

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 667 871**

51 Int. Cl.:

**G10L 19/16** (2013.01)

**H03G 5/00** (2006.01)

**H03G 9/02** (2006.01)

**H03G 9/00** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **15.01.2014 E 16164479 (4)**

97 Fecha y número de publicación de la concesión europea: **21.03.2018 EP 3082128**

54 Título: **Decodificador de audio con sonoridad y metadatos de límite 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:

**14.05.2018**

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:

**LEHMANN NOVO, María Isabel**

**ES 2 667 871 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

Decodificador de audio con sonoridad y metadatos de límite de programa

### REFERENCIA CRUZADA A SOLICITUDES RELACIONADAS

- 5 Esta solicitud reivindica prioridad a la Solicitud de Patente Provisional de los Estados Unidos N° 61/754.882, presentada el 21 de enero de 2013 y a la Solicitud de Patente Provisional de los Estados Unidos N° 61/824.010, presentada el 16 de mayo de 2013.

### CAMPO TÉCNICO

- 10 La invención se refiere al procesamiento de señales de audio y, más particularmente, a la codificación y decodificación de flujos de bits de datos de audio con metadatos indicativos del estado de procesamiento de sonoridad del contenido de audio y la ubicación de límites de programa de audio indicados por los flujos de bits. Algunas realizaciones de la invención generan o decodifican 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 INVENCION

- 15 Dolby, Dolby Digital, Dolby Digital Plus y Dolby E son marcas comerciales de Dolby Laboratories Licensing Corporation. Dolby Laboratories proporciona implementaciones patentadas de AC-3 y E-AC-3 conocidas como Dolby Digital y Dolby Digital Plus, respectivamente.

- 20 Las unidades de procesamiento de datos de audio normalmente funcionan a ciegas y no prestan atención al historial de procesamiento de los datos de audio que ocurren antes de que se reciban los datos. Esto puede funcionar en una trama de procesamiento en el que una sola entidad hace todo el procesamiento y codificación de datos de audio para una variedad de dispositivos de renderización de medios de destino, mientras que un dispositivo de renderizado de medios de destino hace toda la decodificación y renderización de los datos de audio codificados. Sin embargo, este procesamiento ciego no funciona bien (o no funciona) en situaciones en las que una pluralidad de unidades de procesamiento de audio están diseminadas a través de una red diversa o se colocan en tándem (es decir, cadena) y se espera que ejecuten sus respectivos procesamientos de tipos de audio de forma óptima. Por ejemplo, algunos datos de audio pueden estar codificados para sistemas de medios de alto rendimiento y pueden tener que convertirse a una forma reducida adecuada para un dispositivo móvil a lo largo de una cadena de procesamiento de medios. En consecuencia, una unidad de procesamiento de audio puede realizar innecesariamente un tipo de procesamiento sobre los datos de audio que ya se ha realizado. Por ejemplo, una unidad de nivelación de volumen puede realizar el procesamiento en un clip de audio de entrada, independientemente de si se ha realizado anteriormente o no la misma nivelación de volumen en el clip de audio de entrada. Como resultado, la unidad de nivelación de volumen puede realizar la nivelación incluso cuando no es necesario. Este procesamiento innecesario también puede causar la degradación y/o la eliminación de características específicas mientras se renderiza el contenido de los datos de audio.

- 35 Un flujo 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 flujo de bits AC-3 hay varios parámetros de metadatos de audio que están específicamente destinados a ser utilizados para cambiar el sonido del programa entregado a un entorno de escucha. Uno de los parámetros de metadatos es el parámetro DIALNORM, el cual está destinado a indicar el nivel medio de diálogo que se produce en un programa de audio, y se utiliza para determinar el nivel de señal de reproducción de audio.

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

- 50 Normalmente, DIALNORM lo configura un usuario y no se genera automáticamente, aunque hay un valor predeterminado de DIALNORM si no establece ningún valor por el usuario. 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 ajustar el valor de DIALNORM. Por lo tanto, se depende del creador del contenido para ajustar correctamente el parámetro DIALNORM.

5 Hay varias razones diferentes por las que el parámetro DIALNORM en un flujo de bits AC-3 puede ser incorrecto. En primer lugar, cada uno de los codificadores AC-3 tiene un valor predeterminado de DIALNORM que se utiliza durante la generación del flujo de bits si un valor de DIALNORM no se ajusta por el creador del contenido. Este valor predeterminado puede ser sustancialmente diferente del nivel de sonoridad de diálogo real del audio. En segundo lugar, incluso si un creador de contenido mide la sonoridad y ajusta el valor de DIALNORM en consecuencia, es posible que se haya utilizado un algoritmo de medición o medidor de sonoridad que no cumpla con el método de medición de sonoridad AC-3 recomendado, dando como resultado un valor de DIALNORM incorrecto. En tercer lugar, incluso si se ha creado un flujo de bits AC-3 con el valor de DIALNORM medido y configurado correctamente por el creador del contenido, puede haberse cambiado a un valor incorrecto durante la transmisión y/o el almacenamiento del flujo de bits. Por ejemplo, no es raro que las aplicaciones de transmisión de televisión para flujos de bits AC-3 se decodifiquen, modifiquen y luego se vuelvan a codificar utilizando información de metadatos DIALNORM incorrecta. Por lo tanto, un valor de DIALNORM incluido en un flujo de bits AC-3 puede ser incorrecto o impreciso y, por lo tanto, puede tener un impacto negativo en la calidad de la experiencia de escucha.

Además, el parámetro DIALNORM no indica el estado de procesamiento de sonoridad de los correspondientes datos de audio (p. ej., qué tipo(s) de procesamiento de sonoridad se ha(n) realizado sobre los datos de audio). Hasta la presente invención, un flujo de bits de audio no incluía metadatos, indicativos del estado de procesamiento de sonoridad (p. ej., tipo(s) de procesamiento de sonoridad aplicado(s) a) el contenido de audio del flujo de bits de audio o el estado de procesamiento de sonoridad y sonoridad del contenido de audio del flujo de bits, en un formato del tipo descrito en la presente divulgación. Los metadatos de estado de procesamiento de sonoridad en tal formato son útiles para facilitar el procesamiento de sonoridad adaptativo de un flujo de bits de audio y/o la verificación de la validez del estado de procesamiento de sonoridad y sonoridad del contenido de audio, de una manera particularmente eficiente.

Aunque la presente invención no está limitada a ser utilizada con un flujo de bits AC-3, un flujo de bits E-AC-3, o un flujo de bits Dolby E, por conveniencia se describirá en realizaciones en las que genera, decodifica o procesa de otro modo, tal flujo de bits que incluye metadatos de estado de procesamiento de sonoridad.

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

35 Los detalles de la codificación AC-3 (también conocida como Dolby Digital) son bien conocidos y se exponen muchas referencias publicadas incluyendo las siguientes:

*Estándar ATSC A52/A: estándar de compresión de audio digital (AC-3), revisión A*, Comité de Sistemas Avanzados de Televisión, 20 de agosto de 2001; y

Solicitud de Patente Internacional WO 2006/113062 por B.G. Crockett, "Audio Metadata Verification", 26.10.2006; y las patentes de Estados Unidos 5.583.962; 5.632.005; 5.633.981; 5.727.119; y 6.021.386.

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

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

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

50 Cada una de las tramas de un flujo 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 frecuencia de muestreo de 48 kHz, esto representa 5,333, 10,667, 16 o 32 milisegundos de audio digital, respectivamente, o una tasa de 189,9, 93,75, 62,5 o 31,25 tramas por segundo de audio, respectivamente.

5 Como se indica en la Fig. 4, cada una de las tramas AC-3 está dividida en secciones (segmentos), que incluyen: una sección de Información de Sincronización (SI) que contiene (como se muestra en la Fig. 5) una palabra de sincronización (SW) y la primera de dos palabras de corrección de errores (CRC1); una sección de Información de Flujo de Bits (BSI) que contiene la mayoría de los metadatos; seis Bloques de Audio (AB0 a AB5) que contienen datos de contenido de audio comprimido (y también pueden incluir metadatos); segmentos de bits residuales (W) que contienen los bits no utilizados 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 error (CRC2). El segmento de bits residuales (W) también se puede denominar "campo de omisión".

15 Como se indica en la Fig. 7, cada una de las tramas E-AC-3 está dividida en secciones (segmentos), que incluyen: una sección de Información de Sincronización (SI) que contiene (como se muestra en la Fig. 5) una palabra de sincronización (SW); una sección de Información de Flujo de Bits (BSI) que contiene la mayoría de los metadatos; entre uno y seis Bloques de Audio (AB0 a AB5) que contienen datos de contenido de audio comprimido (y también pueden incluir metadatos); segmentos de bits residuales (W) que contienen los bits no utilizados que quedan después de comprimir el contenido de audio (aunque solo se muestra un segmento de bits residuales, un segmento de bits residuales diferente típicamente seguiría a cada uno de los bloques de audio); una sección de información Auxiliar (AUX) que puede contener más metadatos; y una palabra de corrección de errores (CRC). El segmento de bits residuales (W) también se puede denominar "campo de omisión".

20 En un flujo de bits AC-3 (o E-AC-3) hay varios parámetros de metadatos de audio que están específicamente destinados para ser utilizados para cambiar el sonido del programa entregado a un entorno de escucha. Uno de los parámetros de metadatos es el parámetro DIALNORM, el cual se incluye en el segmento BSI.

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

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

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

#### Breve Descripción de la Invención

35 En una clase de realizaciones, la invención es una unidad de procesamiento de audio que incluye una memoria de búfer, un decodificador de audio y un analizador. La memoria de búfer almacena al menos una trama de un flujo de bits de audio codificado. El flujo de bits de audio codificado incluye datos de audio y un contenedor de metadatos. El contenedor de metadatos incluye una cabecera, una o más cargas útiles de metadatos y datos de protección. La cabecera incluye una palabra de sincronización que identifica el inicio del contenedor. La una o más cargas útiles de metadatos describen un programa de audio asociado con los datos de audio. Los datos de protección están ubicados después de la una o más cargas útiles de metadatos. Los datos de protección también pueden utilizarse para verificar la integridad del contenedor de metadatos y de la una o más cargas útiles dentro del contenedor de metadatos. El decodificador de audio está acoplado a la memoria de búfer y es capaz de decodificar los datos de audio. El analizador está acoplado a o integrado con el decodificador de audio y es capaz de analizar el contenedor de metadatos.

40 En realizaciones típicas, el método incluye recibir un flujo de bits de audio codificado, donde el flujo de bits de audio codificado se segmenta en una o más tramas. Los datos de audio se extraen del flujo de bits de audio codificado, junto con un contenedor de metadatos. El contenedor de metadatos incluye una cabecera seguida de una o más cargas útiles de metadatos seguidas de datos de protección. Finalmente, se verifica la integridad del contenedor y de la una o más cargas útiles de metadatos mediante la utilización de los datos de protección. La una o más cargas útiles de metadatos pueden incluir una carga útil de sonoridad de programa que contiene datos indicativos de la sonoridad medido de un programa de audio asociado con los datos de audio.

Se pueden autenticar y validar una carga útil de metadatos de sonoridad de programa, referidos como metadatos de estado de procesamiento de sonoridad (“LPSM”), incrustados en un flujo de bits de audio de acuerdo con las realizaciones típicas de la invención, p. ej., para permitir que las entidades reguladoras de sonoridad verifiquen si una sonoridad de un programa particular ya se encuentra dentro de un rango específico y que los correspondientes datos de audio no se han modificado (garantizando así el cumplimiento de la normativa aplicable). Se puede leer un valor de sonoridad incluido en un bloque de datos que comprende los metadatos de estado de procesamiento de sonoridad para verificar esto, en lugar de calcular la sonoridad de nuevo. En respuesta a LPSM, una agencia reguladora puede determinar que el contenido de audio correspondiente cumple (según lo indicado por los LPSM) con requisitos reglamentarios y/o regulatorios de sonoridad (p. ej., las regulaciones promulgadas bajo la Ley de mitigación de la sonoridad publicitaria comercial, también conocida como la ley “CALM”) sin la necesidad de calcular la sonoridad del contenido de audio.

Las mediciones de sonoridad que se requieren para cumplir con algunos requisitos reglamentarios y/o regulatorios de sonoridad (p. ej., las regulaciones promulgadas en virtud de la Ley CALM) se basan en la sonoridad de programa integrada. La sonoridad de programa integrada requiere que se realice una medición de sonoridad, ya sea del nivel de diálogo o del nivel de mezcla completa, sobre un programa de audio completo. Por lo tanto, para hacer mediciones de sonoridad de programa (p. ej., en varias etapas de la cadena de transmisión) para verificar el cumplimiento de los requisitos legales típicos, es esencial que las mediciones se realicen con conocimiento de qué datos de audio (y metadatos) determinan un programa de audio completo, y esto típicamente requiere conocimiento de la ubicación del comienzo y el final del programa (p. ej., durante el procesamiento de un flujo de bits indicativo de una secuencia de programas de audio).

De acuerdo con las realizaciones típicas de la presente invención, un flujo de bits de audio codificado es indicativo de al menos un programa de audio (p. ej., una secuencia de programas de audio) y los metadatos de límite de programa y los LPSM incluidos en el flujo de bits permiten reiniciar la medición de sonoridad de programa al final de un programa y, por lo tanto, proporciona una forma automatizada de medir la sonoridad de programa integrada. Las realizaciones típicas de la invención incluyen metadatos de límite de programa en un flujo de bits de audio codificado de una manera eficiente, lo cual permite la determinación precisa y robusta de al menos un límite entre programas de audio consecutivos indicados por el flujo de bits. Las realizaciones típicas permiten la determinación precisa y robusta de un límite de programa en el sentido de que permiten la determinación de límite de programa precisa, incluso en casos en los que flujos de bits indicativos de diferentes programas se empalman (para generar el flujo de bits inventivo) de una manera que trunca uno o los dos flujos de bits empalmados (y por lo tanto descarta los metadatos de límite de programa que se han incluido en al menos uno de los flujos de bits previo al empalme).

En realizaciones típicas, los metadatos de límite de programa en una trama del flujo de bits inventivo son un indicador de límite de programa indicativo de un recuento de tramas. Típicamente, el indicador es indicativo del número de tramas entre la trama actual (la trama que incluye el indicador) y un límite de programa (el comienzo o el final del programa de audio actual). En algunas realizaciones preferidas, los indicadores de límite de programa se insertan de manera simétrica y eficiente al comienzo y al final de cada uno de los segmentos de flujo de bits que es indicativo de un único programa (es decir, en tramas que se producen dentro de un número predeterminado de tramas después del comienzo del segmento, y en tramas que se producen dentro de un número predeterminado de tramas antes del final del segmento), de modo que cuando se concatenan dos tales segmentos de flujo de bits (para ser indicativos de una secuencia de dos programas), los metadatos de límite de programa pueden estar presentes (p. ej., simétricamente) en ambos lados del límite entre los dos programas.

Para limitar el aumento de tasa de datos que resulta de incluir metadatos de límite de programa en un flujo de bits de audio codificado (que puede ser indicativo de un programa de audio o una secuencia de programas de audio), en las realizaciones típicas, los indicadores de límite de programa se insertan solo en un subconjunto de las tramas del flujo de bits. Típicamente, la tasa de inserción de indicador de límite es una función no creciente de separación creciente de cada una de las tramas del flujo de bits (en la que se inserta un indicador) desde el límite de programa que está más cercano a cada una de dichas tramas, donde “tasa de inserción de indicador de límite” se refiere a la proporción promedio del número de tramas (indicativas de un programa) que incluyen un indicador de límite de programa al número de tramas (indicativas del programa) que no incluyen un indicador de límite de programa, donde el promedio es un promedio continuo a lo largo de un número (p. ej., un número relativamente pequeño) de tramas consecutivas del flujo de bits de audio codificado. En una clase de realizaciones, la tasa de inserción de indicador de límite es una función logarítmicamente decreciente de distancia creciente (de cada una de las ubicaciones de inserción de indicador) desde el límite de programa más cercano, y para cada una de las tramas que contiene indicador, que incluye uno de los indicadores, el tamaño del indicador en dicha trama que contiene indicador es igual o mayor que el tamaño de cada uno de los indicadores en una trama ubicada más cerca del límite del programa más cercano que dicha trama que contiene indicador (es decir, el tamaño del indicador de límite de programa en cada una de las tramas que contiene indicador es una función no decreciente de separación creciente de dicha trama que contiene indicador desde el límite de programa más cercano).

Otro aspecto de la invención es una unidad de procesamiento de audio (APU) configurada para realizar cualquier realización del método inventivo. En otra clase de realizaciones, la invención es una APU que incluye una memoria de búfer (búfer) que almacena (p. ej., de manera no transitoria) al menos una trama de un flujo de bits de audio codificado que ha sido generado por cualquier realización del método inventivo. Los ejemplos de APU incluyen, pero no están limitados a, codificadores (p. ej., transcodificadores), decodificadores, códecs, sistemas de preprocesamiento (preprocesadores), sistemas de postprocesamiento (postprocesadores), sistemas de procesamiento de flujo de bits de audio y combinaciones de tales elementos.

En otra clase de realizaciones, la invención es una unidad de procesamiento de audio (APU) configurada para generar un flujo 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 incluyen metadatos de estado de procesamiento de sonoridad (LPSM) y, opcionalmente, también metadatos de límite de programa. Típicamente, al menos uno de tales segmentos de metadatos en una trama del flujo de bits, incluye al menos un segmento de LPSM indicativo de si se ha realizado un primer tipo de procesamiento de sonoridad sobre los datos de audio de la trama (es decir, 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 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 las realizaciones típicas de esta clase, cada uno de los segmentos de metadatos tiene un formato preferido que se describirá en el presente documento.

