Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2331611 C 2001/09/11

(11)(21) 2 331 611

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(22) Date de dépôt/Filing Date: 1996/11/21

(41) Mise à la disp. pub./Open to Public Insp.: 1997/06/12

(45) Date de délivrance/Issue Date: 2001/09/11

(62) Demande originale/Original Application: 2 238 026

(30) Priorités/Priorities: 1995/12/01 (60/007,896) US;
1996/05/02 (08/642,254) US

(51) Cl.Int.^{7}/Int.Cl.^{7} G10L 19/12

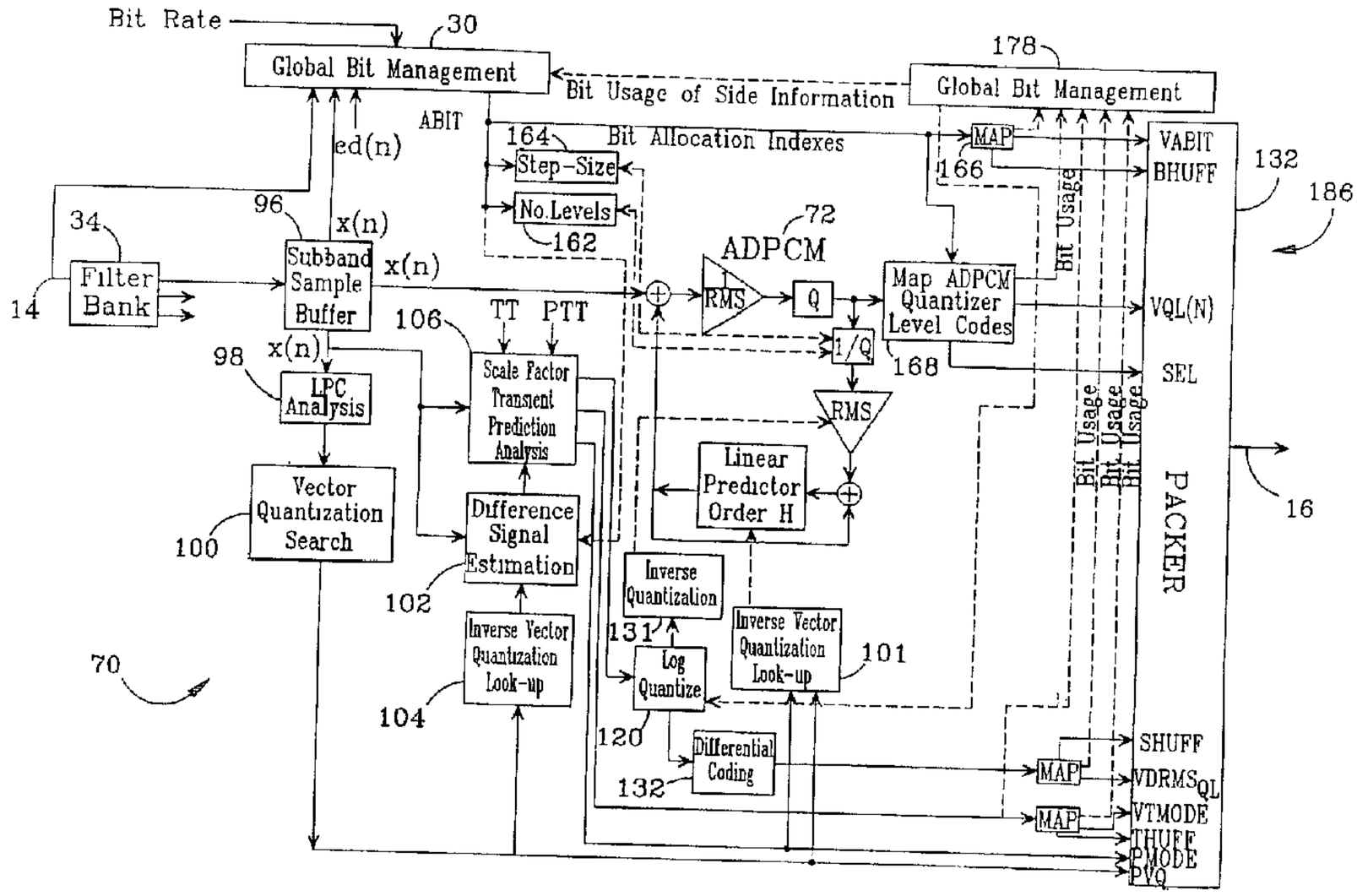
(72) Inventeurs/Inventors:
SMITH, WILLIAM P., US;
SMYTH, MICHAEL H., GB;
SMYTH, STEPHEN M., US

(73) Propriétaire/Owner:
DIGITAL THEATER SYSTEMS, INC., US

(74) Agent: OYEN WIGGS GREEN & MUTALA

(54) Titre : CODEUR PREDICTIF EN SOUS-BANDE MULTIVOIE A ATTRIBUTION PSYCHO-ACOUSTIQUE ADAPTATIVE DES BITS

(54) Title: MULTI-CHANNEL PREDICTIVE SUBBAND CODER USING PSYCHOACOUSTIC ADAPTIVE BIT ALLOCATION



(57) Abrégé/Abstract:
A subband audio coder (12) employs perfect/non-perfect reconstruction filters (34), predictive/non-predictive subband encoding (72), transient analysis (106), and psycho-acoustic/minimum mean-square-error (mmse) bit allocation (30) over time, frequency and the multiple audio channels to encode/decode a data stream to generate high fidelity reconstructed audio. The audio coder windows (64) the multi-channel audio signal such that the frame size, i.e., number of bytes, is constrained to lie in a desired range, and formats the encoded data so that the individual subframes can be played back as they are received thereby reducing latency. Furthermore, the audio coder processes the baseband portion (0-24kHz) of the audio bandwidth for sampling frequencies of 48kHz and higher with the same encoding/decoding algorithm so that audio coder architecture is future compatible.

Abstract of the Disclosure

A subband audio coder (12) employs perfect/non-perfect reconstruction filters (34), predictive/non-predictive subband encoding (72), transient analysis (106), and psycho-acoustic/minimum mean-square-error (mmse) bit allocation (30) over time, frequency and the multiple audio channels to encode/decode a data stream to generate high fidelity reconstructed audio. The audio coder windows (64) the multi-channel audio signal such that the frame size, i.e., number of bytes, is constrained to lie in a desired range, and formats the encoded data so that the individual subframes can be played back as they are received thereby reducing latency. Furthermore, the audio coder processes the baseband portion (0-24kHz) of the audio bandwidth for sampling frequencies of 48kHz and higher with the same encoding/decoding algorithm so that audio coder architecture is future compatible.

MULTI-CHANNEL PREDICTIVE SUBBAND CODER USING PSYCHOACOUSTIC ADAPTIVE BIT ALLOCATION

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to high quality encoding and decoding of multi-channel audio signals and more specifically to a subband encoder that employs perfect/non-perfect reconstruction filters, predictive/non-predictive subband encoding, transient analysis, and psycho-acousti c/minimum mean-square-error (mmse) bit allocation over time, frequency and the multiple audio channels to generate a data stream with a constrained decoding computational load.

Description of the Related Art

Known high quality audio and music coders can be divided into two broad classes of schemes. First, medium to high frequency resolution subband/transform coders which adaptively quantize the subband or coefficient samples within the analysis window according to a psychoacoustic mask calculation. Second, Low resolution subband coders which make-up for their poor frequency resolution by processing the subband samples using ADPCM.

The first class of coders exploit the large short-term spectral variances of general music signals by allowing the bit-allocations to adapt according to the spectral energy of the signal. The high resolution of these coders allows the frequency transformed signal to be applied directly to the psychoacoustic model, which is based on a critical band

theory of hearing. Dolby's AC-3 audio coder, Todd et al., "AC-3: Flexible Perceptual Coding for Audio Transmission and Storage" Convention of the Audio Engineering Society, February, 1994, typically computes 1024-ffts on the respective PCM signals and applies a psychoacoustic model to the 1024 frequency coefficients in each channel to determine the bit rate for each coefficient. The Dolby system uses a transient analysis that reduces the window size to 256 samples to isolate the transients. The AC-3 coder uses a proprietary backward adaptation algorithm to decode the bit allocation. This reduces the amount of bit allocation information that is sent along side the encoded audio data. As a result, the bandwidth available to audio is increased over forward adaptive schemes which leads to an improvement in sound quality.

In the second class of coders, the quantization of the differential subband signals is either fixed or adapts to minimize the quantization noise power across all or some of the subbands, without any explicit reference to psychoacoustic masking theory. It is commonly accepted that a direct psychoacoustic distortion threshold cannot be applied to predictive/differential subband signals because of the difficulty in estimating the predictor performance ahead of the bit allocation process. The problems is further compounded by the interaction of quantization noise on the prediction process.

These coders work because perceptually critical audio signals are generally periodic over long periods of time. This periodicity is exploited by predictive differential quantization. Splitting the signal into a small number of sub-bands reduces the audible effects of noise modulation and allows the exploitation of long-term spectral variances in audio signals. If the number of subbands is increased, the prediction gain within each sub-band is reduced and at some point the prediction gain will tend to zero.

Digital Theater Systems, L.P. (DTS) makes use of an audio coder in which each PCM audio channel is filtered into

four subbands and each subband is encoded using a backward ADPCM encoder that adapts the predictor coefficients to the sub-band data. The bit allocation is fixed and the same for each channel, with the lower frequency subbands being assigned more bits than the higher frequency subbands. The bit allocation provides a fixed compression ratio, for example, 4:1. The DTS coder is described by Mike Smyth and Stephen Smyth, "APT-X100: A LOW-DELAY, LOW BIT-RATE, SUB-BAN D ADPCM AUDIO CODER FOR BROADCASTING," Proceedings of the 10th International AES Conference 1991, pp. 41-56.

Both types of audio coders have other common limitations. First, known audio coders encode/decode with a fixed frame size, i.e. the number of samples or period of time represented by a frame is fixed. As a result, as the encoded transmission rate increases relative to the sampling rate, the amount of data (bytes) in the frame also increases. Thus, the decoder buffer size must be designed to accommodate the worst case scenario to avoid data overflow. This increases the amount of RAM, which is a primary cost component of the decoder. Secondly, the known audio coders are not easily expandable to sampling frequencies greater than 48 kHz. To do so would make the existing decoders incompatible with the format required for the new encoders. This lack of future compatibility is a serious limitation. Furthermore, the known formats used to encode the PCM data require that the entire frame be read in by the decoder before playback can be initiated. This requires that the buffer size be limited to approximately 100ms blocks of data such that the delay or latency does not annoy the listener.

In addition, although these coders have encoding capability up to 24kHz, often times the higher subbands are dropped. This reduces the high frequency fidelity or ambiance of the reconstructed signal. Known encoders typically employ one of two types of error detection schemes. The most common is Read Solomon coding, in which the encoder adds error detection bits to the side information in the data stream. This facilitates the detection and correction

of any errors in the side information. However, errors in the audio data go undetected. Another approach is to check the frame and audio headers for invalid code states. For example, a particular 3-bit parameter may have only 3 valid states. If one of the other 5 states is identified then an error must have occurred. This only provides detection capability and does not detect errors in the audio data.

SUMMARY OF THE INVENTION

In view of the above problems, the present invention provides a multi-channel audio coder with the flexibility to accommodate a wide range of compression levels with better than CD quality at high bit rates and improved perceptual quality at low bit rates, with reduced playback latency, simplified error detection, improved pre-echo distortion, and future expandability to higher sampling rates.

This is accomplished with a subband coder that windows each audio channel into a sequence of audio frames, filters the frames into baseband and high frequency ranges, and decomposes each baseband signal into a plurality of subbands. The subband coder normally selects a non-perfect filter to decompose the baseband signal when the bit rate is low, but selects a perfect filter when the bit rate is sufficiently high. A high frequency coding stage encodes the high frequency signal independently of the baseband signal. A baseband coding stage includes a VQ and an ADPCM coder that encode the higher and lower frequency subbands, respectively. Each subband frame includes at least one subframe, each of which are further subdivided into a plurality of sub-subf rames. Each subframe is analyzed to estimate the prediction gain of the ADPCM coder, where the prediction capability is disabled when the prediction gain is low, and to detect transients to adjust the pre and post-transient SFs.

A global bit management (GBM) system allocates bits to each subframe by taking advantage of the differences between the multiple audio channels, the multiple subbands, and the

subframes within the current frame. The GBM system initially allocates bits to each subframe by calculating its SMR modified by the prediction gain to satisfy a psychoacoustic model. The GBM system then allocates any remaining bits according to a MMSE approach to either immediately switch to a MMSE allocation, lower the overall noise floor, or gradually morph to a MMSE allocation.

A multiplexer generates output frames that include a sync word, a frame header, an audio header and at least one subframe, and which are multiplexed into a data stream at a transmission rate. The frame header includes the window size and the size of the current output frame. The audio header indicates a packing arrangement and a coding format for the audio frame. Each audio subframe includes side information for decoding the audio subframe without reference to any other subframe, high frequency VQ codes, a plurality of baseband audio sub-subframes, in which audio data for each channel's lower frequency subbands is packed and multiplexed with the other channels, a high frequency audio block, in which audio data in the high frequency range for each channel is packed and multiplexed with the other channels so that the multi-channel audio signal is decodable at a plurality of decoding sampling rates, and an unpack sync for verifying the end of the subframe.

The window size is selected as a function of the ratio of the transmission rate to the encoder sampling rate so that the size of the output frame is constrained to lie in a desired range. When the amount of compression is relatively low the window size is reduced so that the frame size does not exceed an upper maximum. As a result, a decoder can use an input buffer with a fixed and relatively small amount of RAM. When the amount of compression is relatively high, the window size is increased. As a result, the GBM system can distribute bits over a larger time window thereby improving encoder performance.

These and other features and advantages of the invention will be apparent to those skilled in the art from

the following detailed description of preferred embodiments, taken together with the accompanying drawings and tables, in which:

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a 5-channel audio coder in accordance with the present invention;

FIG. 2 is a block diagram of a multi-channel encoder;

FIG. 3 is a block diagram of the baseband encoder and decoder;

FIGs. 4a and 4b are block diagrams of a high sampling rate encoder and decoder, respectively;

FIG. 5 is a block diagram of a single channel encoder;

FIG. 6 is a plot of the bytes per frame versus frame size for variable transmission rates;

FIG. 7 is a plot of the amplitude response for the NPR and PR reconstruction filters;

FIG. 8 is a plot of the subband aliasing for a reconstruction filter;

FIG. 9 is a plot of the distortion curves for the NPR and PR filters;

FIG. 10 is a schematic diagram of a single subband encoder;

FIGs. 11a and 11b transient detection and scale factor computation, respectively, for a subframe;

FIG. 12 illustrates the entropy coding process for the quantized TMODES;

FIG. 13 illustrates the scale factor quantization process;

FIG. 14 illustrates the convolution of a signal mask with the signal's frequency response to generate the SMRs;

FIG. 15 is a plot of the human auditory response;

FIG. 16 is a plot of the SMRs for the subbands;

FIG. 17 is a plot of the error signals for the psychoacoustic and mmse bit allocations;

FIGs. 18a and 18b are a plot of the subband energy levels and the inverted plot, respectively, illustrating the

mmse "waterfilling" bit allocation process;

FIG. 19 is a block diagram of a single frame in the data stream;

FIG. 20 is a schematic diagram of the decoder;

FIG. 21 is a block diagram of a hardware implementation for the encoder; and

FIG. 22 is a block diagram of a hardware implementation for the decoder.

BRIEF DESCRIPTION OF THE TABLES

Table 1 tabulates the maximum frame size versus sampling rate and transmission rate;

Table 2 tabulates the maximum allowed frame size (bytes) versus sampling rate and transmission rate; and

Table 3 illustrates the relationship between ABIT index value, the number of quantization levels and the resulting subband SNR.

DETAILED DESCRIPTION OF THE INVENTIOn

Multi-Channel Audio Coding System

As shown in **FIG. 1**, the present invention combines the features of both of the known encoding schemes plus additional features in a single multi-channel audio coder **10**. The encoding algorithm is designed to perform at studio quality levels i.e. "better than CD" quality and provide a wide range of applications for varying compression levels, sampling rates, word lengths, number of channels and perceptual quality.

The encoder **12** encodes multiple channels of PCM audio data **14**, typically sampled at 48kHz with word lengths between 16 and 24 bits, into a data stream **16** at a known transmission rate, suitably in the range of 32-4096kbps. Unlike known audio coders, the present architecture can be expanded to higher sampling rates (48-192kHz) without making the existing decoders, which were designed for the baseband sampling rate or any intermediate sampling rate, incompati-

ble. Furthermore, the PCM data **14** is windowed and encoded a frame at a time where each frame is preferably split into 1-4 subframes. The size of the audio window, i.e. the number of PCM samples, is based on the relative values of the sampling rate and transmission rate such that the size of an output frame, i.e. the number of bytes, read out by the decoder **18** per frame is constrained, suitably between 5.3 and 8 kbytes.

As a result, the amount of RAM required at the decoder to buffer the incoming data stream is kept relatively low, which reduces the cost of the decoder. At low rates larger window sizes can be used to frame the PCM data, which improves the coding performance. At higher bit rates, smaller window sizes must be used to satisfy the data constraint. This necessarily reduces coding performance, but at the higher rates it is insignificant. Also, the manner in which the PCM data is framed allows the decoder **18** to initiate playback before the entire output frame is read into the buffer. This reduces the delay or latency of the audio coder.

The encoder **12** uses a high resolution filterbank, which preferably switches between non-perfect (NPR) and perfect (PR) reconstruction filters based on the bit rate, to decompose each audio channel **14** into a number of subband signals. Predictive and vector quantization (VQ) coders are used to encode the lower and upper frequency subbands, respectively. The start VQ subband can be fixed or may be determined dynamically as a function of the current signal properties. Joint frequency coding may be employed at low bit rates to simultaneously encode multiple channels in the higher frequency subbands.

The predictive coder preferably switches between APCM and ADPCM modes based on the subband prediction gain. A transient analyzer segments each subband subframe into pre and post-echo signals (sub-subframes) and computes respective scale factors for the pre and post-echo sub-subfr ames thereby reducing pre-echo distortion. The encoder

adaptively allocates the available bit rate across all of the PCM channels and subbands for the current frame according to their respective needs (psychoacoustic or mse) to optimize the coding efficiency. By combining predictive coding and psychoacoustic modeling, the low bit rate coding efficiency is enhanced thereby lowering the bit rate at which subjective transparency is achieved. A programmable controller **19** such as a computer or a key pad interfaces with the encoder 12 to relay audio mode information including parameters such as the desired bit rate, the number of channels, PR or NPR reconstruction, sampling rate and transmission rate.

The encoded signals and sideband information are packed and multiplexed into the data stream **16** such that the decoding computational load is constrained to lie in the desired range. The data stream **16** is encoded on or broadcast over a transmission medium **20** such as a CD, a digital video disk (DVD), or a direct broadcast satellite. The decoder **18** decodes the individual subband signals and performs the inverse filtering operation to generate a multi-channel audio signal **22** that is subjectively equivalent to the original multi-channel audio signal **14**. An audio system **24** such as a home theater system or a multimedia computer play back the audio signal for the user.

Multi-Channel Encoder

As shown in **FIG. 2**, the encoder **12** includes a plurality of individual channel encoders **26**, suitably five (left front, center, right front, left rear and right rear), that produce respective sets of encoded subband signals **28**, suitably 32 subband signals per channel. The encoder **12** employs a global bit management (GBM) system **30** that dynamically allocates the bits from a common bit-pool among the channels, between the subbands within a channel, and within an individual frame in a given subband. The encoder **12** may also use joint frequency coding techniques to take advantage of inter-channel correlations in the higher frequency subbands. Furthermore, the encoder **12** can use VQ on the

higher frequency subbands that are not specifically perceptible to provide a basic high frequency fidelity or ambiance at a very low bit rate. In this way, the coder takes advantage of the disparate signal demands, e.g. the subbands' rms values and psychoacoustic masking levels, of the multiple channels and the non-uniform distribution of signal energy over frequency in each channel and over time in a given frame.

Bit Allocation Overview

The GBM system **30** first decides which channels' subbands will be joint frequency coded and averages that data, and then determines which subbands will be encoded using VQ and subtracts those bits from the available bit rate. The decision of which subbands to VQ can be made a priori in that all subbands above a threshold frequency are VQ or can be made based on the psychoacoustic masking effects of the individual subbands in each frame. Thereafter, the GBM system **30** allocates bits (ABIT) using psychoacoustic masking on the remaining subbands to optimize the subjective quality of the decoded audio signal. If additional bits are available, the encoder can switch to a pure mmse scheme, i.e. "waterfilling", and reallocate all of the bits based on the subbands relative rms values to minimize the rms value of the error signal. This is applicable at very high bit rates. The preferred approach is to retain the psychoacoustic bit allocation and allocate only the additional bits according to the mmse scheme. This maintains the shape of the noise signal created by the psychoacoustic masking, but uniformly shifts the noise floor downwards.

Alternately, the preferred approach can be modified such that the additional bits are allocated according to the difference between the rms and psychoacoustic levels. As a result, the psychoacoustic allocation morphs to a mmse allocation as the bit rate increases thereby providing a smooth transition between the two techniques. The above techniques are specifically applicable for fixed bit rate

systems. Alternately, the encoder **12** can set a distortion level, subjective or mse, and allow the overall bit rate to vary to maintain the distortion level. A multiplexer **32** multiplexes the subband signals and side information into the data stream **16** in accordance with a specified data format. Details of the data format are discussed in **FIG. 20** below.

Baseband Encoding

For sampling rates in the range 8 - 48kHz, the channel encoder **26**, as shown in **FIG. 3**, employs a uniform 512-tap 32-band analysis filter bank **34** operating at a sampling rate of 48kHz to split the audio spectrum, 0 - 24kHz, of each channel into 32 subbands having a bandwidth of 750 Hz per subband. The coding stage **36** codes each subband signal and multiplexes **38** them into the compressed data stream **16**. The decoder **18** receives the compressed data stream, separates out the coded data for each subband using an unpacker **40**, decodes each subband signal **42** and reconstructs the PCM digital audio signals (Fsamp=48kHz) using a 512-tap 32-band uniform interpolation filter bank **44** for each channel.

In the present architecture, all of the coding strategies, e.g. sampling rates of 48, 96 or 192 kHz, use the 32-band encoding/decoding process on the lowest (baseband) audio frequencies, for example between 0 - 24kHz. Thus, decoders that are designed and built today based upon a 48kHz sampling rate will be compatible with future encoders that are designed to take advantage of higher frequency components. The existing decoder would read the baseband signal (0-24kHz) and ignore the encoded data for the higher frequencies.

High Sampling Rate Encoding

For sampling rates in the range 48 - 96kHz, the channel encoder **26** preferably splits the audio spectrum in two and employs a uniform 32-band analysis filter bank for the bottom half and an 8-band analysis filter bank for the top half. As shown in **FIGs. 4a and 4b** the audio spectrum, 0 -

48kHz, is initially split using a 256-tap 2-band decimation pre-filter bank **46** giving an audio bandwidth of 24kHz per band. The bottom band (0 - 24kHz) is split and encoded in 32 uniform bands in the manner described above in **FIG. 3**. The top band (24 - 48kHz) however, is split and encoded in 8 uniform bands. If the delay of the 8-band decimatio n/interpolation filter bank **48** is not equal to that of the 32-band filter banks then a delay compensation stage **50** must be employed somewhere in the 24 - 48kHz signal path to ensure that both time waveforms line up prior to the 2-band recombination filter bank at the decoder. In the 96kHz sampling encoding system, the 24 - 48kHz audio band is delayed by 384 samples and then split into the 8 uniform bands using a 128-tap interpolation filter bank. Each of the 3kHz subbands is encoded **52** and packed **54** with the coded data from the 0 - 24kHz band to form the compressed data stream **16**.

