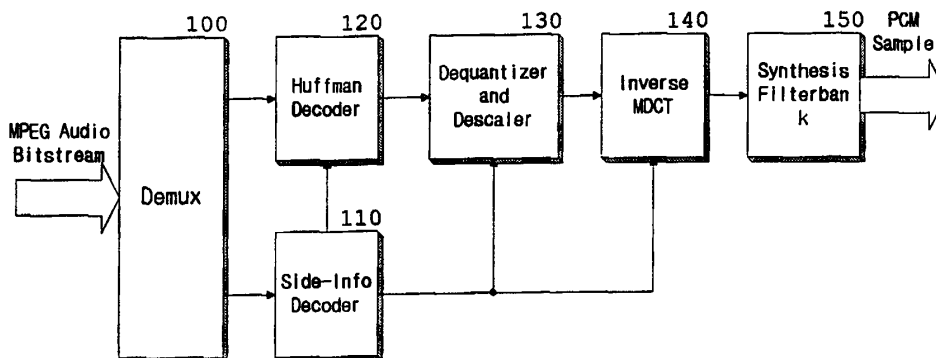




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : H03M 7/40, H04N 7/50	A1	(11) International Publication Number: WO 00/51243 (43) International Publication Date: 31 August 2000 (31.08.00)
(21) International Application Number: PCT/KR99/00764 (22) International Filing Date: 11 December 1999 (11.12.99) (30) Priority Data: 1999/6157 24 February 1999 (24.02.99) KR (71)(72) Applicant and Inventor: YOU, Soo, Geun [KR/KR]; Jamwon Hansin Apt. 1-1103, 56-3, Jamwon-dong, Seocho-gu, Seoul 137-030 (KR). (72) Inventor; and (75) Inventor/Applicant (for US only): PARK, Jung, Jae [KR/KR]; 6516, Taepyeong 1-dong, Sujeong-gu, Seongnam, Kyunggi-do 461-191 (KR). (74) Agent: PARK, Lae, Bong; 4F TLBS B/D, 464-1, Kunja-dong, Kwangjin-gu, Seoul 143-150 (KR).		(81) Designated States: AU, BR, CA, CN, DE, ES, GB, IN, JP, RU, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

(54) Title: A BACKWARD DECODING METHOD OF DIGITAL AUDIO DATA



(57) Abstract

This invention provides a method of backward decoding compressed digital audio data into an analog audio data reversed in time. The method according to this invention comprises the steps of locating a header of a last frame of the compressed digital audio data; dequantizing a plurality of data blocks constructing the frame based on information contained in the located header; extracting time signals of each frequency subband from the dequantized data blocks, reducing discontinuities between the dequantized data blocks; and synthesizing the extracted time signals of all subbands backward into real audio signal reversed in time. Therefore, this invention enables to record the decoded analog signal on both tracks on a magnetic tape simultaneously while the magnetic tape travels in one direction with little increase of computation load and memory size, resulting in a high speed recording.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

DESCRIPTION

A BACKWARD DECODING METHOD OF DIGITAL AUDIO DATA

1. Technical Field

The present invention relates to a method of decoding
5 compressed digital audio data backward, more particularly,
to a method of backward decoding an MPEG (Moving Picture
Experts Group) encoded audio data into analog audio
signal with little increase of computation load and
memory size.

10 2. Background Art

Digital audio signal is in general more robust to noise
than analog signal and thus the quality is not subject to
degradation during copy or transmission over network. The
digital audio signals are, moreover, transmitted more
15 rapidly and stored in storage media of less capacity due
to effective compression methods recently developed.

Many compression methods have been proposed to
effectively encode audio signals into digital data. MPEG
(Moving Picture Experts Group) audio coding schemes have
20 been used for the standard in this area. The MPEG audio
standards that are standardized as ISO (International
Standardization Organization) - MPEG audio layer-1,

layer-2, and layer-3 were devised to encode high-quality stereo audio signals with little or no perceptible loss of quality. They have been widely adopted in digital music broadcasting area and in addition have been used
5 with MPEG video standards to encode multimedia data. In addition to MPEG-1, standard specifications for digital environments have been proposed; MPEG-2 includes standards on compression of multimedia data. Standards for object oriented multimedia communication are included
10 in MPEG-4, which is in progress.

