

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 596 275**

51 Int. Cl.:

G06F 7/499 (2006.01)

G06F 7/544 (2006.01)

G10L 19/02 (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **13.12.2005 PCT/EP2005/013383**

87 Fecha y número de publicación internacional: **22.06.2006 WO06063797**

96 Fecha de presentación y número de la solicitud europea: **13.12.2005 E 05850069 (5)**

97 Fecha y número de publicación de la concesión europea: **03.08.2016 EP 1825356**

54 Título: **Método para crear una representación de un resultado de cálculo dependiente linealmente de un cuadrado de un valor**

30 Prioridad:

13.12.2004 DE 102004059979

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

05.01.2017

73 Titular/es:

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

72 Inventor/es:

**GAYER, MARC;
LUTZKY, MANFRED;
LOHWASSER, MARKUS;
DISCH, SASCHA;
HILPERT, JOHANNES;
GEYERSBERGER, STEFAN y
GRILL, BERNHARD**

74 Agente/Representante:

ARIZTI ACHA, Monica

ES 2 596 275 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).

Método para crear una representación de un resultado de cálculo dependiente linealmente de un cuadrado de un valor

DESCRIPCIÓN

5

La presente invención se refiere a la codificación de audio con cálculo de energías de grupo espectral de una señal de audio, tal como las necesarias por ejemplo en algunos codificadores de audio en los que, conforme a un modelo psicoacústico, se agrupan líneas de frecuencia en grupos de frecuencias con el fin de realizar la codificación ulterior por medio de las energías de señal de los grupos de frecuencias.

10

En métodos modernos de codificación de audio, basados en bancos de filtros, tales como por ejemplo MPEG Layer 3 o MPEG AAC, se usa en el codificador un modelo psicoacústico. En este modelo psicoacústico, el espectro total de la señal de audio, transformada en el dominio de frecuencia, se divide en grupos de frecuencias individuales de anchos diferentes y/o números diferentes de líneas de frecuencia por grupo de frecuencias. Para el cálculo de los umbrales de escucha psicoacústicos, para la decisión de si debe usarse estéreo-codificación central/lateral y para la estimación y/o el cálculo de los factores de escala en el módulo de cuantificación del codificador de audio, las energías de señal de las fracciones de señal de audio en los grupos de frecuencias individuales se calculan en el modelo psicoacústico. Esto se efectúa elevando al cuadrado cada línea de frecuencia individual, lo que resulta en las energías de línea, y sumando a continuación todas las energías de línea en un grupo de frecuencias para obtener la energía de banda de un grupo de frecuencias, de las que puede haber aproximadamente de 40 a 60 por canal de audio por ejemplo en MPEG AAC.

15

20

25

A continuación se considerará el caso de aplicación especial de una implementación de un método de este tipo en un procesador de coma fija.

30

En la representación de coma fija, las líneas de frecuencia se representan con un ancho de palabra de, por ejemplo, 24 bits o 32 bits. Un ancho de palabra de solamente 16 bits no es suficiente. Se usa un factor de escalado global o un factor de desplazamiento, el cual es válido para todas las líneas de frecuencia de un canal de audio o también para todas las líneas de frecuencia de todos los canales de audio procesadas en un codificador, y que determina cuántas posiciones debe ser desplazado cada valor de línea de frecuencia hacia la izquierda o la derecha, de modo que el escalado original pueda ser obtenido de nuevo, lo que se denomina representación de coma flotante por bloques. Es decir que todas las líneas de frecuencia de al menos un canal de audio se regulan igualmente y/o están en el mismo nivel de escalado. En consideración de la dinámica, generalmente alta, de las amplitudes individuales de las líneas de frecuencia, algunas de las líneas de frecuencia están representadas por tanto con una precisión relativamente alta, tal como, por ejemplo, 22 bits válidos con un ancho de palabra de 24 bits o 30 bits válidos con un ancho de palabra de 32 bits, mientras que otras están representadas con solamente unos cuantos bits válidos y, por lo tanto, de manera bastante imprecisa.

35

40

En la implementación del método de codificación de audio basado en banco de filtros descrito antes en un procesador de coma fija surgirán problemas como los descritos a continuación.

45

Primeramente, los problemas se refieren a la representación de las propias energías de señal. Las energías de señal, es decir los cuadrados sumados de las líneas de frecuencia, tienen debido a la elevación al cuadrado, una dinámica el doble de grande que las líneas de frecuencia, si ha de mantenerse la precisión total.

50

Existen posibilidades verdaderamente diferentes para representar las energías de señal. Una posibilidad es la representación de las energías de señal por un tipo de datos que tiene un ancho de palabra el doble de grande que en el tipo de datos usado para la representación de las líneas de frecuencia, es decir, por ejemplo, un tipo de datos con un ancho de 48 bits o de 64 bits. Imagínese, por ejemplo, una línea de frecuencia con 22 bits válidos, representada por un tipo de datos de 24 bits. Debido a la suma junto con las otras líneas de frecuencia, la energía de señal presentaría entonces al menos 44 bits válidos y se representaría en un tipo de datos de 48 bits. Sin embargo, este procedimiento no es factible al menos para energías de 64 bits, es decir, en casos en los que las líneas de frecuencia están representadas en un tipo de datos de 32 bits, ya que los procesadores de coma fija más usados o bien no soportan en absoluto tipos de datos de 64 bits o bien los accesos a memoria y los cálculos con un tipo de datos de 64 bits son extremadamente lentos comparados, por ejemplo, con los accesos y cálculos de 32 bits. Además, el consumo de memoria es significativamente mayor en el caso de datos de 64 bits.

55

60

Otra posibilidad para representar las energías de señal es la representación por un tipo de datos de coma flotante, por medio de la mantisa y exponente. De nuevo, supongamos el caso ejemplar antes mencionado en el que una línea de frecuencia se representa con 22 bits válidos en un tipo de datos de 24 bits. La energía de señal del grupo correspondiente se representaría entonces en un tipo de datos de coma flotante, normalizado o propietario, con mantisa de 16 bits incluyendo el bit de signo y un exponente de 8 bits. A este respecto, no es importante si se trata de un tipo de datos de coma flotante normalizado, como por ejemplo IEEE-P754, o de un tipo de datos de coma

flotante propietario con anchos de mantisa y exponente escogidos libremente. En un procesador de coma fija sin unidad de cálculo de coma flotante, los cálculos con los tipos de datos de coma flotante siempre tendrán que ser emulados mediante varias etapas de cálculo y, por lo tanto, serán extremadamente lentos, por lo que este procedimiento no es factible.

5 Otro aspecto problemático en la implementación del método de codificación de audio mencionado, basado en bancos de filtros, en un procesador de coma fija se refiere al procesamiento posterior de las energías de señal en el transcurso del método de codificación. Las energías de señal y los umbrales de escucha derivados de las mismas se usan en el proceso ulterior del codificador de audio en numerosos puntos diferentes en el algoritmo, por ejemplo, con el fin de calcular relaciones o cocientes, por ejemplo entre la energía de señal y el umbral de escucha. La división requerida no es fácil de realizar en un procesador de coma fija.

15 Una posibilidad para realizar una división en un procesador de coma fija es el uso de comandos de división de un solo bit, las cuales están implementados en algunos procesadores de coma fija y suministran un bit adicional de precisión en el cociente por operación. Para una división con una precisión de, por ejemplo 48 bits, serían necesarios por tanto 48 comandos de división individuales para una sola división de dos energías de señal o una energía de señal y un umbral de escucha. Esto no es factible y es muy ineficiente, debido al elevado consumo de tiempo que implica para los cálculos.

20 Otra posibilidad para implementar una división en un procesador de coma fija es el uso de tablas, dado el caso en conexión con etapas de iteración subsiguientes para aumentar la precisión de los resultados de la división. Este procedimiento, sin embargo, a menudo no es factible ya que para la precisión requerida del resultado de división, o bien debe ser usarse una tabla muy grande o bien las etapas de iteración subsiguientes a su vez tendrán una alta demanda de tiempo de cálculo.

25 Ambos métodos mencionados pueden ser usados en un procesador de coma fija en relación con los tipos de datos de coma fija o tipos de datos de coma flotante, emulados por software, lo que, sin embargo, en ninguno de los casos proporciona una aplicación suficientemente eficaz con respecto al tiempo de cálculo y el consumo de memoria con al mismo tiempo precisión del resultado.

30 Los problemas antes mencionados no ocurrirían con el uso de un GPP (*general purpose processor*, procesador de uso general). Para muchas aplicaciones, sin embargo, el uso de procesadores que tienen un mayor rendimiento que los procesadores de coma fija sin unidad de cálculo de coma flotante está automáticamente fuera de cuestión para la implementación de codificadores de audio debido a la presión del alto precio y el número alto de piezas. Ejemplos de estas aplicaciones son los teléfonos móviles y los PDA.

35 El documento US 6.756.618 B1 responde al problema del cálculo de la SMR, es decir el cálculo de la relación entre la energía de señal y el umbral de escucha, y concretamente ante el uso de chips de DSP de coma fija. De acuerdo con el procedimiento ahí propuesto, en primer lugar se realiza la división en ventanas usual y luego la transformación de Fourier para la descomposición de una señal de audio en sus constituyentes espectrales, seguido de lo cual la energía de cada señal de entrada y/o señal de línea de frecuencia, es decir, la energía de línea, se calcula a partir de las fracciones reales e imaginarias del valor de línea de frecuencia respectivo. Sin entrar a comentar más en relación al cálculo de las energías de señal de los grupos de líneas de frecuencia, el método puede también continuar partiendo de las energías de señal de estos grupos. En el documento se explica ahora que se trata de evitar el problema de que los datos de entrada, es decir las energías, presenten en su mayoría un intervalo dinámico demasiado grande, ya que la mayoría de los chips de DSP de coma fija presentarían un ancho de datos de solamente 16 a 24 bits, mientras que el estándar de MPEG requeriría un ancho de datos de 34 bits, es decir un intervalo dinámico de 101 dB. Por lo tanto, las energías primero tendrán que ser escaladas. En particular, se propone apartarse del procediendo anterior y usar dos valores de escala diferentes. Más precisamente, de acuerdo con este documento, la energía se compara con un valor umbral y en función de ello se escala en forma ascendente o descendente, con el fin de poder representar suficientemente el resultado del logaritmo con 16 bits pasando a un dominio logarítmico, y con el fin de poder calcular la relación SMR en el dominio logarítmico con 16 bits. Dependiendo de si se realiza un escalado ascendente o descendente, se usa una tabla diferente para los valores umbral. Para la logaritimización se usa un logaritmo en base 10 multiplicado por 10, de modo que se obtiene la unidad dB. Si el resultado de la logaritimización de las energías de línea escaladas en forma ascendente es cero, la relación SMR se calculará logaritimizando la energía de línea escalada en forma ascendente menos el logaritmo de la energía de valor umbral multiplicado por 10. De lo contrario, los resultados de la energía de línea escalada en forma ascendente y la energía de línea escalada en forma descendente se combinan entre sí.

60 El procedimiento propuesto en el documento US 6.754.618 B1 evita algunos de los problemas antes mencionados con respecto al procesamiento posterior de las energías de señal, al proponer calcular la relación SMR en el dominio logarítmico. Esto elimina concretamente el complicado cálculo de división. Sin embargo, el procedimiento es desventajoso por que el cálculo del logaritmo es aún relativamente complejo, ya que el intervalo de valores para una representación de coma fija de 16 bits, tal como la adecuada para procesadores de coma fija de DSP de 16 bits,

solo se establece después del cálculo logarítmico, mientras que la logaritmicación como tal se realiza aún con las energías presentes en dinámicas altas, por lo que se requieren aún incluso dos logaritmicaciones por valor de energía.