En algunas realizaciones, cada uno de los segmentos de metadatos del flujo de bits codificado (un flujo de bits AC-3 o un flujo 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 deshecho del segmento de campo de omisión de una trama del flujo de bits (p. ej., un segmento de bits residuales W del tipo mostrado en la Fig. 4 o en la Fig. 7). En otras realizaciones, cada uno de los segmentos de metadatos del flujo de bits codificado (un flujo de bits AC-3 o un flujo 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 flujo de bits en el campo "addbsi" del segmento de Información de Flujo de Bits ("BSI") de una trama del flujo de bits, o en un campo de auxdata (p. ej., un segmento AUX del tipo mostrado en la Fig. 4 o en la Fig. 7) al final de una trama del flujo de bits. Cada uno de los segmentos de metadatos que incluye LPSM, puede tener el formato especificado en el presente documento con referencia a las Tablas 1 y 2 de abajo (es decir, incluye los elementos centrales especificados en la Tabla 1 o una variación al respecto, seguidos por la ID de carga útil (identificando los metadatos como LPSM) y valores de tamaño de carga útil, seguidos por la carga útil (datos de LPSM que tienen el formato indicado en la Tabla 2, o el formato como se indica en una variación de la Tabla 2 descrita en el presente documento). 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 de 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 los pasos de codificación de datos de audio para generar un flujo de bits de audio codificado AC-3 o E-AC-3, que incluye incluir en un segmento de metadatos (de al menos una trama del flujo de bits) LPSM y metadatos de límite de programa y, opcionalmente, también otros metadatos para el programa de audio al que pertenece la trama. En algunas realizaciones, cada uno de tales segmentos de metadatos se incluye en un campo de addbsi de la trama, o en un campo de auxdata de la trama. En otras realizaciones, cada uno de tales segmentos de metadatos se incluye en un segmento de bits residuales de la trama. En algunas realizaciones, cada uno de los segmentos de metadatos que contiene LPSM y metadatos de límite de programa, contiene una cabecera central (y, opcionalmente, también elementos centrales adicionales) y después de la cabecera central (o la cabecera central y otros elementos centrales) un segmento de carga útil de LPSM (o contenedor) que tiene el siguiente formato:

una cabecera, que típicamente incluye al menos un valor de identificación (p. ej., la versión de formato de LPSM, la longitud, el período, el recuento y los valores de asociación del subflujo de bits, como se indica en la Tabla 2 que se expone en el presente documento), y

después de la cabecera, los LPSM y los metadatos de límite de programa. Los metadatos de límite de programa pueden incluir un recuento de tramas de límite de programa y un valor de código (p. ej., un valor "existe\_desplazamiento") indicativo de si la trama incluye solamente un recuento de tramas de límite de programa o tanto un recuento de tramas de límite de programa y 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 correspondientes datos de audio indican diálogo o no indican diálogo (p. ej., qué canales de los correspondientes datos de audio indican diálogo). El (los) valor(es) de

indicación de diálogo puede(n) indicar si el diálogo está presente en cualquier combinación de, o todos, los canales de los correspondientes datos de audio;

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

5 al menos un valor de procesamiento de sonoridad que indica al menos un tipo de procesamiento de sonoridad que se ha realizado sobre los correspondientes datos de audio; y

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

10 En otras realizaciones, el flujo de bits codificado es un flujo de bits que no es un flujo de bits AC-3 o un flujo de bits E-AC-3, y cada uno de los segmentos de metadatos que incluye LPSM (y, opcionalmente, también metadatos de límite de programa) está incluido en un segmento (o campo o ranura) del flujo de bits reservado para el almacenamiento de datos adicionales. Cada uno de los segmentos de metadatos que incluye LPSM pueden tener un formato similar o idéntico al especificado en el presente documento con referencia a las Tablas 1 y 2 de abajo (es decir, incluye elementos centrales similares o idénticos a los especificados en la Tabla 1, seguidos por la ID de carga útil (identificando los metadatos como LPSM) y los valores de tamaño de carga útil, seguidos de la carga útil (datos de LPSM que tienen un formato similar o idéntico al formato indicado en la Tabla 2 o una variación de la Tabla 2 descrita en el presente documento).

20 En algunas realizaciones, el flujo de bits codificado comprende una secuencia de tramas, cada una de las tramas incluye un segmento de Información de Flujo de Bits ("BSI") que incluye un campo "addbsi" (a veces denominado segmento o ranura) y un campo o ranura auxdata (p. ej., el flujo de bits codificado es un flujo de bits AC-3 o un flujo de bits E-AC-3) y comprende segmentos de datos de audio (p. ej., los segmentos AB0-AB5 de la trama mostrada en la Fig. 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, opcionalmente, también metadatos de límite de programa. Los LPSM están presentes en el flujo de bits en el siguiente formato. Cada uno de los segmentos de metadatos que incluye LPSM está incluido en un campo "addbsi" del segmento BSI de una trama del flujo de bits, o en un campo de auxdata de una trama del flujo de bits, o en un segmento de bits residuales de una trama del flujo de bits. Cada uno de los segmentos de metadatos que incluye LPSM incluye un segmento de carga útil de LPSM (o contenedor) que tiene el siguiente formato:

30 una cabecera (que típicamente incluye al menos un valor de identificación, p. ej., la versión de formato de LPSM, la longitud, el período, el recuento y los valores de asociación del subflujo de datos indicados en la Tabla 2 de abajo); y

35 después de la cabecera, los LPSM y, opcionalmente, también los metadatos de límite de programa. Los metadatos de límite de programa pueden incluir un recuento de tramas de límite de programa y un valor de código (p. ej., un valor "existe\_desplazamiento") indicativo de si la trama incluye solamente un recuento de tramas de límite de programa o un recuento de tramas de límite de programa y un valor de desplazamiento), y (en algunos casos) un valor de desplazamiento. Los LPSM pueden incluir:

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

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 correspondientes datos de audio cumplen con un conjunto indicado de regulaciones de sonoridad;

45 al menos un valor de procesamiento de sonoridad (p. ej., uno o más de los parámetros "Indicador de Corrección de Sonoridad de Diálogo sincronizado", "Tipo de Corrección de Sonoridad" de la Tabla 2) que indican al menos un tipo de procesamiento de sonoridad que se ha realizado sobre los correspondientes datos de audio; y

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

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

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

una cabecera central (que típicamente incluye una palabra de sincronización que identifica el inicio del segmento de metadatos, seguido de valores de identificación, p. ej., la versión del elemento central, la longitud y el período, el recuento de elementos extendidos y los valores de asociación de subflujo de bits indicados en la Tabla 1 de abajo); y

10 después de la cabecera central, al menos un valor de protección (p. ej., unos valores de resumen de HMAC y valores de huella digital de audio, donde el resumen de HMAC puede ser un resumen de HMAC de 256 bits (utilizando el algoritmo SHA-2) calculado sobre los datos de audio, el elemento central y todos los elementos expandidos de una trama completa, como se indica en la Tabla 1) útiles para al menos uno de descifrado, autenticación o validación de al menos uno de los metadatos de estado de procesamiento de sonoridad o los correspondientes datos de audio); y

15 también después de la cabecera central, si el segmento de metadatos incluye LPSM, la identificación ("ID") de carga útil de LPSM y los 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 anteriormente) sigue a la ID de carga útil de LPSM y a los valores de carga útil de LPSM.

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

25 una estructura de alto nivel, que incluye un indicador que indica si el campo de auxdata (o addbsi) incluye metadatos, al menos un valor de ID que indica qué tipo(s) de metadatos está(n) presente(s), y generalmente también un valor que indica cuántos bits de metadatos (p. ej., de cada uno de los tipos) están presentes (si los metadatos están presentes). Un tipo de metadatos que podría estar presente son los LSPM, otro tipo de metadatos que podría estar presente son los metadatos de límite de programa, y otro tipo de metadatos que podría estar presente son los metadatos de investigación de medios;

30 una estructura de nivel intermedio, que comprende un elemento central para cada uno de los tipos de metadatos identificado (p. ej., cabecera central, valores de protección e ID de carga útil y valores de tamaño de carga útil, p. ej., del tipo mencionado anteriormente, para cada uno de los tipos de metadatos identificado); y

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

35 Los valores de datos en una estructura de tres niveles pueden anidar se. Por ejemplo, el (los) valor(es) de protección para una carga útil de LPSM y/u otra carga útil de metadatos identificados por un elemento central, pueden incluirse después de cada una de las cargas útiles identificada por el elemento central (y por lo tanto después de la cabecera central del elemento central). En un ejemplo, una cabecera central podría identificar una carga útil de LPSM y otra carga útil de metadatos, la ID de carga útil y los valores de tamaño de carga útil para la primera carga útil (p. ej., la carga útil de LPSM) podrían seguir a la cabecera principal, la primera carga útil en sí podría seguir a la ID y a los valores de tamaño, la ID de carga útil y el valor de tamaño de carga útil para la segunda carga útil podrían seguir a la primera carga útil, la segunda carga útil podría seguir a estas ID y valores de tamaño, y el (los) valor(es) de protección para una o ambas cargas útiles (o para los valores de elemento central y una o ambas de las cargas útiles) podrían seguir a la última carga útil.

45 En algunas realizaciones, el elemento central de un segmento de metadatos en un campo de auxdata (o campo "addbsi" o segmento de bits residuales) de una trama comprende una cabecera central (que típicamente incluye valores de identificación, p. ej., versión del elemento central) y después de la cabecera central: valores indicativos de si se incluyen datos de huellas digitales para metadatos del segmento de metadatos, los valores indicativos de si existen datos externos (relacionados con datos de audio correspondientes a los metadatos del segmento de metadatos), la ID de carga útil y los valores de tamaño de carga útil para cada uno de los tipos de metadatos (p. ej., LPSM y/o metadatos de un tipo distinto de LPSM) identificados por el elemento central, y los valores de protección



para al menos un tipo de metadatos identificado por el elemento central. La(s) carga(s) útil(es) de metadatos del segmento de metadatos sigue(n) a la cabecera central, y está(n) (en algunos casos) anidada(s) dentro de los valores del elemento central.

5 En otro formato preferido, el flujo de bits codificado es un flujo de bits Dolby E, y cada uno de los segmentos de metadatos que incluye LPSM (y, opcionalmente, también metadatos de límite de programa) está incluido en las primeras N ubicaciones de muestra del intervalo de banda de guarda de Dolby E.

10 En otra clase de realizaciones, la invención es una APU (p. ej., un decodificador) acoplada y configurada para recibir un flujo 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 los 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, opcionalmente, también metadatos de límite de programa, y para extraer los LPSM del flujo de bits para generar datos de audio decodificados en respuesta a los datos de audio y para realizar al menos una operación de procesamiento de sonoridad adaptativo sobre los datos de audio utilizando los LPSM. Algunas realizaciones en esta clase también incluyen un postprocesador acoplado a la APU, en donde el postprocesador está acoplado y configurado para realizar al menos una operación de procesamiento de sonoridad adaptativo sobre los datos de audio utilizando los LPSM.

20 En otra clase de realizaciones, la invención es una unidad de procesamiento de audio (APU) que incluye una memoria de búfer (búfer) y un subsistema de procesamiento acoplado al búfer, en donde la APU está acoplada para recibir un flujo 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, opcionalmente, también metadatos de límite de programa, el búfer almacena (p. ej., de manera no transitoria) al menos una trama del flujo de bits de audio codificado, y el subsistema de procesamiento está configurado para extraer los LPSM del flujo de bits y para realizar al menos una operación de procesamiento de sonoridad adaptativo sobre los datos de audio utilizando los LPSM. En realizaciones típicas de esta clase, la APU es una de un codificador, un decodificador y un postprocesador.

25 En algunas implementaciones del método inventivo, el flujo de bits de audio generado es uno de un flujo de bits codificado AC-3, un flujo de bits E-AC-3, o un flujo de bits Dolby E, incluyendo 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 flujo de bits de audio generado es un flujo de bits codificado de otro tipo.

30 Los aspectos de la invención incluyen un sistema o dispositivo configurado (p. ej., programado) para realizar cualquier realización del método inventivo, y un medio legible por ordenador (p. ej., un disco) que almacena código (p. ej., de manera no transitoria) para implementar cualquier realización del método inventivo o sus pasos. Por ejemplo, el sistema inventivo puede ser o incluir un procesador de propósito general programable, un procesador de señal digital o un microprocesador, programado con software o firmware y/o configurado de otra manera para realizar cualquiera de una variedad de operaciones sobre datos, incluyendo una realización del método inventivo o sus pasos. Tal procesador de propósito general puede ser o incluir un sistema informático que incluye un dispositivo de entrada, una memoria y una circuitería de procesamiento programada (y/o configurada de otro manera) para realizar una realización del método inventivo (o sus pasos) en respuesta a los datos declarados a ello.

#### Breve Descripción de los Dibujos

La FIG. 1 es un diagrama de bloques de una realización de un sistema que puede configurarse para realizar una realización del método inventivo.

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

La FIG. 3 es un diagrama de bloques de un decodificador que es una realización de la unidad de procesamiento de audio inventiva, y un postprocesador acoplado a la misma, que es otra realización de la unidad de procesamiento de audio inventiva.

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

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

La FIG. 6 es un diagrama del segmento de Información de Flujo de Bits (BSI) de una trama AC-3, que incluye los segmentos en los que está dividida.

5 La FIG. 7 es un diagrama de una trama E-AC-3, que incluye los segmentos en los que está dividida.

La FIG. 8 es un diagrama de tramas de un flujo de bits de audio codificado que incluye metadatos de límite de programa, cuyo formato está de acuerdo con una realización de la invención.

La FIG. 9 es un diagrama de otras tramas del flujo de bits de audio codificado de la Fig. 9. Algunas de estas tramas incluyen metadatos de límite de programa que tienen un formato de acuerdo con una realización de la invención.

10 La FIG. 10 es un diagrama de dos flujos de bits de audio codificados: un flujo de bits (IEB) en el que un límite de programa (etiquetado "Límite") está alineado con una transición entre dos tramas del flujo de bits, y otro flujo de bits (TB) en el que un límite de programa (etiquetado "Límite Real") se desplaza por 512 muestras de una transición entre dos tramas del flujo de bits.

15 La FIG. 11 es un conjunto de diagramas que muestran cuatro flujos de bits de audio codificados. El flujo de bits en la parte superior de la FIG. 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 flujo 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 los metadatos de límite de programa; el tercer flujo de bits (etiquetado "Escenario 3") es indicativo de un primer programa de audio (P1) truncado que incluye metadatos de límite de programa, y que ha sido empalmado con un segundo programa de audio (P2) completo que incluye metadatos de límite de programa; y el cuarto flujo de bits (etiquetado "Escenario 4") es indicativo de un primer programa de audio (P1) truncado que incluye metadatos de límite de programa y un segundo programa de audio (P2) truncado que incluye metadatos de límite de programa y que ha sido empalmado con una porción del primer programa de audio.

25 Notación y Nomenclatura

A lo largo de esta divulgación, incluyendo en las reivindicaciones, la expresión realizar una operación "sobre" una señal o unos datos (p. ej., filtrado, escalado, transformación, o la aplicación de ganancia a la señal o los datos) se utiliza en un sentido amplio para referirse a realizar la operación directamente sobre la señal o los datos, o sobre una versión procesada de la señal o los datos (p. ej., sobre una versión de la señal que se ha sometido a un filtrado preliminar o preprocesamiento antes de la ejecución de la operación).

A lo largo de esta divulgación, incluyendo en las reivindicaciones, la expresión "sistema" se utiliza en un sentido amplio para referirse a un dispositivo, sistema o subsistema. Por ejemplo, un subsistema que implementa un decodificador se puede denominar un sistema decodificador, y un sistema que incluye un tal subsistema (p. ej., un sistema que genera X señales de salida en respuesta a múltiples entradas, en las cuales el subsistema genera M de las entradas y las otras X - M entradas se reciben de una fuente externa) también se puede denominar un sistema decodificador.

A lo largo de esta divulgación, incluyendo en las reivindicaciones, el término "procesador" se utiliza en un sentido amplio para referirse a un sistema o dispositivo programable o de otro modo configurable (p. ej., con software o firmware) para realizar operaciones sobre los datos (p. ej., audio, o video u otros datos de imagen). Ejemplos de procesadores incluyen una matriz de puertas programables (u otro circuito integrado o conjunto de chips configurable), un procesador de señal digital programado y/o configurado de otra manera para realizar el procesamiento canalizado sobre datos de audio o de otro sonido, un procesador de propósito general u ordenador programable, y un chip microprocesador o conjunto de chips programable.

A lo largo de esta divulgación, incluyendo en las reivindicaciones, las expresiones "procesador de audio" y "unidad de procesamiento de audio" se utilizan indistintamente y en un sentido amplio, para referirse a un sistema configurado para procesar datos de audio. Los ejemplos de unidades de procesamiento de audio incluyen, pero no se limitan a codificadores (p. ej., transcodificadores), decodificadores, códecs, sistemas de preprocesamiento, sistemas de postprocesamiento y sistemas de procesamiento de flujo de bits (a veces denominados herramientas de procesamiento de flujo de bits).

5 A lo largo de esta divulgación, incluyendo en 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 correspondientes datos de audio (el contenido de audio de un flujo de datos de audio que también incluye metadatos de estado de procesamiento). Los metadatos de estado de procesamiento están asociados con los datos de audio, indican el estado de procesamiento de sonoridad de los correspondientes datos de audio (p. ej., qué tipo(s) de procesamiento ya se ha(n) realizado sobre los datos de audio), y, típicamente, también indica al menos una particularidad o característica de los datos de audio. La asociación de los metadatos de estado de procesamiento con los datos de audio está sincronizada en el tiempo. Por lo tanto, los metadatos de estado de procesamiento presentes (recibidos o actualizados más recientemente) indican que los correspondientes datos de audio comprenden contemporáneamente los resultados del (de los) tipo(s) indicado(s) de procesamiento de datos de audio. En algunos casos, los metadatos de estado de procesamiento pueden incluir el historial de procesamiento y/o algunos o todos los parámetros que se utilizan y/o derivan de los tipos de procesamiento indicados. Además, los metadatos de estado de procesamiento pueden incluir al menos una particularidad o característica de los correspondientes datos de audio, que se ha calculado o extraído de los datos de audio. Los metadatos de estado de procesamiento también pueden incluir otros metadatos que no están relacionados con o derivados de ningún procesamiento de los correspondientes datos de audio. Por ejemplo, datos de terceros, información de seguimiento, identificadores, información de propiedad o estándar, datos de anotación de usuario, datos de preferencia de usuario, etc. pueden ser añadidos por una unidad de procesamiento de audio particular para pasar a otras unidades de procesamiento de audio.