MPEG-1 consists of five coding standards for compressing and storing moving picture and audio signals in digital storage media. MPEG audio standard includes three audio coding methods: layer-1, layer-2, and layer-3.
15 MPEG audio layer-3 (hereinafter referred to as "MP3") algorithm includes a much more refined approach than in layer-1 and layer-2 to achieve higher compression ratio and sound quality, which will be described briefly below.

MPEG audio layer-1, 2, 3 compress audio data using
20 perceptual coding techniques which address perception of sound waves of the human auditory system. To be specific, they take an advantage of the human auditory system's inability to hear quantization noise under conditions of auditory masking. The "masking" is a perceptual property

of the human ear which occurs whenever the presence of a strong audio signal makes a temporal or spectral neighborhood of weaker audio signals imperceptible. Let us suppose that a pianist plays the piano in front of
5 audience. When the pianist does not touch keyboard, the audience can hear trailing sounds, but is no longer able to hear the trailing sounds at the instant of touching the keyboard. This is because, in presence of masking sounds, or the newly generated sounds, the trailing
10 sounds which fall inside frequency bands centering the masking sound, so-called critical bands, and loudness of which is lower than a masking threshold are not audible. This phenomenon is called spectral masking effect. The masking ability of a given signal component depends on
15 its frequency position and its loudness. The masking threshold is low in the sensitive frequency bands of the human ear, i.e., 2KHz to 5KHz, but high in other frequency bands.

There is the temporal masking phenomenon in the human
20 auditory system. That is, after hearing a loud sound, it takes a period of time for us to be able to hear a new sound that is not louder than the sound. For instance, it requires 5 milliseconds for us to be able to hear a new sound of 40 dB after hearing a sound of 60 dB during 5

milliseconds. The temporal delay time also depends on frequency band.

Based on a psychoacoustic model of the human ear, the MP3 works by dividing the audio signal into frequency
5 subbands that approximate critical bands, then quantizing each subband according to the audibility of quantization noise within that band, so that the quantization noise is inaudible due to the spectral and temporal masking.

The MP3 encoding process is described below in detail,
10 step by step, with reference to FIGS. 1 and 2.

(1). Subband coding and MDCT (Modified Discrete Cosine Transform)

In the MP3 encoder, PCM format audio signal is, first, windowed and converted into spectral subband components
15 via a filter bank 10, shown in FIG. 1, which consists of 32 equally spaced bandpass filters. The filtered bandpass output signals are critically sub-sampled at the rate of 1/32 of the sampling rate and then encoded.

Polyphase filterbank is, in general, used to cancel the
20 aliasing of adjacent overlapping bands that occurs otherwise because of the low sampling rate at the sub-sampling step. As another method, MDCT (Modified Discrete Cosine Transform) unit 20 and aliasing reduction unit 30 are adopted to cancel the aliasing, thereby preventing

deterioration of the quality.

Because MDCT is essentially critically sampled DCT (Discrete Cosine Transform), the input PCM audio signal can be reconstructed perfectly in the absence of
5 quantization errors. Discontinuities between transformed blocks occur since quantization is carried out.

For each subband, the number of quantization bits is allocated by taking into account the masking effect by neighboring subbands. That is, quantization and bit
10 allocation is performed to keep the quantization noise in all critical bands below the masking threshold.

(2). Scaling

Samples in each of the 32 subbands are normalized by a scale factor such that the sample of the largest
15 magnitude is unity, and the scale factor is encoded for use in the decoder. With the scaling process, the amplitude of signal is compressed, therefore, the quantization noise is reduced and become inaudible due to the psychoacoustic phenomenon.

20 (3). Huffman Coding

Variable-length Huffman codes are used to get better data compression rate of the quantized samples. The Huffman coding is called entropy coding whereby redundancy reduction is carried out based on statistical

property of the digital data. The principle behind the Huffman coding is that codewords of small length are assigned to symbols having higher probability, while large-length codewords are assigned to symbols with lower probability. In effect, the average length of encoded data are reduced as small as possible.

Let us consider an example for illustration. The quantized samples are 00, 01, 10, and 11. Their probabilities are 0.6, 0.2, 0.1, and 0.1, respectively.

