

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 674 244**

51 Int. Cl.:

**H04N 7/24** (2011.01)

**H04N 21/845** (2011.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **06.09.2010** E 16179330 (2)

97 Fecha y número de publicación de la concesión europea: **28.03.2018** EP 3104601

54 Título: **Concepto de transmisión para un flujo que comprende unidades de acceso**

30 Prioridad:

**09.09.2009 US 240833 P**

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

**28.06.2018**

73 Titular/es:

**FRAUNHOFER-GESELLSCHAFT ZUR  
FÖRDERUNG DER ANGEWANDTEN  
FORSCHUNG E.V. (100.0%)  
Hansastraße 27c  
80686 München, DE**

72 Inventor/es:

**ZINK, ALEXANDER;  
KELLERMANN, CHRISTIAN;  
MOHAMMED, HUSSAIN;  
FAERBER, NIKOLAUS y  
THOMA, HERBERT**

74 Agente/Representante:

**ARIZTI ACHA, Monica**

**ES 2 674 244 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).

## Concepto de transmisión para un flujo que comprende unidades de acceso

**DESCRIPCIÓN**

5 La presente invención se refiere a transmisión de, - o preparación, para transmisión, de - un flujo de unidad de acceso tal como un flujo de medios, tal como unidades de acceso de vídeo, audio, texto u otros datos. En algunas realizaciones, la transmisión implica la transmisión de unidades de acceso de vídeo mediante una señal de difusión de audio, tal como DRM.

10 Existen muchas capacidades de transmisión especializadas de manera especial para la transmisión de tipos específicos de datos tales como datos de vídeo y de audio, respectivamente. Por ejemplo, DVB-T se ha diseñado para difundir datos de vídeo. DRM, a su vez, se diseñó para alcanzar, mediante contenido de audio, audiencias dispersadas a través de un área geográfica amplia para mantener a los ciudadanos que viven en el extranjero informados y actualizados acerca de lo que está ocurriendo en el país local, por ejemplo. Cualquiera que sea la capacidad de transmisión considerada, los parámetros de diseño de estas capacidades de transmisión, tales como ancho de banda máximo para los datos útiles, el número de canales o programas que se soportan y el rendimiento en términos de ajuste de la distribución del ancho de banda entre los canales o programas soportados, está configurado para adaptar las demandas impuestas por la forma específica de datos para los que se ha diseñado la respectiva capacidad de transmisión. Por ejemplo, el contenido de audio necesita menos ancho de banda que el contenido de vídeo. Además, las tolerancias de error son diferentes. Algunas de las capacidades de transmisión están especializadas para transmitirse mediante capas físicas que, en sí mismas, tienen una capacidad de ancho de banda limitada. DRM, por ejemplo, se diseñó originalmente para transmitirse en las bandas con las que se permiten distancias de difusión más grandes, sin embargo, aceptando las capacidades de ancho de banda limitadas resultantes de las mismas.

25 El documento D1 (ISO/IEC 13818-1) desvela las especificaciones de MPEG-2 TS. El documento D2 (US2004/083417) desvela separar el contenido en dos partes: datos importantes y datos menos importantes, y usar diferentes códigos de corrección de errores para cada una de las dos partes. El documento D3 (ETSI TS 102 821) desvela las especificaciones de la Radio Digital Mundial. El documento D4 (US5481543) desvela multiplexación variando la tasa de bits y añadiendo datos adicionales.

35 Basándose en las capacidades de transmisión que están disponibles hasta ahora, existe una necesidad de un concepto de transmisión o de un concepto de preparación de transmisión que posibilite la transmisión de un flujo de unidad de acceso mediante alguna capacidad de transmisión existente que, por ejemplo, no se haya diseñado para transmitir tal flujo de unidad de acceso en términos de ancho de banda, por ejemplo. Considérese, por ejemplo, intentar proporcionar un servicio de transmisión de vídeo en una capacidad de transmisión tal como en la estructura de DRM. DRM transmite los datos útiles a ancho de banda relativamente bajo cuando se considera las necesidades de contenido de vídeo y usando ráfagas a un paso de ráfaga considerable. En una situación de este tipo, resultaría un retardo de tiempo considerable cuando se cambia de un programa al contenido de vídeo recién mencionado, durante el cual, en el lado de decodificación, el decodificador incluso no tendría la capacidad de sincronizar al contenido de vídeo.

45 Por consiguiente, es un objeto de la presente invención proporcionar un concepto de transmisión o un concepto de preparación de transmisión que posibilite una transmisión de ancho de banda bajo y sincronización rápida de un flujo de unidad de acceso mediante una capacidad o capa de transmisión subyacente.

50 Este objeto se consigue mediante un aparato para preparar un flujo de unidad de acceso de acuerdo con la reivindicación 1, un aparato para recuperar un flujo de unidad de acceso de acuerdo con la reivindicación 26, un método para preparar un flujo de unidad de acceso de acuerdo con la reivindicación 45, un método para recuperar una unidad de acceso de acuerdo con la reivindicación 46, una señal de transmisión de acuerdo con la reivindicación 47 y un programa informático de acuerdo con la reivindicación 48.

55 La presente invención proporciona un aparato para preparar un flujo de unidad de acceso de unidades de acceso consecutivas, que representan contenido de medios, para una transmisión mediante una señal de transmisión, el aparato configurado para generar una secuencia de tramas lógicas desde el flujo de unidad de acceso insertando consecutivamente las unidades de acceso consecutivas en una sección de datos útiles de las tramas lógicas de la secuencia de tramas lógicas, y proporcionando cada trama lógica en la que cae un comienzo de una unidad de acceso, comprendiendo una tabla de unidad de acceso, por comienzo de las unidades de acceso que caen en la respectiva trama lógica, un puntero que apunta a la misma.

60 A medida que cada trama lógica en la que cae un comienzo de una unidad de acceso, se proporciona con una tabla de unidad de acceso que comprende, por comienzo de las unidades de acceso que caen en la respectiva trama lógica, un puntero que apunta a la misma, un decodificador que recibe la señal de transmisión y la secuencia de tramas lógicas, respectivamente, puede localizar y acceder a las unidades de acceso en las tramas lógicas tan

pronto como sea posible usando los punteros.

Por consiguiente, de acuerdo con una realización de la presente invención, se proporciona un aparato para recuperar un flujo de unidad de acceso de unidades de acceso consecutivas, que representan contenidos de medios, desde una secuencia de tramas lógicas de una señal de transmisión, comprendiendo cada trama lógica una sección de datos útiles. Las unidades de acceso consecutivas se insertan de manera consecutiva en la sección de datos útiles de la secuencia de tramas lógicas y el aparato de recuperación está configurado para, para una trama lógica predeterminada en la que cae un comienzo de una unidad de acceso, extraer una tabla de unidad de acceso desde la trama lógica predeterminada que comprende, por comienzo de las unidades de acceso que caen en la trama lógica predeterminada, un puntero que apunta a una respectiva posición de inicio en la trama lógica predeterminada, y localizar, y comenzar la extracción de, la respectiva unidad de acceso el comienzo de la cual cae en la trama lógica predeterminada, por uso del respectivo puntero, estando configurado también el aparato para extraer de manera consecutiva las unidades de acceso consecutivas del flujo de unidad de acceso desde la sección de datos útiles de las tramas lógicas de la secuencia de tramas lógicas.

Implementaciones ventajosas son el objeto de las reivindicaciones dependientes adjuntas.

En particular, se describen a continuación realizaciones preferidas de la presente solicitud con respecto a las figuras, entre las cuales:

La Figura 1 muestra un diagrama esquemático que ilustra la inserción consecutiva de un flujo de unidad de acceso en una secuencia de tramas lógicas y la estructura de las tramas lógicas de acuerdo con una realización de la presente invención;

La Figura 2 muestra un diagrama de bloques de una cadena de transmisión de acuerdo con una realización;

La Figura 3 muestra un diagrama de bloques de una cadena de recepción de acuerdo con una realización;

La Figura 4 muestra un diagrama de flujo de una preparación de flujo de unidad de acceso realizada mediante el preparador de flujo de unidad de acceso en la Figura 2 de acuerdo con una realización;

La Figura 5 muestra un diagrama esquemático que ilustra la estructura de una trama lógica de acuerdo con una realización;

La Figura 6 muestra un diagrama esquemático que visualiza el enfoque de intercalación virtual de acuerdo con una realización;

Las Figuras 7a a 7m muestran diagramas de flujo que ilustran el modo de operación del recuperador de flujo de unidad de acceso en la Figura 3 de acuerdo con una realización;

La Figura 8 muestra un diagrama de flujo de otra realización para un modo de operación del recuperador de flujo de unidad de acceso en la Figura 3 en relación con FEC; y

La Figura 9 muestra un diagrama de flujo que ilustra un modo de operación en el recuperador de flujo de unidad de acceso en la Figura 3 después de haber realizado FEC o sin FEC.

En primer lugar, con respecto a las Figuras 1 y 2, se describe una realización para la preparación de un flujo de unidad de acceso de unidades de acceso consecutivas, que representa contenido de medios, para una transmisión mediante una señal de transmisión. La Figura 2 muestra un transmisor o cadena de transmisión 10 que comprende un generador de flujo de unidad de acceso 12 configurado para generar un flujo de unidad de acceso 14 de unidades de acceso consecutivas, representando el flujo de unidad de acceso 14 contenido de medios, tal como contenido de vídeo o contenido de vídeo y de audio o contenido de vídeo junto con contenido textual o de datos alineados en tiempo tal como noticias o similares.

El generador de flujo de unidad de acceso 12 puede comprender un codificador de vídeo, un codificador de audio y/o un generador de contenido de texto o similar, o incluso puede aceptar AU desde alguna fuente o fuentes externas. Por consiguiente, el flujo de unidad de acceso 14 puede de hecho estar compuesto de uno o más subflujos separados de unidades de acceso estando dispuestas las unidades de acceso de los diferentes subflujos en el flujo de unidad de acceso 14 en una forma intercalada de modo que las unidades de acceso de los subflujos individuales que pertenecen al mismo tiempo de presentación, se agrupan juntas para estar tan cerca como sea posible o, en otras palabras, se sigan inmediatamente - o dentro de algún límite de tiempo máximo predeterminado - en el flujo de

unidad de acceso 14.

Además, la cadena de transmisión 10 de la Figura 2 comprende un preparador de flujo de unidad de acceso 16 configurado para preparar el flujo de unidad de acceso 14 de unidades de acceso consecutivas para una transmisión mediante una señal de transmisión. Para este fin, el preparador de flujo de unidad de acceso 16 está configurado para generar una secuencia 18 de tramas lógicas desde el flujo de unidad de acceso 14 insertando de manera consecutiva las unidades de acceso consecutivas en una sección de datos útiles de las tramas lógicas de la secuencia de las tramas lógicas y proporcionando cada trama lógica en la que cae un comienzo de una unidad de acceso, con una tabla de unidad de acceso que comprende, por comienzo de las unidades de acceso que caen en la respectiva trama lógica, un puntero que apunta a la misma. La Figura 1, por ejemplo, muestra una porción ejemplar de un flujo de unidad de acceso 14 que incluye, de manera ejemplar, cuatro unidades de acceso AU<sub>1</sub> a AU<sub>4</sub> y la correspondiente porción de la secuencia 18 de las tramas lógicas que abarca, de manera ejemplar, las tramas lógicas LF<sub>1</sub>, LF<sub>2</sub> y LF<sub>3</sub>. Como se muestra en la Figura 1, el preparador de flujo de unidad de acceso 16 puede estar configurado de manera que cada trama lógica 20 comprende un encabezamiento de trama lógica 22 y una sección de datos útiles 24. Como se describirá en más detalle a continuación, las tramas lógicas 20 no es necesario que sean de longitud constante aunque las tramas lógicas LF<sub>1</sub> a LF<sub>3</sub> representadas en la Figura 1 se ilustren de esta manera.

Las líneas discontinuas 26 en la Figura 1 ilustran la inserción consecutiva de las unidades de acceso 28 en la sección de datos útiles 24 de la secuencia 18 de tramas lógicas 20. Como es visible a partir de la Figura 1, el preparador de flujo de unidad de acceso 16 puede estar configurado para proporcionar simplemente aquellas tramas lógicas 20 con una tabla de unidad de acceso 30 en la que realmente cae un comienzo 32 de cualquiera de las unidades de acceso 28. Entre las tramas lógicas LF<sub>1</sub> a LF<sub>3</sub>, las tramas lógicas 20 son tales tramas lógicas LF<sub>1</sub> y LF<sub>3</sub> mientras que la trama lógica LF<sub>2</sub> no comprende un comienzo de una unidad de acceso 24 y, por consiguiente, ninguna tabla de unidad de acceso.

Además, como se muestra en la Figura 1, el preparador de flujo de unidad de acceso 16 puede estar configurado adicionalmente de manera que los encabezamientos de la trama lógica opcionales 22 estén registrados con su extremo inicial en el extremo inicial de la respectiva trama lógica a la que pertenece el encabezamiento de trama lógica 22. Como se ilustra en la Figura 1, los encabezamientos de tramas lógicas 22 pueden ser de tamaño constante, es decir, el tamaño puede ser igual entre sí entre las tramas lógicas 20. Sin embargo, en lo que se refiere a la tabla de unidad de acceso 30, el preparador de flujo de unidad de acceso 16 puede estar configurado para registrar las tablas de unidad de acceso 30 con su extremo final al extremo final de la respectiva trama lógica a la que pertenece la respectiva tabla de unidad de acceso 30, como se ilustra en la Figura 1, o, como alternativa, al extremo inicial de la respectiva trama lógica a la que pertenece la respectiva tabla de unidad de acceso 30 (es decir inserción hacia adelante o hacia atrás). Las tablas de unidad de acceso 30 pueden tener un tamaño o longitud variable 34 dependiendo del número de unidades de acceso 20 el comienzo 32 de las cuales cae en la respectiva trama lógica 20. Comparando las tramas lógicas LF<sub>1</sub> y LF<sub>3</sub>, por ejemplo, dos unidades de acceso AU<sub>3</sub> y AU<sub>4</sub> tienen su respectivo comienzo 32 dispuesto en la sección de datos útiles 24 de trama lógica LF<sub>3</sub>, mientras que solamente una unidad de acceso, en concreto la unidad de acceso AU<sub>2</sub> tiene su comienzo 32 dispuesto en la sección de datos útiles de trama lógica LF<sub>1</sub> de modo que la tabla de unidad de acceso 34 de la trama lógica LF<sub>1</sub> tiene una longitud 34 menor que la longitud 34 de la tabla de unidad de acceso 30 de la trama lógica LF<sub>3</sub>.

En caso de que el preparador de flujo de unidad de acceso 16 esté configurado para disponer el encabezamiento de trama lógica 22 y la tabla de unidad de acceso 30 a unos diferentes del extremo inicial y extremo final de las tramas lógicas 20 como se acaba de describir, la sección de datos útiles 24 está dispuesta entre el encabezamiento de trama lógica 22 y la tabla de unidad de acceso 30 para las tramas lógicas 20 en las que cae un comienzo 32 de una unidad de acceso 28, y el encabezamiento de trama lógica 22 y el extremo opuesto de la respectiva trama lógica para tramas lógicas en las que no cae el comienzo 32 de una unidad de acceso 28.

La cadena de transmisión 10 de la Figura 2 comprende adicionalmente, de manera opcional, una etapa de transmisión 36 para transmitir una señal de transmisión 38 que incluye, o que tiene embebida en la misma, la secuencia 18 de tramas lógicas 20. Por ejemplo, la etapa de transmisión 36 puede difundir la señal de transmisión 38. La etapa de transmisión 36 puede representar una capa de transporte que, de acuerdo con el modelo OSI, está debajo de la capa de transporte a la que pertenece el preparador de flujo de unidad de acceso 16. Por ejemplo, la secuencia de tramas lógicas puede embeberse en un flujo de MSC que, a su vez, se transmite mediante la etapa de transmisión 36 en forma de una secuencia de tramas de transmisión, la última de las cuales se transmiten mediante respectivos símbolos de modulación. La etapa de transmisión 36 puede transmitir, por ejemplo, la señal de transmisión 38 en ráfagas y, por ejemplo, por medio de una señal de OFDM o similar. El tamaño de las tramas lógicas 20 puede ser constante durante el tiempo o puede ser variable, caso en el que el tamaño de las respectivas tramas lógicas 20 puede indicarse en un canal de información secundaria de la señal de transmisión 38. Además, el canal de información secundaria de la señal de transmisión 38 puede comprender información tal como tiempos de datos que indican al receptor cuándo tiene lugar la siguiente ráfaga que incluye la siguiente trama lógica en la señal de transmisión 38 para posibilitar ahorros de potencia eficaces en el lado de receptor, y/o indicaciones en cuanto a

dónde está situado el inicio y fin de las tramas lógicas en la señal de transmisión 38.

Por lo tanto, en operación, el generador de flujo de unidad de acceso 12 genera las unidades de acceso 28, y el preparador de flujo de unidad de acceso 16 inserta de manera consecutiva la unidad de acceso consecutiva 28 en la sección de datos útiles 24 de las tramas lógicas 20 proporcionando cada trama lógica 20 en la que cae el comienzo 32 de una unidad de acceso 28, con una tabla de unidad de acceso 30. Cada tabla de unidad de acceso 30 comprende, por comienzo 32 de las unidades de acceso 28 que caen en la respectiva trama lógica 20, un puntero 40 que apunta al respectivo comienzo 32. Debido a la existencia de los punteros 40, un receptor que recibe las tramas lógicas 20 en la señal de transmisión 38 puede localizar y acceder inmediatamente a la primera unidad de acceso tan pronto como el decodificador recibe la primera trama lógica en el tiempo, en el que cae un comienzo 32 de las unidades de acceso 28. Para este fin, el receptor puede usar las indicaciones adicionales anteriormente mencionadas en, por ejemplo, el canal de información secundaria de la señal de transmisión para tener conocimiento con antelación acerca de las posiciones de inicio y fin de las tramas lógicas 20, o los límites de las tramas lógicas pueden determinarse implícitamente mediante la estructura global de la señal de transmisión. Por lo tanto, incluso cuando el ancho de banda usado por la etapa de transmisión 36 es pequeño, el retardo del decodificador al sincronizar el flujo de unidad de acceso transportado mediante la señal de transmisión 38 no se aumenta adicionalmente por las necesidades de sincronización adicionales que serían necesarias de otra manera para que el decodificador localizara las unidades de acceso.

La Figura 3 muestra una cadena de recepción o receptor 50 adecuado para recibir la señal de transmisión 38 que comprende la secuencia 18 de tramas lógicas 20 o que tiene la última embebida en la misma, respectivamente. La cadena de recepción 50 comprende, opcionalmente, una etapa de recepción 52 como una parte opuesta a la etapa de transmisión 36. En otras palabras, la etapa de recepción 52 puede pertenecer a la misma capa de transporte a la que pertenece la etapa de transmisión 36. La etapa de recepción 52 puede comprender una antena, amplificadores, un demodulador, un corrector de errores hacia delante tal como incluyendo, por ejemplo, un turbo decodificador, y/o un desintercalador así como alguna unidad de gestión para localizar las tramas lógicas en la señal de transmisión 38 basándose, por ejemplo, en información secundaria transmitida en la señal de transmisión 38 en un cierto canal o similar, como ya se ha señalado anteriormente.

La etapa de recepción 52 reenvía la secuencia 18 de tramas lógicas 20 al recuperador de flujo de unidad de acceso 54 también comprendido mediante el decodificador 50. El recuperador de flujo de unidad de acceso 54 está configurado para recuperar el flujo de unidad de acceso 14 de unidades de acceso consecutivas 28 desde la secuencia 18 de tramas lógicas 20. En particular, el recuperador 54 puede estar configurado para extraer desde una trama lógica predeterminada 20 tal como la primera recibida mediante la señal de transmisión 38, que tiene un comienzo 32 de una unidad de acceso 28 dispuesto en la misma, la tabla de unidad de acceso 30 y localizar, y comenzar la extracción de, la respectiva unidad de acceso 28 el comienzo 32 de la cual cae en la respectiva trama lógica 20, mediante el uso del respectivo puntero 40 comprendido mediante la tabla de unidad de acceso extraída 30. Además de esto, el recuperador 54 está configurado para extraer de manera consecutiva las unidades de acceso consecutivas 28 del flujo de unidad de acceso 14 desde la sección de datos útiles 24 de las tramas lógicas 20 de la secuencia 18 de tramas lógicas 20 recibidas desde etapa de recepción 52. Además, el decodificador 50 puede comprender un presentador 56 para decodificar y/o presentar el contenido de medios transportado mediante la secuencia 14 de unidades de acceso 28 como se recuperó mediante el recuperador 54 desde las tramas lógicas 20. El presentador 56 puede comprender, por ejemplo, un decodificador de vídeo, un decodificador de audio y/o un manejador de texto o de datos. Además, el presentador 56 puede comprender una pantalla de vídeo y/o un altavoz.

Los detalles específicos que se han descrito anteriormente con respecto a las Figuras 1 a 3, son ventajosos, pero opcionales. A continuación, se describen ventajas de detalles específicos y alternativas. Por ejemplo, como se ha descrito anteriormente, el preparador de flujo de unidad de acceso 16 puede estar configurado para generar la secuencia 18 de tramas lógicas 20 de manera que la tabla de unidad de acceso 30 rodea el extremo final de la respectiva trama lógica 20. En este sentido, se observa que el extremo final de una trama lógica 20 se entiende que es el fin de la trama lógica 20 que llega más tarde en el tiempo en la señal de transmisión 38 en el decodificador 50 con la dirección del tiempo en la Figura 1 apuntando al lado a la derecha, por ejemplo. Sin embargo, como alternativa, la tabla de unidad de acceso 30 podría rodear el extremo inicial de la respectiva trama lógica 20. Incluso como alternativa, la tabla de unidad de acceso 30 puede tener un desplazamiento constante predeterminado desde el extremo final o inicial de la respectiva trama lógica 20. En todos estos casos, el recuperador 54 puede localizar la tabla de unidad de acceso 30 de la trama lógica predeterminada 20, la tabla de unidad de acceso 30 de la cual se ha de evaluar, en, o al desplazamiento constante predeterminado desde, el extremo final o inicial de la respectiva trama lógica 20.

Además, como se ha descrito también anteriormente, el preparador de flujo de unidad de acceso 16 puede estar configurado para generar la secuencia 18 de tramas lógicas 20 de manera que los punteros 40 apuntan al comienzo 32 de las unidades de acceso 28 que caen en la respectiva trama lógica 20 desde un punto de registro situado con respecto al extremo final o inicial de la respectiva trama lógica 20 de una manera constante entre las tramas lógicas 20 en las que cae un comienzo 32 de una unidad de acceso 28. En las realizaciones específicas descritas a

continuación, por ejemplo, los punteros 40 se indican en las respectivas tablas de unidad de acceso 30 en unidades de bytes o bits o alguna otra longitud de unidad, medida desde el extremo inicial de la respectiva trama lógica.

5 Sin embargo, como alternativa, otros puntos en las tramas lógicas distintos al extremo inicial de las mismas pueden servir como el punto de registro que se acaba de mencionar desde el que los punteros 40 apuntan al comienzo 32 de las unidades de acceso 28. Por consiguiente, el recuperador 54 puede estar configurado para usar, al localizar una respectiva unidad de acceso el comienzo 32 de la cual cae en una trama lógica actualmente inspeccionada 20, el respectivo puntero 40 como un desplazamiento desde el punto de registro.

10 Además, aunque no se establece explícitamente con respecto a la Figura 1, la Figura 1 ilustra un caso de acuerdo con el que el preparador de flujo de unidad de acceso 16 está configurado para insertar de manera ininterrumpida las unidades de acceso consecutivas 28 en la sección de datos útiles 24 de las tramas lógicas 20, al menos, siempre que sea posible. El hueco 58 entre la unidad de acceso  $AU_2$  y  $AU_3$  en la Figura 1, por ejemplo, es simplemente el resultado de las tablas de unidad de acceso 30 de la realización de la Figura 1 que tiene una longitud 34 que aumenta con el número creciente de comienzos 32 que caen en la respectiva trama lógica de modo que, en consecuencia, la sección de datos útiles disminuye por comienzo adicional 32 que tiene lugar. Sin embargo, además de tales huecos 58 la unidad de acceso 28 de la Figura 1 se ha insertado de manera ininterrumpida en la sección de datos útiles 24 de las tramas lógicas 20.

20 Como alternativa, sin embargo, la unidad de acceso 28 puede insertarse en la sección de datos útiles 24 de las tramas lógicas 20 sin importar los datos de relleno dispuestos entre ellas. Por ejemplo, dependiendo de la aplicación, las unidades de acceso 28 pueden haberse generado mediante el generador de flujo de unidad de acceso 12 independientemente de una cierta tasa de bits por la cual las mismas se han de transportar mediante la señal de transmisión 38, y para obedecer exactamente tal tasa de transmisión, pueden introducirse tales datos de relleno entre unas ciertas de las unidades de acceso 28. Por lo tanto, los datos de relleno pueden estar integrados en la secuencia de tramas lógicas por medio de ajustar en consecuencia la indicación de longitud opcional introducida a continuación y los punteros 40 en el lado de transmisión. Como alternativa, sin embargo, un "ID de flujo" no usado o específico indicado, por ejemplo, en la tabla de AU para una respectiva AU puede indicar que esta AU simplemente contiene "datos de relleno", es decir es una "AU de relleno", caso en el que el flujo de AU resultante, a su vez, mantendría las propiedades señaladas anteriormente de un flujo de AU insertado de manera ininterrumpida posibilitando, sin embargo, relleno en el lado de transmisión. En el lado de recepción, esta AU de relleno se saltaría o desprendería, y simplemente se procesarían adicionalmente las otras AU.