On arrival at the decoder **18**, the compressed data stream **16** is unpacked **56** and the codes for both the 32-ban d decoder (0 - 24kHz region) and 8-band decoder (24 - 48kHz) are separated out and fed to their respective decoding stages **42** and **58**, respectively. The eight and 32 decoded subbands are reconstructed using 128-tap and 512-tap uniform interpolation filter banks **60** and **44**, respectively. The decoded subbands are subsequently recombined using a 256-tap 2-band uniform interpolation filter bank **62** to produce a single PCM digital audio signal with a sampling rate of 96kHz. In the case when it is desirable for the decoder to operate at half the sampling rate of the compressed data stream, this can be conveniently carried out by discarding the upper band encoded data (24 - 48kHz) and decoding only the 32-subbands in the 0 - 24kHz audio region.

Channel Encoder

In all the coding strategies described, the 32-band encoding/decoding process is carried out for the baseband portion of the audio bandwidth between 0 - 24kHz. As shown in **FIG. 5**, a frame grabber **64** windows the PCM audio channel

14 to segment it into successive data frames **66**. The PCM audio window defines the number of contiguous input samples for which the encoding process generates an output frame in the data stream. The window size is set based upon the amount of compression, i.e. the ratio of the transmission rate to the sampling rate, such that the amount of data encoded in each frame is constrained. Each successive data frame **66** is split into 32 uniform frequency bands **68** by a 32-band 512-tap FIR decimation filter bank **34**. The samples output from each subband are buffered and applied to the 32-band coding stage **36**.

An analysis stage **70** (described in detail in **FIGs. 10-19**) generates optimal predictor coefficients, differential quantizer bit allocations and optimal quantizer scale factors for the buffered subband samples. The analysis stage **70** can also decide which subbands will be VQ and which will be joint frequency coded if these decisions are not fixed. This data, or side information, is fed forward to the selected ADPCM stage **72**, VQ stage **73** or Joint Frequency Coding (JFC) stage **74**, and to the data multiplexer **32** (packer). The subband samples are then encoded by the ADPCM or VQ process and the quantization codes input to the multiplexer. The JFC stage **74** does not actually encode subband samples but generates codes that indicate which channels' subbands are joined and where they are placed in the data stream. The quantization codes and the side information from each subband are packed into the data stream **16** and transmitted to the decoder.

On arrival at the decoder **18**, the data stream is demultiplexed **40**, or unpacked, back into the individual subbands. The scale factors and bit allocations are first installed into the inverse quantizers **75** together with the predictor coefficients for each subband. The differential codes are then reconstructed using either the ADPCM process **76** or the inverse VQ process **77** directly or the inverse JFC process **78** for designated subbands. The subbands are finally amalgamated back to a single PCM audio signal **22**

using the 32-band interpolation filter bank **44**.

PCM Signal Framing

As shown in **FIG. 6**, the frame grabber **64** shown in **FIG. 5** varies the size of the window **79** as the transmission rate changes for a given sampling rate so that the number of bytes per output frame **80** is constrained to lie between, for example, 5.3k bytes and 8k bytes. **Tables 1** and **2** are design tables that allow a designer to select the optimum window size and decoder buffer size (frame size), respectively, for a given sampling rate and transmission rate. At low transmission rates the frame size can be relatively large. This allows the encoder to exploit the non-flat variance distribution of the audio signal over time and improve the audio coder's performance. At high rates, the frame size is reduced so that the total number of bytes does not overflow the decoder buffer. As a result, a designer can provide the decoder with 8k bytes of RAM to satisfy all transmission rates. This reduces the cost of the decoder.

In general, the size of the audio window is given by:

$$\text{Audio Window} = (\text{Frame Size}) * F_{samp} * \left(\frac{8}{T_{rate}}\right)$$

where Frame Size is the size of the decoder buffer, F_{samp} is the sampling rate, and T_{rate} is the transmission rate. The size of the audio window is independent of the number of audio channels. However, as the number of channels is increased the amount of compression must also increase to maintain the desired transmission rate.

Table 1

F_{samp} (kHz)

T_{rate}	8-12	16-24	32-48	64-96	128-192
≤512kbps	1024	2048	4096	*	*
≤1024kbps	*	1024	2048	*	*
≤2048kbps	*	*	1024	2048	*
≤4096kbps	*	*	*	1024	2048

Table 2

	F_{samp} (kHz)				
T_{rate}	8-12	16-24	32-48	64-96	128-192
<512kbps	8-5.3k	8-5.3k	8-5.3k	*	*
<1024kbps	*	8-5.3k	8-5.3k	*	*
<2048kbps	*	*	8-5.3k	8-5.3k	*
<4096kbps	*	*	*	8-5.3k	8-5.3k

Subband Filtering

The 32-band 512-tap uniform decimation filterbank **34** selects from two polyphase filterbanks to split the data frames **66** into the 32 uniform subbands **68** shown in **FIG. 5**. The two filterbanks have different reconstruction properties that trade off subband coding gain against reconstruction precision. One class of filters is called perfect reconstruction (PR) filters. When the PR decimation (encoding) filter and its interpolation (decoding) filter are placed back-to-back the reconstructed signal is "perfect," where perfect is defined as being within 0.5 lsb at 24 bits of resolution. The other class of filters is called non-perfect reconstruction (NPR) filters because the reconstructed signal has a non-zero noise floor that is associated with the non-perfect aliasing cancellation properties of the filtering process.

The transfer functions **82** and **84** of the NPR and PR filters, respectively, for a single subband are shown in **FIG. 7**. Because the NPR filters are not constrained to provide perfect reconstruction, they exhibit much larger near stop band rejection (NSBR) ratios, i.e. the ratio of the passband to the first side lobe, than the PR filters (110 dB v. 85 dB). As shown in **FIG. 8**, the sidelobes of the filter cause a signal **86** that naturally lies in the third subband to alias into the neighboring subbands. The subband gain measures the rejection of the signal in the neighboring subbands, and hence indicates the filter's ability to decorrelate the audio signal. Because the NPR

filters' have a much larger NSBR ratio than the PR filters they will also have a much larger subband gain. As a result, the NPR filters provide better encoding efficiency.

As shown in **FIG. 9**, the total distortion in the compressed data stream is reduced as the overall bit rate increases for both the PR and NPR filters. However, at low rates the difference in subband gain performance between the two filter types is greater than the noise floor associated with NPR filter. Thus, the NPR filter's associated distortion curve **90** lies below the PR filter's associated distortion curve **92**. Hence, at low rates the audio coder selects the NPR filter bank. At some point **94**, the encoder's quantization error falls below the NPR filter's noise floor such that adding additional bits to the ADPCM coder provides no additional benefits. At this point, the audio coder switches to the PR filter bank.

ADPCM Encoding

The ADPCM encoder **72** generates a predicted sample p(n) from a linear combination of H previous reconstructed samples. This prediction sample is then subtracted from the input x(n) to give a difference sample d(n). The difference samples are scaled by dividing them by the RMS (or PEAK) scale factor to match the RMS amplitudes of the difference samples to that of the quantizer characteristic Q. The scaled difference sample ud(n) is applied to a quantizer characteristic with L levels of step-size SZ, as determined by the number of bits ABIT allocated for the current sample. The quantizer produces a level code QL(n) for each scaled difference sample ud(n). These level codes are ultimately transmitted to the decoder ADPCM stage. To update the predictor history, the quantizer level codes QL(n) are locally decoded using an inverse quantizer 1/Q with identical characteristics to that of Q to produce a quantized scaled difference sample $\hat{ud}(n)$. The sample $\hat{ud}(n)$ is rescaled by multiplying it with the RMS (or PEAK) scale factor, to produce $\hat{d}(n)$. A quantized version $\hat{x}(n)$ of the original input sample

x(n) is reconstructed by adding the initial prediction sample p(n) to the quantized difference sample $\hat{d}(n)$. This sample is then used to update the predictor history.

Vector Quantization

The predictor coefficients and high frequency subband samples are encoded using vector quantization (VQ). The predictor VQ has a vector dimension of 4 samples and a bit rate of 3 bits per sample. The final codebook therefore consists of 4096 codevectors of dimension 4. The search of matching vectors is structured as a two level tree with each node in the tree having 64 branches. The top level stores 64 node codevectors which are only needed at the encoder to help the searching process. The bottom level contacts 4096 final codevectors, which are required at both the encoder and the decoder. For each search, 128 MSE computations of dimension 4 are required. The codebook and the node vectors at the top level are trained using the LBG method, with over 5 million prediction coefficient training vectors. The training vectors are accumulated for all subband which exhibit a positive prediction gain while coding a wide range of audio material. For test vectors in a training set, average SNRs of approximately 30dB are obtained.

The high frequency VQ has a vector dimension of 32 samples (the length of a subframe) and a bit rate of 0.3125 bits per sample. The final codebook therefore consists of 1024 codevectors of dimension 32. The search of matching vectors is structured as a two level tree with each node in the tree having 32 branches. The top level stores 32 node codevectors, which are only needed at the encoder. The bottom level contains 1024 final codevectors which are required at both the encoder and the decoder. For each search, 64 MSE computations of dimension 32 are required. The codebook and the node vectors at the top level are trained using the LBG method with over 7 million high frequency subband sample training vectors. The samples which make up the vectors are accumulated from the outputs of subbands 16 through 32 for a sampling rate of 48 kHz for

a wide range of audio material. At a sampling rate of 48kHz, the training samples represent audio frequencies in the range 12 to 24 kHz. For test vectors in the train set, an average SNR of about 3dB is expected. Although 3dB is a small SNR, it is sufficient to provide high frequency fidelity or ambiance at these high frequencies. It is perceptually much better than the known techniques which simple drop the high frequency subbands.

Joint Frequency Coding

In very low bit rate applications overall reconstruction fidelity can be improved by coding only a summation of high frequency subband signals from two or more audio channels instead of coding them independently. Joint frequency coding is possible because the high frequency subbands oftentimes have similar energy distributions and because the human auditory system is sensitive primarily to the "intensity" of the high frequency components, rather than their fine structure. Thus, the reconstructed average signal provides good overall fidelity since at any bit rate more bits are available to code the perceptually important low frequencies.

Joint frequency coding indexes (JOINX) are transmitted directly to the decoder to indicate which channels and subbands have been joined and where the encoded signal is positioned in the data stream. The decoder reconstructs the signal in the designated channel and then copies it to each of the other channels. Each channel is then scaled in accordance with its particular RMS scale factor. Because joint frequency coding averages the time signals based on the similarity of their energy distributions, the reconstruction fidelity is reduced. Therefore, its application is typically limited to low bit rate applications and mainly to the 10-20kHz signals. In the medium to high bit rate applications joint frequency coding is typically disabled.

Subband Encoder

The encoding process for a single sideband that is en-

coded using the ADPCM/APCM processes, and specifically the interaction of the analysis stage **70** and ADPCM coder **72** shown in **FIG. 5** and the global bit management system **30** shown in **FIG. 2**, is illustrated in detail in **FIG. 10**. **FIGs. 11-19** detail the component processes shown in **FIG. 13**. The filterbank **34** splits the PCM audio signal **14** into 32 subband signals x(n) that are written into respective subband sample buffers **96**. Assuming a audio window size of 4096 samples, each subband sample buffer **96** stores a complete frame of 128 samples, which are divided into 4 32-sample subframes. A window size of 1024 samples would produce a single 32-sample subframe. The samples x(n) are directed to the analysis stage **70** to determine the prediction coefficients, the predictor mode (PMODE), the transient mode (TMODE) and the scale factors (SF) for each subframe. The samples x(n) are also provided to the GBM system **30**, which determines the bit allocation (ABIT) for each subframe per subband per audio channel. Thereafter, the samples x(n) are passed to the ADPCM coder **72** a subframe at a time.

Estimation of Optimal Prediction Coefficients

The H, suitably 4th order, prediction coefficients are generated separately for each subframe using the standard autocorrelation method **98** optimized over a block of subband samples x(n), i.e. the Weiner-Hopf or Yule-Walker equations.

Quantization of Optimal Prediction Coefficients

Each set of four predictor coefficients is preferably quantized using a 4-element tree-search 12-bit vector codebook (3 bits per coefficient) described above. The 12-bit vector codebook contains 4096 coefficient vectors that are optimized for a desired probability distribution using a standard clustering algorithm. A vector quantization (VQ) search **100** selects the coefficient vector which exhibits the lowest weighted mean squared error between itself and the optimal coefficients. The optimal coefficients for each subframe are then replaced with these "quantized" vectors. An inverse VQ LUT **101** is used to provide the quantized predictor coefficients to the ADPCM coder **72**.

Estimation of Prediction Difference Signal d(n)

A significant quandary with ADPCM is that the difference sample sequence d(n) cannot be easily predicted ahead of the actual recursive process **72**. A fundamental requirement of forward adaptive subband ADPCM is that the difference signal energy be known ahead of the ADPCM coding in order to calculate an appropriate bit allocation for the quantizer which will produce a known quantization error, or noise level in the reconstructed samples. Knowledge of the difference signal energy is also required to allow an optimal difference scale factor to be determined prior to encoding.

Unfortunately, the difference signal energy not only depends on the characteristics of the input signal but also on the performance of the predictor. Apart from the known limitations such as the predictor order and the optimality of the predictor coefficients, the predictor performance is also affected by the level of quantization error, or noise, induced in the reconstructed samples. Since the quantization noise is dictated by the final bit allocation ABIT and the difference scale factor RMS (or PEAK) values themselves, the difference signal energy estimate must be arrived at iteratively **102**.

Step 1. Assume Zero Quantization Error

The first difference signal estimation is made by passing the buffered subband samples x(n) through an ADPCM process which does not quantize the difference signal. This is accomplished by disabling the quantization and RMS scaling in the ADPCM encoding loop. By estimating the difference signal d(n) in this way, the effects of the scale factor and the bit allocation values are removed from the calculation. However, the effect of the quantization error on the predictor coefficients is taken into account by the process by using the vector quantized prediction coefficients. An inverse VQ LUT **104** is used to provide the quantized prediction coefficients. To further enhance the accuracy of the estimate predictor, the history samples from

the actual ADPCM predictor that were accumulated at the end of the previous block are copied into the predictor prior to the calculation. This ensures that the predictor starts off from where the real ADPCM predictor left off at the end of the previous input buffer.

The main discrepancy between this estimate ed(n) and the actual process d(n) is that the effect of quantization noise on the reconstructed samples x(n) and on the reduced prediction accuracy is ignored. For quantizers with a large number of levels the noise level will generally be small (assuming proper scaling) and therefore the actual difference signal energy will closely match that calculated in the estimate. However, when the number of quantizer levels is small, as is the case for typical low bit rate audio coders, the actual predicted signal, and hence the difference signal energy, may differ significantly from the estimated one. This produces coding noise floors that are different from those predicted earlier in the adaptive bit allocation process.

Despite this, the variation in prediction performance may not be significant for the application or bit rate. Thus, the estimate can be used directly to calculate the bit allocations and the scale factors without iterating. An additional refinement would be to compensate for the performance loss by deliberately over-estimating the difference signal energy if it is likely that a quantizer with a small number of levels is to be allocated to that subband. The over-estimation may also be graded according to the changing number of quantizer levels for improved accuracy.

Step 2. Recalculate using Estimated Bit Allocations and Scale Factors

Once the bit allocations (ABIT) and scale factors (SF) have been generated using the first estimation difference signal, their optimality may be tested by running a further ADPCM estimation process using the estimated ABIT and RMS (or PEAK) values in the ADPCM loop **72**. As with the first

estimate, the estimate predictor history is copied from the actual ADPCM predictor prior to starting the calculation to ensure that both predictors start from the same point. Once the buffered input samples have all passed through this second estimation loop, the resulting noise floor in each subband is compared to the assumed noise floor in the adaptive bit allocation process. Any significant discrepancies can be compensated for by modifying the bit allocation an d/or scale factors.

Step 2 can be repeated to suitably refine the distributed noise floor across the subbands, each time using the most current difference signal estimate to calculate the next set of bit allocations and scale factors. In general, if the scale factors would change by more than approximately 2-3 dB, then they are recalculated. Otherwise the bit allocation would risk violating the signal-to-mask ratios generating by the psychoacoustic masking process, or alternately the mmse process. Typically, a single iteration is sufficient.

Calculation of Subband Prediction Modes (PMODE)

To improve the coding efficiency, a controller **106** can arbitrarily switch the prediction process off when the prediction gain in the current subframe falls below a threshold by setting a PMODE flag. The PMODE flag is set to one when the prediction gain (ratio of the input signal energy and the estimated difference signal energy), measured during the estimation stage for a block of input samples, exceeds some positive threshold. Conversely, if the prediction gain is measured to be less than the positive threshold the ADPCM predictor coefficients are set to zero at both encoder and decoder, for that subband, and the respective PMODE is set to zero. The prediction gain threshold is set such that it equals the distortion rate of the transmitted predictor coefficient vector overhead. This is done in an attempt to ensure that when PMODE=1, the coding gain for the ADPCM process is always greater than or equal to that of a forward adaptive PCM (APCM) coding process. Otherwise by setting

PMODE to zero and resetting the predictor coefficients, the ADPCM process simply reverts to APCM.

The PMODEs can be set high in any or all subbands if the ADPCM coding gain variations are not important to the application. Conversely, the PMODES can be set low if, for example, certain subbands are not going to be coded at all, the bit rate of the application is high enough that prediction gains are not required to maintain the subjective quality of the audio, the transient content of the signal is high, or the splicing characteristic of ADPCM encoded audio is simply not desirable, as might be the case for audio editing applications.

Separate prediction modes (PMODEs) are transmitted for each subband at a rate equal to the update rate of the linear predictors in the encoder and decoder ADPCM processes. The purpose of the PMODE parameter is to indicate to the decoder if the particular subband will have any prediction coefficient vector address associated with its coded audio data block. When PMODE=1 in any subband then a predictor coefficient vector address will always be included in the data stream. When PMODE=0 in any subband then a predictor coefficient vector address will never be included in the data stream and the predictor coefficients are set to zero at both encoder and decoder ADPCM stages.

The calculation of the PMODEs begins by analyzing the buffered subband input signal energies with respect to the corresponding buffered estimated difference signal energies obtained in the first stage estimation, i.e. assuming no quantization error. Both the input samples x(n) and the estimated difference samples ed(n) are buffered for each subband separately. The buffer size equals the number of samples contained in each predictor update period, e.g. the size of a subframe. The prediction gain is then calculated as:

$$\mathbf{P_{gain}\ (dB)\ =\ 20.0*Log_{10}(RMS_{x(n)}/RMS_{ed(n)})}$$

where $RMS_{x(n)}$ = root mean square value of the buffered input

samples x(n) and $RMS_{ed(n)}$ = root mean square value of the buffered estimated difference samples ed(n).

For positive prediction gains, the difference signal is, on average, smaller than the input signal, and hence a reduced reconstruction noise floor may be attainable using the ADPCM process over APCM for the same bit rate. For negative gains, the ADPCM coder is making the difference signal, on average, greater than the input signal, which results in higher noise floors than APCM for the same bit rate. Normally, the prediction gain threshold, which switches PMODE on, will be positive and will have a value which takes into account the extra channel capacity consumed by transmitting the predictor coefficients vector address.

Calculation of Subband Transient Modes (TMODE)

The controller **106** calculates the transient modes (TMODE) for each subframe in each subband. The TMODEs indicate the number of scale factors and the samples in the estimated difference signal ed(n) buffer when PMODE=1 or in the input subband signal x(n) buffer when PMODE=0, for which they are valid. The TMODEs are updated at the same rate as the prediction coefficient vector addresses and are transmitted to the decoder. The purpose of the transient modes is to reduce audible coding "pre-echo" artifacts in the presence of signal transients.

A transient is defined as a rapid transition between a low amplitude signal and a high amplitude signal. Because the scale factors are averaged over a block of subband difference samples, if a rapid change in signal amplitude takes place in a block, i.e. a transient occurs, the calculated scale factor tends to be much larger than would be optimal for the low amplitude samples preceding the transient. Hence, the quantization error in samples preceding transients can be very high. This noise is perceived as pre-echo distortion.

In practice, the transient mode is used to modify the subband scale factor averaging block length to limit the influence of a transient on the scaling of the differential

samples immediately preceding it. The motivation for doing this is the pre-masking phenomena inherent in the human auditory system, which suggests that in the presence of transients noise can be masked prior to a transient provided that its duration is kept short.

Depending on the value of PMODE either the contents, i.e. the subframe, of the subband sample buffer x(n) or that of the estimated difference buffer ed(n) are copied into a transient analysis buffer. Here the buffer contents are divided uniformly into either 2, 3 or 4 sub-subframes depending on the sample size of the analysis buffer. For example, if the analysis buffer contains 32 subband samples (21.3ms @1500Hz), the buffer is partitioned into 4 sub-subfr ames of 8 samples each, giving a time resolution of 5.3ms for a subband sampling rate of 1500Hz. Alternately, if the analysis window was configured at 16 subband samples, then the buffer need only be divided into two sub-subframes to give the same time resolution.

The signal in each sub-subframe is analyzed and the transient status of each, other than the first, is determined. If any sub-subframes are declared transient, two separate scale factors are generated for the analysis buffer, i.e. the current subframe. The first scale factor is calculated from samples in the sub-subframes preceding the transient sub-subframe. The second scale factor is calculated from samples in the transient sub-subframe together with all proceeding sub-subframes.

The transient status of the first sub-subframe is not calculated since the quantization noise is automatically limited by the start of the analysis window itself. If more than one sub-subframe is declared transient, then only the one which occurs first is considered. If no transient sub-buffers are detected at all, then only a single scale factor is calculated using all of the samples in the analysis buffer. In this way scale factor values which include transient samples are not used to scale earlier samples more than a sub-subframe period back in time. Hence, the pre-tra

nsient quantization noise is limited to a sub-subframe period.

Transient Declaration

A sub-subframe is declared transient if the ratio of its energy over the preceding sub-buffer exceeds a transient threshold (TT), and the energy in the preceding sub-subframe is below a pre-transient threshold (PTT). The values of TT and PTT will depend on the bit rate and the degree of pre-ec ho suppression required. They are normally varied until perceived pre-echo distortion matches the level of other coding artifacts if they exist. Increasing TT and/or decreasing PTT values will reduce the likelihood of sub-subf rames being declared transient, and hence will reduce the bit rate associated with the transmission of the scale factors. Conversely, reducing TT and/or increasing PTT values will increase the likelihood of sub-subframes being declared transient, and hence will increase the bit rate associated with the transmission of the scale factors.

Since TT and PTT are individually set for each subband, the sensitivity of the transient detection at the encoder can be arbitrarily set for any subband. For example, if it is found that pre-echo in high frequency subbands is less perceptible than in lower frequency subbands, then the thresholds can be set to reduce the likelihood of transients being declared in the higher subbands. Moreover, since TMODEs are embedded in the compressed data stream, the decoder never needs to know the transient detection algorithm in use at the encoder in order to properly decode the TMODE information.

Four Sub-buffer Configuration

As shown in **FIG. 11a**, if the first sub-subframe **108** in the subband analysis buffer **109** is transient, or if no transient sub-subframes are detected, then TMODE=0. If the second sub-subframe is transient but not the first, then TMODE=1. If the third sub-subframe is transient but not the first or second, then TMODE=2. If only the fourth sub-subfra me is transient then TMODE=3.