20 A lo largo de esta divulgación, incluyendo en las reivindicaciones, la expresión “metadatos de estado de procesamiento de sonoridad” (o “LPSM”) se refiere a metadatos de estado de procesamiento indicativos del estado de procesamiento de sonoridad de los correspondientes datos de audio (p. ej., qué tipo(s) de procesamiento de sonoridad se ha(n) realizado sobre los datos de audio) y, típicamente, también al menos una particularidad o característica (p. ej., sonoridad) de los correspondientes datos de audio. Los metadatos de estado de procesamiento de sonoridad pueden incluir datos (p. ej., otros metadatos) que no son (es decir, cuando se consideran separados) metadatos de estado de procesamiento de sonoridad.

A lo largo de esta divulgación, incluyendo en las reivindicaciones, la expresión “canal” (o “canal de audio”) se refiere a una señal de audio monofónica.

30 A lo largo de esta divulgación, incluyendo en las reivindicaciones, la expresión “programa de audio” se refiere a un conjunto de uno o más canales de audio y, opcionalmente, también los metadatos asociados (p. ej., metadatos que describen una presentación de audio espacial deseada, y/o LPSM, y/o metadatos de límite de programa).

35 A lo largo de esta divulgación, incluyendo en las reivindicaciones, la expresión “metadatos de límite de programa” se refiere a metadatos de un flujo de bits de audio codificado, donde el flujo 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 flujo de bits de al menos un límite (comienzo y/o final) de al menos uno de dichos programas de audio. Por ejemplo, los metadatos de límite de programa (de un flujo de bits de audio codificado indicativo de un programa de audio) pueden incluir metadatos indicativos de la ubicación (p. ej., el inicio de la “N”-ésima trama del flujo de bits o la ubicación de la “M”-ésima muestra de la “N”-ésima trama del flujo de bits) del comienzo del programa, y metadatos adicionales indicativos de la ubicación (p. ej., el inicio de la “J”-ésima trama del flujo de bits, o la ubicación de la “K”-ésima muestra de la “J”-ésima trama del flujo de bits) del final del programa.

A lo largo de esta divulgación, incluyendo en las reivindicaciones, el término “se acopla” o “acoplado” se utiliza para indicar bien una conexión directa o indirecta. Por lo tanto, si un primer dispositivo se acopla a un segundo dispositivo, esa conexión puede ser a través de una conexión directa, o a través de una conexión indirecta a través de otros dispositivos y conexiones.

#### 45 Descripción Detallada de las Realizaciones de la Invención

50 De acuerdo con realizaciones típicas de la invención, una carga útil de metadatos de sonoridad de programa, referidos como metadatos de estado de procesamiento de sonoridad (“LPSM”) y, opcionalmente, también los metadatos de límite de programa están incrustados en uno o más campos reservados (o ranuras) de segmentos de metadatos de un flujo de bits de audio que también incluye datos de audio en otros segmentos (segmentos de datos de audio). Típicamente, al menos un segmento de cada una de las tramas del flujo de bits incluye LPSM, y al menos otro segmento de la trama incluye datos de audio correspondientes (es decir, datos de audio cuyo estado de procesamiento de sonoridad y sonoridad están indicados por los LPSM). En algunas realizaciones, el volumen de datos de los LPSM puede ser suficientemente pequeño para ser transportado sin afectar a la tasa de bits asignada para transportar los datos de audio.

La comunicación de 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 trabajar en tándem entre sí a lo largo de la cadena de procesamiento (o ciclo de vida del contenido). Sin la inclusión de metadatos de estado de procesamiento de sonoridad en un flujo de bits de audio, pueden producirse graves problemas de procesamiento de medios tales como calidad, degradaciones de nivel y espaciales, por ejemplo, cuando se utilizan dos o más códecs de audio en la cadena y la nivelación de volumen en modo común se aplica más de una vez durante el viaje del flujo de bits a un dispositivo que consume medios (o un punto de renderización del contenido de audio del flujo de bits).

La FIG. 1 es un diagrama de bloques de una cadena de procesamiento de audio (un sistema de procesamiento de datos de audio) a modo de ejemplo, en la que uno o más de los elementos del sistema pueden estar configurados de acuerdo con una realización de la presente invención. El sistema incluye los siguientes elementos acoplados entre sí como se muestra: una unidad de preproceso, un codificador, una unidad de análisis de señal y de corrección de metadatos, un transcodificador, un decodificador y una unidad de preproceso. En las variaciones del sistema mostrado, se omiten uno o más de los elementos, o se incluyen unidades adicionales de procesamiento de datos de audio.

En algunas implementaciones, la unidad de preproceso de la FIG. 1 está configurada para aceptar muestras de PCM (dominio del tiempo) que comprenden contenido de audio como entrada, y para emitir muestras de PCM procesadas. El codificador puede estar configurado para aceptar las muestras de PCM como entrada y para emitir un flujo de bits de audio codificado (p. ej., comprimido) indicativo del contenido de audio. Los datos del flujo de bits que son indicativos del contenido de audio, en el presente documento a veces se denominan "datos de audio". Si el codificador está configurado de acuerdo con una realización típica de la presente invención, la salida de flujo de bits de audio del codificador incluye metadatos de estado de procesamiento de sonoridad (y típicamente también otros metadatos, que incluyen opcionalmente metadatos de límite de programa) así como datos de audio.

La unidad de análisis de señal y de corrección de metadatos de la Fig. 1 puede aceptar uno o más flujos de bits de audio codificados como entrada y determinar (p. ej., validar) si los metadatos de estado de procesamiento en cada uno de los flujos de bits de audio codificado son correctos, realizando el análisis de señal (p. ej., utilizando metadatos de límite de programa en un flujo de bits de audio codificado). Si la unidad de análisis de señal y de corrección de metadatos encuentra que los metadatos incluidos no son válidos, típicamente reemplaza el (los) valor(es) incorrecto(s) con el (los) valor(es) correcto(s) obtenido(s) del análisis de señal. Por lo tanto, cada una de las salidas de flujo de bits de audio codificado de la unidad de análisis de señal y de corrección de metadatos puede incluir metadatos de estado de procesamiento corregidos (o no corregidos) así como datos de audio codificados.

El transcodificador de la Fig. 1 puede aceptar flujos de bits de audio codificados como entrada, y emitir flujos de bits de audio modificados (p. ej., codificados de manera diferente) en respuesta (p. ej., decodificando un flujo de entrada y recodificando el flujo decodificado en un formato de codificación diferente). Si el transcodificador está configurado de acuerdo con una realización típica de la presente invención, la salida de flujo de bits de audio desde el transcodificador incluye metadatos de estado de procesamiento de sonoridad (y típicamente también otros metadatos) así como datos de audio codificados. Los metadatos pueden haberse incluido en el flujo de bits.

El decodificador de la Fig. 1 puede aceptar flujos de bits de audio codificados (p. ej., comprimidos) como entrada y emitir (en respuesta) flujos de muestras de audio PCM decodificadas. Si el decodificador está configurado de acuerdo con una realización típica de la presente invención, la salida del decodificador en funcionamiento típico es o incluye cualquiera de los siguientes:

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

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

un flujo de muestras de audio, sin un correspondiente flujo 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 decodificador puede extraer metadatos de estado de procesamiento de sonoridad (y/o otros metadatos) del flujo de bits codificado de entrada y realizar al menos una operación sobre los metadatos extraídos (p. ej., validación), aunque no genere los metadatos extraídos o los bits de control determinados a partir de ellos.

Configurando la unidad de postproceso de la Fig. 1 de acuerdo con una realización típica de la presente invención, la unidad de postproceso está configurada para aceptar un flujo de muestras de audio PCM decodificadas, y para realizar el postproceso en las mismas (p. ej., nivelación de volumen del contenido de audio) utilizando metadatos de estado de procesamiento de sonoridad (y típicamente también otros metadatos) recibidos con las muestras, o bits de control (determinados por el decodificador a partir de metadatos de estado de procesamiento de sonoridad y típicamente también otros metadatos) recibidos con las muestras. La unidad de postproceso también está configurada típicamente para renderizar el contenido de audio postprocesado para su reproducción por uno o más altavoces.

Las realizaciones típicas de la presente invención proporcionan una cadena de procesamiento de audio mejorada en la que las unidades de procesamiento de audio (p. ej., codificadores, decodificadores, transcodificadores y unidades de pre y postproceso) adaptan su respectivo procesamiento para ser aplicado a datos de audio de acuerdo con un estado contemporáneo de los datos de medios, según lo indicado por los metadatos de estado de procesamiento de sonoridad respectivamente recibidos por las unidades de procesamiento de audio.

La entrada de datos de audio a cualquier unidad de procesamiento de audio del sistema de la Fig. 1 (p. ej., el codificador o transcodificador de la Fig. 1) puede incluir metadatos de estado de procesamiento de sonoridad (y opcionalmente 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 Fig. 1 (u otra fuente, no mostrada en la Fig. 1) de acuerdo con una realización de la presente invención. La unidad de procesamiento que recibe el audio de entrada (con metadatos) puede estar configurada para realizar al menos una operación sobre los metadatos (p. ej., validación) o en respuesta a los metadatos (p. ej., procesamiento adaptativo del audio de entrada) y típicamente 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) está configurada para realizar un procesamiento adaptativo de datos de audio en base al estado de los datos de audio según lo indicado por los metadatos de estado de procesamiento de sonoridad correspondientes a los datos de audio. En algunas realizaciones, el procesamiento adaptativo es (o incluye) procesamiento de sonoridad (si los metadatos indican que el procesamiento de sonoridad, o procesamiento similar al mismo, aún no se ha realizado sobre los datos de audio, pero no es (y no incluye) procesamiento de sonoridad (si los metadatos indican que tal procesamiento de sonoridad, o procesamiento similar al mismo, ya se ha realizado sobre los datos de audio). En algunas realizaciones, el procesamiento adaptativo es o incluye la validación de metadatos (p. ej., realizado en una subunidad de validación de metadatos) para garantizar que la unidad de procesamiento de audio realice otro procesamiento adaptativo de los datos de audio en base al estado de los datos de audio según lo indicado 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 con (p. ej., incluidos en un flujo de bits con) los datos de audio. Por ejemplo, si los metadatos se validan para ser fiables, los resultados de un tipo de procesamiento de audio previamente realizado, pueden reutilizarse y se puede evitar la nueva ejecución del mismo tipo de procesamiento de audio. Por otro lado, si se encuentra que los metadatos han sido alterados (o de otra manera no fiables), entonces el tipo de procesamiento de medios supuestamente realizado previamente (según lo indicado por los metadatos no fiables) puede repetirse por la unidad de procesamiento de audio y/o otro procesamiento puede ser realizado por la unidad de procesamiento de audio sobre los metadatos y/o los datos de audio. La unidad de procesamiento de audio también puede estar configurada para señalar a otras unidades de procesamiento de audio de flujo descendente en una cadena de procesamiento de medios mejorada, que los metadatos de estado de procesamiento de sonoridad (p. ej., presentes en un flujo de bits de medios) son válidos si la unidad determina que los metadatos de estado de procesamiento son válidos (p. ej., en base a una coincidencia de un valor criptográfico extraído y un valor criptográfico de referencia).

La FIG. 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 puede 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 tramas 110, analizador 111, decodificador 101, validador de estado de audio 102, etapa de procesamiento de sonoridad 103, etapa de selección de flujo de audio 104, codificador 105, etapa relleno/formatador 107, etapa de generación de metadatos 106, subsistema de medición de sonoridad de diálogo 108 y búfer de tramas 109, conectados como se muestra. Típicamente, también el codificador 100 incluye otros elementos de procesamiento (no mostrados).

El codificador 100 (que es un transcodificador) está configurado para convertir un flujo de bits de audio de entrada (el cual, por ejemplo, puede ser uno de un flujo de bits AC-3, un flujo de bits E-AC-3 o un flujo de bits Dolby E) a un flujo de bits de audio codificado (el cual, por ejemplo, puede ser otro de un flujo de bits AC-3, un flujo de bits E-AC-3 o un flujo de bits Dolby E) que incluye la realización de procesamiento de sonoridad adaptativo y automatizado, utilizando metadatos de estado de procesamiento de sonoridad incluidos en el flujo de bits de entrada. Por ejemplo,

el codificador 100 puede estar configurado para convertir un flujo de bits Dolby E de entrada (un formato típicamente utilizado en instalaciones de producción y transmisión pero no en dispositivos de consumo que reciben programas de audio que han sido transmitidos a los mismos) a un flujo de bits de audio de salida codificado (adecuado para transmisión a dispositivos de consumo) en formato AC-3 o E-AC-3.

5 El sistema de la FIG. 2 también incluye el subsistema de entrega de audio codificado 150 (el cual almacena y/o entrega la salida de flujos de bits codificados del codificador 100) y el decodificador 152. El subsistema 150 puede almacenar una salida de flujo de bits de audio codificado del codificador 100 (p. ej., en forma de DVD o Blu-ray Disc), o transmitido por el subsistema 150 (el cual puede implementar un enlace de transmisión o red), o puede ser almacenado y transmitido por el subsistema 150. El decodificador 152 está configurado para decodificar un flujo de bits de audio codificado (generado por el codificador 100) que recibe a través del subsistema 150, que incluye la extracción de metadatos de estado de procesamiento de sonoridad (LPSM) de cada una de las tramas del flujo de bits (y, opcionalmente, también extrayendo metadatos de límite de programa del flujo de bits), y generando datos de audio decodificados. Típicamente el decodificador 152 está configurado para realizar un procesamiento de sonoridad adaptativo sobre los datos de audio decodificados utilizando los LPSM (y, opcionalmente, también metadatos de límite de programa), y/o reenviar los datos de audio decodificados y los LPSM a un postprocesador configurado para realizar procesamiento de sonoridad adaptativo sobre los datos de audio decodificados utilizando los LPSM (y, opcionalmente, también metadatos de límite de programa). Típicamente, el decodificador 152 incluye un búfer que almacena (p. ej., de manera no transitoria) el flujo de bits de audio codificado recibido desde el subsistema 150.

20 Varias implementaciones del codificador 100 y del decodificador 152 están configuradas para realizar diferentes realizaciones del método inventivo. El búfer de tramas 110 es una memoria de búfer acoplada para recibir un flujo de bits de audio de entrada codificado. En funcionamiento, el búfer 110 almacena (p. ej., de manera no transitoria) al menos una trama del flujo de bits de audio codificado, y se declara una secuencia de las tramas del flujo de bits de audio codificado desde el búfer 110 al analizador 111.

25 El analizador 111 está acoplado y configurado para extraer metadatos de estado de procesamiento de sonoridad (LPSM) y, opcionalmente, también metadatos de límite de programa (y/u otros metadatos) de cada una de las tramas del audio de entrada codificado en las que tales metadatos están incluidos, para declarar al menos los LPSM (y, opcionalmente, también metadatos de límite de programa y/u otros metadatos) al validador de estado de audio 102, a la etapa de procesamiento de sonoridad 103, a la etapa 106 y al subsistema 108, para extraer datos de audio del audio de entrada codificado y para declarar los datos de audio al decodificador 101. El decodificador 101 del codificador 100 está configurado para decodificar los datos de audio para generar datos de audio decodificados y para confirmar los datos de audio decodificados a la etapa de procesamiento de sonoridad 103, a la etapa de selección de flujo de audio 104, al subsistema 108 y típicamente también al validador de estado 102.

35 El validador de estado 102 está configurado para autenticar y validar los LPSM (y opcionalmente otros metadatos) declarados al mismo. En algunas realizaciones, los LPSM son (o están incluidos en) un bloque de datos que se ha incluido en el flujo de bits de entrada (p. ej., de acuerdo con una realización de la presente invención). El bloque puede comprender un hash criptográfico (un código de autenticación de mensajes en clave-hash o "HMAC") para procesar los LPSM (y, opcionalmente, también otros metadatos) y/o los datos de audio subyacentes (proporcionados desde el decodificador 101 al validador 102). El bloque de datos puede firmarse digitalmente en estas realizaciones, de modo que una unidad de procesamiento de audio de flujo descendente puede autenticar y validar con relativa facilidad los metadatos de estado de procesamiento.

Por ejemplo, el HMAC se utiliza para generar un resumen, y el (los) valor(es) de protección incluido(s) en el flujo de bits inventivo pueden incluir el resumen. El resumen se puede generar de la siguiente manera para una trama AC-3:

45 1. Después de que los datos de AC-3 y los LPSM están codificados, los bytes de datos de trama (frame\_data#1 y frame\_data#2 concatenados) y los datos de LPSM se utilizan como entrada para la función hash HMAC. Otros datos, que pueden estar presentes dentro de un campo de auxdata, no se tienen en cuenta para calcular el resumen. Tales otros datos pueden ser bytes que no pertenecen a los datos AC-3 ni a los datos de LPSM. Los bits de protección incluidos en LPSM pueden no ser considerados para calcular el resumen de HMAC.

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

50 3. El último paso de la generación de la trama AC-3 completa es el cálculo de la verificación de CRC. Esto se escribe al final de la trama y se tienen en cuenta todos los datos que pertenecen a esta trama, incluidos los bits LPSM.

Otros métodos criptográficos que incluyen, pero no se limitan a cualquiera de uno o más métodos criptográficos no HMAC, pueden utilizarse para la validación de LPSM (p. ej., en el validador 102) para garantizar la transmisión segura y la recepción de los LPSM y/o de los datos de audio subyacentes. Por ejemplo, la validación (utilizando tal método criptográfico) puede realizarse en cada una de las unidades de procesamiento de audio que recibe una realización del flujo de bits de audio inventivo para determinar si los metadatos de estado de procesamiento de sonoridad y los correspondientes datos de audio incluidos en el flujo de bits, han sido sometidos a (y/o han resultado de) un procesamiento de sonoridad específico (como lo indican los metadatos) y no se han modificado después de la ejecución de tal procesamiento de sonoridad específico.