30 Por consiguiente, el recuperador de flujo de unidad de acceso 54 puede extraer de manera ininterrumpida o simplemente de manera consecutiva las unidades de acceso consecutivas 28 desde las secciones de datos útiles 24 de las tramas lógicas 20. Para extraer las unidades de acceso consecutivas 28, el recuperador de flujo de unidades de acceso 54 puede localizar los comienzos 32 de la siguientes unidades de acceso 28 mediante el uso de los punteros anteriormente mencionados 40 o, como alternativa, detectando el final de una respectiva unidad de acceso 28 analizando las unidades de acceso, representando este fin de manera concurrente el comienzo 32 de la siguiente unidad de acceso 28, excepto para la presencia de una situación similar a hueco 58 en la Figura 1 que, sin embargo, es previsible mediante el recuperador de flujo de unidad de acceso 54.

45 Además, el preparador de flujo de unidad de acceso 16 puede estar configurado para indicar en los encabezamientos de tramas lógicas 22 la ausencia de una tabla de unidad de acceso 30 en la respectiva trama lógica 20 para las tramas lógicas 20 en las que no cae el comienzo 32 de ninguna de la unidad de acceso consecutiva 28, y la longitud 34 de la tabla de unidad de acceso 30 de la respectiva trama lógica para las tramas lógicas 20 en las que cae el comienzo 32 de al menos una de las unidades de acceso consecutivas 28. Mediante esta medida, el recuperador de flujo de unidad de acceso 54 puede extraer desde el encabezamiento de trama lógica 22 de cada trama lógica 20 una información que indica la ausencia de o la longitud de una tabla de unidad de acceso 30 en la respectiva trama lógica 20, y localizar una extensión de la sección de datos útiles 24 de la respectiva trama lógica 20 dependiendo de la misma. En particular, el recuperador de flujo de unidad de acceso 54 puede localizar la extensión de la sección de datos útiles 24 incluso en caso de que la respectiva tabla de unidad de acceso 30 de la trama lógica actual esté corrupta debido a corrupción de datos, y por lo tanto, el recuperador de flujo de unidad de acceso 54 pudiera continuar correctamente con la extracción de una unidad de acceso 28 que se extiende a través del límite entre esta tabla de unidad de acceso 30 y la sección de datos útiles 24 hasta la sección de datos útiles 24 de la siguiente trama lógica 20, incluso a través del límite de la trama lógica entre ellas.

60 De manera similar, el preparador de flujo de unidad de acceso 16 podría configurarse de manera que cada encabezamiento de trama lógica 22 rodee, o tenga un desplazamiento constante predeterminado desde, el extremo final o inicial de la respectiva trama lógica 20. En la realización de la Figura 1, todos los encabezamientos de tramas lógicas 22 rodean el extremo inicial de la respectiva trama lógica 20. Por consiguiente, el recuperador de flujo de unidad de acceso 54 puede estar configurado para localizar, en cada trama lógica, el encabezamiento de trama lógica 22 en, o a un desplazamiento constante predeterminado desde, el extremo inicial o final de la respectiva trama lógica 20, es decir, independientemente de la corrupción de datos en la parte restante de las tramas lógicas 20, tal como ciertas porciones de la sección de datos útiles 24.

En el caso de la Figura 1, por ejemplo, el preparador de flujo de unidad de acceso 16 se configuró para realizar la inserción continua de las unidades de acceso consecutivas 28 en la sección de datos útiles 24 de las tramas lógicas 20 usando una dirección de inserción de datos útiles 60 y para disponer la tabla de unidad de acceso 30 en las tramas lógicas 20 en las que cae un comienzo 32 de una unidad de acceso 28, para ocupar una porción conectada de la respectiva trama lógica 20 que tiene un extremo situado de manera constante que apunta en la dirección de inserción de datos útiles 60, es decir, un extremo final situado de manera constante, y un extremo variable que apunta al contrario de la dirección de inserción de datos útiles 60, es decir un extremo inicial situado de manera variable, que está desplazado desde el extremo situado de manera constante por la longitud 34 de la tabla de unidad de acceso 30. En otras palabras, la indicación de la longitud 34 en los encabezamientos de tramas lógicas 22 puede medir la longitud o tamaño de las tablas de unidad de acceso 30 medidas desde el extremo situado de manera constante de las tablas de unidad de acceso 31, en concreto el extremo final de las mismas. Por consiguiente, el recuperador de flujo de unidad de acceso 54 puede estar configurado para realizar la extracción consecutiva de la unidad de acceso consecutiva 28 desde las tramas lógicas 20 usando una dirección de extracción de datos útiles igual a la dirección de inserción de datos útiles en las tramas lógicas 20, y para localizar el extremo situado de manera variable de la tabla de unidad de acceso 30 aplicando la longitud 34 de la tabla de unidad de acceso 30 en una dirección en contra contraria a la dirección de extracción de datos útiles 60 desde un extremo situado de manera constante de la misma.

Puede ser ventajoso si el preparador de flujo de unidad de acceso 16 está configurado para generar la secuencia de las tramas lógicas 20 de manera que las tablas de unidad de acceso 30 y los encabezamientos de tramas lógicas 22 rodean, o están desplazados de manera constante de, unos opuestos de los extremos inicial y final de las tramas lógicas 20 y si, por consiguiente, el recuperador de flujo de unidad de acceso 54 está configurado para localizar la tabla de unidad de acceso 30 y el encabezamiento de trama lógica 22 de las respectivas tramas lógicas 20 en, o a un desplazamiento constante a, unos diferentes de los extremos inicial y final de las respectivas tramas lógicas 20 como es el caso en la Figura 1. Esto se cumple particularmente, si los encabezamientos de tramas lógicas 22 son de longitud variable, dependiendo la longitud del contenido del mismo encabezamiento de LF. En este caso, el recuperador de flujo de unidad de acceso 54 puede localizar la tabla de unidad de acceso 30 y el encabezamiento de trama lógica 22 independientemente de la corrupción de datos en la sección de datos útiles 24 de la respectiva trama lógica, y puede localizar la tabla de unidad de acceso 30 independientemente de la corrupción de datos del encabezamiento de trama lógica 22 y viceversa.

Para ser incluso más precisos, el preparador de flujo de unidad de acceso 16 puede estar configurado para generar la secuencia 18 de tramas lógicas 20 de manera que la tabla de unidad de acceso 30 y el encabezamiento de trama lógica 22 rodeen los opuestos de los extremos inicial y final de las tramas lógicas 20 de modo que la sección de datos útiles 24 es una porción conectada que se extiende, durante tramas lógicas 20 en las que cae un comienzo 32 de una unidad de acceso 28, entre la tabla de unidad de acceso 30 y el encabezamiento de trama lógica 22, respectivamente, y durante otras tramas lógicas 20 entre el encabezamiento de trama lógica 22 y el opuesto de los extremos inicial y final de las tramas lógicas 20.

Como ya se ha ilustrado en la Figura 1, el preparador de flujo de unidad de acceso 16 puede estar configurado para proporcionar en cada trama lógica 20 en la que cae un comienzo 32 de una unidad de acceso 28, que se extiende más allá de un extremo final de la respectiva trama lógica 20 en una siguiente trama lógica 20, tal como la unidad de acceso AU<sub>2</sub> en la Figura 1, la tabla de unidad de acceso 30 con una indicación de longitud que indica una longitud 62 de la respectiva unidad de acceso 28, es decir una indicación que posibilita determinar el fin de la respectiva trama lógica 20 cuando se combina con el puntero 40 que apunta al comienzo de la misma. En la Figura 1, el preparador de flujo de unidad de acceso 16 está configurado de manera ejemplar para acompañar cada puntero 40 con una indicación de longitud 62 de este tipo de la unidad de acceso 28 el comienzo 32 de la cual se apunta mediante el puntero 40. En caso de que las unidades de acceso 28, el extremo final de las cuales (en dirección de análisis) no sea determinable analizando las unidades de acceso 28, o se evite que se detecte debido a errores de datos locales, el recuperador de flujo de unidad de acceso 54 puede usar la indicación de la longitud 62 para separar entre el contenido de unidad de acceso y datos de relleno en las secciones de datos útiles 24. Sin embargo, si el fin de las unidades de acceso 28 es detectable mediante el recuperador de flujo de unidad de acceso 54 analizando las unidades de acceso, tal como detectando un respectivo fin de bandera de unidad de acceso en la misma unidad de acceso 28, el recuperador de flujo de unidad de acceso 54 puede detectar el fin de la respectiva unidad de acceso 28 incluso en caso de que la indicación de longitud 62 en la respectiva tabla de unidad de acceso 30 se haya corrompido. En cualquier caso, es ventajoso que el recuperador de flujo de unidad de acceso 54 pueda extraer una indicación de longitud 62 de este tipo desde las tablas de unidad de acceso 30 para obtener la longitud 62 de las unidades de acceso 28 el comienzo de las cuales cae en una respectiva trama lógica 20 que se extiende más allá de un extremo final de la respectiva trama lógica 20 en una siguiente trama lógica incluso cuando, por ejemplo, la tabla de unidad de acceso de la siguiente trama lógica se pierde o corrompe. En caso de una inserción de manera ininterrumpida de las unidades de acceso 28 en las porciones de trama lógica 24, el recuperador 54 puede incluso usar la indicación de longitud de la longitud 62 de una unidad de acceso 28 de este tipo que se extiende a través de tramas lógicas consecutivas 20 para localizar el comienzo 32 de una siguiente unidad de acceso. Por ejemplo, el

recuperador 54 puede usar la longitud 62 de la unidad de acceso  $AU_1$  para detectar el comienzo 32 de la unidad de acceso  $AU_2$  en caso, por ejemplo, que la unidad de acceso 30 de trama lógica  $LF_1$  esté corrupta hasta el punto que el puntero 40 que apunta al comienzo 32 de la unidad de acceso  $AU_2$  no esté disponible.

- 5 Debería observarse, sin embargo, que en lugar de una indicación de longitud, como alternativa, puede usarse una indicación de puntero final para apuntar al fin de la respectiva AU directamente, es decir independientemente del puntero 40 que apunta al comienzo 32. El efecto sería similar a las ventajas anteriormente señaladas. La indicación de puntero final puede comprender, por ejemplo, un puntero que apunta desde el punto de registro anteriormente mencionado, tal como el extremo inicial, de la trama lógica en la que cae el respectivo fin de la respectiva AU, hasta el fin de esta AU. Adicionalmente, la indicación de puntero final puede comprender un indicador de LF que indica en cuál de las siguientes LF cae el fin de las respectivas AU, tal como contando el número de LF empezando desde la LF siguiente a la LF actual. Por motivos de facilitar la descripción presentada a continuación, de acuerdo con las siguientes realizaciones se usa una indicación de longitud. Sin embargo, en estas realizaciones, la indicación de longitud se ha de entender simplemente representativa para una indicación que permite localizar el fin de la respectiva AU.

Como ya se ha mencionado anteriormente, el preparador de flujo de unidad de acceso puede estar configurado para proporcionar cada trama lógica 20 con un encabezamiento de trama lógica 22 que indica la longitud de la respectiva tabla de unidad de acceso 30. De acuerdo con las realizaciones expuestas a continuación en más detalle, el preparador 16 proporciona cada tabla de unidad de acceso 30 con una entrada de tabla de unidad de acceso por unidad de acceso 28 el comienzo 32 de la cual cae en la respectiva trama lógica 20, indicando el encabezamiento de trama lógica 22 la longitud de la tabla de unidad de acceso 30 medida, por ejemplo, en número de entradas de tabla de unidad de acceso que tienen longitud constante cada una en la respectiva tabla de unidad de acceso 30. Una realización específica de este tipo se describe con respecto a una realización de DRM más tarde con respecto a la Figura 5. Analizando brevemente la Figura 5, puede derivarse a partir de la Figura 5 que la trama lógica 20 muestra ejemplarmente en la misma que consiste en un número de entradas de tabla de unidad de acceso 64. Como puede derivarse también a partir de la Figura 5, la sección de datos útiles 24 puede no cubrir completamente la porción restante de la trama lógica 20 excepto la del encabezamiento de trama lógica 22 y la tabla de unidad de acceso 30. En su lugar, como se muestra en la Figura 5, la sección de datos útiles 24 puede desplazarse, en un número predeterminado de bytes o bits, desde el encabezamiento de trama lógica 22 y/o la tabla de unidad de acceso 30, siendo el número predeterminado, por ejemplo, conocido para el decodificador o transmitido al decodificador en un canal adicional en la señal de transmisión 38. En el ejemplo de la Figura 5, por ejemplo, los datos de FEC 66 - en este punto, de manera ejemplar datos de RS (Reed Solomon) - están, o pueden estar, situados entre el encabezamiento de trama lógica 22 y la sección de datos útiles 24 teniendo estos datos de RS 66 una longitud predeterminada, y una sección de mejora 68 puede situarse entre la sección de datos útiles 24 y la tabla de unidad de acceso 30 teniendo la última una longitud constante, o, como es el caso con la siguiente realización, una longitud variable que, por ejemplo, también depende de una manera conocida del número de unidades de acceso 28 el comienzo de las cuales cae en la trama lógica actual 20, o la indicación de la longitud de las cuales está presente en la misma sección de mejora, pero en el límite conocido entre la sección de mejora y la tabla de unidad de acceso. Los comienzos de las entradas de tabla de unidad de acceso 64 pueden espaciarse del extremo final 70 - o, en una realización alternativa, del extremo inicial - de la trama lógica 20 en unidades de una longitud constante. Por consiguiente, el recuperador 54 puede estar configurado para, al extraer la tabla de unidad de acceso 30 de una trama lógica 20, extraer de manera consecutiva el número de entradas de tabla de unidad de acceso 64, indicado por el encabezamiento de LF 22, empezando en el extremo final 70 de la trama lógica 20 o en una posición en la trama lógica 20, que tiene un desplazamiento constante predeterminando desde la misma, avanzando a través de las entradas de la unidad de acceso 64 en unidades de una longitud constante de estas entradas de tabla de unidad de acceso 64 desde una entrada de tabla de unidad de acceso a la siguiente, para obtener, para cada entrada de tabla de unidad de acceso, el puntero 40 que apunta al respectivo comienzo 32 de una correspondiente unidad de acceso 28. Puede ser ventajoso si el preparador de flujo de unidad de acceso 16 rellena la tabla de unidad de acceso 30 desde el extremo final registrado, es decir comenzando desde el extremo final 70 de la trama lógica 20. En otras palabras, el primer comienzo 32 de una unidad de acceso 28 que cae en una respectiva trama lógica 20 a lo largo de la dirección de inserción de datos útiles 60 se indica en la tabla de unidad de acceso 30 por medio de un respectivo puntero en la primera entrada de tabla de unidad de acceso 64 desde el extremo final 70 de la trama lógica 20, es decir la entrada de tabla de unidad de acceso 64 más cercana a, o que rodea, el extremo final 70 de la trama lógica 20. Los comienzos 32 que siguen en la dirección de extracción de datos útiles 60 siguen a lo largo de una dirección contraria a la dirección de inserción de datos del usuario 60. Esto es ventajoso porque el preparador 16 puede construir de manera continua las tramas lógicas 20 en lugar de construir las a nivel de trama lógica a la vez.

- 60 Como también es el caso con la realización descrita a continuación, el preparador 16 puede proporcionar cada tabla de unidad de acceso 30 con datos de redundancia adicional que permiten una detección y/o corrección de corrupción de datos individuales de entrada de tabla de unidad de acceso. En particular, cada propia entrada de tabla de unidad de acceso 64 puede comprender datos de redundancia adicional individuales. Cada entrada de tabla de unidad de acceso 64 puede proporcionarse con primeros datos de redundancia calculados a través de, y que

5 permiten detección de corrección de datos de, al menos el puntero 40 y, opcionalmente, la indicación de longitud opcional que indica la longitud 62 de la respectiva entrada de tabla de unidad de acceso 64. Como consecuencia, el recuperador 54 puede comprobar la corrección de las entradas de AUT individualmente, y una corrupción de una entrada de AUT no compromete la usabilidad de otra entrada de AUT de la misma tabla de AU. Además, como se señalará en más detalle a continuación, la existencia de los primeros datos de redundancia posibilita incluso que se compruebe la validez o existencia de una siguiente entrada de AUT en línea - cuando se avanza a través de las entradas de AUT que tienen una longitud constante - si, por ejemplo, el número total de entradas de AUT es desconocido debido a, por ejemplo, un encabezamiento de LF corrupto.

10 Adicionalmente o como alternativa, cada entrada de tabla de unidad de acceso 64 puede proporcionarse con segundos datos de redundancia calculados a través de, y que permiten detección de corrupción de datos de, el contenido de la unidad de acceso asociado con la respectiva entrada de tabla de unidad de acceso 64. Por lo tanto, la corrección de cada AU es detectable individualmente.

15 Además, de acuerdo con una realización, los primeros datos de redundancia se calculan a través de, y permiten la detección de corrección de datos de, el puntero 40, la indicación de longitud opcional - opcionalmente - y los segundos datos de redundancia de la respectiva entrada de tabla de unidad de acceso 64.

20 Con respecto a la Figura 4, se describe el modo de operación del preparador 16 al insertar de manera consecutiva las unidades de acceso 28 en las tramas lógicas 20 insertando los punteros 40 en las tablas de unidad de acceso 30 de acuerdo con una realización. El proceso de la Figura 4 empieza con el preparador 16 volviendo a la siguiente unidad de acceso 28 en la etapa 80, para continuar insertando de manera consecutiva las unidades de acceso 28 en las tramas lógicas 20. En la etapa 82, el preparador 16 a continuación comprueba en cuanto a si el comienzo 32 de la unidad de acceso actual 28 se ha de colocar o no en la trama lógica actual 20. En este sentido, se observa que la sección de datos útiles 24 de la trama lógica actual se reduce en tamaño tan pronto como un comienzo adicional 32 de una unidad de acceso 28 cae en esta trama lógica actual 20 a medida que se añade una entrada de tabla de unidad de acceso adicional 64 a la tabla de unidad de acceso 30 o, expresado de manera alternativa, la longitud de la tabla de unidad de acceso 30 aumenta. Debido a esto, el preparador 16 puede decidir colocar el comienzo 32 de la unidad de acceso 28 en la siguiente trama lógica aunque, actualmente, la sección de datos útiles 24 de la trama lógica actual 20 tuviera la capacidad de adaptar una pequeña cantidad de contenido de unidad de acceso adicional en el momento, y aunque el preparador 16 buscara, por ejemplo, insertar de manera ininterrumpida las unidades de acceso 28 en las tramas lógicas: tan pronto como el comienzo 32 se hubiera colocado en la trama lógica actual, la capacidad restante de la sección de datos útiles 24 de la trama lógica actual 20 no sería suficiente para adaptar la cantidad necesaria para añadir la nueva entrada de tabla de unidad de acceso 64 necesaria para indicar el puntero 14 a este nuevo comienzo 32 de la nueva unidad de acceso. Un ejemplo para una situación de este tipo se muestra de manera ejemplar en la Figura 1 en caso del comienzo 32 de la unidad de acceso AU<sub>3</sub>.

40 Sin embargo, los huecos como el hueco 58 pueden tener otras razones a aquellas mencionadas de manera ejemplar antes. Por ejemplo, los siguientes datos de AU en línea a procesarse mediante el preparador de flujo de unidad de acceso pueden simplemente no estar aún disponibles cuando la LF necesita ensamblarlos y emitirlos, por ejemplo, a la etapa de transmisión 36 para asegurar, por ejemplo, una señal de transmisión ininterrumpida 38, caso en el que se introduce relleno mediante el preparador 16 como se ha mencionado anteriormente.

45 Si el preparador 16 decide en la etapa 82 colocar el comienzo 32 de la unidad de acceso actual 28 en la siguiente trama lógica, el preparador 16 vuelve a la siguiente trama lógica en la etapa 84 que incluye, por ejemplo, cerrar la trama lógica actual y abrir la siguiente trama lógica. El cierre de la trama lógica actual puede implicar que el preparador 16 calcule los datos de redundancia adicional anteriormente mencionados para detección/corrección de corrupción de datos tales como datos de detección/corrección de errores hacia delante que se describe en más detalle a continuación, y enviar la trama lógica a la etapa de transmisión 36. La apertura de la siguiente trama lógica 20 puede implicar pre-configurar el encabezamiento de trama lógica a un estado inicial que indica, por ejemplo, que - hasta ahora - no existe tabla de unidad de acceso en estas siguientes tramas lógicas. Después de la etapa 84, el proceso vuelve a la etapa 82 de nuevo.

55 Si la comprobación en la etapa 82 produce que la unidad de acceso actual 28 tiene que tener su comienzo 32 colocado en la trama lógica actual, el proceso continúa a la etapa 86 donde el preparador 16 rellena la tabla de unidad de acceso y actualiza el encabezamiento de trama lógica. En particular, la etapa 86 puede implicar la generación de la tabla de unidad de acceso 30 en la trama lógica actual 20 en caso de que, el comienzo 32 que somete en la etapa 82 sea el primer comienzo 32 en la trama lógica actual. Para este comienzo 32, se inserta un puntero 40 y, opcionalmente, una indicación de longitud que indica la longitud 62, en la respectiva entrada de tabla de unidad de acceso 64, incluyendo opcionalmente datos de redundancia adicional para la entrada de la tabla de unidad de acceso 64 y, opcionalmente, de manera separada para la indicación de longitud.

60 La actualización del encabezamiento de trama lógica 22 comprende aumentar el número de entradas de tabla de unidad de acceso indicadas en este encabezamiento de trama lógica 22.

Como una consecuencia de la opción sí en la etapa 82, el preparador 16 también inserta la unidad de acceso actual 28 en la trama lógica actual 20 en la etapa 88. Durante esto, el preparador 16 comprueba continuamente en la etapa 90 en cuanto a si la trama lógica actual 20 está completa, es decir no pueden insertarse datos de unidad de acceso en la sección de datos útiles 24 de la trama lógica actual 20. Si este es el caso, el preparador 16 pasa a la etapa 92 para volver a la siguiente trama lógica 20, es decir cerrar la trama lógica actual y abrir la siguiente trama lógica. Después de la etapa 92, el proceso vuelve a la etapa 88. Sin embargo, si la trama lógica actual 20 aún no se ha completado o rellenado por completo, el preparador 16 realiza otra comprobación continuamente a través de la inserción en la etapa 88, en concreto la comprobación 94 en cuanto a si la unidad de acceso actualmente insertada se ha insertado por completo, es decir en cuanto a si se ha alcanzado el fin de la unidad de acceso actual. Si este es el caso, el proceso vuelve a la etapa 80 para volver a la siguiente unidad de acceso en secuencia 14. Si no, el proceso vuelve a la etapa 88.

Después de haber descrito, bastante en general, las realizaciones para la cadena de transmisión 10, el preparador de flujo de unidad de acceso 16, el decodificador 50 y el recuperador de flujo de unidad de acceso 54, así como los otros elementos de las Figuras 2 y 3, a continuación, se describen las realizaciones de la presente invención que deberán representar una posibilidad para ampliar la capacidad de transmisión proporcionada mediante DRM (DRM=Radio Digital Mundial) hasta el punto que no sea transmisible únicamente audio, texto o información de datos mediante DRM, sino que también sea transmisible vídeo o una mezcla de vídeo y audio u otra información textual alineada en el tiempo mediante DRM de una manera de modo que la calidad sea aceptable en el lado de recepción. Este concepto de transmisión para ampliar la capacidad de DRM se denomina Diveemo a continuación. Mediante Diveemo, pueden transmitirse programas de vídeo de educación e información mediante DRM, por ejemplo.

Las realizaciones para Diveemo, anteriormente conocidas como DrTV, descritas a continuación posibilitan la transmisión de vídeo y audio (posiblemente juntos con servicios de datos) mediante DRM (Radio Digital Mundial). Las posibles aplicaciones pueden implicar suministrar a los ciudadanos que viven en el extranjero con programas de información y de educación. La calidad de imagen y sonido obtenible está más en el extremo inferior de las expectativas del cliente, que es, sin embargo absolutamente suficiente para muchos campos de uso.

Diveemo forma una realización para aprovechar la idea de posibilitar servicios de vídeo mediante DRM. Los servicios de vídeo se han de adaptar a la norma de DRM, es decir tasas de bits disponibles bajas, señalización y configuración de servicio compatible con DRM, compatible con estructuras de transmisión proporcionadas por DRM, manejo eficaz de errores de recepción, etc.

De acuerdo con un aspecto adicional, las realizaciones de Diveemo descritas a continuación forman un esquema de transmisión que usa un enfoque genérico e independiente de DRM de transmisión de una serie de paquetes de datos independientes ("unidades de acceso") sin la necesidad de relleno o tara adicional como "el flujo de datos en serie" libre de interrupciones con máximo uso de la tasa de bits del canal de transmisión, en el que se usan simultáneamente las normas temporales y las estructuras de transmisión del esquema de transmisión usado, tal como DRM, para embeber la información requerida para decodificación y sincronización ascendente de la recepción en el flujo de datos, de manera que la extracción de esta información es rápidamente posible e, incluso con condiciones de recepción muy adversas que tienen muchos errores de bits, únicamente tiene pocas consecuencias con respecto a la pérdida de datos de cabida útil.

Funcionalidad de Diveemo: Diveemo abre la puerta a una amplia gama de nuevos servicios de información y de educación. Es una plataforma ideal para alcanzar audiencias dispersadas a través de un área geográfica extensa con un único transmisor, y mantener a los ciudadanos que viven en el extranjero informados y actualizados acerca de lo que está ocurriendo en el país local. Esta aplicación multimedia puede basarse en la norma de difusión terrestre global rentable DRM. Las transmisiones de DRM a través de onda corta tienen prácticamente posibilidades de cobertura ilimitadas que varían desde 100 kilómetros cuadrados hasta bien por encima de 5.000.000 kilómetros cuadrados dependiendo del sistema de transmisión. La aplicación Diveemo ofrece la posibilidad de recepción gratuita y es independiente de controladores de acceso y proveedores de terceros como redes de satélites y de cable. Las posibilidades son ilimitadas: un transmisor puede alcanzar millones de personas en cualquier lugar y en cualquier momento.

