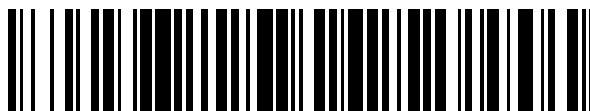


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 777 474**

51 Int. Cl.:

G10L 19/00 (2013.01)

G10L 19/16 (2013.01)

G10L 19/008 (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **12.06.2014 E 18156452 (7)**

97 Fecha y número de publicación de la concesión europea: **12.02.2020 EP 3373295**

54 Título: **Codificador y descodificador de audio con metadatos de información de programa**

30 Prioridad:

19.06.2013 US 201361836865 P

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

05.08.2020

73 Titular/es:

**DOLBY LABORATORIES LICENSING
CORPORATION (100.0%)
1275 Market Street
San Francisco, CA 94103, US**

72 Inventor/es:

**RIEDMILLER, JEFFREY y
WARD, MICHAEL**

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 777 474 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Codificador y decodificador de audio con metadatos de información de programa

Referencia cruzada a aplicaciones relacionadas

5 Esta solicitud reivindica la prioridad de la Solicitud de Patente Provisional de los Estados Unidos Nº 61/836.865, presentada el 19 de junio de 2013.

Esta solicitud es una solicitud divisional europea de la solicitud de patente Euro-PCT, EP14813862.1 (referencia: D13004EP01), presentada el 12 de junio de 2014.

Campo técnico

10 La invención se refiere al procesamiento de señales de audio y, más particularmente, a la codificación y decodificación de flujos de bits de datos de audio con metadatos indicativos de estructura de flujo secundario y/o información de programa con respecto al contenido de audio indicado por los flujos de bits. Algunas formas de realización de la invención generan o decodifican datos de audio en uno de los formatos conocidos como Dolby Digital (AC-3), Dolby Digital Plus (AC-3 Mejorado o E- AC-3) o Dolby E.

Antecedentes de la invención

15 Dolby, Dolby Digital, Dolby Digital Plus y Dolby E son marcas registradas de Dolby Laboratories Licensing Corporation. Dolby Laboratories proporciona puestas en práctica propietarias en AC-3 y E-AC-3, conocidas como Dolby Digital y Dolby Digital Plus, respectivamente.

20 Unidades de procesamiento de datos de audio operan normalmente en una forma denominada 'a ciegas' y no prestan atención al historial de procesamiento de los datos de audio que se produce antes de que se reciban los datos. Esto puede funcionar en un marco de procesamiento en el que una sola entidad realiza todas las funciones de procesamiento y codificación de los datos de audio para una diversidad de dispositivos de representación de medios objetivo, mientras que un dispositivo de representación de medios objetivo realiza toda la decodificación y representación de los datos de audio codificados. Sin embargo, este procesamiento 'a ciegas' no funciona bien (o no funciona) en situaciones en las que una pluralidad de unidades de procesamiento de audio está dispersada a través de una red diversa o se colocan en tándem (es decir, en cadena) y se espera que realicen de forma óptima sus respectivos tipos procesamiento de audio. Por ejemplo, algunos datos de audio pueden estar codificados para sistemas de medios de alto rendimiento, y pueden tener que convertirse a una forma reducida adecuada para un dispositivo móvil a lo largo de una cadena de procesamiento de medios. En consecuencia, una unidad de procesamiento de audio puede realizar, de forma innecesaria, un tipo de procesamiento en los datos de audio que ya se han realizado. Por ejemplo, una unidad de nivelación de sonoridad puede realizar el procesamiento sobre un clip de audio de entrada, independientemente de si se ha realizado anteriormente, o no, la misma o similar nivelación de sonoridad en el clip de audio de entrada. En consecuencia, la unidad de nivelación de sonoridad puede realizar la nivelación incluso cuando no sea necesario. Este procesamiento innecesario puede causar también la degradación y/o la eliminación de características específicas mientras se representa el contenido de los datos de audio. La solicitud internacional WO02/091361A1 describe el uso de bits residuales (campo de omisión) para añadir datos a una trama de datos comprimida. El documento ATSC Standard: Compresión de audio digital (AC-3, E-AC-3), doc. A/52: 2012, 17-12-2012, da a conocer las especificaciones de los flujos de datos de AC-3 y E-AC-3. El documento "Specification of the Broadcast Wave Format; a format for audio data files", EBU Tech 3285 supl.6, octubre de 2009, describe algunos Metadatos en Dolby y segmentos de Metadatos en Dolby.

Compendio de la invención

40 La invención da a conocer un método para generar un flujo de bits de audio codificado según la reivindicación 1, un método para decodificar un flujo bits de audio codificado según la reivindicación 2, un medio de memorización o almacenamiento legible por ordenador según la reivindicación 6, y una unidad de procesamiento de audio según la reivindicación 7.

45 En la descripción, a partir de aquí, siempre que aparezca la palabra "realización/realizaciones", si se refiere a combinaciones de características no compatibles con las definidas en las reivindicaciones, ha de considerarse que define ejemplos que se presentaron originalmente, pero que no representan realizaciones de la invención reivindicada, mostrándose estos ejemplos sólo con fines ilustrativos.

50 En un tipo de realizaciones, la invención es una unidad de procesamiento de audio capaz de decodificar un flujo de bits codificado que incluye metadatos de estructura de flujo secundario y/o metadatos de información sobre el programa (y, de modo opcional, también otros metadatos, por ejemplo metadatos de estado de procesamiento de sonoridad) en al menos un segmento de al menos una trama del flujo de bits y datos de audio en al menos otro segmento de la trama. En este caso, los metadatos de estructura flujo secundario (o "SSM") indican metadatos de un flujo de bits codificado (o conjunto de flujos de bits codificados), que indican la estructura de flujo secundario del contenido de audio del flujo o flujos de bits codificados y "metadatos de información sobre el programa" (o "PIM")

indican metadatos de un flujo de bits de audio codificado, indicativo de al menos un programa de audio (por ejemplo dos o más programas de audio), en donde los metadatos de información sobre el programa son indicativos de al menos una propiedad o característica del contenido de audio de al menos uno de dichos programas (por ejemplo metadatos que indican un tipo o parámetro de procesamiento realizado en datos de audio del programa, o metadatos que indican qué canales del programa son canales activos).

En casos típicos (por ejemplo, en los que el flujo de bits codificado es un flujo de bits en AC-3 o E-AC-3), los metadatos de información sobre el programa (PIM) son indicativos de información de programa que, prácticamente, no pueden transmitirse en otras partes del flujo de bits. Por ejemplo, los PIM pueden ser indicativos del procesamiento aplicado al audio de PCM antes de la codificación (por ejemplo, codificación AC-3 o E-AC-3), cuyas bandas de frecuencia del programa de audio se han codificado utilizando técnicas específicas de codificación de audio, y el perfil de compresión utilizado para crear datos de compresión de margen dinámico (DRC) en el flujo de bits.

En otro tipo de realizaciones, un método incluye una etapa de multiplexación de datos de audio codificados con SSM y/o PIM en cada trama (o de cada una de al menos algunas tramas) del flujo de bits. En la descodificación típica, un descodificador extrae el SSM y/o PIM del flujo de bits (incluyendo, mediante análisis y desmultiplexación de SSM y/o PIM y los datos de audio) y procesa los datos de audio para generar un flujo de datos de audio descodificados (y, en algunos casos, realiza además un procesamiento adaptativo de los datos de audio). En algunas formas de realización, los datos de audio descodificados y SSM y/o PIM, se reenvían desde el descodificador a un post-procesador configurado para realizar un procesamiento adaptativo sobre los datos de audio descodificados utilizando los SSM y/o los PIM.

En un tipo de realizaciones, el método de codificación inventivo genera un flujo de bits de audio codificado (por ejemplo, un flujo de bits en AC-3 o E-AC-3), que incluye segmentos de datos de audio (por ejemplo, los segmentos AB0-AB5 de la trama ilustrada en la Figura 4, o la totalidad o algunos de los segmentos AB0-AB5 de la trama ilustrada en la Figura 7), que incluye datos de audio codificados y segmentos de metadatos (incluyendo SSM y/o PIM y, opcionalmente, otros metadatos) multiplexados por división de tiempo con los segmentos de datos de audio. En algunas realizaciones, cada segmento de metadatos (a veces denominado en este documento como un "contenedor") tiene un formato que incluye una cabecera de segmento de metadatos (y, de forma opcional, también otros elementos obligatorios o "principales" o de núcleo), y una o más cargas útiles de metadatos después de la cabecera de segmento de metadatos. SIM, si está presente, se incluye en una de las cargas útiles de metadatos (que se identifica por una cabecera de carga útil y, por lo general, tiene un formato de un primer tipo). PIM, si está presente, se incluye en otra de las cargas útiles de metadatos (identificada por una cabecera de carga útil y que tiene normalmente un formato de un segundo tipo). De modo similar, cada otro tipo de metadatos (si está presente) se incluye en una distinta de las cargas útiles de metadatos (que se identifica por una cabecera de carga y que suele tener un formato específico para el tipo de metadatos). El formato, a modo de ejemplo, permite un acceso conveniente para los SSM, PIM y otros metadatos en momentos diferentes a la descodificación (por ejemplo, por un post-procesador que sigue a la descodificación, o mediante un procesador configurado para reconocer los metadatos sin realizar una descodificación completa en el flujo de bits codificado), y permite la detección y corrección de errores de forma conveniente y eficiente (por ejemplo, de identificación de flujo secundario) durante la descodificación del flujo de bits. Por ejemplo, sin acceso a SSM en el formato de ejemplo, un descodificador podría identificar, incorrectamente, el número correcto de flujos secundarios asociados con un programa. Una carga útil de metadatos, en un segmento de metadatos, puede incluir SSM, otra carga útil de metadatos, en el segmento de metadatos, puede incluir PIM y, opcionalmente, al menos otra carga útil de metadatos, en el segmento de metadatos, puede incluir otros metadatos (por ejemplo, metadatos de estado de procesamiento de sonoridad o "LPSM").

Breve descripción de los dibujos

La Figura 1 es un diagrama de bloques de una forma de realización de un sistema que puede configurarse para realizar una forma de realización del método de la invención.

La Figura 2 es un diagrama de bloques de un codificador; que es una realización de la unidad de procesamiento de audio inventiva.

La Figura 3 es un diagrama de bloques de un descodificador, que es una realización de la unidad de procesamiento de audio inventiva, y un post-procesador acoplado al mismo, que es otra realización de la unidad de procesamiento de audio inventiva.

La Figura 4 es un diagrama de una trama en AC-3, que incluye los segmentos en los que está dividido.

La Figura 5 es un diagrama del segmento de Información de Sincronización (SI) de una trama en AC-3, que incluye los segmentos en los que está dividido.

La Figura 6 es un diagrama del segmento de Información de Flujo de bits (BSI) de una trama en AC-3, incluidos los segmentos en los que está dividido.

La Figura 7 es un diagrama de una trama en E-AC-3, incluidos los segmentos en los que está dividido.

La Figura 8 es un diagrama de un segmento de metadatos de un flujo de bits codificado, generado de conformidad con una forma de realización de la invención, que incluye una cabecera de segmento de metadatos que comprende una palabra de sincronización de contenedor (identificada como "sincronización de contenedor" en la Figura 8) y valores de versión y de ID de clave, seguido de múltiples cargas útiles de metadatos y bits de protección.

NOTACIÓN Y TERMINOLOGÍA

A través de esta descripción, incluyendo las reivindicaciones, la expresión que realiza una operación "sobre" una señal o datos (por ejemplo, filtrado, escalado, transformación o aplicación de ganancia a, la señal o datos) se usa en un sentido amplio para indicar la realización de la operación directamente en la señal o datos o, sobre una versión procesada, de la señal o datos (por ejemplo sobre una versión de la señal que ha sido sometida a filtrado preliminar o pre-procesamiento antes de la ejecución de la propia operación).

A lo largo de esta descripción, que incluye las reivindicaciones, la expresión "sistema" se utiliza en un sentido amplio para designar un dispositivo, sistema o subsistema. A modo de ejemplo, un subsistema que pone en práctica un decodificador se puede referir como un sistema decodificador, y un sistema que incluye dicho subsistema (por ejemplo, un sistema que genera X señales de salida en respuesta a múltiples entradas, en donde el subsistema genera M de las entradas y las otras X - M entradas se reciben desde una fuente externa) se puede denominar también un sistema decodificador.

A través de esta descripción, que incluye las reivindicaciones, el término "procesador" se utiliza en un sentido amplio para designar un sistema o dispositivo programable o, de cualquier otro modo, configurable (por ejemplo, con software o firmware) para realizar operaciones en datos (por ejemplo, audio, o vídeo u otros datos de imagen). Ejemplos de procesadores incluyen una matriz de puertas programable in situ (u otro conjunto de circuitos integrados configurables o de chips), un procesador de señal digital programado y/o, de otro modo, configurado para realizar un procesamiento más eficiente en datos de audio u otros datos de sonido, un procesador u ordenador de uso general programable, y un chip o conjunto de chips o de microprocesador programable.

A lo largo de esta descripción, que incluye las reivindicaciones, las expresiones "procesador de audio" y "unidad de procesamiento de audio" se utilizan indistintamente y, en un sentido amplio, para indicar un sistema configurado para procesar datos de audio. Ejemplos de unidades de procesamiento de audio incluyen, pero sin limitación, codificadores (por ejemplo, transcodificadores), decodificadores, codificadores-descodificadores (codes), sistemas de procesamiento previo, sistemas de procesamiento posterior y sistemas de procesamiento de flujo de bits (a veces referidos como herramientas de procesamiento de flujo de bits).

A través de esta descripción, que incluye las reivindicaciones, la expresión "metadatos" (de un flujo de bits de audio codificado), se refiere a datos separados y diferentes de los datos de audio correspondientes del flujo de bits.

A lo largo de esta descripción, que incluye las reivindicaciones, la expresión "metadatos de estructura de flujo secundario" (o "SSM"), indica metadatos de un flujo de bits de audio codificado (o un conjunto de flujos de bits de audio codificados), que indican la estructura flujo secundario del contenido de audio del o de los flujos de bits codificados.

A través de esta descripción, que incluye las reivindicaciones, la expresión "metadatos de información sobre el programa" (o "PIM") indica metadatos de un flujo de bits de audio codificado, indicativo de al menos un programa de audio (por ejemplo, dos o más programas de audio), en donde dichos metadatos indican, al menos, una propiedad o característica del contenido de audio de al menos uno de dichos programas (por ejemplo, metadatos que indican un tipo o parámetro de procesamiento realizado sobre datos de audio del programa, o metadatos que indican qué canales del programa son canales activos).

A lo largo de esta descripción, que incluye las reivindicaciones, la expresión "metadatos de estado de procesamiento" (por ejemplo, como en la expresión "metadatos de estado de procesamiento de sonoridad"), se refiere a metadatos (de un flujo de bits de audio codificado) asociados con datos de audio del flujo de bits, que indican el estado de procesamiento de los datos de audio correspondientes (asociados) (por ejemplo, qué tipo o tipos de procesamiento ya se han realizado en los datos de audio) y normalmente indica, además, al menos una función o característica de los datos de audio. La asociación de los metadatos del estado de procesamiento con los datos de audio está sincronizada en el tiempo. Por lo tanto, los metadatos de estado de procesamiento presentes (más recientemente recibidos o actualizados) indican que los correspondientes datos de audio comprenden, contemporáneamente, los resultados del tipo o tipos indicados de procesamiento de datos de audio. En algunos casos, metadatos de estado de procesamiento pueden incluir el historial de procesamiento y/o algunos, o la totalidad, de los parámetros que se utilizan y/o derivan de los tipos de procesamiento indicados. Además, los metadatos de estado de procesamiento pueden incluir al menos una función o característica de los correspondientes datos de audio, que se ha calculado o extraído a partir de los datos de audio. Metadatos de estado de procesamiento pueden incluir, además, otros metadatos que no están relacionados, ni derivados, de ningún procesamiento de los datos de audio correspondientes. Por ejemplo, datos de terceros, información de seguimiento, identificadores, información de propiedad o estándar, datos de anotación de usuario, datos de preferencia de usuario, etc., pueden ser añadidos por una unidad de procesamiento de audio

particular para pasar a otras unidades de procesamiento de audio.

5 A través de esta descripción, que incluye las reivindicaciones, la expresión "metadatos de estado de procesamiento de sonoridad" (o "LPSM"), indica metadatos de estado de procesamiento, indicativos del estado de procesamiento de sonoridad de los correspondientes datos de audio (por ejemplo, qué tipo(s) de procesamiento(s) de sonoridad se ha(n) realizado en los datos de audio) y normalmente también al menos una función o característica (por ejemplo, sonoridad) de los correspondientes datos de audio. Los metadatos de estado de procesamiento de sonoridad pueden incluir datos (por ejemplo, otros metadatos) que no son (es decir, cuando se considera por sí solos) metadatos de estado de procesamiento de sonoridad.

10 A lo largo de esta descripción, que incluye las reivindicaciones, la expresión "canal" (o "canal de audio"), indica una señal de audio monofónica.

A través de esta descripción, que incluye las reivindicaciones, la expresión "programa de audio" indica un conjunto de uno o más canales de audio y, opcionalmente, también metadatos asociados (por ejemplo, metadatos que describen una presentación de audio espacial deseada, y/o PIM, y/o SSM, y/o LPSM y/o metadatos de límite de programa).

15 A lo largo de esta descripción, que incluye las reivindicaciones, la expresión "metadatos de límite de programa", indica metadatos de un flujo de bits de audio codificado, en donde el flujo de bits de audio codificado es indicativo de al menos un programa de audio (por ejemplo, dos o más programas de audio), y los metadatos de límite del programa indican la ubicación, en el flujo de bits, de al menos un límite (comienzo y/o final) de al menos uno de dichos programas de audio. Por ejemplo, los metadatos de límite de programa (de un flujo de bits de audio codificado, indicativo de un programa de audio), pueden incluir metadatos que indican la localización (por ejemplo, el inicio de la "N"-sima trama del flujo de bits, o la "M"-sima localización de muestra de la "N"-sima trama del flujo de bits), desde el comienzo del programa, y metadatos adicionales indicativos de la ubicación (por ejemplo, la "J"-ésima trama del flujo de bits, o la "K"-ésima localización de muestra de la "J"-ésima trama del flujo de bits) del final de programa.

20 A través de esta descripción, que incluye las reivindicaciones, el término "acopla" o "acoplado" se utiliza para indicar una conexión directa o indirecta. En consecuencia, si un primer dispositivo se acopla a un segundo dispositivo, esa conexión puede ser a través de una conexión directa, o a través de una conexión indirecta, a través de otros dispositivos y conexiones.

Descripción detallada de las formas de realización de la invención

30 Un flujo típico de datos de audio incluye tanto contenido de audio (por ejemplo, uno o más canales de contenido de audio) como metadatos indicativos de al menos una característica del contenido de audio. Por ejemplo, en un flujo de bits en AC-3 existen varios parámetros de metadatos de audio que están específicamente previstos para ser utilizados para cambiar el sonido del programa entregado a un entorno de escucha. Uno de los parámetros de metadatos es el parámetro DIALNORM, que está destinado a indicar el nivel medio de diálogo en un programa de audio, y se utiliza para determinar el nivel de señal de reproducción de audio.

