RECOMENDACIÓN UIT-R BS.1196-1*,**

Codificación de audio para la radiodifusión de la televisión terrenal digital

(Cuestiones UIT-R 78/10, UIT-R 19/6, UIT-R 37/6 y UIT-R 31/6)

(1995-2001)

La Asamblea de Radiocomunicaciones de la UIT,

considerando

a) que se establecerán sistemas de radiodifusión de televisión terrenal digital en las bandas de ondas métricas/decimétricas;

b) que para tales sistemas resulta esencial la utilización de una señal de sonido multicanal de alta calidad, que emplee una reducción eficaz de la velocidad binaria;

c) que los sistemas de transmisión de sonido con velocidad de bits reducida, deben protegerse contra los errores de bit residuales debidos a la decodificación de canal y al proceso de demultiplexación;

d) que la Recomendación UIT-R BS.775 se ocupa de un sistema de sonido multicanal con y sin acompañamiento de imagen;

e) que la Recomendación UIT-R BS.1116 se ocupa de métodos para la evaluación subjetiva de sistemas de audio con pequeñas degradaciones, incluyendo los sistemas de sonido multicanal;

f) que el empleo común de métodos de codificación de fuentes de audio entre diferentes servicios puede mejorar la flexibilidad de los sistemas y reducir los costes del receptor;

g) que las Recomendaciones UIT-R BS.774 y UIT-R BS.1114 tratan de la radiodifusión sonora digital para receptores instalados en vehículos, portátiles y fijos empleando transmisores terrenales en las bandas de ondas métricas y decimétricas;

h) que la ISO/CEI, en colaboración con el UIT-R, ha realizado estudios sobre sistemas genéricos de reducción de la velocidad binaria de las señales de audio, elaborándose como consecuencia de esos trabajos las Normas IS 11172-3 (MPEG-1 audio) e IS 13818-3 (MPEG-2 audio), que son objeto de la Recomendación UIT-R BS.1115;

j) que varios servicios de radiodifusión sonora por satélite y numerosos sistemas de distribución secundaria (televisión por cable) utilizan o han especificado, como parte de sus servicios digitales planificados, las Normas MPEG-1 audio, MPEG-2 o AC-3 audio multicanal (véanse los Anexos);

k) que en una familia de equipos se utiliza extensamente la Norma IS 11172-3 (MPEG-1 audio) y la Norma IS 13818-3 (MPEG-2 audio);

l) que un importante sistema de película de sonido digital utiliza la Norma AC-3;

m) que el sistema de televisión digital europeo (DVB) utilizará la Norma MPEG-2 de audio;

---

* La Comisión de Estudio 6 de Radiocomunicaciones efectuó modificaciones de redacción en esta Recomendación en 2003 de conformidad con la Resolución UIT-R 44.

** Esta Recomendación debe señalarse a la atención de la Organización Internacional de Normalización (ISO) y de la Comisión Electrotécnica Internacional (CEI).

n) que el sistema de televisión avanzada digital de América del Norte (ATV) utilizará la Norma AC-3;

o) que es muy conveniente el interfuncionamiento con otros medios tales como el disco óptico que utilicen audio con la Norma MPEG-2 o con la Norma AC-3,

recomienda

1 que los sistemas de radiodifusión de televisión terrenal digital utilicen para la codificación de audio la Norma Internacional especificada en el Anexo 1 o la Norma de Estados Unidos de América especificada en el Anexo 2.

NOTA 1 – Cabe señalar que las velocidades binarias de audio necesarias para lograr los niveles de calidad especificados para el sonido multicanal con estos sistemas no han sido completamente evaluadas y documentadas en el UIT-R.

NOTA 2 – Cabe señalar además que se están desarrollando actualmente mejoras compatibles (por ejemplo, explotación de las características sintácticas disponibles y mejora del modelo psicoacústico) que pueden incrementar de manera significativa a lo largo del tiempo la calidad de funcionamiento del sistema.

NOTA 3 – Considerando que la evaluación de la calidad de funcionamiento actual y futura de estos sistemas de codificación es fundamentalmente responsabilidad de la Comisión de Estudio 6 de Radiocomunicaciones, se alienta la continuación de los trabajos de esta Comisión de Estudio en este campo, con carácter de urgencia, a fin de ofrecer un texto adicional contrastado a la Recomendación y de detallar las características de calidad de funcionamiento de las opciones de codificación disponibles.

NOTA 4 – El sistema de codificación de audio especificado en el Anexo 2 es un códec no compatible hacia atrás (NBC) que es incompatible con los dos canales de codificación, de acuerdo con la Recomendación UIT-R BS.1115.

NOTA 5 – Se insta a la Comisión de Estudio 6 de Radiocomunicaciones a continuar sus esfuerzos para lograr la unificación de las especificaciones de codificación.

Anexo 1

Capa II de audio MPEG (ISO/CEI IS 13818-3): Norma de codificación genérica de sonido bicanal y sonido multicanal para radiodifusión de vídeo digital, radiodifusión de audio digital y multimedios por computadora

1 Introducción

De 1988 a 1992 la Organización Internacional de Normalización (ISO) elaboró y preparó una norma sobre «Tecnología de la Información – Codificación de imágenes en movimiento y audio asociado para medios de almacenamiento digital hasta 1,5 Mbit/s». El Subgrupo de Audio del Grupo MPEG fue responsable de elaborar una norma para la codificación genérica de señales de audio MIC con velocidades de muestreo de 32, 44,1 y 48 kHz a velocidades binarias comprendidas en una gama de 32 a 192 kbit/s por canal de audio monofónico y 64 a 384 kbit/s por canal de audio estereofónico. El resultado de este trabajo es la parte audio de la norma MPEG-1, que consiste en tres capas con complejidad diferente para aplicaciones diferentes y se denomina Norma ISO/CEI 11172-3. Después de muchas pruebas realizadas en 1992 y en 1993, el UIT-R recomienda utilizar la capa II de MPEG-1 para contribución, distribución y emisión que son aplicaciones típicas de radiodifusión. Con respecto a las aplicaciones de telecomunicación, el UIT-T ha formulado la

Recomendación UIT-T J.52, que es la norma para la transmisión de datos de audio MPEG por la red digital de servicios integrados (RDSI).

El primer objetivo de audio MPEG-2 fue la extensión de la codificación de audio de alta calidad de 2 a 5 canales de una manera compatible hacia atrás, basada en Recomendaciones UIT-R, de la Asociación de Ingenieros de Televisión e Imágenes en Movimiento (SMPTE) y de la Unión Europea de Radio y Televisión (UER). Este objetivo se logró en noviembre de 1994 con la aprobación de la Norma ISO/CEI 13818-3, conocida como audio MPEG-2. Esta norma proporciona codificación de alta calidad de 5,1 canales de audio, es decir, cuatro canales con anchura de banda completa más un canal de mejora de baja frecuencia de banda estrecha, y con compatibilidad hacia atrás con MPEG-1, la clave para garantizar que los decodificadores de dos canales existentes serán capaces de decodificar la información estereofónica compatible de señales multicanal. Para la reproducción del audio del sonido ambiente, se utilizan las posiciones de altavoz izquierda, centro, derecha, izquierda y derecha que recogen el sonido ambiente de acuerdo con la norma 3/2. Las aplicaciones previstas son sistemas de televisión digital, tales como dTTb, HDTVT (TVAD-T), HD-SAT, ADTT, medios de almacenamientos digital, por ejemplo, el disco vídeo digital y el sistema de radiodifusión de audio digital de la Recomendación UIT-R BS.1114, Sistema DAB (EU147).

El segundo objetivo del audio MPEG-2 era la extensión del audio MPEG-1 a velocidad de muestreo más baja para mejorar la calidad de audio a velocidades binarias inferiores a 64 kbit/s por canal, en particular para aplicaciones vocales. Esto es de particular interés para las aplicaciones de la RDSI de banda estrecha, donde por sencillos motivos operacionales se puede evitar la multiplexación de varios canales B proporcionando aún una excelente calidad de audio, incluso a velocidades binarias inferiores a 48 kbit/s. Otra aplicación importante es el sistema EU147 DAB. La capacidad de programas del canal del servicio principal se puede aumentar aplicando la opción de frecuencia de muestreo más baja a canales de noticias de alta calidad que necesitan menos bits para la misma calidad en comparación con la frecuencia de muestreo completa.

2 Principios de la técnica de codificación de audio de la capa II de MPEG

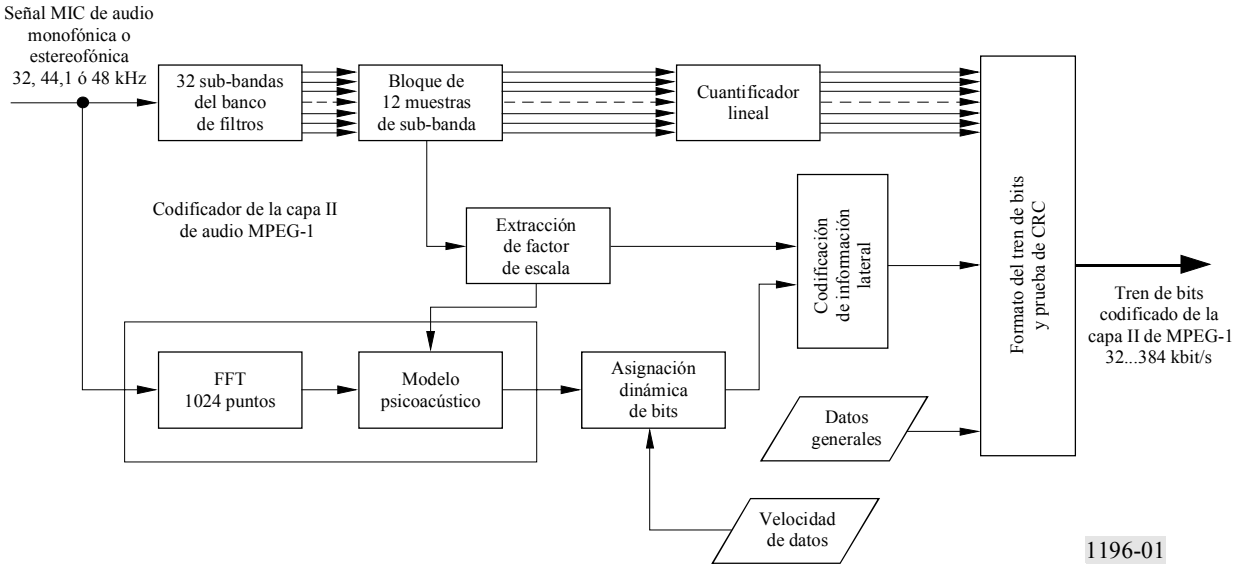
Se pueden emplear dos mecanismos para reducir la velocidad binaria de las señales de audio. Un mecanismo es determinado principalmente por la supresión de la redundancia de la señal de audio mediante correlaciones estadísticas. Además, esta nueva generación de esquemas de codificación reduce la no pertinencia de la señal de audio considerando los fenómenos psicoacústicos, como el enmascaramiento espectral y temporal. Solamente con estas dos técnicas, que utilizan las correlaciones estadísticas de los efectos de enmascaramiento del oído humano, se podrá obtener una reducción importante de la velocidad binaria hasta 200 kbit/s por señal estereofónica y velocidades más bajas.

La capa II es idéntica al conocido sistema de codificador de audio MUSICAM mientras que la capa I ha de ser interpretada con una versión simplificada del sistema MUSICAM. La estructura básica de la técnica de codificación, que es más o menos común a la capa I y la capa II, se caracteriza por el hecho de que el audio MPEG se basa en la codificación de audio porcentual. Por consiguiente, el codificador consiste en los siguientes módulos claves:

– Una de las funciones básicas del codificador es la transformación de la señal de entrada MIC con una anchura de 20 kHz del dominio del tiempo en componentes espectrales submuestreados. Para proporcionar esta funcionalidad, se utiliza para ambas capas un banco de filtros polifásicos que contienen 32 sub-bandas espaciadas igualmente.

FIGURA 1

Diagrama de bloques del codificador de la capa II (audio MPEG-1) de la Norma ISO/CEI 11172-3



– La salida de una transformada de Fourier que se aplica a la señal audio MIC de banda ancha en paralelo al proceso de filtrado, se utiliza para calcular una estimación del umbral enmascarado real, que depende del tiempo. Para esto, se emplea un modelo psicoacústico basado en reglas conocidas de la psicoacústica como un bloque de función adicional en el codificador. Este bloque estimula el enmascaramiento espectral y, en cierta medida, también el enmascaramiento temporal. La base fundamental para calcular el umbral enmascarado en el codificador viene dada por los resultados de mediciones de umbral enmascarado para señales de banda estrecha, considerando el ruido de enmascaramiento de tono y viceversa. En relación con la distancia en frecuencia y la diferencia en el nivel de presión acústica, en la literatura existente se describen relaciones de enmascarador/tono de prueba muy limitadas y artificiales y se han considerado los resultados del caso más desfavorable en relación con las pendientes superior e inferior de las curvas de enmascaramiento para formular la hipótesis de que los mismos umbrales enmascarados se pueden utilizar para situaciones de audio simples y situaciones de audio complejas.

– Las muestras de sub-bandas son cuantificadas y codificadas con el propósito de mantener el ruido, introducido por la cuantificación, por debajo del umbral enmascarado. Las capas I y II utilizan una técnica de compresión-expansión de bloques con un factor de escala que consiste en 6 bits válidos para una gama de dinámica de unos 120 dB y una longitud de bloque de 12 muestras de sub-bandas. Debido a esta clase de técnica de aplicación de factor de escala, las capas I y II pueden tratar una gama dinámica muchos más alta que el disco compacto o DAT, es decir, MIC de 16 bits convencional.

– En el caso de señales estereofónicas, se puede añadir la codificación estereofónica mixta como una característica adicional para explotar la redundancia y la no pertenencia del material de programas estereofónicos típico y se puede utilizar para aumentar la calidad de audio a bajas velocidades binarias y/o reducir la velocidad binaria para señales estereofónicas. El aumento de la complejidad del codificador es pequeño y la complejidad adicional que se requiere en el decodificador es también despreciable. Es importante mencionar que la codificación de estereofonía mixta no aumenta el retardo de codificación global.

– Tras codificar la señal de audio, se utiliza un bloque de ensamblado para entramar el tren de bits audio MPEG que consiste en tramas de audio consecutivas. La longitud de trama de la capa I corresponde a 384 muestras de audio MIC, la longitud de trama de la capa II, a 1 152 muestras de audio MIC. Cada trama de audio, mostrada en la Fig. 2, comienza con un encabezamiento, seguido por la información de asignación de bits, el factor de escala y las muestras de sub-bandas cuantificadas y codificadas. Al final de cada trama de audio está el campo de datos auxiliares, de longitud variable, que se puede especificar para ciertas aplicaciones.

2.1 Modelo psicoacústico

El modelo psicoacústico calcula el umbral enmascarado mínimo que es necesario para determinar el nivel de ruido perceptible para cada banda en el banco de filtros. La diferencia entre el nivel de señal máximo y el umbral enmascarado mínimo se utiliza en la asignación de bits o de ruido para determinar el nivel real del cuantificador en cada sub-banda para cada bloque. En la parte informativa de la Norma ISO/CEI 11172-3 se muestran dos modelos psicoacústicos. Aunque ambos se pueden aplicar a cualquier capa del algoritmo audio MPEG, en la práctica se utilizará el modelo 1 para las capas I y II y el modelo 2 para la capa III. En ambos modelos, la salida final del modelo es una relación señal/enmascaramiento para cada sub-banda de la capa II. El modelo psicoacústico es necesario solamente en el codificador, lo que permite que los decodificadores sean mucho menos complejos. Por consiguiente, es posible mejorar incluso ulteriormente el funcionamiento del codificador, relacionando la velocidad binaria y la calidad subjetiva. Para algunas aplicaciones que no necesitan una velocidad binaria muy baja, es posible utilizar un codificador muy sencillo sin ningún modelo psicoacústico.

Una resolución de alta frecuencia, es decir, pequeñas sub-bandas en la región de frecuencias más bajas, y una resolución más baja en la región de frecuencias más altas con sub-bandas grandes, debe ser la base para un cálculo adecuado de los umbrales enmascarados en el dominio de la frecuencia. Esto conduciría a una estructura arborescente del banco de filtros. La red de filtros polifásicos utilizada para el filtrado de sub-banda tiene una estructura paralela que no proporciona sub-bandas de diferentes anchuras. No obstante, una ventaja importante del banco de filtros es la adaptación óptima de los bloques de audio a los requisitos de los efectos de enmascaramiento temporal y ecos previos imperceptibles. La segunda ventaja importante consiste en que el retardo y la complejidad son pequeños. Para compensar la falta de precisión del análisis del espectro del banco de filtros, se utiliza una transformada de Fourier rápida (FFT) de 1 024 puntos para la capa II en paralelo al proceso de filtrado de la señal de audio en 32 sub-bandas. La salida de la FFT se usa para determinar los componentes tonales pertinentes, es decir, enmascarados de ruido sinusoidales y no tonales, de la señal de audio real. De acuerdo con la investigación psicoacústica, es bien sabido que la tonalidad de un componente de enmascaramiento tiene una influencia sobre el umbral enmascarado. Por ese motivo, es conveniente discriminar entre componentes tonales y no tonales. Se calcula cada umbral enmascarado para cada enmascarador por encima del umbral enmascarado absoluto según la posición en frecuencia, el nivel de sonoridad y la tonalidad. Todos los umbrales enmascarados, incluido el umbral absoluto, se añaden al denominado umbral enmascarado global. Para cada sub-banda, se determina el valor mínimo de esta curva de enmascaramiento. Por último, se calcula la diferencia entre el nivel de señal máximo, calculado por ambos, los factores de escala y el espectro de densidad de potencia de la FFT, y el umbral enmascarado mínimo para cada sub-banda y cada bloque. La longitud de bloque de la capa II es determinada por 36 muestras de sub-bandas, que corresponden a 1152 muestras MIC de audio de entrada. Esta diferencia de nivel de señal máximo y umbral enmascarado mínimo se denomina relación señal/enmascaramiento (SMR, *signal-to-mask ratio*) y es la función de entrada pertinente para la asignación de bits.

En la Fig. 1 se muestra un diagrama de bloques del codificador de la capa II. En los puntos siguientes se explican detalladamente los pasos del proceso de codificación y de decodificación, incluida la división de la señal audio MIC de entrada por un banco de filtros de análisis polifásico en 32 sub-bandas espaciadas igualmente, una asignación dinámica de bits derivada de un modelo psicoacústico, la técnica de compresión-expansión de bloques de las muestras de sub-bandas y el formato del tren de bits.

2.2 Banco de filtros

El filtro QMF prototipo es del orden 511, optimizado desde el punto de vista de la resolución espectral y del rechazo de lóbulos laterales, lo que es mejor que 96 dB. Este rechazo es necesario para una compensación suficiente de las distorsiones por solape. Este banco de filtro proporciona un compromiso razonable entre el comportamiento temporal por un lado y la exactitud espectral por el otro. Una correspondencia de tiempo/frecuencia que proporciona un alto número de sub-bandas facilita la reducción de la velocidad binaria, debido a que el oído humano percibe la información de audio en el dominio espectral con una resolución correspondiente a las bandas críticas del oído o incluso más baja. Estas bandas críticas tienen una anchura de unos 100 Hz en la región de bajas frecuencias, es decir, 500 Hz, y anchuras de 20% aproximadamente de la frecuencia central a frecuencias más altas. El requisito de tener una resolución espectral adecuada es lamentablemente contradictorio con la necesidad de mantener la respuesta de impulsos transitorios, denominada eco previo y eco posterior, dentro de ciertos límites en términos de posición temporal y amplitud comparadas con el ataque de un sonido percutiente. El conocimiento del comportamiento del enmascaramiento temporal da una indicación de la posición temporal y amplitud necesarias del eco previo generado por una correspondencia tiempo/frecuencia, de manera que este eco previo, que normalmente es mucho más crítico comparado con el eco posterior, es enmascarado por el ataque original. Asociada al banco de filtros de síntesis dual colocado en el decodificador, esta técnica de filtrado proporciona una función de transferencia global optimizada desde el punto de vista de una percepción de respuesta de impulso perfecta.

En el decodificador, el banco de filtros de síntesis dual reconstruye un bloque de 32 muestras de salida. La estructura del filtro es extremadamente eficaz para realizarla en un decodificador de poca complejidad y que no es DSP y requiere generalmente menos de 80 multiplicaciones/adiciones de enteros por muestra de salida MIC. Además, el filtro de análisis y síntesis completo da un retardo de tiempo global de sólo 10,5 ms a una velocidad de muestreo de 48 kHz.

2.3 Determinación y codificación de factores de escala

Se calcula el factor de escala para cada sub-banda para un bloque de 12 muestras de sub-banda. El máximo del valor absoluto de estas 12 muestras es determinado y cuantificado con una longitud de palabra de 6 bits, que abarca una gama dinámica global de 120 dB por sub-banda con una resolución de 2 dB por clase de factor de escala. En la capa I, se transmite un factor de escala para cada bloque y cada sub-banda que no tiene asignación de bits cero.

La capa II utiliza una codificación adicional para reducir la velocidad de transmisión para los factores de escala. Debido a que en la capa II una trama corresponde a 36 muestras de sub-banda, es decir, tres veces la longitud de una trama de la capa I, en principio hay que transmitir tres factores de escala. Con el fin de reducir la velocidad binaria para los factores de escala, se ha estudiado una estrategia de codificación que explota los efectos de enmascaramiento temporal del oído. Tres factores de escala sucesivos de cada sub-banda de una trama se consideran juntos y se clasifican en determinados patrones de factor de escala. Según el patrón, se transmiten juntos uno, dos o

tres factores de escala con una información de selección de factor de escala adicional que consiste en 2 bits por sub-banda. Si sólo hay pequeñas desviaciones de una factor de escala al siguiente, habrá que transmitir solamente el mayor. Esto ocurre relativamente a menudo para sonidos tonales estacionarios. Si hay que codificar ataques de sonido percutientes, habrá que transmitir dos o los tres factores de escala, según el borde ascendente y descendente del ataque. Esta técnica de codificación adicional permite, por término medio, un factor de dos de reducción de la velocidad binaria para los factores de escala, en comparación con la capa I.

2.4 Asignación de bits y codificación de la información de asignación de bits

Antes del ajuste a una velocidad binaria fija, se debe determinar el número de bits que están disponibles para codificar las muestras. Este número depende del número de bits requeridos para los factores de escala, la información de selección de factor de escala, la información de asignación de bits y los datos auxiliares.

El procedimiento de asignación de bits se determina minimizando la relación ruido/enmascaramiento total en cada sub-banda y en toda la trama. Este procedimiento es un proceso iterativo donde, en cada paso repetitivo se aumenta el número de niveles de cuantificación de la sub-banda que tiene la mayor ventaja, con la constricción de que el número de bits utilizado no rebase el número de bits disponibles para esa trama. La capa II utiliza 4 bits para codificar la información de asignación de bits solamente para las sub-bandas más bajas y sólo 2 bits para las sub-bandas más altas por trama de audio.

2.5 Cuantificación y codificación de muestras de sub-bandas

En primer lugar, se normaliza cada una de las 12 muestras de sub-banda de un bloque dividiendo su valor por el factor de escala. El resultado se cuantifica de acuerdo con el número de bits empleados por el bloque de asignación de bits. Sólo son posibles números impares de niveles de cuantificación, lo que permite una representación exacta de un cero digital. La capa I utiliza 14 clases de cuantificación diferentes, que contienen $2^n - 1$ pasos, con $2 \leq n \leq 15$ niveles de cuantificación diferentes. Esto es igual para todas las sub-bandas. Además, no se puede utilizar la cuantificación si no hay bits asignados a una sub-banda.

En la capa II, el número de niveles de cuantificación diferentes depende del número de sub-bandas, pero la gama de los niveles de cuantificación comprende siempre una gama de 3 a 65 535 con la posibilidad adicional de ninguna cuantificación. Las muestras de sub-banda en la región de bajas frecuencias pueden ser cuantificadas con 15 niveles de cuantificación diferentes, en la gama de frecuencias medias con 7 y en la gama de frecuencias altas con sólo 3. Las clases pueden contener 3, 5, 7, 9 15, 63 ...., 65 535 niveles de cuantificación. Como los niveles de cuantificación 3, 5, y 9 no permiten una utilización eficaz de una palabra de código, que consiste solamente en 2, 3 ó 4 bits, se agrupan juntas tres muestras de sub-bandas sucesivas en un «gránulo». Después el gránulo es codificado con una palabra de código. La ganancia de codificación que se obtiene utilizando la agrupación es de hasta 37,5%. Debido a que muchas sub-bandas, especialmente en la región de altas frecuencias, son cuantificadas típicamente con sólo 3, 5, 7 y 9 niveles de cuantificación, el factor de reducción de longitud de las palabras de código es considerable.

2.6 Estructura del tren de bits de la capa II

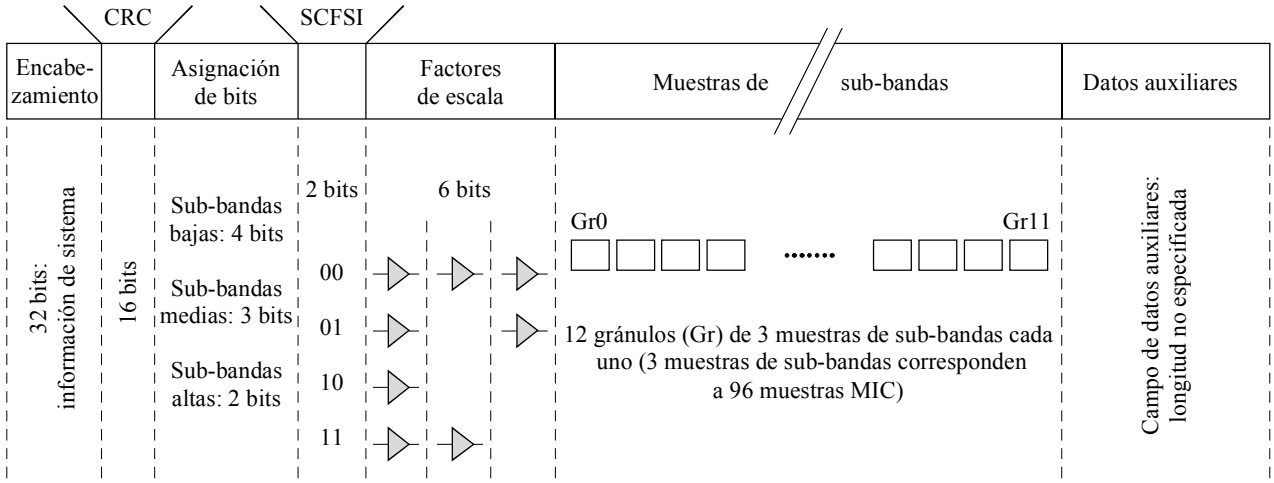
El tren de bits de la capa II se construyó de manera que se pueda utilizar un decodificador de poca complejidad y un retardo de codificación bajo, y que la señal de audio codificada contenga muchos puntos de entrada con intervalos de tiempo cortos y constantes. La representación digital codificada de un algoritmo de codificación eficaz especialmente adaptado a la aplicación de almacenamiento debe permitir múltiples puntos de entrada en el tren de datos codificado para grabar, reproducir y editar secuencias de audio cortas y definir con precisión las posiciones de edición. Para que la realización del codificador sea sencilla, la trama entre estos puntos de entrada debe contener toda la

información necesaria para decodificar el tren de bits. Debido a las diferentes aplicaciones, esta trama tiene que transportar además toda la información necesaria para permitir una amplia gama de codificación con muchos parámetros diferentes. Estas características son importantes también en el campo de la radiodifusión de audio digital, donde se necesitan decodificadores de poca complejidad por razones económicas y donde se necesitan puntos de entrada frecuentes en el tren de bits para ocultar fácilmente los bloques de muestras erróneas consecutivas, degradadas por errores en ráfaga.

El formato del tren de bits de audio codificado para la capa II se muestra en la Fig. 2. La estructura del tren de bits se caracteriza por tramas de audio autónomas cortas que corresponden a 1152 muestras MIC. Cada trama que comienza con una palabra de sincronización de 12 bits puede ser accedida y decodificada independientemente y tiene una duración de 24 ms a una frecuencia de muestreo de 48 kHz.

FIGURA 2

Estructura de trama de audio del tren de bits codificado de la capa II de la Norma ISO/CEI 11172-3



La trama se ensambla sobre la base de 1152 muestras MIC de audio.
Con una frecuencia de muestreo de 48 kHz, la duración de la trama es de 24 ms.

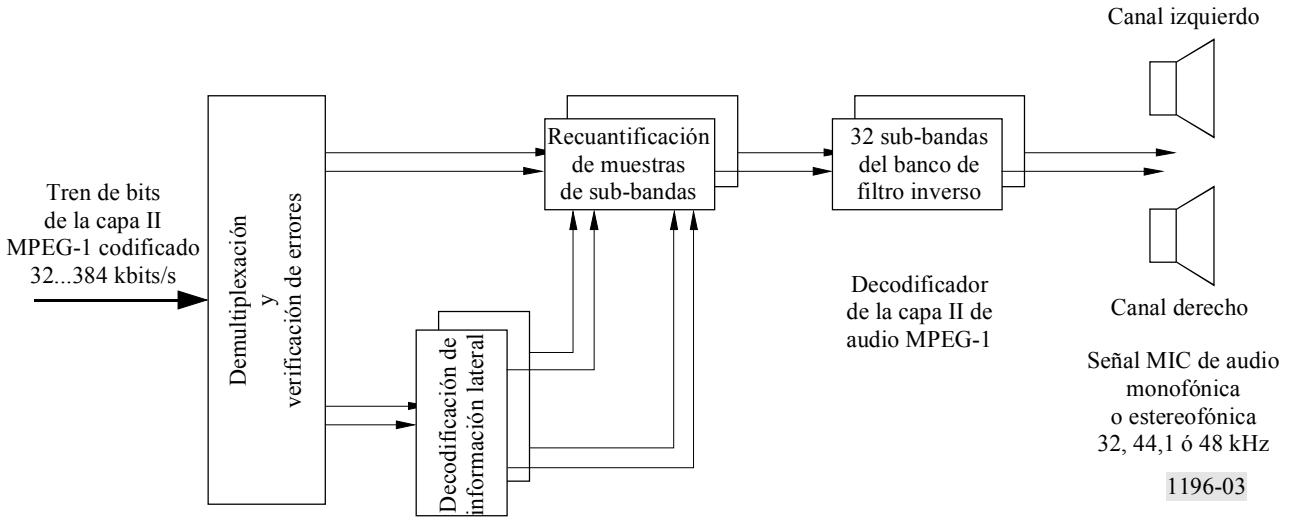
1196-02

2.7 Decodificación de la capa II

En la Fig. 3 se muestra el diagrama de bloques del decodificador. En primer lugar, la información de encabezamiento, la prueba CRC, la información lateral, es decir, la información de asignación de bits con factores de escala, y doce muestras sucesivas de cada señal de sub-banda, están separadas del tren de bits de la capa II de audio MPEG de ISO.

FIGURA 3

Diagramas de bloques del decodificador de la capa II (audio MPEG-1) de la Norma ISO/CEI 11172-3



El proceso de reconstrucción para obtener de nuevo el audio MIC se caracteriza rellenando el formato de datos de las muestras de sub-banda en relación con el factor de escala y la asignación de bits para cada sub-banda y trama. El banco de filtros de síntesis reconstruye la señal de audio de banda ancha completa con una anchura de banda de hasta 24 kHz. El proceso de decodificación necesita una potencia de cómputo mucho menor que el proceso de codificación. La relación para la capa II es aproximadamente 1/3. En vista de la baja potencia de cómputo necesaria y de la estructura directa del algoritmo, la capa II podrá realizarse fácilmente en VLSI especiales. Desde 1993, están disponibles microplaquetas de decodificador de señales estereofónicas de diversos fabricantes. Se dispone también de codificadores de señales estereofónicas de capa I y capa II que se han realizado solamente en un DSP de puntos fijos (DSP56002).

3 Audio MPEG-2: Codificación genérica de audio multicanal

Una de las características básicas de la norma de audio MPEG-2 (Norma ISO/CEI 13818-3) es la compatibilidad hacia atrás con los programas de audio monocanal, estereofónico o bicanal codificados de la Norma ISO/CEI 11172-3. Esto significa que un decodificador de audio de la Norma ISO/CEI 11172-3 o MPEG-1 puede decodificar adecuadamente la información estereofónica básica de un programa multicanal. La información estereofónica básica se mantiene en los canales izquierdo y derecho, lo que constituye un mezclado reductor apropiado de la información de audio en todos los canales.

La compatibilidad hacia atrás con estereofonía bicanal es un requisito importante para muchos proveedores de servicio, que podrán proporcionar en el futuro sonido ambiental digital de alta calidad. Con la excepción del mundo cinematográfico, actualmente no existe audio multicanal digital discreto. Sin embargo, ya están muy generalizadas las microplaquetas de decodificador de capa I y capa II de MPEG-1 que apoyan sonido monofónico y estereofónico. Debido a la compatibilidad hacia atrás de la norma de codificación de audio multicanal MPEG, este decodificador bicanal entregará siempre una señal estereofónica correcta con toda la información de audio del tren de bits de audio multicanal MPEG-2.

El audio MPEG-1 se amplió, como parte de la actividad de MPEG-2, a frecuencias de muestreo más bajas con el fin de mejorar la calidad de audio para señales monofónicas y estereofónicas convencionales a velocidades binarias de 64 kbit/s o inferiores por canal, en particular para aplicaciones de comentarios. Este objetivo se ha logrado reduciendo la velocidad de muestreo a 16, 22,05 ó 24 kHz lo que proporciona una anchura de banda de hasta 7,5, 10,5 u 11,5 kHz. La única diferencia comparada con MPEG-1 es un cambio en las tablas de velocidades binarias y asignaciones de bits del codificador y del decodificador. Se mantienen plenamente los principios de codificación y de decodificación de las capas de audio MPEG-1.

3.1 Características del sistema de codificación de audio multicanal MPEG-2

Un sistema de sonido multicanal digital genérico aplicable a radiodifusión sonora y de televisión y almacenamiento, así como a otras aplicaciones que no son de radiodifusión, debe satisfacer varios requisitos básicos y proporcionar varias características técnicas y operacionales. Debido a que durante los próximos años la representación estereofónica normal será aún predominante para la mayoría de las aplicaciones de consumidor, la compatibilidad bicanal es uno de los requisitos básicos. Otros requisitos importantes son la interoperabilidad entre medios diferentes, la compatibilidad con formatos de sonido que consisten en un número menor de canales de audio y que por tanto proporcionan una calidad de sonido ambiental reducida. Para que las aplicaciones puedan ser lo más universales posibles, son importantes también otros aspectos, tales como los servicios multilingües, diálogo sin ruidos y compresión de gama dinámica.

El audio MPEG-2 admite una amplia gama de velocidades binarias desde 32 kbit/s hasta 1 066 kbit/s. Esta amplia gama podrá realizarse dividiendo la trama audio MPEG-2 en dos partes:

– el tren de bits primario, que transporta la información estereofónica compatible MPEG-1 a una velocidad máxima de 384 kbit/s; y

– el tren de bits de extensión, que transporta toda la información específica MPEG-2 o parte de ésta, es decir, la información multicanal y multilingüe, que no es pertinente a un decodificador de audio MPEG-1.

El tren de bits primario realiza un máximo de 448 kbit/s para la capa I y 384 kbit/s para la capa II. El tren de bits de extensión realiza la velocidad binaria excedente. Si, en el caso de la capa II, se selecciona un total de 384 kbit/s, se puede omitir el tren de bits de extensión. La velocidad binaria no tiene que ser fija, porque MPEG-2 admite velocidad binaria variable, lo que podrá ser de interés en aplicaciones de transmisión en modo transferencia asíncrono o de almacenamiento, por ejemplo, disco vídeo digital.

Esta amplia gama de velocidades binarias permite aplicaciones que requieren una velocidad binaria baja y alta calidad de audio, por ejemplo, si sólo hay que considerar un proceso de codificación y se puede evitar la puesta en cascada. Permite también aplicaciones donde podría ser conveniente utilizar velocidades de datos más altas, es decir, hasta 180 kbit/s por canal, si hay que tener en cuenta la puesta en cascada o el procesamiento posterior. Experimentos realizados por un grupo de especialistas del UIT-R han mostrado que un proceso de codificación se puede repetir nueve veces con la capa II de MPEG-1 sin ninguna degradación subjetiva importante, si la velocidad binaria es suficientemente alta, es decir, 180 kbit/s por canal. Sin embargo, si la velocidad binaria es de sólo 120 kbit/s, no debe haber más de tres procesos de codificación.

3.1.1 Calidad de presentación estereofónica 3/2

El sistema de cinco canales recomendados por UIT-R, SMPTE y UER se denomina estereofonía 3/2 (3 canales frontales/2 canales ambiente) y requiere el tratamiento de cinco canales en el estudio, en los medios de almacenamiento, contribución, distribución, enlaces de emisión y en el hogar.

3.1.2 Compatibilidad hacia atrás/hacia adelante con la Norma ISO/CEI 11172-3

Para varias aplicaciones se piensa mejorar el actual sistema de sonido estereofónico 2/0 paso por paso, transmitiendo canales sonoros adicionales (central, ambiental) sin utilizar funcionamiento de radiodifusión simultánea. El decodificador de sonido multicanal tiene que ser compatible hacia atrás/hacia adelante con el formato de sonido existente.

La *compatibilidad hacia atrás* significa que el actual decodificador bicanal (barato) debe decodificar adecuadamente la información estereofónica 2/0 básica del tren de bits multicanal (véase la Fig. 4). Esto supone la provisión de matrices de compatibilidad que utilizan coeficientes adecuados de mezclado reductor para crear las señales estereofónicas compatibles L_0 y R_0, que se muestran en la Fig. 5. La matriz inversa para recuperar los cinco canales de audio separados en el decodificador MPEG-2 se muestra también en la misma Figura. Las ecuaciones de matriz básicas utilizadas en el codificador para convertir las cinco señales de entrada L, R, C, L_s y R_s en los cinco canales de transporte T_0, T_1, T_2, T_3 y T_4 son:

$$T_0 = L_0 = (\alpha \cdot L) + \beta\ (\alpha \cdot C) + \gamma\ (\alpha \cdot L_s)$$

$$T_1 = R_0 = (\alpha \cdot R) + \beta\ (\alpha \cdot C) + \gamma\ (\alpha \cdot R_s)$$

$$T_2 = C^W = \alpha \cdot \beta \cdot C$$

$$T_3 = L_s^W = \alpha \cdot \gamma \cdot L_s$$

$$T_4 = R_s^W = \alpha \cdot \gamma \cdot R_s$$

Para obtener la reducción máxima de la velocidad binaria, T_2, T_3 y T_4 pueden transportar también $(\alpha \cdot L)$ y $(\alpha \cdot R)$ en vez de $(\alpha \cdot \beta \cdot C)$, $(\alpha \cdot \gamma \cdot L_s)$ y $(\alpha \cdot \gamma \cdot R_s)$.

Se han definido cuatro procedimientos de matriz con diferentes coeficientes, α, β, e γ y se pueden elegir en el codificador multicanal MPEG-2. Tres de estos procedimientos añaden la señal central con una atenuación de 3 dB a las señales L y R. Las señales ambientales L_s y R_s se añaden, respectivamente a las señales L y R con 3 dB ó 6 dB de atenuación. La posibilidad de sobrecarga de las señales estereofónicas compatibles L_0 y R_0 se evita mediante el factor de atenuación α que se utiliza en las señales individuales L, R, C, L_s y R_s antes de la codificación en matriz. Uno de estos procedimientos proporciona la compatibilidad con sonido ambiental Dolby ®. Con un formato bicanal, la compatibilidad se puede realizar en MPEG-1. MPEG-2 permite ampliar estas señales a un tamaño de cinco canales discretos.

El cuarto procedimiento significa que no hay ninguna matriz incluida que realmente constituya una clase de modo no compatible hacia atrás para el códec multicanal MPEG-2, en el sentido de que un decodificador MPEG-1 producirá la señal L y R de la mezcla multicanal. En determinadas condiciones de grabación, esta matriz proporcionará la mezcla estereofónica óptima.

FIGURA 4

Compatibilidad hacia atrás de audio MPEG-2 con la Norma ISO/CEI 11172-3 relativa a la información de audio

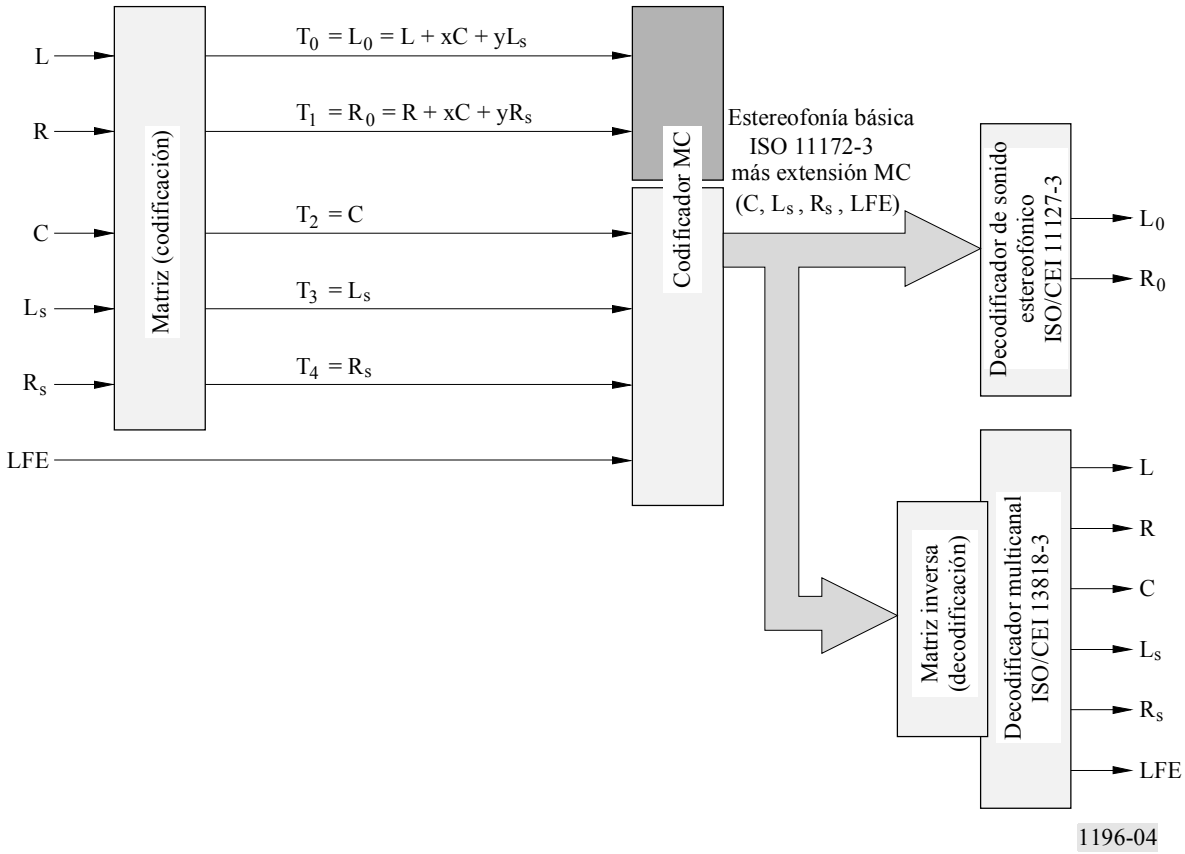
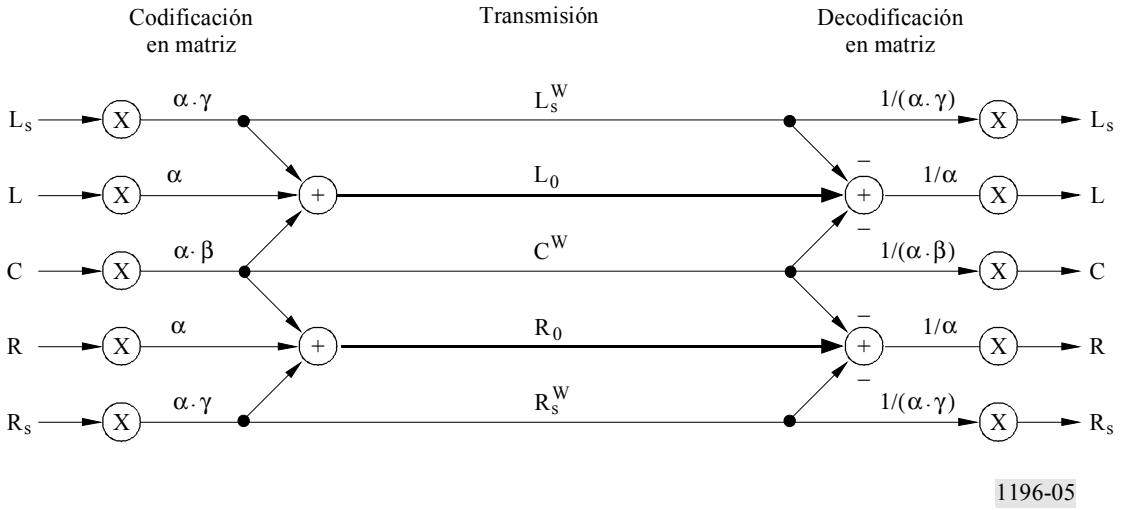


FIGURA 5

Matriz de compatibilidad (codificador) para crear la señal estereofónica básica compatible y la matriz inversa (decodificador) para establecer los cinco canales de audio discretos



Compatibilidad hacia adelante significa que un futuro decodificador multicanal deberá ser capaz de decodificar adecuadamente el tren de bits estereofónico 2/0 básico.