Diveemo puede proporcionar una codificación de transporte y empaquetamiento muy eficaz mientras al mismo tiempo permite que los receptores decodifiquen de manera robusta y (re-)sincronicen rápidamente el contenido transmitido. Un flujo de vídeo puede acompañarse por uno o más flujos de audio, que permite soporte de múltiples idiomas síncronos. El sistema también presenta todos los beneficios de la plataforma de DRM, como selección de servicio por etiquetas compatibles con Unicode, señalización y conmutación de frecuencia alternativa, características de anuncios y advertencias/alertas, etc.

Cuando se combina con otras tecnologías de DRM como el servicio de información basado en texto de tasa de bits muy eficaz Journaline®, que acompaña información de fondo textual sobre el programa, pueden estar

inmediatamente disponibles los subtítulos y servicios de subtulado para personas con problemas de audición en una multitud de idiomas y lenguaje de guiones.

De acuerdo con una realización, Diveemo puede implementarse como se señala a continuación. Para los fines de la siguiente descripción, se aplican los [siguientes] términos y definiciones. En particular, para los fines del presente documento, se aplican los siguientes símbolos:

5	$N_x$	El valor N se expresa en base x. La base x deberá ser decimal, por lo tanto $2A_{16}$ es la representación hexadecimal del número 42.
10	$\lceil X \rceil$	El valor integral más pequeño numéricamente mayor que x. En ocasiones conocido como la función "techo".
	$\lfloor X \rfloor$	El valor integral más grande numéricamente menor que x. En ocasiones conocido como la función "suelo".
	$\frac{x}{y}$	
15	$\text{MIN}\{a, \dots, z\}$	El resultado de dividir el valor x por el valor y. El valor más pequeño en la lista.

Además, el orden de los bits y bytes en cada descripción deberá usar la siguiente notación a menos que se establezca de otra manera:

- 20
- en las figuras, el bit o byte mostrado en la posición a la izquierda se considera que es el primero;
  - en las tablas, el bit o byte mostrado en la posición a la izquierda se considera que es el primero;
- 25
- en campos de bytes, el bit Más Significativo (MSb) se considera que es el primero y se indica mediante el número superior. Por ejemplo, el MSb de un único byte se indica "b<sub>7</sub>" y el bit Menos Significativo (LSb) se indica "b<sub>0</sub>";
  - en vectores (expresiones matemáticas), el bit con el índice más bajo se considera que es el primero.

30 El orden de transmisión para todos los valores numéricos definidos en este documento deberá ser MSb en primer lugar ('orden de byte de red').

### 35 Formato múltiplex

Diveemo puede llevar una serie de 'unidades de acceso' con audio, vídeo y potencialmente otro contenido en un subcanal de MSC. Este subcanal de MSC está configurado para llevar un servicio de datos en 'modo de flujo', es decir no usando el modo de paquete de DRM. Todas las unidades de acceso que pertenecen a un flujo virtual de contenido (por ejemplo todas aquellas que llevan los datos de vídeo) se marcan con un identificador de flujo virtual, denominado "Tipo de AU". Las unidades de acceso con el mismo tipo de AU se llevan en el subcanal de MSC en su orden de presentación. Las unidades de acceso con diferentes tipos de AU se llevan en el subcanal de MSC en forma intercalada, de manera que aquellas unidades de acceso que cubren los mismos tiempos de presentación se transportan agrupadas tan estrechamente como sea posible. Opcionalmente los datos de cabida útil en la sección de datos útiles y toda la información de encabezamiento se protege contra corrección de errores hacia delante (FEC) de errores de transmisión basándose en el algoritmo Reed-Solomon. Esto es similar a la especificación de modo de paquete mejorado del modo de paquete de DRM para permitir reutilizar el decodificador de RS así como el intercalador virtual.

La estructura de señalización de DRM en la que está basado Diveemo, ya posibilita las siguientes características:

- 50
- Hasta 4 servicios de DRM definidos en FAC (Canal de Acceso Rápido); Los servicios de DRM son elementos virtuales presentados al usuario para selección; cada servicio de DRM es de tipo audio (más componente de servicio de datos opcional como PAD - Datos Asociados de Programa) o un servicio de datos independiente
- 55
- MSC (Canal de Servicio Principal) lleva los flujos de bits reales en hasta 4 flujos de MSC; cada flujo de MSC lleva un componente de servicio de tasa de bits constante (contenido de audio o de Diveemo), o hasta 4 componentes de servicio en modo de paquete
- 60
- La protección de flujos de MSC puede seleccionarse desde hasta 2 niveles de protección/tasas de código (definidos por múltiplex de DRM); EEP (Protección de Error Igual) asigna uno de estos a un flujo de MSC, UEP usa ambos por flujo de MSC (con bytes mejor protegidos en el inicio de cada trama de transmisión)

- SDC (Canal de Descripción de Servicio) lleva información descriptiva para servicios de DRM (por ejemplo etiqueta, idioma, país de origen, etc.) e información de señalización general (frecuencias alternativas, fecha/hora actual, etc.); también lleva entidades que describen los componentes de servicio de la codificación de audio (entidad 9) y datos (entidad 5); el último puede usarse para Diveemo

5

La estructura de transporte de datos de MSC en la que se basa Diveemo, tiene las siguientes características:

- Estructura de trama de transmisión (TF) y súper trama de transmisión (TSF) con flujo de datos de tasa de bits constante para Diveemo;
- 10 • DRM30: TF = 400 ms; TSF = 1200 ms
- DRM+: TF = 100 ms; TSF = 400 ms
- TF en TSF se indexa: DRM30: 0-3; DRM+ 0-4;

15 En otras palabras, las tramas de transmisión representan porciones de una cierta longitud de la señal de transmisión y se sincronizan en la señal de transmisión. Las tramas lógicas describen el contenido de las tramas de transmisión.

### Estructura de trama

20 A continuación se describe cómo transportar la información de Diveemo en el subcanal de MSC de una Especificación de Sistema de Radio Digital Mundial DRM ETSI ES 201 980 V3.1.1. Una trama lógica de DRM contiene datos para 100 ms (modo de DRM A-D) o 400 ms (modo de DRM E) de valor de señal de difusión, respectivamente. Si una trama lógica de DRM lleva información de Diveemo, se denomina una trama lógica de Diveemo.

25 La estructura de una trama lógica de Diveemo 20 se muestra en la Figura 5. Comprende el encabezamiento de LF obligatorio 22, información de redundancia de Reed-Solomon en la sección de datos de RS opcional 66, la sección de datos útiles obligatoria 24, seguido por la sección de mejora opcional 68, y la tabla de AU al final 70.

30 La sección de datos útiles 24 contiene el contenido útil (tal como información de audio y vídeo) en forma de unidades de acceso 28. Cada unidad de acceso 28 describe contenido que cubre un cierto tiempo de presentación. Sigue directamente la unidad de acceso anterior en la sección de datos útiles 24 para formar un flujo de bytes continuo. Esta información de flujo de bytes 14 de unidad de acceso se divide en bloques de datos. Estos bloques de datos a continuación se colocan en la sección de datos útiles 24 de tramas lógicas de Diveemo 20 sucesivas. Por lo tanto una unidad de acceso 28 puede empezar en cualquier lugar en la sección de datos útiles 24, y puede abarcar

35 múltiples secciones de datos útiles 26 de tramas lógicas consecutivas 20.

Para cada inicio de unidad de acceso 32 que inicia en la sección de datos útiles 24 de la trama lógica de Diveemo actual 20, la sección de tabla de AU 30 lleva una entrada de tabla de AU 64. Estas entradas de tabla de AU 64 se ordenan de manera que la "Entrada 0 de Tabla de AU" que describe la primera unidad de acceso que se inicia en esta sección de datos útiles 24 se lleva al final de la sección de tabla de AU 36, la "Entrada 1 de Tabla de AU" que describe la segunda unidad de acceso que se inicia en la sección de datos útiles 24 se lleva justo antes de la "Entrada 0 de Tabla de AU", y así sucesivamente.

45 El encabezamiento de LF 22 está compuesto de los siguientes bits - mencionados en su orden de transmisión:

- 1 bit: bandera de mejora
- 7 bits: número de entradas de tabla de AU
- 8 bits: CRC calculada a través del primer byte del encabezamiento de LF, es decir a través del número de entradas de tabla de AU.

50

Los datos de RS 66 son un bloque de información de redundancia de Reed-Solomon como se describe a continuación. Evidentemente, otro código de redundancia, tal como un código de redundancia sistemática, puede usarse para proteger la LF y para colocarse en la sección 66, respectivamente.

55 Cada entrada de tabla de AU 64 está compuesta de los siguientes bits - mencionados en su orden de transmisión, es decir de izquierda a derecha en la Figura 5:

- 3 bits: ID de flujo de AU (el "identificador de flujo virtual"), toma los valores de 0-7; "7" puede reservarse para llevar datos de relleno); estos bits son opcionales y pueden no utilizarse de acuerdo con una realización alternativa.
- 60 1 bit: bandera de contenido; si el contenido es de tipo vídeo: bandera de fotograma I; de otra manera: rfa. Este bit puede omitirse también de acuerdo con una realización diferente.
- 12 bits: desplazamiento de AU 40 (valor de índice absoluto, indicando 0 el primer byte de la trama lógica de Diveemo 20). El desplazamiento de AU, por lo tanto, corresponde al puntero anteriormente

- mencionado 40 medido desde el extremo inicial de la trama lógica actual 20 en bytes. El número de bits, es decir 12, puede variarse de acuerdo con otra realización usando, por ejemplo, tramas lógicas más cortas, y adicionalmente o como alternativa, el desplazamiento de AU puede medir la longitud del puntero 40 en unidades distintas de bytes.
- 5 16 bits: longitud de AU. Longitud de AU, por lo tanto, corresponde a la longitud anteriormente mencionada 62. De nuevo, el número de bits, es decir 16, depende de la aplicación y puede variarse de acuerdo con una realización alternativa. Adicionalmente o como alternativa, la longitud de AU puede medir la longitud de la respectiva AU en unidades distintas a bytes, e incluso como alternativa, el punto de registro puede elegirse de manera diferente para apuntar al final de la respectiva AU de una manera diferente como se ha indicado anteriormente.
- 10 16 bits: indicación de tiempo de AU (véase descripción de temporización detallada a continuación. Esta información en la entrada de tabla de AU 64 es también opcional y puede omitirse de acuerdo con otras realizaciones.
- 15 16 bits: CRC de AU. La CRC de AU se calcula a través de los bytes de 'longitud de las AU' del contenido de la unidad de acceso. Por lo tanto, la CRC de AU también se ha mencionado anteriormente, en concreto como segundos datos de redundancia, y posibilita la detección de corrupción de datos en el contenido de la AU asociada con la respectiva entrada de AUT. De nuevo, el número de bits es opcional y puede variarse.
- 20 8 bits: CRC de entrada de tabla de AU calculada a través de los primeros 8 bytes de esta entrada de tabla de AU, es decir a través de ID de flujo de AU, bandera de contenido de AU, desplazamiento de AU, longitud de AU, indicación de tiempo de AU y la CRC de AU. La CRC de entrada de tabla de AU también se ha mencionado anteriormente como primeros datos de redundancia calculados a través de, opcionalmente, la indicación de longitud que indica la longitud 62, el puntero 40 y, también opcionalmente, los segundos datos de redundancia. En este punto, la CRC de entrada de tabla de AU protege también de manera ejemplar la información adicional en la entrada de tabla de AU. Esto es, por supuesto, opcional. Esto se cumple también para el número de bits gastados para la CRC de entrada de tabla de AU.
- 25

30 Si la bandera de mejora se establece a 1, existe la sección de mejora 68 insertada inmediatamente antes de la tabla de AU 30. De otra manera, no. La sección de mejora 68 puede usarse para futuras extensiones, es decir futuras funcionalidades. La sección de mejora 68 tiene el siguiente formato o está compuesta de los siguientes bits - mencionados en su orden de transmisión.

- 35
- |             |  |
|-------------|--|
| n x 8 bits: | datos de sección de mejora                                     |
| 8 bits:     | CRC calculada a través del último byte de la sección de mejora |
| 8 bits:     | longitud "n" de los datos de sección de mejora                 |

40 Obsérvese que la información estática se lleva al final de la sección de mejora 68, de modo que la longitud de la sección de mejora 68 puede derivarse en el lado del decodificador y mediante el recuperador de flujo de unidad de acceso 54, respectivamente, iniciando desde el límite bien conocido 26 de la sección de tabla de AU 30.

45 Por lo tanto, el preparador 16 establece, de acuerdo con la realización de Diveemo descrita a continuación, los bits anteriormente mencionados en el encabezamiento de LF 22, la entrada de tabla de AU 64 y los otros bits de la trama lógica como se ha indicado anteriormente usando, por ejemplo, un proceso de acuerdo con la Figura 4.

50 Opcionalmente, el contenido de la trama lógica de Diveemo o incluso múltiples LF sucesivas juntas pueden protegerse por corrección de errores hacia delante (FEC) de Reed-Solomon. Para calcular el código de Reed-Solomon, la información de redundancia 66 se calcula a través de la sección de encabezamiento de LF 22, la sección de datos útiles 24, la sección de mejora 68 (si está presente) y la información de tabla 30 (si está presente). Para aumentar la robustez del esquema de FEC, estos datos introducidos en el algoritmo de Reed-Solomon están virtualmente intercalados como se describe a continuación, es decir el preparador 16 calcula los datos de redundancia 66 del código de RS sistemático intercalando virtualmente los datos anteriormente mencionados de la trama lógica 20, pero envía la trama lógica en un formato intercalado, y el recuperador 54 puede o puede no comprobar la corrección de, y corregir, la información en una trama lógica recibida - recibida en el orden correcto - desintercalando la respectiva porción de la trama lógica recibida y comprobar los datos de trama lógica intercalados de esta manera mediante el uso de los datos de redundancia 66.

55

60 El esquema de FEC puede aplicarse basándose en cada trama lógica de Diveemo individual 20, o basándose en una súper trama de transmisión de DRM, que cubre 3 (modo de DRM A-D) o 4 (modo de DRM E) tramas lógicas de Diveemo 20, respectivamente. Si se activa o no la protección de FEC, y la configuración exacta del esquema de FEC puede definirse mediante los datos específicos de aplicación en el tipo 5 de entidad de datos de SDC de DRM. El algoritmo de Reed-Solomon puede definirse por RS(255; 239; 8), es decir generando 16 bytes de información de redundancia por 239 bytes de contenido.

El diagrama en la Figura 6 visualiza el enfoque de intercalación virtual. Es decir, para la intercalación virtual anteriormente mencionada, el preparador 16 puede insertar los datos de LF anteriormente mencionados relevantes, es decir todos menos los mismos datos de redundancia, por ejemplo - en la tabla de datos de aplicación 98, a nivel de columnas a lo largo de la trayectoria de inserción 100, emulando el recuperador 54 este procedimiento con los datos recibidos. La tabla de datos de aplicación 98 y la tabla de datos de RS 102 están yuxtapuestas entre sí columna a columna. El preparador 16 calcula los datos de RS en la tabla de datos de RS 102 a nivel de filas, es decir cada fila de la combinación de tablas 98 y 102 forma una palabra de código de RS, y la preparación 16 a continuación lee los bits de la tabla 102 a nivel de columnas a lo largo de la trayectoria 104 y rellena la sección de datos de RS 66 en este orden intercalado. El recuperador 54 desintercala los datos de RS 66 cuando se rellena la tabla 102.

Las siguientes definiciones se aplican a los valores R y C en la Figura 6:

- R: el número de filas de la tabla de intercalación virtual, los valores permitidos son, por ejemplo, de 1 a 511.
- C: se proporciona implícitamente por el valor de R puesto que el número de bytes a protegerse en la LF es conocido

El valor de R puede señalizarse en el tipo 14 de entidad de datos de SDC. El valor de C puede calcularse a partir de la tabla de datos de aplicación, que deberá ser justo lo suficientemente grande para mantener 1, 3 o 4 trama o tramas lógicas de Diveemo, como se señala en el SDC.

El número de columnas determina la cabida útil de los datos de FEC 66; cuanto más pequeño es el valor de C más alta es la cabida útil. El número de filas determina la profundidad de intercalación y el retardo de bloque; cuanto más pequeño es el valor de R menor es la intercalación y más bajo el retardo antes de que puedan procesarse los datos recibidos.

El relleno implícito puede aplicarse si se transmiten menos datos para rellenar todas las celdas en las tablas 98 y 102, respectivamente.

### 30 Señalización de DRM

Un servicio de Diveemo puede señalizarse en el canal de acceso rápido (FAC) con el valor de id de aplicación "27<sub>10</sub>" (5 bits).

35 La entidad 5 de datos de SDC puede tener la siguiente estructura:

- 1 bit: bandera de PM: 0 (modo de flujo de DRM)
- 3 bits: rfa
- 1 bit: bandera de mejora
- 3 bits: dominio de aplicación: 0x00 (aplicación de DRM)
- 16 bits: Id de aplicación: 0x5456 (ASCII para "TV")
- m x 8 bits: datos de aplicación (véase a continuación)

Formato de la entidad 5 de datos de SDC sección de datos de aplicación:

- 2 bits: versión mayor, valor actualmente 0
- 3 bits: versión menor, valor actualmente 0
- 1 bit: bandera de FEC (activada: 1; FEC no usado: 0)
- 1 bit: bandera de súper trama (FEC calculada a través de trama lógica de Diveemo: 0; sobre 3 o 4 LF, es decir una súper trama de transmisión de DRM: 1)
- 9 bits: número de filas para intercalación virtual (0..511; 0 únicamente si bandera de FEC = 0)
- n x 8 bits: uno o más bloques de configuración de AU (véase a continuación)

Cada bloque de configuración de AU:

- 5 bits: longitud de bloque de configuración en bytes
- 3 bits: Id de flujo de AU (elegido libremente para identificar flujo virtual de unidades de acceso que llevan mismo tipo de contenido; puede tomar valores de 0-7, aunque "7" está reservado para llevar datos de relleno)
- 3 bits: tipo de contenido (0: vídeo, 1: audio, otros valores: rfa)
- 5 bits: ID de códec (véase a continuación)
- n x 8 bits: configuración específica de códec (véase a continuación)

Configuración específica de códec para HE AAC v2 con audio de MPS opcional, ID de códec 0x00: (tipo de

contenido 1)

- 1 bit: bandera de SBR
- 2 bits: modo de audio (véase sistemas de DRM)
- 3 bits: tasa de muestreo de audio (véase sistemas de DRM)
- 2 bits: MPEG Surround (envolvente) (0: ninguno, 1: 5:1, 2: 7.1, 3: en banda)

5

Configuración específica de códec para vídeo H.264/AVC, Id de códec 0x00: (tipo de contenido 0)

- 2 bits: relación de aspecto (0: 4:3, 1: 16:9, otros valores: rfa)
- 11 bits: número de píxeles horizontales por trama
- 11 bits: número de píxeles verticales por trama
- 8 bits: número de tramas por segundo en pasos de 1/4.

10

15 Evidentemente, las realizaciones recién señaladas son simplemente ilustrativas y pueden usarse también otros códigos y valores o en el futuro.

Diveemo forma una realización para aprovechar la idea de posibilitar servicios de vídeo mediante DRM. Los servicios de vídeo se han de adaptar a la norma de DRM, es decir, tasa de bits disponible baja, señalización y configuración de servicio compatible con DRM, compatible con estructuras de transmisión proporcionadas por DRM, manejo eficaz de recepción de errores, etc.

20

Posibles consideraciones/estructuras para definición de Diveemo fueron que la señalización podría realizarse como la entidad 5 de SDC (nuevo tipo de aplicación de datos de 'Diveemo') y que la transmisión se realiza como un flujo de datos síncrono. Deberían cumplirse las siguientes restricciones: longitud de trama fija 400 ms (DRM30) / 100 ms (DRM+), y bytes/trama fijos (bps) en el intervalo: DRM30: 1..3598 bpf (71,960 bps) o DRM+: 1..2325 bpf (186,000 bps). Deberían cumplirse las siguientes restricciones/requisitos al definir formatos de contenido: asignación variable y dinámica de tasas de bits de audio/vídeo en el canal; debería existir algún requisito de almacenamiento en memoria intermedia mínimo; el decodificador de audio y vídeo debería aceptar cualquier tamaño de unidad de acceso flexible (equivalente tasa de bit); el decodificador de vídeo debería poder manejar 'cualquier' velocidad de fotogramas (dinámica), es decir el codificador puede ajustarse dinámicamente al contenido; el decodificador de vídeo debería poder manejar tramas omitidas de modo que los fotogramas I puedan usar segmentación (transporte en AU independientes); debería indicarse una indicación de tiempo por AU (contador de desbordamiento con relación a reloj básico común).

25

30

35

Los formatos que pueden usarse para el contenido de vídeo en las AU, son AVC/H.264 para vídeo, y HE-AAC v.2 (+ Surround (Envolvente)) o en la futura norma MPEG USAC ("Códec de Habla y Audio Unificados") para audio. Son posibles más adelante códec más nuevos/más eficaces.

40

Una suma de retardo de acceso cuando se aplica Diveemo a DRM, puede provenir a partir de los siguientes factores: retardo de recepción de DRM (decodificación de FAC/SDC, intercalador de MSC, etc.), FEC de Diveemo (intercalador) (opcional), tamaño de GoP (para recibir primer fotograma I) del códec de vídeo.

45

Además, los parámetros de vídeo que deberían considerarse cuando se transmite vídeo mediante Diveemo, son: fotogramas I que ocupan el 50 % de la tasa de bits (crítico para errores de recepción), únicamente debería usarse predicción hacia delante por razones de estabilidad, y velocidad de fotogramas dinámicamente adoptable (mediante el decodificador).

50

Con respecto a las indicaciones anteriormente mencionadas, deberían cubrirse las siguientes consideraciones: debería usarse base de reloj común para audio y vídeo; reloj básico con una granularidad de 1 ms parece ser un buen compromiso de modo que da como resultado una máxima fluctuación de 1/3 ms con velocidades de fotograma típicas (por ejemplo 15 fps); 16 bits por contador de reloj por AU debería estar OK (alrededor de 65 s. de ida y vuelta).

55

El retardo de inicio de presentación al que se enfrenta un receptor de Diveemo, es: duración de 1xGoP máxima (ajustado después de los primeros bits del fotograma i) + duración de 1x GoP (transmisión de siguiente fotograma i).

60

Además, cuando se implementa la realización de Diveemo descrita a continuación, debería considerarse lo siguiente: en la sincronización inicial, el receptor necesitaría esperar la entrada de encabezamiento de Diveemo con la bandera de fotograma i activa (→ primera AU de vídeo de un GoP y AU de audio correspondiente). Para añadir redundancia el encabezamiento de Diveemo podría espejarse en el final de una MSC-LF. De manera completa, por lo que un receptor pueda corregir entradas rotas fácilmente comparando las dos copias, o simplemente el primer byte del encabezamiento de Diveemo + CRC incluyendo el primer byte de cada entrada al final de una MSC-LF. Cada AU puede definirse por su ID de flujo de AU. Podría usarse el ID 7 de flujo de AU para describir datos de AU

virtuales que llevan bytes de relleno en el flujo continuo de un contenido de AU. El valor de indicación de tiempo por AU podría basarse en granularidad de 1 ms (es decir 16 que abarcan 65 segundos) como se ha mencionado anteriormente.

5 Se describen diversos diagramas de flujo de decodificación de Diveemo con respecto a las Figuras 7a a 7m. La decodificación de Diveemo descrita con respecto a estas figuras puede realizarse mediante el recuperador 54. En las Figuras 7a-7m, la decodificación se divide ampliamente en dos tipos diferentes. En primer lugar, se describe la decodificación de "trama lógica de Diveemo" que, en resumen, se denomina decodificación de DLF. En segundo lugar, se describe la decodificación de "súper trama de Diveemo" que, en resumen, se denomina decodificación de DSF.  
10

En decodificación de DLF, una trama lógica (LF) se almacena en memoria intermedia antes de que se inicie la decodificación real ya que se lleva a cabo FEC a través de una LF. En decodificación de DSF, dependiendo de la norma, se almacenan en memoria intermedia tres o cuatro tramas lógicas consecutivas antes de que se inicie la decodificación real ya que se lleva a cabo FEC a través de tres o cuatro tramas lógicas.  
15

En primer lugar, el recuperador de flujo de unidad de acceso 54 tiene que leer algunos parámetros de SDC, es decir parámetros de canal secundario en la señal de transmisión 38, como una bandera de FEC (FECF) que indica en cuanto a si la totalidad de las tramas lógicas se protegen por FEC, y una bandera de súper trama (SFF) que indica en cuanto a si las tramas lógicas se agrupan juntas en súper tramas, caso en el que se usa la decodificación de DSF anterior, número de filas en el intercaldor virtual, en concreto R como ya se ha indicado anteriormente y similares. Basándose en estos parámetros de servicio, el recuperador de flujo de unidad de acceso 54 a continuación empieza el proceso de decodificación que se describe en lo sucesivo con respecto a las Figuras 7a-7m.  
20

25 Como se muestra en la Figura 7a, el recuperador 54 intermitentemente, es decir cada LF en DLF, y cada SF en DSF, realiza las etapas el comienzo de las cuales se indica en la Figura 7a. En este comienzo, el recuperador 54 tiene conocimiento acerca de la información de canal secundario de la señal de transmisión 38, en concreto la porción de SDC de la misma, en concreto acerca de los parámetros de SDC FECF, SFF y R como se indica en 150. El proceso se inicia con el recuperador 54 comprobando en la etapa 152 en cuanto a si la FECF señala que se usa o no protección de FEC. En caso afirmativo, el preparador 16 ha embebido las tramas lógicas 20 en paquetes de transporte con datos de FEC 66 para proteger el contenido de las tramas lógicas. Si se usa agrupación de súper trama, el código de FEC del flujo de trama de transporte se define a través de tres o cuatro tramas lógicas consecutivas como se ha descrito anteriormente. Si se activa la FECF, la decodificación con FEC tiene lugar en 154, de otra manera la decodificación sin FEC tiene lugar en la etapa 156. Los detalles con respecto a las etapas 154 y 35 156 se describen con respecto a la Figura 7b y la Figura 7i, respectivamente.