35 Durante la reproducción de un flujo de bits que comprende una secuencia de diferentes segmentos de programa de audio, (cada uno con un parámetro DIALNORM diferente), un decodificador en AC-3 utiliza el parámetro DIALNORM de cada segmento para realizar un tipo de procesamiento de sonoridad, en el que modifica el nivel de reproducción o sonoridad de tal manera que la sonoridad percibida del diálogo de la secuencia de segmentos está en un nivel constante. Cada segmento (ítem) de audio codificado, en una secuencia de ítems de audio codificados, tendría (en general) un parámetro DIALNORM diferente, y el decodificador establecería a escala el nivel de cada uno de los ítems de modo que el nivel de reproducción o la sonoridad del diálogo, para cada elemento, sea igual o muy similar, aunque esto podría requerir la aplicación de diferentes cantidades de ganancia a diferentes ítems durante la reproducción.

40 El parámetro DIALNORM se suele establecer por un usuario, y no se genera de forma automática, aunque existe un valor por defecto del parámetro DIALNORM si el usuario no establece ningún valor. Por ejemplo, un creador de contenido puede realizar mediciones de sonoridad con un dispositivo externo a un codificador en AC-3 y a continuación, transmitir el resultado (indicativo de la intensidad del diálogo hablado de un programa de audio) al codificador para establecer el valor de DIALNORM. Por lo tanto, se confía en el creador del contenido para establecer correctamente el parámetro DIALNORM.

50 Existen varias razones diferentes por las que el parámetro DIALNORM, en un flujo de bits en AC-3, puede ser incorrecto. En primer lugar, cada codificador en AC-3 tiene un valor de DIALNORM por defecto que se utiliza durante la generación del flujo de bits si el creador de contenido no establece un valor de DIALNORM. Este valor por defecto puede ser esencialmente distinto del nivel de sonoridad de diálogo real del audio. En segundo lugar, incluso si un creador de contenido mide la sonoridad y establece el valor de DIALNORM en consecuencia, es posible que se haya utilizado un algoritmo de medición de sonoridad, o medidor, que no esté en conformidad con el método de medición de sonoridad en AC-3 recomendado, lo que da como resultado un valor de DIALNORM incorrecto. En tercer lugar, incluso si se ha creado un flujo de bits en AC-3 con el valor de DIALNORM medido y establecido correctamente por el creador del contenido, puede haberse cambiado a un valor incorrecto durante la transmisión y/o memorización del flujo de bits. Por ejemplo, no es infrecuente en aplicaciones de difusión de televisión que flujos de bits en AC-3 se

descodifiquen, modifiquen y a continuación, se vuelvan a codificar utilizando información de metadatos de DIALNORM incorrecta. De este modo, un valor de DIALNORM, incluido en un flujo de bits en AC-3, puede ser incorrecto o impreciso y, por lo tanto, puede tener un impacto negativo sobre la calidad de la experiencia de escucha.

5 Además, el parámetro DIALNORM no indica el estado de procesamiento de la sonoridad de los correspondientes datos de audio (por ejemplo, qué tipo de procesamiento de sonoridad se ha realizado en los datos de audio). Metadatos de estado de procesamiento de sonoridad (en el formato en el que se proporcionan en algunas formas de realización de la presente invención), son útiles para facilitar el procesamiento de sonoridad adaptativo de un flujo de bits de audio y/o la verificación de la validez del estado de procesamiento de sonoridad y la sonoridad del contenido de audio, de una manera particularmente eficiente.

10 Un flujo de bits codificado en AC-3 comprende metadatos, y uno a seis canales de contenido de audio. El contenido de audio son datos de audio que se han comprimido utilizando una codificación de audio perceptivo. Los metadatos incluyen varios parámetros de metadatos de audio que están destinados a ser utilizados para cambiar el sonido de un programa proporcionado a un entorno de escucha.

15 Cada trama de un flujo de bits de audio codificado en AC-3, incluye contenido de audio y metadatos para 1536 muestras de audio digital. Para una tasa de muestreo de 48 kHz, esto representa 32 milisegundos de audio digital o una tasa de 31,25 tramas por segundo de audio.

20 Cada trama de un flujo de bits de audio codificado en E-AC-3, incluye contenido de audio y metadatos para 256, 512, 768 o 1536 muestras de audio digital, dependiendo de si la trama contiene uno, dos, tres o seis bloques de datos de audio, respectivamente. Para una tasa de muestreo de 48 kHz, esto representa 5,333, 10,667, 16 ó 32 milisegundos de audio digital, respectivamente, o una tasa de 189,9, 93,75, 62,5 ó 31,25 tramas por segundo de audio, respectivamente.

25 Tal como se indica en la Figura 4, cada trama en AC-3 está dividida en secciones (segmentos), que incluyen: una sección de Información de Sincronización (SI), que contiene (tal como se muestra en la Figura 5), una palabra de sincronización (SW) y la primera de dos palabras de corrección de errores (CRC1); una sección de Información de Flujo de Bits (BSI), que contiene la mayoría de los metadatos; seis Bloques de Audio (AB0 a AB5), que tienen contenido de audio comprimidos de datos (y además, pueden incluir metadatos); segmentos de bits residuales (W) (también conocidos como "campos de omisión"), que contienen cualesquiera bits no utilizados que queden después de que se comprima el contenido de audio; una sección de información Auxiliar (AUX) que puede incluir más metadatos; y la segunda de dos palabras de corrección de error (CRC2).

30 Tal como se indica en la Figura 7, cada trama en E-AC-3 está dividida en secciones (segmentos), que incluyen: una sección de Información de Sincronización (SI), que contiene (tal como se muestra en la Figura 5) una palabra de sincronización (SW); una sección de Información de Flujo de bits (BSI), que contiene la mayoría de los metadatos; entre uno y seis Bloques de Audio (AB0 a AB5), que tienen contenido de audio comprimidos de datos (y, además, pueden incluir metadatos); segmentos de bits residuales (W) (también conocidos como "campos de omisión"), que contienen cualesquiera bits no utilizados que queden después de comprimir el contenido de audio (aunque solamente se ilustra un segmento de bits residuales, un segmento de bits residuales diferente, o segmento de campo de omisión, que normalmente seguiría cada bloque de audio); una sección de información Auxiliar (AUX) que puede incluir más metadatos; y una palabra de corrección de error (CRC).

40 En un flujo de bits en AC-3 (o E-AC-3) existen varios parámetros de metadatos de audio que están previstos, específicamente, a ser utilizados para cambiar el sonido del programa que se proporciona a un entorno de escucha. Uno de los parámetros de metadatos es el parámetro DIALNORM, que está incluido en el segmento de BSI.

45 Según se ilustra en la Figura 6, el segmento de BSI de una trama en AC-3 incluye un parámetro de cinco bits ("DIALNORM"), que indica el valor de DIALNORM para el programa. Se incluye un parámetro de cinco bits ("DIALNORM2"), que indica el valor de DIALNORM para un segundo programa de audio que se transmite en la misma trama en AC-3 si el modo de codificación de audio ("acmod"), de la trama en AC-3, es "0", lo que indica que se está utilizando una configuración de canal dual-mono o "1+1".

50 El segmento de BSI incluye, además, un indicador ("addbsie"), que indica la presencia, (o ausencia), de información de flujo de bits adicional que sigue al bit de "addbsie", un parámetro ("addbsil"), que indica la longitud de cualquier información adicional de flujo de bits que sigue al valor de "addbsil", y hasta 64 bits de información de flujo de bits adicional ("addbsi") que sigue al valor de "addbsil".

El segmento de BSI incluye otros valores de metadatos que no se ilustran específicamente en la Figura 6.

55 De conformidad con una clase de formas de realización, un flujo de bits de audio codificado es indicativo de múltiples flujos secundarios de contenido de audio. En algunos casos, los flujos secundarios son indicativos del contenido de audio de un programa multicanal, y cada uno de los flujos secundarios indica uno o más de los canales del programa. En otros casos, múltiples flujos secundarios de un flujo bits de audio codificado, son indicativos del contenido de audio de varios programas de audio, normalmente un programa de audio "principal" (que puede ser un programa multicanal), y al menos otro programa de audio (por ejemplo, un programa que es un comentario sobre el programa de audio

principal).

5 Un flujo de bits de audio codificado, que es indicativo de al menos un programa de audio, incluye necesariamente al menos un flujo secundario "independiente" de contenido de audio. El flujo secundario independiente es indicativo de al menos un canal de un programa de audio (por ejemplo, el flujo secundario independiente puede ser indicativo de los cinco canales de margen completo de un programa de audio de canal 5.1 convencional). En este caso, este programa de audio se refiere como un programa "principal".

10 En algunas clases de formas de realización, un flujo de bits de audio codificado es indicativo de dos o más programas de audio (un programa "principal" y al menos otro programa de audio). En tales casos, el flujo de bits incluye dos o más flujos secundarios independientes: un primer flujo secundario independiente, que indica al menos un canal del programa principal; y al menos otro flujo secundario independiente, indicativo de al menos un canal de otro programa de audio (un programa distinto del programa principal). Cada flujo de bits independiente se puede descodificar de forma independiente, y un descodificador podría funcionar para descodificar solamente un subconjunto (no la totalidad) de los flujos secundarios independientes de un flujo de bits codificado.

15 En un ejemplo típico de un flujo de bits de audio codificado, que es indicativo de dos flujos secundarios independientes, uno de los flujos secundarios independientes es indicativo de canales de altavoz de formato estándar de un programa principal multicanal (por ejemplo, Izquierda, Derecha, Centro, Rodeando la Izquierda, Rodeando la Derecha como canales de altavoz de gama completa de un programa principal de canal 5.1), y el otro flujo secundario independiente es indicativo de un comentario de audio monofónico en el programa principal (por ejemplo, un comentario de un director sobre una película, en donde el programa principal es la banda sonora de la película). En otro ejemplo de un flujo de bits de audio codificado, que indica múltiples flujos secundarios independientes, uno de los flujos secundarios independientes es indicativo de canales de altavoz de formato estándar de un programa principal multicanal (por ejemplo, un programa principal de canal 5.1), que incluye diálogo en un primer idioma (por ejemplo, uno de los canales de altavoz del programa principal puede ser indicativo del diálogo), y cada uno de los otros flujos secundarios independientes es indicativo de una traducción monofónica (en un idioma diferente) del diálogo.

25 Opcionalmente, un flujo de bits de audio codificado, que es indicativo de un programa principal (y, de forma opcional, también de al menos otro programa de audio) incluye al menos un flujo secundario "dependiente" de contenido de audio. Cada flujo secundario dependiente está asociado con un flujo secundario independiente del flujo de bits, y es indicativo de al menos un canal adicional del programa (por ejemplo, el programa principal), cuyo contenido está indicado por el flujo secundario independiente asociado (es decir, el flujo secundario dependiente indica al menos un canal de un programa que no está indicado por el flujo secundario independiente asociado, y el flujo secundario independiente asociado, indica al menos un canal del programa).

35 En un ejemplo de un flujo de bits codificado que incluye un flujo secundario independiente (indicativo de al menos un canal de un programa principal), el flujo de bits incluye además un flujo secundario dependiente (asociado con el flujo de bits independiente) que es indicativo de uno o más canales de altavoz del programa principal. Dichos canales de altavoz adicionales, son adicionales al canal o los canales del programa principal indicados por el flujo secundario independiente. Por ejemplo, si el flujo secundario independiente es indicativo de los formatos estándar izquierdo, derecho, central, rodeando al izquierdo, rodeando al derecho, como canales de altavoz de gama completa de un programa principal de canal 7.1, pudiendo el flujo secundario dependiente ser indicativo de los otros dos canales de altavoz de gama completa del programa principal.

40 De conformidad con la norma E-AC-3, un flujo de bits en E-AC-3 debe ser indicativo de al menos un flujo secundario independiente (por ejemplo, un único flujo de bits en AC-3), y puede ser indicativo de hasta ocho flujos secundarios independientes. Cada flujo secundario independiente, de un flujo de bits en E-AC-3, puede asociarse con hasta ocho flujos secundarios dependientes.

45 Un flujo de bits en E-AC-3 incluye metadatos que indican la estructura de flujo secundario del flujo de bits. Por ejemplo, un campo "chanmap", en la sección de Información de Flujo de Bits (BSI), de un flujo de bits en E-AC-3, determina un mapa de canales para los canales de programa indicados por un flujo secundario dependiente del flujo bits. Sin embargo, los metadatos indicativos de la estructura de flujo secundario se incluyen, de modo convencional, en un flujo de bits en E-AC-3, en un formato tal que es conveniente para el acceso y uso (durante la descodificación del flujo de bits codificado en E-AC-3) solamente por un descodificador en E-AC-3; no para el acceso y uso después de la descodificación (por ejemplo, por un post-procesador), o antes de la descodificación (por ejemplo, por un procesador configurado para reconocer los metadatos). Además, existe el riesgo de que un descodificador identifique incorrectamente los flujos secundarios de un flujo de bits codificado en E-AC-3 convencional, utilizando los metadatos incluidos convencionalmente, y hasta la presente invención no se ha tenido conocimiento de cómo incluir metadatos de estructura de flujo secundario en un flujo de bits codificado (por ejemplo, un flujo de bits en E-AC-3 codificado) en un formato tal que permita la detección y corrección, convenientes y eficientes, de errores en la identificación del flujo secundario durante la descodificación del flujo de bits.

Un flujo de bits en E-AC-3 puede incluir también metadatos con respecto al contenido de audio de un programa de audio. Por ejemplo, un flujo de bits en E-AC-3, indicativo de un programa de audio, incluye metadatos indicativos de frecuencias mínima y máxima en los que se ha utilizado el procesamiento de extensión espectral (y la codificación de

acoplamiento de canal), con el fin de codificar el contenido del programa. Sin embargo, tales metadatos se suelen incluir en un flujo de bits en E-AC-3 en un formato tal que es conveniente para el acceso y uso (durante la descodificación del flujo de bits en E-AC-3 codificado) solamente por un descodificador en E-AC-3; no para el acceso y uso después de la descodificación (por ejemplo, por un post-procesador), o antes de la descodificación (por ejemplo, por un procesador configurado para reconocer los metadatos). Además, dichos metadatos no se incluyen en un flujo de bits en E-AC-3 en un formato que permita la detección de error y la corrección de error convenientes y eficientes de dichos metadatos durante la descodificación del flujo de bits.

De conformidad con las formas de realización típicas de la invención, PIM y/o SSM (y opcionalmente, también otros metadatos, por ejemplo metadatos de estado de procesamiento de sonoridad o "LPSM") están incluidos en uno o más campos reservados (o ranuras) de segmentos de metadatos de un flujo de bits de audio que incluye, además, datos de audio en otros segmentos (segmentos de datos de audio). Normalmente, al menos un segmento de cada trama del flujo de bits incluye PIM o SSM, y al menos otro segmento de la trama incluye datos de audio correspondientes (es decir, datos de audio cuya estructura de flujo secundario está indicada por SSM y/o tiene al menos una característica o propiedad indicada por los PIM).

En una clase de formas de realización, cada segmento de metadatos es una estructura de datos (a veces referida aquí como un contenedor) que puede incluir una o más cargas útiles de metadatos. Cada carga útil incluye una cabecera que comprende un identificador de carga útil específico (y datos de configuración de carga útil), para proporcionar una indicación inequívoca del tipo de metadatos presentes en la carga útil. El orden de las cargas dentro del contenedor no está definido, de modo que las cargas útiles se pueden memorizar en cualquier orden, y un analizador sintáctico debe poder analizar el contenedor completo con el fin de extraer cargas útiles pertinentes e ignorar las cargas útiles que no son pertinentes o no están soportadas. La Figura 8 (que se describirá a continuación) ilustra la estructura de dicho contenedor y cargas útiles dentro del contenedor.

La comunicación de metadatos (por ejemplo., SSM y/o PIM y/o LPSM) en una cadena de procesamiento de datos de audio, es particularmente útil cuando dos o más unidades de procesamiento de audio necesitan funcionar 'en tándem' entre sí a lo largo de la cadena de procesamiento (o ciclo de vida del contenido). Sin la inclusión de metadatos en un flujo de bits de audio, pueden producirse graves problemas de procesamiento del medio tales como degradaciones de calidad, de nivel y espaciales, por ejemplo, cuando se utilizan dos o más codecs de audio en la cadena y la nivelación de volumen de terminación única se aplica más de una vez durante una ruta de flujo de bits a un dispositivo que consume medios (o un punto de representación del contenido de audio del flujo de bits).

Los metadatos de estado de procesamiento de sonoridad (LPSM), que se incluyen en un flujo de bits de audio de conformidad con algunas formas de realización de la invención, se pueden autenticar y validar, por ejemplo para permitir a las entidades reguladoras de sonoridad la comprobación de si la sonoridad de un programa particular está ya dentro de un margen especificado, y que los propios datos de audio correspondientes no han sido modificados (con lo que se asegura el cumplimiento de la normativa aplicable). Un valor de sonoridad, incluido en un bloque de datos que comprende los metadatos de estado de procesamiento de sonoridad, puede ser objeto de lectura para la verificación de lo que antecede, en lugar de calcular la sonoridad de nuevo. En respuesta a LPSM, una agencia reguladora puede determinar que el contenido de audio correspondiente está en cumplimiento (según lo indicado por el LPSM) con requisitos reglamentarios y/o estatutarios de intensidad sonora (por ejemplo, las normativas promulgadas bajo la Ley de Mitigación de la Sonoridad de Anuncios Comerciales, también conocida como la Ley de "CALM") sin la necesidad de calcular la sonoridad del contenido de audio.

La Figura 1 es un diagrama de bloques de un ejemplo de cadena de procesamiento de audio (un sistema de procesamiento de datos de audio), en el que uno o más de los elementos del sistema se pueden configurar de conformidad con una forma de realización de la presente invención. El sistema incluye los siguientes elementos, acoplados conjuntamente según se ilustra: una unidad de procesamiento previo, un codificador, una unidad de corrección de metadatos y análisis de señal, un transcodificador, un descodificador y una unidad de procesamiento previo. En las variaciones del sistema ilustrado, se omiten uno o más de los elementos, o se incluyen unidades adicionales de procesamiento de datos de audio.

En algunas ejecuciones prácticas, la unidad de procesamiento previo de la Figura 1 está configurada para aceptar muestras de PCM (dominio de tiempo), que comprenden contenido de audio como entrada, y para emitir muestras de PCM procesadas. El codificador se puede configurar para aceptar las muestras de PCM como entrada, y para proporcionar, a la salida, un flujo de bits de audio codificado (por ejemplo, comprimido), que indica el contenido de audio. Los datos del flujo de bits que son indicativos del contenido de audio a veces se denominan aquí "datos de audio". Si el codificador está configurado de conformidad con una forma de realización típica de la presente invención, la salida de flujo de bits de audio desde el codificador incluye PIM y/o SSM (y opcionalmente, además, metadatos de estado de procesamiento de sonoridad y/o otros metadatos) así como datos de audio.