- 5 Por lo tanto sería deseable simplificar también el paso al dominio logarítmico, sin que ocurra una pérdida de dinámica.

10 El documento US 5.608.663 se refiere a la ejecución rápida de multiplicaciones paralelas de números de coma flotante, por medio de la conversión a un formato de coma fija logarítmico, la adición en el dominio logarítmico y la reconversión subsiguiente.

El documento US 5.197.024 trata en general de un cálculo exponencial / logarítmico y un dispositivo correspondiente.

- 15 El documento US 6.732.071 se refiere a una solución eficaz para un control de tasa de transmisión en la codificación de audio y para la determinación de un valor del parámetro de cuantificación usa una iteración de bucle con una condición de terminación según la cual se compara el valor del parámetro de cuantificación con una expresión que se deriva de un logaritmo de base dos de una expresión que depende de un valor de línea de la frecuencia máxima.

- 20 El documento US 6.351.730 describe el uso de un logaritmo en base 2 para un cálculo de ganancia en el marco de la codificación de audio. Los valores de ganancia son usados para la asignación de bits en un códec de audio codificado por MDCT.

- 25 El documento US 5.764.698 describe el uso de un logaritmo natural para la representación de energías de señal de audio. No se efectúa una descripción más detallada del paso al intervalo logarítmico.

El objetivo de la presente invención consiste en crear un codificador de audio y un método de codificación de audio de modo sea posible el cálculo anteriormente explicado de manera más sencilla o con menos esfuerzo de hardware.

- 30 Este objetivo se soluciona mediante el objeto de las reivindicaciones independientes.

35 El hallazgo de la presente invención es que, en el paso al dominio logarítmico, no es necesario tomar en consideración todo el ancho de bits del resultado dependiente linealmente del cuadrado del valor. Más bien, es posible escalar el resultado de un valor con x bits, de modo que una representación con menos de x bits del resultado sea suficiente para obtener la representación logarítmica basándose en el mismo. El efecto del factor de escalado en la representación logarítmica resultante puede ser cancelado sumando o restando un valor de corrección, obtenido por la función logarítmica aplicada al factor de escalado, a o de la representación logarítmica escalada, sin pérdida de dinámica.

- 40 Por lo tanto, es una ventaja de la presente invención que, de la manera de acuerdo con la invención, una pluralidad de resultados pueden ser transferidos a una representación logarítmica de una manera tal que, subsiguientemente, el nivel de escalado será el mismo para todos, con mantenimiento sustancial de la dinámica.

45 El procesamiento de la representación de x bits del valor consiste en el cálculo de una suma de cuadrados de valores, con el fin de obtener el resultado del cálculo dependiente linealmente del valor. Sin embargo, en lugar de realizar el factor de escalado efectivo solamente después del cálculo de la suma de cuadrados, las representaciones de x bits son escaladas ya previamente por un factor de escalado común. Este factor de escalado común se determina a partir del número de valores y o por medio del número más grande de posiciones válidas entre las representaciones de x bits de los valores. Este procedimiento hace posible permanecer constantemente, partiendo de la situación inicial con representaciones de x bits de los valores, en el formato de representación de x bits. Esto es válido igualmente para los cuadrados de los valores individuales y a la suma de los cuadrados individuales. Aquí, el factor de escalado común se determina de modo que la suma de cuadrados no experimente desbordamiento como resultado de esta suma. La función de logaritmo se realiza en un fragmento de la representación de x bits no logaritimizada de la suma de cuadrados, Se mantiene la ventaja de que el procedimiento puede ser usado para
50 varios grupos de valores, de modo que, por una parte, los valores de los grupos individuales son escalados por un factor de escalado común diferente y , por otra, el nivel de escalado es posteriormente el mismo para la
55 representación logarítmica de todas las sumas de cuadrados.

- 60 De acuerdo con un ejemplo de realización preferido de la presente invención, se usa un logaritmo en base dos como función de logaritmo, junto con un factor que es menor o igual a $1/y$. De esta manera, es posible el mapeo entre la representación del y bits del resultado del cálculo y la representación escalada logaritimizada del resultado de cálculo de una manera que utiliza óptimamente los intervalos de valores de ambas representaciones.

Para la codificación de audio, los valores de línea espectral de un grupo pueden someterse al cálculo de suma de

cuadrados de la manera antes mencionada, grupo por grupo, concretamente mediante escalado de los valores de línea espectral antes del cálculo de suma de cuadrados, estando presentes los valores de línea de frecuencia en un formato de datos de coma fija de x bits no logaritmizado, suma de los cuadrados individuales resultantes, los cuales, igualmente, deben estar presentes en el formato de datos de coma fija de x bits únicamente, aplicación de la función de logaritmo, que presenta la logaritmización con el logaritmo en base dos y el escalado por el factor menor o igual a 1/x, a la suma de cuadrados, y suma o resta de un valor de corrección, obtenido por la función del logaritmo a partir del factor de escalado común, a o de la representación de coma fija de x bits del resultado de logaritmización. De esta manera, las energías de señal de todos los grupos espectrales se mantienen al mismo nivel de escalado, sin pérdida de la dinámica. Además, cada una de las operaciones de cálculo puede ser realizada por medio de una unidad de cálculo configurada para procesar números de coma fija con u x bits solamente. Resulta ventajoso, por lo tanto, que sea posible implementar un codificador de audio en un DSP de coma fija de 16 o 24 bits, sin la necesidad de una programación compleja con el esfuerzo aumentado implicado de tiempo de cálculo.

A continuación se explican más detalladamente ejemplos de realización preferidos de la presente invención haciendo referencia a los dibujos adjuntos. Muestran:

la Figura 1 una representación esquemática que ilustra la estructura de un formato de datos de coma fija de 32 bits;

las Figuras 2a y 2b representaciones esquemáticas de una elevación al cuadrado de un valor de coma fija de 32 bits, para ilustrar el efecto de valores demasiado pequeños en el resultado de coma fija de 64 bits;

la Figura 3 una gráfica de una función de logaritmo, de acuerdo con un ejemplo de realización de la presente invención;

la Figura 4 un diagrama de bloques esquemático de un dispositivo de cálculo de suma de cuadrados;

la Figura 5 un diagrama de bloques de una posible implementación del dispositivo de la Figura 4;

la Figura 6 un diagrama de flujo para ilustrar la función del dispositivo según la Figura 5, y

la Figura 7 un diagrama de bloques esquemático para ilustrar un codificador de audio de acuerdo con un ejemplo de realización de la presente invención.

Antes de explicar ejemplos de realización de la presente invención en mayor detalle se proporciona a continuación, haciendo referencia a las Figuras 1 – 3, una introducción a estos ejemplos de realización, que es más bien una especie de revisión y que permite identificar las ventajas de los siguientes ejemplos de realización.

La Figura 1 muestra la estructura de un posible formato de datos de coma fija de 32 bits, el cual sirve como base a modo de ejemplo de los ejemplos de realización explicados a continuación. El formato de datos de 32 bits representado puede fácilmente ser aplicado igualmente a otros números de bits. Como se puede ver, un valor almacenado en el formato de datos de coma fija de 32 bits presenta 32 bits. Por lo tanto, un registro de 32 bits es suficiente para almacenar este valor en el formato de datos de coma fija de 32 bits. Tal registro se indica en la Figura 1 con 10. Las 32 posiciones de bit están indicadas con cuadrados individuales, numerados $X_0...X_{31}$, desde el bit menos significativo (LSB) al bit más significativo (MSB). Los significados de las posiciones de bit individuales, de acuerdo con el formato de datos de coma fija de 32 bits, están indicados debajo de las posiciones de bit individuales. Como se puede ver, el bit más significativo indica el signo del valor, es decir + o -. Los bits restantes, X_{30} a X_0 , expresan el valor absoluto del valor. De acuerdo con el ejemplo de realización de la Figura 1, estos bits representan una fracción real, es decir, el formato de datos 10 es un formato de datos de coma fija fraccionario, en el que, por convención, el punto / coma se dispone en la ubicación más a la izquierda, es decir en el lado izquierdo antes del bit X_{30} . El valor en el registro 10, por lo tanto, puede ser expresado como:

$$X = (-1)^{X_{31}} \sum_{i=0}^{30} X_i \cdot 2^{i-31}.$$

Como se puede ver fácilmente, el intervalo de valores representable del formato de datos de coma fija fraccionario se extiende aproximadamente desde ~1 no incluido hasta 1 no incluido.

Como se describió en la introducción de la descripción de la presente solicitud, se obtienen energías de señal, por ejemplo, por medio de la elevación al cuadrado de las líneas de frecuencia y la suma de todas las líneas de

frecuencia al cuadrado, es decir, las energías de línea, en un grupo de frecuencia. El número de líneas contenidas en un grupo de frecuencias varía por ejemplo de 4 a 96 en MPEG Layer-3 y MPEF AAC.

Como también se describió en la introducción de la descripción de la presente solicitud, es posible representar los valores de línea de frecuencia en un formato de datos de coma fija, usando un factor de escalado global, que es válido para todas las líneas de frecuencia de un canal de audio o incluso para todas las líneas de frecuencia de todos los canales de audio procesadas en un codificador y que determina cuántas posiciones ha de desplazarse cada valor de línea de frecuencia hacia la izquierda o la derecha en el lado del descodificador, de modo que se obtenga de nuevo el escalado original. Con el fin de someter tales valores de datos de coma fija de 32 bits, en un procesador de coma fija con ancho de palabras de datos de 32 bits, a una suma de cuadrados para el cálculo de las energías de señal de audio, los siguientes requisitos básicos deben ser cumplidos para una implementación eficaz del cálculo de las energías de señal de audio a partir de las líneas de frecuencia:

- las líneas de frecuencia deben representarse con un ancho de palabra de 32 bits. 16 bits no son suficientes;
- las energías de señal de audio deben también representarse en un formato de datos de coma fija y no en un formato de datos de coma flotante;
- las energías de señal de audio deben representarse en un tipo de datos con un ancho de palabra no mayor de 32 bits. De ello se deriva también un ancho de palabra de 32 bits para las energías de señal de audio.

Los ejemplos de realización especiales de la presente invención descritos a continuación cumplen con todos estos requisitos básicos, aunque se debe observar que una implementación menos eficaz puede ser obtenida si no se cumplen todos estos requisitos básicos. Como se explicará más adelante, por ejemplo, no es absolutamente necesario que las energías de la señal de audio sean limitadas en su ancho de palabra a 32 bits. Solamente el procesamiento posterior de las energías de señal deberá limitarse a una representación de coma fija de 32 bits.

De los requisitos básicos anteriores surge el siguiente problema en el cálculo de las energías de señal de audio. La elevación al cuadrado de los valores de línea de frecuencia de 32 bits proporciona en primer lugar energías de línea con un ancho de palabra de 64 bits, pero los 32 bits - en el caso de los siguientes ejemplos de realización, los 32 bits inferiores - se descartan en el procesamiento ulterior y la aritmética fraccional continúa con solamente los 32 bits superiores.