La decodificación sin FEC en la etapa 156 se muestra en más detalle en la Figura 7b. El proceso de decodificación de una trama lógica sin FEC en el proceso 154 se inicia con información a priori conocida a partir de la decodificación de una trama lógica anterior como se indica en 158. Esta información se denomina CAUB. La información de CAUB es una estructura que consiste en variables que ayudan al recuperador 54 a decodificar, entre otros, la CAU, es decir la AU que continúa, es decir, la unidad de acceso el comienzo 32 de la cual radica en la trama lógica 20 que precede a la trama lógica 20 actualmente bajo consideración. Se usan las siguientes abreviaturas en la siguiente descripción y son conocidas a partir de la información de CAUB:  
40

45 AU: unidad de acceso; a continuación, la abreviatura AU se usa para indicar las AU el comienzo de las cuales cae en la LF actual, al contrario de las CAU  
CAU: AU que continúa;  
CAUF: bandera de CAU, es decir una bandera que indica la existencia o no existencia de una AU que continúa, que se extiende en la LF actual  
50 PCAUB: bytes de CAU parciales, que indican los bytes de la CAU que preceden al límite entre la LF actual y la LF anterior, es decir los bytes de la CAU que ya se han leído.  
LPCAUB: longitud de PCAUB, es decir el número de bytes o la longitud de, PCAUB  
CAUSID: ID de flujo de CAU, es decir el valor de ID de flujo de AU de CAU.  
CAUL: longitud de CAU, es decir la longitud de CAU, es decir longitud 62 de la Figura 1, indicado por la  
55 indicación de longitud de la respectiva entrada de tabla de unidad de acceso en la trama lógica anterior  
CAUCB: bits de CAUCRC, es decir bits de CRC para posibilitar nuestra detección hacia delante en la CAU transmitida en la sección de mejora anteriormente mencionada.

60 Otros valores pueden pertenecer a la información de CAUB también, tal como el tipo de contenido de AU, indicación de tiempo de AU, el valor de la bandera de mejora en las LF, etc.

Siempre que las etapas de la Figura 7b se realicen mediante el recuperador 54, la información de CAUB permanece estática.

En la etapa 160, el recuperador 54 lee la siguiente trama lógica, es decir la trama lógica actual, en una memoria intermedia interna del recuperador 54 no mostrada en la Figura 3. En la siguiente etapa, en concreto la etapa 162, el recuperador 54 decodifica esta trama lógica 20 en unidades de acceso describiéndose adicionalmente esta etapa en la Figura 7c. A continuación, el recuperador de flujo de unidad de acceso 54 almacena en memoria intermedia las unidades de acceso decodificadas de esta manera en la etapa 164 y actualiza la información de CAUB en la etapa 166 para iniciar de nuevo el procesamiento en la etapa 158.

El proceso de decodificación de la trama lógica actual en unidades de acceso, en concreto la etapa 162, se inicia, como se muestra en la Figura 7c, teniendo conocimiento el recuperador 54 de dos cosas, en concreto bytes de LF, es decir los bytes de la trama lógica actual, e información de CAUB. El número de bytes de las tramas lógicas 20 está fijado o se indica de otra manera, tal como varía por medio de respectiva señalización de información secundaria en el canal de información secundaria separado anteriormente mencionado de la señal de transmisión 38. En la etapa 170, el recuperador 54 lee el encabezamiento de LF y su CRC. En la etapa 172, el recuperador 54 comprueba la CRC en cuanto a si la misma coincide con la información de encabezamiento de LF, es decir la bandera de mejora así como el número de entradas de tabla de unidad de acceso contenidas en el encabezamiento de trama lógica 22. Si la CRC no coincide en la etapa 172, el recuperador comprueba en la etapa 174 en cuanto a si hay una CAU, es decir el recuperador 54 comprueba la bandera de CAU interna en cuanto a si la misma indica que hay una unidad de acceso 28 el comienzo 32 de la cual radica en cualquiera de las tramas lógicas anteriores mientras que el fin de esta unidad de acceso no se haya alcanzado aún. Si este es el caso, el recuperador 54 realiza en la etapa 176 la decodificación de los bytes de CAU. La etapa 176 se explica adicionalmente en la Figura 7e. Sin embargo, si la bandera de CAU resulta que no está activada en la etapa 174, el recuperador 54 continúa a la etapa 178, donde la trama lógica actual se descarta o, más preferido - y este caso se considera en este punto, se somete a unas entradas de tabla de AU de decodificación y pruebas de bytes de Au localizando entradas de AUT válidas evaluando la CRC de entrada de tabla de AU, después de lo cual el proceso vuelve a la etapa 158 en la Figura 7b. La etapa 178 corresponde a una concatenación de las porciones de procesos de la Figura 7f - 7g. De manera análoga, como se indica mediante líneas discontinuas, aunque el aprovechamiento de LF puede detenerse después de la etapa 176, es también posible que el recuperador 54 continúe después de la etapa 176 intentando recuperar tantas entradas de AUT como sean posibles a partir del número de entradas de AUT realmente presentes conocidas a priori (debido a la corrupción del encabezamiento de LF).

Sin embargo, si la comprobación de CRC en la etapa 172 da como resultado una coincidencia entre la información de CRC y el encabezamiento de información de LF, el recuperador 54 continúa intentando extraer información desde las siguientes porciones de la trama lógica actual. En particular, como se indica en 180, si la CRC coincide en la etapa 172, el recuperador 54 aprovecha adicionalmente al menos dos elementos de información proporcionados en el encabezamiento de trama lógica 22 correctamente transmitidos de la trama lógica actual 20, en concreto el número de entradas de tabla de unidad de acceso de la tabla de unidad de acceso de la trama lógica actual, y el conocimiento acerca del valor de la bandera de mejora EF. Resultando a partir del conocimiento del número de entradas de tabla de unidad de acceso en la tabla de unidad de acceso de la trama lógica actual, el recuperador 54 puede obtener TAUB, es decir el número total de bytes de unidad de acceso. Resultando a partir del conocimiento de EF, el recuperador 54 tiene conocimiento acerca de la existencia o no existencia de la sección de mejora 68 anteriormente mencionada. En particular, en la etapa 182, el recuperador 54 calcula TAUB multiplicando el número de entradas de tabla de unidad de acceso presentadas en el encabezamiento de trama lógica 22 por la longitud constante común a todas las entradas de tabla de unidad de acceso. Posteriormente, en la etapa 184, el recuperador 54 decodifica los bytes en una sección de datos útiles 24 de la trama lógica actual, procedimiento que se señala en más detalle con respecto a la Figura 7d.

Por lo tanto, la porción de procedimiento mostrada en la Figura 7c crea tres casos:

- a) El encabezamiento de LF es correctamente decodificable y por consiguiente, las unidades de acceso se han de decodificar
- b) El encabezamiento de LF no es decodificable y por consiguiente, las unidades de acceso no pueden decodificarse, pero la CAU está presente y puede decodificarse
- c) El encabezamiento de LF no es decodificable y por consiguiente, las unidades de acceso no pueden decodificarse pero puede llevarse a cabo un intento para encontrar entradas de AUT válidas. Adicionalmente, la CAU tampoco está presente y por lo tanto, no se ha de codificar.

La decodificación de los bytes de aplicación de LF en la sección de datos útiles 24 se describe a continuación con respecto a la Figura 7d. Cuando se entra en esta sección de proceso, el recuperador 54 tiene el conocimiento acerca de los bytes de LF, es decir los bytes de la trama lógica, información de CAUB, EF, TAUE, es decir el número total de entradas de tabla de unidad de acceso, y TAUB, es decir el número total de bytes de la tabla de unidad de acceso como se muestra en 186. El recuperador 54 comprueba en la etapa 188 CAUF para comprobar en cuanto a si hay o no una CAU. En caso afirmativo, el recuperador 54 decodifica en la etapa 190 los bytes de CAU en la LF actual, una etapa que se describe adicionalmente a continuación con respecto a la Figura 7e. Después de la etapa

188 o 190, el recuperador 54 continúa a la etapa 192 para comprobar TAUE en cuanto a si el número de entradas de tabla de unidad de acceso es o no cero. Si este número es cero, es decir no hay tabla de unidad de acceso en la trama lógica actual, el proceso de la Figura 7e finaliza. En caso negativo, sin embargo, el recuperador 54 decodifica en la etapa 194 las entradas de tabla de AU y decodifica en la etapa 196 los bytes de unidad de acceso desde la sección de datos útiles de LF. La etapa 194 se explica adicionalmente con respecto a la Figura 7f, y la etapa 196 con respecto a la Figura 7g. Al final de la porción de proceso de la Figura 7d, el recuperador 54 posee conocimiento acerca de una información de CAUB actualizada y tiene versiones almacenadas en memoria intermedia de las unidades de acceso como se indica en 198.

10 Por lo tanto, la porción de proceso de la Figura 7d, pertenece a los siguientes casos que pueden tener lugar:

- a) los datos de AU en la sección de datos útiles 24 de la trama lógica contienen tanto CAU como AU,
- b) los datos de AU en la sección de datos útiles 24 de la trama lógica actual contienen únicamente CAU, o
- c) los datos de AU en la sección de datos útiles de la trama lógica actual contienen únicamente AU.

15 A continuación, se describe en más detalle la decodificación de bytes de CAU en las etapas 176 y 190 con respecto a la Figura 7e. En el comienzo de esta porción de proceso, el recuperador 54 ya posee conocimiento acerca de los bytes de LF, es decir el número de bytes en la trama lógica actual, e información de CAUB como se indica en 200. En el comienzo, el recuperador 54 se inicia leyendo datos de CAU en la etapa 202. El recuperador 54 puede localizar el comienzo de la sección de datos útiles 24 de la trama lógica actual desde donde se ha de comenzar la lectura en la etapa 202, puesto que este punto de inicio es constante durante el tiempo a través de las tramas lógicas debido a la longitud constante del encabezamiento de trama lógica 22 y el conocimiento acerca de la existencia de (véase FECF), y la longitud constante de, la sección de datos de FEC opcional 66. En particular, en la etapa 202, el recuperador 54 intenta leer tantos bits desde el comienzo de la sección de datos útiles 24 de la trama lógica actual como pertenezcan a la CAU, es decir la unidad de acceso que se extiende en la trama lógica actual desde la trama lógica anterior. Pueden tener lugar dos casos. En primer lugar, el recuperador 54 puede encontrar el fin de la CAU antes del fin de la sección de datos útiles de la trama lógica actual. En segundo lugar, el recuperador 54 puede encontrar el fin de la sección de datos útiles 24 de la trama lógica actual antes del fin de la CAU. El recuperador 54 puede prever la situación basándose en dos informaciones, en concreto CAUL, es decir la longitud de CAU conocida desde cualquiera de las tramas lógicas anteriores, en particular la respectiva entrada de tabla de unidad de acceso de la respectiva tabla de unidad de acceso en la misma, y el número de bytes de la trama lógica junto con el conocimiento acerca de la existencia o no existencia de la sección de mejora 68 (véase bandera de mejora) y acerca de la longitud de esta sección 68 como puede derivarse a partir de la posición predeterminada en esta sección que revela la longitud de la sección 68 - registrada con relación a la AUT - puesto que esta información también define el número máximo de bytes disponibles para una trama lógica en caso de que no exista tabla de unidad de acceso. Después de haber leído los bits de CAU, es decir la porción de la sección de datos útiles 24 desde el comienzo de esa porción, el recuperador 54 actualiza en la etapa 204 el estado interno de LPCAU, es decir la longitud de la parte de CAU que ya se ha recuperado desde la secuencia de las tramas lógicas hasta ahora. En la etapa 206, el recuperador 54 comprueba en cuanto a si la LPCAU equivale a la CAUL, es decir en cuanto a si la totalidad de la CAU se ha recuperado desde la secuencia de las tramas lógicas hasta ahora. En caso negativo, la CAU continúa extendiéndose a la siguiente trama lógica y el recuperador 54 actualiza en la etapa 208 PCAUB y LPCAUB en información de CAUB en consecuencia, es decir actualiza la información en la parte de CAU que ya se ha recuperado desde la secuencia de las tramas lógicas incluyendo la trama lógica actual. Sin embargo, si la comprobación en la etapa 206 revela que la CAU se ha recuperado desde la secuencia de las tramas lógicas por completo, es decir que el fin de la CAU cayó en la sección de datos útiles 24 de la trama lógica actual, el recuperador 54 comprueba en la etapa 210 en cuanto a si la información de CRC en CAU que puede derivarse desde la respectiva entrada de tabla de AU 64 en la trama lógica en la que cayó el comienzo 32 de la CAU coincide con los bits de CAU recuperados y almacenados en memoria intermedia hasta ahora para la CAU. Si este es el caso, el recuperador 54 almacena en memoria intermedia la unidad de CAU decodificada en la etapa 212 y resetea en la etapa 214 los parámetros en información de CAUB como hasta el punto de que no existan actualmente más CAU. Sin embargo, si la comprobación de la CRC en la etapa 210 revela que los bits de CAU que se han recuperado desde la secuencia de las tramas lógicas están corruptos, el recuperador 54 descarta esta unidad de acceso CAU en la etapa 216 y continúa a la etapa 214 donde esta porción de proceso de la Figura 7e finaliza. Como alternativa, el recuperador 54 puede marcar la CAU en la etapa 216 como errónea y pasar la misma al presentador 56 que, a su vez, puede estar configurado para derivar contenido útil a partir de la AU errónea por otros medios tales como FEC interno de AU o datos de CRC o similares, o simplemente analizando satisfactoriamente la AU marcada de manera falsa como errónea debido a una CAUCRC corrupta.

60 Por lo tanto, la Figura 7e trata tres casos diferentes, en concreto

- a) CAU se inicia en una trama lógica anterior que tiene, por ejemplo, número de LF n y finaliza en la trama lógica actual que tiene, por ejemplo, número n+1, donde CAUCRC coincide con el byte de CAU, es decir el byte de CAU se ha recuperado correctamente desde la secuencia de las tramas lógicas.
- b) CAU se inicia en cualquiera de las tramas lógicas anteriores, tal como la trama lógica que tiene número n y

finaliza en la trama lógica actual que tiene, por ejemplo, número  $n+1$ , y CAUCRC no coincide con el byte de CAU, es decir los bytes que se han recuperado para CAU o el valor de CAUCRC están corruptos.

5 c) CAU se inicia en cualquiera de las tramas lógicas anteriores, tal como la trama lógica que tiene número de trama lógica  $n$ , pero esta CAU se extiende más allá de la trama lógica actual que tiene, por ejemplo, número de LF  $n+1$ , y se extiende en y posiblemente finaliza en la siguiente trama lógica que tiene, por ejemplo, número de trama lógica número de LF  $n+2$ .

10 La porción de proceso de decodificación de las entradas de tabla de unidad de acceso en la etapa 194 y la inspección de la porción de la LF que potencialmente corresponde a entradas de AUT válidas en la etapa 178 se analiza en más detalle a continuación con respecto a la Figura 7f. Cuando se entra en esta porción de proceso desde la etapa 194, el recuperador 54 tiene conocimiento acerca de TAUE, es decir el número de entradas de tabla de unidad de acceso en la trama lógica actual, y los bytes de LF, es decir los bytes de la trama lógica actual, como se indica en 218. Cuando se entra en esta porción de proceso desde la etapa 178, el recuperador 54 no tiene conocimiento acerca de TAUE, y el recuperador 54 puede establecer TAUE al número de entradas de tabla de unidad de acceso máximamente posible en las tramas lógicas que es, en el presente caso (debido a que el encabezamiento de FL gasta simplemente un número constante - en este punto, de manera ejemplar 7 - bits para indicar el número de entradas de AUT) 128.

20 En esta porción de proceso de la Figura 7f, el recuperador 54 se inicia en la etapa 220 con la inicialización de un valor de recuento interno, en concreto RAUE, es decir el número de entradas de unidad de acceso restantes, para igualar TAUE, el número total de entradas de unidad de acceso. Por supuesto, sería posible contar las entradas procesadas iniciando el valor de recuento en uno en lugar de reducir el valor de recuento.

25 Después de haber comprobado en la etapa 222 en cuanto a si el valor de RAUE equivale a cero, es decir en cuanto a si no quedan entradas de tabla de unidad de acceso para leerse, el proceso continúa con la etapa 224 en caso de que hayan entradas de tabla de unidad de acceso restantes para leerse desde la trama lógica actual y su tabla de unidad de acceso, respectivamente, donde el recuperador 54 lee AUTEb, es decir los bytes que corresponden a la entrada de tabla de unidad de acceso en línea, y la AUTEcB, es decir los bits de CRC de entrada de tabla de unidad de acceso que corresponden a la misma. Como ya se ha descrito anteriormente, el orden en el que el recuperador 30 54 accede a la entrada de tabla de unidad de acceso 64 puede ser desde el extremo final 70 hacia el extremo inicial 72 de la trama lógica que corresponde al orden de la unidad de acceso el comienzo 32 de la cual se indica en estas entradas de tabla de unidad de acceso 64. Después de la etapa 224, el recuperador 54 comprueba en la etapa 226 la CRC de la entrada de tabla de unidad de acceso que se acaba de leer en la etapa 224 para comprobar en cuanto a si los datos de entrada de tabla de unidad de acceso que se han recuperado desde la trama lógica actual están o no corruptos. Si es así, es decir si los datos están corruptos, el recuperador 54 descarta la respectiva entrada de tabla de unidad de acceso 64 y establece una bandera correspondiente que indica el descarte en la etapa 228. Como alternativa, el recuperador 54 puede intentar re-construir al menos parcialmente la información inválida a través de otros medios. Por ejemplo, el recuperador 54 puede intentar predecir desplazamiento de AU desde la longitud de AU o viceversa (suponiendo, por ejemplo, inserción de manera ininterrumpida de los datos útiles en la sección 24, y rehacer la comprobación de CRC con la respectiva porción prevista de la entrada de tabla de AU sustituida por el resultado de predicción. Mediante esta manera, el recuperador 54 puede obtener, a pesar de la corrupción de datos, una entrada de tabla de AU correcta que coincide con la correspondiente CRC.

45 Si la CRC para la entrada de tabla de unidad de acceso actual 64 coincide con los datos correspondientes, el recuperador 54 interpreta en la etapa 230 la entrada de tabla de unidad de acceso actual 64 y la graba. Debería destacarse que el caso de coincidencia de CRC es también una indicación de que el recuperador 54 acaba de encontrar una entrada de AUT válida 64. Incluso en otras palabras, cuando se realiza la etapa 226 y se ha introducido la porción de proceso de la Figura 7f desde la etapa 178, el recuperador 54 no tiene conocimiento con antelación en cuanto a si la porción actualmente inspeccionada de la LF - localizada aprovechando el hecho de que la tabla de AU está registrada en el fin de LF y las entradas de tabla de AU están situadas a paso constante - realmente forma parte de la sección de datos útiles 24 o la sección de mejora 68, o la AUT 30 respectivamente. El recuperador 54 puede usar la coincidencia de CRC como un resultado de comprobación suficiente para interpretar la porción actualmente inspeccionada de la LF como una entrada de AUT 64. Como alternativa, el recuperador 54 puede realizar pruebas adicionales, tales como comprobaciones de plausibilidad, dependiendo de que la entrada de AUT posible actual se considere como válida o inválida en caso de que el encabezamiento de LF estuviera corrupto. Por ejemplo, el comienzo 32 de la AU de la entrada de AUT que se acaba de encontrar 64 debería permanecer después del fin de la AU anterior y, a la inversa, el fin de la AU de la entrada de AUT que se acaba de encontrar 64 debería permanecer antes del fin de la siguiente AU, y por consiguiente, si alguna de las comprobaciones de plausibilidad diera como resultado una contradicción, la entrada de AUT supuesta actual se rechaza y se considera como inválida.

60 Después de la etapa 230, el recuperador 54 tiene conocimiento acerca del comienzo 32 de la correspondiente unidad de acceso asociada con la entrada de tabla de unidad de acceso actual 64 así como, opcionalmente, acerca de la longitud 62 de la misma. Considerando los contenidos adicionales de las entradas de tabla de unidad de

acceso 64, se hace referencia al análisis anterior de tales opciones adicionales. Independientemente de la comprobación de coincidencia de CRC en la etapa 226, el recuperador 54 reduce después de cualquiera de las etapas 228 y 230 en la etapa 232 el estado del contador interno RAUE en uno y vuelve a la etapa 222. Tan pronto como esta comprobación 222 revela que el restante número de entradas de tabla de unidad de acceso 64 a recuperarse desde la tabla de unidad de acceso 30 de la trama lógica actual es cero, la porción de proceso de la Figura 7f finaliza habiendo rellenado el recuperador 54 la réplica interna de la tabla de unidad de acceso 30, es decir la estructura de entrada de tabla de unidad de acceso como se indica en la etapa 234.

En otras palabras, la porción de proceso de la Figura 7f se introduce mediante el recuperador 54 si existe alguna tabla de unidad de acceso en la trama lógica actual, es decir cuando existe alguna entrada de tabla de unidad de acceso 64, o para evaluar en cuanto a si existe alguna entrada de AUT 64 debido a que no se conoce la existencia o no existencia (y número) de entradas de AUT. Puede ser posible que la CAU pueda tomar la totalidad de la trama lógica de modo que en este caso, por ejemplo, el recuperador 54 puede no entrar en la porción de proceso de la Figura 7f. Se prevén los siguientes casos diferentes en la porción de proceso de la Figura 7f:

- a) la entrada de tabla de unidad de acceso actual y su CRC coinciden (y han pasado todas las comprobaciones de plausibilidad)
- b) la entrada de tabla de unidad de acceso actual y su CRC no coinciden (o la comprobación de plausibilidad falla)
- c) todas las entradas de tabla de unidad de acceso - todas las entradas conocidas que son existentes desde el encabezamiento de LF, o todas las posibles entradas de AUT, simplemente existentes de manera presumible, (debido a corrupción del encabezamiento de LF) - se han procesado independientemente de que su CRC haya coincidido o no coincido (casos a y casos b), en el que la condición c es también la condición de salida para esta porción de proceso de la Figura 7f.

Debería observarse, que la razón para que el recuperador 54 pueda leer AUTEb y AUTEcB independientemente de que cualquiera de las entradas de tabla de unidad de acceso anteriores de la misma tabla de unidad de acceso de la trama lógica actual se hayan corrompido o no, es que todas las entradas de tabla de unidad de acceso son del mismo tamaño, y que la tabla de unidad de acceso 30 se registra con su extremo final al extremo final 70 de la trama lógica actual de modo que el recuperador 54 puede localizar las entradas de tabla de unidad de acceso en cualquier caso. Existen alternativas, ya se ha mencionado anteriormente.

Además, debido a que los datos de redundancia adicional mediante los que se proporciona la tabla de unidad de acceso y la trama lógica actual, que posibilita una comprobación de corrupción de datos individuales de entrada de tabla de unidad de acceso, una entrada de tabla de unidad de acceso que se ha transmitido sin errores en el lado de decodificación, puede evaluarse mediante el recuperador 54 independientemente del éxito o denegación al transmitir cualquiera de las otras entradas de tabla de unidad de acceso.

A continuación, con respecto a la Figura 7g, se describen detalles de la decodificación de bytes de AU en la etapa 196 o 178. Cuando se entra en esta sección de proceso, el recuperador 54 tiene conocimiento de la siguiente información: TAUE, es decir el número total de entradas de tabla de unidad de acceso, información de AUE, es decir el contenido de las entradas de tabla de unidad de acceso de la trama lógica actual así como tramas lógicas anteriores, información de CAUB, bytes de LF y TAUB, es decir el número total de bytes disponibles en la sección de datos útiles 24 de la trama lógica actual para las unidades de acceso 28 el comienzo 32 de las cuales cae en la trama lógica actual, indicándose esta posesión de pre-conocimiento en 236. El recuperador 54 tiene conocimiento acerca de TAUB debido a la siguiente razón: el recuperador 54 tiene conocimiento que los datos de unidad de acceso que pertenecen a unidades de acceso el comienzo de las cuales cae en la trama lógica actual 20 se extiende desde el comienzo 32 más cercano al extremo inicial 72 de la trama lógica 20. Esto corresponde a la posición señalada mediante el puntero 40 de la entrada de tabla de unidad de acceso 64 que se ha recibido sin corrupción de datos y el tiempo más anterior en la porción de proceso de la Figura 7f - o se ha obtenido, aunque está corrupto, mediante FEC como se describe más adelante con respecto a decodificación de DSF o DLF. El fin de la sección de datos útiles 24 es conocido para el recuperador 54 basándose en la información de encabezamiento de trama lógica, es decir el número de entradas de tabla de unidad de acceso 64 junto con la indicación (véase el encabezamiento de LF) de la existencia o no existencia de la sección de mejora 68 directamente contigua a la tabla de AU 30 y que está situada entre la sección de datos útiles 24 y la tabla de unidad de acceso 30, así como la longitud de la sección 68 como puede derivarse desde el byte dentro de esta sección inmediatamente adyacente a - o, como alternativa, que tiene un desplazamiento predeterminado desde - el límite de la AUT 30 frente a la sección 68. Si la última información no está disponible, es decir ni el número de entradas de AUT ni la existencia o no existencia de la sección 68, el recuperador 54 puede restringir TAUB para medir la sub-parte de la sección de datos útiles 24 que se extiende desde el comienzo más a la izquierda 32 anteriormente mencionado señalado mediante cualquiera de las entradas de AUT válidamente encontradas y la última, es decir el comienzo más a la derecha 32 señalado mediante cualquiera de las entradas de AUT válidamente encontradas puesto que esta porción simplemente no contiene las CAU.

