

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 416 056**

51 Int. Cl.:

G10L 19/24 (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **04.07.2008 E 08806166 (8)**

97 Fecha y número de publicación de la concesión europea: **27.03.2013 EP 2176861**

54 Título: **Codificación jerárquica de señales digitales de audio**

30 Prioridad:

06.07.2007 FR 0756326

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

30.07.2013

73 Titular/es:

**FRANCE TELECOM (100.0%)
6 PLACE D'ALLERAY
75015 PARIS, FR**

72 Inventor/es:

**KOVESI, BALAZS y
RAGOT, STÉPHANE**

74 Agente/Representante:

PÉREZ BARQUÍN, Eliana

ES 2 416 056 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Codificación jerárquica de señales digitales de audio

5 La presente invención se refiere a un procedimiento jerárquico de codificación de datos sonoros más particularmente para una codificación mediante cuantificación escalar.

Esa codificación está particularmente adaptada para la transmisión y/o el almacenamiento de señales digitales tales como las señales de frecuencia de audio (palabra, música u otras).

10 La presente invención se relaciona más particularmente con la codificación de formas de onda tal como la codificación MIC (de "Modulación de Impulsos Codificados"), denominada PCM (de "*Pulse Code Modulation*") en inglés, en donde cada muestra de entrada se codifica individualmente, sin predicción.

15 El principio general de codificación/decodificación MIC especificado por la recomendación UIT-T G.711 es tal como el descrito con referencia a la figura 1. La señal de entrada se supone definida con una amplitud de banda mínima de [300-3400 Hz] y muestreada a 8 kHz, con una resolución de 16 bits por muestra (en el formato denominado "MIC lineal").

20 El codificador MIC 13 comprende un módulo de cuantificación Q_{MIC} 10 que recibe en la entrada la señal de entrada S. El índice de cuantificación I_{MIC} en la salida del módulo de cuantificación 10 se trasmite a través del canal de transmisión 11 al decodificador 14.

25 El decodificador MIC 14 recibe en la entrada los índices I'_{MIC} resultantes del canal de transmisión, versión eventualmente perturbada por unos errores binarios de I_{MIC} , y realiza una cuantificación inversa mediante el módulo de cuantificación inversa Q'_{MIC} 12 para obtener la señal codificada S'_{MIC} .

30 La codificación MIC normalizada UIT-T G.711 (de aquí en adelante denominada G. 711) realiza una compresión de la amplitud de las señales mediante una curva logarítmica antes de una cuantificación escalar uniforme, lo que permite obtener una relación de señal a ruido aproximadamente constante para una gran dinámica de las señales. El paso de cuantificación en el dominio de la señal original es proporcionar por lo tanto a la amplitud de las señales.

35 Las muestras sucesivas de la señal comprimida se cuantifican a 8 bits, es decir 256 niveles. En la red telefónica conmutada (RTC) denominada *Public Switched Telephone Network* (PSTN) en inglés, estos 8 bits se transmiten a una frecuencia 8 kHz para dar una velocidad de 64 kbit/s.

40 Una trama de la señal cuantificada de acuerdo con la norma G.711 está constituida por índices de cuantificación codificados a 8 bits. De ese modo, si la cuantificación inversa se implementa mediante una tabla, ésta consiste simplemente en un puntero mediante el índice de uno de los 256 valores decodificados posibles.

Por razones de complejidad de implementación, la compresión MIC se ha aproximado mediante una curva lineal por segmentos.

45 Se definen en la norma G.711 dos leyes de codificación, la ley A utilizada principalmente en Europa y la ley μ (μ) utilizada en América del Norte y en Japón.

50 Estas leyes de codificación permiten efectuar en la señal una compresión de la amplitud (o también "companding" en inglés). La amplitud de la señal se comprime así mediante una función no lineal en el codificador, se envía sobre un canal de transmisión y se descomprime mediante la función inversa en el decodificador. El interés de la compresión de la amplitud es que permite transformar la distribución de probabilidad de la amplitud de la señal de audio de entrada en una ley de probabilidad casi uniforme, en la que se puede aplicar una cuantificación escalar uniforme.

55 Las leyes de compresión de amplitud son en general unas leyes de tipo logarítmico que permiten por lo tanto codificar una señal muestreada con una resolución de 16 bits (en formato "PCM lineal") sobre 8 bits (en formato "PCM" del tipo ley A o μ).

Los 8 bits por muestra en G.711 se reparten de la manera siguiente tal como se representa en 15 en la figura 1:

60 - 1 bit de signo S (0 para un valor negativo, 1 en caso contrario), que lleva la referencia *sgn* en la figura 1,

- 3 bits para indicar el segmento (referencia ID-SEG en la figura 1), viniendo dado cada extremo del segmento por $256 \cdot 2^n$ para la ley A y $256 \cdot 2^n - 132$ para la ley μ , en donde $n = 0, 1, \dots, 7$. El paso de cuantificación se multiplica por tanto por 2 cuando se pasa sobre el segmento superior (a partir del 2º segmento para la ley A).

65 - 4 bits para indicar el emplazamiento en el segmento, llevando la referencia ID-POS en la figura 1.

Los últimos 7 bis constituyen por lo tanto el valor absoluto codificado. A continuación se estudiará de entrada el caso de la ley A, después se generalizarán los resultados para la ley μ . De acuerdo con la norma G.711 ley A, el índice final se obtiene invirtiendo cada segundo bit a partir del bit de peso más reducido (*Least Significant Bit* LSB en inglés). Esta ley de codificación permite tener una precisión de cuantificación escalar de 12 bits (por lo tanto un paso de cuantificación de 16) cuando se está en los dos primeros segmentos, después la precisión disminuye en 1 bit cuando el número de segmentos incrementa en 1.

Se puede observar que es posible realizar la cuantificación MIC G.711 a partir de una señal digital representada en 16 bits efectuando unas simples comparaciones entre la amplitud de la muestra a codificar y los umbrales de decisión del cuantificador. La utilización de una dicotomía acelera de manera significativa estas comparaciones. Esta solución necesita el almacenamiento de una tabla de 256 entradas, la Tabla 1 a continuación da un extracto de una tabla de ese tipo para la ley A G.711.

Tabla 1. Umbrales para la búsqueda por dicotomía.

Nº de intervalo	Umbral inferior	Umbral superior	Signo	Valor absoluto codificado	Índice final	Valor cuantificado
0	-32768	-31745	0	127	0x2a	-32256
1	-31744	-30721	0	126	0x2b	-31232
...
122	-96	-81	0	5	0x50	-88
123	-80	-65	0	4	0x51	-72
124	-64	-49	0	3	0x56	-56
125	-48	-33	0	2	0x57	-40
126	-32	-17	0	1	0x54	-24
127	-16	-1	0	0	0x55	-8
128	0	15	1	0	0xd5	8
129	16	31	1	1	0xd4	24
130	32	47	1	2	0xd7	40
131	48	63	1	3	0xd6	56
132	64	79	1	4	0xd1	72
133	80	95	1	5	0xd0	88
...
254	30720	31743	1	126	0xab	31232
255	31744	32767	1	127	0xaa	32256

Por ejemplo, una muestra de origen de la señal S a codificar con una amplitud igual a -75. En consecuencia, esta amplitud está comprendida en el intervalo [-80, -65] de la línea 123 (o "nivel" 123) de la tabla. La codificación de esta información consiste en entregar un índice final codificado, referenciado como I'_{Mic} en la figura 1 y en la tabla 1, que es igual a 0x51. En la decodificación, la operación de cuantificación inversa consiste por tanto en recuperar el índice $I'_{Mic} = 0x51$ y hacerle corresponder un valor cuantificado VQ, tal que $VQ = -72$. En consecuencia, la decodificación asigna a la amplitud de la muestra correspondiente, de la señal decodificada S'_{Mic} , este valor -72. Se resaltaré que se asignaría este mismo valor $VQ = -72$ a todas las muestras a decodificar y cuyo valor inicial tuviera un valor en el intervalo [-80, -65], o sea en todos los 16 valores posibles en el intervalo, lo que corresponde en este caso al paso de cuantificación de 16. Por el contrario, se resaltaré que se asignaría el mismo valor $VQ = 32256$ a todas las muestras cuya amplitud inicial estuviera en el intervalo [31744, 32767], o sea en todos los 1024 valores posibles, lo que corresponde a un paso de cuantificación de 1024.