El validador de estado 102 declara los datos de control a la etapa de selección de flujo de audio 104, al generador de metadatos 106 y al subsistema de medida de sonoridad de diálogo 108, para indicar los resultados de la operación de validación. En respuesta a los datos de control, la etapa 104 puede seleccionar (y pasar al codificador 105) bien:

la salida procesada adaptativamente de la etapa de procesamiento de sonoridad 103 (p. ej., cuando los LPSM indican que la salida de datos de audio del decodificador 101 no se ha sometido a un tipo específico de procesamiento de sonoridad, y los bits de control del validador 102 indican que los LPSM son válidos); o

la salida de datos de audio del decodificador 101 (p. ej., cuando los LPSM indican que la salida de datos de audio del decodificador 101 ya se ha sometido al tipo específico de procesamiento de sonoridad que se realizaría en la etapa 103, y los bits de control del validador 102 indican que los LPSM son válidos).

La etapa 103 del codificador 100 está configurada para realizar el procesamiento de sonoridad adaptativo sobre la salida de datos de audio decodificados del decodificador 101, en base a una o más características de los datos de audio indicadas por LPSM, extraídas por el decodificador 101. La etapa 103 puede ser un procesador de sonoridad en tiempo real de dominio de transformación adaptativo y de control de rango dinámico. La etapa 103 puede recibir entrada de usuario (p. ej., valores de sonoridad objetivo/rango dinámico de usuario o valores de dialnorm) u otra entrada de metadatos (p. ej., uno o más tipos de datos de terceros, información de seguimiento, identificadores, información patentada o estándar, datos de anotación de usuario, datos de preferencia de usuario, etc.) y/u otra entrada (p. ej., de un proceso de huella digital), y utilizar dicha entrada para procesar la salida de datos de audio decodificados desde el decodificador 101. La etapa 103 puede realizar el procesamiento de sonoridad adaptativo sobre datos de audio decodificados (salida del decodificador 101) indicativos de un único programa de audio (como lo indican 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 decodificados (salida del decodificador 101) indicativo de un programa de audio diferente según lo indicado por los metadatos de límite de programa extraídos por el analizador 111.

El subsistema de medición de sonoridad de diálogo 108 puede operar para determinar la sonoridad de segmentos del audio decodificado (del decodificador 101) que son indicativos de diálogo (u otra voz), p. ej., utilizando los LPSM (y/u otros metadatos) extraídos por el decodificador 101, cuando los bits de control del validador 102 indican que los LPSM son inválidos. La operación del subsistema de medición de sonoridad de diálogo 108 puede desactivarse cuando los LPSM indican la sonoridad previamente determinada de los segmentos de diálogo (u otra voz) del audio decodificado (del decodificador 101) cuando los bits de control del validador 102 indican que los LPSM son válidos. El subsistema 108 puede realizar una medición de sonoridad en datos de audio decodificados indicativos de un único programa de audio (según lo indicado por los metadatos de límite de programa extraídos por el analizador 111), y puede reestablecer la medición en respuesta a recibir datos de audio decodificados indicativos de un programa de audio diferente como se indica por tales 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., etapa 108 del codificador 100) se implementan para incluir (o para realizar las funciones de) tal herramienta para medir la sonoridad de diálogo media del contenido de audio de un flujo de bits de audio (p. ej., un flujo de bits AC-3 decodificado declarado a la etapa 108 desde el decodificador 101 del codificador 100).

Si la etapa 108 se implementa para medir la verdadera sonoridad de diálogo media de datos de audio, la medición puede incluir un paso de aislamiento de segmentos del contenido de audio que contienen voz predominantemente. Los segmentos de audio que predominantemente son voz, se procesan entonces de acuerdo con un algoritmo de medición de sonoridad. Para los datos de audio decodificados de un flujo de bits AC-3, este algoritmo puede ser una medida de sonoridad ponderada K estándar (de acuerdo con el estándar internacional UIT-R BS.1770). Alternativamente, se pueden utilizar otras medidas de sonoridad (p. ej., aquellas basadas en modelos psicoacústicos de sonoridad).

- 5 El aislamiento de segmentos de voz no es esencial para medir la sonoridad de diálogo media de datos de audio. Sin embargo, mejora la precisión de la medida y, por lo general, proporciona resultados más satisfactorios desde la perspectiva del oyente. Debido a 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.
- 10 El generador de metadatos 106 genera (y/o pasa a la etapa 107) metadatos para ser incluidos por la etapa 107 en el flujo de bits codificado a ser emitido desde el codificador 100. El generador de metadatos 106 puede pasar a la etapa 107 los LPSM (y, opcionalmente, también metadatos de límite de programa y/u otros metadatos) extraídos por el codificador 101 y/o el analizador 111 (p. ej., cuando los bits de control del validador 102 indican que los LPSM y/u otros metadatos son válidos) o generan un nuevo LPSM (y, opcionalmente, también metadatos de límite de programa y/u otros metadatos) y declaran los nuevos metadatos a la etapa 107 (p. ej., cuando los bits de control del validador 102 indican que los LPSM y/u otros metadatos extraídos por el decodificador 101 son inválidos o puede declarar a la etapa 107 una combinación de metadatos extraídos por el decodificador 101 y/o el 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 realizado por el subsistema 108, en los LPSM declara a la etapa 107 para la inclusión en el flujo de bits codificado a ser emitido desde el codificador 100.
- 15 El generador 106 de metadatos puede generar bits de protección (que pueden consistir en o incluir un código de autenticación de mensajes en clave-hash o "HMAC") útil para al menos uno de descryptación, autenticación o validación de los LPSM (y, opcionalmente, también otros metadatos) a ser incluidos en el flujo de bits codificado y/o los datos de audio subyacentes a ser incluidos en el flujo de bits codificado. El generador de metadatos 106 puede proporcionar tales bits de protección a la etapa 107 para la inclusión en el flujo de bits codificado.
- 20 En el funcionamiento típico, el subsistema de medida de sonoridad de diálogo 108 procesa la salida de datos de audio del decodificador 101 para generar en respuesta a la misma, valores de sonoridad (p. ej., valores de sonoridad de diálogo sincronizados y dessincronizados) 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 la inclusión (por el rellenador/formateador 107) en el flujo de bits codificado a ser enviado desde el codificador 100.
- 25 Además, opcional o alternativamente, los subsistemas de 106 y/o de 108 del codificador 100 pueden realizar un análisis adicional de los datos de audio para generar metadatos indicativos de al menos una característica de los datos de audio para la inclusión en el flujo de bits codificado a ser emitido desde la etapa 107.
- 30 El codificador 105 codifica (p. ej., realizando compresión sobre los mismos) la salida de datos de audio de la etapa de selección 104, y declara el audio codificado a la etapa 107 para la inclusión en el flujo de bits codificado a ser enviado desde la etapa 107.
- 35 La etapa 107 multiplexa el audio codificado del codificador 105 y los metadatos (incluyendo LPSM) del generador 106 para generar el flujo de bits codificado a ser emitido desde la etapa 107, preferiblemente de modo que el flujo de bits codificado tenga el formato según los especificado por una realización preferida del presente invención.
- El búfer de tramas 109 es una memoria de búfer que almacena (p. ej., de manera no transitoria) al menos una trama de la salida de flujo de bits codificado de audio de la etapa 107, y luego se declara una secuencia de las tramas del flujo de bits de audio codificado del buffer 109 como salida del codificador 100 al sistema de entrega 150.
- 40 Los LPSM generados por el generador de metadatos 106 e incluidos en el flujo de bits codificado por la etapa 107, es indicativo del estado de procesamiento de sonoridad de los correspondientes datos de audio (p. ej., qué tipo(s) de procesamiento de sonoridad se ha(n) realizado sobre los datos de audio) y sonoridad (p. ej., sonoridad de diálogo medido, sonoridad sincronizada y/o dessincronizada y/o rango dinámico) de los correspondientes datos de audio.
- 45 En el presente documento, "sincronización" de sonoridad y/o mediciones de nivel realizadas sobre los datos de audio, se refiere a un umbral de nivel o de sonoridad específico, donde el (los) valor(es) calculado(s) que supera(n) el umbral está(n) incluido(s) en la medición final (p. ej., ignorando valores de sonoridad a corto plazo por debajo de -60 dBFS en los valores medidos finales). La sincronización sobre un valor absoluto se refiere a un nivel o sonoridad fijo, mientras que la sincronización de un valor relativo se refiere a un valor que depende de un valor de medición "dessincronizado" actual.
- 50 En algunas implementaciones del codificador 100, el flujo de bits codificado almacenado temporalmente en la memoria 109 (y emitido al sistema de entrega 150) es un flujo de bits AC-3 o un flujo de bits E-AC-3, y comprende



segmentos de datos de audio (p. ej., los segmentos AB0-AB5 de la trama mostrada en la Fig. 4) y los 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 etapa 107 inserta LPSM (y, opcionalmente, también metadatos de límite de programa) en el flujo de bits en el siguiente formato. Cada uno de los segmentos de metadatos que incluye LPSM (y, opcionalmente, también metadatos de límite de programa) se incluye en un segmento de bits residuales del flujo de bits (p. ej., un segmento de bits residuales "W" como se muestra en la Fig. 4 o en la Fig. 7), o un campo "addbsi" del segmento de Información de Flujo de Bits ("BSI") de una trama del flujo de bits, o en un campo de auxdata (p. ej., el segmento AUX mostrado en la Fig. 4 o en la Fig. 7) al final de una trama del flujo de bits. Una trama del flujo de bits puede incluir uno o dos segmentos de metadatos, cada uno de los cuales incluye LPSM, y si la trama incluye dos segmentos de metadatos, uno puede estar presente en el campo de addbsi de la trama y el otro en el campo AUX de la trama. En algunas realizaciones, cada uno de los segmentos de metadatos que incluye LPSM, incluye un segmento de carga útil de LPSM (o contenedor) que tiene el siguiente formato:

una cabecera (que típicamente 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., la versión de formato de LPSM, la longitud, el período, el recuento y los valores de asociación de subflujo indicados en la Tabla 2 de abajo); y

después de la cabecera,

al menos un valor de indicación de diálogo (p. ej., parámetro "Canal(es) de diálogo" de la Tabla 2) que indica si los correspondientes datos de audio indican diálogo o no indican diálogo (p. ej., qué canales de los correspondientes datos de audio 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 correspondientes datos de audio 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 "indicador de Corrección de Sonoridad de Diálogo sincronizado", "Tipo de Corrección de Sonoridad" de la Tabla 2) que indican al menos un tipo de procesamiento de sonoridad que se ha realizado sobre los correspondientes datos de audio; y

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

En algunas realizaciones, cada uno de los segmentos de metadatos que contiene LPSM y metadatos de límite de programa, contiene una cabecera central (y, opcionalmente, también elementos centrales adicionales), y después de la cabecera central (o la cabecera central y otros elementos centrales), un segmento de carga útil de LPSM (o contenedor) que tiene el siguiente formato:

una cabecera, que típicamente incluye al menos un valor de identificación (p. ej., la versión de formato de LPSM, la longitud, el período, el recuento y los valores de asociación de subflujo, como se indica en la Tabla 2 que se expone en el presente documento), y

después de la cabecera, los LPSM y los metadatos de límite de programa. Los metadatos de límite de programa pueden incluir un recuento de tramas de límite de programa y un valor de código (p. ej., un valor "existe\_desplazamiento") indicativo de si la trama incluye solamente un recuento de tramas de límite de programa o un recuento de tramas de límite de programa y 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 etapa 107 en un segmento de bits residuales o en un campo "addbsi" o en un campo de auxdata de una trama del flujo de bits, tiene el siguiente formato:

una cabecera central (que típicamente incluye una palabra de sincronización que identifica el inicio del segmento de metadatos, seguida de valores de identificación, p. ej., la versión del elemento central, la longitud y el período, el recuento de elementos extendidos y los valores de asociación de subflujo indicados en la Tabla 1 de abajo); y

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

5 también después de la cabecera central, si el segmento de metadatos incluye LPSM, la identificación (“ID”) de carga útil de LPSM y los 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 anteriormente) sigue a la ID de carga útil de LPSM y a los valores de tamaño de carga útil de LPSM.

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

15 una estructura de alto nivel, que incluye una indicador que indica si el campo de auxdata (o addbsi) incluye metadatos, al menos un valor de ID que indica qué tipo(s) de metadatos está(n) presente(s), y, generalmente, también un valor que indica cuántos bits de metadatos (p. ej., de cada uno de los tipos) están presentes (si los metadatos están presentes). Un tipo de metadatos que podría estar presente son LSPM, otro tipo de metadatos que podría estar presente son los metadatos de límite de programa, y otro tipo de metadatos que podría estar presente son los metadatos de investigación de medios (p. ej., metadatos de Nielsen Media Research);

una estructura de nivel intermedio, que comprende un elemento central para cada uno de los tipos de metadatos identificado (p. ej., cabecera central, valores de protección y valores de ID de carga útil de LPSM y de tamaño de LPSM y, como se menciona anteriormente, para cada uno de los tipos de metadatos identificado); y

20 una estructura de bajo nivel, que comprende cada una de las cargas útiles para un elemento central (p. ej., una carga útil de LPSM, si se identificada una por el elemento central como presente, y/o una carga útil de metadatos de otro tipo, si una se identifica una por el elemento central como presente).

25 Los valores de datos en tal estructura de tres niveles pueden anidarse. Por ejemplo, el (los) valor(es) de protección para una carga útil de LPSM y/u otra carga útil de metadatos, identificado(s) por un elemento central, puede(n) incluirse después de cada una de las cargas útiles identificada por el elemento central (y, por lo tanto, después de la cabecera central del elemento central). En un ejemplo, una cabecera central podría identificar una carga útil de LPSM y otra carga útil de metadatos, la ID de carga útil y los valores de tamaño de carga útil para la primera carga útil (p. ej., la carga útil de LPSM) podrían seguir a la cabecera central, la primera carga útil podría seguir a los valores de ID y de tamaño, la ID de carga útil y el valor de tamaño de carga útil para la segunda carga útil podrían seguir a la primera carga útil, la segunda carga útil podría seguir a estos valores de ID y de tamaño, y a los bits de protección para ambas cargas útiles (o para los valores de elemento central y de ambas cargas útiles) podrían seguir a la última carga útil.

35 En algunas realizaciones, si el decodificador 101 recibe un flujo de bits de audio generado de acuerdo con una realización de la invención con hash criptográfico, el decodificador está configurado para analizar y recuperar el hash criptográfico de un bloque de datos determinado a partir del flujo de bits, comprendiendo dicho bloque los metadatos de estado de procesamiento de sonoridad (LPSM) y, opcionalmente, también los metadatos de límite de programa. El validador 102 puede utilizar el hash criptográfico para validar el flujo de bits recibido y/o los metadatos asociados. Por ejemplo, si el validador 102 encuentra que los LPSM son válidos en base a una coincidencia entre un hash criptográfico de referencia y el hash criptográfico recuperado desde el bloque de datos, entonces puede deshabilitar la operación del procesador 103 sobre los correspondientes datos de audio y hacer que la etapa de selección 104 pase (sin cambios) los datos de audio. Además, opcional o alternativamente, se pueden utilizar otros tipos de técnicas criptográficas en lugar de un método basado en un hash criptográfico.

45 El codificador 100 de la FIG. 2 puede determinar (en respuesta a LPSM, y, opcionalmente, también a metadatos de límite de programa, extraídos por el decodificador 101) que una unidad de post/preproceso ha realizado un tipo de procesamiento de sonoridad sobre los datos de audio a ser codificados (en los elementos 105, 106 y 107) y, por lo tanto, pueden crear (en el generador 106) metadatos de estado de procesamiento de sonoridad que incluyen los parámetros específicos utilizados y/o derivados del procesamiento de sonoridad realizado previamente. En algunas implementaciones, el codificador 100 puede crear (e incluir en la salida de flujo de bits codificado) metadatos de estado de procesamiento indicativos del historial de procesamiento sobre el contenido de audio siempre que el  
50 codificador conozca los tipos de procesamiento que se han realizado sobre el contenido de audio.

La FIG. 3 es un diagrama de bloques de un decodificador (200) que es una realización de la unidad de procesamiento de audio inventiva, y de un postprocesador (300) acoplado a la misma. El postprocesador (300) es

también una realización de la unidad de procesamiento de audio inventiva. Cualquiera de los componentes o elementos del decodificador 200 y del postprocesador 300 puede implementarse como uno o más procesos y/o uno o más circuitos (p. ej., ASIC, FPGA u otros circuitos integrados), en hardware, en software o en una combinación de hardware y software. El decodificador 200 comprende el búfer de tramas 201, el analizador 205, el decodificador de audio 202, la etapa de validación de estado de audio (validador) 203 y la etapa de generación de bits de control 204, conectados como se muestra. Típicamente también, el decodificador 200 incluye otros elementos de procesamiento (no mostrados).

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

El analizador 205 está acoplado y configurado para extraer metadatos de estado de procesamiento de sonoridad (LPSM) y, opcionalmente, también metadatos de límite de programa y otros metadatos, de cada una de las tramas del audio de entrada codificada para declarar al menos los LPSM (y se extraen metadatos de límite de programa, si los hay) al validador de estado de audio 203 y a la etapa 204, para declarar los LPSM (y, opcionalmente, también metadatos de límite de programa) como salida (p. ej., al postprocesador 300), para extraer datos de audio del audio entrada de codificado, y para declarar los datos de audio extraídos al decodificador 202.

La entrada del flujo de bits de audio codificado al decodificador 200 puede ser una de un flujo de bits AC-3, un flujo de bits E-AC-3 o un flujo de bits Dolby E.

El sistema de la FIG. 3 también incluye el postprocesador 300. El postprocesador 300 comprende el búfer de tramas 301 y otros elementos de procesamiento (no mostrados) que incluyen al menos un elemento de procesamiento acoplado al búfer 301. El búfer de tramas 301 almacena (p. ej., de manera no transitoria) al menos una trama del flujo de bits de audio decodificado recibido por el postprocesador 300 desde el decodificador 200. Los elementos de procesamiento del postprocesador 300 están acoplados y configurados para recibir y procesar de manera adaptativa una secuencia de las tramas de la salida de flujo de bits de audio decodificado del búfer 301, utilizando metadatos (que incluyen valores de LPSM) emitidos desde el decodificador 202 y/o bits de control emitidos desde la etapa 204 del decodificador 200. Típicamente, el postprocesador 300 está configurado para realizar procesamiento de sonoridad adaptativo sobre los datos de audio decodificados utilizando los valores de LPSM y, opcionalmente, también metadatos de límite de programa (p. ej., en base al estado de procesamiento de sonoridad, y/o a una o más características de datos de audio, indicada por LPSM para datos de audio indicativos de un único programa de audio).