En el comienzo de la porción de proceso de la Figura 7g, el recuperador 54 inicializa en la etapa 238 dos parámetros internos, en concreto RAU, es decir el número de unidades de acceso restantes que no se han procesado todavía en la sección de proceso de la Figura 7g, y RAUB, es decir el número de bytes de AU restantes que no se han leído todavía desde la sección de datos útiles 24 de la trama lógica actual. Ambos parámetros se establecen igual a TAU, es decir el número total de unidades de acceso el comienzo 32 de las cuales cae en la trama lógica actual siendo este número conocido desde el encabezamiento de trama lógica, y TAUB, respectivamente. En la etapa 240, el recuperador 54 comprueba en cuanto si RAU equivale a cero, es decir en cuanto si quedan unidades de acceso a procesarse, que tienen su comienzo 32 en la trama lógica actual. Si no, el recuperador 54 continúa en la etapa 242 leyendo los bytes de la unidad de acceso actual desde la sección de datos útiles 54 de la trama lógica actual en el que la etapa se describe en más detalle a continuación con respecto a la Figura 7h. Posteriormente, el recuperador 54 comprueba en la etapa 244 en cuanto a si la CRC asociada con la unidad de acceso actual coincide con los datos de la unidad de acceso actual leídos en la etapa 242. Si este no es el caso, el recuperador 246 descarta la unidad de acceso actual en la etapa 246, o, como ya se ha señalado anteriormente, marca esta AU como errónea y la pasa para un procesamiento/prueba adicional. Sin embargo, si la CRC coincide con la unidad de acceso actual, el proceso de la Figura 7g continúa con el recuperador 54 almacenando en memoria intermedia la unidad de acceso actual en la etapa 248 para enviar la misma a los datos actuales 56, por ejemplo. Después de cualquiera de las etapas 246 y 248, el recuperador 54 actualiza en la etapa 250 los estados internos de RAU y RAUB. En particular, RAU, es decir el número de unidades de acceso a procesar, se reduce en uno, y RAUB, es decir el número de bytes de unidad de acceso disponibles en la sección de datos útiles 24 se actualiza, es decir, se reduce en el número de bytes de la unidad de acceso actualmente procesada o, se diferencia de la misma, a una diferencia entre el extremo final de la sección de datos útiles 24 y el comienzo 32 de la siguiente unidad de acceso a procesar. Como una medida excepcional, el recuperador 54 puede establecer RAUB a cero cuando encuentra el fin de la sección de datos útiles 24.

Después de la etapa 250, la Figura 7g vuelve a la etapa 240 para procesar la unidad de acceso restante a procesar y comienza en la trama lógica actual.

Tan pronto como la comprobación en la etapa 240 da como resultado que RAU es igual a cero o que RAUB es igual a cero, el recuperador 54 finaliza la porción de proceso de la Figura 7g habiendo almacenado en memoria intermedia la unidad de acceso recibida satisfactoriamente como se indica en la etapa 252.

Con respecto a la Figura 7g, se observa que la descripción anterior de la Figura 7g olvida el hecho de que alguno del número - el número máximamente representable por, o el número correctamente transportado por, el encabezamiento de LF - de entradas de tabla de unidad de acceso puede haberse corrompido o no y que, por consiguiente, puede haberse establecido una respectiva bandera en la etapa 228 en la Figura 7f. En el análisis anterior de la Figura 7g, por ejemplo, TAU puede ya haberse reducido por este número de entradas de tabla de unidad de acceso corruptas, de modo que TAU simplemente establece el número de unidades de acceso el comienzo de las cuales cae en la trama lógica actual y para las que las entradas de tabla de unidad de acceso asociadas eran usables/válidas.

En la descripción de la porción de proceso de la Figura 7g, de nuevo, se diferencian tres casos, en concreto

- a) el contenido de AU re-ensamblado desde la sección de datos útiles 24 para la unidad de acceso actual coincide con la CRC asociada,
- b) el contenido de AU re-ensamblado desde la sección de datos útiles 24 para la unidad de acceso actual y la CRC asociada no coinciden,
- c) todas las AU se han procesado o todos los datos en la sección de datos útiles 24 se han procesado, que es la condición de salida para la sección de proceso de la Figura 7g.

En la Figura 7h, se muestra la porción de proceso introducida por el recuperador 54 para leer los bytes de una unidad de acceso actual en la etapa 242. Cuando se entra en esta porción de proceso, el recuperador 54 puede aprovechar la siguiente información como se indica en 254, en concreto la información de AUE, información de CAUB y RAUB. En primer lugar, el recuperador 54 extrae la longitud de la unidad de acceso actual en la etapa 256 desde la entrada de tabla de unidad de acceso asociada. En la etapa 258 el recuperador 54 comprueba en cuanto a si la AUL, es decir la longitud de la unidad de acceso actual, es mayor que RAUB, es decir el número de bytes restantes en la sección de datos útiles 24. Si la respuesta a la pregunta 258 es sí, el recuperador 54 establece en 260 parámetros en información de CAUB en consecuencia. En particular, en la etapa 260, el recuperador 54 establece CAUF para indicar que hay, de nuevo, una CAU, es decir una unidad de acceso que se extiende en la siguiente trama lógica. LPCAUB indica el número de bytes de la CAU que ya se han recuperado desde la trama lógica actual, es decir la longitud de PCAUB. CAUL es la longitud 62 de la CAU y CAUC es la CRC de la CAU.

Sin embargo, si la unidad de acceso actual se ajusta en la porción restante de la sección de datos útiles 24, el recuperador 54 lee en la etapa 262 los bytes de la unidad de acceso actual hasta el fin de la unidad de acceso actual como se indica mediante su longitud 62, es decir AUL. Posteriormente, el recuperador 54 actualiza los bytes de AU.

Por lo tanto, se diferencian los siguientes casos en la Figura 7h:

- 5 a) todos los bytes que corresponden a la unidad de acceso actual podrían leerse desde la trama lógica actual
- b) no todos los bytes de la unidad de acceso actual podrían leerse desde la trama lógica actual, es decir la unidad de acceso actual continúa o se vuelve una CAU.

10 Con respecto a la Figura 7i, se muestra la porción de proceso realizada mediante el recuperador 54 para el caso que la bandera de FEC resulta que está activada en la comprobación 152. En otras palabras, la Figura 7i ejemplifica la decodificación con FEC en la etapa 154. En este caso, el recuperador 54 tiene acceso, como se indica en 266, a la bandera de SFF, los bytes de LF y R. En la etapa 268, el recuperador 54 comprueba en cuanto a si SFF está activada. Si es así, el recuperador 54 continúa en la etapa 270 con decodificación de SF con FEC, y de otra manera en la etapa 272 con decodificación de LF con FEC. La primera porción de proceso de la etapa 272 se muestra en la Figura 7j. Con el conocimiento acerca de la información de CAUB como se indica en 274, el recuperador 54 lee una única trama lógica en una memoria intermedia interna en la etapa 276 y somete la única trama lógica a una RS-FEC en la etapa 278. En lo sucesivo, esta trama lógica se somete a decodificación en unidades de acceso en la etapa 280 justo como se ha hecho en la etapa 162 como se describe con respecto a la Figura 7c. Posteriormente, las unidades de acceso decodificadas se almacenan en memoria intermedia en la etapa 282 y la información de CAUB se actualiza en 284.

20 Es decir, en caso de decodificación de LF con FEC, la trama lógica actual se pasa adicionalmente a través de una corrección/decodificación de errores hacia delante de RS antes de que comience la decodificación de trama lógica real en unidades de acceso.

25 Someter la trama lógica a través de RS-FEC en la etapa 278 se ejemplifica adicionalmente en la Figura 7k. En particular, basándose en el conocimiento acerca de los bytes de LF y R, como se indica en 286, el recuperador 54 establece en la etapa 288 el desplazamiento de encabezamiento de LF, el desplazamiento de datos de LFAU, el desplazamiento de RS, y los bits de paridad de  $RS = 16 + xR$  y rellena, en las etapas 290 y 292, la tabla de aplicación de RS 98 y la tabla de paridad de RS 102, respectivamente. Después de haber comprobado en cuanto a si R equivale a cero en la etapa 294, el recuperador 54 lleva a cabo corrección de errores de RS en la etapa 296, si este no es el caso y reduce R en la etapa 298 después de lo cual el proceso vuelve a la etapa 294. Si la etapa comprobada 294 revela que R equivale a cero, los bytes de aplicación de RS en la tabla de datos de aplicación 98 se leen en el sentido des-intercalado para producir los bytes de LF en la etapa 300. Si la FEC de RS tiene éxito al corregir la fila R entonces la FEC de RS puede configurarse para devolver el número de los símbolos corregidos y actualizar la fila con los bytes corregidos, y para devolver, si falla, menos uno y retener los símbolos de fila como están. Sin embargo, son posibles también otras implementaciones. Además, pueden usarse diferentes códigos de FEC.

40 La Figura 7i muestra los detalles de la decodificación de SF con FEC en la etapa 270 de la Figura 7i. Como puede observarse, teniendo conocimiento acerca de la información de CAUB, como se indica en 302, el recuperador 54 lee en la etapa 304 tres o cuatro, o - de acuerdo con una realización alternativa - cualquier otro número, de tramas lógicas consecutivas, es decir una súper trama, en una memoria intermedia interna y somete esta súper trama de SF en la etapa 306 a una FEC de RS, decodificando a continuación las tres/cuatro tramas lógicas en la SF en la etapa 308 en las AU como se describe con respecto a la Figura 7c y almacenando en memoria intermedia en la etapa 310 las unidades de acceso decodificadas. Finalmente, el recuperador 54 actualiza la información de CAUB en la etapa 312. Similar a la Figura 7k, la Figura 7m muestra el caso de someter la SF a una FEC de RS. Como puede observarse, habiendo accedido a las tres/cuatro o - más en general, n - tramas lógicas, es decir los bytes de las tramas lógicas, y el valor de R como se indica en 314, el recuperador 54 realiza los ajustes en la etapa 316, rellena la tabla de aplicación de RS y la tabla de paridad de RS en las etapas 318 y 320, respectivamente y comprueba en cuanto a si R equivale a cero en la etapa 322. Si no, el recuperador 54 lleva a cabo la corrección de error de RS en la etapa 324 y reduce R en la etapa 326 para volver a la etapa 322. Tan pronto como R equivale a cero, el recuperador 54 dispone los bytes de aplicación de RS desde la tabla de datos de aplicación 98 en el formato intercalado para obtener las n tramas lógicas en la etapa 328.

55 Finalmente, las Figuras 8 y 9 muestran una vista general del manejo de protección de transporte de FEC en el recuperador 54 de acuerdo con una cierta realización adicional. De acuerdo con esta realización, se construye una tabla de 100 filas y 36 columnas que corresponde a 3600 bytes. Realmente la LF contiene 3598 bytes, sin embargo está espaciada en el lado de emisor y receptor a 3600 bytes por simplicidad. Todos los bytes se rellenan en esta tabla, columna a columna de la siguiente manera (1,1),(2,1)...(100,1),(1,2),(2,2) ....(10,100). Proporcionando por lo tanto intercalación aparente. Dependiendo de las columnas de aplicación y columnas de FEC, se determina la capacidad de corrección de FEC de Reed Solomon y se usa para decodificar cada fila de la tabla. Puesto que no tenemos conocimiento de la localización de error en la tabla tenemos que emplear decodificación de error de RS. Una vez que la tabla se pasa a través del decodificador de FEC, la salida puede o puede no contener los bytes completamente libres de errores. En ambos casos se emplea decodificación normal en serie, que es que

inicialmente el primer byte del encabezamiento de LF puede decodificarse que, de acuerdo con una realización alternativa relacionada con la anterior - proporciona el tamaño del encabezamiento de LF, si se decodifica satisfactoriamente a continuación el encabezamiento de LF completo y se compara con su CRC. El encabezamiento de LF se considera - de acuerdo con la presente realización - como que comprende la información de AUT, y se analiza adicionalmente para AU individuales. El diagrama de flujo del decodificador completo está en la Figura 8 y 9.

Por lo tanto, la realización de Diveemo consigue las siguientes ventajas mediante los siguientes aspectos

1. Método y aparato para codificar, transmitir y decodificar señales de vídeo mediante el sistema de DRM

- fuera de señalización de banda
- configuración del acceso de datos en banda
- al menos un flujo de datos lógico
- audio / vídeo / datos (tal como Journaline(R) para subtítulos)
- por ejemplo 1x vídeo, 5x audio (diferentes idiomas)
- cualquier códec para audio y vídeo es codificable (compatibilidad hacia atrás para mejora futura)
- compatibilidad de DRM (formatear en el MSC, conforme a la norma SDC)
- uso de tasa de bits eficaz (por ejemplo sin empaquetado)
- flexibilidad configurable (velocidad de fotogramas, tasa de bits, ...)
- protección frente a fallos opcional (FEC), parametrizable de manera flexible, intercalación virtual, intercalador de 2 tiempos,
- sincronización ascendente rápida del receptor posible
- estructura de datos posibilita extensiones futuras
- robusto frente a errores de recepción

2. Método para codificación y señalización de la transmisión de señales de vídeo mediante el sistema de Radio Digital Mundial

- se transmiten AU de audio, vídeo y datos sin encabezamientos adicionales como un 'flujo de bits en serie'
- la definición de las AU y sus límites/longitudes se registra con los límites de las tramas de transmisión del sistema de difusión (DRM) para posibilitar un acceso rápido y una fácil (re-) sincronización ascendente
- el índice se transmite de manera redundante en una trama lógica (LF)
- en caso de un fallo en el encabezamiento, sin embargo, el decodificador puede extraer datos de AU mediante la cadena de la definición/descripción de entidad
- en caso de un fallo en las AU o en la definición/descripción de entidad, simplemente se pierden las AU individuales, no necesariamente todas las AU en la trama de transmisión de DRM actual
- uno o más flujos de datos lógicos (intercalados) se subdividen en paquetes de datos de diferente longitud
- posible sincronización rápida en datos de cabida útil
- se usa estructura de transmisión (puntero hacia delante)
- mecanismo para futuras extensiones proporcionado
- parámetro de FEC es adaptable a errores de transmisión
- protección de error en el índice y en los paquetes de datos posibilita la extracción de todos los datos de cabida útil sin errores

Resumiendo las realizaciones anteriores, es decir las realizaciones descritas en primer lugar con respecto a las Figuras 1 a 3 así como las siguientes realizaciones que incluyen las realizaciones de Diveemo, una ventaja de registrar la tabla de unidad de acceso con el extremo inicial/final de las tramas lógicas es que las tablas se localizan rápida y fiablemente mediante el decodificador y aunque su longitud depende del número de comienzos de unidad de acceso en las tramas lógicas, el comienzo de la incrustación de la unidad de acceso, es decir el comienzo de la sección de datos útiles puede mantenerse constante tal como, por ejemplo, colocando la tabla de unidad de acceso al final cuando se ve en la dirección de inserción de datos útiles. Los punteros apuntan a las posiciones de los comienzos de la unidad de acceso en un sentido absoluto de modo que los errores locales en la trama lógica no evitan que se analicen y usen otras unidades de acceso que están sin errores. Si una tabla de unidad de acceso se pierde, la primera unidad de acceso que continúa puede completarse, incluso cuando la tabla de unidad de acceso se situó en el extremo inicial, y si la primera unidad de acceso que continúa está corrupta, y, por lo tanto, la detección del final de esta unidad de acceso y el comienzo de la siguiente unidad de acceso no es detectable mediante análisis a pesar de una inserción de manera ininterrumpida, las siguientes unidades de acceso pueden usarse aunque - independientemente de la posición exacta de esta tabla de unidad de acceso - puesto que los punteros en la tabla de unidad de acceso se definen en un sentido absoluto, es decir desde punto de registro que es estático.

Adicionalmente, puesto que la tabla de unidad de acceso y el encabezamiento de trama lógica se sitúan registrados

al comienzo absoluto o al fin absoluto de las tramas lógicas o a un desplazamiento conocido a partir de las mismas, la tabla de unidad de acceso y/o el encabezamiento de trama lógica se localizan fácil y fiablemente por el lado de decodificación basándose en las tramas de transmisión. Incluso si ambos, el encabezamiento de la trama lógica así como la tabla de unidad de acceso se registraran con relación al mismo fin, tal como el extremo inicial o final de la trama lógica, la sección de datos útiles de la trama lógica podría localizarse si al menos uno de los encabezamientos de trama lógica y la tabla de unidad de acceso tuvieran una longitud constante como con las realizaciones anteriores. Sin embargo, colocando el encabezamiento de trama lógica y la tabla de unidad de acceso a lados opuestos, es posible un uso óptimo de las unidades de acceso: si un encabezamiento de trama lógica está corrupto, el comienzo de la sección de datos útiles, sin embargo, es localizable en el lado de decodificación, y la unidad de acceso actual puede procesarse. Si la tabla de unidad de acceso está corrupta, entonces el fin de la sección de datos útiles es conocido en el decodificador y posiblemente todas las unidades de acceso pueden reconstruirse mediante análisis.

La ventaja de presentar la información de longitud de unidad de acceso en la tabla de unidad de acceso es que la información en la misma puede usarse en el lado de decodificación para acceder a la siguiente unidad de acceso, es decir, la primera unidad de acceso el comienzo de la cual cae en la trama lógica actual incluso entonces si la unidad de acceso que continúa está corrupta y por lo tanto no es pasable. En otras palabras, en caso de inserción de manera ininterrumpida de las unidades de acceso, la indicación de longitud de unidad de acceso ayuda a acceder a una unidad de acceso incluso cuando el puntero que apunta al comienzo de esta unidad de acceso está corrupto. Si es necesario relleno, este relleno puede hacerse en una forma de tipo de unidad de acceso especialmente marcada, que está integrada de manera ininterrumpida en las otras unidades de acceso de los tipos de unidad de acceso normal, de modo que se mantiene la inserción de manera ininterrumpida.

La posibilidad anterior de protección de FEC de las entradas de tabla de unidad de acceso posibilita individualmente el procesamiento de las unidades de acceso individualmente incluso si algunas de las entradas de tabla de unidad de acceso están corruptas. Por lo tanto, las entradas corruptas en la tabla de unidad de acceso pueden saltarse debido a su longitud constante y las posteriores entradas en la tabla de unidad de acceso pueden evaluarse sin problema por el recuperador. Esto es una ventaja de la protección de CRC individual. Además, incluso si el número de, o incluso la existencia de, entradas de AUT es desconocido, pueden rastrearse entradas válidas.

Pueden realizarse muchas modificaciones en las realizaciones anteriores. Por ejemplo, son fácilmente derivables muchas alternativas para los detalles específicos en las realizaciones de Diveemo partir de las declaraciones con respecto a las Figuras 1 a 3. Con respecto a la FEC de RS se indica, que pueden usarse otros códigos de FEC también. Además, en lugar de CRC en las LF, los datos de FEC pueden incorporarse en la LF.

Aunque se han descrito algunos aspectos en el contexto de un aparato, es evidente que estos aspectos también representan una descripción del método correspondiente, donde un bloque o dispositivo corresponde a una etapa de método o a una característica de una etapa de método. De manera análoga, los aspectos descritos en el contexto de una etapa de método también representan una descripción de un bloque o elemento o característica correspondiente de un aparato correspondiente. Alguna o todas las etapas del método pueden ejecutarse mediante (o usando) un aparato de hardware, como por ejemplo, un microprocesador, un ordenador programable o un circuito electrónico. En algunas realizaciones, alguna o más de las etapas de método más importantes pueden ejecutarse mediante un aparato de este tipo.

La señal de transmisión resultante anterior podría almacenarse en un medio de almacenamiento digital o puede transmitirse en un medio de transmisión tal como un medio de transmisión inalámbrica o un medio de transmisión alámbrica tal como internet.

Dependiendo de ciertos requisitos de implementación, las realizaciones de la invención pueden implementarse en hardware o en software. Las implementaciones pueden realizarse usando un medio de almacenamiento digital, por ejemplo un disco flexible, un DVD, un Blue-Ray, un CD, una ROM, una PROM, una EPROM, una EEPROM o una memoria FLASH, que tiene señales de control electrónicamente legibles almacenadas en la misma, que cooperan (o son capaces de cooperar) con un sistema informático programable de manera que se realiza el respectivo método. Por lo tanto, el medio de almacenamiento digital puede ser legible por ordenador.

Algunas realizaciones de acuerdo con la invención comprenden un soporte de datos que tiene señales de control electrónicamente legibles, que pueden cooperar con un sistema informático programable, de manera que se realiza uno de los métodos descritos en el presente documento.

En general, las realizaciones de la presente invención pueden implementarse como un producto de programa informático con un código de programa, el código de programa siendo operativo para realizar uno de los métodos cuando el producto de programa informático se ejecuta en un ordenador. El código de programa puede almacenarse por ejemplo en un soporte legible por máquina.

Otras realizaciones comprenden el programa informático para realizar uno de los métodos descritos en el presente

documento, almacenado en un soporte legible por máquina.

5 En otras palabras, una realización del método inventivo es, por lo tanto, un programa informático que tiene un código de programa para realizar uno de los métodos descritos en el presente documento, cuando el programa informático se ejecuta en un ordenador.

10 Una realización adicional del método inventivo es, por lo tanto, un soporte de datos (o un medio de almacenamiento digital, o un medio legible por ordenador) que comprende, grabado en el mismo, el programa informático para realizar uno de los métodos descritos en el presente documento.

15 Una realización adicional del método inventivo es, por lo tanto, un flujo de datos o una secuencia de señales que representan el programa informático para realizar uno de los métodos descritos en el presente documento. El flujo de datos o la secuencia de señales pueden configurarse, por ejemplo, para transferirse mediante una conexión de comunicación de datos, por ejemplo mediante internet.

20 Una realización adicional comprende un medio de procesamiento, por ejemplo un ordenador, o un dispositivo de lógica programable, configurado para o adaptado para realizar uno de los métodos descritos en el presente documento.

25 Una realización adicional comprende un ordenador que tiene instalado en el mismo el programa informático para realizar uno de los métodos descritos en el presente documento.

30 En algunas realizaciones, un dispositivo de lógica programable (por ejemplo un campo de matriz de puertas programables) puede usarse para realizar algunas o todas las funcionalidades de los métodos descritos en el presente documento. En algunas realizaciones, un campo de matriz de puertas programables puede cooperar con un microprocesador para para realizar uno de los métodos descritos en el presente documento. En general, los métodos se realizan preferentemente mediante cualquier aparato de hardware.

Las realizaciones anteriormente descritas son meramente ilustrativas para los principios de la presente invención. Se entiende que serán evidentes modificaciones y variaciones de las disposiciones y los detalles descritos en el presente documento para los expertos en la materia. Se pretende, por lo tanto, que estén limitadas únicamente por el alcance de las reivindicaciones de patente siguientes y no por los detalles específicos presentados por medio de descripción y explicación de las realizaciones en el presente documento.

## REIVINDICACIONES

1. Aparato para recuperar un flujo de unidad de acceso de unidades de acceso consecutivas, que representan contenido de vídeo y/o de audio, desde una secuencia de tramas lógicas de una señal de transmisión, comprendiendo cada trama lógica una sección de datos útiles, en el que las unidades de acceso consecutivas se insertan de manera consecutiva en la sección de datos útiles de la secuencia de las tramas lógicas, en el que las tramas lógicas comprenden tramas lógicas en las que no cae comienzo de una unidad de acceso (28), cae un comienzo de una unidad de acceso, y cae un comienzo de dos unidades de acceso, respectivamente, estando configurado el aparato para, para una trama lógica predeterminada, en la que cae un comienzo de una unidad de acceso, extraer una tabla de unidad de acceso desde la trama lógica predeterminada que comprende, por comienzo de las unidades de acceso que caen en la trama lógica predeterminada, un puntero que apunta a una posición de comienzo respectiva en la trama lógica predeterminada, y localizar, y comenzar la extracción de, la respectiva unidad de acceso el comienzo de la cual cae en la trama lógica predeterminada, mediante el uso del respectivo puntero, estando el aparato configurado también para extraer de manera consecutiva las unidades de acceso consecutivas del flujo de unidad de acceso desde la sección de datos útiles de las tramas lógicas de la secuencia de las tramas lógicas, en el que el aparato está configurado para extraer desde cada trama lógica (20) un encabezamiento de trama lógica (22) que indica la ausencia de, o la longitud (34) de una tabla de unidad de acceso (30) en la respectiva trama lógica (20), y localizar una extensión de la sección de datos útiles (24) de la respectiva trama lógica (20) que depende de la misma,
- en el que el aparato está configurado para realizar la extracción consecutiva de las unidades de acceso consecutivas (28) desde las tramas lógicas (20) usando una dirección de extracción de datos útiles en las tramas lógicas (20), y para localizar un extremo situado de manera variable de la tabla de unidad de acceso (30) aplicando la longitud (34) de la tabla de unidad de acceso (30) en una dirección en contra contraria a la dirección de extracción de datos útiles desde un extremo situado de manera constante de la misma, y
- en el que el encabezamiento de trama lógica (32) indica una longitud de la tabla de unidad de acceso (30) de las respectivas tramas lógicas (20) y el aparato está configurado para, al extraer la tabla de unidad de acceso (30) desde la trama lógica predeterminada (20), extraer de manera consecutiva un número de entradas de tabla de unidad de acceso (64), iniciando gradualmente en el extremo inicial o final de la trama lógica (20) o en una posición en la trama lógica predeterminada (20), que tiene un desplazamiento constante predeterminado desde la misma, en unidades de una longitud constante desde una entrada de tabla de unidad de acceso (64) a la siguiente, para obtener, para cada entrada de tabla de unidad de acceso (64), un puntero (40) que apunta a una respectiva posición de comienzo de la respectiva unidad de acceso (28),
- en el que el aparato está configurado para extraer información de redundancia adicional para el encabezamiento de trama lógica desde la trama lógica predeterminada, y usar la misma para realizar detección de corrupción de datos (172) o intentos de corrección de errores en el encabezamiento de trama lógica, y, si el encabezamiento de trama lógica resulta estar corrupto, etapa (178), empezar en el extremo inicial o final de la trama lógica (20) o en la posición en la trama lógica predeterminada (20), que tiene el desplazamiento constante predeterminado desde la misma, en unidades de la longitud constante desde una entrada de tabla de unidad de acceso (64) posiblemente presente a la siguiente, para
- obtener, para cada entrada de tabla de unidad de acceso (64) posiblemente presente, un supuesto puntero (40) que apunta a una supuesta posición de comienzo de una respectiva unidad de acceso (28) y datos de redundancia adicionales,
- realizar una detección de corrupción de datos individual de entrada de tabla de unidad de acceso en las entradas de tabla de unidad de acceso posiblemente presentes, y
- localizar, y comenzar la extracción de, simplemente en unidades de acceso en supuestas posiciones de comienzo a las que apuntan los punteros de aquellas entradas de tabla de unidad de acceso posiblemente presentes que resultan ser correctas por la detección de corrupción de datos.
2. Aparato de acuerdo con la reivindicación 1, en el que el aparato está configurado para localizar, en cada trama lógica (20), el encabezamiento de trama lógica (22) en, o a un desplazamiento constante predeterminado desde, el extremo inicial o final de la respectiva trama lógica (20).
3. Aparato de acuerdo con la reivindicación 1 o 2, en el que el aparato está configurado para extraer información de redundancia adicional para el encabezamiento de trama lógica desde la trama lógica predeterminada, y usar la misma para realizar detección de corrupción de datos (172) o intentos de corrección de errores en el encabezamiento de trama lógica, y, si el último resulta estar corrupto, extraer (176, 190), si está presente, una porción restante de la sección de datos útiles de la trama lógica predeterminada que pertenece a una unidad de acceso el comienzo de la cual cualquier trama lógica anterior y se extiende en la trama lógica predeterminada, empezando la porción restante en una posición de inicio de la sección de datos útiles de la trama lógica predeterminada situada de manera constante para todas las tramas lógicas.
4. Aparato de acuerdo con cualquiera de las reivindicaciones 1 a 3, en el que el aparato está configurado para, al extraer la tabla de unidad de acceso desde la trama lógica predeterminada, extraer desde cada entrada de tabla de

unidad de acceso el puntero, y segundos datos de redundancia adicionales, en el que el aparato está configurado para usar los segundos de redundancia adicionales para realizar detección de corrupción de datos con respecto a un contenido de la unidad de acceso al comienzo del cual apunta el puntero de la respectiva entrada de tabla de unidad de acceso (64).