Con el fin de ilustrar esto, se hace referencia a las Figuras 2a y 2b. Estas muestran dos registros de entrada 12a y 12b de un multiplicador usado como un medio de elevación al cuadrado, escribiéndose los mismos valores en los registros de multiplicando de entrada 12a y 12b. Ambos registros, 12a y 12b, son registros de 32 bits. Además, las Figuras 2a y 2b muestran un registro 14 de salida de 64 bits, que consiste de una parte superior 14a y una parte inferior 14b. El multiplicador, el cual no se representa con mayor detalle en las Figuras 2a y 2b, está configurado para leer los contenidos de registro de los registros 12a y 12b en el formato de datos de coma fija de 32 bits y para multiplicar 16 los dos valores, de modo que se obtiene un resultado de 64 bits, el cual se emite en el registro de salida 14 en un formato de datos de coma fija de 64 bits. Así, la parte superior 14a del registro de salida 14 abarca las primeras 31 posiciones binarias detrás de la coma, mientras que la parte inferior 14b abarca las posiciones binarias restantes.

Las Figuras 2a y 2b muestran situaciones en las que se han escrito valores de línea de frecuencia diferentes en los registros de entrada 12a y 12b, de modo que se obtienen resultados diferentes en el registro de salida 14.

La Figura 2a muestra específicamente una situación en la cual el valor de línea de frecuencia es alto, concretamente 0x12345678. En tanto que los valores de línea de frecuencia son controlados en un nivel alto, es decir, ocupan un número grande de bits válidos en una palabra de datos de 32 bits, los 32 bits superiores del resultado con un ancho realmente de 64 bits en el registro de salida 14 son suficientemente exactos para expresar el resultado y/o el cuadrado. El error que se obtiene cuando solamente se usa la parte superior 14a en lugar del resultado completo asciende a solamente 0x000000003be9b080 en el caso de la Figura 2a.

La Figura 2b muestra el caso de un valor de línea de frecuencia pequeño en los registros de entrada 12a y 12b, respectivamente, concretamente 0x00004333321. Tales valores de línea de frecuencia también ocurren, como se mencionó antes, porque todas las líneas de frecuencia de un canal de audio tienen un factor de escalado global y, por lo tanto, típicamente un gran número de las líneas de frecuencia ocupará un número pequeño de bits válidos en una palabra de datos de 32 bits solamente, es decir, presentan una amplitud pequeña. Como se puede ver a partir de la Figura 2b, en el caso de valores de línea de frecuencia pequeños, los 32 bits superiores del registro de salida 14 no expresan el resultado de elevación al cuadrado con suficiente precisión. La suma de varias de tales energías de línea para formar un valor de energía de señal, en particular, por lo tanto, suministrará un resultado incorrecto, es decir, por ejemplo cero, el cual en el transcurso posterior de la codificación de audio conducirá a una peor calidad de audio. En el ejemplo de la Figura 2b, la parte superior del registro de salida, por ejemplo, indicará un

resultado de cero, mientras que el valor del resultado real será de 0x0000000023349482.

En vista de los ejemplos de las Figuras 2a y 2b, parece ser necesario considerar primero todos los 64 bits del registro de salida 14 en procesamientos subsiguientes de las energías de señal. Sin embargo, esto requerirá numerosas operaciones complicadas de datos de 64 bits, es decir operaciones con operandos de 64 bits, e infringirá los requisitos básicos antes mencionados para una implementación eficaz.

Sin embargo, si solo se va a usar realmente la parte de 32 bits superior de cada valor de energía de señal en el registro de salida 14 una mejora puede ser lograda en primer lugar desplazando todos los valores de línea de frecuencia en un grupo de frecuencia a la izquierda el mismo número de bits antes de la elevación al cuadrado, por lo que el resultado contiene un número mayor de bits válidos en la parte de 32 bits superior del resultado de 64 bits, o bien desplazando el resultado en el registro de resultado 14.

La parte superior 14a de cada valor de energía de señal podría luego ser convertida a un dominio logarítmico aplicando una unidad de cálculo para calcular la función de logaritmo a la respectiva parte superior 14a de las energías de señal. La unidad de cálculo solamente tendría que poder logaritmizar un valor de datos de coma fija de 32 bits, con el fin de obtener un valor de datos de coma fija de 64 bits. La función de logaritmo, la cual convierte el valor x en el valor y , puede ser $y = \log_2(X)/64$, cuyo desarrollo se ilustra en la Figura 3. Esta Figura 3 muestra solamente la sección de la función de logaritmo con $\epsilon]0,1[$, el cual es el único uno relevante para los valores de energía de señal elevados al cuadrado en el formato de datos de coma fija de 32 bits. Aquí, el factor de escalado $1/64$ asegura que el valor más pequeño posible, es decir un valor de la señal de energía de 2^{-63} (1 bit está reservado al menos para el signo), también entra dentro del posible intervalo de valores de partida, que se extiende desde -1 hasta 1 no incluidos, en el caso de representaciones de coma fija, incluso cuando los valores de señal de energía que se logaritmizan de acuerdo con la función de logaritmo de la Figura 3 son valores de coma fija de 64 bits.

Sin embargo, este procedimiento no será muy satisfactorio ya que en las etapas de procesamiento posteriores de la codificación de audio dentro de un codificador de audio todas las energías de señal deben presentar obligatoriamente la misma escala y, por lo tanto, el desplazamiento hacia la izquierda o el factor de escalado, que aumentan la precisión, interferirá precisamente con este nivel de escalado igual.

Precisamente en este punto surte efecto, sin embargo, una propiedad positiva de la logaritmización. Como se comentó ya anteriormente, la representación de las energías de línea es de precisión máximo gracias al desplazamiento hacia la izquierda prescrito, por ejemplo, s posiciones de bit en el dominio de línea de frecuencia o de amplitud o gracias al escalado por efectivamente 2^{2s} en el dominio de energía de señal manteniendo dicha distancia de seguridad. Temporalmente tiene que anotarse ahora el número de desplazamientos hacia la izquierda realizados en este grupo de frecuencias, es decir, el valor s . Si el valor de energía de señal de la función de logaritmo mostrada en la Figura 3 en lo siguiente denominada LD64(), el cual se calculó así con alta precisión, es ahora transferido al intervalo logarítmico - en lo siguiente, algunas veces denominado como formato LD64 - entonces el mismo está siempre presente con la máxima precisión posible. Un desplazamiento hacia la izquierda de las líneas de frecuencia de 1 bit corresponde ahora, por ejemplo, a un desplazamiento hacia la izquierda de 2 bits en las energías de señal en el dominio no logarítmico lineal, es decir, después de la elevación al cuadrado, y esto, a su vez, corresponde a una adición de $2/64$ en el dominio logarítmico puesto que $2/64 = \log_2(2)^2/64 = LD64(2^2)$.

Con el fin de restaurar la escala original en las energías de señal en el formato LD64, solamente debe ser efectuada una resta con $2^s/64$, donde s corresponde al desplazamiento hacia la izquierda anotado previamente de los valores de línea de la frecuencia del grupo. Sin embargo, esta resta no provoca una pérdida de precisión, ya que no se pierden bits válidos ni se desplazan fuera, como sería el caso con un desplazamiento hacia la derecha en el dominio lineal. Así, las energías de señal están presentes en el formato LD64 que, tal como se explicará más adelante, es ventajoso para los cálculos posteriores de cualquier manera en el codificador de audio, con una precisión alta y el mismo escalado para todos los valores de energía de señal.

Después de la explicación anterior de los principios y ventajas de los ejemplos de realización de la presente invención, a continuación se explicarán en mayor detalle por medio de las Figuras 4 a 6 ejemplos de dispositivos que pueden usarse en un codificador de audio de acuerdo con el ejemplo de realización de la figura 7.

La Figura 4 en primer lugar muestra una representación esquemática de un dispositivo adecuado para calcular la suma de cuadrados de un grupo de N valores de coma fija de x bits, y a este respecto para mantener, por una parte, el rango dinámico y, por otra, el esfuerzo de hardware dentro de límites razonables. El dispositivo se basa a este respecto en las consideraciones previas.

El dispositivo de la Figura 4, el cual se designa generalmente con 20, incluye registros $22_1, 22_2, \dots, 22_N$, de x bits, para almacenar los valores de coma fija de N bits, un medio de suma de cuadrados 24 para determinar la suma de cuadrados de los valores en los registros $22_1, \dots, 22_N$, en el dominio no logarítmico, escalado por un factor de escalado efectivo ajustable, un primer registro de resultado 26, para almacenar de manera intermedia el resultado del medio

de suma de cuadrados 24, un medio de logaritmización 28 para transferir el resultado en el registro de resultado 26 al formato LD64, un segundo registro de resultado 30 para almacenar de manera intermedia el resultado del medio de logaritmización 28, un medio de reescalado 32 para reescalar o revertir el escalado del resultado en el registro de resultado 30, un registro de salida 34 para almacenar de manera intermedia el resultado final, conforme sale del medio de reescalado 32, y un medio de control 36 que se encarga de que el escalado efectuado por el medio de suma de cuadrados 24 se revierta por el reescalado en el medio de reescalado 32 y que determina además el factor de escalado común.

Una vez descritos los componentes individuales del dispositivo 20, a continuación se describirá su interacción en el cálculo de suma de cuadrados, indicándose la interacción respectiva mediante flechas.

Como ya se ha mencionado, los valores que se van a someter a la suma de cuadrados están presentes inicialmente en los registros $22_1 - 22_N$, en el formato de coma fija de x bits. Como también se explicó previamente, estos valores, que pueden ser valores de línea de frecuencia, pueden divergir claramente uno de otro con respecto al número de sus bits válidos.

El medio de suma de cuadrados 24 ahora recibe estos valores en los registros 22_1 a 22_N , con el fin de someter los mismos a un cálculo de suma de cuadrados, de tal forma que finalmente la suma de cuadrados de los valores en los registros 22_1 a 22_N estará presente en una representación de coma fija en el registro 26, escalada por un factor de escalado efectivo, ajustable, al menos indirectamente, por el medio de control 36. Aquí, no es absolutamente necesario que el registro de resultado 26, en el cual el medio de suma de cuadrados 24 escribe el resultado de esta suma de cuadrados, tenga $2x$ bits. Más bien, será suficiente como se indica mediante la representación parcial en línea de rayas del registro 26, que el medio de suma de cuadrados 24 suministre una representación de coma fija de y bits del resultado, siendo preferiblemente $y = x$.

Se describirá en primer lugar cómo el medio de suma de cuadrados 24 puede considerar la información del factor de escalado del medio de control 36, con el fin de entregar el resultado escalado al registro 26 y de qué manera el medio de control 36 puede determinar la información del factor de escalado o el factor de escalado efectivo por el cual se escala el resultado en el registro 26, de manera que ningún bit válido se pierda debido a desbordamiento.

Conforme a la descripción precedente, una posibilidad consiste ahora en que el elemento de control 36 analice los contenidos de los registros $22_1 - 22_N$, con el fin de determinar el valor de escalado efectivo de antemano, de manera que no pueda ocurrir un desbordamiento por la suma subsiguiente de los cuadrados de los valores en los registros $22_1 - 22_N$. Esto ocurriría si ningún bit de desbordamiento estuviera presente en una unidad del cálculo de adición del medio de suma de cuadrados 24, indicada en la Figura 4 con 38, o a más tardar cuando el valor del cuadrado o de energía "desbordado" se escribe en una celda de memoria que no ofrece definitivamente ningún bit de desbordamiento, como es el caso por ejemplo con el registro 26. Por lo tanto, el medio de control 36 ajusta la información del factor de escalado en función del número N y el número máximo de bits libres en los registros $22_1 - 22_N$, o el máximo de x menos el número de posiciones válidas de los valores individuales en los registros $22_1 - 22_N$, de manera que no se produzca ningún desbordamiento durante la entrega al registro 26 o que el resultado no abandone el intervalo de valores de -1 a 1 no incluidos.

