

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 948 839**

51 Int. Cl.:

G10L 21/038 (2013.01)

G10L 19/00 (2013.01)

G10L 19/18 (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **28.01.2019** **E 22181854 (5)**

97 Fecha y número de publicación de la concesión europea: **17.05.2023** **EP 4099325**

54 Título: **Integración retrocompatible de técnicas de reconstrucción de alta frecuencia para señales de audio**

30 Prioridad:

26.01.2018 EP 18153683

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

20.09.2023

73 Titular/es:

DOLBY INTERNATIONAL AB (100.0%)
77 Sir John Rogerson's Quay Block C Grand
Canal Docklands
Dublin, D02 VK60, IE

72 Inventor/es:

KJOERLING, KRISTOFER;
VILLEMOES, LARS;
PUHNHAGEN, HEIKO y
EKSTRAND, PER

74 Agente/Representante:

LINAGE GONZÁLEZ, Rafael

ES 2 948 839 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

Integración retrocompatible de técnicas de reconstrucción de alta frecuencia para señales de audio

5 Referencia cruzada a solicitud relacionada

Esta solicitud es una solicitud divisional europea de la solicitud de patente europea EP 21164481.0 (referencia: D17806EP02), cuyo formulario 1001 de la OEP fue presentado el 24 de marzo de 2021.

10 Campo técnico

Las realizaciones pertenecen a un método y una unidad de procesamiento de audio para realizar reconstrucción de alta frecuencia de una señal de audio.

15 Antecedentes de la invención

Un flujo de bits de audio típico incluye tanto datos de audio (por ejemplo, datos de audio codificados) indicativos de uno o más canales de contenido de audio, como metadatos indicativos de al menos una característica de los datos de audio o del contenido de audio. Un formato bien conocido para generar un flujo de bits de audio codificado es el formato de codificación de audio avanzada (AAC) MPEG-4, descrito en el estándar MPEG ISO/IEC 14496-3: 2009. En el estándar MPEG-4, AAC denota "codificación de audio avanzada" y HE-AAC denota "codificación de audio avanzada de alta eficiencia".

El estándar MPEG-4 AAC define varios perfiles de audio, que determinan qué objetos y herramientas de codificación están presentes en un codificador o decodificador compatible. Tres de estos perfiles de audio son (1) el perfil AAC, (2) el perfil HE-AAC y (3) el perfil HE-AAC v2. El perfil AAC incluye el tipo de objeto AAC de baja complejidad (o "AAC-LC"). El objeto AAC-LC es la contraparte del perfil de baja complejidad MPEG-2 AAC, con algunos ajustes, y no incluye ni el tipo de objeto de replicación de banda espectral ("SBR") ni el tipo de objeto estéreo paramétrico ("PS"). El perfil HE-AAC es un superconjunto del perfil AAC y además incluye el tipo de objeto SBR. El perfil HE-AAC v2 es un superconjunto del perfil HE-AAC y además incluye el tipo de objeto PS.

El tipo de objeto SBR contiene la herramienta de replicación de banda espectral, que es una importante herramienta de codificación de reconstrucción de alta frecuencia ("HFR") que mejora significativamente la eficiencia de compresión de los códecs de audio de percepción. SBR reconstruye los componentes de alta frecuencia de una señal de audio en el lado del receptor (por ejemplo, en el decodificador). Por lo tanto, el codificador solo necesita codificar y transmitir componentes de baja frecuencia, lo que permite una calidad de audio mucho mayor a tasas de datos bajas. SBR se basa en la replicación de las secuencias de armónicos, previamente truncadas para reducir la tasa de datos, a partir de la señal limitada de ancho de banda disponible y los datos de control obtenidos del codificador. La relación entre los componentes tonales y similares al ruido se mantiene mediante el filtrado inverso adaptativo, así como la adición opcional de ruido y sinusoidales. En el estándar MPEG-4 AAC, la herramienta SBR realiza el parcheo espectral (también llamado traslación lineal o traslación espectral), donde se copian (o se "parchean") varias subbandas consecutivas del filtro de espejo en cuadratura (QMF) desde una porción de banda baja transmitida de una señal de audio a una porción de banda alta de la señal de audio, que se genera en el decodificador.

El parcheo espectral o la traslación lineal pueden no ser ideales para ciertos tipos de audio, como el contenido musical con frecuencias de cruce relativamente bajas. Por lo tanto, se necesitan técnicas para mejorar la replicación de la banda espectral.

El documento WO 2016/149015 A1 describe una unidad de procesamiento de audio que incluye un búfer, un desformateador de carga útil de flujo de bits y un subsistema de decodificación. El búfer almacena al menos un bloque de un flujo de bits de audio codificado. El bloque incluye un elemento de relleno que comienza con un identificador seguido de datos de relleno. Los datos de relleno incluyen al menos un indicador que identifica si se va a realizar el procesamiento de replicación de banda espectral mejorada (eSBR) en el contenido de audio del bloque. También se proporciona un método correspondiente para decodificar un flujo de bits de audio codificado.

Sumario de la invención

La invención se define mediante las reivindicaciones independientes. Las realizaciones específicas se definen en las reivindicaciones dependientes.

Breve descripción de las realizaciones de la invención

Una primera clase de realizaciones se refiere a un método de acuerdo con la reivindicación 1.

Una segunda clase de realizaciones se refiere a una unidad de procesamiento de audio de acuerdo con la reivindicación 7.

Breve descripción de los dibujos

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

La figura 2 es un diagrama de bloques de un codificador que es una realización de la unidad de procesamiento de audio de la invención.

La figura 3 es un diagrama de bloques de un sistema que incluye un decodificador que es una realización de la unidad de procesamiento de audio de la invención y, opcionalmente, también un posprocesador acoplado a la misma.

La figura 4 es un diagrama de bloques de un decodificador que es una realización de la unidad de procesamiento de audio de la invención.

La figura 5 es un diagrama de bloques de un decodificador que es otra realización de la unidad de procesamiento de audio de la invención.

La figura 6 es un diagrama de bloques de otra realización de la unidad de procesamiento de audio de la invención.

La figura 7 es un diagrama de un bloque de un flujo de bits MPEG-4 AAC, incluidos los segmentos en los que se divide.

Nomenclatura y notación

A lo largo de esta divulgación, incluidas las reivindicaciones, la expresión realizar una operación "en" una señal o datos (por ejemplo, filtrar, escalar, transformar o aplicar ganancia a la señal o a los datos) se usa en un sentido amplio para denotar la realización de la operación directamente en la señal o datos, o en una versión procesada de la señal o datos (por ejemplo, en una versión de la señal que ha sido sometida a un filtrado preliminar o preprocesamiento antes de la realización de la operación en los mismos).

A lo largo de esta divulgación, incluidas las reivindicaciones, la expresión "unidad de procesamiento de audio" o "procesador de audio" se usa en un sentido amplio, para indicar un sistema, dispositivo o aparato, configurado para procesar datos de audio. Los ejemplos de unidades de procesamiento de audio incluyen, entre otros, codificadores, transcodificadores, decodificadores, códecs, sistemas de preprocesamiento, sistemas de posprocesamiento y sistemas de procesamiento de flujo de bits (a veces denominados herramientas de procesamiento de flujo de bits). Prácticamente todos los productos electrónicos de consumo, como teléfonos móviles, televisores, computadoras portátiles y tabletas, contienen una unidad de procesamiento de audio o un procesador de audio.

A lo largo de esta divulgación, incluidas las reivindicaciones, el término "acopla" o "acoplado" se usa en un sentido amplio para significar 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 mediante una conexión indirecta a través de otros dispositivos y conexiones. Además, los componentes que están integrados en o con otros componentes también están acoplados entre sí.

Descripción detallada de las realizaciones de la invención

El estándar MPEG-4 AAC contempla que un flujo de bits MPEG-4 AAC codificado incluye metadatos indicativos de cada tipo de procesamiento de reconstrucción de alta frecuencia ("HFR") que se aplicará (si se aplica alguno) por un decodificador para decodificar el contenido de audio del flujo de bits, y/o que controla tal procesamiento HFR, y/o es indicativo de al menos una característica o parámetro de al menos una herramienta HFR que se usará para decodificar el contenido de audio del flujo de bits. En el presente documento, se usa la expresión "metadatos SBR" para denotar metadatos de este tipo que se describen o mencionan en el estándar MPEG-4 AAC para su uso con la replicación de banda espectral ("SBR"). Como apreciará un experto en la técnica, SBR es una forma de HFR.

SBR se usa preferiblemente como un sistema de tasa dual, con el códec subyacente operando a la mitad de la tasa de muestreo original, mientras que SBR opera a la tasa de muestreo original. El codificador SBR funciona en paralelo con el códec central subyacente, aunque a una frecuencia de muestreo más alta. Aunque SBR es principalmente un proceso posterior en el decodificador, se extraen parámetros importantes en el codificador con el fin de garantizar la reconstrucción de alta frecuencia más precisa en el decodificador. El codificador estima la envolvente espectral del rango SBR para un rango/resolución de tiempo y frecuencia adecuados para las características de los segmentos de la señal de entrada actual. La envolvente espectral se estima mediante un análisis QMF complejo y un cálculo de energía posterior. Las resoluciones de tiempo y frecuencia de las envolventes

espectrales se pueden elegir con un alto nivel de libertad, con el fin de garantizar la resolución de frecuencia de tiempo más adecuada para el segmento de entrada dado. La estimación de la envolvente debe tener en cuenta que un transitorio en el original, principalmente situado en la región de alta frecuencia (por ejemplo, una zona alta), estará presente en menor medida en la banda alta generada por SBR antes del ajuste de la envolvente, ya que la

banda alta en el decodificador se basa en la banda baja donde el transitorio es mucho menos pronunciado en comparación con la banda alta. Este aspecto impone diferentes requisitos para la resolución de tiempo frecuencia de los datos de la envolvente espectral, en comparación con la estimación de la envolvente espectral ordinaria como se usa en otros algoritmos de codificación de audio.

Además de la envolvente espectral, se extraen varios parámetros adicionales que representan las características espectrales de la señal de entrada para diferentes regiones de tiempo y frecuencia. Dado que el codificador naturalmente tiene acceso a la señal original, así como a información sobre cómo la unidad SBR en el decodificador creará la banda alta, dado el conjunto específico de parámetros de control, es posible que el sistema maneje situaciones en las que la banda baja constituye una serie de armónicos fuertes y la banda alta, que se va a recrear, constituye principalmente componentes de señal aleatorios, así como situaciones en las que están presentes componentes tonales fuertes en la banda alta original sin contrapartes en la banda baja, sobre la que se basa la región de banda alta. Además, el codificador SBR trabaja en estrecha relación con el códec central subyacente para evaluar qué rango de frecuencia debería ser cubierto por SBR en un momento dado. Los datos SBR se codifican eficazmente antes de la transmisión explotando la codificación de entropía, así como las dependencias de canal de los datos de control, en el caso de señales estéreo.

Típicamente, los algoritmos de extracción de parámetros de control deben ajustarse cuidadosamente al códec subyacente a una tasa de bits determinada y una frecuencia de muestreo determinada. Esto se debe al hecho de que una tasa de bits más baja, generalmente implica un rango SBR más grande en comparación con una tasa de bits alta, y diferentes tasas de muestreo corresponden a diferentes resoluciones de tiempo de las tramas SBR.

Un decodificador SBR típicamente incluye varias partes diferentes. Comprende un módulo de decodificación de flujo de bits, un módulo de reconstrucción de alta frecuencia (HFR), un módulo de componentes de alta frecuencia adicional y un módulo de ajuste de envolvente. El sistema se basa en un banco de filtros QMF de valor complejo (para SBR de alta calidad) o un banco de filtros QMF de valor real (para SBR de baja potencia). Las realizaciones de la invención son aplicables tanto a SBR de alta calidad como a SBR de baja potencia. En el módulo de extracción de flujo de bits, los datos de control se leen del flujo de bits y se decodifican. La cuadrícula de tiempo frecuencia se obtiene para la trama actual, antes de leer los datos de la envolvente del flujo de bits. El decodificador central subyacente decodifica la señal de audio de la trama actual (aunque a la tasa de muestreo más baja) para producir muestras de audio en el dominio tiempo. La trama resultante de datos de audio se usa para la reconstrucción de alta frecuencia por el módulo HFR. La señal de banda baja decodificada se analiza luego usando un banco de filtros QMF. La reconstrucción de alta frecuencia y el ajuste de envolvente se realizan posteriormente en las muestras de subbanda del banco de filtros QMF. Las altas frecuencias se reconstruyen a partir de la banda baja de forma flexible, basándose en los parámetros de control dados. Además, la banda alta reconstruida se filtra de forma adaptativa sobre una base de canal de subbanda de acuerdo con los datos de control para asegurar las características espectrales apropiadas de la región de tiempo/frecuencia dada.

El nivel superior de un flujo de bits MPEG-4 AAC es una secuencia de bloques de datos (elementos "raw_data_block"), cada uno de los cuales es un segmento de datos (en el presente documento denominado "bloque") que contiene datos de audio (típicamente durante un tiempo período de 1024 o 960 muestras) e información relacionada y/u otros datos. En el presente documento, se usa el término "bloque" para denotar un segmento de un flujo de bits MPEG-4 AAC que comprende datos de audio (y los metadatos correspondientes y, opcionalmente, también otros datos relacionados) que determina o es indicativo de un elemento "raw_data_block" (pero no más de uno).

Cada bloque de un flujo de bits MPEG-4 AAC puede incluir varios elementos sintácticos (cada uno de los cuales también se materializa en el flujo de bits como un segmento de datos). En el estándar MPEG-4 AAC se definen siete tipos de tales elementos sintácticos. Cada elemento sintáctico se identifica por un valor diferente del elemento de datos "id_syn_ele". Los ejemplos de elementos sintácticos incluyen un "single_channel_element()", un "channel_pair_element()" y un "fill_element()". Un elemento de un solo canal es un contenedor que incluye datos de audio de un solo canal de audio (una señal de audio monofónica). Un elemento de par de canales incluye datos de audio de dos canales de audio (es decir, una señal de audio estéreo).

Un elemento de relleno es un contenedor de información que incluye un identificador (por ejemplo, el valor del elemento "id_syn_ele" indicado anteriormente) seguido de datos, que se denomina "datos de relleno". Los elementos de relleno se han usado históricamente para ajustar la tasa de bits instantánea de los flujos de bits que se transmitirán a través de un canal de tasa constante. Agregando la cantidad apropiada de datos de relleno a cada bloque, se puede lograr una tasa de datos constante.

De acuerdo con las realizaciones de la invención, los datos de relleno pueden incluir una o más cargas útiles de extensión que amplían el tipo de datos (por ejemplo, metadatos) que pueden transmitirse en un flujo de bits. Un

decodificador que recibe flujos de bits con datos de relleno que contienen un nuevo tipo de datos puede ser usado opcionalmente por un dispositivo que recibe el flujo de bits (por ejemplo, un decodificador) para ampliar la funcionalidad del dispositivo. Por tanto, como puede apreciar un experto en la técnica, los elementos de relleno son un tipo especial de estructura de datos y son diferentes de las estructuras de datos que se usan típicamente para transmitir datos de audio (por ejemplo, cargas útiles de audio que contienen datos de canal).

En algunas realizaciones de la invención, el identificador usado para identificar un elemento de relleno puede consistir en un entero sin signo de tres bits transmitido primero el bit más significativo ("uimbsb") que tiene un valor de 0x6. En un bloque, pueden ocurrir varias instancias del mismo tipo de elemento sintáctico (por ejemplo, varios elementos de relleno).

Otro estándar para codificar flujos de bits de audio es el estándar de codificación de audio y voz unificado (USAC) MPEG (ISO/IEC 23003-3: 2012). El estándar USAC MPEG describe la codificación y decodificación de contenido de audio usando procesamiento de replicación de banda espectral (incluido el procesamiento SBR como se describe en el estándar MPEG-4 AAC, y también incluye otras formas mejoradas de procesamiento de replicación de banda espectral). Este procesamiento aplica herramientas de replicación de banda espectral (a veces denominadas en el presente documento "herramientas SBR mejoradas" o "herramientas eSBR") de una versión ampliada y mejorada del conjunto de herramientas SBR descritas en el estándar MPEG-4 AAC. Por lo tanto, eSBR (como se define en el estándar USAC) es una mejora de SBR (como se define en el estándar MPEG-4 AAC).

En el presente documento, se usa la expresión "procesamiento de SBR mejorada" (o "procesamiento eSBR") para denotar el procesamiento de replicación de banda espectral usando al menos una herramienta eSBR (por ejemplo, al menos una herramienta eSBR que se describe o menciona en el estándar USAC MPEG) que no se describe ni se menciona en el estándar MPEG-4 AAC. Ejemplos de tales herramientas eSBR son la transposición armónica y el preprocesamiento adicional de parcheo QMF o "preaplanamiento".