5  
5. Aparato de acuerdo con cualquiera de las reivindicaciones 1 a 4, en el que el aparato está configurado para continuar la extracción consecutiva de las unidades de acceso consecutivas desde las tramas lógicas cuando se pasa desde la trama lógica predeterminada a una siguiente trama lógica, comenzando la extracción en una posición en la siguiente trama lógica, independiente de un número de unidades de acceso el comienzo de las cuales cae en la trama lógica predeterminada.

15  
6. Aparato de acuerdo con la reivindicación 2 a 5, en el que el flujo de unidad de acceso comprende al menos dos flujos de entrada separados de unidades de acceso consecutivas, cada uno de los cuales representa contenido de vídeo o de audio, en el que, para cada flujo de entrada, las unidades de acceso del respectivo flujo de entrada están dispuestas en el flujo de unidad de acceso en un orden de presentación de los al menos dos flujos de entrada, en el que la tabla de unidad de acceso comprende una entrada de tabla de unidad de acceso por unidad de acceso el comienzo de la cual cae en la respectiva trama lógica, y con la que está asociada la respectiva entrada de tabla de unidad de acceso, y cada entrada de tabla de unidad de acceso comprende un puntero que apunta a una posición de su unidad de acceso asociada en la respectiva trama lógica, y un ID de flujo que indica a qué flujo de entrada pertenece su unidad de acceso asociada, en el que el aparato está configurado para asignar las unidades de acceso extraídas desde la trama lógica predeterminada al correspondiente flujo de entrada mediante el uso del ID de flujo de la entrada de unidad de acceso asociada.

25  
7. Aparato de acuerdo con cualquiera de las reivindicaciones 1 a 6, en el que los encabezamientos de tramas lógicas rodean, o tienen un desplazamiento predeterminado desde, uno de un extremo inicial o un extremo final de las respectivas tramas lógicas, y las tramas lógicas cada una comprende adicionalmente una sección de datos de corrección de errores hacia delante (66) de longitud constante y situada adyacente al respectivo encabezamiento de trama lógica, definiendo las secciones de datos de corrección de errores hacia delante datos de corrección de errores hacia delante sistemáticos y porciones de protección de corrección de errores hacia delante de las tramas lógicas externas a las secciones de datos de corrección de errores hacia delante y que comprende, al menos, la sección de datos útiles, el encabezamiento de trama lógica y la tabla de unidad de acceso, si está presente, de las mismas, en el que el aparato está configurado para realizar procesamiento de corrección de errores hacia delante usando los datos de corrección de errores hacia delante en las secciones de datos de corrección de errores hacia delante en las porciones protegidas de corrección de errores hacia delante de las tramas lógicas.

35  
8. Aparato de acuerdo con cualquiera de las reivindicaciones 1 a 7, en el que el aparato está configurado adicionalmente para inspeccionar, en cada encabezamiento de trama lógica, una bandera en cuanto a si existe o no una sección de mejora en la respectiva trama lógica, y usar esta información para localizar una extensión de la sección de datos útiles en las tramas lógicas.

40  
9. Método para recuperar un flujo de unidad de acceso de unidades de acceso consecutivas, que representan contenido de vídeo y/o de audio, desde una secuencia de tramas lógicas de una señal de transmisión, comprendiendo cada trama lógica una sección de datos útiles, en el que las unidades de acceso consecutivas se insertan de manera consecutiva en la sección de datos útiles de la secuencia de las tramas lógicas, en el que las tramas lógicas comprenden tramas lógicas en las que no cae comienzo de una unidad de acceso (28), cae un comienzo de una unidad de acceso, y cae un comienzo de dos unidades de acceso, respectivamente, comprendiendo el método, para una trama lógica predeterminada, en la que cae un comienzo de una unidad de acceso, extraer una tabla de unidad de acceso desde la trama lógica predeterminada que comprende, por comienzo de las unidades de acceso que caen en la trama lógica predeterminada, un puntero que apunta a una respectiva posición de comienzo en la trama lógica predeterminada, y localizar, y comenzar la extracción de, la respectiva unidad de acceso el comienzo de la cual cae en la trama lógica predeterminada, mediante el uso del respectivo puntero, comprendiendo el método también extraer de manera consecutiva las unidades de acceso consecutivas del flujo de unidad de acceso desde la sección de datos útiles de las tramas lógicas de la secuencia de las tramas lógicas,

55  
en el que el método comprende extraer desde cada trama lógica (20) un encabezamiento de trama lógica (22) que indica la ausencia de, o la longitud (34) de una tabla de unidad de acceso (30) en la respectiva trama lógica (20), y localizar una extensión de la sección de datos útiles (24) de la respectiva trama lógica (20) que depende de la misma, realizar la extracción consecutiva de las unidades de acceso consecutivas (28) desde las tramas lógicas (20) usando una dirección de extracción de datos útiles en las tramas lógicas (20), y localizar un extremo situado de manera variable de la tabla de unidad de acceso (30) aplicando la longitud (34) de la tabla de unidad de acceso (30) en una dirección en contra contraria a la dirección de extracción de datos útiles desde un extremo situado de manera constante de la misma, y en el que el encabezamiento de trama lógica (32) indica una longitud de la tabla de unidad de acceso (30) de las respectivas tramas lógicas (20) y la extracción de la tabla de unidad de acceso (30) desde la

- trama lógica predeterminada (20) comprende extraer de manera consecutiva un número de entradas de tabla de unidad de acceso (64), iniciando gradualmente en el extremo inicial o final de la trama lógica (20) o en una posición en la trama lógica predeterminada (20), que tiene un desplazamiento constante predeterminado desde la misma, en unidades de una longitud constante desde una entrada de tabla de unidad de acceso (64) a la siguiente, para
- 5 obtener, para cada entrada de tabla de unidad de acceso (64), un puntero (40) que apunta a una respectiva posición de comienzo de la respectiva unidad de acceso (28),
- en el que el método comprende adicionalmente extraer información de redundancia adicional para el encabezamiento de trama lógica desde la trama lógica predeterminada, y usar la misma para realizar detección de corrupción de datos (172) o intentos de corrección de errores en el encabezamiento de trama lógica, y, si el
- 10 encabezamiento de trama lógica resulta corrupto,
- iniciar gradualmente (178), en el extremo inicial o final de la trama lógica (20) o en la posición en la trama lógica predeterminada (20), que tiene el desplazamiento constante predeterminado desde la misma, en unidades de la longitud constante desde una entrada de tabla de unidad de acceso (64) posiblemente presente a la siguiente, para
- 15 obtener, para cada entrada de tabla de unidad de acceso (64) posiblemente presente, un supuesto puntero (40) que apunta a una supuesta posición de comienzo de una respectiva unidad de acceso (28) y datos de redundancia adicionales,
- realizar una detección de corrupción de datos individual de entrada de tabla de unidad de acceso en las entradas de tabla de unidad de acceso posiblemente presentes, y
- 20 localizar, y comenzar la extracción de, simplemente en unidades de acceso en supuestas posiciones de comienzo a las que apuntan los punteros de aquellas entradas de tabla de unidad de acceso posiblemente presentes que resultan ser correctas por la detección de corrupción de datos.
10. Un programa informático que tiene un código de programa para realizar, cuando se ejecuta en un ordenador, un
- 25 método de acuerdo con la reivindicación 9.