En particular, el medio de control 36 puede, como se mencionó previamente, ajustar el factor de escala efectivo, por el cual se escala el resultado en el registro, por medio de un factor de escalado común, por el cual el medio de suma de cuadrados 24 escala los contenidos de registro 22_1 a 22_N , en el inicio, es decir, antes de la elevación al cuadrado 40. En particular, el factor de escalado común, como ya se ha explicado previamente, puede corresponder a una potencia de 2, de modo que el escalado previo de los valores en los registros $22_1 - 22_N$ puede lograrse mediante una operación de desplazamiento hacia la izquierda. En este caso, no será necesario, como ya se ha explicado antes, que el medio de suma de cuadrados 24 sume 38 los resultados intermedios de las elevaciones al cuadrado 40 de los contenidos de registro individuales escalados en un formato de datos de coma fija $2x$, con el fin de producir el resultado en el registro de resultado 26, sino que más bien el medio de suma de cuadrados 24 requiere únicamente una representación de coma fija de y bits de los resultados de elevación al cuadrado intermedios $42_1 \dots 42_N$. La suma 38 de todos los resultados de elevación al cuadrado intermedios $41_1 - 42_N$ proporciona entonces el resultado final del medio de suma de cuadrados 24 en el registro 26.

Una alternativa, no de acuerdo con la invención, para determinar y considerar el factor de escalado consistiría en que el medio de suma de cuadrados 24 someta en primer lugar los contenidos de registro $22_1 - 22_N$ individualmente en cada caso a elevación al cuadrado 40, con el fin de obtener resultados intermedios $42_1 - 42_N$ en el formato de datos de coma fija $2x$. Estos serán sometido entonces a suma 38 por el medio de suma de cuadrados 24. El resultado de la suma de cuadrados de coma fija $2x$ resultante será examinado en primer lugar por el medio de control 36 en cuanto al número de bits no usados o la diferencia entre $2x$ y el número de posiciones válidas, con el fin de permitir efectuar de manera correspondiente un desplazamiento hacia la izquierda o una operación de desplazamiento hacia la izquierda por un número correspondiente de bits en este resultado de suma de cuadrados de $2x$ bits por el medio de suma de cuadrados 24, antes de que este último introduzca el resultado en el registro 26.

La desventaja en este procedimiento comparado con el descrito primero es que el esfuerzo de cálculo interno en el medio de suma de cuadrados 24 habría aumentado, ya que los valores $42_1 - 42_N$ tendrían que ser procesados con un ancho de palabra de $2x$, en particular tendrían que ser sometidos a la suma 38. Para esto, o bien se requerirá una unidad de cálculo de adición con capacidad de desempeño aumentada o bien una unidad de cálculo de adición de formato de datos de coma fija de y bits, con bit desbordamiento asociado, deberá activarse correspondientemente más a menudo. Con esta posibilidad, el medio de suma de cuadrados 24 primero elevará al cuadrado los valores en los registros 22_1 a 22_N , individualmente y sumará 38 los valores cuadrados de coma fija de $2x$ bits, 42_1 a 42_N y solo entonces escalará el resultado de acuerdo con la información de factor de escalado del medio de control 36 y lo entregará al registro 26.

En un ejemplo de realización, implementado preferiblemente con $x = 24$ bits, el registro 26 tiene una longitud de $y = 2x$ posiciones. En otro ejemplo de realización, implementado preferiblemente con $x = 32$ bits, se realizará sin embargo una reducción de posiciones a x posiciones ya en la elevación al cuadrado en un medio 42_1 a 42_N por así decir ya durante la elevación al cuadrado o después de la elevación al cuadrado, pero antes del almacenamiento en el registro. Después se realiza la suma con un ancho de palabra de x bits. Aquí, el registro 26 así solo comprenderá $y = x$ posiciones. El medio de logaritimización puede entonces también comprender preferiblemente x posiciones.

Por medio del factor de escalado efectivo, el número de bits válidos en el registro de resultado 26 puede ajustarse casi óptimamente. Ahora, el medio de logaritimización 28 recibe la representación de coma fija de y bits del registro 26, sin tener en cuenta, dado el caso, las posiciones de bit restantes de mayor precisión y la somete a la conversión al formato LD64, comentada anteriormente, con respecto a la Figura 3, con el fin de introducir el resultado en el registro 30 en el formato de datos de coma fija de z bits, siendo preferiblemente $z = y$ y al igual que preferiblemente $y = x$.

El resultado en el registro de resultado 30 muestra una representación logaritmica escalada de la suma de cuadrados de los valores en los registros $22_1 - 22_N$. El medio de reescalado 32 ahora revierte de nuevo el escalado restando del valor escalado en el registro 30 un valor 44 de corrección que obtiene del medio de control 36. En el caso del formato LD64, el valor de corrección, como se mencionó previamente, asciende a LD64(s), siendo s el factor de escalado efectivo por el cual se escala la suma de cuadrados en el registro de resultado 26.

El medio de reescalado 32 entrega entonces el resultado en el registro 34 en el formato de datos de coma fija.

Haciendo referencia a la Figura 4 se ha descrito un dispositivo para el cálculo de suma de cuadrados a grandes rasgos en cuanto a su modo de funcionamiento sin entrar más al detalle en una posible implementación del hardware. La Figura 5 muestra una posible implementación del dispositivo de la Figura 4 en algo más de detalle. El dispositivo de la Figura 5, el cual se designa generalmente con 60, incluye una memoria 62, un medio de control 64, un medio de desplazamiento 66, un medio de elevación al cuadrado 68, un sumador 70, un logaritimizador 73, un divisor 74 y un restador 76. Todos los módulos 62 - 76 se conectan en comunicación mutua por medio, por ejemplo, de un bus o también de una interfaz de programa 78. Los medios 66 - 76 pueden estar implementados en hardware, pero pueden también en parte ser códigos de programa respectivos ejecutados por el medio de control 64, con el fin de realizar las tareas respectivas. Los módulos 66 - 76 solo tienen que poder procesar cada uno operandos de formatos de datos de coma fija de x bits. De entre los módulos 66 - 76, los módulos 66 - 70 forman un medio de suma de cuadrados 80, mientras que el logaritimizador 72 y el divisor 74 forman juntos un medio de logaritimización 82.

El medio de control 64 está, por ejemplo, controlado por programa y provoca el cálculo de suma de cuadrados de valores que están depositados en el formato de datos de coma fija de x bits en la memoria 62 en las ubicaciones de memoria $84_1 \dots 84_N$ de x bits. El modo de funcionamiento preciso del medio de control 64 se comenta con respecto a la Figura 6. Aquí, se supone específicamente que los valores de coma fija de x bits en las ubicaciones de memoria $84_1 \dots 84_N$ son valores de línea de frecuencia de un grupo de frecuencias.

Primero, el medio de control 100 examina los valores de línea de frecuencia del grupo en las ubicaciones de memoria $84_1 - 84_N$ y ajusta un factor de escalado común. El ajuste en la etapa 100 se realiza, como se explicó anteriormente, basándose en el número N y el número mínimo de posiciones de bit sin usar en las ubicaciones de memoria 84_1 a 94_N . Específicamente ajusta al factor de escalado común a una potencia de 2, tal como 2^s . Alternativamente, el medio de control 64 ajusta un valor s de desplazamiento, el cual corresponde al factor de escalado común 2^s . El medio de control introduce el valor s o 2^s en un registro 101, interno o externo, para su almacenamiento intermedio.

En una etapa subsiguiente 102, el medio de control 64 hace que el medio de desplazamiento 66 desplace los contenidos de las ubicaciones de memoria 84_1 a 84_N con los valores de línea de frecuencia del grupo hacia la izquierda s posiciones de bit, de acuerdo con el factor de escala común o el valor de desplazamiento.

En una etapa 104, el medio de control 64 luego hace que el elevador al cuadrado 68 eleve al cuadrado cada valor en

las ubicaciones de memoria 84_1 a 84_N y escriba la mitad superior del resultado de $2x$ bits de nuevo en la ubicación de memoria 84_1 a 84_N correspondiente. El elevador al cuadrado 58 es, por ejemplo, un multiplicador, escribiendo el medio de control 104 el valor respectivo que se va a elevar al cuadrado, procedente de una de las ubicaciones de memoria. 84_1 a 84_N , en ambos registros de multiplicando de x bits del multiplicador. El elevador al cuadrado puede comprender internamente, por ejemplo, un registro de resultado de $2x$ bits, aunque el medio de control 64 se encarga de que solo la mitad superior, es decir el valor de elevación al cuadrado de coma fija de x bits, se escriba de nuevo en la ubicación de memoria 84_1 a 84_N correspondiente. Alternativamente, el elevador al cuadrado calcula solamente una representación de x bits, de antemano, y, por lo tanto, tiene solo un registro de salida de x bits.

Después de eso, en una etapa 106, el medio de control 64 calcula una suma de todos los contenidos de las posiciones de memoria 84_1 a 84_N , por medio del sumador 70. Puede comenzar con la suma de los primeros dos valores y escribe el resultado en un registro de suma de x bits 86, en el formato de datos de coma fija de x bits. A continuación, el medio de control 64 puede usar el sumador 70 con el fin de sumar el valor con la posición de memoria 86 a los valores subsiguientes, uno después de otro, y sobrescribe en cada caso el valor previo en la ubicación de memoria 86 con la suma resultante. Alternativamente, el medio de control 64 puede encargarse de que la suma sea escrita en una de las posiciones de memoria 84_1 a 84_N , y se acumule en la misma.

En una etapa 108, el medio de control 64 ordena entonces al logaritmizador 72 que logaritme la suma en la ubicación de memoria 86, y después de eso, ordena al divisor 74 que divida el resultado por 64. El resultado intermedio del logaritmizador 72 se escribe, por ejemplo, de nuevo en la ubicación de memoria 86, al igual que, por ejemplo, el resultado del divisor 74. Si el término divisor es una potencia de 2, el divisor se realizará como un simple medio de desplazamiento.

Después de la etapa 108, una representación escalada logaritmizada escalada de la suma de cuadrados está, por lo tanto, presente en la ubicación de memoria 86. En una etapa 110, el medio de control, por lo tanto, ordena al restador 76 que reste del valor logaritmizado y dividido en la ubicación de la memoria 86 un valor que es dependiente del factor de escalado s almacenado en la memoria intermedia 101, es decir, el valor $2^s/64$.

A continuación se describirá un ejemplo de realización de un codificador de audio, en el cual está implementado un dispositivo de acuerdo con los ejemplos de realización de las Fig. 4 a 6.

El codificador de audio de la Figura 7, el cual se designa generalmente con 150, comprende un medio de transformación 152, un medio de cálculo de energía de grupo 154 y un generador de corriente de código 156, que están conectados en serie ente una entrada 158 y una salida 160 del codificador 150. El generador de corriente de código 156 usa un modelo psicoacústico 162, con el fin de eliminar información irrelevante, por ejemplo, de una señal de audio 164 presente en la entrada 158 y que va ser codificada, cuya eliminación solo afecta levemente o nada a la calidad de audio de la corriente de datos codificada emitida en la salida 160. Aunque no se muestra en la Figura 7, el generador de corriente de código 156 puede estar acoplado al medio de transformación 152 o al medio de cálculo de energía de grupo 154 por medio de una trayectoria de retroalimentación.