La unidad de corrección de metadatos y análisis de señal de la Figura 1 puede aceptar uno o más flujos de bits de audio codificados como entrada, y determinar (por ejemplo, validar), si los metadatos (por ejemplo, metadatos de estado de procesamiento), en cada flujo de bits de audio codificado son correctos, realizando análisis de señal (por ejemplo, utilizando metadatos de límite de programa en un flujo de bits de audio codificado). Si la unidad de corrección de metadatos y análisis de señal encuentra que los metadatos incluidos no son válidos, normalmente, sustituye los

valores incorrectos con los valores correctos obtenidos a partir del análisis de señal. De este modo, cada salida de flujo de bits de audio codificado, procedente de la unidad de corrección de metadatos y análisis de señal, puede incluir metadatos de estado de procesamiento corregidos, (o no corregidos), así como datos de audio codificados.

5 El transcodificador de la Figura 1 puede aceptar flujos de bits de audio codificados como entrada, y a la salida, responder con flujos de bits de audio modificados (por ejemplo, codificados de modo distinto) (por ejemplo, descodificando un flujo de entrada y recodificando el flujo descodificado en un formato de codificación diferente). Si el transcodificador está configurado de conformidad con una forma de realización típica de la presente invención, el flujo de bits de audio de salida, procedente del transcodificador, incluye SSM y/o PIM (y además, normalmente otros metadatos), así como datos de audio codificados. Los metadatos pueden haberse incluido en el flujo de bits de entrada.

10 El descodificador de la Figura 1 puede aceptar flujos de bits de audio codificados (por ejemplo, comprimidos) como entrada y, a la salida (en respuesta) flujos de muestras de audio de PCM descodificadas. Si el descodificador está configurado de conformidad con una forma de realización típica de la presente invención, la salida del descodificador, en funcionamiento típico, es o incluye cualquiera de lo que sigue:

15 un flujo de muestras de audio, y al menos un flujo correspondiente de SSM y/o PIM (y normalmente también otros metadatos), extraídos de un flujo de bits codificado de entrada; o

un flujo de muestras de audio, y un flujo correspondiente de bits de control determinados a partir de SSM y/o PIM (y normalmente también otros metadatos, por ejemplo, LPSM), extraídos de un flujo bits codificado de entrada; o

20 un flujo de muestras de audio, sin un flujo correspondiente de metadatos o bits de control, determinados a partir de metadatos. En este último caso, el descodificador puede extraer metadatos a partir del flujo de bits codificado de entrada y realizar al menos una operación en los metadatos extraídos (por ejemplo, validación), aun cuando no da la salida los metadatos extraídos o bits de control determinados a partir de los mismos.

25 Mediante la configuración de la unidad de procesamiento posterior de la Figura 1 de conformidad con una forma de realización típica de la presente invención, la unidad de procesamiento posterior está configurada para aceptar un flujo de muestras de audio de PCM descodificadas, y para realizar un procesamiento posterior sobre el mismo (por ejemplo, nivelación de volumen del contenido de audio) utilizando SSM y/o PIM (y, normalmente, también otros metadatos, por ejemplo, LPSM), recibidos con las muestras, o bits de control determinados por el descodificador a partir de los metadatos recibidos con las muestras. La unidad de post-procesamiento está configurada normalmente, además, para interpretar el contenido de audio procesado posteriormente para su reproducción por uno o más altavoces.

30 Formas de realización típicas de la presente invención proporcionan una cadena de procesamiento de audio mejorada, en la que unidades de procesamiento de audio (por ejemplo, codificadores, descodificadores, transcodificadores y unidades de procesamiento previo y posterior), adaptan su respectivo procesamiento para aplicarlo a datos de audio de conformidad con un estado contemporáneo de los datos del medio, según lo indicado por los metadatos recibidos, respectivamente, por las unidades de procesamiento de audio.

35 La entrada de datos de audio a cualquier unidad de procesamiento de audio del sistema de la Figura 1 (por ejemplo, el codificador o transcodificador de la Figura 1) puede incluir SSM y/o PIM (y, opcionalmente, también otros metadatos) así como datos de audio (por ejemplo, datos de audio codificados). Estos metadatos pueden haber sido incluidos en el audio de entrada por otro elemento del sistema de la Figura 1 (u otra fuente, no ilustrada en la Figura 1), de conformidad con una forma de realización de la presente invención. La unidad de procesamiento que recibe el audio de entrada (con metadatos) se puede configurar para realizar al menos una operación en los metadatos (por ejemplo, validación), o en respuesta a los metadatos (por ejemplo, procesamiento adaptativo del audio de entrada) y normalmente incluir además en su audio de salida los metadatos, una versión procesada de los metadatos, o bits de control determinados a partir de los metadatos.

45 Una forma de realización típica de la unidad de procesamiento de audio inventiva (o procesador de audio) está configurada para realizar un procesamiento adaptativo de datos de audio, sobre la base del estado de los datos de audio, tal como se indica por los metadatos correspondientes a los datos de audio. En algunas formas de realización, el procesamiento adaptativo es, (o incluye) procesamiento de sonoridad (si los metadatos indican que el procesamiento de sonoridad, o procesamiento similar al mismo, aún no se ha realizado en los datos de audio, pero no es (y no incluye) procesamiento de sonoridad (si los metadatos indican que dicho procesamiento de sonoridad, o procesamiento similar al mismo, ya se ha realizado en los datos de audio). En algunas formas de realización, el procesamiento adaptativo es, o incluye, la validación de metadatos (por ejemplo, realizada en una sub-unidad de validación de metadatos) con el fin de garantizar que la unidad de procesamiento de audio realice otro procesamiento adaptativo de los datos de audio sobre la base del estado de los datos de audio, según lo indicado por los metadatos. En algunas formas de realización, la validación determina la fiabilidad de los metadatos asociados (por ejemplo, incluidos en un flujo de bits) con los datos de audio. Por ejemplo, si los metadatos se validan para ser fiables, en ese caso los resultados de un tipo de procesamiento de audio previamente realizado se pueden reutilizar y se puede evitar una nueva ejecución del mismo tipo de procesamiento de audio. Por otro lado, si se descubre que los metadatos han sido manipulados (o de otro modo, no son fiables) entonces, se puede repetir el tipo de procesamiento multimedia supuestamente realizado con anterioridad por la unidad de procesamiento de audio sobre los metadatos y/o los datos

de audio. La unidad de procesamiento de audio puede, además, estar configurada para señalar a otras unidades de procesamiento de audio, aguas abajo en una cadena de procesamiento del medio mejorada, que los metadatos (por ejemplo, presentes en un flujo de bits del medio) son válidos, si la unidad determina que los metadatos son válidos (por ejemplo, sobre la base de una coincidencia de un valor criptográfico extraído y un valor criptográfico de referencia).

5 La Figura 2 es un diagrama de bloques de un codificador (100) que es una forma de realización de la unidad de procesamiento de audio inventiva. Cualquiera de los componentes o elementos del codificador 100 se puede poner en práctica como uno o más procesos y/o uno o más circuitos (por ejemplo, ASICs, FPGAs u otros circuitos integrados) en hardware, software o una combinación de hardware y software. El codificador 100 comprende una memoria intermedia de trama 110, un analizador sintáctico 111, un descodificador 101, un validador de estado de audio 102,
10 una etapa de procesamiento de sonoridad 103, una etapa de selección de flujo de audio 104, un codificador 105, una etapa de rellenador/formateador 107, una etapa de generación de metadatos 106, un subsistema 108 de medición de sonoridad de diálogo y una memoria intermedia de trama 109, conectados según se ilustra. Normalmente, también el codificador 100 incluye otros elementos de procesamiento (no ilustrados).

15 El codificador 100 (que es un transcodificador) está configurado para convertir un flujo de bits de audio de entrada (que, por ejemplo, puede ser uno de entre un flujo de bits en AC-3, un flujo de bits en E-AC-3 o un flujo de bits en Dolby E), en un flujo de bits de audio de salida codificado (que, por ejemplo, puede ser otro de entre un flujo de bits en AC-3, un flujo de bits en E-AC-3 o un flujo de bits en Dolby E) que incluye la realización de un procesamiento de sonoridad automatizado y adaptativo, utilizando metadatos de estado de procesamiento de sonoridad incluidos en el flujo de bits de entrada. Por ejemplo, el codificador 100 se puede configurar para convertir un flujo de bits en Dolby E
20 de entrada (un formato normalmente utilizado en instalaciones de producción y difusión, pero no en dispositivos de consumo que reciben programas de audio que han sido difundidos) a un flujo de bits de audio codificado de salida (adecuado para la difusión a dispositivos de consumo) en formato AC-3 o E-AC-3.

25 El sistema de la Figura 2 incluye, además, el subsistema 150 de entrega de audio codificado (que memoriza y/o entrega, los flujos de bits codificados procedentes del codificador 100) y el descodificador 152. El subsistema 150 puede memorizar un flujo de bits de audio codificado, de salida, procedente del codificador 100 (por ejemplo, en la forma de un DVD o disco de Blu-ray), o transmitido por el subsistema 150 (que puede poner en práctica un enlace o red de transmisión), o puede ser, a la vez, memorizado y transmitido por el subsistema 150. El descodificador 152 está configurado para descodificar un flujo de bits de audio codificado (generado por el codificador 100), que se recibe a través del subsistema 150, incluyendo mediante la extracción de metadatos (PIM y/o SSM y, opcionalmente, también
30 metadatos de estado de procesamiento de sonoridad y/u otros metadatos) desde cada trama del flujo de bits (y, opcionalmente, extrayendo, además, metadatos de límite de programa del flujo de bits), y generando datos de audio descodificados. En condiciones normales, el descodificador 152 está configurado para realizar un procesamiento adaptativo sobre los datos de audio descodificados utilizando PIM y/o SSM, y/o LPSM (y, de modo opcional, también metadatos de límite de programa), y/o para enviar los datos de audio descodificados y los metadatos a un post-procesador configurado para realizar un procesamiento adaptativo en los datos de audio descodificados utilizando los metadatos. Normalmente, el descodificador 152 incluye una memoria intermedia que memoriza (por ejemplo, de
35 manera no transitoria) el flujo de bits de audio codificado, que se recibe desde el subsistema 150.

40 Diversas ejecuciones prácticas del codificador 100 y del descodificador 152 están configuradas para realizar diferentes formas de realización del método inventivo. La memoria intermedia de trama 110 es una memoria intermedia acoplada para recibir un flujo de bits de audio de entrada codificado. En funcionamiento, la memoria intermedia 110 memoriza (por ejemplo, de forma no transitoria), al menos una trama del flujo de bits de audio codificado, y una secuencia de las tramas del flujo de bits de audio codificada se reafirma desde la memoria intermedia 110 al analizador 111.

45 El analizador 111 está acoplado y configurado para extraer PIM y/o SSM, y metadatos de estado de procesamiento de sonoridad (LPSM) y, opcionalmente, también metadatos de límite de programa (y/u otros metadatos) desde cada trama del audio de entrada codificado en donde están incluidos dichos metadatos, para establecer al menos el LPSM (y, opcionalmente, también metadatos de límite de programa y/u otros metadatos) al validador de estado de audio 102, etapa de procesamiento de sonoridad 103, etapa 106 y subsistema 108, para extraer datos de audio del audio de entrada codificado y para llevar los datos de audio al descodificador 101. El descodificador 101, del codificador
50 100, está configurado para descodificar los datos de audio para generar datos de audio descodificados, y para llevar los datos de audio descodificados a la etapa de procesamiento de sonoridad 103, a la etapa de selección de flujo de audio 104, al subsistema 108, y normalmente, también al validador de estado 102.

55 El validador de estado 102 está configurado para autenticar y validar el LPSM (y, opcionalmente, otros metadatos) establecido en el mismo. En algunas formas de realización, el LPSM es, (o está incluido en), un bloque de datos que se ha incluido en el flujo de bits de entrada (por ejemplo, de conformidad con una forma de realización de la presente invención). El bloque puede incluir un denominado *hash* criptográfico (un código de autenticación de mensaje basado en *hash* o "HMAC") para el procesamiento de los LPSM (y, opcionalmente, también otros metadatos) y/o los datos de audio subyacentes (que se proporcionan a partir del descodificador 101, al validador 102). El bloque de datos puede estar firmado digitalmente en estas formas de realización, de modo que una unidad de procesamiento de audio de flujo del pueda autenticar y validar, con relativa facilidad, los metadatos del estado de procesamiento.

60 A modo de ejemplo, el HMAC se usa para generar un *digest*, y el o los valores de protección incluidos en el flujo de

bits de la invención pueden incluir el *digest*. El *digest* se puede generar del modo siguiente para una trama en AC-3:

1. Después de que se codifiquen los datos de AC-3 y LPSM, bytes de datos de trama (concatenadas *frame_data* (trama_datos) #1 y *frame_data* #2) y los bytes de datos de LPSM se utilizan como entrada para la función de hash HMAC. Otros datos, que pueden estar presentes dentro de un campo *auxdata*, no se tienen en cuenta para calcular el *digest*. Dichos otros datos pueden ser bytes que no pertenezcan a los datos en AC-3, ni a los datos de LSPSM. Bits de protección, incluidos en LPSM, pueden no ser considerados para el cálculo del *digest* de HMAC.

2. Después de calcular el *digest*, se escribe en el flujo de bits en un campo reservado para bits de protección.

3. La última etapa de la generación de la trama en AC-3 completa es el cálculo de la verificación de CRC. Esto se escribe en el mismo extremo de la trama, y se tienen en cuenta todos los datos que pertenecen a esta trama, incluidos los bits de LPSM.

Otros métodos criptográficos que incluyen, pero no se limitan a, uno o más métodos criptográficos no de HMAC, se pueden utilizar para la validación de LPSM y/u otros metadatos (por ejemplo, en el validador 102) con el fin de garantizar la transmisión y recepción seguras de los metadatos y/o los datos de audio subyacentes. Por ejemplo, la validación (utilizando dicho método criptográfico) se puede realizar en cada unidad de procesamiento de audio que recibe, en una forma de realización de la invención, el flujo de bits de audio para determinar si los metadatos y los datos de audio correspondientes, incluidos en el flujo de bits, se han sometido (y/o son el resultado del) al procesamiento específico (como lo indican los metadatos) y no se han modificado después de la realización de dicho procesamiento específico.

El validador de estado 102 establece datos de control para la etapa de selección de flujo de audio 104, el generador de metadatos 106 y el subsistema de medición de sonoridad de diálogo 108, para indicar los resultados de la operación de validación. En respuesta a los datos de control, la etapa 104 puede seleccionar (y pasar a través del codificador 105), ya sea:

la salida procesada adaptativamente de la etapa de procesamiento de sonoridad 103 (por ejemplo, cuando LPSM indican que la salida de datos de audio desde el descodificador 101 no ha experimentado un tipo específico de procesamiento de sonoridad, y los bits de control procedentes del validador 102 indican que los LPSM son válidos); o

los datos de audio de salida del descodificador 101 (por ejemplo, cuando LPSM indica que los datos de audio, a la salida del descodificador 101, ya ha experimentado el tipo específico de procesamiento de sonoridad que se realizaría en la etapa 103, y los bits de control del validador 102 indican que los LPSM son válidos).

La etapa 103 del codificador 100 está configurada para realizar un procesamiento de sonoridad adaptativo sobre la salida de datos de audio descodificados procedentes del descodificador 101, sobre la base de una o más características de datos de audio que se indican por LPSM, extraídas por el descodificador 101. La etapa 103 puede ser un procesador de dominio-transformación adaptativo de sonoridad en tiempo real y de control de margen dinámico. La etapa 103 puede recibir una entrada del usuario (por ejemplo valores del margen de sonoridad objetivo /margen dinámico del usuario o valores de *dialnorm*) u otra entrada de metadatos (por ejemplo, uno o más tipos de datos de terceros, información de seguimiento, identificadores, información de propiedad o estándar, datos de anotación del usuario, datos de preferencia del usuario, etc.) y/u otra entrada (por ejemplo, desde un proceso de huella dactilar), y utilizar dicha entrada para procesar la salida de datos de audio descodificados, procedentes del descodificador 101. La etapa 103 puede realizar un procesamiento de sonoridad adaptativo sobre datos de audio descodificados (salida del descodificador 101), indicativo de un único programa de audio (como es indicado por los metadatos de límite de programa extraídos por el analizador sintáctico 111) y puede reiniciar el procesamiento de sonoridad en respuesta a la recepción de datos de audio descodificados (salida del descodificador 101), que indican un programa de audio diferente, tal como es indicado por metadatos de límite de programa extraídos por el analizador sintáctico 111.

El subsistema 108 de medición de sonoridad de diálogo puede funcionar para determinar la sonoridad de segmentos del audio descodificado (procedente del descodificador 101) que son indicativos de diálogo (u otra expresión vocal), por ejemplo utilizando LPSM (y/u otros metadatos) extraídos por el descodificador 101, cuando los bits de control del validador 102 indican que los LPSM no son válidos. La operación del subsistema 108 de medición de sonoridad de diálogo puede desactivarse cuando los LPSM indican previamente una determinada sonoridad de segmentos de diálogo (u otra expresión vocal) del audio descodificado (procedente del descodificador 101) cuando los bits de control del validador 102 indican que los LPSM son válidos. El subsistema 108 puede realizar una medición de sonoridad en datos de audio descodificados, indicativos de un único programa de audio (según lo indicado por los metadatos de límites del programa extraídos por el analizador 111), y puede reiniciar la medición en respuesta a la recepción de datos de audio descodificados indicativos de un programa de audio diferente, según lo indicado por tales metadatos de límites del programa.

Existen herramientas útiles (por ejemplo el medidor de sonoridad en Dolby LM100) para medir el nivel del diálogo en contenido de audio, de manera conveniente y fácil. Algunas formas de realización de la APU inventiva (por ejemplo, etapa 108 del codificador 100) se ponen en práctica para incluir (o para realizar las funciones de) dicha herramienta para medir la sonoridad media de diálogo del contenido de audio de un flujo de bits de audio (por ejemplo, un flujo de bits en AC-3 descodificado, establecido en la etapa 108 del descodificador 101, del codificador 100).