10 In case of using codewords of constant length, say, 2 bits, the average length of a codeword is 2 bits without calculation of $(2 \times 0.6 + 2 \times 0.2 + 2 \times 0.1 + 2 \times 0.1) / 4 = 2$ bits. However, if variable-length codewords are used, i.e., 1 bit is assigned to 00 with the highest

15 probability, 2 bits for 01 with the second highest probability, and 3 bits for 10 and 11, the average length of the codeword leads to 1.6 bits ($= (1 \times 0.6 + 2 \times 0.2 + 3 \times 0.1 + 3 \times 0.1) / 4$).

In addition, in order to achieve high compression rate,

20 MP3 adopts bit reservoir buffering technique whereby unused bits in the frames in which the size of coded data are relatively small are used when the encoder needs more bits than the average number of bits to code a frame. After being processed by the above processes, the audio

signal is formatted into a bitstream. FIG. 3 shows the arrangement of the various fields in a frame of an MP3 encoded bitstream.

Without data reduction, digital audio signals typically consist of 16 bit samples recorded at several sampling rates than twice the actual audio bandwidth, (e.g., 32KHz, 44.1KHz, and 48KHz). In case of two channels stereo audio signals at a sampling rate of 44.1KHz with 16 bits per sample, the bit rate is $16 \times 44100 \times 2 = 1411200$, or about 1.4 10 Mbps. By using MP3 audio coding, the original sound data can be encoded at the bit rate of 128 to 256 Kbps. That is, 1.5 to 3 bits are, on the average, needed for sampling instead of 16 bits, and therefore the MP3 enables to shrink down the original sound data from a CD- 15 DA by a factor of about 12 without loss of the sound quality.

Despite its advantages, digital audio recorders and players are in infancy for several reasons and analog audio recorders and players have still been the majority 20 in the market. Accordingly, it would be attractive in terms of commercial products if it is possible that digital audio signals are recorded on analog signal storage media like magnetic tapes because users can enjoy digital audio without buying new digital audio recorders

and players.

Digital audio data are first decoded and then recorded on either track on a magnetic tape on which a forward track and a backward track are provided, That is, when
5 the tape travels in the forward (backward) direction, the audio signals are recorded on the forward (backward) track. After completion of recording the audio signals on the forward track, the tape begins to travel in the backward and the audio signals are recorded thereon. As a
10 result, it needs the time for two times tape travels to record the digital audio signals on a magnetic tape.

For fast recording, it is possible to encode analog audio signals which were backward-reproduced and to decode and record the encoded signal on the tape during
15 only one tape travel. However, the method has weak points of more storage spaces for the encoded backward-reproduced signals in addition to the encoded forward-reproduced signals, and imperfect reproduction of the audio signals due to MP3 encoding using masking
20 phenomenon since small amplitude preceding large amplitude in view of normal reproduction was suppressed while encoding audio signal reproduced backward.

3. Disclosure of Invention

It is a primary object of the present invention to

provide a method of backward decoding an MPEG digital audio data into an analog audio data which enables to record the decoded analog signal on analog signal storage media like magnetic tapes at a high speed with little
5 increase of computation load and memory size.

To achieve the object, the present invention provides a method of a method of backward decoding an MPEG audio data into an analog audio data, comprising the steps of locating a header of a last frame of the compressed
10 digital audio data; dequantizing a plurality of data blocks constructing the frame based on information contained in the located header; extracting time signals of each frequency subband from the dequantized data blocks, reducing discontinuities between the dequantized
15 data blocks; and synthesizing the extracted time signals of all subbands backward into real audio signal reversed in time.

According to the method of backward decoding MPEG audio data according to the present invention, when MPEG audio
20 data are asked to be recorded on a magnetic tape at a high speed, the MPEG audio data can be decoded and recorded on both of the two tracks on the magnetic tape simultaneously while the tape travels in one direction. Therefore, the backward decoding method according to

present invention enables fast recording of MPEG audio data on both of tracks on the magnetic tape.

4. Brief Description of Drawings

The accompanying drawings, which are included to
5 provide a further understanding of the invention,
illustrate the preferred embodiment of this invention,
and together with the description, serve to explain the
principles of the present invention.