Calculation of Scale Factors

As shown in **FIG. 11b**, when TMODE=0 the scale factors **110** are calculated over all sub-subframes. When TMODE=1, the first scale factor is calculated over the first sub-subframe and the second scale factor over all proceeding sub-subframes. When TMODE=2 the first scale factor is calculated over the first and second sub-subframes and the second scale factor over all proceeding sub-subframes. When TMODE=3 the first scale factor is calculated over the first, second and third sub-subframes and the second scale factor is calculated over the fourth sub-subframe.

ADPCM Encoding and Decoding using TMODE

When TMODE=0 the single scale factor is used to scale the subband difference samples for the duration of the entire analysis buffer, i.e. a subframe, and is transmitted to the decoder to facilitate inverse scaling. When TMODE>0 then two scale factors are used to scale the subband difference samples and both transmitted to the decoder. For any TMODE, each scale factor is used to scale the differential samples used to generate the it in the first place.

Calculation of Subband Scale Factors (RMS or PEAK)

Depending on the value of PMODE for that subband, either the estimated difference samples ed(n) or input subband samples x(n) are used to calculate the appropriate scale factor(s). The TMODEs are used in this calculation to determine both the number of scale factors and to identify the corresponding sub-subframes in the buffer.

RMS scale factor calculation

For the jth subband, the rms scale factors are calculated as follows:

When TMODE=0 then the single rms value is;

$$RMS_j = \left(\sum_{n=1}^{L} ed(n)^2 / L \right)^{0.5}$$

where L is the number of samples in the subframe.

When TMODE >0 then the two rms values are;

$$RMS1_j = \left(\sum_{n=1}^{k} ed(n)^2 / L \right)^{0.5}$$

$$RMS2_j = \left(\sum_{n=1}^{k+1} ed(n)^2 / L \right)^{0.5}$$

where k =(TMODE*L/NSB) and NSB is the number of uniform subframes.

If PMODE=0 then the $ed_j(n)$ samples are replaced with the input samples $x_j(n)$.

PEAK scale factor calculation

For the jth subband, the peak scale factors are calculated as follows;

When TMODE=0 then the single peak value is;

$$\mathbf{PEAK_j = MAX(ABS(ed_j(n))) \ for \ n=1, \ L}$$

When TMODE>0 then the two peak values are;

$$\mathbf{PEAK1_j = MAX(ABS(ed_j(n))) \ for \ n=1, \ (TMODE*L/NSB)}$$

$$\mathbf{PEAK2_j = MAX(ABS(ed_j(n))) \ for \ n=(1+TMODE*L/NSB), \ L}$$

If PMODE=0 then the $ed_j(n)$ samples are replaced with the input samples $x_j(n)$.

Quantization of PMODE, TMODE and Scale Factors

Quantization of PMODEs

The prediction mode flags have only two values, on or off, and are transmitted to the decoder directly as 1-bit codes.

Quantization of TMODEs

The transient mode flags have a maximum of 4 values; 0, 1, 2 and 3, and are either transmitted to the decoder directly using 2-bit unsigned integer code words or optionally via a 4-level entropy table in an attempt to reduce the average word length of the TMODEs to below 2 bits. Typically the optional entropy coding is used for low-bit rate applications in order to conserve bits.

The entropy coding process **112** illustrated in detail in **FIG. 12** is as follows; the transient mode codes TMODE(j) for the j subbands are mapped to a number (p) of 4-level

mid-riser variable length code book, where each code book is optimized for a different input statistical characteristic. The TMODE values are mapped to the 4-level tables **114** and the total bit usage associated with each table (NB_p) is calculated **116**. The table that provides the lowest bit usage over the mapping process is selected **118** using the THUFF index. The mapped codes, VTMODE(j), are extracted from this table, packed and transmitted to the decoder along with the THUFF index word. The decoder, which holds the same set of 4-level inverse tables, uses the THUFF index to direct the incoming variable length codes, VTMODE(j), to the proper table for decoding back to the TMODE indexes.

Quantization of Subband Scale Factors

To transmit the scale factors to the decoder they must be quantized to a known code format. In this system they are quantized using either a uniform 64-level logarithmic characteristic, a uniform 128-level logarithmic characteristic, or a variable rate encoded uniform 64-level logarithmic characteristic **120**. The 64-level quantizer exhibits a 2.25dB step-size in both cases, and the 128-level a 1.25dB step-size. The 64-level quantization is used for low to medium bit-rates, the additional variable rate coding is used for low bit-rate applications, and the 128-level is generally used for high bit-rates.

The quantization process **120** is illustrated in **FIG. 13**. The scale factors, RMS or PEAK, are read out of a buffer **121**, converted to the log domain **122**, and then applied either to a 64-level or 128-level uniform quantizers **124, 126** as determined by the encoder mode control **128**. The log quantized scale factors are then written into a buffer **130**. The range of the 128 and 64-level quantizers are sufficient to cover scale factors with a dynamic range of approximately 160dB and 144dB, respectively. The 128-lev el upper limit is set to cover the dynamic range of 24-bit input PCM digital audio signals. The 64-level upper limit is set to cover the dynamic range of 20-bit input PCM digital audio signals.

The log scale factors are mapped to the quantizer and the scale factor is replaced with the nearest quantizer level code RMS_{QL} (or $PEAK_{QL}$). In the case of the 64-level quantizer these codes are 6-bits long and range between 0-63. In the case of the 128-level quantizer, the codes are 7-bits long and range between 0-127.

Inverse quantization **131** is achieved simply by mapping the level codes back to the respective inverse quantization characteristic to give RMS_q (or $PEAK_q$) values. Quantized scale factors are used both at the encoder and decoder for the ADPCM (or APCM if PMODE=0) differential sample scaling, thus ensuring that both scaling and inverse scaling processes are identical.

If the bit-rate of the 64-level quantizer codes needs to be reduced, additional entropy, or variable length coding is performed. The 64-level codes are first order differentially encoded **132** across the j subbands, starting at the second subband (j=2) to the highest active subband. The process can also be used to code PEAK scale factors. The signed differential codes $DRMS_{QL}(j)$, (or $DPEAK_{QL}(j)$) have a maximum range of +/-63 and are stored in a buffer **134**. To reduce their bit rate over the original 6-bit codes, the differential codes are mapped to a number (p) of 127-level mid-riser variable length code books. Each code book is optimized for a different input statistical characteristic.

The process for entropy coding the signed differential codes is the same as entropy coding process for transient modes illustrated in FIG. 12 except that p 127-level variable length code tables are used. The table which provides the lowest bit usage over the mapping process is selected using the SHUFF index. The mapped codes $VDRMS_{QL}(j)$ are extracted from this table, packed and transmitted to the decoder along with the SHUFF index word. The decoder, which holds the same set of (p) 127-level inverse tables, uses the SHUFF index to direct the incoming variable length codes to the proper table for decoding back to differential quantizer

code levels. The differential code levels are returned to absolute values using the following routines;

$$RMS_{QL}(1) = DRMS_{QL}(1)$$

$$RMS_{QL}(j) = DRMS_{QL}(j) + RMS_{QL}(j-1) \text{ for } j=2, \ldots K$$

and PEAK differential code levels are returned to absolute values using the following routines;

$$PEAK_{QL}(1) = DPEAK_{QL}(1)$$

$$PEAK_{QL}(j) = DPEAK_{QL}(j) + PEAK_{QL}(j-1) \text{ for } j=2, \ldots K$$

where in both cases K = number of active subbands.

Global Bit Allocation

The Global Bit Management system **30** shown in **FIG. 10** manages the bit allocation (ABIT), determines the number of active subbands (SUBS) and the joint frequency strategy (JOINX) and VQ strategy for the multi-channel audio encoder to provide subjectively transparent encoding at a reduced bit rate. This increases the number of audio channels an d/or the playback time that can be encoded and stored on a fixed medium while maintaining or improving audio fidelity. In general, the GBM system **30** first allocates bits to each subband according to a psychoacoustic analysis modified by the prediction gain of the encoder. The remaining bits are then allocated in accordance with a mmse scheme to lower the overall noise floor. To optimize encoding efficiency, the GBM system simultaneously allocates bits over all of the audio channels, all of the subbands, and across the entire frame. Furthermore, a joint frequency coding strategy can be employed. In this manner, the system takes advantage of the non-uniform distribution of signal energy between the audio channels, across frequency, and over time.

Psychoacoustic Analysis

Psychoacoustic measurements are used to determine perceptually irrelevant information in the audio signal. Perceptually irrelevant information is defined as those parts of the audio signal which cannot be heard by human listeners, and can be measured in the time domain, the frequency domain, or in some other basis. J.D. Johnston:

"Transform Coding of Audio Signals Using Perceptual Noise Criteria" IEEE Journal on Selected Areas in Communications, vol JSAC-6, no. 2, pp. 314-323, Feb. 1988 described the general principles of psychoacoustic coding.

Two main factors influence the psychoacoustic measurement. One is the frequency dependent absolute threshold of hearing applicable to humans. The other is the masking effect that one sound has on the ability of humans to hear a second sound played simultaneously or even after the first sound. In other words the first sound prevents us from hearing the second sound, and is said to mask it out.

In a subband coder the final outcome of a psychoacoustic calculation is a set of numbers which specify the inaudible level of noise for each subband at that instant. This computation is well known and is incorporated in the MPEG 1 compression standard ISO/IEC DIS 11172 "Information technology - Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbits/s," 1992. These numbers vary dynamically with the audio signal. The coder attempts to adjust the quantization noise floor in the subbands by way of the bit allocation process so that the quantization noise in these subbands is less than the audible level.

An accurate psychoacoustic calculation normally requires a high frequency resolution in the time-to-frequenc y transform. This implies a large analysis window for the time-to-frequency transform. The standard analysis window size is 1024 samples which corresponds to a subframe of compressed audio data. The frequency resolution of a length 1024 fft approximately matches the temporal resolution of the human ear.

The output of the psychoacoustic model is a signal-to-mask (SMR) ratio for each of the 32 subbands. The SMR is indicative of the amount of quantization noise that a particular subband can endure, and hence is also indicative of the number of bits required to quantize the samples in the subband. Specifically, a large SMR (>>1) indicates that

a large number of bits are required and a small SMR (>0) indicates that fewer bits are required. If the SMR < 0 then the audio signal lies below the noise mask threshold, and no bits are required for quantization.

As shown in **FIG. 14**, the SMRs for each successive frame are generated, in general, by 1) computing an fft, preferably of length 1024, on the PCM audio samples to produce a sequence of frequency coefficients **142**, 2) convolving the frequency coefficients with frequency dependent tone and noise psychoacoustic masks **144** for each subband, 3) averaging the resulting coefficients over each subband to produce the SMR levels, and 4) optionally normalizing the SMRs in accordance with the human auditory response **146** shown in **FIG. 15**.

The sensitivity of the human ear is a maximum at frequencies near 4kHz and falls off as the frequency is increased or decreased. Thus, in order to be perceived at the same level, a 20kHz signal must be much stronger than a 4kHz signal. Therefore, in general, the SMRs at frequencies near 4kHz are relatively more important than the outlying frequencies. However, the precise shape of the curve depends on the average power of the signal delivered to the listener. As the volume increases, the auditory response **146** is compressed. Thus, a system optimized for a particular volume will be suboptimal at other volumes. As a result, either a nominal power level is selected for normalizing the SMR levels or normalization is disabled. The resulting SMRs **148** for the 32 subbands are shown in **FIG. 16**.

Bit Allocation Routine

The GBM system **30** first selects the appropriate encoding strategy, which subbands will be encoded with the VQ and ADPCM algorithms and whether JFC will be enabled. Thereafter, the GBM system selects either a psychoacoustic or a MMSE bit allocation approach. For example, at high bit rates the system may disable the psychoacoustic modeling and use a true mmse allocation scheme. This reduces the compu-

tational complexity without any perceptual change in the reconstructed audio signal. Conversely, at low rates the system can activate the joint frequency coding scheme discussed above to improve the reconstruction fidelity at lower frequencies. The GBM system can switch between the normal psychoacoustic allocation and the mmse allocation based on the transient content of the signal on a frame-by-frame basis. When the transient content is high, the assumption of stationarity that is used to compute the SMRs is no longer true, and thus the mmse scheme provides better performance.

For a psychoacoustic allocation, the GBM system first allocates the available bits to satisfy the psychoacoustic effects and then allocates the remaining bits to lower the overall noise floor. The first step is to determine the SMRs for each subband for the current frame as described above. The next step is to adjust the SMRs for the prediction gain (Pgain) in the respective subbands to generate mask-to-noise rations (MNRs). The principle being that the ADPCM encoder will provide a portion of the required SMR. As a result, inaudible psychoacoustic noise levels can be achieved with fewer bits.

The MNR for the j^{th} subband, assuming PMODE=1, is given by:

$$\mathbf{MNR(j) = SMR(j) - Pgain(j)*PEF(ABIT)}$$

where PEF(ABIT) is the prediction efficiency factor of the quantizer. To calculate MNR(j), the designer must have an estimate of the bit allocation (ABIT), which can be generated by either allocating bits solely based on the SMR(j) or by assuming that PEF(ABIT)=1. At medium to high bit rates, the effective prediction gain is approximately equal to the calculated prediction gain. However, at low bit rates the effective prediction gain is reduced. The effective prediction gain that is achieved using, for example, a 5-level quantizer is approximately 0.7 of the estimated prediction gain, while a 65-level quantizer allows the effective prediction gain to be approximately equal to

the estimated prediction gain, PEF = 1.0. In the limit, when the bit rate is zero, predictive encoding is essentially disabled and the effective prediction gain is zero.

In the next step, the GBM system **30** generates a bit allocation scheme that satisfies the MNR for each subband. This is done using the approximation that 1 bit equals 6dB of signal distortion. To ensure that the encoding distortion is less than the psychoacoustically audible threshold, the assigned bit rate is the greatest integer of the MNR divided by 6dB, which is given by:

$$ABIT(j) = \left\lceil \frac{MNR(j)}{6dB} \right\rceil$$

By allocating bits in this manner, the noise level **156** in the reconstructed signal will tend to follow the signal itself **157** shown in **FIG. 17**. Thus, at frequencies where the signal is very strong the noise level will be relatively high, but will remain inaudible. At frequencies where the signal is relatively weak, the noise floor will be very small and inaudible. The average error associated with this type of psychoacoustic modeling will always be greater than a mmse noise level **158**, but the audible performance may be better, particularly at low bit rates.

In the event that the sum of the allocated bits for each subband over all audio channels is greater or less than the target bit-rate, the GBM routine will iteratively reduce or increase the bit allocation for individual subbands. Alternately, the target bit rate can be calculated for each audio channel. This is suboptimum but simpler especially in a hardware implementation. For example, the available bits can be distributed uniformly among the audio channels or can be distributed in proportion to the average SMR or RMS of each channel.

In the event that the target bit rate is exceeded by the sum of the local bit allocations, including the VQ code bits and side information, the global bit management routine

will progressively reduce the local subband bit allocations. A number of specific techniques are available for reducing the average bit rate. First, the bit rates that were rounded up by the greatest integer function can be rounded down. Next, one bit can be taken away from the subbands having the smallest MNRs. Furthermore, the higher frequency subbands can be turned off or joint frequency coding can be enabled. All bit rate reduction strategies follow the general principle of gradually reducing the coding resolution in a graceful manner, with the perceptually least offensive strategy introduced first and the most offensive strategy used last.

In the event that the target bit rate is greater than the sum of the local bit allocations, including the VQ code bits and side information, the global bit management routine will progressively and iteratively increase the local subband bit allocations to reduce the reconstructed signal's overall noise floor. This may cause subbands to be coded which previously have been allocated zero bits. The bit overhead in 'switching on' subbands in this way may need to reflect the cost in transmitting any predictor coefficients if PMODE is enabled.

The GBM routine can select from one of three different schemes for allocating the remaining bits. One option is to use a mmse approach that reallocates <u>all</u> of the bits such that the resulting noise floor is approximately flat. This is equivalent to disabling the psychoacoustic modeling initially. To achieve a mmse noise floor, the plot **160** of the subbands' RMS values shown in **FIG. 18a** is turned upside down as shown in **FIG. 18b** and "waterfilled" until all of the bits are exhausted. This well known technique is called waterfilling because the distortion level falls uniformly as the number of allocated bits increases. In the example shown, the first bit is assigned to subband 1, the second and third bits are assigned to subbands 1 and 2, the fourth through seventh bits are assigned to subbands 1, 2, 4 and 7, and so forth. Alternately, one bit can be assigned to each

subband to guarantee that each subband will be encoded, and then the remaining bits waterfilled.

A second, and preferred, option is to allocate the remaining bits according to the mmse approach and RMS plot described above. The effect of this method is to uniformly lower the noise floor **157** shown in **FIG. 17** while maintaining the shape associated with the psychoacoustic masking. This provides a good compromise between the psychoacoustic and mse distortion.

The third approach is to allocate the remaining bits using the mmse approach as applied to a plot of the difference between the RMS and MNR values for the subbands. The effect of this approach is to smoothly morph the shape of the noise floor from the optimal psychoacoustic shape **157** to the optimal (flat) mmse shape **158** as the bit rate increases. In any of these schemes, if the coding error in any subband drops below 0.5 LSB, with respect to the source PCM, then no more bits are allocated to that subband. Optionally fixed maximum values of subband bit allocations may be used to limit the maximum number of bits allocated to particular subbands.

In the encoding system discussed above, we have assumed that the average bit rate per sample is fixed and have generated the bit allocation to maximize the fidelity of the reconstructed audio signal. Alternately, the distortion level, mse or perceptual, can be fixed and the bit rate allowed to vary to satisfy the distortion level. In the mmse approach, the RMS plot is simply waterfilled until the distortion level is satisfied. The required bit rate will vary based upon the RMS levels of the subbands. In the psychoacoustic approach, the bits are allocated to satisfy the individual MNRs. As a result, the bit rate will vary based upon the individual SMRs and prediction gains. This type of allocation is not presently useful because contemporary decoders operate at a fixed rate. However, alternative delivery systems such as ATM or random access storage media may make variable rate coding practical in the

near future.

Quantization of Bit Allocation Indexes (ABIT)

The bit allocation indexes (ABIT) are generated for each subband and each audio channel by an adaptive bit allocation routine in the global bit management process. The purpose of the indexes at the encoder is to indicate the number of levels **162** shown in **FIG. 10** that are necessary to quantize the difference signal to obtain a subjectively optimum reconstruction noise floor in the decoder audio. At the decoder they indicate the number of levels necessary for inverse quantization. Indexes are generated for every analysis buffer and their values can range from 0 to 27. The relationship between index value, the number of quantizer levels and the approximate resulting differential subband SN_QR is shown in **Table 3**. Because the difference signal is normalized, the step-size **164** is set equal to one.

Table 3

ABIT Index	# of Q Levels	Code Length (bits)	SN_QR(dB)
0	0	0	-
1	3	variable	8
2	5	variable	12
3	7 (or 8)	variable (or 3)	16
4	9	variable	19
5	13	variable	21
6	17 (or 16)	variable (or 4)	24
7	25	variable	27
8	33 (or 32)	variable (or 5)	30
9	65 (or 64)	variable (or 6)	36
10	129 (or 128)	variable (or 7)	42
11	256	8	48
12	512	9	54
13	1024	10	60
14	2048	11	66
15	4096	12	72
16	8192	13	78
17	16384	14	84

18	32768	15	90
19	65536	16	96
20	131072	17	102
21	262144	18	108
22	524268	19	114
23	1048576	20	120
24	2097152	21	126
25	4194304	22	132
26	8388608	23	138
27	16777216	24	144

The bit allocation indexes (ABIT) are either transmitted to the decoder directly using 4-bit unsigned integer code words, 5-bit unsigned integer code words, or using a 12-level entropy table. Typically, entropy coding would be employed for low-bit rate applications to conserve bits. The method of encoding ABIT is set by the mode control at the encoder and is transmitted to the decoder. The entropy coder maps **166** the ABIT indexes to a particular codebook identified by a BHUFF index and a specific code VABIT in the codebook using the process shown in FIG. 12 with 12-level ABIT tables.

Global Bit Rate Control

Since both the side information and differential subband samples can optionally be encoded using entropy variable length code books, some mechanism must be employed to adjust the resulting bit rate of the encoder when the compressed bit stream is to be transmitted at a fixed rate. Because it is not normally desirable to modify the side information once calculated, bit rate adjustments are best achieved by iteratively altering the differential subband sample quantization process within the ADPCM encoder until the rate constraint is met.

In the system described, a global rate control (GRC) system **178** in **FIG. 10** adjusts the bit rate, which results from the process of mapping the quantizer level codes to the entropy table, by altering the statistical distribution of the level code values. The entropy tables are all assumed

to exhibit a similar trend of higher code lengths for higher level code values. In this case the average bit rate is reduced as the probability of low value code levels increases and vice-versa. In the ADPCM (or APCM) quantization process, the size of the scale factor determines the distribution, or usage, of the level code values. For example, as the scale factor size increases the differential samples will tend to be quantized by the lower levels, and hence the code values will become progressively smaller. This, in turn, will result in smaller entropy code word lengths and a lower bit rate.

The disadvantage of this method is that by increasing the scale factor size the reconstruction noise in the subband samples is also raised by the same degree. In practice, however, the adjustment of the scale factors is normally no greater than 1dB to 3dB. If a greater adjustment is required it would be better to return to the bit allocation and reduce the overall bit allocation rather than risk the possibility of audible quantization noise occurring in subbands which would use the inflated scale factor.

To adjust the entropy encoded ADPCM bit allocation, the predictor history samples for each subband are stored in a temporary bufferin case the ADPCM coding cycle is repeated. Next, the subband sample buffers are all encoded by the full ADPCM process using prediction coefficients A_H derived from the subband LPC analysis together with scale factors RMS (or PEAK), quantizer bit allocations ABIT, transient modes TMODE, and prediction modes PMODE derived from the estimated difference signal. The resulting quantizer level codes are buffered and mapped to the entropy variable length code book, which exhibits the lowest bit usage again using the bit allocation index to determine the code book sizes.

The GRC system then analyzes the number of bits used for each subband using the same bit allocation index over all indexes. For example, when ABIT=1 the bit allocation calculation in the global bit management could have assumed

an average rate of 1.4 per subband sample (i.e. the average rate for the entropy code book assuming optimal level code amplitude distribution). If the total bit usage of all the subbands for which ABIT=1 is greater than 1.4/(total number of subband samples) then the scale factors could be increased throughout all of these subbands to affect a bit rate reduction. The decision to adjust the subband scale factors is preferably left until all the ABIT index rates have been accessed. As a result, the indexes with bit rates lower than that assumed in the bit allocation process may compensate for those with bit rates above that level. This assessment may also be extended to cover all audio channels where appropriate.

The recommended procedure for reducing overall bit rate is to start with the lowest ABIT index bit rate which exceeds the threshold and increase the scale factors in each of the subbands which have this bit allocation. The actual bit usage is reduced by the number of bits that these subbands were originally over the nominal rate for that allocation. If the modified bit usage is still in excess of the maximum allowed, then the subband scale factors for the next highest ABIT index, for which the bit usage exceeds the nominal, are increased. This process is continued until the modified bit usage is below the maximum.

Once this has been achieved, the old history data is loaded into the predictors and the ADPCM encoding process **72** is repeated for those subbands which have had their scale factors modified. Following this, the level codes are again mapped to the most optimal entropy codebooks and the bit usage is recalculated. If any of the bit usage's still exceed the nominal rates then the scale factors are further increased and the cycle is repeated.

The modification to the scale factors can be done in two ways. The first is to transmit to the decoder an adjustment factor for each ABIT index. For example a 2-bit word could signal an adjustment range of say 0, 1, 2 and 3dB. Since the same adjustment factor is used for all

subbands which use the ABIT index, and only indexes 1-10 can use entropy encoding, the maximum number of adjustment factors that need to be transmitted for all subbands is 10. Alternately, the scale factor can be changed in each subband by selecting a high quantizer level. However, since the scale factor quantizers have step-sizes of 1.25 and 2.5dB respectively the scale factor adjustment is limited to these steps. Moreover, when using this technique the differential encoding of the scale factors and the resulting bit usage may need to be recalculated if entropy encoding is enabled.