Un transpondedor armónico de orden entero T mapea una senoide con frecuencia ω en una senoide con frecuencia $T\omega$, mientras preserva la duración de la señal. Típicamente, se usan tres órdenes, $T = 2, 3, 4$ en secuencia para producir cada parte del rango de frecuencia de salida deseado usando el orden de transposición más pequeño posible. Si se requiere una salida por encima del rango de transposición de cuarto orden, puede generarse mediante cambios de frecuencia. Cuando es posible, se crean dominios tiempo de banda base muestreados casi críticamente para que el procesamiento minimice la complejidad computacional.

El transpondedor armónico puede estar basado en QMF o DFT. Cuando se usa el transpondedor armónico basado en QMF, la extensión del ancho de banda de la señal de dominio tiempo del codificador central se lleva a cabo completamente en el dominio QMF, usando una estructura de codificador de voz de fase modificada, realizando decimación seguida de estiramiento de tiempo para cada subbanda QMF. La transposición usando varios factores de transposiciones (por ejemplo, $T = 2, 3, 4$) se lleva a cabo en una etapa común de transformación de análisis/síntesis QMF. Dado que el transpondedor armónico basado en QMF no presenta sobremuestreo de dominio frecuencia adaptativo de señal, el indicador correspondiente en el flujo de bits (`sbrOversamplingFlag [ch]`) puede ignorarse.

Cuando se usa el transpondedor armónico basado en DFT, los transpondedores de factor 3 y 4 (transpondedores de 3^{er} y 4^o orden) se integran preferiblemente en el transpondedor de factor 2 (transpondedor de 2^o orden) por medio de interpolación para reducir la complejidad. Para cada trama (correspondiente a las muestras del codificador central `coreCoderFrameLength`), el tamaño de transformación nominal de "tamaño completo" del transpondedor se determina primero mediante el indicador de sobremuestreo en el dominio frecuencia adaptativa de la señal (`sbrOversamplingFlag [ch]`) en el flujo de bits.

Cuando `sbrPatchingMode == 1`, que indica que se va a usar la transposición lineal para generar la banda alta, se puede introducir un paso adicional para evitar discontinuidades en la forma de la envolvente espectral de la señal de alta frecuencia que se introduce en el ajustador de envolvente posterior. Esto mejora el funcionamiento de la siguiente etapa de ajuste de la envolvente, lo que da como resultado una señal de banda alta que se percibe como más estable. El funcionamiento del preprocesamiento adicional es beneficioso para los tipos de señal en los que la envolvente espectral gruesa de la señal de banda baja que se usa para la reconstrucción de alta frecuencia muestra grandes variaciones de nivel. Sin embargo, el valor del elemento de flujo de bits puede determinarse en el codificador aplicando cualquier tipo de clasificación dependiente de la señal. El preprocesamiento adicional se activa preferiblemente a través de un elemento de flujo de bits de un bit, `bs_sbr_preprocessing`. Cuando `bs_sbr_preprocessing` se establece en uno, se habilita el procesamiento adicional. Cuando `bs_sbr_preprocessing` se establece en cero, el preprocesamiento adicional está deshabilitado. El procesamiento adicional preferible utiliza una curva `preGain` que usa el generador de alta frecuencia para escalar la banda baja, `XLow`, para cada parche. Por ejemplo, la curva `preGain` se puede calcular de acuerdo con:

$$preGain(k) = 10^{(meanNrg - lowEnvSlope(k))/20}, 0 \leq k < k_0$$

donde k_0 es la primera subbanda QMF en la tabla de bandas de frecuencia maestra y $lowEnvSlope$ se calcula usando una función que calcula los coeficientes de un polinomio de mejor ajuste (en un sentido de menos cuadrados), como $polyfit()$. Por ejemplo,

5 $polyfit(3, k_0, x_lowband, lowEnv, lowEnvSlope);$

puede emplearse (usando un polinomio de tercer grado) y donde

$$lowEnv(k) = 10 \log_{10} \frac{\varphi_k(0,0)}{numTimeSlots \cdot RATE + 6}, 0 \leq k < k_0$$

10 donde $x_lowband(k) = [0 \dots k_0-1]$, $numTimeSlot$ es el número de intervalos de tiempo de la envolvente SBR que existen dentro de una trama, $RATE$ es una constante que indica el número de muestras de subbanda QMF por intervalo de tiempo (por ejemplo, 2), φ_k es un coeficiente de filtro de predicción lineal (potencialmente obtenido del método de covarianza) y donde

15

$$meanNrg = \frac{\sum_{k=0}^{k_0-1} lowEnv(k)}{k_0}.$$

Un flujo de bits generado de acuerdo con el estándar USAC MPEG (a veces denominado en el presente documento como un "flujo de bits USAC") incluye contenido de audio codificado y típicamente incluye metadatos indicativos de cada tipo de procesamiento de replicación de banda espectral que aplicará un decodificador para decodificar el contenido de audio del flujo de bits USAC y/o metadatos que controlan tal procesamiento de replicación de banda espectral y/o es indicativo de al menos una característica o parámetro de al menos una herramienta SBR y/o herramienta eSBR que se empleará para decodificar el contenido de audio del flujo de bits USAC.

25 En el presente documento, se usa la expresión "metadatos de SBR mejorada" (o "metadatos eSBR") para denotar los metadatos indicativos de cada tipo de procesamiento de replicación de banda espectral que aplicará un decodificador para decodificar el contenido de audio de un flujo de bits de audio codificado (por ejemplo, un flujo de datos USAC) y/o que controla tal procesamiento de replicación de banda espectral, y/o es indicativo de al menos una característica o parámetro de al menos una herramienta SBR y/o herramienta eSBR que se empleará para decodificar tal contenido de audio, pero que no se describe o menciona en el estándar MPEG-4 AAC. Un ejemplo de metadatos eSBR son los metadatos (indicativos de, o para controlar, el procesamiento de replicación de banda espectral) que se describen o mencionan en el estándar USAC MPEG pero no en el estándar MPEG-4 AAC. Por lo tanto, los metadatos eSBR denotan en el presente documento metadatos que no son metadatos SBR, y los metadatos SBR denotan en el presente documento metadatos que no son metadatos eSBR.

35 Un flujo de bits USAC puede incluir tanto metadatos SBR como metadatos eSBR. Más específicamente, un flujo de bits USAC puede incluir metadatos eSBR que controlan el rendimiento del procesamiento de eSBR por un decodificador, y metadatos SBR que controlan el rendimiento del procesamiento de SBR por el decodificador. De acuerdo con las realizaciones típicas de la presente invención, los metadatos eSBR (por ejemplo, datos de configuración específicos de eSBR) se incluyen (de acuerdo con la presente invención) en un flujo de bits MPEG-4 AAC (por ejemplo, en el contenedor `sbr_extension()` al final de una carga útil SBR).

45 El rendimiento del procesamiento eSBR, durante la decodificación de un flujo de bits codificado usando un conjunto de herramientas eSBR (que comprende al menos una herramienta eSBR), mediante un decodificador regenera la banda de alta frecuencia de la señal de audio, basándose en la replicación de secuencias de armónicos que se truncaron durante la codificación. Tal procesamiento de eSBR ajusta típicamente la envolvente espectral de la banda de alta frecuencia generada y aplica filtrado inverso, y agrega componentes de ruido y sinusoidales para recrear las características espectrales de la señal de audio original.

50 De acuerdo con las realizaciones típicas de la invención, se incluyen metadatos eSBR (por ejemplo, se incluye una pequeña cantidad de bits de control que son metadatos eSBR) en uno o más de los segmentos de metadatos de un flujo de bits de audio codificado (por ejemplo, un flujo de bits MPEG-4 AAC) que también incluye datos de audio codificados en otros segmentos (segmentos de datos de audio). Típicamente, al menos uno de esos segmentos de metadatos de cada bloque del flujo de bits es (o incluye) un elemento de relleno (incluido un identificador que indica el inicio del elemento de relleno), y los metadatos eSBR se incluyen en el elemento de relleno después del identificador.

60 La figura 1 es un diagrama de bloques de una cadena de procesamiento de audio de ejemplo (un sistema de procesamiento de datos de audio), donde uno o más de los elementos del sistema pueden configurarse de acuerdo con una realización de la presente invención. El sistema incluye los siguientes elementos, acoplados como se muestra: codificador 1, subsistema 2 de entrega, decodificador 3 y unidad 4 de posprocesamiento. En variaciones

del sistema mostrado, se omiten uno o más de los elementos, o se incluyen unidades de procesamiento de datos de audio adicionales.

En algunas implementaciones, el codificador 1 (que opcionalmente incluye una unidad de preprocesamiento) está configurado para aceptar muestras PCM (dominio tiempo) que comprenden contenido de audio como entrada, y para emitir un flujo de bits de audio codificado (que tiene un formato compatible con el estándar MPEG-4 AAC) que es indicativo del contenido de audio. Los datos del flujo de bits que son indicativos del contenido de audio se denominan a veces en el presente documento "datos de audio" o "datos de audio codificados". 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 eSBR (y típicamente también otros metadatos) así como datos de audio.

Uno o más flujos de bits de audio codificados emitidos desde el codificador 1 pueden confirmarse en el subsistema 2 de entrega de audio codificado. El subsistema 2 está configurado para almacenar y/o entregar cada salida de flujo de bits codificado desde el codificador 1. Una salida de flujo de bits de audio codificado del codificador 1 puede ser almacenada por el subsistema 2 (por ejemplo, en forma de un disco DVD o Blu ray), o transmitida por el subsistema 2 (que puede implementar un enlace de transmisión o red), o pueden tanto almacenarse como transmitirse por el subsistema 2.

El decodificador 3 está configurado para decodificar un flujo de bits de audio MPEG-4 AAC codificado (generado por el codificador 1) que recibe a través del subsistema 2. En algunas realizaciones, el decodificador 3 está configurado para extraer metadatos eSBR de cada bloque del flujo de bits y decodificar el flujo de bits (incluso realizando el procesamiento de eSBR usando los metadatos eSBR extraídos) para generar datos de audio decodificados (por ejemplo, flujos de muestras de audio PCM decodificadas). En algunas realizaciones, el decodificador 3 está configurado para extraer metadatos SBR del flujo de bits (pero para ignorar los metadatos eSBR incluidos en el flujo de bits) y para decodificar el flujo de bits (incluso realizando el procesamiento SBR usando los metadatos SBR extraídos) para generar datos de audio decodificados (por ejemplo, flujos de muestras de audio PCM decodificadas). Típicamente, el decodificador 3 incluye un búfer que almacena (por ejemplo, de manera no transitoria) segmentos del flujo de bits de audio codificado recibido del subsistema 2.

La unidad 4 de posprocesamiento de la figura 1 está configurada para aceptar un flujo de datos de audio decodificados del decodificador 3 (por ejemplo, muestras de audio PCM decodificadas) y para realizar el posprocesamiento en el mismo. La unidad de posprocesamiento también puede configurarse para reproducir el contenido de audio posprocesado (o el audio decodificado recibido del decodificador 3) para su reproducción por uno o más altavoces.

La figura 2 es un diagrama de bloques de un codificador 100 que es una realización de la unidad de procesamiento de audio de la invención. Cualquiera de los componentes o elementos del codificador 100 puede implementarse como uno o más procesos y/o uno o más circuitos (por ejemplo, ASIC, FPGA u otros circuitos integrados), en hardware, software o una combinación de hardware y software. El codificador 100 incluye el codificador 105, la etapa 107 de relleno/formateador, la etapa 106 de generación de metadatos y la memoria 109 de búfer, conectadas como se muestra. Típicamente también, el codificador 100 incluye otros elementos de procesamiento (no mostrados). El codificador 100 está configurado para convertir un flujo de bits de audio de entrada en un flujo de bits de salida codificado MPEG-4 AAC.

El generador 106 de metadatos está acoplado y configurado para generar (y/o pasar a la etapa 107) metadatos (incluidos metadatos eSBR y metadatos SBR) que se incluirán en la etapa 107 en el flujo de bits codificado que se emitirá desde el codificador 100.

El codificador 105 está acoplado y configurado para codificar (por ejemplo, realizando compresión sobre el mismo) los datos de audio de entrada y para confirmar el audio codificado resultante en la etapa 107 para su inclusión en el flujo de bits codificado que se emitirá desde la etapa 107.

La etapa 107 está configurada para multiplexar el audio codificado del codificador 105 y los metadatos (incluidos los metadatos eSBR y los metadatos SBR) del generador 106 para generar el flujo de bits codificado que se emitirá desde la etapa 107, preferiblemente para que el flujo de bits codificado tenga el formato especificado por una de las realizaciones de la presente invención.

La memoria 109 de búfer está configurada para almacenar (por ejemplo, de manera no transitoria) al menos un bloque de la salida del flujo de bits de audio codificado de la etapa 107, y una secuencia de los bloques del flujo de bits de audio codificado se confirma desde la memoria 109 de búfer como salida del codificador 100 a un sistema de entrega.

La figura 3 es un diagrama de bloques de un sistema que incluye un decodificador 200 que es una realización de la unidad de procesamiento de audio de la invención y, opcionalmente, también un posprocesador 300 acoplado a la misma. Cualquiera de los componentes o elementos del decodificador 200 y posprocesador 300 puede implementarse como uno o más procesos y/o uno o más circuitos (por ejemplo, ASIC, FPGA u otros circuitos

integrados), en hardware, software o una combinación de hardware y software. El decodificador 200 comprende memoria 201 de búfer, desformateador (analizador) 205 de carga útil de flujo de bits, subsistema 202 de decodificación de audio (a veces denominado etapa de decodificación "central" o subsistema de decodificación "central"), etapa 203 de procesamiento de eSBR y etapa 204 de generación de bits de control, conectados como se muestra. Típicamente también, el decodificador 200 incluye otros elementos de procesamiento (no mostrados).

La memoria 201 de búfer (búfer) almacena (por ejemplo, de manera no transitoria) al menos un bloque de un flujo de bits de audio MPEG-4 AAC codificado recibido por el decodificador 200. En el funcionamiento del decodificador 200, se confirma una secuencia de los bloques del flujo de bits desde el búfer 201 al desformateador 205.

En variaciones de la realización de la figura 3 (o la realización de la figura 4 que se describirá), una APU que no es un decodificador (por ejemplo, APU 500 de la figura 6) incluye una memoria de búfer (por ejemplo, una memoria de búfer idéntica al búfer 201) que almacena (por ejemplo, de manera no transitoria) al menos un bloque de un flujo de bits de audio codificado (por ejemplo, un flujo de bits de audio MPEG-4 AAC) del mismo tipo recibido por el búfer 201 de la figura 3 o la figura 4 (es decir, un flujo de bits de audio codificado que incluye metadatos eSBR).

Con referencia nuevamente a la figura 3, el desformateador 205 está acoplado y configurado para demultiplexar cada bloque del flujo de bits para extraer metadatos SBR (incluidos datos de envolvente cuantificados) y metadatos eSBR (y típicamente también otros metadatos) de los mismos, para confirmar al menos los metadatos eSBR y los metadatos SBR a la etapa 203 de procesamiento de eSBR, y típicamente también para confirmar otros metadatos extraídos al subsistema 202 de decodificación (y opcionalmente también al generador 204 de bits de control). El desformateador 205 también está acoplado y configurado para extraer datos de audio de cada bloque del flujo de bits y para confirmar los datos de audio extraídos al subsistema 202 de decodificación (etapa de decodificación).

El sistema de la figura 3 opcionalmente también incluye el posprocesador 300. El posprocesador 300 incluye una memoria 301 de búfer (búfer) y otros elementos de procesamiento (no mostrados) que incluyen al menos un elemento de procesamiento acoplado al búfer 301. El búfer 301 almacena (por ejemplo, de manera no transitoria) al menos un bloque (o trama) de los datos de audio decodificados recibidos por el posprocesador 300 desde el decodificador 200. Los elementos de procesamiento del posprocesador 300 están acoplados y configurados para recibir y procesar de manera adaptativa una secuencia de los bloques (o tramas) de la salida de audio decodificada del búfer 301, usando la salida de metadatos del subsistema 202 de decodificación (y/o desformateador 205) y/o bits de control emitidos desde la etapa 204 del decodificador 200.