La relación señal a ruido (RSR) obtenida mediante la codificación MIC es aproximadamente constante (~38 dB) para una dinámica amplia de las señales. El paso de cuantificación en el dominio de la señal original es proporcional a la amplitud de las señales. Esta relación señal a ruido no es suficiente para hacer el ruido de cuantificación inaudible en toda la banda de frecuencias 0-4000 Hz. Además, para las señales de niveles reducidos (que se codifican con el primer segmento) la RSR es muy mala.

La norma G.711 se considera generalmente como de buena calidad para las aplicaciones de telefonía en banda estrecha con los terminales limitados a la banda de [300-3400 Hz]. No obstante, la calidad no es satisfactoria cuando

se utiliza G.711 para otras aplicaciones como por ejemplo para los terminales de buena fidelidad en la banda [50, 4000 Hz] o para la extensión jerárquica en banda ampliada de la codificación G.711.

5 DAVIS A G, TURNBULL R S: "A Scaleable Audio Coded", EXTERNAL RESEARCH WEB SITE OF BRITISH TELECOM, 1998, divulga un método para hacer escalable un flujo de datos codificado mediante una cuantificación escalar efectuada muestra por muestra.

10 Existen en efecto unos métodos de codificación jerárquica que consisten en aportar una capa de mejora determinada a partir del ruido de codificación del codificador G.711. Este ruido de codificación se codifica entonces mediante una técnica diferente de G.711, que constituye la capa denominada de base (o capa de núcleo). Un método de este tipo de codificación jerárquica se describe por ejemplo en el documento de Y. Hiwasaki, H. Ohmuro, T. Mori, S Kurihara y A. Kataoka. "A G.711 embedded wideband speech coding for VoIP conferences", IEICE Trans. Inf. & Syst., Vol. E89-D, nº9, septiembre de 2006. Este tipo de método tiene el inconveniente de incrementar de manera muy significativa la complejidad en el codificador mientras que la codificación de tipo MIC es considerada como de reducida complejidad. Además, el ruido de codificación MIC, al ser un ruido blanco, por lo tanto no correlacionado, la codificación de este tipo de ruido es difícil de implementar puesto que las técnicas de compresión se basan esencialmente en las propiedades de extracción de la correlación de la señal a codificar.

20 La presente invención ofrece una solución que mejora la situación.

Con este fin, la invención propone un procedimiento de codificación como se enuncia en la reivindicación 1.

Así, se transmite un flujo de mejora al mismo tiempo que la trama binaria de índices de cuantificación.

25 Este flujo de extensión se determina aprovechando unos bits de peso reducido que no se utilizan durante la codificación. Este método tiene por tanto la ventaja de no añadir complejidad al codificador y aportar la mejora de calidad deseada añadiendo al decodificador la posibilidad de obtener una mejor precisión de decodificación.

30 En un modo de realización, los bits memorizados son los bits de peso más alto entre los bits que no lo son tenidos en cuenta en la trama binaria de índices de cuantificación.

35 Todos los bits dejados de lado durante la aplicación de la ley de codificación logarítmica no son recobrados forzosamente en el flujo de extensión. Es posible así determinar un flujo de extensión en función de las necesidades de calidad y disponibilidad en términos de velocidad.

En una variante de realización, el número de bits tenidos en cuenta para determinar el flujo de mejora es función de la velocidad disponible en el curso de una transmisión hacia un decodificador.

40 De ese modo, el flujo de extensión es modulable en el curso de la transmisión en función de la velocidad disponible.

La invención se adapta particularmente al caso en el que la etapa de cuantificación escalar es una cuantificación del tipo MIC de acuerdo con una ley de codificación logarítmica de compresión de la amplitud del tipo A o del tipo mu conforme a la norma ITU-T G.711.

45 La invención se aplica igualmente a un procedimiento de codificación como se enuncia en la reivindicación 5.

50 El decodificador que recibe unos bits de extensión, mejora de ese modo la precisión de su expansión o "descompresión" concatenando los bits de extensión recibidos con los presentes en la trama de índices de cuantificación recibido del flujo de base.

En un modo preferido de realización, el procedimiento comprende además una etapa de adaptación de un valor de redondeo en función del número de bits de extensión recibido para obtener la señal de audio decodificada.

55 La detección de la señal de audio codificada se adapta de ese modo en función del número de bits del flujo de extensión.

La invención se relaciona igualmente con un codificador de audio como se enuncia en la reivindicación 7.

La invención se relaciona con un decodificador de audio como se enuncia la reivindicación 8.

60 La invención concierne finalmente a un programa informático como se enuncia en la reivindicación 9.

Igualmente, la invención concierne a un programa informático como se enuncia en la reivindicación 10.

65 Surgirán otras características y ventajas de la invención más claramente con la lectura de la descripción a continuación, dada únicamente a título de ejemplo no limitativo, y realizada con referencia a los dibujos adjuntos, en

los que:

- La figura 1 ilustra un sistema de codificación/decodificación MIC G.711 clásico del estado de la técnica;

5 - la figura 2 ilustra un sistema de codificación/decodificación de acuerdo con la invención así como los procedimientos de acuerdo con la invención implementados por los elementos de este sistema;

- las figuras 3a y 3b representan los valores cuantificados con relación a los valores de entrada después de la aplicación de las leyes de codificación respectivas A y mu de acuerdo con la norma G.711;

10 - las figuras 4 y 5 representan una comparación con y sin implementación de la invención, de los valores cuantificados con relación a los valores de entrada después de la aplicación de las leyes de codificación A y mu respectivamente.

15 La figura 2 ilustra un sistema de codificación/decodificación de acuerdo con la invención.

Un codificador 23 comprende un cuantificador Q_{MIC} 20 adaptado para cuantificar la señal de entrada S para tener una trama de índices de cuantificación I_{MIC} que se transmiten por el canal de transmisión 21 hacia un decodificador 24.

20 En un modo particular de realización, este codificador es del tipo codificador MIC e implementa una ley de codificación del tipo A o mu tal como la descrita en la norma G.711.

La trama de índices de cuantificación obtenida se representa por lo tanto en 15 y está de acuerdo con la trama del tipo G.711 ley A o mu.

Se proponen en la norma G.711 unos métodos de implementación de las leyes de codificación A y mu. Consisten en determinar el índice final de cuantificación mediante unas operaciones simples de baja complejidad que evitan el almacenamiento de importantes tablas de valores.