La compatibilidad se realiza utilizando el campo de datos auxiliares de la trama de audio ISO/CEI 11172-3 para la provisión de canales adicionales (véase la Fig. 6). La «longitud variable» del campo de datos auxiliares ofrece la posibilidad de transportar la información de extensión multicanal completa. Un decodificador audio MPEG-1 bicanal normalizado pasa por alto esta parte del campo de datos auxiliares. Si para la capa II la velocidad binaria para la señal de audio multicanal excede de 384 kbit/s, se añade una parte de extensión a la parte compatible MPEG-1. Sin embargo, toda la información sobre la señal estereofónica compatible tiene que mantenerse en la parte compatible MPEG-1. En este caso, la trama de audio MPEG-2 consiste en la parte compatible MPEG-1 y la parte de extensión (no compatible). Esto se muestra en la Fig. 7.

Un ejemplo de esta estrategia es el sistema EU147 DAB, que no proporcionará sonido multicanal en la primera generación. Por consiguiente, la extensión a sonido ambiente digital tiene que ser compatible hacia atrás y hacia adelante con un decodificador de audio MPEG-1.

FIGURA 6

Compatibilidad hacia atrás con la Norma ISO/CEI 11172-3 y la sintaxis de audio MPEG: Campo de datos auxiliares de la trama de la capa II de MPEG-1 que transporta información de extensión multicanal

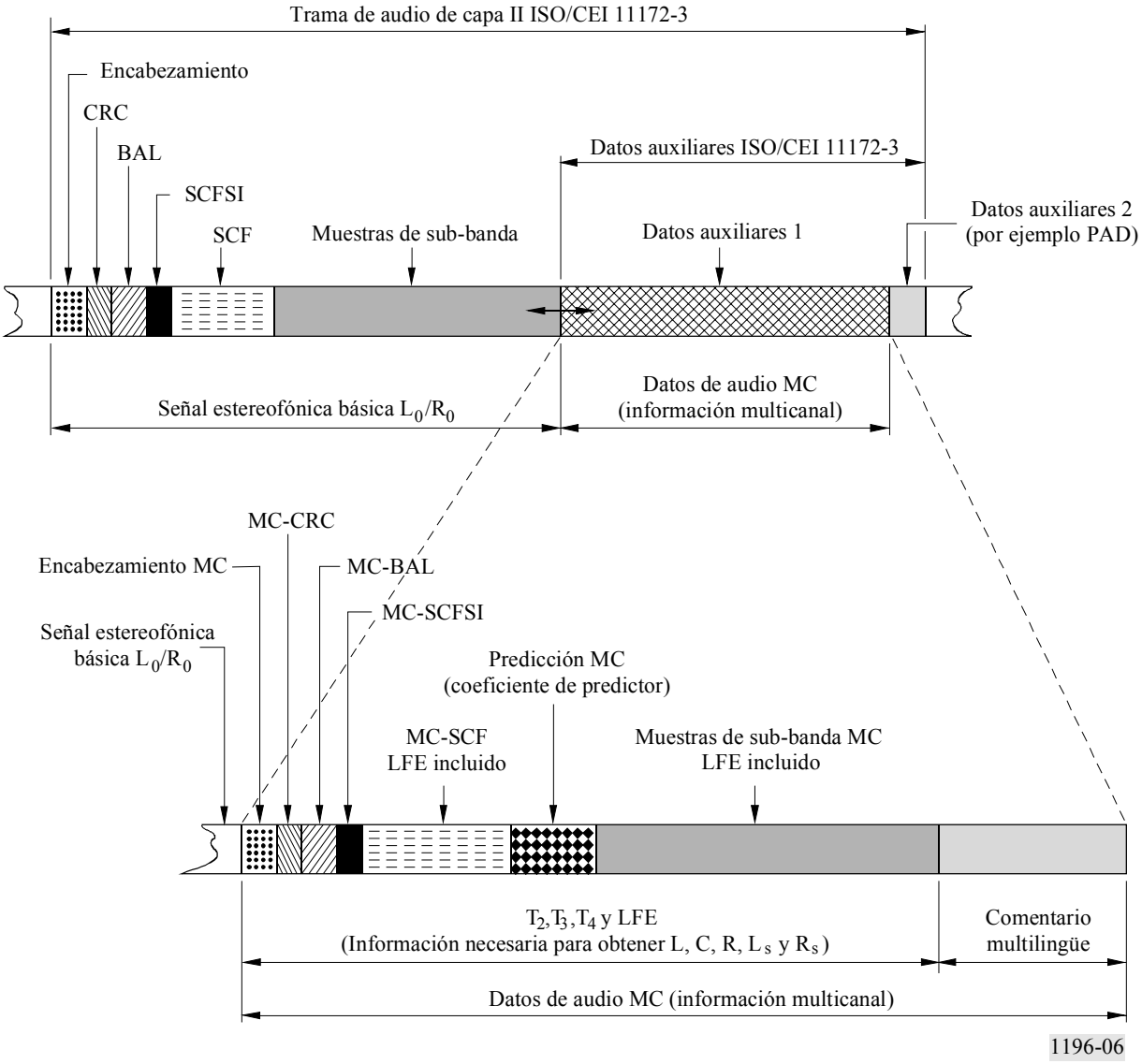
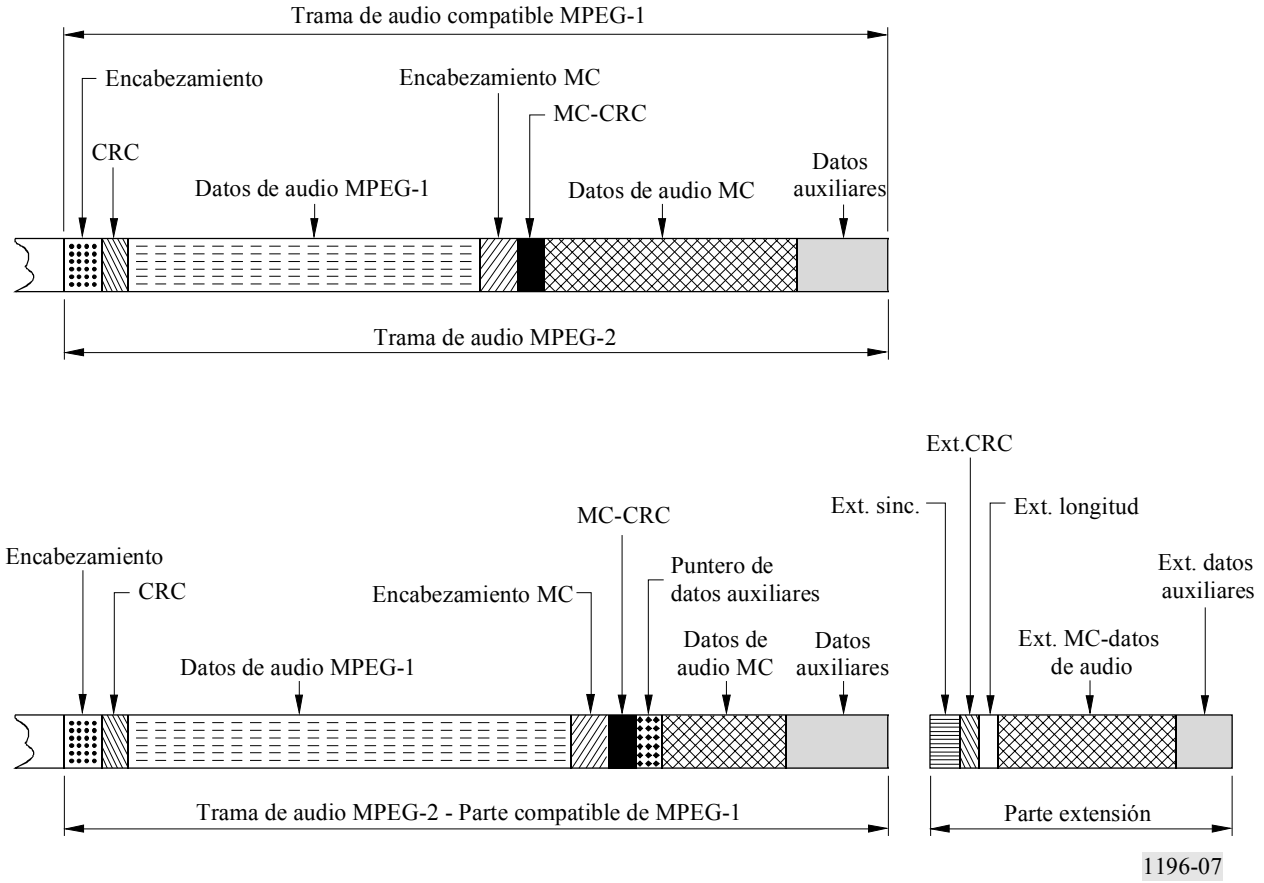


FIGURA 7

Trama de audio multicanal de capa II (audio MPEG-2) de ISO/CEI 13818-3 que consiste en la parte compatible MPEG-1 y la parte extensión



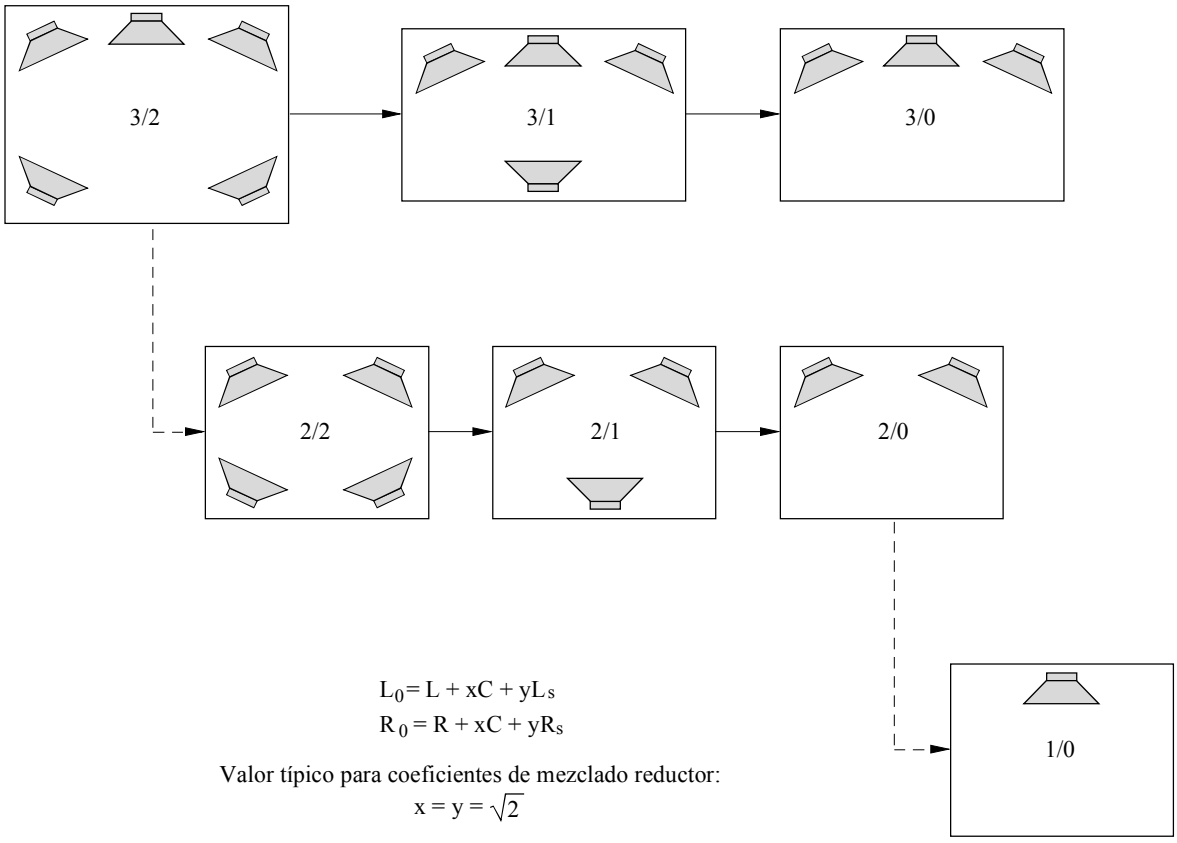
3.1.3 Compatibilidad descendente

En relación con la presentación estereofónica de la señal de audio, los grupos de especialistas del UIT-R, SMPTE, y UER recomiendan un sistema de cinco canales como el formato de sonido ambiente de referencia con un canal central, C, y dos canales de ambiente, L_s, R_s, además de los canales estereofónicos izquierdo y derecho frontales, L y R. Esto se denomina «estereofonía 3/2» (tres canales frontales y dos canales de ambiente) y requiere el tratamiento de cinco canales en el estudio, medios de almacenamiento, contribución, distribución, enlaces de misión y en el hogar.

Con una jerarquía de formatos de sonido que proporciona un número mayor de canales y calidad de presentación estereofónica reducida (hasta estereofonía 2/0 o incluso monofonía) y un conjunto correspondiente de ecuaciones de mezclado reductor, la capa II de audio MPEG-2 proporciona compatibilidad descendente como se muestra en la Fig. 8. Son formatos de sonido de nivel más bajo alternativos útiles 3/1, 3/0, así como 2/2, 2/1, 2/0 y 1/0 que se pueden utilizar cuando se aplican constricciones económicas o de capacidad de canales en el enlace de transmisión o cuando sólo se desea un número menor de canales de reproducción, como es el caso en la recepción portátil de programas de televisión.

FIGURA 8

Opciones de mezclado reductor de canales ambiente de audio MPEG-2 con mezclados reductores de 3/2 hasta 1/0



1196-08

3.1.4 Extensión multilingüe y servicios asociados

En particular para las aplicaciones de TVAD se requiere no sólo la estereofonía multicanal sino también servicios asociados, tales como programas bilingües o diálogos/comentarios multilingües, además del servicio principal. La capa II de audio MPEG-2 proporciona configuraciones de canales sonoros alternativos en el sistema de sonido multicanal, por ejemplo, la aplicación del segundo programa estereofónico pudiera ser un programa estereofónico 2/0 bilingüe o la transmisión de una señal binaural adicional. Otras configuraciones podrían consistir en un sonido ambiental 3/2 más servicios acompañantes (por ejemplo, diálogo sin ruidos para las personas con deficiencias auditivas, comentario para las personas con deficiencias visuales, comentarios multilingües, etc.). Para estos servicios, se puede utilizar la extensión multilingüe o el campo de datos auxiliares, ambos proporcionados por el tren de bits de la capa II de MPEG-2.

Un caso fácil de prestación de un servicio multilingüe en combinación con sonido ambiental es cuando la contribución hablada no forma parte del entorno acústico que se está visualizando. En otras palabras, es relativamente fácil proporcionar efectos deportivos de sonido ambiente más múltiples canales de comentarios monofónicos de idiomas. En cambio, el sonido ambiente con drama requerirá una nueva mezcla de cinco canales para cada idioma adicional.

Un aspecto importante es ciertamente la mezcla final en el decodificador, que significa, la reproducción de un comentario/dialogo seleccionado (por ejemplo, mediante un altavoz central) junto con el mezclado reductor estereofónico común de música/efectos (como ejemplo cabe citar los documentales, los reportajes deportivos). Si se requiere compatibilidad hacia atrás, las señales básicas tienen que contener la información de la señal primaria de comentario/dialogo, que tiene que ser substraída en el decodificador multicanal cuando se selecciona un comentario/dialogo alternativo.

Además de estos servicios, los radiodifusores deben considerar también los servicios para las personas con deficiencias auditivas y visuales. En el caso de personas con deficiencias auditivas, sería más ventajoso un canal de diálogo sin ruidos (es decir, sin efectos sonoros). Para las personas con deficiencias visuales, se necesitaría un canal descriptivo. En ambos casos, estos servicios se podrán transmitir a una velocidad binaria baja de 48 kbit/s aproximadamente con la técnica de codificación de frecuencia de muestreo más baja que proporciona una calidad vocal excelente a una velocidad binaria de 64 kbit/s e incluso más baja, lo que exigiría muy poca capacidad disponible del canal de transmisión.

3.1.5 Canal de efectos de baja frecuencia

De acuerdo con los proyectos de nuevas Recomendaciones UIT-R del ex Grupo de Tareas Especiales 10/1 de Radiocomunicaciones, el formato de sonido estereofónico 3/2 debe proporcionar un canal de efecto de baja frecuencia (LFE) facultativo además de los canales principales de gama completa capaces de transportar señales en la gama de frecuencias 20 Hz a 120 Hz. La finalidad de este canal es que los oyentes que así lo eligen puedan ampliar el contenido de bajas frecuencias del programa de audio desde el punto de vista de las bajas frecuencias y de su nivel. Desde la perspectiva del productor, esto puede traducirse en fijaciones más pequeñas del espacio sobre los canales de audio principales.

3.2 Estrategias de codificación compuesta para audio multicanal

Si se utilizan métodos de codificación compuesta para un programa de audio que consiste en más de un canal, la velocidad binaria requerida no aumenta proporcionalmente en función del número de canales. Para audio multicanal, la técnica de codificación compuesta es muy eficaz, porque hay muchas correlaciones, tanto en la propia señal, como en la percepción estereofónica de esta señal. En el modo codificación compuesta se eliminan las porciones no pertinentes y redundantes de las señales estereofónicas. Se pueden utilizar los siguientes efectos:

3.2.1 Diafonía dinámica

Una cierta porción de las señales estereofónicas, típicamente en la región de altas frecuencias, no contribuye a la localización de las fuentes de sonido. Esta porción puede ser reproducida por cualquier altavoz. Se puede aplicar codificación estereofónica de intensidad basándose en el hecho de que para frecuencias más altas la localización depende más de la forma espectral, es decir, la energía de la señal en función de la frecuencia, que de la información de fase. En comparación con la codificación de estereofonía mixta o de intensidad definida para la capa I y la capa II de MPEG-1, la diafonía dinámica representa una manera mucho más flexible de codificar la señal de extensión multicanal de MPEG-2. La gama de frecuencias de audio se divide en 12 grupos de sub-bandas. Para cada uno de estos grupos, se puede aplicar uno de los 15 casos diferentes. Es posible no transmitir la información de asignación de bits y las muestras cuantificadas de uno, dos o tres de los canales de transmisión T_2, T_3, T_4. Sólo hay que transmitir los factores de escala correspondientes. En el decodificador, las muestras faltantes son sustituidas por las muestras del canal de transmisión correspondiente.

3.2.2 Codificación fantasma de canal central

El canal central proporciona una posición estable, en particular para las señales de audio que se supone estén en el centro, tales como un diálogo, y especialmente en el caso de una zona de gran escucha. Los experimentos han demostrado que la ventaja de un canal central no es afectada si dicho canal central está limitado en banda a una frecuencia superior de unos 9 kHz y las altas frecuencias restantes se transmiten por los canales L y R y representan así un centro fantasma en altas frecuencias.

3.2.3 Predicción multicanal adaptable

Algunas señales estereofónicas contienen porciones coherentes entre canales, que en principio se podrán transmitir por un canal en vez de por dos. En el caso de predicción multicanal, que se puede utilizar individualmente en cada uno de los 12 grupos de sub-bandas, las señales T_2, T_3 y T_4 se predicen a partir de los canales de transmisión T_0 y T_1 de la señal estereofónica básica. En vez de transmitir las muestras de sub-banda cuantificadas, sólo se transmite el error de predicción junto con los coeficientes de predicción y la información sobre la compensación de retardo de tiempo que se puede utilizar por razones de mayor eficacia. La ganancia de predicción depende más bien de la estructura de señal de sub-banda. Las señales tonales estacionarias muestran una ganancia mucho mayor que los componentes transitorios de una señal de audio.

3.2.4 Umbral enmascarado común

La capacidad de procesamiento del sistema de audio está limitada a un determinado grado. No podrá percibir ciertos detalles de canales sonoros individuales en una presentación multicanal. En la capa II de MPEG-2 se puede explotar el enmascaramiento multicanal en forma de umbral enmascarado común. En el decodificador, cada umbral enmascarado intracanal para cada una de las cinco señales sonoras de entrada, L, C, R, L_s, R_s, se calcula de la misma manera que en el codificador MUSICAM estereofónico básico. Sin embargo, las muestras de sub-banda por canal son cuantificadas considerando el umbral individual más alto, teniendo en cuenta el efecto de enmascaramiento entre canales, denominado diferencias de nivel de enmascaramiento. Esto se caracteriza por un umbral enmascarado decreciente cuando el enmascarador está separado en el espacio.

Sin embargo, la utilización del umbral enmascarado común en vez del umbral enmascarado intracanal entraña que hay que tener en cuenta la disposición de altavoces y la zona de escucha máxima. La escucha muy cerca en un altavoz puede resultar en la percepción de ruido de codificación. Por consiguiente, este algoritmo se utiliza solamente en caso de una falta predominante de capacidad de bits. Si las crestas de la velocidad binaria requerida que varían dinámicamente son más altas que la velocidad binaria disponible, se selecciona en el codificador el método de codificación de diafonía dinámica y umbral enmascarado común.

3.2.5 Agrupación de bits común

La velocidad binaria por canal requerida para la codificación porcentual depende de la señal. En consecuencia, cada canal se codifica con velocidad binaria variable, que varía dinámicamente en la gama de unos 100 kbit/s. Si el tren de bits tiene que ser de velocidad binaria constante, la velocidad binaria global de todos los canales tiene que mantenerse constante. Como las velocidades binarias dinámicas individuales de las señales central y ambiental no están completamente correlacionadas (incluso pueden no estar correlacionadas), resulta un efecto de nivelación de las crestas de velocidad binaria global. Esta agrupación de bits común, que es utilizada por las técnicas de intercambio de bits de la capa II, es particularmente eficaz en el modo de codificación independiente.

3.2.6 Conmutación de canal de transmisión

Mientras se transmiten las dos señales estereofónicas básicas, L_0 y R_0 por los canales de transmisión compatibles MPEG-1 T_0 y T_1, se puede transmitir cualquier combinación de las señales adicionales por los canales de transmisión T_2, T_3 y T_4. Esto significa que la matriz presente en la Fig. 3 no es la única versión. La elección del subconjunto de 8 combinaciones posibles se efectúa trama por trama para minimizar la velocidad binaria global. Esto se puede hacer como diafonía dinámica y la predicción de multicanal adaptable en grupos de sub-bandas individuales.

4 Resumen

Las Normas Internacionales ISO/CEI 11172-3 y 13818-3 proporcionan métodos de codificación de audio eficaces y flexibles que las hacen particularmente adecuadas para una amplia gama de aplicaciones a los servicios de radiodifusión. Audio MPEG-1 ha establecido una técnica de codificación para señales monofónicas o estereofónicas que se puede utilizar con o sin un esquema de codificación de imágenes, y que puede codificar señales de audio de alta calidad en la gama de 192 a unos 100 kbit/s por programa monofónico, proporcionando un margen suficiente para la puesta en cascada y el posprocesamiento a velocidades binarias más altas.

Se ha completado un paso importante de la primera fase del desarrollo de la codificación de audio de alta calidad para uso generalizado en aplicaciones de radiodifusión, telecomunicación, computador y consumidor, con la Norma ISO/CEI 11172-3, pero la compleción de MPEG-1 no es el fin de la normalización de sistemas de codificación de audio de alta calidad. El sistema de codificación multicanal de audio MPEG-2, que asegura compatibilidad hacia adelante y hacia atrás con las señales de audio codificadas según la Norma ISO/CEI 11172-3, se ha diseñado para aplicaciones universales con y sin imagen acompañante. Las aplicaciones previstas, además de radiodifusión de audio digital, son sistemas de televisión digital, grabación en cinta de vídeo digital y medios de almacenamiento interactivos.

La configurabilidad con respecto a la asignación de canales de sonido y a la velocidad binaria ofrece combinaciones útiles de diversos niveles de funcionamiento estereofónico multicanal y diversos números de canales en el modo de codificación compuesta e independiente.

Anexo 2

Norma de compresión de audio digital (AC-3) (Norma ATSC)

ÍNDICE

Página

Preámbulo 23

1 Introducción 23

1.1 Justificación 23

1.2 Codificación 25

1.3 Decodificación 26

2 Alcance 26

3 Referencias 27

3.1 Referencias normativas 27

3.2 Referencias informativas 27

4 Notación, definiciones y terminología 28

4.1 Notación de conformidad 28

4.2 Definiciones 28

4.3 Abreviaturas de terminología 29

5 Sintaxis del tren de bits 33

5.1 Trama de sincronización 33

5.2 Semántica de la especificación de sintaxis 34

5.3 Especificación de la sintaxis 34

5.3.1 Información de sincronización – syncinfo 35

5.3.2 Información del tren de bits – bsi 35

5.3.3 Bloque de audio – audblk 36

5.3.4 Datos auxiliares – auxdata 39

5.3.5 Código detector de errores – errorcheck 40

5.4 Descripción de los elementos del tren de bits 40

5.4.1 Información de sincronización – syncinfo 40

5.4.2 bsi – Información del tren de bits 41

Página

5.4.3 audblk – Bloque de audio 46
5.4.4 auxdata – Campo de datos auxiliares 55
5.4.5 errorcheck – Campo de detección de errores de la trama 57
5.5 Limitaciones del tren de bits 57
6 Decodificación del tren de bits AC-3 57
6.1 Introducción 57
6.2 Resumen del proceso de decodificación 60
6.2.1 Tren de bits de entrada 60
6.2.2 Sincronización y detección de errores 60
6.2.3 Desempaquetado de BSI, información lateral 61
6.2.4 Decodificación de exponentes 61
6.2.5 Atribución de bits 61
6.2.6 Procesamiento de las mantisas 61
6.2.7 Desacoplamiento 62
6.2.8 Rematrización 62
6.2.9 Compresión de la gama dinámica 62
6.2.10 Transformación inversa 62
6.2.11 Ventana, superposición/adición 62
6.2.12 Submezclado 62
6.2.13 Memoria tampón de salida MIC 62
6.2.14 MIC de salida 62
7 Detalles del algoritmo 63
7.1 Codificación de exponentes 63
7.1.1 Visión de conjunto 63
7.1.2 Estrategia de exponente 63
7.1.3 Decodificación de exponentes 65
7.2 Atribución de bits 69
7.2.1 Visión de conjunto 69
7.2.2 Atribución de bits paramétrica 69
7.2.3 Cuadros de atribución de bits 76

Página

7.3 Cuantificación y decodificación de las mantisas 83

7.3.1 Visión de conjunto 83

7.3.2 Desarrollo de las mantisas para la cuantificación asimétrica ($6 \leq bap \leq 15$) 84

7.3.3 Desarrollo de las mantisas para la cuantificación simétrica ($1 \leq bap \leq 5$) 85

7.3.4 Adición de ruido para mantisas del bit 0 (bap = 0) 85

7.3.5 Desagrupación de las mantisas 87

7.4 Acoplamiento del canal 88

7.4.1 Visión de conjunto 88

7.4.2 Estructura de sub-bandas para el acoplamiento 88

7.4.3 Formato de las coordenadas de acoplamiento 89

7.5 Rematrización 90

7.5.1 Visión de conjunto 90

7.5.2 Definiciones de bandas de frecuencias 91

7.5.3 Técnica de codificación 93

7.5.4 Técnica de decodificación 93

7.6 Normalización del diálogo 93

7.6.1 Visión de conjunto 93

7.7 Compresión de la gama dinámica 95

7.7.1 Control de la gama dinámica; dynrng, dynrng2 95

7.7.2 Compresión intensa, compr, compr2 98

7.8 Submezclado 100

7.8.1 Procedimiento general downmix 101

7.8.2 Submezclado a dos canales 104

7.9 Ecuaciones de la transformación y conmutación de bloques 106

7.9.1 Visión de conjunto 106

7.9.2 Técnica 106

7.9.3 Realización del decodificador 107

7.9.4 Ecuaciones de la transformación 107

7.9.5 Código de gama de ganancia de canal 112

Página

7.10 Detección de errores 112

7.10.1 Verificación de CRC 112

7.10.2 Verificación de la coherencia del tren de bits 115

8 Codificación del tren de bits AC-3 116

8.1 Introducción 116

8.2 Resumen del proceso de codificación 116

8.2.1 MIC de entrada 116

8.2.2 Detección de transitorios 118

8.2.3 Transformación progresiva 119

8.2.4 Estrategia de acoplamiento 120

8.2.5 Construcción del canal de acoplamiento 120

8.2.6 Rematrización 121

8.2.7 Extracción de exponentes 121

8.2.8 Estrategia de exponentes 121

8.2.9 Estrategia de adición de ruido 121

8.2.10 Codificación de exponentes 121

8.2.11 Normalización de mantisas 122

8.2.12 Atribución de bits de núcleo 122

8.2.13 Cuantificación de las mantisas 123

8.2.14 Empaquetamiento de la trama AC-3 123

Apéndice 1 – Trenes elementales AC-3 en el múltiplex MPEG-2. 123

Norma de compresión de audio digital (AC-3) (Norma ATSC)

Preámbulo

Las organizaciones miembros del Joint Committe on InterSociety Coordination (JCIC)* formaron el Advanced Television System Committee (ATSC) de los Estados Unidos, reconociendo que el desarrollo temprano, eficaz y efectivo de un conjunto de normas nacionales resulta esencial para el desarrollo futuro de los servicios de televisión domésticos.

Una de las actividades que investiga el ATSC es la necesidad y, en su caso, la coordinación del desarrollo de normas técnicas nacionales voluntarias destinadas a sistemas de televisión perfeccionados (ATV – Advanced Television Systems). La comisión ejecutiva del ATSC encmendó la tarea de elaborar la norma de ATV norteamericana a un conjunto de grupos de especialistas que trabajaban en el grupo de tecnologías de distribución (T3). Se encomendó al grupo de especialistas de audio (T3/S7) la elaboración de la norma de audio ATV.

Esta Recomendación se redactó inicialmente por el grupo de especialistas de audio como parte de su esfuerzo para documentar la norma de radiodifusión de televisión perfeccionada de Estados Unidos de América. El grupo de tecnologías de distribución en su reunión del 26 de septiembre de 1994 y la totalidad de miembros del ATSC en el 10 de noviembre de 1994 han aprobado este texto como norma ATSC. El Apéndice 1 al Anexo 2 «Trenes elementales AC-3 en el múltiplex MPEG-2» fue aprobado en 2001.

1 Introducción

1.1 Justificación

Con el fin de lograr una grabación o radiodifusión más eficientes de las señales de audio, debe reducirse el volumen de información necesario para representar tales señales. En el caso de señales de audio digital, la cuantía de información digital necesaria para reproducir las muestras de modulación por impulsos codificados (MIC) original puede reducirse aplicando un algoritmo de compresión digital, que produce una representación comprimida digitalmente de la señal original. (El término compresión empleado en este contexto significa la compresión del volumen de información digital que debe almacenarse o grabarse y no la compresión de la gama dinámica de la señal de audio.) El objetivo primordial de los algoritmos de compresión digital es la producción de una representación digital de una señal de audio que, tras su decodificación y reproducción, suene de la misma forma que la señal original, empleando un mínimo de información digital (velocidad de bits) para la representación comprimida (o codificada). El algoritmo de compresión digital AC-3 especificado en esta Recomendación puede codificar de 1 a 5,1 canales audio de origen con

---

* Actualmente el JCIC está constituido por la Electronic Industries Association (EIA), el Institute of Electrical and Electronic Engineers (IEEE), la National Association of Broadcasters (NAB), la National Cable Television Association (NCTA) y la Society of Motion Picture and Television Engineers (SMPTE).

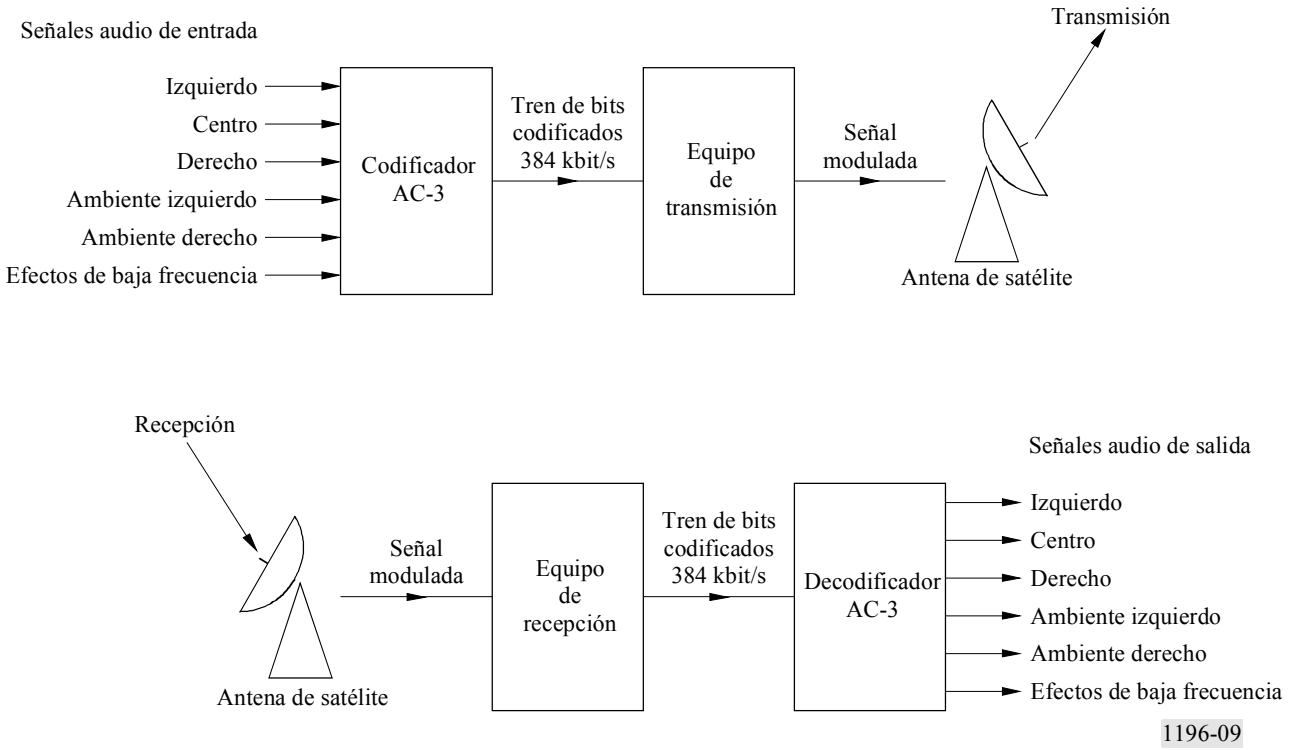
NOTA 1 – Se señala a la atención del usuario la posibilidad de que la cumplimentación de esta norma puede exigir la utilización de inventos amparados por derechos de patentes. Con la publicación de esta norma no se adopta ninguna postura con respecto a la validez de estas reclamaciones o de cualesquiera derechos de patentes conexos. El propietario de la patente sin embargo ha cumplimentado un formulario de complacencia para garantizar la concesión de una licencia sobre unas bases razonables y no discriminatorias a quienes deseen obtener tal licencia. El editor facilitará detalles adicionales.

estructura MIC, transformándolos en un flujo de bits en serie con velocidades de datos comprendidas entre 32 kbit/s y 640 kbit/s. El canal 0,1 se refiere a un canal de anchura de banda fraccional previsto para transportar únicamente señales de baja frecuencia (sonidos graves).

En la Fig. 9 se representa una aplicación práctica del algoritmo. En este ejemplo, se convierte un canal de programa de audio 5,1 en formato MIC que requiere más de 5 Mbit/s (6 canales × 48 kHz × 18 bits = 5,184 Mbit/s) en un tren de bits en serie de 384 kbit/s empleando el codificador AC-3. El equipo de transmisión por satélite convierte este flujo de bits en una transmisión RF que se encamina a un transpondedor de satélite. La compresión digital AC-3, reduce en más de trece veces la anchura de banda y la potencia requeridas por la transmisión. La señal recibida desde el satélite se demodula para formar un tren de bits en serie de 384 kbit/s y se decodifica en el decodificador AC-3. El resultado es la señal original del programa de audio del canal 5,1.

FIGURA 9

Ejemplo de aplicación de AC-3 a la transmisión de audio por satélite



La compresión digital de audio resulta útil siempre que se obtengan ventajas económicas con la reducción del volumen de información digital necesaria para representar la señal de audio. Como ejemplos típicos pueden citarse la radiodifusión de audio terrenal o por satélite, la distribución de audio a través de cables ópticos o metálicos o la grabación de audio en soportes magnéticos, ópticos, de semiconductores o de otro tipo.

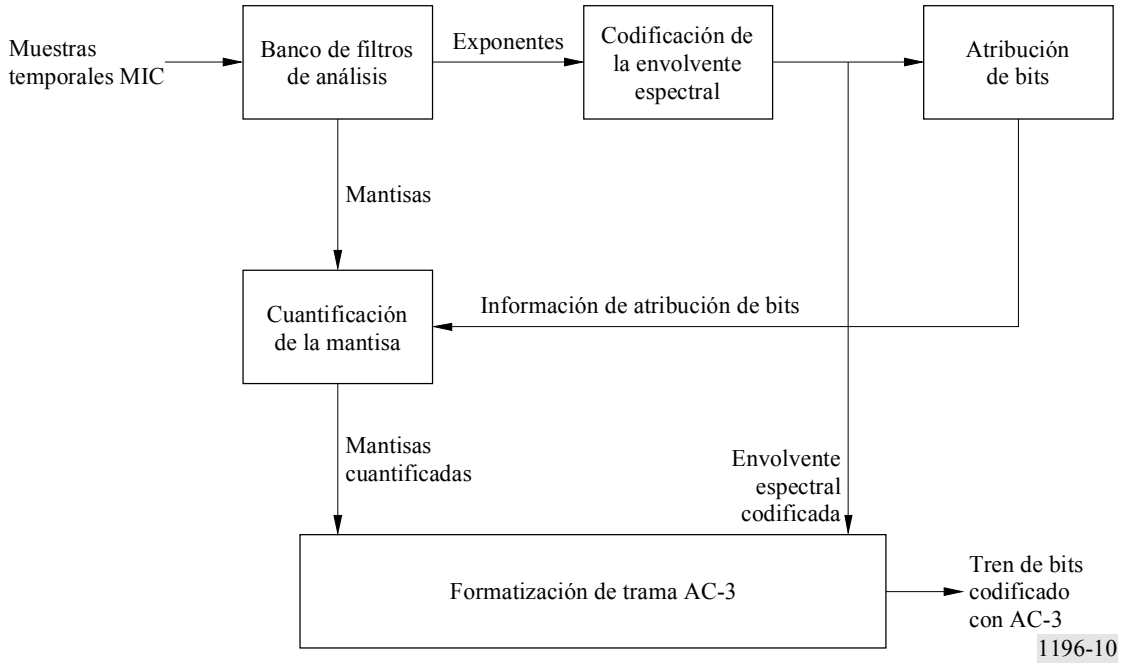
1.2 Codificación

El codificador AC-3 acepta a su entrada una señal audio MIC y genera un tren de bits codificados coherente con esta norma. Los aspectos específicos del proceso de codificación del audio no constituyen requisitos normativos de esta especificación. No obstante, el codificador debe generar un flujo de bits acorde con la sintaxis descrita en el § 5, de forma que, cuando se decodifique según lo establecido en los § 6 y 7, se produzca una señal audio de calidad suficiente para la aplicación prevista. El § 8 contiene información descriptiva del proceso de codificación. Seguidamente se describe brevemente el proceso de codificación.

El algoritmo AC-3 proporciona una elevada ganancia de codificación (relación entre las velocidades de bits de entrada y de salida) mediante la cuantificación gruesa de una representación en el dominio de la frecuencia de la señal de audio. En la Fig. 10 se ilustra un diagrama de bloques de este proceso. La primera fase del proceso de codificación consiste en la transformación de la representación de la señal audio de una sucesión de muestras temporales MIC en un conjunto de bloques de coeficientes de frecuencia. Esto se realiza en el banco de filtros de análisis. Se toman bloques superpuestos de 512 muestras temporales y se multiplican por una ventana temporal, transformándose al dominio de la frecuencia. Debido a la superposición de los bloques, cada muestra de entrada MIC queda representada en dos bloques transformados secuenciales. En consecuencia, la representación en el dominio de la frecuencia puede diezmarse utilizando un factor igual a 2 de forma que cada bloque contiene 256 coeficientes de frecuencia. Los coeficientes de frecuencia individuales se representan utilizando una notación exponencial binaria en forma de una mantisa y un exponente binario. El conjunto de exponentes se codifica a una representación aproximada del espectro de la señal, conocida como envolvente espectral. La rutina de atribución de bits de núcleo emplea esta envolvente espectral para determinar cuántos bits han de utilizarse para la codificación de cada mantisa individual. La envolvente espectral y las mantisas cuantificadas aproximadamente correspondientes a 6 bloques de audio (1 536 muestras de audio), se ensamblan en una trama AC-3. El tren de bits AC-3 es una sucesión de tramas AC-3.

FIGURA 10

Codificador AC-3



El codificador AC-3 real es más complejo que el indicado en la Fig. 10 y comprende las siguientes funciones no representadas en esa figura:

– se agrega una cabecera de trama que contiene información (velocidad de bits, velocidad de muestras, número de canales codificados, etc.) necesaria para la sincronización y la decodificación del flujo de bits codificados;
– se utilizan códigos detectores de errores para permitir que el decodificador verifique si una trama de datos recibida está exenta de errores;
– la resolución espectral del banco de filtros de análisis puede modificarse dinámicamente para que se adapte mejor a las características tiempo/frecuencia de cada bloque de audio;
– la envolvente espectral puede codificarse con una resolución tiempo/frecuencia variable;
– puede efectuarse una atribución de bits más compleja y modificarse la rutina de atribución de los bits núcleo a fin de conseguir una atribución de bits más optimizada;
– en frecuencias altas pueden acoplarse los canales conjuntamente a fin de lograr una mayor ganancia de codificación para el funcionamiento a velocidades de bits reducidas;
– en el modo bicanal puede realizarse selectivamente un proceso de rematrización a fin de proporcionar una ganancia de codificación adicional y permitir la obtención de resultados mejorados en el caso de que se decodifique la señal bicanal con un decodificador de entorno matricial.

1.3 Decodificación

Básicamente el proceso de decodificación es el inverso del proceso de codificación. El codificador representado en la Fig. 11, debe sincronizar el flujo de bits codificados, comprobar la presencia de errores y desformatizar los diversos tipos de datos tales como la envolvente espectral codificada y las mantisas cuantificadas. Se ejecuta la rutina de atribución de bits empleándose los resultados para desempaquetar y descuantificar las mantisas. La decodificación de la envolvente espectral proporciona los exponentes. Los exponentes y las mantisas se transforman al dominio del tiempo para generar las muestras temporales MIC decodificadas.

El decodificador AC-3 real es más complejo que el representado en la Fig. 11 y comprende las siguientes funciones no indicadas anteriormente:

– cuando se detecte un error de datos debe aplicarse un enmudecimiento u ocultación del error;
– deben desacoplarse los canales cuyo contenido de altas frecuencias se acopló conjuntamente;
– cuando se hayan rematrizado los canales debe aplicarse la desmatrización (en el modo de 2 canales);
– debe alterarse dinámicamente la resolución del banco de filtros de síntesis, de la misma forma que se hizo con el banco de filtros de análisis del codificador en el proceso de codificación.