Generally speaking the same procedure can also be used to increase the bit rate, i.e. when the bit rate is lower than the desired bit rate. In this case the scale factors would be decreased to force the differential samples to make greater use of the outer quantizer levels, and hence use longer code words in the entropy table.

If the bit usage for bit allocation indexes cannot be reduced within a reasonable number of iterations, or in the case when the scale factor adjustment factors are transmitted, the number of adjustment steps has reached the limit then two remedies are possible. First, the scale factors of subbands which are within the nominal rate may be increased, thereby lowering the overall bit rate. Alternately, the entire ADPCM encoding process can be aborted and the adaptive bit allocations across the subbands recalculated, this time using fewer bits.

Data Stream Format

The multiplexer **32** shown in **FIG. 10** packs the data for each channel and then multiplexes the packed data for each channel into an output frame to form the data stream **16**. The method of packing and multiplexing the data, i.e. the frame format **186** shown in **FIG. 19**, was designed so that the audio coder can be used over a wide range of applications and can be expanded to higher sampling frequencies, the amount of data in each frame is constrained, playback can be initiated on each sub-subframe independently to reduce latency, and decoding errors are reduced.

As shown, a single frame **186** (4096 PCM samples/ch) defines the bit stream boundaries in which sufficient information resides to properly decode a block of audio and consists of 4 subframes **188** (1024 PCM samples/ch), which in turn are each made up of 4 sub-subframes **190** (256 PCM samples/ch). The frame synchronization word **192** is placed at the beginning of each audio frame. The frame header information **194** primarily gives information regarding the construction of the frame **186**, the configuration of the encoder which generated the stream and various optional operational features such as embedded dynamic range control and time code. The optional header information **196** tells the decoder if downmixing is required, if dynamic range compensation was done and if auxiliary data bytes are included in the data stream. The audio coding headers **198** indicate the packing arrangement and coding formats used at the encoder to assemble the coding 'side information', i.e. bit allocations, scale factors, PMODES, TMODES, codebooks, etc. The remainder of the frame is made up of SUBFS consecutive audio subframes **188**.

Each subframe begins with the audio coding side information **200** which relays information regarding a number of key encoding systems used to compress the audio to the decoder. These include transient detection, predictive coding, adaptive bit allocation, high frequency vector quantization, intensity coding and adaptive scaling. Much of this data is unpacked from the data stream using the audio coding header information above. The hgih frequency VQ code array **202** consists of 10-bit indexes per high frequency subband indicated by VQSUB indexes. The low frequency effects array **204** is optional and represents the very low frequency data that can be used to drive, for example, a subwoofer.

The audio array **206** is decoded using Huffman/fixed inverse quantizers and is divided into a number of sub-subfr ames (SSC), each decoding up to 256 PCM samples per audio channel. The oversampled audio array **208** is only present

if the sampling frequency is greater than 48kHz. To remain compatible, decoders which cannot operate at sampling rates above 48kHz should skip this audio data array. DSYNC **210** is used to verify the end of the subframe position in audio frame. If the position does not verify, the audio decoded in the subframe is declared unreliable. As a result, either that frame is muted or the previous frame is repeated.

Subband Decoder

FIG. 20 is a block diagram of the subband sample decoder **18**, respectively. The decoder is quite simple compared to the encoder and does not involve calculations that are of fundamental importance to the quality of the reconstructed audio such as bit allocations. After synchronization the unpacker **40** unpacks the compressed audio data stream **16**, detects and if necessary corrects transmission induced errors, and demultiplexes the data into individual audio channels. The subband differential signals are requantized into PCM signals and each audio channel is inverse filtered to convert the signal back into the time domain.

Receive Audio Frame and unpack Headers

The coded data stream is packed (or framed) at the encoder and includes in each frame additional data for decoder synchronization, error detection and correction, audio coding status flags and coding side information, apart from the actual audio codes themselves. The unpacker **40** detects the SYNC word and extracts the frame size FSIZE. The coded bit stream consists of consecutive audio frames, each beginning with a 32-bit (0x7ffe8001) synchronization word (SYNC). The physical size of the audio frame, FSIZE is extracted from the bytes following the sync word. This allows the programmer to set an 'end of frame' timer to reduce software overheads. Next NBlks is extracted which allows the decoder to compute the Audio Window Size (32 (Nblks+1)). This tells the decoder what side information to extract and how many reconstructed samples to generate.

As soon as the frame header bytes (sync,ftype,sur

p,nblks,fsize,amode,sfreq,rate,mixt,dynf,dynct,time,auxcnt, lff,hflag) have been received, the validity of the first 12 bytes may checked using the Reed Solomon check bytes, HCRC. These will correct 1 erroneous byte out of the 14 bytes or flag 2 erroneous bytes. After error checking is complete the header information is used to update the decoder flags.

The headers (filts,vernum,chist,pcmr,unspec) following HCRC and up to the optional information, may be extracted and used to update the decoder flags. Since this information will not change from frame to frame, a majority vote scheme may be used to compensate for bit errors. The optional header data (times,mcoeff,dcoeff,auxd,ocrc) is extracted according to the mixct, dynf, time and auxcnt headers. The optional data may be verified using the optional Reed Solomon check bytes OCRC.

The audio coding frame headers (subfs, subs,chs,vqsub,joinx,thuff,shuff,bhuff,sel5,sel7,sel9,sel13,sel17,sel25, sel33,sel65,sel129,ahcrc) are transmitted once in every frame. They may be verified using the audio Reed Solomon check bytes AHCRC. Most headers are repeated for each audio channel as defined by CHS.

Unpack Subframe Coding Side Information

The audio coding frame is divided into a number of subframes (SUBFS). All the necessary side information (pmode, pvq, tmode, scales, abits, hfreq) is included to properly decode each subframe of audio without reference to any other subframe. Each successive subframe is decoded by first unpacking its side information.

A 1-bit prediction mode (PMODE) flag is transmitted for every active subband and across all audio channel. The PMODE flags are valid for the current subframe. PMODE=0 implies that the predictor coefficients are not included in the audio frame for that subband. In this case the predictor coefficients in this band are reset to zero for the duration of the subframe. PMODE=1 implies that the side information contains predictor coefficients for this subband. In this case the predictor coefficients are

extracted and installed in its predictor for the duration of the subframe.

For every PMODE=1 in the pmode array a corresponding prediction coefficient VQ address index is located in array PVQ. The indexes are fixed unsigned 12-bit integer words and the 4 prediction coefficients are extracted from the look-up table by mapping the 12-bit integer to the vector table **266**.

The bit allocation indexes (ABIT) indicate the number of levels in the inverse quantizer which will convert the subband audio codes back to absolute values. The unpacking format differs for the ABITs in each audio channel, depending on the BHUFF index and a specific VABIT code **256**.

The transient mode side information (TMODE) **238** is used to indicate the position of transients in each subband with respect to the subframe. Each subframe is divided into 1 to 4 sub-subframes. In terms of subband samples each sub-subframe consists of 8 samples. The maximum subframe size is 32 subband samples. If a transient occurs in the first sub-subframe then tmode=0. A transient in the second sub-subframe is indicated when tmode=1, and so on. To control transient distortion, such as pre-echo, two scale factors are transmitted for subframe subbands where TMODE is greater then 0. The THUFF indexes extracted from the audio headers determine the method required to decode the TMODEs. When THUFF=3, the TMODEs are unpacked as un-signed 2-bit integers.

Scale factor indexes are transmitted to allow for the proper scaling of the subband audio codes within each subframe. If TMODE is equal to zero then one scale factor is transmitted. If TMODE is greater than zero for any subband, then two scale factors are transmitted together. The SHUFF indexes **240** extracted from the audio headers determine the method required to decode the SCALES for each separate audio channel. The $VDRMS_{QL}$ indexes determine the value of the RMS scale factor.

In certain modes SCALES indexes are unpacked using a

choice of five 129-level signed Huffman inverse quantizers. The resulting inverse quantized indexes are, however, differentially encoded and are converted to absolute as follows;

ABS_SCALE(n+1)=SCALES(n)-SCALES(n+1) where n is the nth differential scale factor in the audio channel starting from the first subband.

At low bit-rate audio coding modes, the audio coder uses vector quantization to efficiently encode high frequency subband audio samples directly. No differential encoding is used in these subbands and all arrays relating to the normal ADPCM processes must be held in reset. The first subband which is encoded using VQ is indicated by VQSUB and all subbands up to SUBS are also encoded in this way.

The high frequency indexes (HFREQ) are unpacked **248** as fixed 10-bit unsigned integers. The 32 samples required for each subband subframe are extracted from the Q4 fractional binary LUT by applying the appropriate indexes. This is repeated for each channel in which the high frequency VQ mode is active

The decimation factor for the effects channel is always X128. The number of 8-bit effect samples present in LFE is given by SSC*2 when PSC=0 or (SSC+1)*2 when PSC is non zero. An additional 7-bit scale factor (unsigned integer) is also included at the end of the LFE array and this is converted to rms using a 7-bit LUT.

Unpack Sub-subframe Audio codes array

The extraction process for the subband audio codes is driven by the ABIT indexes and, in the case when ABIT<11, the SEL indexes also. The audio codes are formatted either using variable length Huffman codes or fixed linear codes. Generally ABIT indexes of 10 or less will imply a Huffman variable length codes, which are selected by codes VQL(n) **258**, while ABIT above 10 always signify fixed codes. All quantizers have a mid-tread, uniform characteristic. For the fixed code (Y^2) quantizers the most negative level is

dropped. The audio codes are packed into sub-subframes, each representing a maximum of 8 subband samples, and these sub-subframes are repeated up to four times in the current subframe.

If the sampling rate flag (SFREQ) indicates a rate higher than 48kHz then the over_audio data array will exist in the audio frame. The first two bytes in this array will indicate the byte size of over_audio. Further, the sampling rate of the decoder hardware should be set to operate at SFREQ/2 or SFREQ/4 depending on the high frequency sampling rate.

Unpack Synchronization Check

A data unpacking synchronization check word DSYNC=0xffff is detected at the end of every subframe to allow the unpacking integrity to be verified. The use of variable code words in the side information and audio codes, as is the case for low audio bit rates, can lead to unpacking misalignment if either the headers, side information or audio arrays have been corrupted with bit errors. If the unpacking pointer does not point to the start of DSYNC then it can be assumed the previous subframe audio is unreliable.

Once all of the side information and audio data is unpacked, the decoder reconstructs the multi-channel audio signal a subframe at a time. **FIG. 20** illustrates the baseband decoder portion for a single subband in a single channel.

Reconstruct RMS Scale Factors

The decoder reconstructs the RMS scale factors (SCALES) for the ADPCM, VQ and JFC algorithms. In particular, the VTMODE and THUFF indexes are inverse mapped to identify the transient mode (TMODE) for the current subframe. Thereafter, the SHUFF index, $VDRMS_{QL}$ codes and TMODE are inverse mapped to reconstruct the differential RMS code. The differential RMS code is inverse differential coded **242** to select the RMS code, which is them inverse quantized **244** to produce the RMS scale factor.

Inverse Quantize High Frequency Vectors

The decoder inverse quantizes the high frequency vectors to reconstruct the subband audio signals. In particular, the extracted high frequency samples (HFREQ), which are signed 8-bit fractional (Q4) binary number, as identified by the start VQ subband (VQSUBS) are mapped to an inverse VQ lut **248**. The selected table value is inverse quantized **250**, and scaled **252** by the RMS scale factor.

Inverse Quantize Audio Codes

Before entering the ADPCM loop the audio codes are inverse quantized and scaled to produce reconstructed subband difference samples. The inverse quantization is achieved by first inverse mapping the VABIT and BHUFF index to specify the ABIT index which determines the step-size and the number of quantization levels and inverse mapping the SEL index and the VQL(n) audio codes which produces the quantizer level codes QL(n). Thereafter, the code words QL(n) are mapped to the inverse quantizer look-up table **260** specified by ABIT and SEL indexes. Although the codes are ordered by ABIT, each separate audio channel will have a separate SEL specifier. The look-up process results in a signed quantizer level number which can be converted to unit rms by multiplying with the quantizer step-size. The unit rms values are then converted to the full difference samples by multiplying with the designated RMS scale factor (SCALES) **262**.

1. QL[n] = 1/Q[code[n]] where 1/Q is the inverse quantizer look-up table
2. Y[n] = QL[n] * StepSize[abits]
3. Rd[n] = Y[n] * scale_factor where Rd=reconstructed difference samples

Inverse ADPCM

The ADPCM decoding process is executed for each subband difference sample as follows;

1. Load the prediction coefficients from the inverse VQ lut **268**.
2. Generate the prediction sample by convolving the current

predictor coefficients with the previous 4 reconstructed subband samples held in the predictors history array **268**.

P[n] = sum (Coeff[i]*R[n-i]) for i=1, 4 where n=current sample period

3. Add the prediction sample to the reconstructed difference sample to produce a reconstructed subband sample **270**.

R[n]=Rd[n]+P[n]

4. Update the history of the predictor, ie copy the current reconstructed subband sample to the top of the history list.

R[n-i]=R[n-i+1] for I =4, 1

In the case when PMODE=0 the predictor coefficients will be zero, the prediction sample zero, and the reconstructed subband sample equates to the differential subband sample. Although in this case the calculation of the prediction is unnecessary, it is essential that the predictor history is kept updated in case PMODE should become active in future subframes. Further, if the HFLAG is active in the current audio frame, the predictor history should be cleared prior to decoding the very first sub-subfr ame in the frame. The history should be updated as usual from that point on.

In the case of high frequency VQ subbands or where subbands are deselected (i.e. above SUBS limit) the predictor history should remain cleared until such time that the subband predictor becomes active.

Selection Control of ADPCM, VQ and JFC Decoding

A first "switch" controls the selection of either the ADPCM or VQ output. The VQSUBS index identifies the start subband for VQ encoding. Therefore if the current subband is lower than VQSUBS, the switch selects the ADPCM output. Otherwise it selects the VQ output. A second "switch" **278** controls the selection of either the direct channel output or the JFC coding output. The JOINX index identifies which channels are joined and in which channel the reconstructed signal is generated. The reconstructed JFC signal forms the intensity source for the JFC inputs in the other channels. Therefore, if the current subband is part of a JFC and is

not the designated channel than, the switch selects the JFC output. Normally, the switch selects the channel output.

Down Matrixing

The audio coding mode for the data stream is indicated by AMODE. The decoded audio channels can then be redirected to match the physical output channel arrangement on the decoder hardware **280**.

Dynamic Range Control Data

Dynamic range coefficients DCOEFF may be optionally embedded in the audio frame at the encoding stage **282**. The purpose of this feature is to allow for the convenient compression of the audio dynamic range at the output of the decoder. Dynamic range compression is particularly important in listening environments where high ambient noise levels make it impossible to discriminate low level signals without risking damaging the loudspeakers during loud passages. This problem is further compounded by the growing use of 20-bit PCM audio recordings which exhibit dynamic ranges as high as 110dB.

Depending on the window size of the frame (NBLKS) either one, two or four coefficients are transmitted per audio channel for any coding mode (DYNF). If a single coefficient is transmitted, this is used for the entire frame. With two coefficients the first is used for the first half of the frame and the second for the second half of the frame. Four coefficients are distributed over each frame quadrant. Higher time resolution is possible by interpolating between the transmitted values locally.

Each coefficient is 8-bit signed fractional Q2 binary, and represents a logarithmic gain value as shown in table (53) giving a range of +/- 31.75dB in steps of 0.25dB. The coefficients are ordered by channel number. Dynamic range compression is affected by multiplying the decoded audio samples by the linear coefficient.

The degree of compression can be altered with the appropriate adjustment to the coefficient values at the decoder or switched off completely by ignoring the

coefficients.

32-band Interpolation Filterbank

The 32-band interpolation filter bank **44** converts the 32 subbands for each audio channel into a single PCM time domain signal. Non-perfect reconstruction coefficients (512-tap FIR filters) are used when FILTS=0. Perfect reconstruction coefficients are used when FILTS=1. Normally the cosine modulation coefficients will be pre-calculated and stored in ROM. The interpolation procedure can be expanded to reconstruct larger data blocks to reduce loop overheads. However, in the case of termination frames, the minimum resolution which may be called for is 32 PCM samples. The interpolation algorithm is as follows: create cosine modulation coefficients, read in 32 new subband samples to array XIN, multiply by cosine modulation coefficients and create temporary arrays SUM and DIFF, store history, multiply by filter coefficients, create 32 PCM output samples, update working arrays, and output 32 new PCM samples

Depending on the bit rate and the coding scheme in operation, the bit stream can specify either non-perfect or perfect reconstruction interpolation filter bank coefficients (FILTS). Since the encoder decimation filter banks are computed with 40-bit floating precision, the ability of the decoder to achieve the maximum theoretical reconstruction precision will depend on the source PCM word length and the precision of DSP core used to compute the convolutions and the way that the operations are scaled.

Low frequency Effects PCM interpolation

The audio data associated with the low-frequency effects channel is independent of the main audio channels. This channel is encoded using an 8-bit APCM process operating on a X128 decimated (120Hz bandwidth) 20-bit PCM input. The decimated effects audio is time aligned with the current subframe audio in the main audio channels. Hence, since the delay across the 32-band interpolation filterbank is 256 samples (512 taps), care must be taken to ensure that

the interpolated low-frequency effect channel is also aligned with the rest of the audio channels prior to output. No compensation is required if the effects interpolation FIR is also 512 taps.

The LFT algorithm uses a 512 tap 128X interpolation FIR as follows: map 7-bit scale factor to rms, multiply by step-size of 7-bit quantizer, generate sub sample values from the normalized values, and interpolate by 128 using a low pass filter such as that given for each sub sample.

Hardware Implementation

Figures 21 and **22** describe the basic functional structure of the hardware implementation of a six channel version of the encoder and decoder for operation at 32, 44.1 and 48kHz sampling rates. Referring to **Fig. 22**, Eight Analog Devices ADSP21020 40-bit floating point digital signal processor (DSP) chips **296** are used to implement a six channel digital audio encoder **298**. Six DSPs are used to encode each of the channels while the seventh and eighth are used to implement the "Global Bit Allocation and Management" and "Data Stream Formatter and Error Encoding" functions respectively. Each ADSP21020 is clocked at 33 MHz and utilize external 48bit X 32k program ram (PRAM) **300**, 40bit X 32k data ram (SRAM) **302** to run the algorithms. In the case of the encoders an 8bit X 512k EPROM **304** is also used for storage of fixed constants such as the variable length entropy code books. The data stream formatting DSP uses a Reed Solomon CRC chip **306** to facilitate error detection and protection at the decoder. Communications between the encoder DSPs and the global bit allocation and management is implemented using dual port static RAM **308**.

The encode processing flow is as follows. A 2-channel digital audio PCM data stream **310** is extracted at the output of each of the three AES/EBU digital audio receivers. The first channel of each pair is directed to CH1, 3 and 5 Encoder DSPs respectively while the second channel of each is directed to CH2, 4 and 6 respectively. The PCM samples are read into the DSPs by converting the serial PCM words to

parallel (s/p). Each encoder accumulates a frame of PCM samples and proceeds to encode the frame data as described previously. Information regarding the estimated difference signal (ed(n) and the subband samples (x(n)) for each channel is transmitted to the global bit allocation and management DSP via the dual port RAM. The bit allocation strategies for each encoder are then read back in the same manner. Once the encoding process is complete, the coded data and side information for the six channels is transmitted to the data stream formatter DSP via the global bit allocation and management DSP. At this stage CRC check bytes are generated selectively and added to the encoded data for the purposes of providing error protection at the decoder. Finally the entire data packet **16** is assembled and output.

A six channel hardware decoder implementation is described in **Fig. 22**. A single Analog Devices ADSP21020 40-bit floating point digital signal processor (DSP) chip **324** is used to implement the six channel digital audio decoder. The ADSP21020 is clocked at 33 MHz and utilize external 48bit X 32k program ram (PRAM) **326**, 40bit X 32k data ram (SRAM) **328** to run the decoding algorithm. An additional 8bit X 512k EPROM **330** is also used for storage of fixed constants such as the variable length entropy and prediction coefficient vector code books.

The decode processing flow is as follows. The compressed data stream **16** is input to the DSP via a serial to parallel converter (s/p) **332**. The data is unpacked and decoded as illustrated previously. The subband samples are reconstructed into a single PCM data stream **22** for each channel and output to three AES/EBU digital audio transmitter chips **334** via three parallel to serial converters (p/s) **335**.

While several illustrative embodiments of the invention have been shown and described, numerous variations and alternate embodiments will occur to those skilled in the art. For example, as processor speeds increase and the cost of memory is reduced, the sampling frequencies, transmission

rates and buffer size will most likely increase. Such variations and alternate embodiments are contemplated, and can be made without departing from the spirit and scope of the invention as defined in the appended claims.

WE CLAIM:

1. A multi-channel audio decoder for reconstructing multiple audio channels up to a decoder sampling rate from a data stream, in which each audio channel was sampled at an encoder sampling rate that is at least as high as the decoder sampling rate, subdivided into a plurality of frequency subbands, compressed and multiplexed into the data stream at a transmission rate, comprising:

an input buffer (324) for reading in and storing the data stream a frame at a time, each of said frames including a sync word, a frame header, an audio header, and at least one subframe, which includes audio side information, a plurality of sub-subframes having baseband audio codes over a baseband frequency range, a block of high sampling rate audio codes over a high sampling rate frequency range, and an unpack sync;

a demultiplexer (40) that a) detects the sync word, b) unpacks the frame header to extract a window size that indicates a number of audio samples in the frame and a frame size that indicates a number of bytes in the frame, said window size being set as a function of the ratio of the transmission rate to the encoder sampling rate so that the frame size is constrained to be less than the size of the input buffer, c) unpacks the audio header to extract the number of subframes in the frame and the number of encoded audio channels, and d) sequentially unpacks each subframe to extract the audio side information, demultiplex the baseband audio codes in each sub-subframe into the multiple audio channels and unpack each audio channel into its subband audio codes, demultiplex the high sampling rate audio codes into the multiple audio channels up to the decoder sampling rate and skip the remaining high sampling rate audio codes up to the encoder sampling rate, and detects the unpack sync to verify the end of the subframe;

a baseband decoder (42,44) that uses the side information to decode the subband audio codes into reconstructed subband signals a subframe at a time without reference to any other subframes;

a baseband reconstruction filter (44) that combines each channel's reconstructed subband signals into a reconstructed baseband signal a subframe at a time;

a high sampling rate decoder (58,60) that uses the side information to decode the high sampling rate audio codes into a reconstructed high sampling rate signal for each audio channel a subframe at a time; and

a channel reconstruction filter (62) that combines the reconstructed baseband and high sampling rate signals into a reconstructed multi-channel audio signal a subframe at a time.

2. The multi-channel audio decoder of claim 1, wherein the baseband reconstruction filter (44) comprises a non-perfect reconstruction (NPR) filterbank and a perfect reconstruction (PR) filterbank, and said frame header includes a filter code that selects one of said NPR and PR filterbanks.

3. The multi-channel audio decoder of claim 1, wherein the baseband decoder comprises a plurality of inverse adaptive differential pulse code modulation (ADPCM) coders (268,270) for decoding the respective subband audio codes, said side information including prediction coefficients for the respective ADPCM coders and a prediction mode (PMODE) for controlling the application of the prediction coefficients to the respective ADPCM coders to selectively enable and disable their prediction capabilities.