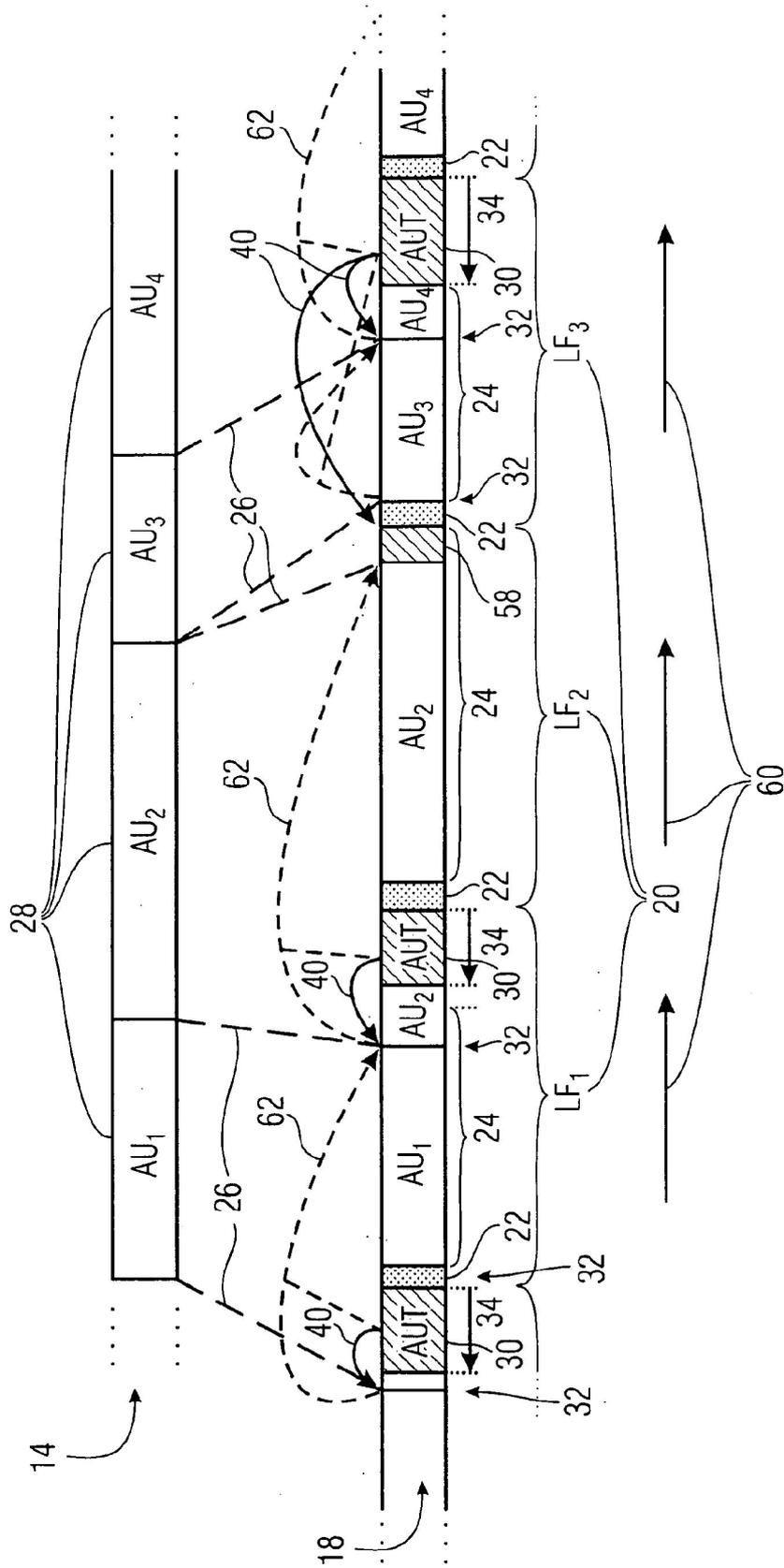


FIG 1

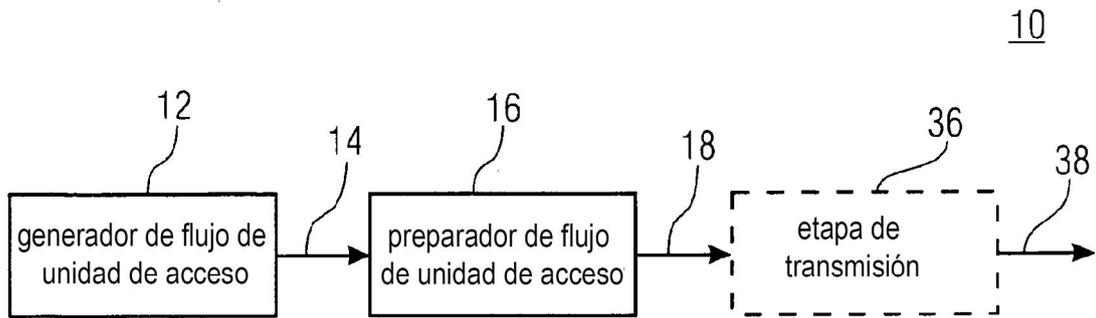


FIG 2

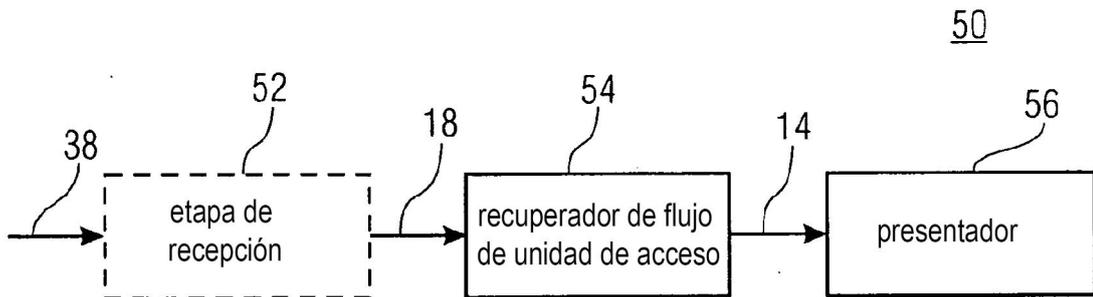


FIG 3

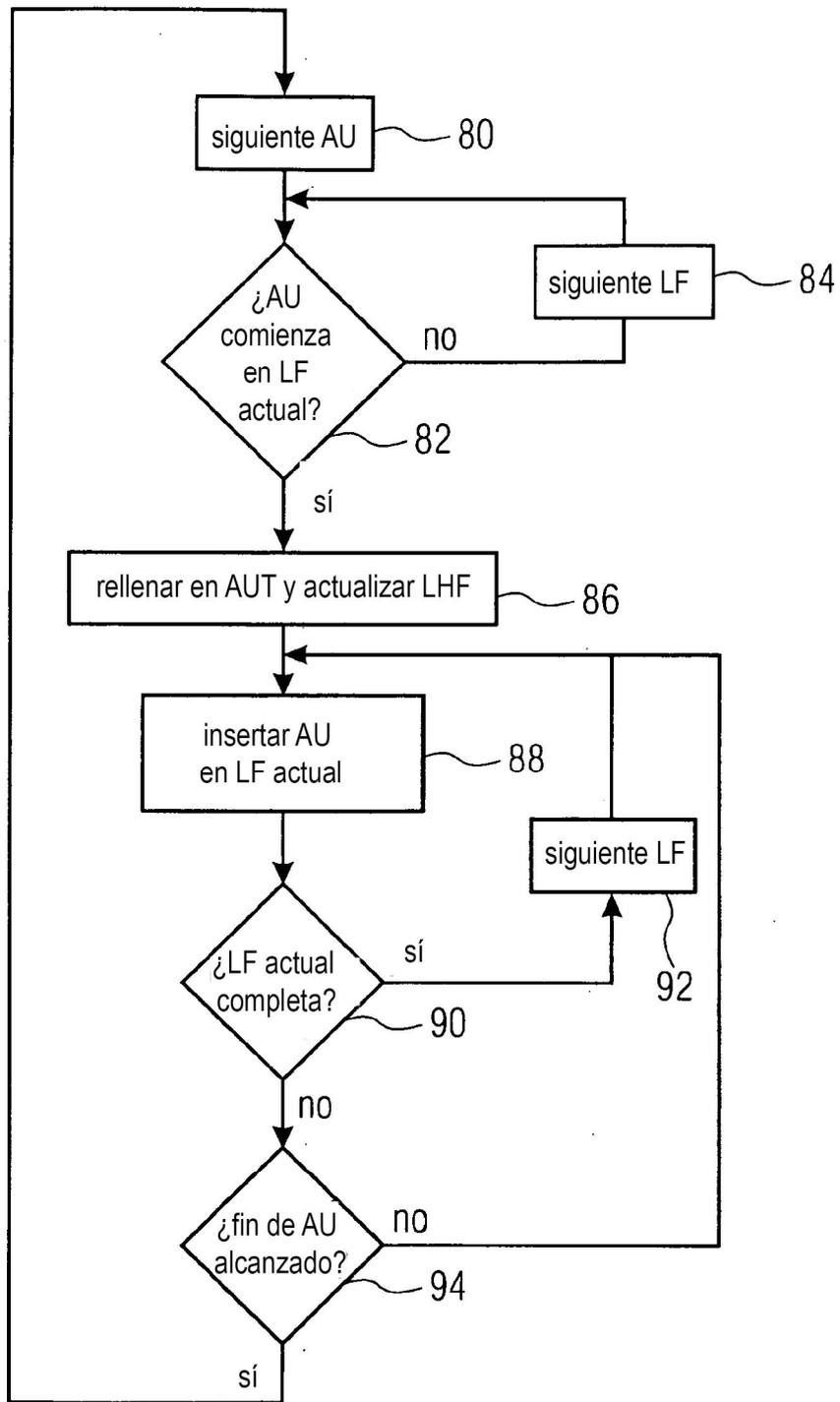


FIG 4

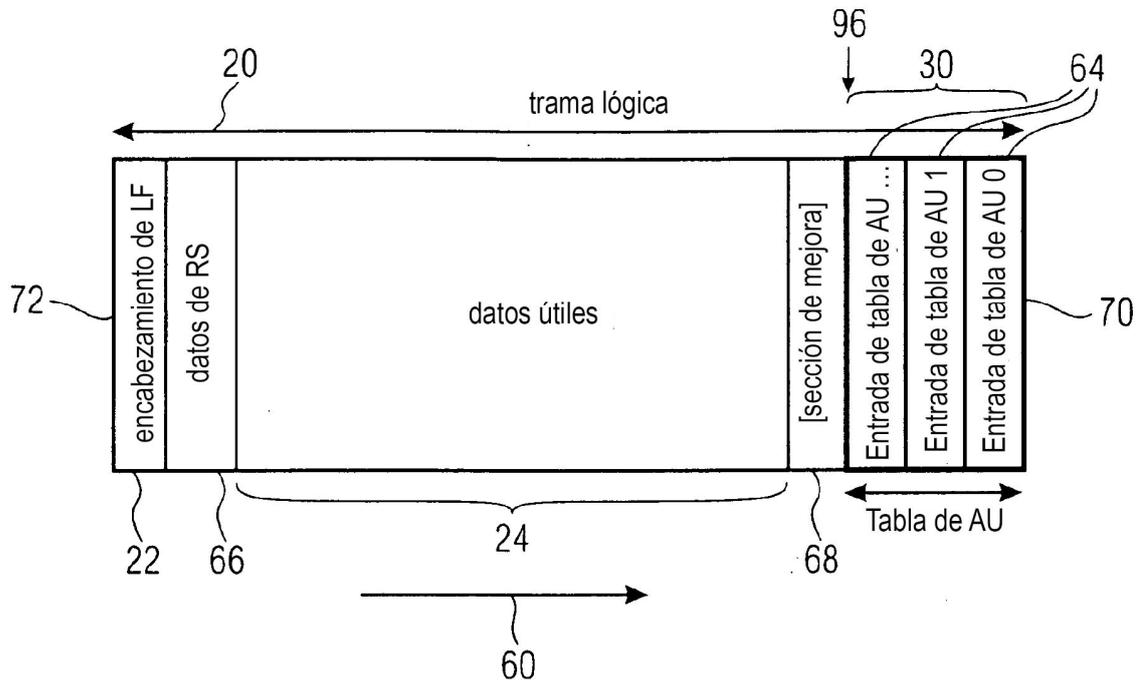


FIG 5

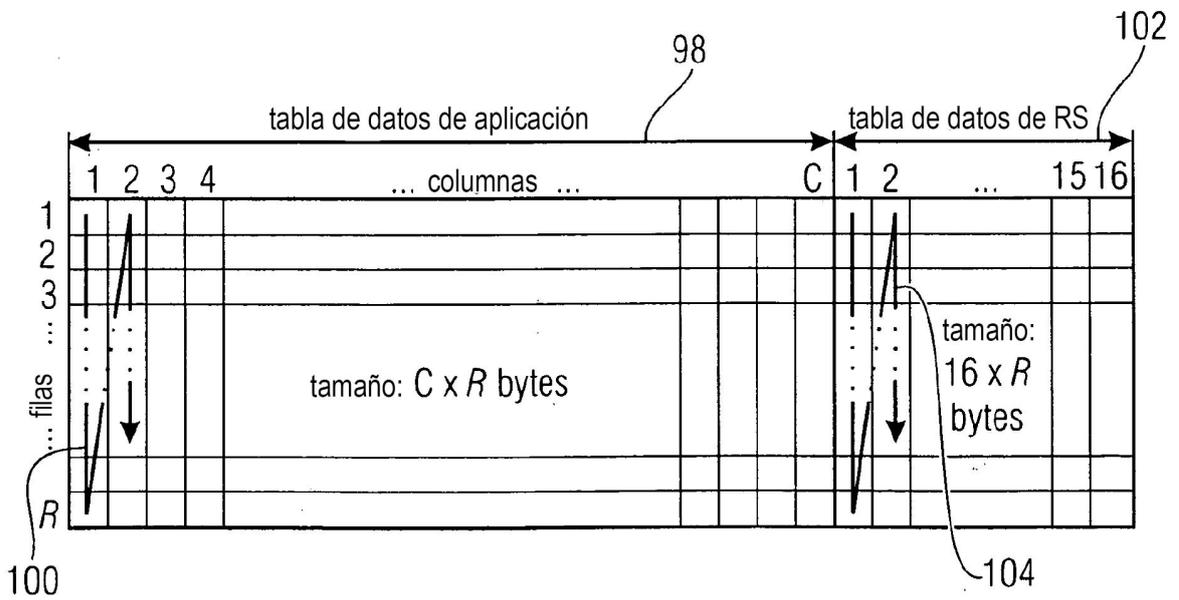


FIG 6

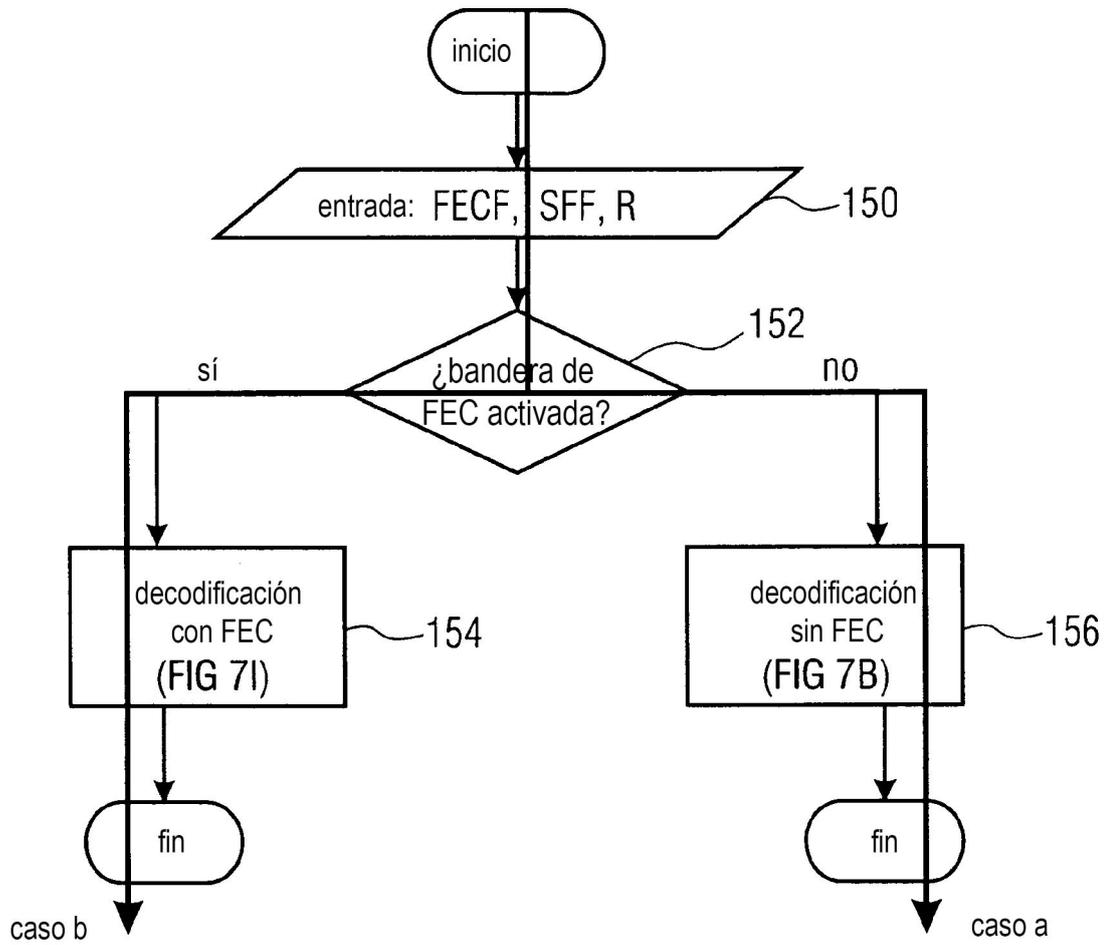


FIG 7A

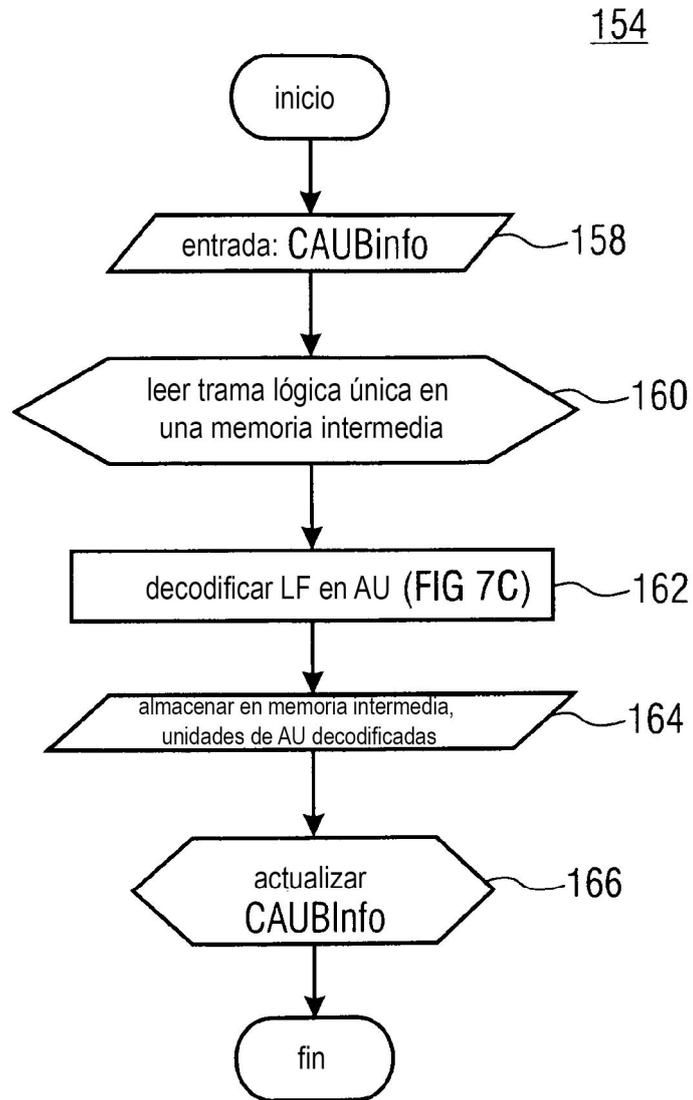


FIG 7B

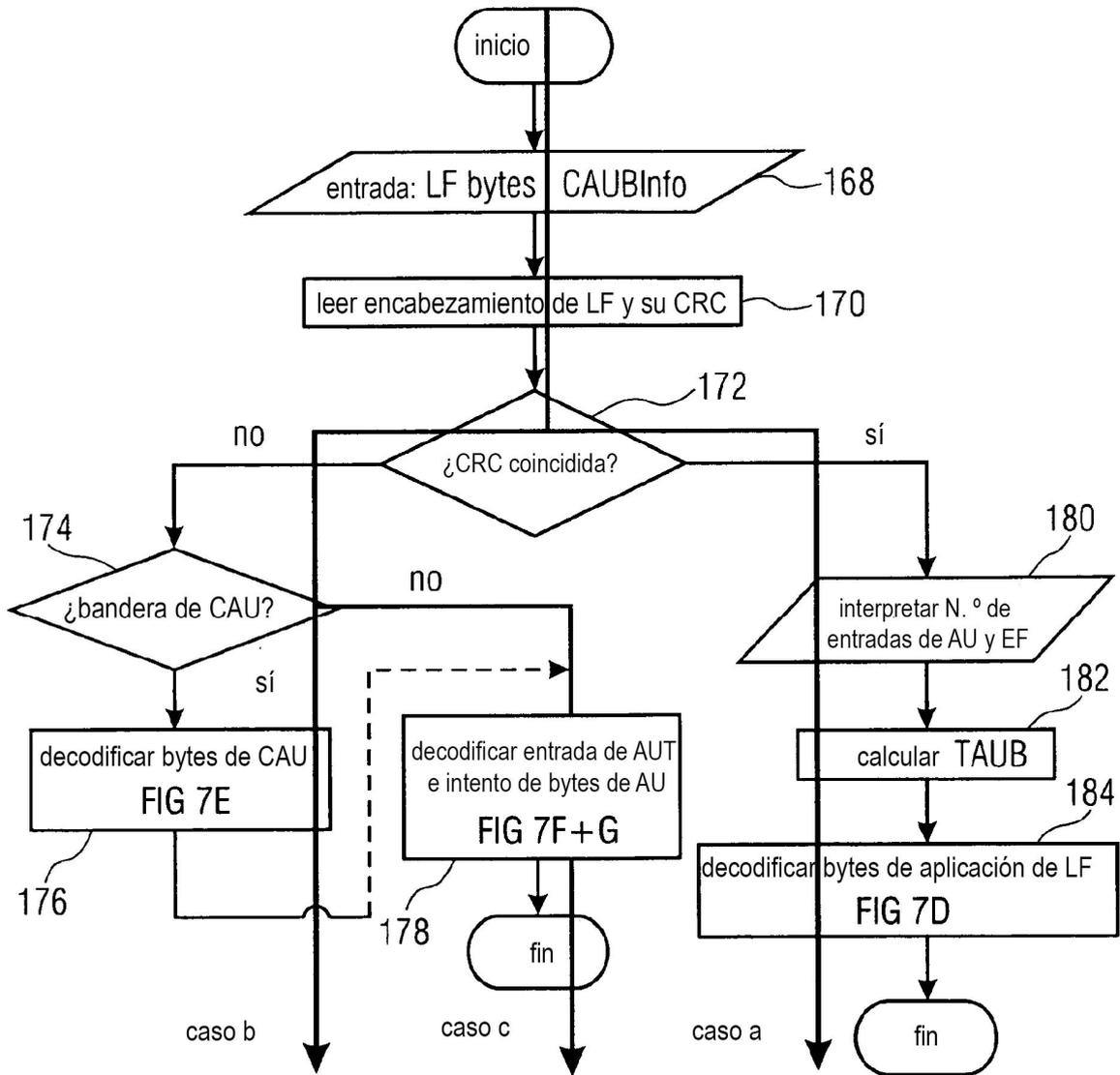


FIG 7C

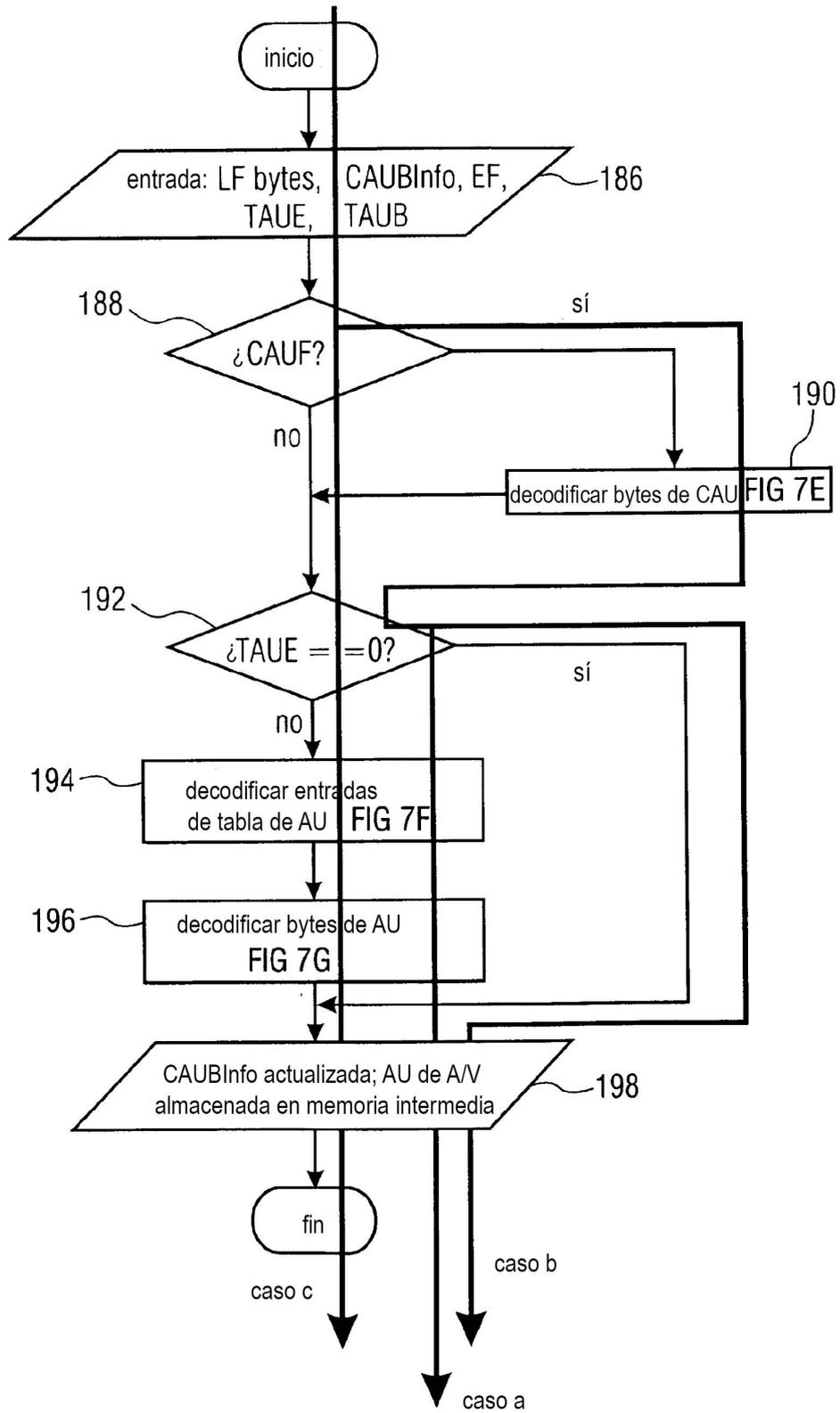


FIG 7D

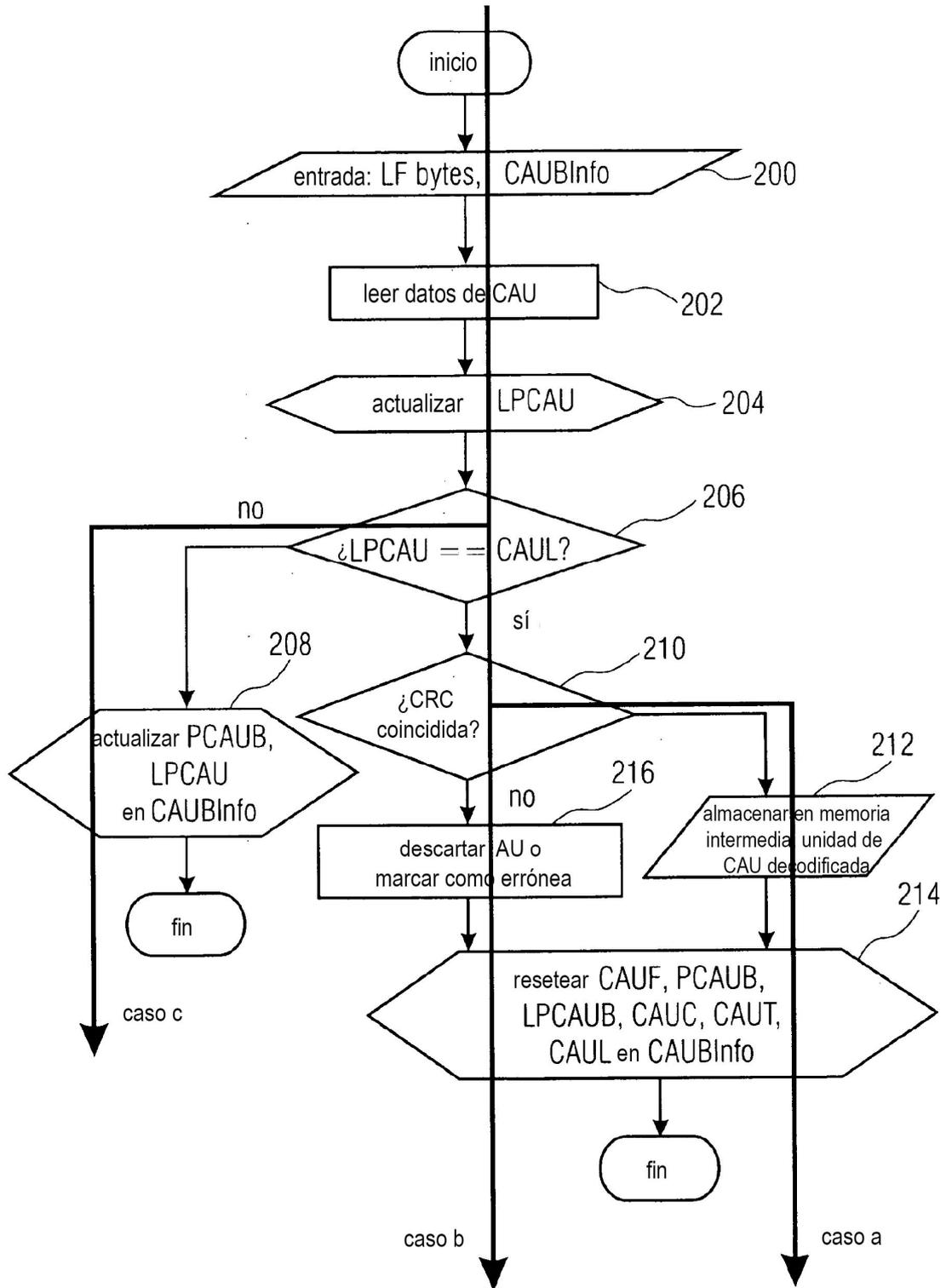