El subsistema 202 de decodificación de audio del decodificador 200 está configurado para decodificar los datos de audio extraídos por el analizador 205 (tal decodificación puede denominarse operación de decodificación "central") para generar datos de audio decodificados y para confirmar los datos de audio decodificados en la etapa 203 de procesamiento eSBR. La decodificación se realiza en el dominio frecuencia y típicamente incluye cuantificación inversa seguida de procesamiento espectral. Típicamente, una etapa final de procesamiento en el subsistema 202 aplica una transformación de dominio frecuencia a dominio tiempo a los datos de audio de dominio frecuencia decodificados, de modo que la salida del subsistema son datos de audio decodificados de dominio tiempo. La etapa 203 está configurada para aplicar herramientas SBR y herramientas eSBR indicadas por los metadatos eSBR y la eSBR (extraída por el analizador 205) a los datos de audio decodificados (es decir, para realizar el procesamiento SBR y eSBR en la salida del subsistema 202 de decodificación usando los metadatos SBR y eSBR) para generar los datos de audio completamente decodificados que se envían (por ejemplo, al posprocesador 300) desde el decodificador 200. Típicamente, el decodificador 200 incluye una memoria (accesible por el subsistema 202 y la etapa 203) que almacena los datos de audio desformateados y la salida de metadatos del desformateador 205, y la etapa 203 está configurada para acceder a los datos de audio y metadatos (incluidos los metadatos SBR y los metadatos eSBR) como necesario durante el procesamiento de SBR y eSBR. Se puede considerar que el procesamiento SBR y el procesamiento de eSBR en la etapa 203 son posprocesamiento en la salida del subsistema 202 de decodificación central. Opcionalmente, el decodificador 200 también incluye un subsistema de mezcla ascendente final (que puede aplicar herramientas estéreo paramétricas ("PS") definidas en el estándar MPEG-4 AAC, usando metadatos PS extraídos por el desformateador 205 y/o bits de control generados en el subsistema 204) que es acoplado y configurado para realizar una mezcla ascendente en la salida de la etapa 203 para generar audio de mezcla ascendente totalmente decodificado que se emite desde el decodificador 200. Alternativamente, el posprocesador 300 está configurado para realizar una mezcla ascendente en la salida del decodificador 200 (por ejemplo, usando metadatos PS extraídos por el desformateador 205 y/o bits de control generados en el subsistema 204).

En respuesta a los metadatos extraídos por el desformateador 205, el generador 204 de bits de control puede generar datos de control, y los datos de control pueden usarse dentro del decodificador 200 (por ejemplo, en un subsistema de mezcla ascendente final) y/o confirmarse como salida del decodificador 200 (por ejemplo, para el posprocesador 300 para su uso en posprocesamiento). En respuesta a los metadatos extraídos del flujo de bits de entrada (y opcionalmente también en respuesta a los datos de control), la etapa 204 puede generar (y confirmar al posprocesador 300) bits de control que indican que la salida de datos de audio decodificados de la etapa 203 de procesamiento de eSBR debe someterse a un tipo específico de posprocesamiento. En algunas implementaciones, el decodificador 200 está configurado para confirmar metadatos extraídos por el desformateador 205 del flujo de bits

de entrada al posprocesador 300, y el posprocesador 300 está configurado para realizar posprocesamiento en la salida de datos de audio decodificados desde el decodificador 200 usando los metadatos.

La figura 4 es un diagrama de bloques de una unidad 210 de procesamiento de audio ("APU") que es otra realización de la unidad 210 de procesamiento de audio APU de la invención es un decodificador heredado que no está configurado para realizar procesamiento eSBR. Cualquiera de los componentes o elementos de la APU 210 puede implementarse como uno o más procesos y/o uno o más circuitos (por ejemplo, ASIC, FPGA u otros circuitos integrados), en hardware, software o una combinación de hardware y software. La APU 210 comprende memoria 201 de búfer, desformateador 215 de carga útil de flujo de bits (analizador), subsistema 202 de decodificación de audio (a veces denominado etapa de decodificación "central" o subsistema de decodificación "central") y etapa 213 de procesamiento SBR, conectada como se muestra. Típicamente también, la APU 210 incluye otros elementos de procesamiento (no mostrados). La APU 210 puede representar, por ejemplo, un codificador, decodificador o transcodificador de audio.

Los elementos 201 y 202 de la APU 210 son idénticos a los elementos numerados idénticamente del decodificador 200 (de la figura 3) y la descripción anterior de ellos no se repetirá. En el funcionamiento de la APU 210, una secuencia de bloques de un flujo de bits de audio codificado (un flujo de bits MPEG-4 AAC) recibido por la APU 210 se confirma desde el búfer 201 al desformateador 215.

El desformateador 215 se acopla y se configura para demultiplexar cada bloque del flujo de bits para extraer metadatos SBR (incluidos los datos de envolvente cuantificados) y típicamente también otros metadatos de los mismos, pero para ignorar los metadatos eSBR que pueden incluirse en el flujo de bits de acuerdo con cualquier realización de la presente invención. El desformateador 215 está configurado para confirmar al menos los metadatos SBR en la etapa 213 de procesamiento de SBR. El desformateador 215 también está acoplado y configurado para extraer datos de audio de cada bloque del flujo de bits y para confirmar los datos de audio extraídos en el subsistema 202 de decodificación (etapa de decodificación).

El subsistema 202 de decodificación de audio del decodificador 200 está configurado para decodificar los datos de audio extraídos por el desformateador 215 (tal decodificación puede denominarse operación de decodificación "central") para generar datos de audio decodificados y para confirmar los datos de audio decodificados en la etapa 213 de procesamiento SBR. La decodificación se realiza en el dominio frecuencia. Típicamente, una etapa final de procesamiento en el subsistema 202 aplica una transformación de dominio frecuencia a dominio tiempo a los datos de audio de dominio frecuencia decodificados, de modo que la salida del subsistema son datos de audio decodificados de dominio tiempo. La etapa 213 está configurada para aplicar herramientas SBR (pero no herramientas eSBR) indicadas por los metadatos SBR (extraídos por el desformateador 215) a los datos de audio decodificados (es decir, para realizar el procesamiento SBR en la salida del subsistema 202 de decodificación usando los metadatos SBR) para generar los datos de audio totalmente decodificados que se envían (por ejemplo, al posprocesador 300) desde la APU 210. Típicamente, la APU 210 incluye una memoria (accesible por el subsistema 202 y la etapa 213) que almacena los datos de audio desformateados y la salida de metadatos del desformateador 215, y la etapa 213 está configurada para acceder a los datos de audio y metadatos (incluidos los metadatos SBR) según sea necesario durante el procesamiento SBR. Se puede considerar que el procesamiento SBR en la etapa 213 es un posprocesamiento en la salida del subsistema 202 de decodificación central. Opcionalmente, la APU 210 también incluye un subsistema de mezcla ascendente final (que puede aplicar herramientas estéreo paramétricas ("PS") definidas en el estándar MPEG-4 AAC, usando metadatos PS extraídos por el desformateador 215) que está acoplado y configurado para realizar mezcla ascendente en la salida de la etapa 213 para generar audio de mezcla ascendente y totalmente decodificado que se emite desde la APU 210. Alternativamente, se configura un posprocesador para realizar una mezcla ascendente en la salida de la APU 210 (por ejemplo, usando metadatos PS extraídos por el desformateador 215 y/o bits de control generados en la APU 210).

Varias implementaciones del codificador 100, el decodificador 200 y la APU 210 están configuradas para realizar diferentes realizaciones del método de la invención.

De acuerdo con algunas realizaciones, se incluyen metadatos eSBR (por ejemplo, se incluye una pequeña cantidad de bits de control que son metadatos eSBR) en un flujo de bits de audio codificado (por ejemplo, un flujo de bits MPEG-4 AAC), de modo que los decodificadores heredados (que no son configurado para analizar los metadatos eSBR, o para usar cualquier herramienta eSBR a la que pertenezcan los metadatos eSBR) puede ignorar los metadatos eSBR pero, sin embargo, decodificar el flujo de bits en la medida de lo posible sin usar los metadatos eSBR o cualquier herramienta eSBR a la que pertenezcan los metadatos eSBR, típicamente sin ninguna penalización significativa en la calidad del audio decodificado. Sin embargo, los decodificadores eSBR configurados para analizar el flujo de bits para identificar los metadatos eSBR y para usar al menos una herramienta eSBR en respuesta a los metadatos eSBR, disfrutarán de los beneficios de usar al menos una de tales herramientas eSBR. Por lo tanto, las realizaciones de la invención proporcionan un medio para transmitir eficazmente datos o metadatos de control de replicación de banda espectral mejorada (eSBR) de una manera compatible con versiones anteriores.

Típicamente, los metadatos eSBR en el flujo de bits son indicativos de (por ejemplo, son indicativos de al menos una característica o parámetro de) una o más de las siguientes herramientas eSBR (que se describen en el estándar USAC MPEG, y que pueden o no haber sido aplicados por un codificador durante la generación del flujo de bits):

- 5 - transposición armónica; y
- preprocesamiento adicional de parcheo QMF (preaplanamiento).

10 Por ejemplo, los metadatos eSBR incluidos en el flujo de bits pueden ser indicativos de los valores de los parámetros (descritos en el estándar USAC MPEG y en la presente divulgación): `sbrPatchingMode[ch]`, `sbrOversamplingFlag[ch]`, `sbrPitchInBins[ch]`, `sbrPitchInBins[ch]` y `bs_sbr_preprocessing`.

15 En el presente documento, la notación `X[ch]`, donde `X` es algún parámetro, denota que el parámetro pertenece al canal ("ch") del contenido de audio de un flujo de bits codificado que se va a decodificar. Para simplificar, a veces omitimos la expresión `[ch]` y asumimos que el parámetro relevante pertenece a un canal de contenido de audio.

20 En el presente documento, la notación `X[ch][env]`, donde `X` es algún parámetro, denota que el parámetro pertenece a la envolvente SBR ("env") del canal ("ch") del contenido de audio de un flujo de bits codificado que se va a decodificar. Para simplificar, a veces omitimos las expresiones `[env]` y `[ch]`, y asumimos que el parámetro relevante pertenece a una envolvente SBR de un canal de contenido de audio.

25 Durante la decodificación de un flujo de bits codificado, el rendimiento de la transposición armónica durante una etapa de procesamiento de eSBR de la decodificación (para cada canal, "ch", del contenido de audio indicado por el flujo de bits) se controla mediante los siguientes parámetros de metadatos de eSBR: `sbrPatchingMode[ch]`, `sbrOversamplingFlag[ch]`, `sbrPitchInBinsFlag[ch]`; y `sbrPitchInBins[ch]`.

30 El valor "`sbrPatchingMode[ch]`" indica el tipo de transpondedor usado en eSBR: `sbrPatchingMode[ch] = 1` indica el parcheo de transposición lineal como se describe en la sección 4.6.18 del estándar MPEG-4 AAC (como se usa con SBR de alta calidad o SBR de baja potencia); `sbrPatchingMode[ch] = 0` indica el parcheo armónico SBR como se describe en la sección 7.5.3 o 7.5.4 del estándar USAC MPEG.

35 El valor "`sbrOversamplingFlag[ch]`" indica el uso de sobremuestreo de dominio frecuencia adaptativo de señal en eSBR en combinación con el parcheo SBR armónico basado en DFT como se describe en la sección 7.5.3 del estándar USAC MPEG. Este indicador controla el tamaño de las DFT que se utilizan en el transpondedor: 1 indica sobremuestreo de dominio frecuencia adaptativo de señal habilitado como se describe en la sección 7.5.3.1 del estándar USAC MPEG; 0 indica que el sobremuestreo del dominio frecuencia adaptativo de la señal está desactivado, como se describe en la sección 7.5.3.1 del estándar USAC MPEG.

40 El valor "`sbrPitchInBinsFlag[ch]`" controla la interpretación del parámetro `sbrPitchInBins[ch]`: 1 indica que el valor en `sbrPitchInBins[ch]` es válido y mayor que cero; 0 indica que el valor de `sbrPitchInBins[ch]` se establece en cero.

45 El valor "`sbrPitchInBins[ch]`" controla la adición de términos de productos cruzados en el transpondedor armónico SBR. El valor `sbrPitchInBins[ch]` es un valor entero en el rango [0,127] y representa la distancia medida en intervalos de frecuencia para una DFT de 1536 líneas que actúa sobre la frecuencia de muestreo del codificador central.

En el caso de que un flujo de bits MPEG-4 AAC sea indicativo de un par de canales SBR cuyos canales no están acoplados (en lugar de un solo canal SBR), el flujo de bits es indicativo de dos instancias de la sintaxis anterior (para transposición armónica o no armónica), uno para cada canal del `sbr_channel_pair_element()`.

50 La transposición armónica de la herramienta eSBR mejora típicamente la calidad de las señales musicales decodificadas a frecuencias de cruce relativamente bajas. La transposición no armónica (es decir, el parcheo espectral heredado) mejora típicamente las señales de voz. Por lo tanto, un punto de partida en la decisión sobre qué tipo de transposición es preferible para codificar contenido de audio específico es seleccionar el método de transposición dependiendo de la detección de voz/música con transposición armónica empleada en el contenido musical y el parcheo espectral en el contenido de velocidad.

60 El rendimiento del preaplanamiento durante el procesamiento de eSBR se controla mediante el valor de un parámetro de metadatos eSBR de un bit conocido como "`bs_sbr_preprocessing`", en el sentido de que el preaplanamiento se realiza o no dependiendo del valor de este bit único. Cuando se usa el algoritmo de parcheo QMF de SBR, como se describe en la sección 4.6.18.6.3 del estándar MPEG-4 AAC, se puede realizar el paso de preaplanamiento (cuando lo indique el parámetro "`bs_sbr_preprocessing`") en un esfuerzo para evitar discontinuidades en la forma de la envolvente espectral de una señal de alta frecuencia que se introduce en un ajustador de envolvente posterior (el ajustador de envolvente realiza otra etapa del procesamiento eSBR). El preaplanamiento típicamente mejora el funcionamiento de la etapa de ajuste de envolvente posterior, dando como resultado una señal de banda alta que se percibe como más estable.

Se espera que el requisito general de tasa de bits para incluir en los metadatos eSBR de un flujo de bits MPEG-4 AAC indicativos de las herramientas eSBR mencionadas anteriormente (transposición armónica y preaplanamiento) sea del orden de unos pocos cientos de bits por segundo porque solo los datos de control diferencial necesarios para realizar el procesamiento de eSBR se transmiten de acuerdo con algunas realizaciones de la invención. Los decodificadores heredados pueden ignorar esta información porque se incluye de una manera compatible con versiones anteriores (como se explicará más adelante). Por lo tanto, el efecto perjudicial sobre la tasa de bits asociado con la inclusión de metadatos eSBR es insignificante, por varias razones, incluidas las siguientes:

- la penalización de la tasa de bits (debido a que se incluyen los metadatos de eSBR) es una fracción muy pequeña de la tasa de bits total porque solo se transmiten los datos de control diferencial necesarios para realizar el procesamiento de eSBR (y no una transmisión simultánea de los datos de control de SBR); y

- el ajuste de la información de control relacionada con SBR no depende típicamente de los detalles de la transposición. Los ejemplos de cuándo los datos de control dependen del funcionamiento del transpondedor se describen más adelante en esta solicitud.

Por tanto, las realizaciones de la invención proporcionan un medio para transmitir eficazmente datos o metadatos de control de replicación de banda espectral mejorada (eSBR) de una manera compatible con versiones anteriores. Esta transmisión eficiente de los datos de control eSBR reduce los requisitos de memoria en los decodificadores, codificadores y transcodificadores que emplean aspectos de la invención, mientras que no tiene un efecto adverso tangible sobre la tasa de bits. Además, la complejidad y los requisitos de procesamiento asociados con la realización de eSBR de acuerdo con las realizaciones de la invención también se reducen porque los datos de SBR deben procesarse solo una vez y no difundirse simultáneamente, lo que sería el caso si eSBR se tratara como un tipo de objeto completamente separado en MPEG-4 AAC en lugar de integrarse en el códec MPEG-4 AAC de forma compatible con versiones anteriores.

A continuación, con referencia a la figura 7, describimos elementos de un bloque ("raw_data_block") de un flujo de bits MPEG-4 AAC donde se incluyen metadatos eSBR de acuerdo con algunas realizaciones de la presente invención. La figura 7 es un diagrama de un bloque (un "raw_data_block") del flujo de bits MPEG-4 AAC, que muestra algunos de sus segmentos.

Un bloque de un flujo de bits MPEG-4 AAC puede incluir al menos un "single_channel_element()" (por ejemplo, el elemento de un solo canal que se muestra en la figura 7), y/o al menos un "channel_pair_element()" (no se muestra específicamente en la figura 7 aunque puede estar presente), incluidos los datos de audio de un programa de audio. El bloque también puede incluir una serie de "fill_elements" (por ejemplo, elemento 1 de relleno y/o elemento 2 de relleno de la figura 7) que incluyen datos (por ejemplo, metadatos) relacionados con el programa. Cada "single_channel_element()" incluye un identificador (por ejemplo, "ID1" de la figura 7) que indica el inicio de un elemento de canal único, y puede incluir datos de audio indicativos de un canal diferente de un programa de audio multicanal. Cada elemento de par de canales incluye un identificador (no mostrado en la figura 7) que indica el inicio de un elemento de par de canales, y puede incluir datos de audio indicativos de dos canales del programa.

