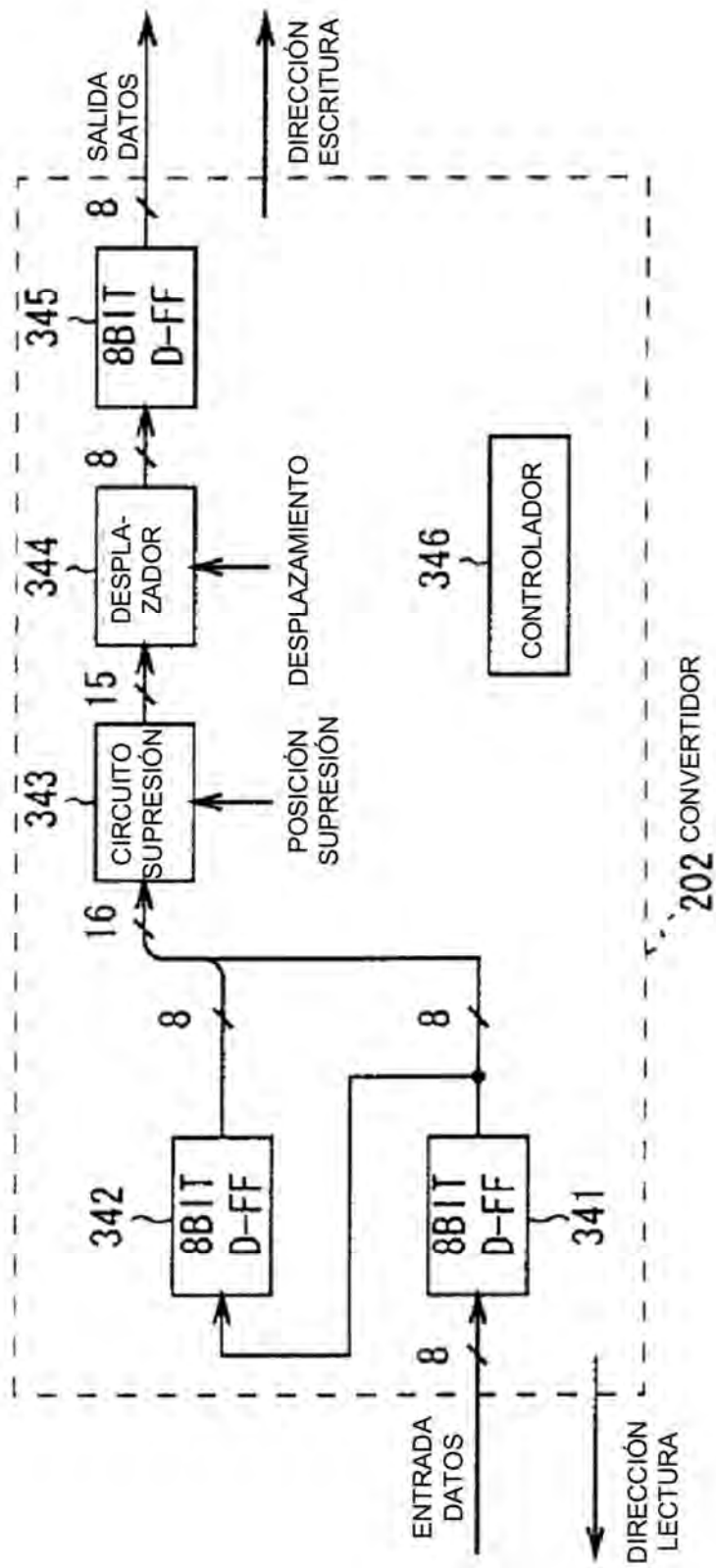


FIG. 25

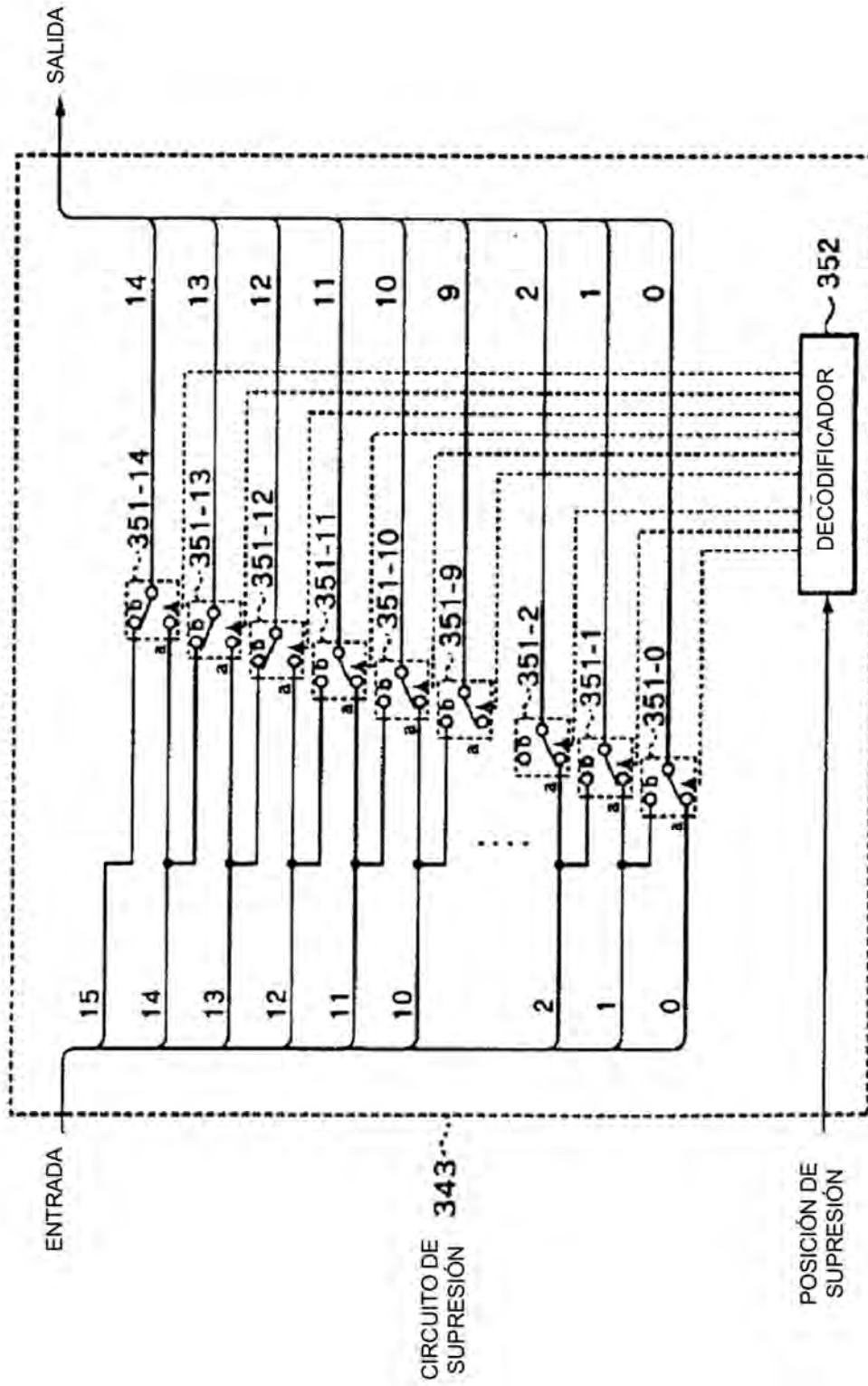


FIG. 26

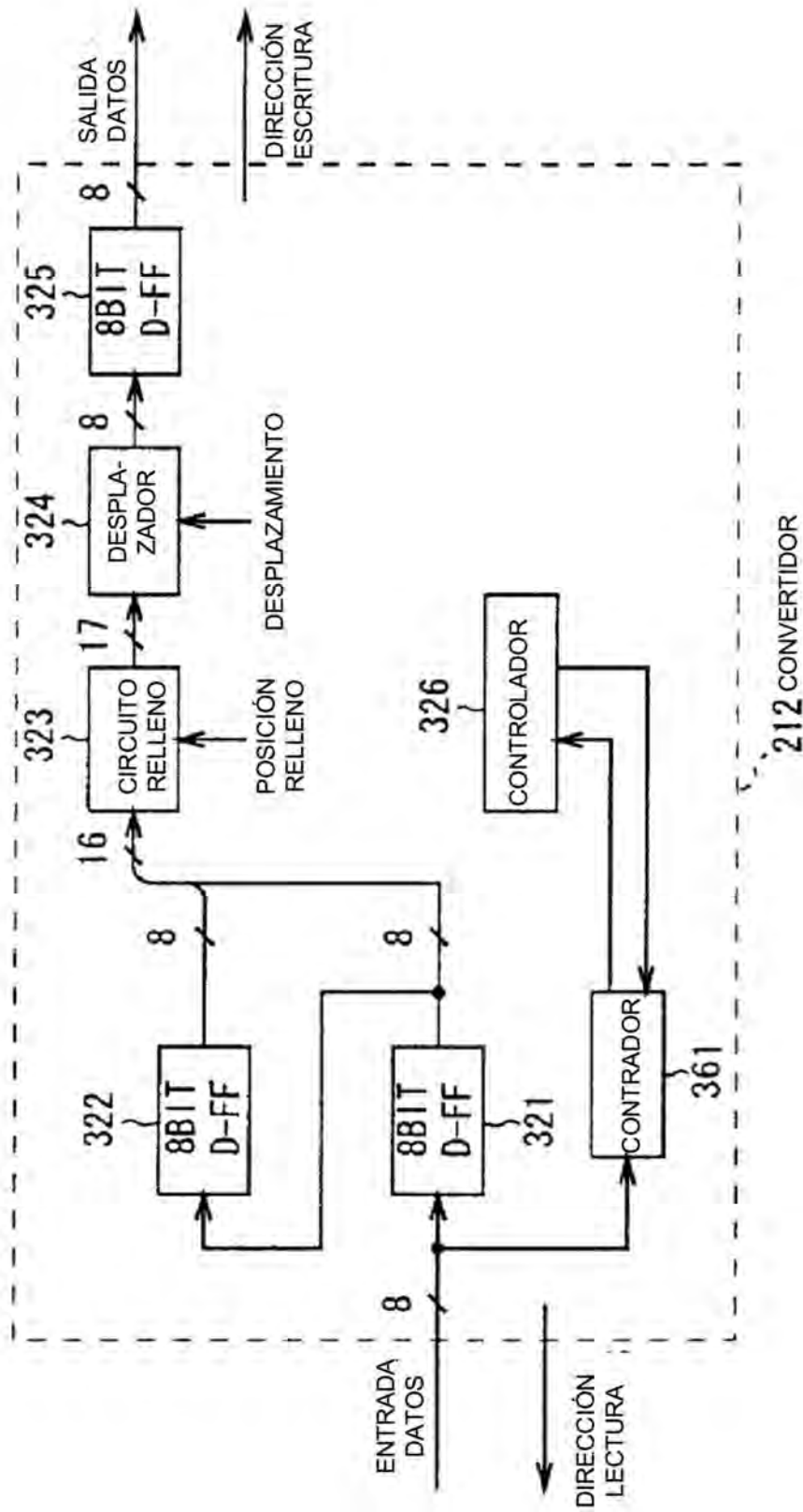


FIG. 27

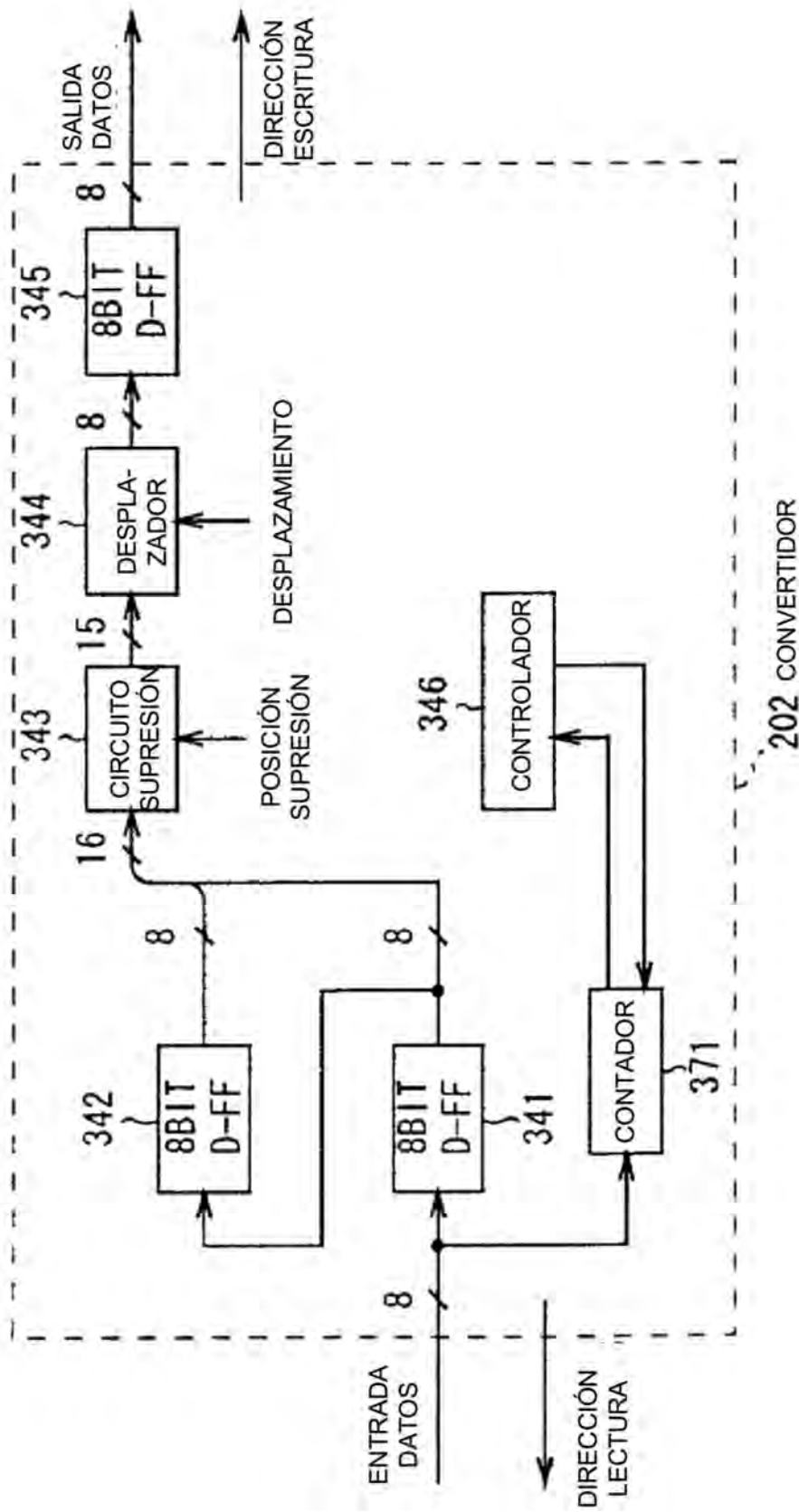


FIG. 28

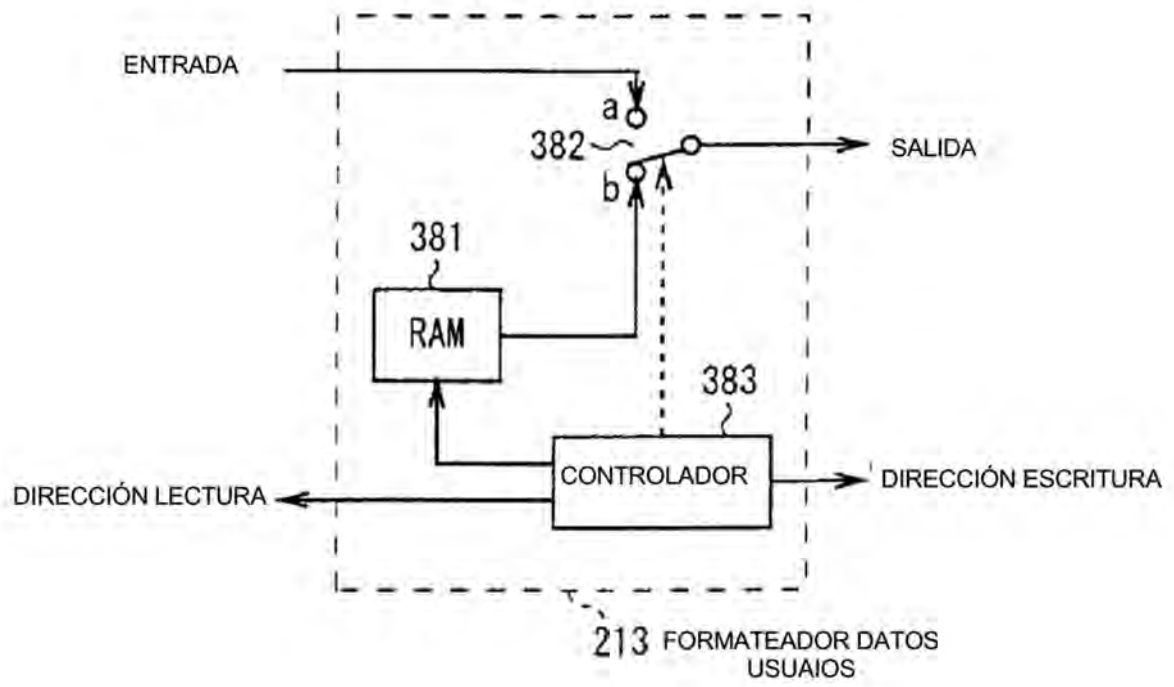


FIG. 29

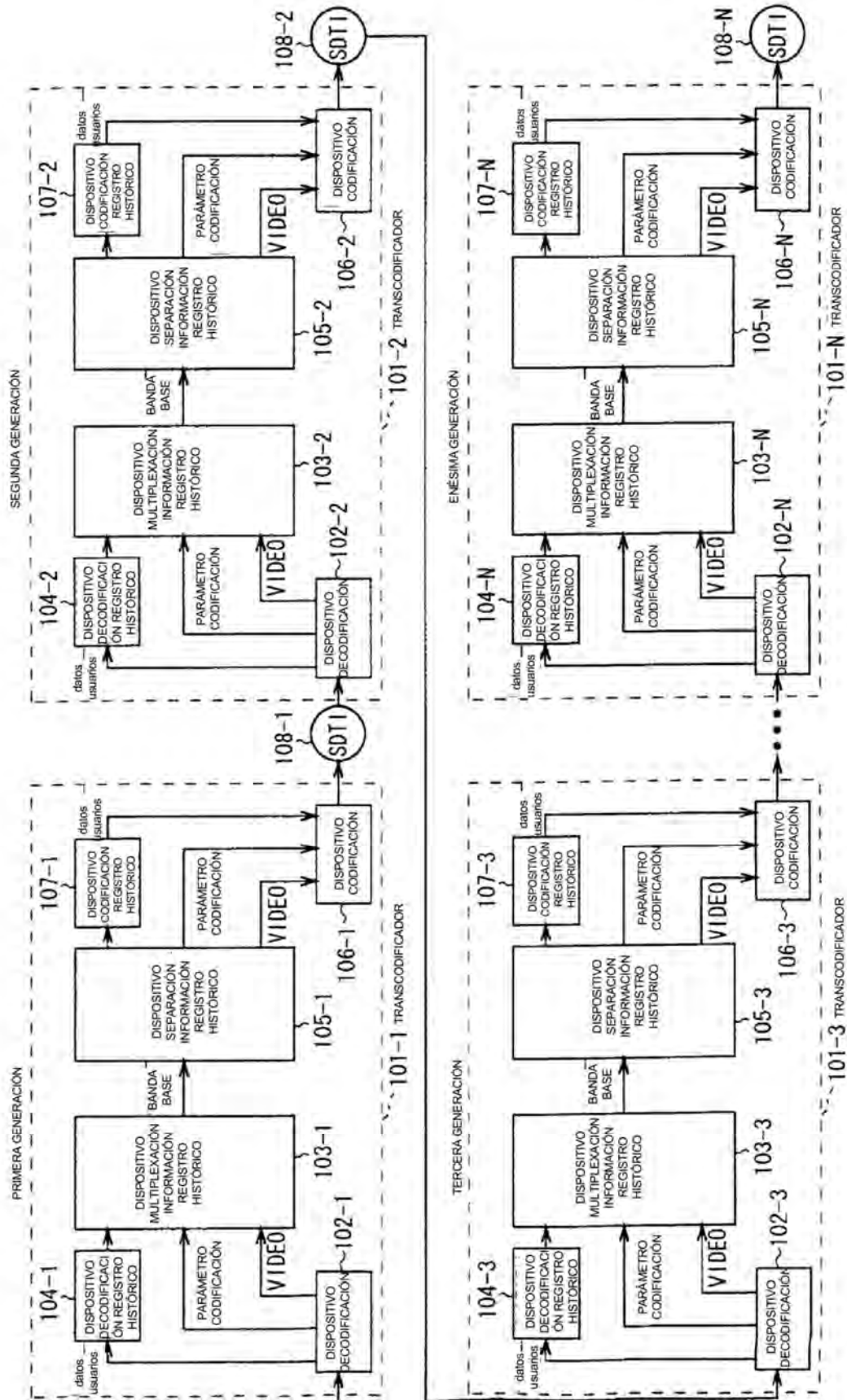


FIG. 30

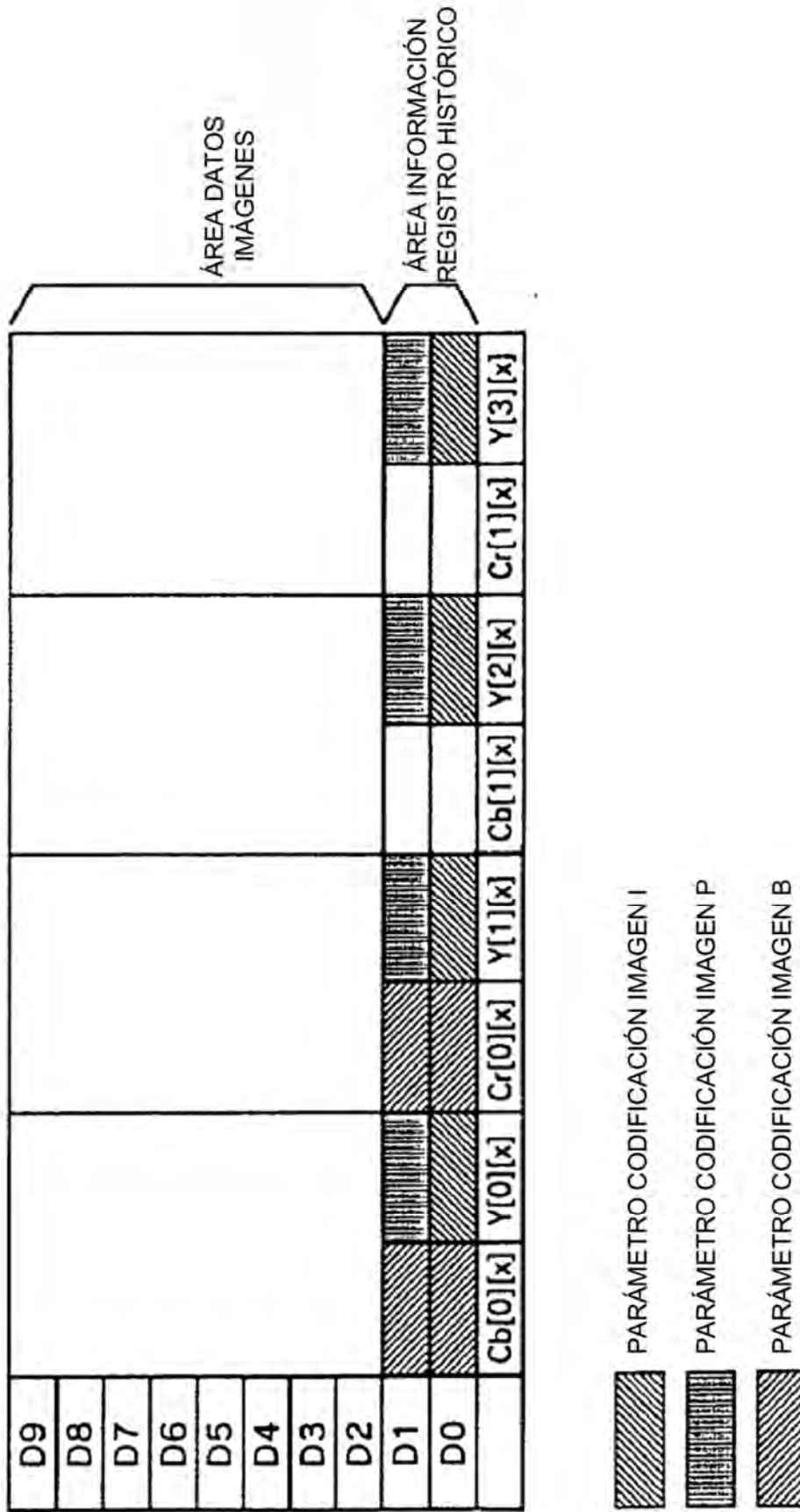


FIG. 31

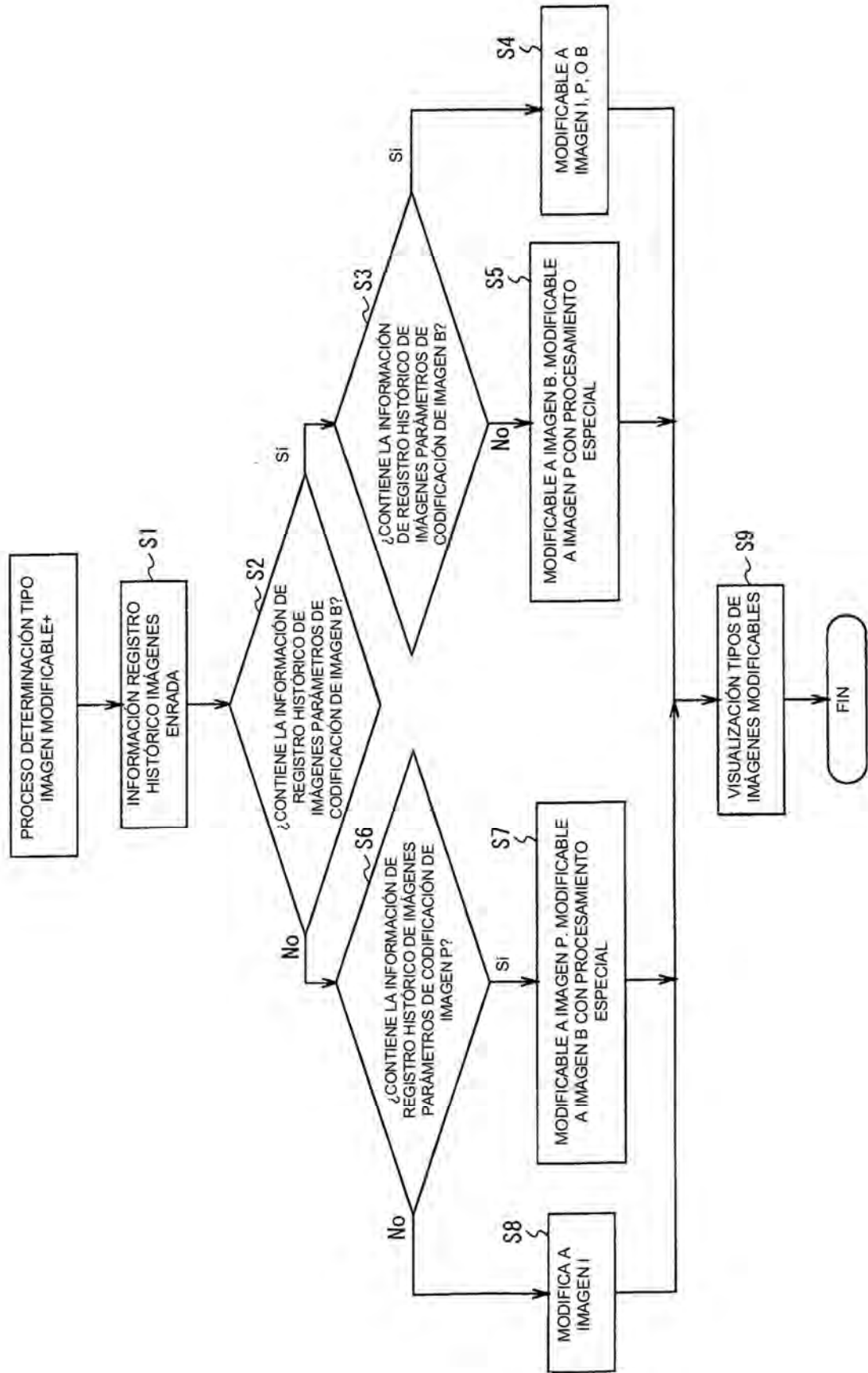


