(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 749 089**

(51) Int. Cl.:

H03G 9/00 (2006.01)
H03G 9/02 (2006.01)
G10L 19/16 (2013.01)
G10L 19/002 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **15.01.2014** **E 16164487 (7)**

(97) Fecha y número de publicación de la concesión europea: **31.07.2019** **EP 3079257**

(54) Título: **Codificador y descodificador de audio con metadatos de límite y sonoridad de programa**

(30) Prioridad:

21.01.2013 US 201361754882 P
16.05.2013 US 201361824010 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
19.03.2020

(73) Titular/es:

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

(72) Inventor/es:

GRANT, MICHAEL;
NORCROSS, SCOTT GREGORY;
RIEDMILLER, JEFFREY y
WARD, MICHAEL

(74) Agente/Representante:

ELZABURU, S.L.P

ES 2 749 089 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 descodificador de audio con metadatos de límite y sonoridad de programa

Referencia cruzada a solicitudes relacionadas

La presente solicitud reivindica la prioridad de la Solicitud de Patente Provisional de los Estados Unidos No. 61/754.882, presentada el 21 de enero de 2013 y la Solicitud de Patente Provisional de los Estados Unidos No. 61/824.010, presentada el 16 de mayo de 2013.

Campo técnico

La invención se refiere al procesamiento de señales de audio y, más concretamente, a la codificación y descodificación de trenes de bits de datos de audio con metadatos indicativos del estado de procesamiento de sonoridad del contenido de audio y la ubicación de los límites de programa de audio indicados por los trenes de bits. Algunas realizaciones de la invención generan o descodifican datos de audio en uno de los formatos conocidos como AC-3, Enhanced AC-3 o E-AC-3, o Dolby E.

Antecedentes de la invención

Dolby, Dolby Digital, Dolby Digital Plus y Dolby E son marcas comerciales de Dolby Laboratories Licensing Corporation. Dolby Laboratories provee implementaciones en propiedad de AC-3 y E-AC-3 conocidas como Dolby Digital y Dolby Digital Plus, respectivamente.

Las unidades de procesamiento de datos de audio normalmente funcionan a ciegas y no prestan atención al historial de procesamiento de datos de audio que ocurre antes de recibir los datos. Ello puede funcionar en un marco de procesamiento en el cual una sola entidad hace toda codificación y el procesamiento de datos de audio para una variedad de dispositivos de renderización de medios de destino mientras un dispositivo de renderización de medios de destino lleva a cabo toda la descodificación y renderización de los datos de audio codificados. Sin embargo, este procesamiento ciego no funciona bien (o directamente no funciona) en situaciones donde una pluralidad de unidades de procesamiento de audio se dispersan a lo largo de una red diversa o se ubican en serie (a saber, cadena) y se espera que lleven a cabo, de forma óptima, sus respectivos tipos de procesamiento de audio. Por ejemplo, algunos datos de audio pueden codificarse para sistemas de medios de alto rendimiento y pueden tener que convertirse en una forma reducida apropiada para un dispositivo móvil a lo largo de una cadena de procesamiento de medios. Por consiguiente, una unidad de procesamiento de audio puede llevar a cabo, de forma innecesaria, un tipo de procesamiento en los datos de audio que ya se ha llevado a cabo. Por ejemplo, una unidad de nivelación de volumen puede llevar a cabo el procesamiento en un clip de audio de entrada, independientemente de si la misma nivelación o una nivelación de volumen similar se han llevado a cabo o no previamente en el clip de audio de entrada. Como resultado, la unidad de nivelación de volumen puede llevar a cabo la nivelación incluso cuando no es necesaria. Este procesamiento innecesario puede también causar la degradación y/o eliminación de rasgos específicos durante la renderización del contenido de los datos de audio.

Un tren típico de datos de audio incluye tanto contenido de audio (p. ej., uno o más canales de contenido de audio) como metadatos indicativos de al menos una característica del contenido de audio. Por ejemplo, en un tren de bits AC-3, existen varios parámetros de metadatos de audio que están pretendidos específicamente para su uso en el cambio de sonido del programa entregado a un entorno de audición. Uno de los parámetros de metadatos es el parámetro DIALNORM, el cual pretende indicar el nivel medio de diálogo que ocurre en un programa de audio y se usa para determinar el nivel de señal de reproducción de audio.

Durante la reproducción de un tren de bits que comprende una secuencia de diferentes segmentos de programa de audio (cada uno de los cuales tiene un parámetro DIALNORM diferente), un descodificador AC-3 usa el parámetro DIALNORM de cada segmento para llevar a cabo un tipo de procesamiento de sonoridad en el que modifica el nivel de reproducción o sonoridad de modo que la sonoridad percibida del diálogo de la secuencia de segmentos se encuentra en un nivel coherente. Cada segmento de audio codificado (artículo) en una secuencia de artículos de audio codificados tiene (en general) un parámetro DIALNORM diferente y el descodificador escala el nivel de cada uno de los artículos de modo que el nivel de reproducción o sonoridad del diálogo para cada artículo es igual o muy similar, aunque ello puede requerir la aplicación de diferentes cantidades de ganancia a diferentes artículos durante la reproducción.

DIALNORM se establece, normalmente, por un usuario y no se genera de forma automática, aunque existe un valor DIALNORM por defecto 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 AC-3 y luego transferir el resultado (indicativo de la sonoridad del diálogo hablado de un programa de audio) al codificador para establecer el valor DIALNORM. Por consiguiente, existe una dependencia del creador de contenido para establecer el parámetro DIALNORM de forma correcta.

Existen varias razones diferentes por las que el parámetro DIALNORM en un tren de bits AC-3 puede ser incorrecto. Primero, cada codificador AC-3 tiene un valor DIALNORM por defecto que se usa durante la generación del tren de

bits si un valor DIALNORM no es establecido por el creador de contenido. Este valor por defecto puede ser sustancialmente diferente del nivel de sonoridad de diálogo real del audio. Segundo, incluso si un creador de contenido mide la sonoridad y establece, por consiguiente, el valor DIALNORM, puede haberse usado un medidor o algoritmo de medición de sonoridad que no se adapta al método de medición de sonoridad AC-3 recomendado, lo que resulta en un valor DIALNORM incorrecto. Tercero, incluso si un tren de bits AC-3 se ha creado con el valor DIALNORM medido y establecido de forma correcta por el creador de contenido, puede haberse cambiado a un valor incorrecto durante la transmisión y/o almacenamiento del tren de bits. Por ejemplo, es común en aplicaciones de difusión por televisión que los trenes de bits AC-3 se decodifiquen, modifiquen y luego recodifiquen mediante el uso de información de metadatos DIALNORM incorrectos. Por consiguiente, un valor DIALNORM incluido en un tren de bits AC-3 puede ser incorrecto o inexacto y, por lo tanto, puede tener un impacto negativo en la calidad de la experiencia auditiva.

Además, el parámetro DIALNORM no indica el estado de procesamiento de sonoridad de los datos de audio correspondientes (p. ej. qué tipo(s) de procesamiento de sonoridad puede haberse llevado a cabo en los datos de audio). Hasta la presente invención, un tren de bits de audio no incluía metadatos, indicativos del estado de procesamiento de sonoridad del (p. ej., tipo(s) de procesamiento de sonoridad aplicado al) contenido de audio del tren de bits de audio o el estado de procesamiento de sonoridad y sonoridad del contenido de audio del tren de bits, en un formato de un tipo descrito en la presente descripción. Los metadatos de estado de procesamiento de sonoridad en dicho formato son útiles para facilitar el procesamiento de sonoridad adaptativo de un tren de bits de audio y/o verificación de validez del estado de procesamiento de sonoridad y sonoridad del contenido de audio, de una manera particularmente eficaz.

Aunque la presente invención no se encuentra limitada al uso con un tren de bits AC-3, un tren de bits E-AC-3 o un tren de bits Dolby E, en aras de la conveniencia, se describirá en realizaciones en las que genera, descodifica o de otra forma procesa dicho tren de bits que incluye metadatos de estado de procesamiento de sonoridad.

Un tren de bits codificado AC-3 comprende metadatos y uno a seis canales de contenido de audio. El contenido de audio consiste en datos de audio que se han comprimido mediante el uso de codificación de audio perceptual. Los metadatos incluyen varios parámetros de metadatos de audio se pretenden para uso en el cambio de sonido de un programa entregado a un entorno de audición.

Los detalles de la codificación AC-3 (también conocida como Dolby Digital) son bien conocidos y se presentan en muchas referencias publicadas, incluidas las siguientes:

ATSC Standard A52/a: Digital Audio Compression Standard (AC-3), Revisión A, Advanced Television Systems Committee, 20 de agosto de 2001; y

Patentes de Estados Unidos 5.583.962; 5.632.005; 5.633.981; 5.727.119 y 6.021.386.

Los detalles de la codificación Dolby Digital Plus (E-AC-3) se presentan en "Introduction to Dolby Digital Plus, an Enhancement to the Dolby Digital Coding System", AES Convention Paper 6196, 117th AES Convention, 28 de octubre de 2004.

Los detalles de la codificación Dolby E se presentan en "Efficient Bit Allocation, Quantization, and Coding in an Audio Distribution System", AES Preprint 5068, 107th AES Conference, agosto 1999 y "Professional Audio Coder Optimized for Use with Video", AES Preprint 5033, 107th AES Conference, agosto 1999.

Cada trama de un tren de bits de audio codificado AC-3 contiene contenido de audio y metadatos para 1536 muestras de audio digital. Para una velocidad de muestreo de 48 kHz, ello representa 32 milisegundos de audio digital o una velocidad de 31.25 tramas por segundo de audio.

Cada trama de un tren de bits de audio codificado E-AC-3 contiene 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 velocidad de muestreo de 48 kHz, ello representa 5.333, 10.667, 16 o 32 milisegundos de audio digital respectivamente o una velocidad de 189.9, 93.75, 62.5 o 31.25 tramas por segundo de audio, respectivamente.

Como se indica en la Figura 4, cada trama AC-3 se divide en secciones (segmentos), que incluyen: una sección de Información de Sincronización (IS) que contiene (como se muestra en la Figura 5) una palabra de sincronización (PS), y la primera de dos palabras de corrección de errores (CRC1); una sección de Información de Tren de Bits (BSI, por sus siglas en inglés) que contiene la mayoría de los metadatos; seis Bloques de Audio (BA0 a BA5) que contienen contenido de audio de datos comprimidos (y pueden incluir también metadatos); segmentos de bits de residuo (W) que contienen bits no usados que quedan después de comprimir el contenido de audio; una sección de información Auxiliar (AUX) que puede contener más metadatos; y la segunda de dos palabras de corrección de errores (CRC2). También puede hacerse referencia al segmento de bits de residuo (W) como "campo de salto".

Como se indica en la Figura 7, cada trama E-AC-3 se divide en secciones (segmentos), que incluyen: una sección de Información de Sincronización (IS) que contiene (como se muestra en la Figura 5) una palabra de sincronización (PS); una sección de Información de Tren de Bits (BSI) que contiene la mayoría de los metadatos; entre uno y seis Bloques

de Audio (BA0 a BA5) que contienen contenido de audio de datos comprimidos (y pueden incluir también metadatos); segmentos de bits de residuo (W) que contienen bits no usados que quedan después de comprimir el contenido de audio (aunque solo se muestra un segmento de bits de residuo, un segmento de bits de residuo diferente seguiría normalmente a cada bloque de audio); una sección de información Auxiliar (AUX) que puede contener más metadatos; y una palabra de corrección de errores (CRC). También puede hacerse referencia al segmento de bits de residuo (W) como "campo de salto".

En un tren de bits AC-3 (o E-AC-3), existen varios parámetros de metadatos de audio que se pretenden específicamente para uso en el cambio de sonido del programa entregado a un entorno de audición. Uno de los parámetros de metadatos es el parámetro DIALNORM, que se incluye en el segmento BSI.

Como se muestra en la Figura 6, el segmento BSI de una trama AC-3 incluye un parámetro de cinco bits ("DIALNORM") que indica el valor DIALNORM para el programa. Un parámetro de cinco bits ("DIALNORM2") que indica el valor DIALNORM para un segundo programa de audio transportado en la misma trama AC-3 se incluye si el modo de codificación de audio ("acmod") de la trama AC-3 es "0", que indica que se encuentra en uso una configuración de canal dual mono o "1+1".

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

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

El documento WO2012075246 (A2) concierne a procesamiento adaptativo con múltiples nodos de procesamiento de medios. El método presentado comprende determinar, mediante un primer dispositivo en una cadena de procesamiento de datos, si un tipo del procesamiento de datos se ha llevado a cabo en una versión de salida de los datos de medios. En respuesta a determinar, mediante un primer dispositivo, que el tipo de procesamiento de medios ha sido realizado en la versión de salida de los datos de medios, el primer dispositivo crea un estado de los datos de medios. El estado especifica el tipo de procesamiento de medios realizado en la versión de salida de los datos de medios. Posteriormente, la versión de salida de los datos de medios y el estado de los datos de medios se comunican desde el primer dispositivo a un segundo dispositivo corriente abajo en la cadena de procesamiento de medios.

El documento WO 2006/113062 A1 describe un sistema para la verificación de metadatos de audio. Un tren de bits digital, que comprende bits de datos que representan audio, metadatos pretendidos para ser correctos para el audio e información de verificación de metadatos, en donde todos o una parte de los metadatos pueden no ser correctos para el audio. La información de verificación de metadatos puede usarse para detectar si los metadatos son correctos o no para el audio y, si no son correctos, para cambiarlos para que sean correctos. La información de verificación de metadatos que puede usarse para detectar y cambiar metadatos puede incluir una copia, o una copia de datos comprimidos, de una versión correcta de los metadatos.

El documento US2008080722 (A1) describe un controlador de sonoridad con control remoto y local. Se describe un sistema y un método asociados con un equipo de audio del destinatario para mantener sonoridad deseada de una señal de audio recibida. El sistema incluye un dispositivo receptor para recibir una señal de entrada difundida y para trasformar la señal de entrada en un tren de trasporte. La señal de entrada y dicho tren de trasporte tienen un componente de señal de audio no procesada que contiene su rango dinámico original. En el sistema se incorpora un controlador de sonoridad de modo que recibe el componente de señal de audio no procesada de dicho tren de trasporte y realiza un proceso de control de rango dinámico en el componente de señal de audio no procesada. Se usa una señal de datos que contiene ajustes para configurar el controlador de sonoridad para controlar selectivamente las operaciones del controlador de sonoridad. La señal de datos permite aceptar metadatos de audio externo y que sean usados para control del controlador sonoridad. Como alternativa, el controlador de sonoridad acepta entrada de usuario externo por el receptor para modificar o confeccionar este controlador de sonoridad de lado consumidor al gusto o las necesidades del receptor. El controlador de sonoridad también puede ser baipaseado para permitir que el audio original pase sin modificar.

Breve descripción de la invención

En una clase de realizaciones, la invención es una unidad de procesamiento de audio que incluye una memoria intermedia (búfer), un descodificador de audio y un analizador. La memoria intermedia almacena al menos una trama de un tren de bits de audio codificado. El tren de bits de audio codificado incluye datos de audio y un contenedor de metadatos. El contenedor de metadatos incluye un encabezamiento, una o más cargas útiles de metadatos y datos de protección. El encabezamiento incluye una palabra de sincronización que identifica el inicio del contenedor. La única o más cargas útiles de metadatos describen un programa de audio asociado a los datos de audio. Los datos de protección se ubican después de la única o más cargas útiles de metadatos. Los datos de protección también pueden usarse para verificar la integridad del contenedor de metadatos y la única o más cargas útiles dentro del contenedor de metadatos. El descodificador de audio se acopla a la memoria intermedia y puede descodificar los datos de audio. El analizador se acopla al descodificador de audio, o se integra en este, y puede analizar el contenedor de metadatos.

En realizaciones típicas, el método incluye recibir un tren de bits de audio codificado donde el tren de bits de audio codificado se segmenta en una o más tramas. Los datos de audio se extraen del tren de bits de audio codificado, junto con un contenedor de metadatos. El contenedor de metadatos incluye un encabezamiento seguido de una o más cargas útiles de metadatos seguidas de datos de protección. Finalmente, la integridad del contenedor y de la única o más cargas útiles de metadatos se verifica a través del uso de los datos de protección. La única o más cargas útiles de metadatos pueden incluir una carga útil de sonoridad de programa que contiene datos indicativos de la sonoridad medida de un programa de audio asociado a los datos de audio.

Una carga útil de los metadatos de sonoridad de programa, a los que se hace referencia como metadatos de estado de procesamiento de sonoridad ("LPSM", por sus siglas en inglés), incorporados en un tren de bits de audio según realizaciones típicas de la invención.

En las realizaciones típicas, los metadatos de límite de programa en una trama del tren de bits inventivo son una bandera de límite de programa indicativa de un cómputo de tramas. Normalmente, la bandera es indicativa del número de tramas entre la trama actual (la trama que incluye la bandera) y un límite de programa (el comienzo o fin del programa de audio actual). En algunas realizaciones preferidas, las banderas de límite de programa se insertan de una manera simétrica y eficaz al comienzo y fin de cada segmento de tren de bits que es indicativo de un solo programa (a saber, en tramas que ocurren dentro de cierto número predeterminado de tramas después del comienzo del segmento, y en tramas que ocurren dentro de cierto número predeterminado de tramas antes del fin del segmento), de modo que cuando dos de dichos segmentos de tren de bits se concatenan (para ser indicativos de una secuencia de dos programas), los metadatos de límite de programa pueden estar presentes (p. ej., de forma simétrica) a ambos lados del límite entre los dos programas.

Con el fin de limitar el aumento de velocidad de datos que resulta de incluir metadatos de límite de programa en un tren de bits de audio codificado (que puede ser indicativo de un programa de audio o una secuencia de programas de audio), en realizaciones típicas, las banderas de límite de programa se insertan únicamente en un subconjunto de las tramas del tren de bits solamente. Normalmente, la velocidad de inserción de bandera de límite es una función no creciente de la separación creciente de cada una de las tramas del tren de bits (en las que se inserta una bandera) desde el límite de programa que está más cerca de cada una de las tramas, donde "velocidad de inserción de bandera de límite" denota la relación promedio del número de tramas (indicativas de un programa) que incluyen una bandera de límite de programa al número de tramas (indicativas del programa) que no incluyen una bandera de límite de programa, donde el promedio es un promedio móvil de un número (p. ej., un número relativamente pequeño) de tramas consecutivas del tren de bits de audio codificado. En una clase de realizaciones, la velocidad de inserción de bandera de límite es una función logarítmicamente decreciente de la distancia creciente (de cada ubicación de inserción de bandera) desde el límite de programa más cercano, y para cada trama que contiene bandera que incluye una de las banderas, el tamaño de la bandera en dicha trama que contiene bandera es igual a o mayor que el tamaño de cada bandera en una trama ubicada más cerca del límite de programa más cercano que lo que se encuentra dicha trama que contiene bandera (a saber, el tamaño de la bandera de límite de programa en cada trama que contiene bandera es una función no decreciente de la separación creciente de dicha trama que contiene bandera desde el límite de programa más cercano).

Otro aspecto de la invención es una unidad de procesamiento de audio (APU, por sus siglas en inglés) configurada para llevar a cabo cualquier realización del método inventivo. En otra clase de realizaciones, la invención es una APU que incluye una memoria intermedia (búfer) que almacena (p. ej., de manera no transitoria) al menos una trama de un tren de bits de audio codificado que se ha generado por cualquier realización del método inventivo. Ejemplos de APU incluyen, pero sin limitación, codificadores (p. ej., transcodificadores), descodificadores, códecs, sistemas de preprocesamiento (preprocesadores), sistemas de postprocesamiento (postprocesadores), sistemas de procesamiento de tren de bits de audio y combinaciones de dichos elementos.

En otra clase de realizaciones, la invención es una unidad de procesamiento de audio (APU) configurada para generar un tren de bits de audio codificado que comprende segmentos de datos de audio y segmentos de metadatos, 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 metadatos de estado de procesamiento de sonoridad (LPSM) y, de forma opcional, también metadatos de límite de programa. Normalmente, al menos un segmento de metadatos en una trama del tren de bits incluye al menos un segmento de LPSM indicativo de si un primer tipo de procesamiento de sonoridad se ha llevado a cabo en los datos de audio de la trama (a saber, datos de audio en al menos un segmento de datos de audio de la trama), y al menos otro segmento de LPSM indicativo de la sonoridad de al menos algunos de los datos de audio de la trama (p. ej., sonoridad de diálogo de al menos algunos de los datos de audio de la trama que son indicativos de diálogo). En una realización en la esta clase, la APU es un codificador configurado para codificar audio de entrada para generar audio codificado, y los segmentos de datos de audio incluyen el audio codificado. En realizaciones típicas en la esta clase, cada uno de los segmentos de metadatos tiene un formato preferido que se describirá en la presente memoria.