Un fill_element (denominado en el presente documento elemento de relleno) de un flujo de bits MPEG-4 AAC incluye un identificador ("ID2" de la figura 7) que indica el inicio de un elemento de relleno y datos de relleno después del identificador. El identificador ID2 puede consistir en un entero sin signo de tres bits transmitido primero el bit más significativo ("uimsbf") que tiene un valor de 0x6. Los datos de relleno pueden incluir un elemento extension_payload() (a veces denominado en el presente documento como una carga útil de extensión) cuya sintaxis se muestra en la tabla 4.57 del estándar MPEG-4 AAC. Existen varios tipos de cargas útiles de extensión y se identifican mediante el parámetro "extension_type", que es un entero sin signo de cuatro bits que se transmite primero con el bit más significativo ("uimsbf").

Los datos de relleno (por ejemplo, una carga útil de extensión de los mismos) pueden incluir un encabezado o identificador (por ejemplo, "encabezado1" de la figura 7) que indica un segmento de datos de relleno que es indicativo de un objeto SBR (es decir, el encabezado inicializa un tipo "de objeto SBR", denominado sbr_extension_data() en el estándar MPEG-4 AAC). Por ejemplo, una carga útil de extensión de replicación de banda espectral (SBR) se identifica con el valor de '1101' o '1110' para el campo extension_type en el encabezado, con el identificador '1101' que identifica una carga útil de extensión con datos SBR y '1110' que identifica una carga útil de extensión con datos SBR con una verificación de redundancia cíclica (CRC) para verificar la exactitud de los datos SBR.

Cuando el encabezado (por ejemplo, el campo extension_type) inicializa un tipo de objeto SBR, los metadatos SBR (a veces denominados en el presente documento "datos de replicación de banda espectral" y denominados sbr_data() en el estándar MPEG-4 AAC) siguen al encabezado, y al menos un elemento de extensión de replicación de banda espectral (por ejemplo, el "elemento de extensión de SBR" del elemento 1 de relleno de la figura 7) puede seguir los metadatos SBR. Tal elemento de extensión de replicación de banda espectral (un segmento del flujo de bits) se denomina contenedor "sbr_extension()" en el estándar MPEG-4 AAC. Un elemento de extensión de

replicación de banda espectral incluye opcionalmente un encabezado (por ejemplo, "encabezado de extensión SBR" del elemento 1 de relleno de la figura 7).

El estándar MPEG-4 AAC contempla que un elemento de extensión de replicación de banda espectral puede incluir datos PS (estéreo paramétrico) para datos de audio de un programa. El estándar MPEG-4 AAC contempla que cuando el encabezado de un elemento de relleno (por ejemplo, de una carga útil de extensión del mismo) inicializa un tipo de objeto SBR (al igual que "encabezado1" de la figura 7) y un elemento de extensión de replicación de banda espectral del elemento de relleno incluye datos PS, el elemento de relleno (por ejemplo, la carga útil de extensión del mismo) incluye datos de replicación de banda espectral y un parámetro "bs_extension_id" cuyo valor (es decir, bs_extension_id = 2) indica que los datos PS están incluidos en un elemento de extensión de replicación de banda espectral del elemento de relleno.

De acuerdo con algunas realizaciones de la presente invención, los metadatos eSBR (por ejemplo, un indicador indicativo de si el procesamiento de replicación de banda espectral mejorada (eSBR) se realizará en el contenido de audio del bloque) se incluyen en un elemento de extensión de replicación de banda espectral de un elemento de relleno. Por ejemplo, tal indicador se indica en el elemento 1 de relleno de la figura 7, donde el indicador aparece después del encabezado (el "encabezado de extensión SBR" del elemento 1 de relleno) del "elemento de extensión SBR" del elemento 1 de relleno. Opcionalmente, tal indicador y metadatos eSBR adicionales se incluyen en un elemento de extensión de replicación de banda espectral después del encabezado del elemento de extensión de replicación de banda espectral (por ejemplo, en el elemento de extensión SBR del elemento 1 de relleno en la figura 7, después del encabezado de extensión SBR). De acuerdo con algunas realizaciones de la presente invención, un elemento de relleno que incluye metadatos eSBR también incluye un parámetro "bs_extension_id" cuyo valor (por ejemplo, bs_extension_id = 3) indica que los metadatos eSBR están incluidos en el elemento de relleno y que el procesamiento de eSBR debe ser realizado en el contenido de audio del bloque relevante.

De acuerdo con algunas realizaciones de la invención, los metadatos eSBR se incluyen en un elemento de relleno (por ejemplo, el elemento 2 de relleno de la figura 7) de un flujo de bits MPEG-4 AAC que no sea un elemento de extensión de replicación de banda espectral (elemento de extensión de SBR) del elemento de relleno. Esto se debe a que los elementos de relleno que contienen un extension_payload() con datos SBR o datos SBR con una CRC no contienen ninguna otra carga útil de extensión de ningún otro tipo de extensión. Por lo tanto, en las realizaciones en las que los metadatos eSBR almacenan su propia carga útil de extensión, se usa un elemento de relleno independiente para almacenar los metadatos de eSBR. Tal elemento de relleno incluye un identificador (por ejemplo, "ID2" de la figura 7) que indica el inicio de un elemento de relleno y datos de relleno después del identificador. Los datos de relleno pueden incluir un elemento extension_payload() (a veces denominado en el presente documento como una carga útil de extensión) cuya sintaxis se muestra en la tabla 4.57 del estándar MPEG-4 AAC. Los datos de relleno (por ejemplo, una carga útil de extensión de los mismos) incluyen un encabezado (por ejemplo, "encabezado2" del elemento 2 de relleno de la figura 7) que es indicativo de un objeto eSBR (es decir, el encabezado inicializa un tipo de objeto de replicación de banda espectral mejorada (eSBR)), y los datos de relleno (por ejemplo, una carga útil de extensión de los mismos) incluyen metadatos eSBR después del encabezado. Por ejemplo, el elemento 2 de relleno de la figura 7 incluye dicho encabezado ("encabezado2") y también incluye, después del encabezado, metadatos eSBR (es decir, el "indicador" en el elemento 2 de relleno, que indica si la replicación de banda espectral mejorada (eSBR) se debe realizar en el contenido de audio del bloque). Opcionalmente, también se incluyen metadatos eSBR adicionales en los datos de relleno del elemento 2 de relleno de la figura 7, después del encabezado2. En las realizaciones que se describen en el presente párrafo, el encabezado (por ejemplo, encabezado2 de la figura 7) tiene un valor de identificación que no es uno de los valores convencionales especificados en la tabla 4.57 del estándar MPEG-4 AAC, y en cambio es indicativo de una carga útil de extensión eSBR (de modo que el campo extension_type del encabezado indique que los datos de relleno incluyen metadatos eSBR).

En una primera clase de realizaciones, la invención es una unidad de procesamiento de audio (por ejemplo, un decodificador), que comprende:

una memoria (por ejemplo, el búfer 201 de la figura 3 o 4) configurada para almacenar al menos un bloque de un flujo de bits de audio codificado (por ejemplo, al menos un bloque de un flujo de bits MPEG-4 AAC);

un desformateador de carga útil de flujo de bits (por ejemplo, el elemento 205 de la figura 3 o el elemento 215 de la figura 4) acoplado a la memoria y configurado para demultiplexar al menos una porción de dicho bloque del flujo de bits; y

un subsistema de decodificación (por ejemplo, elementos 202 y 203 de la figura 3, o elementos 202 y 213 de la figura 4), acoplado y configurado para decodificar al menos una porción del contenido de audio de dicho bloque del flujo de bits, donde el bloque incluye:

un elemento de relleno, que incluye un identificador que indica un inicio del elemento de relleno (por ejemplo, el identificador "id_syn_ele" que tiene el valor 0x6, de la tabla 4.85 del estándar MPEG-4 AAC), y datos de relleno después del identificador, donde los datos de relleno incluyen:

al menos un indicador que identifica si el procesamiento de replicación de banda espectral mejorada (eSBR) debe realizarse en el contenido de audio del bloque (por ejemplo, usando datos de replicación de banda espectral y metadatos eSBR incluidos en el bloque).

El indicador son los metadatos eSBR y un ejemplo del indicador es el indicador `sbrPatchingMode`. Otro ejemplo del indicador es el indicador `harmonicSBR`. Ambos indicadores indican si se va a realizar una forma de base de replicación de banda espectral o una forma mejorada de replicación espectral en los datos de audio del bloque. La forma de base de la replicación espectral es el parcheo espectral, y la forma mejorada de replicación de la banda espectral es la transposición armónica.

En algunas realizaciones, los datos de relleno también incluyen metadatos eSBR adicionales (es decir, metadatos eSBR distintos del indicador).

La memoria puede ser un búfer (por ejemplo, una implementación de la memoria 201 de búfer de la figura 4) que almacena (por ejemplo, de manera no transitoria) al menos dicho bloque del flujo de bits de audio codificado.

Se estima que la complejidad del rendimiento del procesamiento de eSBR (usando la transposición armónica eSBR y el preaplanamiento) por un decodificador eSBR durante la decodificación de un flujo de bits MPEG-4 AAC que incluye metadatos eSBR (indicativos de estas herramientas eSBR) sería la siguiente (para decodificación típica con los parámetros indicados):

- Transposición armónica (16 kbps, 14400/28800 Hz)
 - Basada en DFT: 3,68 WMOPS (millones de operaciones ponderadas por segundo);
 - Basada en QMF: 0,98 WMOPS;
- Preprocesamiento de parcheo QMF (preaplanamiento): 0,1WMOPS.

Se sabe que la transposición basada en DFT funciona típicamente mejor que la transposición basada en QMF para transitorios.

De acuerdo con algunas realizaciones de la presente invención, un elemento de relleno (de un flujo de bits de audio codificado) que incluye metadatos eSBR también incluye un parámetro (por ejemplo, un parámetro `"bs_extension_id"`) cuyo valor (por ejemplo, `bs_extension_id = 3`) indica que los metadatos eSBR está incluido en el elemento de relleno y que el procesamiento de eSBR debe realizarse en el contenido de audio del bloque relevante, y/o un parámetro (por ejemplo, el mismo parámetro `"bs_extension_id"`) cuyo valor (por ejemplo, `bs_extension_id = 2`) indica que un contenedor `sbr_extension()` del elemento de relleno incluye datos PS. Por ejemplo, como se indica en la tabla 1 a continuación, tal parámetro que tiene el valor `bs_extension_id = 2` puede indicar que un contenedor `sbr_extension()` del elemento de relleno incluye datos PS, y tal parámetro que tiene el valor `bs_extension_id = 3` puede indicar que un contenedor `sbr_extension()` del elemento de relleno incluye metadatos eSBR:

Tabla 1

bs_extension_id	Significado
0	Reservado
1	Reservado
2	EXTENSION_ID_PS
3	EXTENSION_ID_ESBR

De acuerdo con algunas realizaciones de la invención, la sintaxis de cada elemento de extensión de replicación de banda espectral que incluye metadatos eSBR y/o datos PS es como se indica en la tabla 2 a continuación (en la que `"sbr_extension()"` denota un contenedor que es el elemento de extensión de replicación de banda espectral, `"bs_extension_id"` es como se describe en la tabla 1 anterior, `"ps_data"` denota datos PS y `"esbr_data"` denota metadatos eSBR):

Tabla 2

<code>sbr_extension(bs_extension_id, num_bits_left)</code>	
{	
switch (bs_extension_id) {	

case EXTENSION_ID_PS:	
num_bits_left -= ps_data();	Nota 1
break;	
case EXTENSION_ID_ESBR:	
num_bits_left -= esbr_data();	Nota 2
break;	
default:	
bs_fill_bits;	
num_bits_left = 0;	
break;	
}	
}	
Nota 1: ps_data() devuelve el número de bits leídos.	
Nota 2: esbr_data() devuelve el número de bits leídos	

En una realización de ejemplo, esbr_data() al que se hace referencia en la tabla 2 anterior es indicativo de los valores de los siguientes parámetros de metadatos:

- 5 1. El parámetro de metadatos de un bit, "bs_sbr_preprocessing"; y
2. Para cada canal ("ch") de contenido de audio del flujo de bits codificado que se va a decodificar, cada uno de los parámetros descritos anteriormente: "sbrPatchingMode[ch]"; "sbrOversamplingFlag[ch]"; "sbrPitchInBinsFlag[ch]"; y "sbrPitchInBins[ch]".

10 Por ejemplo, en algunas realizaciones, esbr_data() puede tener la sintaxis indicada en la tabla 3, para indicar estos parámetros de metadatos:

Tabla 3

Sintaxis	Nº de bits
esbr_data(id_aac, bs_coupling)	
{	
bs_sbr_preprocessing;	1
if (id_aac == ID_SCE) {	
if (sbrPatchingMode[0] == 0) {	1
sbrOversamplingFlag[0];	1
if (sbrPitchInBinsFlag[0])	1
sbrPitchInBins[0];	7
else	
sbrPitchInBins[0] = 0;	
} else {	
sbrOversamplingFlag[0] = 0;	
sbrPitchInBins[0] = 0;	
}	
} else if (id_aac == ID_CPE) {	
If (bs_coupling) {	
if (sbrPatchingMode[0,1] == 0) {	1

Sintaxis	Nº de bits
sbrOversamplingFlag[0,1];	1
if (sbrPitchInBinsFlag[0,1])	1
sbrPitchInBins[0,1];	7
else	
sbrPitchInBins[0,1] = 0;	
} else {	
sbrOversamplingFlag[0,1] = 0;	
sbrPitchInBins[0,1] = 0;	
}	
} else { /* bs_coupling == 0 */	
if (sbrPatchingMode[0] == 0) {	1
sbrOversamplingFlag[0];	1
if (sbrPitchInBinsFlag[0])	1
sbrPitchInBins[0];	7
else	
sbrPitchInBins[0] = 0;	
} else {	
sbrOversamplingFlag[0] = 0;	
sbrPitchInBins[0] = 0;	
}	
if (sbrPatchingMode[1] == 0) {	1
sbrOversamplingFlag[1];	1
if (sbrPitchInBinsFlag[1])	1
sbrPitchInBins[1];	7
else	
sbrPitchInBins[1] = 0;	
} else {	
sbrOversamplingFlag[1] = 0;	
sbrPitchInBins[1] = 0;	
}	
}	
}	

Nota: *bs_sbr_preprocessing* se define como se describe en la sección 6.2.12 of ISO/IEC 23003-3:2012. *sbrPatchingMode[ch]*, *sbrOversamplingFlag[ch]*, *sbrPitchInBinsFlag[ch]* y *sbrPitchInBins[ch]* se definen como se describe en la sección 7.5 de ISO/IEC 23003-3:2012.

- 5 La sintaxis anterior permite una implementación eficiente de una forma mejorada de replicación de banda espectral, como la transposición armónica, como una extensión de un decodificador heredado. Específicamente, los datos eSBR de la tabla 3 incluyen solo aquellos parámetros necesarios para realizar la forma mejorada de replicación de banda espectral que no están ya soportados en el flujo de bits o directamente derivables de los parámetros ya soportados en el flujo de bits. Todos los demás parámetros y datos de procesamiento necesarios para realizar la forma mejorada de replicación de banda espectral se extraen de parámetros preexistentes en ubicaciones ya definidas en el flujo de bits.

Por ejemplo, un decodificador compatible con MPEG-4 HE-AAC o HE-AAC v2 puede ampliarse para incluir una forma mejorada de replicación de banda espectral, como la transposición armónica. Esta forma mejorada de replicación de banda espectral se suma a la forma de base de replicación de banda espectral que ya es soportada por el decodificador. En el contexto de un decodificador compatible con MPEG-4 HE-AAC o HE-AAC v2, esta forma de base de replicación de banda espectral es la herramienta SBR de parcheo espectral QMF como se define en la sección 4.6.18 del estándar MPEG-4 AAC.

Al realizar la forma mejorada de replicación de banda espectral, un decodificador HE-AAC extendido puede reutilizar muchos de los parámetros del flujo de bits ya incluidos en la carga útil de extensión SBR del flujo de bits. Los parámetros específicos que pueden reutilizarse incluyen, por ejemplo, los diversos parámetros que determinan la tabla de bandas de frecuencia maestra. Estos parámetros incluyen `bs_start_freq` (parámetro que determina el inicio de la tabla de frecuencia maestra), `bs_stop_freq` (parámetro que determina la parada de la tabla de frecuencia maestra), `bs_freq_scale` (parámetro que determina el número de bandas de frecuencia por octava) y `bs_alter_scale` (parámetro que modifica la escala de las bandas de frecuencia). Los parámetros que pueden reutilizarse también incluyen parámetros que determinan la tabla de bandas de ruido (`bs_noise_bands`) y los parámetros de la tabla de bandas limitadoras (`bs_limiter_bands`). Por consiguiente, en varias realizaciones, al menos algunos de los parámetros equivalentes especificados en el estándar USAC se omiten del flujo de bits, reduciendo así la sobrecarga de control en el flujo de bits. Típicamente, cuando un parámetro especificado en el estándar AAC tiene un parámetro equivalente especificado en el estándar USAC, el parámetro equivalente especificado en el estándar USAC tiene el mismo nombre que el parámetro especificado en el estándar AAC, por ejemplo, el factor de escala de la envolvente `EOrigMapped`. Sin embargo, el parámetro equivalente especificado en el estándar USAC tiene típicamente un valor diferente, que se "ajusta" para el procesamiento de SBR mejorada definido en el estándar USAC en lugar del procesamiento SBR definido en el estándar AAC.