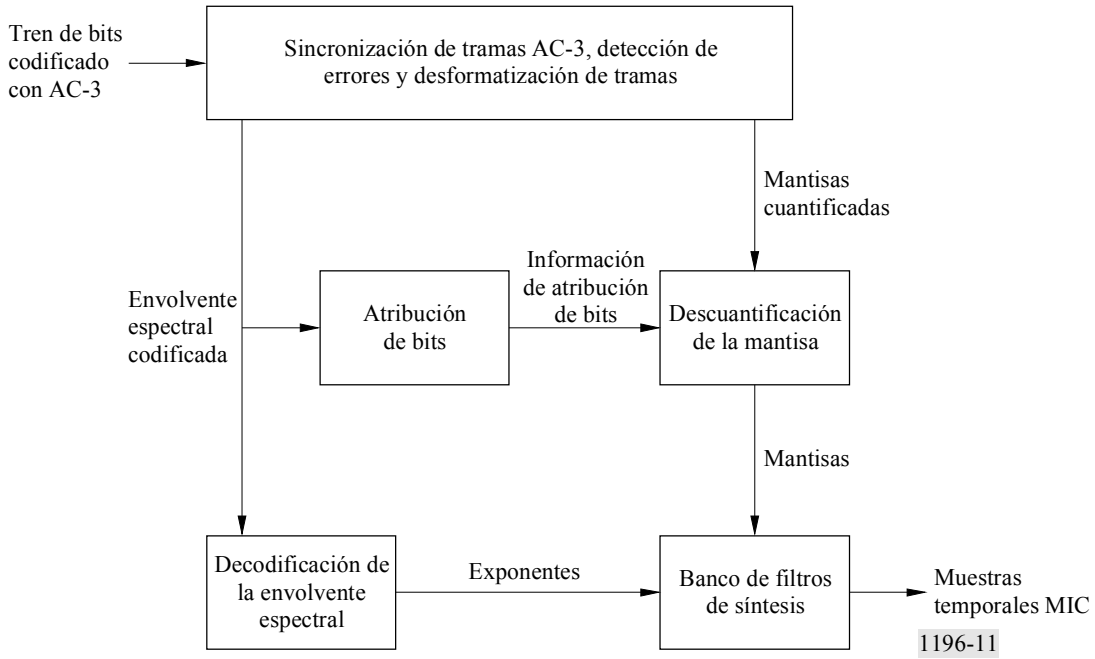
2 Alcance

La parte dispositiva de esta norma especifica una representación codificada de información de audio así como el proceso de decodificación. Se incluye información descriptiva del proceso de codificación. La representación codificada aquí representada resulta adecuada para su empleo en aplicaciones de transmisión o grabación de señales de audio. La representación codificada puede abarcar de 1 a 5 canales de audio de anchura de banda total junto con un canal de realce de baja frecuencia. Esta especificación sustenta una amplia gama de velocidades de bits codificados.

La designación abreviada de este algoritmo de codificación de señales de audio es AC-3.

FIGURA 11

Decodificador AC-3



3 Referencias

3.1 Referencias normativas

Los siguientes textos contienen disposiciones que mediante referencia en esta Recomendación constituyen estipulaciones de esta norma. En el momento de la publicación estaban vigentes las ediciones indicadas. Todas las normas son objeto de revisión, por lo que se alienta a los usuarios de esta norma que investiguen la posibilidad de aplicar las ediciones más recientes de los documentos enumerados seguidamente.

Ninguna.

3.2 Referencias informativas

Los textos que siguen contienen información sobre el algoritmo descrito en esta norma y pueden ser útiles a quienes utilicen o deseen comprender esta especificación. En caso de informaciones contradictorias se considerará correcta la información contenida en esta norma.

TODD, C. y otros [febrero de 1994] AC-3: Flexible perceptual coding for audio transmission and storage. AES 96th Convention. Preprint 3796.

EHMER, R.H. [agosto de 1959] Masking patterns of tones. *J. Acoust. Soc. Am.*, Vol.31, 1115-1120.

EHMER, R.H. [septiembre de 1959] Masking of tones vs. noise bands. *J. Acoust. Soc. Am.*, Vol 31, 1253-1256.

MOORE, B.C.J. y GLASBERG, B.R. [1987] Formulae describing frequency selectivity as a function of frequency and level, and their use in calculating excitation patterns. *Hearing Resear*ch, Vol. 28, 209-225.

ZWICKER, E. [febrero de 1961] Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *J. Acoust. Soc. Am.*, Vol. 33, 248.

4 Notación, definiciones y terminología

4.1 Notación de conformidad

Cuando se utiliza en esta Recomendación el tiempo futuro (de mandato) o la palabra «*deberá*» ello indica una disposición obligatoria en esta norma. La palabra «*debería*» indica una disposición recomendada pero no obligatoria. La palabra «*puede*» o «*podrá*» implica una característica cuya presencia no excluye la conformidad y que puede o no estar presente, a voluntad del realizador.

4.2 Definiciones

En esta Recomendación se utilizan diversos términos. A continuación se facilitan definiciones explicativas del significado de alguno de los términos utilizados.

Banda de acoplamiento:	Banda de coeficientes de la transformación del canal acoplado que comprende una o más sub-bandas del canal de acoplamiento.
Bloque de audio:	Conjunto de 512 muestras de audio constituidas por 256 muestras del bloque de audio precedente y 256 muestras temporales nuevas. Cada 256 muestras de audio se genera un nuevo bloque de audio. Cada muestra de audio está representada en dos bloques de audio.
Canal acoplado:	Canal de anchura de banda total cuya información de alta frecuencia se combina en el canal acoplado.
Canal de acoplamiento:	Canal constituido por la combinación de la información de alta frecuencia procedente de los canales acoplados.
Canal de anchura de banda total (fbw):	Canal de audio con la anchura de banda total de audio. Todos los canales (izquierdo, central, derecho, ambiente izquierdo ambiente derecho), salvo el canal lfe, son canales fbw.
Canal de efectos de baja frecuencia (lfe):	Canal único facultativo de anchura de banda limitada (<120 Hz) previsto para su reproducción a un nivel de +10 dB con respecto a los canales fbw. El canal lfe facultativo proporciona elevados niveles de presión sonora para los sonidos de baja frecuencia.
Canal independiente:	Canal cuya información de alta frecuencia no está combinada en el canal de acoplamiento (el canal lfe siempre es independiente).
Coeficiente:	La transformación convierte muestras en el dominio del tiempo en coeficientes en el dominio de la frecuencia.
Envolvente espectral:	Estimación espectral consistente en el conjunto de exponentes obtenidos mediante la decodificación de los exponentes codificados. Es similar (pero no idéntica) al conjunto de coeficientes originales.
Juego de exponentes:	Conjunto de exponentes de un canal independiente, del canal de acoplamiento o de la porción de baja frecuencia de un canal acoplado.

Segmento: Número del coeficiente de frecuencia, segmento de frecuencia número *n*. La transformación TDAC de 512 puntos genera 256 coeficientes de frecuencia o segmentos de frecuencia.

Sub-banda de acoplamiento: Sub-banda constituida por un grupo de 12 coeficientes de transformación del canal de acoplamiento.

Submezclado: Combinación (o submezclado) del contenido de *n* canales originales para generar *m* canales siendo $m < n$.

Trama de sincronización: Unidad del tren de bits en serie que puede decodificarse completamente. La trama de sincronización comienza con un código de sincronización y contiene 1 536 muestras de audio codificadas.

Ventana: Vector temporal que se multiplica por un bloque de audio para proporcionar un bloque de audio enventanado. La forma de la ventana determina la selectividad de frecuencia del banco de filtros y proporciona las características de solapamiento/adición adecuadas para evitar los artefactos de bloqueo.

4.3 Abreviaturas de terminología

Para hacer referencia a los elementos empleados en el formato AC-3 se utilizan diversas abreviaturas. A continuación se facilita una lista de correspondencias entre cada abreviatura y el término que representa. En la mayoría de los casos se proporciona una referencia a información ulterior. En esta Recomendación se efectúa un uso exhaustivo de estas abreviaturas. Las abreviaturas se escriben con letras minúsculas y tienen una longitud máxima de 12 caracteres, siendo adecuadas para su utilización con cualquier código de programación o lenguaje ensamblador de alto nivel de un computador. Se alienta a los usuarios de esta norma a que utilicen las mismas abreviaturas en cualquier código fuente de ordenador o en la documentación relativa a realizaciones físicas o lógicas.

Abreviatura	Término	Referencia
acmod	Modo codificación de audio (audio coding mode)	§ 5.4.2.3
addbsi	Información de flujo de bits adicional (additional bit stream information)	§ 5.4.2.31
addbsie	Existe información de flujos de bits adicional (additional bit stream information exists)	§ 5.4.2.29
addbsil	Longitud de la información de flujo de bits adicional (additional bit stream information length)	§ 5.4.2.30
audblk	Bloque de audio (audio block)	§ 5.4.3
audprodie	Existe información de producción de audio (audio production information exists)	§ 5.4.2.13
audprodi2e	Existe información de producción de audio, ch2 (audio production information exists, ch2)	§ 5.4.2.21
auxbits	Bits de datos auxiliares (auxiliary bata bits)	§ 5.4.4.1
auxdata	Campo de datos auxiliares (auxiliary data field)	§ 5.4.4.1
auxdatae	Existen datos auxiliares (auxiliary data exists)	§ 5.4.4.3

Abreviatura	Término	Referencia
auxdatal	Longitud de los datos auxiliares (auxiliary data length)	§ 5.4.4.2
baie	Existe información de atribución de bits (bit allocation information exists)	§ 5.4.3.30
bap	Puntero de atribución de bits (bit allocation pointer)	
bin	Segmento de coeficientes de frecuencia en el índice [bin] (frequency coefficient bin in index [bin])	§ 5.4.3.13
blk	Bloque en el índice de filas [blk] (block in array index [blk])	
blksw	Bandera de conmutación de bloques (block switch flag)	§ 5.4.3.1
bnd	Banda en el índice de filas [bnd] (band in array index)	
bsi	Información del tren de bits (bit stream information)	§ 5.4.2
bsid	Identificación del tren de bits (bit stream identification)	§ 5.4.2.1
bsmod	Modo tren de bits (bit stream mode)	§ 5.4.2.2
ch	Canal en el índice de filas [ch] (channel in array index [ch])	
chbwcod	Código de anchura de banda de canal (channel bandwidth code)	§ 5.4.3.24
chexpstr	Estrategia del exponente del canal (channel exponent strategy)	§ 5.4.3.22
chincpl	Canal en el acoplamiento (channel in coupling)	§ 5.4.3.9
chmant	Mantisas del canal (channel mantissas)	§ 5.4.3.61
clev	Coeficiente de nivel de mezcla central (center mixing level coefficient)	§ 5.4.2.4
cmixlev	Nivel de mezcla central (center mix level)	§ 5.4.2.4
compr	Palabra de ganancia de compresión (compression gain word)	§ 5.4.2.10
compr2	Palabra de ganancia en compresión, ch2 (compression gain word, ch2)	§ 5.4.2.18
compre	Existe palabra de ganancia de compresión (compression gain word exists)	§ 5.4.2.9
compr2e	Existe palabra de ganancia de compresión, ch2 (compression gain word exists, ch2)	§ 5.4.2.17
copyrightb	Bit de derechos de autor (copyright bit)	§ 5.4.2.24
cplabsexp	Exponente absoluto de acoplamiento (coupling absolute exponent)	§ 5.4.3.25
cplbegf	Código de frecuencia de comienzo de acoplamiento (coupling begin frequency code)	§ 5.4.3.11
cplbndstrc	Estructura de banda de acoplamiento (coupling band structure)	§ 5.4.3.13
cplco	Coordenada de acoplamiento (coupling coordinate)	§ 7.4.3
cplcoe	Existen coordenadas de acoplamiento (coupling coordinates exists)	§ 5.4.3.14
cplcoexp	Exponente de la coordenada de acoplamiento (coupling coordinate exponent)	§ 5.4.3.16
cplcomant	Mantisa de la coordenada de acoplamiento (coupling coordinate mantissa)	§ 5.4.3.17
cpldeltba	dba de acoplamiento (coupling dba)	§ 5.4.3.53
cpldeltbae	Existe dba de acoplamiento (coupling dba exists)	§ 5.4.3.48
cpldeltlen	Longitud dba de acoplamiento (coupling dba length)	§ 5.4.3.52
cpldeltnseg	Número de segmentos dba de acoplamiento (coupling dba number of segments)	§ 5.4.3.50
cpldeltoffst	Desplazamiento dba de acoplamiento (coupling dba offset)	§ 5.4.3.51
cplendf	Código de frecuencia de final de acoplamiento (coupling end frequency code)	§ 5.4.3.12

Abreviatura	Término	Referencia
cplexps	Exponentes de acoplamiento (coupling exponents)	§ 5.4.3.26
cplexpstr	Estrategia de exponente de acoplamiento (coupling exponent strategy)	§ 5.4.3.21
cplfgaincod	Código de ganancia rápida de acoplamiento (coupling fast gain code)	§ 5.4.3.39
cplfleak	Inicialización pérdida rápida de acoplamiento (coupling fast leak initialization)	§ 5.4.3.45
cplfsnroffst	Desplazamiento snr fino de acoplamiento (coupling fine snr offset)	§ 5.4.3.38
cplinu	Acoplamiento en uso (coupling in use)	§ 5.4.3.8
cplleake	Existe inicialización pérdida de acoplamiento (coupling leak initialization exists)	§ 5.4.3.44
cplmant	Mantisas de acoplamiento (coupling mantissas)	§ 5.4.3.61
cplsleak	Inicialización pérdida lenta acoplamiento (coupling slow leak initialization)	§ 5.4.3.46
cplstre	Existe estrategia de acoplamiento (coupling strategy exists)	§ 5.4.3.7
crc1	Palabra 1 de verificación de redundancia cíclica-crc (crc-cyclic redundancy check word 1)	§ 5.4.1.2
crc2	Palabra 2 de verificación de redundancia cíclica-crc (crc-cyclic redundancy check word 2)	§ 5.4.5.2
crcrsv	Bit reservado para crc (crc reserved bit)	§ 5.4.5.1
csnroffst	Desplazamiento snr grueso (coarse snr offset)	§ 5.4.3.37
d15	Modo codificación exponente d15 (d15 exponent coding mode)	§ 5.4.3.21
d25	Modo codificación exponente d25 (d25 exponent coding mode)	§ 5.4.3.21
d45	Modo codificación exponente d45 (d45 exponent coding mode)	§ 5.4.3.21
dba	Atribución de bit delta (delta bit allocation)	§ 5.4.3.47
dbpbcod	dB por código de bit	§ 5.4.3.34
deltba	dba de canal (channel dba)	§ 5.4.3.57
deltbae	Existe dba de canal (channel dba exists)	§ 5.4.3.49
deltbaie	Existe información dba (dba information exists)	§ 5.4.3.47
deltlen	Longitud dba de canal (channel dba length)	§ 5.4.3.56
deltnseg	Número de segmentos dba de canal (channel dba number of segments)	§ 5.4.3.54
deltoffst	Desplazamiento dba del canal (channel dba offset)	§ 5.4.3.55
dialnorm	Palabra de normalización de diálogo (dialog normalization word)	§ 5.4.2.8
dialnorm2	Palabra de normalización de diálogo, ch2 (dialog normalization word, ch2)	§ 5.4.2.16
dithflag	Bandera adición de ruido (dither flag)	§ 5.4.3.2
dsurmod	Modo ambiente Dolby (Dolby surround mode)	§ 5.4.2.6
dynrng	Palabra ganancia gama dinámica (dynamic range gain word)	§ 5.4.3.4
dynrng2	Palabra ganancia gama dinámica, ch2 (dynamic range gain word, ch2)	§ 5.4.3.6
dynrnge	Existe palabra ganancia gama dinámica (dynamic range gain word exists)	§ 5.4.3.3
dynrng2e	Existe palabra ganancia gama dinámica, ch2 (dynamic range gain word exists, ch2)	§ 5.4.3.5
exps	Exponentes de canal (channel exponents)	§ 5.4.3.27

Abreviatura	Término	Referencia
fbw	Anchura de banda total (full bandwidth)	
fdcycod	Código decremento rápido (fast decay code)	§ 5.4.3.32
fgaincod	Código de ganancia rápida de canal (channel fast gain code)	§ 5.4.3.41
floorcod	Código de umbral de enmascaramiento (masking floor code)	§ 5.4.3.35
floortab	Cuadro de umbral de enmascaramiento (masking floor table)	§ 7.2.2.7
frmsizecod	Código de tamaño de trama (frame size code)	§ 5.4.1.4
fscod	Código de frecuencia de muestreo (sampling frequency code)	§ 5.4.1.3
fsnroffst	Desplazamiento snr fino de canal (channel fine snr offset)	§ 5.4.3.40
gainrng	Código de gama de ganancia de canal (channel gain range code)	§ 5.4.3.28
grp	Grupo en índice [grp] (group in index [grp])	
langcod	Código de lenguaje (language code)	§ 5.4.2.12
langcod2	Código de lenguaje, ch2 (language code, ch2)	§ 5.4.2.20
langcode	Existe código de lenguaje (language code exists)	§ 5.4.2.11
langcod2e	Existe código de lenguaje, ch2 (language code exists, ch2)	§ 5.4.2.19
lfe	Efectos de baja frecuencia (low frequency effects)	
lfeexps	Exponentes lfe (lfe exponents)	§ 5.4.3.29
lfeexpstr	Estrategia exponente lfe (lfe exponent strategy)	§ 5.4.3.23
lfefgaincod	Código de ganancia rápida lfe (lfe fast gain code)	§ 5.4.3.43
lfefsnroffst	Desplazamiento snr fino lfe (lfe fine snr offset)	§ 5.4.3.42
lfemant	Mantisas lfe (lfe mantissas)	§ 5.4.3.63
lfeon	lfe activado (lfe on)	§ 5.4.2.7
mixlevel	Nivel de mezcla (mixing level)	§ 5.4.2.14
mixlevel2	Nivel de mezcla, ch2 (mixing level, ch2)	§ 5.4.2.22
mstrcplco	Coordenada de acoplamiento principal (master coupling coordinate)	§ 5.4.3.15
nauxbits	Número de bits auxiliares (number of auxiliary bits)	§ 5.4.4.1
nchans	Número de canales (number of channels)	§ 5.4.2.3
nchgrps	Número de grupos de exponentes del canal fbw (number of fbw channel exponent groups)	§ 5.4.3.27
nchmant	Número de mantisas del canal fbw (number of fbw channel mantissas)	§ 5.4.3.61
ncplbnd	Número de bandas acopladas estructuradas (number of structured coupled bands)	§ 5.4.3.13
ncplgrps	Número de grupos de exponentes acoplados (number of coupled exponent groups)	§ 5.4.3.26
ncplmant	Número de mantisas acopladas (number of coupled mantissas)	§ 5.4.3.62
ncplsubnd	Número de sub-bandas acopladas (number of coupling subbands)	§ 5.4.3.12
nfchans	Número de canales fbw (number of fbw channels)	§ 5.4.2.3
nlfegrps	Número de grupos de exponentes del canal lfe (number of lfe channel exponent groups)	§ 5.4.3.29
nlfemant	Número de mantisas del canal lfe (number of lfe channel mantissas)	§ 5.4.3.63
origbs	Tren de bits original (original bit stream)	§ 5.4.2.25
phsflg	Bandera de fase (phase flag)	§ 5.4.3.18
phsflginu	Banderas de fase en uso (phase flags in use)	§ 5.4.3.10
rbnd	Banda de rematrización en el índice [rbnd] (rematrix band in index [rbnd])	

Abreviatura	Término	Referencia
rematflg	Bandera de rematrización (rematrix flag)	§ 5.4.3.20
rematstr	Estrategia de rematrización (rematrixing strategy)	§ 5.4.3.19
roomtyp	Tipo de sala (room type)	§ 5.4.2.15
roomtyp2	Tipo de sala, ch2 (room type, ch2)	§ 5.4.2.23
sbnd	Sub-banda en el índice [sbnd] (subband in index[sbnd])	
sdcycod	Código de decremento lento (slow decay code)	§ 5.4.3.31
seg	Segmento en el índice [seg] (segment in index [seg])	
sgaincod	Código de ganancia lenta (slow gain code)	§ 5.4.3.33
skipfld	Campo de salto (skip field)	§ 5.4.3.60
skipl	Longitud de salto (skip length)	§ 5.4.3.59
skiple	Existe longitud de salto (skip length exists)	§ 5.4.3.58
slev	Coeficiente de nivel de mezcla de entorno (surround mixing level coefficient)	§ 5.4.2.5
snroffste	Existe desplazamiento SNR (SNR offset exists)	§ 5.4.3.36
surmixlev	Nivel de mezcla ambiente (surround mix level)	§ 5.4.2.5
syncframe	Trama de sincronización (synchronization frame)	§ 5.1
syncinfo	Información de sincronización (synchronization information)	§ 5.3.1
syncword	Palabra de sincronización (synchronization word)	§ 5.4.1.1
tdac	Cancelación de solapamiento por división temporal (time division aliasing cancellation)	
timecod1	Primera mitad del código temporal (time code first half)	§ 5.4.2.27
timecod2	Segunda mitad del código temporal (time code second half)	§ 5.4.2.28
timecod1e	Existe primera mitad del código temporal (time code first half exists)	§ 5.4.2.26
timecod2e	Existe segunda mitad del código temporal (time code second half exists)	§ 5.4.2.26

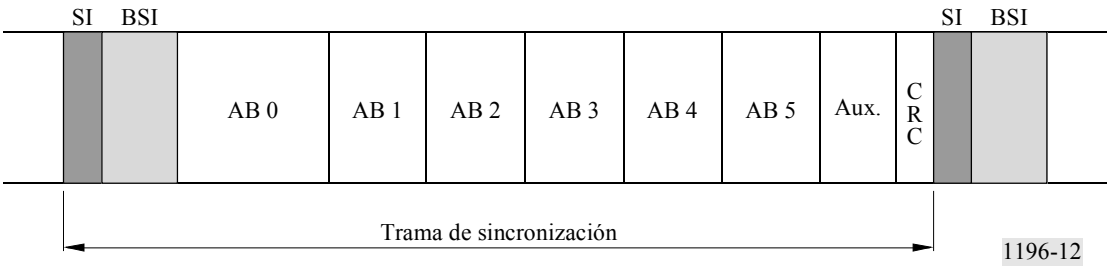
5 Sintaxis del tren de bits

5.1 Trama de sincronización

Un tren de bits de audio codificados en serie con AC-3 está constituido por una sucesión de tramas de sincronización (véase la Fig. 12). Cada trama de sincronización contiene 6 bloques de audio codificados (AB), cada uno de los cuales representa 256 muestras de audio nuevas. La cabecera de información de sincronización (SI) situada al principio de cada trama contiene la información necesaria para la adquisición y el mantenimiento de la sincronización. A la cabecera SI le sigue otra cabecera de información de flujo de bits (BSI) que contiene parámetros descriptivos del servicio de audio codificado. Los bloques de audio codificados pueden ir seguidos de un campo de datos auxiliares (Aux.). Al final de cada trama hay un campo de verificación de errores que incluye una palabra de redundancia (CRC) para la detección de los errores. En la cabecera SI hay una CRC adicional cuyo empleo es facultativo.

FIGURA 12

Trama de sincronización AC-3



5.2 Semántica de la especificación de sintaxis

En el pseudocódigo que sigue se describe el orden de llegada de la información dentro del tren de bits. Tal pseudocódigo se basa aproximadamente en la sintaxis del lenguaje C, si bien se ha simplificado para facilitar su lectura. Para aquellos elementos del tren de bits de longitud superior a un bit, el orden de los bits en el tren de bits en serie es empezando por el bit más significativo (en caso de valores numéricos) o por el bit situado en el extremo izquierdo (para valores de campos de bits). Los campos o elementos contenidos en el tren de bits se representan en **negrita**. Los elementos sintácticos se distinguen tipográficamente mediante la utilización de un tipo de letra diferente (por ejemplo, dynrng).

Algunos elementos del tren de bits de AC-3 se estructuran formando filas. En esta especificación de sintaxis se trata a todos los elementos del tren de bits de forma individual, tanto si están incluidos de una manera natural en filas como si no lo están. Por consiguiente, las filas se describen como elementos múltiples (como en blksw[ch] en oposición a blksw o blksw[]) empleándose estructuras de control tales como los bucles *for* para incrementar el índice ([ch] para canal en este ejemplo).

5.3 Especificación de la sintaxis

Un flujo de bits de audio continuos constará de una sucesión de tramas de sincronización:

Sintaxis

AC-3_bitstream()
{
  while(true)
  {
  syncframe() ;
  }
} /* fin del flujo binario AC-3 */

La trama de sincronización (syncframe) está constituida por los campos **syncinfo** y **bsi**, los 6 campos **audblk**, el campo **auxdata** y el campo **errorcheck**.

Sintaxis

syncframe()
{
  syncinfo() ;
  bsi() ;
  for(blk = 0; blk < 6; blk++)
  {
  audblk() ;
  }
  auxdata() ;
  errorcheck() ;
} /* fin de trama de sincronización */

Cada uno de los elementos del tren de bits y sus longitudes se indican en el siguiente pseudocódigo. Obsérvese que todos los elementos del tren de bits llegan temporalmente en el siguiente orden, primero el bit más significativo o primero el bit de la izquierda.

5.3.1 Información de sincronización – syncinfo

Sintaxis	Tamaño de la palabra
syncinfo()	
{	
syncword	16
crc1	16
fscod	2
frmsizecod	6
} /* fin de información de sincronización */	

5.3.2 Información del tren de bits – bsi

Sintaxis	Tamaño de la palabra
bsi()	
{	
bsid	5
bsmod	3
acmod	3
if((acmod & 0x1) && (acmod != 0x1)) /* si hay 3 canales frontales */ **{cmixlev}**	2
if(acmod & 0x4) /* si existe un canal ambiente */ **{surmixlev}**	2
if(acmod == 0x2) /* si se trata del modo 2/0 activo */ **{dsurmod}**	2
lfeon	1
dialnorm	5
compre	1
if(compre) **{compr}**	8
langcode	1
if(langcode) **{langcod}**	8
audprodie	1
if(audprodie)	
{	
mixlevel	5
roomtyp	2
}	
if(acmod == 0) /* si se trata del modo 1+1 (dual, mono, de forma que algunos elementos necesitan un segundo valor) */	
{	
dialnorm2	5
compr2e	1
if(compr2e) **{compr2}**	8
lngcod2e	1
if(langcod2e) **{langcod2}**	8
audprodi2e	1
if(audprodi2e)	
{	
mixlevel2	5
roomtyp2	2
}	
}	
copyrightb	1
origbs	1
timecod1e	1
if(timecod1e) **{timecod1}**	14
timecod2e	1
if(timecod2e) **{timecod2}**	14
addbsie	1
if(addbsie)	
{	
addbsil	6
addbsi	(addbsil+1)×8
}	
} /* fin de información del tren de bits */	

5.3.3 Bloque de audio – audblk

Sintaxis	Tamaño de la palabra
audblk()	
{	
/* Campos de banderas de conmutación de bloque y adición de ruido */	
for(ch = 0; ch < nfchans; ch++) **{blksw[ch]}**	1
for(ch = 0; ch < nfchans; ch++) **{dithflag[ch]}**	1
/* Campos de control de gama dinámica */	
dynrnge	1
if(dynrnge) **{dynrng}**	8
if(acmod == 0) /* si modo 1+1 */	
{	
dynrng2e	1
if(dynrng2e) **{dynrng2}**	8
}	
/* Campos de información estrategia de acoplamiento */	
cplstre	1
if(cplstre)	
{	
cplinu	1
if(cplinu)	
{	
for(ch = 0; ch < nfchans; ch++) **{chincpl[ch]}**	1
if(acmod == 0×2) **{phsflginu}** /* if in 2/0 mode */	1
cplbegf	4
cplendf	4
/* ncplsubnd = 3 + cplendf – cplbegf */	
for(bnd = 1; bnd < ncplsubnd; bnd++) **{cplbndstrc[bnd]}**	1
}	
}	
/* Campos de coordenadas de acoplamiento, banderas de fase */	
if(cplinu)	
{	
for(ch = 0; ch < nfchans; ch++)	
{	
if(chincpl[ch])	
{	
cplcoe[ch]	1
if(cplcoe[ch])	
{	
mstrcplco[ch]	2
/* ncplbnd se obtiene a partir de ncplsubnd, y cplbndstrc */	
for(bnd = 0; bnd < ncplbnd; bnd++)	
{	
cplcoexp[ch][bnd]	4
cplcomant[ch][bnd]	4
}	
}	
}	
}	
if((acmod == 0×2) && phsflginu && (cplcoe[0] || cplcoe[1]))	
{	
for(bnd = 0; bnd < ncplbnd; bnd+ +) **{phsflg[bnd]}**	1
}	
}	

Sintaxis	Tamaño de la palabra
/* Campos de operación rematrización en el modo 2/0 */	
if(acmod == 0x2) /* si en el modo 2/0 */	
{	
rematstr	1
if(rematstr)	
{	
if((cplbegf > 2) || (cplinu == 0))	
{	
for(rbnd = 0; rbnd < 4; rbnd++) **{rematflg[rbnd]}**	1
}	
if((2 ≥ cplbegf > 0) && cplinu)	
{	
for(rbnd = 0; rbnd < 3; rbnd++) **{rematflg[rbnd]}**	1
}	
if((cplbegf == 0) && cplinu)	
{	
for(rbnd = 0; rbnd < 2; rbnd++) **{rematflg[rbnd]}**	1
}	
}	
}	
/* Campos de estrategia de exponente */	
if(cplinu) **{cplexpstr}**	2
for(ch = 0; ch < nfchans; ch++) **{chexpstr[ch]}**	2
if(lfeon) **{lfeexpstr}**	1
for(ch = 0; ch < nfchans; ch++)	
{	
if(chexpstr[ch] != reuse)	
{	
if(!chincpl[ch]) **{chbwcod[ch]}**	6
}	
}	
/* Campos de exponentes */	
if(cplinu) /* exponentes para el canal de acoplo */	
{	
if(cplexpstr != reuse)	
{	
cplabsexp	4
/* ncplgrps se obtiene a partir de ncplsubnd, y cplexpstr */	
for(grp = 0; grp < ncplgrps; grp++) **{cplexps[grp]}**	7
}	
}	
for(ch = 0; ch < nfchans; ch++) /* exponentes para los canales de anchura de banda total */	
{	
if(chexpstr[ch] != reuse)	
{	
exps[ch][0]	4
/* nchgrps se obtiene a partir de chexpstr[ch], y cplbegf o chbwcod[ch] */	
for(grp = 1; grp <= nchgrps[ch]; grp++) **{exps[ch][grp]}**	7
gainrng[ch]	2
}	
}	
if(lfeon) /* exponentes para el canal de efectos de baja frecuencia */	
{	
if(lfeexpstr != reuse)	
{	
lfeexps[0]	4
/* nlfegrps = 2 */	
for(grp = 1; grp <= nlfegrps; grp++) **{lfeexps[grp]}**	7
}	
}	

Sintaxis	Tamaño de la palabra
/* Campos de información paramétrica de atribución de bits */	
baie	1
if(baie)	
{	
sdcycod	2
fdcycod	2
sgaincod	2
dbpbcod	2
floorcod	3
}	
snroffste	1
if(snroffste)	
{	
csnroffst	6
if(cplinu)	
{	
cplfsnroffst	4
cplfgaincod	3
}	
for(ch = 0; ch < nfchans; ch++)	
{	
fsnroffst[ch]	4
fgaincod[ch]	3
}	
if(lfeon)	
{	
lfefsnroffst	4
lfefgaincod	3
}	
}	
if(cplinu)	
{	
cplleake	1
if(cplleake)	
{	
cplfleak	3
cplsleak	3
}	
}	
/* Campos de información sobre atribución de bit delta */	
deltbaie	1
if(deltbaie)	
{	
if(cplinu) **{cpldeltbae}**	2
for(ch = 0; ch < nfchans; ch++) **{deltbae[ch]}**	2
if(cplinu)	
{	
if(cpldeltbae==new info follows)	
{	
cpldeltnseg	3
for(seg = 0; seg <= cpldeltnseg; seg++)	
{	
cpldeltoffst[seg]	5
cpldeltlen[seg]	4
cpldeltba[seg]	3
}	
}	
}	
for(ch = 0; ch < nfchans; ch++)	

Sintaxis	Tamaño de la palabra
{	
if(deltbae[ch]==new info follows)	
{	
deltnseg[ch]	3
for(seg = 0; seg <= deltnseg[ch]; seg++)	
{	
deltoffst[ch][seg]	5
deltlen[ch][seg]	4
deltba[ch][seg]	3
}	
}	
}	
}	
/* Campos para inclusión de datos ficticios sin utilizar */	
skiple	1
if(skiple)	
{	
skipl	9
skipfld	skipl × 8
}	
/* Campos de valores de mantisa cuantificados */	
ch = 0	
do /* mantisas de los canales que van hasta el primer canal acoplado, inclusive */	
{	
for(bin = 0; bin < nchmant[ch]; bin++) **{chmant[ch][bin]}**	(0-16)
ch += 1	
}	
while(chinclp[ch] = = 0 && ch < nfchans)	
if(cplinu) /* mantisas del canal de acoplo */	
{	
for(bin = 0; bin < ncplmant; bin++) **{cplmant[bin]}**	(0-16)
}	
while(ch<nfchans) /* mantisas de los canales restantes, acoplados o no */	
{	
for(bin = 0; bin < nchmant[ch]; bin++) **{chmant[ch][bin]}**	(0-16)
ch += 1	
}	
if(lfeon) /* mantisas del canal de efectos de baja frecuencia */	
{	
for(bin = 0; bin < nlfemant; bin++) **{lfemant[bin]}**	(0-16)
}	
} /* fin de bloque de audio */	

5.3.4 Datos auxiliares – auxdata

Sintaxis	Tamaño de la palabra
auxdata()	
{	
auxbits	nauxbits
if(auxdatae)	
{	
auxdatal	14
}	
auxdatae	1
} /* fin de datos auxiliares */	

5.3.5 Código detector de errores – errorcheck

Sintaxis	Tamaño de la palabra
errorcheck()	
{	
crcrsv	1
crc2	16
} /* fin de comprobación de errores */	

5.4 Descripción de los elementos del tren de bits

Algunos elementos del tren de bits tienen valores que deben transmitirse aunque su significado sea reservado. Si un decodificador recibe un tren de bits que contiene valores reservados puede o no ser capaz de decodificarlo y producir una salida de audio. En la descripción de los elementos del tren de bits que tienen códigos reservados existe una indicación acerca de lo que puede hacer el decodificador si recibe el código reservado. En algunos casos, el decodificador no puede decodificar la señal de audio. En otros, el decodificador puede decodificar el audio utilizando un valor por defecto para el parámetro indicado por un código reservado.

5.4.1 Información de sincronización – syncinfo

5.4.1.1 syncword – Palabra de sincronización – 16 bits

La syncword es siempre 0×0B77 ó 0000 1011 0111 0111. La transmisión de syncword, como la de otros elementos del campo de bits se inicia por el bit de la izquierda.

5.4.1.2 crc1 – Verificación de redundancia cíclica 1 – 16 bits

Esta CRC de 16 bits se aplica a los primeros 5/8 de la trama. La transmisión de la CRC, como la de otros valores numéricos se inicia por el bit más significativo.

5.4.1.3 fscod – Código de velocidad de muestreo – 2 bits

Se trata de un código de 2 bits que indica la velocidad de muestreo de conformidad con el Cuadro 1. Si se ha señalado el código reservado, el decodificador no intentará decodificar el audio y quedará enmudecido.

CUADRO 1

Códigos de velocidad de muestreo

fscod	**Velocidad de muestreo (kHz)**
00	48
01	44,1
10	32
11	Reservado

5.4.1.4 frmsizecod – Código de tamaño de trama – 6 bits

El código de tamaño de trama se usa junto con el código de velocidad de muestreo para determinar el número de palabras (de 2 bytes) anteriores a la siguiente syncword (véase el Cuadro 13).

5.4.2 bsi – Información del tren de bits

5.4.2.1 bsid – Identificación del tren de bits – 5 bits

Este campo de bits tiene un valor igual 01000 (= 8) en la presente versión de esta norma. En futuras modificaciones de esta norma podrán definirse otros valores. Valores de bsid inferiores a 8 se utilizarán en versiones de AC-3 que realicen subconjuntos de la versión de sintaxis 8. Los decodificadores que puedan decodificar la versión 8 serán también capaces de decodificar números de versiones inferiores a 8. De ampliarse esta norma con la agregación de elementos o características adicionales se utilizará un bsid superior a 8. Los decodificadores fabricados con arreglo a la presente versión de la norma no podrán decodificar versiones con bsid superior a 8. En consecuencia, tales decodificadores deberán permanecer en silencio si el valor del bsid es superior a 8 y deberán decodificar y reproducir la señal audio si el valor del bsid es menor o igual que 8.

5.4.2.2 bsmod – Modo tren de bits – 3 bits

Este código de 3 bits indica el tipo de servicio transportado por el tren de bits como se define en el Cuadro 2.

CUADRO 2

Modo tren de bits

bsmod	acmod	**Tipo de servicio**
000	Cualquiera	Servicio de audio principal: principal completo (CM)
001	Cualquiera	Servicio de audio principal: música y efectos (ME)
010	Cualquiera	Servicio asociado: degradado visulamente (VI)
011	Cualquiera	Servicio asociado: degradado auditivamente (HI)
100	Cualquiera	Servicio asociado: diálogo (D)
101	Cualquiera	Servicio asociado: comentarios (C)
110	Cualquiera	Servicio asociado: emergencia (E)
111	‘001’	Servicio asociado: sobrevoz (VO)
111	‘010’-‘111’	Servicio de audio principal: karaoke

5.4.2.3 acmod – Modo codificación audio – 3 bits

Este código de 3 bits, representado en el Cuadro 3, indica cuáles de los canales de servicio principal están en uso variando de 3/2 a 1/0. Si el MSB de acmod es un 1, se utilizan los canales ambiente y van seguidos de surmixlev en el tren de bits. Si el MSB de acmod es un 0 no se usan los canales ambiente, por lo que en el tren de bits no sigue a continuación surmixlev. Si el LSB de acmod es un 0 no está en uso el canal central. Si el LSB de acmod es un 1 se utiliza el canal central. Nótese que el estado de acmod fija el número del parámetro de canales de anchura de banda total nfchans (por ejemplo, para el modo 3/2, nfchans = 5; para el modo 2/1, nfchans = 3; etc.). El número de

canales total, nchans es igual a nfchans si el canal lfe está desactivado (véase el preámbulo) y es igual a 1 + nfchans si el canal lfe está activado. Si acmod es cero se codifican en el flujo de bits 2 canales de programa independientes completamente (mono dual) denominados Ch1, Ch2. En este caso, en BSI o audblk están presentes algunos elementos adicionales, a fin de describir completamente Ch2. En el Cuadro 3 se indica asimismo la ordenación de los canales (orden en el cual se procesan los canales) para cada uno de los modos.

CUADRO 3

Modo codificación de audio

acmod	**Modo codificación audio**	nfchans	**Ordenación de la fila de canales**
000	1 + 1	2	Canal 1, canal 2
001	1/0	1	C
010	2/0	2	L, R
011	3/0	3	L, C, R
100	2/1	3	L, R, S
101	3/1	4	L, C, R, S
110	2/2	4	L, R, SL, SR
111	3/2	5	L, C, R, SL, SR

5.4.2.4 cmixlev – Nivel de mezcla central – 2 bits

Cuando se emplean tres canales frontales, este código de 2 bits mostrado en el Cuadro 4, indica el nivel de submezclado nominal del canal central con respecto a los canales izquierdo y derecho. Si cmixlev se fija a código reservado, los decodificadores deberán sin embargo reproducir la señal de audio. En este caso puede utilizarse el valor intermedio de cmixlev (–4,5 dB).

CUADRO 4

Nivel de mezcla central

cmixlev	clev
00	0,707 (–3,0 dB)
01	0,596 (–4,5 dB)
10	0,500 (–6,0 dB)
11	Reservado

5.4.2.5 surmixlev – Nivel de mezcla de ambiente – 2 bits

Si se utilizan canales ambiente, este código de 2 bits representado en el Cuadro 5, indica el nivel de submezclado nominal de los canales ambiente. Si surmixlev se fija a un código reservado, el decodificador deberá, no obstante, reproducir la señal audio. Puede utilizarse en este caso el valor intermedio de surmixlev (–6 dB).

CUADRO 5

Nivel mezcla de ambiente

surmixlev	slev
00	0,707 (–3 dB)
01	0,500 (–6 dB)
10	0
11	Reservado

5.4.2.6 dsurmod – Modo ambiente Dolby – 2 bits

En funcionamiento en el modo de dos canales, este código de 2 bits indica, como se muestra en el Cuadro 6, si el programa se ha codificado con ambiente Dolby o no. El decodificador AC-3 no utiliza esta información, pero sí la pueden emplear otras partes del equipo de reproducción de audio. Si dsurmod se fija a un código reservado, el decodificador deberá aún reproducir la señal de audio. El código reservado puede interpretarse como «no indicado».

CUADRO 6

Modo ambiente Dolby

dsurmod	**Indicación**
00	No indicado
01	Ambiente NO codificado con Dolby
10	Ambiente codificado con Dolby
11	Reservado

5.4.2.7 lfeon – Canal de efectos de baja frecuencia activado – 1 bit

Este bit tiene un valor igual a 1 si está activado el canal lfe (graves) y toma un valor igual a 0 si está desactivado el canal lfe.

5.4.2.8 dialnorm – Normalización de diálogo – 5 bits

Este código de 5 bits indica en qué grado el nivel de diálogo medio está por debajo del valor 100% digital. Los valores válidos están comprendidos en la gama 1-31. El valor 0 es reservado. Los valores de 1 a 31 se interpretan como –1 dB a –31 dB con respecto a 100% digital. Si se recibe el valor reservado 0, el decodificador utilizará –31 dB. El valor de dialnorm afectará al nivel de reproducción sonora. Si el valor no se utiliza en el propio decodificador AC-3 puede emplearse en otras partes del equipo de reproducción de audio. En el § 7.6 se explica ulteriormente la normalización del diálogo.

5.4.2.9 compre – Existe palabra de ganancia de compresión – 1 bit

Si este bit es igual a 1, los 8 bits siguientes representan una palabra de control de compresión.

5.4.2.10 compr – Palabra de ganancia de compresión – 8 bits

Esta palabra de ganancia generada en el codificador puede estar presente en el flujo de bits. De ser así, puede emplearse para graduar el nivel de audio reproducido a fin de reproducir una gama dinámica muy estrecha con un valor garantizado para el límite superior del nivel de cresta instantáneo de la señal reproducida en el submezclador monofónico. El significado y la utilización de compr se describen ulteriormente en el § 7.7.2.

5.4.2.11 langcode – Existe código de lenguaje – 1 bit

Si este bit es un 1, los 8 bits siguientes representan un código de lenguaje. Si este bit es un 0 el lenguaje del servicio de audio no está indicado.

5.4.2.12 langcod – Código de lenguaje – 8 bits

Se trata de un código de 8 bits que representa el lenguaje del servicio de audio. La correspondencia entre langcod se indica en el Cuadro 14.

5.4.2.13 audprodie – Existe información de producción de audio – 1 bit

Si este bit es un 1, existen los campos mixlevel y roomtyp, conteniendo información sobre el ambiente de producción de audio (sala de mezclas).

5.4.2.14 mixlevel – Nivel de mezcla – 5 bits

Este código de 5 bits indica el nivel de presión sonora acústica absoluta de un canal individual durante la sesión final de mezcla de audio. El código de 5 bits representa un valor en la gama de 0 a 31. El nivel de mezcla máximo es 80 más el valor de mixlevel dB SPL, u 80 a 111 dB SPL. El nivel de mezcla máximo es el nivel acústico de una onda sinusoidal en un canal cuyas crestas alcanzan el 100% en la representación MIC. Generalmente el valor SPL absoluto se mide mediante ruido rosado con un valor RMS de –20 o –30 dB con respecto al nivel RMS máximo de la onda sinusoidal. El valor de mixlevel no se suele utilizar dentro del decodificador AC-3, pero puede ser empleado por otras partes del equipo de reproducción de audio.

5.4.2.15 roomtyp – Tipo de sala – 2 bits

Este código de 2 bits mostrado en el Cuadro 7, indica el tipo y la calibración de la sala de mezclas utilizada en la sesión de mezcla de audio final. Generalmente, en el decodificador AC-3 no se emplea el valor de roomtyp si bien puede utilizarse en otras partes del equipo de reproducción de audio. Si roomtyp se fija al código reservado, el decodificador deberá reproducir la señal de audio. El código reservado puede interpretarse como «no indicado».

CUADRO 7

Tipo de sala

roomtyp	**Tipo de sala de mezclas**
00	No indicado
01	Sala amplia, monitor de curva X
10	Sala pequeña, monitor plano
11	Reservado

5.4.2.16 dialnorm2 – Normalización de diálogo, Ch2 – 5 bits

Este código de 5 bits tiene el mismo significado que dialnorm, salvo que se aplica al segundo canal de audio cuando acmod indica dos canales independientes (modo 1 + 1 mono dual).

5.4.2.17 compr2e – Existe palabra de ganancia de compresión, Ch2 – 1 bit

Si este bit es un 1, los 8 bits siguientes representan la palabra de ganancia de compresión de Ch2.

5.4.2.18 compr2 – Palabra de ganancia de compresión, Ch2 – 8 bits

Esta palabra de 8 bits tiene el mismo significado que compr, salvo que se aplica al segundo canal de audio cuando acmod indica dos canales independientes (modo 1 + 1 mono dual).

5.4.2.19 langcod2e – Existe código de lenguaje, Ch2 – 1 bit

Si este bit es un 1, los 8 bits siguientes representan un código de lenguaje para Ch2. Si este bit es un 0, el lenguaje de Ch2 no está indicado.

5.4.2.20 langcod2 – Código de lenguaje, Ch2 – 8 bits

Este código de 8 bits tiene el mismo significado que langcod, salvo que se aplica al segundo canal de audio cuando acmod indica dos canales independientes (modo 1 + 1, mono dual).

5.4.2.21 audprodi2e – Existe información de producción audio, Ch2 – 1 bit

Si este bit es 1 existen los dos campos de datos siguientes que proporcionan información sobre la producción de audio para Ch2.