FIG. 32



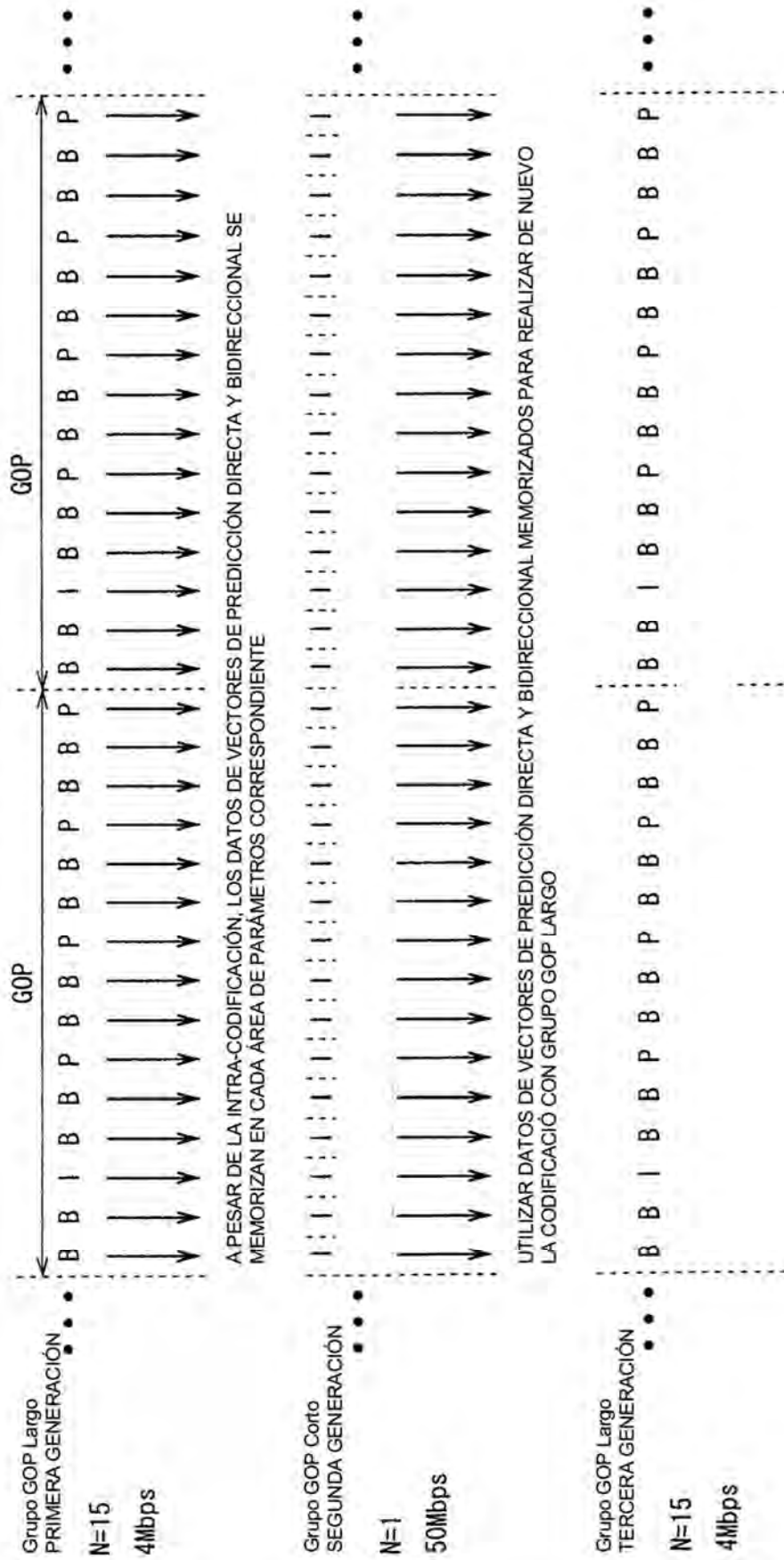


FIG. 33

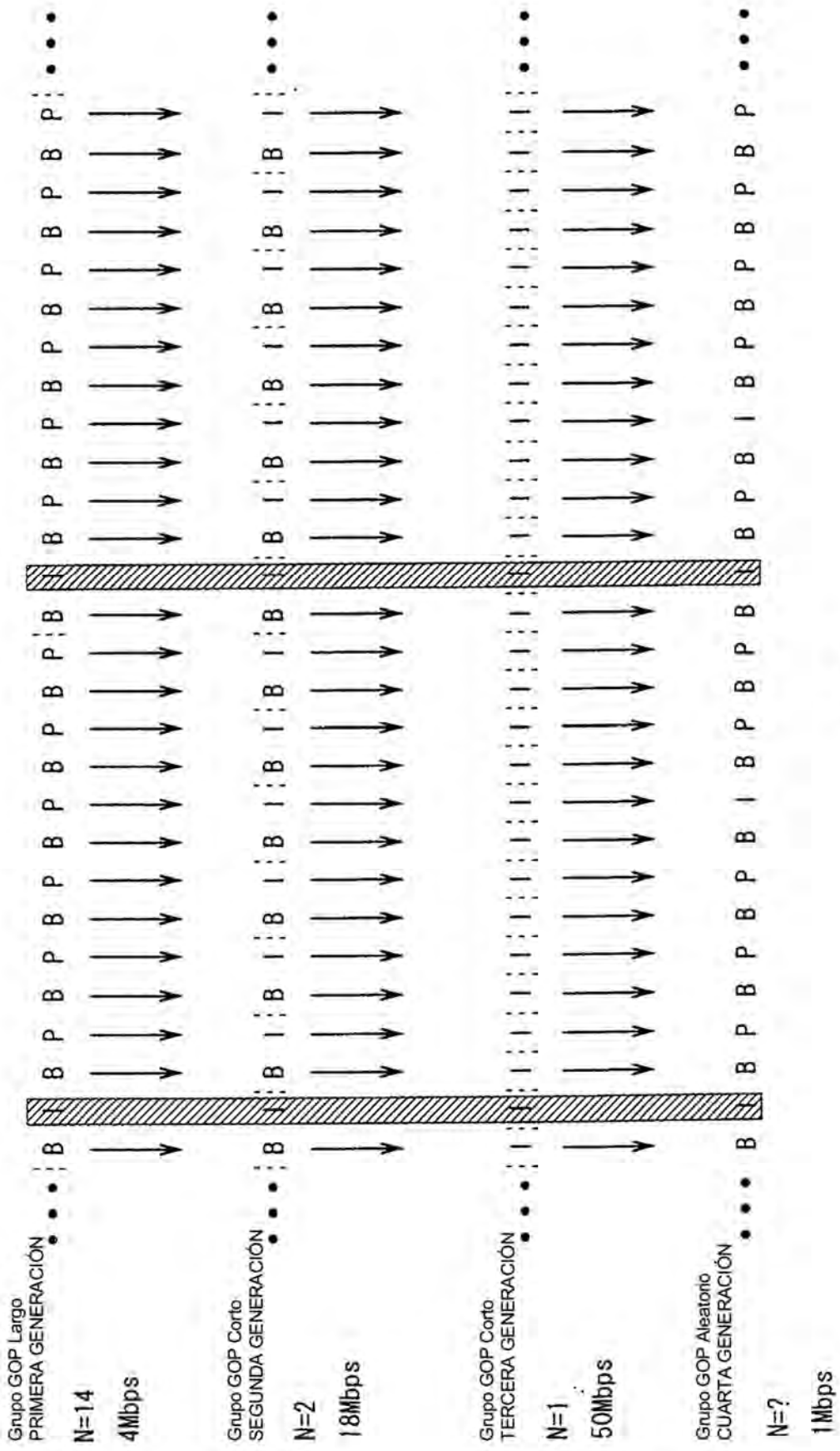


FIG. 34

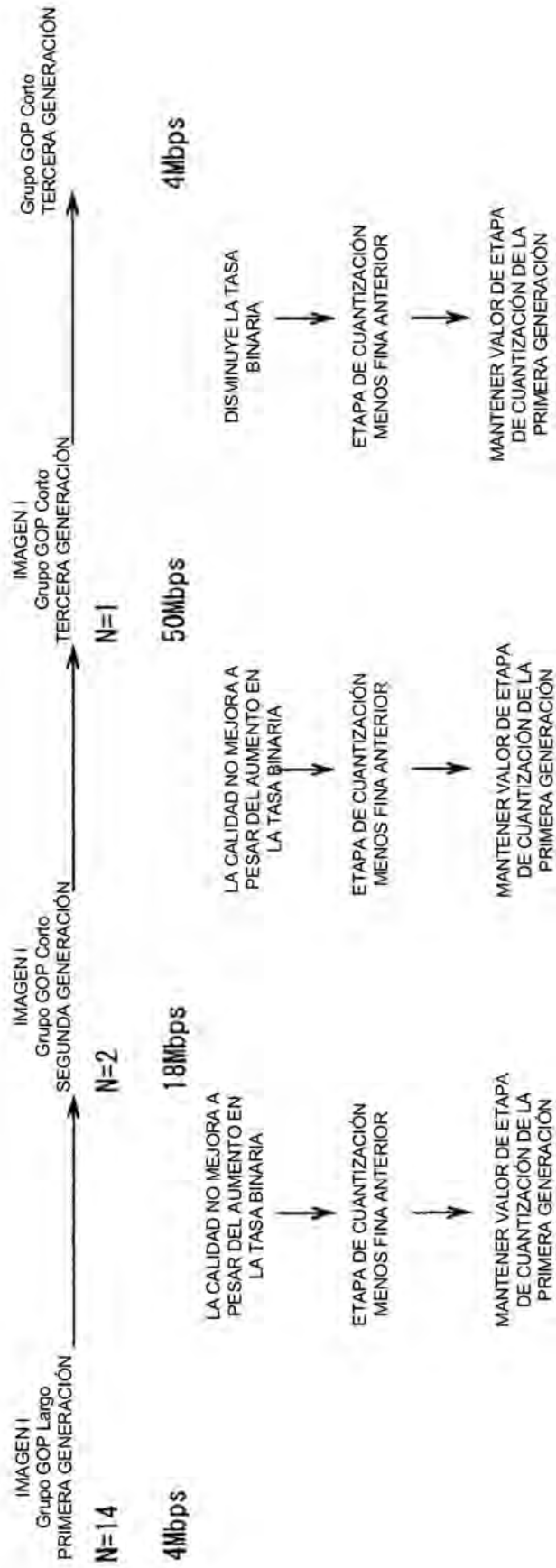


FIG. 35

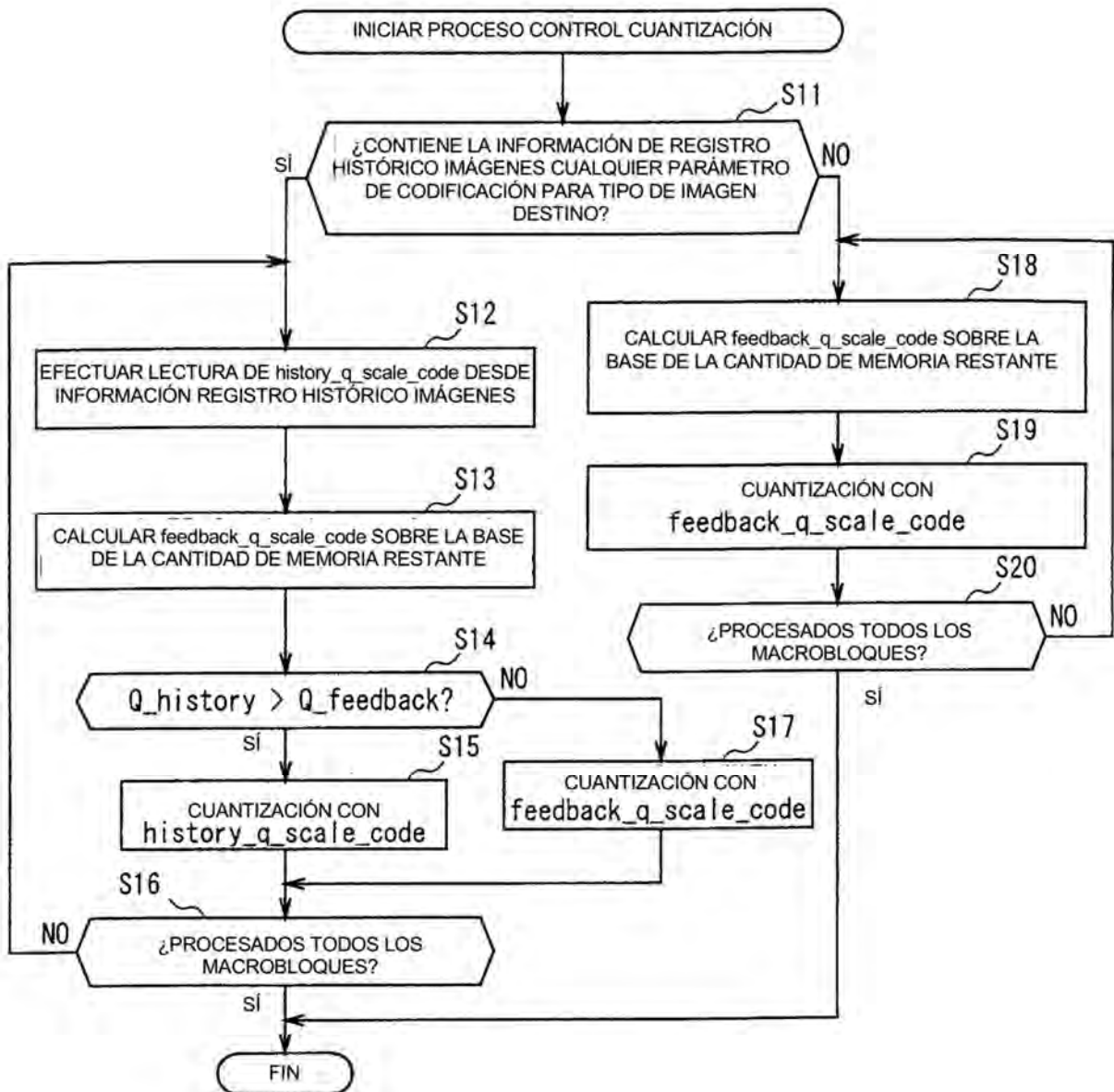


FIG. 36

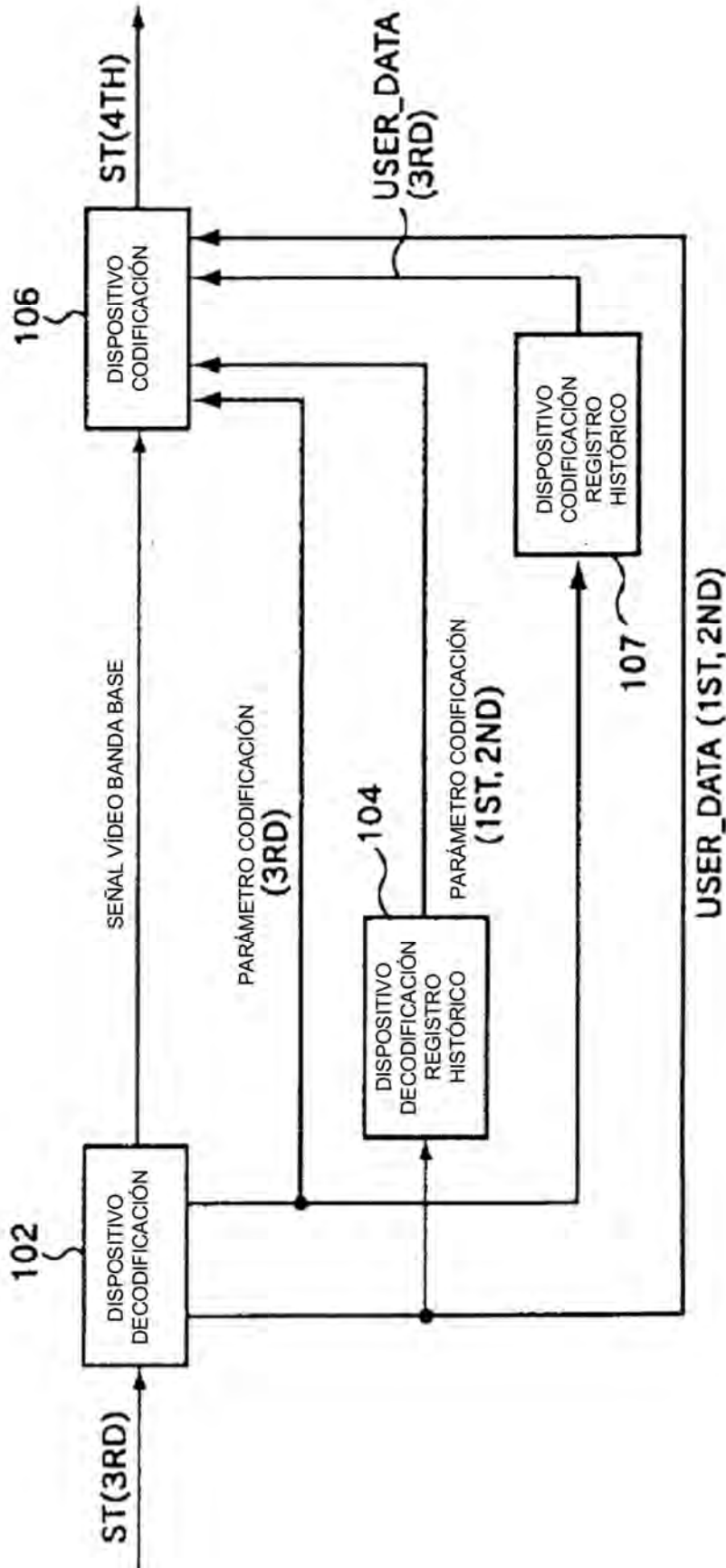


FIG. 37

flujo con datos de registro histórico

	Nº de bits	Mnemónico
video_sequence ( ) {		
next_start_code ( )		
sequence_header ( )		
sequence_extension ( )		
do {		
extension_and_user_data (0)		
do {		
if(nextbits( ) == group_start_code){		
group_of_pictures_header(1)		
extension_and_user_data (1)		
}		
picture_header( )		
picture_coding_extension( )		
while((nextbits( ) == extension_start_code)		
(nextbits( ) == user_data_start_code)){		
if(nextbits( ) == extension_start_code)		
extension_data(2)		
if(nextbits( ) == user_data_start_code){		
user_data_start_code	32	bslbf
if(nextbits( ) == History_Data_ID){		
History_Data_ID	8	bslbf
converted_history_stream( )		
}		
else{		
user_data( )		
}		
}		
}		
picture_data( )		
}while((nextbits( ) == picture_start_code)		
(nextbits( ) == groupe_start_code))		
if(nextbits( ) != sequence_end_code){		
sequence_header ( )		
sequence_extension ( )		
}		
}while(nextbits( ) != sequence_end_code)		
sequence_end_code		
}	32	bslbf