Para mejorar la calidad subjetiva del contenido de audio con una estructura de frecuencia armónica y características tonales fuertes, en particular a tasas de bits bajas, se recomienda la activación de SBR mejorada. Los valores del elemento de flujo de bits correspondiente (es decir, `esbr_data()`), que controla estas herramientas, se pueden determinar en el codificador aplicando un mecanismo de clasificación dependiente de la señal. Generalmente, el uso del método de parcheo armónico (`sbrPatchingMode == 1`) es preferible para codificar señales musicales a tasas de bits muy bajas, donde el códec central puede estar considerablemente limitado en el ancho de banda de audio. Esto es especialmente cierto si estas señales incluyen una estructura armónica pronunciada. Por el contrario, se prefiere el uso del método de parcheo de SBR normal para señales de voz y mixtas, ya que proporciona una mejor conservación de la estructura temporal en la voz.

Para mejorar el rendimiento del transpondedor armónico, se puede activar un paso de preprocesamiento (`bs_sbr_preprocessing == 1`) que se esfuerza por evitar la introducción de discontinuidades espectrales de la señal que entran en el ajustador de envolvente posterior. El funcionamiento de la herramienta es beneficioso para los tipos de señal en los que la envolvente espectral gruesa de la señal de banda baja que se usa para la reconstrucción de alta frecuencia muestra grandes variaciones de nivel.

Para mejorar la respuesta transitoria del parcheo armónico SBR, se puede aplicar un sobremuestreo de dominio frecuencia adaptativo de señal (`sbrOversamplingFlag == 1`). Dado que el sobremuestreo de dominio frecuencia adaptativo de la señal aumenta la complejidad computacional del transpondedor, pero solo brinda beneficios para las tramas que contienen transitorios, el uso de esta herramienta está controlado por el elemento de flujo de bits, que se transmite una vez por trama y por canal SBR independiente.

Un decodificador que opera en el modo de SBR mejorada propuesto necesita típicamente poder cambiar entre el parcheo SBR heredado y mejorado. Por lo tanto, se puede introducir un retardo que puede ser tan largo como la duración de una trama de audio central, dependiendo de la configuración del decodificador. Típicamente, el retardo tanto para el parcheo SBR heredado como mejorado será similar.

Además de los numerosos parámetros, un decodificador HE-AAC extendido también puede reutilizar otros elementos de datos cuando se realiza una forma mejorada de replicación de banda espectral de acuerdo con realizaciones de la invención. Por ejemplo, los datos de la envolvente y los datos del ruido de fondo también pueden extraerse de los datos `bs_data_env` (factores de escala de la envolvente) y `bs_noise_env` (factores de escala del ruido de fondo) y usarse durante la forma mejorada de replicación de banda espectral.

En esencia, estas realizaciones explotan los parámetros de configuración y los datos de envolvente ya soportados por un decodificador HE-AAC o HE-AAC v2 heredado en la carga útil de extensión SBR para permitir una forma mejorada de replicación de banda espectral que requiere la menor cantidad posible de datos transmitidos adicionales. Los metadatos se sintonizaron originalmente para una forma de base de HFR (por ejemplo, la operación de traslación espectral de SBR), pero de acuerdo con las realizaciones, se usan para una forma mejorada de HFR (por ejemplo, la transposición armónica de eSBR). Como se explicó anteriormente, los metadatos generalmente representan parámetros operativos (por ejemplo, factores de escala de envolvente, factores de escala de ruido de fondo, parámetros de cuadrícula de tiempo/frecuencia, información de adición de senoide, cruce variable sobre

frecuencia/banda, modo de filtrado inverso, resolución de envolvente, modo de suavizado, modo de interpolación de frecuencia) sintonizados y destinados a ser usados con la forma de base de HFR (por ejemplo, traslación espectral lineal). Sin embargo, estos metadatos, combinados con parámetros de metadatos adicionales específicos de la forma mejorada de HFR (por ejemplo, transposición armónica), pueden usarse para procesar de manera eficiente y efectiva los datos de audio usando la forma mejorada de HFR.

En consecuencia, los decodificadores extendidos que soportan una forma mejorada de replicación de banda espectral pueden crearse de una manera muy eficiente confiando en elementos de flujo de bits ya definidos (por ejemplo, aquellos en la carga útil de extensión SBR) y agregando solo los parámetros necesarios para soportar la forma mejorada de replicación de banda espectral (en una carga útil de extensión de elemento de relleno). Esta función de reducción de datos combinada con la ubicación de los parámetros recién agregados en un campo de datos reservado, como un contenedor de extensión, reduce sustancialmente las barreras para crear un decodificador que soporte una forma mejorada de replicación de banda espectral al garantizar que el flujo de bits sea retrocompatible con el decodificador heredado que no soporta la forma mejorada de replicación de banda espectral. Se apreciará que el campo de datos reservado es un campo de datos retrocompatible, es decir, es un campo de datos que ya es soportador por decodificadores anteriores, como los decodificadores HE-AAC o HE-AAC v2 heredados. De manera similar, el contenedor de extensión es retrocompatible, es decir, es un contenedor de extensión que ya es soportador por decodificadores anteriores, como los decodificadores HE-AAC o HE-AAC v2 heredados.

En la tabla 3, el número de la columna de la derecha indica el número de bits del parámetro correspondiente en la columna de la izquierda.

En algunas realizaciones, el tipo de objeto SBR definido en MPEG-4 AAC se actualiza para contener la herramienta SBR y aspectos de la herramienta SBR mejorada (eSBR) como se indica en el elemento de extensión SBR (bs_extension_id == EXTENSION_ID_ESBR). Si un decodificador detecta este elemento de extensión SBR, el decodificador emplea los aspectos señalizados de la herramienta SBR mejorada.

En algunas realizaciones, la invención es un método que incluye un paso de codificación de datos de audio para generar un flujo de bits codificado (por ejemplo, un flujo de bits MPEG-4 AAC), incluyendo los metadatos eSBR en al menos un segmento de al menos un bloque del flujo de bits codificado y datos de audio en al menos otro segmento del bloque. En realizaciones típicas, el método incluye un paso de multiplexar los datos de audio con los metadatos eSBR en cada bloque del flujo de bits codificado. En la decodificación típica del flujo de bits codificado en un decodificador eSBR, el decodificador extrae los metadatos eSBR del flujo de bits (incluido el análisis y el demultiplexado de los metadatos eSBR y los datos de audio) y usa los metadatos eSBR para procesar los datos de audio para generar un flujo de datos de audio decodificados.

Otro aspecto de la invención es un decodificador eSBR configurado para realizar el procesamiento de eSBR (por ejemplo, usando al menos una de las herramientas eSBR conocidas como transposición armónica o preaplanamiento) durante la decodificación de un flujo de bits de audio codificado (por ejemplo, un flujo de bits MPEG-4 AAC) que no incluye metadatos eSBR. Se describirá un ejemplo de tal decodificador con referencia a la figura 5.

El decodificador eSBR 400 de la figura 5 incluye la memoria 201 de búfer (que es idéntica a la memoria 201 de las figuras 3 y 4), el desformateador 215 de carga útil de flujo de bits (que es idéntico al desformateador 215 de la figura 4), el subsistema 202 de decodificación de audio (a veces denominada etapa de decodificación "central" o subsistema de decodificación "central", y que es idéntico al subsistema 202 de decodificación de la figura 3), el subsistema 401 de generación de datos de control eSBR y la etapa 203 de procesamiento de eSBR (que es idéntica a la etapa 203 de la figura 3), conectados como se muestra. También típicamente, el decodificador 400 incluye otros elementos de procesamiento (no mostrados).

En el funcionamiento del decodificador 400, una secuencia de bloques de un flujo de bits de audio codificado (un flujo de bits MPEG-4 AAC) recibido por el decodificador 400 se confirma desde el búfer 201 al desformateador 215.

El desformateador 215 está acoplado y configurado para demultiplexar cada bloque del flujo de bits para extraer metadatos SBR (incluidos datos de envolvente cuantificados) y típicamente también otros metadatos de los mismos. El desformateador 215 está configurado para confirmar al menos los metadatos SBR en la etapa 203 de procesamiento de eSBR. El desformateador 215 también está acoplado y configurado para extraer datos de audio de cada bloque del flujo de bits y para confirmar los datos de audio extraídos en el subsistema 202 de decodificación (etapa de decodificación).

El subsistema 202 de decodificación de audio del decodificador 400 está configurado para decodificar los datos de audio extraídos por el desformateador 215 (tal decodificación puede denominarse operación de decodificación "central") para generar datos de audio decodificados y para confirmar los datos de audio decodificados en la etapa 203 de procesamiento eSBR. La decodificación se realiza en el dominio frecuencia. Típicamente, una etapa final de procesamiento en el subsistema 202 aplica una transformación de dominio frecuencia a dominio tiempo a los datos

de audio de dominio frecuencia decodificados, de modo que la salida del subsistema son datos de audio decodificados de dominio tiempo. La etapa 203 está configurada para aplicar herramientas SBR (y herramientas eSBR) indicadas por los metadatos SBR (extraídos por el desformateador 215) y por los metadatos eSBR generados en el subsistema 401, a los datos de audio decodificados (es decir, para realizar el procesamiento SBR y eSBR en la salida del subsistema 202 de decodificación usando los metadatos SBR y eSBR) para generar los datos de audio completamente decodificados que se emiten desde el decodificador 400. Típicamente, el decodificador 400 incluye una memoria (accesible por el subsistema 202 y la etapa 203) que almacena la salida de datos y metadatos de audio desformateados del desformateador 215 (y opcionalmente también el subsistema 401), y la etapa 203 está configurada para acceder a los datos de audio y metadatos según sea necesario durante el procesamiento de SBR y eSBR. Se puede considerar que el procesamiento SBR en la etapa 203 es un posprocesamiento en la salida del subsistema 202 de decodificación central. Opcionalmente, el decodificador 400 también incluye un subsistema de mezcla ascendente final (que puede aplicar herramientas estéreo paramétricas ("PS") definidas en el estándar MPEG-4 AAC, usando metadatos PS extraídos por el desformateador 215) que está acoplado y configurado para realizar mezcla ascendente en la salida de la etapa 203 para generar audio de mezcla ascendente y totalmente decodificado que se emite desde la APU 210.

El estéreo paramétrico es una herramienta de codificación que representa una señal estéreo usando una mezcla descendente lineal de los canales izquierdo y derecho de la señal estéreo y conjuntos de parámetros espaciales que describen la imagen estéreo. El estéreo paramétrico típicamente emplea tres tipos de parámetros espaciales: (1) diferencias de intensidad entre canales (IID) que describen las diferencias de intensidad entre los canales; (2) diferencias de fase entre canales (IPD) que describen las diferencias de fase entre los canales; y (3) coherencia entre canales (ICC) que describe la coherencia (o similitud) entre los canales. La coherencia puede medirse como el máximo de la correlación cruzada en función del tiempo o la fase. Estos tres parámetros generalmente permiten una reconstrucción de alta calidad de la imagen estéreo. Sin embargo, los parámetros de IPD solo especifican las diferencias de fase relativas entre los canales de la señal de entrada estéreo y no indican la distribución de estas diferencias de fase en los canales izquierdo y derecho. Por lo tanto, se puede usar adicionalmente un cuarto tipo de parámetro que describe un desplazamiento de fase general o una diferencia de fase general (OPD). En el proceso de reconstrucción estéreo, los segmentos de ventana consecutivos de tanto la señal de mezcla descendente recibida, $s[n]$, como una versión descorrelacionada de la mezcla descendente recibida, $d[n]$, se procesan junto con los parámetros espaciales para generar la izquierda ($l_k(n)$) y derecha ($r_k(n)$) reconstruyeron las señales de acuerdo con:

$$l_k(n) = H_{11}(k,n)s_k(n) + H_{21}(k,n)d_k(n)$$

$$r_k(n) = H_{12}(k,n)s_k(n) + H_{22}(k,n)d_k(n)$$

donde H_{11} , H_{12} , H_{21} y H_{22} están definidos por los parámetros estéreo. Las señales $l_k(n)$ y $r_k(n)$ se transforman finalmente de nuevo al dominio tiempo mediante una transformación de frecuencia a tiempo.

El subsistema 401 de generación de datos de control de la figura 5 está acoplado y configurado para detectar al menos una propiedad del flujo de bits de audio codificado que se va a decodificar, y para generar datos de control eSBR (que pueden ser o incluir metadatos eSBR de cualquiera de los tipos incluidos en flujos de bits de audio codificados de acuerdo con otras realizaciones de la invención) en respuesta a al menos un resultado del paso de detección. Los datos de control eSBR se confirman en la etapa 203 para activar la aplicación de herramientas eSBR individuales o combinaciones de herramientas eSBR al detectar una propiedad específica (o combinación de propiedades) del flujo de bits y/o para controlar la aplicación de tales herramientas eSBR. Por ejemplo, para controlar el rendimiento del procesamiento de eSBR mediante la transposición armónica, algunas realizaciones del subsistema 401 de generación de datos de control incluirían: un detector de música (por ejemplo, una versión simplificada de un detector de música convencional) para configurar el parámetro `sbrPatchingMode[ch]` (y confirmar el parámetro establecido en la etapa 203) en respuesta a la detección de que el flujo de bits es o no indicativo de música; un detector de transitorios para establecer el parámetro `sbrOversamplingFlag[ch]` (y confirmar el parámetro establecido en la etapa 203) en respuesta a la detección de la presencia o ausencia de transitorios en el contenido de audio indicado por el flujo de bits; y/o un detector de tono para establecer los parámetros `sbrPitchInBinsFlag[ch]` y `sbrPitchInBins[ch]` (y confirmar los parámetros establecidos en la etapa 203) en respuesta a la detección del tono del contenido de audio indicado por el flujo de bits. Otros aspectos de la invención son los métodos de decodificación de flujo de bits de audio realizados por cualquier realización del decodificador de la invención descrito en este párrafo y en el párrafo anterior.

Los aspectos de la invención incluyen un método de codificación o decodificación del tipo para el que está configurada (por ejemplo, programada) cualquier realización de la APU, sistema o dispositivo de la invención. Otros aspectos de la invención incluyen un sistema o dispositivo configurado (por ejemplo, programado) para realizar cualquier realización del método de la invención, y un medio legible por computadora (por ejemplo, un disco) que almacena código (por ejemplo, de manera no transitoria) para implementar cualquier realización del método de la invención o pasos del mismo. Por ejemplo, el sistema de la invención puede ser o incluir un procesador de propósito

general programable, procesador de señal digital o microprocesador, programado con software o firmware y/o configurado de otra manera para realizar cualquiera de una variedad de operaciones sobre datos, incluida una realización del método de la invención o pasos del mismo. 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 programado (y/o configurado de otra manera) para realizar una realización del método de la invención (o pasos del mismo) en respuesta a los datos confirmados en el mismo.

Las realizaciones de la presente invención pueden implementarse en hardware, firmware o software, o una combinación de ambos (por ejemplo, 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 u otro aparato en particular. En particular, se pueden usar varias máquinas de uso general con programas escritos de acuerdo con las enseñanzas del presente documento, o puede ser más conveniente construir aparatos más especializados (por ejemplo, circuitos integrados) para realizar los pasos requeridos del método. Por tanto, la invención puede implementarse en uno o más programas informáticos que se ejecutan en uno o más sistemas informáticos programables (por ejemplo, una implementación de cualquiera de los elementos de la figura 1, o el codificador 100 de la figura 2 (o un elemento del mismo), o decodificador 200 de la figura 3 (o un elemento del mismo), o decodificador 210 de la figura 4 (o un elemento del mismo), o decodificador 400 de la figura 5 (o un elemento del mismo)) cada uno de los cuales comprende al menos un procesador, en al menos un sistema de almacenamiento de datos (que incluye memoria y/o elementos de almacenamiento volátiles y no volátiles), al menos un dispositivo o puerto de entrada y al menos un dispositivo o puerto de salida. El código de programa se aplica a los datos de entrada para 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.

Cada uno de estos programas puede implementarse en cualquier lenguaje informático deseado (incluyendo máquina, ensamblaje o lenguajes de programación de procedimiento, lógico u orientado a objetos de alto nivel) para comunicarse con un sistema informático. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado.

Por ejemplo, cuando se implementan mediante secuencias de instrucciones de software informático, varias funciones y pasos de realizaciones de la invención pueden implementarse mediante secuencias de instrucciones de software multiproceso que se ejecutan en hardware de procesamiento de señales digitales adecuado, en cuyo caso los diversos dispositivos, pasos y funciones de las realizaciones pueden corresponder a porciones de las instrucciones del software.