30 De ese modo, el pseudocódigo representado en el anexo A-10 da un ejemplo de implementación de la ley A tal como se describe en la norma G.711 (con una aproximación lineal mediante fragmentos de la ley de compresión de amplitud). Una implementación concreta de este pseudocódigo se da igualmente a título de ejemplo en el anexo A-10. Esta implementación está de acuerdo con la recomendación ITU-T G.191 Software Tool Library (STL-2005), capítulo 13 "ITU-T Basic operators". Esta recomendación está accesible en el sitio de Internet del ITU:

http://www.itu.int/rec/T_REC-G.191-200508-I/en

40 Se ve en este pseudocódigo que el índice de cuantificación sobre 8 bits comprende el bit de signo (signe), el índice del segmento (exp) y la posición en el segmento (mant).

En una primera parte de esta codificación, se determina el bit de signo que se pone en la posición 0 como se indica en 15 en la figura 1. A continuación, se busca la posición del bit del peso más alto "pos" y se calcula el número de segmento que lo codifica en 3 bits y lo pone en la posición 1, 2 y 3 como se representa en 15 en la figura 1.

45 Los 4 bits que constituyen la posición en el segmento se ponen en las posiciones 4, 5, 6 y 7 como se representa en 15.

50 Hay siempre un desplazamiento de bits a la derecha de al menos 4 bits ($x = \text{shift_right}(x, \text{pos} - 4)$) y por lo tanto 4 bits perdidos;

No se utilizan por lo tanto más que los bits de peso más alto (*Most Significant Bit* MSB en inglés) para constituir la trama de índices de cuantificación. El mínimo variable de la variable "pos" para la codificación de acuerdo con la ley A es de 8. Hay por lo tanto, para todos los elementos, al menos 4 bits de peso más reducido que se pierden. Se efectúa así la compresión del proceso de compresión de amplitud.

60 Para una señal de entrada de una resolución de 16 bis por muestra (en formato "PCM lineal"), el paso más pequeño de cuantificación es 16, perdiéndose los 4 bits de peso más reducido. La tabla 2 a continuación da los umbrales y paso de cuantificación en función de cada segmento para la G.711 ley A.

Tabla 2. Paso de cuantificación G.711 ley A.

Segmento	Umbral inferior	Umbral superior	Paso de cuantificación
0	0	255	16
1	256	511	16

ES 2 416 056 T3

2	512	1023	32
3	1024	2047	64
4	2048	4095	128
5	4096	8191	256
6	8192	16383	512
7	16384	32767	1024

De la misma manera, la decodificación se puede implementar mediante unas operaciones simples como lo ilustra el pseudocódigo y la implementación ITU-T STL-2005 representados en el anexo A-11.

- 5 Se puede ver en el pseudocódigo que el signo (signe). El segmento (exp) del valor en el segmento (val) se encuentran a partir del índice de 8 bits (indice). Se aplica un valor de redondeo igual a 8, y que corresponde a la mitad del paso de cuantificación utilizado para un segmento, para tener el valor de la mitad del intervalo de cuantificación. De ese modo, se efectúa la inversión del proceso de compresión de la amplitud. Los bits menos significativos que han sido rechazados en la codificación son recuperados en este caso a continuación de la aproximación.

10 La versión de la ley mu de G.711 es similar a la ley A. La diferencia principal es que se añade 128 a los valores para asegurar que en el primer segmento el bit 7 es siempre igual a 1 lo que hace inútil la transmisión de este bit y por lo tanto incrementa la precisión del primer segmento (ninguna cuantificación 8 en el primer segmento contra 16 en la ley A). Esto permite igualmente un tratamiento idéntico de todos los segmentos. Además se añade 4 (por lo tanto $128 + 4 = 132$ al total) para el redondeo para tener un nivel 0 entre los valores cuantificados (la ley A no tiene el nivel 0, los valores más pequeños son 8 ó -8). El precio de esta mejor resolución en el primer segmento es el desplazamiento de todos los elementos en 132. La Tabla 3 a continuación da los umbrales y el paso de cuantificación en función de cada segmento para la G.711 ley mu.

20

Tabla 3. Paso de cuantificación G.711 ley mu.

Segmento	Umbral inferior	Umbral superior	Paso de cuantificación
0	0	123	8
1	124	379	16
2	380	891	32
3	892	1915	64
4	1916	3963	128
5	3964	8059	256
6	8060	16251	512
7	16252	32635	1024

Las figuras 3a y 3b permiten comparar la resolución de estas dos leyes para los 512 primeros valores.

- 25 De la misma manera que para la ley A, un método de implementación sin almacenamiento de tablas de valores se da mediante un ejemplo de pseudocódigo de codificación de acuerdo con la norma G.711 ley mu, representado en el anexo A-12.

30 De la misma manera que para la ley A, se ve en el pseudocódigo que hay siempre un desplazamiento de bits a la derecha de al menos 3 bits ($x = \text{shift_right}(x, \text{pos} - 4)$), siendo el valor mínimo de "pos" 7 para la ley mu.

No se utilizan más que los bits de peso más alto (MSB) para constituir la trama de índices de cuantificación y efectuar de ese modo la etapa de compresión de amplitud.

- 35 El valor mínimo de la variable "pos" para la codificación de acuerdo con la ley mu es de 7, puesto que, como se ha mencionado anteriormente, en el caso de la ley mu el primer segmento es tratado de la misma manera que los otros segmentos. Hay por lo tanto para todos los segmentos al menos 3 bits de peso más reducido que se pierden.

40 Como para la ley A, la codificación se puede efectuar simplemente mediante un algoritmo simple, dándose un ejemplo en el anexo A-13.

El codificador 23 de acuerdo con la invención extrae parte del método de codificación de acuerdo con las leyes A o

mu memorizando en un espacio de memoria, representado en 27, una parte de los bits de peso más reducido que no han sido tenidos en cuenta para la codificación de la trama binaria de índices de cuantificación I_{MIC} .

5 De ese modo, como se ha mencionado anteriormente para la codificación logarítmica de acuerdo con las leyes A o mu, se pueden memorizar al menos 3 bits para todos los segmentos.

El número de bits perdidos por los métodos de codificación de acuerdo con la ley A o mu, aumenta con el número del segmento, hasta 10 bits para el último segmento.

10 El procedimiento de acuerdo con la invención permite recuperar al menos los bits de peso más alto de entre estos bits perdidos.

Para determinar un flujo de mejora de una velocidad de 16 kbit/s, por lo tanto de 2 bits por muestra, el procedimiento de acuerdo con la invención memorizará en la memoria 27, los dos bits de peso más alto de entre los bits que no han sido tenidos en cuenta en la operación de compresión, para determinar la trama de índices de cuantificación.

15 Estos bits se recuperan para determinar en 28 mediante unos medios de determinación del flujo de extensión, el flujo de mejora I_{EXT} . Este flujo de mejora se transmite a continuación a través de otro canal de transmisión 25 hacia un decodificador 24.

20 De ese modo, el decodificador 24 comprende un cuantificador inverso, en este caso un cuantificador MIC inverso Q^{-1}_{MIC} 22, recibe en paralelo el flujo de base I'_{MIC} y el flujo de mejora I'_{EXT} .

25 Estos flujos I'_{MIC} y I'_{EXT} son unas versiones eventualmente perturbadas por unos errores binarios de I_{MIC} y I_{EXT} respectivamente.

30 En caso de recepción de este flujo de mejora por los medios de recepción 29 del decodificador 24, el decodificador tendrá entonces una precisión más grande para la colocación de la muestra decodificada en el segmento. Para ello, concatena los bits de extensión a los bits recibidos en el flujo de base I'_{MIC} mediante los medios 30 de concatenación de bits, para a continuación efectuar una cuantificación inversa en 22.