La señal de audio 174 presente en la entrada 158 es, por ejemplo, una serie de valores de muestreo de audio, los cuales se han muestreado con una frecuencia de muestreo predeterminada. La señal de audio 164 puede, por ejemplo, estar presente en un formato PCM. En 164, la señal de audio se representa con respecto al tiempo t , representando el eje vertical la amplitud A en unidades arbitrarias.

Luego, el medio de transformación 152 transforma la señal de audio 164 de un dominio de tiempo a un dominio espectral, descomponiendo la señal de audio 164 en sus constituyentes espectrales. El medio de transformación 152 puede, por ejemplo, consistir específicamente en un banco de filtros de análisis, que tiene 32 filtros de paso de banda. Más precisamente, el medio de transformación 152 descompone la señal de audio 164 en componentes espectrales, sección por sección. Las secciones o tramas 166, para las cuales la descomposición espectral es efectuada, se solapan en el tiempo, por ejemplo, en un 50 %. Para cada componente espectral se crea un valor espectral o valor de línea de frecuencia para cada trama consecutiva, que se ilustra por medio de puntos en el espectrograma 168, generado por el medio de transformación 152. De esta manera, una línea de frecuencia formada por una serie de valores de línea de frecuencia se crea por cada componente espectral, estando indicadas las líneas de frecuencia en 168 con flechas horizontales, que están divididas como puede verse en valores de línea de frecuencia para las tres tramas 166 indicadas solamente. En el espectrograma 168, un número arbitrario de líneas de frecuencia se dispone a lo largo del eje espectral o el eje de frecuencia f , aunque el número real de líneas de frecuencia será mayor.

Basándose en el espectrograma 168 así creado, el generador de corriente de código 156 generará la corriente de datos codificada. Para este fin, el generador de corriente de código 156, sin embargo, no o no siempre requiere la descomposición espectral en todas las componentes espectrales. Más bien, las líneas de frecuencia están divididas, de acuerdo con aspectos psicoacústicos en grupos 170, tal como se indica mediante círculos.

El generador de corriente de código 156 requiere ahora específicamente, para la generación de corriente de código para cada grupo 170 de líneas de frecuencia, y para cada una de las tramas 166, el valor de energía de señal asociado, es decir, la suma de cuadrados de los valores de amplitud individuales, o sea la suma de cuadrados de los valores de línea de frecuencia.

5 Este cálculo se realiza por el medio de cálculo de energía de grupo 154. Este calcula la energía de señal para cada grupo 170 de líneas de frecuencia, como la suma de cuadrados de las líneas de frecuencia, estando construido el medio de cálculo de energía de grupo 154, por ejemplo, tal como se muestra en las Figuras 4 y 5 y/o funcionando tal como se describió en las figuras 4 a 6. El resultado del medio de cálculo de energía de grupo 154 son series 172 de valores de energía de señal, concretamente una serie 172 por cada grupo 170. Por lo tanto, las series 172 de energía de señal tienen un valor de energía de señal por trama 166, que se ilustra de nuevo a lo largo de las flechas 172 por medio de un punto.

15 Basándose en estas series 172 de valores de energía de señal, el generador de corriente de código 156 genera entonces la corriente de datos 160 codificada tomando como base el modelo psicoacústico 162. Aquí, una ventaja en el procesamiento posterior de los valores de energía de señal en el medio 156 no consiste solo en que los valores están presentes en un formato de datos de coma fija con el mismo número de bits, tal como lo estaban los valores de línea de frecuencia, sino también en que los valores están presentes en el dominio logarítmico, dado que este dominio permite realizar de manera más sencilla multiplicaciones, exponenciaciones y divisiones, ya que estas operaciones en el dominio logarítmico se convierten en adiciones/restas y multiplicaciones/divisiones más sencillas.

Con el fin de ilustrar esto, consideremos el caso en el que el codificador de la Figura 7 produce una corriente de datos MPEG.

25 El codificador de audio de la Figura 7 es, por ejemplo, un codificador MPEG AAC. En este caso, el generador de corriente de código 156 comprende un módulo TNS (*temporal noise shaping*), en el cual el generador de corriente de código 156 realiza un cálculo eficaz del espectro ponderado en energía. Este módulo TNS en particular realiza un cálculo con las energías de señal SE del tipo $X = 1 / \sqrt{SE}$. Esto puede ser efectuado más fácilmente con las energías de señal en el formato logarítmico LD64, tal como se usa por el medio de cálculo de energía de grupo 154. El módulo TNS calcula simplemente X para dar LD64⁻¹ (-LD64_SE >>1), donde a >> b indica una operación de desplazamiento hacia la derecha del operando a en el lado izquierdo por el número b de bits en el lado derecho, mientras que LD64⁻¹ () es una función inversa de la función de logaritmo LD64(), es decir una función inversa de la función LD64, y LD64_SE es el valor de energía de señal en el formato LD64 logarítmico, conforme se produce por el elemento de cálculo de energía de grupo 154. ">> 1", por lo tanto, designa un desplazamiento hacia la derecha por 1 bit, lo que corresponde a una división por 2. La división por 2 corresponde, dado que se realiza en el intervalo logarítmico, a la raíz cuadrada en el dominio lineal. El signo menos corresponde a la función inversa 1/x en el dominio lineal. El cálculo de la función LD64⁻¹ se puede realizar con menos complejidad que el cálculo de la raíz y la inversión del resultado, tal como se requeriría si las energías de señal se alimentaran al generador de corriente del código 156 en el dominio lineal.

40 El generador de corriente de código 156 podría también generar una corriente de datos MPEG-4-AAC codificada y comprenderían entonces un módulo PNS (*perceptual noise substitution*). El módulo PNS calcula internamente la distancia entre el umbral de escucha y la energía de señal en un grupo de frecuencias 170. El módulo PNS determina específicamente, a partir de la distancia entre el umbral de escucha y la energía de señal, si puede o no realizarse una sustitución de ruido. La mayor dinámica y precisión asociada a ello de la representación logarítmica de las energías de señal, conforme se producen por el medio de cálculo de energía de grupo 154, y los umbrales de escucha, que se proporcionan también en la representación LD64 logarítmica en el generador de corriente de código 156, contribuyen esencialmente a la determinación de PNS correcta. En particular, en una función de coherencia en el módulo de PNS, se produce una división y un cálculo de raíz en forma de

50 $Y = (SE1 / \sqrt{SE2 \cdot SE3})$, donde SE1, SE2 y SE3 son energías de señal producidas por el medio de cálculo de energía de grupo 154. La representación logarítmica de las energías de señal facilita y acelera este cálculo en gran medida. Específicamente, el módulo PNS en el generador de corriente de código 156 puede realizar el cálculo de LD64_Y, es decir de Y en el formato LD64, calculando lo siguiente:

55
$$LD64_Y = LD64_SE1 - ((LD64_SE2 + LD64_SE3) \gg 1),$$

donde LD64_SE# es la representación logarítmica del valor de energía de señal respectivo SE#, conforme es producido por el medio de cálculo de energía de grupo 154.

60 El generador de corriente de código 156 puede, por ejemplo, también comprender un módulo de estéreo central/lateral que realiza una determinación de codificación central/lateral. En este módulo CS, se calculan numerosas divisiones, concretamente del umbral de escucha con respecto a la energía de señal, para determinar si ha de usarse una codificación central/lateral (CS) o no. Gracias al uso del formato LD64, conforme es producido por

5 el medio de cálculo de energía de grupo 154, estas divisiones se convierten en simples restas. Los umbrales, a partir de los cuales ha de usarse una codificación CS, son constantes y pueden, por lo tanto, transferirse al dominio logarítmico ya en el código de fuente, por lo cual no se requiere un tiempo de cálculo ulterior para el tiempo de ejecución del programa. Esto se aplica a muchas más constantes, usadas en el código modificado en relación con los cálculos de LD64. Lo mencionado antes, en particular, también se aplica al módulo de estéreo de intensidad de un codificador de MPE Layer 33 o AAC.

10 El generador de corriente de código 156 puede además presentar un estimador del factor de escala, que realiza un cálculo de un volumen de sonido. En el estimador del factor de escala, que está dispuesto por ejemplo en un módulo de cuantificación del generador de corriente de código 156, se produce un cálculo de la raíz cuarta de los umbrales de escucha, es decir un cálculo del volumen de sonido. Mediante la representación del umbral de escucha por medio de un formato de datos logarítmicos, concretamente LD64, el cálculo de esta raíz cuarta puede realizarse muy eficientemente desplazando simplemente los umbrales medios logarítmicos hacia la derecha 2 bits. La etapa inversa, es decir una potenciación por 4, puede también ser realizada de una manera muy sencilla, mediante desplazamiento hacia la izquierda 2 bits.

15 En el estimador del factor de escala del generador de corriente de código 156, asimismo, por ejemplo, un cálculo de los factores de escala SCF, que forman parte del ámbito de cuantificación, puede ser calculado a partir de los umbrales de escucha modificados, MHS, y las energías de señal, SE. El cálculo incluye la etapa de cálculo de tipo $SCF = \log(k \cdot MHS/SE)$, donde $\log()$ es un logaritmo en base diez y k es una constante. Este cálculo puede resolverse de manera muy elegante y eficaz gracias al uso del formato LD64 y una conversión correspondiente de la fórmula original. El estimador del factor de escala realizará el cálculo específicamente de la siguiente manera: $SCF = \log(k) + \log_2^{64}(LD64_MHS - LD64_SE)$

20 Los ejemplos anteriores de módulos posibles dentro del generador de corriente de código 156 muestran que el uso del formato de datos logarítmico LD64 posibilita un aumento de la eficacia en procesadores de coma fija. El formato de datos logarítmico LD64 posibilita, para muchos casos, que un códec de audio con alta calidad de audio pueda implementarse en estas plataformas, solo si no ha de bajarse de una determinada calidad de audio.

30 Los ejemplos anteriores de módulos posibles en el generador de corriente de código 156, en particular, muestran que la representación de las energías de señal, umbrales de escucha y valores de energía adicionales en un formato de datos logarítmico es muy apropiada dado que en el procesamiento de energías de señal y umbrales de escucha, la mayoría de las etapas de cálculo se realizan por medio de una división o multiplicación. Por lo tanto, una división se convierte en una resta y una multiplicación en una suma, lo cual resulta en una representación y procesamiento eficientes de las energías de señal y los umbrales de escucha en un codificador de audio por medio del formato de datos logarítmico ilustrado. Las energías de señal en el dominio lineal, es decir en el dominio no logarítmico, se representan en el formato de datos de coma fija, fraccionario, que presenta un intervalo de valores de -1,0 hasta +0.99999...

40 En los ejemplos de realización anteriores se ha usado un logaritmo en base 2, el logaritmo binario o *dualis* LD. En un logaritmo en base 2, un valor de energía de señal de 0,25 en el dominio lineal corresponde a un valor LD de -2. Un valor de energía de 0,3 en el dominio lineal corresponde a un valor LD de -1,7369656. Como en un procesador de coma fija por regla general pueden representarse valores de número entero o también valores fraccionarios, pero habitualmente no valores mixtos, en los ejemplos de realización precedentes los valores de LD se escalaron en el contexto del formato LD64 de modo que solo se obtuvieran valores fraccionarios. Por supuesto, otra posibilidad sería escoger la escala de modo que solo se obtengan valores enteros. Se escogió una escala con un factor de 1/64 específicamente, porque este factor es aplicable a todos los anchos de palabra hasta 64 bits y, por lo tanto, también abarca los anchos de palabra convencionales de 48 bits y de 32 bits. El cálculo del logaritmo binario escalado por 1/64, es decir el formato LD64, se realiza según la fórmula $y = \log_2(x)/64 - LD64(x)$. Prescindiendo de la portabilidad extendida, este factor puede, por supuesto, ser también aumentado, concretamente hasta 1/y, siendo y el número de bits de la suma de cuadrados escalada no logaritmizada.