FIG. 38

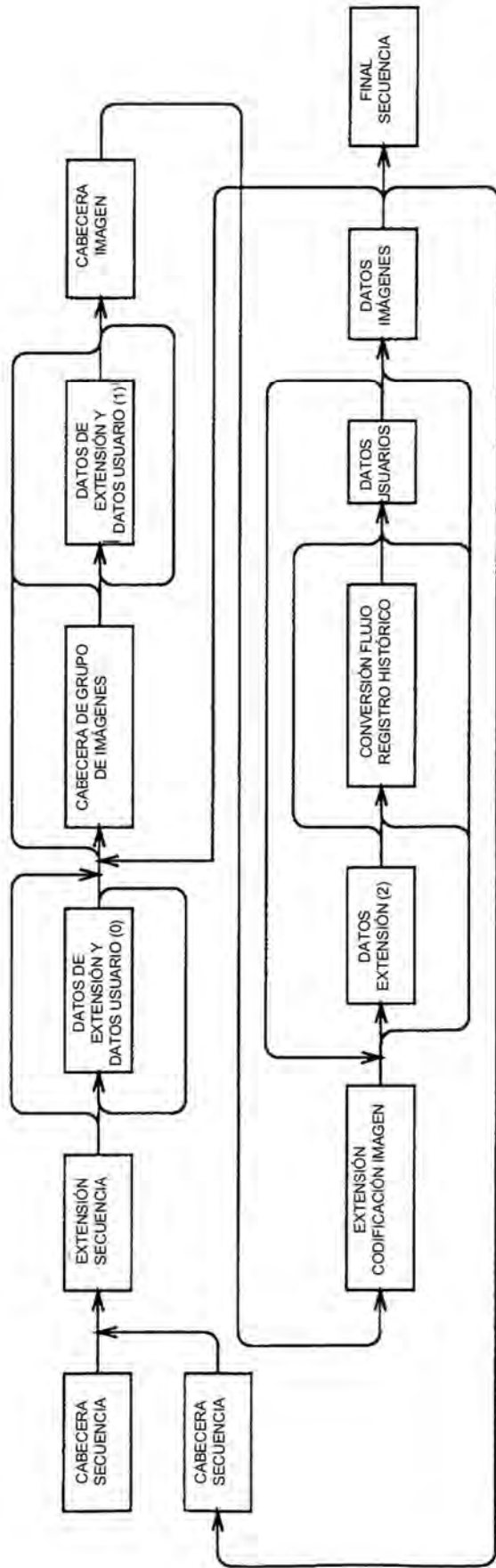


FIG. 39

Flujo de registro histórico (40-1)

history_stream( ) {	Bits	Valor
sequence_header		
sequence_header_code	32	000001B3
sequence_header_present_flag	1	
horizontal_size_value	12	
marker_bit	1	1
vertical_size_value	12	
aspect_ratio_information	4	
frame_rate_code	4	
marker_bit	1	1
bit_rate_value	18	
marker_bit	1	1
vbv_buffer_size_value	10	
constrained_parameter_flag	1	0
load_intra_quantiser_matrix	1	
load_non_intra_quantiser_matrix	1	
marker_bits	5	1F
intra_quantiser_matrix[64]	8*64	
non_intra_quantiser_matrix[64]	8*64	
sequence_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	1
sequence_extension_present_flag	1	
profile_and_level_indication	8	
progressive_sequence	1	
chroma_format	2	
horizontal_size_extension	2	
vertical_size_extension	2	
marker_bit	1	1
bit_rate_extension	12	
vbv_buffer_size_extension	8	
low_delay	1	
marker_bit	1	1

FIG. 40



Flujo de registro histórico (40-2)

	Bits	Valor
frame_rate_extension_n	2	
frame_rate_extension_d	5	
marker_bits	6	3F
sequence_display_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	2
sequence_display_extension_present_flag	1	
video_format	3	
colour_description	1	
colour_primaries	8	
transfer_characteristics	8	
marker_bit	1	1
matrix_coefficients	8	
display_horizontal_size	14	
marker_bit	1	1
display_vertical_size	14	
marker_bit	1	1
macroblock_assignment_in_user_data		
macroblock_assignment_present_flag	1	
marker_bit	7	7F
v_phase	8	
h_phase	8	
group_of_picture_header		
group_start_code	32	000001B8
group_of_picture_header_present_flag	1	
time_code	25	
closed_gop	1	
broken_link	1	
marker_bits	4	F
picture_header		
picture_start_code	32	00000100

FIG. 41

Flujo de registro histórico (40-3)

	Bits	Valor
temporal_reference	10	
picture_coding_type	3	
marker_bit	1	1
vbv_delay	16	
full_pel_forward_vector	1	
forward_f_code	3	
full_pel_backward_vector	1	
marker_bit	1	1
backward_f_code	3	
marker_bit	1	1
picture_coding_extension		
extension_start_code	32	000001B5
extension_start_code_identifer	4	8
f_code[0][0]	4	
f_code[0][1]	4	-
f_code[1][0]	4	
f_code[1][1]	4	
intra_dc_precision	2	
picture_structure	2	
top_field_first	1	
frame_pred_frame_dct	1	
concealment_motion_vectors	1	
q_scale_type	1	
marker_bit	1	1
intra_vlc_format	1	
alternate_scan	1	
repeat_first_field	1	
chroma_420_type	1	
progressive_frame	1	
composite_display_flag	1	
v_axis	1	
field_sequence	3	
sub_carrier	1	
burst_amplitude	7	

FIG. 42

Flujo de registro histórico (40-4)-

	Bits	Valor
marker_bits	1	1
sub_carrier_phase	8	
quant_matrix_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	3
quant_matrix_extension_present_flag	1	
load_intra_quantiser_matrix	1	
marker_bits	2	3
intra_quantiser_matrix[64]	8*64	
load_non_intra_quantiser_matrix	1	
marker_bits	7	7F
non_intra_quantiser_matrix[64]	8*64	
load_chroma_intra_quantiser_matrix	1	
marker_bits	7	7F
chroma_intra_quantiser_matrix[64]	8*64	
load_chroma_non_intra_quantiser_matrix	1	
marker_bits	7	7F
chroma_non_intra_quantiser_matrix[64]	8*64	
copyright_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	4
copyright_extension_present_flag	1	
copyright_flag	1	
copyright_identifier	8	
original_or_copy	1	
marker_bit	1	
copyright_number_1	20	
marker_bit	1	
copyright_number_2	22	
marker_bit	1	
copyright_number_3	22	3F
marker_bits	6	

FIG. 43

Flujo de registro histórico (40-5)

	Bits	Valor
picture_display_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	7
picture_display_extension_present_flag	1	
frame_centre_horizontal_offset_1	16	
marker_bit	1	1
frame_centre_vertical_offset_1	16	
marker_bit	1	1
frame_centre_horizontal_offset_2	16	
marker_bit	1	1
frame_centre_vertical_offset_2	16	
marker_bit	1	1
frame_centre_horizontal_offset_3	16	
marker_bit	1	1
frame_centre_vertical_offset_3	16	
marker_bit	6	3F
re_coding_stream_information		
user_data_start_code	32	000001B2
re_coding_stream_info_ID	16	91EC
red_bw_flag	1	
red_bw_indicator	2	
marker_bit	5	1F
user_data		
user_data_start_code	32	000001B2
user_data	2048	
while(macroblock i=macroblock_count){		
macroblock		
macroblock_address_h	8	
macroblock_address_v	8	
slice_header_present_flag	1	
skipped_macroblock_flag	1	
marker_bit	1	1
macroblock_modes( )		
macroblock_quant	1	
macroblock_motion_forward	1	
macroblock_motion_backward	1	
macroblock_pattern	1	
macroblock_intra	1	

FIG. 44

Flujo de registro histórico (40-6)

	Bits	Valor
spatial_temporal_weight_code_flag	1	
frame_motion_type	2	
field_motion_type	2	
dct_type	1	
marker_bits	2	3
quantiser_scale_code	5	
marker_bits	3	7
PMV[0][0][0]	14	
marker_bits	2	3
PMV[0][0][1]	14	
motion_vertical_field_select[0][0]	1	
marker_bit	1	1
PMV[0][1][0]	14	
marker_bits	2	3
PMV[0][1][1]	14	
motion_vertical_field_select[0][1]	1	
marker_bit	1	1
PMV[1][0][0]	14	
marker_bits	2	3
PMV[1][0][1]	14	
motion_vertical_field_select[1][0]	1	
marker_bit	1	1
PMV[1][1][0]	14	
marker_bits	2	3
PMV[1][1][1]	14	
motion_vertical_field_select[1][1]	1	
marker_bit	1	1
coded_block_pattern	12	
marker_bits	4	F
num_mv_bits	8	
num_coef_bits	14	
marker_bits	2	3

FIG. 45

Flujo de registro histórico (40-7)

	Bits	Valor
num_other_bits	7	
marker_bit	1	1
}		

FIG. 46

history_stream( ){	Nº de bits	Mnemónico
next_start_code ( )		
sequence_header ( )		
sequence_extension ( )		
extension_and_user_data (0)		
if(nextbits( ) == group_start_code){		
group_of_pictures_header( )		
extension_and_user_data (1)		
}		
picture_header( )		
picture_coding_extension( )		
re_coding_stream_info( )		
extensions_and_user_data (2)		
picture_data( )		
sequence_end_code	32	bslbf
}		

FIG. 47

sequence_header( ) {	Nº de bits	Mnemónico
sequence_header_code ( )	32	bslbf
horizontal_size_value	12	uimsbf
vertical_size_value	12	uimsbf
aspect_ratio_information	4	uimsbf
frame_rate_code	4	uimsbf
bit_rate_value	18	uimsbf
marker_bit	1	bslbf
vbv_buffer_size_value	10	uimsbf
constrained_parameters_flag	1	bslbf
load_intra_quantiser_matrix	1	uimsbf
if(load_intra_quantiser_matrix)		
intra_quantiser_matrix[64]	8*64	uimsbf
load_non_intra_quantiser_matrix	1	uimsbf
if(load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix[64]	8*64	uimsbf
next_start_code ( )		
}		

FIG. 48

sequence_extension( ){	Nº de bits	Mnemónico
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
profile_and_level_indication	8	uimsbf
progressive_sequence	1	uimsbf
chroma_format	2	uimsbf
horizontal_size_extension	2	uimsbf
vertical_size_extension	2	uimsbf
bit_rate_extension	12	uimsbf
marker_bit	1	bslbf
vbv_buffer_size_extension	8	uimsbf
low_delay	1	uimsbf
frame_rate_extension_n	2	uimsbf
frame_rate_extension_d	5	uimsbf
next_start_code ( )		
}		

FIG. 49



extension_and_user_data (i){	Nº de bits	Mnemónico
while((nextbits( ) == extension_start_code)		
(nextbits( ) == user_data_start_code)){		
if((i==2) && (nextbits( ) == extension_start_code))		
extension_data( )		
if(nextbits( ) == user_data_start_code)		
user_data( )		
}		
}		

FIG. 50

user_data ( ) {	Nº de bits	Mnemónico
user_data_start_code	32	bslbf
while(nextbits( ) != '0000 0000 0000 0000 0000 0001'){		
user_data	8	uimsbf
}		
next_start_code( )		
}		

FIG. 51

group_of_pictures_header( ){	Nº de bits	Mnemónico
group_start_code	32	bslbf
time_code	25	bslbf
closed_gop	1	uimsbf
broken_link	1	uimsbf
next_start_code( )		
}		

FIG. 52

picture_header( ){	Nº de bits	Mnemónico
picture_start_code	32	bslbf
temporal_reference	10	uimsbf
picture_coding_type	3	uimsbf
vbv_delay	16	uimsbf
if(picture_coding_type==2    picture_coding_type==3){		
full_pel_forward_vector	1	bslbf
forward_f_code	3	bslbf
}		
if(picture_coding_type==3){		
full_pel_forward_vector	1	bslbf
backward_f_code	3	bslbf
}		
while(nextbits( )=='1'){		
extra_bit_picture/*con el valor '1'*/	1	uimsbf
extra_information_picture	8	uimsbf
}		
extra_bit_picture/*con el valor '0'*/	1	uimsbf
next_start_code( )		
}		

FIG. 53

picture_coding_extension( ) {	Nº de bits	Mnemónico
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
f_code [0] [0] /*horizontal directa*/	4	uimsbf
f_code [0] [1] /*vertical directa*/	4	uimsbf
f_code [1] [0] /*horizontal inversa*/	4	uimsbf
f_code [1] [1] /*vertical inversa*/	4	uimsbf
intra_dc_precision	2	uimsbf
picture_structure	2	uimsbf
top_field_first	1	uimsbf
frame_pred_frame_dct	1	uimsbf
concealment_motion_vectors	1	uimsbf
q_scale_type	1	uimsbf
intra_vlc_format	1	uimsbf
alternate_scan	1	uimsbf
repeat_first_field	1	uimsbf
chroma_420_type	1	uimsbf
progressive_frame	1	uimsbf
composite_display_flag	1	uimsbf
if(composite_display_flag){		
v_axis	1	uimsbf
field_sequence	3	uimsbf
sub_carrier	1	uimsbf
burst_amplitude	7	uimsbf
sub_carrier_phase	8	uimsbf
}		
next_start_code( )		
}		