Cada uno de estos programas informáticos se almacena o se descarga preferiblemente en un medio o dispositivo de almacenamiento (por ejemplo, memoria o medio de estado sólido, o medio magnético u óptico) legible por una computadora programable de propósito general o especial, para configurar y operar la computadora cuando el medio o dispositivo de almacenamiento es leído por el sistema informático para realizar los procedimientos descritos en el presente documento. El sistema de la invención también puede implementarse como un medio de almacenamiento legible por computadora, configurado con (es decir, almacenando) un programa informático, donde el medio de almacenamiento así configurado hace que un sistema informático opere de una manera específica y predefinida para realizar las funciones descritas en el presente documento.

Se han descrito varias realizaciones de la invención. No obstante, se entenderá que se pueden realizar varias modificaciones. Son posibles numerosas modificaciones y variaciones de la presente invención a la luz de las enseñanzas anteriores. Por ejemplo, para facilitar implementaciones eficientes, los cambios de fase pueden usarse en combinación con los bancos de filtros de síntesis y análisis QMF complejos. El banco de filtros de análisis es responsable de filtrar la señal de banda baja en el dominio tiempo generada por el decodificador central en una pluralidad de subbandas (por ejemplo, subbandas QMF). El banco de filtros de síntesis es responsable de combinar la banda alta regenerada producida por la técnica HFR seleccionada (como lo indica el parámetro `sbrPatchingMode` recibido) con la banda baja decodificada para producir una señal de audio de salida de banda ancha. Sin embargo, una implementación de banco de filtros dada que opera en un cierto modo de frecuencia de muestreo, por ejemplo, operación normal de tasa dual o modo SBR con muestreo descendente, no debería tener cambios de fase que dependan del flujo de bits. Los bancos QMF usados en SBR son una extensión exponencial compleja de la teoría de los bancos de filtros modulados por coseno. Se puede demostrar que las restricciones de cancelación de alias se vuelven obsoletas cuando se amplía el banco de filtros con modulación de coseno con modulación exponencial compleja. Por lo tanto, para los bancos QMF de SBR, tanto los filtros de análisis, $h_k(n)$ como los filtros de síntesis, $f_k(n)$, pueden definirse por:

$$h_k(n) = f_k(n) = p_0(n) \exp \left\{ i \frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(n - \frac{N}{2} \right) \right\}, \quad 0 \leq n \leq N; \quad 0 \leq k < M \quad (1)$$

donde $p_0(n)$ es un filtro prototipo simétrico o asimétrico de valor real (típicamente, un filtro prototipo de paso bajo), M denota el número de canales y N es el orden del filtro prototipo. El número de canales usados en el banco de filtros de análisis puede ser diferente al número de canales usados en el banco de filtros de síntesis. Por ejemplo, el banco de filtros de análisis puede tener 32 canales y el banco de filtros de síntesis puede tener 64 canales. Cuando se

- opera el banco de filtros de síntesis en modo de muestreo descendente, el banco de filtros de síntesis puede tener solo 32 canales. Dado que las muestras de subbanda del banco de filtros tienen un valor complejo, puede añadirse al banco de filtros de análisis un paso de cambio de fase aditivo posiblemente dependiente del canal. Estos cambios de fase adicionales deben compensarse antes del banco de filtros de síntesis. Si bien los términos de cambio de fase en principio pueden ser de valores arbitrarios sin destruir el funcionamiento de la cadena de análisis/síntesis de QMF, también pueden estar limitados a ciertos valores para la verificación de conformidad. La señal SBR se verá afectada por la elección de los factores de fase, mientras que la señal de paso bajo que proviene del decodificador central no lo hará. La calidad de audio de la señal de salida no se ve afectada.
- 5
- 10 Los coeficientes del filtro prototipo, $p_0(n)$, pueden definirse con una longitud, L , de 640, como se muestra en la tabla 4 a continuación.

Tabla 4

n	$p_0(n)$	n	$p_0(n)$	n	$p_0(n)$
0	0.0000000000	214	0.0019765601	428	0.0117623832
1	-0.0005525286	215	-0.0032086896	429	0.0163701258
2	-0.0005617692	216	-0.0085711749	430	0.0207997072
3	-0.0004947518	217	-0.0141288827	431	0.0250307561
4	-0.0004875227	218	-0.0198834129	432	0.0290824006
5	-0.0004893791	219	-0.0258227288	433	0.0329583930
6	-0.0005040714	220	-0.0319531274	434	0.0366418116
7	-0.0005226564	221	-0.0382776572	435	0.0401458278
8	-0.0005466565	222	-0.0447806821	436	0.0434768782
9	-0.0005677802	223	-0.0514804176	437	0.0466303305
10	-0.0005870930	224	-0.0583705326	438	0.0495978676
11	-0.0006132747	225	-0.0654409853	439	0.0524093821
12	-0.0006312493	226	-0.0726943300	440	0.0550460034
13	-0.0006540333	227	-0.0801372934	441	0.0575152691
14	-0.0006777690	228	-0.0877547536	442	0.0598166570
15	-0.0006941614	229	-0.0955533352	443	0.0619602779
16	-0.0007157736	230	-0.1035329531	444	0.0639444805
17	-0.0007255043	231	-0.1116826931	445	0.0657690668
18	-0.0007440941	232	-0.1200077984	446	0.0674525021
19	-0.0007490598	233	-0.1285002850	447	0.0689664013
20	-0.0007681371	234	-0.1371551761	448	0.0703533073
21	-0.0007724848	235	-0.1459766491	449	0.0715826364
22	-0.0007834332	236	-0.1549607071	450	0.0726774642
23	-0.0007779869	237	-0.1640958855	451	0.0736406005
24	-0.0007803664	238	-0.1733808172	452	0.0744664394
25	-0.0007801449	239	-0.1828172548	453	0.0751576255
26	-0.0007757977	240	-0.1923966745	454	0.0757305756
27	-0.0007630793	241	-0.2021250176	455	0.0761748321
28	-0.0007530001	242	-0.2119735853	456	0.0765050718
29	-0.0007319357	243	-0.2219652696	457	0.0767204924
30	-0.0007215391	244	-0.2320690870	458	0.0768230011
31	-0.0006917937	245	-0.2423016884	459	0.0768173975

ES 2 948 839 T3

n	po(n)	n	po(n)	n	po(n)
32	-0.0006650415	246	-0.2526480309	460	0.0767093490
33	-0.0006341594	247	-0.2631053299	461	0.0764992170
34	-0.0005946118	248	-0.2736634040	462	0.0761992479
35	-0.0005564576	249	-0.2843214189	463	0.0758008358
36	-0.0005145572	250	-0.2950716717	464	0.0753137336
37	-0.0004606325	251	-0.3059098575	465	0.0747452558
38	-0.0004095121	252	-0.3168278913	466	0.0741003642
39	-0.0003501175	253	-0.3278113727	467	0.0733620255
40	-0.0002896981	254	-0.3388722693	468	0.0725682583
41	-0.0002098337	255	-0.3499914122	469	0.0717002673
42	-0.0001446380	256	0.3611589903	470	0.0707628710
43	-0.0000617334	257	0.3723795546	471	0.0697630244
44	0.0000134949	258	0.3836350013	472	0.0687043828
45	0.0001094383	259	0.3949211761	473	0.0676075985
46	0.0002043017	260	0.4062317676	474	0.0664367512
47	0.0002949531	261	0.4175696896	475	0.0652247106
48	0.0004026540	262	0.4289119920	476	0.0639715898
49	0.0005107388	263	0.4402553754	477	0.0626857808
50	0.0006239376	264	0.4515996535	478	0.0613455171
51	0.0007458025	265	0.4629308085	479	0.0599837480
52	0.0008608443	266	0.4742453214	480	0.0585915683
53	0.0009885988	267	0.4855253091	481	0.0571616450
54	0.0011250155	268	0.4967708254	482	0.0557173648
55	0.0012577884	269	0.5079817500	483	0.0542452768
56	0.0013902494	270	0.5191234970	484	0.0527630746
57	0.0015443219	271	0.5302240895	485	0.0512556155
58	0.0016868083	272	0.5412553448	486	0.0497385755
59	0.0018348265	273	0.5522051258	487	0.0482165720
60	0.0019841140	274	0.5630789140	488	0.0466843027
61	0.0021461583	275	0.5738524131	489	0.0451488405
62	0.0023017254	276	0.5845403235	490	0.0436097542
63	0.0024625616	277	0.5951123086	491	0.0420649094
64	0.0026201758	278	0.6055783538	492	0.0405349170
65	0.0027870464	279	0.6159109932	493	0.0390053679
66	0.0029469447	280	0.6261242695	494	0.0374812850
67	0.0031125420	281	0.6361980107	495	0.0359697560
68	0.0032739613	282	0.6461269695	496	0.0344620948
69	0.0034418874	283	0.6559016302	497	0.0329754081
70	0.0036008268	284	0.6655139880	498	0.0315017608
71	0.0037603922	285	0.6749663190	499	0.0300502657

ES 2 948 839 T3

n	po(n)	n	po(n)	n	po(n)
72	0.0039207432	286	0.6842353293	500	0.0286072173
73	0.0040819753	287	0.6933282376	501	0.0271859429
74	0.0042264269	288	0.7022388719	502	0.0257875847
75	0.0043730719	289	0.7109410426	503	0.0244160992
76	0.0045209852	290	0.7194462634	504	0.0230680169
77	0.0046606460	291	0.7277448900	505	0.0217467550
78	0.0047932560	292	0.7358211758	506	0.0204531793
79	0.0049137603	293	0.7436827863	507	0.0191872431
80	0.0050393022	294	0.7513137456	508	0.0179433381
81	0.0051407353	295	0.7587080760	509	0.0167324712
82	0.0052461166	296	0.7658674865	510	0.0155405553
83	0.0053471681	297	0.7727780881	511	0.0143904666
84	0.0054196775	298	0.7794287519	512	-0.0132718220
85	0.0054876040	299	0.7858353120	513	-0.0121849995
86	0.0055475714	300	0.7919735841	514	-0.0111315548
87	0.0055938023	301	0.7978466413	515	-0.0101150215
88	0.0056220643	302	0.8034485751	516	-0.0091325329
89	0.0056455196	303	0.8087695004	517	-0.0081798233
90	0.0056389199	304	0.8138191270	518	-0.0072615816
91	0.0056266114	305	0.8185776004	519	-0.0063792293
92	0.0055917128	306	0.8230419890	520	-0.0055337211
93	0.0055404363	307	0.8272275347	521	-0.0047222596
94	0.0054753783	308	0.8311038457	522	-0.0039401124
95	0.0053838975	309	0.8346937361	523	-0.0031933778
96	0.0052715758	310	0.8379717337	524	-0.0024826723
97	0.0051382275	311	0.8409541392	525	-0.0018039472
98	0.0049839687	312	0.8436238281	526	-0.0011568135
99	0.0048109469	313	0.8459818469	527	-0.0005464280
100	0.0046039530	314	0.8480315777	528	0.0000276045
101	0.0043801861	315	0.8497805198	529	0.0005832264
102	0.0041251642	316	0.8511971524	530	0.0010902329
103	0.0038456408	317	0.8523047035	531	0.0015784682
104	0.0035401246	318	0.8531020949	532	0.0020274176
105	0.0032091885	319	0.8535720573	533	0.0024508540
106	0.0028446757	320	0.8537385600	534	0.0028446757
107	0.0024508540	321	0.8535720573	535	0.0032091885
108	0.0020274176	322	0.8531020949	536	0.0035401246
109	0.0015784682	323	0.8523047035	537	0.0038456408
110	0.0010902329	324	0.8511971524	538	0.0041251642
111	0.0005832264	325	0.8497805198	539	0.0043801861

ES 2 948 839 T3

n	po(n)	n	po(n)	n	po(n)
112	0.0000276045	326	0.8480315777	540	0.0046039530
113	-0.0005464280	327	0.8459818469	541	0.0048109469
114	-0.0011568135	328	0.8436238281	542	0.0049839687
115	-0.0018039472	329	0.8409541392	543	0.0051382275
116	-0.0024826723	330	0.8379717337	544	0.0052715758
117	-0.0031933778	331	0.8346937361	545	0.0053838975
118	-0.0039401124	332	0.8311038457	546	0.0054753783
119	-0.0047222596	333	0.8272275347	547	0.0055404363
120	-0.0055337211	334	0.8230419890	548	0.0055917128
121	-0.0063792293	335	0.8185776004	549	0.0056266114
122	-0.0072615816	336	0.8138191270	550	0.0056389199
123	-0.0081798233	337	0.8087695004	551	0.0056455196
124	-0.0091325329	338	0.8034485751	552	0.0056220643
125	-0.0101150215	339	0.7978466413	553	0.0055938023
126	-0.0111315548	340	0.7919735841	554	0.0055475714
127	-0.0121849995	341	0.7858353120	555	0.0054876040
128	0.0132718220	342	0.7794287519	556	0.0054196775
129	0.0143904666	343	0.7727780881	557	0.0053471681
130	0.0155405553	344	0.7658674865	558	0.0052461166
131	0.0167324712	345	0.7587080760	559	0.0051407353
132	0.0179433381	346	0.7513137456	560	0.0050393022
133	0.0191872431	347	0.7436827863	561	0.0049137603
134	0.0204531793	348	0.7358211758	562	0.0047932560
135	0.0217467550	349	0.7277448900	563	0.0046606460
136	0.0230680169	350	0.7194462634	564	0.0045209852
137	0.0244160992	351	0.7109410426	565	0.0043730719
138	0.0257875847	352	0.7022388719	566	0.0042264269
139	0.0271859429	353	0.6933282376	567	0.0040819753
140	0.0286072173	354	0.6842353293	568	0.0039207432
141	0.0300502657	355	0.6749663190	569	0.0037603922
142	0.0315017608	356	0.6655139880	570	0.0036008268
143	0.0329754081	357	0.6559016302	571	0.0034418874
144	0.0344620948	358	0.6461269695	572	0.0032739613
145	0.0359697560	359	0.6361980107	573	0.0031125420
146	0.0374812850	360	0.6261242695	574	0.0029469447
147	0.0390053679	361	0.6159109932	575	0.0027870464
148	0.0405349170	362	0.6055783538	576	0.0026201758
149	0.0420649094	363	0.5951123086	577	0.0024625616
150	0.0436097542	364	0.5845403235	578	0.0023017254
151	0.0451488405	365	0.5738524131	579	0.0021461583

ES 2 948 839 T3

n	po(n)	n	po(n)	n	po(n)
152	0.0466843027	366	0.5630789140	580	0.0019841140
153	0.0482165720	367	0.5522051258	581	0.0018348265
154	0.0497385755	368	0.5412553448	582	0.0016868083
155	0.0512556155	369	0.5302240895	583	0.0015443219
156	0.0527630746	370	0.5191234970	584	0.0013902494
157	0.0542452768	371	0.5079817500	585	0.0012577884
158	0.0557173648	372	0.4967708254	586	0.0011250155
159	0.0571616450	373	0.4855253091	587	0.0009885988
160	0.0585915683	374	0.4742453214	588	0.0008608443
161	0.0599837480	375	0.4629308085	589	0.0007458025
162	0.0613455171	376	0.4515996535	590	0.0006239376
163	0.0626857808	377	0.4402553754	591	0.0005107388
164	0.0639715898	378	0.4289119920	592	0.0004026540
165	0.0652247106	379	0.4175696896	593	0.0002949531
166	0.0664367512	380	0.4062317676	594	0.0002043017
167	0.0676075985	381	0.3949211761	595	0.0001094383
168	0.0687043828	382	0.3836350013	596	0.0000134949
169	0.0697630244	383	0.3723795546	597	-0.0000617334
170	0.0707628710	384	-0.3611589903	598	-0.0001446380
171	0.0717002673	385	-0.3499914122	599	-0.0002098337
172	0.0725682583	386	-0.3388722693	600	-0.0002896981
173	0.0733620255	387	-0.3278113727	601	-0.0003501175
174	0.0741003642	388	-0.3168278913	602	-0.0004095121
175	0.0747452558	389	-0.3059098575	603	-0.0004606325
176	0.0753137336	390	-0.2950716717	604	-0.0005145572
177	0.0758008358	391	-0.2843214189	605	-0.0005564576
178	0.0761992479	392	-0.2736634040	606	-0.0005946118
179	0.0764992170	393	-0.2631053299	607	-0.0006341594
180	0.0767093490	394	-0.2526480309	608	-0.0006650415
181	0.0768173975	395	-0.2423016884	609	-0.0006917937
182	0.0768230011	396	-0.2320690870	610	-0.0007215391
183	0.0767204924	397	-0.2219652696	611	-0.0007319357
184	0.0765050718	398	-0.2119735853	612	-0.0007530001
185	0.0761748321	399	-0.2021250176	613	-0.0007630793
186	0.0757305756	400	-0.1923966745	614	-0.0007757977
187	0.0751576255	401	-0.1828172548	615	-0.0007801449
188	0.0744664394	402	-0.1733808172	616	-0.0007803664
189	0.0736406005	403	-0.1640958855	617	-0.0007779869
190	0.0726774642	404	-0.1549607071	618	-0.0007834332
191	0.0715826364	405	-0.1459766491	619	-0.0007724848

n	po(n)	n	po(n)	n	po(n)
192	0.0703533073	406	-0.1371551761	620	-0.0007681371
193	0.0689664013	407	-0.1285002850	621	-0.0007490598
194	0.0674525021	408	-0.1200077984	622	-0.0007440941
195	0.0657690668	409	-0.1116826931	623	-0.0007255043
196	0.0639444805	410	-0.1035329531	624	-0.0007157736
197	0.0619602779	411	-0.0955533352	625	-0.0006941614
198	0.0598166570	412	-0.0877547536	626	-0.0006777690
199	0.0575152691	413	-0.0801372934	627	-0.0006540333
200	0.0550460034	414	-0.0726943300	628	-0.0006312493
201	0.0524093821	415	-0.0654409853	629	-0.0006132747
202	0.0495978676	416	-0.0583705326	630	-0.0005870930
203	0.0466303305	417	-0.0514804176	631	-0.0005677802
204	0.0434768782	418	-0.0447806821	632	-0.0005466565
205	0.0401458278	419	-0.0382776572	633	-0.0005226564
206	0.0366418116	420	-0.0319531274	634	-0.0005040714
207	0.0329583930	421	-0.0258227288	635	-0.0004893791
208	0.0290824006	422	-0.0198834129	636	-0.0004875227
209	0.0250307561	423	-0.0141288827	637	-0.0004947518
210	0.0207997072	424	-0.0085711749	638	-0.0005617692
211	0.0163701258	425	-0.0032086896	639	-0.0005525280
212	0.0117623832	426	0.0019765601		
213	0.0069636862	427	0.0069636862		

El filtro prototipo, $p_0(n)$, también puede derivarse de la tabla 4 mediante una o más operaciones matemáticas como redondeo, submuestreo, interpolación y decimación.