4. The multi-channel audio decoder of claim 1, wherein said side information comprises:

a bit allocation table for each channel's subbands, in which each subband's bit rate is fixed over the subframe;

at least one scale factor for each subband in each channel; and

a transient mode (TMODE) for each subband in each channel that identifies the number of scale factors and their associated sub-subframes, said baseband decoder scaling the subbands' audio codes by the respective scale factors in accordance with their TMODEs to facilitate decoding.

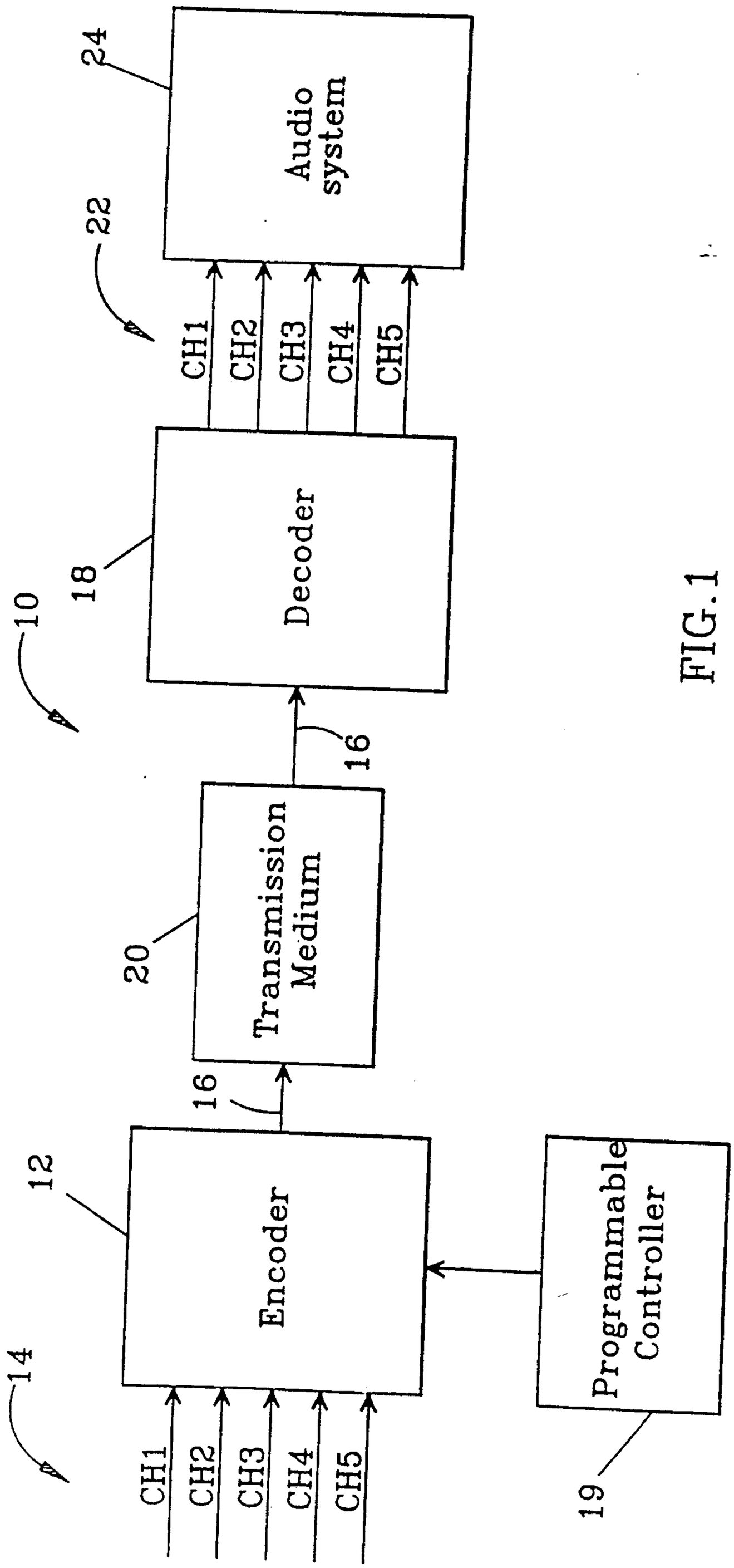
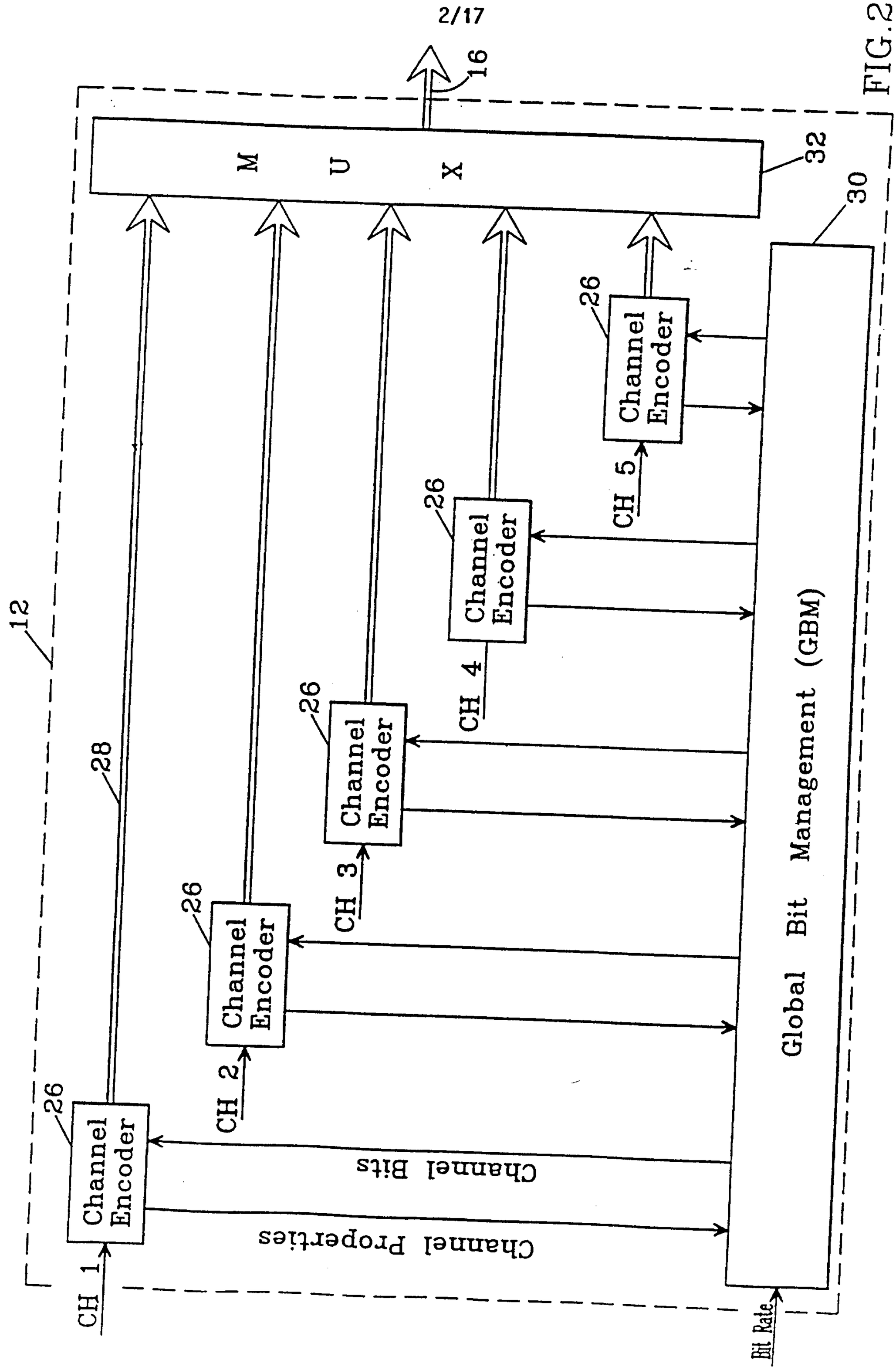

24
Audio system
22
CH1
CH2
CH3
CH4
CH5
18
Decoder
10
16
Transmission Medium
20
16
12
Encoder
Programmable Controller
14
CH1
CH2
CH3
CH4
CH5
19

FIG.1

FIG.2


16
MUX
32
30
12
28
26
Channel Encoder
CH 5
CH 4
CH 3
CH 2
CH 1
Global Bit Management (GBM)
Channel Bits
Channel Properties
Bit Rate

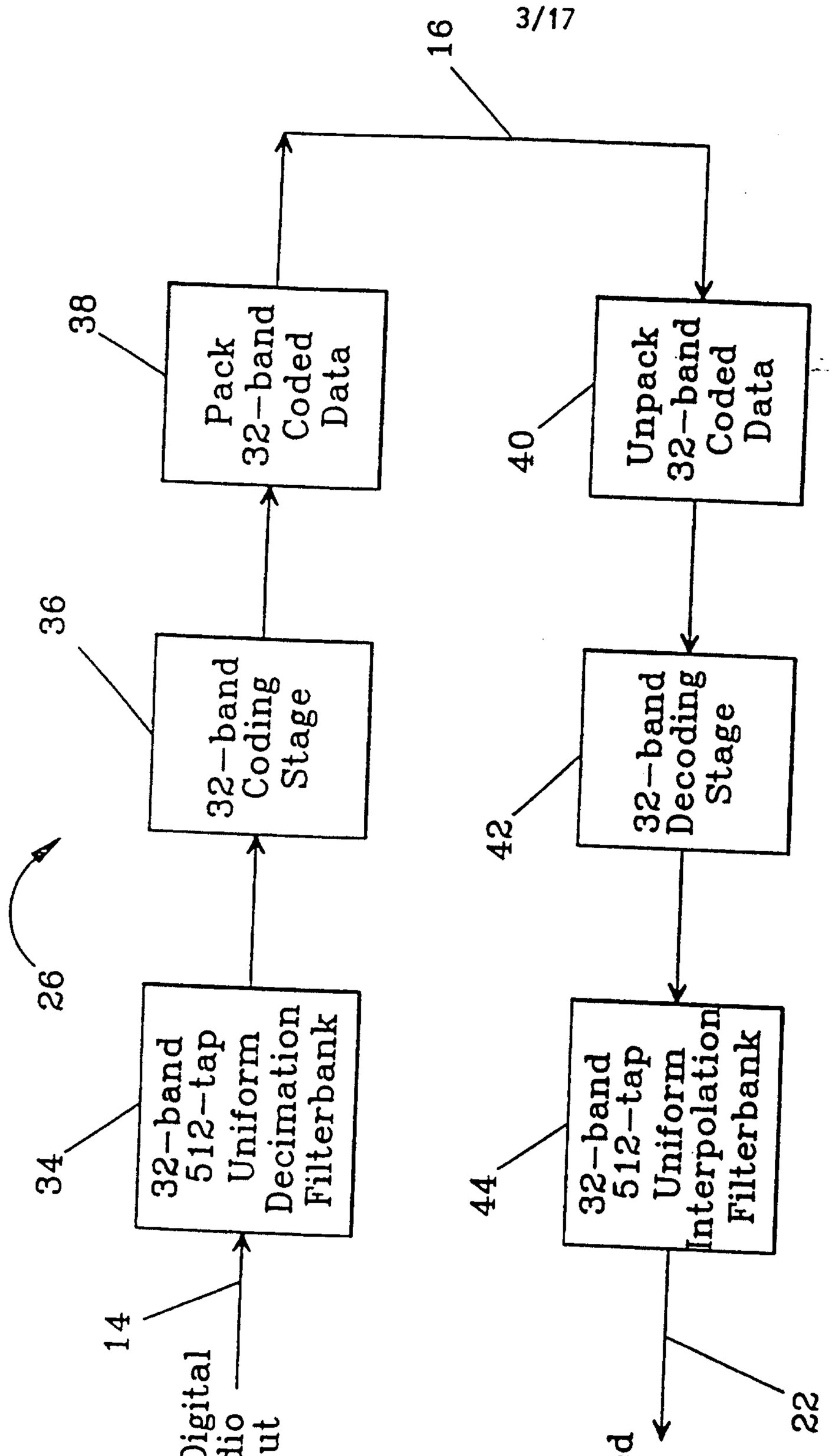

PCM Digital Audio Input
14
32-band 512-tap Uniform Decimation Filterbank
34
32-band Coding Stage
36
Pack 32-band Coded Data
38
16
Unpack 32-band Coded Data
40
32-band Decoding Stage
42
32-band 512-tap Uniform Interpolation Filterbank
44
Reconstructed PCM Digital Audio Signal
22
26
18

FIG.3

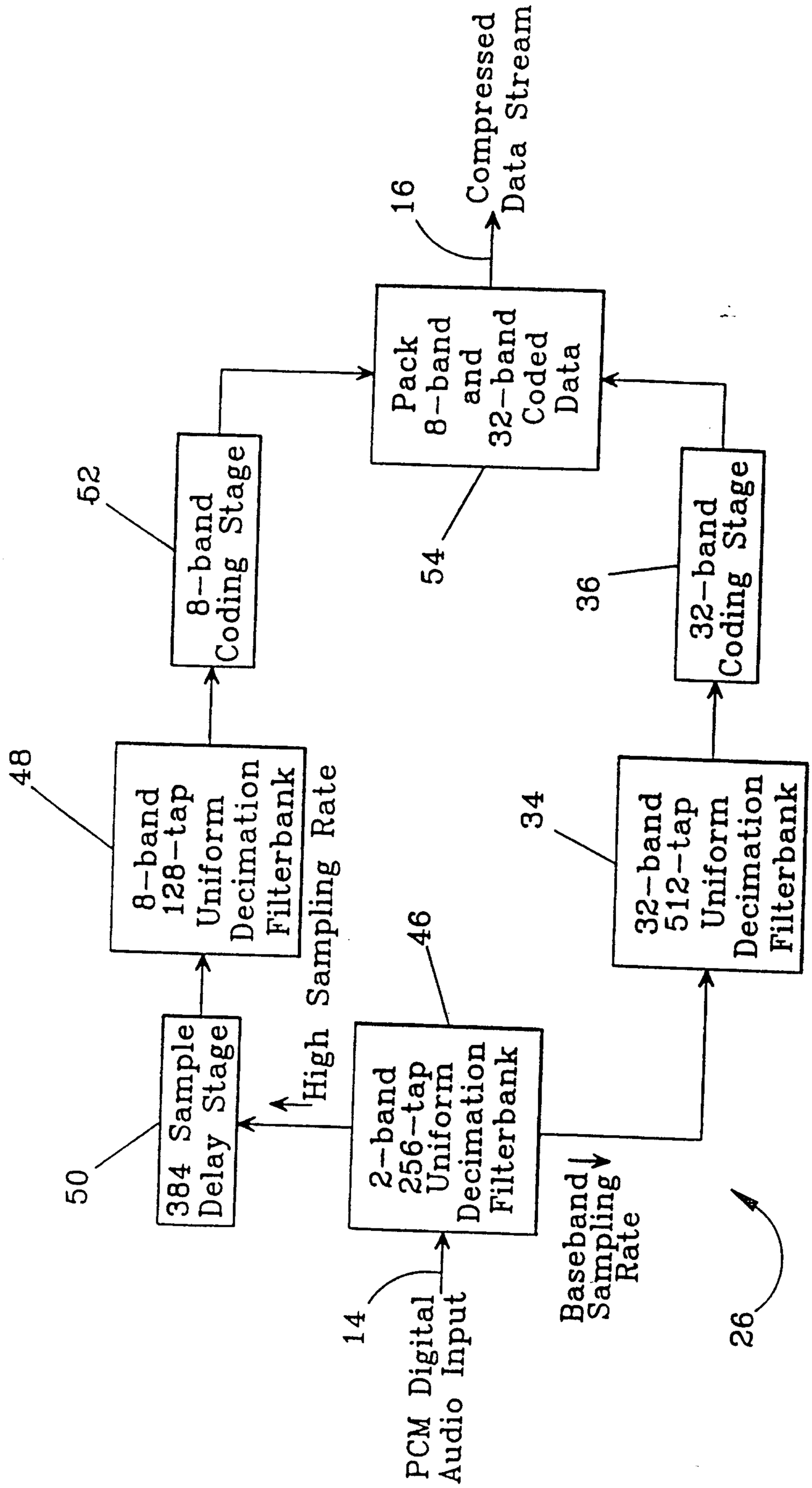

PCM Digital Audio Input
14
Baseband Sampling Rate
2-band 256-tap Uniform Decimation Filterbank
46
High Sampling Rate
384 Sample Delay Stage
50
8-band 128-tap Uniform Decimation Filterbank
48
32-band 512-tap Uniform Decimation Filterbank
34
8-band Coding Stage
52
32-band Coding Stage
36
Pack 8-band and 32-band Coded Data
54
16
Compressed Data Stream
26

FIG.4a

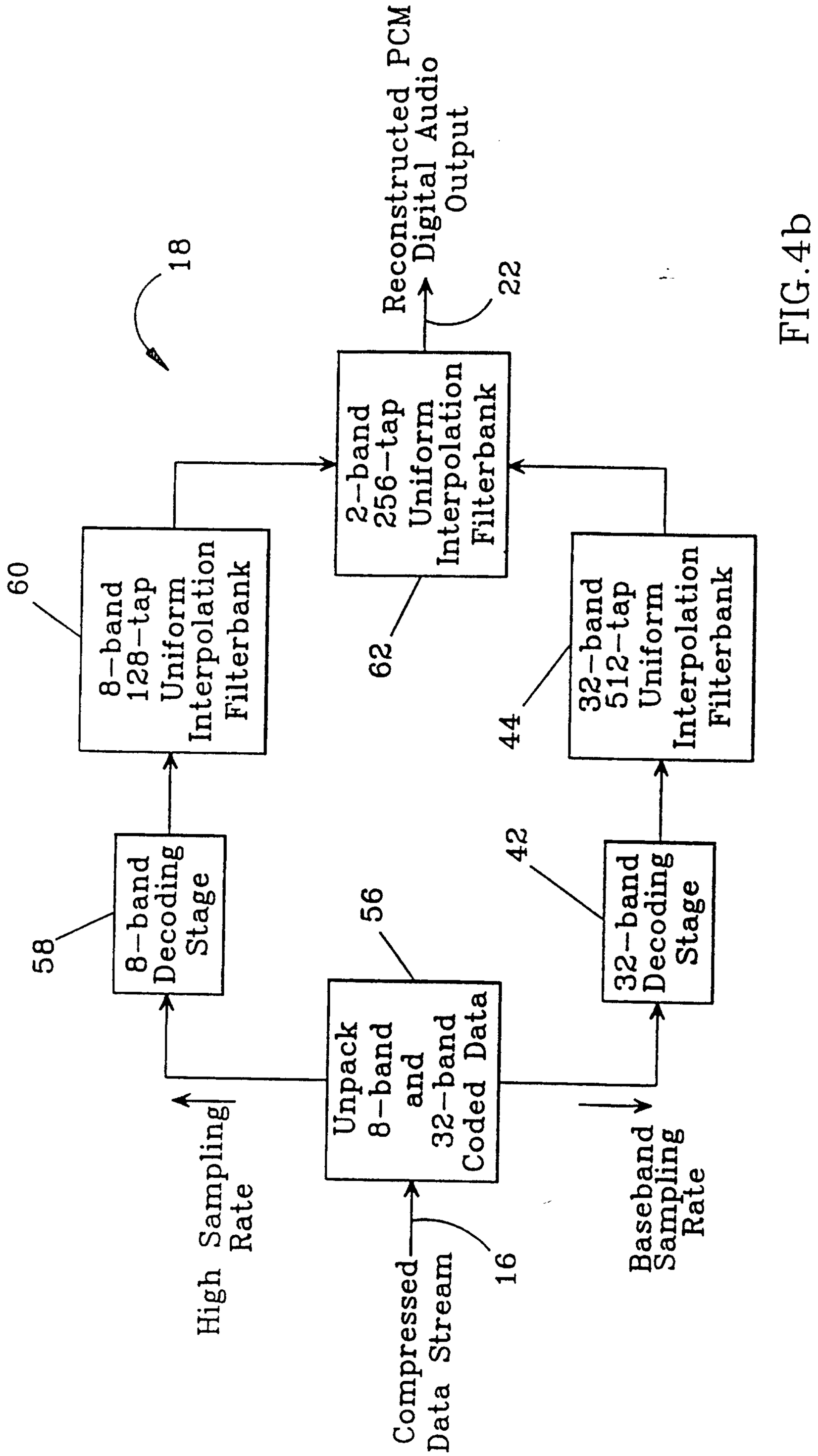

18
60
8–band
128–tap
Uniform
Interpolation
Filterbank
62
2–band
256–tap
Uniform
Interpolation
Filterbank
22
Reconstructed PCM
Digital Audio
Output
44
32–band
512–tap
Uniform
Interpolation
Filterbank
58
8–band
Decoding
Stage
56
Unpack
8–band
and
32–band
Coded Data
42
32–band
Decoding
Stage
High Sampling
Rate
Compressed
Data Stream
16
Baseband
Sampling
Rate

FIG.4b

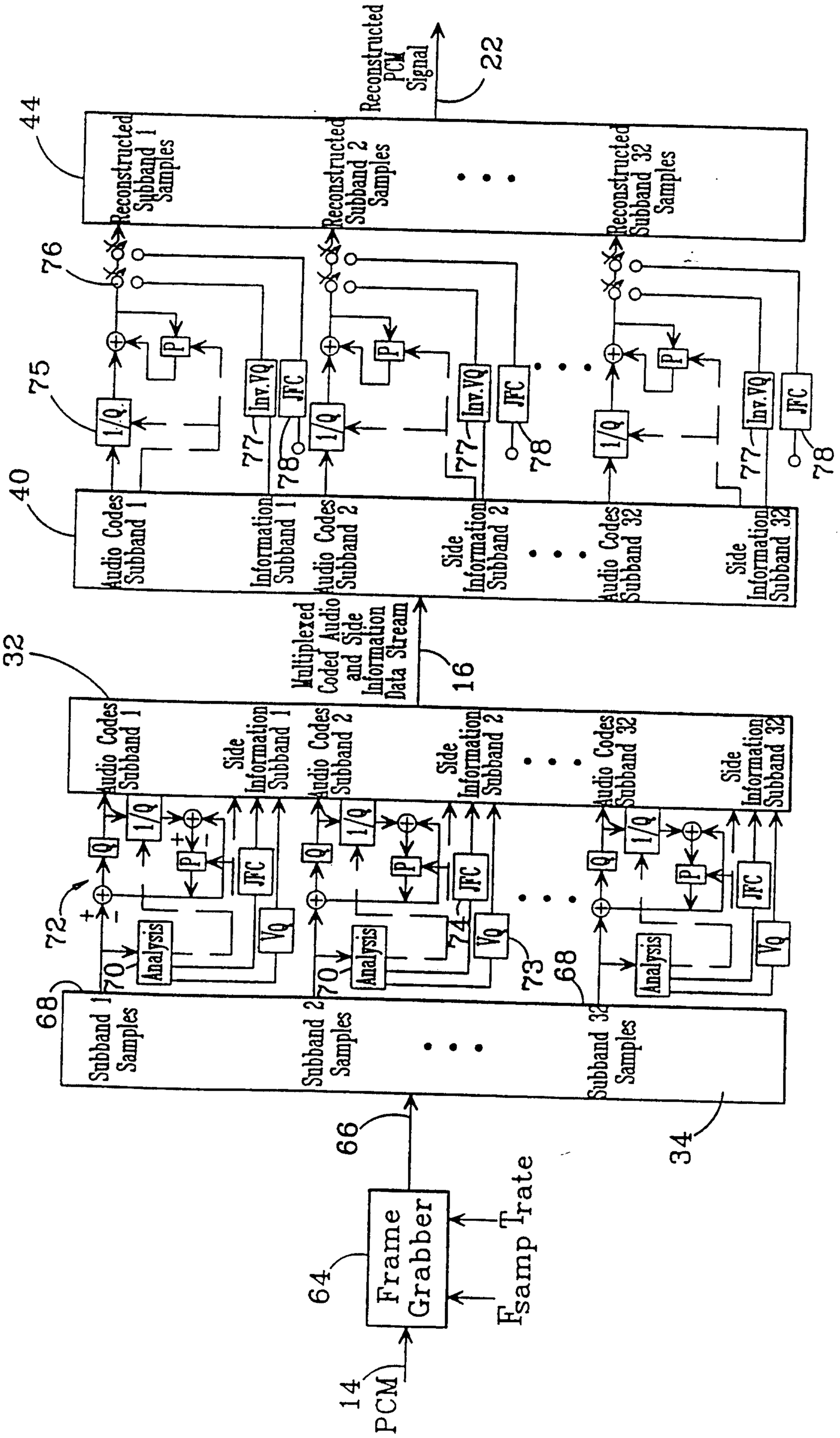

Frame Grabber
64
14
PCM
66
34
Subband 1 Samples
Subband 2 Samples
Subband 32 Samples
68
70
Analysis
72
Q
1/Q
P
VQ
73
74
JFC
Audio Codes Subband 1
Side Information Subband 1
Audio Codes Subband 2
Side Information Subband 2
Audio Codes Subband 32
Side Information Subband 32
32
Multiplexed Coded Audio and Side Information Data Stream
16
40
75
Inv.VQ
77
78
76
44
Reconstructed PCM Signal
22
Reconstructed Subband 1 Samples
Reconstructed Subband 2 Samples
Reconstructed Subband 32 Samples
F_{samp}
T_{rate}