Diversas implementaciones del decodificador 200 y del postprocesador 300 están configuradas para realizar diferentes realizaciones del método de la invención.

El decodificador de audio 202 del decodificador 200 está configurado para decodificar los datos de audio extraídos por el analizador 205 para generar datos de audio decodificados, y para declarar los datos de audio decodificados como salida (p. ej., al postprocesador 300).

El validador de estado 203 está configurado para autenticar y validar los LPSM (y opcionalmente otros metadatos) declarados al mismo. En algunas realizaciones, los LPSM son (o están incluidos en) un bloque de datos que se ha incluido en el flujo de bits de entrada (p. ej., de acuerdo con una realización de la presente invención). El bloque puede comprender un hash criptográfico (un código de autenticación de mensajes en clave-hash o "HMAC") para procesar los LPSM (y, opcionalmente, también otros metadatos) y/o los datos de audio subyacentes (proporcionados desde el analizador 205 y/o el decodificador 202 al validador 203). El bloque de datos puede estar firmado digitalmente en estas realizaciones, de modo que una unidad de procesamiento de audio de flujo descendente puede autenticar y validar con relativa facilidad los metadatos de estado de procesamiento.

Otros métodos criptográficos incluyendo, pero no limitados a, cualquiera de uno o más métodos criptográficos no HMAC puede ser utilizado para la validación de LPSM (p. ej., en el validador 203) para garantizar la transmisión y recepción segura de los LPSM y/o de los datos de audio subyacentes. Por ejemplo, la validación (utilizando tal método criptográfico) puede realizarse en cada una de las unidades de procesamiento de audio que recibe una realización del flujo de bits de audio inventivo para determinar si los metadatos de estado de procesamiento de sonoridad y los correspondientes datos de audio incluidos en el flujo de bits, han sido sometidos a (y/o han resultado de) un procesamiento de sonoridad específico (como lo indican los metadatos) y no se han modificado después de la ejecución de dicho procesamiento de sonoridad específico.

El validador de estado 203 declara los datos de control al generador de bits de control 204, y/o declara 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, opcionalmente, también a otros metadatos extraídos del flujo de bits de entrada), la etapa 204 puede generar (y generar al postprocesador 300) bien:

5 bits de control que indican que los datos de audio decodificados emitidos desde el decodificador 202 se han sometido a un tipo específico de procesamiento de sonoridad (cuando los LPSM indican que los datos de audio emitidos desde el decodificador 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

10 bits de control que indican que los datos de audio decodificados emitidos desde el decodificador 202 deben someterse a un tipo específico de procesamiento de sonoridad (p. ej., cuando los LPSM indican que la salida de datos de audio del decodificador 202 no se ha sometido al tipo específico de procesamiento de sonoridad o cuando los LPSM indican que la salida de datos de audio del decodificador 202 se ha sometido al tipo específico de procesamiento de sonoridad, pero los bits de control del validador 203 indican que los LPSM no son válidos).

15 Alternativamente, el decodificador 200 declara los metadatos extraídos por el decodificador 202 del flujo de bits de entrada, y los LPSM (y, opcionalmente, también los metadatos de límite de programa) extraídos por el analizador 205 del flujo de bits de entrada para el postprocesador 300, y el postprocesador 300 realiza el procesamiento de sonoridad sobre los datos de audio decodificados utilizando los LPSM (y, opcionalmente, también los metadatos de límite de programa), o realiza la validación de los LPSM y luego realiza el procesamiento de sonoridad sobre los datos de audio decodificados utilizando los LPSM (y, opcionalmente, también metadatos de límite de programa) si la validación indica que los LPSM son válidos.

20

En algunas realizaciones, si el decodificador 200 recibe un flujo de bits de audio generado de acuerdo con una realización de la invención con hash criptográfico, el decodificador está configurado para analizar y recuperar el hash criptográfico de un bloque de datos determinado del flujo de bits, comprendiendo dicho bloque los metadatos de estado de procesamiento de sonoridad (LPSM). El validador 203 puede utilizar el hash criptográfico para validar el flujo de bits recibido y/o los metadatos asociados. Por ejemplo, si el validador 203 encuentra que los LPSM son válidos en base a una coincidencia entre un hash criptográfico de referencia y el hash criptográfico recuperado del bloque de datos, entonces puede señalar a una unidad de procesamiento de audio de flujo descendente (p. ej., el postprocesador 300, que puede ser o incluir una unidad de nivelación de volumen) para pasar (sin cambios) los datos de audio del flujo de bits. Además, opcional o alternativamente, se pueden utilizar otros tipos de técnicas criptográficas en lugar de un método basado en un hash criptográfico.

25

30

En algunas implementaciones del decodificador 200, el flujo de bits codificado recibido (y almacenado temporalmente en la memoria 201) es un flujo de bits AC-3 o un flujo de bits E-AC-3, y comprende segmentos de datos de audio (p. ej., los segmentos AB0-AB5 de la trama mostrada en la Fig. 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, opcionalmente, también metadatos de límite de programa. La etapa 202 del decodificador (y/o del analizador 205) está configurada para extraer LPSM del flujo de bits (y, opcionalmente, también metadatos de límite de programa) que tienen el siguiente formato. Cada uno de los segmentos de metadatos que incluye LPSM (y, opcionalmente, también metadatos de límite de programa) se incluye en un segmento de bits residuales de una trama del flujo de bits, o en un campo "addbsi" del segmento de Información de Flujo de Bits ("BSI") de una trama de flujo de bits, o en un campo de auxdata (p. ej., el segmento AUX mostrado en la Fig. 4) al final de una trama del flujo de bits. Una trama del flujo de bits puede incluir uno o dos segmentos de metadatos, cada uno de los cuales puede incluir LPSM, y si la trama incluye dos segmentos de metadatos, uno puede estar presente en el campo de addbsi de la trama y el otro en el campo AUX de la trama. En algunas realizaciones, cada uno de los segmentos de metadatos que incluye LPSM incluye un segmento de carga útil de LPSM (o contenedor) que tiene el siguiente formato:

35

40

45

una cabecera (que típicamente incluye una palabra de sincronización que identifica el inicio de la carga útil de LPSM, seguida de valores de identificación, p. ej., la versión de formato de LPSM, la longitud, el período, el recuento y los valores de asociación de la subflujo indicados en la Tabla 2 de abajo); y

después de la cabecera,

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

5 al menos un valor de procesamiento de sonoridad (p. ej., uno o más de los parámetros “indicador de Corrección de Sonoridad de Diálogo”, “Tipo de Corrección de Sonoridad” de la Tabla 2) que indican al menos un tipo de procesamiento de sonoridad que se ha realizado sobre los correspondientes datos de audio; y

10 al menos un valor de sonoridad (p. ej., uno o más de los parámetros “Sonoridad Relativa Sincronizada de UIT”, “Sonoridad de Voz Sincronizada de UIT”, “Sonoridad de 3s a Corto plazo de UIT (EBU 3341)” y “Pico Verdadero” de la Tabla 2) que indican al menos una característica de sonoridad (p. ej., sonoridad pico o promedio) de los correspondientes datos de audio.

En algunas realizaciones, cada uno de los segmentos de metadatos que contiene LPSM y metadatos de límite de programa, contiene una cabecera central (y, opcionalmente, también elementos centrales adicionales), y después de la cabecera central (o de la cabecera central y de otros elementos centrales) un segmento de carga útil de LPSM (o contenedor) que tiene el siguiente formato:

15 una cabecera, que generalmente incluye al menos un valor de identificación (p. ej., la versión de formato de LPSM, la longitud, el período, el recuento y los valores de asociación de subflujo, como se indica en la Tabla 2 de abajo), y

20 después de la cabecera, los LPSM y los metadatos de límite de programa. Los metadatos de límite de programa pueden incluir un recuento de tramas de límite de programa y un valor de código (p. ej., un valor “existe\_desplazamiento”) indicativo de si la trama incluye solamente un recuento de tramas de límite de programa o un recuento de tramas de límite de programa y un valor de desplazamiento), y (en algunos casos) un valor de desplazamiento.

25 En algunas implementaciones, el analizador 205 (y/o la etapa de decodificador 202) está configurado para extraer, de un segmento de bits residuales, o de un campo “addbsi”, o de un campo de auxdata, de una trama del flujo de bits, cada uno de los segmentos de metadatos que tiene el siguiente formato:

una cabecera central (que típicamente incluye una palabra de sincronización que identifica el inicio del segmento de metadatos, seguido de al menos un valor de identificación, p. ej., la versión de elemento Central, la longitud y el período, el recuento extendido de elementos y los valores de asociación de subflujo indicados en la Tabla 1 de abajo); y

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

35 también después de la cabecera central, si el segmento de metadatos incluye LPSM, la identificación (“ID”) de carga útil de LPSM y los 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 anteriormente) sigue a los valores de ID de carga útil de LPSM y de tamaño de carga útil de LPSM.

40 De manera más general, el flujo de bits de audio codificado generado por realizaciones preferidas de la invención, tiene una estructura que proporciona un mecanismo para etiquetar elementos y subelementos de metadatos como centrales (obligatorios) o expandidos (elementos opcionales). Esto permite que la tasa de datos del flujo de bits (incluidos sus metadatos) se escale en numerosas aplicaciones. Los elementos centrales (obligatorios) de la sintaxis del flujo de bits preferida, también deberían ser capaces de señalar que los elementos expandidos (opcionales), asociados con el contenido de audio están presentes (en banda) y/o en una ubicación remota (fuera de banda).

45 Se requiere que los elementos centrales estén presentes en cada una de las tramas del flujo de bits. Algunos subelementos de los elementos centrales son opcionales y pueden estar presentes en cualquier combinación. No se requiere que los elementos expandidos estén presentes en cada una de las tramas (para limitar la sobrecarga de la tasa de bits). Por lo tanto, los elementos expandidos pueden estar presentes en algunas tramas y en otras no. 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 (es decir, si el elemento expandido está presente en una trama del flujo de bits).

50

5 En una clase de realizaciones, se genera un flujo de bits de audio codificado que comprende una secuencia de segmentos de datos de audio y de segmentos de metadatos (p. ej., mediante una unidad de procesamiento de audio que incorpora la invención). Los segmentos de datos de audio son indicativos de datos de audio, cada uno de al menos algunos de los segmentos de metadatos incluye metadatos de estado de procesamiento de sonoridad (LPSM) y, opcionalmente, también metadatos de límite de programa, y los segmentos de datos de audio son multiplexados por división de tiempo con los segmentos de metadatos. En realizaciones preferidas en esta clase, cada uno de los segmentos de metadatos tiene un formato preferido a ser descrito en el presente documento.

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

15 En el formato preferido, cada una de las tramas incluye un elemento central que tiene el formato mostrado en la Tabla 1 de abajo, en el campo de addbsi (o segmento de bits residuales) de la trama:

Tabla 1

Parámetro	Descripción	Obligatorio/Opcional
SYNC [ID]	La palabra de sincronización puede ser un valor de 16 bits ajustado al valor de 0x5838	Ob
Versión de elemento central		Ob
Longitud de elemento central		Ob
Periodo de elemento central(xxx)		Ob
Recuento de elementos extendidos	Indica la cantidad de elementos de metadatos extendidos asociados con el elemento central. Este valor puede incrementarse/decrementarse a medida que el flujo de bits pasa de producción a través de distribución y emisión final.	Ob
Asociación de subflujo	Describe a qué subflujo(s) está asociado el elemento central.	Ob
Firma (resumen de HMAC)	El resumen de HMAC de 256 bits (utilizando el algoritmo SHA-2) calculado sobre los datos de audio, el elemento central y todos los elementos expandidos de la trama completa.	Ob
Cuenta hacia abajo de límite de PGM	El campo solo aparece para algún número de tramas en la cabeza o cola de un archivo/flujo de programa de audio. Por lo tanto, un cambio de versión de elemento central podría utilizarse para señalar la inclusión de este parámetro.	Op
Huella Digital de Audio	La Huella Digital de Audio tomada sobre algún número de muestras de audio PCM representadas por el campo de período de elemento central.	Op
Huella Digital de Video	La Huella Digital de Video tomada sobre algún número de muestras de video comprimido (si las hay) representadas por el campo de período de elemento central.	Op
URL/UUID	Este campo está definido para transportar una URL y/o una UUID (puede ser redundante para la huella digital) que hace referencia a una ubicación externa de contenido de programa adicional (esencia) y/o metadatos asociados con el flujo de bits.	Op

En el formato preferido, cada uno de los campos de addbsi (o auxdata) o segmentos de bits residuales que contienen LPSM, contienen una cabecera central (y, opcionalmente, también elementos centrales adicionales), y

## ES 2 667 871 T3

después de la cabecera central (o la cabecera central y otros elementos centrales), los siguientes valores de LPSM (parámetros):

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

5 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 de LPSM (que siguen a la ID de carga útil y al valor del tamaño de carga útil) que tiene el formato que se indica en la siguiente tabla (Tabla 2):

Tabla 2

Parámetro LPSM [Sonoridad inteligente]	Descripción	número de estados únicos	Obligatorio/Opcional	Tasa de inserción (Periodo de actualización del parámetro)
Versión de LPSM			Ob	
Período de LPSM (xxx)	Aplicable solo a los campos xxx		Ob	
Recuento de LPSM			Ob	
Asociación de subflujo LPSM			Ob	
Canal(es) de diálogo	Indica qué combinación de canales de audio L, C y R contienen voz durante los 0,5 segundos anteriores. Cuando la voz no está presente en ninguna combinación L, C o R, entonces este parámetro indicará "sin diálogo"	8	Ob	~ 0,5 segundos (típico)
Tipo de Regulación de Sonoridad	Indica que el flujo de datos de audio asociado está en conformidad con un conjunto específico de regulaciones (p. Ej., ATSC A/85 o EBU R128)	8	Ob	Trama
Indicador de Corrección de Sonoridad de Diálogo Sincronizado	Indica si el flujo de audio asociado se ha corregido en base a la sincronización de diálogo	2	Op (solo presente si Tipo_Regulación_Sonoridad indica que el correspondiente audio está SIN CORREGIR)	Trama
Tipo de Corrección de Sonoridad	Indica si el flujo de audio asociado se ha corregido con una búsqueda infinita anticipada (basada en archivos) o con un controlador de sonoridad en tiempo real (RT) y de rango dinámico.	2	Op (solo presente si Tipo_Regulación_Sonoridad indica que el correspondiente audio está SIN CORREGIR)	Trama
Sonoridad Relativa Sincronizada de UIT (INF)	Indica la sonoridad integrada de UIT-R BS.1770-3 del flujo de audio asociado sin metadatos aplicados (p. ej., 7 bits: -58 -> +5,5 LKFS 0,5 pasos de LKFS)	128	Op	1 seg.

Sonoridad de Voz Sincronizada de UIT (INF)	Indica la sonoridad integrada de UIT-R BS.1770-1/3 de la voz/diálogo del flujo de audio asociado sin metadatos aplicados (p. ej., 7 bits: -58 -> +5,5 LKFS 0,5 pasos de LKFS)	128	Op	1 seg.
Sonoridad de 3s a Corto plazo de UIT (EBU 3341)	Indica la sonoridad de UIT (UIT-BS.1771-1) dessincronizada de 3 segundos del flujo de audio asociado sin metadatos aplicados (ventana deslizante) @ ~ tasa de inserción de 10Hz (p. ej., 8bits: 116 -> +11,5 LKFS 0,5 pasos de LKFS)	256	Op	0,1 seg.
Valor de Pico Verdadero	Indica el valor de PicoVerdadero (dB TP) de UIT-R BS.1770-3 Anexo 2 del flujo de audio asociado sin metadatos aplicados. (Es decir, el valor más grande durante el período de trama señalizado en el campo de período de elemento) 116 -> +11,5 LKFS 0,5 pasos de LKFS	256	Op	0,5 seg.
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 está en el límite de trama, el desplazamiento de muestra opcional indicará qué tan lejos en la trama se produce el límite real de programa			

5 En otro formato preferido de un flujo de bits codificado generado de acuerdo con la invención, el flujo de bits es un flujo de bits AC-3 o un flujo de bits E-AC-3, y cada uno de los segmentos de metadatos que incluye LPSM (y, opcionalmente, también metadatos de límite de programa) se incluye (p. ej., mediante la etapa 107 de una implementación preferida del codificador 100) en cualquiera de: un segmento de bits residuales de una trama del flujo de bits; o un campo "addbsi" (mostrado en la Fig. 6) del segmento de Información de Flujo de Bits ("BSI") de una trama del flujo de bits; o un campo de auxdata (p. ej., el segmento AUX mostrado en la Fig. 4) al final de una trama del flujo de bits. Una trama puede incluir uno o dos segmentos de metadatos, cada uno de los cuales incluye LPSM, y si la trama incluye dos segmentos de metadatos, uno puede estar presente en el campo de addbsi de la trama y el otro en el campo AUX de la trama. Cada uno de los segmentos de metadatos que incluye LPSM tiene el formato especificado anteriormente con referencia a las Tablas 1 y 2 anteriores (es decir, incluye los elementos centrales especificados en la Tabla 1, seguidos por la ID de carga útil (identificando los metadatos como LPSM) y los valores de tamaño de carga útil especificados anteriormente, seguidos de la carga útil (los datos de LPSM que tienen el formato indicado en la Tabla 2).

15 En otro formato preferido, el flujo de bits codificado es un flujo de bits Dolby E, y cada uno de los segmentos de metadatos que incluye LPSM (y, opcionalmente, también metadatos de límite de programa), son las primeras N ubicaciones de muestra del intervalo de banda de guarda de Dolby E. Un flujo de bits Dolby E que incluye un segmento de metadatos que incluye LPSM, incluye preferiblemente un valor indicativo de la longitud de carga útil de LPSM señalizada en la palabra Pd del preámbulo SMPTE 337M (la tasa de repetición de palabra SMPTE 337M Pa permanece preferiblemente idéntica a la tasa de tramas de video asociada).