FIG. 54

extension_data( ){	Nº de bits	Mnemónico
while(nextbits( )==extension_start_code) {		
extension_start_code	32	bslbf
if (nextbits( )=="ID Extensión Matriz Cuantiz.")		
quant_matrix_extension( )		
else if (nextbits( )=="ID Extensión Copyright")		
copyright_extension( )		
else		
picture_display_extension( )		
}		
}		

FIG. 55

quant_matrix_extension( ){	Nº de bits	Mnemónico
extension_start_code_identifier	4	uimsbf
load_intra_quantiser_matrix	1	uimsbf
if (load_intra_quantiser_matrix)		
intra_quantiser_matrix [64]	8*64	uimsbf
load_non_intra_quantiser_matrix	1	uimsbf
if (load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix [64]	8*64	uimsbf
load_chroma_intra_quantiser_matrix	1	uimsbf
if (load_chroma_intra_quantiser_matrix)		
chroma_intra_quantiser_matrix [64]	8*64	uimsbf
load_chroma_non_intra_quantiser_matrix	1	uimsbf
if (load_chroma_non_intra_quantiser_matrix)		
chroma_non_intra_quantiser_matrix [64]	8*64	uimsbf
next_start_code( )		
}		

FIG. 56

copyright_extension( ) {	Nº de bits	Mnemónico
extension_start_code_identifier	4	uimsbf
copyright_flag	1	bslbf
copyright_identifier	8	uimsbf
original_or_copy	1	bslbf
reserved	7	uimsbf
marker_bit	1	bslbf
copyright_number_1	20	uimsbf
marker_bit	1	bslbf
copyright_number_2	22	uimsbf
marker_bit	1	bslbf
copyright_number_3	22	uimsbf
next_start_code( )		
}		

FIG. 57

picture_display_extension( ) {	Nº de bits	Mnemónico
extension_start_code_identifier	4	uimsbf
for(i=0;i<number_of_frame_centre_offsets;i++) {		
frame_centre_horizontal_offset	16	simsbf
marker_bit	1	bslbf
frame_centre_vertical_offset	16	simsbf
marker_bit	1	bslbf
}		
next_start_code( )		
}		

FIG. 58

	Nº de bits	Mnemónico
picture_data( ) {		
while(nextbits( )==slice_start_code) {		
slice( )		
}		
next_start_code( )		
}		

FIG. 59

	Nº de bits	Mnemónico
slice( ) {		
slice_start_code	32	bslbf
slice_quantiser_scale_code	5	uimsbf
if (nextbit( )=='1') {		
intra_slice_flag	1	bslbf
intra_slice	1	uimsbf
reserved_bits	7	uimsbf
while(nextbits( )=='1') {		
extra_bit_slice/*con el valor '1'*/	1	uimsbf
extra_information_slice	8	uimsbf
}		
}		
extra_bit_slice/*con el valor '0'*/	1	uimsbf
do {		
macroblock( )		
} while(nextbit( )!='000 0000 0000 0000 0000 0000')		
next_start_code( )		
}		

FIG. 60

macrobloque( ){	Nº de bits	Mnemónico
while(nextbits( )=='0000 0001 000')		
macroblock_escape	11	bslbf
macroblock_address_increment	1-11	viclbf
macroblock_modes( )		
if (macroblock_quant)		
macroblock_quantiser_scale_code	5	uimsbf
if (macroblock_motion_forward    (macroblock_intra && concealment_motion_vectors))		
motion_vectors(0)		
if (macroblock_motion_backward)		
motion_vectors(1)		
if (macroblock_intra && concealment_motion_vectors)		
marker_bit	1	bslbf
}		
}		

FIG. 61

	Nº de bits	Mnemónico
macroblock_modes( ) {		
macroblock_type	1-9	vlc1bf
if (macroblock_motion_forward		
macroblock_motion_backward) {		
if (picture_structure== 'trama') {		
if (frame_pred_frame_dct==0)		
frame_motion_type	2	uimsbf
} else {		
field_motion_type	2	uimsbf
}		
}		
if ((picture_structure=="Imagen tramas") &&		
(frame_pred_frame_dct==0) &&		
(dct_type_flag==1)) {		
dct_type	1	uimsbf
}		
}		

FIG. 62



motion_vectors(s) {	Nº de bits	Mnemónico
if (motion_vector_count==1) {		
if ((mv_format==field)&&(dmv!=1)		
motion_vertical_field_select [0][s]	1	uimsbf
motion_vector(0,s)		
} else {		
motion_vertical_field_select [0][s]	1	uimsbf
motion_vector(0,s)		
motion_vertical_field_select [1][s]	1	uimsbf
motion_vector(1,s)		
}		
}		

FIG. 63

motion_vectors(r, s) {	Nº de bits	Mnemónico
motion_code[r][s][0]	1-11	vlclbf
if ((f_code[s][0]!=1)&&(motion_code[r][s][0]!=0)		
motion_residual [r][s][0]	1-8	uimsbf
if (dmv==1)		
dmvector [0]	1-2	vlclbf
motion_code[r][s][1]	1-11	vlclbf
if ((f_code[s][1]!=1)&&(motion_code[r][s][1]!=0)		
motion_residual [r][s][1]	1-8	uimsbf
if (dmv==1)		
dmvector [1]	1-2	vlclbf
}		

FIG. 64

macroblock_type VLC code					
	macroblock_quant				
		dct_type_flag			
			macroblock_motion_forward		
			macroblock_motion_backward		
				Descripción	
1	0	1	0	0	Intra
01	1	1	0	0	Intra, Quant

FIG. 65

macroblock_type VLC code					
	macroblock_quant				
		dct_type_flag			
			macroblock_motion_forward		
			macroblock_motion_backward		
				Descripción	
1	0	1	1	0	MC, Coded
01	0	1	0	0	No MC, Coded
001	0	0	0	0	MC, Not Coded
0001 1	0	1	0	0	Intra
0001 0	1	1	1	0	MC, Coded, Quant
0000 1	1	1	0	0	No MC, Coded, Quant
0000 01	1	1	0	0	Intra, Quant

FIG. 66

macroblock_type V L C code					
	macroblock_quant				
	dct_type_flag				
	macroblock_motion_forward				
	macroblock_motion_backward				
	Descripción				
10	0	0	0	0	Interp, Not Coded
11	0	1	1	1	Interp, Coded
010	0	0	0	0	Bwd, Not Coded
011	0	1	0	1	Bwd, Coded
0010	0	0	0	0	Fwd, Not Coded
0011	0	1	1	0	Fwd, Coded
0001 1	0	1	0	0	Intra
0001 0	1	1	1	1	Interp, Coded, Quant
0000 11	1	1	1	0	Fwd, Coded, Quant
0000 10	1	1	0	1	Bwd, Coded, Quant
0000 01	1	1	0	0	Intra, Quant

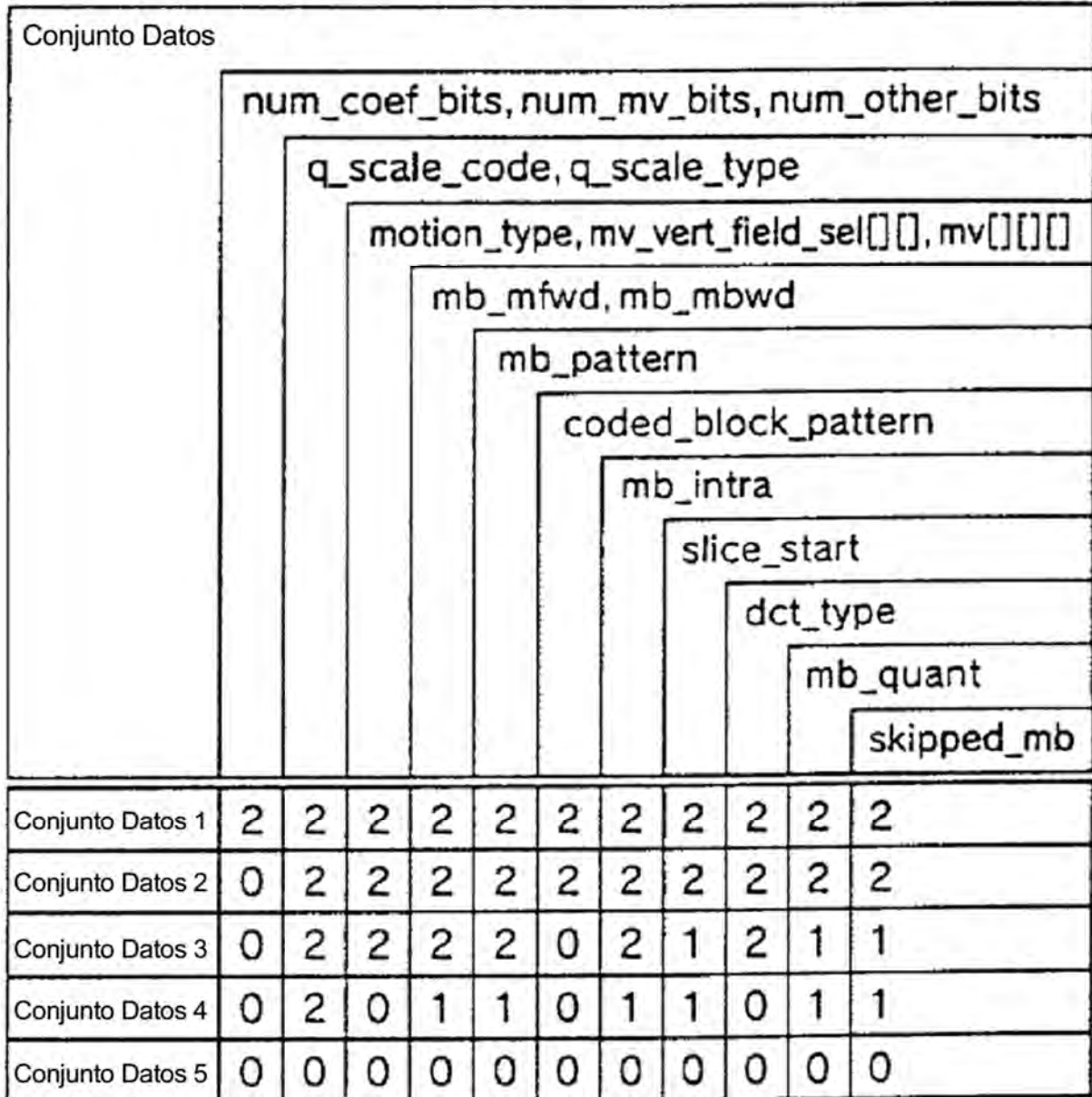
FIG. 67

re_coding_stream_info ( ) {	Nº de bits	Mnemónico
user_data_start_code	32	bslbf
re_coding_stream_info_ID	16	bslbf
red_bw_flag	1	uimsbf
if (red_bw_flag)		
red_bw_indicator	2	uimsbf
if (!red_bw_flag) {		
for(i=0;i<number_of_macroblock:++) {		
marker_bit	3	bslbf
num_other_bits	7	uimsbf
num_mv_bits	8	uimsbf
num_coef_bits	14	uimsbf
}		
}		
next_start_code( )		
}		

FIG. 68

Conjunto Datos	red_bw_flag	
	red_bw_flag	red_bw_indicator
Conjunto Datos 1	0	-
Conjunto Datos 2	1	0
Conjunto Datos 3	1	1
Conjunto Datos 4	1	2
Conjunto Datos 5	1	3

FIG. 69



COMBINACIÓN DE ELEMENTOS DE  
INFORMACIÓN DE REGISTRO  
HISTÓRICO

FIG. 70

Línea	Codificación tramas	Codificación campos
0	0	SRIB_sync_code=11112
1	1	Reservado
2	2	rolling_SRIB_mb_ref [15:0]
3	3	picrate_element [picrate_element_index][31:16]
4	4	picrate_element [picrate_element_index][15:0]
5	5	mb_quant [mb_mtwid mb_inbwid]
6	6	mb_pattern [mb_intra mb_start_flag]
7	7	DCT_type [motion_type]
8	8	skip_mb [skip_mb_type]
9	9	Reservado
10	10	coded_block_pattern [7:0]
11	11	Reservado
12	12	coded_block_pattern [3:0]
13	13	Reservado
14	14	num_mv_bits [1:0]
15	15	Reservado