35 En efecto, la aportación de un bit suplementario permite multiplicar por dos el número de niveles de los segmentos. El hecho de duplicar el número de niveles, incrementa también la relación señal a ruido en 6 dB. De ese modo, para cada bit añadido en el flujo de mejora y recibido en el decodificador, la relación señal a ruido se incrementara en 6 dB, lo que incrementa de ese modo la calidad de la señal decodificada sin por lo tanto incrementar considerablemente la complejidad en el codificador.

40 En el ejemplo ilustrado en la figura 2, el flujo de mejora I_{EXT} está constituido por dos bits de extensión por muestra, es decir una velocidad de 16 bit/s. Estos bits de extensión se pueden obtener realizando un desplazamiento de bits en dos operaciones como lo muestra el pseudocódigo representado en el anexo A-14.

45 Se puede ver que en lugar de desplazar en un único golpe los bits en "pos-4" posiciones para guardar nada más que los 5 bits de peso más alto, como es el caso en el codificación según la ley A, se desplazan en un primer tiempo 2 posiciones de menos (por lo tanto "pos-6" posiciones) para guardar los 7 bits de peso más alto y se memorizan en 27 los dos últimos bits. A continuación, en otra etapa, se desplazan aún más los dos bits para obtener los 5 bits de peso alto cuyo primer bit, siempre a 1, no se transmite. Los otros 4 se utilizan para el flujo de base.

Los dos bits memorizados se envían en el flujo de extensión.

50 Como se representa en la figura 2, se puede considerar que estos dos bits de extensión son del 8º y el 9º bit de la señal comprimida.

55 El pseudocódigo que permite realizar el conjunto de operaciones en el codificador para la ley A se da en el anexo A-15.

Se ve que las diferencias con relación a la codificación G.711 clásica (pasos subrayados y en grueso en el anexo) son las etapas de desplazamiento en dos tiempos como se ha expuesto anteriormente y la toma en consideración de los dos bits memorizados para determinar el flujo de mejora "ext" y transmitirlo.

60 Igualmente para la implementación de la ley mu, el pseudocódigo correspondiente para la codificación se representa en el anexo A-16.

Se remarcan las mismas diferencias con la codificación clásica que para la codificación de acuerdo con la ley A.

65 La figura 4 muestra una comparación de los valores cuantificados con relación a los valores de entrada entre la ley A clásica (en puntos) y la ley A con extensión de dos bits por muestra (trazado continuo), para los 128 primeros

valores.

Igualmente, la figura 5 muestra una comparación de los valores cuantificados con relación a los valores de entrada entre la ley mu clásica (en puntos) y la ley mu con extensión de dos bits por muestra (trazado continuo), para los 128 primeros valores.

Con la recepción del flujo de mejora I'_{EXT} , el decodificador concatena en 30 los bits de extensión así recibidos después de los bits de posición del flujo de base I'_{MIC} para efectuar la descompresión de amplitud —o expansión— que es la operación inversa del proceso de compresión de amplitud.

La toma en consideración de estos bits suplementarios permite obtener de ese modo una precisión más grande del emplazamiento de la muestra decodificada en el segmento.

En efecto, para un bit suplementario, el segmento se divide en dos. La precisión sobre el emplazamiento en el segmento del valor decodificado es entonces más importante.

El valor de redondeo “roundval” que permite encontrar el valor del centro del segmento se adapta también en función del número de bits de extensión recibidos.

La información del número de los bits de extensión recibidos se da por ejemplo por medio de una señalización externa como se representa por la flecha 26 en la figura 2.

Esta información podría deducirse igualmente directamente mediante el análisis del flujo de extensión.

Un ejemplo de decodificación que tiene en cuenta estos bits de extensión se da en el anexo A-17 mediante los pseudocódigos para la ley A y la ley mu respectivamente:

Las diferencias entre la decodificación clásica y la de la invención (pasos subrayados y en grueso en el anexo) representa la toma en consideración de los bits del flujo de extensión y la aplicación de un valor de redondeo “roundval”.

El codificador tal como se representa en la figura 2 comprende un procesador del tipo DSP (de “Digital Signal Processor”), no representado aquí, un espacio de memoria 27 para memorizar al menos los bits que servirán para determinar el flujo de extensión.

Este espacio de memoria 27 puede formar parte de un bloque de memoria que comprende además una memoria de almacenamiento y/o una memoria de trabajo.

El medio de almacenamiento puede comprender un programa informático que comprende unas instrucciones de código para la implementación de las etapas del procedimiento de codificación de acuerdo con la invención cuando se ejecutan mediante un procesador del codificador.

El programa informático puede estar igualmente almacenado en un soporte de memoria que pueda leer un lector del codificador o que pueda tele cargarse en el espacio de memoria del codificador.

Este codificador implementa así el procedimiento de acuerdo con la invención como se enuncia en la reivindicación 1.

Igualmente, el decodificador de acuerdo con la invención comprende un procesador del tipo DSP no representado aquí y es adecuado para poner en práctica el procedimiento como se enuncia en la reivindicación 5.

Este codificador comprende además un medio de almacenamiento (no representado), adecuado para almacenar un programa informático que comprende unas instrucciones de código para poner en práctica las etapas del procedimiento de decodificación de acuerdo con la invención cuando se ejecutan por el procesador del decodificador.

El programa informático se puede almacenar igualmente en un soporte de memoria que pueda leerse por un lector del decodificador o tele cargarse en el espacio de memoria del decodificador.

El ejemplo representado y explicado con referencia a la figura 2 se da para una capa de extensión de 2 bits por muestra. Este procedimiento se puede generalizar, por supuesto, para otro número de bits, por ejemplo 1, 2, 3 bits o más. El pseudocódigo correspondiente sería entonces como el representado en el anexo A-18.

Los “ext_bits” LSB de la variable “ext” se envían en el flujo de mejora.

Se ha de observar que el término “pos-4-ext_bits” puede ser negativo para ext_bits > 3 en los primeros segmentos y

según la ley utilizada (A o mu). Igualmente en estas condiciones el pseudocódigo dado funcionaría correctamente porque $\text{shift_right}(x, -v) = \text{shift_left}(x, v)$. En otros términos, en el caso en el que el número de bits de peso reducido que no son tenidos en cuenta en la trama de índices de cuantificación sea inferior al número de bits del flujo de extensión, en particular en los primeros segmentos, es suficiente completar en el flujo de extensión los bits faltantes por unos ceros. De ese modo, los bits de peso más alto del flujo de extensión serán los bits memorizados y recuperados de acuerdo con la invención, los bits de peso más reducido serán puestos a 0.

Para los segmentos siguientes, el número de bits memorizados aumenta, no será ya necesario completarlos por unos ceros.

Del mismo modo, la invención se aplica igualmente al caso de que en el curso de la transmisión la velocidad deba reducirse. En el caso de que el flujo de extensión comprenda dos bits, el bit de peso reducido de este flujo de extensión ya no se transmitirá.

El decodificador no recibe entonces más que un bit de extensión por muestra. El decodificador tal como se ha descrito en el código a título de ejemplo funcionará correctamente con esta capa de extensión reducida a un bit por muestra con la condición de que el bit de extensión recibido sea puesto en la variable "ext" en la posición 1, el bit de posición 0 de la variable "ext" se pone entonces a 0 y el valor de "roundval" se adapta en consecuencia.

El valor de la variable "roundval" tal como se utiliza en los ejemplos dados será por lo tanto función del número de bits recibidos por el codificador y de la ley utilizada (A o mu). La tabla 4 a continuación da el valor de la variable "roundval" en las diferentes situaciones.