20 En un formato preferido, en el que el flujo de bits codificado es un flujo de bits E-AC-3, cada uno de los segmentos de metadatos que incluye LPSM (y, opcionalmente, también metadatos de límite de programa) se incluye (p. ej., en la etapa 107 de una implementación preferida del codificador 100) como información de flujo de bits adicional en un segmento de bits residuales, o en el campo "addbsi" del segmento de Información de Flujo de Bits ("BSI"), de una



trama del flujo de bits. A continuación, describimos aspectos adicionales de la codificación de un flujo de bits E-AC-3 con LPSM en este formato preferido:

1. durante la generación de un flujo de bits E-AC-3, mientras que el codificador E-AC-3 (que inserta los valores de LPSM en el flujo de bits) está “activo”, para cada una de las tramas (trama de sincronización) generada, el flujo de bits debe incluir un bloque de metadatos (incluyendo LPSM) transportado en el campo de addbsi (o segmento de bits residuales) de la trama. Los bits necesarios para transportar el bloque de metadatos no deberían aumentar la tasa de bits del codificador (longitud de trama);

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

indicador\_tipo\_corrección\_sonoridad :donde ‘1’ indica que la sonoridad de los correspondientes datos de audio se corrigió aguas arriba del codificador, y ‘0’ indica que la sonoridad fue corregida por un corrector de sonoridad incrustado en el codificador (p. ej., el procesador de sonoridad 103 del codificador 100 de la Fig. 2);

canal\_voz: indica qué canal(es) de origen contiene(n) voz (durante los 0,5 segundos previos). Si no se detecta la voz, esto se indicará como tal;

sonoridad\_voz: indica la sonoridad de voz integrada de cada uno de los correspondientes canales de audio que contiene voz (durante los 0,5 segundos previos);

sonoridad UIT: indica la sonoridad de UIT BS.1770-3 integrada de cada uno de los correspondientes canales de audio; y

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

3. Mientras que el codificador E-AC-3 (que inserta los valores de LPSM en el flujo de bits) está “activo” y está recibiendo una trama AC-3 con un indicador de “confianza”, el controlador de sonoridad en el codificador (p. ej., procesador de sonoridad 103 del codificador 100 de la Fig. 2) se debe anular. Los valores de dialnorm y DRC de origen “de confianza” deberían pasar (p. ej., por el generador 106 del codificador 100) al componente del codificador E-AC-3 (p. ej., la etapa 107 del codificador 100). La generación de bloque de LPSM continúa y el indicador\_tipo\_corrección\_sonoridad se ajusta a ‘1’. La secuencia de omisión del controlador de sonoridad debe estar sincronizada con el inicio de la trama AC-3 decodificada donde aparece el indicador ‘confianza’. La secuencia de omisión del controlador de sonoridad debe implementarse de la siguiente manera: el control de cantidad\_nivelador decrementa desde un valor de 9 a un valor de 0 en 10 periodos de bloque de audio (es decir, 53,3 mseg.) y el control medidor\_final\_vuelta\_nivelador se coloca en modo de omisión (esta operación debería dar como resultado una transición sin interrupciones). El término omisión ‘de confianza’ del nivelador implica que el valor de dialnorm del flujo de bits de origen también se reutiliza en la salida del codificador. (P. ej., si el flujo de bits de origen ‘de confianza’ tiene un valor de dialnorm de -30, entonces la salida del codificador debe utilizar -30 para el valor de dialnorm de salida);

4. Mientras que el codificador E-AC-3 (que inserta los valores de LPSM en el flujo de bits) está “activo” y está recibiendo una trama AC-3 sin el indicador de ‘confianza’, el controlador de sonoridad incrustado en el codificador (p. ej., el procesador de sonoridad 103 del codificador 100 de la Fig. 2) debe estar activo. La generación de bloque de LPSM continúa y el indicador\_tipo\_corrección\_sonoridad se ajusta a ‘0’. La secuencia de activación del controlador de sonoridad debe sincronizarse con el inicio de la trama AC-3 decodificada, donde desaparece el indicador de ‘confianza’. La secuencia de activación del controlador de sonoridad debe implementarse de la siguiente manera: el control cantidad\_nivelador se incrementa desde un valor de 0 a un valor de 9 en 1 período de bloque de audio. (Es decir, 5,3 mseg.) Y el control medidor\_final\_vuelta\_nivelador se coloca en modo ‘activo’ (esta operación debería dar como resultado una transición sin interrupciones e incluir un reinicio de medidor\_final\_vuelta\_nivelador); y

5. durante la codificación, una interfaz gráfica de usuario (GUI) debe indicar a un usuario los siguientes parámetros: “Programa de Audio de entrada: [Confiable/No Confiable]” -el estado de este parámetro se basa en la presencia del indicador “confianza” dentro de la señal de entrada; y “Corrección de Sonoridad en Tiempo real: [Activado/Desactivado]” -el estado de este parámetro se basa en si este controlador de sonoridad incrustado en el codificador está activo.

Quando se decodifica un flujo de bits AC-3 o E-AC-3 que tiene LPSM (en el formato preferido) incluido en un segmento de bits residuales, o en el campo “addbsi” del segmento de Información de Flujo de Bits (“BSI”), de cada una de las tramas del flujo de bits, el decodificador debe analizar los datos del bloque de LPSM (en el segmento de

bits residuales o en el campo de addbsi) y pasar todos los valores de LPSM extraídos a una interfaz gráfica de usuario (GUI). El conjunto de valores de LPSM extraídos se actualiza cada trama.

5 En otro formato preferido de un flujo de bits codificado, generado de acuerdo con la invención, el flujo de bits codificado es un flujo de bits AC-3 o un flujo de bits E-AC-3, y cada uno de los segmentos de metadatos que incluye LPSM se incluye (p. ej., por la etapa 107 de una implementación preferida del codificador 100) en un segmento de bits residuales, o en un segmento Aux, o como información de flujo de bits adicional en el campo "addbsi" (mostrado en la Fig. 6) del segmento de Información de Flujo de Bits ("BSI"), de una trama del flujo de bits. En este formato (que es una variación del formato descrito anteriormente con referencias a las Tablas 1 y 2), cada uno de los campos addbsi (o Aux o bits residuales) que contiene LPSM contiene los siguientes valores de LPSM:

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

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

15 dialchan: un campo de 3 bits que indica si los canales izquierdo, derecho y/o central de los correspondientes datos de audio contienen diálogo hablado. La asignación de bits del campo dialchan puede ser la siguiente: el bit 0, que indica la presencia de diálogo en el canal izquierdo, se almacena en el bit más significativo del campo dialchan; y el bit 2, que indica la presencia de diálogo en el canal central, se almacena en el bit menos significativo del campo dialchan.

20 Cada uno de los bits del campo dialchan se ajusta a '1' si el correspondiente canal contiene diálogo hablado durante los 0,5 segundos anteriores del programa;

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

30 loudcorrldialgat: campo de un bit que indica si se ha aplicado la corrección de sonoridad de diálogo sincronizado. Si la sonoridad del programa se ha corregido utilizando sincronización de diálogo, el valor del campo loudcorrldialgat se ajusta a '1'. De lo contrario, se ajusta a '0';

35 loudcorrtyp: campo de un bit que indica el tipo de corrección de sonoridad aplicado al programa. Si la sonoridad del programa se ha corregido con un proceso de corrección de sonoridad de búsqueda anticipada infinita (basada en archivo), el valor del campo loudcorrtyp se ajusta a '0'. Si la sonoridad del programa se ha corregido utilizando una combinación de medición de sonoridad en tiempo real y control de rango dinámico, el valor de este campo se ajusta a '1';

loudrelgate: campo de un bit que indica si existen datos de sonoridad relativa sincronizada (UIT). Si el campo de loudrelgate se ajusta a '1', en la carga útil debe seguir un campo ituloudrelgat de 7 bits;

40 loudrelgat: un campo de 7 bits que indica la sonoridad de programa relativa sincronizada (UIT). Este campo indica la sonoridad integrada del programa de audio, medida de acuerdo con BS.1770-3 de UIT-R sin ningún ajuste de ganancia debido a la aplicación de dialnorm y de compresión de rango de dinámico. Los valores de 0 a 127 se interpretan como -58 LKFS a +5,5 LKFS, en 0,5 pasos de LKFS;

loudspchgate: campo de un bit que indica si existen datos de sonoridad de voz sincronizada (UIT). Si el campo loudspchgate se ajusta a '1', un campo de loudspchgat de 7 bits debe seguir en la carga útil;

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

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

5 loudstrm3s: un campo de 7 bits que indica la sonoridad dessincronizada de los 3 segundos anteriores del correspondiente programa de audio, medido de acuerdo con la UIT-R BS.1771-1 y con ningún ajuste de ganancia debido a la aplicación de dialnorm y de compresión de rango dinámico. Los valores de 0 a 256 se interpretan como -116 LKFS a +11,5 LKFS en 0,5 pasos de LKFS;

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

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

15 En algunas realizaciones, el elemento central de un segmento de metadatos en un segmento de bits residuales o en un campo de auxdata (o "addbsi") de una trama de un flujo de bits AC-3 o un flujo de bits E-AC-3, comprende una cabecera central (que típicamente incluye valores de identificación, p. ej. versión de elemento central) y después de la cabecera central: 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 los datos externos (relacionados con datos de audio correspondiente a los metadatos del segmento de metadatos), ID de carga útil y tamaño de carga útil para cada uno de los tipos de metadatos (p. ej., LPSM y/o metadatos de un tipo distinto de LPSM) identificados por el elemento central, y valores de protección para al menos un tipo de metadatos identificado por el elemento central. La(s) carga(s) útil(es) de metadatos del segmento de metadatos sigue(n) a la cabecera central, y está(n) (en algunos casos) anidada(s) dentro de los valores del elemento central.

20

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

30

35 En realizaciones típicas, los metadatos de límite de programa en una trama del flujo de bits inventivo son un indicador de límite de programa indicativo de un recuento de tramas. Típicamente, el indicador es indicativo del número de tramas entre la trama actual (la trama que incluye el indicador) y un límite de programa (el comienzo o el final del programa de audio actual). En algunas realizaciones preferidas, los indicadores de límite de programa se insertan de manera simétrica y eficiente al comienzo y al final de cada uno de los segmentos de flujo de bits, lo cual es indicativo de un único programa (es decir, en tramas que se producen dentro de un número predeterminado de tramas después del comienzo del segmento, y en tramas que se producen dentro de un número predeterminado de tramas antes del final del segmento), de modo que cuando se concatenan dos tales segmentos de flujo de bits (para que sean indicativos de una secuencia de dos programas), los metadatos de límite de programa pueden estar presentes (p. ej., simétricamente) en ambos lados del límite entre los dos programas.

40

45 La máxima robustez se puede lograr insertando un indicador de límite de programa en cada una de las tramas de un flujo de bits indicativo de un programa, pero esto normalmente no sería práctico debido al aumento asociado en la tasa de datos. En realizaciones típicas, los indicadores de límite de programa se insertan solo en un subconjunto de tramas de un flujo de bits de audio codificado (el cual puede ser indicativo de un programa de audio o una secuencia de programas de audio) y la tasa de inserción de indicador de límite, es una función no creciente de separación creciente de cada una de las tramas del flujo de bits (en la que se inserta un indicador) desde el límite del programa que está más cerca de cada una de dichas tramas, donde la "tasa de inserción de indicador de límite" se refiere a la proporción promedio del número de tramas (indicativas de un programa) que incluyen un indicador de límite de programa con el número de tramas (indicativo del programa), que no incluyen un indicador de límite de programa, donde el promedio es un promedio continuo sobre un número (p. ej., número relativamente pequeño) de tramas consecutivas del flujo de bits de audio codificado.

50

El aumento de la tasa de inserción del indicador de límite (p. ej., en ubicaciones en el flujo de bits más cercanas a un límite de programa) aumenta la tasa de datos requerida para la entrega del flujo de bits. Para compensar esto, el tamaño (número de bits) de cada uno de los indicadores insertado se reduce preferiblemente a medida que aumenta

la tasa de inserción del indicador de límite (p. ej., de modo que el tamaño del indicador de límite de programa en la “N”-ésima trama del flujo de bits, donde N es un número 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 tasa de inserción de indicador de límite es una función logarítmicamente decreciente de distancia creciente (de cada una de las ubicaciones de inserción de indicador) desde el límite de programa más cercano, y para cada una de las tramas que contiene indicador que incluye uno de los indicadores, el tamaño del indicador en dicha trama que contiene indicador es igual o mayor que el tamaño de cada uno de los indicadores en una trama ubicada más cerca del límite del programa más cercano que dicha trama que contiene el indicador. Típicamente, el tamaño de cada uno de los indicadores se determina mediante una función creciente del número de tramas desde la ubicación de inserción del indicador hasta el límite de programa más cercano.

Por ejemplo, considere la realización de las Figs. 8 y 9, en la que cada una de las columnas identificada por un número de trama (en la fila superior) indica una trama de un flujo de bits de audio codificado. El flujo de bits es indicativo de un programa de audio que tiene un primer límite de programa (indicativo del comienzo del programa) que se produce inmediatamente a la izquierda de la columna identificada por el número de trama “17” en el lado izquierdo de la Fig. 9, y un segundo límite de programa (indicativo del final del programa) que se produce inmediatamente a la derecha de la columna identificada por el número de trama “1” en el lado derecho de la Fig. 8. Los indicadores de límite de programa incluidos en las tramas mostradas en la Fig. 8 cuentan hacia abajo el número de tramas entre la trama actual y el segundo límite de programa. Los indicadores de límite de programa incluidos en las tramas que se muestran en la Fig. 9 cuentan el número de tramas entre la trama actual y el primer límite de programa.

En la realización de las Figs. 8 y 9, se inserta un indicador de límite de programa solo en cada una de las “ $2^N$ ”-ésimas tramas de las primeras X tramas del flujo de bits codificado después del comienzo del programa de audio indicado por el flujo de bits, y en cada una de las “ $2^N$ ”-ésimas tramas (de las últimas X tramas del flujo de bits) más cercanas al final del programa indicado por el flujo de bits, donde el programa comprende Y tramas, X es un número entero menor o igual que  $Y/2$ , y N es un entero positivo en un rango desde 1 hasta  $\log_2(X)$ . Por lo tanto, (como se indica en las Figs. 8 y 9), se inserta un indicador de límite de programa en la segunda trama (N = 1) del flujo de bits (la trama que contiene el indicador más cercano al comienzo del programa), en la cuarta trama (N = 2), en la octava trama (N = 3), y así sucesivamente, y en la octava trama desde el final del flujo de bits, en la cuarta trama desde el final del flujo de bits, y en la segunda trama desde el final del flujo de bits (la trama que contiene el indicador más cercano al final del programa). En este ejemplo, el indicador de límite de programa en la “ $2^N$ ”-ésima trama desde el comienzo (o final) del programa comprende  $\log_2(2^{N+2})$  bits binarios, como se indica en las Figs. 8 y 9. Por lo tanto, el indicador de límite de programa en la segunda trama (N = 1) desde el comienzo (o final) del programa comprende  $\log_2(2^{N+2}) = \log_2(2^3) = 3$  bits binarios, y el indicador en la cuarta trama (N = 2) desde el comienzo (o final) del programa comprende  $\log_2(2^{N+2}) = \log_2(2^4) = 4$  bits binarios, y así sucesivamente.

En el ejemplo de las Figs. 8 y 9, el formato de cada uno de los indicadores de límite de programa es el siguiente. Cada uno de los indicadores de límite de programa consiste en un bit “1” inicial, una secuencia de bits “0” (ya sea ningún bit “0” o uno o más bits “0” consecutivos) después del bit inicial y un código final de dos bits. El código final es “11” para los indicadores en las últimas X tramas del flujo de bits (las tramas más cercanas al final de programa), como se indica en la Fig. 8. El código final es “10” para los indicadores en las primeras X tramas del flujo de bits (las tramas más cercanas al comienzo del programa), como se indica en la Fig. 9. Por lo tanto, para leer (decodificar) cada uno de los indicadores, se cuenta el número de ceros entre el bit “1” inicial y el código final. Si el código final se identifica como “11”, el indicador indica que hay  $(2^{Z+1} - 1)$  tramas entre la trama actual (la trama que incluye el indicador) y el final del programa, donde Z es el número de ceros entre el bit “1” inicial del indicador y el código final. El decodificador se puede implementar de manera eficiente para ignorar el primer y el último bit de cada uno de los indicadores, para determinar la inversa de la secuencia de los otros bits (intermedios) del indicador (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” siendo el bit “1” 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 el indicador) relativo al final del programa. Por ejemplo, si la secuencia invertida de bits intermedios es “1000”, esta secuencia invertida tiene el valor binario de  $2^4 = 16$ , y la trama se identifica como la 16ª trama antes del final del programa (como se indica en la columna de la Fig. 8 que describe la trama “0”).

Si el código final se identifica como “10”, el indicador indica que hay  $(2^{Z+1} - 1)$  tramas entre el comienzo del programa y la trama actual (la trama que incluye el indicador), donde Z es el número de ceros entre el bit “1” inicial del indicador y el código final. El decodificador puede implementarse de manera eficiente para ignorar el primer y el último bit de cada uno de tales indicadores, para determinar la inversa de la secuencia de los bits intermedios del indicador (p. ej., si la secuencia de bits intermedios es “0001” siendo el bit “1” el último bit en la secuencia, la secuencia invertida de bits intermedios es “1000” siendo el bit “1” 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 está incluido el indicador) relativo al comienzo del programa. Por ejemplo, si la secuencia invertida de bits

intermedios es "1000", esta secuencia invertida tiene el valor binario  $2^4 = 16$ , y la trama se identifica como el 16ª trama después del comienzo del programa (como se indica en la columna de la Fig. 9 que describe la trama "32").

En el ejemplo de las Figs. 8 y 9, un indicador de límite de programa solo está presente en cada una de las " $2^N$ "-ésimas tramas de las primeras X tramas de un flujo de bits codificado después del comienzo de un programa de audio indicado por el flujo de bits, y en cada una de las " $2^N$ "-ésimas tramas (de las últimas X tramas del flujo de bits) más cercanas al final del programa indicado por el flujo de bits, donde el programa comprende Y tramas, X es un entero menor que o igual que  $Y / 2$ , y N es un entero positivo en un rango desde 1 hasta  $\log_2(X)$ . La inclusión de los indicadores de límite de programa agrega solo una tasa de bits promedio de 1,875 bits/trama a la tasa de bits requerida para transmitir el flujo de bits sin los indicadores.