5.4.2.22 mixlevel2 – Nivel de mezcla, Ch2 – 5 bits

Este código de 5 bits tiene el mismo significado que mixlevel, salvo que se aplica al segundo canal de audio cuando acmod indica dos canales independientes (modo 1 + 1 mono dual).

5.4.2.23 roomtyp2 – Tipo de sala, Ch2 – 2 bits

Este código de 2 bits tiene el mismo significado que roomtyp, salvo que se aplica al segundo canal de audio cuando acmod indica dos canales independientes (modo 1 + 1 mono dual).

5.4.2.24 copyrightb – Bit de derechos de autor – 1 bit

Si este bit tiene un valor igual a 1, ello indica que la información del tren de bits está protegida por derechos de autor. Si tiene un valor igual a 0, indica que la información no está protegida.

5.4.2.25 origbs – Tren de bits original – 1 bit

Este bit tiene el valor 1 si se trata del tren de bits original. Este bit tiene el valor 0 si se trata de una copia de otro tren de bits.

5.4.2.26 timecod1e, timecod2e – Existen mitades (1ª y 2ª) del código temporal – 2 bits

Como se muestra en el Cuadro 8, estos valores indican si aparecen los códigos temporales en el tren de bits. El código temporal puede tener una resolución de 1/64 de una trama (1 trama = 1/30 de un segundo). Como para la sincronización fina únicamente es necesaria la porción de alta resolución del código temporal, se divide el código temporal de 28 bits en dos mitades de 14 bits. La primera mitad de baja resolución representa el código en incrementos de 8 s hasta 24 h. La segunda mitad de alta resolución representa el código en incrementos de 1/64 de trama hasta 8 s.

CUADRO 8

Existe código temporal

timecod2e, timecod1e	**Código temporal presente**
0,0	No presente
0,1	Primera trama (14 bits) presente
1,0	Segunda trama (14 bits) presente
1,1	Ambas mitades (28 bits) presentes

5.4.2.27 timecod1 – Primera mitad del código temporal – 14 bits

Los primeros 5 bits de este campo de 14 bits representan el tiempo en horas, con valores válidos comprendidos entre 0 y 23. Los siguientes 6 bits representan el tiempo en minutos, con valores comprendidos entre 0 y 59. Los últimos 3 bits representan el tiempo en incrementos de 8 s, con valores comprendidos entre 0 y 7 (representativos de 0, 8, 16, ... 56 s).

5.4.2.28 timecod2 – Segunda mitad del código temporal – 14 bits

Los primeros 3 bits de este campo de 14 bits representan el tiempo en segundos, con valores comprendidos entre 0 y 7 (representativos de 0-7 s). Los 5 bits siguientes representan el tiempo en tramas, con valores de 0 a 29. Los últimos 6 bits representan fracciones de 1/64 de una trama, con valores comprendidos entre 0 y 63.

5.4.2.29 addbsie – Existe información del flujo de bits adicional – 1 bit

Si este bit tiene el valor 1 hay información del flujo de bits adicional cuya longitud viene indicada por el siguiente campo. Si este bit tiene un valor 0, no hay información del flujo de bits adicional.

5.4.2.30 addbsil – Longitud de la información del flujo de bits adicional – 6 bits

Este código de 6 bits, que existe únicamente si addbsie es un 1, indica la longitud en octetos de la información del flujo de bits adicional. La gama válida de valores de addbsil es 0-63, indicando de 1 a 64 bits adicionales, respectivamente. No se exige que el decodificador interprete esta información, por lo que puede ignorar este número de octetos que siguen en el tren de datos.

5.4.2.31 addbsi – Información del tren de bits adicional – ((addbsil+1) × 8) bits

Este campo contiene de 1 a 64 octetos de cualquier información adicional incluida con la estructura de información del tren de bits.

5.4.3 audblk – Bloque de audio

5.4.3.1 blksw[ch] – Bandera de conmutación de bloque – 1 bit

Esta bandera, para el canal [ch], indica si el bloque de audio en curso se ha divido en dos sub-bloques durante la transformación del dominio del tiempo al dominio de la frecuencia. El valor 0 indica que el bloque no se ha dividido y que se ha realizado una transformación TDAC de 512 puntos. El valor 1 indica que el bloque se ha dividido en dos sub-bloques de longitud 256, que la longitud de la transformación TDAC ha pasado de 512 puntos a 256 puntos y que se han realizado las 2 transformaciones en el bloque de audio (una en cada sub-bloque). En el § 7.9 se describe con más detalle la conmutación de la longitud de la transformación.

5.4.3.2 dithflag[ch] – Bandera adición de ruido – 1 bit

Para el canal [ch] esta bandera expresa que el decodificador debería activar la adición de ruido en el bloque vigente. En el § 7.3.4 se describe con detalle la adición de ruido.

5.4.3.3 dynrnge – Existe palabra de ganancia de gama dinámica – 1 bit

Si este bit es un 1, la palabra de ganancia de gama dinámica aparece seguidamente en el tren de bits. Si es un 0, no está presente la palabra de ganancia reutilizándose el valor anterior, salvo en el caso del bloque 0 de una trama en el que si no está presente la palabra de control el valor vigente de dynrng se pone a 0.

5.4.3.4 dynrng – Palabra de ganancia de gama dinámica – 8 bits

Esta palabra de ganancia generada en el codificador se aplica para graduar el nivel de audio reproducido como se describe en el § 7.7.1.

5.4.3.5 dynrng2e – Existe palabra de ganancia de gama dinámica, Ch2 – 1 bit

Si este bit es un 1, la palabra de ganancia de gama dinámica para el canal 2 se inserta en el tren de bits. Si es un 0, la palabra no está presente, reutilizándose el valor anterior salvo en el caso del bloque 0 de una trama en el que si no está presente la palabra de control el valor vigente de dynrng2 se pone a 0.

5.4.3.6 dynrng2 – Palabra de ganancia de gama dinámica, Ch2 – 8 bits

Esta palabra de ganancia generada en el codificador se aplica para graduar el nivel del audio reproducido en Ch2 de la misma forma que dynrng se aplica al Ch1, como se describe en el § 7.7.1.

5.4.3.7 cplstre – Existe estrategia de acoplamiento – 1 bit

Si este bit es un 1, se inserta a continuación la información de acoplamiento en el tren de bits. Si es un 0, la información de acoplamiento no está presente, reutilizándose los parámetros de acoplamiento enviados anteriormente.

5.4.3.8 cplinu – Acoplamiento en uso – 1 bit

Si este bit es un 1, se utiliza actualmente el acoplamiento transmitiéndose a continuación los parámetros del acoplamiento. Si es un 0 no se está empleando el acoplamiento (todos los canales son independientes) por lo que en el tren de bits no se incluyen los parámetros de acoplamiento.

5.4.3.9 chincpl[ch] – Canal con acoplamiento – 1 bit

Si este bit es un 1, el canal indicado por el índice [ch] es un canal acoplado. Si el bit es un 0 tal canal no está acoplado. Como en el modo 1/0 no se emplea el acoplamiento, si existen valores de chincpl[] estarán en la gama de 2 a 5. De entre los valores presentes al menos 2 serán iguales a 1 ya que el acoplamiento requiere que exista más de un canal acoplado.

5.4.3.10 phsflginu – Banderas de fase en uso – 1 bit

Si este bit (definido únicamente para el modo 2/0) es un 1, se incluyen las banderas de fase junto con la información de coordenada de acoplamiento. En el § 7.4 se describen las banderas de acoplamiento.

5.4.3.11 cplbegf – Código de frecuencia de comienzo de acoplamiento – 4 bits

Este código de 4 bits se interpreta como el número de sub-banda (0 a 15) que indica el extremo de la banda de frecuencias más bajas del canal acoplado (o la primera sub-banda activa), como se muestra en el Cuadro 38.

5.4.3.12 cplendf – Código de frecuencia de fin de acoplamiento – 4 bits

Este código de 4 bits indica el extremo de la banda superior del canal de acoplamiento. Tal extremo (o la última sub-banda activa) es cplendf+2 ó un valor comprendido entre 2 y 17 (véase el Cuadro 38).

El número de sub-bandas de acoplamiento activas es igual a ncplsubnd, que se calcula como sigue:

ncplsubnd = 3 + cplendf – cplbegf;

5.4.3.13 cplbnstrc[sbnd] – Estructura de la banda de acoplamiento – 1 bit

En el Cuadro 38 se definen 18 sub-bandas de acoplamiento cada una de las cuales contiene 12 coeficientes de frecuencia. Las sub-bandas de acoplamiento fijas con una anchura de 12 bin se convierten en bandas de acoplamiento cada una de las cuales puede tener una anchura mayor (un múltiplo de) 12 bins de frecuencia. Cada banda de acoplamiento puede contener una o más sub-bandas de acoplamiento. Para cada banda de acoplamiento se transmiten las coordenadas de acoplamiento. Cada coordenada de acoplamiento de las bandas debe aplicarse a la totalidad de los coeficientes de la banda de acoplamiento.

La estructura de la banda de acoplamiento indica qué sub-bandas de acoplamiento se combinan en bandas de acoplamiento más anchas. Cuando cplbndstrc[sbnd] es un 0 el número de sub-banda [sbnd] no se coordina con la banda anterior para formar una banda más ancha, sino que inicia una nueva banda de acoplamiento de anchura 12. Cuando cplbndstrc[sbnd] es un 1 la sub-banda [sbnd] se combina con la banda anterior haciendo la banda anterior 12 bins más ancha. Cada valor sucesivo de cplbndstrc que sea un 1 continuará combinando sub-bandas en la banda vigente. Cuando se reciba otro valor de cplbndstrc igual a 0, se formará una nueva banda comenzando con los 12 bins de la sub-banda vigente. Típicamente el conjunto de valores de cplbndstrc[sbnd] se estructura en una fila.

Cada bit de la fila se corresponde con una sub-banda de acoplamiento específica en orden de frecuencias creciente. El primer elemento de la fila que corresponde a la sub-banda cplbegf es siempre 0 y no se transmite. (No hay motivo para enviar un 1 bit de cplbndstrc para la primera sub-banda en cplbegf ya que este bit siempre será 0.) Por consiguiente, se transmiten ncplsubnd-1 valores de cplbndstrc. Si únicamente existe una sub-banda de acoplamiento no se envían los bits cplbndstrc.

El número de bandas de acoplamiento ncplbnd, puede calcularse mediante ncplsubnd y cplbnstrc como sigue:

ncplbnd = (ncplsubnd – (cplbndstrc[cplbegf+1] + ... + cplbndstrc[cplendf+2]));

5.4.3.14 cplcoe[ch] – Existen coordenadas de acoplamiento – 1 bit

La coordenadas de acoplamiento indican, para un canal determinado dentro de una banda de acoplamiento dada, la fracción de coeficientes de frecuencia del canal de acoplamiento que ha de utilizarse para regenerar los coeficientes de frecuencia del canal individual. En el tren de bits se transmiten condicionalmente las coordenadas de acoplamiento. Si no se entregan nuevos valores mantienen su vigencia los valores enviados previamente. En el § 7.4 se facilita información adicional sobre el acoplamiento.

Si cplcoe[ch] es 1, existen las coordenadas de acoplamiento para el canal correspondiente [ch] enviándose seguidamente en el flujo de bits. Si el bit es un 0, se reutilizan para este canal las coordenadas de acoplamiento transmitidas anteriormente. Todas las coordenadas de acoplamiento se transmiten siempre en el bloque 0 de cada sync frame.

5.4.3.15 mstrcplco[ch] – Coordenada de acoplamiento principal – 2 bits

Este parámetro propio de cada canal establece un factor de ganancia canal por canal (que aumenta la gama dinámica) para las coordenadas de acoplamiento, como se muestra en el Cuadro 9.

CUADRO 9

Coordenada de acoplamiento principal

mstrcplco[ch]	cplco[ch][bnd] **Multiplicador de ganancia**
00	1
01	2^{-3}
10	2^{-6}
11	2^{-9}

5.4.3.16 cplcoexp[ch][bnd] – Exponente de la coordenada de acoplamiento – 4 bits

Cada coordenada de acoplamiento está constituida por un exponente de 4 bits y una mantisa de 4 bits. Este elemento es el valor del exponente de la coordenada de acoplamiento para el canal [ch] y la banda [bnd]. El índice [ch] existirá únicamente para los canales que estén acoplados. El índice [bnd] variará entre 0 y ncplbnds. En el § 7.4.3 se facilita información adicional acerca de la interpretación de las coordenadas de acoplamiento.

5.4.3.17 cplcomant[ch][bnd] – Mantisa de la coordenada de acoplamiento – 4 bits

Este elemento es la mantisa de la coordenada de acoplamiento de 4 bits para el canal [ch] y la banda [bnd].

5.4.3.18 phsflg[bnd] – Bandera de fase – 1 bit

Este elemento (empleado únicamente en el modo 2/0) indica si el decodificador deberá invertir la fase de las mantisas del canal de acoplamiento cuando reconstruya el canal de salida derecho. El índice [bnd] puede variar entre 0 y ncplbnd. En el § 7.4 se describen las banderas de fase.

5.4.3.19 rematstr – Estrategia de rematrización – 1 bit

Si este bit es un 1, en el flujo de bits están presentes las nuevas banderas de rematrización. Si es un 0, tales banderas no están presentes debiendo reutilizarse los valores anteriores. El parámetro rematstr únicamente está presente en el modo codificación de audio 2/0.

5.4.3.20 rematflg[rbnd] – Bandera de rematrización – 1 bit

Este bit indica si se han rematrizado los coeficientes de la transformación en la banda de rematrización [rbnd]. Si este bit es un 1, los coeficientes de la transformación de [rbnd] se han rematrizado en los canales suma y diferencia. Si este bit es un 0, no se ha efectuado la rematrización en la banda [rbnd]. El número de bandas de rematrización (y el número de valores de [rbnd]) dependen de los parámetros de acoplamiento, como se muestra en el Cuadro 10. La rematrización se describe en el § 7.5.

CUADRO 10

Número de bandas de rematrización

Condición	Número de bandas de rematrización
cplinu == 0	4
(cplinu == 1) && (cplbegf > 2)	4
(cplinu == 1) && (2 ≥ cplbegf > 0)	3
(cplinu == 1) && (cplbegf == 0)	2

5.4.3.21 cplexpstr – Estrategia del exponente de acoplamiento – 2 bits

Este elemento indica el método de codificación de exponentes que se utiliza para el canal de acoplamiento, según se muestra en el Cuadro 18. Para una explicación de cada estrategia de exponente, véase el § 7.1.

5.4.3.22 chexpstr[ch] – Estrategia del exponente del canal – 2 bits

Este elemento indica el método de codificación de exponentes utilizado para canal [ch], como se muestra en el Cuadro 18. Este elemento existe para cada canal de anchura de banda completa.

5.4.3.23 lfeexpstr – Estrategia del exponente del canal de efectos de baja frecuencia – 1 bit

Este elemento indica el método de codificación de exponentes que se utiliza para el canal lfe como se muestra en el Cuadro 19.

5.4.3.24 chbwcod[ch] – Código de anchura de banda de canal – 6 bits

El elemento chwcod[ch] es un entero sin signo que define el extremo de banda superior para el canal de anchura de banda total [ch]. Unicamente se incluye este parámetro para canales fbw no acoplados (para la definición de este parámetro consúltese el § 7.1.3 sobre exponentes). Los valores válidos están en la gama de 0 a 60. Si se recibe un valor mayor que 60 se considera inválido el tren de bits y el decodificador interrumpirá la decodificación de audio y permanecerá silenciado.

5.4.3.25 cplabsexp – Exponente absoluto de acoplamiento – 4 bits

Se trata de un exponente absoluto utilizado como referencia en la decodificación de los exponentes diferenciales para el canal de acoplamiento.

5.4.3.26 cplexps[grp] – Exponentes de acoplamiento – 7 bits

Cada valor de cplexps indica el valor de 3, 6 ó 12 exponentes de canal de acoplamiento codificados diferencialmente para el grupo de exponente de acoplamiento [grp] en el caso de la codificación d15, d25 o d45, respectivamente. El número de valores de cplexps es igual a ncplgrps, que puede determinarse a partir de cplbegf, cplendf y cplexpstr. Para más información consúltese el § 7.1.3.

5.4.3.27 exps[ch][grp] – Exponentes de canal – 4 ó 7 bits

Estos elementos representan los exponentes codificados para el canal [ch]. El primer elemento ([grp] = 0) es un exponente absoluto de 4 bits para el primer coeficiente de la transformación (término DC). Los elementos subsiguientes ([grp] > 0) son las representaciones con 7 bits de un grupo de 3, 6 ó 12 exponentes codificados diferencialmente (correspondientes a las estrategias de

exponente d15, d25 y d45, respectivamente). El número de grupos de cada canal, nchgrps[ch] se determina a partir de cplbegf si el canal está acoplado o de chbwcod[ch] si el canal no está acoplado. Para más información consúltese el § 7.1.3.

5.4.3.28 gainrng[ch] – Código de gama de ganancia de canal – 2 bits

Este elemento de 2 bits por canal puede utilizarse para determinar un valor en punto flotante de desplazamiento de bloque para el banco de filtros de la transformación TDAC inversa. El empleo de este código permite la obtención de una gama dinámica aumentada a partir del cálculo de la transformación de una longitud de palabra limitada. Para más información, véase el § 7.9.5.

5.4.3.29 lfeexps[grp] – Exponentes del canal de efectos de baja frecuencia – 4 ó 7 bits

Estos elementos representan los exponentes codificados para el canal lfe. El primer elemento ([grp] = 0) es un exponente absoluto de 4 bits para el primer coeficiente de la transformación (término DC). Hay dos elementos adicionales (nlfegrps = 2) que son representaciones de 7 bits de un grupo de 3 exponentes codificados diferencialmente. El número total de exponentes del canal lfe (nlfemant) es igual a 7.

5.4.3.30 baie – Existe información de atribución de bit – 1 bit

Si este bit es un 1, se envían seguidamente 5 campos separados (con un total de 11 bits) en el tren de bits. Cada campo contiene valores de parámetro para el proceso de atribución de bits. Si este bit es un 0, no existen esos campos. En el § 7.2 se facilitan más detalles sobre esos campos.

5.4.3.31 sdcycod – Código de decremento lento – 2 bits

Se trata de un código de 2 bits que especifica el parámetro de decremento lento en el proceso de atribución de bits.

5.4.3.32 fdcycod – Código de decremento rápido – 2 bits

Código de 2 bits que especifica el parámetro de decremento rápido en el proceso de atribución de bits de decodificación.

5.4.3.33 sgaincod – Código de ganancia lenta – 2 bits

Código de 2 bits que especifica el parámetro ganancia lenta en el proceso de atribución de bits de decodificación.

5.4.3.34 dbpbcod – Código dB por bit – 2 bits

Código de 2 bits que especifica los dB por parámetro de bit en el proceso de atribución de bits.

5.4.3.35 floorcod – Código de umbral de enmascaramiento – 3 bits

Código de 3 bits que especifica el parámetro código umbral en el proceso de atribución de bits.

5.4.3.36 snroffste – Existe desplazamiento de SNR – 1 bit

Si este bit tiene un valor igual a 1, se envían seguidamente en el flujo de bits ciertos parámetros de atribución de bits. Si el valor de este bit es 0 no se envía a continuación la información de desplazamiento de SNR, debiendo utilizarse para este bloque los valores transmitidos anteriormente. En el § 7.2.2 se describen estos parámetros y el proceso de atribución de bits.

5.4.3.37 csnroffst – Desplazamiento SNR grueso – 6 bits

Código de 6 bits que especifica el parámetro desplazamiento SNR grueso del proceso de atribución de bits.

5.4.3.38 cplfsnroffst – Desplazamiento SNR fino de acoplamiento – 4 bits

Código de 4 bits que especifica el desplazamiento SNR fino del canal de acoplamiento en el proceso de atribución de bits.

5.4.3.39 cplfgaincod – Código de ganancia rápida de acoplamiento – 3 bits

Código de 3 bits que especifica el código de ganancia rápida del canal de acoplamiento empleado en el proceso de atribución de bits.

5.4.3.40 fsnroffst[ch] – Desplazamiento de SNR fino de canal – 4 bits

Código de 4 bits que especifica el desplazamiento SNR fino empleado en el proceso de atribución de bit para el canal [ch].

5.4.3.41 fgaincod[ch] – Código de ganancia rápida de canal – 3 bits

Código de 3 bits que especifica el parámetro ganancia rápida empleado en el proceso de atribución de bits para el canal [ch].

5.4.3.42 lfefsnroffst – Desplazamiento SNR fino del canal de efectos de baja frecuencia – 4 bits

Código de 4 bits que especifica el parámetro de desplazamiento SNR fino utilizado en el proceso de atribución de bits para el canal lfe.

5.4.3.43 lfefgaincod – Código de ganancia rápida del canal de efectos de baja frecuencia – 3 bits

Código de 3 bits que especifica el parámetro de ganancia rápida empleado en el proceso de atribución de bits para el canal lfe.

5.4.3.44 cplleake – Existe inicialización de fuga de acoplamiento – 1 bit

Si este bit es un 1, se envían seguidamente en el tren de bits los parámetros de inicialización de la fuga. Si este bit es un 0 se aplicarán los valores transmitidos previamente.

5.4.3.45 cplfleak – Inicialización de la fuga rápida de acoplamiento – 3 bits

Código de 3 bits que especifica el valor de inicialización de la fuga rápida para el cálculo de la función de excitación del canal en el proceso de atribución de bits.

5.4.3.46 cplsleak – Inicialización de la fuga lenta de acoplamiento – 3 bits

Código de 3 bits que especifica el valor de inicialización de la fuga lenta para el cálculo de la función de excitación del canal en el proceso de atribución de bits.

5.4.3.47 deltbaie – Existe información de atribución del bit delta – 1 bit

Si este bit es un 1, se envía seguidamente en el tren de bits alguna información de atribución del bit delta. Si este bit es un 0, se aplica todavía la información de atribución del bit delta transmitida previamente, salvo para el bloque 0. Si deltbaie en el bloque es un 0, se fijan a 0 cpldeltnseg y deltnseg[ch], no aplicándose la atribución del bit delta. En el § 7.2.2.6 se describe la atribución del bit delta.

5.4.3.48 cpldeltbae – Existe atribución del bit delta de acoplamiento – 2 bits

Código de 2 bits que indica la estrategia de atribución del bit delta para el canal de acoplamiento como se muestra en el Cuadro 11. Si se recibe el estado reservado, el decodificador no decodificará el audio y permanecerá en silencio.

5.4.3.49 deltbae[ch] – Existe atribución de bit delta – 2 bits

Código de 2 bits por canal de anchura de banda total que indica la estrategia de atribución del bit delta para el canal correspondiente, como se muestra en el Cuadro 11.

5.4.3.50 cpldeltnseg – Número de segmentos de la atribución del bit delta de acoplamiento – 3 bits

Código de 3 bits que indica el número de segmentos de atribución del bit delta existentes para el canal de acoplamiento. El valor de este parámetro varía de 1 a 8 y se calcula sumando 1 al número binario de 3 bits representado por el código.

CUADRO 11

Estados de existencia de atribución del bit delta

cpldeltbae, deltbae	**Código**
00	Se reutiliza el estado anterior
01	Sigue información nueva
10	No se realiza atribución delta
11	Reservado

5.4.3.51 cpldeltoffst[seg] – Desplazamiento de atribución del bit delta de acoplamiento – 5 bits

El primer código de 5 bits ([seg] = 0) indica el número de la primera banda de atribución de bit (como se especifica en el § 7.4.2) del canal de acoplamiento para el que se proporcionan valores de atribución del bit delta. Los códigos subsiguientes indican el desplazamiento desde el punto final del segmento delta anterior hasta la siguiente banda de atribución de bits para la que se proporcionan valores de atribución del bit delta.

5.4.3.52 cpldeltlen[seg] – Longitud de atribución del bit delta de acoplamiento – 4 bits

Cada código de 4 bits indica el número de bandas de atribución de bit a la que se aplica el segmento correspondiente.

5.4.3.53 cpldeltba[seg] – Atribución del bit delta de acoplamiento – 3 bits

Valor de 3 bits utilizado en el proceso de atribución de bits para el canal de acoplamiento.

Cada código de 3 bits indica un ajuste a la curva de enmascaramiento por defecto calculada en el decodificador. Las deltas se codifican como se muestra en el Cuadro 12.

CUADRO 12

Deltas de atribución de bits

cpldeltba, deltba	**Ajuste (dB)**
000	–24
001	–18
010	–12
011	–6
100	+6
101	+12
110	+18
111	+24

5.4.3.54 deltnseg[ch] – Número de segmentos de atribución de bits delta en el canal – 3 bits

Estos elementos asociados al canal de anchura de banda total son códigos de 3 bits que indican el número de segmentos de atribución de bits delta que existen para el canal correspondiente. El valor de este parámetro está comprendido entre 1 y 8 y se calcula añadiendo 1 al código binario de 3 bits.

5.4.3.55 deltoffst[ch][seg] – Desplazamiento de atribución del bit delta de canal – 5 bits

El primer código de 5 bits ([seg] = 0) indica el número de la primera banda de atribución de bits (véase el § 7.2.2.6) del canal correspondiente para el cual se han proporcionado valores de atribución del bit delta. Los códigos subsiguientes indican el desplazamiento desde el punto final del segmento delta anterior hasta la siguiente banda de atribución de bits para la cual se han proporcionado valores de atribución del bit delta.

5.4.3.56 deltlen[ch][seg] – Longitud de atribución del bit delta de canal – 4 bits

Cada código de 4 bits indica el número de bandas de atribución de bits que abarca cada segmento correspondiente.

5.4.3.57 deltba[ch][seg] – Atribución del bit delta de canal – 3 bits

Valor de 3 bits empleado en el proceso de atribución de bit para el canal indicado. Cada código de 3 bits indica un ajuste a la curva de enmascaramiento por defecto calculada en el decodificador. Las deltas se codifican como se muestra en el Cuadro 12.

5.4.3.58 skiple – Existe longitud de salto – 1 bit

Si este bit es un 1 se envía seguidamente el parámetro skipl en el tren de bits. Si este bit es un 0 no existe el skipl.

5.4.3.59 skipl – Longitud del salto – 9 bits

Código de 9 bits que indica el número de octetos ficticios que deben despreciarse antes de desempaquetar las mantisas del bloque de audio vigente.

5.4.3.60 skipfld – Campo de salto – (skipl × 8) bits

Este campo contiene los octetos nulos de los datos que han de ignorarse según lo indica el parámetro skipl.

5.4.3.61 chmant[ch][bin] – Mantisas de canal – 0 a 16 bits

Valores de la mantisa cuantificada real para el canal indicado. Cada valor puede contener un número de bits comprendido entre 0 y 16. El número de mantisas para el canal indicado es igual a nchmant[ch] que puede determinarse a partir de chbwcod[ch] (véase el § 7.1.3) si el canal no está acoplado o a partir de cplbegf (véase el § 7.4.2) si el canal está acoplado. En el § 7.3 se facilita información detallada sobre los datos de la mantisa empaquetada.

5.4.3.62 cplmant[bin] – Mantisas de acoplamiento – 0 a 16 bits

Valores reales de la mantisa cuantificada para el canal de acoplamiento. Cada valor puede contener un número de bits comprendido entre 0 y 16. El número de mantisas del canal de acoplamiento es igual a ncplmant, que puede calcularse mediante:

ncplmant = 12 x cplsubnd;

5.4.3.63 lfemant[bin] – Mantisas del canal de efectos de baja frecuencia – 0 a 16 bits

Valores reales de la mantisa cuantificada correspondientes al canal lfe. Cada valor puede contener un número de bits comprendido entre 0 y 16. El valor de nlfemant es igual a 7, por lo que hay 7 valores de mantisa para el canal lfe.

5.4.4 auxdata – Campo de datos auxiliares

Existirán datos no utilizados al final de una trama cuando el codificador no emplee todos los datos disponibles para la codificación de la señal audio. Esto puede suceder si la atribución de bits final es incapaz de utilizar todos los bits disponibles o sencillamente si la señal audio de entrada no necesita que se codifiquen de forma transparente todos los bits disponibles. O, alternativamente, puede ordenarse al codificador que deje adrede algunos bits sin utilizar por parte del audio de forma que queden disponibles para su empleo por datos auxiliares. Como el número de bits necesarios para los datos auxiliares puede ser inferior al número de bits disponibles (que variarán con el tiempo) en una trama determinada, se ha previsto un método para marcar el número real de bits de datos auxiliares en cada trama.

5.4.4.1 auxbits – Bits de datos auxiliares – bits nauxbits

Este campo contiene datos auxiliares. El número total de bits en este campo es:

nauxbits = (bits en la trama) – (bits utilizados por todos los elementos del tren de bits salvo por auxbits);

El número de bits en la trama puede determinarse a partir del código de tamaño de trama (frmsizcod) y el Cuadro 13. El número de bits utilizados comprende todos los bits empleados por los elementos del tren de bits con excepción de auxbits. Cualquier dato ficticio que se haya incluido con campos de salto (skipfld) se incluye en el cómputo de bits utilizados. El codificador ajusta la longitud del campo auxbits de forma que el elemento crc2 queda situado en la última palabra de 16 bits de la trama.

CUADRO 13

Cuadro de códigos de tamaño de trama (1 palabra = 16 bits)

frmsizecod	**Velocidad de bits nominal (kbit/s)**	**Palabras**/syncframe f_S **= 32 kHz**	**Palabras**/syncframe f_S **= 44,1 kHz**	**Palabras**/syncframe f_S **= 48 kHz**
‘000000’(0)	32	96	69	64
‘000001’(0)	32	96	70	64
‘000010’(1)	40	120	87	80
‘000011’(1)	40	120	88	80
‘000100’(2)	48	144	104	96
‘000101’(2)	48	144	105	96
‘000110’(3)	56	168	121	112
‘000111’(3)	56	168	122	112
‘001000’(4)	64	192	139	128
‘001001’(4)	64	192	140	128
‘001010’(5)	80	240	174	160
‘001011’(5)	80	240	175	160
‘001100’(6)	96	288	208	192
‘001101’(6)	96	288	209	192
‘001110’(7)	112	336	243	224
‘001111’(7)	112	336	244	224
‘010000’(8)	128	384	278	256
‘010001’(8)	128	384	279	256
‘010010’(9)	160	480	348	320
‘010011’(9)	160	480	349	320
‘010100’(10)	192	576	417	384
‘010101’(10)	192	576	418	384
‘010110’(11)	224	672	487	448
‘010111’(11)	224	672	488	448
‘011000’(12)	256	768	557	512
‘011001’(12)	256	768	558	512
‘011010’(13)	320	960	696	640
‘011011’(13)	320	960	697	640
‘011100’(14)	384	1152	835	768
‘011101’(14)	384	1152	836	768
‘011110’(15)	448	1344	975	896
‘011111’(15)	448	1344	976	896
‘100000’(16)	512	1536	1114	1024
‘100001’(16)	512	1536	1115	1024
‘100010’(17)	576	1728	1253	1152
‘100011’(17)	576	1728	1254	1152
‘100100’(18)	640	1920	1393	1280
‘100101’(18)	640	1920	1394	1280

f_S: frecuencia de muestreo

Si el número de bits de usuario indicado por auxdatal es menor que el número auxbits disponibles nauxbits, los datos de usuario se sitúan al final del campo auxbits. Esto le permite al decodificador encontrar y desempaquetar los bits de usuario auxdatal sin necesidad de conocer el valor de nauxbits (que únicamente puede determinarse mediante la decodificación del audio en la trama total). El orden de los datos de usuario en el campo auxbits es progresivo. Por consiguiente el decodificador de datos auxiliares (que puede no decodificar ninguna señal audio) simplemente debe observar el final de la syncframe AC-3 para encontrar auxdatal, recuperar los bits auxdatal (desde el principio de auxdatal) en el tren de datos y, seguidamente, desempaquetar los bits auxdatal desplazándose progresivamente en el flujo de datos.

5.4.4.2 auxdatal – Longitud de datos auxiliares – 14 bits

Valor entero de 14 bits que indica la longitud, en bits, de los datos de usuario en el campo auxiliar auxbits.

5.4.4.3 auxdatae – Existen datos auxiliares – 1 bit

Si este bit es un 1 precede el parámetro auxdatal en el tren de bits. Si este bit es un 0 no existe auxdatal, por lo que tampoco hay datos de usuario.

5.4.5 errorcheck – Campo de detección de errores de la trama

5.4.5.1 crcrsv – Bit reservado para CRC – 1 bit

Reservado para su empleo en aplicaciones específicas a fin de asegurar que crc2 será distinto de la palabra sync. Los codificadores pueden utilizar, facultativamente, este bit. Si el cálculo de crc2 proporciona un valor igual a syncword debe invertirse el bit crcrsv. Esto producirá un valor de crc2 distinto de syncword.

5.4.5.2 crc2 – Verificación de redundancia cíclica 2 – 16 bits

La CRC de 16 bits se aplica a la totalidad de la trama. En el § 7.10.1 se facilitan detalles sobre la verificación de CRC.

5.5 Limitaciones del tren de bits

El codificador AC-3 impone las siguientes limitaciones al tren de bits codificado. Tales limitaciones permiten la fabricación de decodificadores AC-3 con memorias tampón de entrada más pequeñas.

– El tamaño del bloque 0 y del bloque 1 combinados no rebasará nunca los 5/8 de la trama.
– La suma de los datos de mantisa del bloque 5 y de los datos auxiliares no rebasará nunca los 3/8 finales de la trama.
– El bloque 0 siempre contiene toda la información necesaria para comenzar a decodificar correctamente el tren de bits.
– Siempre que cambie el estado de cplinu de desactivado a activado se incluirá toda la información de acoplamiento en el bloque en el cual se conecta el acoplamiento. No se reutiliza ninguna información relacionada con el acoplamiento procedente de bloques anteriores en los que pudiera estar conectado el acoplamiento.

6 Decodificación del tren de bits AC-3

6.1 Introducción

En el preámbulo de esta norma se especifican los detalles de la sintaxis del tren de bits AC-3. En este punto se facilita una visión de conjunto del proceso de decodificación AC-3 como se indica

en el diagrama de la Fig. 13, en el que se muestra el flujo del proceso de decodificación como una secuencia de bloques descendente en el centro de la página, representándose parte de los flujos de información mediante las líneas con flechas situadas en los laterales de la página. Para más información sobre alguno de los bloques de tratamiento, consúltese el § 7. El decodificador descrito en este punto debe considerarse como un decodificado ejemplo. Puede haber otros métodos de realizar decodificadores, los cuales pueden presentar ventajas en algunos aspectos (tales como el cómputo de instrucciones, requisitos de memoria, número de transformaciones, etc.).

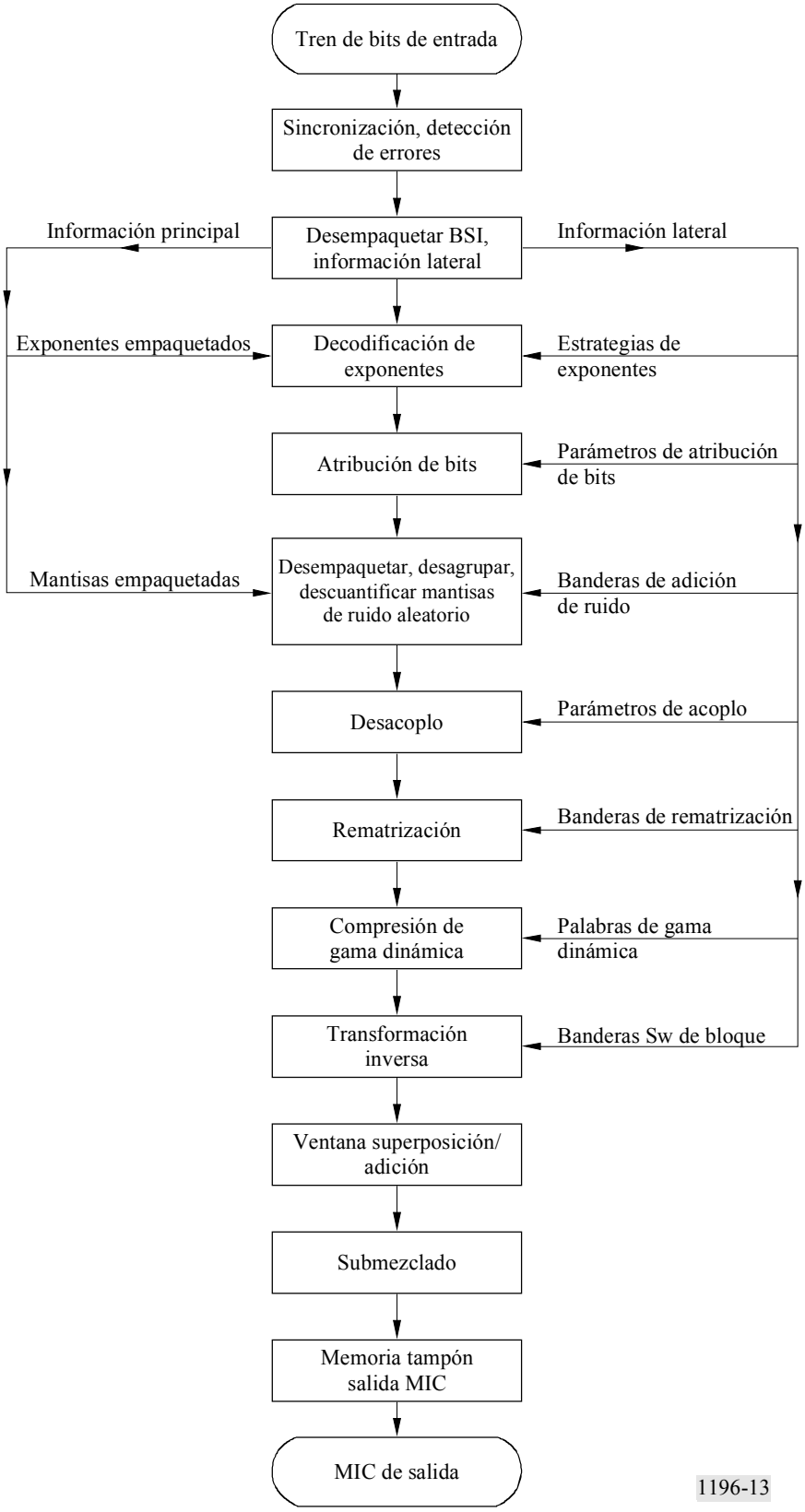
CUADRO 14

Cuadro de códigos de idiomas

langcod	**Idioma**	langcod	**Idioma**	langcod	**Idioma**	langcod	**Idioma**
00	Desconocido/ no aplicable	20	Polaco	40	Sonido de fondo/ alimentación limpia	60	Moldávico
01	Albanés	21	Portugués	41		61	Malasio
02	Bretón	22	Romanés	42		62	Malagasay
03	Catalán	23	Romanche	43		63	Macedonio
04	Croata	24	Servio	44		64	Letón
05	Galés	25	Eslovaco	45	Zulú	65	Coreano
06	Checo	26	Esloveno	46	Vietnamita	66	Khmer
07	Danés	27	Finlandés	47	Uzbek	67	Kazakh
08	Alemán	28	Sueco	48	Urdu	68	Kannada
09	Inglés	29	Turco	49	Ucraniano	69	Japonés
0A	Español	2A	Flamenco	4A	Thai	6A	Indonesio
0B	Esperanto	2B	Valón	4B	Telegu	6B	Hindi
0C	Estoniano	2C		4C	Tatar	6C	Hebreo
0D	Vasco	2D		4D	Tamil	6D	Hausa
0E	Faroese	2E		4E	Tadzhik	6E	Guaraní
0F	Francés	2F		4F	Swahili	6F	Gujurati
10	Frisio	30	Reservado para asignación nacional	50	Sranan Tongo	70	Griego
11	Irlandés	31		51	Somalí	71	Georgiano
12	Gaélico	32		52	Sinhalese	72	Fulani
13	Gallego	33		53	Shona	73	Dari
14	Islandés	34		54	Servocroata	74	Churash
15	Italiano	35		55	Rutenio	75	Chino
16	Lappish	36		56	Ruso	76	Burmese
17	Latín	37		57	Quechua	77	Búlgaro
18	Letón	38		58	Pustu	78	Bengalí
19	Luxemburgué	39		59	Punjabi	79	Bielorruso
1A	Lituano	3A		5A	Persa	7A	Bambora
1B	Húngaro	3B		5B	Papamiento	7B	Azebanés
1C	Maltés	3C		5C	Oriya	7C	Assamese
1D	Holandés	3D		5D	Nepalí	7D	Armenio
1E	Noruego	3E		5E	Ndebele	7E	Árabe
1F	Occitano	3F		5F	Marathi	7F	Amárico

FIGURA 13

Diagrama de flujo del proceso de decodificación



6.2 Resumen del proceso de decodificación

6.2.1 Tren de bits de entrada

Generalmente el tren de bits de entrada provendrá de un sistema de transmisión o de grabación. En esta norma no se especifica el interfaz entre la fuente de datos AC-3 y el decodificador AC-3. Los detalles del interfaz afectan a algunas características de realización del decodificador.

6.2.1.1 Entrada continua o en ráfagas

Los datos codificados con AC-3 pueden aplicarse al decodificador en forma de un tren de datos continuo o en forma de ráfagas con velocidad elevada y un ciclo de trabajo reducido. En el caso de funcionamiento en el modo ráfagas, la temporización de las ráfagas puede ser controlada por la fuente de datos o por el decodificador. La memoria tampón de entrada al decodificador AC-3 puede ser de tamaño más reducido si el decodificador es capaz de solicitar ráfagas de datos a medida que lo necesita. Sin embargo, en este caso, la memoria tampón externa debe ser más grande.

6.2.1.2 Alineación en octetos o en palabras

La mayoría de las aplicaciones de esta norma transportarán el tren de bits AC-3 elemental con una alineación en octetos o en palabras (16 bits). La syncframe tiene siempre una longitud igual a un número entero de palabras. El decodificador puede recibir datos en forma de un tren continuo de bits en serie sin ninguna alineación. Alternativamente, pueden aplicarse datos al decodificador con alineación en octetos o en palabras (16 bits). La alineación en octetos o en palabras de los datos de entrada puede permitir alguna simplificación del decodificador. La alineación reduce la probabilidad de detección falsa de la palabra de sincronización.

6.2.2 Sincronización y detección de errores

El formato del tren de bits de AC-3, permite una rápida sincronización. La palabra de sincronización de 16 bits tiene una probabilidad de detección falsa muy baja. Cuando no existe alineación del flujo de entrada, la probabilidad de detección falsa de la palabra de sincronización es igual a 0,0015% por cada posición de bit del tren de entrada. Para una velocidad de bits de 384 kbit/s, la probabilidad de detección falsa de la palabra de sincronización es el 19%. La alineación en octetos del flujo de entrada reduce esta probabilidad al 2,5% y la alineación en palabras la reduce al 1,2%.

Cuando se detecta una estructura de sincronización, el decodificador puede considerar que está en sincronismo, debiendo verificarse una de las palabras de CRC (crc1 ó crc2). Como la crc1 llega primero y abarca los primeros 5/8 de la trama el resultado de una comprobación crc1 únicamente estará disponible cuando se hayan recibido los 5/8 de la trama. Alternativamente, pueden recibirse la totalidad de la trama y comprobarse la crc2. Si se cumple cualquiera de las CRC, puede considerarse con seguridad que el decodificador está sincronizado, pudiendo procederse a la decodificación y reproducción del audio. En este caso el riesgo de sincronización falsa sería igual a la concatenación de probabilidades de una detección falsa de la palabra de sincronización y una detección errónea del CRC. La fiabilidad de la comprobación de CRC es del 0,0015%. Esta probabilidad combinada con la probabilidad de una detección falsa del sincronismo en un flujo de bits de entrada alineado en octetos proporciona una probabilidad de sincronización falsa del 0,000035% (aproximadamente una vez cada 3 millones de tentativas de sincronización).

Si esta probabilidad tan pequeña de sincronización falsa no es suficiente para alguna aplicación hay varios métodos para reducirla. El decodificador puede únicamente considerar que la sincronización es correcta cuando se verifiquen adecuadamente las dos palabras de CRC. El decodificador puede exigir la recepción de múltiples de sincronización con la alineación correcta. Si el sistema de transmisión o de grabación tiene constancia de que hay errores en los datos puede facilitarse esta información al decodificador.

En esta norma no se facilitan detalles adicionales sobre los métodos de sincronización del flujo de bits. En el § 7.10, se proporcionan más detalles sobre el cálculo de la CRC.

6.2.3 Desempaquetado de BSI, información lateral

Es inherente al proceso de decodificación el desempaquetado (demultiplexación) de los distintos tipos de información comprendidos en el tren de bits. Alguno de estos elementos pueden trasladarse desde la memoria tampón de entrada a registros dedicados, otros pueden copiarse en posiciones de memoria de trabajo específicas y otros pueden simplemente ubicarse en la memoria tampón de entrada con punteros almacenados en otra ubicación para emplearlos cuando se necesite la información. La información que debe desempaquetarse se especifica con detalle en el § 5.3. En esta norma no se facilitan detalles ulteriores sobre el desempaquetamiento del BSI y la información lateral.