Tabla 4. Valor de la variable "roundval" en diferentes configuraciones.

bits de mejora recibidos por el codificador	0	1	2	3
Ley A	8	4	2	1
Ley mu	4	2	1	0

Este ejemplo muestra por lo tanto otra ventaja de la solución presentada que es que el tren binario de la capa de extensión es jerárquico. Es posible por lo tanto disminuir su velocidad en el curso de la transmisión.

De ese modo, si se reciben dos bits por el decodificador, el incremento de la RSR es de 12 dB, si se recibe un bit, el incremento de la RSR es de 6 dB.

Por supuesto este ejemplo se puede generalizar igualmente, por ejemplo el codificador puede enviar 4 bits por muestra en la capa de extensión y el decodificador puede recibir 4, 3, 2, 1 o 0 de estos bits, la calidad de la señal decodificada será proporcional al número de los bits de extensión recibidos.

Se puede observar en los pseudocódigos dados que la complejidad adicional de la decodificación de la capa de extensión es solamente de dos operaciones por muestra en el codificador y 4 operaciones por muestra en el decodificador, o sea ~0,05 millón de operaciones ponderadas por segundo, o "*Weighted Million Operations per Second*" (WMOPS) en inglés, lo que es despreciable. Esta reducida complejidad se puede explotar en el caso de una codificación jerárquica que extiende G.711 mientras permite por ejemplo en unas aplicaciones de conferencia de audio realizar una mezcla "*convencional*" de reducida complejidad de los flujos G.711 o G.711 extendido de acuerdo con la invención, mientras que en el artículo de Hiwasaki se implementa una mezcla denominada "*parcial*", que implica una degradación de la calidad con relación a la mezcla convencional, para limitar la complejidad de la mezcla con la codificación G.711 escalable.

En un modo de realización alternativo, la invención se realizará sin seguir los algoritmos especificados anteriormente mediante el pseudocódigo, sino calculando previamente y almacenando en unas tablas en el codificador y/o en el decodificador los niveles que permitan obtener los bits de extensión. Esta solución tiene no obstante el inconveniente de que necesita una capacidad más grande de memoria a la vez en el codificador y en el decodificador para una ganancia en complejidad reducida.

Anexos

A-10:

```
function lin_to_Alaw(input_16bit)
  x = input_16bit
  signe = 0x80 /*supposing + */
  if x < 0
    x = ~x /*abs(x) - 1*/
  signe = 0
```

```

end
if x > 255 /* 1st bit 1 + 4 saved bits */
    pos = cherche_position_bit_1_poids_fort(x) /* 14 >= pos >= 8 */
    exp = shift_left(pos - 7, 4)
5     x = shift_right(x, pos - 4)
    mant = x - 16 /* remove leading 1 */
else
    exp = 0
    mant = shift_right(x, 4)
10    end
    ind_tmp = signe + exp + mant
    indice = xor(ind_tmp, 0x0055) /* toggle odd bits */
return indice /* only 8LSB bits are used */

```

15 Versión ITU-T STL-2005:

```

short lin_to_Alaw(short input_16bit) {
    short x, signe, pos, exp, mant, ind_tmp, indice;
    x = input_16bit;
20    signe = 0x80; /*supposing + */
    IF(x < 0)
    {
        x = s_xor(x, (short)0xFFFF); /*abs(x) - 1*/
        signe = 0;
25    }
    IF (sub(x, 255) > 0) /* 1st bit 1 + 4 saved bits */
    {
        pos = sub(14, norm_s(x)); /* 14 >= pos >= 8 */
        exp = shl(sub(pos, 7), 4);
        x = shr(x, sub(pos, 4));
30        mant = sub(x, 16); /* remove leading 1 */
    }
    ELSE
    {
35        exp = 0;
        mant = shr(x, 4);
    }
    ind_tmp = add(signe, add(exp, mant) );
    indice = s_xor(ind_tmp, 0x0055); /* toggle odd bits */
40    return(indice); /* only 8LSB bits are used */
}

```

A-11:

```

45    function Alaw_to_lin(indice)
        signe = and(indice, 0x80);
        y = and(xor(indice, 0x0055), 0x7F) /* without sign */
        exp = shift_right(y, 4)
        val = shift_left(and(y, 0xF), 4) + 8 /* with rounding */
50        if exp > 0
            val = shift_left(val + 256, exp - 1) /* add leading 1 */
        end
        if signe == 0 /* sign bit ==0 → negative value */
            val = -val
55        end
    return val

```

Versión ITU-T STL-2005 :

```

60    short Alaw_to_lin (short indice)
    {
        short y, signe, exp, val;
        signe = s_and(indice, 0x80);
        y = s_and(s_xor (indice, 0x0055), 0x7F); /* without sign */
65        exp = shr(y, 4);
        val = add(shl(s_and(y, 0xF), 4), 8); /* rounding */
    }

```

```

    if(exp > 0)
    {
        val = shl(add(val, 256), sub(exp, 1)); /*add leading 1 */
    }
5   if(signe == 0) /* sign bit ==0 'negative value */
    {
        val = negate(val);
    }
10  return(val);
}

```

A-12:

```

function lin_to_mlaw(input_16bit)
15  x = input_16bit
    signe = 0x80 /* supposing + */
    if x > 32635 /* to avoid overflow after adding 132*/
        x = 32635
    end
20  if x < -32635
        x = -32635
    end
    if x < 0
        x = ~x /*abs(x) - 1*/
25  signe = 0x00
    end
    x = x + 132
    /* always 1st bit 1 + 4 saved bits */
    pos = cherche_position_bit_1_poids_fort(x) /* 14 >= pos >= 7 */
30  exp = shift_left(pos - 7, 4)
    x = shift_right(x, pos - 4)
    mant = x - 16 /* remove leading 1 */
    ind_tmp = signe + exp + mant
    indice = xor(ind_tmp, 0x007F) /* toggle all bits */
35  return indice /* only 8LSB bits are used */

```

A-13:

```

function mlaw_to_lin(indice)
40  signe = and(indice, 0x80);
    y = and(xor(indice, 0x00FF), 0x7F) /* without sign */
    exp = shift_right(y, 4)
    val = shift_left(and(y, 0xF), 3) + 132 /* leading 1 & rounding */
    val = shift_left(val, exp) - 132 /* suppress encoder offset */
45  if signe == 0 /* sign bit ==0 → negative value */
        val = -val
    end
    return val

```

50 A-14:

```

x = shift_right(x, pos - 6) /* first part of shift*/
ext = and(x, 0x3) /*save last two bits*/
x = shift_right(x, 2) /* finish shift*/

```

55

A-15:

```

function lin_to_Alaw_enh(input_16bit)
60  x = input_16bit
    signe = 0x80 /*supposing + */
    if x < 0
        x = ~x /*abs(x) - 1*/
        signe = 0
    end
65  if x > 255 /* 1st bit 1 + 4 saved bits */
        pos = cherche_position_bit_1_poids_fort (x) /* 14 >= pos >= 8 */

```

```

    exp = shift_left(pos - 7, 4)
    x = shift_right(x, pos - 6) /* first part of shift */
    ext = and(x, 0x3) /* save last to bits */
    x = shift_right(x, 2) /* finish shift */
5   mant = x - 16 /* remove leading 1 */
    else
        exp = 0
        x = shift_right(x, 2)
        ext = and(x, 0x3) /* save last two bits */
10   x = shift_right(x, 2) /* finish shift */
    end
    ind_tmp = signe + exp + mant
    indice = xor(ind_tmp, 0x0055) /* toggle odd bits */
return indice, ext /* only 8LSB bits are used in indice and 2LSB bits in ext*/
15
A-16:

```

```

function lin_to_mlaw_enh(input_16bit)
    x = input_16bit
20   signe = 0x80 /* supposing + */
    if x > 32635 /* to avoid overflow after adding 132*/
        x = 32635
    end
    if x < -32635
25   x = -32635
    end
    if x < 0
        x = ~x /*abs(x) - 1*/
        signe = 0x00
30   end
    x = x + 132
    /* always 1st bit 1 + 4 saved bits */
    pos = cherche_position_bit_1_poids_fort(x) /* 14 >= pos >=
7 */
35   exp = shift_left(pos - 7, 4)
    x = shift_right(x, pos - 6) /* first part of shift */
    ext = and(x, 0x3) /* save last two bits */
    x = shift_right(x, 2) /* finish shift */
    mant = x - 16 /* remove leading 1 */
40   ind_tmp = signe + exp + mant
    indice = xor(ind_tmp, 0x007F) /* toggle all bits */
return indice, ext /* only 8LSB bits are used in indice and 2LSB bits in ext*/

```

A-17:

```

45 Ley A:
function Alaw_to_lin_enh(indice, ext, roundval)
    signe = and(indice, 0x80);
    y = and(xor(indice, 0x0055), 0x7F) /* without sign */
50   exp = shift_right(y, 4)
    ext = shift_left(and(ext, 0x03), 2) /* put extension bits in
position 2 & 3 */
    val = shift_left(and(y, 0xF), 4) + ext + roundval /* with rounding */
    if exp > 0
55   val = shift_left(val + 256, exp - 1) /* adding leading 1 */
    end
    if signe == 0 /* sign bit ==0 → negative value */
        val = -val
    end
60   return val

```

```

Ley mu:
function mlaw_to_lin_enh(indice, ext, roundval)
    signe = and(indice, 0x80);
65   y = and(xor(indice, 0x007F), 0x7F) /* without sign */
    exp = shift_right(y, 4)

```

```
    ext = shift_left(and(ext, 0x03), 1) /* put extension bits in position 1 & 2 */
    val = shift_left(and(y, 0xF), 3) + 128 + ext + roundval /* leading 1 & rounding */
    val = shift_left(val, exp) - 132 /* suppress encoder offset */
    if signe == 0 /* sign bit ==0 → negative value */
5      val = -val
      end
    return val

A-18:
10 x = shift_right(x, pos - 4 - ext_bits) /* first part of shift*/
    ext = and(x, shift_left(1, ext_bits) - 1) /* last ext_bits bits*/
    x = shift_right(x, ext_bits) /* finish shift*/
```

REIVINDICACIONES

1. Procedimiento de codificación mediante cuantificación escalar de las muestras de una señal de audio digital (S), siendo codificadas las muestras en un número predeterminado de bits para obtener una trama binaria de índices de cuantificación (I_{MIC}), efectuándose la codificación de acuerdo con una ley logarítmica de compresión de la amplitud, no siendo tenidos en cuenta, en la operación de compresión para formar la trama binaria de índices de cuantificación, un número predeterminado de bits de peso reducido de la señal de audio digital en formato MIC lineal, caracterizado porque comprende las etapas siguientes:
- 5
- 10 - memorización (27) de al menos una parte de los bits de peso reducido que no son tenidos en cuenta en la operación de compresión para formar la trama binaria del índice de cuantificación;
- determinación (28) de un flujo de mejora (I_{EXT}) que comprende al menos un bit así memorizado.
- 15 2. Procedimiento de acuerdo con la reivindicación 1, caracterizado porque los bits memorizados son los bits de peso más alto entre los bits que no son tenidos en cuenta en la operación de compresión para formar la trama binaria de índices de cuantificación.
3. Procedimiento de acuerdo con una de las reivindicaciones 1 a 2, caracterizado porque el número de bits tenidos en cuenta para determinar el flujo de mejora es función de la velocidad disponible en el curso de una transmisión hacia un decodificador.
- 20 4. Procedimiento de acuerdo con una de las reivindicaciones 1 a 3, caracterizado porque la etapa de cuantificación escalar es una cuantificación del tipo MIC según una ley de codificación logarítmica de compresión de la amplitud del tipo A o del tipo μ , conforme a la norma ITU-T G.711.
- 25 5. Procedimiento de decodificación de una trama binaria de índices de cuantificación (I'_{MIC}) que comprende un número predeterminado de bits mediante una capa de cuantificación inversa (22) y según una ley logarítmica de compresión de la amplitud, caracterizado porque comprende las etapas siguientes:
- 30 - recepción (29) de un flujo de mejora (I'_{EXT}) que comprende uno o varios bits de extensión determinados de acuerdo con un procedimiento de codificación según la reivindicación 1;
- concatenación (30) de los bits de extensión detrás de los bits resultantes de la trama binaria para obtener una señal de audio decodificada.
- 35 6. El procedimiento de decodificación de acuerdo con la reivindicación 5, caracterizado porque comprende además una etapa de adaptación de un valor de redondeo en función del número de los bits de extensión recibidos para obtener la señal de audio decodificada.
- 40 7. Codificador de audio que comprende un módulo de cuantificación escalar (20) de las muestras de la señal de audio digital (S), estando codificadas las muestras en un número predeterminado de bits para obtener una trama binaria de índices de cuantificación (I_{MIC}), efectuándose la codificación de acuerdo con una ley logarítmica de compresión de amplitud, no siendo tenidos en cuenta, en la operación de compresión para formar la trama binaria de índices de cuantificación, un número predeterminado de bits de peso reducido de la señal digital de audio en formato MIC lineal, caracterizado porque comprende:
- 45 - un espacio de memoria (27) adecuado para memorizar al menos una parte de los bits de peso reducido que no son tenidos en cuenta en la operación de compresión para formar la trama binaria de índices de cuantificación;
- 50 - unos medios de determinación (28) de un flujo de mejora (I_{EXT}) que comprende al menos un bit así memorizado.
8. Decodificador de audio adecuado para decodificar una trama binaria de índices de cuantificación (I'_{MIC}) que comprende el número predeterminado de bits mediante un módulo de cuantificación inversa (22) y de acuerdo con una ley logarítmica de compresión de amplitud caracterizado porque comprende:
- 55 - unos medios de recepción (29) de un flujo de mejora que comprende uno o varios bits de extensión determinados por un codificador según la reivindicación 7;
- unos medios de concatenación (30) de los bits de extensión detrás de los resultantes de la trama binaria para obtener una señal de audio decodificada.
- 60 9. Programa informático destinado a estar almacenado en una memoria de un codificador y/o un soporte de memoria adecuado para cooperar con un lector del codificador, que comprende unas instrucciones de código para la implementación de las etapas del procedimiento de codificación de acuerdo con una cualquiera de las reivindicaciones 1 a 4 cuando se ejecuta mediante un procesador del codificador.
- 65

10. Programa informático destinado a estar almacenado en una memoria de un decodificador y/o un soporte de memoria adecuado para cooperar con un lector del decodificador, que comprende unas instrucciones de código para la implementación de las etapas del procedimiento de decodificación de acuerdo con una cualquiera de las reivindicaciones 5 a 6 cuando se ejecuta mediante un procesador del decodificador.