- Si la etapa 108 se ejecuta para medir la sonoridad de diálogo media real de los datos de audio, la medición puede incluir una etapa de aislamiento de segmentos del contenido de audio que contienen, de forma predominante, voz. Los segmentos de audio predominantemente hablados se procesan entonces de conformidad con un algoritmo de medición de sonoridad. Para datos de audio descodificados a partir de un flujo de bits en AC-3, este algoritmo puede ser una medida de sonoridad ponderada K estándar (de conformidad con la norma internacional ITU-R BS.1770). Como alternativa, se pueden utilizar otras medidas de sonoridad (por ejemplo, las basadas en modelos psico-acústicos de sonoridad).
- El aislamiento de los segmentos de voz no es esencial para medir la sonoridad de diálogo media de los datos de audio. Sin embargo, mejora la precisión de la medida y, por lo general, proporciona resultados más satisfactorios desde la perspectiva del oyente. Puesto que no todo el contenido de audio incluye diálogo (expresión vocal), la medición de sonoridad del contenido de audio completo puede proporcionar una aproximación suficiente del nivel de diálogo del audio, si la expresión vocal hubiera estado presente.
- El generador de metadatos 106 genera (y/o pasa a través de la etapa 107) metadatos para ser incluidos por la etapa 107 en el flujo de bits codificado, a proporcionarse, para ser emitidos desde el codificador 100. El generador de metadatos 106 puede pasar a través de la etapa 107 los LPSM (y además, opcionalmente, LIM y/o PIM y/o metadatos de límite de programa y/u otros metadatos) extraídos por el codificador 101 y/o el analizador sintáctico 111 (por ejemplo cuando los bits de control procedentes del validador 102 indican que los LPSM y/u otros metadatos son válidos), o generar nuevos LIM y/o PIM y/o LPSM y/o metadatos de límite de programa y/u otros metadatos y establecer los nuevos metadatos para la etapa 107 (por ejemplo, cuando los bits de control del validador 102 indican que los metadatos extraídos por el descodificador 101 no son válidos) o puede establecer, en la etapa 107, una combinación de metadatos extraídos por el descodificador 101 y/o el analizador 111, y los metadatos recientemente generados. El generador de metadatos 106 puede incluir datos de sonoridad generados por el subsistema 108, y al menos un valor indicativo del tipo de procesamiento de sonoridad realizado por el subsistema 108, en LPSM que son establecidos en la etapa 107 para su inclusión en el flujo de bits codificado que se enviará desde el codificador 100.
- El generador de metadatos 106 puede generar bits de protección (que pueden consistir en, o incluir, un código de autenticación de mensaje basado en el *hash*, o "HMAC") útil para al menos una de entre las funciones de descifrado, autenticación o validación de los LPSM (y, opcionalmente, también otros metadatos) que han de incluirse en el flujo de bits codificado, y/o los datos de audio subyacentes que han de incluirse en el flujo de bits codificado. El generador de metadatos 106 puede proporcionar dichos bits de protección a la etapa 107 para su inclusión en el flujo de bits codificado.
- En una operación típica, el subsistema 108 de medición de sonoridad de diálogo procesa los datos de audio de salida del descodificador 101, para generar en respuesta a sus valores de sonoridad (por ejemplo, valores de sonoridad de diálogo bloqueados y no bloqueados) y valores de margen dinámico. En respuesta a estos valores, el generador de metadatos 106 puede generar metadatos de estado de procesamiento de sonoridad (LPSM) para su inclusión (por el relleno/formateador 107) en el flujo de bits codificado que se enviará desde el codificador 100.
- De forma adicional, opcionalmente o como alternativa, los subsistemas de 106 y/o 108 del codificador 100 pueden realizar un análisis adicional de los datos de audio para generar metadatos indicativos de al menos una característica de los datos de audio, para su inclusión en el flujo de bits codificado para darle salida desde la etapa 107.
- El codificador 105 codifica (por ejemplo, realizando una compresión) los datos de audio emitidos desde la etapa de selección 104, y establece el audio codificado a la etapa 107 para su inclusión en el flujo de bits codificado para darle salida, desde la etapa 107.
- La etapa 107 realiza la multiplexación del audio codificado procedente del codificador 105, y los metadatos (incluyendo PIM y/o SSM) del generador 106, para generar el flujo de bits codificado que se emitirá desde la etapa 107, preferiblemente de modo que el flujo de bits codificado tiene el formato que se especifica mediante una forma de realización preferida de la presente invención.
- La memoria intermedia de trama 109 es una memoria intermedia que memoriza, (por ejemplo, de forma no transitoria), al menos una trama del flujo de bits de audio codificado a la salida de la etapa 107 y, a continuación, una secuencia de las tramas del flujo de bits de audio codificado se establecida desde la memoria intermedia 109 como salida del codificador 100 al sistema de entrega 150.
- LPSM generados por el generador de metadatos 106, e incluidos en el flujo de bits codificado por la etapa 107, son normalmente indicativos del estado de procesamiento de sonoridad de los datos de audio correspondientes (por ejemplo, qué tipos de procesamientos de sonoridad se han realizado sobre los datos de audio) y sonoridad (por ejemplo, sonoridad de diálogo medida, sonoridad bloqueada y/o no bloqueada, y/o margen dinámico) de los correspondientes datos de audio.
- En este caso, "efecto compuerta" ("gating") de sonoridad y/o mediciones de nivel realizadas en datos de audio, se refiere a un nivel específico o umbral de sonoridad, en donde el o los valores calculados que superen el umbral se incluyen en la medición final (por ejemplo, ignorando los valores de sonoridad a corto plazo por debajo de -60 dBFS en los valores de medición final). La función *gating* aplicada sobre un valor absoluto se refiere a un nivel o sonoridad

fijo, mientras que dicha misma función aplicada a un valor relativo se refiere a un valor que depende de un valor de medición "sin gating" actual.

5 En algunas ejecuciones en prácticas del codificador 100, el flujo de bits codificado que se almacena en memoria intermedia 109 (y se proporciona como salida al sistema de distribución 150), es un flujo de bits en AC-3 o un flujo de bits en E-AC-3, y comprende segmentos de datos de audio (por ejemplo, los segmentos AB0-AB5 de la trama ilustrada en la Figura 4) y segmentos de metadatos, en donde los segmentos de datos de audio son indicativos de datos de audio, y cada uno de al menos algunos de los segmentos de metadatos incluyen PIM y/o SSM (y, opcionalmente, también otros metadatos). La etapa 107 establece segmentos de metadatos (que incluyen metadatos) en el flujo de bits en el siguiente formato. Cada uno de los segmentos de metadatos que incluye PIM y/o SSM se incluye en un segmento de bits residuales del flujo de bits (por ejemplo, un segmento de bits residuales "W", tal como se muestra en la Figura 4 o Figura 7) o un campo "addbsi" del segmento de Información de Flujo De bits ("BSI") de una trama del flujo de bits, o en un campo auxdata (por ejemplo, el segmento AUX, ilustrado en la Figura 4 o la Figura 7) al final de una trama del flujo de bits. Una trama del flujo de bits puede incluir uno o dos segmentos de metadatos, cada uno de los cuales incluya metadatos, y si la trama incluye dos segmentos de metadatos, uno puede estar presente en el campo addbsi de la trama y el otro en el campo AUX de la trama.

20 En algunas formas de realización, cada segmento de metadatos (a veces referido aquí como un "contenedor") insertado por la etapa 107 tiene un formato que incluye una cabecera de segmento de metadatos (y, opcionalmente, también otros elementos obligatorios o "principales"), y una o más cargas útiles de metadatos, siguiendo la cabecera del segmento de metadatos. SIM, si está presente, se incluye en una de las cargas útiles de metadatos (identificada por una cabecera de carga útil y, por lo general, tiene un formato de un primer tipo). PIM, se está presente, está incluido en otra de las cargas útiles de metadatos (identificada por una cabecera de carga útil y que, típicamente, tiene un formato de un segundo tipo). De modo similar, cada otro tipo de metadatos (si está presente) se incluye en otra de las cargas útiles de metadatos (que se identifica por una cabecera de carga y, normalmente, tiene un formato específico para el tipo de metadatos). El formato de ejemplo permite un acceso cómodo a los SSM, PIM y otros metadatos en momentos distintos a la duración de la descodificación (por ejemplo, mediante un post-procesador que sigue la descodificación, o mediante un procesador configurado para reconocer los metadatos sin realizar una descodificación completa en el flujo de bits codificado), y permite la detección y corrección de errores, cómoda y eficiente, (por ejemplo, de identificación del flujo secundario) durante la descodificación del flujo de bits. A modo de ejemplo, sin acceso a SSM en el formato de ejemplo, un descodificador podría identificar, de forma incorrecta, el número correcto de flujos secundarios asociados con un programa. Una carga útil de metadatos, en un segmento de metadatos, puede incluir SSM, otra carga útil de metadatos, en el segmento de metadatos, puede incluir PIM y, opcionalmente, además al menos otra carga útil de metadatos en el segmento de metadatos puede incluir otros metadatos (por ejemplo, metadatos de estado de procesamiento de sonoridad o "LPSM").

35 En algunas formas de realización, una carga útil de metadatos de estructura de flujo secundario (SSM) incluida (por la etapa 107), en una trama de un flujo de bits codificado (por ejemplo, un flujo de bits en E-AC-3 indicativo de al menos un programa de audio), incluye SSM en el siguiente formato:

una cabecera de carga útil, que incluye normalmente al menos un valor de identificación (por ejemplo, un valor de 2 bits indicativo de la versión de formato SSM y, de forma opcional, también valores de longitud, período, conteo y de asociación de flujos secundario); y después de la cabecera:

40 metadatos de flujo secundario independientes, que indican el número de flujos secundarios independientes del programa indicado por el flujo de bits; y

45 metadatos de flujo secundario dependientes, que indican si cada flujo secundario independiente del programa tiene al menos un flujo secundario dependiente asociado (es decir, si al menos un flujo secundario dependiente está asociado con dicho flujo secundario independiente) y, de ser así, el número de flujos secundarios dependientes asociados con cada flujo secundario independiente del programa.

50 Se contempla que un flujo secundario independiente, de un flujo de bits codificado, puede ser indicativo de un conjunto de canales de altavoz de un programa de audio (por ejemplo, los canales de altavoz de un programa de audio de canal de altavoz 5.1), y que cada uno de entre los uno o más flujos secundarios dependientes (asociados con el flujo secundario independiente, tal como se indica por los metadatos de flujo secundario dependiente) puede ser indicativo de un canal de objeto del programa. Normalmente, sin embargo, un flujo secundario independiente de un flujo de bits codificado, es indicativo de un conjunto de canales de altavoz de un programa, y cada flujo secundario dependiente, asociado con el flujo secundario independiente (tal como lo indican los metadatos de flujo secundario dependiente) es indicativo de al menos un canal de altavoz adicional del programa.

55 En algunas formas de realización, una carga útil de metadatos de información sobre el programa (PIM), incluida (por la etapa 107) en una trama de un flujo de bits codificado (por ejemplo, un flujo de bits en E-AC-3 indicativo de al menos un programa de audio) tiene el siguiente formato:

una cabecera de carga útil, que suele incluir, al menos un valor de identificación (por ejemplo, un valor indicativo de la versión de formato PIM y, de forma opcional, también valores de asociación de longitud, período, conteo y flujo

secundario); y

después de la cabecera, PIM en el siguiente formato:

5 metadatos de canales activos, indicativos de cada canal silencioso y cada canal no silencioso, de un programa de audio (es decir, qué canal(es) del programa contiene(n) información de audio y cuáles (si los hay) contienen solamente silencio (normalmente durante la duración de la trama)). En formas de realización en las que el flujo de bits codificado es un flujo de bits en AC-3 o E-AC-3, los metadatos de canal activo, en una trama del flujo de bits, se pueden utilizar junto con metadatos adicionales del flujo de bits (por ejemplo, el campo de modo de codificación de audio ("acmod") de la trama y, si está presente, el campo 'chanmap' en la trama o las tramas de flujo secundario dependiente asociadas) para determinar qué canal(es) del programa contiene(n) información de audio y cuáles contienen silencio.

10 El campo "acmod" de una trama en AC-3 o E-AC-3 indica el número de canales de margen completo de un programa de audio, indicado por el contenido de audio de la trama (por ejemplo, si el programa es un programa monofónico de canal 1.0, un programa estéreo de canal 2.0, o un programa que comprende canales de margen completo L, R, C, Ls, Rs), o que la trama es indicativa de dos programas monofónicos de canal 1.0 independientes. Un campo "chanmap" de un flujo de bits en E-AC-3 indica un mapa de canales para un flujo secundario dependiente, indicado por el flujo de bits. Los metadatos de canal activo pueden ser útiles para poner en práctica la mezcla ascendente (en un post-procesador) aguas abajo de un descodificador, por ejemplo para añadir audio a canales que contienen silencio en la salida del descodificador;

los metadatos de estado de procesamiento de mezcla descendente indican si el programa fue objeto de mezcla descendente (antes o durante la codificación), y si es así, el tipo de operación de mezcla descendente que se aplicó.

20 Los metadatos de estado de procesamiento de mezcla descendente pueden ser útiles para poner en práctica una mezcla ascendente (en un post-procesador) aguas abajo de un descodificador, por ejemplo para la mezcla ascendente del contenido de audio del programa utilizando los parámetros que más se asemejan a un tipo de operación de mezcla descendente que se aplicó. En formas de realización en las que el flujo de bits codificado es un flujo de bits en AC-3 o E-AC-3, los metadatos de estado de procesamiento de mezcla descendente pueden utilizarse junto con el campo de modo de codificación de audio ("acmod") de la trama para determinar el tipo de operación de mezcla descendente (si corresponde) aplicado al(a los) canal(es) del programa;

25

metadatos de estado de procesamiento de mezcla ascendente, indicativos de si el programa fue objeto de mezcla ascendente (por ejemplo de un número menor de canales) antes, o durante, la codificación y, de ser así, el tipo de mezcla ascendente que se aplicó. Los metadatos de estado de procesamiento de mezcla ascendente pueden ser útiles para poner en práctica una operación de mezcla descendente (en un post-procesador) aguas abajo de un descodificador, por ejemplo para la mezcla descendente del contenido de audio del programa de un modo que sea compatible con un tipo de operación de mezcla ascendente (por ejemplo, Dolby Pro Logic, o Dolby Pro Logic II Movie Mode, o Dolby Pro Logic II Music Mode, o Dolby Professional Upmixer) que se aplicó al programa. En formas de realización en las que el flujo de bits codificado es un flujo de bits en E-AC-3, los metadatos de estado de procesamiento de mezcla ascendente se pueden utilizar junto con otros metadatos (por ejemplo, el valor de un campo "strmtyp" de la trama), para determinar el tipo de operación de mezcla ascendente (si corresponde) aplicado al (a los) canal(es) del programa. El valor del campo "strmtyp" (en el segmento BSI de una trama de un flujo de bits en E-AC-3) indica si el contenido de audio de la trama pertenece a un flujo independiente (que determina un programa) o un flujo secundario independiente (de un programa que incluye, o está asociado con, múltiples flujos secundarios) y por lo tanto, puede descodificarse independientemente de cualquier otro flujo secundario indicado por el flujo de bits en E-AC-3, o si el contenido de audio de la trama pertenece a un flujo secundario dependiente (de un programa que incluye, o está asociado con, múltiples flujos secundarios) y, en consecuencia, debe descodificarse junto con un flujo secundario independiente con el que está asociado; y

30

35

40

45 metadatos de estado de pre-procesamiento, que indican si el pre-procesamiento se realizó en el contenido de audio de la trama (antes de la codificación del contenido de audio para generar el flujo de bits codificado) y, en ese caso, el tipo de pre-procesamiento que se realizó.

En algunas realizaciones, los metadatos del estado de pre-procesamiento son indicativos de:

si se aplicó una atenuación circundante (por ejemplo, si los canales circundantes del programa de audio fueron atenuados en 3 dB antes de la codificación),

50 si se aplicó un cambio de fase de 90 grados (por ejemplo, para canales Ls circundantes y canales Rs del programa de audio antes de la codificación),

si se aplicó un filtro de paso bajo a un canal de LFE del programa de audio antes de la codificación,

si el nivel de un canal de LFE del programa fue supervisado durante la producción y, de ser así, el nivel supervisado del canal de LFE relativo al nivel de los canales de audio de margen completo del programa,

55 si se debe realizar una compresión de margen dinámico (por ejemplo, en el descodificador) en cada bloque de contenido de audio descodificado del programa y, si es así, el tipo (y/o parámetros) de compresión de margen dinámico que ha de realizarse (por ejemplo, este tipo de estado de pre-procesamiento los metadatos pueden ser indicativo de

- cuál de los siguientes tipos de perfil de compresión fue asumido por el codificador para generar valores de control de compresión de margen dinámico que están incluidos en el flujo de bits codificado: Clase de Película, Luz de película, Clase de Música, Luz de Música o Voz. Como alternativa, este tipo de metadatos de estado de pre-procesamiento puede indicar que ha de realizarse la compresión de margen dinámico intensa (compresión "compr") en cada trama de contenido de audio descodificado del programa, de un modo determinado por los valores de control de compresión de margen dinámico que se incluyen en el flujo de bits codificado).
- si el procesamiento de extensión espectral y/o codificación de acoplamiento de canal se empleó para codificar márgenes de frecuencia específicos del contenido del programa y, en caso afirmativo, las frecuencias mínima y máxima de los componentes de frecuencia del contenido en que se realizó la codificación de extensión espectral, y las frecuencias mínima y máxima de componentes de frecuencia del contenido en el que se realizó la codificación de acoplamiento de canal. Este tipo de información de metadatos de estado de pre-procesamiento puede ser útil para realizar la igualación (en un post-procesador) aguas abajo de un descodificador. Tanto el acoplamiento de canales como la información de extensión espectral son útiles, además, para optimizar la calidad durante las operaciones y aplicaciones de trans-codificación. Por ejemplo, un codificador puede optimizar su comportamiento (incluida la adaptación de etapas de pre-procesado, como la virtualización de auriculares, operación de mezcla ascendente, etc.) sobre la base del estado de los parámetros, tales como la extensión espectral y la información de acoplamiento de canal. Además, el codificador podría adaptar sus parámetros de acoplamiento y extensión espectral de forma dinámica para que coincidan y/o para valores óptimos, en función del estado de los metadatos entrantes (y autenticados), y
- si los datos de margen de ajuste de mejora de diálogo se incluyen en el flujo de bits codificado y, si es así, el margen de ajuste disponible durante la realización del procesamiento de mejora de diálogo (por ejemplo, en un post-procesador de aguas abajo de un descodificador) para ajustar el nivel de contenido de diálogo relativo al nivel de contenido sin diálogo en el programa de audio.
- En algunas ejecuciones prácticas, metadatos de estado de pre-procesamiento adicionales (por ejemplo, metadatos indicativos de parámetros relacionados con auriculares) se incluyen (por la etapa 107) en una carga útil de PIM de un flujo de bits codificado para darles salida desde el codificador 100.
- En algunas formas de realización, una carga útil de LPSM incluida (por la etapa 107) en una trama de un flujo de bits codificado (por ejemplo, un flujo de bits en E-AC-3, indicativo de al menos un programa de audio) incluye LPSM en el siguiente formato:
- una cabecera (que incluye normalmente una palabra de sincronización que identifica el inicio de la carga útil de LPSM, seguida por al menos un valor de identificación, por ejemplo la versión de formato de LPSM, longitud, período, conteo y valores de asociación flujo secundario indicados en la Tabla 2 siguiente); y
- después de la cabecera,
- al menos un valor de indicación de diálogo (por ejemplo parámetro "Canal(es) de Diálogo" de la Tabla 2), que indica si correspondientes datos de audio indican diálogo, o no indican diálogo, (por ejemplo, qué canales de los correspondientes datos de audio indican diálogo);
- al menos un valor de cumplimiento de normativa de sonoridad (por ejemplo parámetro "Tipo de Normativa de Sonoridad" de la Tabla 2), que indica si los correspondientes datos de audio cumplen con un conjunto indicado de normativas de sonoridad;
- al menos un valor de procesamiento de sonoridad (por ejemplo uno o más de los parámetros "Indicador de corrección de sonoridad bloqueada por diálogo", "Tipo de Corrección de Sonoridad" de la Tabla 2), que indica al menos un tipo de procesamiento de sonoridad que se ha realizado en los correspondientes datos de audio; y
- al menos un valor de sonoridad (por ejemplo uno o más de los parámetros "Sonoridad bloqueada relativa de ITU", "Sonoridad bloqueada de expresión vocal de ITU", "Sonoridad de corto plazo de 3s de ITU (EBU 3341)" y "Pico verdadero" de la Tabla 2), que indican al menos una característica de sonoridad (por ejemplo, pico o promedio de sonoridad) de los datos de audio correspondientes.
- En algunas formas de realización, cada segmento de metadatos que contiene PIM y/o SSM (y opcionalmente, también otros metadatos) contiene una cabecera de segmento de metadatos (y, como opción, elementos principales adicionales) y después de la cabecera del segmento de metadatos (o la cabecera del segmento de metadatos y otros elementos principales), al menos un segmento de carga útil de metadatos que tiene el siguiente formato:
- una cabecera de carga útil, que incluye normalmente al menos un valor de identificación (por ejemplo, versión de formato SSM o PIM, longitud, periodo, conteo y valores de asociación de flujo secundario), y
- después de la cabecera de la carga útil, los SSM o PIM (o metadatos de otro tipo).
- En algunas realizaciones, cada uno de los segmentos de metadatos (a veces denominados en este documento como