FIG 7E

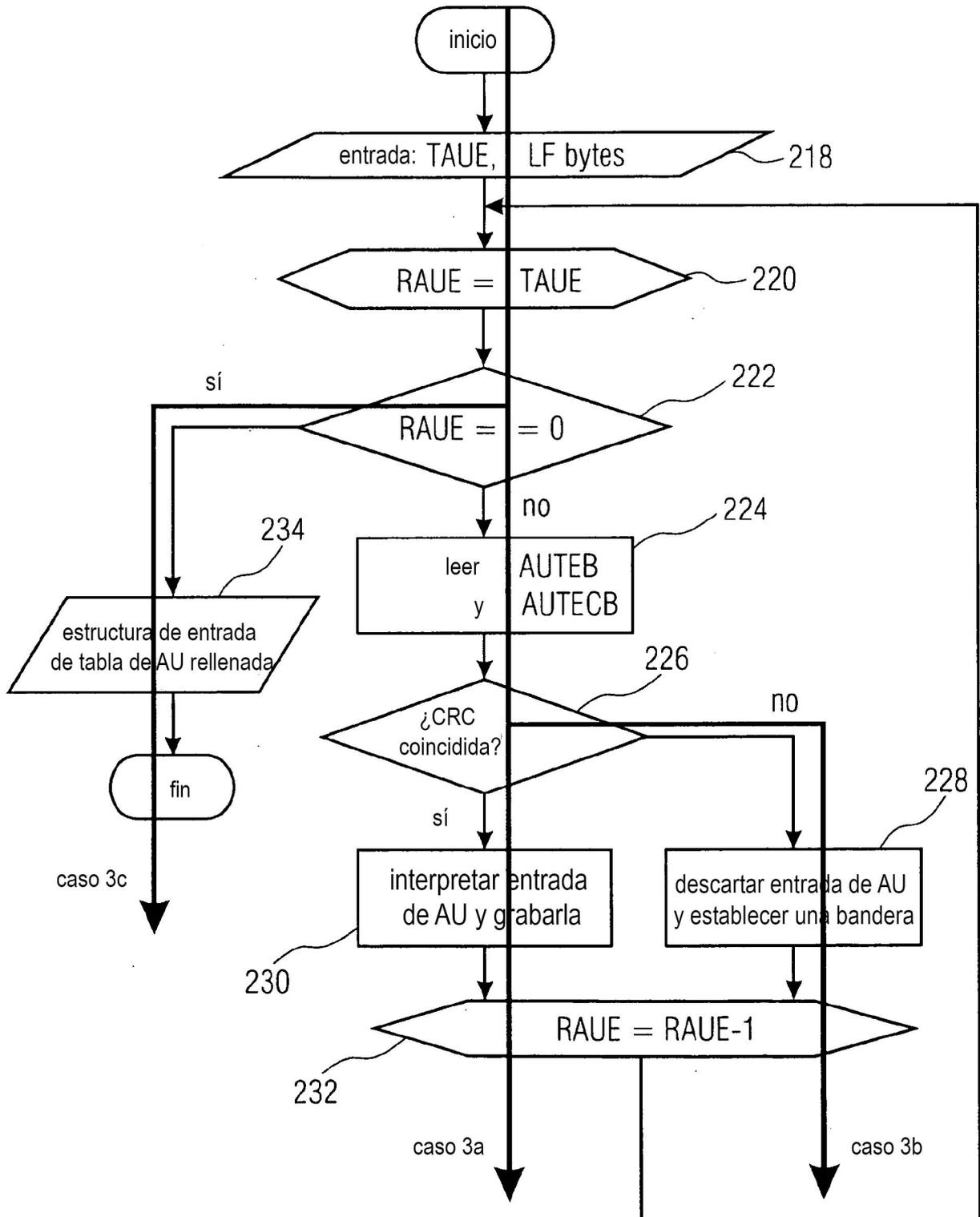


FIG 7F

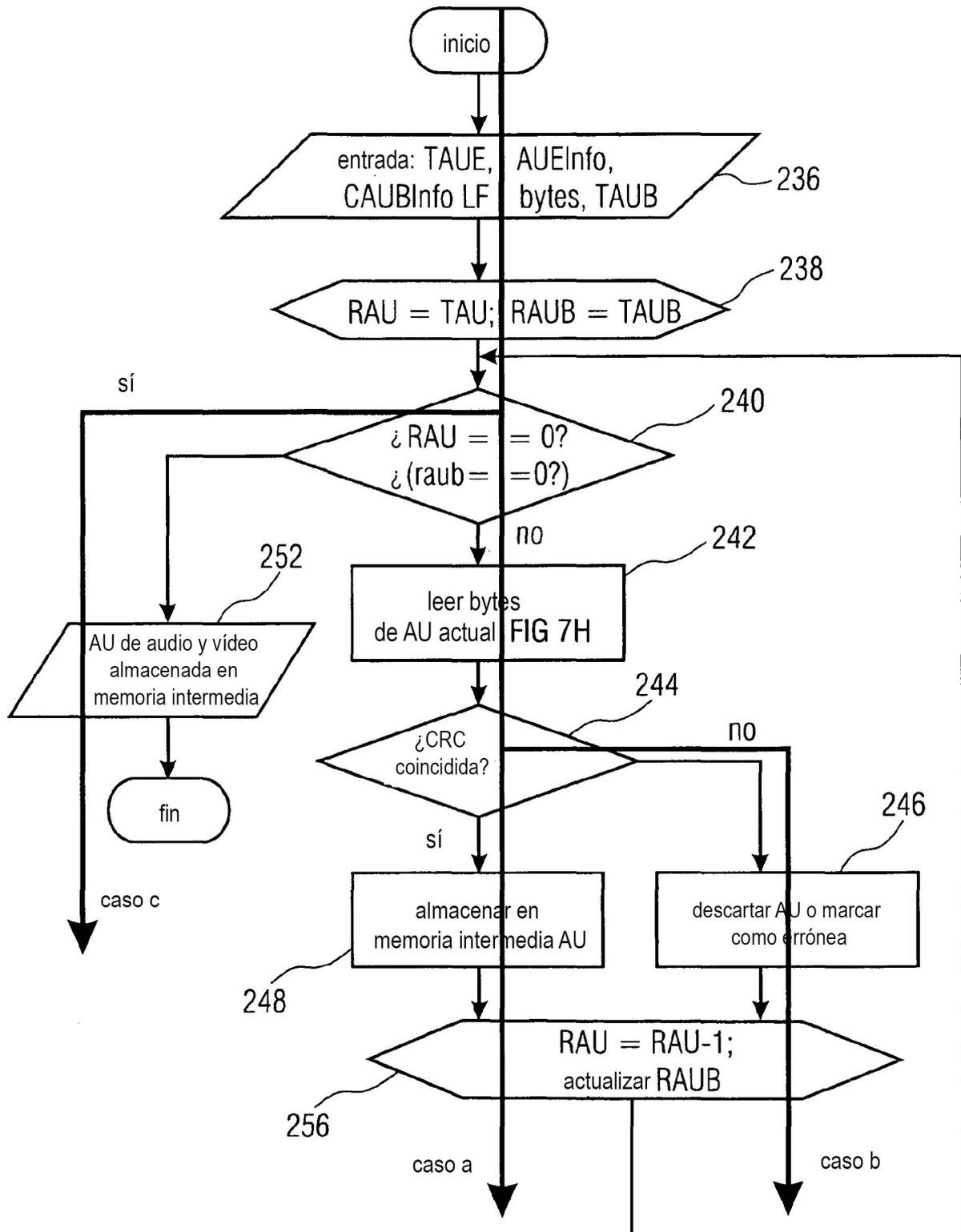


FIG 7G

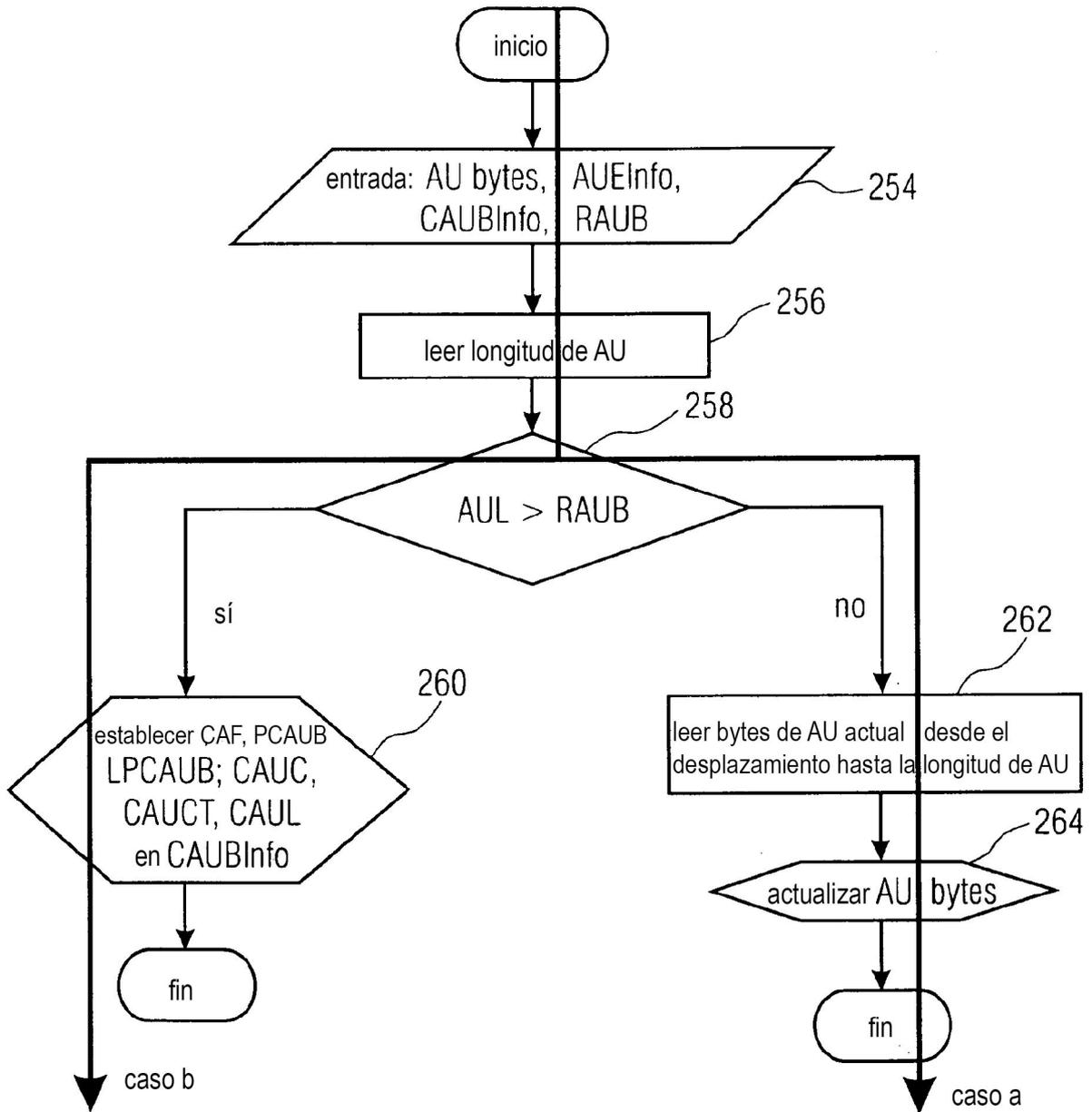


FIG 7H

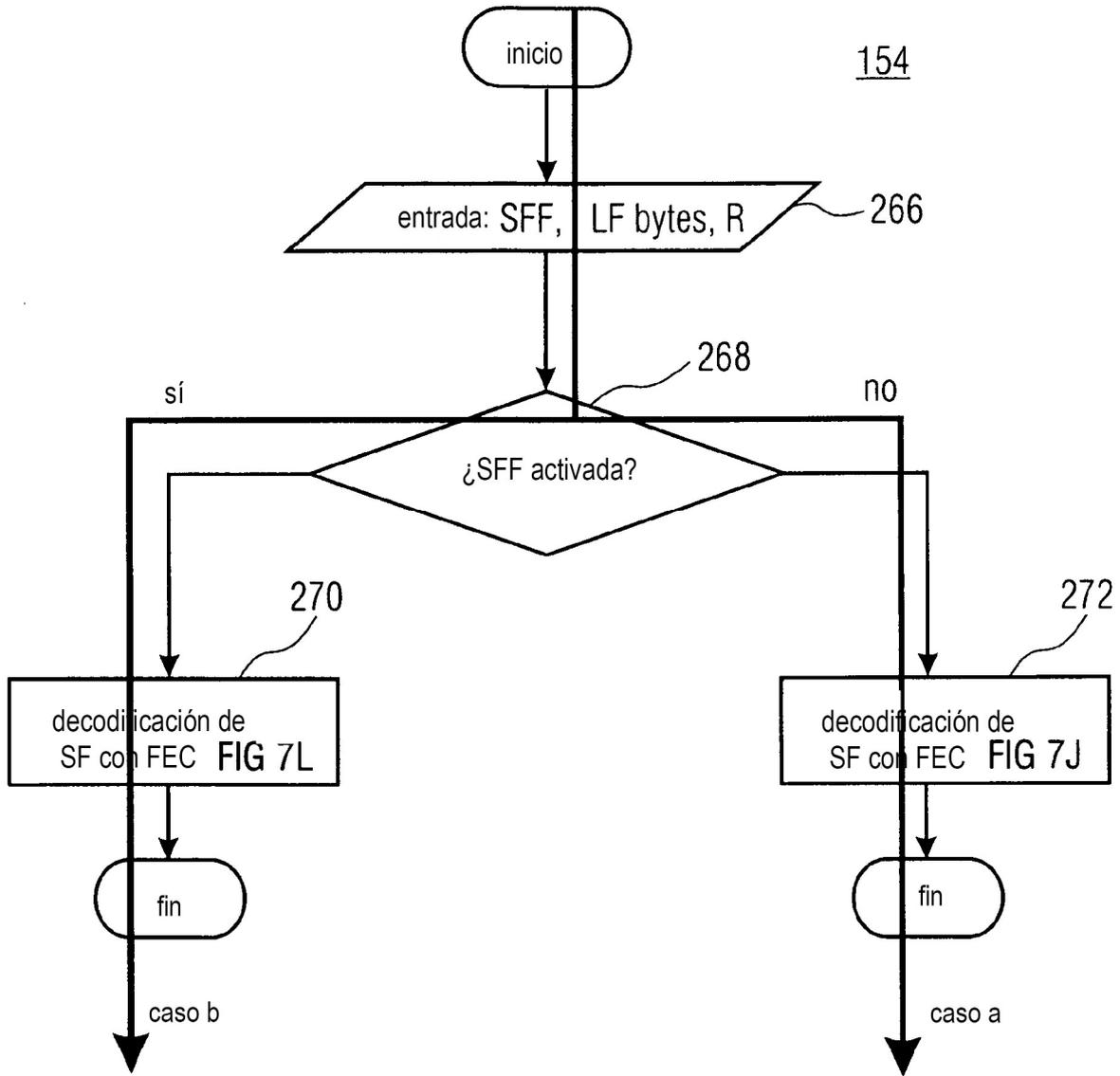


FIG 7I

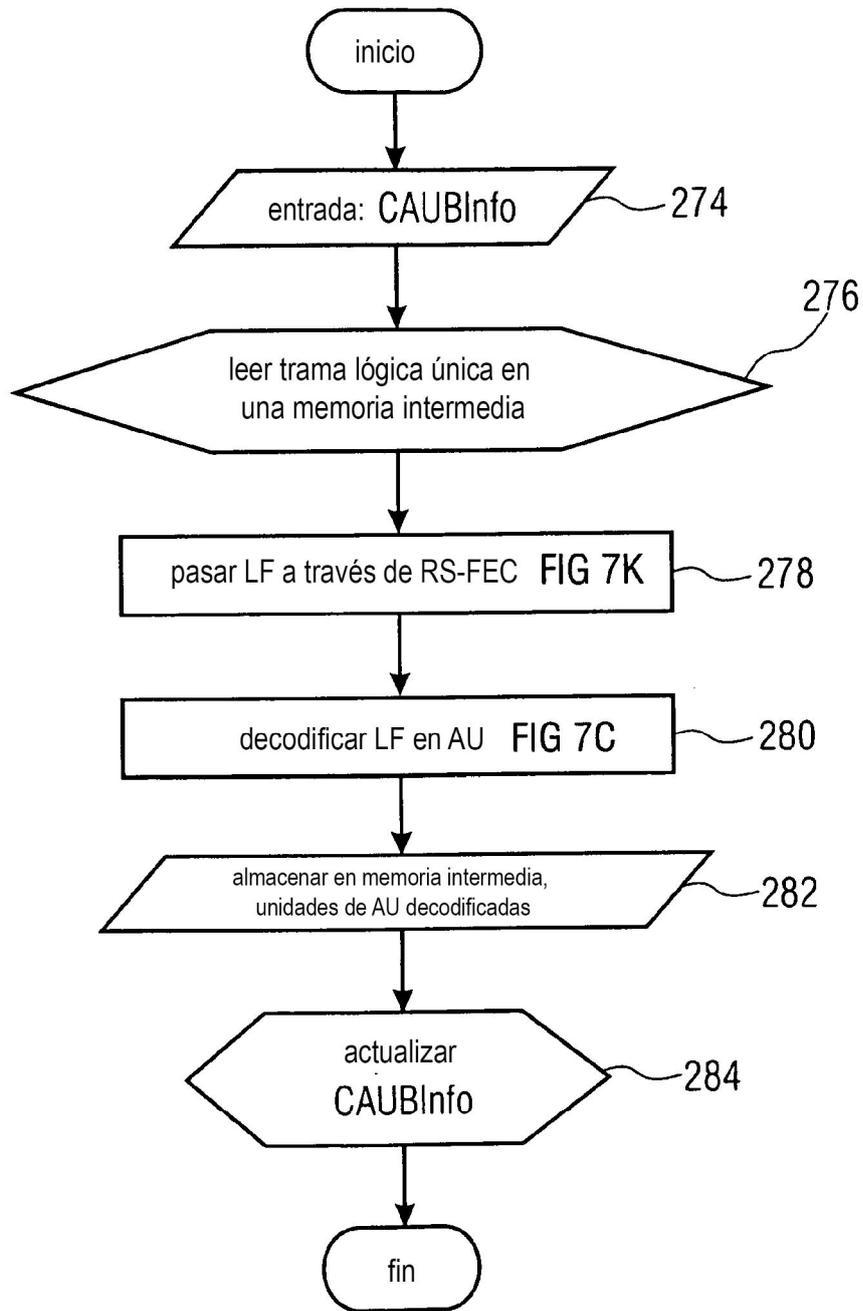


FIG 7J

278

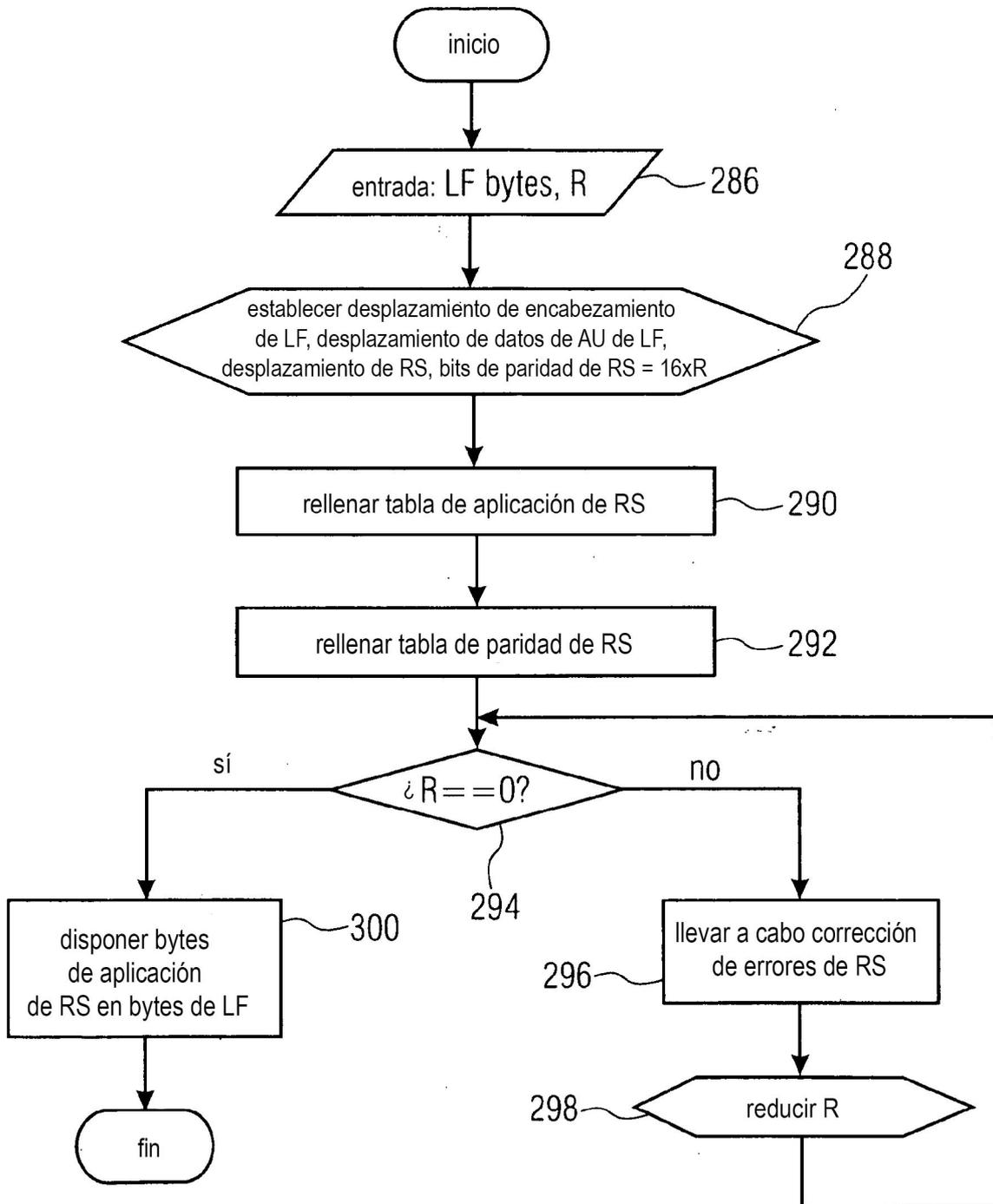


FIG 7K

270

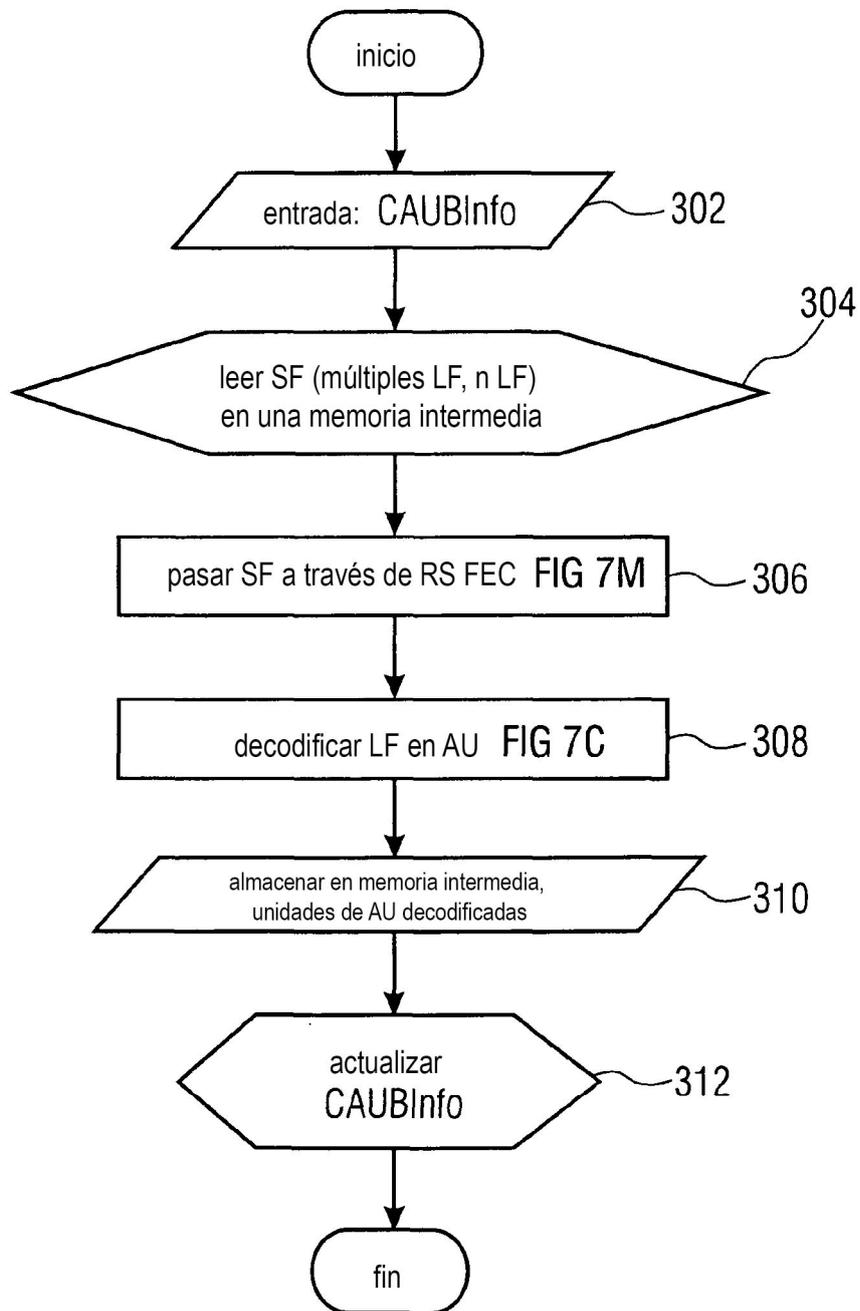


FIG 7L

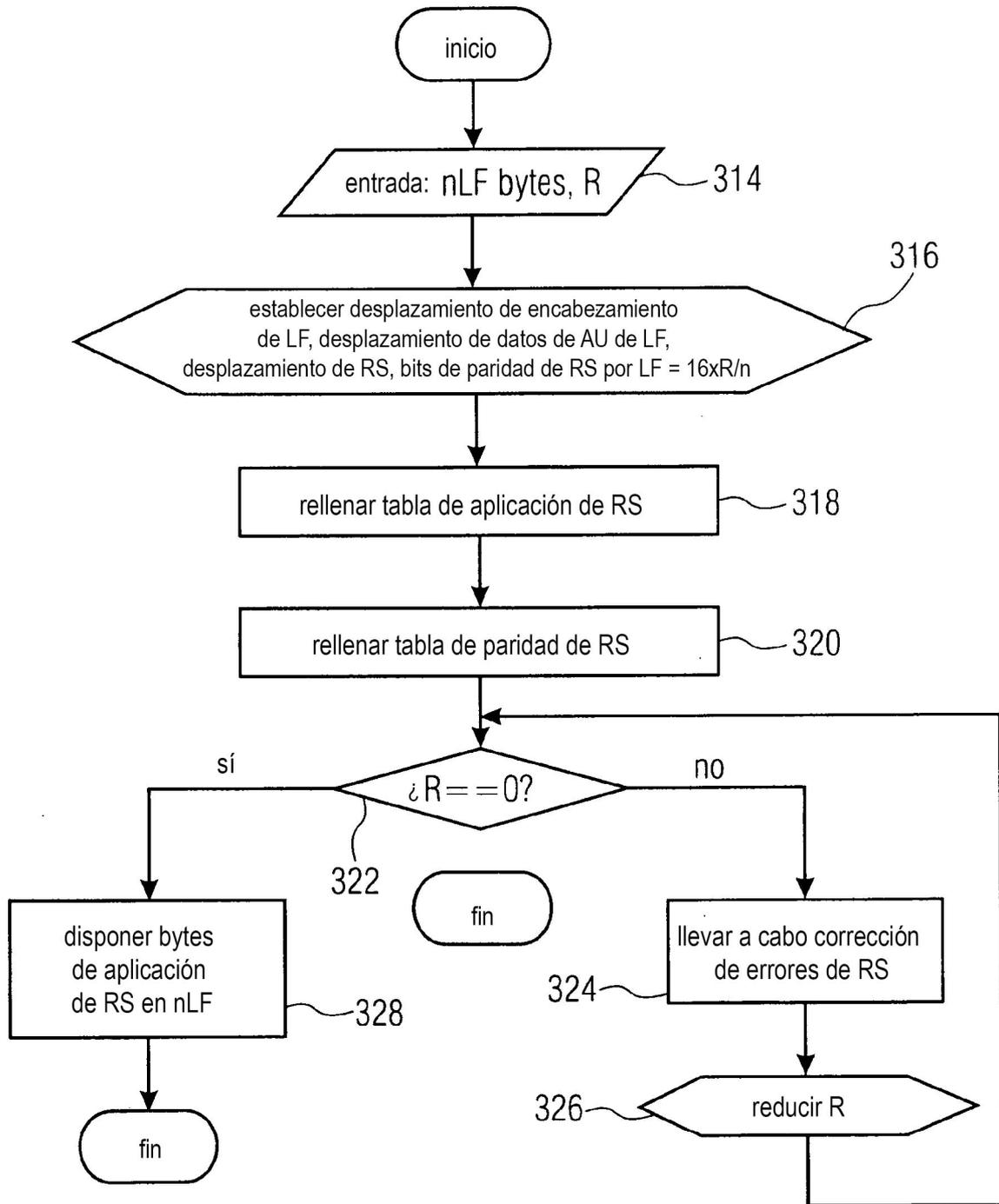


FIG 7M

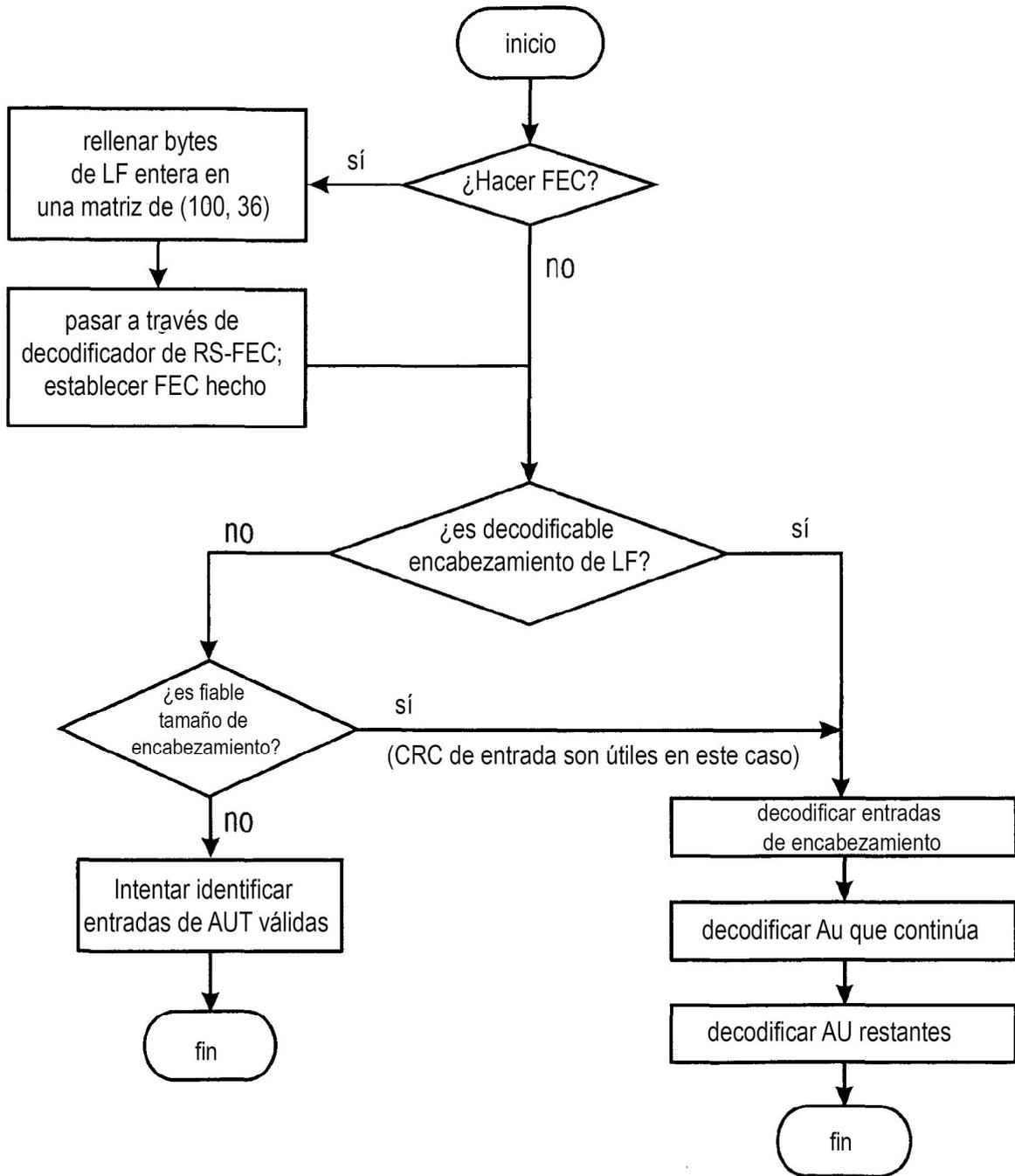


FIG 8

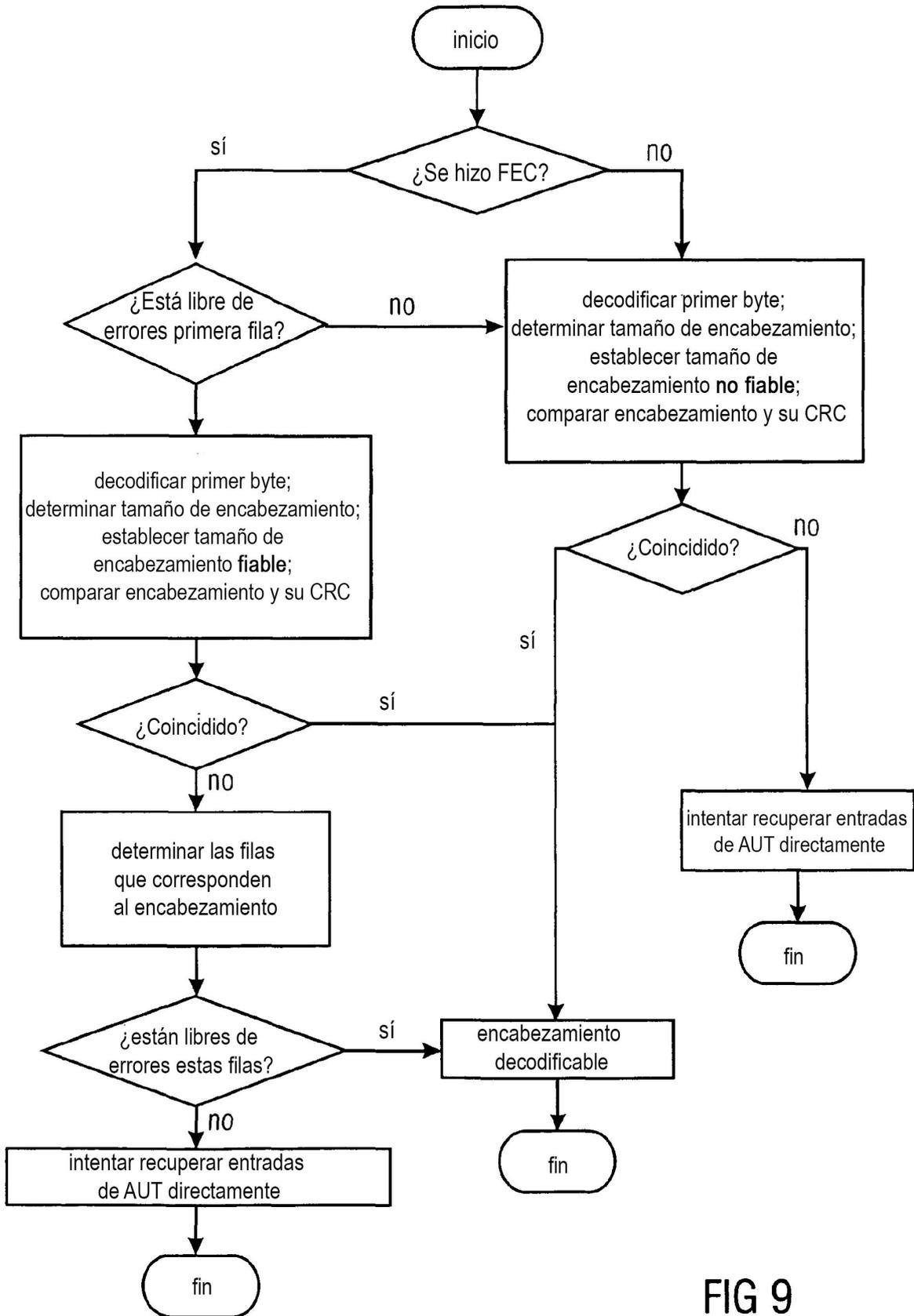


FIG 9