55 La Figura 3 muestra la gráfica de la función de logaritmo. Aquí puede observarse la característica de la función de logaritmo que, para valores de entrada x menores, es decir realmente más inexactos en una representación de coma fija (más cercanos a 0,0), ofrece un número mayor de valores de salida y , es decir una resolución más fina en el dominio logarítmico, los cuales, al ser mayores en valor absoluto, es decir, más cercanos a -1,0, son, por lo tanto, más precisos en una representación de coma fija. De acuerdo con los ejemplos de realización anteriores, una pérdida de precisión en el cálculo de las energías de señal antes del paso al intervalo logarítmico, se evita por el factor de escalado común o el valor de desplazamiento común, antes de elevar al cuadrado.

60 Por lo tanto, el ejemplo de realización de la Figura 7 representa un codificador que usa un formato de datos logarítmico escalado o también no escalado, en el modelo psicoacústico y/o en el módulo de cuantificación para la representación y el procesamiento de las energías de señal y los umbrales de escucha. Sin embargo, se entenderá que la descripción anterior del logaritmo en base dos y de la escala por 1/64 es solamente un ejemplo. En cualquier

caso, la representación logarítmica sirve para lograr una ejecución más rápida de numerosos cálculos, tal como en el módulo de determinación central/lateral y el estimador del factor de escala del módulo de cuantificación, debido a que, como se ha explicado, las divisiones presentes originalmente se convierten, gracias al uso del formato logarítmico, en simples restas, las cuales pueden ser realizadas mucho más rápido en un procesador de coma fija.

5 Lo mismo se aplica a los cálculos de raíces, tal como el cálculo de una raíz cuadrada, una raíz cuarta, etc., las cuales se convierten en simples desplazamientos hacia la derecha. El esfuerzo cálculo que ha de realizarse para el paso al dominio logarítmico es compensado por mucho por la ganancia de velocidad en las etapas posteriores del algoritmo. Por lo tanto, las ventajas de los ejemplo de realización anteriores específicamente son la mayor precisión de los valores representados logarítmicamente, al igual que la mayor velocidad de procesamiento con el uso de la
10 representación logarítmica.

Según los ejemplos de realización anteriores se ha usado una función de logaritmización en la cual se realiza un escalado por $1/64$. Este factor es razonable si ha de ejecutarse una misma codificación en diferentes plataformas con diferentes formatos de datos de coma fija. Es, sin embargo, también posible que el factor en la función
15 logarítmica solo pueda ser mayor que x , es decir, mayor que el número de bits de los valores que se suministran a la suma de cuadrados.

Se entenderá especialmente que, dependiendo de las circunstancias, el esquema de acuerdo con la invención puede también ser llevado a cabo en software. La implementación puede efectuarse en un medio de almacenamiento digital, en particular en un disquete o un CD con señales de control legibles electrónicamente, que interaccionan con un sistema informático programable, de modo que se realice el método correspondiente. En general, la invención, por lo tanto, también consiste en un producto de programa informático con un código de programa, almacenado en un soporte legible por la máquina, para realizar el método de la invención cuando este producto de programa informático se ejecuta en un ordenador. En otras palabras, la invención, puede, por lo tanto,
20 ser llevada a cabo como un programa informático, con un código de programa para realizar el método, cuando este programa informático se ejecuta en un ordenador.
25

REIVINDICACIONES

1. Codificador de audio con
 - 5 un dispositivo implementado en un procesador de coma fija para el cálculo de energías de grupos espectrales de una señal de audio con un medio de transformación (152) para descomponer la señal de audio (164) en tramas en un dominio de tiempo y para transformar la señal de audio en cada trama del dominio de tiempo a un dominio espectral, a fin de obtener para cada trama un grupo de componentes espectrales de las que cada una presenta un valor de amplitud en una representación de x bits no logaritmizada, de modo que se producen series de valores de amplitud que están repartidas en grupos espectrales (170) diferentes;
 - 10 un medio de cálculo de energía de grupo (24; 80) para calcular una energía de grupo espectral para cada grupo espectral en la respectiva trama como suma de cuadrados de los valores de amplitud (22_1-22_N ; 84_1-84_N), presentando el medio de cálculo de energía de grupo (24; 80):
 - 15 un medio de control (36; 64) para ajustar un respectivo factor de escalado para cada grupo espectral en función del número (N) de valores de amplitud del respectivo grupo espectral y el número más pequeño de posiciones de bit no utilizadas en las representaciones de x bits de los valores de amplitud del respectivo grupo espectral,
 - 20 un medio de escalado (66) para escalar la representación de x bits no logaritmizada (84_1-84_N) de cada valor de amplitud para cada grupo espectral por el respectivo factor de escalado, a fin de obtener valores de amplitud escalados en una representación de x bits,
 - un medio de elevación al cuadrado (68) para elevar al cuadrado los valores de amplitud escalados en la representación de x bits, a fin de obtener valores de amplitud escalados y elevados al cuadrado en una representación de 2x bits,
 - 25 un medio de suma (70) para sumar los valores de amplitud escalados y elevados al cuadrado para cada grupo espectral, a fin de obtener una respectiva suma de cuadrados en una representación no logaritmizada (26),
 - un medio de logaritmo (28; 82) para aplicar una función de logaritmo a y bits de la representación no logaritmizada (26) de cada suma de cuadrados para cada grupo espectral, a fin de obtener una respectiva representación escalada y logaritmizada (30), siendo y inferior a 2x,
 - 30 un medio de reescalado (32; 76) para sumar o restar (44) un respectivo valor de corrección, que se obtiene aplicando la función de logaritmo al cuadrado del respectivo factor de escalado, a o de la representación escalada y logaritmizada (30) para cada representación logaritmizada (30), a fin de obtener una representación logaritmizada (34) de cada energía de grupo espectral, presentando las energías de grupo espectral de todos los grupos espectrales el mismo nivel de escalado; y
 - 35 un generador de corriente de código MPEG (156) para, basándose en las energías de grupo espectral de los grupos espectrales en representación logaritmizada con el mismo nivel de escalado (34), generar una corriente de datos MPEG codificada, que representa una versión codificada de la señal de audio (164).
- 40 2. Codificador de audio de acuerdo con la reivindicación 1, en el que el medio de control (64) está configurado para ajustar el respectivo factor de escalado a una k-ésima potencia de 2, y el medio de escalado (66) está configurado para desplazar k bits la representación de x bits de un valor de amplitud del grupo en un registro de valores de x bits.
- 45 3. Codificador de audio de acuerdo con la reivindicación 1 o 2, en el que el medio de elevación al cuadrado (68) presenta un registro de 2x resultados y está configurado para depositar un valor de amplitud escalado y elevado al cuadrado predeterminado de los valores de amplitud escalados y elevados al cuadrado en la representación de 2x bits en el registro de 2x resultados, y el medio de suma (70) está configurado para leer un fragmento de y bits de la representación de 2x bits en el registro de 2x resultados, mientras que no tiene en cuenta el resto de bits en el registro de 2x resultados, a fin de sumar el fragmento de y bits junto con fragmentos de y bits de los demás valores de amplitud escalados y elevados al cuadrado y escribir la suma de cuadrados en un registro de y bits.
- 50 4. Codificador de audio de acuerdo con una de las reivindicaciones anteriores, en el que el medio de logaritmización (28; 72) está configurado para tratar los y bits de la representación no logaritmizada (26) de tal manera que la representación escalada y logaritmizada (30) corresponda a una representación de y bits del logaritmo en base dos de la representación de y bits no logaritmizada (26) dividido por un número mayor o igual que y, y el medio de reescalado (32) está configurado para sumar un valor absoluto a la representación escalada y logaritmizada (30) o restarlo de la misma, que corresponde al logaritmo en base dos del cuadrado del respectivo factor de escalado dividido por el número.
- 60 5. Codificador de audio de acuerdo con la reivindicación 4, en el que el número es 64 e $y = x = 32$.
6. Codificador de audio de acuerdo con una de las reivindicaciones anteriores, en el que y es igual a x.
7. Codificador de audio de acuerdo con una de las reivindicaciones anteriores, en el que el generador de corriente

de código (156) está configurado para generar la corriente de código codificada basándose además en un modelo psicoacústico (162).

8. Codificador de audio según la reivindicación 7, en el que el generador de corriente de código es un codificador de MPEG © Layer 3 o un codificador de MPEG 2/4 AAC.

5 9. Codificador de audio de acuerdo con una de las reivindicaciones anteriores, en el que el generador de corriente de código (156) presenta un módulo TNS, un módulo PNS para realizar el cálculo y un estimador del factor de escala.

10 10. Método de codificación de audio con las siguientes etapas:

calcular energías de grupo espectral de una señal de audio por medio de un procesador de coma fija descomponiendo la señal de audio (164) en tramas en un dominio de tiempo,

15 transformar la señal de audio en cada trama del dominio de tiempo a un dominio espectral, a fin de obtener para cada trama un grupo de componentes espectrales, de las que cada una presenta un valor de amplitud en una representación de x bits no logaritmizada, de modo que se producen series de valores de amplitud que están repartidos en grupos espectrales (170) diferentes,

20 calcular una energía de grupo espectral para cada grupo espectral en la respectiva trama como la suma de cuadrados de los valores de amplitud (22_1-22_N ; 84_1-84_N) por medio del ajuste de un respectivo factor de escalado para cada grupo espectral en función del número (N) de valores de amplitud del respectivo grupo espectral y el número más pequeño de posiciones de bit no utilizadas en las representaciones de x bits de los valores de amplitud del respectivo grupo espectral,

25 escalar la representación de x bits no logaritmizada (84_1-84_N) de cada valor de amplitud para cada grupo espectral con el respectivo factor de escalado, a fin de obtener valores de amplitud escalados en una representación de x bits,

elevar al cuadrado los valores de amplitud escalados en la representación de x bits, a fin de obtener valores de amplitud escalados y elevados al cuadrado en una representación de 2x bits,

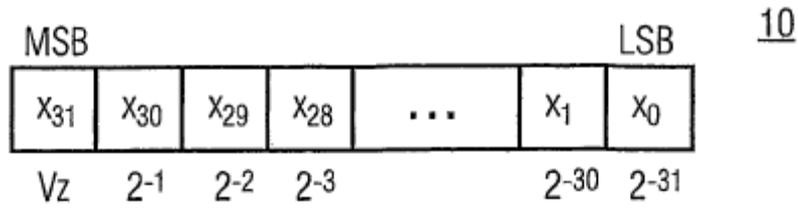
30 sumar los valores de amplitud escalados y elevados al cuadrado para cada grupo espectral, a fin de obtener una respectiva suma de cuadrados en una representación no logaritmizada (26),

35 aplicar una función de logaritmo a y bits de la representación no logaritmizada (26) de cada suma de cuadrados para cada grupo espectral, a fin de obtener una respectiva representación escalada y logaritmizada (30), siendo y inferior a 2x, y

sumar o restar (44) un respectivo valor de corrección, que se obtiene al aplicar la función de logaritmo al cuadrado del respectivo factor de escalado, a o de la representación escalada y logaritmizada (30) para cada representación escalada y logaritmizada, a fin de obtener una representación logaritmizada (34) de cada energía de grupo espectral, presentando las energías de grupo espectral de todos los grupos espectrales el mismo nivel de escalado; y

40 generar una corriente de datos MPEG codificada, que representa una versión codificada de la señal de audio (164), basándose en las energías de grupo espectral de los grupos espectrales en representación logaritmizada con el mismo nivel de escalado (34).