"contenedores de metadatos" o "contenedores") insertados en la etapa 107 en un segmento de campo de omisión/bit residual (o un campo "addbsi", o un campo auxdata) de una trama del flujo de bits que tiene el siguiente formato:

5 una cabecera de segmento de metadatos (que incluye normalmente una palabra de sincronización que identifica el inicio del segmento de metadatos, seguida por valores de identificación, por ejemplo, versión, longitud, período, conteo de elementos expandidos, y valores de asociación de flujo secundario, tal como se indica en la Tabla 1 siguiente); y

después de la cabecera del segmento de metadatos, al menos un valor de protección (por ejemplo, los valores de HMAC digest y de huella dactilar de audio de la Tabla 1), útiles para al menos una función de entre descriptación, autenticación o validación de al menos uno de los metadatos del segmento de metadatos o los correspondientes datos de audio); y

10 también después de la cabecera de segmento de metadatos, identificación de carga útil de metadatos ("ID") y valores de configuración de carga útil, que identifican el tipo de metadatos en cada carga útil de metadatos, e indican al menos un aspecto de la configuración (por ejemplo tamaño) de cada carga útil.

Cada carga útil de metadatos sigue a los correspondientes valores de ID de carga útil y de configuración de carga útil.

15 En algunas formas de realización, cada uno de los segmentos de metadatos, en el segmento de bits residuales (o campo auxdata, o campo "addbsi") de una trama, tiene tres niveles de estructura:

20 una estructura de alto nivel (por ejemplo una cabecera de segmento de metadatos), que incluye un indicador que indica si el campo de bits residuales (o auxdata o addbsi) incluye metadatos, al menos un valor de ID que indica qué tipo(s) de metadatos está(n) presente(s) y normalmente también un valor que indica cuántos bits de metadatos (por ejemplo, de cada tipo) están presentes (si metadatos están presentes). Un tipo de metadatos que podría estar presente es PIM, otro tipo de metadatos que podría estar presente es SSM, y otros tipos de metadatos que podrían estar presentes son LPSM, y/o metadatos de límite de programa y/o metadatos de investigación de medios;

una estructura de nivel intermedio, que comprende datos asociados con cada tipo de metadatos identificado (por ejemplo valores de cabecera de carga útil de metadatos, de protección, y valores de ID de carga útil y de configuración de carga útil, para cada tipo de metadatos identificado); y

25 una estructura de nivel bajo, que comprende una carga útil de metadatos para cada tipo de metadatos identificado (por ejemplo una secuencia de valores de PIM, si se identifica PIM como estando presente, y/o valores de metadatos de otro tipo (por ejemplo, SSM o LPSM), si otro tipo de metadatos se identifica como estando presente).

30 Los valores de datos en dicha estructura de tres niveles pueden estar acoplados. Por ejemplo, los valores de protección para cada carga útil (por ejemplo cada PIM o SSM u otra carga útil de metadatos), que se identifican por las estructuras de nivel alto e intermedio, pueden incluirse después de la carga útil (y, por lo tanto, después de la cabecera de carga útil de metadatos de la carga útil), o el(los) valor(es) de protección para todas las cargas útiles de metadatos, identificados por las estructuras de nivel alto e intermedio pueden incluirse después de la carga útil de metadatos final en el segmento de metadatos (y, por lo tanto, después de las cabeceras de carga útil de metadatos de todas las cargas útiles del segmento de metadatos).

35 En un ejemplo (que se describirá con referencia al segmento de metadatos o "contenedor" de la Figura 8), una cabecera de segmento de metadatos identifica cuatro cargas útiles de metadatos. Según se ilustra en la Figura 8, la cabecera del segmento de metadatos incluye una palabra de sincronización de contenedor (identificada como "container sync") y valores de ID de versión y clave. La cabecera del segmento de metadatos es seguida por las cuatro cargas útiles de metadatos y bits de protección. Los valores de ID y configuración de carga útil (por ejemplo, tamaño de carga útil), para la primera carga útil (por ejemplo una carga útil de PIM), siguen a la cabecera de segmento de metadatos, la propia primera carga útil sigue los valores de ID y configuración, valores de ID de carga útil y configuración de carga útil (por ejemplo, tamaño de carga útil) para la segunda carga útil (por ejemplo una carga útil de SSM) siguen a la primera carga útil, la propia segunda carga útil sigue a estos valores de ID y configuración, los valores de ID de carga útil y configuración de carga útil (por ejemplo, tamaño de carga útil) para la tercera carga útil (por ejemplo, una carga útil de LPSM) siguen a la segunda carga útil, la propia tercera carga útil sigue a estos valores de ID y configuración, los valores de ID de carga útil y configuración de carga útil (por ejemplo, tamaño de carga útil) para la cuarta carga útil, siguen a la tercera carga útil, la propia cuarta carga útil sigue estos valores de ID y configuración, y los valores de protección (identificados como "Datos de Protección" en la Figura 8) para la totalidad o algunas de las cargas útiles (o para la estructura de niveles alto e intermedio, y la totalidad o algunas de las cargas útiles), siguen a la última carga útil.

55 En algunas formas de realización, si el descodificador 101 recibe un flujo de bits de audio generado de conformidad con una forma de realización de la invención con un *hash* criptográfico, el descodificador está configurado para analizar y recuperar el *hash* criptográfico desde un bloque de datos determinado a partir del flujo de bits, en donde dicho bloque incluye metadatos. El Validador 102 puede utilizar el *hash* criptográfico para validar el flujo de bits recibido y/o los metadatos asociados. A modo de ejemplo, si el validador 102 encuentra que los metadatos son válidos sobre la base

de una coincidencia entre un *hash* criptográfico de referencia y el *hash* criptográfico recuperado del bloque de datos, entonces, se puede inhabilitar la operación del procesador 103 en los datos de audio correspondientes y hacer que la etapa de selección 104 pase a través de (sin cambiar) los datos de audio. De modo adicional, como opción, o de forma alternativa, se pueden utilizar otros tipos de técnicas criptográficas en lugar de un método basado en un *hash* criptográfico.

El codificador 100 de la Figura 2 puede determinar (en respuesta a LPSM y opcionalmente también metadatos de límite de programa, extraídos por el descodificador 101) que una unidad de post/pre-procesamiento ha realizado un tipo de procesamiento de sonoridad sobre los datos de audio que han de codificarse (en los elementos 105, 106, y 107) y, por lo tanto, pueden crear (en el generador 106) metadatos de estado de procesamiento de sonoridad que incluyan los parámetros específicos utilizados y/o derivados del procesamiento de sonoridad realizado con anterioridad. En algunas ejecuciones prácticas, el codificador 100 puede crear (e incluir en la salida el flujo de bits codificado del mismo) metadatos indicativos del historial de procesamiento sobre el contenido de audio siempre que el codificador conozca los tipos de procesamiento que se han realizado sobre el contenido de audio.

La Figura 3 es un diagrama de bloques de un descodificador (200) que es una forma de realización de la unidad de procesamiento de audio inventiva, y de un post-procesador (300) acoplado al mismo. El post-procesador (300) es también una forma de realización de la unidad de procesamiento de audio inventiva. Cualquiera de los componentes o elementos del descodificador 200 y del post-procesador 300 se puede poner en práctica como uno o más procesos y/o uno o más circuitos (por ejemplo, ASICs, FPGAs u otros circuitos integrados), en hardware, software o una combinación de hardware y software. El descodificador 200 incluye la memoria intermedia de trama 201, el analizador sintáctico 205, el descodificador de audio 202, la etapa 203 de validación de estado de audio (validador), y la etapa 204 de generación de bits de control, conectados según se ilustra. Normalmente también, el descodificador 200 incluye otros elementos de procesamiento (no ilustrados).

La memoria intermedia de trama 201 (una memoria intermedia) memoriza (por ejemplo, de manera no transitoria) al menos una trama del flujo de bits de audio codificado, recibido por el descodificador 200. Una secuencia de las tramas del flujo de bits de audio codificado es establecida desde la memoria intermedia 201 al analizador 205.

El analizador sintáctico 205 está acoplado y configurado para extraer PIM y/o SSM (y, además, como opción, otros metadatos, por ejemplo, LPSM) a partir de cada trama del audio de entrada codificado, para establecer al menos algunos de los metadatos (por ejemplo, LPSM y metadatos de límite de programa, si se extrae alguno, y/o PIM y/o SSM) al validador de estado de audio 203 y la etapa 204, con el fin de establecer los metadatos extraídos como salida (por ejemplo, al post-procesador 300) para extraer datos de audio procedentes del audio de entrada codificado, y para establecer los datos de audio extraídos al descodificador 202.

La entrada del flujo de bits de audio codificado al descodificador 200 puede ser uno de entre un flujo de bits en AC-3, un flujo de bits en E-AC-3, o un flujo de bits en Dolby E.

El sistema de la Figura 3 incluye, además, el post-procesador 300. El post-procesador 300 comprende la memoria intermedia de trama 301 y otros elementos de procesamiento (no ilustrados), que incluyen al menos un elemento de procesamiento acoplado a la memoria intermedia 301. La memoria intermedia de trama 301 realiza la memorización (por ejemplo de forma no transitoria), en al menos una trama del flujo de bits de audio descodificado, recibido por el post-procesador 300 desde el descodificador 200. Los elementos de procesamiento del post-procesador 300 están acoplados y configurados para recibir y procesar, de forma adaptativa, una secuencia de las tramas de la salida de flujo de bits de audio descodificado procedente de la memoria intermedia 301, utilizando la salida de metadatos procedentes del descodificador 200 y/o bits de control de salida desde la etapa 204 del descodificador 200. Normalmente el post-procesador 300 está configurado para realizar un procesamiento adaptativo sobre los datos de audio descodificados utilizando metadatos procedentes del descodificador 200 (por ejemplo, procesamiento de sonoridad adaptativo en los datos de audio descodificados utilizando valores de LPSM y, además, de forma opcional, metadatos de límite de programa, en donde el procesamiento adaptativo puede estar basado en el estado de procesamiento de sonoridad y/o una o más características de datos de audio, indicadas por LPSM para datos de audio indicativos de un único programa de audio).

Varias ejecuciones prácticas del descodificador 200 y el post-procesador 300 están configuradas para realizar diferentes formas de realización del método de la invención.

El descodificador de audio 202, del descodificador 200, está configurado para descodificar los datos de audio extraídos por el analizador sintáctico 205, con el fin de generar datos de audio descodificados, y para establecer los datos de audio descodificados como salida (por ejemplo, al post-procesador 300).

El validador de estado 203 está configurado para autenticar y validar los metadatos establecidos en el mismo. En algunas formas de realización, los metadatos son (o están incluidos en) un bloque de datos que se ha incluido en el flujo de bits de entrada (por ejemplo, de conformidad con una forma de realización de la presente invención). El bloque puede incluir un *hash* criptográfico (un código de autenticación de mensaje basado en *hash*, o "HMAC") para el procesamiento de los metadatos y/o los datos de audio subyacentes (proporcionados desde el analizador sintáctico 205 y/o el descodificador 202 al validador 203). El bloque de datos puede estar firmado de forma digital en estas

formas de realización, de modo que una unidad de procesamiento de audio de flujo descendente pueda autenticar y validar con relativa facilidad los metadatos de estado de procesamiento.

Otros métodos criptográficos que incluyen, pero no se limitan a, uno o más métodos criptográficos no de HMAC, se pueden utilizar para la validación de metadatos (por ejemplo, en el validador 203) con el fin de garantizar la transmisión y recepción segura de los metadatos y/o los datos de audio subyacentes. Por ejemplo, la validación (utilizando dicho método criptográfico) se puede realizar en cada unidad de procesamiento de audio que recibe una forma de realización del flujo de bits de audio inventivo, para determinar si los metadatos de estado de procesamiento de sonoridad y los correspondientes datos de audio, incluidos en el flujo de bits, se han sometido a (y/o han resultado de) un procesamiento de sonoridad específico (tal como se indica por los metadatos) y no se han modificado después de la realización de dicho procesamiento de sonoridad específico.

El validador de estado 203 establece datos de control para controlar el generador de bits 204, y/o establece los datos de control como salida (por ejemplo, al post-procesador 300), para indicar los resultados de la operación de validación. En respuesta a los datos de control (y opcionalmente también otros metadatos extraídos del flujo de bits de entrada), la etapa 204 puede generar (y establecer al post-procesador 300) cualquiera de lo que sigue:

bits de control que indican que la salida de datos de audio descodificados, procedentes del descodificador 202, ha experimentado un tipo específico de procesamiento de sonoridad (cuando LPSM indica que la salida de datos de audio del descodificador 202 ha experimentado el tipo específico de procesamiento de sonoridad, y los bits de control, procedentes del validador 203, indican que los LPSM son válidos); o

bits de control que indican que la salida de datos de audio descodificados del descodificador 202, deben someterse a un tipo específico de procesamiento de sonoridad (por ejemplo, cuando LPSM indica que la salida de datos de audio procedentes del descodificador 202, no se han sometido al tipo específico de procesamiento de sonoridad, o cuando el LPSM indica que la salida de datos de audio del descodificador 202 ha experimentado el tipo específico de procesamiento de sonoridad, pero los bits de control del validador 203 indican que los LPSM no son válidos).

Como alternativa, el descodificador 200 establece metadatos extraídos por el descodificador 202 desde el flujo de bits de entrada, y metadatos extraídos por el analizador sintáctico 205 desde el flujo de bits de entrada hasta el post-procesador 300, y el post-procesador 300 realiza un procesamiento adaptativo sobre los datos de audio descodificados utilizando los metadatos, o realiza la validación de los metadatos y luego realiza un procesamiento adaptativo en los datos de audio descodificados utilizando los metadatos, si la validación indica que los metadatos son válidos.

En algunas formas de realización, si el descodificador 200 recibe un flujo de bits de audio generado de conformidad con una forma de realización de la invención, con un *hash* criptográfico, el descodificador está configurado para analizar y recuperar el *hash* criptográfico desde un bloque de datos determinado a partir del flujo de bits, comprendiendo dicho bloque metadatos de estado de procesamiento de sonoridad (LPSM). El validador 203 puede utilizar el *hash* criptográfico para validar el flujo de bits recibido y/o metadatos asociados. Por ejemplo, si el validador 203 encuentra que los metadatos LPSM son válidos, sobre la base de una coincidencia entre un *hash* criptográfico de referencia y el *hash* criptográfico recuperado del bloque de datos, entonces puede indicar a una unidad de procesamiento de audio de flujo descendente (por ejemplo, post-procesador 300, que puede ser, o incluir, una unidad de nivelación de volumen) para pasar a través (sin cambio) de los datos de audio del flujo de bits. Además, de modo opcional, o como alternativa, se pueden utilizar otros tipos de técnicas criptográficas en lugar de un método basado en un *hash* criptográfico.

En algunas ejecuciones prácticas del descodificador 200, el flujo de bits codificado recibido (y guardado temporalmente en la memoria 201), es un flujo de bits en AC-3 o un flujo de bits en E-AC-3, e incluye segmentos de datos de audio (por ejemplo, los segmentos AB0-AB5 de la trama ilustrada en la Figura 4) y segmentos de metadatos, en donde los segmentos de datos de audio son indicativos de datos de audio, y cada uno de al menos algunos de los segmentos de metadatos incluye PIM o SSM (u otros metadatos). La etapa del descodificador 202 (y/o el analizador sintáctico 205) está configurada para extraer los metadatos del flujo de bits. Cada uno de los segmentos de metadatos que incluye PIM y/o SSM (y, opcionalmente, también otros metadatos), se incluye en un segmento de bits residuales de una trama del flujo de bits, o un campo "addbsi" del segmento de Información de Flujo de Bits ("BSI") de una trama del flujo de bits, o en un campo auxdata (por ejemplo, el segmento AUX, que se ilustra en la Figura 4) al final de una trama del flujo de bits. Una trama del flujo de bits puede incluir uno o dos segmentos de metadatos, cada uno de los cuales incluya metadatos, y si la trama incluye dos segmentos de metadatos, uno puede estar presente en el campo addbsi de la trama, y el otro en el campo AUX de la trama.

En algunas formas de realización, cada segmento de metadatos (a veces referido aquí como un "contenedor") del flujo de bits guardado en la memoria intermedia 201, tiene un formato que incluye una cabecera de segmento de metadatos (y además, de forma opcional, otros elementos obligatorios o "principales"), y una o más cargas útiles de metadatos que siguen a la cabecera del segmento de metadatos. SIM, si está presente, se incluye en una de las cargas útiles de metadatos (identificada por una cabecera de carga útil y, normalmente, tiene un formato de un primer tipo). PIM, si está presente, se incluye en otra de las cargas útiles de metadatos (identificada por una cabecera de carga útil y que suele tener un formato de un segundo tipo). De modo similar, cada otro tipo de metadatos (si está presente), se incluye en otra de las cargas útiles de metadatos (que se identifica por una cabecera de carga útil y que tiene normalmente

un formato específico para el tipo de metadatos). El formato a modo de ejemplo permite un acceso conveniente a los SSM, PIM y otros metadatos en momentos distintos a cuando se realiza la descodificación (por ejemplo, por del post-procesador 300 después de la descodificación, o por un procesador configurado para reconocer los metadatos sin realizar una descodificación completa en el flujo de bits codificado), y permite las cómodas y eficientes detección y corrección de errores, (por ejemplo, de identificación de flujo secundario) durante la descodificación del flujo de bits. A modo de ejemplo, sin acceso a SSM en el formato de ejemplo, el descodificador 200 podría identificar incorrectamente el número correcto de flujos secundarios asociados con un programa. Una carga útil de metadatos, en un segmento de metadatos, puede incluir SSM, otra carga útil de metadatos en el segmento de metadatos puede incluir PIM y, opcionalmente, también al menos otra carga útil de metadatos en el segmento de metadatos puede incluir otros metadatos (por ejemplo, metadatos de estado de procesamiento de sonoridad o "LPSM").)