6.2.4 Decodificación de exponentes

Los exponentes se transmiten en el tren de bits en forma codificada. Con el fin de desempaquetar y decodificar los exponentes se necesitan dos tipos de información lateral. En primer lugar, debe conocerse el número de exponentes. Para canales fbw tal número puede obtenerse a partir de chbwcod[ch] (para canales no acoplados) o a partir de cplbegf (para canales acoplados). Para el canal de acoplamiento, el número de exponentes puede determinarse a partir de cplbegf y cplendf. En el caso de canal lfe (cuando esté activado) siempre hay 7 exponentes. En segundo lugar debe conocerse la estrategia de exponentes que utiliza cada canal (d15, etc.). En el § 7.1, se facilitan detalles sobre como desempaquetar y decodificar los exponentes.

6.2.5 Atribución de bits

El cálculo de la atribución de bits revela como se utilizan los diversos bits para cada mantisa. Las entradas al cálculo de la atribución de bits son los exponentes decodificados y la información lateral de atribución. Las salidas del cálculo de atribución de bits son un conjunto de punteros de atribución de bits (baps) empleándose un bap para cada mantisa codificada. El bap indica el cuantificador utilizado para la mantisa y cuántos bits del tren de bits se emplearon para cada mantisa. En el § 7.2 se describe con detalle el cálculo de la atribución de bits.

6.2.6 Procesamiento de las mantisas

Las mantisas cuantificadas aproximadamente constituyen el núcleo del tren de datos AC-3. Cada mantisa se cuantifica con un nivel de precisión indicado por el bap correspondiente. A fin de empaquetar los datos de la mantisa con más eficiencia, se agrupan conjuntamente algunas mantisas para constituir un único valor transmitido. Por ejemplo, dos valores cuantificados de 11 niveles se transmiten mediante un único código de 7 bits (3,5 bits/valor) en el tren de bits.

Los datos de mantisa se desempaquetan separando los grupos de bits según lo indican los baps. Las mantisas agrupadas deben desagruparse. Los valores individuales de la mantisa codificada se convierten a un valor descuantificado. Las mantisas carentes de bits pueden reproducirse como 0 o como un valor de ruido aleatorio (bajo control de la bandera de adición de ruido). En el § 7.3 se describe con detalle el procesamiento de las mantisas.

6.2.7 Desacoplamiento

Cuando se utiliza el acoplamiento deben desacoplarse los canales acoplados. El desacoplamiento supone la reconstrucción de la sección de alta frecuencia (exponentes y mantisas) de cada canal acoplado, a partir del canal de acoplamiento común y las coordenadas de acoplamiento del canal individual. Dentro de cada banda de acoplamiento, se multiplican los coeficientes (exponente y mantisa) del canal de acoplamiento por las coordenadas de acoplamiento del canal individual. En el § 7.4 se describe con detalle el proceso de acoplamiento.

6.2.8 Rematrización

En el modo de codificación de audio 2/0 debe emplearse la rematrización, como lo indican las banderas de rematrización (rematflg[rbnd]). Cuando la bandera indica que una banda está rematrizada, los coeficientes codificados en el tren de bits son valores suma y diferencia en vez de valores izquierda y derecha. En el § 7.5 se describe con detalle la rematrización.

6.2.9 Compresión de la gama dinámica

Para cada bloque de audio puede incluirse en el tren de bits un valor de control de gama dinámica (dynrng). El decodificador utilizará, por defecto, este valor para alterar la magnitud del coeficiente (exponente y mantisa) como se especifica en el § 7.7.1.

6.2.10 Transformación inversa

Las fases de decodificación descritas anteriormente producirán un conjunto de coeficientes de frecuencia para cada canal codificado. La transformación inversa convierte los bloques de coeficientes de frecuencia en bloques de muestras temporales. La transformación inversa se detalla en el § 7.9.

6.2.11 Ventana, superposición/adición

Los bloques individuales de muestras temporales deben enventanarse superponiéndose bloques adyacentes y sumándose conjuntamente a fin de reconstruir la señal audio MIC continua en el tiempo. En el § 7.9 se describen las fases de enventanado y superposición/adición junto con la transformación inversa.

6.2.12 Submezclado

Si el número de canales necesarios a la salida del decodificador es inferior al número de canales codificados en el tren de bits será necesario aplicar el submezclado. En este decodificador ejemplo se muestra un submezclado en el dominio del tiempo. Como la transformación inversa es una operación lineal, resulta también posible efectuar el submezclado en el dominio de la frecuencia antes de la transformación. En el § 7.8 se describe el submezclado y se especifican los coeficientes de submezclado que deberán utilizar los decodificadores.

6.2.13 Memoria tampón de salida MIC

Los decodificadores típicos proporcionarán muestras de salida MIC a la velocidad de muestreo de MIC. Como el proceso de decodificación proporciona bloques de muestras, generalmente se necesitará una memoria de tampón de salida. En esta norma no se especifica ni se describe con ningún detalle ulterior la memoria tampón de salida.

6.2.14 MIC de salida

Las muestras MIC de salida pueden entregarse en un forma adecuada para su aplicación a un convertidor digital a analógico (DAC) o en cualquier otra forma. En esta norma no se especifica el formato MIC de salida.

7 Detalles del algoritmo

En los puntos que siguen se describen con detalle algunos aspectos de la codificación AC-3.

7.1 Codificación de exponentes

7.1.1 Visión de conjunto

La información de audio real transportada por el tren de bits AC-3 está constituida por los coeficientes de frecuencia cuantificados. Los coeficientes se suministran en forma de punto flotante, consistiendo cada coeficiente en un exponente y una mantisa. En este punto se describe como se codifican los exponentes y se insertan en el tren de bits.

Los exponentes son valores de 5 bits que indican el número de ceros delanteros en la representación binaria de un coeficiente de frecuencia. El exponente actúa como factor de escala para cada mantisa siendo igual a 2^{-exp}. Los valores del exponente varían entre 0 (en el caso de los coeficientes de máximo valor sin ceros delanteros) y 24. Los exponentes de coeficientes que tengan más de 24 ceros delanteros se fijan a 24, permitiéndose que las mantisas correspondientes tengan ceros delanteros. Son necesarios 5 bits para los exponentes a fin de poder representar todos los valores admisibles.

Los trenes de bits AC-3 contienen exponentes codificados para todos los canales correspondientes, todos los canales acoplados y para los canales de acoplamiento y los canales de efectos de baja frecuencia (cuando estén activados). Como la información de audio no se comparte a través de las tramas, el bloque 0 de cada trama incluirá nuevos exponentes para cada canal. La información de exponente puede compartirse dentro de los bloques de una trama, de forma que los bloques de 1 a 5 pueden reutilizar exponentes de los bloques anteriores.

La transmisión en AC-3 utiliza la codificación diferencial, en la cual los exponentes de un canal se codifican diferencialmente en función de la frecuencia. El primer exponente de un canal fbw o lfe se envía siempre en forma de un valor absoluto de 4 bits, variable entre 0 y 15. El valor indica el número de ceros delanteros del primer coeficiente de la transformada (término DC). Los exponentes sucesivos (en sentido de frecuencias crecientes) se envían como valores diferenciales que deben añadirse al valor del exponente anterior a fin de constituir el siguiente valor absoluto.

Los exponentes diferenciales se combinan en grupos en el bloque de audio. La agrupación se realiza mediante alguno de los tres métodos d15, d25 o d45, denominados estrategias de exponentes. El número de exponentes diferenciales agrupados situados en el bloque de audio de un canal concreto depende de la estrategia de exponente y de la información sobre la anchura de banda de frecuencias de ese canal. El número de exponentes de cada grupo depende únicamente de la estrategia de exponente.

Un bloque de audio AC-3 contiene dos tipos de campos con información de exponente. El primer tipo define la estrategia de codificación del exponente para cada canal y el segundo tipo contiene los exponentes codificados reales para los canales que requieran nuevos exponentes. Para canales independientes, se incluye información sobre la anchura de banda de frecuencias junto con los campos de estrategia de exponente. Para canales acoplados y el canal de acoplamiento, la información de frecuencia figura en los campos de estrategia de acoplamiento.

7.1.2 Estrategia de exponente

En cada bloque de audio AC-3 se incluye información de estrategia de exponente para cada canal. La información nunca se comparte a través de las tramas, por lo que el bloque 0 deberá siempre contener una indicación de estrategia (d15, d25 ó d45) para cada canal. Los bloques 1 a 5 pueden indicar la reutilización de los exponentes anteriores (dentro de la misma trama). Las tres estrategias de codificación de exponente proporcionan un intercambio entre la velocidad de datos requerida

para los exponentes y su resolución de frecuencia. El modo d15 proporciona la resolución de frecuencia más elevada y el modo d45 requiere el mínimo volumen de datos. En los tres modos, varios exponentes diferenciales se combinan en palabras de 7 bits cuando se codifican en un bloque de audio. La diferencia principal entre los modos reside en cómo se combinan conjuntamente diversos exponentes diferenciales.

Los exponentes absolutos que aparecen en el tren de bits al principio de los conjuntos de exponentes codificados diferencialmente se transmiten en forma de valores de 4 bits limitados en gama o en resolución a fin de ahorrar 1 bit. Para canales fbw y lfe, el exponente absoluto de 4 bits inicial representa un valor comprendido entre 0 y 15. Los valores de exponente superiores a 15 se limitan a un valor igual a 15. En el caso del canal acoplado, el exponente absoluto de 5 bits se limita a valores pares y no se transmite el LSB. Se ha limitado la resolución a valores válidos de 0, 2, 4, ... 24. Cada exponente diferencial puede tomar uno de entre 5 valores: –2, –1, 0, +1, +2. Esto permite deltas de hasta ±2 (±12 dB) entre exponentes. Estos cinco valores se ponen en correspondencia con los valores 0, 1, 2, 3 y 4 antes de agruparse como se muestra en el Cuadro 15:

CUADRO 15

Correspondencia de valores de exponente diferencial, modo d15

Exponente diferencial	Valor de correspondencia
+2	4
+1	3
0	2
–1	1
–2	0

Valor de correspondencia: exponente diferencial + 2
Exponente diferencial: valor de correspondencia – 2

En el modo d15, se aplica la correspondencia anterior a cada exponente diferencial individual para su codificación en el tren de bits. En el modo d25, cada pareja de exponentes diferenciales se representa mediante un único valor de correspondencia en el tren de bits. En este modo, el segundo exponente diferencial de cada pareja está implícito como una delta de 0 desde el primer elemento de la pareja como se indica en el Cuadro 16.

CUADRO 16

orrespondencia de valores de exponente diferencial, modo d25

Exponente diferencial *n*	Exponente diferencial *n* + 1	Valor de correspondencia
+2	0	4
+1	0	3
0	0	2
–1	0	1
–2	0	0

El modo d45 es similar al modo d25 salvo en que las *cuaternas* de exponentes diferenciales se representan por un único valor de correspondencia como se indica en el Cuadro 17.

CUADRO 17

Correspondencia de valores de exponente diferencial, modo d45

Exponente diferencial *n*	**Exponente diferencial *n* + 1**	**Exponente diferencial *n* + 2**	**Exponente diferencial *n* + 3**	**Valor de correspondencia**
+2	0	0	0	4
+1	0	0	0	3
0	0	0	0	2
–1	0	0	0	1
–2	0	0	0	0

Como un único exponente se comparte de hecho por 2 ó 4 mantisas diferentes, los decodificadores deben asegurar que el exponente elegido para la pareja o la cuaterna es el valor absoluto mínimo (correspondiente al exponente mayor) necesario para representar todas las mantisas.

Para todos los modos, se agrupan conjuntamente juegos de 3 valores de correspondencia (en frecuencia) adyacentes (M1, M2 y M3) y se codifican como un valor de 7 bits según la expresión siguiente:

$$\text{Valor agrupado de 7 bits codificado} = (25 \times M1) + (5 \times M2) + M3$$

El campo de exponentes para un canal determinado en un bloque de audio AC-3 consta de un único exponente absoluto seguido por varios valores agrupados de este tipo.

7.1.3 Decodificación de exponentes

La estrategia de exponente para cada canal acoplado e independiente se incluye en un conjunto de campos de 2 bits designado mediante chexpstr[ch]. Cuando está presente el canal de acoplamiento se incluye asimismo un código de estrategia cplexpstr. En el Cuadro 18 se representa la correspondencia entre el código de estrategia de exponente y la estrategia de exponente.

CUADRO 18

Codificación de estrategia de exponente

chexpstr[ch], cplexpstr	**Estrategia de exponente**	**Exponentes por grupo**
00	Reutilización de exponentes anteriores	0
01	d15	3
10	d25	6
11	d45	12

Cuando está activado el canal de efectos de baja frecuencia, está presente el campo lfeexpstr. Se decodifica como se muestra en el Cuadro 19.

CUADRO 19

Codificación de estrategia de exponente del canal lfe

lfeexpstr	**Estrategia de exponente**	**Exponentes por grupo**
0	Reutilización de exponentes anteriores	0
1	d15	3

Siguiendo a los campos de estrategia de exponente del tren de bits aparece un conjunto de códigos de anchura de banda de canal, chbwcod[ch]. Tales códigos únicamente están presentes en el caso de canales independientes (canales no acoplados) que poseen nuevos exponentes en el bloque vigente. El código de anchura de banda de canal define el número bin de mantisa final de ese canal de acuerdo con la siguiente expresión:

```
endmant[ch] = ((chbwcod[ch] + 12) * 3) + 37;        /* (ch no acoplado) */
```

Para canales acoplados el número bin de la mantisa final se define como el número bin de comienzo del canal acoplado:

```
endmant[ch] = cplstrtmant;                          /* (ch acoplado) */
```

donde cplstrtmant es el parámetro que se deduce seguidamente. Por definición el número bin de la mantisa de inicio para los canales independientes y acoplados es 0.

```
strtmant[ch] = 0;
```

Para el canal de acoplamiento, la información de anchura de banda de frecuencias se deduce de los campos cplbegf y cplendf que aparecen en la información de estrategia de acoplamiento. Los bins de las mantisas de inicio y final del canal de acoplamiento se definen así:

```
cplstrtmant = (cplbegf * 12) + 37;

cplendmant = ((cplendf + 3) * 12) + 37;
```

El canal de efectos de baja frecuencia, cuando esté presente, comienza siempre con el bin 0 y tiene siempre el mismo número de mantisas:

```
lfestrtmant = 0;

lfeendmant = 7;
```

El segundo conjunto de campos contiene exponentes codificados para aquellos canales que tienen exponentes nuevos en el bloque vigente. Se designan estos campos como exps[ch][grp] para los canales independientes y acoplados, cplexps[grp] para el canal de acoplamiento y lfeexps[grp] para el canal de efectos de baja frecuencia. El primer elemento de los campos (exps[ch][0]) y el campo lfeexps (lfeexps[0]) es siempre un número absoluto de 4 bits. Para estos canales el exponente absoluto contiene siempre el valor de exponente del primer coeficiente de la transformación (bin #0). Estos valores de 4 bits corresponden a un exponente de 5 bits limitado en gama (0 a 15 en

vez de 0 a 24) es decir el bit más significativo es 0. El exponente absoluto del canal acoplado, cplabsexp únicamente se emplea como referencia para iniciar la decodificación de los exponentes diferenciales del canal acoplado (esto es, no representa un exponente real). El cplabsexp está contenido en el bloque de audio en forma de un valor de 4 bits, aunque corresponde a un valor de 5 bits. El LSB del exponente inicial del canal acoplado siempre es 0, de forma que el decodificador debe tomar el valor de 4 bits transmitido y duplicarlo (desplazarlo a la izquierda una unidad) con el fin de obtener el valor de arranque de 5 bits.

Para cada conjunto de exponentes codificados se obtiene el número de exponentes agrupados (sin incluir el primer exponente absoluto) que ha de decodificarse del tren de bits como sigue:

Para canales independientes y acoplados:

```
nchgrps[grp] = truncate ((endmant[grp] – 1) / 3);          /* para el modo d15 */
             = truncate ((endmant[grp] – 1 + 3) / 6);      /* para el modo d25 */
             = truncate ((endmant[grp] – 1 + 9) / 12);     /* para el modo d45 */
```

Para el canal de acoplamiento:

```
ncplgrps     = (cplendmant – cplstrtmant) / 3;             /* para el modo d15 */
             = (cplendmant – cplstrtmant) / 6;             /* para el modo d25 */
             = (cplendmant – cplstrtmant) / 12;            /* para el modo d45 */
```

Para el canal de efectos de baja frecuencia:

```
nlfegrps     = 2;
```

La decodificación de un conjunto de exponentes agrupados codificados generará un juego de exponentes absolutos de 5 bits. Los exponentes se decodifican como sigue:

1. Cada agrupación de 7 bits de valores de correspondencia (gexp) se decodifica empleando la inversa del procedimiento de codificación:

   ```
   M1 = truncate (gexp / 25);
   M2 = truncate ((gexp % 25) / 5);
   M3 = (gexp % 25) % 5;
   ```

2. Cada valor de correspondencia se convierte en un exponente diferencial (dexp) restando el desplazamiento de correspondencia:

   ```
   dexp = M – 2;
   ```

3. El conjunto de exponentes diferenciales se convierte en exponentes absolutos añadiendo cada exponente diferencial al exponente absoluto del bin de frecuencia anterior:

   ```
   exp[n] = exp[n–1] + dexp[n];
   ```

4. Para los modos d25 y d45 se transfiere cada exponente absoluto a los miembros restantes de la pareja o de la cuaterna.

El procedimiento anterior puede resumirse como sigue:

Pseudocódigo

```
/* decodificación de los valores de correspondencia */
for (grp = 0; grp < ngrps; grp++)
{
    expacc = gexp[grp];
    dexp[grp * 3] = truncate (expacc / 25);
    expacc = expacc – ( 25 * dexp[grp * 3]);
    dexp[(grp * 3) + 1] = truncate (expacc / 5);
    expacc = expacc – (5 * dexp[(grp * 3) + 1]);
    dexp[(grp * 3) + 2] = expacc;
}

/* valores de correspondencia sin sesgo */
for (grp = 0; grp < (ngrps * 3); grp++)
{
    dexp[grp] = dexp[grp] – 2;
}

/* conversión de valores diferenciales a valores absolutos */
prevexp = absexp;
for (i = 0; i < (ngrps * 3); i++)
{
    aexp[i] = prevexp + dexp[i];
    prevexp = aexp[i];
}

/* expansión al grupo completo de exponentes absolutos, utilizando tamaño de grupo */
exp[0] = absexp;
for (i = 0; i < (ngrps * 3); i++)
{
    for (j = 0; j < grpsize; j++)
    {
        exp[(i * grpsize) + j +1] = aexp[i];
    }
}
```

donde:

ngrps:	número de exponentes agrupados (nchgrps[ch], ncplgrps, o nlfegrps)
grpsize	= 1 para d15
	= 2 para d25
	= 4 para d45
absexp:	exponente absoluto (exps[ch][0], (cplabsexp<<1), o lfeexps[0])

Para el canal de acoplamiento la fila de salida anterior, exp[n] debe desplazarse para que se corresponda con el bin de la mantisa de inicio de acoplamiento:

```
cplexp[n + cplstrtmant] = exp[n + 1];
```

Para los canales restantes exp[n] se corresponderá directamente con la fila de exponente absoluto para ese canal.

7.2 Atribución de bits

7.2.1 Visión de conjunto

La rutina de atribución de bits analiza la envolvente espectral de la señal audio que se está codificando con relación a los efectos de enmascaramiento a fin de determinar el número de bits que debe asignarse a cada mantisa del coeficiente de la transformación. En el codificador, la atribución de bits se efectúa globalmente en el conjunto de canales como un todo a partir de una reserva de bits común. No existen bits de exponente o de mantisa preasignados, permitiendo así que la rutina asigne de modo flexible los bits a través de los canales, frecuencias y bloques de audio de conformidad con la demanda de señal.

La atribución de bits contiene un modelo paramétrico de la audición humana con el fin de estimar un umbral de nivel de ruido expresado en función de la frecuencia, que distingue entre componentes espectrales audibles e inaudibles. El codificador puede ajustar diversos parámetros del modelo de audición dependiendo de las características de la señal. Por ejemplo, se define una curva tipo de enmascaramiento mediante dos segmentos de línea continuos, cada uno con su propia pendiente y ordenada en el eje y. Para cada segmento de línea el codificador elige una de las diversas pendientes y ordenadas posibles. El codificador puede efectuar iteraciones con uno o más de estos parámetros hasta alcanzar un resultado óptimo. Cuando el decodificador ha seleccionado la totalidad de los parámetros utilizados para la estimación del umbral del nivel de ruido, se determina la atribución de bits final. Los parámetros del modelo se transfieren al decodificador junto con otra información colateral. El decodificador ejecuta la rutina en una sola operación.

Se calcula el umbral de nivel de ruido estimado sobre 50 bandas de anchura de banda no uniforme (aproximadamente una escala de 1/6 de octava). La estructura de bandas definidas mediante cuadros en la siguiente sección es independiente de la frecuencia de muestreo. La atribución de bits requerida por cada mantisa se establece realizando una búsqueda en la tabla basada en la diferencia entre la densidad espectral de potencia (PSD) de la señal de entrada evaluada en una escala de frecuencias uniforme de gran resolución, en tanto que el umbral del nivel de ruido estimado se evalúa en una escala de frecuencias (dividida en bandas) de escasa resolución. En consecuencia, el resultado de atribución de bits para un canal concreto tiene una granularidad espectral que se corresponde con la estrategia de exponente utilizada. Más específicamente, se calculará para cada mantisa una atribución de bits separada dentro de un conjunto de exponentes d15, para cada par de mantisas dentro de un conjunto de exponentes d25 y para cada cuaterna de mantisas dentro de un conjunto de exponentes d45.

En el decodificador deberá calcularse la atribución de bits siempre que la estrategia de exponente (chexpstr, cplexpstr, lfeexpstr) para uno o más canales no indique reutilización o cuando baie, snroffste, o deltbaie = 1. En consecuencia, la atribución de bits puede actualizarse a un ritmo variable entre una vez por cada bloque de audio y una vez por cada 6 bloques de audio, incluyendo entre medias escalones enteros. En el bloque de audio 0 se transmite siempre un conjunto completo de información nueva de atribución de bits.

Como la rutina paramétrica de atribución de bits debe generar resultados idénticos en todas las realizaciones del codificador y del decodificador, cada escalón se define exactamente en función de operaciones con enteros de punto fijo y búsquedas en tablas. En el texto que sigue se utiliza una aritmética de complemento a dos con signo. Todas las sumas se realizan con un acumulador de 14 o más bits. Todos los resultados intermedios y valores almacenados son valores de 8 bits.

7.2.2 Atribución de bits paramétrica

En este punto se describe un procedimiento de 7 etapas para calcular la salida de la rutina de atribución de bits paramétrica en el decodificador. El enfoque esbozado aquí comienza con un conjunto de exponentes único acoplado o desacoplado y procesa en cada paso todos los datos de

entrada antes de proseguir al paso siguiente. Esta técnica denominada ejecución vertical, conceptualmente es inmediata de describir y realizar. Alternativamente, los 7 pasos pueden ejecutarse de forma horizontal, en cuyo caso se realizan múltiples recorridos a través de las 7 etapas para subconjuntos separados del conjunto de exponentes de entrada.

La elección entre la ejecución vertical u horizontal depende de la importancia relativa del tiempo de ejecución con respecto a la utilización de memoria en la realización final. Generalmente la ejecución vertical del algoritmo es más rápida, debido a que se reduce el número de bloques así como la tara asociada al contexto. Sin embargo, la ejecución horizontal requiere menos memoria RAM para almacenar las filas temporales generadas en cada etapa. Es también posible utilizar métodos de realización mixtos horizontal/vertical que combinan las ventajas de ambas técnicas.

7.2.2.1 Inicialización

Se calcularán las frecuencias de comienzo/final correspondientes al canal que se está decodificando. Tales frecuencias se calculan a partir de los parámetros del tren de bits como sigue:

Pseudocódigo

```
/* para canales con anchura de banda total */
for(ch=0; ch<nfchans; ch++)
{
    strtmant[ch] = 0;
    if(chincpl[ch]) endmant[ch] = 37 + (12 × cplbegf); /* el canal está acoplado */
    else endmant[ch] = 37 + (3 × (chbwcod + 12)); /* el canal no está acoplado */
}

/* para canal de acoplamiento */
cplstrtmant = 37 + (12 × cplbegf);
cplendmant = 37 + (12 × (cplendf + 3));

/* para canales de efectos de baja frecuencia */
lfestartmant = 0;
lfeendmant = 7;
```

Paso de procesamiento de caso especial:

Antes de continuar con el procedimiento de inicialización, deberán evaluarse todos los parámetros de desplazamiento SNR a partir del tren de bits. Estos parámetros comprenden csnroffst, fsnroffst[ch], cplfsnroffst y lfefsnroffst. Si se observa que todos ellos son iguales a 0, deberán ponerse a 0 todos los elementos de la fila bap[] del puntero de atribución de bits, no necesitándose ningún otro proceso de atribución de bits para el bloque de audio vigente.

Se ejecutarán las búsquedas en tabla para determinar los valores de sdecay, fdecay, sgain, dbknee y floor obtenidos de los parámetros del tren de bits:

Pseudocódigo

```
sdecay = slowdec[sdcycod];        /* Cuadro 20 */
fdecay = fastdec[fdcycod];        /* Cuadro 21 */
sgain = slowgain[sgaincod];       /* Cuadro 22 */
dbknee = dbpbtab[dbpbcod];        /* Cuadro 23 */
floor = floortab[floorcod];       /* Cuadro 24 */
```

Se efectuará la inicialización siguiente para la porción no acoplada del canal fbw:

Pseudocódigo

start = strtmant[ch];
end = endmant[ch];
lowcomp = 0;
fgain = fastgain[fgaincod[ch]]; /* Cuadro 25 */
snroffset[ch] = ((csnroffst – 15) << 4 + fsnroffst[ch]) << 2;

Se efectuará la inicialización para el canal de acoplamiento como sigue:

Pseudocódigo

start = cplstrtmant;
end = cplendmant;
fgain = fastgain[cplfgaincod]; /* Cuadro 25 */
snroffset = ((csnroffst – 15) << 4 + cplfsnroffst) << 2;
if (cplleake)
{
  fastleak = (cplfleak << 8) + 768;
  slowleak = (cplsleak << 8) + 768;
}

Se efectuará la inicialización del canal lfe como sigue:

Pseudocódigo

start = lfestrtmant;
end = lfeendmant;
lowcomp = 0;
fgain = fastgain[lfefgaincod];
snroffset = ((csnroffst – 15) << 4 + lfefsnroffst) << 2;

7.2.2.2 Correspondencia de exponentes con psd

Esta etapa pone en correspondencia los exponentes decodificados con un número de 13 bits con signo asociado al algoritmo de la función de densidad espectral de potencia.

Pseudocódigo

for (bin=start; bin<end; bin++)
{
  psd[bin] = (3072 – (exp[bin] << 7));
}

Como exp[k] supone valores enteros comprendidos entre 0 y 24, la gama dinámica de los valores psd[] va de 0 (para la señal de nivel más bajo) a 3 072 para la señal de nivel más alto. La función resultante se representa en una escala de frecuencias lineal de gran resolución.

7.2.2.3 Integración de la psd

Esta etapa del algoritmo integra valores de la psd de gran resolución dentro de cada multiplicidad de bandas de 1/6 de octava. El Cuadro 26 contiene los 50 valores filas de bndtab[] y bndsz[]. La fila bndtab[] proporciona el primer número de mantisa de cada banda. La fila bndsz[] proporciona la anchura de cada banda en forma del número de mantisas incluida. El Cuadro 27 contiene los 256 valores fila para masktab[], mostrando la correspondencia entre el número de mantisa y el número de banda de 1/6 de octava asociado. Estos dos cuadros contienen información duplicada, la cual no tiene por qué estar disponible en una realización real. Se muestran aquí únicamente para mayor sencillez de la presentación.

La integración en cada banda de los valores de la psd se realiza con registro-adición logarítmica. Se establece el registro-adición calculando la diferencia entre los dos operandos y utilizando la diferencia absoluta dividida por 2 como dirección en una tabla de búsqueda de longitud 256 latab[] representada en el Cuadro 28.

Pseudocódigo

```
j = start;
k = masktab[start];
do
{
    bndpsd[k] = psd[j];
    j++;
    for (i = j; i < min(bndtab[k+1], end); i++)
    {
        bndpsd[k] = logadd(bndpsd[k], psd[j]);
        j++;
    }
    k++;
}
while (end > bndtab[k++]);
logadd(a, b)
{
    c = a – b;
    address = min((abs(c) >> 1), 255);
    if (c >= 0)
    {
        return(a + latab(address));
    }
    else
    {
        return(b + latab(address));
    }
}
```

7.2.2.4 Cálculo de la función de excitación

La función de excitación se calcula aplicando la curva máscara prototipo seleccionada por el codificador (y transmitida al decodificador) al espectro de psd integrado (bndpsd[]). Seguidamente el resultado de este cálculo se desplaza en amplitud hacia abajo utilizando los parámetros fgain y sgain que se obtienen también del tren de bits.

Pseudocódigo

```
bndstrt = masktab[start];
bndend = masktab[end – 1] + 1;
if (bndstrt == 0) /* para los canales fbw y lfe */
{ /* Nota: No llamar calc_lowcomp() a la última banda del canal lfe, (bin = 6) */
    lowcomp = calc_lowcomp(lowcomp, bndpsd[0], bndpsd[1], 0);
    excite[0] = bndpsd[0] – fgain – lowcomp;
    lowcomp = calc_lowcomp(lowcomp, bndpsd[1], bndpsd[2], 1);
    excite[1] = bndpsd[1] – fgain – lowcomp;
    begin = 7;
    for (bin = 2; bin < 7; bin++)
    {
        lowcomp = calc_lowcomp(lowcomp, bndpsd[bin], bndpsd[bin+1], bin);
        fastleak = bndpsd[bin] – fgain;
        slowleak = bndpsd[bin] – sgain;
        excite[bin] = (fastleak – lowcomp);
        if (bndpsd[bin] <= bndpsd[bin+1])
        {
            begin = bin + 1;
            break;
        }
    }
    for (bin = begin; bin < min(bndend, 22); bin++)
    {
        lowcomp = calc_lowcomp(lowcomp, bndpsd[bin], bndpsd[bin+1], bin);
        fastleak –= fdecay;
        fastleak = max(fastleak, bndpsd[bin] – fgain);
        slowleak –= sdecay;
        slowleak = max(slowleak, bndpsd[bin] – sgain);
        excite[bin] = max(fastleak – lowcomp, slowleak);
    }
    begin = 22;
}
else /* para el canal acoplado */
{
    begin = bndstrt;
}
for (bin = begin; bin < bndend; bin++)
{
    fastleak –= fdecay;
    fastleak = max(fastleak, bndpsd[bin] – fgain);
    slowleak –= sdecay;
    slowleak = max(slowleak, bndpsd[bin] – sgain);
    excite[bin] = max(fastleak, slowleak);
}
calc_lowcomp(a, b0, b1, bin)
{
    if (bin < 7)
    {
        if ((b0 + 256) == b1);
        {
            a = 384;
        }
        else if (b0 > b1)
        {
            a = max(0, a – 64);
        }
    }
    else if (bin < 20)
    {
        if ((b0 + 256) == b1)
        {
            a = 320;
        }
        else if (b0 > b1)
        {
            a = max(0, a – 64);
        }
    }
    else
    {
        a = max(0, a – 128);
    }
    return(a);
}
```

7.2.2.5 Cálculo de la curva de enmascaramiento

Esta etapa calcula la curva de enmascaramiento (umbral de nivel de ruido) a partir de la función de excitación, como se indica a continuación. En el Cuadro 29 se muestra el umbral de audición hth[][]. El decodificador recibe en el tren de bits las variables fscod y dbpbcod.

Pseudocódigo

for (bin = bndstrt; bin < bndend; bin++)
{
  if (bndpsd[bin] < dbknee)
  {
  excite[bin] += ((dbknee – bndpsd[bin]) >> 2);
  }
  mask[bin] = max(excite[bin], hth[fscod][bin]);
}

7.2.2.6 Aplicación de la atribución del bit delta

La información facultativa de atribución del bit delta, dba, en el tren de bits proporciona al codificador un medio para transmitir al decodificador información colateral que aumenta o disminuye directamente la curva de enmascaramiento obtenida por la rutina paramétrica. El codificador puede habilitar la atribución de bit delta para bloques de audio que proporcionan una mejora de la calidad de audio, cuando se modifica de forma apropiada la atribución de bit por defecto. La opción de atribución del bit delta está disponible para cada canal fbw y para el canal de acoplamiento.

En el caso en que no se utilice la atribución de bit delta y no se incluya la información de dba en el tren de bits, el decodificador no deberá modificar la atribución por defecto. Una manera de asegurar esto es inicializar a 0, al comienzo de cada trama, las variables de atribución del bit delta cpldeltnseg y deltnseg[ch]. Esto provoca la conclusión inmediata del procesamiento dba (indicado más adelante), a menos que se incluya en el tren de bits la información dba (incluidas cpldeltnseg y deltnseg[ch]).

La información dba que modifica la atribución de bits del decodificador se transmite como información lateral. Las modificaciones de la atribución se producen en forma de ajustes a la curva de enmascaramiento por defecto calculada en el decodificador. Los ajustes pueden efectuarse en múltiplos de ±6 dB. Por término medio, un ajuste de la curva de enmascaramiento de –6 dB corresponde a un incremento de 1 bit en la resolución de todas las mantisas de la banda de 1/6 de octava afectada. El código que sigue indica, para un único canal, cómo se efectúa la modificación. El cálculo de la modificación se realiza en el canal de acoplamiento (siendo deltnseg igual a cpdeltnseg) y en cada canal fbw (siendo deltnseg igual deltnseg[ch]).

Pseudocódigo

```
if ((deltbae == 0) || (deltbae == 1))
{
    band = 0;
    for (seg = 0; seg < deltnseg+1; seg++)
    {
        band += deltoffst[seg];
        if (deltba[seg] >= 4)
        {
            delta = (deltba[seg] – 3) << 7;
        }
        else
        {
            delta = (deltba[seg] – 4) << 7;
        }
        for (k = 0; k < deltlen[seg]; k++)
        {
            mask[band] += delta;
            band++;
        }
    }
}
```

7.2.2.7 Cálculo de la atribución de bits

En esta etapa se calcula la fila de puntero de atribución de bits (bap[]). De la fila psd[] de gran resolución, se resta la curva de enmascaramiento ajustada por snroffset en una fase anterior y truncada subsiguientemente. A continuación se desplaza la diferencia 5 bits a la derecha, se le aplica un umbral y el resultado se utiliza como dirección baptab[] para obtener la atribución final. En el Cuadro 30 se muestra la fila baptab[].

El codificador limita la suma de todas las atribuciones de mantisa del canal a fin de que sea menor o igual que el número total de bits de mantisa disponibles para esa trama. El codificador ejecuta esta tarea iterando los valores de csnroffst y fsnroffst (o cplfsnroffst o lfefsnroffst para los canales de acoplamiento y de efectos de baja frecuencia) para obtener un resultado apropiado. Se garantiza que el decodificador recibirá una atribución de mantisa que cumpla las limitaciones de una velocidad de transmisión de bits fija.

Al final de esta etapa la fila bap[] contiene una serie de punteros de 4 bits. Los punteros indican el número de bits asignado a cada mantisa. En el Cuadro 31 se muestra la correspondencia entre el valor del puntero bap y la exactitud de la cuantificación.

Pseudocódigo

```
i = start;
j = masktab[start];
do
{
    mask[j] –= snroffset;
    mask[j] –= floor;
    if (mask[j] < 0)
    {
        mask[j] = 0;
    }
    mask[j] &= 0x1fe0;
    mask[j] += floor;
    for (k = i; k < min(bndtab[j] + bndsz[j], end); k++)
    {
        address = (psd[i] – mask[j]) >> 5;
        address = min(63, max(0, address));
        bap[i] = baptab[address];
        i++;
    }
}
while (end > bndtab[j++]);
```

7.2.3 Cuadros de atribución de bits

CUADRO 20

Cuadro de decremento lento, slowdec[ ]

Dirección	slowdec[address]
0	0×0f
1	0×11
2	0×13
3	0×15

CUADRO 21

Cuadro de decremento rápido, fastdec[ ]

Dirección	fastdec[address]
0	0×3f
1	0×53
2	0×67
3	0×7b

CUADRO 22

Cuadro de ganancia lenta, slowgain[ ]

Dirección	slowgain[address]
0	0×540
1	0×4d8
2	0×478
3	0×410

CUADRO 23

Cuadro dB/bit, dbpbtab[ ]

Dirección	dbpbtab[address]
0	0×000
1	0×700
2	0×900
3	0×b00

CUADRO 24

Cuadro de umbral, floortab[ ]

Dirección	floortab[address]
0	0×2f0
1	0×2b0
2	0×270
3	0×230
4	0×1f0
5	0×170
6	0×0f0
7	0×f800

CUADRO 25

Cuadro de ganancia rápida, fastgain[ ]

Dirección	fastgain[address]
0	0×080
1	0×100
2	0×180
3	0×200
4	0×280
5	0×300
6	0×380
7	0×400

CUADRO 26

Cuadros de estructura de formación de bandas, bndtab[ ], bndsz[ ]

N.° de la banda	bndtab[band]	bndsz[band]	**N.° de la banda**	bndtab[band]	bndsz[band]
0	0	1	25	25	1
1	1	1	26	26	1
2	2	1	27	27	1
3	3	1	28	28	3
4	4	1	29	31	3
5	5	1	30	34	3
6	6	1	31	37	3
7	7	1	32	40	3
8	8	1	33	43	3
9	9	1	34	46	3
10	10	1	35	49	6
11	11	1	36	55	6
12	12	1	37	61	6
13	13	1	38	67	6
14	14	1	39	73	6
15	15	1	40	79	6
16	16	1	41	85	12
17	17	1	42	97	12
18	18	1	43	109	12
19	19	1	44	121	12
20	20	1	45	133	24
21	21	1	46	157	24
22	22	1	47	181	24
23	23	1	48	205	24
24	24	1	49	229	24

CUADRO 27

Cuadro de correspondencia número de segmento (bin)-número de banda, masktab[bin] bin = (10 × A) + B

	B = 0	B = 1	B = 2	B = 3	B = 4	B = 5	B = 6	B = 7	B = 8	B = 9
A = 0	0	1	2	3	4	5	6	7	8	9
A = 1	10	11	12	13	14	15	16	17	18	19
A = 2	20	21	22	23	24	25	26	27	28	28
A = 3	28	29	29	29	30	30	30	31	31	31
A = 4	32	32	32	33	33	33	34	34	34	35
A = 5	35	35	35	35	35	36	36	36	36	36
A = 6	36	37	37	37	37	37	37	38	38	38
A = 7	38	38	38	39	39	39	39	39	39	40
A = 8	40	40	40	40	40	41	41	41	41	41
A = 9	41	41	41	41	41	41	41	42	42	42
A = 10	42	42	42	42	42	42	42	42	42	43
A = 11	43	43	43	43	43	43	43	43	43	43
A = 12	43	44	44	44	44	44	44	44	44	44
A = 13	44	44	44	45	45	45	45	45	45	45
A = 14	45	45	45	45	45	45	45	45	45	45
A = 15	45	45	45	45	45	45	45	46	46	46
A = 16	46	46	46	46	46	46	46	46	46	46
A = 17	46	46	46	46	46	46	46	46	46	46
A = 18	46	47	47	47	47	47	47	47	47	47
A = 19	47	47	47	47	47	47	47	47	47	47
A = 20	47	47	47	47	47	48	48	48	48	48
A = 21	48	48	48	48	48	48	48	48	48	48
A = 22	48	48	48	48	48	48	48	48	48	49
A = 23	49	49	49	49	49	49	49	49	49	49
A = 24	49	49	49	49	49	49	49	49	49	49
A = 25	49	49	49	0	0	0				

CUADRO 28

Tabla de suma logarítmica, latab[val]
val = (10 × A) + B

	B = 0	B = 1	B = 2	B = 3	B = 4	B = 5	B = 6	B = 7	B = 8	B = 9
A = 0	0×0040	0×003f	0×003e	0×003d	0×003c	0×003b	0×003a	0×0039	0×0038	0×0037
A = 1	0×0036	0×0035	0×0034	0×0034	0×0033	0×0032	0×0031	0×0030	0×002f	0×002f
A = 2	0×002e	0×002d	0×002c	0×002c	0×002b	0×002a	0×0029	0×0029	0×0028	0×0027
A = 3	0×0026	0×0026	0×0025	0×0024	0×0024	0×0023	0×0023	0×0022	0×0021	0×0021
A = 4	0×0020	0×0020	0×001f	0×001e	0×001e	0×001d	0×001d	0×001c	0×001c	0×001b
A = 5	0×001b	0×001a	0×001a	0×0019	0×0019	0×0018	0×0018	0×0017	0×0017	0×0016
A = 6	0×0016	0×0015	0×0015	0×0015	0×0014	0×0014	0×0013	0×0013	0×0013	0×0012
A = 7	0×0012	0×0012	0×0011	0×0011	0×0011	0×0010	0×0010	0×0010	0×000f	0×000f
A = 8	0×000f	0×000e	0×000e	0×000e	0×000d	0×000d	0×000d	0×000d	0×000c	0×000c
A = 9	0×000c	0×000c	0×000b	0×000b	0×000b	0×000b	0×000a	0×000a	0×000a	0×000a
A = 10	0×000a	0×0009	0×0009	0×0009	0×0009	0×0009	0×0008	0×0008	0×0008	0×0008
A = 11	0×0008	0×0008	0×0007	0×0007	0×0007	0×0007	0×0007	0×0007	0×0006	0×0006
A = 12	0×0006	0×0006	0×0006	0×0006	0×0006	0×0006	0×0005	0×0005	0×0005	0×0005
A = 13	0×0005	0×0005	0×0005	0×0005	0×0004	0×0004	0×0004	0×0004	0×0004	0×0004
A = 14	0×0004	0×0004	0×0004	0×0004	0×0004	0×0003	0×0003	0×0003	0×0003	0×0003
A = 15	0×0003	0×0003	0×0003	0×0003	0×0003	0×0003	0×0003	0×0003	0×0003	0×0002
A = 16	0×0002	0×0002	0×0002	0×0002	0×0002	0×0002	0×0002	0×0002	0×0002	0×0002
A = 17	0×0002	0×0002	0×0002	0×0002	0×0002	0×0002	0×0002	0×0002	0×0001	0×0001
A = 18	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001
A = 19	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001
A = 20	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001	0×0001
A = 21	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000
A = 22	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000
A = 23	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000
A = 24	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000
A = 25	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000	0×0000

CUADRO 29

Cuadro de umbral de audición, hth[fscod][band]

Número de banda	hth[0][band] **(f_s = 48 kHz)**	hth[1][band] **(f_s = 44,1 kHz)**	hth[2][band] **(f_s = 32 kHz)**
0	0×04d0	0×04f0	0×0580
1	0×04d0	0×04f0	0×0580
2	0×0440	0×0460	0×04b0
3	0×0400	0×0410	0×0450
4	0×03e0	0×03e0	0×0420
5	0×03c0	0×03d0	0×03f0
6	0×03b0	0×03c0	0×03e0
7	0×03b0	0×03b0	0×03d0
8	0×03a0	0×03b0	0×03c0
9	0×03a0	0×03a0	0×03b0
10	0×03a0	0×03a0	0×03b0
11	0×03a0	0×03a0	0×03b0
12	0×03a0	0×03a0	0×03a0
13	0×0390	0×03a0	0×03a0
14	0×0390	0×0390	0×03a0
15	0×0390	0×0390	0×03a0
16	0×0380	0×0390	0×03a0
17	0×0380	0×0380	0×03a0
18	0×0370	0×0380	0×03a0
19	0×0370	0×0380	0×03a0
20	0×0360	0×0370	0×0390
21	0×0360	0×0370	0×0390
22	0×0350	0×0360	0×0390
23	0×0350	0×0360	0×0390
24	0×0340	0×0350	0×0380

Número de banda	hth[0][band] **(f_s = 48 kHz)**	hth[1][band] **(f_s = 44,1 kHz)**	hth[2][band] **(f_s = 32 kHz)**
25	0×0340	0×0350	0×0380
26	0×0330	0×0340	0×0380
27	0×0320	0×0340	0×0370
28	0×0310	0×0320	0×0360
29	0×0300	0×0310	0×0350
30	0×02f0	0×0300	0×0340
31	0×02f0	0×02f0	0×0330
32	0×02f0	0×02f0	0×0320
33	0×02f0	0×02f0	0×0310
34	0×0300	0×02f0	0×0300
35	0×0310	0×0300	0×02f0
36	0×0340	0×0320	0×02f0
37	0×0390	0×0350	0×02f0
38	0×03e0	0×0390	0×0300
39	0×0420	0×03e0	0×0310
40	0×0460	0×0420	0×0330
41	0×0490	0×0450	0×0350
42	0×04a0	0×04a0	0×03c0
43	0×0460	0×0490	0×0410
44	0×0440	0×0460	0×0470
45	0×0440	0×0440	0×04a0
46	0×0520	0×0480	0×0460
47	0×0800	0×0630	0×0440
48	0×0840	0×0840	0×0450
49	0×0840	0×0840	0×04e0

CUADRO 30

Cuadro de puntero de atribución de bits, baptab[ ]

Dirección	baptab[address]	**Dirección**	baptab[address]
0	0	32	10
1	1	33	10
2	1	34	10
3	1	35	11
4	1	36	11
5	1	37	11
6	2	38	11
7	2	39	12
8	3	40	12
9	3	41	12
10	3	42	12
11	4	43	13
12	4	44	13
13	5	45	13
14	5	46	13
15	6	47	14
16	6	48	14
17	6	49	14
18	6	50	14
19	7	51	14
20	7	52	14
21	7	53	14
22	7	54	14
23	8	55	15
24	8	56	15
25	8	57	15
26	8	58	15
27	9	59	15
28	9	60	15
29	9	61	15
30	9	62	15
31	10	63	15

CUADRO 31

Niveles de cuantificación y bits de mantisa en funcion de bap

bap	Niveles de cuantificación	Bits de mantisa (bits de grupo / número del grupo)
0	0	0
1	3	1,67 (5/3)
2	5	2,33 (7/3)
3	7	3
4	11	3,5 (7/2)
5	15	4
6	32	5
7	64	6
8	128	7
9	256	8
10	512	9
11	1024	10
12	2048	11
13	4096	12
14	16384	14
15	65536	16

7.3 Cuantificación y decodificación de las mantisas

7.3.1 Visión de conjunto

Todas las mantisas se cuantifican con un nivel fijo de precisión indicado por el bap correspondiente. Para las mantisas cuantificadas con un número de niveles menor o igual que 15 se utiliza la cuantificación simétrica. Para las mantisas cuantificadas con más de 15 niveles se emplea la cuantificación asimétrica que es una representación convencional de complemento a 2.