En algunas realizaciones, cada uno de los segmentos de metadatos del tren de bits codificado (un tren de bits AC-3 o un tren de bits E-AC-3 en algunas realizaciones) que incluye LPSM (p. ej., LPSM y metadatos de límite de programa) se incluye en un bit de residuo del segmento de campo de salto de una trama del tren de bits (p. ej., un segmento de bits de residuo W del tipo que se muestra en la Figura 4 o la Figura 7). En otras realizaciones, cada uno de los

segmentos de metadatos del tren de bits codificado (un tren de bits AC-3 o un tren de bits E-AC-3 en algunas realizaciones) que incluye LPSM (p. ej., LPSM y metadatos de límite de programa) se incluye como información adicional de tren de bits en el campo "addbsi" del segmento de Información de Tren de Bits ("BSI") de una trama del tren de bits o en un campo auxdata (p. ej., un segmento AUX del tipo que se muestra en la Figura 4 o la Figura 7) al final de una trama del tren de bits. Cada segmento de metadatos que incluye LPSM puede tener el formato especificado en la presente memoria con referencia a las Tablas 1 y 2 más abajo (a saber, incluye los elementos principales especificados en la Tabla 1 o una variación de ellos, seguidos de la ID de carga útil (que identifica los metadatos como LPSM) y valores de tamaño de carga útil, seguidos de la carga útil (los datos LPSM que tienen el formato indicado en la Tabla 2, o el formato indicado en una variación de la Tabla 2 descrita en la presente memoria). En algunas realizaciones, una trama puede incluir uno o dos segmentos de metadatos, cada uno de los cuales incluye LPSM, 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 una clase de realizaciones, la invención es un método que incluye las etapas de codificar datos de audio para generar un tren de bits de audio codificado AC-3 o E-AC-3, incluidos, mediante la inclusión en un segmento de metadatos (de al menos una trama del tren de bits), LPSM y metadatos de límite de programa y, de forma opcional, también otros metadatos para el programa de audio al que pertenece la trama. En algunas realizaciones, cada segmento de metadatos se incluye en un campo addbsi de la trama, o un campo auxdata de la trama. En otras realizaciones, cada segmento de metadatos se incluye en un segmento de bits de residuo de la trama. En algunas realizaciones, cada segmento de metadatos que contiene LPSM y metadatos de límite de programa contiene un encabezamiento principal (y, de forma opcional, también elementos principales adicionales), y después del encabezamiento principal (o el encabezamiento principal y otros elementos principales) un segmento de carga útil de LPSM (o contenedor) que tiene el siguiente formato:

un encabezamiento, que normalmente incluye al menos un valor de identificación (p. ej., versión de formato de LPSM, longitud, período, cómputo y valores de asociación de subtren, como se indica en la Tabla 2 presentada en la presente memoria), y

después del encabezamiento, los LPSM y metadatos de límite de programa. Los metadatos de límite de programa pueden incluir un cómputo de tramas de límite de programa y un valor de código (p. ej., un valor "desplazamiento_existe") indicativo de si la trama incluye solamente un cómputo de tramas de límite de programa o tanto un cómputo de tramas de límite de programa como un valor de desplazamiento), y (en algunos casos) un valor de desplazamiento. Los LPSM pueden incluir:

al menos un valor de indicación de diálogo que indica si los datos de audio correspondientes indican diálogo o no indican diálogo (p. ej., qué canales de los datos de audio correspondientes indican diálogo). El valor(es) de indicación de diálogo puede indicar si el diálogo está presente en cualquier combinación de, o todos, los canales de los datos de audio correspondientes;

al menos un valor de cumplimiento de regulación de sonoridad que indica si los datos de audio correspondientes cumplen con un conjunto indicado de regulaciones de sonoridad;

al menos un valor de procesamiento de sonoridad que indica al menos un tipo de procesamiento de sonoridad que se ha llevado a cabo en los datos de audio correspondientes; y

al menos un valor de sonoridad que indica al menos una característica de sonoridad (p. ej., sonoridad pico o promedio) de los datos de audio correspondientes.

En otras realizaciones, el tren de bits codificado es un tren de bits que no es un tren de bits AC-3 o un tren de bits E-AC-3, y cada uno de los segmentos de metadatos que incluye LPSM (y, de forma opcional, también metadatos de límite de programa) se incluye en un segmento (o campo o intervalo) del tren de bits reservado para el almacenamiento de datos adicionales. Cada segmento de metadatos que incluye LPSM puede tener un formato similar o idéntico al especificado en la presente memoria con referencia a las Tablas 1 y 2 más abajo (a saber, incluye los elementos principales similares o idénticos a los especificados en la Tabla 1, seguidos de la ID de carga útil (que identifica los metadatos como LPSM) y valores de tamaño de carga útil, seguidos de la carga útil (los datos LPSM que tienen un formato similar o idéntico al formato indicado en la Tabla 2 o una variación de la Tabla 2 descrita en la presente memoria).

En algunas realizaciones, el tren de bits codificado comprende una secuencia de tramas, cada una de las tramas incluye un segmento de Información de Tren de Bits ("BSI") que incluye un campo "addbsi" (al que algunas veces se hace referencia como segmento o intervalo) y un campo auxdata o intervalo (p. ej., el tren de bits codificado es un tren de bits AC-3 o un tren de bits E-AC-3), y comprende segmentos de datos de audio (p. ej., los segmentos BA0- BA5 de la trama que se muestra en la Figura 4) y segmentos de metadatos, 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 metadatos de estado de procesamiento de sonoridad (LPSM) y, de forma opcional, también metadatos de límite de programa. Los LPSM se encuentran presentes en el tren de bits en el siguiente formato. Cada uno de los segmentos de metadatos que incluye LPSM se incluye en un campo "addbsi" del segmento BSI de una trama del tren de bits, o en un campo

auxdata de una trama del tren de bits, o en un segmento de bits de residuo de una trama del tren de bits. Cada segmento de metadatos que incluye LPSM incluye un segmento de carga útil de LPSM (o contenedor) que tiene el siguiente formato:

un encabezamiento (que normalmente incluye al menos un valor de identificación, p. ej., la versión de formato de LPSM, longitud, período, cómputo y valores de asociación de subtren indicados en la Tabla 2 más abajo); y

después del encabezamiento, los LPSM y, de forma opcional, también los metadatos de límite de programa. Los metadatos de límite de programa pueden incluir un cómputo de tramas de límite de programa y un valor de código (p. ej., un valor "desplazamiento_existe") indicativo de si la trama incluye solamente un cómputo de tramas de límite de programa o tanto un cómputo de tramas de límite de programa como un valor de desplazamiento), y (en algunos casos) un valor de desplazamiento. Los LPSM pueden incluir:

al menos un valor de indicación de diálogo (p. ej., parámetro "Canal(es) de diálogo" de la Tabla 2) que indica si los datos de audio correspondientes indican diálogo o no indican diálogo (p. ej., qué canales de los datos de audio correspondientes indican diálogo). El valor(es) de indicación de diálogo puede indicar si el diálogo está presente en cualquier combinación de, o todos, los canales de los datos de audio correspondientes;

al menos un valor de cumplimiento de regulación de sonoridad (p. ej., el parámetro "Tipo de Regulación de Sonoridad" de la Tabla 2) que indica si los datos de audio correspondientes cumplen con un conjunto indicado de regulaciones de sonoridad;

al menos un valor de procesamiento de sonoridad (p. ej., uno o más de los parámetros "bandera de Corrección de Sonoridad con puerta de Diálogo", "Tipo de Corrección de Sonoridad", de la Tabla 2) que indica al menos un tipo de procesamiento de sonoridad que se ha llevado a cabo en los datos de audio correspondientes; y

al menos un valor de sonoridad (p. ej., uno o más de los parámetros "Sonoridad con Puerta Relativa a ITU", "Sonoridad con Puerta de Voz ITU", "Sonoridad 3s a Corto Plazo ITU (EBU 3341)", y "Pico Verdadero" de la Tabla 2) que indica al menos una característica de sonoridad (p. ej., sonoridad pico o promedio) de los datos de audio correspondientes.

En cualquier realización de la invención que contempla, usa o genera al menos un valor de sonoridad indicativo de datos de audio correspondientes, el valor(es) de sonoridad puede indicar al menos una característica de medición de sonoridad utilizada para procesar la sonoridad y/o rango dinámico de los datos de audio.

En algunas implementaciones, cada uno de los segmentos de metadatos en un campo "addbsi", o un campo auxdata, o un segmento de bits de residuo, de una trama del tren de bits tiene el siguiente formato:

un encabezamiento principal (que, normalmente, incluye una palabra de sincronización que identifica el inicio del segmento de metadatos, seguida de valores de identificación, p. ej., la versión de elemento Principal, longitud y período, cómputo de elementos extendidos, y valores de asociación de subtren indicados en la Tabla 1 más abajo); y

después del encabezamiento principal, al menos un valor de protección (p. ej., resumen de HMAC y valores de Huella Digital de Audio, donde el resumen de HMAC puede ser un resumen de HMAC de 256 bits (mediante el uso del algoritmo SHA- 2) computado en los datos de audio, el elemento principal, y todos los elementos expandidos, de toda una trama, según se indica en la Tabla 1) útiles para al menos una de desencriptación, autenticación o validación de al menos uno de los metadatos de estado de procesamiento de sonoridad o los datos de audio correspondientes); y

también después del encabezamiento principal, si el segmento de metadatos incluye LPSM, la identificación ("ID") de carga útil de LPSM y valores de tamaño de carga útil de LPSM que identifican los siguientes metadatos como una carga útil de LPSM e indican el tamaño de la carga útil de LPSM. El segmento de carga útil de LPSM (que, preferiblemente, tiene el formato especificado más arriba) sigue a la ID de carga útil de LPSM y valores de tamaño de carga útil de LPSM.

En algunas realizaciones del tipo descrito en el párrafo anterior, cada uno de los segmentos de metadatos en el campo auxdata (o campo "addbsi" o segmento de bits de residuo) de la trama tiene tres niveles de estructura:

una estructura de nivel alto, que indica una bandera que indica si el campo auxdata (o addbsi) incluye metadatos, al menos un valor de ID que indica qué tipo de metadatos están presentes y, normalmente, también un valor que indica cuántos bits de metadatos (p. ej., de cada tipo) están presentes (si hay presentes metadatos). Un tipo de metadatos que pueden estar presentes son LSPM, otro tipo de metadatos que pueden estar presentes son los metadatos de límite de programa, y otro tipo de metadatos que pueden estar presentes son los metadatos de búsqueda de medios;

una estructura de nivel intermedio, que comprende un elemento principal para cada tipo identificado de metadatos (p. ej., encabezamiento principal, valores de protección e ID de carga útil y valores de tamaño de carga útil, p. ej., del tipo mencionado más arriba, para cada tipo identificado de metadatos); y

una estructura de nivel bajo, que comprende cada carga útil para un elemento principal (p. ej., una carga útil de LPSM, si una es identificada por el elemento principal como presente, y/o una carga útil de metadatos de otro tipo, si una es

identificada por el elemento principal como presente).

Los valores de datos en dicha estructura de tres niveles pueden anidarse. Por ejemplo, el valor(es) de protección para una carga útil LPSM y/u otra carga útil de metadatos identificada por un elemento principal pueden incluirse después de cada carga útil identificada por el elemento principal (y, por consiguiente, después del encabezamiento principal del elemento principal). En un ejemplo, un encabezamiento principal puede identificar una carga útil de LPSM y otra carga útil de metadatos, ID de carga útil y valores de tamaño de carga útil para la primera carga útil (p. ej., la carga útil de LPSM) puede seguir al encabezamiento principal, la propia primera carga útil puede seguir a la ID y valores de tamaño, la ID de carga útil y valor de tamaño de carga útil para la segunda carga útil pueden seguir a la primera carga útil, la propia segunda carga útil puede seguir a estos ID y valores de tamaño, y valor(es) de protección para cualquiera o ambas cargas útiles (o para valores de elementos principales y cualquiera o ambas cargas útiles) pueden seguir a la última carga útil.

En algunas realizaciones, el elemento principal de un segmento de metadatos en un campo auxdata (o campo "addbsi" o segmento de bits de residuo) de una trama comprende un encabezamiento principal (que, normalmente, incluye valores de identificación, p. ej., versión de elemento principal) y después del encabezamiento principal: valores indicativos de si los datos de huella digital se incluyen para metadatos del segmento de metadatos, valores indicativos de si existen datos externos (relacionados con datos de audio correspondientes a los metadatos del segmento de metadatos), ID de carga útil y valores de tamaño de carga útil para cada tipo de metadatos (p. ej., LPSM y/o metadatos de un tipo diferente de LPSM) identificados por el elemento principal, y valores de protección para al menos un tipo de metadatos identificado por el elemento principal. Las cargas útiles de metadatos del segmento de metadatos siguen al encabezamiento principal, y se anidan (en algunos casos) dentro de valores del elemento principal.

En otro formato preferido, el tren de bits codificado es un tren de bits Dolby E y cada uno de los segmentos de metadatos que incluye LPSM (y, de forma opcional, también metadatos de límite de programa) se incluye en las primeras N ubicaciones de muestras del intervalo de banda de guarda de Dolby E.

En otra clase de realizaciones, la invención es una APU (p. ej., un descodificador) acoplada y configurada para recibir un tren de bits de audio codificado que comprende segmentos de datos de audio y segmentos de metadatos, 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 metadatos de estado de procesamiento de sonoridad (LPSM) y, de forma opcional, también metadatos de límite de programa, y para extraer los LPSM del tren de bits, para generar datos de audio descodificados en respuesta a los datos de audio y para llevar a cabo al menos una operación de procesamiento de sonoridad adaptativa en los datos de audio mediante el uso de los LPSM. Algunas realizaciones en esta clase también incluyen un postprocesador acoplado a la APU, en donde el postprocesador se acopla y configura para llevar a cabo al menos una operación de procesamiento de sonoridad adaptativa en los datos de audio mediante el uso de los LPSM.

En otra clase de realizaciones, la invención es una unidad de procesamiento de audio (APU) que incluye una memoria intermedia (búfer) y un subsistema de procesamiento acoplado al búfer, en donde la APU se acopla para recibir un tren de bits de audio codificado que comprende segmentos de datos de audio y segmentos de metadatos, 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 metadatos de estado de procesamiento de sonoridad (LPSM) y, de forma opcional, también metadatos de límite de programa, el búfer almacena (p. ej., de una manera no transitoria) al menos una trama del tren de bits de audio codificado, y el subsistema de procesamiento se configura para extraer los LPSM del tren de bits y para llevar a cabo al menos una operación de procesamiento de sonoridad adaptativa en los datos de audio mediante el uso de los LPSM. En realizaciones típicas en la esta clase, la APU es uno de un codificador, un descodificador y un postprocesador.

En algunas implementaciones del método inventivo, el tren de bits de audio generado es uno de un tren de bitscodificado AC-3, un tren de bits E-AC-3, o un tren de bits Dolby E, que incluye los metadatos de estado de procesamiento de sonoridad, así como otros metadatos (p. ej., un parámetro de metadatos DIALNORM, parámetros de metadatos de control de rango dinámico y otros parámetros de metadatos). En algunas otras implementaciones del método, el tren de bits de audio generado es un tren de bits codificado de otro tipo.

Los aspectos de la invención incluyen un sistema o dispositivo configurado (p. ej., programado) para llevar a cabo cualquier realización del método inventivo, y un medio legible por ordenador (p. ej., un disco) que almacena el código (p. ej., de una manera no transitoria) para implementar cualquier realización del método inventivo o sus etapas. Por ejemplo, el sistema inventivo puede ser o incluir un procesador de propósito general programable, procesador de señal digital, o microprocesador, programado con software o firmware y/o de otra forma configurado para llevar a cabo cualquiera de una variedad de operaciones en datos, incluida una realización del método inventivo o sus etapas. Dicho procesador de propósito general puede ser o incluir un sistema informático que incluye un dispositivo de entrada, una memoria y circuitos de procesamiento programados (y/o de otra forma configurados) para llevar a cabo una realización del método inventivo (o sus etapas) en respuesta a datos confirmados a aquella.

Breve descripción de los dibujos

La Figura 1 es un diagrama de bloques de una realización de un sistema que puede configurarse para llevar a cabo una realización del método inventivo.

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 postprocesador acoplado a aquella que es otra realización de la unidad de procesamiento de audio inventiva.

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

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

La Figura 6 es un diagrama del segmento de Información de Tren de Bits (BSI) de una trama AC-3, que incluye los segmentos en los que se divide.

La Figura 7 es un diagrama de una trama E-AC-3, que incluye los segmentos en los que se divide.

La Figura 8 es un diagrama de tramas de un tren de bits de audio codificado que incluye metadatos de límite de programa cuyo formato es según una realización de la invención.

La Figura 9 es un diagrama de otras tramas del tren de bits de audio codificado de la Figura 9. Algunas de estas tramas incluyen metadatos de límite de programa que tienen formato según una realización de la invención.

La Figura 10 es un diagrama de dos trenes de bits de audio codificados: un tren de bits (IEB) en el que un límite de programa (etiquetado "Límite") se alinea con una transición entre dos tramas del tren de bits, y otro tren de bits (TB) en el que un límite de programa (etiquetado "Límite Verdadero") se desplaza 512 muestras desde una transición entre dos tramas del tren de bits.

La Figura 11 es un conjunto de diagramas que muestran cuatro trenes de bits de audio codificados. El tren de bits en la parte superior de la Figura 11 (etiquetado "Escenario 1") es indicativo de un primer programa de audio (P1) que incluye metadatos de límite de programa seguidos de un segundo programa de audio (P2) que también incluye metadatos de límite de programa; el segundo tren de bits (etiquetado "Escenario 2") es indicativo de un primer programa de audio (P1) que incluye metadatos de límite de programa seguidos de un segundo programa de audio (P2) que no incluye metadatos de límite de programa; el tercer tren de bits (etiquetado "Escenario 3") es indicativo de un primer programa de audio truncado (P1) que incluye metadatos de límite de programa, y que se ha empalmado con un segundo programa de audio entero (P2) que incluye metadatos de límite de programa; y el cuarto tren de bits (etiquetado "Escenario 4") es indicativo de un primer programa de audio truncado (P1) que incluye metadatos de límite de programa, y un segundo programa de audio truncado (P2) que incluye metadatos de límite de programa y que se ha empalmado con una porción del primer programa de audio.

Notación y Nomenclatura

A lo largo de la presente descripción, incluidas las reivindicaciones, la expresión llevar a cabo una operación "en" una señal o datos (p. ej., filtrar, escalar, transformar o aplicar una ganancia a, la señal o datos) se usa en un sentido amplio para denotar llevar a cabo la operación directamente en la señal o datos, o en una versión procesada de la señal o datos (p. ej., en una versión de la señal que ha experimentado el filtrado preliminar o preprocesamiento previo a llevar a cabo la operación en aquellos).

A lo largo de la presente descripción, incluidas las reivindicaciones, la expresión "sistema" se usa en un sentido amplio para denotar un dispositivo, sistema o subsistema. Por ejemplo, a un subsistema que implementa un descodificador puede hacerse referencia como sistema de descodificador, y a un sistema que incluye dicho subsistema (p. ej., un sistema que genera X señales de salida en respuesta a múltiples entradas, en el que el subsistema genera M de las entradas y las otras X - M entradas se reciben de una fuente externa) también puede hacerse referencia como sistema de descodificador.

A lo largo de la presente descripción, incluidas las reivindicaciones, el término "procesador" se usa en un sentido amplio para denotar un sistema o dispositivo programable o de otra forma configurable (p. ej., con software o firmware) para llevar a cabo operaciones en datos (p. ej., audio o vídeo u otros datos de imagen). Ejemplos de procesadores incluyen una matriz de puertas programables en campo (u otro circuito integrado configurable o conjunto de chips), un procesador de señal digital programado y/o de otra forma configurado para llevar a cabo el procesamiento canalizado en datos de audio u otros datos de sonido, un procesador u ordenador de propósito general programable y un conjunto de chips o un chip de microprocesador programable.

A lo largo de la presente descripción, incluidas las reivindicaciones, las expresiones "procesador de audio" y "unidad de procesamiento de audio" se usan de manera intercambiable y en un sentido amplio para denotar un sistema configurado para procesar datos de audio. Ejemplos de unidades de procesamiento de audio incluyen, pero sin limitación, codificadores (p. ej., transcodificadores), descodificadores, códecs, sistemas de preprocesamiento, sistemas de postprocesamiento y sistemas de procesamiento de tren de bits (a los que algunas veces se hace referencia como herramientas de procesamiento de tren de bits).