FIG. 71

Elementos de tasa de imágenes (72-1)

Parámetro	Formato numérico	Número de bits	Offset bits desde	Offset bits a	Cat. datos	Detalles
Indicador estándar MPEG	1bit flag	1	0	0	3	1=>MPEG 1:0=>MPEG2
red_bw_flag	1bit flag	1	1	1	3	Por defecto = "0"
red_bw_indicator	2bit ui	3	2	4	3	Por defecto = "000"
Indicadores de cabeceras actuales	2bit flags	2	5	6	3	flag presente cabecera secuencia, flag presente cabecera GOP
Indicadores código inicio extensión	16 flags	16	7	22	3	Indica si existe un código de inicio de extensión dado. Los 16 flags corresponden a las 16 entradas en la tabla 6.2 del ISO/IEC 13818-2: la norma de 1996 en el orden en que están listados.
Otros códigos de inicio	3 flags	3	23	25	3	user_data_start_code, sequence_error_code, sequence_end_code
Cabecera de secuencia						
horizontal_size	14bit uimsbf	14	26	39	2	incluye extensión
vertical_size	14bit uimsbf	14	40	53	2	incluye extensión
aspect_ratio_information	4bit uimsbf	4	54	57	2	
frame_rate_code	4bit uimsbf	4	58	61	2	
bit_rate	30bit uimsbf	30	62	91	2	incluye extensión, debe añadirse valor correcto
vbv_buffer_size	18bit uimsbf	18	92	109	2	incluye extensión
constrained_parameters_flag	1bit flag	1	110	110	2	
Extensión de secuencia						
profile_and_level_indication	8bit uimsbf	8	111	118	2	
progressive_sequence	1bit flag	1	119	119	2	
chroma_format	2bit uimsbf	2	120	121	2	
low_delay	1bit flag	1	122	122	2	
Extensión visualización secuencia						
video_format	3bit uimsbf	3	123	125	2	
colour_description	1bit flag	1	126	126	2	
colour primaries	8bit uimsbf	8	127	134	2	
transfer_characteristics	8bit uimsbf	8	135	142	2	
matrix_coefficients	8bit uimsbf	8	143	150	2	
display_horizontal_size	14bit uimsbf	14	151	164	2	
display_vertical_size	14bit uimsbf	14	165	178	2	
Cabecera de grupo de imágenes						
time_code	25bit flag	25	179	203	2	
closed_gop	1bit flag	1	204	204	2	

FIG. 72

Elementos de tasa de imágenes (72-2)

broken_link	1bit flag	1	205	205	2	
Cabecera de imagen						
temporal_reference	10bit uimsbf	10	206	215	1	
picture_coding_type	3bit uimsbf	3	216	218	1	
vbv_delay	16bit uimsbf	16	219	234	1	Debe calcular si no está presente el flujo bits
full_pel_forward_vector	1bit flag	1	235	235	1	
forward_f_code	3bit uimsbf	3	236	238	1	
full_pel_backward_vector	1bit flag	1	239	239	1	
backward_f_code	3bit uimsbf	3	240	242	1	
Extensión codificación imagen						
forward_horizontal_f_code	4bit uimsbf	4	243	246	1	
forward_vertical_f_code	4bit uimsbf	4	247	250	1	
backward_horizontal_f_code	4bit uimsbf	4	251	254	1	
backward_vertical_f_code	4bit uimsbf	4	255	258	1	
intra_dc_precision	2bit uimsbf	2	259	260	1	
picture_structure	2bit uimsbf	2	261	262	1	
top_field_first	1bit flag	1	263	263	1	
frame_pred_frame_dct	1bit flag	1	264	264	1	
concealment_motion_vectors	1bit flag	1	265	265	1	
q_scale_type	1bit flag	1	266	266	1	
intra_vlc_format	1bit flag	1	267	267	1	
alternate_scan	1bit flag	1	268	268	1	
repeat_first_field	1bit flag	1	269	269	1	
chroma_420_type	1bit flag	1	270	270	1	
progressive_frame	1bit flag	1	271	271	1	
composite_display_flag	1bit flag	1	272	272	1	
v_axis	1bit flag	1	273	273	1	
field_sequence	3bit uimsbf	3	274	276	1	
sub_carrier	1bit flag	1	277	277	1	
burst_amplitude	7bit uimsbf	7	278	284	1	
sub_carrier_phase	8bit uimsbf	8	285	292	1	
Extensión matriz cuantización						
load_intra_quantiser_matrix	1bit flag	1	293	293	1	(1)
load_non_intra_quantiser_matrix	1bit flag	1	294	294	1	(1)
load_chroma_intra_quantiser_matrix	1bit flag	1	295	295	1	(1)
load_chroma_non_intra_quantiser_matrix	1bit flag	1	296	296	1	(1)
intra_quantiser_matrix[64]	64*0...255	512	297	808	2	(1)
non_intra_quantiser_matrix[64]	64*0...255	512	809	1320	2	(1)
chroma_intra_quantiser_matrix[64]	64*0...255	512	1321	1832	2	(1)
chroma_non_intra_quantiser_matrix[64]	64*0...255	512	1833	2344	2	(1)
extensión presentación imagen						
frame_centre_horizontal_offset_1	16bit uimsbf	16	2345	2360	2	
frame_centre_vertical_offset_1	16bit uimsbf	16	2361	2376	2	
frame_centre_horizontal_offset_2	16bit uimsbf	16	2377	2392	2	

FIG. 73



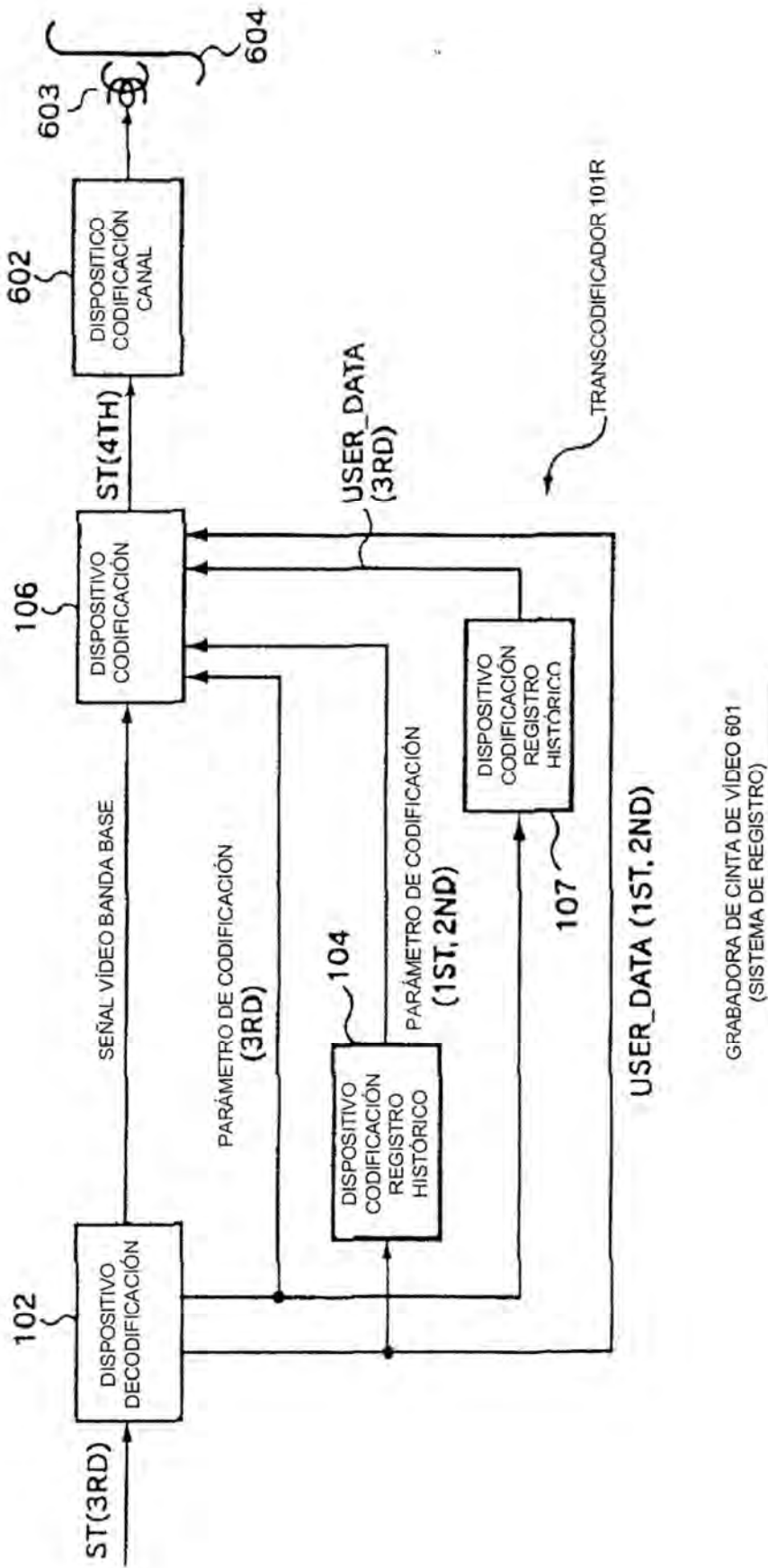
Elementos de tasa de imágenes (72-3)

frame_centre_vertical_offset_2	16bit uimsbf	16	2393	2408	2
frame_centre_horizontal_offset_3	16bit uimsbf	16	2409	2424	2
frame_centre_vertical_offset_3	16bit uimsbf	16	2425	2440	2
Extensión copyright					
copyright_flag	1bit flag	1	2441	2441	2
copyright_identifier	8bit code	8	2442	2449	2
original_or_copy	1bit flag	1	2450	2450	2
copyright_number	64bit uimsbf	64	2451	2514	2
PTS/DTS					
PTS_DTS_flag	2bit flag	2	2515	2516	1
PTS_Value	33bit uimsbf	33	2517	2549	2
DTS_Value	33bit uimsbf	33	2550	2582	2
bits reservados					
reserva	41bit uimsbf	41	2583	2623	
área datos usuarios					
user_data		1664	2624	4287	2
información tasa imágenes CRC					
32-bit_protection_CRT	32bit uimsbf	32	4288	4319	