En una implementación típica de la realización de las Figs. 8 y 9, en la que el flujo de bits es un flujo de bits de audio codificado AC-3, cada una de las tramas contiene contenido de audio y metadatos para 1536 muestras de audio digital. Para una frecuencia de muestreo de 48 kHz, esto representa 32 milisegundos de audio digital o una velocidad de 31,25 tramas por segundo de audio. Por lo tanto, en una realización de este tipo, un indicador de límite de programa en una trama separada por un número de tramas ("X" tramas) desde un límite de programa, indica que el límite se produce 32X milisegundos después del final de la trama que contiene el indicador (o 32X milisegundos antes del comienzo de la trama que contiene el indicador).

En una implementación típica de la realización de las Figs. 8 y 9, en la que el flujo de bits es un flujo de bits de audio codificado E-AC-3, cada una de las tramas del flujo 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 frecuencia de muestreo de 48 kHz, esto representa 5,333, 10,667, 16 o 32 milisegundos de audio digital, respectivamente, o una tasa de 189,9, 93,75, 62,5 o 31,25 tramas por segundo de audio, respectivamente. Por lo tanto, en una realización de este tipo (suponiendo que cada una de las tramas es indicativa de 32 milisegundos de audio digital), un indicador de límite de programa en una trama separada por un cierto número de tramas ("X" tramas) desde un límite de programa, indica que el límite se produce 32X milisegundos después del final de la trama que contiene el indicador (o 32X milisegundos antes del comienzo de la trama que contiene el indicador).

En algunas realizaciones en las que un límite programa puede producirse dentro de una trama de un flujo de bits de audio (es decir, no en alineación con el comienzo o el final de una trama), los metadatos de límite de programa incluidos en una trama del flujo de bits, incluyen un recuento de tramas de límite de programa (es decir, metadatos indicativos del número de tramas completas entre el comienzo o el final de la trama que contienen el recuento de tramas y un límite de programa) y un valor de desplazamiento. El valor de desplazamiento es indicativo de un desplazamiento (típicamente un número de muestras) entre el comienzo o el final de una trama que contiene el límite de programa, y la ubicación real del límite de programa dentro de la trama que contiene el límite de programa.

Un flujo de bits de audio codificado puede ser indicativo de una secuencia de programas (pistas de sonido) de una correspondiente secuencia de programas de video, y los límites de tales programas de audio tienden a producirse en los bordes de las tramas de video en lugar de en los bordes de las tramas de audio. Además, algunos códecs de audio (p. ej., códecs E-AC-3) utilizan tamaños de trama de audio que no están alineados con las tramas de video. Además, en algunos casos, un flujo de bits de audio inicialmente codificado se transcodifica para generar un flujo de bits transcodificado, y el flujo de bits codificado inicialmente tiene un tamaño de trama diferente que el flujo de bits transcodificado, de modo que no se garantiza que un límite de programa (determinado por el flujo de bits codificado inicialmente) se produzca en un límite de trama del flujo de bits transcodificado. Por ejemplo, si el flujo de bits codificado inicialmente (p. ej., el flujo de bits "IEB" de la Fig. 10) tiene un tamaño de trama de 1536 muestras por trama, y el flujo de bits transcodificado (p. ej., flujo de bits "TB" de la Fig. 10) tiene un tamaño de trama de 1024 muestras por trama, el proceso de transcodificación puede hacer que el límite real de programa no se produzca en un límite de trama del flujo de bits transcodificado sino en alguna trama del mismo (p. ej., 512 muestras en una trama del flujo de bits transcodificado, como se indica en la Fig. 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 flujo de bits de audio codificado, incluyen un valor de desplazamiento así como un recuento de tramas de límite de programa, son útiles en los tres casos señalados en este párrafo (así como en otros casos).

La realización descrita anteriormente con referencia a las Figs. 8 y 9 no incluye un valor de desplazamiento (p. ej., un campo de desplazamiento) en ninguna de las tramas del flujo de bits codificado. En variaciones de esta realización, se incluye un valor de desplazamiento en cada una de las tramas de un flujo de bits de audio codificado que incluye un indicador de límite de programa (p. ej., en tramas correspondientes a las tramas numeradas 0, 8, 12 y 14 en la Fig. 8, y tramas numeradas 18, 20, 24 y 32 en la Fig. 9).

5 En una clase de realizaciones, una estructura de datos (en cada una de las tramas de un flujo 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 recuento de tramas de límite de programa, o tanto un recuento de tramas de límite de programa como un valor de desplazamiento. Por ejemplo, el valor del código puede ser el valor de un campo de un solo bit (al que se hará referencia en el presente documento como campo un “existe\_desplazamiento”), el valor “existe\_desplazamiento” = 0 puede indicar que no se incluye un valor de desplazamiento en la trama, y el valor “existe\_desplazamiento” = 1 puede indicar que tanto el recuento de tramas de límite de programa como un valor de desplazamiento están incluidos en la trama.

10 En algunas realizaciones, al menos una trama de un flujo 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, opcionalmente, también otros metadatos) para un programa de audio determinado por el flujo de bits. Cada uno de los segmentos de metadatos (el cual puede incluirse en un campo de addbsi, o un campo de auxdata, o un segmento de bits residuales del flujo de bits) contiene una cabecera central (y, opcionalmente, también elementos centrales adicionales), y después de la cabecera central (o de la cabecera central y de otros elementos principales) un segmento de carga útil de LPSM (o contenedor) con el siguiente formato:

una cabecera (que típicamente incluye al menos un valor de identificación, p. ej., la versión del formato de LPSM, la longitud, el período, el recuento y los valores de asociación de subflujo), y

20 después de la cabecera, los metadatos de límite de programa (que pueden incluir un recuento de tramas de límite de programa, un valor de código (p. ej., un valor de “existe\_desplazamiento”) indicativo de si la trama incluye solo un recuento de tramas de límite de programa o tanto un recuento 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:

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

al menos un valor de cumplimiento de regulación de sonoridad que indica si los correspondientes datos de audio 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 realizado sobre los correspondientes datos de audio; y

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

35 En algunas realizaciones, el segmento de carga útil de LPSM incluye un valor de código (un valor de “existe\_desplazamiento”) indicativo de si la trama incluye solamente un recuento de tramas de límite de programa o tanto un recuento de tramas de límite de programa como un valor de desplazamiento. Por ejemplo, en una de tales realizaciones, cuando tal valor de código indica (p. ej., cuando existe\_desplazamiento = 1) que la trama incluye un recuento 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 sin signo de 11 bits (es decir, que tiene un valor de 0 a 2048) y que indica el número de muestras de audio adicionales entre el límite de trama señalado (el límite de la trama que incluye el límite de programa) y el límite real de programa. Si el recuento de tramas de límite de programa indica el número de tramas (a la tasa de tramas actual) a la trama que contiene el límite de programa, la ubicación precisa (en unidades de número de muestras) del límite de programa (relativo al comienzo o al final de la trama que incluye el segmento de carga útil de LPSM) se calcularía como:

$$S = (\text{contador\_de\_tramas} * \text{tamaño de trama}) + \text{desplazamiento},$$

45 donde S es el número de muestras hasta el límite de programa (desde el comienzo o el final de la trama que incluye el segmento de carga útil de LPSM), “contador\_de\_tramas” es el recuento de tramas indicado por el recuento de tramas de límite de programa, “tamaño de trama” es el número de muestras por trama, y “desplazamiento” es el número de muestras indicado por el valor de desplazamiento.

50 Algunas realizaciones en las que la tasa de inserción de indicadores de límite de programa aumenta cerca del límite real de programa, implementan una regla de que un valor de desplazamiento nunca se incluye en una trama si la trama es menor o igual que cierto número (“Y”) de tramas desde la trama que incluye el límite de programa.

Típicamente,  $Y = 32$ . Para un codificador E-AC-3 que implementa esta regla (con  $Y = 32$ ), el codificador nunca inserta un valor de desplazamiento en el último segundo de un programa de audio. En este caso, el dispositivo receptor es responsable de mantener un temporizador y, por lo tanto, realizar su propio cálculo de desplazamiento (en respuesta a los metadatos de límite de programa, incluyendo un valor de desplazamiento, en una trama del flujo de bits codificado que está a más de  $Y$  tramas desde la trama que contiene el límite del programa).

Para programas cuyos programas de audio se conocen como "alineados con tramas" con tramas de video de correspondientes programas de video (p. ej., fuentes de contribución típicas con audio codificado Dolby E), sería superfluo incluir valores de desplazamiento en los flujos de bits codificados indicativos de los programas de audio. Por lo tanto, los valores de desplazamiento típicamente no se incluirán en tales flujos de bits codificados.

Con referencia a la Fig. 11, a continuación consideramos casos en los que los flujos de bits de audio codificados se empalman juntos para generar una realización del flujo de bits de audio inventivo.

El flujo de bits en la parte superior de la FIG. 11 (etiquetado "Escenario 1") es indicativo de un primer programa de audio (P1) completo incluyendo metadatos de límite de programa (indicadores de límite de programa, F) seguidos de un segundo programa de audio (P2) completo que también incluye metadatos de límite de programa (indicadores de límite de programa, F). Los indicadores de límites de programa en la parte final del primer programa (algunos de los cuales se muestran en la Fig. 11) son idénticos o similares a los descritos con referencia a la Fig. 8, y determinan la ubicación del límite entre los dos programas (es decir, el límite al comienzo del segundo programa). Los indicadores de límite de programa en la parte inicial del segundo programa (algunos de los cuales se muestran en la Fig. 11) son idénticos o similares a los descritos con referencia a la Fig. 9, y también determinan la ubicación del límite. En realizaciones típicas, un codificador o decodificador implementa un temporizador (calibrado por los indicadores en el primer programa) que realiza una cuenta hacia abajo hasta el límite de programa, y el mismo temporizador (calibrado por los indicadores en el segundo programa) realiza una cuenta hacia arriba desde el mismo límite de programa. Como se indica por el gráfico de temporizador de límite en el Escenario 1 de la Fig. 11, una cuenta hacia abajo de este tipo del temporizador (calibrada por indicadores en el primer programa) llega a cero en el límite, y la cuenta hacia arriba del temporizador (calibrado por indicadores en el segundo programa) indica que la misma ubicación del límite.

El segundo flujo de bits desde la parte superior de la FIG. 11 (etiquetado "Escenario 2") es indicativo de un primer programa de audio (P1) completo que incluye metadatos de límite de programa (indicadores de límite de programa, F) seguidos por un segundo programa de audio (P2) completo que no incluye metadatos de límite de programa. Los indicadores de límite de programa en la parte final del primer programa (algunos de los cuales se muestran en la Fig. 11) son idénticos o similares a los descritos con referencia a la Fig. 8, y determinan la ubicación del límite entre los dos programas (es decir, el límite al comienzo del segundo programa), tal como en el Escenario 1. En realizaciones típicas, un codificador o decodificador implementa un temporizador (calibrado por los indicadores en el primer programa) que realiza la cuenta hacia abajo hasta el límite de programa, y el mismo temporizador (sin ser calibrado adicionalmente) continúa la cuenta hacia arriba desde el límite de programa (como lo indica el gráfico de temporizador de límite en el Escenario 2 de la Fig. 11).

El tercer flujo de bits desde la parte superior de la FIG. 11 (etiquetado "Escenario 3") es indicativo de un primer programa de audio (P1) truncado que incluye metadatos de límite de programa (indicadores de límite de programa, F) y que se ha empalmado con un segundo programa de audio (P2) completo que también incluye metadatos de límite de programa (indicadores de límites de programa, F). El empalme ha eliminado las últimas "N" tramas del primer programa. Los indicadores de límite de programa en la parte inicial del segundo programa (algunos de los cuales se muestran en la Fig. 11) son idénticos o similares a los descritos con referencia a la Fig. 9, y determinan la ubicación del límite (empalme) entre el primer programa truncado y el segundo programa completo. En realizaciones típicas, un codificador o decodificador implementa un temporizador (calibrado por los indicadores en el primer programa) que cuenta hacia abajo hasta el final del primer programa no truncado, y el mismo temporizador (calibrado por los indicadores en el segundo programa) cuenta hacia arriba 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 Fig. 11, tal cuenta hacia abajo del temporizador (calibrada por los metadatos de límite de programa en el primer programa) se reinicia (en respuesta a los metadatos de límite de programa en el segundo programa) antes de que hubiera llegado a cero (en respuesta a los metadatos de límite de programa en el primer programa). Por lo tanto, aunque el truncamiento del primer programa (por el empalme) impide que el temporizador identifique el límite de programa entre el primer programa truncado y el comienzo del segundo programa en respuesta a (es decir, bajo calibración por) metadatos de límite de programa en el primer programa solo, los metadatos del programa en el segundo programa reinician el temporizador, de modo que el temporizador de reinicio indica correctamente (como la ubicación correspondiente al recuento de "cero" del temporizador de reinicio) la ubicación del límite del programa entre el primer programa truncado y el comienzo del segundo programa.

5 El cuarto flujo de bits (etiquetado "Escenario 4") es indicativo de un primer programa de audio (P1) truncado que incluye metadatos de límite de programa (indicadores de límite de programa, F) y un segundo programa de audio (P2) truncado que incluye metadatos de límite de programa (indicadores de límite de programa, F) y que se ha empalmado con una parte (la parte no truncada) del primer programa de audio. Los indicadores de límite de programa en la parte inicial del segundo programa completo (previo al truncamiento) (algunos de los cuales se muestran en la Fig. 11), son idénticos o similares a los descritos con referencia a la Fig. 9, y los indicadores de límite de programa en el la parte final del primer programa completo (previo al truncamiento) (algunos de los cuales se muestran en la Fig. 11), son idénticos o similares a los descritos con referencia a la Fig. 8. El empalme ha eliminado las última "N" tramas del primer programa (y, por lo tanto, algunos de los indicadores de límite de programa que se habían incluido antes del empalme) y las primeras "M" tramas del segundo programa (y por lo tanto algunos de los indicadores de límite de programa que se incluyeron antes del empalme). En realizaciones típicas, un codificador o decodificador implementa un temporizador (calibrado por los indicadores en el primer programa truncado) que cuenta hacia abajo hacia el final del primer programa no truncado, y el mismo temporizador (calibrado por los indicadores en el segundo programa truncado) cuenta hacia arriba desde el comienzo del segundo programa no truncado. Como se indica por el gráfico de temporizador de límite en el Escenario 4 de la Fig. 11, tal cuenta hacia abajo del temporizador (calibrada por los metadatos de límite de programa en el primer programa) se reinicia (en respuesta a los metadatos de límite de programa en el segundo programa) antes de que hubiera llegado a 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 (es decir, bajo calibración por) metadatos de límite de programa en el primer programa solo. Sin embargo, el temporizador de reinicio no indica correctamente la ubicación del límite de programa entre el final del primer programa truncado y el comienzo del segundo programa truncado. Por lo tanto, el truncamiento de ambos flujos de bits empalmados puede evitar la determinación precisa del límite entre ellos.

25 Las realizaciones de la presente invención pueden implementarse en hardware, firmware o software, o en una combinación de ambos (p. ej., como una matriz lógica programable). A menos que se especifique lo contrario, los algoritmos o procesos incluidos como parte de la invención no están intrínsecamente relacionados con ninguna computadora particular u otro aparato. En particular, se pueden utilizar diversas máquinas de propósito general con programas escritos de acuerdo con las enseñanzas del presente documento, o puede ser más conveniente construir aparatos más especializados (p. ej., circuitos integrados) para realizar los pasos del método requeridos. Por lo tanto, la invención puede implementarse en uno o más programas informáticos que se ejecutan en uno o más sistemas informáticos programables (p. ej., una implementación de cualquiera de los elementos de la Fig. 1, o el codificador 100 de la Fig. 2 (o un elemento del mismo), o el decodificador 200 de la Fig. 3 (o un elemento del mismo), o postprocesador 300 de la Fig. 3 (o un elemento del mismo) comprendiendo cada uno al menos un procesador, al menos un sistema de almacenamiento de datos (incluyendo memoria volátil y no volátil y/o elementos de almacenamiento), al menos un dispositivo o puerto de entrada, y al menos un dispositivo o puerto de salida. El código de programa se aplica a datos de entrada para realizar las funciones descritas en el presente documento y generar información de salida. La información de salida se aplica a uno o más dispositivos de salida, de manera conocida.

40 Cada uno de tales programas puede implementarse en cualquier lenguaje informático deseado (incluyendo lenguajes máquina, ensamblador o procedimentales, lógicos u orientados a objetos de alto nivel) para comunicarse con un sistema informático. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado.

45 Por ejemplo, cuando se implementa por secuencias de instrucciones de software informático, diversas funciones y pasos de realizaciones de la invención pueden implementarse por secuencias de instrucciones de software multihilo que funcionan en hardware de procesamiento de señal digital adecuado, en cuyo caso los diversos dispositivos, pasos y funciones de las realizaciones pueden corresponder a partes de las instrucciones del software.

50 Cada uno de tales programas informáticos se almacena o descarga preferiblemente en un medio de almacenamiento o dispositivo (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 operar el ordenador cuando el sistema informático lee el dispositivo o medios de almacenamiento para realizar los procedimientos descritos en el presente documento. El sistema inventivo también puede implementarse como un medio de almacenamiento legible por ordenador, configurado con (es decir, que almacena) un programa informático, donde el medio de almacenamiento así configurado hace que un sistema informático opere de manera específica y predefinida para realizar las funciones descritas en el presente documento.

55 Se han descrito varias realizaciones de la invención. Sin embargo, se entenderá que pueden realizarse diversas modificaciones sin apartarse del alcance de la invención. Numerosas modificaciones y variaciones de la presente invención son posibles a la luz de las enseñanzas anteriores. Debe entenderse que dentro del alcance de las



reivindicaciones adjuntas, la invención puede practicarse de otra manera que la específicamente descrita en el presente documento.