En algunas formas de realización, una carga útil de metadatos de estructura de flujo secundario (SSM), incluida en una trama de un flujo de bits codificado (por ejemplo, un flujo de bits en E-AC-3 indicativo de al menos un programa de audio), que se guarda en la memoria intermedia 201, incluye SSM en el formato siguiente:

una cabecera de carga útil, que incluye normalmente al menos un valor de identificación (por ejemplo, un valor de 2 bits indicativo de la versión de formato de SSM y, además como opción, valores de longitud, período, conteo y asociación de flujo secundario); y

después de la cabecera:

metadatos de flujo secundario independiente, que indican el número de flujos secundarios independientes del programa, indicados por el flujo de bits; y

metadatos de flujo secundario dependiente, indicativos de si cada flujo secundario independiente del programa tiene al menos un flujo secundario dependiente asociado con el mismo, y, si es así, el número de flujos secundarios dependientes, asociados con cada flujo secundario independiente del programa.

En algunas formas de realización, una carga útil de metadatos de información sobre el programa (PIM), incluida en una trama de un flujo de bits codificado, (por ejemplo un flujo de bits en E-AC-3, indicativo de al menos un programa de audio), que se memoriza, de forma temporal, en la memoria intermedia 201, tiene el siguiente formato:

una cabecera de carga útil, que incluye normalmente al menos un valor de identificación (por ejemplo, un valor indicativo de la versión de formato de PIM y además, de forma opcional, valores de longitud, período, conteo y asociación de flujo secundario); y

después de la cabecera, PIM en el siguiente formato:

metadatos de canales activos de cada canal silencioso y cada canal no silencioso, de un programa de audio (es decir, cuyos canales del programa contienen información de audio y los que (si los hay) contienen solamente silencio (normalmente durante la duración de la trama)). En formas de realización en las que el flujo de bits codificado es un flujo de bits de AC-3 o un flujo de bits en E-AC-3, los metadatos de canal activo en una trama del flujo de bits se pueden utilizar junto con metadatos adicionales del flujo de bits (por ejemplo, el campo de modo de codificación de audio ("acmod") de la trama y, si está presente, el campo de mapa de canales en la trama o tramas de flujo secundario dependiente asociado, para determinar qué canales del programa contienen información de audio y cuáles contienen silencio;

los metadatos de estado de procesamiento de mezcla descendente que indican si el programa fue objeto de mezcla descendente (antes o durante la codificación), y, si es así, el tipo de operación de mezcla descendente que se aplicó. Los metadatos de estado de procesamiento de mezcla descendente pueden ser útiles para poner en práctica la mezcla ascendente (por ejemplo en el post-procesador 300) de aguas abajo de un descodificador, por ejemplo, para la mezcla ascendente del contenido de audio del programa, utilizando parámetros que coincidan más estrechamente con un tipo de operación de mezcla descendente que fue aplicado. En formas de realización en las que el flujo de bits codificado es un flujo de bits en AC-3 o E-AC-3, los metadatos de estado de procesamiento de mezcla descendente se pueden utilizar junto con el campo de modo de codificación de audio ("acmod") de la trama para determinar el tipo de operación de mezcla descendente (si corresponde) aplicado al(a los) canal(es) del programa;

metadatos de estado de procesamiento de mezcla ascendente, indicativos de si el programa fue objeto de mezcla ascendente, (por ejemplo, a partir de un pequeño número de canales), antes o durante la codificación y, de ser así, el tipo de operación de mezcla ascendente que se aplicó. Los metadatos de estado de procesamiento de mezcla ascendente pueden ser útiles para poner en práctica la operación de mezcla descendente (en un post-procesador) aguas abajo de un descodificador, por ejemplo para la mezcla descendente del contenido de audio del programa, en un modo que sea compatible con un tipo de mezcla ascendente (por ejemplo, Dolby Pro Logic, o Dolby Pro Logic II Movie Mode, o Dolby Pro Logic II Music Mode, o Dolby Professional Upmixer) que se aplicó al programa. En formas de realización en las que el flujo de bits codificado es un flujo de bits en E-AC-3, los metadatos de estado de procesamiento de mezcla ascendente se pueden utilizar junto con otros metadatos (por ejemplo el valor de un campo "strmtyp" de la trama), con el fin de determinar el tipo de operación de mezcla ascendente (si corresponde) aplicado al(a los) canal(es) del programa. El valor del campo "strmtyp" (en el segmento de BSI de una trama de un flujo de bits

- en E-AC-3) indica si el contenido de audio de la trama pertenece a un flujo independiente (que determina un programa) o un flujo secundario independiente (de un programa que incluye, o está asociado con, múltiples flujos secundarios) y, por lo tanto, se puede descodificar, con independencia de cualquier otro flujo secundario indicado por el flujo de bits en E-AC-3, o si el contenido de audio de la trama pertenece a un flujo secundario dependiente (de un programa que incluye, o está asociado con, múltiples flujos secundarios) y, por lo tanto, se debe descodificar junto con un flujo secundario independiente con el que está asociado; y
- 5 metadatos de estado de pre-procesamiento, que indican si el pre-procesamiento se realizó en el contenido de audio de la trama (antes de la codificación del contenido de audio para generar el flujo de bits codificado), y, de ser así, el tipo de pre-procesamiento que fue realizado.
- 10 En algunas realizaciones, los metadatos de estado de pre-procesamiento son indicativos de:
- si se aplicó una atenuación circundante (por ejemplo, si los canales circundantes del programa de audio se atenuaron en 3 dB antes de la codificación),
- si se aplicó un cambio de fase de 90 grados (por ejemplo para canales circundantes Ls y canales Rs del programa de audio antes de la codificación),
- 15 si se aplicó un filtro de paso bajo a un canal de LFE del programa de audio antes de la codificación,
- si el nivel de un canal de LFE del programa fue supervisado durante la producción y, de ser así, el nivel supervisado del canal de LFE relativo al nivel de los canales de audio de margen completo del programa,
- si se debe realizar una compresión de margen dinámico (por ejemplo en el descodificador) en cada bloque de contenido de audio descodificado del programa y, si es así, el tipo (y/o parámetros) de compresión de margen dinámico que ha de realizarse (por ejemplo este tipo de metadatos de estado de pre-procesamiento puede ser indicativo de cuál de los siguientes tipos de perfil de compresión fue asumido por el codificador para generar valores de control de compresión de margen dinámico que están incluidos en el flujo de bits codificado: Clase de Película, Luz de película, Clase de Música, Luz musical o Voz. Como alternativa, este tipo de metadatos de estado de pre-procesamiento puede indicar que debe realizarse la compresión de margen dinámico intensa (compresión "compr") en cada trama de contenido de audio descodificado del programa, de un modo determinado por los valores de control de compresión de margen dinámico que están incluidos en el flujo de bits codificado),
- 20 si el procesamiento de extensión espectral y/o codificación de acoplamiento de canal se empleó para codificar márgenes de frecuencia específicos del contenido del programa y, en caso afirmativo, las frecuencias mínima y máxima de los componentes de frecuencia del contenido en que se realizó la codificación de extensión espectral, y las frecuencias mínima y máxima de componentes de frecuencia del contenido en el que se realizó la codificación de acoplamiento de canal. Este tipo de información de metadatos de estado de pre-procesamiento puede ser útil para realizar la igualación (en un post-procesador) aguas abajo de un descodificador. Tanto el acoplamiento de canales como la información de extensión espectral son útiles, además, para optimizar la calidad durante operaciones y aplicaciones de trans-codificación. Por ejemplo, un codificador puede optimizar su comportamiento (incluida la adaptación de etapas de pre-procesamiento, tales como la virtualización de auriculares, operación de mezcla ascendente, etc.) sobre la base del estado de los parámetros, como la extensión espectral y la información de acoplamiento de canal. Además, el codificador podría adaptar sus parámetros de acoplamiento y extensión espectral, de forma dinámica, para que coincidan y/o a valores óptimos basados en el estado de los metadatos entrantes (y autenticados), y
- 30 si los datos de margen de ajuste de mejora de diálogo están incluidos en el flujo de bits codificado y, si es así, el margen de ajuste disponible durante la realización del procesamiento de mejora de diálogo (por ejemplo en un post-procesador aguas abajo de un descodificador) para ajustar el nivel de contenido de diálogo relativo al nivel de contenido sin diálogo en el programa de audio.
- 45 En algunas formas de realización, una carga útil de LPSM incluida en una trama de un flujo de bits codificado (por ejemplo un flujo de bits en E-AC-3 indicativo de al menos un programa de audio) que se guarda temporalmente en la memoria intermedia 201, incluye LPSM en el siguiente formato:
- una cabecera (que incluye típicamente una palabra de sincronización que identifica el inicio de la carga útil de LPSM, seguida de al menos un valor de identificación, por ejemplo la versión de formato de LPSM, longitud, período, recuento y valores de asociación flujo secundario indicados en la Tabla 2 siguiente); y
- 50 después de la cabecera,
- al menos un valor de indicación de diálogo (por ejemplo, parámetro "Canal(es) de Diálogo" de la Tabla 2), que indica si los correspondientes datos de audio indican diálogo, o no indican diálogo, (por ejemplo, qué canales de los correspondientes datos de audio indican diálogo);
- al menos un valor de cumplimiento de normativa de sonoridad (por ejemplo parámetro "Tipo de Normativa de

Sonoridad" de la Tabla 2), que indica si los correspondientes datos de audio cumplen con un conjunto indicado de normativas de sonoridad;

5 al menos un valor de procesamiento de sonoridad (por ejemplo, uno o más de los parámetros "Indicador de Corrección de Sonoridad bloqueada de Diálogo", "Tipo de Corrección de Sonoridad" de la Tabla 2), que indica al menos un tipo de procesamiento de sonoridad que se ha realizado en los correspondientes datos de audio; y

al menos un valor de sonoridad (por ejemplo, uno o más de los parámetros "Sonoridad Bloqueada Relativa de ITU", "Sonoridad Bloqueada de Expresión vocal de ITU", "Sonoridad de Corto plazo de 3s de ITU (EBU 3341)" y "Pico Verdadero" de la Tabla 2), que indican al menos una característica de sonoridad (por ejemplo pico o promedio de sonoridad) de los datos de audio correspondientes.

10 En algunas ejecuciones prácticas, el analizador sintáctico 205 (y/o la etapa 202 de descodificador) se configuran para extraer, a partir de un segmento de bits residuales, o un campo "addbsi", o un campo de datos auxiliares, de una trama del flujo de bits, teniendo cada segmento de metadatos el siguiente formato:

15 una cabecera de segmento de metadatos (que incluye normalmente una palabra de sincronización que identifica el inicio del segmento de metadatos, seguida de al menos un valor de identificación, por ejemplo valores de versión, longitud y período, conteo de elementos expandidos y de asociación de flujo secundario); y

después de la cabecera del segmento de metadatos, al menos un valor de protección (por ejemplo, los valores HMAC digest y de Huella dactilar de Audio de la Tabla 1) útil para al menos una función de entre desenscriptación, autenticación o validación de al menos uno de los metadatos del segmento de metadatos, o el correspondiente dato de audio); y

20 además, después de la cabecera del segmento de metadatos, valores de identificación ("ID") de carga útil de metadatos y de configuración de carga útil, que identifican el tipo y al menos un aspecto de la configuración (por ejemplo tamaño) de cada carga útil de metadatos siguiente.

Cada segmento de carga útil de metadatos (que tiene, preferentemente, el formato especificado con anterioridad) sigue los correspondientes valores de ID de carga útil de metadatos y de configuración de carga útil.

25 Más generalmente, el flujo de bits de audio codificado, generado por formas de realización de la invención preferidas, tiene una estructura que proporciona un mecanismo para etiquetar elementos de metadatos y sub-elementos como elementos o sub-elementos principales (obligatorios) o expandidos (de forma opcional). Esto permite que la tasa de datos del flujo de bits (incluidos sus metadatos) sea objeto de escala en numerosas aplicaciones. Los elementos principales (obligatorios) de la sintaxis del flujo de bits preferido deberían ser capaces, además, de señalar que los
30 elementos expandidos (opcionales), asociados con el contenido de audio están presentes (en banda) y/o en una localización distante (fuera de banda).

Se requiere que el(los) elemento(s) principal(es) esté(n) presente(s) en cada trama del flujo de bits. Algunos sub-elementos de los elementos principales son opcionales y pueden estar presentes en cualquier combinación. No se requiere que elementos expandidos estén presentes en cada trama (para limitar la sobrecarga de la tasa de bits). En consecuencia, los elementos expandidos pueden estar presentes en algunas tramas y no en otras. Algunos sub-elementos de un elemento expandido son opcionales y pueden estar presentes en cualquier combinación, mientras que algunos sub-elementos de un elemento expandido pueden ser obligatorios (es decir, si el elemento expandido está presente en una trama del flujo de bits).
35

40 En una clase de formas de realización, se genera un flujo de bits de audio codificado que comprende una secuencia de segmentos de datos de audio y segmentos de metadatos (por ejemplo, mediante una unidad de procesamiento de audio que incorpora la invención). Los segmentos de datos de audio son indicativos de datos de audio, cada uno de al menos alguno de los segmentos de metadatos incluye PIM y/o SSM (y además, de modo opcional, metadatos de al menos otro tipo), y los segmentos de datos de audio son objeto de multiplexación por división de tiempo con los segmentos de metadatos. En formas de realización preferidas de esta clase, cada uno de los segmentos de metadatos
45 tiene un formato preferido que se describirá en este documento.

En un formato preferido, el flujo de bits codificado es un flujo de bits en AC-3 o un flujo de bits en E-AC-3, y cada uno de los segmentos de metadatos que incluye SSM y/o PIM está incluido (por ejemplo por la etapa 107 de una ejecución práctica preferida del codificador 100), como información de flujo de bits adicional en el campo "addbsi" (ilustrado en la Figura 6) del segmento de Información de Flujo de Bits ("BSI") de una trama del flujo de bits, o en un campo auxdata de una trama del flujo de bits, o en un segmento de bits residuales de una trama del flujo de bits.
50

En el formato preferido, cada una de las tramas incluye un segmento de metadatos (a veces aquí referido como un contenedor de metadatos, o contenedor), en un segmento de bits residuales (o campo addbsi) de la trama. El segmento de metadatos tiene los elementos obligatorios (referidos, de forma colectiva, como el "elemento principal") que se ilustra en la Tabla 1 siguiente (y pueden incluir los elementos opcionales que se ilustran en la Tabla 1). Al menos alguno de los elementos requeridos que se ilustran en la Tabla 1, están incluidos en la cabecera de segmento de metadatos del segmento de metadatos, pero algunos se pueden incluir en cualquier parte del segmento de
55

metadatos:

Tabla 1

Parámetro	Descripción	Obligatorio/ Opcional
SYNC [ID]		Obligatorio
Versión de elemento principal		Obligatorio
Longitud de elemento principal		Obligatorio
Periodo de elemento principal (xxx)		Obligatorio
Conteo de elementos expandidos	Indica la cantidad de elementos de metadatos expandidos que se asocian con el elemento principal. Este valor puede aumentarse/disminuirse a medida que el flujo de bits pasa de producción a distribución y emisión final.	Obligatorio
Asociación de flujo secundario	Describe a qué flujo secundario está asociado el elemento principal.	Obligatorio
Firma (HMAC digest)	HMAC digest de 256 bits (utilizando el algoritmo SHA-2) calculado sobre los datos de audio, el elemento principal y todos los elementos expandidos de la trama completa.	Obligatorio
Cuenta descendente del límite de PGM	El campo solamente aparece para alguna cantidad de tramas en la parte superior o posterior de un fichero/flujo de programas de audio. Por lo tanto, un cambio en la versión del elemento principal se podría utilizar para señalar la inclusión de este parámetro.	Opcional
Huella Dactilar de audio	Huella Dactilar de audio tomada sobre varias muestras de audio de PCM representadas por el campo de período de elemento principal.	Opcional
Huella Dactilar de video	Huella Dactilar de video tomada sobre varias muestras de video comprimidas (si las hay) representadas por el campo del período de elemento principal.	Opcional
URL/UUID	Este campo está definido para transmitir un URL y/o un UUID (puede ser redundante para la huella dactilar) que hace referencia a una localización externa de contenido de programa adicional (esencia) y/o metadatos asociados con el flujo de bits.	Opcional

- 5 En el formato preferido, cada segmento de metadatos (en un segmento de bits residual, o campo addbsi o auxdata de una trama de un flujo de bits codificado) que contiene metadatos SSM, PIM o LPSM contiene una cabecera de segmento de metadatos (y, además, de modo opcional, elementos principales adicionales), y después de la cabecera del segmento de metadatos (o la cabecera del segmento de metadatos y otros elementos principales), una o más cargas útiles de metadatos. Cada carga útil de metadatos incluye una cabecera de carga útil de metadatos (que indica un tipo específico de metadatos (por ejemplo SSM, PIM o LPSM)) incluido en la carga útil, seguido de metadatos del tipo específico. Normalmente, la cabecera de carga útil de metadatos incluye los valores siguientes (parámetros):
- 10

un ID de carga útil (que identifica el tipo de metadatos, por ejemplo SSM, PIM o LPSM) que sigue a la cabecera del segmento de metadatos (que puede incluir los valores especificados en la Tabla 1);

un valor de configuración de carga útil (que generalmente indica el tamaño de la carga útil) que sigue al identificador ID de carga útil;

- 5 y además, de modo opcional, valores de configuración adicionales de carga útil (por ejemplo, un valor de compensación que indica el número de muestras de audio desde el inicio de la trama a la primera muestra de audio a la que pertenece la carga útil y el valor de prioridad de la carga útil, que indica, por ejemplo, una condición en la que la carga útil puede descartarse).