- 5 Aunque el ajuste de la información de control relacionada con SBR no depende típicamente de los detalles de la transposición (como se explicó anteriormente), en algunas realizaciones, ciertos elementos de los datos de control pueden ser simultaneados en el contenedor de extensión eSBR (`bs_extension_id == EXTENSION_ID_ESBR`) para mejorar la calidad de la señal regenerada. Algunos de los elementos simultaneados pueden incluir los datos del ruido de fondo (por ejemplo, factores de escala del ruido de fondo y un parámetro que indica la dirección, ya sea en la dirección de frecuencia o en la del tiempo, de la codificación delta para cada ruido de fondo), los datos de filtrado inverso (por ejemplo, un parámetro que indica el modo de filtrado inverso seleccionado entre sin filtrado inverso, un nivel bajo de filtrado inverso, un nivel intermedio de filtrado inverso y un nivel alto de filtrado inverso) y los datos de armónicos faltantes (por ejemplo, un parámetro que indica si debe agregarse una senoide a una banda de frecuencia específica de la banda alta regenerada). Todos estos elementos se basan en una emulación sintetizada del transpondedor del decodificador realizada en el codificador y, por lo tanto, si se sintonizan correctamente para el transpondedor seleccionado, pueden aumentar la calidad de la señal regenerada.

- 20 Específicamente, en algunas realizaciones, los armónicos faltantes y los datos de control de filtrado inverso se transmiten en el contenedor de extensión de eSBR (junto con los otros parámetros de flujo de bits de la tabla 3) y se sintonizan para el transpondedor de armónicos de eSBR. La tasa de bits adicional necesaria para transmitir estas dos clases de metadatos para el transpondedor armónico de eSBR es relativamente baja. Por lo tanto, el envío de datos de control de filtrado inverso y/o armónicos faltantes sintonizados en el contenedor de extensión eSBR aumentará la calidad del audio producido por el transpondedor y afectará solo mínimamente a la tasa de bits. Para garantizar la retrocompatibilidad con los decodificadores heredados, los parámetros sintonizados para la operación de traslación espectral de SBR también pueden enviarse en el flujo de bits como parte de los datos de control de SBR usando señalización implícita o explícita.

Debe entenderse que dentro del alcance de las reivindicaciones adjuntas, la invención se puede poner en práctica de varias formas. Los números de referencia contenidos en las siguientes reivindicaciones son solo para fines ilustrativos y no deben usarse para interpretar o limitar las reivindicaciones de ninguna manera.

REIVINDICACIONES

1.- Un método para realizar la reconstrucción de alta frecuencia de una señal de audio, comprendiendo el método:

- 5 recibir un flujo de bits de audio codificado, incluyendo el flujo de bits de audio codificado datos de audio que representan una porción de banda baja de la señal de audio y metadatos de reconstrucción de alta frecuencia;

decodificar los datos de audio para generar una señal de audio de banda baja decodificada;

- 10 extraer del flujo de bits de audio codificado los metadatos de reconstrucción de alta frecuencia, los metadatos de reconstrucción de alta frecuencia incluyendo los parámetros operativos para un proceso de reconstrucción de alta frecuencia, los parámetros operativos incluyendo un parámetro de modo de parcheo ubicado en un contenedor de extensión retrocompatible del flujo de bits de audio codificado, donde un primer valor del parámetro de modo de parcheo indica traslación espectral y un segundo valor del parámetro de modo de parcheo indica transposición armónica por expansión de frecuencia de codificador de voz de fase;

filtrar la señal de audio de banda baja decodificada para generar una señal de audio de banda baja filtrada;

- 20 regenerar una porción de banda alta de la señal de audio usando la señal de audio de banda baja filtrada y los metadatos de reconstrucción de alta frecuencia, donde la regeneración incluye la traslación espectral si el parámetro de modo de parcheo es el primer valor y la regeneración incluye la transposición armónica mediante la expansión de frecuencia del codificador de voz de fase si el parámetro de modo de parcheo es el segundo valor; y

- 25 combinar la señal de audio de banda baja filtrada con la porción de banda alta regenerada para formar una señal de audio de banda ancha;

donde el flujo de bits de audio codificado incluye además un elemento de relleno con un identificador que indica un inicio del elemento de relleno y datos de relleno después del identificador, donde los datos de relleno incluyen el contenedor de extensión retrocompatible, donde los datos de relleno incluyen una carga útil de extensión, la carga útil de extensión incluye datos de extensión de replicación de banda espectral, y la carga útil de extensión se identifica con un entero de cuatro bits sin signo transmitido primero con el bit más significativo y que tiene un valor de '1101' o '1110';

- 35 en donde los metadatos de reconstrucción de alta frecuencia incluyen factores de escala de envolvente.

2.- El método de la reivindicación 1, donde el filtrado se realiza mediante un banco de filtros de análisis que incluye filtros de análisis, $h_k(n)$, que son versiones moduladas de un filtro prototipo, $p_0(n)$, de acuerdo con:

$$h_k(n) = p_0(n) \exp \left\{ i \frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(n - \frac{N}{2} \right) \right\}, \quad 0 \leq n \leq N; \quad 0 \leq k < M$$

- 40 donde $p_0(n)$ es un filtro prototipo simétrico o asimétrico de valor real, M es un número de canales en el banco de filtros de análisis y N es un orden del filtro prototipo.

3.- El método de la reivindicación 2, donde el filtro prototipo, $p_0(n)$, se deriva de los coeficientes de la siguiente tabla.

n	$p_0(n)$	n	$p_0(n)$	n	$p_0(n)$
0	0.0000000000	214	0.0019765601	428	0.0117623832
1	-0.0005525286	215	-0.0032086896	429	0.0163701258
2	-0.0005617692	216	-0.0085711749	430	0.0207997072
3	-0.0004947518	217	-0.0141288827	431	0.0250307561
4	-0.0004875227	218	-0.0198834129	432	0.0290824006
5	-0.0004893791	219	-0.0258227288	433	0.0329583930
6	-0.0005040714	220	-0.0319531274	434	0.0366418116
7	-0.0005226564	221	-0.0382776572	435	0.0401458278
8	-0.0005466565	222	-0.0447806821	436	0.0434768782
9	-0.0005677802	223	-0.0514804176	437	0.0466303305
10	-0.0005870930	224	-0.0583705326	438	0.0495978676

ES 2 948 839 T3

n	po(n)	n	po(n)	n	po(n)
11	-0.0006132747	225	-0.0654409853	439	0.0524093821
12	-0.0006312493	226	-0.0726943300	440	0.0550460034
13	-0.0006540333	227	-0.0801372934	441	0.0575152691
14	-0.0006777690	228	-0.0877547536	442	0.0598166570
15	-0.0006941614	229	-0.0955533352	443	0.0619602779
16	-0.0007157736	230	-0.1035329531	444	0.0639444805
17	-0.0007255043	231	-0.1116826931	445	0.0657690668
18	-0.0007440941	232	-0.1200077984	446	0.0674525021
19	-0.0007490598	233	-0.1285002850	447	0.0689664013
20	-0.0007681371	234	-0.1371551761	448	0.0703533073
21	-0.0007724848	235	-0.1459766491	449	0.0715826364
22	-0.0007834332	236	-0.1549607071	450	0.0726774642
23	-0.0007779869	237	-0.1640958855	451	0.0736406005
24	-0.0007803664	238	-0.1733808172	452	0.0744664394
25	-0.0007801449	239	-0.1828172548	453	0.0751576255
26	-0.0007757977	240	-0.1923966745	454	0.0757305756
27	-0.0007630793	241	-0.2021250176	455	0.0761748321
28	-0.0007530001	242	-0.2119735853	456	0.0765050718
29	-0.0007319357	243	-0.2219652696	457	0.0767204924
30	-0.0007215391	244	-0.2320690870	458	0.0768230011
31	-0.0006917937	245	-0.2423016884	459	0.0768173975
32	-0.0006650415	246	-0.2526480309	460	0.0767093490
33	-0.0006341594	247	-0.2631053299	461	0.0764992170
34	-0.0005946118	248	-0.2736634040	462	0.0761992479
35	-0.0005564576	249	-0.2843214189	463	0.0758008358
36	-0.0005145572	250	-0.2950716717	464	0.0753137336
37	-0.0004606325	251	-0.3059098575	465	0.0747452558
38	-0.0004095121	252	-0.3168278913	466	0.0741003642
39	-0.0003501175	253	-0.3278113727	467	0.0733620255
40	-0.0002896981	254	-0.3388722693	468	0.0725682583
41	-0.0002098337	255	-0.3499914122	469	0.0717002673
42	-0.0001446380	256	0.3611589903	470	0.0707628710
43	-0.0000617334	257	0.3723795546	471	0.0697630244
44	0.0000134949	258	0.3836350013	472	0.0687043828
45	0.0001094383	259	0.3949211761	473	0.0676075985
46	0.0002043017	260	0.4062317676	474	0.0664367512
47	0.0002949531	261	0.4175696896	475	0.0652247106
48	0.0004026540	262	0.4289119920	476	0.0639715898
49	0.0005107388	263	0.4402553754	477	0.0626857808
50	0.0006239376	264	0.4515996535	478	0.0613455171

ES 2 948 839 T3

n	po(n)	n	po(n)	n	po(n)
51	0.0007458025	265	0.4629308085	479	0.0599837480
52	0.0008608443	266	0.4742453214	480	0.0585915683
53	0.0009885988	267	0.4855253091	481	0.0571616450
54	0.0011250155	268	0.4967708254	482	0.0557173648
55	0.0012577884	269	0.5079817500	483	0.0542452768
56	0.0013902494	270	0.5191234970	484	0.0527630746
57	0.0015443219	271	0.5302240895	485	0.0512556155
58	0.0016868083	272	0.5412553448	486	0.0497385755
59	0.0018348265	273	0.5522051258	487	0.0482165720
60	0.0019841140	274	0.5630789140	488	0.0466843027
61	0.0021461583	275	0.5738524131	489	0.0451488405
62	0.0023017254	276	0.5845403235	490	0.0436097542
63	0.0024625616	277	0.5951123086	491	0.0420649094
64	0.0026201758	278	0.6055783538	492	0.0405349170
65	0.0027870464	279	0.6159109932	493	0.0390053679
66	0.0029469447	280	0.6261242695	494	0.0374812850
67	0.0031125420	281	0.6361980107	495	0.0359697560
68	0.0032739613	282	0.6461269695	496	0.0344620948
69	0.0034418874	283	0.6559016302	497	0.0329754081
70	0.0036008268	284	0.6655139880	498	0.0315017608
71	0.0037603922	285	0.6749663190	499	0.0300502657
72	0.0039207432	286	0.6842353293	500	0.0286072173
73	0.0040819753	287	0.6933282376	501	0.0271859429
74	0.0042264269	288	0.7022388719	502	0.0257875847
75	0.0043730719	289	0.7109410426	503	0.0244160992
76	0.0045209852	290	0.7194462634	504	0.0230680169
77	0.0046606460	291	0.7277448900	505	0.0217467550
78	0.0047932560	292	0.7358211758	506	0.0204531793
79	0.0049137603	293	0.7436827863	507	0.0191872431
80	0.0050393022	294	0.7513137456	508	0.0179433381
81	0.0051407353	295	0.7587080760	509	0.0167324712
82	0.0052461166	296	0.7658674865	510	0.0155405553
83	0.0053471681	297	0.7727780881	511	0.0143904666
84	0.0054196775	298	0.7794287519	512	-0.0132718220
85	0.0054876040	299	0.7858353120	513	-0.0121849995
86	0.0055475714	300	0.7919735841	514	-0.0111315548
87	0.0055938023	301	0.7978466413	515	-0.0101150215
88	0.0056220643	302	0.8034485751	516	-0.0091325329
89	0.0056455196	303	0.8087695004	517	-0.0081798233
90	0.0056389199	304	0.8138191270	518	-0.0072615816

ES 2 948 839 T3

n	po(n)	n	po(n)	n	po(n)
91	0.0056266114	305	0.8185776004	519	-0.0063792293
92	0.0055917128	306	0.8230419890	520	-0.0055337211
93	0.0055404363	307	0.8272275347	521	-0.0047222596
94	0.0054753783	308	0.8311038457	522	-0.0039401124
95	0.0053838975	309	0.8346937361	523	-0.0031933778
96	0.0052715758	310	0.8379717337	524	-0.0024826723
97	0.0051382275	311	0.8409541392	525	-0.0018039472
98	0.0049839687	312	0.8436238281	526	-0.0011568135
99	0.0048109469	313	0.8459818469	527	-0.0005464280
100	0.0046039530	314	0.8480315777	528	0.0000276045
101	0.0043801861	315	0.8497805198	529	0.0005832264
102	0.0041251642	316	0.8511971524	530	0.0010902329
103	0.0038456408	317	0.8523047035	531	0.0015784682
104	0.0035401246	318	0.8531020949	532	0.0020274176
105	0.0032091885	319	0.8535720573	533	0.0024508540
106	0.0028446757	320	0.8537385600	534	0.0028446757
107	0.0024508540	321	0.8535720573	535	0.0032091885
108	0.0020274176	322	0.8531020949	536	0.0035401246
109	0.0015784682	323	0.8523047035	537	0.0038456408
110	0.0010902329	324	0.8511971524	538	0.0041251642
111	0.0005832264	325	0.8497805198	539	0.0043801861
112	0.0000276045	326	0.8480315777	540	0.0046039530
113	-0.0005464280	327	0.8459818469	541	0.0048109469
114	-0.0011568135	328	0.8436238281	542	0.0049839687
115	-0.0018039472	329	0.8409541392	543	0.0051382275
116	-0.0024826723	330	0.8379717337	544	0.0052715758
117	-0.0031933778	331	0.8346937361	545	0.0053838975
118	-0.0039401124	332	0.8311038457	546	0.0054753783
119	-0.0047222596	333	0.8272275347	547	0.0055404363
120	-0.0055337211	334	0.8230419890	548	0.0055917128
121	-0.0063792293	335	0.8185776004	549	0.0056266114
122	-0.0072615816	336	0.8138191270	550	0.0056389199
123	-0.0081798233	337	0.8087695004	551	0.0056455196
124	-0.0091325329	338	0.8034485751	552	0.0056220643
125	-0.0101150215	339	0.7978466413	553	0.0055938023
126	-0.0111315548	340	0.7919735841	554	0.0055475714
127	-0.0121849995	341	0.7858353120	555	0.0054876040
128	0.0132718220	342	0.7794287519	556	0.0054196775
129	0.0143904666	343	0.7727780881	557	0.0053471681
130	0.0155405553	344	0.7658674865	558	0.0052461166