In the drawings:

10 FIGS. 1 and 2 are block diagrams showing an MPEG audio
encoder;

FIG. 3 shows the arrangement of the various bit fields
in a frame of MPEG audio data;

FIG. 4 is a block diagram showing an MPEG audio
15 decoder;

FIG. 5 is a schematic diagram showing an illustration
of the bit reservoir within a fixed length frame
structure;

FIG. 6 is a schematic diagram illustrating the overlap
20 of inverse-modified-discrete-cosine-transformed blocks;

FIG. 7 is a flow graph showing a synthesis filterbank;

FIG. 8 is a flowchart showing an algorithm implementing
the synthesis filterbank of FIG. 7;

FIG. 9 is a block diagram of the flowchart of FIG. 8;

FIG. 10 is a flow graph showing a synthesis filterbank for backward decoding according to the present invention;

FIG. 11 is a flowchart showing an algorithm implementing the synthesis filterbank of FIG. 10; and

5 FIG. 12 is a block diagram of the flowchart of FIG. 11.

5. Modes for Carrying out the Invention

The preferred embodiments of the present invention will be described hereinafter in detail referring to the accompanying drawings.

10 FIG. 4 shows a block diagram of an MP3 audio decoder to which an embodiment of the present invention is applied, comprising a demultiplexer 100 for dividing an MP3 audio bitstream into several data of different types; a side-information decoder 110 for decoding side-information
15 contained the bitstream; a Huffman-decoder 120 for Huffman-decoding the divided audio data; a dequantizer 130 for obtaining actual frequency energies from the Huffman-decoded data; an inverse MDCT (IMDCT) unit 140 for applying IMDCT to the energies; and a synthesis
20 filterbank 150 for synthesizing subband values the into PCM samples.

With reference to the MP3 audio decoder of FIG. 4, the method of backward decoding MP3 encoded audio data are described below step by step.

(1). Identifying Frame Header

The first step in the backward decoding process of an MP3 bitstream is to find where decoding is started in the bitstream. In MPEG audio, frames are independent of each other, and consequently the first step is to locate a frame header in the bitstream, requiring knowing the frame length. All MPEG bit streams are generally divided in separate chunks of bits called frames. There is a fixed number of frames per second for each MPEG format, which means that for a given bit rate and sampling frequency, each input frame has a fixed length and produces a fixed number of output samples.

In order to obtain actual frame length, it is required to locate a frame header in the bitstream and to get the required information from it, because the frame length depends on the bit rate and sampling frequency. Locating header information is done by searching for a synchronization bit-pattern marked within the header. However, it happens that locating header information fails because some audio data may contain the same bit pattern as the synchronization bit-pattern.

To alleviate this problem, on the assumption that neither bit rate nor sampling frequency does not change in an MP3 audio clip, the demultiplexer 100 analyzes the

first header in the stream and obtains the length of the frame having no padding bit based on information in the first header. By using the frame length, the header of the last frame is located while traveling the MP3 audio clip from the end.

If padding bit is added to a frame, the frame length is increased by 1 byte. That is, the frame length may change from frame to frame due to the padding bit. Because it is uncertain that the last frame have padding bit, searching for the header of the last frame needs to examine whether the last frame header is away from the end of the clip by the frame length or one more byte away.

(2). Obtaining Side-information

After the frame header is found, the demultiplexer divides the input MP3 audio bitstream into side-information containing how the frame was encoded, scale factor specifying gain of each frequency band, and Huffman-coded data. The side-information decoder decodes the side-information so that the decoder knows what to do with the data contained in the frame.

The number of bits required for MP3 encoding depends on acoustic characteristics of samples to be encoded with equal quality of sound. The coded data do not necessarily fit into a fixed length frame in the code bitstream.

Based on this, MP3 uses bit reservoir technique whereby bit rate may be borrowed from previous frames in order to provide more bits to demanding parts of the input signal. To be specific, the encoder donates bits to a reservoir

5 when it needs less than the average number of bits to code a frame. Later, when the encoder needs more than the average number of bits to code a frame, it borrows bits from the reservoir. The encoder can only borrow bits donated from past frames with limits. It cannot borrow

10 from future frames. On the decoder's side, the current frame being decoded may include audio data belonging to the frames that will be presented subsequently. The starting byte of the audio data for the current frame is limited to 511 bytes away from that frame.

15 A 9-bit pointer is included in each frame's side-information that points to the location of the starting byte of the audio data for that frame, as shown in FIG. 5.