Normalmente, los metadatos de la carga útil tienen uno de los siguientes formatos:

- 10 los metadatos de la carga útil son SSM, que incluyen metadatos de flujo secundario independiente, que indican el número de flujos secundarios independientes del programa indicado por el flujo de bits; y metadatos de flujo secundario dependiente, indicativos de si cada flujo secundario independiente del programa tiene al menos un flujo secundario dependiente asociada con él, y, de ser así, el número de flujos secundarios dependientes asociados con cada flujo secundario independiente del programa;

- 15 los metadatos de la carga útil son PIM, que incluyen metadatos de canal activo, indicativos de qué canales de un programa de audio contienen información de audio, y cuáles (si los hay) contienen solamente silencio (generalmente durante la duración de la trama); metadatos de estado de procesamiento de mezcla descendente, indicativos de si el programa fue objeto de mezcla descendente (antes o durante la codificación), y, si es así, el tipo de operación de mezcla descendente que se aplicó, metadatos de estado de procesamiento de mezcla ascendente, indicativos de si el programa fue objeto de mezcla ascendente (por ejemplo a partir de un menor número de canales) antes o durante la codificación, y, si es así, el tipo de operación de mezcla ascendente que se aplicó, y metadatos de estado de pre-procesamiento, indicativos de si se realizó el pre-procesamiento en el contenido de audio de la trama (antes de codificar el contenido de audio para generar el flujo de bits codificado), y, si es así, el tipo de pre-procesamiento que se realizó; o

- 25 los metadatos de la carga útil son LPSM que tienen el formato que se indica en la siguiente tabla (Tabla 2):

Tabla 2

Parámetro LPSM [Sonoridad inteligente]	Descripción	Número de estados únicos	Obligatorio/Opcional	Tasa de inserción (Periodo de actualización del parámetro)
Versión de LPSM			O	
Período LPSM (xxx)	Aplicable solamente a campos xxx		Obligatorio	
Conteo de LPSM			Obligatorio	
Asociación flujo secundario de LPSM			Obligatorio	
Canal(es) de diálogo	Indica qué combinación de canales de audio L, C y R contienen voz durante los 0,5 segundos anteriores. Cuando la voz no está presente en ninguna combinación de L, C o R, este parámetro deberá indicar "sin diálogo"	8	Obligatorio	-0,5 segundos (típico)
Tipo de normativa de sonoridad	Indica que el flujo de datos de audio asociado cumple con un conjunto específico de normativas (p.ej., ATSC A/85 o EBU R128)	8	Obligatorio	Trama

Indicador de Corrección de Sonoridad Bloqueada de Diálogo	Indica si el flujo de audio asociado se ha corregido según el control de diálogo	2	Obligatorio (solamente está presente si Loudness_Regulation_Type indica que el audio correspondiente NO ESTÁ CORREGIDO)	Trama
Tipo de corrección de sonoridad	Indica si el flujo de audio asociado se ha corregido con una búsqueda infinita anticipada (basada en archivos) o con controlador de margen dinámico y sonoridad en tiempo real (RT)	2	Obligatorio (solamente está presente si Loudness_Regulation_Type indica que el audio correspondiente NO ESTÁ CORREGIDO)	Trama
Sonoridad bloqueada relativa de ITU (INF)	Indica la norma ITU-R BS.1770-3 de sonoridad integrada del flujo de audio asociado sin metadatos aplicados (p.ej., 7 bits: -58 -> +5,5 LKFS en pasos de 0,5 LKFS)	128	Obligatorio	1 seg.
Sonoridad bloqueada de expresión vocal de ITU (INF)	Indica la norma ITU-R BS.1770-1/3 de sonoridad integrada de voz/diálogo del flujo de audio asociado sin metadatos aplicados (p.ej., 7 bits: -58 -> +5,5 LKFS en pasos de 0.5 LKFS)	128	Obligatorio	1 seg.
ITU (EBU 3341) Sonoridad a Corto plazo de 3s	Indica la norma ITU (ITU-BS.1771-1) de 3 segundos sin control de sonoridad del flujo de audio asociado sin metadatos aplicados (ventana deslizante) @ tasa de inserción de ~ 10 Hz (p. ej., 8 bits: 116 -> +11.5 LKFS en pasos de 0,5 LKFS)	256	Obligatorio	0,1 seg.
Valor de pico verdadero	Indica la norma ITU-R BS.1770-3 Anexo 2, de Valor de Pico Verdadero (dB TP) del flujo de audio asociado sin metadatos aplicados (es decir, el valor más alto durante el período de trama señalado en el campo de período de elemento) 116 -> +11,5 LKFS en pasos de 0,5 LKFS	256	Obligatorio	0,5 seg.
Compensación de mezcla descendente	Indica la compensación de sonoridad de mezcla descendente			
Límite del programa	Indica, en tramas, cuándo habrá o ha ocurrido un límite de programa. Cuando el límite del programa no está en el límite de la trama, el desplazamiento de muestra opcional indicará en qué momento ocurre el límite de programa real de la trama.			

En otro formato preferido de un flujo de bits codificado, generado de conformidad con la invención, el flujo de bits es un flujo de bits en AC-3, o un flujo de bits en E-AC-3, y se incluye cada uno de los segmentos de metadatos que

5 incluyen PIM y/o SSM (y opcionalmente, también metadatos de al menos otro tipo) (por ejemplo, mediante la etapa 107 de una puesta en práctica preferida del codificador 100) en cualquiera de los segmentos de bits residuales de una trama del flujo de bits; o un campo "addbsi" (ilustrado en la Figura 6) del segmento de Información de Flujo de Bits ("BSI") de una trama del flujo de bits; o un campo auxdata (por ejemplo, el segmento AUX ilustrado en la Figura 4) al final de una trama del flujo de bits. Una trama puede incluir uno o dos segmentos de metadatos, cada uno de los cuales incluye PIM y/o SSM, y (en algunas formas de realización) si la trama incluye dos segmentos de metadatos, uno puede estar presente en el campo addbsi de la trama, y el otro en el campo AUX de la trama. Cada segmento de metadatos tiene preferentemente el formato especificado anteriormente con referencia a la Tabla 1 anterior (es decir, incluye los elementos principales especificados en la Tabla 1, seguidos por ID de carga útil (que identifica el tipo de metadatos en cada carga útil del segmento de metadatos) y valores de configuración de carga útil, y cada carga útil de metadatos). Cada segmento de metadatos que incluye metadatos LPSM tiene, preferentemente, el formato especificado anteriormente con referencia a las Tablas 1 y 2 anteriores (es decir, incluye los elementos principales especificados en la Tabla 1, seguidos por ID de carga útil (que identifica los metadatos como LPSM) y valores de configuración de carga útil, seguidos por la carga útil (datos de LPSM que tienen un formato según se indica en la Tabla 2)).

20 En otro formato preferido, el flujo de bits codificado es un flujo de bits en Dolby E, y cada uno de los segmentos de metadatos que incluyen PIM y/o SSM (y opcionalmente, también otros metadatos) es la primera ubicación de N muestras del intervalo de banda de guarda en Dolby E. Un flujo de bits en Dolby E que incluye dicho segmento de metadatos que incluye LPSM, incluye preferentemente un valor indicativo de la longitud de carga útil de LPSM señalada en la palabra de Pd del preámbulo de SMPTE 337M (la tasa de repetición de la palabra de SMPTE 337M Pa permanece, preferentemente, idéntica a la tasa de trama de vídeo asociada).

25 En un formato preferido, en el que el flujo de bits codificado es un flujo de bits en E-AC-3, se incluye cada uno de los segmentos de metadatos que incluyen PIM y/o SSM (y opcionalmente, también LPSM y/u otros metadatos) (por ejemplo mediante la etapa 107 de una ejecución práctica preferida del codificador 100) como información de flujo de bits adicional en un segmento de bits residuales, o en el campo "addbsi" del segmento de Información de Flujo de Bits ("BSI"), de una trama del flujo de bits. A continuación se describen aspectos adicionales de la codificación de un flujo de bits en E-AC-3 con LPSM en este formato preferido:

30 1. Durante la generación de un flujo de bits en E-AC-3, mientras que el codificador en E-AC-3 (que inserta los valores de LPSM en el flujo de bits) está "activo" para cada trama (trama de sincronización) generada, el flujo de bits debe incluir un bloque de metadatos (incluyendo LPSM) transmitido en el campo addbsi (o segmento de bits residuales) de la trama. Los bits necesarios para transmitir el bloque de metadatos no deberían aumentar la tasa de bits del codificador (longitud de trama);

2. Cada bloque de metadatos (que contiene LPSM) debe contener la siguiente información:

35 **loudness_correction_type_flag** (sonoridad_corrección_tipo_indicador): en donde '1' indica que la sonoridad de los datos de audio correspondientes se corrigió aguas arriba del codificador, y '0' indica que la sonoridad fue corregida por un corrector de sonoridad integrado en el codificador (por ejemplo, el procesador de sonoridad 103 del codificador 100 de la Figura 2);

speech_channel (habla_canal): indica qué canal(es) origen contiene(n) voz (en los últimos 0,5 segundos). Si no se detecta la voz, esto se indicará como tal;

40 **speech_loudness** (habla_sonoridad): indica la sonoridad de voz integrada de cada canal de audio correspondiente que contiene voz (durante los 0,5 segundos previos);

ITU_loudness: indica la sonoridad integrada de ITU BS.1770-3 de cada canal de audio correspondiente; y

gain: ganancia(s) compuesta(s) de sonoridad para la inversión en un descodificador (para demostrar la reversibilidad);

45 3. Mientras que el codificador en E-AC-3 (que inserta los valores de LPSM en el flujo de bits) está "activo" y recibe una trama en AC-3 con un indicador de "confianza", el controlador de sonoridad en el codificador (por ejemplo procesador de sonoridad 103 del codificador 100 de la Figura 2) debe ser desviado (bypassed). Los valores de dialnorm y DRC origen "de confianza" deben hacerse pasar a través de (por ejemplo, por medio del generador 106 del codificador 100) al componente del codificador de E-AC-3 (por ejemplo, etapa 107 del codificador 100). La generación del bloque de LPSM continúa y **loudness_correction_type_flag** se pone a '1'. La secuencia de derivación del controlador de sonoridad debe estar sincronizada con el inicio de la trama descodificada en AC-3, en donde aparece el indicador 'trust' ("confianza"). La secuencia de derivación del controlador de sonoridad debe ponerse en práctica como sigue: el control **leveller_amount** se disminuye desde un valor de 9 a un valor de 0 sobre 10 periodos de bloque de audio (es decir, 53,3 mseg) y el control **leveler_back_end_meter** se coloca en modo derivación (esta operación debería dar lugar a una transición sin problemas). El término derivación "de confianza" del nivelador implica que el valor dialnorm del flujo de bits origen se reutiliza también a la salida del codificador, (por ejemplo, si el flujo de bits origen 'de confianza' tiene un valor de dialnorm de -30, entonces la salida del codificador debe utilizar -30 para el valor del parámetro dialnorm de salida);

4. Mientras que el codificador en E-AC-3 (que inserta los valores de LPSM en el flujo de bits) está "activo" y está recibiendo una trama en AC-3 sin el indicador de "confianza", el controlador de sonoridad incluido en el codificador (por ejemplo, procesador de sonoridad 103 del codificador 100 de la Figura 2) debe estar activo. La generación del bloque de LPSM continúa y el `loudness_correction_type_flag` se establece a '0'. La secuencia de activación del controlador de sonoridad debe sincronizarse con el inicio de la trama en AC-3 descodificada, en donde desaparece el indicador de 'confianza'. La secuencia de activación del controlador de sonoridad debe ponerse en práctica como sigue: el control de `leveler_amount` (nivelador_cantidad) se incrementa desde un valor de 0 a un valor de 9 sobre 1 período de bloque de audio (es decir, 5,3 mseg) y el control de `leveler_back_end_meter` (nivelador_atrás_final_medidor) se coloca en el modo "activo" (esta operación debería dar como resultado una transición sin interrupciones e incluir un reinicio de integración de `back_end_meter`); y

5. Durante la codificación, una interfaz gráfica de usuario (GUI) debe indicar a un usuario los siguientes parámetros: "Programa de Audio de Entrada: [Fiable/no fiable]" -el estado de este parámetro se basa en la presencia del indicador de "confianza" dentro de la señal de entrada y "Corrección de sonoridad en tiempo real: [Habilitado/Deshabilitado]" - el estado de este parámetro se basa en el hecho de si este controlador de sonoridad, integrado en el codificador, está activo.

Cuando se descodifica un flujo de bits en AC-3 o E-AC-3 que tiene LPSM (en el formato preferido), incluido en un segmento de bits residuales o de campo de omisión, o el campo "addbsi" del segmento de Información de Flujo de Bits ("BSI"), de cada trama del flujo de bits, el descodificador debe analizar los datos del bloque de LPSM (en el segmento de bits residual o campo addbsi) y pasar todos los valores LPSM extraídos a una interfaz gráfica de usuario (GUI). El conjunto de valores de LPSM extraídos se renueva cada trama.

En otro formato preferido de un flujo de bits codificado, generado de conformidad con la invención, el flujo de bits codificado es un flujo de bits en AC-3 o un flujo de bits en E-AC-3, y se incluye cada uno de los segmentos de metadatos, que incluye PIM y/o SSM (y, opcionalmente, también LPSM y/u otros metadatos) (por ejemplo, en la etapa 107 de una ejecución práctica preferida del codificador 100) en un segmento de bits residuales, o en un segmento Aux, o como información de flujo de bits adicional en el campo "addbsi" (ilustrado en la Figura 6) del segmento de Información de Flujo de Bits ("BSI"), de una trama del flujo de bits. En este formato (que es una variación del formato descrito anteriormente con referencias a las Tablas 1 y 2), cada uno de los campos addbsi (o Aux o bits residuales), que contiene LPSM, contiene los siguientes valores de LPSM:

los elementos principales especificados en la Tabla 1, seguidos por el ID de carga útil (que identifica los metadatos como LPSM) y valores de configuración de carga útil, seguidos por la carga útil (datos de LPSM) que tiene el siguiente formato (similar a los elementos obligatorios indicados en la Tabla 2 anterior):

versión de la carga útil de LPSM: un campo de 2 bits que indica la versión de la carga útil de LPSM;

dialchan: un campo de 3 bits que indica si los canales Izquierdo, Derecho y/o Central de los datos de audio correspondientes contienen diálogo hablado. La asignación de bits del campo dialchan puede ser como sigue: bit 0, que indica la presencia de diálogo en el canal izquierdo, se memoriza en el bit más significativo del campo dialchan; y bit 2, que indica la presencia de diálogo en el canal central, se memoriza en el bit menos significativo del campo dialchan. Cada bit del campo dialchan se establece a '1' si el canal correspondiente contiene diálogo hablado durante los 0,5 segundos anteriores del programa;

loudregtyp: un campo de 4 bits que indica la norma de regulación de la intensidad que cumple la sonoridad del programa. Al establecer el campo "loudregtyp" en "000", se indica que los LPSM no indican el cumplimiento de la normativa de intensidad. Por ejemplo, un valor de este campo (por ejemplo, 0000) puede indicar que no está indicado el cumplimiento con una norma de regulación de la sonoridad, otro valor de este campo (por ejemplo, 0001) puede indicar que los datos de audio del programa cumplen con la norma ATSC A/85, y otro valor de este campo (por ejemplo, 0010) puede indicar que los datos de audio del programa cumplen con la norma EBU R128. En el ejemplo, si el campo se pone en cualquier valor distinto de '0000', los campos loudcorrdialgat y loudcorrtyp deberían seguir en la carga útil;

loudcorrdialgat: un campo de un bit que indica si se ha aplicado la corrección de la sonoridad bloqueada de diálogo. Si la sonoridad del programa se ha corregido utilizando el bloqueo (gating) de diálogo, el valor del campo loudcorrdialgat se pone en '1'. En caso contrario, se pone en '0';

loudcorrtyp: un campo de un bit que indica el tipo de corrección de intensidad aplicado al programa. Si la sonoridad del programa se ha corregido con un proceso de corrección de sonoridad infinita anticipada (basado en fichero), el valor del campo loudcorrtyp se establece en '0'. Si la intensidad del programa se ha corregido utilizando una combinación de medición de sonoridad en tiempo real y control de margen dinámico, el valor de este campo se establece en '1';

loudrelgate: un campo de un bit que indica si existen datos de sonoridad bloqueados relativos (ITU). Si el campo loudrelgate se establece en '1', debe seguir un campo ituloudrelgat de 7 bits en la carga útil;

loudrelgat: un campo de 7 bits que indica la sonoridad relativa del programa bloqueado (ITU). Este campo indica la sonoridad integrada del programa de audio, medida de conformidad con la norma ITU-R BS.1770-3 sin ningún ajuste

de ganancia debido a la aplicación de dialnorm y la compresión del margen dinámico (DRC). Los valores de 0 a 127 se interpretan como -58 LKFS a +5.5 LKFS, en pasos de 0,5 LKFS;

loudspchgate: un campo de un bit que indica si existen datos de sonoridad bloqueados por la voz (ITU). Si el campo loudspchgate está establecido en '1', un campo loudspchgat de 7 bits debe seguir en la carga útil;

5 loudspchgat: un campo de 7 bits que indica la sonoridad del programa bloqueado de voz. Este campo indica la sonoridad integrada del correspondiente programa de audio completo, medido de conformidad con la fórmula (2) de la ITU-R BS.1770-3 y sin ningún ajuste de ganancia debido a la aplicación de dialnorm y compresión de margen dinámico. Los valores de 0 a 127 se interpretan como -58 a +5,5 LKFS, en pasos de 0,5 LKFS;

10 loudstrm3se: un campo de un bit que indica si existen datos de sonoridad de corto plazo (3 segundos). Si el campo está establecido en '1', un campo loudstrm3s de 7 bits debe permanecer en la carga útil;

loudstrm3s: un campo de 7 bits que indica la sonoridad no bloqueada de los 3 segundos anteriores del programa de audio correspondiente, medida de conformidad con la norma ITU-R BS.1771-1 y sin ningún ajuste de ganancia debido a la aplicación de dialnorm y compresión de margen dinámico. Los valores de 0 a 256 se interpretan como -116 LKFS a +11,5 LKFS en pasos de 0,5 LKFS;

15 truepk: un campo de un bit que indica si existen datos de sonoridad de pico verdadero. Si el campo truepk está establecido en '1', un campo truepk de 8 bits debe seguir en la carga útil; y

20 truepk: un campo de 8 bits que indica el valor de muestra de pico verdadero del programa, medido de conformidad con el Anexo 2 de la norma ITU-R BS.1770-3 y sin ningún ajuste de ganancia debido a la aplicación de dialnorm y la compresión de margen dinámico. Los valores de 0 a 256 se interpretan como -116 LKFS a +11,5 LKFS en pasos de 0,5 LKFS.

25 En algunas formas de realización, el elemento principal de un segmento de metadatos, en un segmento de bits residuales o en un campo auxdata (o "addbsi") de una trama de un flujo de bits en AC-3, o un flujo de bits en E-AC-3, comprende una cabecera de segmento de metadatos (normalmente incluye valores de identificación, por ejemplo, versión) y después la cabecera del segmento de metadatos; valores indicativos de si los datos de huellas dactilares se incluyen (u otros valores de protección) para los metadatos del segmento de metadatos, valores indicativos de si los datos externos existen (relacionados con datos de audio correspondientes a los metadatos del segmento de metadatos), ID de carga útil y valores de configuración de carga útil para cada tipo de metadatos (por ejemplo, PIM y/o SSM y/o LPSM y/o metadatos de un tipo) identificados por el elemento principal, y valores de protección para al menos un tipo de metadatos identificados por la cabecera del segmento de metadatos (u otros elementos principales del segmento de metadatos). Las cargas útiles de metadatos, del segmento de metadatos, siguen a la cabecera del segmento de metadatos y están (en algunos casos), alojadas dentro de elementos principales del segmento de metadatos.