ES 2 948 839 T3

n	po(n)	n	po(n)	n	po(n)
131	0.0167324712	345	0.7587080760	559	0.0051407353
132	0.0179433381	346	0.7513137456	560	0.0050393022
133	0.0191872431	347	0.7436827863	561	0.0049137603
134	0.0204531793	348	0.7358211758	562	0.0047932560
135	0.0217467550	349	0.7277448900	563	0.0046606460
136	0.0230680169	350	0.7194462634	564	0.0045209852
137	0.0244160992	351	0.7109410426	565	0.0043730719
138	0.0257875847	352	0.7022388719	566	0.0042264269
139	0.0271859429	353	0.6933282376	567	0.0040819753
140	0.0286072173	354	0.6842353293	568	0.0039207432
141	0.0300502657	355	0.6749663190	569	0.0037603922
142	0.0315017608	356	0.6655139880	570	0.0036008268
143	0.0329754081	357	0.6559016302	571	0.0034418874
144	0.0344620948	358	0.6461269695	572	0.0032739613
145	0.0359697560	359	0.6361980107	573	0.0031125420
146	0.0374812850	360	0.6261242695	574	0.0029469447
147	0.0390053679	361	0.6159109932	575	0.0027870464
148	0.0405349170	362	0.6055783538	576	0.0026201758
149	0.0420649094	363	0.5951123086	577	0.0024625616
150	0.0436097542	364	0.5845403235	578	0.0023017254
151	0.0451488405	365	0.5738524131	579	0.0021461583
152	0.0466843027	366	0.5630789140	580	0.0019841140
153	0.0482165720	367	0.5522051258	581	0.0018348265
154	0.0497385755	368	0.5412553448	582	0.0016868083
155	0.0512556155	369	0.5302240895	583	0.0015443219
156	0.0527630746	370	0.5191234970	584	0.0013902494
157	0.0542452768	371	0.5079817500	585	0.0012577884
158	0.0557173648	372	0.4967708254	586	0.0011250155
159	0.0571616450	373	0.4855253091	587	0.0009885988
160	0.0585915683	374	0.4742453214	588	0.0008608443
161	0.0599837480	375	0.4629308085	589	0.0007458025
162	0.0613455171	376	0.4515996535	590	0.0006239376
163	0.0626857808	377	0.4402553754	591	0.0005107388
164	0.0639715898	378	0.4289119920	592	0.0004026540
165	0.0652247106	379	0.4175696896	593	0.0002949531
166	0.0664367512	380	0.4062317676	594	0.0002043017
167	0.0676075985	381	0.3949211761	595	0.0001094383
168	0.0687043828	382	0.3836350013	596	0.0000134949
169	0.0697630244	383	0.3723795546	597	-0.0000617334
170	0.0707628710	384	-0.3611589903	598	-0.0001446380

ES 2 948 839 T3

n	po(n)	n	po(n)	n	po(n)
171	0.0717002673	385	-0.3499914122	599	-0.0002098337
172	0.0725682583	386	-0.3388722693	600	-0.0002896981
173	0.0733620255	387	-0.3278113727	601	-0.0003501175
174	0.0741003642	388	-0.3168278913	602	-0.0004095121
175	0.0747452558	389	-0.3059098575	603	-0.0004606325
176	0.0753137336	390	-0.2950716717	604	-0.0005145572
177	0.0758008358	391	-0.2843214189	605	-0.0005564576
178	0.0761992479	392	-0.2736634040	606	-0.0005946118
179	0.0764992170	393	-0.2631053299	607	-0.0006341594
180	0.0767093490	394	-0.2526480309	608	-0.0006650415
181	0.0768173975	395	-0.2423016884	609	-0.0006917937
182	0.0768230011	396	-0.2320690870	610	-0.0007215391
183	0.0767204924	397	-0.2219652696	611	-0.0007319357
184	0.0765050718	398	-0.2119735853	612	-0.0007530001
185	0.0761748321	399	-0.2021250176	613	-0.0007630793
186	0.0757305756	400	-0.1923966745	614	-0.0007757977
187	0.0751576255	401	-0.1828172548	615	-0.0007801449
188	0.0744664394	402	-0.1733808172	616	-0.0007803664
189	0.0736406005	403	-0.1640958855	617	-0.0007779869
190	0.0726774642	404	-0.1549607071	618	-0.0007834332
191	0.0715826364	405	-0.1459766491	619	-0.0007724848
192	0.0703533073	406	-0.1371551761	620	-0.0007681371
193	0.0689664013	407	-0.1285002850	621	-0.0007490598
194	0.0674525021	408	-0.1200077984	622	-0.0007440941
195	0.0657690668	409	-0.1116826931	623	-0.0007255043
196	0.0639444805	410	-0.1035329531	624	-0.0007157736
197	0.0619602779	411	-0.0955533352	625	-0.0006941614
198	0.0598166570	412	-0.0877547536	626	-0.0006777690
199	0.0575152691	413	-0.0801372934	627	-0.0006540333
200	0.0550460034	414	-0.0726943300	628	-0.0006312493
201	0.0524093821	415	-0.0654409853	629	-0.0006132747
202	0.0495978676	416	-0.0583705326	630	-0.0005870930
203	0.0466303305	417	-0.0514804176	631	-0.0005677802
204	0.0434768782	418	-0.0447806821	632	-0.0005466565
205	0.0401458278	419	-0.0382776572	633	-0.0005226564
206	0.0366418116	420	-0.0319531274	634	-0.0005040714
207	0.0329583930	421	-0.0258227288	635	-0.0004893791
208	0.0290824006	422	-0.0198834129	636	-0.0004875227
209	0.0250307561	423	-0.0141288827	637	-0.0004947518
210	0.0207997072	424	-0.0085711749	638	-0.0005617692

n	$p_0(n)$	n	$p_0(n)$	n	$p_0(n)$
211	0.0163701258	425	-0.0032086896	639	-0.0005525280
212	0.0117623832	426	0.0019765601		
213	0.0069636862	427	0.0069636862		

4.- El método de la reivindicación 2, donde el filtro prototipo, $p_0(n)$, se deriva de los coeficientes de la tabla definida en la reivindicación 3 mediante una o más operaciones matemáticas seleccionadas del grupo que consiste en redondeo, submuestreo, interpolación o decimación.

5 5.- El método de una cualquiera de las reivindicaciones 1 a 4, donde se agrega un cambio de fase a la señal de audio de banda baja filtrada después del filtrado y se compensa antes de la combinación para reducir la complejidad del método.

10 6.- Un medio legible por computadora no transitorio que contiene instrucciones que cuando son ejecutadas por un procesador realizan el método de cualquiera de las reivindicaciones 1 a 5.

7.- Una unidad de procesamiento de audio para realizar la reconstrucción de alta frecuencia de una señal de audio, comprendiendo la unidad de procesamiento de audio:

15 una interfaz de entrada para recibir un flujo de bits de audio codificado, incluyendo el flujo de bits de audio codificado datos de audio que representan una porción de banda baja de la señal de audio y metadatos de reconstrucción de alta frecuencia;

20 un decodificador (202) de audio central para decodificar los datos de audio para generar una señal de audio de banda baja decodificada;

25 un desformateador (205, 215) para extraer del flujo de bits de audio codificado los metadatos de reconstrucción de alta frecuencia, los metadatos de reconstrucción de alta frecuencia incluyendo los parámetros operativos para un proceso de reconstrucción de alta frecuencia, los parámetros operativos incluyendo un parámetro de modo de parcheo ubicado en un contenedor de extensión retrocompatible del flujo de bits de audio codificado, donde un primer valor del parámetro de modo de parcheo indica traslación espectral y un segundo valor del parámetro de modo de parcheo indica transposición armónica mediante expansión de frecuencia de codificador de voz de fase;

30 un banco (203, 213) de filtros de análisis para filtrar la señal de audio de banda baja decodificada para generar una señal de audio de banda baja filtrada;

35 un regenerador (203, 213) de alta frecuencia para reconstruir una porción de banda alta de la señal de audio usando la señal de audio de banda baja filtrada y los metadatos de reconstrucción de alta frecuencia, donde la reconstrucción incluye una traslación espectral si el parámetro de modo de parcheo es el primer valor y la reconstrucción incluye la transposición armónica por la expansión de frecuencia del codificador de voz de fase si el parámetro de modo de parcheo es el segundo valor; y

40 un banco (203, 213) de filtros de síntesis para combinar la señal de audio de banda baja filtrada con la porción de banda alta regenerada para formar una señal de audio de banda ancha;

45 donde el flujo de bits de audio codificado incluye además un elemento de relleno con un identificador que indica un inicio del elemento de relleno y datos de relleno después del identificador, donde los datos de relleno incluyen el contenedor de extensión retrocompatible, donde los datos de relleno incluyen una carga útil de extensión, la carga útil de extensión incluye datos de extensión de replicación de banda espectral, y la carga útil de extensión se identifica con un entero de cuatro bits sin signo transmitido primero con el bit más significativo y que tiene un valor de '1101' o '1110';

50 en donde los metadatos de reconstrucción de alta frecuencia incluyen factores de escala de envolvente.

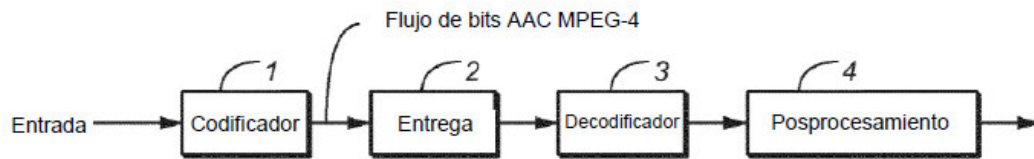


FIG. 1

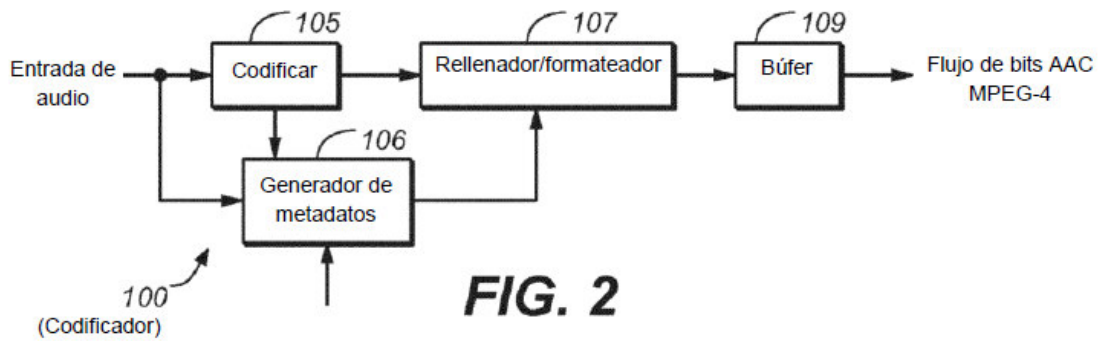


FIG. 2

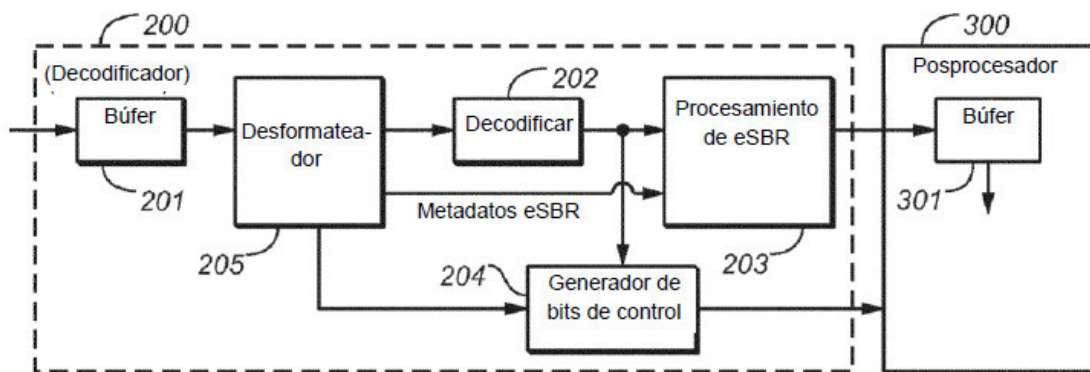


FIG. 3

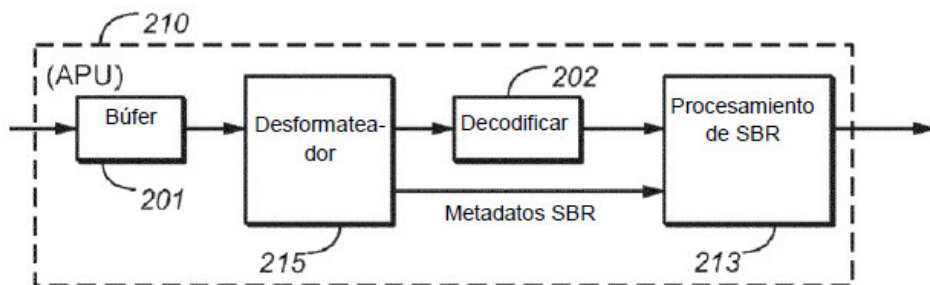


FIG. 4

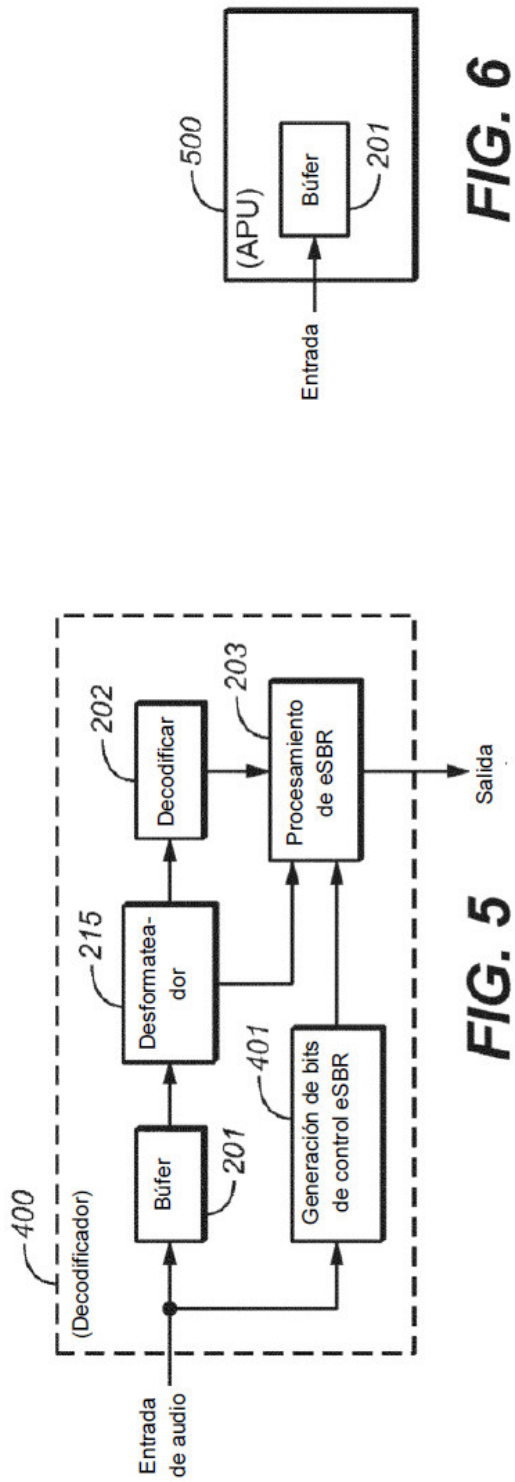


FIG. 6

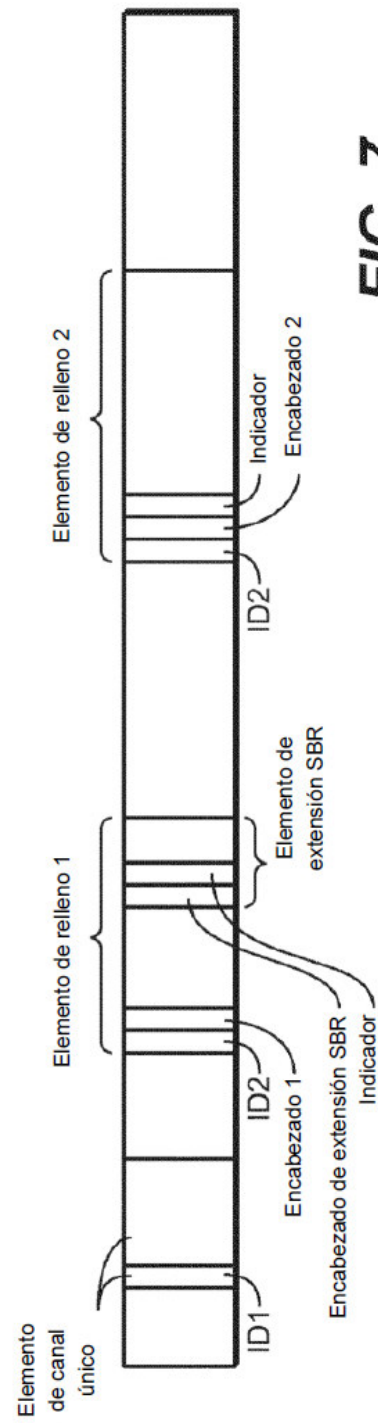


FIG. 7