FIG. 5

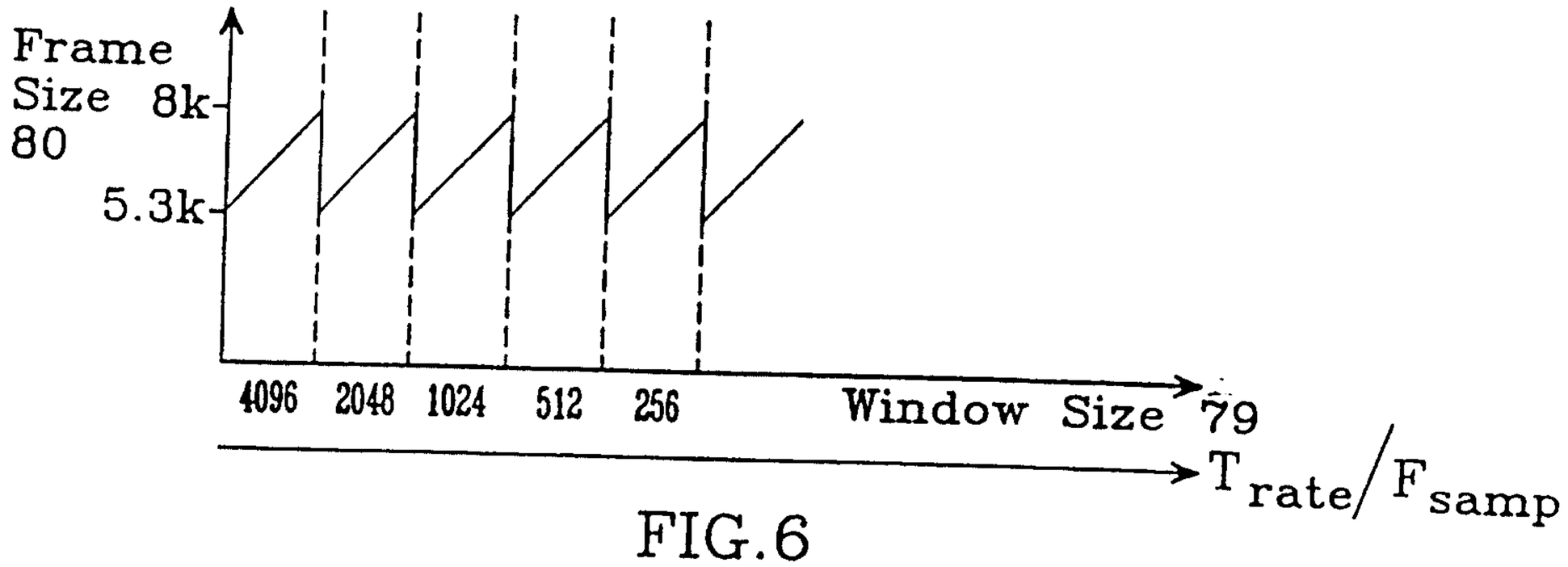


FIG.6

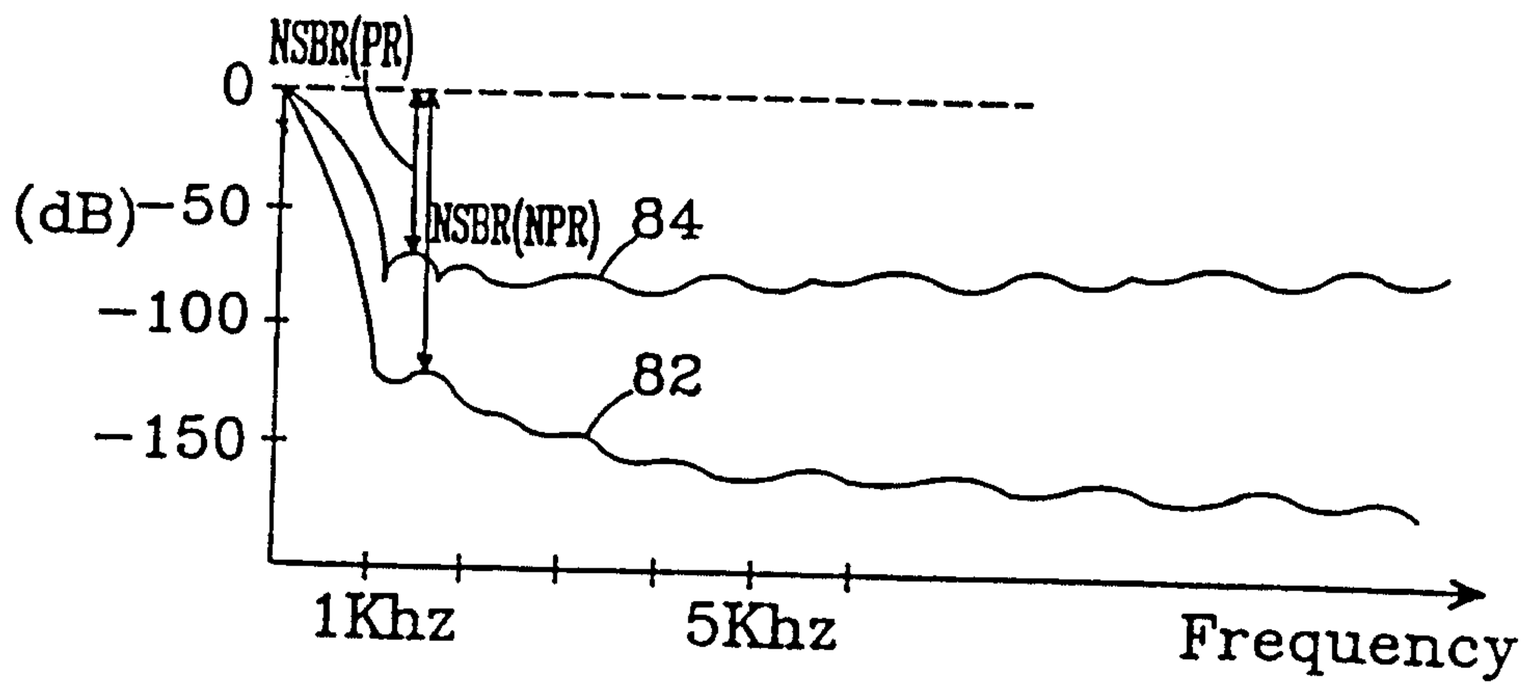


FIG.7

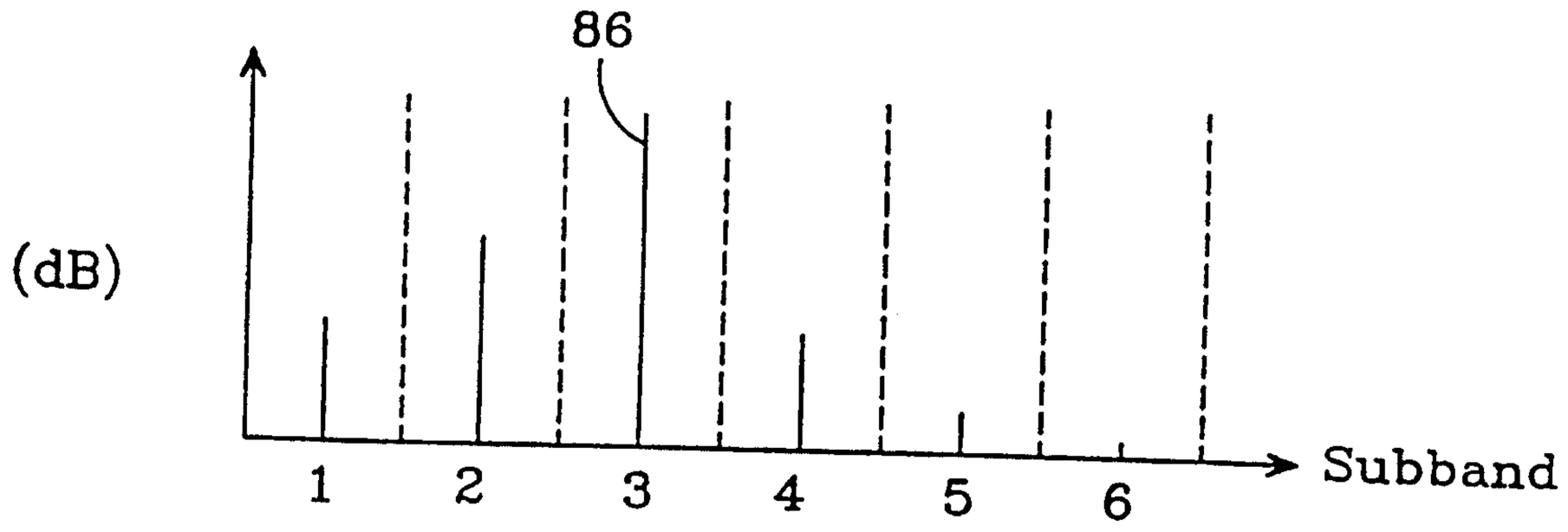


FIG.8

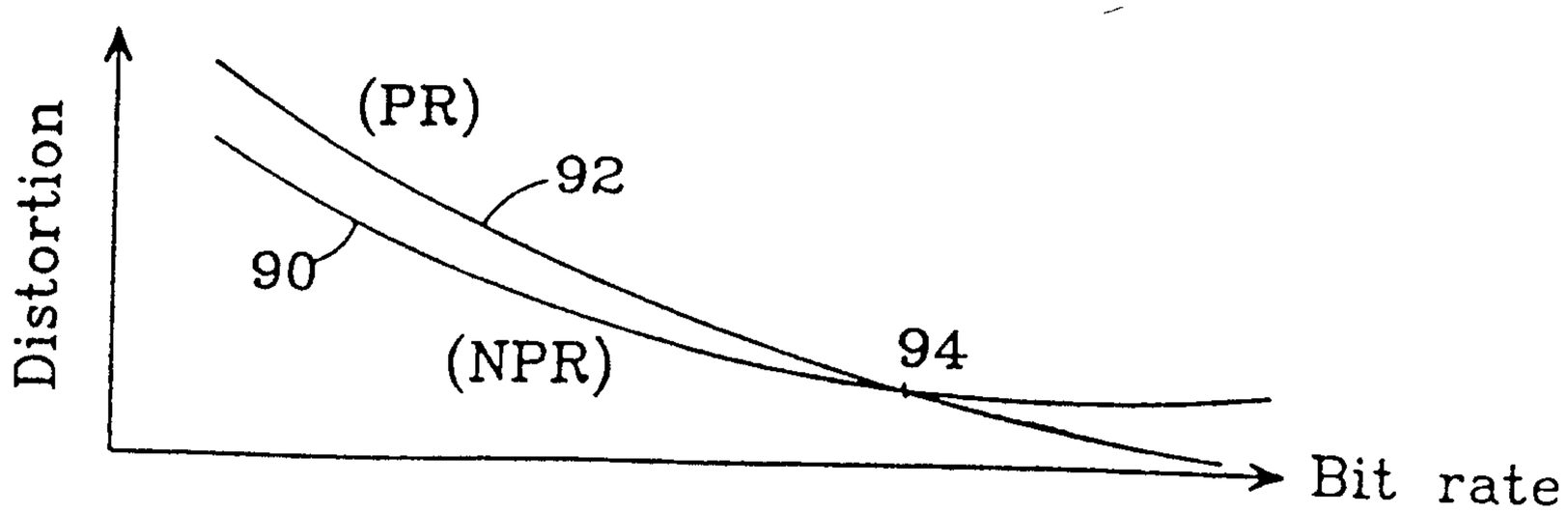


FIG.9

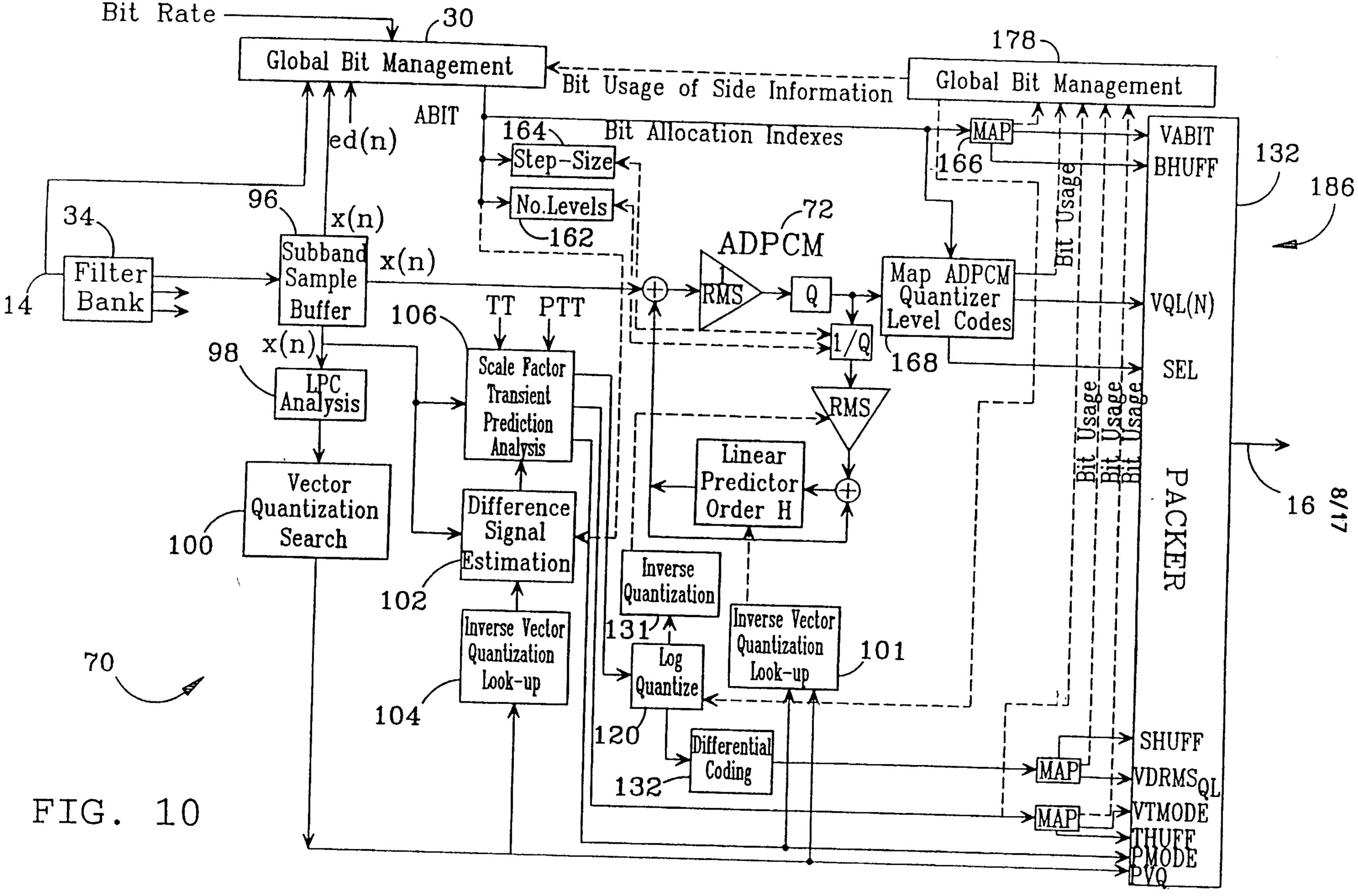


FIG. 10

CA 02331611 2001-01-24

109

Subframe Buffer

TMODE	Sub-subframe 1	Sub-subframe 2	Sub-subframe 3	Sub-subframe 4
0				
1			X X	X X
2				X X
3				

108

FIG. 11A

109

Subframe Buffer

TMODE	Sub-subframe 1	Sub-subframe 2	Sub-subframe 3	Sub-subframe 4
0	RMS 1 or Peak 1			
1	RMS 1 or Peak 1	RMS 2 or Peak 2		
2	RMS 1 or Peak 1		RMS 2 or Peak 2	
3	RMS 1 or Peak 1			RMS 2 or Peak 2

110

FIG. 11B

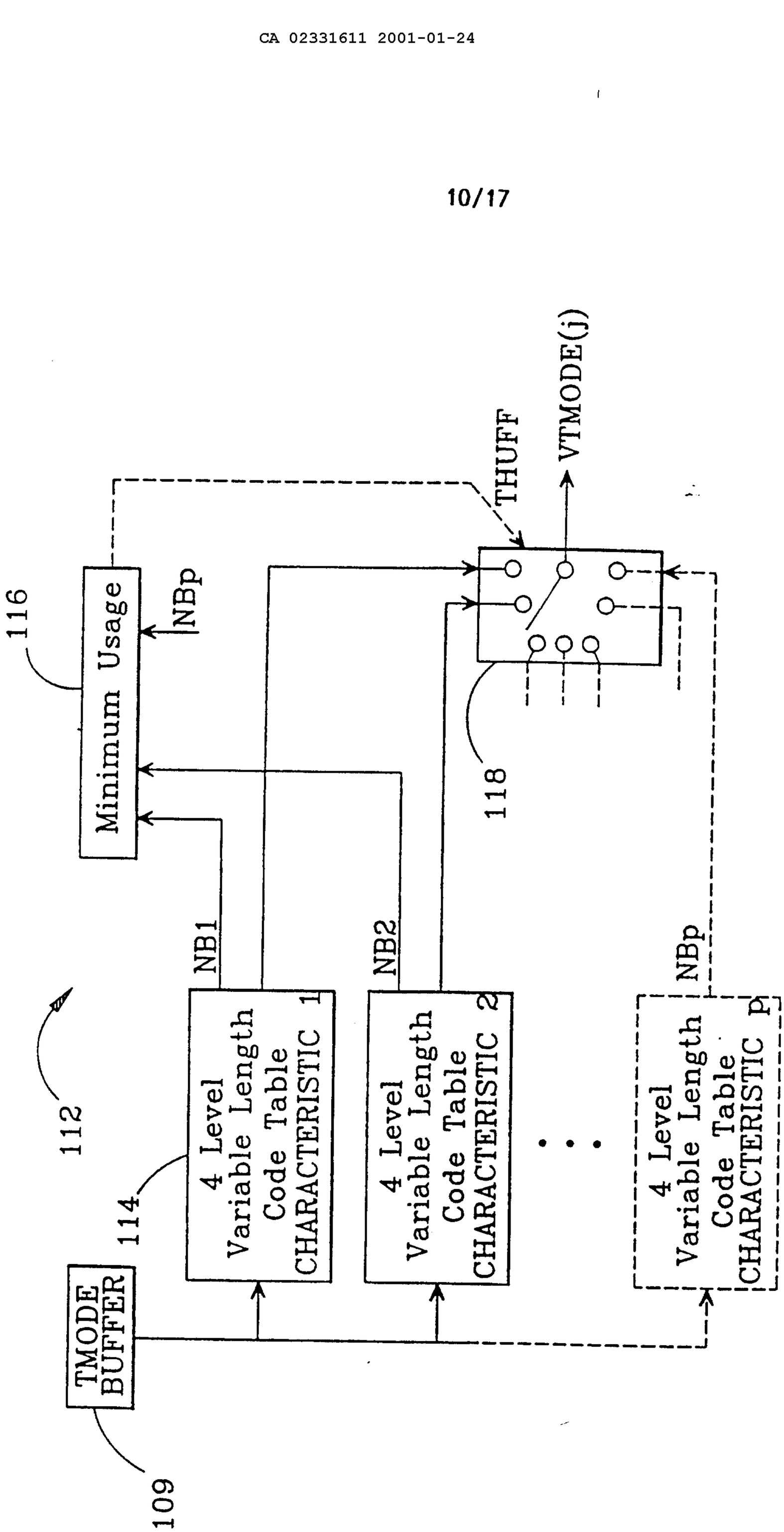

109
TMODE BUFFER
112
114
4 Level Variable Length Code Table CHARACTERISTIC 1
4 Level Variable Length Code Table CHARACTERISTIC 2
4 Level Variable Length Code Table CHARACTERISTIC p
NB1
NB2
NBp
116
Minimum Usage
NBp
118
THVTMODE(j)
BUFF

FIG. 12

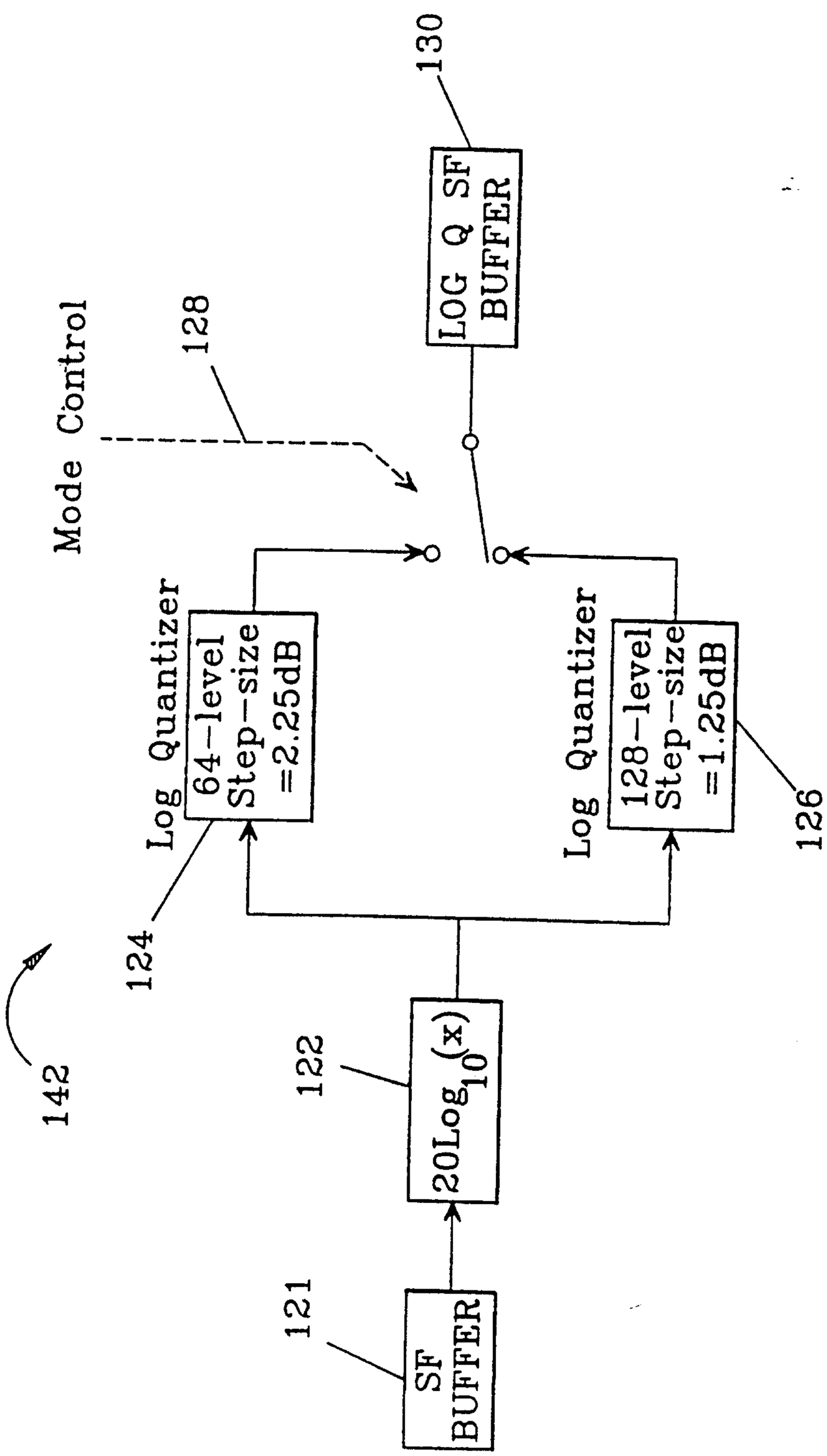

130
LOG Q SF BUFFER
Mode Control
128
Log Quantizer
64-level Step-size =2.25dB
124
Log Quantizer
128-level Step-size =1.25dB
126
142
122
20Log10(x)
121
SF BUFFER