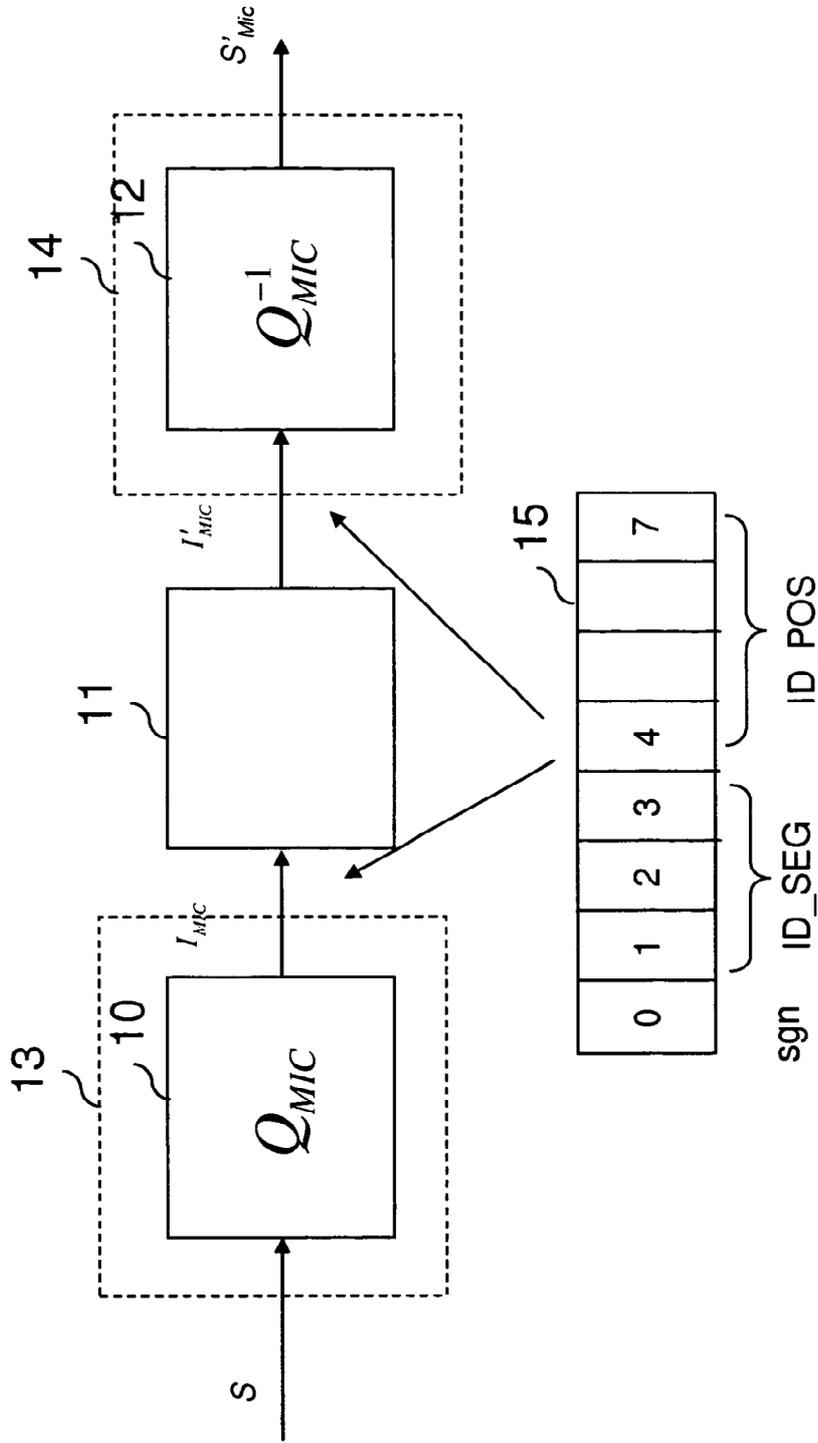


Fig. 1 (Estado de la técnica)