Algunos valores de mantisa cuantificada se agrupan conjuntamente y se codifican para formar una palabra código común. En el caso de un cuantificador de tres niveles, se agrupan conjuntamente 3 valores cuantificados que se representan, en el tren de datos, mediante una palabra código de 5 bits. Para el cuantificador de 5 niveles se agrupan 3 valores cuantificados que se representan mediante una palabra código de 7 bits. En el caso del cuantificador de 11 niveles se agrupan 2 valores cuantificados que se representan por una palabra código de 7 bits.

En el codificador, cada coeficiente de la transformación (que siempre es < 1,0) se justifica por la izquierda desplazando hacia la izquierda su representación binaria un número de veces igual a su exponente (de 0 a 24 desplazamientos a la izquierda). Seguidamente, se cuantifica el coeficiente amplificado con un número de niveles igual al indicado por el bap correspondiente.

En el Cuadro 32 se indica el cuantificador que debe utilizarse con cada bap. Si un bap es igual a 0 no se envían bits para la mantisa. Para valores de bap iguales a 1, 2 y 4 (cuantificadores de 3, 5 y 11 niveles) se emplea el agrupamiento.

CUADRO 32

Correspondencia entre el bap y el cuantificador

bap	**Niveles del cuantificador**	**Tipo de cuantificación**	**Bits de mantisa (qntztab[bap]) (bits de grupo / número del grupo)**
0	0	Ninguno	0
1	3	Simétrica	1,67 (5/3)
2	5	Simétrica	2,33 (7/3)
3	7	Simétrica	3
4	11	Simétrica	3,5 (7/2)
5	15	Simétrica	4
6	32	Asimétrica	5
7	64	Asimétrica	6
8	128	Asimétrica	7
9	256	Asimétrica	8
10	512	Asimétrica	9
11	1024	Asimétrica	10
12	2048	Asimétrica	11
13	4096	Asimétrica	12
14	16384	Asimétrica	14
15	65536	Asimétrica	16

Durante el proceso de decodificación, el tren de datos de las mantisas se transforma en una única mantisa de longitud variable, entreverada con grupos que representan codificaciones combinadas de tripletas o pares de mantisas. En el tren de bits las mantisas de cada conjunto de exponentes se clasifican en orden creciente de frecuencias. Sin embargo, se producen agrupaciones en la posición de la primera mantisa contenida en el grupo. No se desempaqueta nada del tren de bits para las mantisas subsiguientes del grupo.

7.3.2 Desarrollo de las mantisas para la cuantificación asimétrica ($6 \leq$ bap ≤ 15)

Para los valores de fila del puntero de atribución de bits, $6 \leq \text{bap} \leq 15$, se utiliza cuantificación fraccional asimétrica de complemento a 2. Cada mantisa, junto con su exponente, constituye la representación en punto flotante de un coeficiente de la transformación. Se considera que el punto decimal está a la izquierda del MSB, por lo que la palabra de mantisa representa la gama de:

$$(1{,}0 - 2^{-\text{qntztab[bap]}-1}) \text{ a } -1{,}0.$$

Del tren de bits se extrae el número k de mantisa cuya longitud es qntztab[bap[k]]. Se efectúa la conversión a una representación de punto fijo desplazando la mantisa en una cantidad igual a su exponente. El proceso se representa mediante la siguiente expresión:

```
transform_coefficient[k] = mantissa[k] >> exponent[k];
```

Para mantisas cuantificadas asimétricamente no se realiza ninguna agrupación.

7.3.3 Desarrollo de las mantisas para la cuantificación simétrica (1 ≤ bap ≤ 5)

Para valores de bap de 1 a 5 (1 ≤ bap ≤ 5), las mantisas se representan mediante valores codificados. Los valores codificados se convierten a palabras binarias normalizadas fraccionales de complemento a 2 mediante un proceso de búsqueda en tabla. El número de bits indicado por un bap de mantisa se extrae del tren de bits y se justifica por la derecha. El valor codificado se maneja como índice de tabla, empleándose para la búsqueda del valor de la mantisa. El valor de la mantisa resultante se desplaza a la derecha en un número igual al exponente correspondiente para generar el valor del coeficiente de la transformación.

```
transform_coefficient[k] = quantization_table[mantissa_code[k]] >> exponent[k];
```

En los Cuadros 33 a 37, se representa la correspondencia entre el valor de la mantisa codificado y el valor de la mantisa real.

7.3.4 Adición de ruido para mantisas del bit 0 (bap = 0)

El decodificador AC-3 utiliza valores de ruido aleatorio (dither) en lugar de valores cuantificados cuando el número de bits atribuido a la mantisa es 0 (bap = 0). El uso del valor aleatorio está condicionado por el valor de dithflag. Cuando el valor de dithflag es 1, se utiliza el valor de ruido aleatorio. Cuando el valor de dithflag es 0, se emplea el valor 0 verdadero. Para cada canal hay una variable dithflag. La adición de ruido se aplica después de la extracción de los canales individuales del canal de acoplamiento. De esta forma, el ruido aplicado a las frecuencias superiores de cada canal está incorrelado.

Para generar valores de ruido aleatorio puede emplearse cualquier secuencia aleatoria razonable. La longitud de la palabra de los valores de ruido aleatorio no es crítica. Bastan 8 bits. Para la graduación óptima de las palabras de ruido aleatorio se toma una distribución uniforme de valores comprendidos entre –1 y +1, afectándolos por un factor de escala igual a 0,707, lo que produce una distribución uniforme entre –0,707 y +0,707. Un valor escalar de 0,75 es lo suficientemente aproximado para considerarlo óptimo. Asimismo resulta aceptable un valor escalar igual a 0,5 (distribución uniforme entre –0,5 y +0,5).

Una vez asignado un valor de ruido aleatorio a una mantisa, ésta se desplaza a la derecha según su exponente para generar el coeficiente de la transformación correspondiente.

```
transform_coefficient[k] = scaled_dither_value >> exponent[k];
```

CUADRO 33

Cuantificación (3 niveles) bap = 1

Código de mantisa	Valor de mantisa
0	–2./3
1	0
2	2./3

CUADRO 34

Cuantificación (5 niveles) bap = 2

Código de mantisa	Valor de mantisa
0	–4./5
1	–2./5
2	0
3	2./5
4	4./5

CUADRO 35

Cuantificación (7 niveles) bap = 3

Código de mantisa	Valor de mantisa
0	–6./7
1	–4./7
2	–2./7
3	0
4	2./7
5	4./7
6	6./7

CUADRO 36

Cuantificación (11 niveles) bap = 4

Código de mantisa	Valor de mantisa
0	–10./11
1	–8./11
2	–6./11
3	–4./11
4	–2./11
5	0
6	2./11
7	4./11
8	6./11
9	8./11
10	10./11

CUADRO 37

Cuantificación (15 niveles) bap = 5

Código de mantisa	Valor de mantisa
0	–14./15
1	–12./15
2	–10./15
3	–8./15
4	–6./15
5	–4./15
6	–2./15
7	0
8	2./15
9	4./15
10	6./15
11	8./15
12	10./15
13	12./15
14	14./15

7.3.5 Desagrupación de las mantisas

En el caso en que bap = 1, 2 ó 4, se comprimen aún más los valores de la mantisa cuantificada combinando palabras de 3 niveles y palabras de 5 niveles en grupos separados que representan tripletas de mantisas y palabras de 11 niveles en grupos que representan pares de mantisas. Los grupos se rellenan en el orden en que se procesan las mantisas. Si el número de mantisas de un juego de exponentes no rellena un número entero de grupos, se comparten los grupos entre conjuntos de exponentes. El siguiente conjunto de exponentes del bloque continua rellenando los grupos parciales. Si el número total de las palabras derivadas de los coeficientes de la transformación cuantificada de 3 ó 5 niveles, para cada uno de los casos, no es divisible por 3 o si las palabras de 11 niveles no son divisibles por 2, se añaden a los grupos finales de un bloque mantisas ficticias para completar el grupo compuesto. El decodificador desecha los elementos ficticios. Los grupos se extraen del tren de bits utilizando la longitud obtenida del bap. Las mantisas cuantificadas de 3 niveles (bap = 1), se agrupan en tripletas de 5 bits cada una. Las mantisas cuantificadas de 5 niveles (bap = 2), se agrupan en tripletas de 7 bits cada una. Las mantisas cuantificadas de 11 niveles (bap = 4), se agrupan en pares de 7 bits cada uno.

Ecuaciones del codificador

```
bap = 1:
    group_code = 9 * mantissa_code[a] + 3 * mantissa_code[b] + mantissa_code[c];
bap = 2:
    group_code = 25 * mantissa_code[a] + 5 * mantissa_code[b] + mantissa_code[c];
bap = 4:
    group_code = 11 * mantissa_code[a] + mantissa_code[b];
```

Ecuaciones del decodificador

```
bap = 1:
    mantissa_code[a] = truncate (group_code / 9);
    mantissa_code[b] = truncate ((group_code % 9) / 3 );
    mantissa_code[c] = (group_code % 9) % 3;

bap = 2:
    mantissa_code[a] = truncate (group_code / 25);
    mantissa_code[b] = truncate ((group_code % 25) / 5 );
    mantissa_code[c] = (group_code % 25) % 5;

bap = 4:
    mantissa_code[a] = truncate (group_code / 11);
    mantissa_code[b] = group_code % 11;
```

donde la mantisa a viene antes que la mantisa b, que viene antes que la mantisa c.

7.4 Acoplamiento del canal

7.4.1 Visión de conjunto

Cuando está activado, se realiza el acoplamiento de canal en la codificación promediando los coeficientes de la transformación a través de los canales incluidos en el canal de acoplamiento. Cada canal acoplado tiene un conjunto único de coordenadas de acoplamiento que se utilizan para preservar las envolventes de alta frecuencia de los canales originales. El proceso de acoplamiento se ejecuta por encima de una frecuencia de acoplamiento definida por el valor cplbegf.

El decodificador convierte el canal de acoplamiento en canales individuales mediante la multiplicación de los valores de los coeficientes de la transformada del canal acoplado por la coordenada de acoplamiento para ese canal y esa sub-banda de frecuencias. En el caso del modo 2/0 existe una etapa de procesamiento adicional. Si el bit phsflginu = 1 ó permanece el estado equivalente desde el bloque anterior, se envían bits de restauración de la fase en el tren de bits mediante bits de bandera de fase. Los bits de bandera de fase representan las sub-bandas de acoplamiento en un orden de frecuencias ascendente. Si, para una sub-banda determinada, el bit de bandera de fase es 1, se niega la totalidad de los coeficientes de la transformada del canal derecho dentro de esa sub-banda acoplada tras su modificación por la coordenada de acoplamiento, pero antes de la transformación inversa.

7.4.2 Estructura de sub-bandas para el acoplamiento

Los coeficientes de la transformada (tc), números 37 a 252, se agrupan en 18 sub-bandas de 12 coeficientes cada una, como se indica en el Cuadro 38. El parámetro cplbegf indica el número de la sub-banda de acoplamiento que debe incluirse en primer término en el proceso de acoplamiento. Por debajo de la frecuencia (o número del coeficiente de la transformada) indicada por cplbegf, todos los canales se codifican independientemente. Por encima de la frecuencia indicada por cplbegf, los canales incluidos en el proceso de acoplamiento (chincpl[ch] = 1) comparten el canal de acoplamiento común hasta la frecuencia (o número tc) indicada por cplendf. El canal de acoplamiento se codifica hasta la frecuencia (o número tc) indicada por cplendf, que indica la última sub-banda de acoplamiento codificada. Se interpreta el parámetro cplendf añadiendo 2 a su valor, de forma que la última sub-banda codificada puede variar entre 2 y 17.

Las sub-bandas de acoplamiento se combinan en bandas de acoplamiento para las que se generan coordenadas de acoplamiento (y se incluyen en el tren de bits). El parámetro cplbndstrc[sbnd] indica la estructura de la banda de acoplamiento. Cada bit de la fila cplbndstrc[] expresa si la

sub-banda indicada por el índice está combinada en la banda de acoplamiento anterior (de frecuencia más baja). En consecuencia, las bandas de acoplamiento se generan a partir de números enteros de sub-bandas de acoplamiento (véase el § 5.4.3.13).

CUADRO 38

Sub-bandas de acoplamiento

N.° de la sub-banda de acoplamiento	N.° tc bajo	N.° tc alto	lf de corte (kHz) @ f_s = 48 kHz	hf de corte (kHz) @ f_s = 48 kHz	lf de corte (kHz) @ f_s = 44,1 kHz	hf de corte (kHz) @ f_s = 44,1 kHz
0	37	48	3,42	4,55	3,14	4,18
1	49	60	4,55	5,67	4,18	5,21
2	61	72	5,67	6,80	5,21	6,24
3	73	84	6,80	7,92	6,24	7,28
4	85	96	7,92	9,05	7,28	8,31
5	97	108	9,05	10,17	8,31	9,35
6	109	120	10,17	11,30	9,35	10,38
7	121	132	11,30	12,42	10,38	11,41
8	133	144	12,42	13,55	11,41	12,45
9	145	156	13,55	14,67	12,45	13,48
10	157	168	14,67	15,80	13,48	14,51
11	169	180	15,80	16,92	14,51	15,55
12	181	192	16,92	18,05	15,55	16,58
13	193	204	18,05	19,17	16,58	17,61
14	205	216	19,17	20,30	17,61	18,65
15	217	228	20,30	21,42	18,65	19,68
16	229	240	21,42	22,55	19,68	20,71
17	241	252	22,55	23,67	20,71	21,75

f_s: frecuencia de muestreo.

NOTA 1 – A la velocidad de muestreo de 32 kHz las gamas de frecuencia de las sub-bandas son iguales a 2/3 de los valores de las correspondientes a 48 kHz.

7.4.3 Formato de las coordenadas de acoplamiento

Existen coordenadas de acoplamiento para cada banda de acoplamiento [bnd] en cada canal [ch] acoplado (chincp[ch]==1). Las coordenadas de acoplamiento se envían en un formato de coma flotante. El exponente se envía como un valor de 4 bits (cplcoexp[ch][bnd]) que indica el número de desplazamientos hacia la derecha que se deben aplicar al valor de mantisa fraccional. Las mantisas se transmiten como valores de 4 bits (cplcomant[ch][bnd]) a las cuales se les debe aplicar un factor de escala apropiado antes de utilizarlas. Las mantisas son valores sin signo, de modo que no se utiliza un bit de signo. Salvo en el caso límite, donde el valor del exponente = 15, se sabe que el valor de la mantisa se halla entre 0,5 y 1,0. Por lo tanto, cuando el valor del exponente es < 15, el bit más significativo de la mantisa siempre es igual a «1» y no se transmite; se transmiten los 4 bits siguientes de la mantisa. Esto proporciona un bit de resolución adicional. Cuando el valor del exponente = 15, el valor de la mantisa se obtiene dividiendo el valor de 4 bits de cplcomant por 16. Cuando el valor del exponente es < 15, el valor de la mantisa se obtiene sumando 16 al valor de 4 bits de cplcomant y dividiendo después la suma por 32.

La gama dinámica de las coordenadas de acoplamiento se aumenta por encima de lo que puede proporcionar el exponente de 4 bits utilizando una coordenada de acoplamiento principal de 2 bits por canal (mstrcplco[ch]) que se emplea para abarcar todas las coordenadas de acoplamiento dentro de ese canal. Los valores del exponente para cada canal se aumentan en 3 veces el valor de mstrcplco que se aplica a ese canal. Esto aumenta la gama dinámica de las coordenadas de acoplamiento en 54 dB adicionales.

El siguiente pseudocódigo indica cómo generar la coordenada de acoplamiento (cplco) para cada banda de acoplamiento [bnd] en cada canal [ch].

Pseudocódigo

```
if (cplcoexp[ch, bnd] == 15)
{
    cplco_temp[ch, bnd] = cplcomant[ch, bnd] / 16 ;
}
else
{
    cplco_temp[ch, bnd] = (cplcomant[ch, bnd] + 16) / 32 ;
}
cplco[ch, bnd] = cplco_temp[ch, bnd]>> (cplcoexp[ch, bnd] + 3 * mstrcplco[ch]) ;
```

Utilizando la disposición cplbndstrc[], los valores de las coordenadas de acoplamiento que se aplican a las bandas de acoplamiento son convertidos (duplicando los valores indicador por valores de «1» en cplbandstrc[]) en valores que se aplican a las sub-bandas de acoplamiento.

Las mantisas de cada canal se reconstruyen a partir del canal acoplado como sigue:

Pseudocódigo

```
for (sbnd = cplbegf; sbnd < 3 + cplendf; sbnd++)
{
    for (bin = 0; bin < 12; bin++)
    {
        chmant[ch, sbnd*12+bin+37] = cplmant[sbnd*12+bin+37] * cplco[ch, sbnd] * 8;
    }
}
```

7.5 Rematrización

7.5.1 Visión de conjunto

La rematrización en AC-3 es una técnica de combinación de canales en la que se codifican las sumas y diferencias de canales altamente correlados en vez de los propios canales originales. Es decir en vez de empaquetar y codificar los canales izquierdo (left) y derecho (right) con un codificador de dos canales se construye:

left' = 0.5*(left + right);

right' = 0.5*(left – right);

Las operaciones usuales de cuantificación y empaquetamiento de datos se ejecutan seguidamente sobre left' y right'. Obviamente, si la señal estereofónica original fuera idéntica en ambos canales (por ejemplo señal monofónica de dos canales) esta técnica produciría una señal left' idéntica a los canales izquierdo (left) y derecho originales y una señal right' que es idénticamente nula. En consecuencia, puede codificarse el canal right' con muy pocos bits y aumentar la precisión en el más importante canal left'.

Esta técnica es esencialmente importante para mantener la compatibilidad ambiente Dolby. Para convencerse considérese una señal origen monofónica de dos canales como la indicada anteriormente. Un decodificador lógico con procesamiento Dolby tratará de encaminar toda la información en fase hacia el canal central y toda la información fuera de fase hacia el canal ambiente. Si no se ha activado la rematrización, el decodificador lógico de procesamiento recibirá las siguientes señales:

Received left = left + QN1;

Received right = right + QN2;

donde QN1 y QN2 son secuencias de ruido de cuantificación independientes (esto es incorreladas), que corresponden a la cuantificación del algoritmo de codificación AC-3 y dependen del programa. Entonces el decodificador lógico de procesamiento construirá los canales central y de ambiente como sigue:

center = 0.5*(left + QN1) + 0.5*(right + QN2);

surround = 0.5*(left + QN1) – 0.5*(right + QN2); /* despreciando el desplazamiento de fase de 90° */

En el caso del canal central se suman QN1 y QN2, aunque permanece enmascarado por la señal dominante izquierda + derecha (left + right) . Sin embargo en el canal ambiente izquierda – derecha se iguala 0, por lo que los locutores de ambiente han de reproducir la diferencia en secuencias de ruido de cuantificación (QN1 – QN2).

Si la rematrización del canal está activa, los canales central y de ambiente se reproducirán más simplemente como sigue:

center = left' + QN1;
surround = right' + QN2;

En este caso, el ruido de cuantificación en el canal ambiente QN2 tiene un nivel mucho menor y está enmascarado por la señal diferencia right'.

7.5.2 Definiciones de bandas de frecuencias

En AC-3, la rematrización se ejecuta independientemente en bandas de frecuencias separadas. Hay 4 bandas cuyas fronteras mutuas dependen de la información de acoplamiento. Las fronteras se especifican mediante un número bin de coeficiente, variando esas fronteras de las bandas de frecuencias de rematrización correspondientes con la frecuencia de muestreo. En los Cuadros que siguen, se muestran las frecuencias de la banda de rematrización para velocidades de muestreo de 48 kHz y 44,1 kHz. Para la velocidad de muestreo de 32 kHz las frecuencias de la banda de rematrización son iguales a 2/3 de los valores correspondientes a 48 kHz.

7.5.2.1 Acoplamiento desactivado

Si no se utiliza el acoplamiento (cplinu=0) hay 4 bandas de rematrización (nrematbd=4).

CUADRO 39

Cuadro A de bandas de rematrización

Número de la banda	Número del coeficiente inferior	Número del coeficiente superior	Frecuencia baja (kHz) f_s = 48 kHz	Frecuencia alta (kHz) f_s = 48 kHz	Frecuencia baja (kHz) f_s = 44,1 kHz	Frecuencia alta (kHz) f_s = 44,1 kHz
0	13	24	1,17	2,30	1,08	2,11
1	25	36	2,30	3,42	2,11	3,14
2	37	60	3,42	5,67	3,14	5,21
3	61	252	5,67	23,67	5,21	21,75

7.5.2.2 Acoplamiento activado, cplbegf > 2

Si se utiliza el acoplamiento (cplinu = 1) y cplbegf > 2, hay 4 bandas de rematrización (nrematbd = 4). La última (cuarta) banda de rematrización termina en el punto en el que comienza el acoplamiento.

CUADRO 40

Cuadro B de bandas de rematrización

Número de la banda	Número del coeficiente inferior	Número del coeficiente superior	Frecuencia baja (kHz) f_s = 48 kHz	Frecuencia alta (kHz) f_s = 48 kHz	Frecuencia baja (kHz) f_s = 44,1 kHz	Frecuencia alta (kHz) f_s = 44,1 kHz
0	13	24	1,17	2,30	1,08	2,11
1	25	36	2,30	3,42	2,11	3,14
2	37	60	3,42	5,67	3,14	5,21
3	61	A	5,67	B	5,21	C

A = 36 + cplbegf*12 B = (A + 1/2)*0,09375 kHz C = (A + 1/2)*0,08613 kHz

7.5.2.3 Acoplamiento activado, 2 ≥ cplbegf > 0

Si se utiliza el acoplamiento (cplinu = 1) y 2 ≥ cplbegf > 0 hay tres bandas de rematrización (nrematbd = 3). La última (tercera) banda de rematrización termina en el punto donde comienza el acoplamiento.

CUADRO 41

Cuadro C de bandas de rematrización

Número de la banda	Número del coeficiente inferior	Número del coeficiente superior	Frecuencia baja (kHz) f_s = 48 kHz	Frecuencia alta (kHz) f_s = 48 kHz	Frecuencia baja (kHz) f_s = 44,1 kHz	Frecuencia alta (kHz) f_s = 44,1 kHz
0	13	24	1,17	2,30	1,08	2,11
1	25	36	2,30	3,42	2,11	3,14
2	37	A	3,42	B	3,14	C

A = 36 + cplbegf*12 B = (A + 1/2)*0,09375 kHz C = (A + 1/2)*0,08613 kHz

7.5.2.4 Acoplamiento activado, cplbegf = 0

Si se utiliza acoplamiento (cplinu = 1) y cplbegf = 0, hay 2 bandas de rematrización (nrematbd = 2).

CUADRO 42

Cuadro D de bandas de rematrización

Número de la banda	Número del coeficiente inferior	Número del coeficiente superior	Frecuencia baja (kHz) f_s = 48 kHz	Frecuencia alta (kHz) f_s = 48 kHz	Frecuencia baja (kHz) f_s = 44,1 kHz	Frecuencia alta (kHz) f_s = 44,1 kHz
0	13	24	1,17	2,30	1,08	2,11
1	25	36	2,30	3,42	2,11	3,14

7.5.3 Técnica de codificación

Si se ha seleccionado el modo 2/0, el codificador utiliza la rematrización. Los cuadrados de los coeficientes de la transformada se suman en las bandas de frecuencias de rematrización definidas anteriormente para las combinaciones L, R, L + R, L – R.

Pseudocódigo

```
if(minimum sum for a rematrixing sub-band n is L or R)
{
      the variable rematflg[n] = 0;
      transmitted left = input L;
      transmitted right = input R;
}
if(minimum sum for a rematrixing sub-band n is L+R or L-R)
{
      the variable rematflg[n] = 1;
      transmitted left = 0.5* input (L+R);
      transmitted right = 0.5* input (L-R);
}
```

La selección de una combinación matricial se efectúa bloque a bloque. Las operaciones de procesamiento restantes que efectúa el codificador sobre los canales transmitidos izquierdo y derecho son idénticas cualquiera que sea el valor 0 ó 1 de las banderas de rematrización.

7.5.4 Técnica de decodificación

Para cada banda de rematrización, se envía un único bit (bandera de rematrización) en el tren de datos a fin de indicar si para esa banda se han rematrizado o no los dos canales. Si el bit es 0, no se requiere operación ulterior. Si el bit es 1 el decodificador AC-3 ejecuta la siguiente operación para restaurar los canales individuales:

left(band n) = received left(band n) + received right(band n);

right(band n) = received left(band n) – received right(band n);

Obsérvese que si no se emplea el acoplamiento, los dos canales pueden tener anchuras de bandas diferentes. La rematrización, como tal, se aplica únicamente a la anchura de banda más baja de los dos canales. Independientemente de la anchura de banda real, en el tren de datos se envían las 4 banderas de rematrización (supuesto activado el bit de estrategia de rematrización).

7.6 Normalización del diálogo

La sintaxis AC-3 proporciona elementos que permiten al tren de bits codificado satisfacer a los oyentes en numerosas situaciones diferentes. El elemento dialnorm permite la reproducción uniforme de diálogos hablados al decodificar cualquier tren de bits AC-3.

7.6.1 Visión de conjunto

Cuando se reproduce audio procedente de fuentes diferentes, la sonoridad aparente varía a menudo de una fuente a otra. Las distintas fuentes de una señal audio pueden ser segmentos de programas diferentes durante una emisión de radiodifusión (por ejemplo sonido de películas frente a un mensaje comercial), canales de radiodifusión diferentes o medios distintos (disco o cinta magnética). La tecnología de codificación AC-3 resuelve este problema codificando de forma explícita una indicación de la sonoridad en el tren de bits AC-3.

Se utiliza como referencia el nivel subjetivo de un diálogo hablado normal. La palabra de normalización de diálogo de 5 bits dialnorm, contenida en el BSI es una indicación del volumen subjetivo del diálogo hablado normal con relación al 100% digital. El valor de 5 bits se interpreta como un entero sin signo (se transmite en primer lugar el bit más significativo) con una gama de valores posibles de 1 a 31. El entero sin signo indica el gálibo en dB por encima del nivel de diálogo subjetivo. Este valor puede también interpretarse como una indicación del número de dB en que el nivel de diálogo subjetivo es inferior al 100% digital.

El decodificador AC-3 no utiliza directamente el valor dialnorm. Tal valor es utilizado por la sección del sistema de reproducción del sonido que tiene a su cargo el ajuste del volumen de la reproducción, esto es el control del volumen del sistema. Dicho control se gradúa generalmente sobre la base del nivel de sonoridad deseado por el oyente o el nivel de presión sonora (SPL). El oyente ajusta un control de volumen que, en general, actúa directamente sobre la ganancia del sistema de reproducción. En AC-3, con el valor dialnorm, la ganancia del sistema de reproducción es una función del nivel de presión sonora de la reproducción deseado por los oyentes para el diálogo y del valor dialnorm que indica el nivel del diálogo en la señal de audio. Por consiguiente, el oyente puede ajustar con precisión el nivel de volumen de diálogo permaneciendo el nivel subjetivo de diálogo uniforme, con independencia del programa AC-3 decodificado.

Ejemplo:

> El oyente ajusta el control de volumen a 67 dB. (Con la normalización de diálogo AC-3 es posible calibrar un control de volumen del sistema directamente en nivel de presión sonora y la indicación será lo suficientemente precisa para cualquier fuente de audio codificada con AC-3.) Se recibe un programa de entretenimiento de elevada calidad indicando el tren de bits AC-3 que el nivel de diálogo está a 25 dB por debajo del nivel digital 100%. El sistema de reproducción ajusta automáticamente su ganancia de forma que las señales digitales a plena escala se reproduzcan con un nivel de presión sonora de 92 dB. En consecuencia el diálogo hablado (25 dB más bajo) se reproducirá a 67 dB SPL.
>
> Si el programa de radiodifusión pasa a un mensaje publicitario con un nivel de diálogo a –15 dB con respecto al nivel digital del 100%. La ganancia del sistema se reduce automáticamente de forma que el nivel digital del 100% se reproduce ahora a 82 dB SPL. El diálogo del mensaje publicitario (15 dB más bajo) se reproduce a 67 dB SPL, como se deseaba.

A fin de que funcione el sistema del sistema de normalización de diálogo, el valor dialnorm debe transferirse del decodificador AC-3 al controlador de ganancia del sistema, de manera que dialnorm pueda interaccionar con el control de volumen ajustado por el oyente. Si la función de control de volumen de un sistema se ejecuta como una multiplicación digital interior al decodificador AC-3, el ajuste de volumen realizado por el oyente debe comunicarse al decodificador AC-3. Tal ajuste de volumen y el valor de dialnorm deben agruparse conjuntamente y combinarse a fin de ajustar la ganancia del sistema de reproducción final.

El ajuste del control de volumen del sistema no es función de AC-3. El tren de bits de AC-3 simplemente transporta información útil que permite realizar el control de volumen del sistema de una forma que elimine automáticamente variaciones de nivel no deseadas entre fuentes de programas. Es obligatorio utilizar el valor dialnorm y el ajuste de volumen seleccionado por el usuario para establecer la ganancia del sistema de reproducción.

7.7 Compresión de la gama dinámica

7.7.1 Control de la gama dinámica; dynrng, dynrng2

El elemento dynrng permite al proveedor de programas realizar de forma subjetiva la reducción de la gama dinámica para la mayoría de la audiencia prevista, a la vez que proporciona a miembros individuales de la audiencia la posibilidad de experimentar una gama mayor (o la totalidad) de la gama original.

7.7.1.1 Visión de conjunto

Un problema constante asociado a la entrega de programas de audio es el que los distintos miembros de una audiencia desean obtener volúmenes de gama dinámica diferentes. Los programas originales de alta calidad (tales como películas) generalmente se mezclan con una gama dinámica bastante amplia. Tomando como referencia el nivel de un diálogo, los sonidos de gran intensidad como las explosiones tienen, generalmente un nivel 20 dB superior, en tanto que los sonidos débiles como los susurros pueden estar a unos 50 dB por debajo. En muchas situaciones de escucha, no es conveniente admitir que el sonido sea muy intenso, por lo que deben comprimirse a un nivel inferior los sonidos de mayor volumen. Análogamente en muchas situaciones de escucha los sonidos muy débiles serían inaudibles, por lo que debe realzarse su nivel para poderlos oír. Como para la mayor parte de una audiencia será ventajosa la escucha con una gama dinámica de programa limitada, generalmente se comprimen las pistas de sonido mezcladas con una gama dinámica amplia, reduciéndose la gama mediante la atenuación del nivel de sonidos intensos y el realce del nivel de los sonidos débiles. Si bien esto satisface las necesidades de la mayoría de la audiencia, impide a otros oyentes experimentar la sensación del programa sonoro original como fue elaborado. La tecnología de codificación audio AC-3 solventa este problema permitiendo que en el tren de bits AC-3 se inserten valores de control de gama dinámica.

Los valores de control de gama dinámica, dynrng, indican la variación de ganancia que debe aplicarse al decodificador a fin de realizar la compresión de la gama dinámica. Cada valor dynrng puede indicar una variación de ganancia de ±24 dB. La sucesión de valores de dynrng constituye una señal de control de la compresión. Un codificador AC-3 (o un procesador de trenes de bits) generará la secuencia de valores dynrng. El decodificador AC-3 utiliza cada valor para modificar la ganancia de uno o más bloques de audio. Generalmente, los valores de dynrng expresan la reducción de la ganancia en las fases de señal más intensa y el incremento de la ganancia en las fases en que la señal es más débil. Para el oyente es deseable reducir los sonidos más intensos a un nivel similar al nivel del diálogo y realzar los sonidos más débiles también al nivel del diálogo. Generalmente, no se alterará la ganancia de los sonidos cuyo volumen es similar al del diálogo normal.

La compresión se aplica realmente a la señal de audio en el decodificador AC-3. La señal audio codificada tiene su gama dinámica completa. Se admite que el decodificador AC-3 desprecie (facultativamente, bajo control del usuario) los valores de dynrng del tren de bits. Esto provocará la reproducción de la señal audio con su gama dinámica completa. También es admisible (asimismo bajo control del oyente) que el decodificador utilice alguna fracción del valor de control dynrng y que emplee fracciones diferentes de valores positivos y negativos. En consecuencia, el decodificador AC-3 puede reproducir indistintamente una señal audio comprimida (como estaba previsto por el circuito de control de compresión en el codificador AC-3), una señal audio de gama dinámica completa o una señal audio con gama dinámica parcialmente comprimida con distintos grados de compresión para señales de alto nivel y señales de bajo nivel.

Ejemplo:

La pista sonora de una película se codifica con AC-3. La mezcla de programa original tiene un nivel de diálogo igual a –25 dB. Las explosiones alcanzan un nivel de cresta de fondo de escala igual a 0 dB. Algunos sonidos débiles previstos para su escucha por los oyentes están a 50 dB por debajo del nivel de diálogo (esto es, –75 dB). El codificador AC-3 genera una señal de control de compresión (sucesión de valores dynrng). Durante las porciones del programa de audio en que el nivel audio es mayor que el nivel de diálogo, los valores dynrng indican una ganancia negativa o una reducción de la ganancia. Para señales de 0 dB a fondo de escala (explosiones más intensas), se codifica en el dynrng una reducción de ganancia de –15 dB. Para señales muy débiles se codifica en el dynrng un aumento de ganancia de 20 dB.

Supongamos un oyente que desea reproducir esta pista sonora con un volumen reducido para no molestar a nadie y que quiere oir todo el contendido del programa previsto. Se permite al codificador AC-3 reproducir esta situación por defecto, que corresponde a la compresión total. El oyente ajusta el nivel de diálogo a 60 dB SPL. Las explosiones alcanzarán únicamente una intensidad de 70 dB (son 25 dB más intensas que el diálogo pero tienen aplicada una ganancia de –15 dB) y los sonidos débiles se reproducirán con un nivel igual a 30 dB SPL (se aplica una ganancia de 20 dB a su nivel original de 50 dB por debajo del nivel de diálogo). La gama dinámica de la señal reproducida será 70 dB – 30 dB = 40 dB.

Supongamos que la situación de escucha cambia y el oyente desea ahora realzar el nivel de reproducción del diálogo a 70 dB SPL, manteniendo el límite de la intensidad del programa. Se permitirá reproducir los sonidos débiles como en el caso anterior. El oyente ordena al decodificador AC-3 que continúe empleando los valores de dynrng que indican reducción de la ganancia, pero que atenúe en un factor de 1/2 los valores que expresan aumentos de la ganancia. Las explosiones se reproducirán aún a 10 dB por encima del nivel de diálogo que ahora es de 80 dB SPL. Los sonidos débiles se han realzado ahora en un nivel igual a 20 dB/2 = 10 dB. Por consiguiente se reproducirán a 40 dB por debajo del nivel de diálogo, es decir 30 dB SPL. La gama dinámica del sonido reproducido en este caso es 80 dB – 30 dB = 50 dB.

Si otro oyente desea obtener la gama dinámica completa original de la señal de audio, ajusta el nivel de diálogo reproducido a 75 dB SPL y ordena al decodificador AC-3 que ignore la señal de control de la gama dinámica. Para este oyente, los sonidos débiles se reproducen a 25 dB SPL y las explosiones alcanzan 100 dB SPL. La gama dinámica de la señal reproducida es igual a 100 dB – 25 dB = 75 dB. Esta reproducción es exactamente igual a la prevista por el productor del programa.

A fin de que este método de control de la gama dinámica sea eficaz, deberían utilizarlo todos los proveedores de programas. Como todos los organismos de radiodifusión desean proporcionar sus programas en la forma más idónea para su audiencia, casi todos esos organismos aplicarán compresión de gama dinámica a cualquier programa de audio que tenga una amplia gama dinámica. Tal compresión no es reversible, a menos que se realice mediante la técnica incluida en AC-3. Si los organismos de radiodifusión utilizan el sistema de control de gama dinámica incluido en AC-3, los

oyentes deberán poder ejercitar algún tipo de control sobre la gama dinámica de la señal reproducida. Los organismos de radiodifusión deben tener presente que las características de compresión que introducen en AC-3 serán percibidas, por defecto, por los oyentes. En consecuencia, el decodificador AC-3 deberá, por defecto, aplicar la característica de compresión indicada mediante los valores dynrng en el tren de datos. Los decodificadores AC-3 podrán facultativamente permitir que el oyente ejerza cierto grado de control empleando valores de dynrng, de manera que pueda seleccionar la reproducción del audio con una gama dinámica total o parcial.

7.7.1.2 Realización detallada

En el tren de datos AC-3 el campo dynrng tiene una longitud igual a 8 bits. En el caso en que acmod = 0 (modo 1 + 1 ó 2 canales completamente independientes) dynrng se aplica al primer canal (Ch1) y dynrng2 se aplica al segundo canal (Ch2). Seguidamente se describe dynrng. El elemento dynrng2 se maneja de forma idéntica. El valor dynrng puede estar presente en cualquier bloque de audio. Cuando el valor no está presente se utiliza el valor del bloque anterior, salvo en el caso del bloque 0. En este caso si no está presente un nuevo valor de dynrng deberá utilizarse el valor 0000 0000. El bit más significativo de dynrng (y de dynrng2) se transmite en primer lugar. Los primeros 3 bits expresan variaciones de ganancia en incrementos de 6,02 dB que pueden realizarse con una operación de desplazamiento aritmético. Los 5 bits siguientes indican variaciones lineales de ganancia y requieren una multiplicación de 6 bits. Se representarán los campos de 3 y 5 bits de dynrng como sigue:

$$X_0\ X_1\ X_2\ .\ Y_3\ Y_4\ Y_5\ Y_6\ Y_7$$

La significación de los valores X se describe del modo más sencillo considerando que X representa un entero con signo de 3 bits con valores comprendidos entre –4 y 3. En consecuencia la ganancia indicada por X es (X + 1) ∗ 6,02 dB. Esto se muestra con detalle en el Cuadro 43.

CUADRO 43

Significado de los 3 bits más significativos de dynrng

X_0	X_1	X_2	Valor entero	Ganancia indicada (dB)	Dezplazamientos aritméticos
0	1	1	3	+24,08	4 izquierda
0	1	0	2	+18,06	3 izquierda
0	0	1	1	+12,04	2 izquierda
0	0	0	0	+6,02	1 izquierda
1	1	1	–1	0	Ninguno
1	1	0	–2	–6,02	1 derecha
1	0	1	–3	–12,04	2 derecha
1	0	0	–4	–18,06	3 derecha

El valor de Y es una representación lineal de variaciones de ganancia de hasta –6 dB. Se considera que Y es un entero fraccional sin signo con un valor delantero igual a 1 ó: 0,1 Y_3 Y_4 Y_5 Y_6 Y_7 (base 2). Y puede representar valores comprendidos entre $0{,}111111_2$ (ó 63/64) y $0{,}100000_2$ (ó 1/2). Por lo tanto, Y puede representar variaciones de ganancia comprendidas entre –0,14 dB y –6,02 dB.

La combinación de valores de X e Y permite que dynrng exprese variaciones de ganancia comprendidas entre 24,08 – 0,14 = +23,94 dB, y –18,06 – 6 = –24,06 dB. El código de bits 0000 0000 expresa una ganancia de 0 dB (unidad).

Compresión parcial

Puede operarse con el valor dynrng a fin de hacer que represente una variación de ganancia que sea una fracción del valor original. A fin de modificar el grado de compresión que se aplicará, considérese que dynrng representa un número fraccional con signo, o:

$$X_0 \cdot X_1\ X_2\ Y_3\ Y_4\ Y_5\ Y_6\ Y_7$$

donde X_0 es el bit de signo y X_1 X_2 Y_3 Y_4 Y_5 Y_6 Y_7 constituye la fracción de 7 bits. Este número fraccional de 8 bits con signo puede multiplicarse por una fracción que indique la proporción de compresión original que debe aplicarse. Si se multiplica este valor por 1/2, la gama de compresión de ±24 dB se reducirá a ±12 dB. Tras la graduación multiplicativa se considera de nuevo que el resultado de 8 bits es de la forma original X_0 X_1 X_2 . Y_3 Y_4 Y_5 Y_6 Y_7 y se usa normalmente.

7.7.2 Compresión intensa, compr, compr2

El elemento compr permite al proveedor del programa (o radiodifusor) establecer una amplia reducción de la gama dinámica (compresión intensa) de forma que asegure que una submezcla monofónica no rebasará un cierto nivel de cresta. Puede ser conveniente presentar un programa de audio altamente comprimido en ciertas situaciones de escucha, tales como películas entregadas a una habitación de hotel o a un asiento de aeronave. La limitación del nivel de cresta es útil cuando, por ejemplo, una señal mezclada monofónica se aplica a un modulador RF y se desea evitar la sobremodulación.

7.7.2.1 Visión de conjunto

Algunos equipos que decodifican el tren de bits AC-3 necesitarán entregar el audio resultante a través de un enlace con gama dinámica muy restringida. Como ejemplo puede citarse el caso de un decodificador de señal de televisión que debe modular la imagen y el sonido recibidos en un canal RF a fin de proporcionar una señal utilizable por un receptor de televisión de precio módico. En esta situación es necesario restringir el nivel máximo de cresta de salida a un valor conocido con respecto al nivel de diálogo, a fin de evitar la sobremodulación. Durante la mayor parte del tiempo, la señal de control de gama dinámica dynrng proporcionará la reducción de ganancia adecuada de forma que el nivel de cresta absoluto permanecerá acotado. Sin embargo, como se ha previsto que el sistema de control de gama dinámica aplique una reducción subjetivamente agradable en la gama de los sonidos percibidos, no puede asegurarse que controle adecuadamente las crestas de señal instantáneas para evitar la sobremodulación.

A fin de permitir la limitación del nivel de cresta de la señal AC-3 decodificada, puede estar presente en el tren de datos de AC-3 una segunda señal de control denominada compr (compr2 para el Ch2 en el modo 1 + 1). Esta nueva señal de control deberá estar presente en todos los trenes de bits previstos para su recepción, por ejemplo, por un decodificador de receptor de televisión. La señal de control compr es similar a la señal de control dynrng en el sentido en que la utiliza el

decodificador para modificar el nivel de audio reproducido. La señal de control compr tiene una gama de control igual al doble de la correspondiente a dynrng (±48 dB en vez ±24 dB) con la mitad de la resolución (0,5 dB frente a 0,25 dB). Además, como la señal de control compr se presenta en BSI únicamente, tiene un tiempo de resolución igual a una trama AC-3 (32 ms) en vez de un bloque (5,3 ms).

Los equipos que exijan una limitación del nivel de audio de cresta deberán emplear el elemento compr en vez de dynrng cuando esté presente compr en BSI. Como durante la mayor parte del tiempo el empleo de dynrng evitará niveles de cresta intensos, el codificador AC-3 únicamente necesitará insertar compr en forma ocasional, es decir durante aquellos instantes en que el empleo de dynrng implique un nivel de cresta excesivo. Si se ha ordenado al decodificador que utilice compr y en una trama concreta este elemento no está presente, se utilizará en esa trama la señal de control dynrng.

En algunas aplicaciones de AC-3, algunos receptores desearán reproducir una gama dinámica muy restringida. En este caso, puede estar presente la señal de control compr en todo momento. Entonces el empleo de compr en vez de dynrng permitirá la reproducción del audio con una gama dinámica muy limitada. Esto podría ser conveniente, por ejemplo, en el caso de la entrega de audio a una habitación de hotel o a un asiento de aeronave.