FIG. 13

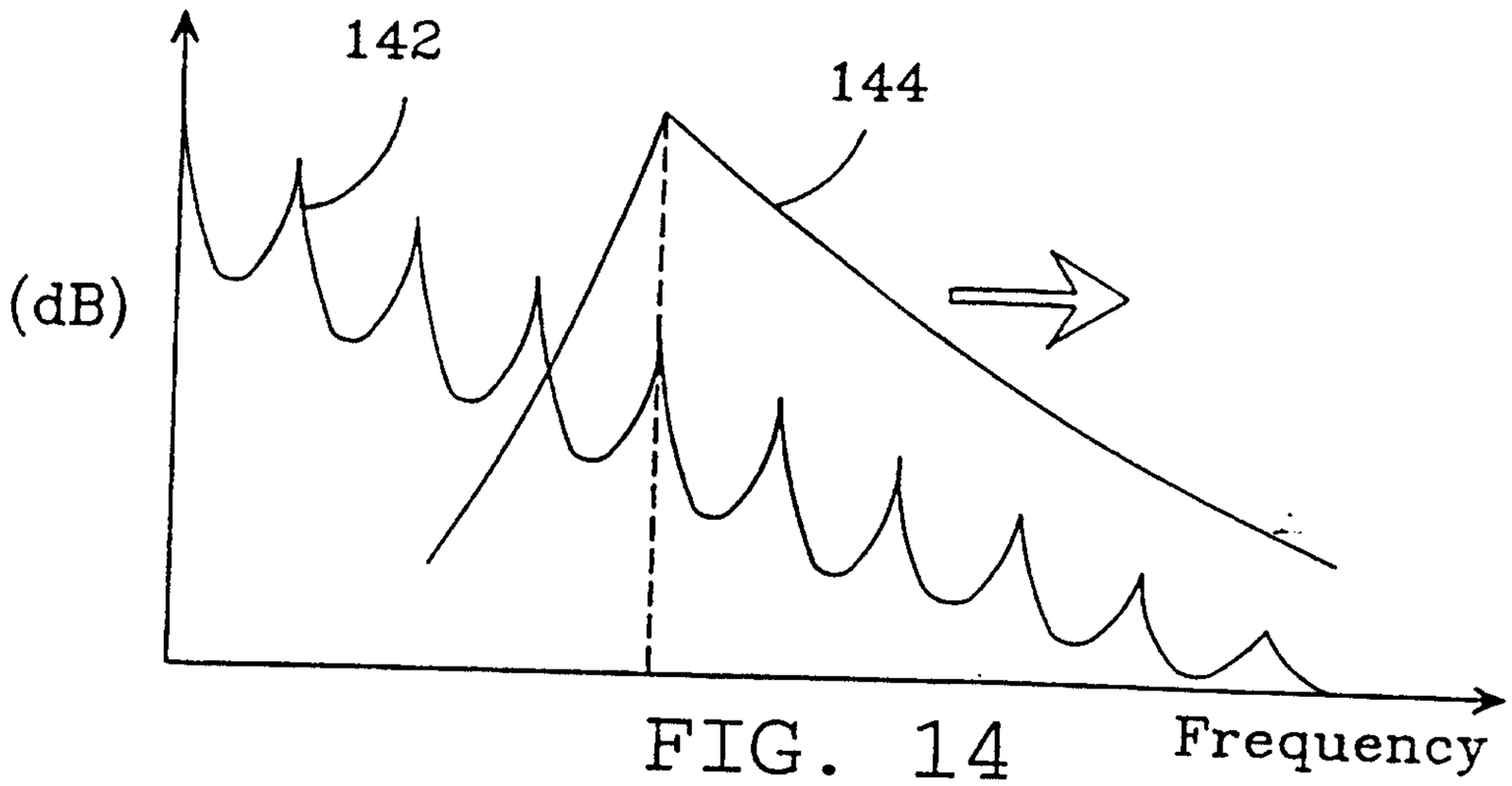


FIG. 14

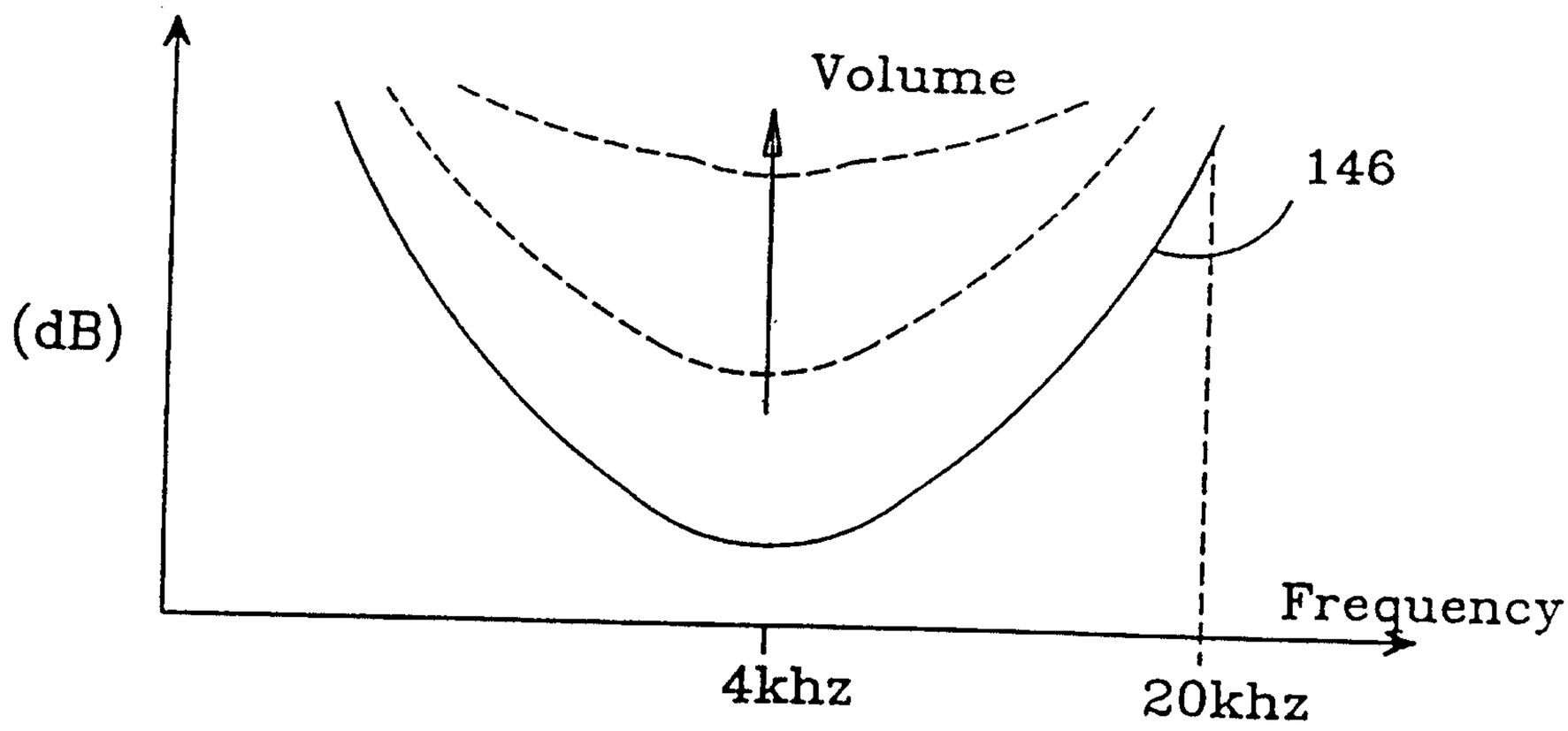


FIG. 15

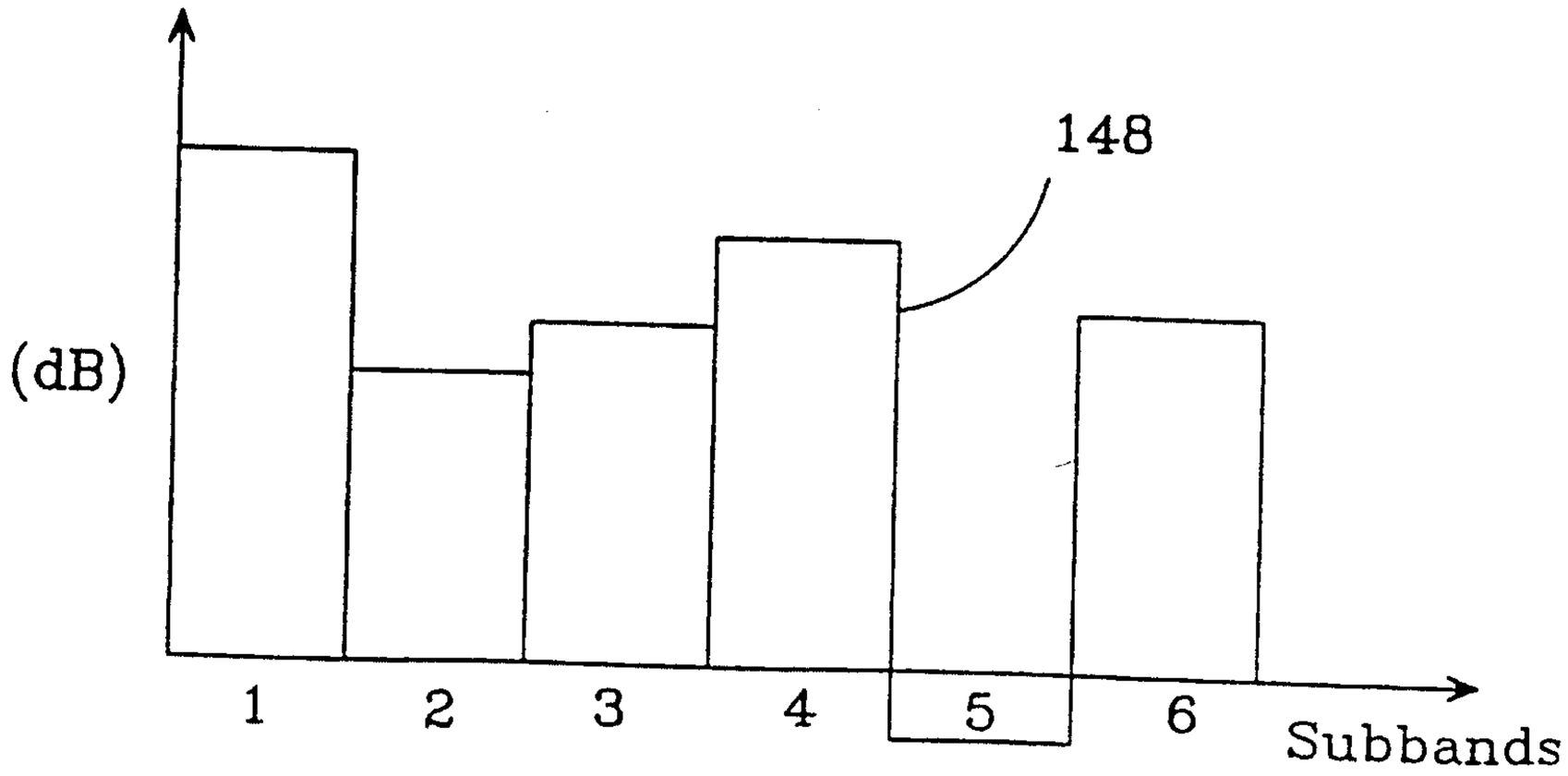


FIG. 16

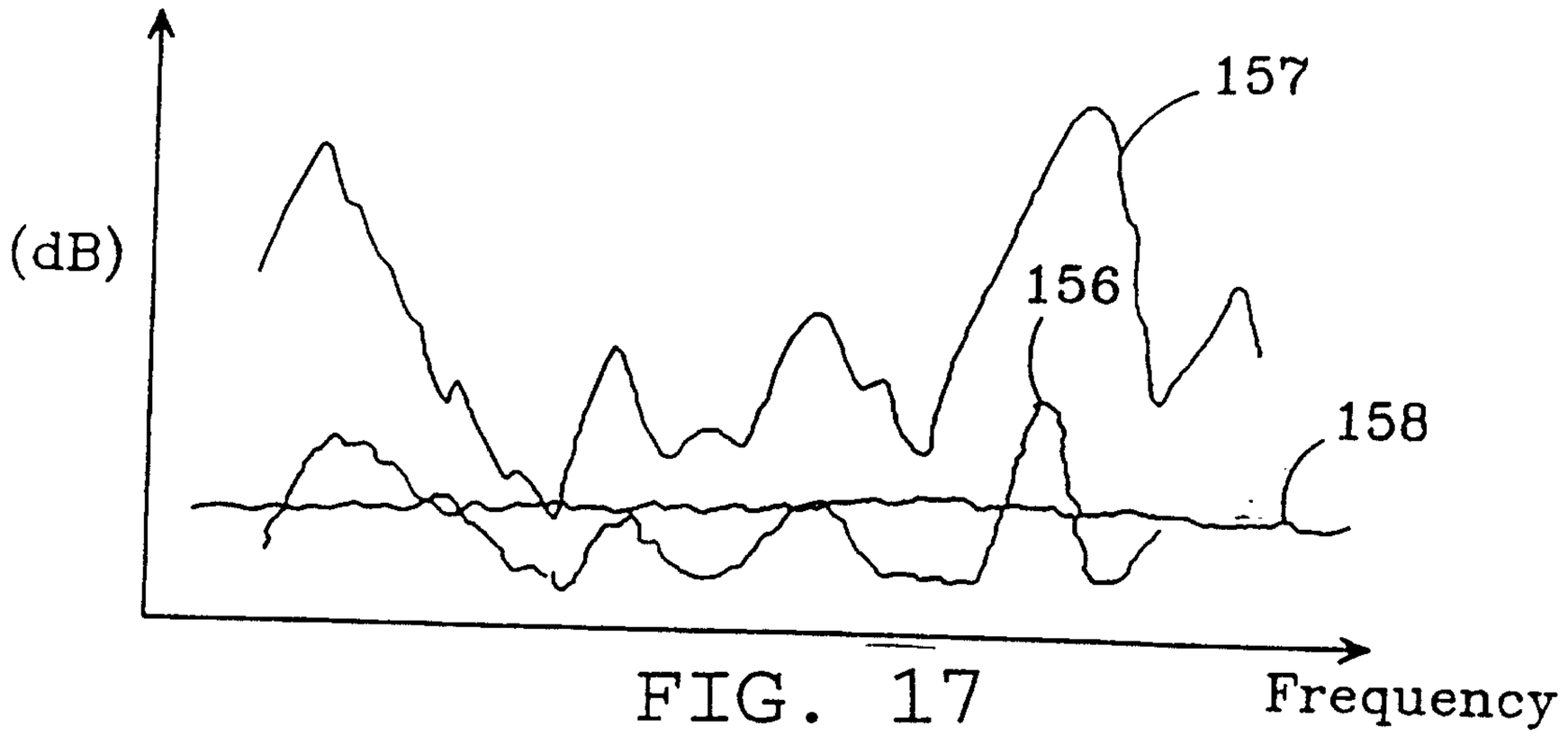


FIG. 17

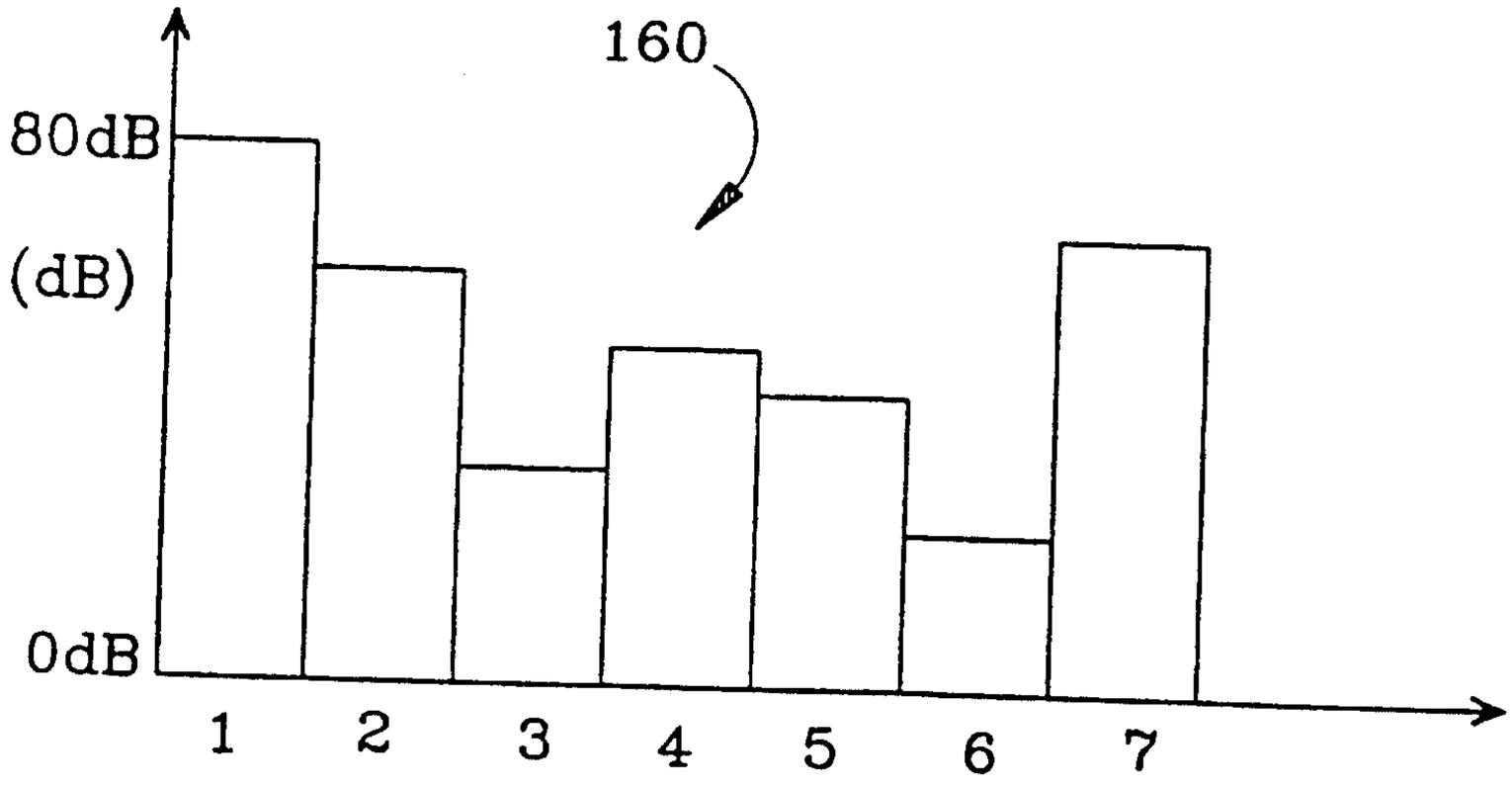


FIG. 18A

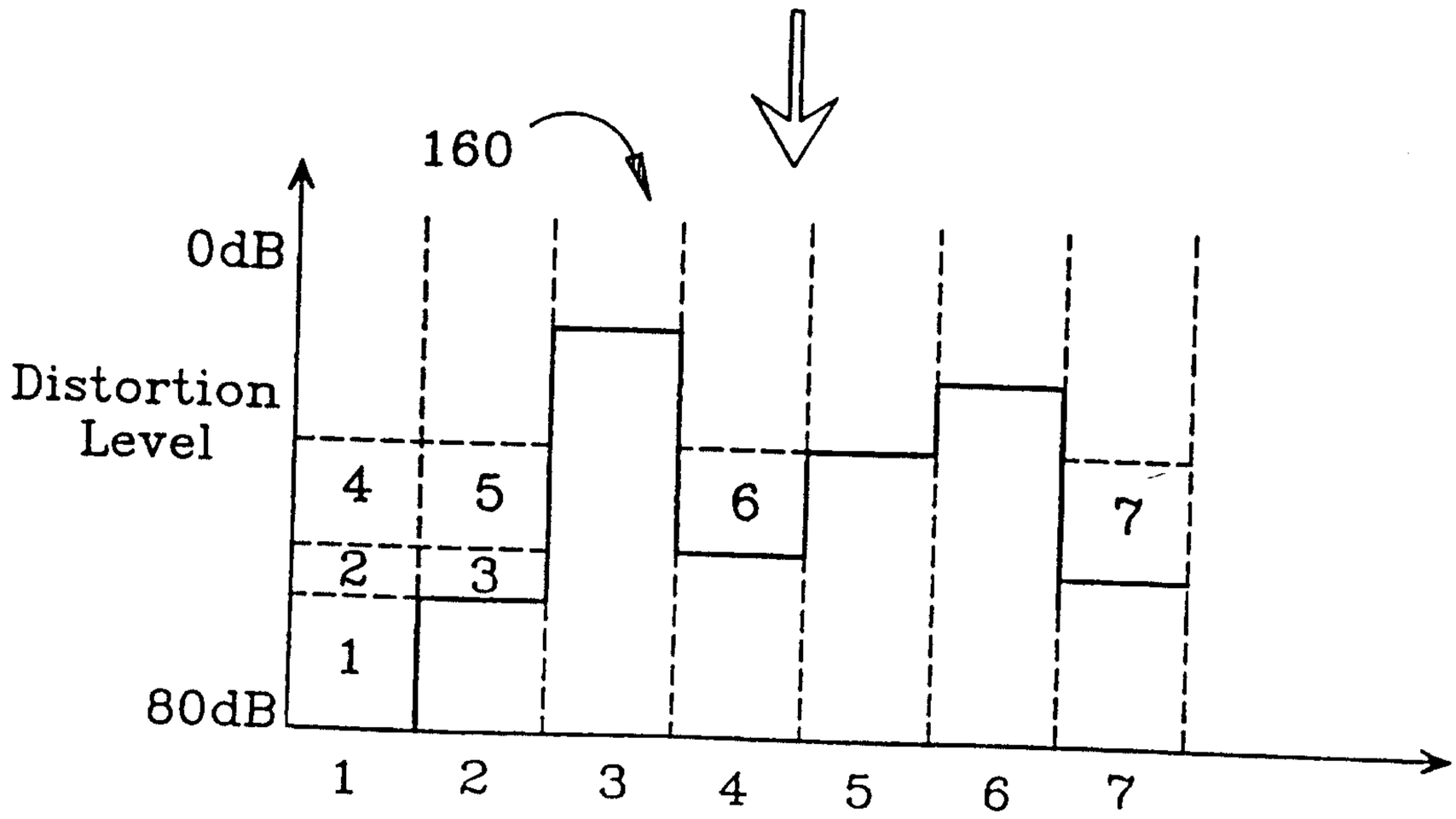
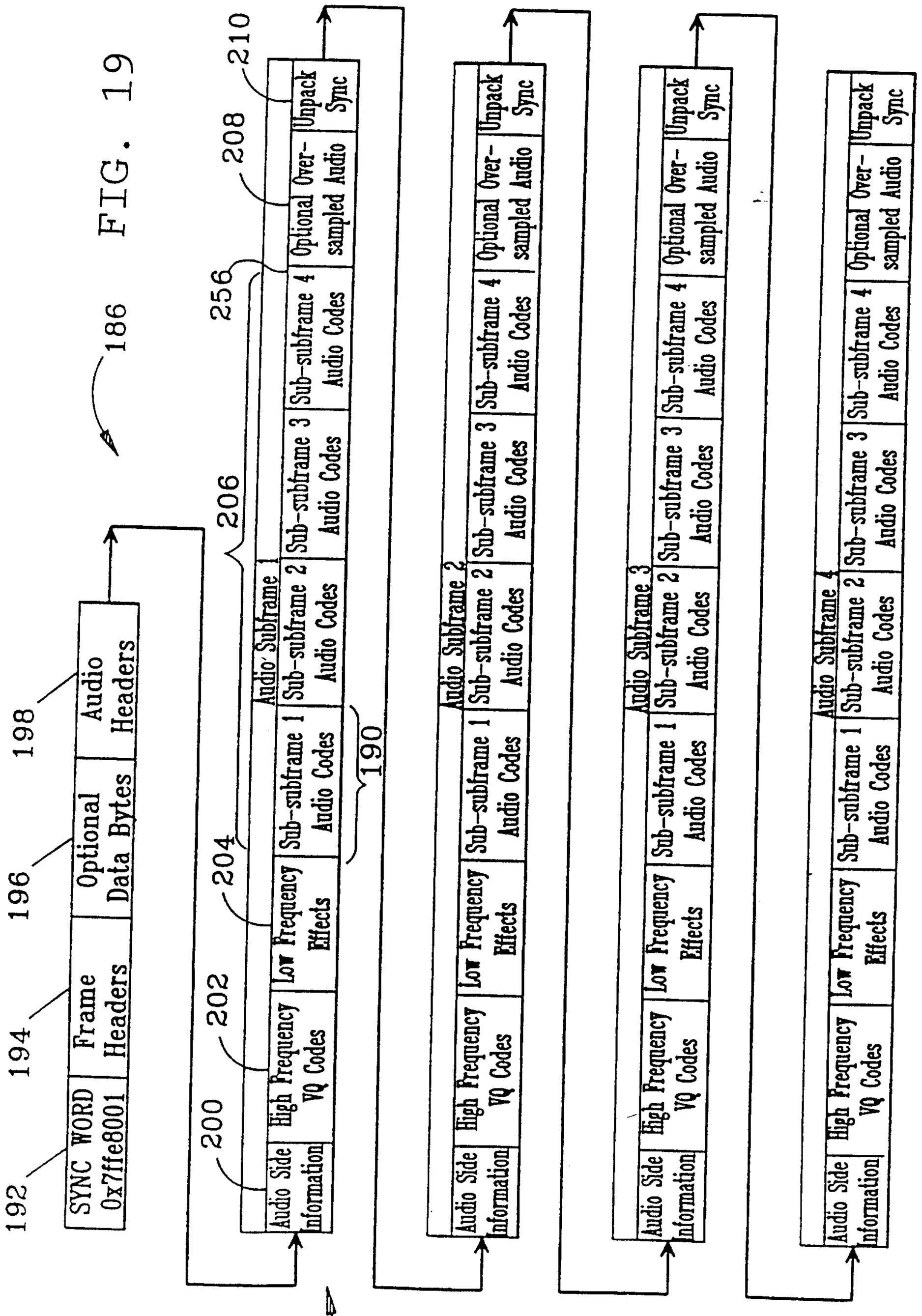


FIG.18B

FIG. 19


186
194
196
198
206
256
208
210
200
202
204
190
188
192
SYNC WORD 0x7ffe8001
Frame Headers
Optional Data Bytes
Audio Headers
Audio Side Information
High Frequency VQ Codes
Low Frequency Effects
Sub-subframe 1 Audio Codes
Sub-subframe 2 Audio Codes
Sub-subframe 3 Audio Codes
Sub-subframe 4 Audio Codes
Optional Over-sampled Audio
Unpack Sync
Audio Subframe 1
Audio Subframe 2
Audio Subframe 3
Audio Subframe 4
FRAME END

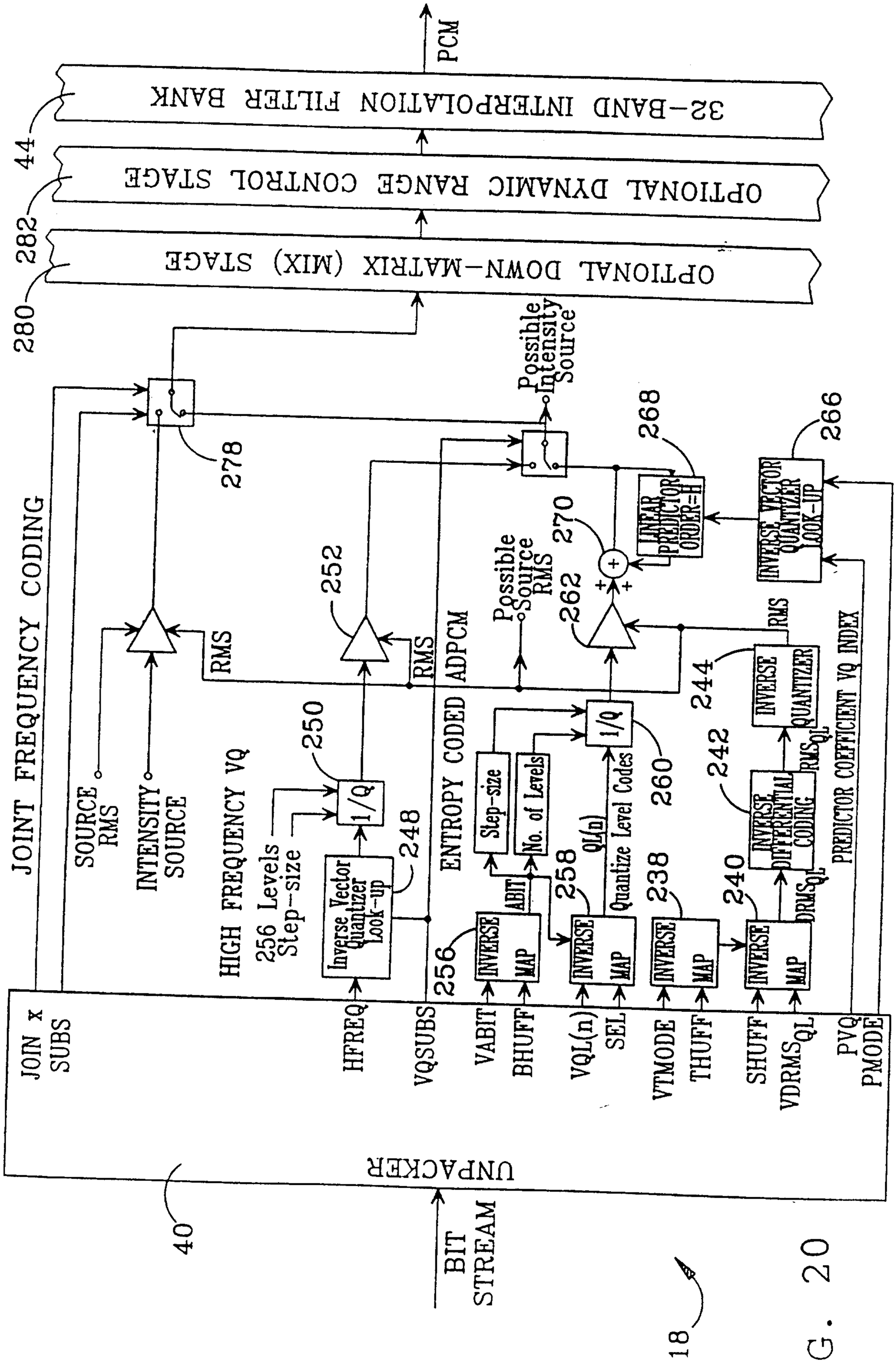

UNPACKER
BIT STREAM
40
18
JOIN x SUBS
HFREQ
VQSUBS
VABIT
BHUFF
VQL(n)
SEL
VTMODE
THUFF
SHUFF
VDRMS$_{QL}$
PVQ
PMODE
JOINT FREQUENCY CODING
SOURCE RMS
INTENSITY SOURCE
HIGH FREQUENCY VQ
256 Levels Step-size
250
Inverse Vector Quantizer Look-up
1/Q
248
256
ENTROPY CODED ADPCM
Step-size
No. of Levels
ABIT
258
INVERSE MAP
QL(n)
Quantize Level Codes
1/Q
260
238
240
242
244
INVERSE DIFFERENTIAL CODING
INVERSE QUANTIZER
RMS$_{QL}$
DRMS$_{QL}$
RMS
Possible Source RMS
262
270
+
268
LINEAR PREDICTOR ORDER=H
INVERSE VECTOR QUANTIZER LOOK-UP
266
PREDICTOR COEFFICIENT VQ INDEX
Possible Intensity Source
278
280
OPTIONAL DOWN-MATRIX (MIX) STAGE
282
OPTIONAL DYNAMIC RANGE CONTROL STAGE
44