11. Programa informático con un código de programa para realizar el método según la reivindicación 10, cuando el programa informático se ejecuta en un ordenador.



10

FIG. 1

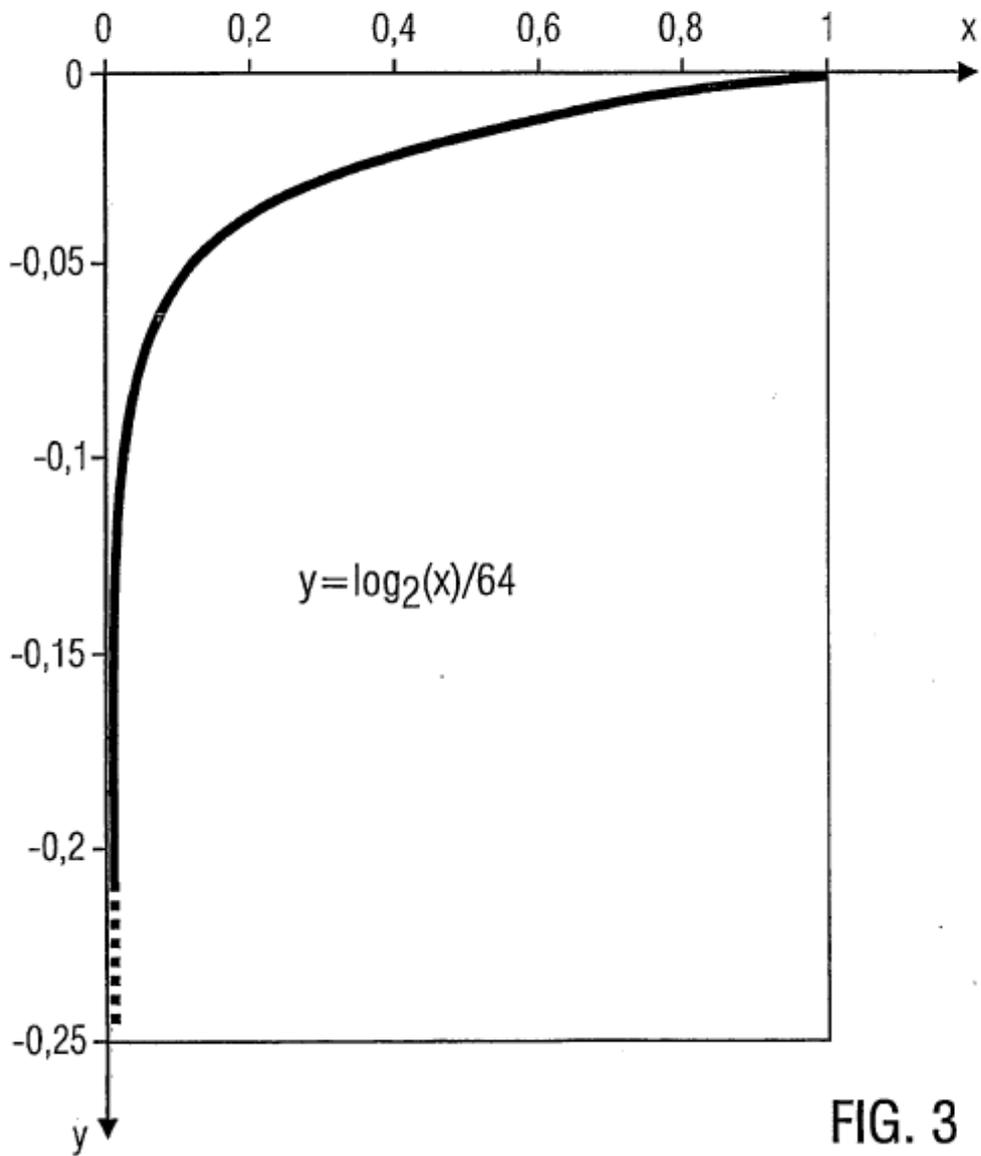


FIG. 3

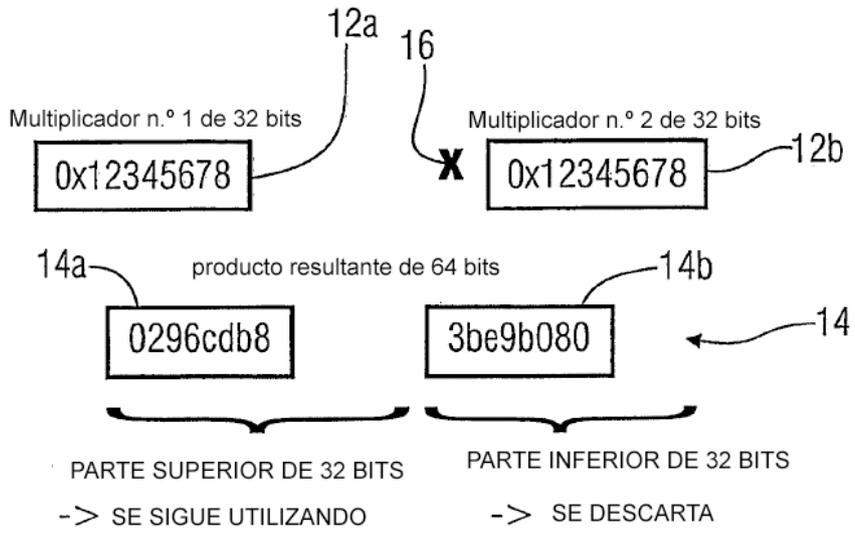