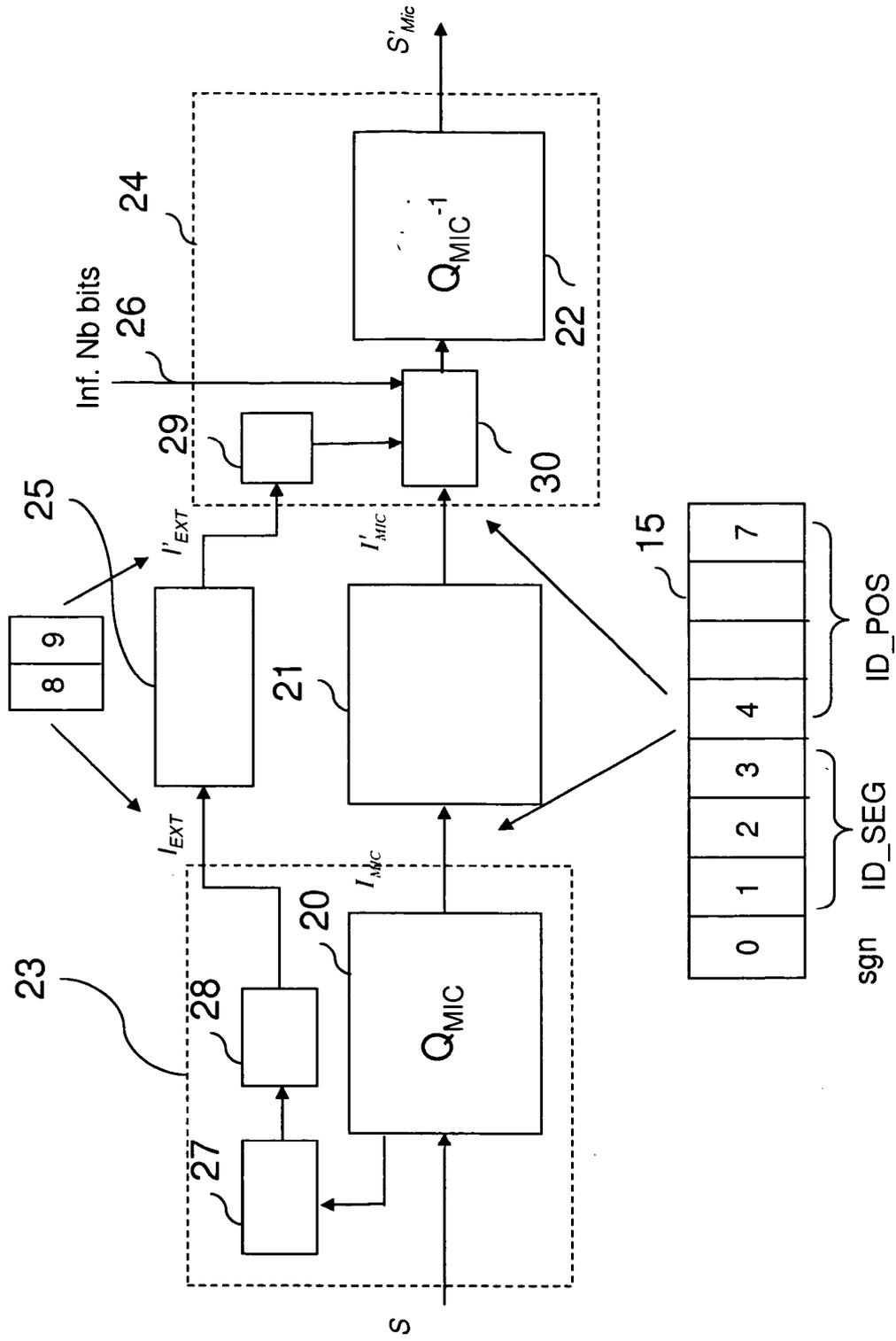


Fig.2

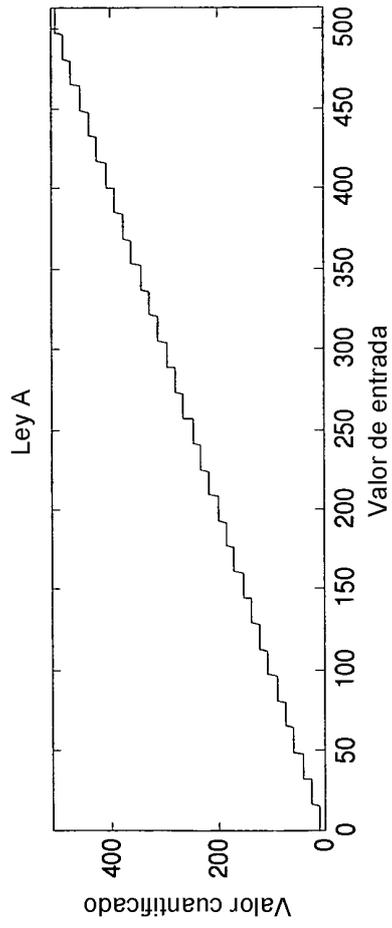


Fig.3a

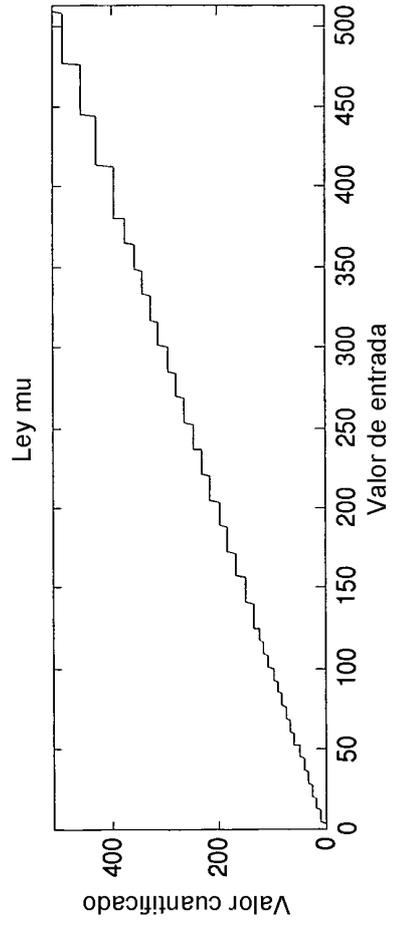


Fig.3b

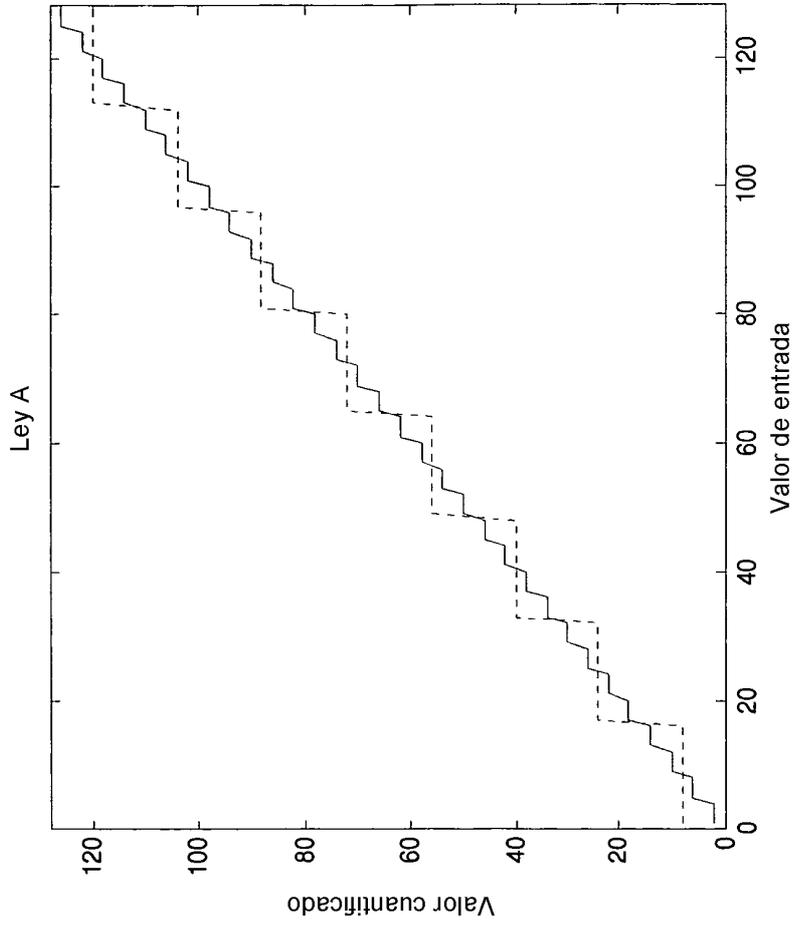


Fig.4

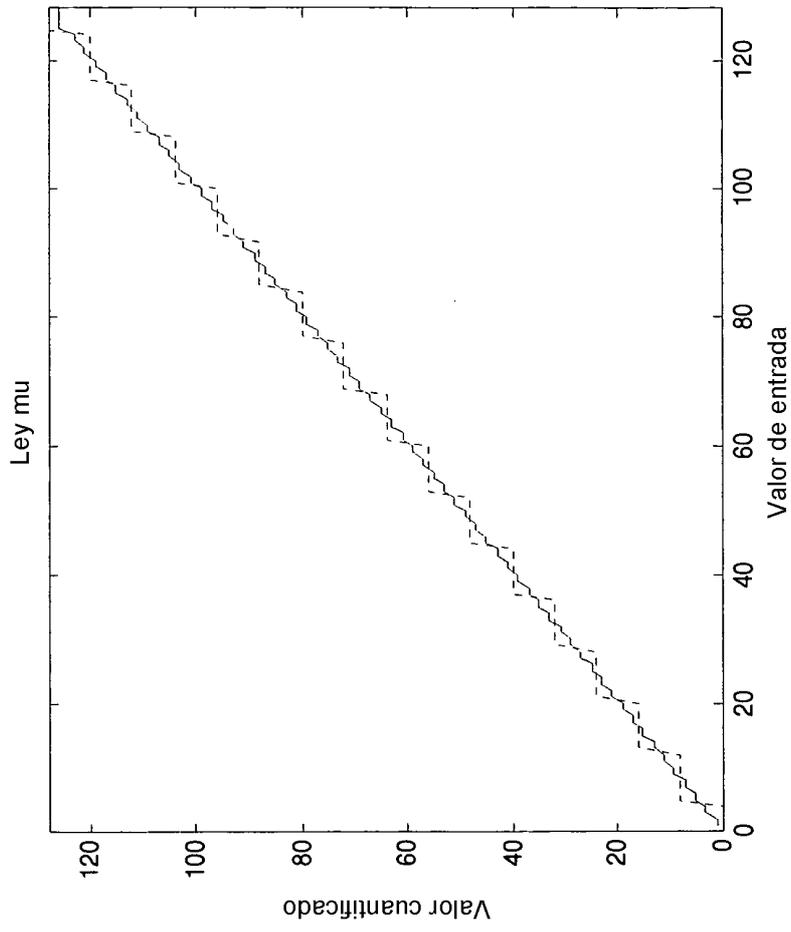


Fig.5