FIG. 74

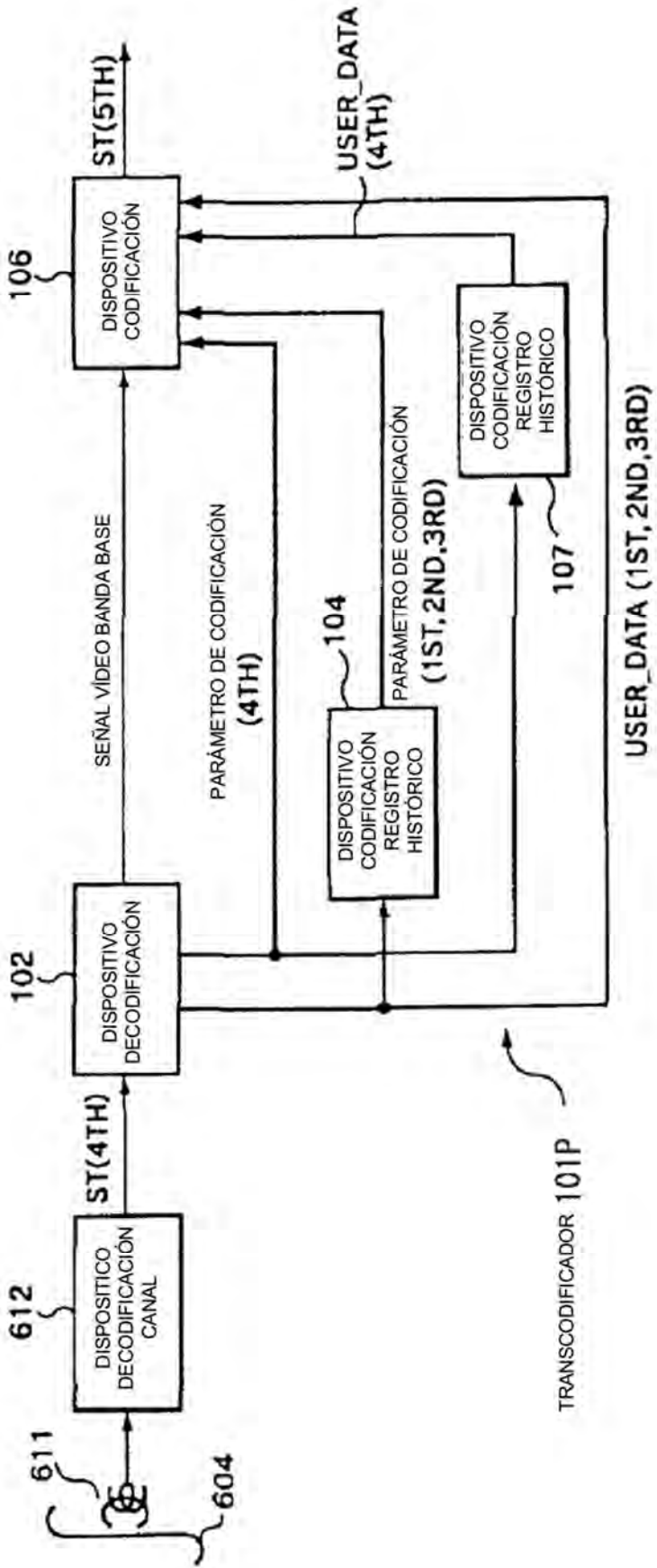
D9	Cb[0][9]	Y[0][9]	Cr[0][9]	Y[1][9]	Cb[1][9]	Y[2][9]	Cr[1][9]	Y[3][9]
D8	Cb[0][8]	Y[0][8]	Cr[0][8]	Y[1][8]	Cb[1][8]	Y[2][8]	Cr[1][8]	Y[3][8]
D7	Cb[0][7]	Y[0][7]	Cr[0][7]	Y[1][7]	Cb[1][7]	Y[2][7]	Cr[1][7]	Y[3][7]
D6	Cb[0][6]	Y[0][6]	Cr[0][6]	Y[1][6]	Cb[1][6]	Y[2][6]	Cr[1][6]	Y[3][6]
D5	Cb[0][5]	Y[0][5]	Cr[0][5]	Y[1][5]	Cb[1][5]	Y[2][5]	Cr[1][5]	Y[3][5]
D4	Cb[0][4]	Y[0][4]	Cr[0][4]	Y[1][4]	Cb[1][4]	Y[2][4]	Cr[1][4]	Y[3][4]
D3	Cb[0][3]	Y[0][3]	Cr[0][3]	Y[1][3]	Cb[1][3]	Y[2][3]	Cr[1][3]	Y[3][3]
D2	Cb[0][2]	Y[0][2]	Cr[0][2]	Y[1][2]	Cb[1][2]	Y[2][2]	Cr[1][2]	Y[3][2]
D1	Cb[0][1]	Y[0][1]	Cr[0][1]	Y[1][1]	Cb[1][1]	Y[2][1]	Cr[1][1]	Y[3][1]
D0	Integrado alineado en MPEG-2 Bus Información de Recodificación	Y[0][0]	Integrado alineado en MPEG-2 Bus Información de Recodificación	Y[1][0]	Integrado alineado en MPEG-2 Bus Información de Recodificación	Y[2][0]	Integrado alineado en MPEG-2 Bus Información de Recodificación	Y[3][0]

FIG. 75



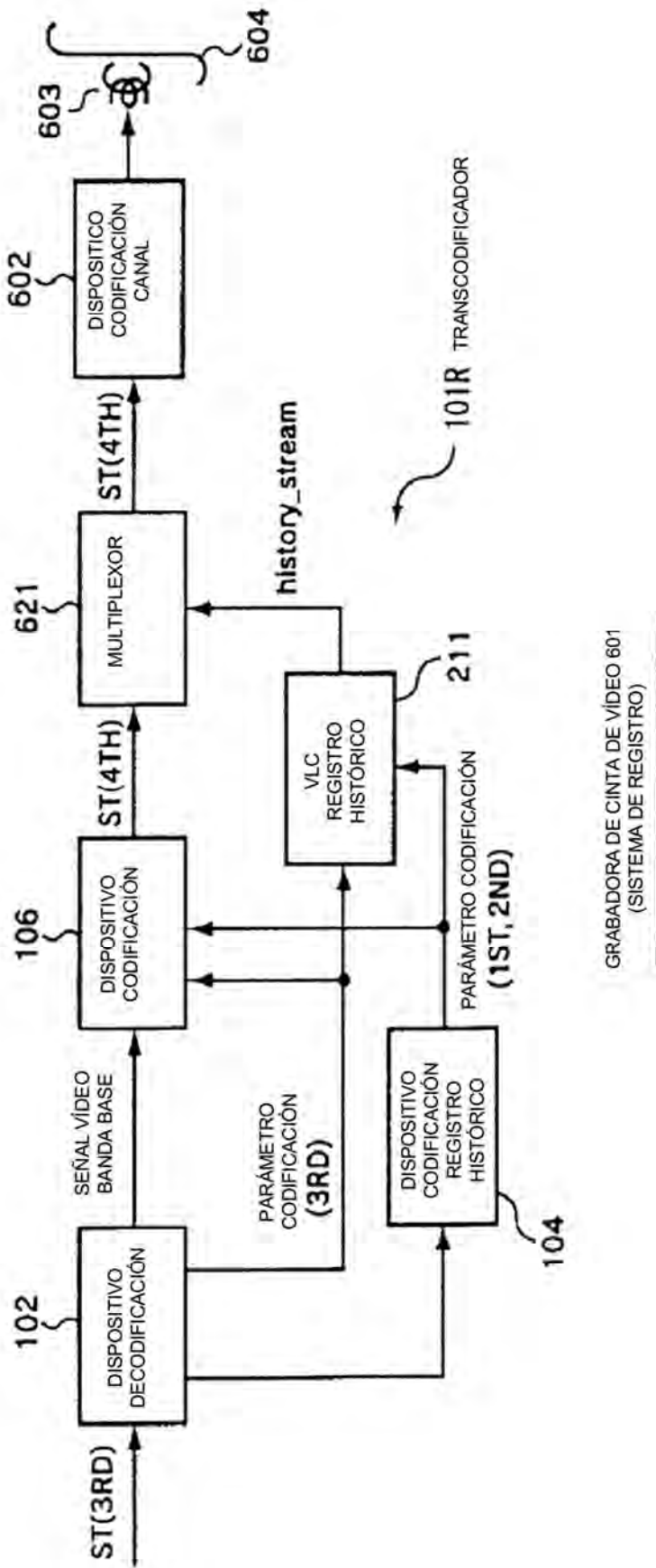
GRABADORA DE CINTA DE VIDEO 601  
(SISTEMA DE REGISTRO)

FIG. 76



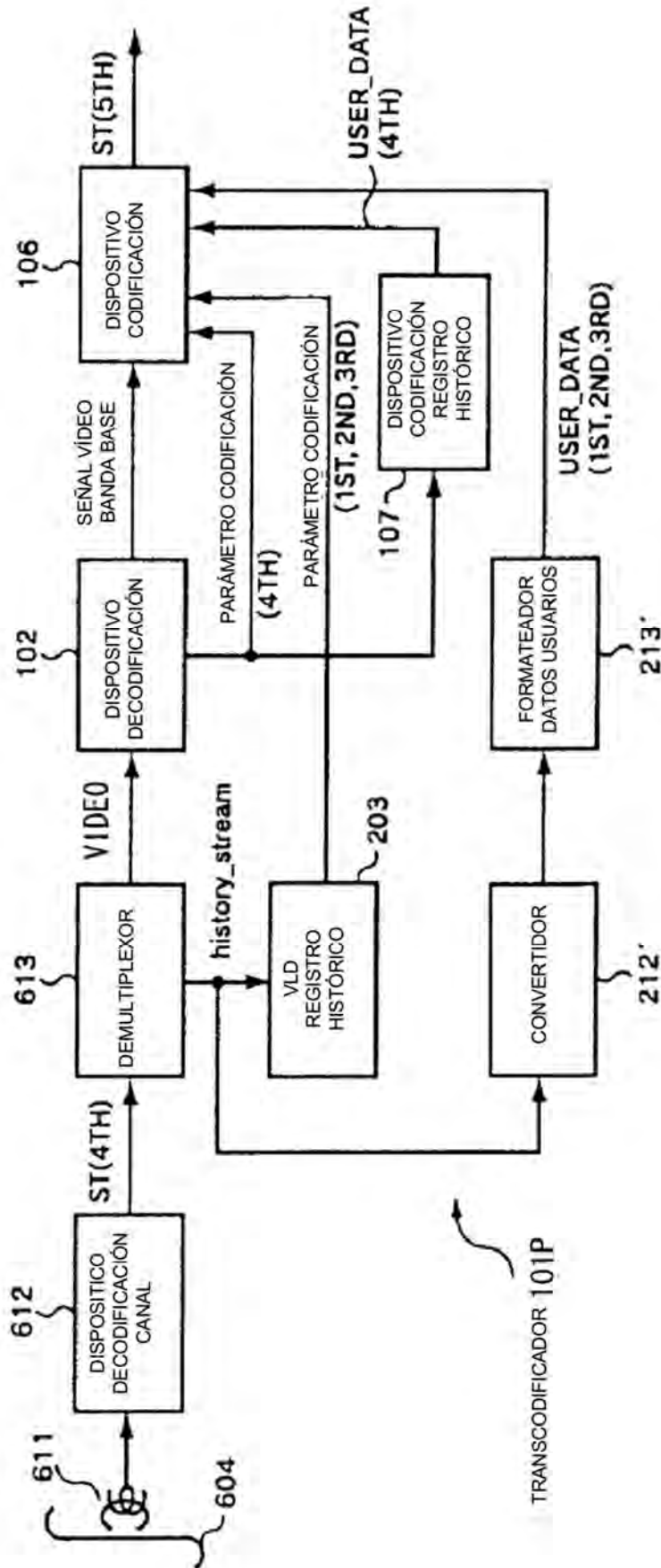
GRABADORA DE CINTA DE VÍDEO 601  
(SISTEMA DE REPRODUCCIÓN)

FIG. 77



GRABADORA DE CINTA DE VIDEO 601  
(SISTEMA DE REGISTRO)

FIG. 78



GRABADORA DE CINTA DE VIDEO 601  
(SISTEMA DE REPRODUCCION)

FIG. 79

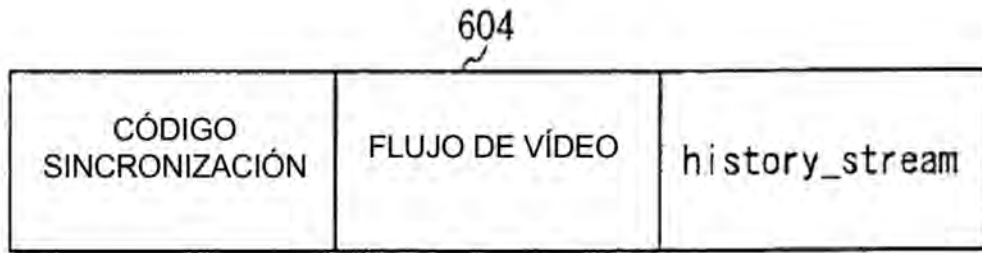


FIG. 80

**Explicación de las referencias numéricas**

1, 106 ... Dispositivo codificación, 2, 102 ... Dispositivo decodificación, 3 ... Soporte de registro, 14, 33 . . . Memoria tramas, 17, 32 . . . Circuito conversión formato, 18 . . . Codificador, 19 . . . Circuito de registro, 30 . . . Circuito de reproducción, 31 ... Decodificador, 50 ... Circuito detección vectores movimiento, 57 ... Circuito de cuantización, 58 . . . Circuito codificación longitud variable, 59 . . . Memoria transmisión, 64, 87 ... Circuito compensación de movimiento, 70 ... Controlador, 81 ... Memoria recepción, 82 ... Circuito decodificación longitud variable, 83 ... Circuito de cuantización, 84 ... Circuito IDCT, 100 ... Ordenador, 101 . . . Transcodificador, 103 . . . Dispositivo multiplexación información registro histórico, 104 ... Dispositivo decodificación registro histórico, 105 ... Dispositivo separación-información registro histórico, 107 ... Dispositivo codificación registro histórico, 201 ... Decodificador datos usuarios, 202, 212 ... Convertidor, 203 ... VLC registro histórico.