A lo largo de la presente descripción, incluidas las reivindicaciones, la expresión "metadatos de estado de procesamiento" (p. ej., como en la expresión "metadatos de estado de procesamiento de sonoridad") se refiere a datos separados y diferentes de los datos de audio correspondientes (el contenido de audio de un tren de datos de audio que también incluye metadatos de estado de procesamiento). Los metadatos de estado de procesamiento se asocian a datos de audio, indican el estado de procesamiento de sonoridad de los datos de audio correspondientes (p. ej., qué tipo(s) de procesamiento ya se ha llevado a cabo en los datos de audio) y, normalmente, también indican al menos un rasgo o una característica de los datos de audio. La asociación de los metadatos de estado de procesamiento a los datos de audio es síncrona en el tiempo. Por consiguiente, los metadatos de estado de procesamiento presentes (recibidos o actualizados más recientemente) indican que los datos de audio correspondientes comprenden, de manera contemporánea, los resultados del tipo(s) indicado(s) de procesamiento de datos de audio. En algunos casos, los metadatos de estado de procesamiento pueden incluir historial de procesamiento y/o algunos o todos los parámetros que se usan en los tipos de procesamiento indicados y/o derivan de estos. Además, los metadatos de estado de procesamiento pueden incluir al menos un rasgo o una característica de los datos de audio correspondientes, que se han computado o extraído de los datos de audio. Los metadatos de estado de procesamiento también pueden incluir otros metadatos que no se relacionan o se derivan de cualquier 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 preferencias de usuario, etc. pueden ser añadidos por una unidad de procesamiento de audio particular para pasar a las otras unidades de procesamiento de audio.

A lo largo de la presente descripción, incluidas las reivindicaciones, la expresión "metadatos de estado de procesamiento de sonoridad" (o "LPSM") denota metadatos de estado de procesamiento indicativos del estado de procesamiento de sonoridad de datos de audio correspondientes (p. ej., qué tipo(s) de procesamiento de sonoridad se ha llevado a cabo en los datos de audio) y, normalmente, también al menos un rasgo o una característica (p. ej., sonoridad) de los datos de audio correspondientes. Los metadatos de estado de procesamiento de sonoridad pueden incluir datos (p. ej., otros metadatos) que no son los metadatos de estado de procesamiento de sonoridad (a saber, cuando se los considera solos).

A lo largo de la presente descripción, incluidas las reivindicaciones, la expresión "canal" (o "canal de audio") denota una señal de audio monofónica.

A lo largo de la presente descripción, incluidas las reivindicaciones, la expresión "programa de audio" denota un conjunto de uno o más canales de audio y, de forma opcional, también metadatos asociados (p. ej., metadatos que describen una presentación de audio espacial deseada y/o LPSM y/o metadatos de límite de programa).

A lo largo de la presente descripción, incluidas las reivindicaciones, la expresión "metadatos de límite de programa" denota metadatos de un tren de bits de audio codificado, donde el tren de bits de audio codificado es indicativo de al menos un programa de audio (p. ej., dos o más programas de audio), y los metadatos de límite de programa son indicativos de la ubicación en el tren de bits de al menos un límite (comienzo y/o fin) de al menos un programa de audio. Por ejemplo, los metadatos de límite de programa (de un tren de bits de audio codificado indicativo de un programa de audio) pueden incluir metadatos indicativos de la ubicación (p. ej., el inicio de la "N"ésima trama del tren de bits, o la "M"ésima ubicación de muestra de la "N"ésima trama del tren de bits) del comienzo del programa, y metadatos adicionales indicativos de la ubicación (p. ej., el inicio de la "J"ésima trama del tren de bits, o la "K"ésima ubicación de muestra de la "J"ésima trama del tren de bits) del fin del programa.

A lo largo de la presente descripción, incluidas las reivindicaciones, el término "se acopla" o "acoplado" se usa para significar una conexión directa o indirecta. Por consiguiente, 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 mediante otros dispositivos y conexiones.

Descripción detallada de realizaciones de la invención

Según realizaciones típicas de la invención, una carga útil de metadatos de sonoridad de programa, a los que se hace referencia como metadatos de estado de procesamiento de sonoridad ("LPSM") y, de forma opcional, también metadatos de límite de programa se incorporan en uno o más campos reservados (o intervalos) de segmentos de metadatos de un tren de bits de audio que también incluyen datos de audio en otros segmentos (segmentos de datos de audio). Normalmente, al menos un segmento de cada trama del tren de bits incluye LPSM, y al menos otro segmento de la trama incluye datos de audio correspondientes (a saber, datos de audio cuyo estado de procesamiento de sonoridad y sonoridad es indicado por los LPSM). En algunas realizaciones, el volumen de datos de los LPSM puede ser suficientemente pequeño como para transportarse sin afectar la velocidad de bits asignada para transportar los datos de audio.

Comunicar los metadatos de estado de procesamiento de sonoridad en una cadena de procesamiento de datos de audio es particularmente útil cuando dos o más unidades de procesamiento de audio necesitan funcionar en serie entre sí a lo largo de la cadena de procesamiento (o vida útil del contenido). Sin inclusión de metadatos de estado de procesamiento de sonoridad en un tren de bits de audio, pueden ocurrir problemas serios de procesamiento de medios como, por ejemplo, degradaciones de calidad, nivel y espacial, por ejemplo, cuando se utilizan dos o más códecs de audio en la cadena y se aplica nivelación de volumen de extremo único más de una vez durante el trayecto del tren de bits a un dispositivo de consumo de medios (o un punto de renderización del contenido de audio del tren de bits).

La Figura 1 es un diagrama de bloques de una cadena de procesamiento de audio a modo de ejemplo (un sistema de procesamiento de datos de audio), en el que uno o más de los elementos del sistema pueden configurarse según una realización de la presente invención. El sistema incluye los siguientes elementos, acoplados, de manera conjunta, como se muestra: una unidad de preprocesamiento, un codificador, una unidad de análisis de señal y corrección de metadatos, un transcodificador, un descodificador y una unidad de preprocesamiento. En variaciones del sistema que se muestra, uno o más de los elementos se omiten, o se incluyen unidades de procesamiento de datos de audio adicionales.

En algunas implementaciones, la unidad de preprocesamiento de la Figura 1 se configura para aceptar muestras PCM (dominio temporal) que comprenden contenido de audio como entrada, y para producir muestras PCM procesadas. El codificador puede configurarse para aceptar las muestras PCM como entrada y para producir un tren de bits de audio codificado (p. ej., comprimido) indicativo del contenido de audio. A los datos del tren de bits que son indicativos del contenido de audio, en la presente memoria, a veces se les hace referencia como "datos de audio". Si el codificador se configura según una realización típica de la presente invención, el tren de bits de audio producido desde el codificador incluye metadatos de estado de procesamiento de sonoridad (y, normalmente, también otros metadatos, incluidos, de forma opcional, metadatos de límite de programa) así como datos de audio.

La unidad de análisis de señal y corrección de metadatos de la Figura 1 puede aceptar uno o más trenes de bits de audio codificados como entrada y determinar (p. ej., validar) si los metadatos de estado de procesamiento en cada tren de bits de audio codificado son correctos, llevando a cabo el análisis de la señal (p. ej., mediante el uso de metadatos de límite de programa en un tren de bits de audio codificado). Si la unidad de análisis de señal y corrección de metadatos descubre que los metadatos incluidos no son válidos, normalmente reemplaza los valores incorrectos por los valores correctos obtenidos del análisis de la señal. Por consiguiente, cada tren de bits de audio codificado producido desde la unidad de análisis de señal y corrección de metadatos puede incluir metadatos de estado de procesamiento corregidos (o no corregidos) así como datos de audio codificados.

El transcodificador de la Figura 1 puede aceptar trenes de bits de audio codificados como entrada, y producir trenes de bits de audio modificados (p. ej., codificados de manera diferente) en respuesta (p. ej., mediante la descodificación de un tren de entrada y la recodificación del tren descodificado en un formato de codificación diferente). Si el transcodificador se configura según una realización típica de la presente invención, el tren de bits de audio producido desde el transcodificador incluye metadatos de estado de procesamiento de sonoridad (y, normalmente, también otros metadatos) así como datos de audio codificados. Los metadatos pueden haberse incluido en el tren de bits.

El descodificador de la Figura 1 puede aceptar trenes de bits de audio codificados (p. ej., comprimidos) como entrada, y producir (en respuesta) trenes de muestras de audio PCM descodificadas. Si el descodificador se configura según una realización típica de la presente invención, la salida del descodificador en una operación típica es o incluye cualquiera de los siguientes:

un tren de muestras de audio, y un tren correspondiente de metadatos de estado de procesamiento de sonoridad (y, normalmente, también otros metadatos) extraídos de un tren de bits codificado de entrada; o

un tren de muestras de audio, y un tren correspondiente de bits de control determinado a partir de los metadatos de estado de procesamiento de sonoridad (y, normalmente, también otros metadatos) extraídos de un tren de bits codificado de entrada; o

un tren de muestras de audio, sin un tren correspondiente de metadatos de estado de procesamiento o bits de control determinados a partir de los metadatos de estado de procesamiento. En este último caso, el descodificador puede extraer metadatos de estado de procesamiento de sonoridad (y/u otros metadatos) del tren de bits codificado de entrada y llevar a cabo al menos una operación en los metadatos extraídos (p. ej., validación), aunque no produce los metadatos extraídos o bits de control determinados a partir de aquellos.

Mediante la configuración de la unidad de postprocesamiento de la Figura 1 según una realización típica de la presente invención, la unidad de postprocesamiento se configura para aceptar un tren de muestras de audio PCM descodificadas y para llevar a cabo el postprocesamiento en aquellas (p. ej., nivelación de volumen del contenido de audio) mediante el uso de metadatos de estado de procesamiento de sonoridad (y, normalmente, también otros metadatos) recibidos con las muestras, o bits de control (determinados por el descodificador a partir de los metadatos de estado de procesamiento de sonoridad y, normalmente, también otros metadatos) recibidos con las muestras. La unidad de postprocesamiento se configura también, normalmente, para renderizar el contenido de audio postprocesado para la reproducción por uno o más altavoces.

Las realizaciones típicas de la presente invención proveen una cadena de procesamiento de audio mejorada en la que las unidades de procesamiento de audio (p. ej., codificadores, descodificadores, transcodificadores y unidades de preprocesamiento y postprocesamiento) adaptan su respectivo procesamiento para aplicarlo a datos de audio según un estado contemporáneo de los datos de medios según se indica por los metadatos de estado de procesamiento de sonoridad respectivamente recibidos por las unidades de procesamiento de audio.

Los datos de audio ingresados en cualquier unidad de procesamiento de audio del sistema de la Figura 1 (p. ej., el codificador o transcodificador de la Figura 1) pueden incluir metadatos de estado de procesamiento de sonoridad (y, de manera opcional, también otros metadatos) así como datos de audio (p. ej., 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 se muestra en la Figura 1) según una realización de la presente invención. La unidad de procesamiento que recibe el audio de entrada (con metadatos) puede configurarse para llevar a cabo al menos una operación en los metadatos (p. ej., validación) o en respuesta a los metadatos (p. ej., procesamiento adaptativo del audio de entrada) y, normalmente, también para incluir en su audio de salida los metadatos, una versión procesada de los metadatos, o bits de control determinados a partir de los metadatos.

Una realización típica de la unidad de procesamiento de audio inventiva (o procesador de audio) se configura para llevar a cabo el procesamiento adaptativo de datos de audio según el estado de los datos de audio como se indica por los metadatos de estado de procesamiento de sonoridad correspondientes a los datos de audio. En algunas realizaciones, el procesamiento adaptativo es (o incluye) el procesamiento de sonoridad (si los metadatos indican que el procesamiento de sonoridad, o procesamiento similar a aquel, no se ha llevado a cabo ya 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 a aquel, ya se ha llevado a cabo en los datos de audio). En algunas realizaciones, el procesamiento adaptativo es o incluye la validación de metadatos (p. ej., llevada a cabo en una subunidad de validación de metadatos) para asegurar que la unidad de procesamiento de audio lleve a cabo otro procesamiento adaptativo de los datos de audio basándose el estado de los datos de audio según se indica por los metadatos de estado de procesamiento de sonoridad. En algunas realizaciones, la validación determina la fiabilidad de los metadatos de estado de procesamiento de sonoridad asociados a (p. ej., incluidos en un tren de bits con) los datos de audio. Por ejemplo, si los metadatos se validan para ser fiables, entonces los resultados de un tipo de procesamiento de audio llevado a cabo previamente pueden reusarse y puede evitarse llevar a cabo otra vez el mismo tipo de procesamiento de audio. Por otro lado, si se descubre que los metadatos se han alterado (o de otra forma no son fiables), entonces el tipo de procesamiento de medios supuestamente llevado a cabo de forma previa (según se indica por los metadatos no fiables) puede ser repetido por la unidad de procesamiento de audio y/o puede llevarse a cabo otro procesamiento por la unidad de procesamiento de audio en los metadatos y/o datos de audio. La unidad de procesamiento de audio también puede configurarse para señalizar a otras unidades de procesamiento de audio corriente abajo en una cadena de procesamiento de medios mejorada que los metadatos de estado de procesamiento de sonoridad (p. ej., presentes en un tren de bits de medios) son válidos, si la unidad determina que los metadatos de estado de procesamiento son válidos (p. ej., según una concordancia de un valor criptográfico extraído y un valor criptográfico de referencia).

La Figura 2 es un diagrama de bloques de un codificador (100) que es una realización de la unidad de procesamiento de audio inventiva. Cualquiera de los componentes o elementos del codificador 100 pueden implementarse como uno o más procesos y/o uno o más circuitos (p. ej., ASIC, FPGA, u otros circuitos integrados), en hardware, software o una combinación de hardware y software. El codificador 100 comprende búfer de trama 110, analizador 111, descodificador 101, validador de estado de audio 102, fase de procesamiento de sonoridad 103, fase de selección de tren de audio 104, codificador 105, fase de relleno/formateador 107, fase de generación de metadatos 106, subsistema de medición de sonoridad de diálogo 108 y búfer de trama 109, conectados como se muestra. También normalmente, el codificador 100 incluye otros elementos de procesamiento (no se muestran).

El codificador 100 (que es un transcodificador) se configura para convertir un tren de bits de audio de entrada (que, por ejemplo, puede ser uno de un tren de bits AC-3, un tren de bits E-AC-3, o un tren de bits Dolby E) en un tren de bits de audio de salida codificado (que, por ejemplo, puede ser otro de un tren de bits AC-3, un tren de bits E-AC-3, o un tren de bits Dolby E) incluso al llevar a cabo el procesamiento de sonoridad adaptativo y automático mediante el uso de los metadatos de estado de procesamiento de sonoridad incluidos en el tren de bits de entrada. Por ejemplo, el codificador 100 puede configurarse para convertir un tren de bits Dolby E de entrada (un formato normalmente usado en las instalaciones de producción y difusión pero no en dispositivos de consumidor que reciben programas de audio que se han difundido a aquellos) en un tren de bits de audio de salida codificado (apropiado para la difusión a dispositivos de consumidor) en formato AC-3 o E-AC-3.

El sistema de la Figura 2 también incluye un subsistema de entrega de audio codificado 150 (que almacena y/o entrega los trenes de bits codificados producidos desde el codificador 100) y descodificador 152. Un tren de bits de audio codificado producido desde el codificador 100 puede almacenarse por el subsistema 150 (p. ej., en forma de DVD o disco Blu-ray) o transmitirse por el subsistema 150 (que puede implementar una red o enlace de transmisión), o puede tanto almacenarse como transmitirse por el subsistema 150. El descodificador 152 se configura para descodificar un tren de bits de audio codificado (generado por el codificador 100) que recibe mediante el subsistema 150, incluido mediante la extracción de metadatos de estado de procesamiento de sonoridad (LPSM) de cada trama del tren de bits (y, de manera opcional, también mediante la extracción de metadatos de límite de programa del tren de bits) y generar

datos de audio descodificados. Normalmente, el descodificador 152 se configura para llevar a cabo el procesamiento de sonoridad adaptativo en los datos de audio descodificados mediante el uso de los LPSM (y, de manera opcional, también metadatos de límite de programa) y/o para reenviar los datos de audio descodificados y LPSM a un postprocesador configurado para llevar a cabo el procesamiento de sonoridad adaptativo en los datos de audio descodificados mediante el uso de los LPSM (y, de manera opcional, también metadatos de límite de programa). Normalmente, el descodificador 152 incluye un búfer que almacena (p. ej., de una manera no transitoria) el tren de bits de audio codificado recibido del subsistema 150.

Varias implementaciones del codificador 100 y descodificador 152 se configuran para llevar a cabo diferentes realizaciones del método inventivo. El búfer de trama 110 es una memoria intermedia acoplada para recibir un tren de bits de audio de entrada codificado. Durante el funcionamiento, el búfer 110 almacena (p. ej., de una manera no transitoria) al menos una trama del tren de bits de audio codificado, y una secuencia de las tramas del tren de bits de audio codificado se confirma del búfer 110 al analizador 111.

El analizador 111 se acopla y configura para extraer metadatos de estado de procesamiento de sonoridad (LPSM) y, de manera opcional, también metadatos de límite de programa (y/u otros metadatos) de cada trama del audio de entrada codificado en el que se incluyen dichos metadatos, para confirmar al menos los LPSM (y, de manera opcional, también metadatos de límite de programa y/u otros metadatos) al validador de estado de audio 102, fase de procesamiento de sonoridad 103, fase 106 y subsistema 108, para extraer datos de audio del audio de entrada codificado, y para confirmar los datos de audio al descodificador 101. El descodificador 101 del codificador 100 se configura para descodificar los datos de audio para generar datos de audio descodificados, y para confirmar los datos de audio descodificados a la fase de procesamiento de sonoridad 103, fase de selección de tren de audio 104, subsistema 108 y, normalmente, también al validador de estado 102.

El validador de estado 102 se configura para autenticar y validar los LPSM (y, de manera opcional, otros metadatos) confirmados a aquel. En algunas realizaciones, los LPSM son (o se incluyen en) un bloque de datos que se ha incluido en el tren de bits de entrada (p. ej., según una realización de la presente invención). El bloque puede comprender un hash criptográfico (un código de autenticación de mensaje basado en hash o "HMAC", por sus siglas en inglés) para procesar los LPSM (y, de manera opcional, también otros metadatos) y/o los datos de audio subyacentes (proporcionadas desde el descodificador 101 al validador 102). El bloque de datos puede firmarse de manera digital en estas realizaciones, de modo que una unidad de procesamiento de audio corriente abajo puede autenticar y validar, de forma relativamente fácil, los metadatos de estado de procesamiento.

Por ejemplo, el HMAC se usa para generar un resumen y los valores de protección incluidos en el tren de bits inventivo pueden incluir el resumen. El resumen puede generarse de la siguiente manera para una trama AC-3:

1. Después de codificar datos AC-3 y LPSM, los bytes de datos de trama (datos_trama #1 y datos_trama #2 concatenados) y los bytes de datos LPSM se usan como entrada para el HMAC de la función de troceado. Otros datos, que pueden estar presentes dentro de un campo auxdata, no se tienen en cuenta para calcular el resumen. Dichos otros datos pueden ser bytes que no pertenecen a los datos AC-3 ni a los datos LPSM. Los bits de protección incluidos en LPSM pueden no considerarse para calcular el resumen de HMAC.

2. Después de calcular el resumen, se escribe en el tren de bits en un campo reservado para los bits de protección.

3. La última etapa de la generación de la trama AC-3 completa es el cálculo del control CRC. Ello se escribe al final de la trama y se tienen en cuenta todos los datos pertenecientes a dicha trama, incluidos los bits LPSM.

Otros métodos criptográficos que incluyen, pero sin limitación, cualquiera de uno o más métodos criptográficos diferentes de HMAC pueden usarse para la validación de LPSM (p. ej., en el validador 102) para garantizar la transmisión y recepción seguras de los LPSM y/o datos de audio subyacentes. Por ejemplo, puede llevarse a cabo validación (mediante el uso de dicho método criptográfico) en cada unidad de procesamiento de audio que recibe una realización del tren de bits de audio inventivo para determinar si los metadatos de estado de procesamiento de sonoridad y los datos de audio correspondientes incluidos en el tren de bits se han sometido al (y/o han resultado del) procesamiento de sonoridad específico (según se indica por los metadatos) y no se han modificado después de llevar a cabo el procesamiento de sonoridad específico.

El validador de estado 102 confirma los datos de control a la fase de selección de tren de audio 104, generador de metadatos 106 y 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 fase 104 puede seleccionar (y transmitir al codificador 105) ya sea:

la salida procesada adaptativamente de la fase de procesamiento de sonoridad 103 (p. ej., cuando los LPSM indican que los datos de audio producidos desde el descodificador 101 no se han sometido a un tipo específico de procesamiento de sonoridad, y los bits de control desde el validador 102 indican que los LPSM son válidos); o

los datos de audio producidos desde el descodificador 101 (p. ej., cuando los LPSM indican que los datos de audio producidos desde el descodificador 101 se han sometido a un tipo específico de procesamiento de sonoridad que se llevaría a cabo por la fase 103, y los bits de control del validador 102 indican que los LPSM son válidos).