That is, the audio data for the current frame being decoded, i.e., scale factor and Huffman-coded data may be

20 included in data region of the previous frames, which are within 511 bytes distance from that frame. When MP3 audio data are forwardly decoded, if it is determined that data belonging to the current frame contains data for the subsequent frames, they are kept until the subsequent

frames are decoded. On the other hand, in order to backward decoding MP3 audio data, when the current frame is decoded, it is checked whether or not the decoding current frame needs data contained in the precedent frame, 5 and if any, the data are obtained in such a manner that headers of the precedent frames and data belonging to the frames are identified.

(3). Huffman decoding

Once obtaining the audio data are completed, the 10 Huffman decoder 120 starts to Huffman-decode the audio data (including the data contained in the precedent frames) based on the side-information and Huffman trees which were constructed and used in the encoding process according to the data contents.

15 This step is the same as that of forward decoding. However, since a frame is encoded in two granules (granule 0 and granule 1) and data of granule 0 must be decoded in order to locate granule 1, two granules must be decoded to output granule 1 in the backward decoding 20 process whereas it is possible to decode the MP3 encoded data from granule 0 to granule 1 sequentially in the forward decoding, whereas data of two granules must be decoded at a time in the backward decoding process.

(4). Dequantizing and dscaling

When the Huffman-decoder 120 has decoded the audio data, they have to be dequantized by the dequantizer 130 and descaled using the scale factors into real spectral energy values. For example, if the Huffman-decoded value
5 is Y , then the real spectral energy value is obtained by multiplying $Y^{(4/3)}$ and the scale factors.

If the bitstream is a stereo signal, each channel can be transmitted separately in every frame, but transmission of the sum and the difference between the
10 two channels is often adopted to reduce redundancies therebetween. If the bitstream was encoded in this way, the decoder has to perform stereo-processing to recover the original two channels.

(5). IMDCT (inverse modified discrete cosine transform)

15 So far the signals have all been in the frequency domain, and to synthesize the output samples, a transform is applied that is the reverse of the time-to-frequency transform used in the encoder.

In MPEG layer-3, MDCT is done to get better frequency
20 resolution than in the other layers. MDCT are essentially critically sampled DCT, implying that if no quantizing had been done, the original signal would be reconstructed perfectly. However, because quantization is performed for each data block in the encoding process, discontinuities

between data blocks occur inevitably. The single data block is the unit block of output samples of the decoder and is corresponding to a granule in inverse MDCT.

- To avoid discontinuities between the granules, which
 5 would lead to perceptible noise and clicks, the inverse MDCT uses 50% overlap, i.e., every inverse-modified-discrete-cosine-transformed granules are overlapped with half of the previous transformed granules to smooth out any discontinuities.
- 10 To be specific, IMDCT produces 36 samples output in a manner that the second half 18 samples of the previous granule is added to the first half 18 samples of the current granule, as shown in FIG. 6. For the backward decoding, the order in which granule is added must be
 15 reversed, i.e., the second half 18 samples of the current granule is added to the first half 18 samples of the precedent granule. For the end frame which is to be decoded at first at the backward decoding process, second granule of that frame is added with zeros or just used
 20 without overlapping.

The IMDCT process in the forward decoding is expressed by the following equation.

$$x_i(n) = y_i(n) + y_{i-1}(n+18) \quad 0 \leq n < 18, \quad i=1, 2, \dots, 2N.$$

where $x_i(n)$ is a target sample output, $y_i(n)$ is inverse-

modified-discrete-cosine-transformed sample, i is the granule index, N is the total number of frames, and $y_0(n+18)$ are all zeros for $0 \leq n < 18$.

The above equation must be changed into the following
 5 equation for the IMDCT process in the backward decoding.

$$x_i(n) = y_i(n+18) + y_{i-1}(n) \quad 0 \leq n < 18, \quad i=2N, 2N-1 \dots, 1.$$

where $y_{2N+1}(n+18)$ are all zeros for $0 \leq n < 18$. The overlapping procedure is the same as that of the forward decoding and therefore computation and memory size needed
 10 are identical.

(6). Synthesis of Subband signals

Once the transformed blocks is overlapped after the IMDCT process, the final step to get the output audio samples is to synthesize 32 subband samples. The subband
 15 synthesis operation is to interpolate 32 subband samples into audio samples in the time domain.