**REIVINDICACIONES**

1. Un aparato para procesar una señal de audio, comprendiendo el aparato:

5 un búfer de entrada para almacenar al menos una trama de un flujo de bits de audio codificado, en donde el flujo de bits de audio codificado incluye datos de audio y un segmento de metadatos, y el segmento de metadatos incluye una cabecera, una o más cargas útiles de metadatos y datos de protección;

un decodificador de audio acoplado al búfer de entrada para decodificar los datos de audio para producir datos de audio decodificados;

un analizador acoplado a o integrado con el decodificador de audio, en donde el analizador está configurado para analizar el flujo de bits de audio codificado; y

10 un búfer de salida para almacenar los datos de audio decodificados,

15 en donde la cabecera del segmento de metadatos incluye una palabra de sincronización que identifica el inicio del segmento de metadatos, la una o más cargas útiles de metadatos incluye metadatos indicativos de al menos un límite entre programas de audio consecutivos, los datos de protección están ubicados después de la una o más cargas útiles de metadatos, y los datos de protección se pueden utilizar para verificar la integridad del segmento de metadatos y de la una o más cargas útiles dentro del segmento de metadatos.

2. El aparato de la reivindicación 1, en donde el segmento de metadatos se almacena en un espacio de datos reservado AC-3 o E-AC-3 seleccionado del grupo que consiste en un campo de omisión, un campo de auxdata, un campo de addbsi y una combinación de los mismos.

20 3. El aparato de cualquier reivindicación precedente, en donde la una o más cargas útiles de metadatos incluye una carga útil de sonoridad de programa que contiene datos indicativos de una sonoridad medida de un programa de audio.

4. El aparato de la reivindicación 3, en donde la carga útil de sonoridad de programa incluye un campo que indica si un canal de audio contiene diálogo hablado.

25 5. El aparato de la reivindicación 3 o la reivindicación 4, en donde la carga útil de sonoridad de programa incluye un campo que indica un método de medición de sonoridad que se ha utilizado para generar datos de sonoridad contenidos en la carga útil de sonoridad de programa.

6. El aparato de una cualquiera de las reivindicaciones 3 a 5, en donde el aparato está configurado para realizar un procesamiento de sonoridad adaptativo utilizando la carga útil de sonoridad de programa.

30 7. El aparato de cualquier reivindicación precedente, en donde el flujo de bits de audio codificado es un flujo de bits AC-3 o un flujo de bits E-AC-3.

8. El aparato de cualquier reivindicación precedente, en donde cada uno de los identificadores de carga útil es único y se encuentra al comienzo de cada una de las cargas útiles de metadatos.

9. El aparato de cualquier reivindicación precedente, en donde la palabra de sincronización es una palabra de sincronización de 16 bits que tiene un valor de 0x5838.

35 10. Un decodificador de audio que comprende:

un búfer de entrada para almacenar al menos una trama de un flujo de bits de audio codificado, en donde el flujo de bits de audio codificado incluye datos de audio y un segmento de metadatos, y el segmento de metadatos incluye una cabecera, una o más cargas útiles de metadatos y datos de protección;

40 un analizador configurado para analizar el flujo de bits de audio codificado, en donde la cabecera del segmento de metadatos incluye una palabra de sincronización que identifica el inicio del segmento de metadatos, los datos de protección están ubicados después de la una o más cargas útiles de metadatos, la una o más cargas útiles de metadatos incluyen metadatos indicativos de al menos un límite entre programas de audio consecutivos, y los

datos de protección pueden utilizarse para verificar la integridad del segmento de metadatos y la una o más cargas útiles dentro del segmento de metadatos; y

un búfer de salida para almacenar datos de audio decodificados.

- 5 11. El decodificador de audio de la reivindicación 10, en donde el segmento de metadatos se almacena en un espacio de datos reservados AC-3 o E-AC-3 seleccionado del grupo que consiste en un campo de omisión, un campo de auxdata, un campo de addbsi y una combinación de los mismos.
12. El decodificador de audio de una cualquiera de las reivindicaciones 10 a 11, en donde el flujo de bits de audio codificado es un flujo de bits AC-3 o un flujo de bits E-AC-3.
- 10 13. El decodificador de audio de una cualquiera de las reivindicaciones 10 a 12, en donde la una o más cargas útiles de metadatos incluye una carga útil de sonoridad de programa que contiene datos indicativos de una sonoridad medida de un programa de audio.
14. El decodificador de audio de una cualquiera de las reivindicaciones 10 a 13, en donde la palabra de sincronización es una palabra de sincronización de 16 bits que tiene un valor de 0x5838.

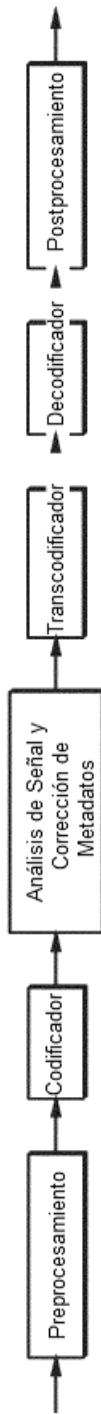


FIG. 1

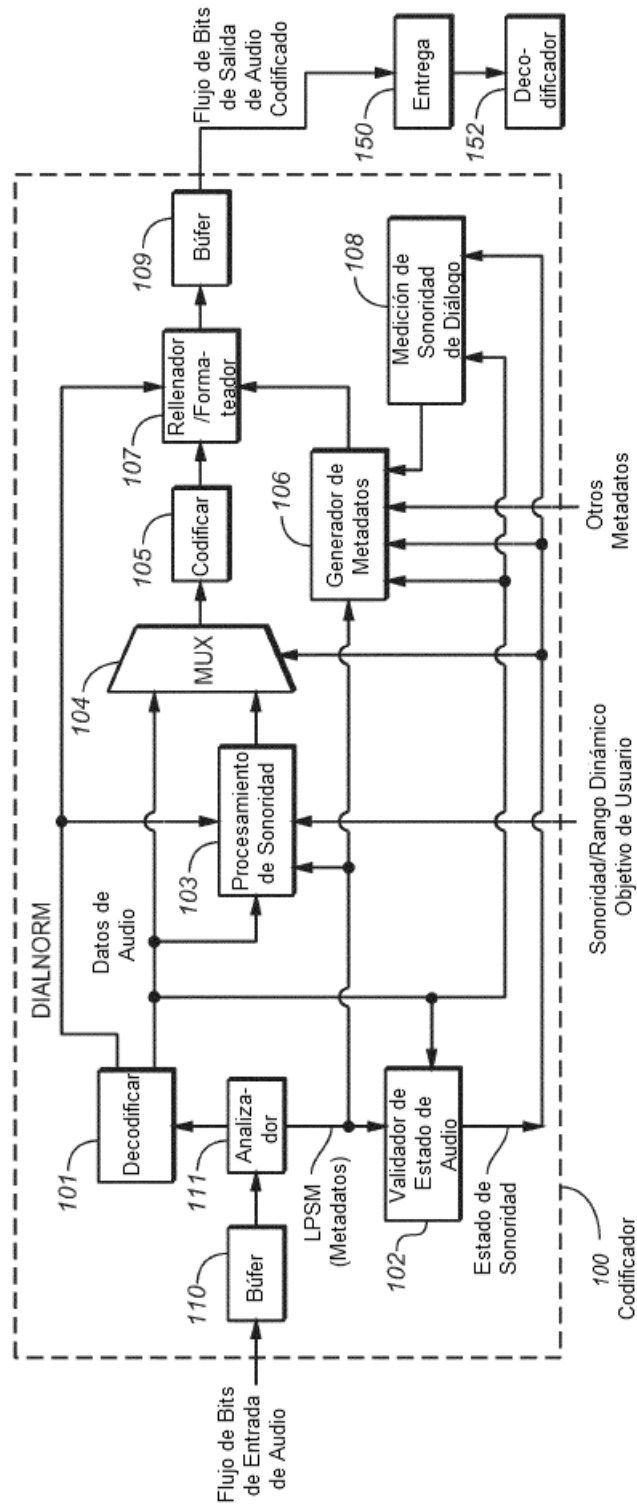


FIG. 2

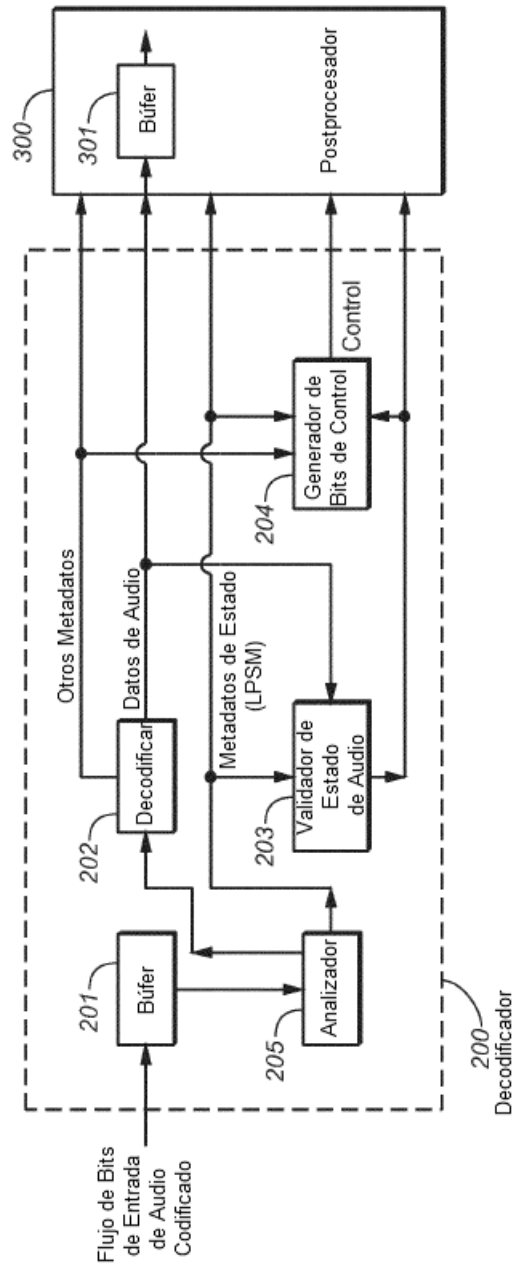
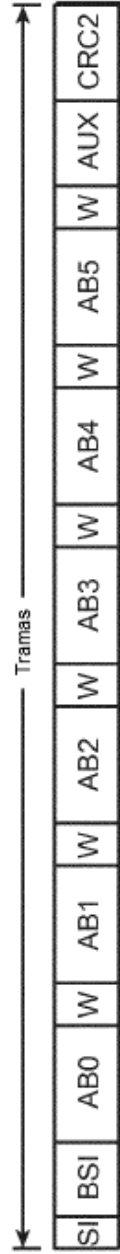
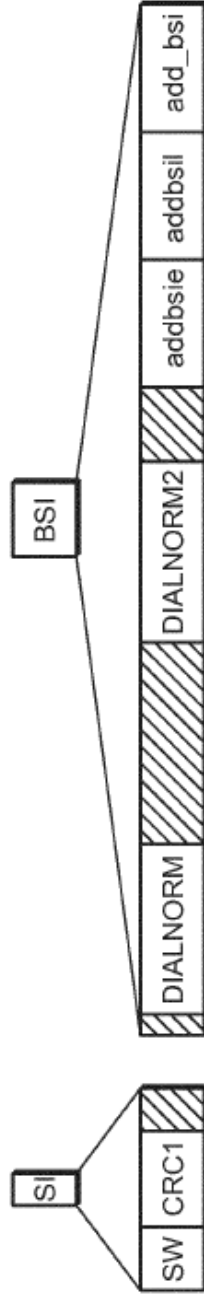


FIG. 3



**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
Tramas hasta Limite	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
Contenido de Indicador de Limite de Programa	100011	0	0	0	0	0	0	0	10011	0	0	0	1011	0	111	0
Existe	1	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0
Contador de Tramas	0001->16								001->8				01->4		1->2	
Cuenta hacia arriba (+) / hacia abajo (-)	1								1						1	

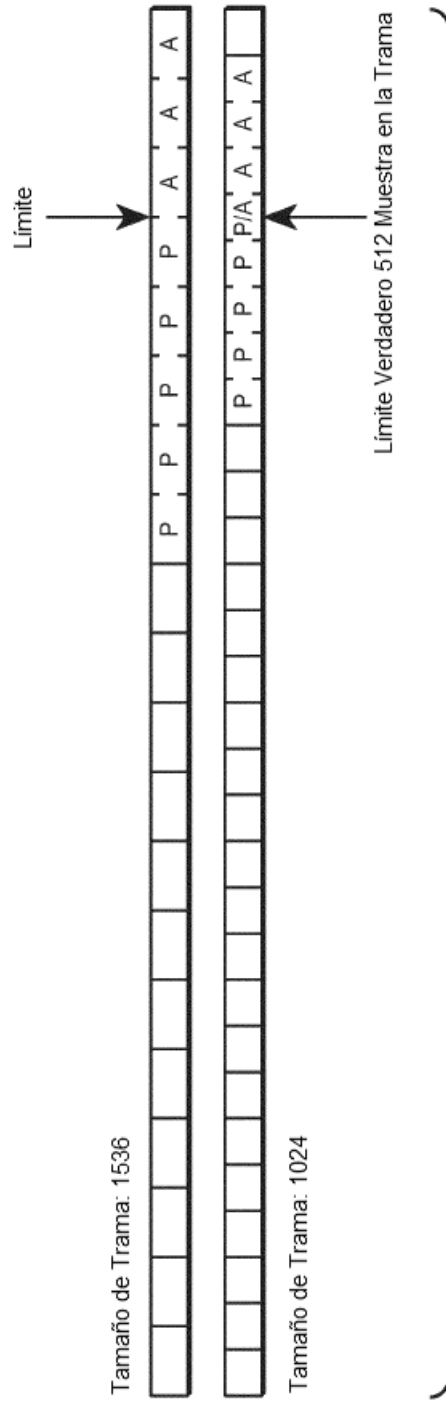
(Sintaxis de Indicador de Limite de Programa Contando Hacia Abajo hasta el Limite de Programa)

**FIG. 8**

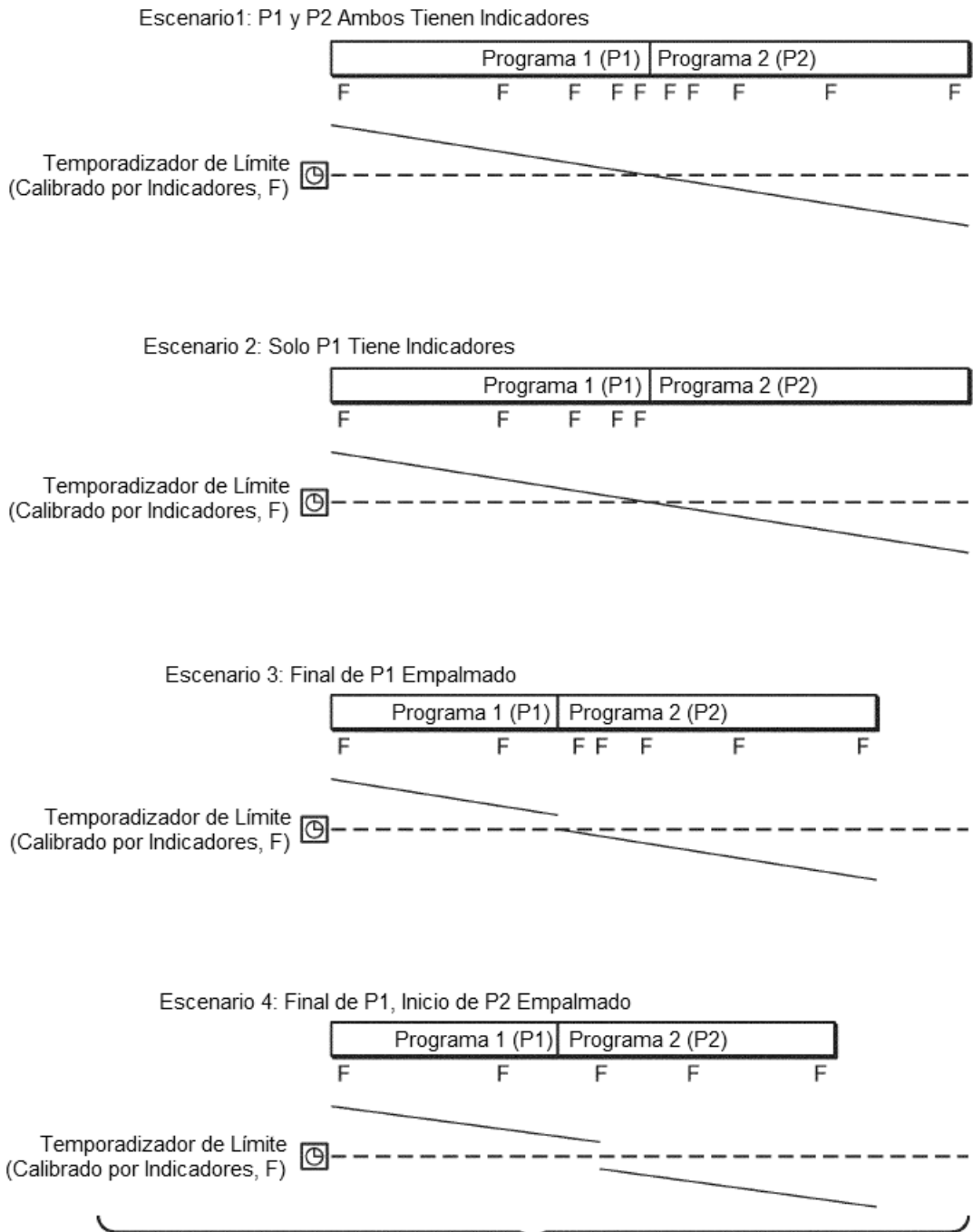
# Trama	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Tramas hasta Limite	B	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Contenido de Indicador de Limite de Programa	B	0	110	0	1010	0	0	0	10010	0	0	0	0	0	0	10010
Existe	B	0	1	0	0	0	0	0	1	0	0	0	0	0	0	
Contador de Tramas	B	1->2		01->4					001->8							0001->16
Cuenta hacia arriba (+) / hacia abajo (-)	B	0		0					0							0

(Sintaxis de Indicador de Limite de Programa Contando Hacia Arriba para el Comienzo de Programa)

**FIG. 9**







**FIG. 11**