7.7.2.2 Realización detallada

El campo compr del tren de datos AC-3 tiene una longitud de 8 bits. En el caso en que acmod = 0 (modo 1 + 1 ó 2 canales completamente independientes), se aplica compr al primer canal (Ch1) y se aplica compr2 al segundo canal (Ch2). Seguidamente se describe el elemento compr (para el canal Ch1) manejándose idénticamente compr2 (aplicado al canal Ch2).

Se transmite en primer lugar el bit más significativo. Los primeros 4 bits expresan variaciones de ganancia en incrementos de 6,02 dB, que pueden realizarse con una operación de desplazamiento aritmético. Los 4 bits siguientes indican variaciones lineales de ganancia y requieren una multiplicación de 5 bits. Se representarán los dos campos de 4 bits de compr como sigue:

$$X_0\ X_1\ X_2\ X_3 \cdot Y_4\ Y_5\ Y_6\ Y_7$$

El significado de los valores de X se describe del modo más sencillo considerando que X representa un entero con signo de 4 bits con valores comprendidos entre −8 y +7. En consecuencia la ganancia expresada por X es (X + 1) ∗ 6,02 dB. Esto se muestra con detalle en el Cuadro 44.

El valor de Y es una representación lineal de la variación de la ganancia de hasta −6 dB como máximo. Se considera que Y es un entero fraccional sin signo con un valor delantero igual a 1, o bien: $0{,}1\ Y_4\ Y_5\ Y_6\ Y_7$ (base 2). Y puede representar valores comprendidos entre $0{,}11111_2$ (ó 31/32) y $0{,}10000_2$ (ó 1/2). En consecuencia Y puede representar variaciones de ganancia entre −0,28 dB y −6,02 dB.

La combinación de valores de X e Y permite a compr expresar variaciones de ganancia desde 48,16 − 0,28 = +47,88 dB a −42,14 − 6 = −48,14 dB.

CUADRO 44

Significado de los 4 MSB de compr

X_0	X_1	X_2	X_3	Valor entero	Ganancia indicada (dB)	Dezplazamientos aritméticos
0	1	1	1	7	+48,16	8 izquierda
0	1	1	0	6	+42,14	7 izquierda
0	1	0	1	5	+36,12	6 izquierda
0	1	0	0	4	+30,10	5 izquierda
0	0	1	1	3	+24,08	4 izquierda
0	0	1	0	2	+18,06	3 izquierda
0	0	0	1	1	+12,04	2 izquierda
0	0	0	0	0	+6,02	1 izquierda
1	1	1	1	–1	0	Ninguno
1	1	1	0	–2	–6,02	1 derecha
1	1	0	1	–3	–12,04	2 derecha
1	1	0	0	–4	–18,06	3 derecha
1	0	1	1	–5	–24,08	4 derecha
1	0	1	0	–6	–30,10	5 derecha
1	0	0	1	–7	–36,12	6 derecha
1	0	0	0	–8	–42,14	7 derecha

7.8 Submezclado

En muchos sistemas de reproducción el número de altavoces no concuerda con el número de canales audio codificados. A fin de reproducir el programa audio completo se necesita realizar un submezclado. Es importante que se normalice el submezclado de forma que los proveedores de programa puedan tener confianza de cómo se reproducirán sus programas en sistemas con diversos números de altavoces. Con ecuaciones de submezclado normalizadas, los productores de programas pueden verificar cómo sonarán las versiones submezcladas de sus programas y producir las modificaciones necesarias de forma que se logren resultados aceptables para todos los oyentes. El proveedor del programa puede utilizar los elementos sintácticos cmixlev y smixlev a fin de modificar el equilibrio relativo de los canales central y de ambiente con respecto a los canales izquierdo y derecho.

El submezclado del canal lfe es facultativo. Un submezclado ideal debería permitir la reproducción del canal lfe con un nivel acústico de +10 dB con respecto a los canales izquierdo y derecho. Como la inclusión de este canal es facultativa, puede utilizarse en la práctica cualquier coeficiente de submezclado. Deben adoptarse precauciones para asegurar que no se saturan los altavoces por la intensidad total de bajas frecuencias del canal lfe.

7.8.1 Procedimiento general downmix

En el pseudocódigo que sigue se describe la forma de obtener coeficientes no normalizados downmix. En una realización práctica puede ser necesario normalizar los coeficientes downmix para prevenir cualquier posibilidad de sobrecarga. Se realiza la normalización atenuando por igual todos los coeficientes downmix, de forma que la suma de los coeficientes utilizados para crear cualquier canal de salida único nunca exceda de 1.

Pseudocódigo

```
downmix()
{
    if (acmod == 0) /* modo 1+1, dos canales monofónicos independientes presentes */
    {
        if (output_nfront == 1) /* 1 altavoz frontal (centro) */
        {
            if (dualmode == Chan 1) /* salida del canal 1 solicitada */
            {
                route left into center;
            }
            else if (dualmode == Chan 2) /* salida del canal 2 solicitada */
            {
                route right into center;
            }
            else
            {
                mix left into center with –6 dB gain;
                mix right into center with –6 dB gain;
            }
        }
        else if (output_nfront == 2) /* 2 altavoces frontales (izquierdo, derecho) */
        {
            if (dualmode == Stereo) /* salida de ambos canales monofónicos solicitada */
            {
                route left into left;
                route right into right;
            }
            else if (dualmode == Chan 1)
            {
                mix left into left with –3 dB gain;
                mix left into right with –3 dB gain;
            }
            else if (dualmode == Chan 2)
            {
                mix right into left with –3 dB gain;
                mix right into right with –3 dB gain;
            }
            else /* suma monofónica de ambos canales monofónicos solicitada */
            {
                mix left into left with –6 dB gain;
                mix right into left with –6 dB gain;
                mix left into right with –6 dB gain;
                mix right into right with –6 dB gain;
            }
        }
```

Pseudocódigo

```
            else /* output_nfront==3 */
            {
                if (dualmode == Stereo)
                {
                    route left into left;
                    route right into right;
                }
                else if (dualmode == Chan 1)
                {
                    route left into center;
                }
                else if (dualmode == Chan 2)
                {
                    route right into center;
                }
                else
                {
                    mix left into center with –6 dB gain;
                    mix right into center with –6 dB gain;
                }
            }
        }
        else /* acmod > 0 */
        {
            for i = { left, center, right, leftsur/monosur, rightsur }
            {
                if (exists(input_chan[i])) and (exists(output_chan[i]))
                {
                    route input_chan[i] into output_chan[i];
                }
            }
            if (output_mode == 2/0 Dolby Surround compatible)
            /* salida codificada de la matriz de 2 canales solicitada */
            {
                if (input_nfront != 2)
                {
                    mix center into left with –3 dB gain;
                    mix center into right with –3 dB gain;
                }
                if (input_nrear == 1)
                {
                    mix -mono surround into left with –3 dB gain;
                    mix mono surround into right with –3 dB gain;
                }
                else if (input_nrear == 2)
                {
                    mix -left surround into left with –3 dB gain;
                    mix -right surround into left with –3 dB gain;
                    mix left surround into right with –3 dB gain;
                    mix right surround into right with –3 dB gain;
                }
            }
            else if (output_mode == 1/0) /* center only */
            {
                if (input_nfront != 1)
                {
                    mix left into center with –3 dB gain;
                    mix right into center with –3 dB gain;
                }
```

Pseudocódigo

```
                if (input_nfront == 3)
                {
                        mix center into center using clev and –3 dB gain;
                }
                if (input_nrear == 1)
                {
                        mix mono surround into center using slev and –3 dB gain;
                }
                else if (input_nrear == 2)
                {
                        mix left surround into center using slev and –3 dB gain;
                        mix right surround into center using slev and –3 dB gain;
                }
        }
        else /* más que la salida central solicitada */
        {
                if (output_nfront == 2)
                {
                        if (input_nfront == 1)
                        {
                        mix center into left with –3 dB gain;
                        mix center into right with –3 dB gain;
                        }
                        else if (input_nfront == 3)
                        {
                        mix center into left using clev;
                        mix center into right using clev;
                        }
                }
                if (input_nrear == 1) /* canal único periférico codificado */
                {
                        if (output_nrear == 0) /* no hay altavoces periféricos */
                        {
                        mix mono surround into left with slev and –3 dB gain;
                        mix mono surround into right with slev and –3 dB gain;
                        }
                        else if (output_nrear == 2) /* dos canales con altavoces periféricos */
                        {
                        mix mono srnd into left surround with –3 dB gain;
                        mix mono srnd into right surround with –3 dB gain;
                        }
                }
                else if (input_nrear == 2) /* dos canales periféricos codificados */
                {
                        if (output_nrear == 0)
                        {
                        mix left surround into left using slev;
                        mix right surround into right using slev;
                        }
                        else if (output_nrear == 1) .
                        {
                        mix left srnd into mono surround with –3 dB gain;
                        mix right srnd into mono surround with –3 dB gain;
                        }
                }
        }
    }
}
```

Los coeficientes reales utilizados para el submezclado afectarán al nivel absoluto del canal central. Esto debe tenerse presente si se desea establecer el nivel de diálogo con una calibración de SPL absoluta.

7.8.2 Submezclado a dos canales

Supóngase que L, C, R, L_s, R_s representan los 5 canales discretos que hay que reducir por submezclado a 2 canales. En el caso de un solo canal ambiente (modos *n*/1), S representa ese canal ambiente. Conviene proporcionar 2 tipos de submezclado: submezclado a un par estereofónico ambiental codificado por matriz, L_tR_t, y el submezclado a una señal estereofónica convencional, L_0R_0. La señal estereofónica submezclada (L_0R_0 o L_tR_t) se puede mezclar otra vez para generar una señal monofónica, M, mediante la simple adición de los 2 canales. Si se combina la L_tR_t submezclada con la señal monofónica, se perderá la información de ambiente. Cuando se desea disponer de una señal monofónica es preferible usar L_0R_0 submezclada. Los coeficientes de submezclado deberán tener una exactitud relativa de al menos ±0,25 dB.

Antes de aplicar el factor de escala necesario para evitar el desbordamiento, las ecuaciones generales downmix 3/2 para una señal estereofónica L_0R_0 son:

$$L_0 = 1.0 * L + clev * C + slev * L_s;$$

$$R_0 = 1.0 * R + clev * C + slev * R_s;$$

Si, a continuación, L_0R_0 se combinan para reproducción monofónica, la ecuación efectiva submezclado monofónico será:

$$M = 1.0 * L + 2.0 * clev * C + 1.0 * R + slev * L_s + slev * R_s;$$

Si sólo está presente un canal ambiente S (modo 3/1), las ecuaciones de submezclado son:

$$L_0 = 1.0 * L + clev * C + 0.7 * slev * S;$$

$$R_0 = 1.0 * R + clev * C + 0.7 * slev * S;$$

$$M = 1.0 * L + 2.0 * clev * C + 1.0 * R + 1.4 * slev * S;$$

Los valores de clev y slev vienen indicados por los campos de bits cmixlev y surmixlev de los datos BSI, como se muestra en el Cuadro 4 y en el Cuadro 5, respectivamente.

Si los campos de bits cmixlev ó surmixlev indican el estado reservado (valor de 1 1), el decodificador deberá utilizar los valores de los coeficientes intermedios indicados por el valor del campo de bits 0 1. Si falta el canal central (modo 2/1 ó 2/2), pueden emplearse las mismas ecuaciones sin el término C. Si faltan los canales ambiente pueden emplearse las mismas ecuaciones sin los términos L_s, R_s o S.

Antes de aplicar el factor de escala necesario para evitar la saturación, las ecuaciones downmix 3/2 para una señal estereofónica L_tR_t son:

$$L_t = 1.0 * L + 0.707 * C – 0.707 * L_s – 0.707 * R_s;$$

$$R_t = 1.0 * R + 0.707 * C + 0.707 * L_s + 0.707 * R_s;$$

Si sólo está presente un canal ambiente de S (modo 3/1) las ecuaciones se transforman en:

$$L_t = 1.0\ L + 0.707\ C – 0.707\ S;$$

$$R_t = 1.0\ R + 0.707\ C + 0.707\ S;$$

Si falta el canal central (modo 2/2 ó 2/1), se descarta el término C.

Cuando todos los canales que contribuyen a una señal submezclada están a su nivel máximo se aplicará un factor de escala reductor a los coeficientes reales utilizados para evitar el desbordamiento aritmético. En cada modo de codificación de audio, hay un número diferente de canales que contribuyen al submezclado, por lo que se podrá utilizar un factor reductor diferente para evitar el desbordamiento. Para simplificar, se puede aplicar en todos los casos el factor reductor del caso más desfavorable, lo que minimiza el número de coeficientes requeridos. La aplicación de un factor de escala del caso más desfavorable se produce cuando clev y slev valen 0,707. En el caso de L_0R_0 downmix, la suma de los coeficientes sin factor reductor es 1 + 0,707 + 0,707 = 2,414, de forma que hay que multiplicar todos los coeficientes por 1/2,414 = 0,4143 (reducción de 7,65 dB por submezclado). En el caso de L_tR_t downmix, la suma de los coeficientes sin factor reductor es 1 + 0,707 + 0,707 + 0,707 = 3,121, de forma que hay que multiplicar todos los coeficientes por 1/3,121 = 0,3204 (reducción de 9,89 dB por submezclado). Los coeficientes así reducidos se convertirán a valores binarios con longitud de palabra limitada. Los coeficientes de 6 bits que se indican a continuación tienen exactitud suficiente.

Para realizar el downmix a 2 canales, L_0R_0, se necesitan valores de coeficientes reducidos (0,453) que correspondan a los valores de 1,0; 0,707; 0,596; 0,500; 0,354.

CUADRO 45

Coeficientes downmix ajustados para L_0R_0

Coeficiente no ajustado	Coeficiente ajustado	Coeficiente cuantificado con 6 bits	Ganancia (dB)	Ganancia relativa (dB)	Error del coeficiente (dB)
1,0	0,414	26/64	–7,8	0,0	–
0,707	0,293	18/64	–11,0	–3,2	–0,2
0,596	0,247	15/64	–12,6	–4,8	+0,3
0,500	0,207	13/64	–13,8	–6,0	0,0
0,354	0,147	9/64	–17,0	–9,2	–0,2

Para realizar downmix a 2 canales de L_tR_t, se necesitan valores de coeficientes ajustados (por 0,3204) que correspondan con los valores 1,0 y 0,707.

CUADRO 46

Coeficientes downmix ajustados para L_tR_t

Coeficiente no ajustado	Coeficiente ajustado	Coeficiente cuantificado con 6 bits	Ganancia (dB)	Ganancia relativa (dB)	Error del coeficiente (dB)
1,0	0,3204	20/64	–10,1	0,0	–
0,707	0,2265	14/64	–13,20	–3,1	–0,10

Si es necesario realizar un downmix a una señal monofónica, deberá aplicarse una graduación adicional de 1/2 a los coeficientes del downmix L_0R_0, para evitar la sobrecarga de la suma monofónica de $L_0 + R_0$.

7.9 Ecuaciones de la transformación y conmutación de bloques

7.9.1 Visión de conjunto

En cualquier sistema de codificación de audio basado en transformaciones de señal es fundamental la elección de la longitud del bloque de análisis. En el caso de señales de entrada cuyo espectro permanezca estacionario o varíe muy lentamente con el tiempo, lo más adecuado es el empleo de una longitud de la transformación grande. Una gran longitud de la transformación proporciona mayor resolución de frecuencia y en consecuencia una mejor calidad de la codificación de esas señales. Por otra parte, una longitud de la transformación reducida que proporciona mayor resolución temporal resulta más conveniente para señales que varían rápidamente con el tiempo. En consecuencia, para la selección de la longitud del bloque de la transformación debe analizarse la posible transacción entre resolución de tiempo y resolución de frecuencia.

El método tradicional de resolver este problema es seleccionar una longitud de la transformación única que proporcione el mejor equilibrio de calidad de codificación tanto para señales estacionarias como para señales dinámicas. El método AC-3 emplea una técnica más eficaz consistente en la adaptación de la resolución frecuencia/tiempo de la transformación en función de las características temporales y espectrales de la señal que se está procesando. Este enfoque es muy parecido al comportamiento conocido de la audición humana. En la codificación de transformada se realiza la adaptación mediante la conmutación de la longitud del bloque en función de la señal.

7.9.2 Técnica

En el procedimiento de conmutación de bloques de la transformación utilizado en AC-3 puede emplearse una longitud de bloque de 512 ó 256 muestras (equivalente a una resolución temporal de 10,7 ó 5,3 ms para la frecuencia de muestreo de 48 kHz). Los bloques normales tienen una longitud de 512 muestras. Cuando se transforma un bloque enventanado normal proporciona 256 coeficientes unívocos de la transformada en el dominio de la frecuencia. Se construyen bloques más cortos tomando el segmento de audio enventanado usual de 512 muestras y dividiéndolo en 2 segmentos de 256 muestras cada uno. La primera mitad de un bloque MDTC se transforma de una manera separada aunque idéntica a la segunda mitad de ese bloque. Cada mitad del bloque proporciona 128 coeficientes únicos y no nulos de la transformación, que representan frecuencias comprendidas entre 0 y $f_s/2$ con un total de 256. Esto es idéntico al número de coeficientes generados por un bloque único de 512 muestras, pero con el doble de resolución temporal. Los coeficientes de la transformación procedentes de los 2 semibloques se intercalan conjuntamente coeficiente a coeficiente para constituir un bloque único de 256 valores. Este bloque se cuantifica y se transmite de forma idéntica a un único bloque largo. Para la reconstrucción de la señal en el decodificador se aplica un procedimiento similar, imagen del anterior.

Los coeficientes de la transformación correspondientes a las 2 transformaciones de longitud 256 llegan al decodificador intercalados conjuntamente bin a bin. Esta secuencia intercalada contiene el mismo número de coeficientes de la transformación que el generado por una transformación única de 512 muestras. El decodificador procesa las secuencias intercaladas de forma idéntica a las secuencias no intercaladas, excepto durante la transformación inversa que se describe a continuación.

Antes de realizar la transformación de la señal audio del dominio del tiempo al de la frecuencia, el codificador efectúa un análisis de la naturaleza espectral y/o temporal de la señal de entrada y selecciona la longitud de bloque apropiada. Este análisis tiene lugar únicamente en el codificador, por lo que puede mejorarse sin alterar la base existente de decodificadores. En el tren de bits se inserta un código de un bit por canal y por bloque de transformación (blksw[ch]) que transporta la información de longitud (blksw[ch] = 0 ó 1 para 512 ó 256 muestras, respectivamente). El decodificador utiliza esta información para desformatizar el tren de bits, reconstruir los datos de mantisa y aplicar las ecuaciones de la transformación inversa apropiadas.

7.9.3 Realización del decodificador

La conmutación de bloques de la transformación TDAC se realiza en AC-3 mediante un ajuste de las ecuaciones convencionales de la transformación directa e inversa para la transformación de longitud 256. Pueden reutilizarse la misma ventana y tablas seno/coseno de la FFT empleadas para los bloques de 512 muestras para realizar la transformación inversa de bloques de 256 muestras. Sin embargo, el desplazamiento de fase de la multiplicación compleja pre y post FFT requiere 128 valores de tabla adicionales para la transformación conmutada de los bloques.

Como las filas de entrada y salida correspondientes a blksw[ch] = 1 tienen exactamente la mitad de la longitud que las correspondientes a blksw = 0, el tamaño de la memoria RAM de la transformación inversa y sus memorias tampón asociadas es el mismo con la conmutación de bloques que sin ella.

Los ajustes necesarios para realizar la transformación inversa de los bloques de 256 muestras son los siguientes:

– La fila de entrada contiene 128 en vez de 256 coeficientes.

– La IFFT pre y post-desplazamiento de fase emplea una tabla de cosenos diferente, lo que exige 128 valores de tabla adicionales (64 de coseno y 64 de seno).

– La IFFT compleja emplea 64 puntos en vez de 128. Puede utilizarse la misma tabla de cosenos FFT con submuestreo para recuperar únicamente los elementos pares.

– Los punteros de entrada correspondientes a la operación de post-enventanado de la IFFT se inicializan con direcciones de arranque distintas y funcionan en módulo 128 en vez de en módulo 256.

7.9.4 Ecuaciones de la transformación

7.9.4.1 Transformación IMDCT de 512 muestras

Seguidamente se describe la técnica utilizada para calcular la IMDCT para un bloque único de datos reales con una longitud N = 512 empleando una sola IFFT compleja de N/4 puntos con operaciones simples de pre y post-desplazamiento de fase. Son las ecuaciones de la transformación inversa utilizadas cuando la bandera blksw está puesta a 0 (lo que indica la ausencia de transitorios y transformaciones de 512 muestras).

Etapa 1: Definición de los coeficientes de la transformación MDCT = X[k], k = 0,1,...N/2–1

Etapa 2: Etapa de multiplicación compleja pre-IFFT

Cálculo del resultado de la multiplicación compleja de N/4 puntos, Z[k], k = 0,1,...N/4–1:

Pseudocódigo

```
for(k=0; k<N/4; k++)
{
    /* Z[k] = (X[N/2-2*k-1] + j * X[2*k]) * (xcos1[k] + j * xsin1[k]); */
    Z[k]=(X[N/2-2*k-1]*xcos1[k]-X[2*k]*xsin1[k])+j*(X[2*k]*xcos1[k]+X[N/2-2*k-1]*xsin1[k]);
}
```

donde:

```
xcos1[k] = –cos(2pi * (8*k+1)/(8*N));
xsin1[k] = –sin(2pi * (8*k+1)/(8*N));
```

Etapa 3: Etapa IFFT compleja

Cálculo de la IFFT compleja de N/4 puntos de Z[k] para generar la sucesión z[n] de valores complejos:

Pseudocódigo

```
for(n=0; n<N/4; n++)
{
    z[n] = 0;
    for(k=0; k<N/4; k++)
    {
        z[n] + = Z[k] * (cos(8*pi*k*n/N) + j * sin(8*pi*k*n/N));
    }
}
```

Etapa 4: Etapa de multiplicación compleja post-IFFT

Cálculo del resultado de la multiplicación compleja de N/4 puntos, y[n], n = 0,1,...N/4–1 como sigue:

Pseudocódigo

```
for(n=0; n<N/4; n++)
{
    /* y[n] = z[n] * (xcos1[n] + j * xsin1[n]); */
    y[n] = (zr[n] * xcos1[n] - zi[n] * xsin1[n]) + j * (zi[n] * xcos1[n] + zr[n] * xsin1[n]);
}
```

donde:

zr[n] = real(z[n]);
zi[n] = imag(z[n]);
xcos1[n] y xsin1[n] son las definidas en la etapa 2 anterior.

Etapa 5: Etapa de enventanado y desentrelazado

Cálculo de las muestras enventanadas en el dominio del tiempo x[n]:

Pseudocódigo

```
for(n=0; n<N/8; n++)
{
    x[2*n] = -yi[N/8+n] * w[2*n];
    x[2*n+1] = yr[N/8-n-1] * w[2*n+1];
    x[N/4+2*n] = -yr[n] * w[N/4+2*n];
    x[N/4+2*n+1] = yi[N/4-n-1] * w[N/4+2*n+1];
    x[N/2+2*n] = -yr[N/8+n] * w[N/2-2*n-1];
    x[N/2+2*n+1] = yi[N/8-n-1] * w[N/2-2*n-2];
    x[3*N/4+2*n] = yi[n] * w[N/4-2*n-1];
    x[3*N/4+2*n+1] = -yr[N/4-n-1] * w[N/4-2*n-2];
}
```

donde:

yr[n] = real(y[n]);
yi[n] = imag(y[n]);
w[n] es la secuencia de ventana de transformación (véase el Cuadro 47).

Etapa 6: Etapa de superposición y adición

Se produce la superposición entre la primera mitad del bloque enventanado y la segunda mitad del bloque anterior para generar muestras MIC (el factor de graduación 2 anula la graduación de gálibo realizada en el codificador).

Pseudocódigo

```
for(n=0; n<N/2; n++)
{
    pcm[n] = 2 * (x[n] + delay[n]);
    delay[n] = x[N/2+n);
}
```

Obsérvese que el tratamiento aritmético del procesamiento superposición/adición debe emplear aritmética de saturación para evitar la sobrecarga (enrrollamiento). Como la señal de salida está constituida por la señal original más un error de codificación es posible que la señal de salida rebase el nivel del 100% aunque la señal de entrada original tuviera un nivel menor o igual que el 100%.

7.9.4.2 Transformadas IMDCT de 256 muestras

En el caso en que blksw = 1 lo que indica la presencia de un transitorio y dos transformaciones de 256 muestras (el valor de N se mantiene igual a 512) deben utilizarse las ecuaciones que siguen para calcular las transformaciones inversas:

Etapa 1: Definición de los coeficientes de la transformación MDCT = X[k], k = 0,1,...N/2

Pseudocódigo

```
    for(k=0; k<N/4; k++)
    {
        X1[k] = X[2*k];
        X2[k] = X[2*k+1];
    }
```

Etapa 2: Etapa de multiplicación compleja pre-IFFT

Cálculo de los resultados de la multiplicación compleja de N/8 puntos Z1[k] y Z2[k], k = 0,1,...N/8-1.

Pseudocódigo

```
for(k=0; k<N/8; k++)
{
    /* Z1[k] = (X1[N/4-2*k-1] + j * X1[2*k]) * (xcos2[k] + j * xsin2[k]); */
    Z1[k]=(X1[N/4-2*k-1]*xcos2[k]-X1[2k]*xsin2[k])+j*(X1[2*k]*xcos2[k]+X1[N/4-2*k-1]*xsin2[k]);
    /* Z2[k] = (X2[N/4-2*k-1] + j * X2[2*k]) * (xcos2[k] + j * xsin2[k]); */
    Z2[k]=(X2[N/4-2*k-1]*xcos2[k]-X2[2*k]*xsin2[k])+j*(X2[2*k]*xcos2[k]+X2[N/4-2*k-1]*xsin2[k]);
}
```

donde:

xcos2[k] = –cos(2pi*(8*k+1)/(4*N)), xsin2(k) = –sin(2pi*(8*k+1)/(4*N))

Etapa 3: Etapa IFFT compleja

Se calculan las IFFT complejas de N/8 puntos de Z1[k] y Z2[k] para generar las secuencias complejas z1[n] y z2[n].

Pseudocódigo

```
for(n=0; n<N/8; n++)
{
    z1[n] = 0.;
    z2[n] = 0.;
    for(k=0; k<N/8; k++)
    {
        z1[n] + = Z1[k] * (cos(16*pi*k*n/N) + j * sin(16*pi*k*n/N));
        z2[n] + = Z2[k] * (cos(16*pi*k*n/N) + j * sin(16*pi*k*n/N));
    }
}
```

Etapa 4: Etapa de multiplicación compleja post-IFFT

Se calculan los resultados de las multiplicaciones complejas de N/8 puntos y1[n] e y2[n], n = 0,1,...N/8–1.

Pseudocódigo

```
for(n=0; n<N/8; n++)
{
    /* y1[n] = z1[n] * (xcos2[n] + j * xsin2[n]); */
    y1[n] = (zr1[n] * xcos2[n] - zi1[n] * xsin2[n]) + j * (zi1[n] * xcos2[n] + zr1[n] * xsin2[n]);
    /* y2[n] = z2[n] * (xcos2[n] + j * xsin2[n]); */
    y2[n] = (zr2[n] * xcos2[n] - zi2[n] * xsin2[n]) + j * (zi2[n] * xcos2[n] + zr2[n] * xsin2[n]);
}
```

donde:

zr1[n] = real(z1[n]);
zi1[n] = imag(z1[n]);
zr2[n] = real(z2[n]);
zi2[n] = imag(z2[n]);

y xcos2[n] y xsin2[n] son las mismas que en la etapa 2 anterior.

Etapa 5: Etapa de enventanado y desentrelazado

Se calculan las muestras x[n] enventanadas en el dominio del tiempo

Pseudocódigo

```
for(n=0; n<N/8; n++)
{
    x[2*n] = -yi1[n] * w[2*n];
    x[2*n+1] = yr1[N/8-n-1] * w[2*n+1];
    x[N/4+2*n] = -yr1[n] * w[N/4+2*n];
    x[N/4+2*n+1] = yi1[N/8-n-1] * w[N/4+2*n+1];
    x[N/2+2*n] = -yr2[n] * w[N/2-2*n-1];
    x[N/2+2*n+1] = yi2[N/8-n-1] * w[N/2-2*n-2];
    x[3N/4+2*n] = yi2[n] * w[N/4-2*n-1];
    x[3N/4+2*n+1] = -yr2[N/8-n-1] * w[N/4-2*n-2];
}
```

donde:

yr1[n] = real(y1[n]);
yi1[n] = imag(y1[n]);
yr2[n] = real(y2[n]);
yi2[n] = imag(y2[n]);

y w[n] es la secuencia ventana de la transformación (véase el Cuadro 47).

Etapa 6: Etapa de superposición y adición

Se realiza la superposición de la primera mitad del bloque enventanado con la segunda mitad del bloque anterior para generar muestras MIC (el factor de graduación 2 deshace la graduación de gálibo efectuada en el codificador).

Pseudocódigo

```
for(n=0; n<N/2; n++)
{
      pcm[n] = 2 * (x[n] + delay[n]);
      delay[n] = x[N/2+n];
}
```

Obsérvese que el tratamiento aritmético del procesamiento superposición/adición debe emplear aritmética de saturación para evitar la sobrecarga (enrrollamiento). Como la señal de salida consta de la señal original más el error de codificación, es posible que la señal de salida exceda el nivel del 100%, aun cuando el nivel de la señal de entrada original sea menor o igual que el 100%.

CUADRO 47

Secuencia de ventana de la transformación (w[addr]), donde addr = (10 ∗ A) + B

	B = 0	B = 1	B = 2	B = 3	B = 4	B = 5	B = 6	B = 7	B = 8	B = 9
A = 0	0,00014	0,00024	0,00037	0,00051	0,00067	0,00086	0,00107	0,00130	0,00157	0,00187
A = 1	0,00220	0,00256	0,00297	0,00341	0,00390	0,00443	0,00501	0,00564	0,00632	0,00706
A = 2	0,00785	0,00871	0,00962	0,01061	0,01166	0,01279	0,01399	0,01526	0,01662	0,01806
A = 3	0,01959	0,02121	0,02292	0,02472	0,02662	0,02863	0,03073	0,03294	0,03527	0,03770
A = 4	0,04025	0,04292	0,04571	0,04862	0,05165	0,05481	0,05810	0,06153	0,06508	0,06878
A = 5	0,07261	0,07658	0,08069	0,08495	0,08935	0,09389	0,09859	0,10343	0,10842	0,11356
A = 6	0,11885	0,12429	0,12988	0,13563	0,14152	0,14757	0,15376	0,16011	0,16661	0,17325
A = 7	0,18005	0,18699	0,19407	0,20130	0,20867	0,21618	0,22382	0,23161	0,23952	0,24757
A = 8	0,25574	0,26404	0,27246	0,28100	0,28965	0,29841	0,30729	0,31626	0,32533	0,33450
A = 9	0,34376	0,35311	0,36253	0,37204	0,38161	0,39126	0,40096	0,41072	0,42054	0,43040
A = 10	0,44030	0,45023	0,46020	0,47019	0,48020	0,49022	0,50025	0,51028	0,52031	0,53033
A = 11	0,54033	0,55031	0,56026	0,57019	0,58007	0,58991	0,59970	0,60944	0,61912	0,62873
A = 12	0,63827	0,64774	0,65713	0,66643	0,67564	0,68476	0,69377	0,70269	0,71150	0,72019
A = 13	0,72877	0,73723	0,74557	0,75378	0,76186	0,76981	0,77762	0,78530	0,79283	0,80022
A = 14	0,80747	0,81457	0,82151	0,82831	0,83496	0,84145	0,84779	0,85398	0,86001	0,86588
A = 15	0,87160	0,87716	0,88257	0,88782	0,89291	0,89785	0,90264	0,90728	0,91176	0,91610
A = 16	0,92028	0,92432	0,92822	0,93197	0,93558	0,93906	0,94240	0,94560	0,94867	0,95162
A = 17	0,95444	0,95713	0,95971	0,96217	0,96451	0,96674	0,96887	0,97089	0,97281	0,97463
A = 18	0,97635	0,97799	0,97953	0,98099	0,98236	0,98366	0,98488	0,98602	0,98710	0,98811
A = 19	0,98905	0,98994	0,99076	0,99153	0,99225	0,99291	0,99353	0,99411	0,99464	0,99513
A = 20	0,99558	0,99600	0,99639	0,99674	0,99706	0,99736	0,99763	0,99788	0,99811	0,99831
A = 21	0,99850	0,99867	0,99882	0,99895	0,99908	0,99919	0,99929	0,99938	0,99946	0,99953
A = 22	0,99959	0,99965	0,99969	0,99974	0,99978	0,99981	0,99984	0,99986	0,99988	0,99990
A = 23	0,99992	0,99993	0,99994	0,99995	0,99996	0,99997	0,99998	0,99998	0,99998	0,99999
A = 24	0,99999	0,99999	0,99999	1,00000	1,00000	1,00000	1,00000	1,00000	1,00000	1,00000
A = 25	1,00000	1,00000	1,00000	1,00000	1,00000	1,00000				

7.9.5 Código de gama de ganancia de canal

Cuando el nivel de señal es bajo, la gama dinámica del audio decodificado suele estar limitada por la longitud de palabra utilizada en el cálculo de transformación. El uso de longitud de palabra mayor mejora la gama dinámica pero aumenta el coste, pues se debe aumentar la longitud de palabra tanto de las unidades aritméticas como de la RAM de trabajo. Para poder reducir la longitud de palabra del cálculo de transformación, los trenes binarios AC-3 contienen un elemento sintáctico gainrng[ch]. Este elemento de 2 bits existe para cada bloque codificado para cada canal.

El elemento gainrng es un valor comprendido en la gama de 0-3. El valor es una indicación del nivel de muestra máximo dentro del bloque codificado. Cada bloque representa 256 muestras de audio nuevas y 256 muestras de audio previas. Antes de la aplicación de la ventana de 512 puntos, se determina el valor absoluto máximo de los 512 valores MIC. Sobre la base del **valor máximo dentro del bloque**, se fija el valor de gainrng, según se indica a continuación:

Valor absoluto máximo (máx)	**gainrng**
máx ≥ 0,5	**0**
0,5 > máx ≥ 0,25	**1**
0,25 > máx ≥ 0,125	**2**
0,125 > máx	**3**

Si el codificador no ejecuta el paso de encontrar el valor absoluto máximo dentro de cada bloque, el valor de gainrng se debe poner a 0.

El decodificador puede utilizar el valor de gainrng para aplicar un factor de escala previo a los coeficientes de transformación antes de la transformación y un factor de escala posterior a los valores después de la transformación. Con un diseño cuidadoso, el proceso de aplicación de factor de escala posterior se puede realizar en la etapa de salida MIC, lo que permite que una RAM tampón de salida de 16 bits proporcione un audio de gama dinámica de 18 bits.

7.10 Detección de errores

Los datos de AC-3 pueden determinar la existencia de errores en una trama de datos de varias formas. El sistema de transporte que ha entregado los datos puede informar de este hecho al decodificador. Puede verificarse la integridad de los datos utilizando la CRC insertada. Asimismo, la aplicación de algún tipo de verificación de coherencia de los datos recibidos puede indicar la presencia de errores. La estrategia del decodificador cuando se detectan errores es definible por el usuario. Las posibles respuestas incluyen el silenciamiento, las repeticiones de bloques o las repeticiones de tramas. En esta norma no se especifica el alcance de la verificación de errores ni el comportamiento del sistema en presencia de errores, sino que se dejan para la aplicación y realización prácticas.

7.10.1 Verificación de CRC

Cada trama de AC-3 contiene 2 palabras de CRC de 16 bits; crc1 es la segunda palabra de 16 bits de la trama situada inmediatamente después de la palabra sync y crc2 es la última palabra de 16 bits de la trama que precede inmediatamente a la palabra sync de la trama siguiente. El control crc1 se aplica a los primeros 5/8 de la trama, sin incluir la palabra sync. El control crc2 proporciona cobertura de los últimos 3/8 de la trama así como de la trama completa (sin incluir la palabra sync). La decodificación de la palabra o palabras CRC permite la detección de los errores.

Para generar cada una de las palabras de CRC de 16 bits se emplea el polinomio generador: $x^{16} + x^{15} + x^2 + 1$.

En el Cuadro 48 se definen los 5/8 de una trama que pueden calcularse como sigue:

5/8_framesize = truncate(framesize ÷ 2) + truncate(framesize ÷ 8);

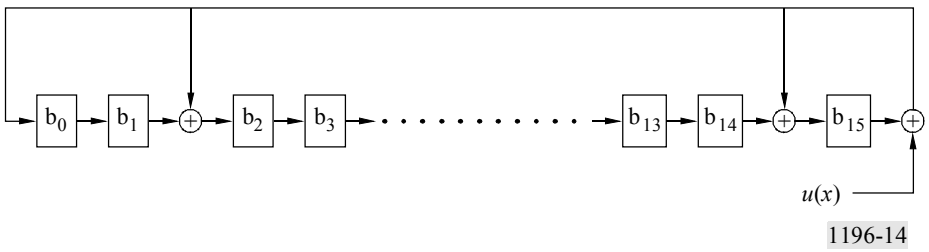
ó

5/8_framesize = (int) (framesize>>1) + (int) (framesize>>3);

donde el tamaño de trama está expresada en unidades de palabras de 16 bits. En el Cuadro 48 se muestra el valor de 5/8 del tamaño de la trama en función de la velocidad de bits en AC-3 y de la velocidad de muestreo de audio.

El cálculo de la CRC puede realizarse mediante alguna de las técnicas convencionales. Una realización física conveniente es el registro de desplazamiento con realimentación (LFSR). En la Fig. 14 se representa un ejemplo de circuito LFSR para el polinomio generador anterior:

FIGURA 14



La verificación de una CRC válida con el circuito anterior consiste en reponer a 0 todos los registros e introducir seguidamente los bits de datos AC-3 en el circuito en el orden en que aparecen en el tren de datos. Como la palabra sync no está amparada por ninguna CRC (aunque se incluye en el 5/8_framesize), no debe incluirse en el cálculo de la CRC. Se considera válida crc1 si la totalidad del contenido del registro son ceros una vez introducidos los primeros 5/8 de la trama. Se prosigue el cálculo hasta que se han introducido todos los datos de la trama y si de nuevo el contenido de los registros es 0, se considera válida crc2. Algunos decodificadores pueden elegir la verificación de crc2 únicamente y no comprobar la validez de crc1 en el punto 5/8 de la trama. Si crc1 es inválida cabe la posibilidad de reponer los registros a 0 y comprobar crc2. Si entonces crc2 concuerda, los últimos 3/8 de la trama están probablemente exentos de errores. Esto, sin embargo, es de poca utilidad ya que si hay errores en los 5/8 iniciales de la trama no es posible decodificar el audio de esa trama aun cuando sus 3/8 finales estén libres de errores.

Obsérvese que crc1 se genera mediante codificadores tales que el cálculo de la CRC producirá un 0 en el punto 5/8 de la trama. Este *no* es el valor obtenido mediante el cálculo de la CRC en los primeros 5/8 de la trama utilizando el polinomio generador anterior. Por consiguiente, los decodificadores no deberán intentar almacenar crc1, calcular la CRC para los primeros 5/8 de la trama y después comparar los dos.

Las restricciones sintácticas del tamaño de bloque dentro de cada trama (forzadas por los codificadores), aseguran que crc1 ampara completamente a los bloques 0 y 1. En consecuencia, los decodificadores pueden comenzar inmediatamente el procesamiento del bloque 0 cuando se ha alcanzado el punto 5/8 de la trama de datos. Esto permite la utilización de memorias tampón de entrada más pequeñas en algunas aplicaciones. Los decodificadores que tengan capacidad de almacenamiento de una trama completa pueden optar por el procesamiento de crc2 únicamente. Tales decodificadores no comenzarán el procesamiento del bloque 0 de una trama hasta que se haya recibido la totalidad de la misma.

CUADRO 48

Cuadro de 5/8_framesize; número de palabras en los primeros 5/8 de la trama

frmsizecod	**Velocidad de bits nominal (kbit/s)**	5/8_framesize f_s = 32 kHz	5/8_framesize f_s = 44,1 kHz	5/8_framesize f_s = 48 kHz
‘000000’ (0)	32	60	42	40
‘000001’ (0)	32	60	43	40
‘000010’ (1)	40	75	53	50
‘000011’ (1)	40	75	55	50
‘000100’ (2)	48	90	65	60
‘000101’ (2)	48	90	65	60
‘000110’ (3)	56	105	75	70
‘000111’ (3)	56	105	76	70
‘001000’ (4)	64	120	86	80
‘001001’ (4)	64	120	87	80
‘001010’ (5)	80	150	108	100
‘001011’ (5)	80	150	108	100
‘001100’ (6)	96	180	130	120
‘001101’ (6)	96	180	130	120
‘001110’ (7)	112	210	151	140
‘001111’ (7)	112	210	152	140
‘010000’ (8)	128	240	173	160
‘010001’ (8)	128	240	173	160
‘010010’ (9)	160	300	217	200
‘010011’ (9)	160	300	217	200
‘010100’ (10)	192	360	260	240
‘010101’ (10)	192	360	261	240
‘010110’ (11)	224	420	303	280
‘010111’ (11)	224	420	305	280
‘011000’ (12)	256	480	347	320
‘011001’ (12)	256	480	348	320
‘011010’ (13)	320	600	435	400
‘011011’ (13)	320	600	435	400
‘011100’ (14)	384	720	521	480
‘011101’ (14)	384	720	522	480
‘011110’ (15)	448	840	608	560
‘011111’ (15)	448	840	610	560
‘100000’ (16)	512	960	696	640
‘100001’ (16)	512	960	696	640
‘100010’ (17)	576	1080	782	720
‘100011’ (17)	576	1080	783	720
‘100100’ (18)	640	1200	870	800
‘100101’ (18)	640	1200	871	800

7.10.2 Verificación de la coherencia del tren de bits

Siempre es posible que una trama AC-3 tenga información de sync válida y CRC válidos ya que en cualquier otro caso no podría decodificarse. Esta última condición puede producirse si una trama está deteriorada de forma que la palabra CRC no tiene validez o en el caso de un error del codificador. Una protección frente a estas situaciones es la realización de algún tipo de pruebas de verificación de errores en el decodificador AC-3 y el clasificador del tren de bits. Pese a su rendimiento de codificación existen redundancias propias en el tren de bits de AC-3. Si un tren de bits AC-3 contiene errores es posible que surjan construcciones sintácticas ilegales. La verificación de estas construcciones ilegales detectará numerosas condiciones de error significativas.

Seguidamente se facilita una lista de condiciones de error conocidas en el tren de datos. En algunas realizaciones puede ser importante que el decodificador tenga aptitud para tratar en primer lugar estos errores. Concretamente puede ser conveniente para los decodificadores asegurarse de que estos errores no provocan la sobreescritura de memoria reservada con datos inválidos y no provocan retardos de procesamiento por entrada en bucles con cómputos de bucles ilegales. Puede admitirse la reproducción de señales de audio inválidas siempre que se mantenga la estabilidad del sistema.

1) (blknum == 0) &&
(cplstre == 0);

2) (cplinu == 1) &&
(no channels in coupling);

3) (cplinu == 1) &&
(cplbegf > (cplendf+2));

4) (cplinu == 1) &&
((blknum == 0) || (previous cplinu == 0)) &&
(chincpl[n] == 1) &&
(cplcoe[n] == 0);

5) (blknum == 0) &&
(acmod == 2) &&
(rematstr == 0);

6) (cplinu == 1) &&
((blknum == 0) || (previous cplinu == 0)) &&
(cplexpstr == 0);

7) (cplinu == 1) &&
((cplbegf != previous cplbegf) || (cplendf != previous cplendf)) &&
(cplexpstr == 0);

8) (blknum == 0) &&
(chexpstr[n] == 0);

9) (cplinu == 1) &&
(cplbegf != previous cplbegf) &&
(chincpl[n] == 1) &&
(chexpstr[n] == 0);

10) (blknum == 0) &&
(lfeon == 1) &&
(lfeexpstr == 0);

11) (chincpl[n] == 0) &&
(chbwcod[n] > 60);

12) (blknum == 0) &&
(baie == 0);

13) (blknum == 0) &&
(snroffste == 0);

14) (blknum == 0) &&
(cplinu == 1) &&
(cplleake == 0);

15) (cplinu == 1) &&
(longitud aumentada de la atribución del bit delta cpl > 50);

16) longitud aumentada de la atribución del bit delta[n] > 50;

17) valor del exponente codificado de forma compuesta con 5 niveles > 124;

18) valor de la mantisa codificado de forma compuesta con 3 niveles > 26;

19) valor de la mantisa codificado de forma compuesta con 5 niveles > 124;

20) valor de la mantisa codificado de forma compuesta con 11 niveles > 120;

21) el desempaquetamiento del tren de bits continua más allá del final de la trama.

Obsérvese que alguna de estas condiciones (tales como las número 17 a número 20) únicamente pueden comprobarse a niveles inferiores dentro del soporte lógico del decodificador lo que conduce a una incidencia MIPS potencialmente importante. En tanto y cuanto tales condiciones no afecten a la estabilidad del sistema, no es necesario prevenirlas específicamente.