32-BAND INTERPOLATION FILTER BANK
PCM

FIG. 20

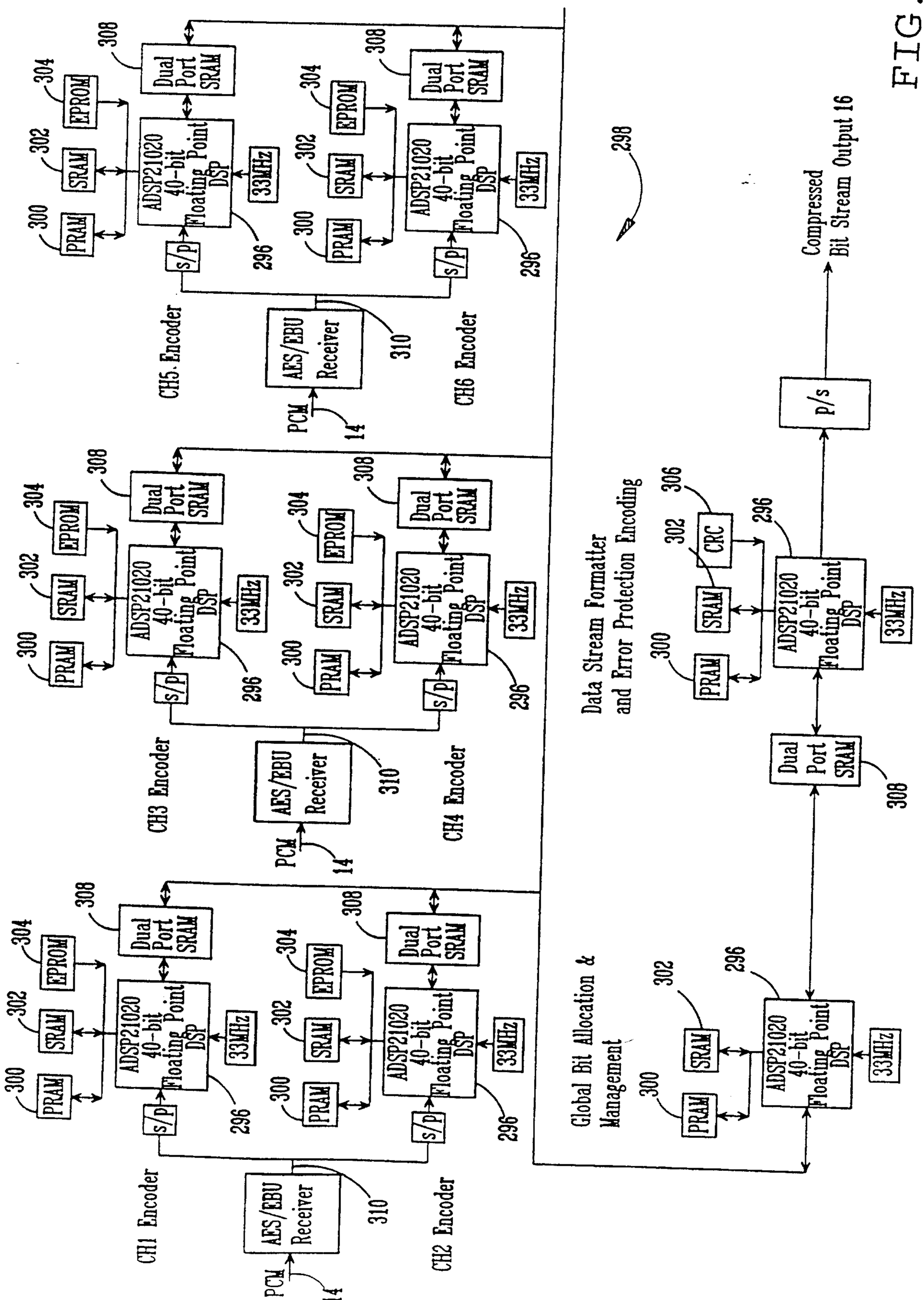

CH5 Encoder
CH6 Encoder
CH3 Encoder
CH4 Encoder
CH1 Encoder
CH2 Encoder
AES/EBU Receiver
PCM
14
310
296
298
ADSP21020 40-bit Floating Point DSP
33MHz
Dual Port SRAM
EPROM
SRAM
PRAM
S/P
P/S
CRC
Compressed Bit Stream Output 16
Data Stream Formatter and Error Protection Encoding
Global Bit Allocation & Management
300
302
304
306
308

FIG. 21

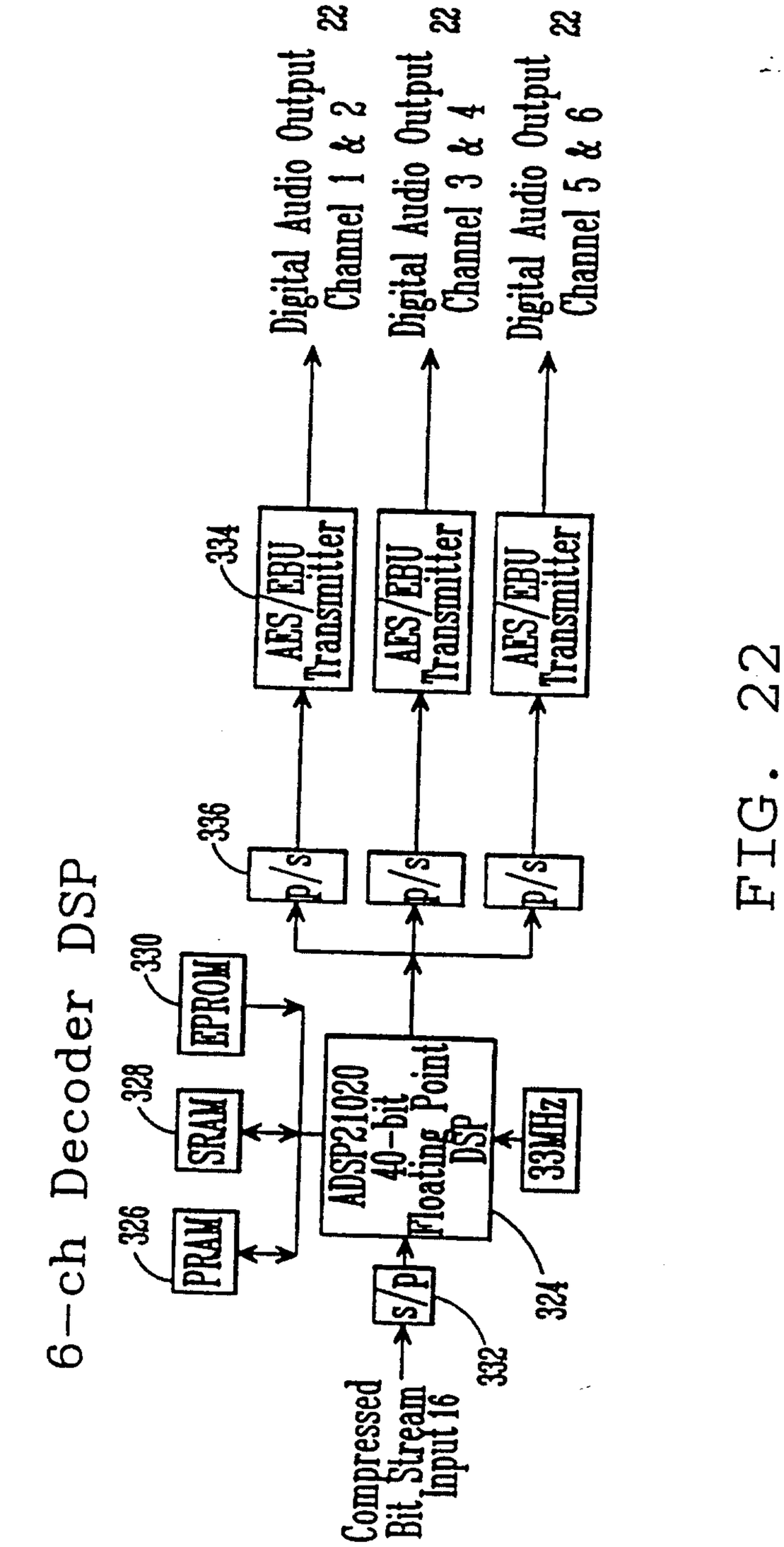
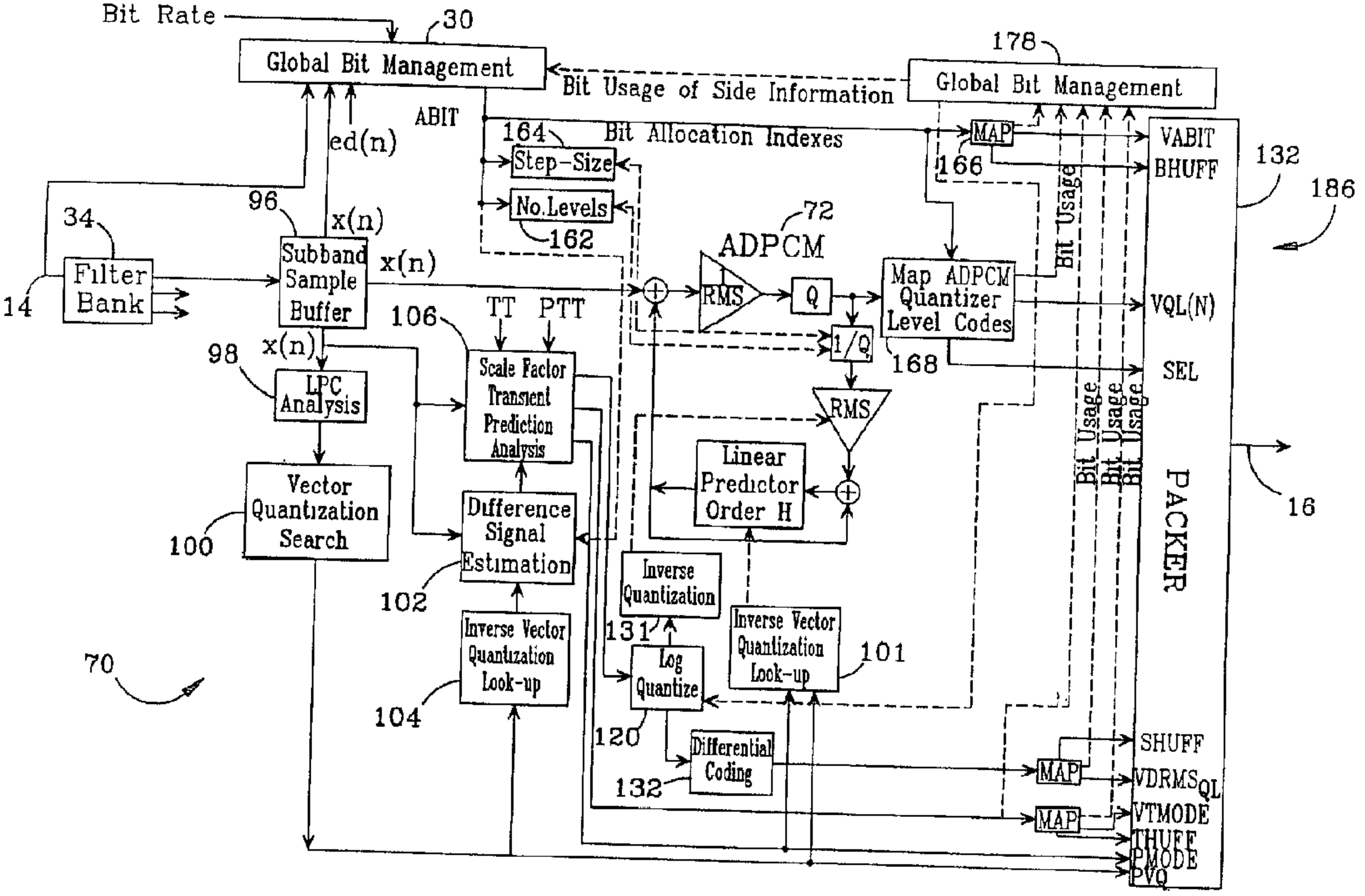

6-ch Decoder DSP
Compressed Bit Stream Input 16
332
S/P
324
ADSP21020 40-bit Floating Point DSP
33MHz
326
PRAM
328
SRAM
330
EPROM
336
P/S
P/S
P/S
334
AES/EBU Transmitter
AES/EBU Transmitter
AES/EBU Transmitter
Digital Audio Output Channel 1 & 2
Digital Audio Output Channel 3 & 4
Digital Audio Output Channel 5 & 6
22
22
22

FIG.


Bit Rate
30
Global Bit Management
178
Global Bit Management
Bit Usage of Side Information
ABIT
ed(n)
164
Bit Allocation Indexes
Step-Size
No.Levels
162
MAP
166
VABIT
BHUFF
132
186
Bit Usage
96
x(n)
34
Filter Bank
14
Subband Sample Buffer
x(n)
72
ADPCM
RMS
Q
1/Q
Map ADPCM Quantizer Level Codes
168
VQL(N)
SEL
TT
PTT
106
98
x(n)
LPC Analysis
Scale Factor Transient Prediction Analysis
RMS
Linear Predictor Order H
Vector Quantization Search
100
Difference Signal Estimation
102
Inverse Quantization
131
Inverse Vector Quantization Look-up
101
Inverse Vector Quantization Look-up
104
Log Quantize
120
Differential Coding
132
70
MAP
MAP
SHUFF
VDRMS$_{QL}$
VTMODE
THUFF
PMODE
PVQ
PACKER
16