La fase 103 del codificador 100 se configura para llevar a cabo procesamiento de sonoridad adaptativo en los datos de audio descodificados producidos desde el descodificador 101, basándose en una o más características de datos de audio indicadas por los LPSM extraídos por el descodificador 101. La fase 103 puede ser un procesador adaptativo de control de rango dinámico y sonoridad en tiempo real de dominio de transformada. La fase 103 puede recibir una entrada de usuario (p. ej., valores de sonoridad/rango dinámico de destino de usuario o valores dialnorm) u otra entrada de metadatos (p. ej., uno o más tipos de datos de terceros, información de seguimiento, identificadores, información de propiedad o estándar, datos de anotación de usuario, datos de preferencias de usuario, etc.) y/u otra entrada (p. ej., de un proceso de huella digital) y usar dicha entrada para procesar los datos de audio descodificados producidos desde el descodificador 101. La fase 103 puede llevar a cabo el procesamiento de sonoridad adaptativo en datos de audio descodificados (producidos desde el descodificador 101) indicativos de un solo programa de audio (según se indica por los metadatos de límite de programa extraídos por el analizador 111) y puede restablecer el procesamiento de sonoridad en respuesta a la recepción de datos de audio descodificados (producidos desde el descodificador 101) indicativos de un programa de audio diferente según se indica por metadatos de límite de programa extraídos por el analizador 111.

El subsistema de medición de sonoridad de diálogo 108 puede funcionar para determinar la sonoridad de segmentos del audio descodificado (desde el descodificador 101) que son indicativos de diálogo (u otra voz), p. ej., mediante el uso de los 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. El funcionamiento del subsistema de medición de sonoridad de diálogo 108 puede inhabilitarse cuando los LPSM indican sonoridad previamente determinada de segmentos de diálogo (u otra voz) del audio descodificado (del descodificador 101) cuando los bits de control del validador 102 indican que los LPSM son válidos. El subsistema 108 puede llevar a cabo una medición de sonoridad en datos de audio descodificados indicativos de un solo programa de audio (según se indica por los metadatos de límite de programa extraídos por el analizador 111) y puede restablecer la medición en respuesta a la recepción de datos de audio descodificados indicativos de un programa de audio diferente según se indica por dichos metadatos de límite de programa.

Existen herramientas útiles (p. ej., el medidor de sonoridad Dolby LM100) para medir el nivel de diálogo en contenido de audio de manera conveniente y fácil. Algunas realizaciones de la APU inventiva (p. ej., la fase 108 del codificador 100) se implementan para incluir (o para llevar a cabo las funciones de) dicha herramienta para medir la sonoridad de diálogo media de contenido de audio de un tren de bits de audio (p. ej., un tren de bits AC-3 descodificado confirmado a la fase 108 desde el descodificador 101 del codificador 100).

Si la fase 108 se implementa para medir la verdadera sonoridad de diálogo media de los datos de audio, la medición puede incluir una etapa de aislamiento de segmentos del contenido de audio que contienen, principalmente, voz. Los segmentos de audio que son, principalmente, voz se procesan entonces según un algoritmo de medición de sonoridad. Para datos de audio descodificados desde un tren de bits AC-3, este algoritmo puede ser una medida de sonoridad ponderada K estándar (según el estándar internacional ITU-R BS.1770). De manera alternativa, pueden usarse otras medidas de sonoridad (p. ej., aquellas basadas en modelos psicoacústicos de sonoridad).

El aislamiento de segmentos de voz no es esencial para medir la sonoridad de diálogo media de los datos de audio. Sin embargo, mejora la exactitud de la medida y, normalmente, provee resultados más satisfactorios desde la perspectiva de un oyente. Dado que no todo el contenido de audio contiene diálogo (voz), la medida de sonoridad de todo el contenido de audio puede proporcionar una aproximación suficiente del nivel de diálogo del audio, si la voz hubiera estado presente.

El generador de metadatos 106 genera (y/o transmite a la fase 107) metadatos que se incluirán por la fase 107 en el tren de bits codificado que se producirán desde el codificador 100. El generador de metadatos 106 puede transmitir a la fase 107 los LPSM (y, de manera opcional, también metadatos de límite de programa y/u otros metadatos) extraídos por el codificador 101 y/o analizador 111 (p.ej., cuando los bits de control del validador 102 indican que los LPSM y/u otros metadatos son válidos), o generar nuevos LPSM (y, de manera opcional, también metadatos de límite de programa y/u otros metadatos) y confirmar los nuevos metadatos a la fase 107 (p. ej., cuando los bits de control del validador 102 indican que los LPSM y/u otros metadatos extraídos por el descodificador 101 no son válidos), o puede confirmar a la fase 107 una combinación de metadatos extraídos por el descodificador 101 y/o analizador 111 y 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 llevado a cabo por el subsistema 108, en los LPSM que confirma a la fase 107 para su inclusión en el tren de bits codificado que se producirá desde el codificador 100.

El generador de metadatos 106 puede generar bits de protección (que pueden consistir en un código de autenticación de mensaje basado en hash o "HMAC", o incluir este) útil para al menos una de desencriptación, autenticación o validación de los LPSM (y, de manera opcional, también otros metadatos) que se incluirán en el tren de bits codificado y/o los datos de audio subyacentes que se incluirán en el tren de bits codificado. El generador de metadatos 106 puede proveer dichos bits de protección a la fase 107 para su inclusión en el tren de bits codificado.

En el funcionamiento típico, el subsistema de medición de sonoridad de diálogo 108 procesa los datos de audio producidos desde el descodificador 101 para generar en respuesta a aquellos valores de sonoridad (p. ej., valores de sonoridad de diálogo con puerta y sin puerta) y valores de rango 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 tren de bits codificado que se producirá desde el codificador 100.

De manera adicional, opcional o alternativa, los subsistemas de 106 y/o 108 del codificador 100 pueden llevar a cabo 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 tren de bits codificado que se producirá desde la fase 107.

El codificador 105 codifica (p. ej., llevando a cabo compresión) los datos de audio producidos desde la fase de selección 104, y confirma el audio codificado a la fase 107 para su inclusión en el tren de bits codificado que se producirá desde la fase 107.

La fase 107 multiplexa el audio codificado del codificador 105 y los metadatos (incluidos los LPSM) del generador 106 para generar el tren de bits codificado que se producirá desde la fase 107, preferiblemente de modo que el tren de bits codificado tenga un formato según se especifica por una realización preferida de la presente invención.

El búfer de trama 109 es una memoria intermedia que almacena (p. ej., de una manera no transitoria) al menos una trama del tren de bits de audio codificado producido desde la fase 107, y una secuencia de las tramas del tren de bits de audio codificado se confirma entonces desde el búfer 109 como salida del codificador 100 al sistema de entrega 150.

Los LPSM generados por el generador de metadatos 106 e incluidos en el tren de bits codificado por la fase 107 son indicativos del estado de procesamiento de sonoridad de datos de audio correspondientes (p. ej., qué tipo de procesamiento de sonoridad se ha llevado a cabo en los datos de audio) y sonoridad (p. ej., sonoridad de diálogo medida, sonoridad con puerta y/o sin puerta, y/o rango dinámico) de los datos de audio correspondientes.

En la presente memoria, la "puerta" de sonoridad y/o mediciones de nivel llevadas a cabo en los datos de audio se refiere a un nivel específico o umbral de sonoridad donde los valores computados que superan el umbral se incluyen en la medición final (p. ej., ignorando valores de sonoridad a corto plazo por debajo de -60 dBFS en los valores medidos finales). Las puertas en un valor absoluto se refieren a una sonoridad o nivel fijo, mientras que las puertas en un valor relativo se refieren a un valor que depende de un valor de medición "sin puerta" actual.

En algunas implementaciones del codificador 100, el tren de bits codificado almacenado en la memoria 109 (y sacado al sistema de entrega 150) es un tren de bits AC-3 o un tren de bits E-AC-3, y comprende segmentos de datos de audio (p. ej., los segmentos BA0-BA5 de la trama que se muestra en la Figura 4) y segmentos de metadatos, 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 metadatos de estado de procesamiento de sonoridad (LPSM). La fase 107 inserta LPSM (y, de manera opcional, también metadatos de límite de programa) en el tren de bits en el siguiente formato. Cada uno de los segmentos de metadatos que incluye LPSM (y, de manera opcional, también metadatos de límite de programa) se incluye en un segmento de bits de residuo del tren de bits (p. ej., un segmento de bits de residuo "W" según se muestra en la Figura 4 o la Figura 7), o un campo "addbsi" del segmento de Información de Tren de Bits ("BSI") de una trama del tren de bits, o un campo auxdata (p. ej., el segmento AUX que se muestra en la Figura 4 o Figura 7) al final de una trama del tren de bits. Una trama del tren de bits puede incluir uno o dos segmentos de metadatos, cada uno de los cuales incluye LPSM, y si la trama incluye dos segmentos de metadatos, uno puede estar presente en el campo addbsi de la trama y el otro puede estar presente en el campo AUX de la trama. En algunas realizaciones, cada segmento de metadatos, que incluye LPSM, incluye un segmento de carga útil LPSM (o contenedor) que tiene el siguiente formato:

un encabezamiento (que normalmente incluye una palabra de sincronización que identifica el inicio de la carga útil de LPSM), seguido de al menos un valor de identificación, p. ej., versión de formato de LPSM, longitud, período, cómputo y valores de asociación de subtren indicados en la Tabla 2 más abajo); y

después del encabezamiento,

al menos un valor de indicación de diálogo (p. ej., parámetro "Canal(es) de diálogo" de la Tabla 2) que indica si los datos de audio correspondientes indican diálogo o no indican diálogo (p. ej., qué canales de los datos de audio correspondientes indican diálogo);

al menos un valor de cumplimiento de regulación de sonoridad (p. ej., el parámetro "Tipo de Regulación de Sonoridad" de la Tabla 2) que indica si los datos de audio correspondientes cumplen con un conjunto indicado de regulaciones de sonoridad;

al menos un valor de procesamiento de sonoridad (p. ej., uno o más de los parámetros "bandera de Corrección de Sonoridad con puerta de Diálogo", "Tipo de Corrección de Sonoridad", de la Tabla 2) que indica al menos un tipo de procesamiento de sonoridad que se ha llevado a cabo en los datos de audio correspondientes; y

al menos un valor de sonoridad (p. ej., uno o más de los parámetros "Sonoridad con Puerta Relativa a ITU", "Sonoridad con Puerta de Voz ITU", "Sonoridad 3s a Corto Plazo ITU (EBU 3341)", y "Pico Verdadero" de la Tabla 2) que indica al menos una característica de sonoridad (p. ej., sonoridad pico o promedio) de los datos de audio correspondientes.

En algunas realizaciones, cada segmento de metadatos que contiene LPSM y metadatos de límite de programa contiene un encabezamiento principal (y, de forma opcional, también elementos principales adicionales), y después del encabezamiento principal (o el encabezamiento principal y otros elementos principales) un segmento de carga útil de LPSM (o contenedor) que tiene el siguiente formato:

un encabezamiento, que normalmente incluye al menos un valor de identificación (p. ej., versión de formato de LPSM, longitud, período, cómputo y valores de asociación de subtren, como se indica en la Tabla 2 presentada en la presente memoria), y

después del encabezamiento, los LPSM y metadatos de límite de programa. Los metadatos de límite de programa pueden incluir un cómputo de tramas de límite de programa y un valor de código (p. ej., un valor "desplazamiento_existe") indicativo de si la trama incluye solamente un cómputo de tramas de límite de programa o tanto un cómputo de tramas de límite de programa como un valor de desplazamiento), y (en algunos casos) un valor de desplazamiento.

En algunas implementaciones, cada uno de los segmentos de metadatos insertados por la fase 107 en un segmento de bits de residuo o un campo "addbsi" o un campo auxdata de una trama del tren de bits tiene el siguiente formato:

un encabezamiento principal (que, normalmente, incluye una palabra de sincronización que identifica el inicio del segmento de metadatos, seguida de valores de identificación, p. ej., la versión de elemento Principal, longitud y período, cómputo de elementos extendidos, y valores de asociación de subtren indicados en la Tabla 1 más abajo); y

después del encabezamiento principal, al menos un valor de protección (p. ej., el resumen de HMAC y valores de Huella Digital de Audio de la Tabla 1) útil para al menos una de desencriptación, autenticación o validación de al menos uno de los metadatos de estado de procesamiento de sonoridad o los datos de audio correspondientes); y

también después del encabezamiento principal, si el segmento de metadatos incluye LPSM, la identificación ("ID") de carga útil de LPSM y valores de tamaño de carga útil de LPSM que identifican los siguientes metadatos como una carga útil de LPSM e indican el tamaño de la carga útil de LPSM.

El segmento de carga útil de LPSM (o contenedor) (que, preferiblemente, tiene el formato especificado más arriba) sigue a la ID de carga útil de LPSM y valores de tamaño de carga útil LPSM.

En algunas realizaciones, cada uno de los segmentos de metadatos en el campo auxdata (o campo "addbsi") de una trama tiene tres niveles de estructura:

una estructura de nivel alto, que indica una bandera que indica si el campo auxdata (o addbsi) incluye metadatos, al menos un valor de ID que indica qué tipo de metadatos están presentes y, normalmente, también un valor que indica cuántos bits de metadatos (p. ej., de cada tipo) están presentes (si hay presentes metadatos). Un tipo de metadatos que pueden estar presentes son los LSPM, otro tipo de metadatos que pueden estar presentes son los metadatos de límite de programa, y otro tipo de metadatos que pueden estar presentes son los metadatos de búsqueda de medios (p. ej., metadatos de Búsqueda de Medios de Nielsen);

una estructura de nivel intermedio, que comprende un elemento principal para cada tipo identificado de metadatos (p. ej., encabezamiento principal, valores de protección e ID de carga útil de LPSM y valores de tamaño de carga útil de LPSM, como se menciona más arriba, para cada tipo identificado de metadatos); y

una estructura de nivel bajo, que comprende cada carga útil para un elemento principal (p. ej., una carga útil de LPSM, si una es identificada por el elemento principal como presente, y/o una carga útil de metadatos de otro tipo, si una es identificada por el elemento principal como presente).

Los valores de datos en dicha estructura de tres niveles pueden anidarse. Por ejemplo, el valor(es) de protección para una carga útil LPSM y/u otra carga útil de metadatos identificada por un elemento principal pueden incluirse después de cada carga útil identificada por el elemento principal (y, por consiguiente, después del encabezamiento principal del elemento principal). En un ejemplo, un encabezamiento principal puede identificar una carga útil de LPSM y otra carga útil de metadatos, ID de carga útil y valores de tamaño de carga útil para la primera carga útil (p. ej., la carga útil de LPSM) pueden seguir al encabezamiento principal, la propia primera carga útil puede seguir a la ID y valores de tamaño, la ID de carga útil y valor de tamaño de carga útil para la segunda carga útil pueden seguir a la primera carga útil, la propia segunda carga útil puede seguir a estas ID y valores de tamaño, y bits de protección para ambas cargas útiles (o para valores de elementos principales y ambas cargas útiles) pueden seguir a la última carga útil.

En algunas realizaciones, si el descodificador 101 recibe un tren de bits de audio generado según una realización de la invención con hash criptográfico, el descodificador se configura para analizar y recuperar el hash criptográfico de un bloque de datos determinado del tren de bits, dicho bloque comprende metadatos de estado de procesamiento de sonoridad (LPSM) y, de manera opcional, también metadatos de límite de programa. El validador 102 puede usar el hash criptográfico para validar el tren de bits recibido y/o metadatos asociados. Por ejemplo, si el validador 102 descubre que los LPSM son válidos basándose una concordancia entre un hash criptográfico de referencia y el hash criptográfico recuperado del bloque de datos, entonces puede inhabilitar la operación del procesador 103 en los datos

de audio correspondientes y hacer que la fase de selección 104 transmita (sin cambios) los datos de audio. De manera adicional, opcional o alternativa, pueden usarse otros tipos de técnicas criptográficas en lugar de un método según un hash criptográfico.

El codificador 100 de la Figura 2 puede determinar (en respuesta a LPSM y, de manera opcional, también metadatos de límite de programa, extraídos por el descodificador 101) que una unidad de post/preprocesamiento ha llevado a cabo un tipo de procesamiento de sonoridad en los datos de audio que se codificarán (en los elementos 105, 106 y 107) y, por lo tanto, puede crear (en el generador 106) metadatos de estado de procesamiento de sonoridad que incluyen los parámetros específicos usados en y/o derivados del procesamiento de sonoridad previamente llevado a cabo. En algunas implementaciones, el codificador 100 puede crear (e incluir en el tren de bits codificado producido desde él) metadatos de estado de procesamiento indicativos del historial de procesamiento en el contenido de audio siempre y cuando el codificador conozca los tipos de procesamientos que se han llevado a cabo en el contenido de audio.

La Figura 3 es un diagrama de bloques de un descodificador (200) que es una realización de la unidad de procesamiento de audio inventiva, y de un postprocesador (300) acoplado a aquella. El postprocesador (300) también es una realización de la unidad de procesamiento de audio inventiva. Cualquiera de los componentes o elementos del descodificador 200 y postprocesador 300 pueden implementarse como uno o más procesos y/o uno o más circuitos (p. ej., ASIC, FPGA, u otros circuitos integrados), en hardware, software o una combinación de hardware y software. El descodificador 200 comprende búfer de trama 201, analizador 205, descodificador de audio 202, fase de validación de estado de audio (validador) 203 y fase de generación de bits de control 204, conectados como se muestra. También, normalmente, el descodificador 200 incluye otros elementos de procesamiento (no se muestran).

El búfer de trama 201 (una memoria intermedia) almacena (p. ej., de una manera no transitoria) al menos una trama del tren de bits de audio codificado recibido por el descodificador 200. Una secuencia de las tramas del tren de bits de audio codificado se confirma desde el búfer 201 al analizador 205.

El analizador 205 se acopla y configura para extraer metadatos de estado de procesamiento de sonoridad (LPSM) y, de manera opcional, también metadatos de límite de programa, y otros metadatos de cada trama del audio de entrada codificado, para confirmar al menos los LPSM (y metadatos de límite de programa, si se extraen) al validador de estado de audio 203 y fase 204, para confirmar los LPSM (y, de manera opcional, también los metadatos de límite de programa) como salida (p. ej., al postprocesador 300), para extraer datos de audio del audio de entrada codificado, y para confirmar los datos de audio extraídos al descodificador 202.

El tren de bits de audio codificado ingresado al descodificador 200 puede ser uno de un tren de bits AC-3, un tren de bits E-AC-3 o un tren de bits Dolby E.

El sistema de la Figura 3 también incluye el postprocesador 300. El postprocesador 300 comprende un búfer de trama 301 y otros elementos de procesamiento (no se muestran) incluido al menos un elemento de procesamiento acoplado al búfer 301. El búfer de trama 301 almacena (p. ej., de una manera no transitoria) al menos una trama del tren de bits de audio descodificado recibido por el postprocesador 300 desde el descodificador 200. Los elementos de procesamiento del postprocesador 300 se acoplan y configuran para recibir y procesar, de manera adaptativa, una secuencia de las tramas del tren de bits de audio descodificado producido desde el búfer 301, mediante el uso de metadatos (incluidos los valores LPSM) producidos desde el descodificador 202 y/o bits de control producidos desde la fase 204 del descodificador 200. Normalmente, el postprocesador 300 se configura para llevar a cabo el procesamiento de sonoridad adaptativo en los datos de audio descodificados mediante el uso de los valores LPSM y, de manera opcional, también metadatos de límite de programa (p. ej., según el estado de procesamiento de sonoridad y/o una o más características de datos de audio, indicados por LPSM para datos de audio indicativos de un solo programa de audio).

Varias implementaciones del descodificador 200 y postprocesador 300 se configuran para llevar a cabo diferentes realizaciones del método inventivo.

El descodificador de audio 202 del descodificador 200 se configura para descodificar los datos de audio extraídos por el analizador 205 para generar datos de audio descodificados y para confirmar los datos de audio descodificados como salida (p. ej., al postprocesador 300).

El validador de estado 203 se configura para autenticar y validar los LPSM (y, de manera opcional, otros metadatos) confirmados a aquel. En algunas realizaciones, los LPSM son (o se incluyen en) un bloque de datos que se ha incluido en el tren de bits de entrada (p. ej., según una realización de la presente invención). El bloque puede comprender un hash criptográfico (un código de autenticación de mensaje basado en hash o "HMAC") para procesar los LPSM (y, de manera opcional, también otros metadatos) y/o los datos de audio subyacentes (provistos del analizador 205 y/o descodificador 202 al validador 203). El bloque de datos puede firmarse de manera digital en estas realizaciones, de modo que una unidad de procesamiento de audio corriente abajo puede autenticar y validar, de forma relativamente fácil, los metadatos de estado de procesamiento.