FIG. 2a

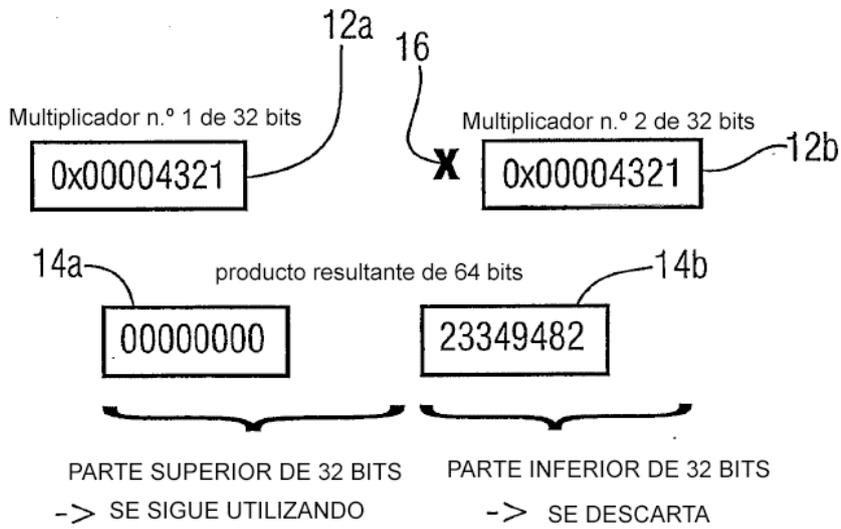


FIG. 2b

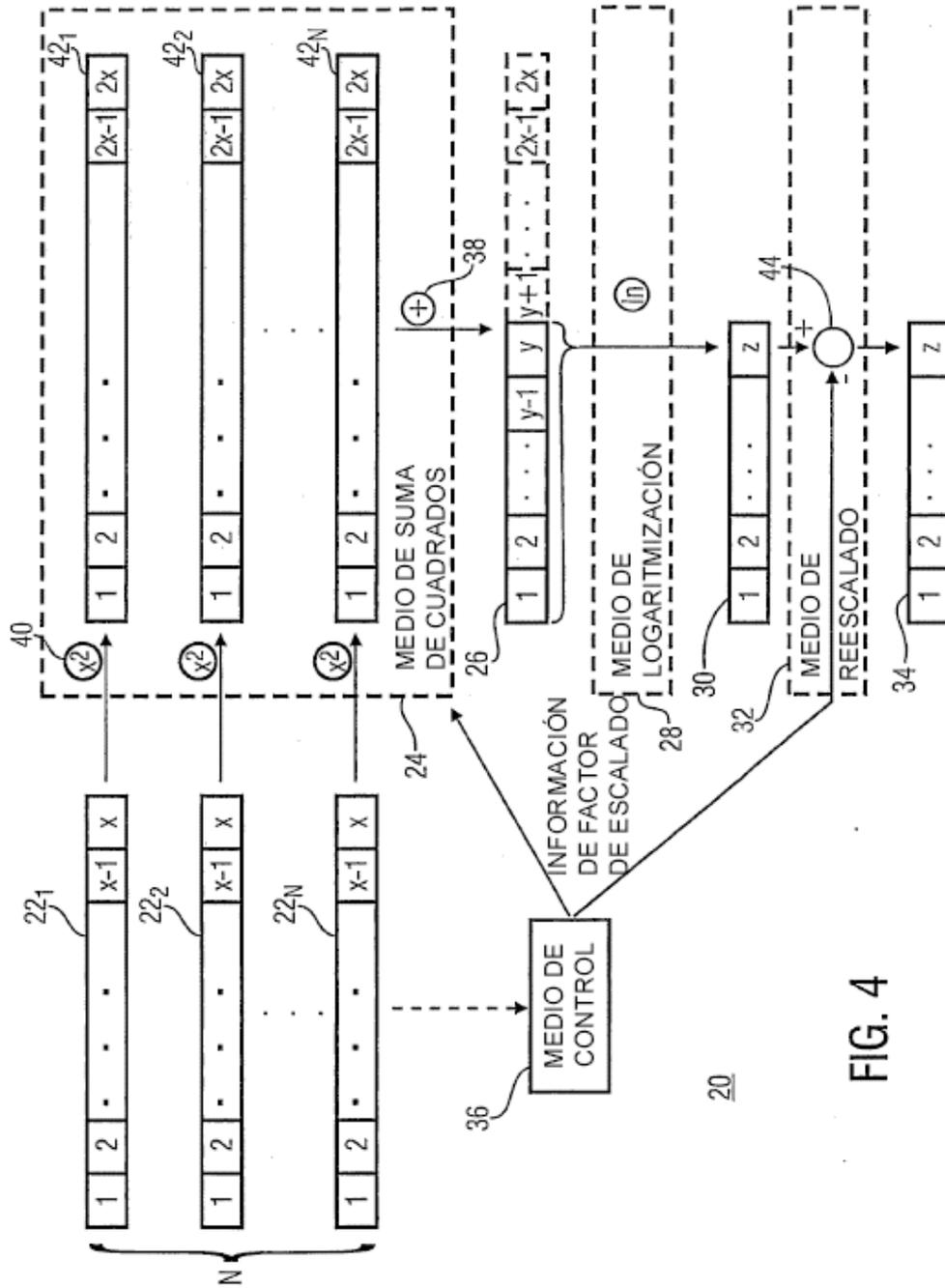


FIG. 4

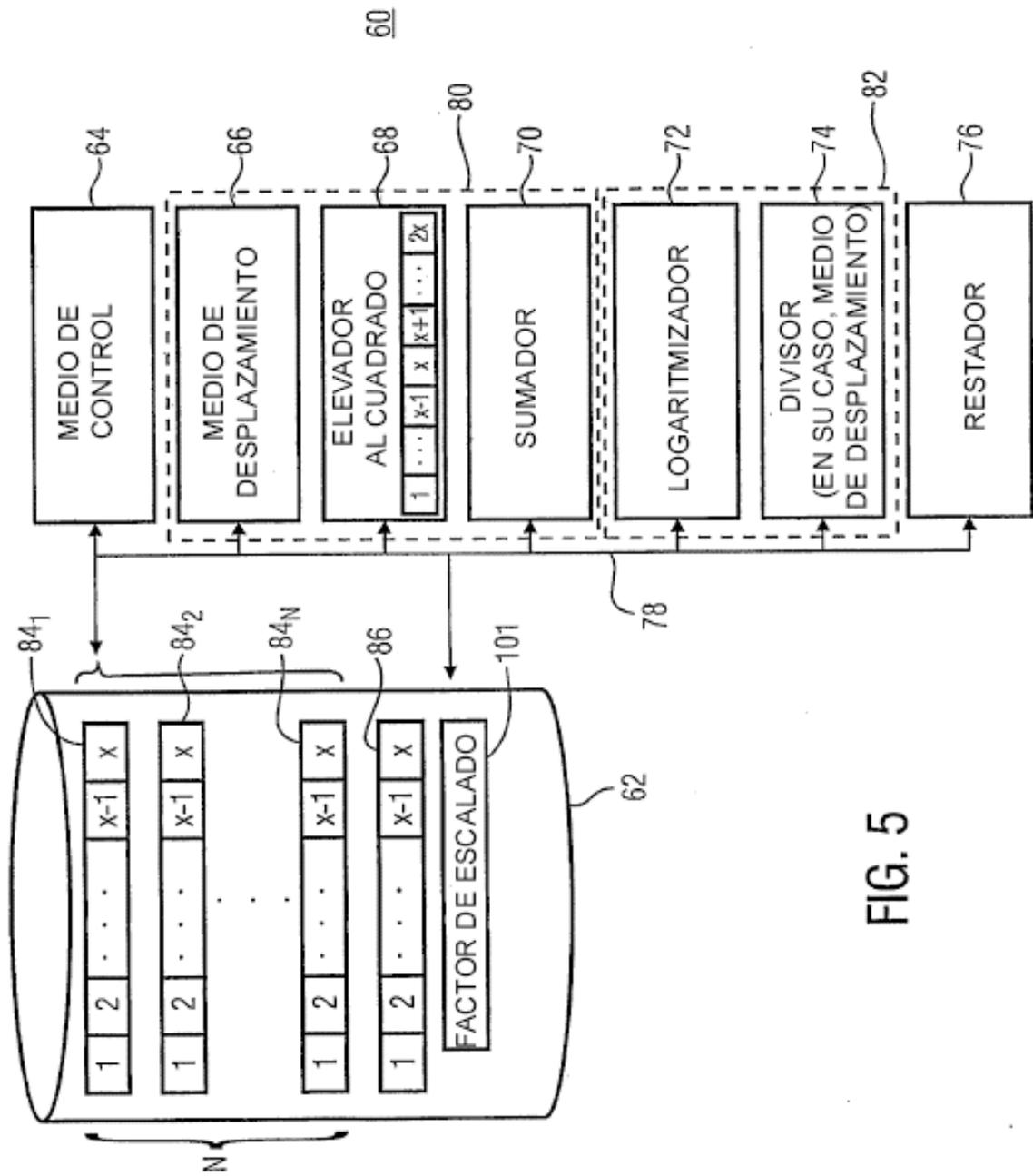


FIG. 5

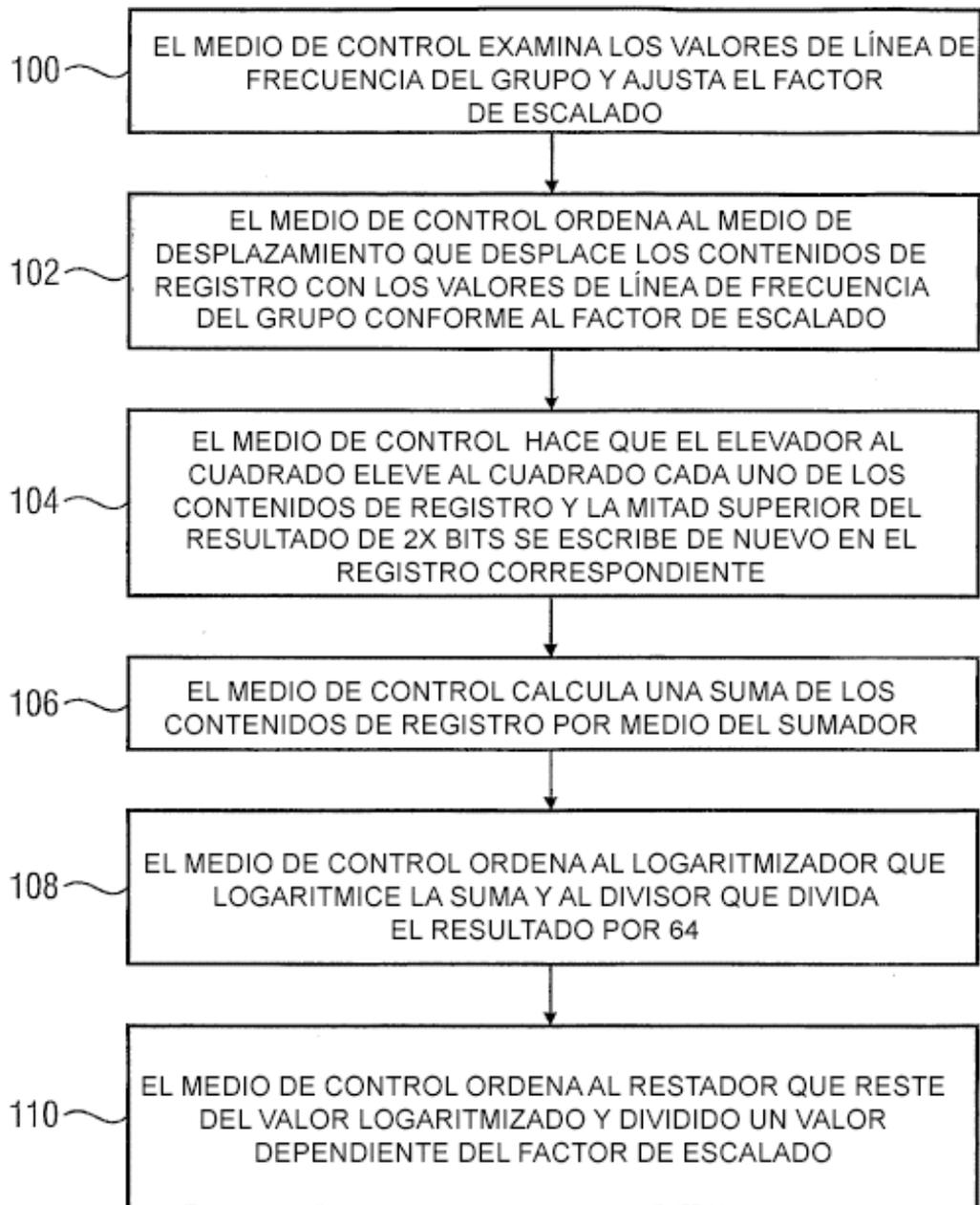


FIG. 6

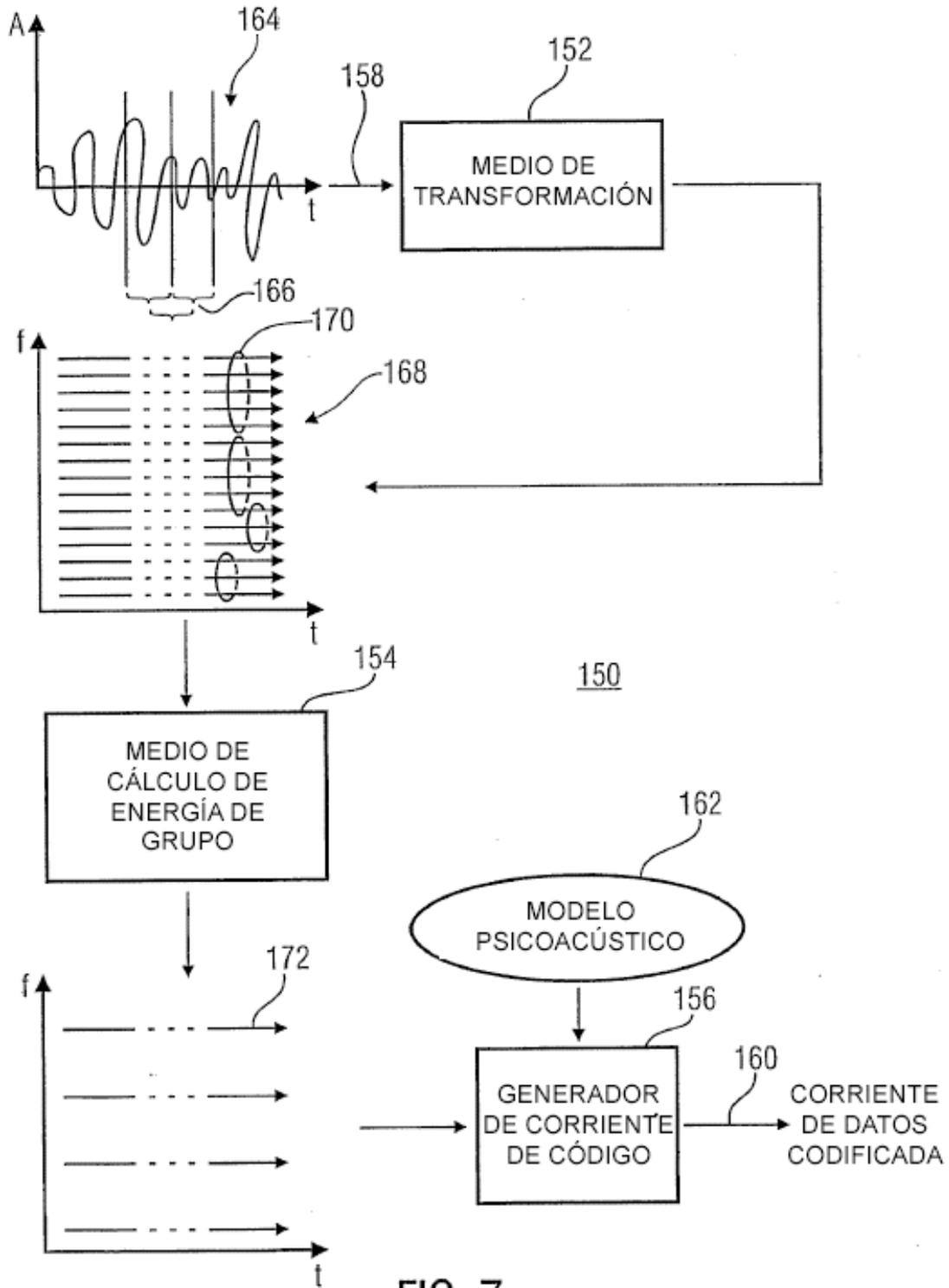


FIG. 7