35 Formas de realización de la presente invención se pueden poner en práctica en hardware, firmware o software, o una combinación de ambos (por ejemplo, como una matriz lógica programable). A no ser que se especifique de otro modo, los algoritmos o procesos incluidos como parte de la invención no están intrínsecamente relacionados con ningún ordenador particular u otro aparato. En particular, se pueden utilizar varias máquinas de uso general con programas escritos de conformidad con las enseñanzas aquí dadas a conocer, o puede ser más conveniente construir aparatos más especializados (por ejemplo, circuitos integrados) para realizar las etapas requeridas del método. Por lo tanto, la invención se puede poner en práctica en uno o más programas informáticos que se ejecutan en uno o más sistemas informáticos programables (por ejemplo, una realización de cualquiera de los elementos de la Figura 1, o el codificador 100 de la Figura 2 (o un elemento del mismo), o descodificador 200 de la Figura 3 (o un elemento del mismo), o post-procesador 300 de la Figura 3 (o un elemento del mismo)) que comprenden cada uno al menos un procesador, al menos un sistema de memorización de datos (incluyendo memoria volátil y no volátil/o elementos de memorización), al menos un dispositivo o puerto de entrada, y al menos un dispositivo o puerto de salida. El código de programa se aplica a datos de entrada para realizar las funciones descritas en este documento y para generar información de salida. La información de salida se aplica a uno o más dispositivos de salida, de manera conocida.

50 Cada uno de dichos programas se puede poner en práctica en cualquier lenguaje informático deseado (incluyendo lenguajes de máquina, montaje o de procedimientos de alto nivel, lógicos, o de programación orientados al objeto) para comunicarse con un sistema informático. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado.

55 A modo de ejemplo, cuando se pone en práctica mediante secuencias de instrucciones de software, varias funciones y etapas de formas de realización de la invención pueden realizarse mediante secuencias de instrucciones de software multiproceso que funcionan en hardware de procesamiento de señal digital adecuado, en cuyo caso los diversos dispositivos, pasos y funciones, de las formas de realización, pueden corresponder a partes de las instrucciones del software.

Cada uno de dichos programas informáticos preferentemente se memoriza, o descarga, en un soporte de almacenamiento o dispositivo (por ejemplo, memoria o medio de estado sólido, o medio magnético u óptico) legible

5 por un ordenador programable de finalidad general o especial, para la configuración y funcionamiento del ordenador cuando el sistema informático lee el dispositivo o medio de memorización, para realizar los procedimientos aquí descritos. El sistema inventivo puede ponerse en práctica, además, como un medio de memorización legible por ordenador, configurado con (es decir, memorizando) un programa informático, en donde el soporte de memorización así configurado hace que un sistema informático opere de manera específica y predefinida para realizar las funciones aquí descritas.

10 Se han descrito varias formas de realización de la invención. Sin embargo, ha de entenderse que pueden realizarse diversas modificaciones sin desviarse del alcance de la invención. Numerosas modificaciones y variaciones de la presente invención son posibles a la luz de las enseñanzas anteriores. Se ha de entender que, dentro del alcance de las reivindicaciones adjuntas, la invención se puede poner en práctica de otros modos distintos al que concretamente aquí se describe.

REIVINDICACIONES

1. Un método para generar un flujo de bits de audio codificado, comprendiendo el método:

la generación de una secuencia de tramas de un flujo de bits de audio codificado, en el que el flujo de bits de audio codificado es un flujo de bits en AC-3 o un flujo de bits en E-AC-3, siendo indicativo el flujo de bits de audio codificado de al menos un programa de audio, incluyendo cada trama de al menos un subconjunto de dichas tramas:

metadatos de información sobre el programa en al menos un segmento de metadatos situado en al menos uno de: a) un campo de omisión de la trama; b) un campo addbsi de la trama; c) un campo aux de la trama; y

datos de audio en al menos otro segmento de la trama,

en el que el segmento de metadatos incluye al menos una carga útil de metadatos, comprendiendo dicha carga útil de metadatos: una cabecera; y, después de la cabecera, al menos alguno de los metadatos sobre información del programa,

en el que los metadatos de información sobre el programa son indicativos de información acerca del al menos un programa de audio no presente en otras porciones del flujo de bits de audio codificado,

en el que los metadatos de información sobre el programa incluyen metadatos de canal activo indicativos de cada canal no silencioso y cada canal silencioso del al menos un programa de audio.

2. Un método para descodificar un flujo de bits de audio codificado, incluyendo dicho método los pasos de:

recibir un flujo de bits de audio codificado, en el que el flujo de bits de audio codificado es un flujo de bits en AC-3 o un flujo de bits en E-AC-3,

en el que el flujo de bits de audio codificado comprende una secuencia de tramas y es indicativo de al menos un programa de audio, incluyendo cada trama de al menos un subconjunto de dichas tramas:

metadatos de información sobre el programa en al menos un segmento de metadatos situado en al menos uno de: a) un campo de omisión de la trama; b) un campo addbsi de la trama; c) un campo aux de la trama; y

los datos de audio en al menos otro segmento de la trama,

en el que el segmento de metadatos incluye al menos una carga útil de metadatos, comprendiendo dicha carga útil de metadatos: una cabecera; y, después de la cabecera, al menos alguno de los metadatos de información sobre el programa,

en el que los metadatos de información sobre el programa son indicativos de información acerca de al menos un programa de audio no presente en otras porciones del flujo de bits de audio codificado,

comprendiendo además el método extraer los datos de audio y los metadatos de información sobre el programa del flujo de bits de audio codificado,

en el que los metadatos de información sobre programa incluyen metadatos de canal activo indicativos de cada canal no silencioso y cada canal silencioso del al menos un programa de audio.

3. El método según la reivindicación 1 o el método según la reivindicación 2, en el que los metadatos de información sobre el programa incluyen además al menos uno de entre:

metadatos del estado de procesamiento de mezcla descendente, que indican si el programa fue objeto de mezcla descendente, y, si es así, un tipo de mezcla descendente que se aplicó al programa;

metadatos del estado de procesamiento de mezcla ascendente, que indican si el programa fue objeto de mezcla ascendente y, de ser así, un tipo de mezcla ascendente que se aplicó al programa;

metadatos de estado de pre-procesamiento, indicativos de si el pre-procesamiento se realizó sobre el contenido de audio de la trama, y, si es así, un tipo de pre-procesamiento que se realizó sobre dicho contenido de audio; o

metadatos de procesamiento de extensión espectral o de acoplamiento de canal, indicativos de si se aplicó al programa procesamiento de extensión espectral o acoplamiento de canal, y, si es así, un margen de frecuencia que se aplicó a la extensión espectral o al acoplamiento de canal.

4. El método según la reivindicación 1 o el método de la reivindicación 2, en el que el segmento de metadatos incluye:

una cabecera de segmento de metadatos;

después de la cabecera de segmento de metadatos, valores de identificación de carga útil de metadatos y de

configuración de carga útil, en el que la carga útil de metadatos sigue a valores de identificación de carga útil de metadatos y de configuración de carga útil;

5 después del segmento de metadatos, al menos un valor de protección útil para al menos una de entre descriptación, autenticación o validación de los metadatos de información sobre el programa o los datos de audio correspondientes a dichos metadatos de información sobre el programa.

5. El método según la reivindicación 4, en el que la cabecera del segmento de metadatos incluye una palabra de sincronización que identifica el inicio del segmento de metadatos, y al menos un valor de identificación que sigue a la palabra de sincronización, y la cabecera de la carga útil de metadatos incluye al menos un valor de identificación.

10 6. Un medio de almacenamiento legible por ordenador, que tiene guardado en el mismo un programa informático configurado para hacer que un sistema informático realice el método de cualquier reivindicación precedente.

7. Una unidad de procesamiento de audio, que comprende:

una memoria intermedia o temporal (109, 110, 201, 301); y

al menos un subsistema de procesamiento, acoplado a la memoria intermedia, y configurado para realizar el método según cualquiera de las reivindicaciones 1 a 5.

15

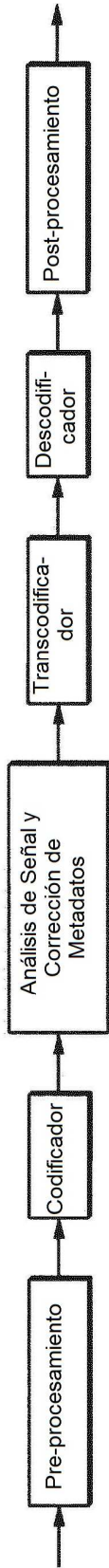


FIG. 1

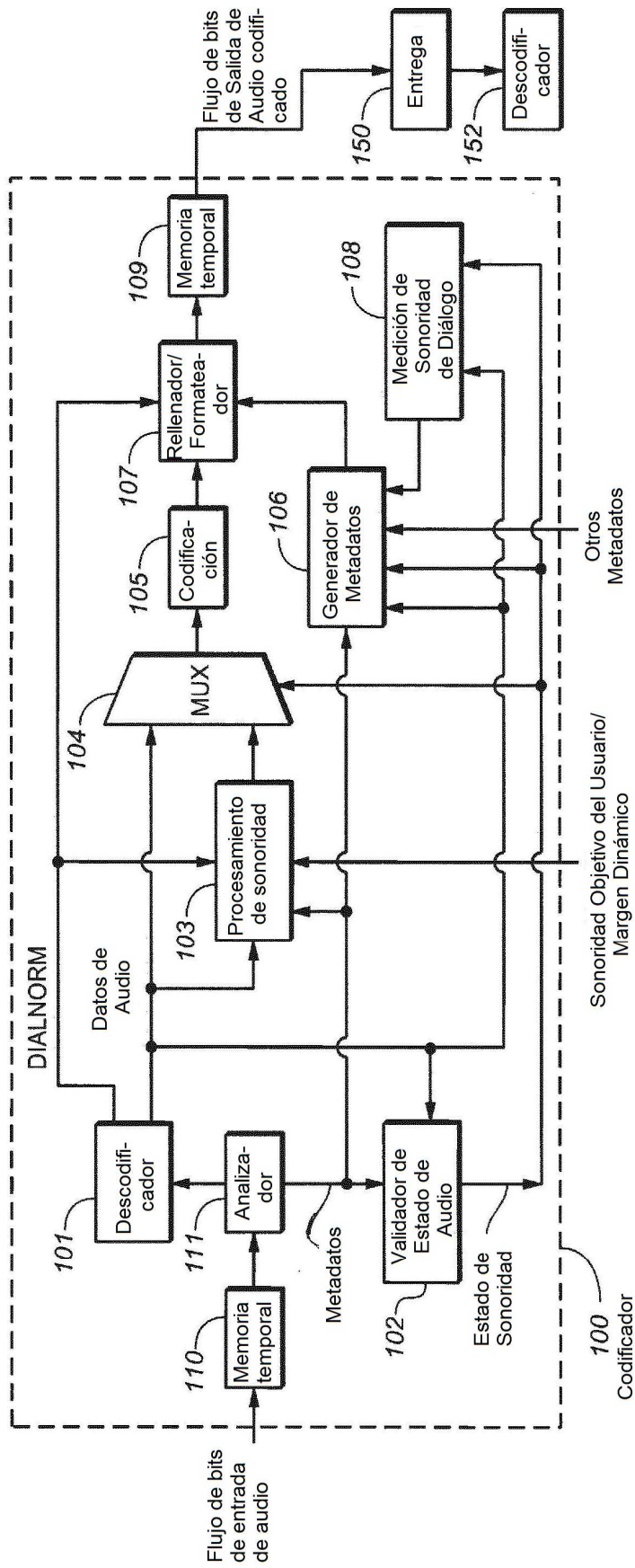


FIG. 2

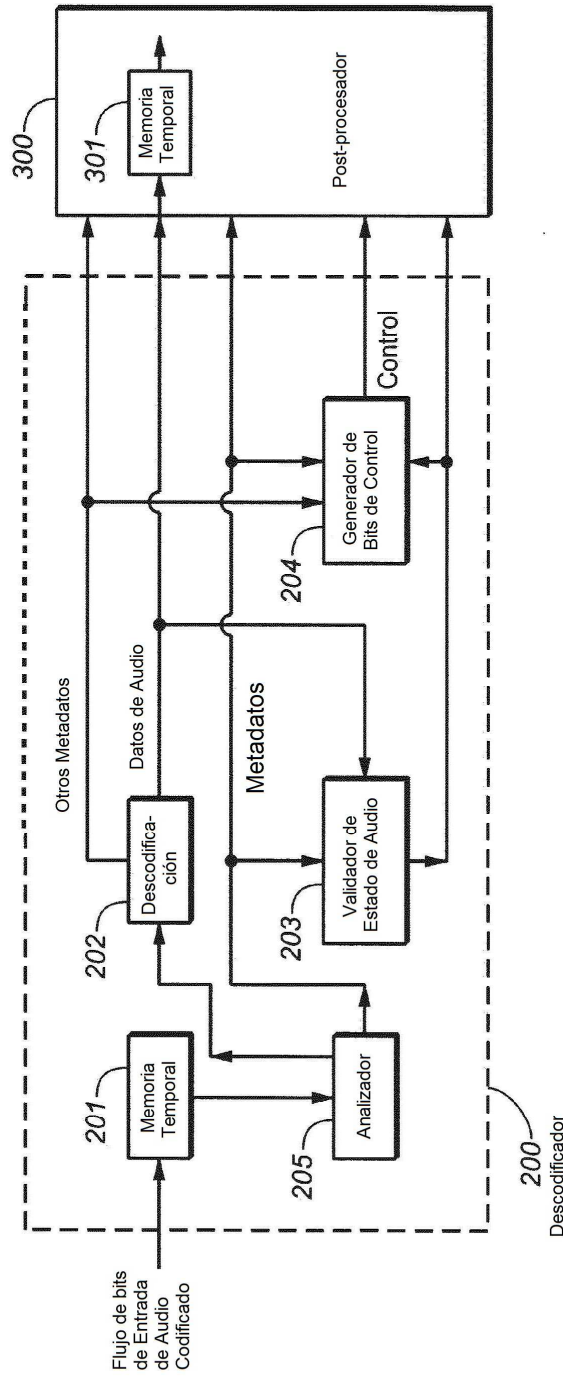


FIG. 3

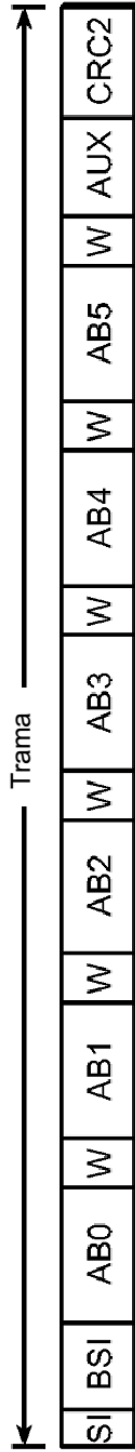


FIG. 4

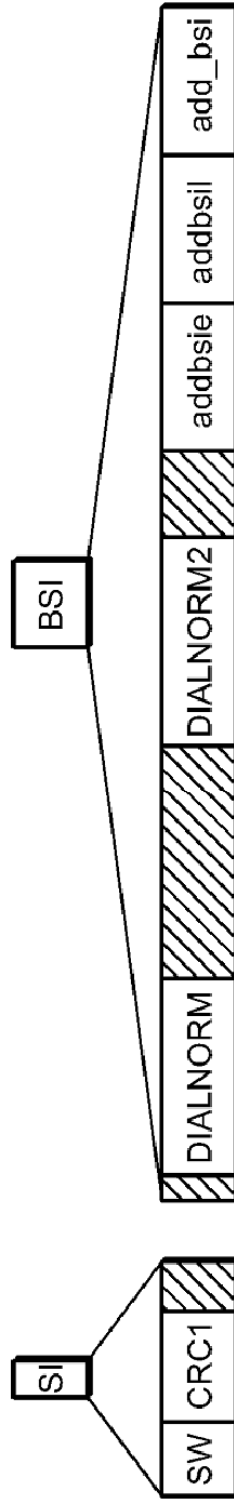


FIG. 5

FIG. 6



FIG. 7

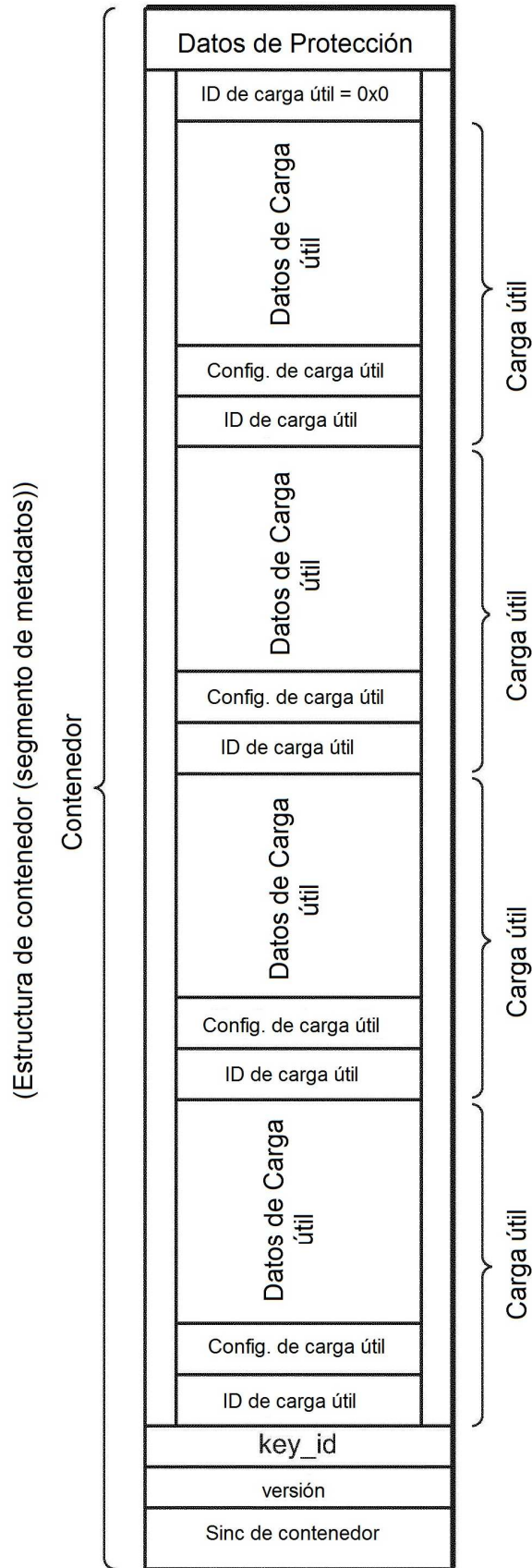


Fig. 8