Otros métodos criptográficos que incluyen, pero sin limitación, cualquiera de uno o más métodos criptográficos

diferentes de HMAC pueden usarse para la validación de LPSM (p. ej., en el validador 203) para garantizar la transmisión y recepción seguras de los LPSM y/o datos de audio subyacentes. Por ejemplo, puede llevarse a cabo validación (mediante el uso de dicho método criptográfico) en cada unidad de procesamiento de audio que recibe una realización del tren de bits de audio inventivo para determinar si los metadatos de estado de procesamiento de sonoridad y los datos de audio correspondientes incluidos en el tren de bits se han sometido al (y/o han resultado del) procesamiento de sonoridad específico (según se indica por los metadatos) y no se han modificado después de llevar a cabo el procesamiento de sonoridad específico.

El validador de estado 203 confirma datos de control al generador de bits de control 204 y/o confirma los datos de control como salida (p. ej., al postprocesador 300), para indicar los resultados de la operación de validación. En respuesta a los datos de control (y, de manera opcional, también otros metadatos extraídos del tren de bits de entrada), la fase 204 puede generar (y confirmar al postprocesador 300) ya sea:

bits de control que indican que los datos de audio descodificados producidos desde el descodificador 202 se han sometido a un tipo específico de procesamiento de sonoridad (cuando los LPSM indican que los datos de audio producidos desde el descodificador 202 se han sometido al tipo específico de procesamiento de sonoridad, y los bits de control del validador 203 indican que los LPSM son válidos); o

bits de control que indican que los datos de audio descodificados producidos desde el descodificador 202 deben someterse a un tipo específico de procesamiento de sonoridad (p. ej., cuando los LPSM indican que los datos de audio producidos desde el descodificador 202 no se han sometido al tipo específico de procesamiento de sonoridad, o cuando los LPSM indican que los datos de audio producidos desde el descodificador 202 se han sometido al tipo específico de procesamiento de sonoridad pero los bits de control del validador 203 indican que los LPSM no son válidos).

De manera alternativa, el descodificador 200 confirma los metadatos extraídos por el descodificador 202 desde el tren de bits de entrada, y los LPSM (y, de manera opcional, también metadatos de límite de programa) extraídos por el analizador 205 del tren de bits de entrada al postprocesador 300, y el postprocesador 300 lleva a cabo el procesamiento de sonoridad en los datos de audio descodificados mediante el uso de los LPSM (y, de manera opcional, también los metadatos de límite de programa), o lleva a cabo la validación de los LPSM y luego lleva a cabo el procesamiento de sonoridad en los datos de audio descodificados mediante el uso de los LPSM (y, de manera opcional, también metadatos de límite de programa) si la validación indica que los LPSM son válidos.

En algunas realizaciones, si el descodificador 200 recibe un tren de bits de audio generado según una realización de la invención con hash criptográfico, el descodificador se configura para analizar y recuperar el hash criptográfico de un bloque de datos determinado del tren de bits, dicho bloque comprendiendo metadatos de estado de procesamiento de sonoridad (LPSM). El validador 203 puede usar el hash criptográfico para validar el tren de bits recibido y/o metadatos asociados. Por ejemplo, si el validador 203 descubre que los LPSM son válidos basándose en una concordancia entre un hash criptográfico de referencia y el hash criptográfico recuperado del bloque de datos, entonces puede señalizar a una unidad de procesamiento de audio corriente abajo (p. ej., postprocesador 300, que puede ser o incluir una unidad de nivelación de volumen) que transmita (sin cambios) los datos de audio del tren de bits. De manera adicional, opcional o alternativa, pueden usarse otros tipos de técnicas criptográficas en lugar de un método según un hash criptográfico.

En algunas implementaciones del descodificador 200, el tren de bits codificado recibido (y almacenado en la memoria 201) es un tren de bits AC-3 o un tren de bits E-AC-3, y comprende segmentos de datos de audio (p. ej., los segmentos BA0-BA5 de la trama que se muestra en la Figura 4) y segmentos de metadatos, 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 metadatos de estado de procesamiento de sonoridad (LPSM) y, de manera opcional, también metadatos de límite de programa. La fase de descodificador 202 (y/o analizador 205) se configura para extraer del tren de bits LPSM (y, de manera opcional, también metadatos de límite de programa) que tienen el siguiente formato. Cada uno de los segmentos de metadatos que incluye LPSM (y, de manera opcional, también metadatos de límite de programa) se incluye en un segmento de bits de residuo de una trama del tren de bits, o un campo "addbsi" del segmento de Información de Tren de Bits ("BSI") de una trama del tren de bits, o en un campo auxdata (p. ej., el segmento AUX que se muestra en la Figura 4) al final de una trama del tren de bits. Una trama del tren de bits puede incluir uno o dos segmentos de metadatos, cada uno de los cuales puede incluir LPSM, y si la trama incluye dos segmentos de metadatos, uno puede estar presente en el campo addbsi de la trama y el otro puede estar presente en el campo AUX de la trama. En algunas realizaciones, cada segmento de metadatos, que incluye LPSM, incluye un segmento de carga útil LPSM (o contenedor) que tiene el siguiente formato:

un encabezamiento (que normalmente incluye una palabra de sincronización que identifica el inicio de la carga útil de LPSM, seguida de valores de identificación, p. ej., versión de formato de LPSM, longitud, período, cómputo y valores de asociación de subtren indicados en la Tabla 2 más abajo); y

después del encabezamiento,

al menos un valor de indicación de diálogo (p. ej., parámetro "Canal(es) de diálogo" de la Tabla 2) que indica si los

datos de audio correspondientes indican diálogo o no indican diálogo (p. ej., qué canales de los datos de audio correspondientes indican diálogo);

al menos un valor de cumplimiento de regulación de sonoridad (p. ej., el parámetro "Tipo de Regulación de Sonoridad" de la Tabla 2) que indica si los datos de audio correspondientes cumplen con un conjunto indicado de regulaciones de sonoridad;

al menos un valor de procesamiento de sonoridad (p. ej., uno o más de los parámetros "bandera de Corrección de Sonoridad con puerta de Diálogo", "Tipo de Corrección de Sonoridad", de la Tabla 2) que indica al menos un tipo de procesamiento de sonoridad que se ha llevado a cabo en los datos de audio correspondientes; y

al menos un valor de sonoridad (p. ej., uno o más de los parámetros "Sonoridad con Puerta Relativa a ITU", "Sonoridad con Puerta de Voz ITU", "Sonoridad 3s a Corto Plazo ITU (EBU 3341)", y "Pico Verdadero" de la Tabla 2) que indica al menos una característica de sonoridad (p. ej., sonoridad pico o promedio) de los datos de audio correspondientes.

En algunas realizaciones, cada segmento de metadatos que contiene LPSM y metadatos de límite de programa contiene un encabezamiento principal (y, de forma opcional, también elementos principales adicionales), y después del encabezamiento principal (o el encabezamiento principal y otros elementos principales) un segmento de carga útil de LPSM (o contenedor) que tiene el siguiente formato:

un encabezamiento, que normalmente incluye al menos un valor de identificación (p. ej., versión de formato de LPSM, longitud, período, cómputo y valores de asociación de subtren, como se indica en la Tabla 2 más abajo), y

después del encabezamiento, los LPSM y metadatos de límite de programa. Los metadatos de límite de programa pueden incluir un cómputo de tramas de límite de programa y un valor de código (p. ej., un valor "desplazamiento_existe") indicativo de si la trama incluye solamente un cómputo de tramas de límite de programa o tanto un cómputo de tramas de límite de programa como un valor de desplazamiento), y (en algunos casos) un valor de desplazamiento.

En algunas implementaciones, el analizador 205 (y/o fase de descodificador 202) se configura para extraer, de un segmento de bits de residuo, o un campo "addbsi", o un campo auxdata, de una trama del tren de bits, cada segmento de metadatos que tiene el siguiente formato:

un encabezamiento principal (que, normalmente, incluye una palabra de sincronización que identifica el inicio del segmento de metadatos, seguida de al menos un valor de identificación, p. ej., la versión de elemento Principal, longitud y período, cómputo de elementos extendidos, y valores de asociación de subtren indicados en la Tabla 1 más abajo); y

después del encabezamiento principal, al menos un valor de protección (p. ej., el resumen de HMAC y valores de Huella Digital de Audio de la Tabla 1) útil para al menos una de desencriptación, autenticación o validación de al menos uno de los metadatos de estado de procesamiento de sonoridad o los datos de audio correspondientes); y

también después del encabezamiento principal, si el segmento de metadatos incluye LPSM, la identificación ("ID") de carga útil de LPSM y valores de tamaño de carga útil de LPSM que identifican los siguientes metadatos como una carga útil de LPSM e indican el tamaño de la carga útil de LPSM.

El segmento de carga útil de LPSM (o contenedor) (que, preferiblemente, tiene el formato especificado más arriba) sigue a la ID de carga útil de LPSM y valores de tamaño de carga útil LPSM.

De manera más general, el tren de bits de audio codificado generado por realizaciones preferidas de la invención tiene una estructura que proporciona un mecanismo para etiquetar elementos de metadatos y subelementos como principales (obligatorios) o expandidos (elementos opcionales). Ello permite a la velocidad de datos del tren de bits (incluidos sus metadatos) escalar a lo largo de numerosas aplicaciones. Los elementos principales (obligatorios) de la sintaxis del tren de bits preferido deben también poder señalizar que los elementos expandidos (opcionales) asociados al contenido de audio están presentes (en banda) y/o en una ubicación remota (fuera de banda).

Se requiere que los elementos principales estén presentes en cada trama del tren de bits. Algunos subelementos de los elementos principales son opcionales y pueden estar presentes en cualquier combinación. No se requiere que los elementos expandidos estén presentes en cada trama (para limitar la sobrecarga de velocidad de bits). Por consiguiente, los elementos expandidos pueden estar presentes en algunas tramas y no en otras. Algunos subelementos de un elemento expandido son opcionales y pueden estar presentes en cualquier combinación, mientras que algunos subelementos de un elemento expandido pueden ser obligatorios (a saber, si el elemento expandido está presente en una trama del tren de bits).

En una clase de realizaciones, se genera un tren de bits de audio codificado que comprende una secuencia de segmentos de datos de audio y segmentos de metadatos (p. ej., por una unidad de procesamiento de audio que realiza la invención). Los segmentos de datos de audio son indicativos de datos de audio, cada uno de al menos algunos de los segmentos de metadatos incluye metadatos de estado de procesamiento de sonoridad (LPSM) y, de manera

opcional, también metadatos de límite de programa, y los segmentos de datos de audio se multiplexan por división del tiempo con los segmentos de metadatos. En realizaciones preferidas en esta clase, cada uno de los segmentos de metadatos tiene un formato preferido que se describirá en la presente memoria.

En un formato preferido, el tren de bits codificado es un tren de bits AC-3 o un tren de bits E-AC-3, y cada uno de los segmentos de metadatos que incluye LPSM se incluye (p. ej., por la fase 107 de una implementación preferida del codificador 100) como información adicional del tren de bits en el campo "addbsi" (que se muestra en la Figura 6) del segmento de Información de Tren de Bits ("BSI") de una trama del tren de bits, o en un campo auxdata de una trama del tren de bits, o en un segmento de bits de residuo de una trama del tren de bits.

En el formato preferido, cada una de las tramas incluye un elemento principal que tiene el formato que se muestra en la Tabla 1 más abajo, en el campo addbsi (o segmento de bits de residuo) de la trama:

Tabla 1

Parámetro	Descripción	Obligatorio/Opcional
SYNC [ID]	La palabra de sincronización puede ser un valor de 16 bits establecido en el valor de 0x5838	Ob.
Versión de elemento principal		Ob.
Longitud del elemento principal		Ob.
Período del elemento principal (xxx)		Ob.
Cómputo de elementos extendidos	Indica el número de elementos de metadatos extendidos asociados al elemento principal. Este valor puede aumentar/disminuir mientras el tren de bits se transmite de la producción a la distribución y emisión final.	Ob.
Asociación de subtren	Describe a qué subtren(es) se asocia el elemento principal.	Ob.
Firma (resumen de HMAC)	resumen de HMAC de 256 bits (mediante el uso del algoritmo SHA-2) computado en los datos de audio, el elemento principal, y todos los elementos expandidos, de toda la trama.	Ob.
Cuenta regresiva de límite PGM	Este campo solo aparece para cierto número de tramas al principio o final de un tren/archivo de programa de audio. Por consiguiente, puede usarse un cambio de versión de elemento principal para señalizar la inclusión de este parámetro.	Op.
Huella Digital de Audio	Huella Digital de Audio tomada en cierto número de muestras de audio PCM representadas por el campo de período de elemento principal.	Op.
Huella Digital de Vídeo	Huella Digital de Vídeo tomada en cierto número de muestras de vídeo comprimidas (si las hubiera) representadas por el campo de período de elemento principal.	Op.

URL/UUID	Este campo se define para transportar un URL y/o un UUID (puede ser redundante a la huella digital) que hace referencia a una ubicación externa del contenido de programa adicional (esencia) y/o metadatos asociados al tren de bits.	Op.

En el formato preferido, cada uno de los campos addbsi (o auxdata) o segmentos de bits de residuo que contienen LPSM contiene un encabezamiento principal (y, de manera opcional, también elementos principales adicionales) y después del encabezamiento principal (o el encabezamiento principal y otros elementos principales), los siguientes valores LPSM (parámetros):

una ID de carga útil (que identifica los metadatos como LPSM) que sigue a los valores de elemento principal (p. ej., según se especifica en la Tabla 1);

un tamaño de carga útil (que indica el tamaño de la carga útil de LPSM) que sigue a la ID de carga útil; y

datos LPSM (que siguen a la ID de carga útil y valor de tamaño de carga útil) que tienen el formato según se indica en la siguiente tabla (Tabla 2):

Tabla 2

Parámetro LPSM [Sonoridad Inteligente]	Descripción	número de estados únicos	Obligatorio/Opcional	Velocidad de Inserción (Período de actualización del parámetro)
Versión LPSM			Ob.	
Período LPSM (xxx)	Aplicable a xxx campos solamente		Ob.	
Cómputo de LPSM			Ob.	
Asociación de subtren de LPSM			Ob.	
Canal(es) de diálogo	Indica qué combinación de canales de audio L, C y R	8	Ob.	~0,5 segundos
	contienen voz en los 0,5 segundos previos. Cuando la voz no está presente en ninguna combinación de L, C o R, entonces este parámetro debe indicar "sin diálogo"			(típico)
Tipo de Regulación de Sonoridad	Indica que el tren de datos de audio asociado cumple con un conjunto específico de regulaciones (p. ej., ATSC A/85 o EBU R128)	8	Ob.	Trama

Bandera de Corrección de Sonoridad con puerta de Diálogo	Indica si el tren de audio asociado se ha corregido según la puerta de diálogo	2	Op. (solo presente si Tipo_Regulación_Sonoridad indica que el audio correspondiente NO SE HA CORREGIDO)	Trama
Tipo de Corrección de Sonoridad	Indica si el tren de audio asociado se ha corregido con una previsión de compatibilidad infinita (basada en archivos) o con un controlador de rango dinámico y sonoridad en tiempo real (TR)	2	Op. (solo presente si Tipo_Regulación_Sonoridad indica que el audio correspondiente NO SE HA CORREGIDO)	Trama
Sonoridad con Puerta Relativa a ITU (INF)	Indica la sonoridad integrada ITU-R BS.1770-3 del tren de audio asociado sin metadatos aplicados (p. ej., 7 bits: -58 -> +5.5 LKFS etapas 0.5 LKFS)	128	Op.	1 s
Sonoridad con Puerta de Voz ITU (INF)	Indica la sonoridad integrada ITU-R BS.1770-1/3 de voz/diálogo del tren de audio asociado sin metadatos aplicados (p. ej., 7 bits: -58 -> +5.5 LKFS etapas 0.5 LKFS)	128	Op.	1 s
Sonoridad 3s a Corto Plazo ITU (EBU 3341)	Indica la sonoridad ITU sin puerta de 3 segundos (ITU- BS.1771-1) del tren de audio asociado sin metadatos aplicados (ventana deslizante) @ velocidad de inserción ~ 10 Hz (p. ej., 8 bits: 116 -> +11.5 LKFS etapas 0.5 LKFS)	256	Op.	0,1 s
Valor Pico Verdadero	Indica el valor de PicoVerdadero del Anexo 2 ITU-R BS.1770- 3 (dB TP) del tren de audio asociado sin metadatos aplicados (a saber, valor más grande en período de trama señalizado en el campo de período de elemento) 116 -> +11.5 LKFS etapas 0.5 LKFS	256	Op.	0,5 s
Desplazamiento de Mezcla Descendente	Indica el desplazamiento de sonoridad de mezcla descendente			
Límite de Programa	Indica, en tramas, cuándo ocurrirá o ha ocurrido un límite de programa. Cuando el límite de programa no se encuentra en un límite de trama, el desplazamiento de muestra opcional indicará cuán lejos ocurre el límite de programa real en la trama			

En otro formato preferido de un tren de bits codificado generado según la invención, el tren de bits es un tren de bits AC-3 o un tren de bits E-AC-3, y cada uno de los segmentos de metadatos que incluye LPSM (y, de manera opcional, también metadatos de límite de programa) se incluye (p. ej., por la fase 107 de una implementación preferida del

codificador 100) en cualquiera de: un segmento de bits de residuo de una trama del tren de bits; o un campo "addbsi" (que se muestra en la Figura 6) del segmento de Información de Tren de Bits ("BSI") de una trama del tren de bits; o un campo auxdata (p. ej., el segmento AUX que se muestra en la Figura 4) al final de una trama del tren de bits. Una trama puede incluir uno o dos segmentos de metadatos, cada uno de los cuales incluye LPSM, 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. Cada segmento de metadatos que incluye LPSM tiene el formato especificado más arriba con referencia a las Tablas 1 y 2 más arriba (a saber, incluye los elementos principales especificados en la Tabla 1, seguidos de la ID de carga útil (que identifica los metadatos como LPSM) y valores de tamaño de carga útil especificados más arriba, seguidos por la carga útil (los datos de LPSM que tienen el formato indicado en la Tabla 2).

En otro formato preferido, el tren de bits codificado es un tren de bits Dolby E y cada uno de los segmentos de metadatos que incluye LPSM (y, de forma opcional, también metadatos de límite de programa) está en las primeras N ubicaciones de muestras del intervalo de banda de guarda de Dolby E. Un tren de bits Dolby E que incluye dicho segmento de metadatos que incluye LPSM preferiblemente incluye un valor indicativo de longitud de carga útil de LPSM señalizado en la palabra Pd del preámbulo SMPTE 337M (la velocidad de repetición de palabra Pa SMPTE 337M preferiblemente permanece idéntica a la velocidad de trama de vídeo asociada).