8 Codificación del tren de bits AC-3

8.1 Introducción

En este punto se proporcionan algunas orientaciones sobre la codificación AC-3. Como el AC-3 viene especificado por la sintaxis y procesamiento en el decodificador, no se especifica de forma precisa el codificador. El único requisito normativo del codificador es que el tren de bits elementales de salida se ajuste a la sintaxis AC-3. Pueden elaborarse codificadores de distintos grados de complejidad. Los codificadores más complejos pueden ofrecer una calidad de audio superior y hacer viable el funcionamiento con velocidades de bits inferiores. Se espera que los codificadores mejoren con el tiempo. Tales mejoras favorecerán a los decodificadores. El codificador descrito en este punto es básico en cuanto a su funcionamiento y proporciona buena calidad. En la descripción que sigue se sugieren varias posibilidades de mejora potencial. En la Fig. 15 se muestra un diagrama de flujo del proceso de codificación.

8.2 Resumen del proceso de codificación

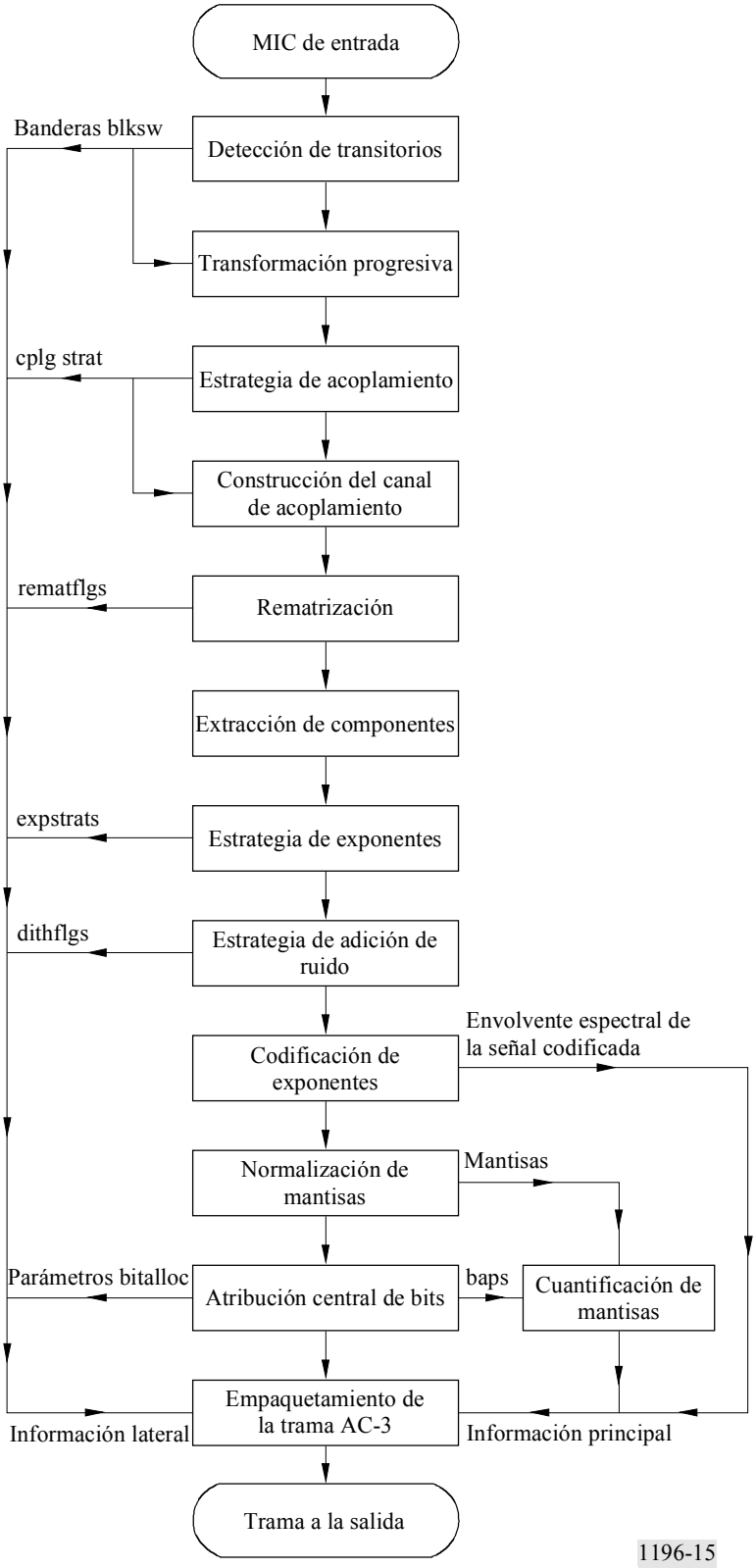
8.2.1 MIC de entrada

8.2.1.1 Longitud de la palabra de entrada

El codificador AC-3 acepta señales audio en forma de palabra MIC. La gama dinámica interna de AC-3 admite longitudes de palabras de hasta 24 bits como válidas.

FIGURA 15

Diagrama de flujo del proceso de codificación



8.2.1.2 Velocidad de muestreo de entrada

La velocidad de muestreo de entrada debe estar sincronizada con la velocidad de bits de salida de forma que cada trama sync de AC-3 contengan 1 536 muestras de audio. Si se dispone de una señal audio de entrada en un formato MIC con una velocidad de muestras distinta de la necesaria, debe efectuarse la conversión de velocidad de muestreo para que haya concordancia con la velocidad de muestreo.

8.2.1.3 Filtrado de entrada

Pueden someterse a los canales de entrada individuales a un filtrado de paso alto. La eliminación de las componentes de corriente continua de la señal puede permitir una codificación más eficaz ya que no se utilizan velocidades de datos para la codificación de la componente continua. Sin embargo, se corre el riesgo de que señales que no llegan al nivel MIC del 100% antes del filtrado de paso alto rebasen ese nivel tras el filtrado y, en consecuencia, queden recortadas. Un codificador típico efectuará el filtrado de paso alto de las señales de entrada con un filtro de un solo polo a 3 Hz.

Deberá aplicarse al canal lfe un filtrado de paso bajo a 120 Hz. Un codificador típico efectuará el filtrado del canal lfe mediante un filtro elíptico de orden 8 con una frecuencia de corte de 120 Hz.

8.2.2 Detección de transitorios

En los canales de anchura de banda completa se detectan los transitorios a fin de decidir cuándo debe efectuarse la conmutación a bloques de audio de longitud acortada para mejorar la calidad pre-ecos. Se examinan versiones de las señales filtradas en paso alto para detectar un aumento de la energía de un segmento temporal de un sub-bloque al siguiente. Los sub-bloques se examinan en escalas temporales diferentes. Si en la segunda mitad del bloque de audio de un canal se detecta un transitorio, se conmuta ese canal a un bloque corto. Un canal cuyo bloque se ha conmutado utiliza la estrategia de exponente d45.

Se emplea el detector de transitorios para determinar cuándo debe efectuarse la conmutación de un bloque transformado largo (longitud 512) al bloque corto (longitud 256). El detector actúa sobre 512 muestras en cada bloque de audio. Esta operación se efectúa en dos recorridos, procesándose en cada recorrido 256 muestras. La detección de transitorios se divide en 4 etapas:

Etapa 1: Filtrado de paso alto.

Etapa 2: Segmentación del bloque en submúltiplos.

Etapa 3: Detección de la amplitud de cresta dentro de cada segmento de sub-bloque.

Etapa 4: Comparación de umbral.

El detector de transitorios genera una bandera blksw[n] para cada canal de anchura de banda completa. Cuando se pone la bandera a «1» ello indica la presencia de un transitorio en la segunda mitad del bloque de entrada de longitud 512 en el canal correspondiente.

Etapa 1: Filtrado de paso alto: Se realiza el filtrado de paso alto mediante un filtro IIR de tipo II directo, bicuadrático en cascada, con una frecuencia de corte de 8 kHz.

Etapa 2: Segmentación del bloque: El bloque de 256 muestras filtradas en paso alto se segmenta en forma de árbol jerárquico de niveles en el cual el nivel 1 representa el bloque de longitud 256, el nivel 2 corresponde a 2 segmentos de longitud 128 y el nivel 3 representa 4 segmentos de longitud 64.

Etapa 3: Detección de cresta: Para cada segmento de cada uno de los niveles del árbol jerárquico se identifica la muestra de valor máximo. Las crestas para un único nivel se localizan como sigue:

$$P[j][k] = \max(x(n))$$
$$\text{for } n = (512 \times (k-1) / 2j), (512 \times (k-1) / 2j) + 1, \ldots (512 \times k / 2j) - 1$$
$$\text{and } k = 1, \ldots 2j-1;$$

donde:

$x(n)$ = muestra n-ésima del bloque de longitud 256

j = 1, 2, 3 es el número del nivel jerárquico

k = número de segmento en el nivel j

Obsérvese que $P[j][0]$, (esto es $k = 0$) se define como la cresta del último segmento en el nivel j del árbol calculado inmediatamente antes del árbol actual. Por ejemplo $P[3][4]$ del árbol precedente es $P[3][0]$ del árbol actual.

Etapa 4: Comparación de umbral: La primera etapa del comparador de umbral comprueba si hay un nivel de señal importante en el bloque vigente. Esto se realiza comparando el nivel de cresta global $P[1][1]$ del bloque vigente con un «umbral de silencio». Si $P[1][1]$ es inferior a este umbral se fuerza un bloque largo. El valor umbral de silencio es 100/32768. La etapa siguiente del comparador verifica los niveles de cresta relativos de segmentos adyacentes en cada nivel del árbol jerárquico. Si, para un nivel determinado, la relación de cresta de cualquier par de segmentos adyacentes excede un umbral predefinido para ese nivel, se activa una bandera que indica la existencia de un transitorio en el bloque vigente de longitud 256. Las relaciones se comparan como sigue:

$$\text{mag}(P[j][k]) \times T[j] > \text{mag}(P[j][(k-1)])$$

donde:

$T[j]$ es el umbral predefinido para el nivel j definido como sigue:

$T[1] = 0{,}1$

$T[2] = 0{,}075$

$T[3] = 0{,}05$

Si esta desigualdad es cierta para cualquier par de crestas de segmento en cualquier nivel, se indica un transitorio en la primera mitad del bloque de entrada de longitud 512. El segundo recorrido por este proceso determina la presencia de transitorios en la segunda mitad del bloque de entrada de longitud 512.

8.2.3 Transformación progresiva

8.2.3.1 Enventanado

El bloque de audio se multiplica por una función ventana para reducir los efectos de contorno de la transformada y mejorar la selectividad de frecuencias en el banco de filtros. En el Cuadro 47 se indican los valores de la función ventana. Obsérvese que los 256 coeficientes proporcionados se usan acoplados para formar una ventana simétrica de 512 puntos.

8.2.3.2 Transformación de tiempo en frecuencia

Sobre la base de las banderas de conmutación de bloques, cada bloque de audio se transforma en el dominio de la frecuencia realizando una transformación larga de $N = 512$ puntos, o dos

transformaciones cortas de $N = 256$ puntos. Sea que $x[n]$ representa la secuencia de tiempo de entrada en ventana. La secuencia de frecuencia de salida, $X_D[k]$ es definida por:

$$X_D[k] = \frac{-2}{N} \sum_{n=0}^{N-1} x[n] \cos\left(\frac{2\pi}{4N} (2n + 1)(2k + 1) + \frac{\pi}{4}(2k + 1)(1 + \alpha) \right) \qquad \text{para } 0 \le k < N/2$$

donde:

$\alpha = -1$ para la primera transformación corta

0 para la transformación larga

+1 para la segunda transformación corta

8.2.4 Estrategia de acoplamiento

8.2.4.1 Codificador básico

Para un codificador básico, puede emplearse una estrategia de acoplamiento estática. A continuación se indican parámetros de acoplamiento adecuados.

```
cplbegf     = 6; /*  el acoplamiento comienza en 10,2 kHz */
cplendf     = 12; /*  el canal de acoplamiento termina en 20,3 kHz */
cplbndstrc  = 0, 0, 1, 1, 0, 1, 1, 1;
cplinu      = 1; /*  acoplamiento activado siempre */
/* acoplados todos los canales conmutados que no tienen estructura de bloque */
for(ch=0; ch<nfchans; ch++) if(blksw[ch]) chincpl[ch] = 0; else chincpl[ch] = 1.
```

Pueden transmitirse las coordenadas de acoplamiento de todos los canales para cada bloque, es decir, los bloques 0, 2 y 4. Para los bloques 1, 3 y 5 se reutilizan las coordenadas de acoplamiento.

8.2.4.2 Codificador perfeccionado

Los codificadores más perfeccionados pueden emplear parámetros de acoplamiento variables dinámicamente. Pueden hacerse variables las frecuencias de acoplamiento sobre la base de la demanda de bits y de un modelo psicoacústico que compara la percepción de los artefactos producidos por la supresión de bits frente a los provocados por el proceso de acoplamiento. Pueden librarse del acoplamiento los canales que tengan un nivel de potencia variable rápidamente con el tiempo. Las coordenadas de acoplamiento de los canales cuyos niveles de potencia varíen con lentitud se envían con menor frecuencia. La estructura de la banda de acoplamiento puede ser dinámica.

8.2.5 Construcción del canal de acoplamiento

8.2.5.1 Canal de acoplamiento

El codificador más básico puede construir el canal de acoplamiento simplemente mediante la adición conjunta de todos los coeficientes de los canales individuales y dividiendo el resultado por 8. La división por 8 impide que el canal de acoplamiento rebase el valor 1. Codificadores algo más complejos pueden modificar el signo de los canales individuales antes de agregarlos a la suma a fin de evitar cancelaciones de fase.

8.2.5.2 Coordenadas de acoplamiento

Se obtienen las coordenadas de acoplamiento calculando relaciones de potencia dentro de cada banda de acoplamiento. La potencia del canal original dentro de una banda de acoplamiento se divide entre la potencia del canal de acoplamiento dentro de esa banda de acoplamiento. Este cociente de potencias es la coordenada de acoplamiento. Las coordenadas de acoplamiento así obtenidas se convierten a un formato de punto flotante y se cuantifican. Se examinan los exponentes de cada canal para ver si pueden graduarse ulteriormente por los factores 3, 6 ó 9. Este proceso genera la coordenada de acoplamiento principal de 2 bits para ese canal (las coordenadas de acoplamiento principales permiten aumentar la gama dinámica representada por la coordenada de acoplamiento).

8.2.6 Rematrización

La rematrización únicamente se aplica al modo 2/0. Dentro de cada banda de rematrización se efectúan mediciones de potencia de las señales L, R, L+R y L–R. Si la potencia máxima aparece en los canales L o R, no se activa la bandera de rematrización para esa banda. Si la potencia máxima se encuentra en la señal L+R o L–R, se activa la bandera de rematrización. Cuando la bandera de rematrización está activada, el codificador codifica L+R y L–R en lugar de L y R. La rematrización se describe en el § 7.5.

8.2.7 Extracción de exponentes

Se examina la representación binaria de cada coeficiente de frecuencia para determinar el número de ceros delanteros. El valor del exponente inicial es igual al número de ceros delanteros (hasta un máximo de 24). Se extraen esos exponentes empleándose los conjuntos de exponentes (uno para cada bloque de cada canal, incluido el canal de acoplamiento) para determinar las estrategias de exponente apropiadas.

8.2.8 Estrategia de exponentes

Para cada canal se examina la variación de los exponentes en la frecuencia y en el tiempo. Si los exponentes indican un espectro relativamente plano, puede utilizarse una estrategia de exponente tal con la d25 o d45. Si el espectro es muy tonal deberá emplearse una estrategia de exponente de elevada resolución espectral, tal como la d15 o d25. Si el espectro varía poco en los 6 bloques de una trama, pueden enviarse los exponentes únicamente para el bloque 0 y reutilizarse para los bloques 1 a 5. Si los exponentes varían rápidamente en una trama, deben enviarse exponentes para el bloque 0 y para aquellos bloques que tengan conjuntos de exponentes que difieran sustancialmente de los exponentes enviados anteriormente. Existe una transacción entre resolución frecuencial precisa, resolución temporal precisa y número de bits necesarios para enviar exponentes. En general cuando se trabaje con velocidades binarias muy pequeñas, es necesario intercambiar resolución temporal con resolución de frecuencias.

En un codificador básico puede emplearse un algoritmo sencillo. En primer lugar, se observa la variación de los exponentes con el tiempo. Cuando la variación exceda un umbral se enviarán nuevos exponentes. La estrategia de exponentes utilizada se hace dependiente del número de bloques que utilizan los nuevos conjuntos de exponentes. Si los exponentes se utilizaran únicamente para un solo bloque, se empleará la estrategia d45. Si deben emplearse nuevos exponentes para 2 ó 3 bloques, la estrategia a utilizar será la d25. En el caso en que deban emplearse nuevos exponentes para 4, 5 ó 6 bloques, se utilizará la estrategia d15.

8.2.9 Estrategia de adición de ruido

El codificador controla, canal por canal, qué coeficientes cuantificados a 0 bits deberán reproducirse con ruido aleatorio. El objetivo es mantener aproximadamente la misma energía en el espectro reproducido, aun cuando no se hayan atribuido bits a porciones del espectro. En función de la estrategia de exponente y de la exactitud de los exponentes codificados puede ser conveniente detraer ruido de algunos bloques.

Un codificador básico puede aplicar una estrategia sencilla de adición de ruido canal por canal. Cuando blksw[ch] es 1 se detraerá ruido de ese bloque y del bloque siguiente.

8.2.10 Codificación de exponentes

Sobre la base de la estrategia de exponentes seleccionada se efectúa el procesamiento previo de los exponentes de cada conjunto de exponentes. Las estrategias de exponente d25 y d45 requieren la compartición de un único exponente por más de una mantisa. Para su transmisión en el tren de bits, los exponentes se codificarán de forma diferencial. La diferencia entre dos exponentes brutos

sucesivos no produce necesariamente códigos diferenciales legales (valor máximo = ±2) si la velocidad de los exponente brutos es mayor que la permitida por la estrategia de exponente. El procesamiento previo ajusta los coeficientes de forma que los coeficientes de la transformada que comparten un exponente tienen el mismo exponente por lo que sus valores diferenciales son valores legales. El resultado de este procesamiento es que los valores de algunos exponentes disminuirán, por lo que las mantisas correspondientes tendrán algunos ceros delanteros.

Para generar la envolvente espectral codificada se someten los exponentes a una codificación diferencial. Como parte del procesamiento del codificador, se genera un conjunto de exponentes igual al conjunto de que dispondrá el decodificador cuando decodifique la envolvente espectral.

8.2.11 Normalización de mantisas

Se normalizan los coeficientes de la transformada de cada canal desplazando cada coeficiente a la izquierda un número de veces igual al indicado por su exponente correspondiente para generar mantisas normalizadas. Los coeficientes binarios de frecuencias originales se desplazan a la izquierda según el valor de los exponentes que utilizará el decodificador. Algunas de las mantisas normalizadas tendrán ceros delanteros. Se cuantifican las mantisas normalizadas.

8.2.12 Atribución de bits de núcleo

Un codificador básico puede utilizar la rutina de atribución de bits de núcleo con todos los parámetros fijados en los valores por defecto nominales.

```
sdcycod = 2;
fdcycod = 1;
sgaincod = 1;
dbpbcod = 2;
floorcod = 4;
cplfgaincod = 4;
fgaincod[ch] = 4;
lfegaincod = 4;
cplsnroffst = fsnroffst[ch] = lfesnroffst = fineoffset;
```

Como los parámetros de atribución de bits son estáticos, únicamente se envían durante el bloque 0. No se utiliza la atribución del bit delta, por lo que deltbaie = 0. Se ejecuta la rutina de atribución de bits de núcleo (descrita en el § 7.2) y se ajustan los desplazamientos SNR grueso y fino hasta que se utilicen todos los bits disponibles de la trama. Los ajustes de desplazamiento SNR grueso se efectúan en incrementos de 6 dB, en tanto que los ajustes finos de desplazamiento se realizan en incrementos de 3/8 dB. Los bits se atribuyen de forma global a partir de una reserva de bits común a todos los canales. Se eligen las combinaciones de csnroffst y fineoffset que hagan uso del máximo número de bits sin rebasar el tamaño de la trama. Esto implica un proceso iterativo. Cuando, para una iteración determinada, el número de bits excede de la reserva, se disminuye el desplazamiento de SNR para la siguiente iteración. Por otro lado, si la atribución es inferior a la reserva, en la siguiente iteración se aumenta el desplazamiento SNR.

La iteración concluye cuando se alcanza el máximo del desplazamiento SNR sin que la atribución exceda de la reserva. El resultado de la rutina de atribución de bits son los valores finales de csnroffst y fineoffset y el conjunto de punteros de atribución de bits (baps). Los valores de desplazamiento SNR se incluyen en el tren de bits de forma que el decodificador no necesita efectuar la iteración.

8.2.13 Cuantificación de las mantisas

El bloque de cuantificación de la mantisa utiliza los baps. Para cada coeficiente de la transformada individual hay un bap. Cada una de las mantisas normalizadas se cuantifica empleando el cuantificador indicado por el bap correspondiente. Las mantisas cuantificadas de forma asimétrica se cuantifican redondeando el número de bits indicado por el bap correspondiente. Las mantisas cuantificadas de forma simétrica se cuantifican mediante la búsqueda en una tabla. Las mantisas con baps iguales a 1, 2 y 4 se agrupan en tripletas o en parejas.

8.2.14 Empaquetamiento de la trama AC-3

Todos los datos se empaquetan en la trama AC-3 codificada. Algunas de las mantisas cuantificadas se agrupan conjuntamente y se codifican con una palabra código única. El formato de salida depende de la aplicación. La trama puede emplearse en forma de ráfaga o como un tren de datos a velocidad constante.

Apéndice 1
al Anexo 2

(Normativo)

Trenes elementales AC-3 en el múltiplex MPEG-2

1 Alcance

Este Apéndice contiene especificaciones sobre la forma de combinar uno o más trenes elementales AC-3 en el tren de transporte MPEG-2 ATSC (Recomendación UIT-R BT.1300, Sistema A) o DVB (Recomendación UIT-R BT.1300, Sistema B) (Norma ISO/CEI 13818-1).

2 Introducción

El tren binario elemental AC-3 se incluye en un tren binario múltiplex MPEG-2 casi de la misma manera como se incluiría un tren de audio MPEG-1. El tren binario AC-3 se organiza en paquetes PES. Un tren binario múltiplex MPEG-2 que contiene trenes elementales AC-3 debe cumplir con todas las constricciones de audio descritas en el modelo STD en el § 3.6 (Sistema A) o en el § 4.4 (Sistema B). Es necesario indicar sin ambigüedades que un tren AC-3 es, de hecho, un tren AC-3 (y no un tren de audio MPEG). La norma MPEG-2 no indica explícitamente los códigos que se han de utilizar para indicar un tren AC-3. La norma MPEG-2 tampoco tiene un descriptor de audio adecuado para describir el contenido del tren binario AC-3 en los cuadros de información específica de programa (PSI – program specific information).

La unidad de acceso (AU) o la unidad de presentación (PU) de audio AC-3 es una trama de sincronización AC-3 que contiene 1 536 muestras de audio. La duración de una unidad de acceso (o de presentación) AC-3 es de 32 ms por audio muestreado a 48 kHz, aproximadamente de 34,83 ms por audio muestreado a 44,1 kHz, y de 48 ms por audio muestreado a 32 kHz.

Los ítems que es necesario especificar para incluir AC-3 dentro del tren binario MPEG-2 son: stream_type, stream_id, descriptor de audio AC-3 y para el Sistema A únicamente, descriptor de registro. El descriptor de registro no es necesario en el Sistema B, pues se considera que el descriptor AC-3 (AC-3_descriptor) es un descriptor público en este Sistema. El descriptor de idiomas ISO 639 puede emplearse para indicar idioma. Se imponen algunas constricciones en la capa PES en el caso de múltiples trenes de audio destinados a ser reproducidos en sincronismo de muestra exacto. En el Sistema A (ATSC) el descriptor de audio AC-3 se titula «descriptor de tren de audio» (audio_stream_descriptor), mientras que en el Sistema B (DVB), el descriptor de audio AC-3 se titula «descriptor_AC-3» (AC-3_descriptor). Debe señalarse que la sintaxis de estos descriptores difiere significativamente entre uno y otro Sistema.

3 Especificación detallada para el Sistema A (ATSC)

3.1 Tipo de tren (stream_type)

El valor de stream_type, para AC-3 será 0×81.

3.2 Id. de tren (stream_id)

El valor de stream_id, en el encabezamiento PES será 0×BD (lo que indica private_stream_1). Múltiples trenes AC-3 pueden compartir el mismo valor de stream_id, puesto que cada tren es transportado con un valor de PID único. La correspondencia entre los valores de PID y stream_type se indica en el cuadro de correspondencia de programa (PMT, *programme map table*) del tren de transporte.

3.3 Descriptor de registro (registration_descriptor)

En el Cuadro 49 se muestra la sintaxis del descriptor de registro AC-3 (registration_descriptor). El descriptor de registro AC-3 (registration_descriptor) se incluirá en la sección de correspondencia de programas de TS (TS_program_map_section).

CUADRO 49

Descriptor de registro AC-3 (registration_descriptor)

Sintaxis	N.° de bits	Mnemónico
registration_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
format_identifier	**32**	**uimsbf**
}		

descriptor_tag — 0×05.
descriptor_length — 0×04.
format_identifier — Le format_identifier AC-3 est 0×41432D33 («AC-3»).

3.4 Descriptor de tren de audio AC-3 (audio_stream_descriptor)

El descriptor de tren de audio AC-3, (audio_stream_descriptor) que se muestra en el Cuadro 50 permite incluir información sobre los diferentes trenes elementales AC-3 en los cuadros de información específica de programa (PSI). Esta información es útil para poder dirigir el o los trenes AC-3 apropiados al decodificador de audio. Obsérvese que las líneas horizontales del Cuadro indican los puntos de terminación admisibles para el descriptor.

CUADRO 50

Sintaxis de descriptor de audio AC-3 (audio_stream_descriptor)

Sintaxis	N.° de bits	Mnemónico
audio_stream_descriptor( ) {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
sample_rate_code	3	bslbf
bsid	5	bslbf
bit_rate_code	6	bslbf
surround_mode	2	bslbf
bsmod	3	bslbf
num_channels	4	bslbf
full_svc	1	bslbf
langcod	8	bslbf
if(num_channels==0) /* 1+1 mode */		
langcod2	8	bslbf
if(bsmod<2) {		
mainid	3	uimsbf
reserved	5	bslbf
}		
else asvcflags	8	bslbf
textlen	7	uimsbf
text_code	1	bslbf
for(i=0; i<M; i++) {		
text[i]	8	bslbf
}		
for(i=0; i<N; i++) {		
additional_info[i]	N×8	bslbf
}		
}		

descriptor_tag – El valor para el rótulo de descriptor AC-3 (descriptor_tag) es 0×81.

descriptor_length (longitud de descriptor) – Éste es un campo de 8 bits que especifica el número de bytes del descriptor que sigue inmediatamente al campo descriptor_length.

sample_rate_code (código de velocidad de muestra) – Éste es un campo de 3 bits que indica la velocidad de muestra del audio codificado. La indicación se puede referir a una velocidad de muestra específica, o a un conjunto de valores que incluyen la velocidad de muestra del audio codificado (véase el Cuadro 51).

CUADRO 51

Cuadro de códigos de velocidad de muestra (sample_rate_code)

sample_rate_code	Velocidad de muestra (kHz)
‘000’	48
‘001’	44,1
‘010’	32
‘011’	Reservado
‘100’	48 ó 44,1
‘101’	48 ó 32
‘110’	44,1 ó 32
‘111’	48 ó 44,1 ó 32

bsid – Éste es un campo de 5 bits que se pone al mismo valor que el campo bsid en el tren elemental AC-3.

bit_rate_code (código de velocidad binaria) – Éste es un campo de 6 bits. Los 5 bits más bajos indican una velocidad binaria nominal. El bit más significativo indica si la velocidad binaria señalada es exacta (bit más significativo = 0) o un límite superior (bit más significativo = 1) (véase el Cuadro 52).

CUADRO 52

Cuadro de códigos de velocidad binaria (bit_rate_code)

bit_rate_code	**Velocidad binaria exacta (kbit/s)**	bit_rate_code	**Límite superior de velocidad binaria (kbit/s)**
‘000000’ (0)	32	‘100000’ (32)	32
‘000001’ (1)	40	‘100001’ (33)	40
‘000010’ (2)	48	‘100010’ (34)	48
‘000011’ (3)	56	‘100011’ (35)	56
‘000100’ (4.)	64	‘100100’ (36)	64
‘000101’ (5)	80	‘100101’ (37)	80
‘000110’ (6)	96	‘100110’ (38)	96
‘000111’ (7)	112	‘100111’ (39)	112
‘001000’ (8)	128	‘101000’ (40)	128
‘001001’ (9)	160	‘101001’ (41)	160
‘001010’ (10)	192	‘101010’ (42)	192
‘001011’ (11)	224	‘101011’ (43)	224
‘001100’ (12)	256	‘101100’ (44)	256
‘001101’ (13)	320	‘101101’ (45)	320
‘001110’ (14)	384	‘101110’ (46.)	384
‘001111’ (15)	448	‘101111’ (47)	448
‘010000’ (16)	512	‘110000’ (48)	512
‘010001’ (17)	576	‘110001’ (49)	576
‘010010’ (18)	640	‘110010’ (50)	640

dsurmod – Éste es un campo de 2 bits que se puede poner al mismo valor que el campo dsurmod en el tren elemental AC-3, o que se puede poner a 00 (no indicado) (véase el Cuadro 53).

CUADRO 53

Cuadro dsurmod

surround_mode	**Significado**
‘00’	No indicado
‘01’	Entorno Dolby NO codificado
‘10’	Entorno Dolby codificado
‘11’	Reservado

bsmod – Éste es un campo de 3 bits que se pone al mismo valor que el campo bsmod en el tren elemental AC-3.

num_channels – Éste es un campo de 4 bits que indica el número de canales del tren elemental AC-3. Cuando el bit más significativo es 0, los 3 bits más bajos se ponen al mismo valor que el campo acmod en el tren elemental AC-3. Cuando el bit más significativo es 1, los 3 bits más bajos indican el número máximo de canales de audio codificados (contando el canal lfe como 1). Si el valor de acmod en el tren elemental AC-3 es 000 (modo 1 + 1), el valor de num_channels se pondrá a 0000 (véase el Cuadro 54).

CUADRO 54

Cuadro de número de canales (num_channels)

num_channels	**Modo de codificación audio (acmod)**	num_channels	**Número de canales codificados**
‘0000’	1 + 1	‘1000’	1
‘0001’	1/0	‘1001’	≤ 2
‘0010’	2/0	‘1010’	≤ 3
‘0011’	3/0	‘1011’	≤ 4
‘0100’	2/1	‘1100’	≤ 5
‘0101’	3/1	‘1101’	≤ 6
‘0110’	2/2	‘1110’	Reservado
‘0111’	3/2	‘1111’	Reservado

full_svc – Éste es un campo de 1 bit que indica si este servicio de audio es o no un servicio completo adecuado para la presentación, o si es sólo un servicio parcial que debe combinarse con otro servicio de audio antes de la presentación. Este bit se debe poner a «1» si este servicio de audio es suficientemente completo como para ser presentado al oyente sin combinarlo con otro servicio de audio (por ejemplo, un servicio visualmente degradado que contiene todos los elementos del programa: música, efectos, diálogo y la descripción narrativa del contenido visual). Este bit se debe poner a «0» si el servicio no es suficientemente completo como para ser presentado sin combinarlo con otro servicio de audio (por ejemplo, un servicio visualmente degradado que sólo contiene una descripción narrativa del contenido de programa visual y se tiene que combinar con otro servicio de audio que contiene música, efectos y diálogo).

langcod – Éste es un campo de 8 bits que se pone al mismo valor que el campo langcod en el tren elemental AC-3.

langcod2 – Éste es un campo de 8 bits que se pone al valor del campo langcod2 en el tren elemental AC-3.

Véase que este campo no se utiliza para indicar idioma, pues la indicación de idioma en los sistemas MPEG-2 se hace por medio del descriptor de idioma ISO 639. (NOTA – Como se han utilizado, el significado de los campos langcod y langcod2 en el tren elemental AC-3 puede redefinirse en el futuro.)

mainid – Éste es un campo de 3 bits que contiene un número en la gama de 0 a 7 e identifica un servicio de audio principal. Cada servicio principal se debe rotular con un número único. Este valor se utiliza como un identificador para vincular los servicios asociados con determinados servicios principales.

asvcflags – Éste es un campo de 8 bits. Cada bit (0-7) indica los servicios principales con los que está asociado este servicio. El bit más a la izquierda, el bit 7, indica si este servicio asociado puede ser reproducido junto con el servicio principal número 7. Si el bit tiene un valor de 1, el servicio está asociado con el servicio principal número 7. Si el bit tiene un valor de 0, el servicio no está asociado con el servicio principal número 7.

textlen – Éste es un entero sin signo que indica la longitud, en bytes, del campo de texto descriptivo que sigue.

text_code – Éste es un campo de 1 bit que indica cómo está codificado el siguiente campo de texto. Si este bit es «1», el texto está codificado como caracteres de 1 byte que utiliza el alfabeto Latino 1 de la Norma ISO (ISO 8859-1). Si este bit es «0», el texto está codificado con caracteres unicódigo de 2 bytes.

text[i] – El campo de texto puede contener una breve descripción textual del servicio de audio.

additional_info[j] – Éste es un conjunto de bytes adicionales que rellenan el resto del descriptor. Normalmente no se define la finalidad de estos bytes. Este campo se proporciona para poder extender el descriptor en el futuro.

3.5 Código de idiomas ISO 639 (ISO_639_language_code)

El descriptor ISO_639_language_code permite rotular un tren con el código de idiomas ISO 639 de 24 bits.

3.6 Tamaño de memoria tampón de audio STD

En el caso de un tren de transporte MPEG-2, el modelo T-STD define el tamaño de la memoria tampón de audio principal, BS_n, como:

$$BS_n = BS_{mux} + BS_{dec} + BS_{oh}$$

donde:

$BS_{mux} =$ 736 bytes

BS_{oh}: tara de encabezamiento de PES

BS_{dec}: memoria tampón de la unidad de acceso.

MPEG-2 especifica un valor fijo para BS_n (3 584 bytes) e indica que cualquier espacio disponible en la memoria tampón se puede utilizar para multiplexación adicional.

Cuando un tren elemental AC-3 es transportado por un tren de transporte MPEG-2, el tren de transporte deberá tener un tamaño de memoria tampón de audio principal de:

$$BS_n = BS_{mux} + BS_{pad} + BS_{dec}$$

donde:

$BS_{mux} =$ 736 bytes

$BS_{pad} =$ 64 bytes

BS_{dec} figura en el Cuadro 13 de la Norma A/52 de ATSC. El valor de BS_{dec} empleado será el de la velocidad binaria más alta que soporte el sistema (es decir, el tamaño de la memoria tampón no disminuye cuando la velocidad binaria de audio es inferior al valor máximo permitido por un sistema específico). Los 64 bytes en BS_{pad} están disponibles para el BS_{oh} y la multiplexación adicional. Esta constricción permite realizar decodificadores con la memoria tampón mínima posible.

4 Especificación detallada para el Sistema B (DVB)

4.1 Tipo de tren (stream_type)

El valor de stream_type para un tren elemental AC-3 será 0×06 (indicativo de que los paquetes PES contienen datos privados).

4.2 Id de tren (stream_id)

El valor de stream_id en el encabezamiento PES será 0×BD (lo que indica private_stream_1). Múltiples trenes AC-3 pueden compartir el mismo valor de stream_id, puesto que cada tren es transportado con un valor de PID único. La correspondencia entre los valores de PID y stream_type se indica en el PMT del tren de transporte.

4.3 Información de servicio

4.3.1 Descriptor AC-3 (AC-3_descriptor)

El AC-3_descriptor identifica un tren elemental de audio AC-3 que se ha codificado conforme a esta Recomendación. El objetivo que se pretende es dar una información de configuración para el IRD. El descriptor está situado en el cuadro PSI PMT y se utiliza una vez en una sección de mapa de programa, program_map_section, tras el campo pertinente ES_info_length para todo tren que contenga AC-3.

El rótulo del descriptor da una identificación única de la presencia del tren elemental AC-3. Pueden utilizarse otros campos facultativos en el descriptor para dar identificación del modo de tipo de componente del audio AC-3 codificado en el tren (campo AC-3_type) e indicar si el tren es un servicio de audio AC-3 principal (campo principal (mainid)) o un servicio AC-3 asociado (campo avsc).

El descriptor tiene una longitud mínima de un byte, pero puede ser mayor dependiendo del estado de las banderas y del bucle de información adicional.

4.3.2 Sintaxis del AC-3_descriptor

El AC-3_descriptor (véase el Cuadro 55) se utilizará en el cuadro PSI PMT para identificar trenes que cursan audio AC-3. El descriptor ha de situarse una vez en una sección de mapa de programa que siga al campo pertinente ES_info_length.

CUADRO 55

Sintaxis del AC-3_descriptor

Sintaxis	N.º de bits	Identificador
AC-3_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
AC-3_type_flag	1	bslbf
bsid_flag	1	bslbf
mainid_flag	1	bslbf
asvc_flag	1	bslbf
reserved	1	bslbf
reserved	1	bslbf
reserved	1	bslbf
reserved	1	bslbf

If (AC-3_type_flag)==1{
AC-3_type
}
	8	uimsbf

If (bsid_flag)==1{
bsid
{
	8	uimsbf

If (mainid_flag)==1{
mainid
}
	8	uimsbf

If (asvc_flag)==1{
asvc
}
	8	bslbf

For (I=0;I<N;I++){
additional_info[I]
}
	N × 8	uimsbf

}		

descriptor_tag – El rótulo de descriptor es un campo de 8 bits que identifica cada descriptor. El descriptor_tag de AC-3 tendrá un valor de 0×6A.

descriptor_length – Este campo de 8 bits especifica el número de bytes de la parte de datos del descriptor que sigue al byte en el que se define el valor de este campo. El descriptor AC-3 tiene una longitud mínima de un byte, pero puede ser mayor dependiendo de la utilización de las banderas facultativas y del bucle adicional info, additional_info.

AC-3_type_flag – Este campo de 1 bit es obligatorio. Debe ponerse en «1» para incluir el campo facultativo AC-3_type en el descriptor.

bsid_flag – Este campo de 1 bit es obligatorio. Debe ponerse en «1» para incluir el campo bsid facultativo en el descriptor.

mainid_flag – Este campo de 1 bit es obligatorio. Debe ponerse en «1» para incluir el campo mainid facultativo en el descriptor.

asvc_flag – Este campo de 1 bit es obligatorio. Debe ponerse en «1» para incluir el campo asvc facultativo en el descriptor.

reserved flags – Estos campos de 1 bit están reservados para utilización futura. Deben estar siempre puestos en «0».

AC-3_type – Este campo facultativo de 8 bits indica el tipo de audio que cursa el tren elemental AC-3. Se pone en el mismo valor que el campo component_type del component_descriptor (véase el Cuadro 56).

bsid – Este campo facultativo de 8 bits indica la versión de codificación AC-3. Los tres MSB deben ponerse siempre en «0». Los cinco LSB han de ponerse en el mismo valor que el campo bsid en el tren elemental «01000» (=8) en la versión actual de AC-3.

mainid – Este campo facultativo de 8 bits identifica un servicio de audio principal y contiene un número comprendido entre 0 y 7 que identifica un servicio de audio principal. Cada servicio principal debe etiquetarse con un número único. Este valor se utiliza como identificador para enlazar servicios asociados con servicios principales particulares.

asvc – Este campo de 8 bits es facultativo. Cada bit (0-7) identifica con qué servicio(s) principal(es) se asocia este servicio asociado. El bit situado más a la izquierda, que es el bit 7 indica si puede reproducirse este servicio asociado junto con el número de servicio principal 7. Si el bit tiene un valor de 1, el servicio se asocia con un número de servicio principal 7. Si el bit tiene un valor de 0, el servicio no se asocia con un número de servicio principal 7.

adicional_info – Estos bytes facultativos se reservan para utilización futura.

4.3.3 Tipos de componente (component_type) AC-3

El Cuadro 56 muestra la asignación de valores de component_type, en el component_descriptor, para el caso en que el valor de stream_content se fije en 0×04, lo que indica la referencia a un tren AC-3.

4.4 Tamaño de la memoria tampón de audio STD

El tamaño de la memoria tampón de audio principal, BS_n, tendrá un valor fijo de 5 696 bytes. Para la obtención de BS_n de trenes elementales de audio véase ISO/CEI 13818-1 (ISO/IEC 13818-1 [1996] Information technology – Generic coding of moving pictures and associated audio information – Systems Part 1. International Standard.)

CUADRO 56

Asignaciones del valor del byte de component_type de AC-3

Valores del byte de component_type (determinaciones permitidas)								**Descripción**
Bandera de estado reservado	**Bandera de servicio pleno**	**Bandera de tipo de servicio**			**Número de banderas de canales**			
b7	b6	b5	b4	b3	b2	b1	b0	
1	X	X	X	X	X	X	X	Reservado
0	X	X	X	X	X	X	X	Interpretar b0-b6 como se indica más adelante
	1	X	X	X	X	X	X	El tren de audio decodificado es un servicio pleno (adecuado para decodificación y presentación al oyente)
	0							Se pretende que el tren de audio decodificado se combine con otro tren de audio decodificado antes de la presentación al oyente
	X	X	X	X	0	0	0	Mono
					0	0	1	Modo 1+1
					0	1	0	2 canales (estéreo)
					0	1	1	2 canales codificados Dolby (estéreo)
					1	0	0	Audio multicanal (>2 canales)
					1	0	1	Reservado
					1	1	0	Reservado
					1	1	1	Reservado
	1	0	0	0	X	X	X	Principal completo (CM)
	0	0	0	1				Música y efectos (ME)
	X	0	1	0				Degradado visualmente (VI)
	X	0	1	1				Degradado auditivamente (HI)
	0	1	0	0				Diálogo (D)
	X	1	0	1	0	0	0	Comentario (C)
	1	1	1	0				Emergencia (E)
	0	1	1	1				Voz superpuesta (VO)
	1	1	1	1	X	X	X	Karaoke (prohibido mono y «1+1»)

5 Constricciones de PES

Este punto se aplica al Sistema A y al Sistema B.

5.1 Codificación

En algunas aplicaciones, el decodificador de audio puede ser capaz de decodificar simultáneamente dos trenes elementales que contienen diferentes elementos de programa, y luego combinar los elementos de programa en un programa completo.

La mayor parte de los elementos de programa se encuentran en el *servicio audio principal*. Otro elemento de programa (como una narración del contenido de imagen destinada al oyente con deficiencias visuales) se puede encontrar en el *servicio audio asociado*.

Para reproducir el audio de los dos trenes elementales en un sincronismo de muestra exacto, es necesario que los codificadores de trenes elementales de audio original hayan codificado síncronamente las dos tramas de elementos de programa de audio, es decir, si el tren de audio 1 ha tomado la muestra 0 de la trama n en el instante $t0$, el tren de audio 2 debe tomar también la trama n que comienza con su muestra 0 en el mismo instante $t0$. Si la codificación de múltiples servicios de audio se efectúa con sincronismo de trama y de muestra, y se prevé que la decodificación sea con sincronismo de trama y de muestra, los paquetes PES de esos servicios de audio contendrán valores idénticos de PTS que se refieren a las unidades de acceso audio destinadas a la decodificación síncrona.

Los servicios de audio que se han de combinar juntos para la reproducción se codificarán a una velocidad de muestra idéntica.

5.2 Decodificación

Si las unidades de acceso de audio de dos servicios de audio que se van a decodificar simultáneamente tienen valores idénticos de PTS indicados en sus correspondientes encabezamientos PES, las correspondientes unidades de acceso de audio se presentarán al decodificador audio para una decodificación síncrona simultánea. La decodificación síncrona significa que para las correspondientes tramas audio (unidades de acceso) las correspondientes muestras de audio se presentan en el mismo instante.

Si los valores PTS no concuerdan (lo que indica que la codificación audio no se efectuó con sincronización de trama), las tramas de audio (unidades de acceso) del servicio de audio principal pueden presentarse al decodificador audio para la decodificación y presentación en el momento indicado por el PTS. Un servicio asociado que está siendo decodificado simultáneamente puede presentar sus tramas de audio (unidades de acceso) que están en alineación temporal más próximas (según indica el PTS) a las del servicio principal que está siendo decodificado, al decodificador de audio para decodificación simultánea. En este caso, el servicio asociado puede ser reproducido fuera de sincronismo hasta la mitad del tiempo de una trama. (Por lo general esto es satisfactorio; una narración visualmente degradada no requiere una sincronización muy precisa.)

6 Alineación de bytes

Este punto se aplica al Sistema A y al Sistema B.

El tren elemental AC-3 estará alineado en bytes en el interior del tren de datos MPEG-2, es decir, los 8 bits iniciales de una trama AC-3 residirán en un único byte que se cursa por el tren de datos MPEG-2.