En un formato preferido, en el que el tren de bits codificado es un tren de bits E-AC-3, cada uno de los segmentos de metadatos incluye LPSM (y, de manera opcional, también metadatos de límite de programa) se incluye (p. ej., por la fase 107 de una implementación preferida del codificador 100) como información de tren de bits adicional en un segmento de bits de residuo, o en el campo "addbsi" del segmento de Información de Tren de Bits ("BSI"), de una trama del tren de bits. A continuación se describen aspectos adicionales de la codificación de un tren de bits E-AC-3 con LPSM en el presente formato preferido:

1. durante la generación de un tren de bits E-AC-3, mientras el codificador E-AC-3 (que inserta los valores LPSM en el tren de bits) estça "activo", para cada trama (syncframe) generada, el tren de bits debe incluir un bloque de metadatos (incluidos los LPSM) transportado en el campo addbsi (o segmento de bits de residuo) de la trama. Los bits requeridos para transportar el bloque de metadatos no deben aumentar la velocidad de bits del codificador (longitud de trama);

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

sonoridad_corrección_tipo_bandera: donde '1' indica que la sonoridad de los datos de audio correspondientes se ha corregido corriente arriba desde el codificador, y '0' indica que la sonoridad se ha corregido por un corrector de sonoridad incorporado en el codificador (p. ej., procesador de sonoridad 103 del codificador 100 de la Figura 2);

canal_voz: indica qué canal(es) de origen contiene(n) voz (en los 0,5 segundos previos). Si no se detecta voz alguna, debe indicarse como tal;

sonoridad_voz: indica la sonoridad de voz integrada de cada canal de audio correspondiente que contiene voz (en los 0,5 segundos previos);

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

ganancia: ganancia compuesta de sonoridad para alternancias en un descodificador (para demostrar la reversibilidad);

3. mientras el codificador E-AC-3 (que inserta los valores LPSM en el tren de bits) está "activo" y recibe una trama AC-3 con una bandera de "fiabilidad", el controlador de sonoridad en el codificador (p. ej., el procesador de sonoridad 103 del codificador 100 de la Figura 2) debe ser baipaseado. Los valores DRC y dialnorm de origen "fiables" deben transmitirse (p. ej., por el generador 106 del codificador 100) al componente de codificador E-AC-3 (p. ej., fase 107 del codificador 100). La generación del bloque LPSM continúa y bandera_tipo_corrección_sonoridad se establece en '1'. La secuencia de baipás de controlador de sonoridad debe sincronizarse con el inicio de la trama AC-3 descodificada donde aparece la bandera "fiabilidad". La secuencia de baipás de controlador de sonoridad debe implementarse de la siguiente manera: el control cantidad_nivelador se disminuye desde un valor de 9 a un valor de 0 en 10 períodos de bloques de audio (a saber, 53,3 ms) y el control medidor_extremo_posterior_nivelador se coloca en modo baipás (dicha operación debe resultar en una transición sin discontinuidades). El término baipás "fiable" del nivelador implica que el valor dialnorm del tren de bits de origen también se reutiliza en la salida del codificador (p. ej., si el tren de bits de origen "fiable" tiene un valor dialnorm de -30 entonces la salida del codificador debe utilizar -30 para el valor dialnorm de salida);

4. Mientras el codificador E-AC-3 (que inserta los valores LPSM en el tren de bits) está "activo" y recibe una trama AC- 3 sin la bandera de "fiabilidad", el controlador de sonoridad incorporado al codificador (p. ej., procesador de sonoridad 103 del codificador 100 de la Figura 2) debe estar activo. La generación de bloque LPSM continúa y sonoridad_corrección_tipo_bandera se establece en '0'. La secuencia de activación de controlador de sonoridad debe sincronizarse con el inicio de la trama AC-3 descodificada donde desaparece la bandera "fiabilidad". La secuencia de activación de controlador de sonoridad debe implementarse de la siguiente manera: el control cantidad_nivelador se aumenta desde un valor de 0 a un valor de 9 en 1 período de bloques de audio (a saber, 5,3 ms) y el control medidor_extremo_posterior_nivelador se coloca en modo "activo" (dicha operación debe resultar en una transición sin

discontinuidades e incluir un restablecimiento de integración medidor_extremo_posterior); y

5. durante la codificación, una interfaz gráfica de usuario (GUI, por sus siglas en inglés) 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 de la bandera "fiabilidad" dentro de la señal de entrada; y "Corrección de Sonoridad en Tiempo Real: [Habilitado/Inhabilitado]" -el estado de este parámetro se basa en si este controlador de sonoridad incorporado al codificador se encuentra activo.

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

En otro formato preferido de un tren de bits codificado generado según la invención, el tren de bits codificado es un tren de bits AC-3 o un tren de bits E-AC-3, y cada uno de los segmentos de metadatos que incluye LPSM se incluye (p. ej., por la fase 107 de una implementación preferida del codificador 100) en un segmento de bits de residuo, o en un segmento Aux, o como información de tren de bits adicional en el campo "addbsi" (que se muestra en la Figura 6) del segmento de Información de Tren de Bits ("BSI"), de una trama del tren de bits. En este formato (que es una variación del formato descrito más arriba con referencia a las Tablas 1 y 2), cada uno de los campos addbsi (o Aux o de bits de residuo) que contiene LPSM contiene los siguientes valores LPSM:

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

versión de 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 se describe a continuación: bit 0, que indica la presencia de diálogo en el canal izquierdo, se almacena en el bit más significativo del campo dialchan; y bit 2, que indica la presencia de diálogo en el canal central, se almacena en el bit menos significativo del campo dialchan.

Cada bit del campo dialchan se establece en "1" si el canal correspondiente contiene diálogo hablado durante los 0,5 segundos anteriores del programa;

loudregtyp: un campo de 4 bits que indica qué estándar de regulación de sonoridad cumple la sonoridad de programa. Establecer el campo "loudregtyp" en "000" indica que los LPSM no indican cumplimiento de regulación de sonoridad. Por ejemplo, un valor de este campo (p. ej., 0000) puede indicar que el cumplimiento de un estándar de regulación de sonoridad no se indica, otro valor de este campo (p. ej., 0001) puede indicar que los datos de audio del programa cumplen el estándar ATSC A/85, y otro valor de este campo (p. ej., 0010) puede indicar que los datos de audio del programa cumplen el estándar EBU R128. En el ejemplo, si el campo se establece en cualquier valor diferente de "0000", los campos loudcorrdialgat y loudcorrtyp deben seguir a la carga útil;

loudcorrdialgat: un campo de un bit que indica si se ha aplicado la corrección de sonoridad con puerta de diálogo. Si la sonoridad del programa se ha corregido mediante el uso de puertas de diálogo, el valor del campo loudcorrdialgat se establece en "1". De lo contrario, se establece en "0";

loudcorrtyp: un campo de un bit que indica el tipo de corrección de sonoridad aplicada al programa. Si la sonoridad del programa se ha corregido con un proceso de corrección de sonoridad de previsión de compatibilidad infinita (basada en archivos), el valor del campo loudcorrtyp se establece en "0". Si la sonoridad del programa se ha corregido mediante el uso de una combinación de medición de sonoridad en tiempo real y control de rango dinámico, el valor de este campo se establece en "1";

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

loudrelgat: un campo de 7 bits que indica la sonoridad de programa con puerta relativa (ITU). Este campo indica la sonoridad integrada del programa de audio, medida según ITU-R BS.1770-3 sin ajustes de ganancia debido a la compresión de dialnorm y rango dinámico aplicada. Los valores de 0 a 127 se interpretan como -58 LKFS a +5.5 LKFS, en las etapas 0.5 LKFS;

loudspchgate: un campo de un bit que indica si existen datos de sonoridad con puerta de voz (ITU). Si el campo loudspchgate se establece en "1", un campo loudspchgat de 7 bits debe seguir a la carga útil;

loudspchgat: un campo de 7 bits que indica la sonoridad de programa con puerta de voz. Este campo indica la sonoridad integrada de todo el programa de audio correspondiente, medida según la fórmula (2) de ITU-R BS.1770- 3

y sin ajustes de ganancia debido a la compresión de dialnorm y rango dinámico aplicada. Los valores de 0 a 127 se interpretan como - 58 a +5.5 LKFS, en las etapas 0.5 LKFS;

loudstrm3se: un campo de un bit que indica si existen datos de sonoridad a corto plazo (3 segundos). Si el campo se establece en "1", un campo loudstrm3s de 7 bits debe seguir a la carga útil;

loudstrm3s: un campo de 7 bits que indica la sonoridad sin puerta de los 3 segundos anteriores del programa de audio correspondiente, medida según ITU-R BS.1771-1 y sin ajustes de ganancia debido a la compresión de dialnorm y rango dinámico aplicada. Los valores de 0 a 256 se interpretan como -116 LKFS a +11.5 LKFS en las etapas 0.5 LKFS;

truepke: un campo de un bit que indica si existen datos de sonoridad de pico verdadero. Si el campo truepke se establece en "1", un campo truepk de 8 bits debe seguir a la carga útil; y

truepk: un campo de 8 bits que indica el valor de muestra de pico verdadero del programa, medido según el Anexo 2 de ITU-R BS.1770-3 y sin ajustes de ganancia debido a la compresión aplicada de dialnorm y rango dinámico. Los valores de 0 a 256 se interpretan como -116 LKFS a +11.5 LKFS en las etapas 0.5 LKFS.

En algunas realizaciones, el elemento principal de un segmento de metadatos en un segmento de bits de residuo o en un campo auxdata (o "addbsi") de una trama de un tren de bits AC-3 o un tren de bits E-AC-3 comprende un encabezamiento principal (que, normalmente, incluye valores de identificación, p. ej., versión de elemento principal) y después del encabezamiento principal: valores indicativos de si los datos de huella digital (u otros valores de protección) se incluyen para metadatos del segmento de metadatos, valores indicativos de si existen datos externos (relacionados con datos de audio correspondientes a los metadatos del segmento de metadatos), ID de carga útil y valores de tamaño de carga útil para cada tipo de metadatos (p. ej., LPSM y/o metadatos de un tipo diferente de LPSM) identificados por el elemento principal, y valores de protección para al menos un tipo de metadatos identificado por el elemento principal. Las cargas útiles de metadatos del segmento de metadatos siguen al encabezamiento principal, y se anidan (en algunos casos) dentro de valores del elemento principal.

Las realizaciones típicas de la invención incluyen metadatos de límite de programa en un tren de bits de audio codificado de manera eficaz, lo que permite la determinación exacta y robusta de al menos un límite entre programas de audio consecutivos indicados por el tren de bits. Las realizaciones típicas permiten la determinación exacta y robusta de un límite de programa en el sentido de que permiten la determinación exacta de límite de programa incluso en casos en los que trenes de bits indicativos de diferentes programas se empalman juntos (para generar el tren de bits inventivo) de una manera que trunca uno o ambos trenes de bits empalmados (y, por consiguiente, descarta metadatos de límite de programa que se habían incluido en al menos uno de los trenes de bits de preempalme).

En las realizaciones típicas, los metadatos de límite de programa en una trama del tren de bits inventivo son una bandera de límite de programa indicativa de un cómputo de tramas. Normalmente, la bandera es indicativa del número de tramas entre la trama actual (la trama que incluye la bandera) y un límite de programa (el comienzo o fin del programa de audio actual). En algunas realizaciones preferidas, las banderas de límite de programa se insertan de una manera simétrica y eficaz al comienzo y fin de cada segmento de tren de bits que es indicativo de un solo programa (a saber, en tramas que ocurren dentro de cierto número predeterminado de tramas después del comienzo del segmento, y en tramas que ocurren dentro de cierto número predeterminado de tramas antes del fin del segmento), de modo que cuando dos de dichos segmentos de tren de bits se concatenan (para ser indicativos de una secuencia de dos programas), los metadatos de límite de programa pueden estar presentes (p. ej., de forma simétrica) a ambos lados del límite entre los dos programas.

La robustez máxima puede lograrse insertando una bandera de límite de programa en cada trama de un tren de bits indicativo de un programa, pero ello normalmente no será práctico debido al aumento asociado de la velocidad de datos. En realizaciones típicas, las banderas de límite de programa se insertan solamente en un subconjunto de las tramas de un tren de bits de audio codificado (el que puede ser indicativo de un programa de audio o una secuencia de programas de audio), y la velocidad de inserción de bandera de límite es una función no creciente de la separación creciente de cada una de las tramas del tren de bits (en las que se inserta una bandera) del límite de programa que está más cerca de cada una de las tramas, donde "velocidad de inserción de bandera de límite" denota la relación promedio del número de tramas (indicativas de un programa) que incluyen una bandera de límite de programa y el número de tramas (indicativas del programa) que no incluyen una bandera de límite de programa, donde el promedio es un promedio móvil de un número (p. ej., un número relativamente pequeño) de tramas consecutivas del tren de bits de audio codificado.

Aumentar la velocidad de inserción de bandera de límite (p. ej., en ubicaciones en el tren de bits más cercanas a un límite de programa) aumenta la velocidad de datos requerida para la entrega del tren de bits. Para compensar esto, el tamaño (número de bits) de cada bandera insertada se reduce, preferiblemente, conforme se aumenta la velocidad de inserción de bandera de límite (p. ej., de modo que el tamaño de la bandera de límite de programa en la "N"ésima trama del tren de bits, donde N es un entero, es una función no creciente de la distancia (número de tramas) entre la "N"ésima trama y el límite de programa más cercano). En una clase de realizaciones, la velocidad de inserción de bandera de límite es una función logarítmicamente decreciente de distancia creciente (de cada ubicación de inserción

de bandera) desde el límite de programa más cercano, y para cada trama que contiene bandera que incluye una de las banderas, el tamaño de la bandera en dicha trama que contiene bandera es igual a o mayor que el tamaño de cada bandera en una trama ubicada más cerca del límite de programa más cercano que lo que se encuentra dicha trama que contiene bandera. Normalmente, el tamaño de cada bandera se determina por una función creciente del número de tramas desde la ubicación de inserción de la bandera al límite de programa más cercano.

Por ejemplo, consideremos la realización de las Figuras 8 y 9, en las que cada columna identificada por un número de trama (en la fila superior) indica una trama de un tren de bits de audio codificado. El tren de bits es indicativo de un programa de audio que tiene un primer límite de programa (indicativo del comienzo del programa) que ocurre inmediatamente a la izquierda de la columna identificada por el número de trama "17" en el lado izquierdo de la Figura 9, y un segundo límite de programa (indicativo del fin del programa) que ocurre inmediatamente a la derecha de la columna identificada por el número de trama "1" en el lado derecho de la Figura 8. Las banderas de límite de programa incluidas en las tramas que se muestran en la Figura 8 cuentan, de manera regresiva, el número de tramas entre la trama actual y el segundo límite de programa. Las banderas de límite de programa incluidas en las tramas que se muestran en la Figura 9 cuentan, de manera ascendente, el número de tramas entre la trama actual y el primer límite de programa.

En la realización de las Figuras 8 y 9, se inserta una bandera de límite de programa solamente en cada una de las ""2^N"ésimas tramas de las primeras X tramas del tren de bits codificado después del inicio del programa de audio indicado por el tren de bits, y en cada una de las "2^N"ésimas tramas (de las últimas X tramas del tren de bits) más cercanas al fin del programa indicado por el tren de bits, donde el programa comprende Y tramas, X es un entero menor o igual a Y/2 y N es un entero positivo en un rango de 1 a $\log_2(X)$. Por consiguiente (según se indica en las Figuras 8 y 9), una bandera de límite de programa se inserta en la segunda trama (N = 1) del tren de bits (la trama que contiene bandera más cercana al comienzo del programa), en la cuarta trama (N = 2), en la octava trama (N = 3) y así sucesivamente, y en la octava trama del final del tren de bits, en la cuarta trama del final del tren de bits, y en la segunda trama del final del tren de bits (la trama que contiene bandera más cercana al fin del programa). En el presente ejemplo, la bandera de límite de programa en la "2^N"ésima trama desde el comienzo (o fin) del programa comprende $\log_2(2^{N+2})$ bits binarios, como se indica en las Figuras 8 y 9. Por consiguiente, la bandera de límite de programa en la segunda trama (N = 1) desde el comienzo (o fin) del programa comprende $\log_2(2^{N+2}) = \log_2(2^3) = 3$ bits binarios, y la bandera en la cuarta trama (N = 2) desde el comienzo (o fin) del programa comprende $\log_2(2^{N+2}) = \log_2(2^4) = 4$ bits binarios, y así sucesivamente.

En el ejemplo de las Figuras 8 y 9, el formato de cada bandera de límite de programa es como se describe a continuación. Cada bandera de límite de programa consiste en un bit "1" líder, una secuencia de "0" bits (ya sea ningún "0" bit o uno o más "0" bits consecutivos) después del bit de cabeza, y un código de cola de dos bits. El código de cola es "11" para las banderas en las últimas X tramas del tren de bits (las tramas más cercanas al fin del programa), como se indica en la Figura 8. El código de cola es "10" para las banderas en las primeras X tramas del tren de bits (las tramas más cercanas al comienzo del programa), como se indica en la Figura 9. Por consiguiente, para leer (descodificar) cada bandera, se cuenta el número de ceros entre el bit de cabeza "1" y el código de cola. Si se identifica que el código de cola es "11", la bandera indica que hay (2^{Z+1} - 1) tramas entre la trama actual (la trama que incluye la bandera) y el fin del programa, donde Z es el número de ceros entre el bit de cabeza "1" de la bandera y el código de cola. El descodificador puede implementarse, de manera eficaz, para ignorar el primer y último bit de cada bandera, para determinar el inverso de la secuencia de otros bits (intermedios) de la bandera (p. ej., si la secuencia de bits intermedios es "0001" con el bit "1" siendo el último bit en la secuencia, la secuencia invertida de bits intermedios es "1000" con el bit "1" siendo el primer bit en la secuencia invertida), y para identificar el valor binario de la secuencia invertida de bits intermedios como el índice de la trama actual (la trama en la que se incluye la bandera) respecto al fin del programa. Por ejemplo, si la secuencia invertida de bits intermedios es "1000", dicha secuencia invertida tiene el valor binario $2^4 = 16$, y la trama se identifica como la 16-ésima trama antes del fin del programa (como se indica en la columna de la Figura 8 que describe la trama "0").

Si se identifica que el código de cola es "10", la bandera indica que hay (2^{Z+1} - 1) tramas entre el inicio del programa y la trama actual (la trama que incluye la bandera), donde Z es el número de ceros entre el bit de cabeza "1" de la bandera y el código de cola. El descodificador puede implementarse, de manera eficaz, para ignorar el primer y último bit de cada bandera, para determinar el inverso de la secuencia de los bits intermedios de la bandera (p. ej., si la secuencia de bits intermedios es "0001" con el bit "1" siendo el último bit en la secuencia, la secuencia invertida de bits intermedios es "1000" con el bit "1" siendo el primer bit en la secuencia invertida), y para identificar el valor binario de la secuencia invertida de bits intermedios como el índice de la trama actual (la trama en la que se incluye la bandera) respecto al comienzo del programa. Por ejemplo, si la secuencia invertida de bits intermedios es "1000", dicha secuencia invertida tiene el valor binario $2^4 = 16$, y la trama se identifica como la 16-ésima trama después del comienzo del programa (como se indica en la columna de la Figura 9 que describe la trama "32").

En el ejemplo de las Figuras 8 y 9, una bandera de límite de programa está presente solamente en cada una de las ""2^N"ésimas tramas de las primeras X tramas de un tren de bits codificado después del inicio de un programa de audio indicado por el tren de bits, y en cada una de las "2^N"ésimas tramas (de las últimas X tramas del tren de bits) más cercanas al fin del programa indicado por el tren de bits, donde el programa comprende Y tramas, X es un entero menor que o igual a Y/2 y N es un entero positivo en un rango de 1 a $\log_2(X)$. La inclusión de las banderas de límite

de programa solo añade una velocidad binaria promedio de 1.875 bits/trama a la velocidad binaria requerida para transmitir el tren de bits sin las banderas.

En una implementación típica de la realización de las Figuras 8 y 9 en la que el tren de bits es un tren de bits de audio codificado AC-3, cada trama contiene contenido de audio y metadatos para 1536 muestras de audio digital. Para una velocidad de muestreo de 48 kHz, ello representa 32 milisegundos de audio digital o una velocidad de 31.25 tramas por segundo de audio. Por consiguiente, en dicha realización, una bandera de límite de programa en una trama separada de cierto número de tramas ("X" tramas) desde un límite de programa indica que el límite ocurre 32X milisegundos después del final de la trama que contiene bandera (o 32X milisegundos antes del inicio de la trama que contiene bandera).

En una implementación típica de la realización de las Figuras 8 y 9 en la que el tren de bits es un tren de bits de audio codificado E-AC-3, cada trama del tren de bits contiene 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 velocidad de muestreo de 48 kHz, ello representa 5.333, 10.667, 16 o 32 milisegundos de audio digital respectivamente o una velocidad de 189.9, 93.75, 62.5 o 31.25 tramas por segundo de audio, respectivamente. Por consiguiente, en dicha realización (suponiendo que cada trama es indicativa de 32 milisegundos de audio digital), una bandera de límite de programa en una trama separada por cierto número de tramas ("X" tramas) de un límite de programa indica que el límite ocurre 32X milisegundos después del final de la trama que contiene bandera (o 32X milisegundos antes del inicio de la trama que contiene la bandera).

En algunas realizaciones en las que un límite de programa puede ocurrir dentro de una trama de un tren de bits de audio (a saber, no en alineación con el comienzo o final de una trama), los metadatos de límite de programa incluidos en una trama del tren de bits incluyen un cómputo de tramas de límite de programa (a saber, metadatos indicativos del número de tramas completas entre el comienzo o final de la trama que contiene el cómputo de tramas y un límite de programa) y un valor de desplazamiento. El valor de desplazamiento es indicativo de un desplazamiento (normalmente, un número de muestras) entre el comienzo o fin de una trama que contiene límite de programa y la ubicación real del límite de programa dentro de la trama que contiene el límite de programa.

Un tren de bits de audio codificado puede ser indicativo de una secuencia de programas (pistas de audio) de una secuencia correspondiente de programas de vídeo, y los límites de dichos programas de audio tienden a ocurrir en los bordes de las tramas de vídeo en lugar de en los bordes de las tramas de audio. Asimismo, algunos códecs de audio (p. ej., códecs E-AC-3) usan tamaños de trama de audio que no se alinean con las tramas de vídeo. Asimismo, en algunos casos un tren de bits de audio inicialmente codificado experimenta la transcodificación para generar un tren de bits transcodificado, y el tren de bits inicialmente codificado tiene un tamaño de trama diferente que el tren de bits transcodificado de modo que no se garantiza que un límite de programa (determinado por el tren de bits inicialmente codificado) ocurra en un límite de trama del tren de bits transcodificado. Por ejemplo, si el tren de bits inicialmente codificado (p. ej., el tren de bits "IEB" de la Figura 10) tiene un tamaño de trama de 1536 muestras por trama, y el tren de bits transcodificado (p. ej., tren de bits "TB" de la Figura 10) tiene un tamaño de trama de 1024 muestras por trama, el proceso de transcodificación puede hacer que el límite de programa real ocurra no en un límite de trama del tren de bits transcodificado sino en un lugar en una trama de aquel (p. ej., 512 muestras en una trama del tren de bits transcodificado, como se indica en la Figura 10), debido a diferentes tamaños de trama de los diferentes códecs. Las realizaciones de la presente invención en las que los metadatos de límite de programa incluidos en una trama de un tren de bits de audio codificado incluyen un valor de desplazamiento así como un cómputo de tramas de límite de programa son útiles en los tres casos descritos en el presente párrafo (así como en otros casos).

La realización descrita más arriba con referencia a las Figuras 8 y 9 no incluye un valor de desplazamiento (p. ej., un campo de desplazamiento) en las tramas del tren de bits codificado. En variaciones de la presente realización, un valor de desplazamiento se incluye en cada trama de un tren de bits de audio codificado que incluye una bandera de límite de programa (p. ej., en tramas correspondientes a las tramas numeradas 0, 8, 12 y 14 en la Figura 8, y las tramas numeradas 18, 20, 24 y 32 en la Figura 9).

En una clase de realizaciones, una estructura de datos (en cada trama de un tren de bits codificado que contiene los metadatos de límite de programa inventivos) incluye un valor de código indicativo de si la trama incluye solamente un cómputo de tramas de límite de programa o tanto un cómputo de tramas de límite de programa como un valor de desplazamiento. Por ejemplo, el valor de código puede ser el valor de un campo de un solo bit (al que se hará referencia en la presente memoria como un campo "desplazamiento_existe"), el valor "desplazamiento_existe" = 0 puede indicar que no se incluye valor de desplazamiento alguno en la trama, y el valor "desplazamiento_existe" = 1 puede indicar que tanto un cómputo de tramas de límite de programa como un valor de desplazamiento se incluyen en la trama.

En algunas realizaciones, al menos una trama de un tren de bits de audio codificado AC-3 o E-AC-3 incluye un segmento de metadatos que incluye LPSM y metadatos de límite de programa (y, de manera opcional, también otros metadatos) para un programa de audio determinado por el tren de bits. Cada segmento de metadatos (que pueden incluirse en un campo addbsi, o un campo auxdata o un segmento de bits de residuo del tren de bits) contiene un encabezamiento principal (y, de forma opcional, también elementos principales adicionales), y después del encabezamiento principal (o el encabezamiento principal y otros elementos principales) un segmento de carga útil de LPSM (o contenedor) que tiene el siguiente formato:

un encabezamiento (que normalmente incluye al menos un valor de identificación, p. ej., versión de formato de LPSM, longitud, período, cómputo y valores de asociación de subtren), y

después del encabezamiento, los metadatos de límite de programa (que pueden incluir un cómputo de tramas de límite de programa, un valor de código (p. ej., un valor "desplazamiento_existe") indicativo de si la trama incluye solamente un cómputo de tramas de límite de programa o tanto un cómputo de tramas de límite de programa como un valor de desplazamiento y, en algunos casos, un valor de desplazamiento) y los LPSM. Los LPSM pueden incluir:

al menos un valor de indicación de diálogo que indica si los datos de audio correspondientes indican diálogo o no indican diálogo (p. ej., qué canales de los datos de audio correspondientes indican diálogo). El valor(es) de indicación de diálogo puede indicar si el diálogo está presente en cualquier combinación de, o todos, los canales de los datos de audio correspondientes;

al menos un valor de cumplimiento de regulación de sonoridad que indica si los datos de audio correspondientes cumplen con un conjunto indicado de regulaciones de sonoridad;

al menos un valor de procesamiento de sonoridad que indica al menos un tipo de procesamiento de sonoridad que se ha llevado a cabo en los datos de audio correspondientes; y

al menos un valor de sonoridad que indica al menos una característica de sonoridad (p. ej., sonoridad pico o promedio) de los datos de audio correspondientes.

En algunas realizaciones, el segmento de carga útil LPSM incluye un valor de código (un valor "desplazamiento_existe") indicativo de si la trama incluye solamente un cómputo de tramas de límite de programa o tanto un cómputo de tramas de límite de programa como un valor de desplazamiento. Por ejemplo, en dicha realización, cuando dicho valor de código indica (p. ej., cuando desplazamiento_existe = 1) que la trama incluye un cómputo de tramas de límite de programa y un valor de desplazamiento, el segmento de carga útil de LPSM puede incluir un valor de desplazamiento que es un entero no firmado de 11 bits (a saber, que tiene un valor de 0 a 2048) y que indica el número de muestras de audio adicionales entre el límite de trama señalizado (el límite de la trama que incluye el límite de programa) y el límite de programa real. Si el cómputo de tramas de límite de programa indica el número de tramas (a la velocidad de trama actual) a la trama que contiene límite de programa, la ubicación precisa (en unidades de número de muestras) del límite de programa (respecto al inicio o final de la trama que incluye el segmento de carga útil LPSM) debe calcularse de la siguiente manera:

M = (cómputo_trama * tamaño de trama) + desplazamiento,

donde M es el número de muestras para el límite de programa (desde el inicio o el final de la trama que incluye el segmento de carga útil de LPSM), "cómputo_trama" es el cómputo de tramas indicado por el cómputo de tramas de límite de programa, "tamaño de trama" es el número de muestras por trama, y "desplazamiento" es el número de muestras indicadas por el valor de desplazamiento.

Algunas realizaciones en las que la velocidad de inserción de banderas de límite de programa aumenta cerca del límite de programa real implementan una norma en la que un valor de desplazamiento nunca se incluye en una trama si la trama es menor o igual a cierto número ("Y") de tramas a partir de la trama que incluye el límite de programa. Normalmente, Y = 32. Para un codificador E-AC-3 que implementa dicha norma (con Y = 32), el codificador nunca inserta un valor de desplazamiento en el segundo final de un programa de audio. En el presente caso, el dispositivo de recepción es responsable de mantener un temporizador y, por consiguiente, de llevar a cabo su propio cálculo de desplazamiento (en respuesta a metadatos de límite de programa, incluido un valor de desplazamiento, en una trama del tren de bits codificado que es mayor que Y tramas a partir de la trama que contiene el límite de programa).

Para programas cuyos programas de audio se conoce que tienen la "trama alineada" a tramas de vídeo de programas de vídeo correspondientes (p. ej., típicas alimentaciones de contribución con audio codificado Dolby E), es superfluo incluir valores de desplazamiento en los trenes de bits codificados indicativos de los programas de audio. Por consiguiente, los valores de desplazamiento no se incluirán, normalmente, en dichos trenes de bits codificados.

Con referencia a la Figura 11, a continuación se considerarán casos en los que los trenes de bits de audio codificados se empalman, de manera conjunta, para generar una realización del tren de bits de audio inventivo.

El tren de bits en la parte superior de la Figura 11 (etiquetado "Escenario 1") es indicativo de todo un primer programa de audio (P1) que incluye metadatos de límite de programa (banderas de límite de programa, B) seguidas de todo un segundo programa de audio (P2) que también incluye metadatos de límite de programa (banderas de límite de programa, B). Las banderas de límite de programa en la porción final del primer programa (algunas de las cuales se muestran en la Figura 11) son idénticas o similares a aquellas descritas con referencia a la Figura 8 y determinan la ubicación del límite entre los dos programas (a saber, el límite al comienzo del segundo programa). Las banderas de límite de programa en la porción inicial del segundo programa (algunas de las cuales se muestran en la Figura 11) son idénticas o similares a aquellas descritas con referencia a la Figura 9 y también determinan la ubicación del límite. En realizaciones típicas, un codificador o descodificador implementa un temporizador (calibrado por las banderas en el primer programa) que cuenta, de manera regresiva, hasta el límite de programa, y el mismo temporizador (calibrado

por las banderas en el segundo programa) cuenta, de manera ascendente, desde el mismo límite de programa. Como se indica por el gráfico de temporizador de límite en el Escenario 1 de la Figura 11, dicha cuenta regresiva del temporizador (calibrado por las banderas en el primer programa) alcanza cero en el límite, y la cuenta ascendente del temporizador (calibrado por las banderas en el segundo programa) indica la misma ubicación del límite.

El segundo tren de bits de la parte superior de la Figura 11 (etiquetado "Escenario 2") es indicativo de todo un primer programa de audio (P1) que incluye metadatos de límite de programa (banderas de límite de programa, B) seguidas de todo un segundo programa de audio (P2) que no incluye metadatos de límite de programa. Las banderas de límite de programa en la porción final del primer programa (algunas de las cuales se muestran en la Figura 11) son idénticas o similares a aquellas descritas con referencia a la Figura 8 y determinan la ubicación del límite entre los dos programas (a saber, el límite al comienzo del segundo programa), como en el Escenario 1. En realizaciones típicas, un codificador o descodificador implementa un temporizador (calibrado por las banderas en el primer programa) que cuenta, de manera regresiva, hasta el límite de programa, y el mismo temporizador (sin calibrarse más) continúa contando, de manera ascendente, desde el límite de programa (como se indica por el gráfico de temporizador de límite en el Escenario 2 de la Figura 11).

El tercer tren de bits de la parte superior de la Figura 11 (etiquetado "Escenario 3") es indicativo de un primer programa de audio truncado (P1) que incluye metadatos de límite de programa (banderas de límite de programa, B) y que se han empalmado con todo un segundo programa de audio (P2) que también incluye metadatos de límite de programa (banderas de límite de programa, B). El empalme ha eliminado las últimas "N" tramas del primer programa. Las banderas de límite de programa en la porción de comienzo del segundo programa (algunas de las cuales se muestran en la Figura 11) son idénticas o similares a aquellas descritas con referencia a la Figura 9 y determinan la ubicación del límite (empalme) entre el primer programa truncado y todo el segundo programa. En realizaciones típicas, un codificador o descodificador implementa un temporizador (calibrado por las banderas en el primer programa) que cuenta, de manera regresiva, hasta el final del primer programa no truncado, y el mismo temporizador (calibrado por las banderas en el segundo programa) cuenta, de manera ascendente, desde el comienzo del segundo programa. El comienzo del segundo programa es el límite de programa en el Escenario 3. Como se indica por el gráfico de temporizador de límite en el Escenario 3 de la Figura 11, dicha cuenta regresiva del temporizador (calibrado por los metadatos de límite de programa en el primer programa) se restablece (en respuesta a los metadatos de límite de programa en el segundo programa) antes de que haya alcanzado cero (en respuesta a los metadatos de límite de programa en el primer programa). Por consiguiente, aunque el truncamiento del primer programa (por el empalme) evita que el temporizador identifique el límite de programa entre el primer programa truncado y el comienzo del segundo programa en respuesta a (a saber, bajo la calibración por) los metadatos de límite de programa en el primer programa solo, los metadatos de programa en el segundo programa restablecen el temporizador, de modo que el temporizador restablecido correctamente indica (como la ubicación correspondiente al cómputo "cero" del temporizador restablecido) la ubicación del límite de programa entre el primer programa truncado y el comienzo del segundo programa.

El cuarto tren de bits (etiquetado "Escenario 4") es indicativo de un primer programa de audio truncado (P1) que incluye metadatos de límite de programa (banderas de límite de programa, B), y un segundo programa de audio truncado (P2) que incluye metadatos de límite de programa (banderas de límite de programa, B), y que se ha empalmado con una porción (la porción no truncada) del primer programa de audio. Las banderas de límite de programa en la porción de comienzo de todo el segundo programa (pretruncamiento) (algunas de las cuales se muestran en la Figura 11) son idénticas o similares a aquellas descritas con referencia a la Figura 9 y las banderas de límite de programa en la porción final de todo el primer programa (pretruncamiento) (algunas de las cuales se muestran en la Figura 11) son idénticas o similares a aquellas descritas con referencia a la Figura 8. El empalme ha eliminado las últimas "N" tramas del primer programa (y, por consiguiente, algunas de las banderas de límite de programa que se habían incluido allí antes del empalme) y las primeras "M" tramas del segundo programa (y, por consiguiente, algunas de las banderas de límite de programa que se habían incluido allí antes del empalme). En realizaciones típicas, un codificador o descodificador implementa un temporizador (calibrado por las banderas en el primer programa truncado) que cuenta, de manera regresiva, hacia el final del primer programa no truncado, y el mismo temporizador (calibrado por las banderas en el segundo programa truncado) cuenta, de manera ascendente, desde el comienzo del segundo programa no truncado. Como se indica por el gráfico de temporizador de límite en el Escenario 4 de la Figura 11, dicha cuenta regresiva del temporizador (calibrado por los metadatos de límite de programa en el primer programa) se restablece (en respuesta a los metadatos de límite de programa en el segundo programa) antes de que haya alcanzado cero (en respuesta a los metadatos de límite de programa en el primer programa). El truncamiento del primer programa (por el empalme) evita que el temporizador identifique el límite de programa entre el primer programa truncado y el comienzo del segundo programa truncado) en respuesta a (a saber, bajo la calibración por) los metadatos de límite de programa en el primer programa solo. Sin embargo, el temporizador restablecido no indica correctamente la ubicación del límite de programa entre el fin del primer programa truncado y el comienzo del segundo programa truncado. Por consiguiente, el truncamiento de ambos trenes de bits empalmados puede prevenir la determinación exacta del límite entre ellos.

Las realizaciones de la presente invención pueden implementarse en hardware, firmware o software, o una combinación de ellos (p. ej., como una matriz lógica programable). Salvo que se especifique lo contrario, los algoritmos o procesos incluidos como parte de la invención no se relacionan de forma inherente con un ordenador particular u

otro aparato. En particular, varias máquinas de propósito general pueden usarse con programas escritos según las enseñanzas de la presente memoria, o puede ser más conveniente construir aparatos más especializados (p. ej., circuitos integrados) para llevar a cabo las etapas requeridas del método. Por consiguiente, la invención puede implementarse en uno o más programas de ordenador mediante la ejecución de uno o más sistemas informáticos programables (p. ej., una implementación de cualquiera de los elementos de la Figura 1, o el codificador 100 de la Figura 2 (o un elemento de este), o el descodificador 200 de la Figura 3 (o un elemento de este) o postprocesador 300 de la Figura 3 (o un elemento de este)) comprendiendo, cada uno, al menos un procesador, al menos un sistema de almacenamiento de datos (incluida una memoria y/o elementos de almacenamiento no permanentes y permanentes), al menos un dispositivo o puerto de entrada y al menos un dispositivo o puerto de salida. El código de programa se aplica a los datos de entrada para llevar a cabo las funciones descritas en la presente memoria y generar información de salida. La información de salida se aplica a uno o más dispositivos de salida, de manera conocida.

Cada programa puede implementarse en cualquier lenguaje informático deseado (incluidos lenguajes de programación de procedimiento, lógicos u orientados al objeto de máquina, conjunto o alto nivel) para comunicarse con un sistema informático. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado.

Por ejemplo, cuando se implementan por secuencias de instrucción de software de ordenador, varias funciones y etapas de las realizaciones de la invención pueden implementarse por secuencias de instrucción de software multihilo que se ejecutan en hardware de procesamiento de señal digital apropiado, en cuyo caso los diferentes dispositivos, etapas y funciones de las realizaciones pueden corresponder a porciones de las instrucciones de software.

Cada programa de ordenador se almacena, preferiblemente, en un medio o dispositivo, o se descarga en este, de almacenamiento (p. ej., memoria o medios de estado sólido, o medios magnéticos u ópticos) legible por un ordenador programable de propósito general o especial, para configurar y hacer funcionar el ordenador cuando el medio o dispositivo de almacenamiento es leído por el sistema informático para llevar a cabo los procedimientos descritos en la presente memoria. El sistema inventivo puede implementarse también como medio de almacenamiento legible por ordenador, configurado con (a saber, que almacena) un programa de ordenador, donde el medio de almacenamiento así configurado hace que un sistema informático funcione de una manera específica y predefinida para llevar a cabo las funciones descritas en la presente memoria.

Se han descrito varias realizaciones de la invención. Sin embargo, se comprenderá que pueden llevarse a cabo diversas modificaciones sin apartarse del alcance de la invención. Son posibles numerosas modificaciones y variaciones de la presente invención a la luz de las enseñanzas de más arriba. Se comprenderá que dentro del alcance de las reivindicaciones anexas, la invención puede practicarse de manera diferente a como se describe específicamente en la presente memoria.

REIVINDICACIONES

1. Una unidad de procesamiento de audio (100, 200), que comprende:

una memoria intermedia (110, 201), configurada para almacenar al menos una trama de un tren de bits de audio codificado, en donde el tren de bits de audio codificado incluye datos de audio y un contenedor de metadatos, el contenedor de metadatos incluye un encabezamiento, una o más cargas útiles de metadatos después del encabezamiento, y datos de protección después de la una o más cargas útiles de metadatos, la una o más cargas útiles de metadatos incluyen metadatos indicativos de estado de procesamiento de sonoridad de los datos de audio, y los metadatos indicativos de estado de procesamiento de sonoridad de los datos de audio son o incluyen metadatos indicativos de al menos un tipo de procesamiento de sonoridad realizado en los datos de audio, en donde los datos de protección pueden ser usados para verificar la integridad del contenedor de metadatos y la una o más cargas útiles dentro del contenedor de metadatos;

un analizador (111, 205) acoplado a la memoria intermedia (110, 201) y configurado para analizar el tren de bits de audio codificado; y

un subsistema acoplado al analizador (111, 205) y configurado para realizar procesamiento de sonoridad adaptativo usando al menos algunos de los metadatos indicativos del estado de procesamiento de sonoridad de los datos de audio.

2. La unidad de procesamiento de audio (100, 200) de la reivindicación 1, en donde los metadatos indicativos de estado de procesamiento de sonoridad de los datos de audio incluyen metadatos indicativos de al menos una sonoridad o rango dinámico característicos de los datos de audio.

3. La unidad de procesamiento de audio (100, 200) de cualquier reivindicación anterior, en donde el procesamiento de sonoridad adaptativo es o incluye llevar a cabo control de rango dinámico.

4. La unidad de procesamiento de audio de cualquier reivindicación anterior, que incluye también:

un descodificador de audio (101, 202) acoplado y configurado para descodificar los datos de audio generando de ese modo datos de audio descodificados.

5. La unidad de procesamiento de audio (100, 200) de la reivindicación 4, en donde el subsistema acoplado al analizador (111, 205) también se acopla al descodificador de audio (101, 202), y dicho subsistema se configura para realizar procesamiento de sonoridad adaptativo en al menos algunos de los datos de audio descodificados usando al menos algunos de los metadatos indicativos de estado de procesamiento de sonoridad de los datos de audio, y opcionalmente,

en donde el procesamiento de sonoridad adaptativo es o incluye llevar a cabo control de rango dinámico.

6. La unidad de procesamiento de audio (100, 200) de la reivindicación 4, en donde la una o más cargas útiles de metadatos incluyen una carga útil de sonoridad de programa que contiene datos indicativos de una sonoridad medida de un programa de audio asociado con los datos de audio, el subsistema acoplado al analizador (111, 205) también se acopla al descodificador de audio (101, 202), y dicho subsistema se configura para realizar procesamiento de sonoridad adaptativo en al menos algunos de los datos de audio usando la carga útil de sonoridad de programa.

7. La unidad de procesamiento de audio (100, 200) de cualquier reivindicación anterior, en donde la una o más cargas útiles de metadatos incluyen una carga útil de sonoridad de programa que contiene datos indicativos de una sonoridad medida de un programa de audio asociado con los datos de audio, y opcionalmente,

en donde la carga útil de sonoridad de programa incluye un campo que indica un método de medición de sonoridad que ha sido usado para generar datos de sonoridad contenidos en la carga útil de sonoridad de programa.

8. Un método de procesamiento de audio, que comprende las etapas de:

recibir un tren de bits de audio codificado, en donde el tren de bits de audio codificado se segmenta en una o más tramas: almacenar la una o más tramas del tren de bits de audio codificado en una memoria intermedia;

extraer datos de audio y un contenedor de metadatos del tren de bits de audio codificado, en donde el contenedor de metadatos incluye un encabezamiento, una o más cargas útiles de metadatos después del encabezamiento, y datos de protección después de la una o más cargas útiles de metadatos la una o más cargas útiles de metadatos incluyen metadatos indicativos de estado de procesamiento de sonoridad de los datos de audio, y los metadatos indicativos de estado de procesamiento de sonoridad de los datos de audio son o incluyen metadatos indicativos de al menos un tipo de procesamiento de sonoridad realizado en los datos de audio, en donde los datos de protección se pueden usar para verificar la integridad del contenedor de metadatos y la una o más cargas útiles dentro del contenedor de metadatos; y

realizar procesamiento de sonoridad adaptativo usando al menos algunos de los metadatos indicativos del estado de procesamiento de sonoridad de los datos de audio.

9. El método de la reivindicación 8, en donde los metadatos indicativos de estado de procesamiento de sonoridad de los datos de audio incluyen metadatos indicativos de al menos una sonoridad o rango dinámico características de los datos de audio.

10. El método de una cualquiera de las reivindicaciones 8 a 9, en donde los datos de audio son datos de audio codificados, y también comprende una etapa de:

descodificar los datos de audio codificados para generar datos de audio descodificados.

11. El método de la reivindicación 10, en donde la etapa de llevar a cabo procesamiento de sonoridad adaptativo comprende llevar a cabo procesamiento de sonoridad adaptativo en al menos algunos de los datos de audio descodificados usando dichos algunos de los metadatos indicativos de estado de procesamiento de sonoridad de los datos de audio.

12. El método de la reivindicación 11, en donde el procesamiento de sonoridad adaptativo es o incluye llevar a cabo control de rango dinámico.

13. El método de una cualquiera de las reivindicaciones 8 a 12, en donde la una o más cargas útiles de metadatos incluyen una carga útil de sonoridad de programa que contiene datos indicativos de una sonoridad medida de un programa de audio asociado con los datos de audio, y en donde la etapa de llevar a cabo procesamiento de sonoridad adaptativo comprende llevar a cabo procesamiento de sonoridad adaptativo en al menos algunos de los datos de audio extraídos del tren de bits de audio codificado usando la carga útil de sonoridad de programa.

14. El método de una cualquiera de las reivindicaciones 8 a 13, en donde el procesamiento de sonoridad adaptativo es o incluye llevar a cabo control de rango dinámico.

15. El método de una cualquiera de las reivindicaciones 8 a 14, en donde la una o más cargas útiles de metadatos incluyen una carga útil de sonoridad de programa que contiene datos indicativos de una sonoridad medida de un programa de audio asociado con los datos de audio, y en donde la carga útil de sonoridad de programa incluye un campo que indica un método de medición de sonoridad que se ha usado para generar datos de sonoridad contenidos en la carga útil de sonoridad de programa.

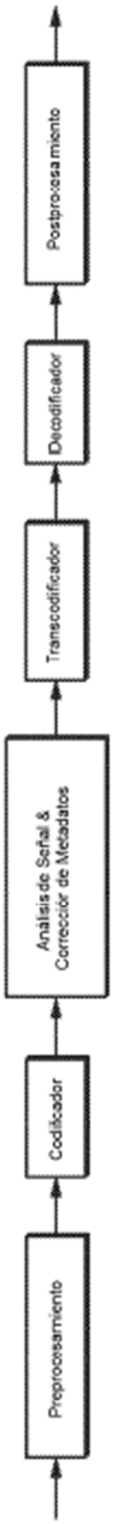
Preprocesamiento
Codificador
Análisis de Señal & Corrección de Metadatos
Transcodificador
Decodificador
Postprocesamiento

FIG. 1

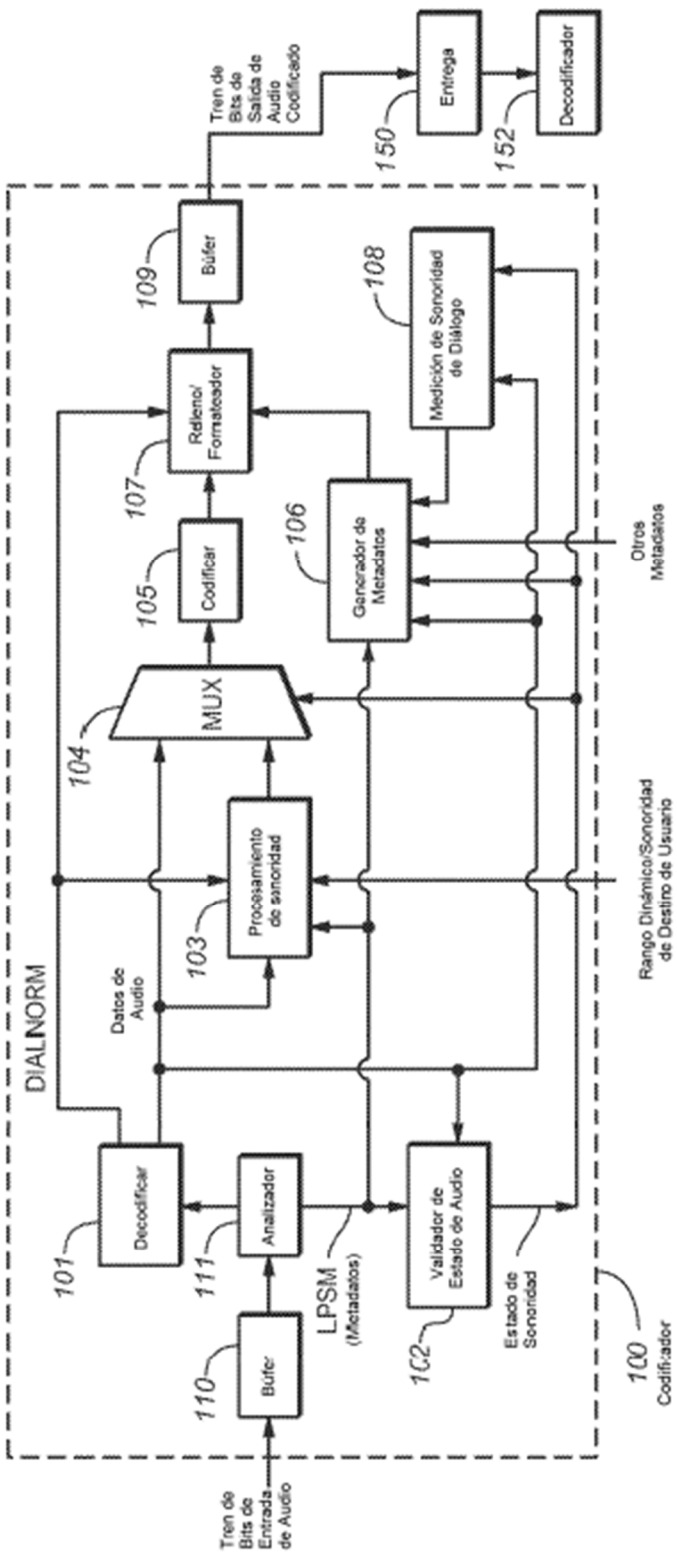
DIALNORM
Tren de Bits de Entrada de Audio
Búfer
110
Analizador
111
Decodificar
101
Datos de Audio
Procesamiento de sonoridad
103
MUX
104
Codificar
105
Relleno/ Formateador
107
Búfer
109
Tren de Bits de Salida de Audio Codificado
Generador de Metadatos
106
Medición de Sonoridad de Diálogo
108
Validador de Estado de Audio
102
LPSM (Metadatos)
Estado de Sonoridad
Rango Dinámico/Sonoridad de Destino de Usuario
Otros Metadatos
Entrega
150
Decodificador
152
100 Codificador

FIG. 2

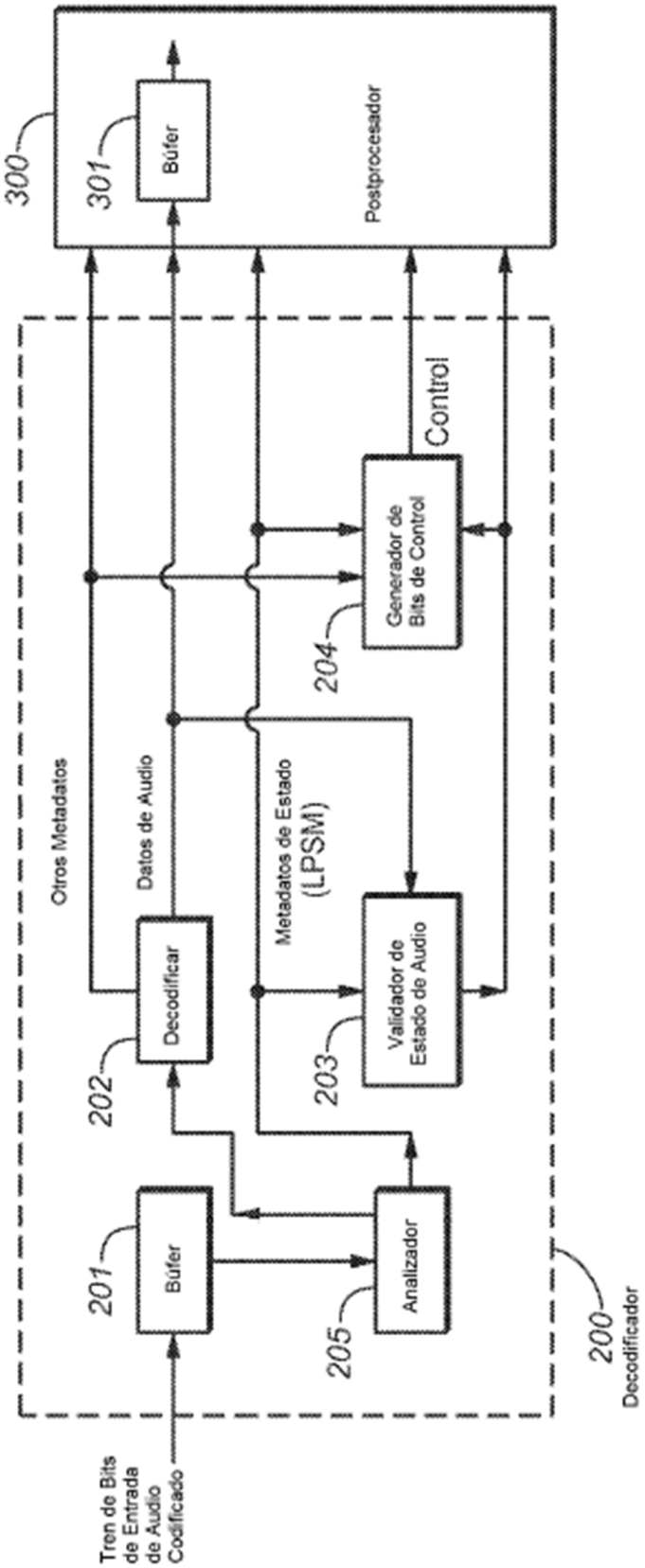
300
301
Búfer
Postprocesador
Control
Generador de Bits de Control
204
Otros Metadatos
Datos de Audio
Metadatos de Estado (LPSM)
Decodificar
202
Validador de Estado de Audio
203
201
Búfer
Analizador
205
200
Decodificador
Tren de Bits de Entrada de Audio Codificado

FIG. 3

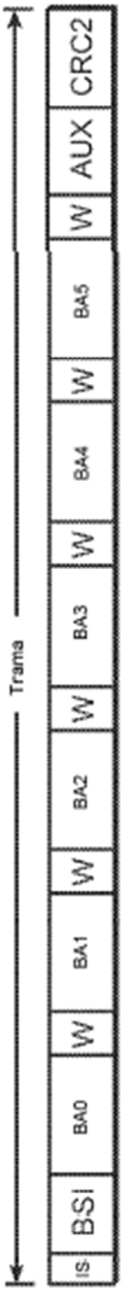

FIG. 4

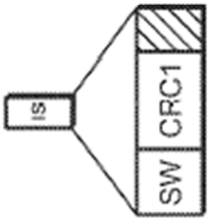

FIG. 5

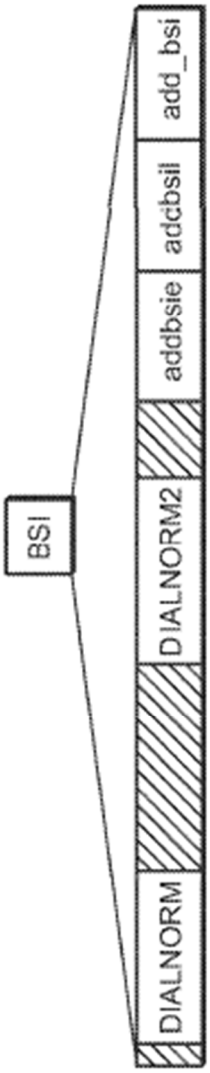

FIG. 6

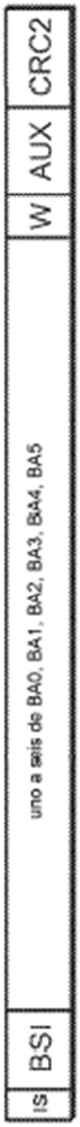

FIG. 7

Trama #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Trama para Límite	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	B
Contenidos de Bandera de Límite de Programa	100011	0	0	0	0	0	0	0	10011	0	0	0	1011	0	111	0	B
																	B
Existen	1	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	B
Contador de Tramas	0001->16								001->8				01->4		1->2		B
Cuenta Ascendente (+) / Regresiva (-)	1								1						1		B

(Sintaxis de Bandera de Límite de Programa que Cuenta de Forma Regresiva hasta el Límite de Programa)

FIG. 8

Trama #	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Tramas para Límite	B	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Contenidos de Bandera de Límite de Programa	B	0	110	0	1010	0	0	0	10010	0	0	0	0	0	0	0	10010
	B																
Existen	B	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	
Contador de Tramas	B		1->2		01->4				001->8								0001->16
Cuenta Ascendente (+) / Regresiva (-)	B		0		0				0								0

(Sintaxis de Bandera de Límite de Programa que Cuenta de Manera Ascendente para el Comienzo del Programa)

FIG. 9

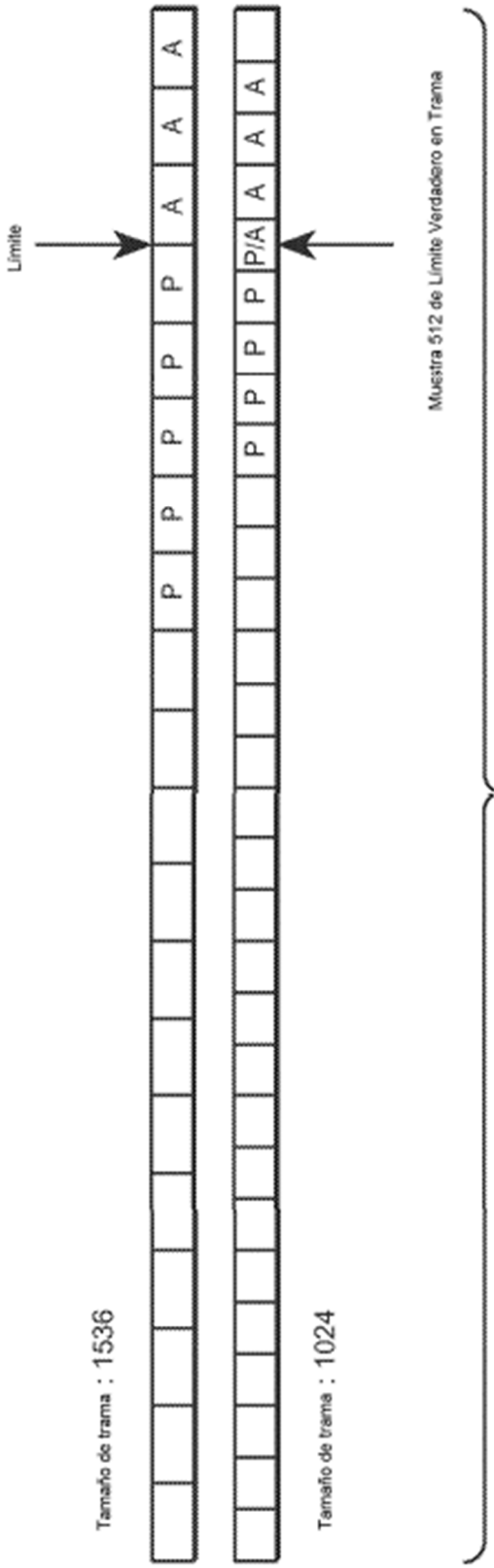
Límite
Tamaño de trama : 1536
P
P
P
P
P
A
A
A
A
Tamaño de trama : 1024
P
P
P
P
P/A
A
A
A
Muestra 512 de Límite Verdadero en Trama

FIG. 10

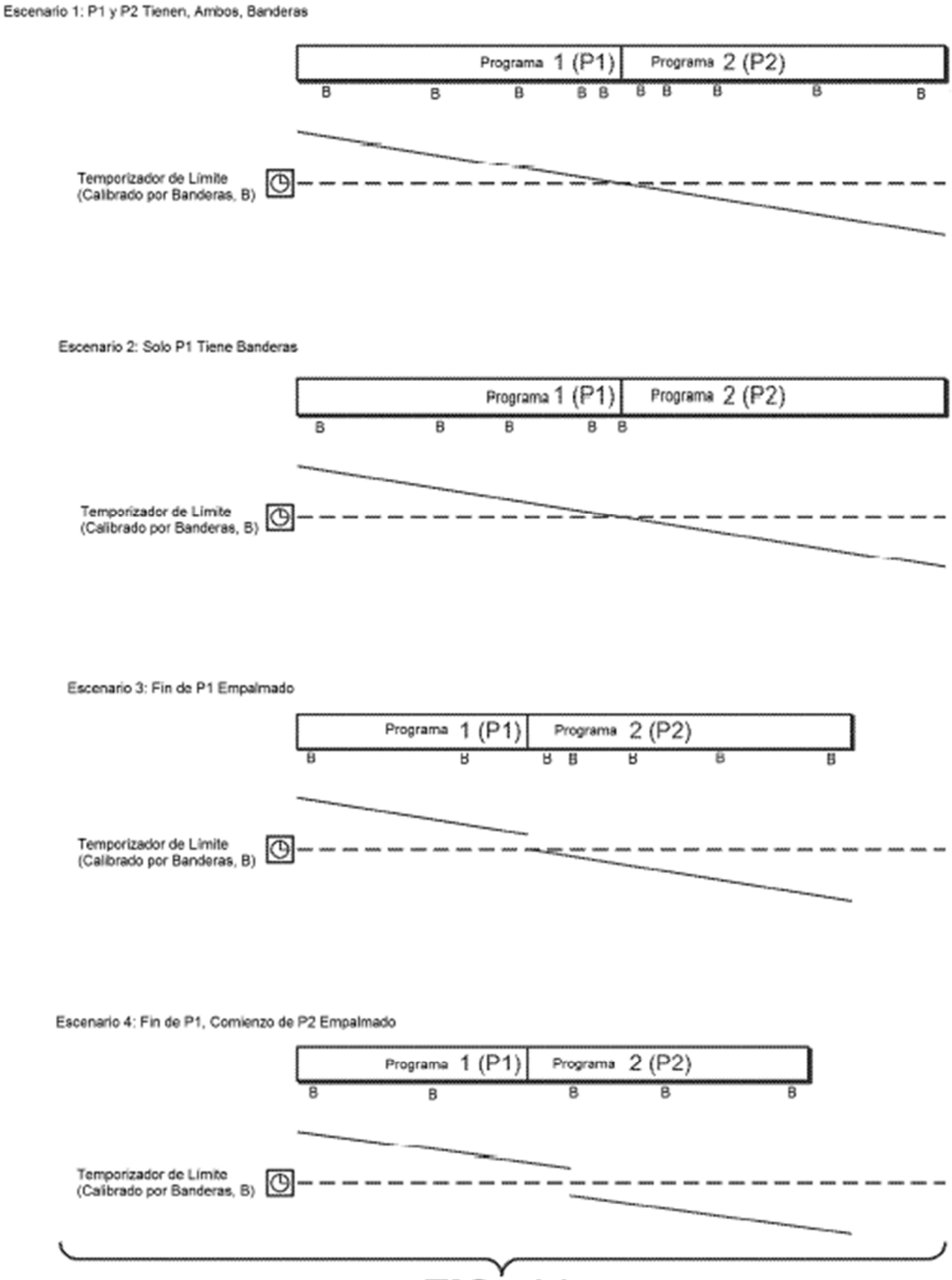
Escenario 1: P1 y P2 Tienen, Ambos, Banderas
Programa 1 (P1)
Programa 2 (P2)
B
Temporizador de Límite
(Calibrado por Banderas, B)
Escenario 2: Solo P1 Tiene Banderas
Programa 1 (P1)
Programa 2 (P2)
B
Temporizador de Límite
(Calibrado por Banderas, B)
Escenario 3: Fin de P1 Empalmado
Programa 1 (P1)
Programa 2 (P2)
B
Temporizador de Límite
(Calibrado por Banderas, B)
Escenario 4: Fin de P1, Comienzo de P2 Empalmado
Programa 1 (P1)
Programa 2 (P2)
B
Temporizador de Límite
(Calibrado por Banderas, B)

FIG. 11