A subband synthesis filter needs the delayed inputs of previous frames, but in case of the backward decoding, subband samples are presented to the synthesis filter in
 20 the reverse order to the forward decoding. Therefore, redesign of MPEG standard synthesis filterbank is required to perform the backward decoding operation. The MPEG standard synthesis filterbank for the forward decoding is described below in detail and then the

synthesis filterbank for the backward decoding according to the present invention is explained in detail.

FIG. 7 shows a flow graph of an MPEG standard synthesis filterbank for forward decoding, whereby 32 subband
 5 samples are synthesized into audio samples of a time-series in the similar way to frequency division multiplexing. To be specific, 32 subband samples or $x_r(mT_{s1})$'s, each of which is critically sampled at a sampling period of T_{s1} , are synthesized into an output
 10 samples or $s(nT_{s2})$ which is critically sampled signal at a sampling period of T_{s2} ($= T_{s1} / 32$).

Here, $x_r(mT_{s1})$ is the r -th subband sample and $x_r(nT_{s2})$ is 32 up-sampled from $x_r(mT_{s1})$ such that thirty-one zeros are inserted into the interval between $(m-1)T_{s1}$ and mT_{s1} for
 15 $x_r(mT_{s1})$ samples. This up-sampling generates 31 images of baseband centered at harmonics of the original sampling frequency, kf_{s1} ($k=1,2,...,31$). That is, sampling frequency is increased from f_{s1} ($= 1/T_{s1}$) to f_{s2} ($=1/T_{s2}$) for the original subband sample of $x_r(mT_{s1})$.

20 For each subband, $x_r(nT_{s2})$ is processed by band-pass filter $H_r(z)$ to pass the signal belonging to frequency-band allocated to each filter. The band-pass filter has 512 orders and is constructed by phase-shifting a prototype low-pass filter.

The flow graph of FIG. 7 is expressed by the equation (1).

$$\begin{aligned}
 S_t(nT_{s2}) &= \sum_{r=0}^{31} \sum_{k=0}^{511} x_r((32t+n-k)T_{s2}) \cdot H_r(kT_{s2}) \\
 &= \sum_{r=0}^{31} \sum_{k=0}^{511} x_r((32t+n-k)T_{s2}) \cdot h(kT_{s2}) \cdot N_r(k) \quad \dots\dots\dots (1) \\
 &= \sum_{r=0}^{31} \sum_{k=0}^{511} x_r((32t+n-k)T_{s2}) \cdot h(kT_{s2}) \cdot \cos\left(\frac{(2r+1)(k+16)\pi}{64}\right)
 \end{aligned}$$

where r is the subband index ranging from 0 to 31, n is the output sample index ranging from 0 to 31, and $S_t(nT_{s2})$ is the synthesized output sample at time t . That is, $S_t(nT_{s2})$ represents the synthesized output sample of 32 subband samples or $x_r(tT_{s1})$'s at time t .

The equation (1) implies the convolution of $x_r(kT_{s2})$ and $H_r(kT_{s2})$, which has 512 coefficients and is constructed by the product of the prototype low-pass filter $h(kT_{s2})$ and $N_r(k)$ that is used for phase-shift thereof.

Reduction of the number of computations, i.e., multiplies and adds is possible in equation (1). By utilizing the symmetry property of cosine terms and zeros that are filled in $x_r(kT_{s2})$ at the time of up-sampling, equation (1) leads to equation (2), hereinafter, sampling period in the following equations is omitted for convenience and is T_{s2} if not explicitly expressed.

$$S_t(n) = \sum_{i=0}^{15} h(n+32i) \cdot (-1)^{[i/2]} \cdot g_t(n+64i+32 \times (i\%2)) \dots\dots\dots (2)$$

$$= \sum_{i=0}^{15} d(n+32i) \cdot g_t(n+64i+32 \times (i\%2))$$

$$g_t(k+64i) = \sum_{r=0}^{31} x_r(32t-32i) \cdot \cos\left(\frac{(2r+1)(k+16)\pi}{64}\right) \dots\dots\dots (3)$$

where r is the subband index ranging from 0 to 31, n , i , and k are computation indices ($n=0,1,2 \dots 31$, $i=0,1,2 \dots 15$,
 5 $k=0,1,2, \dots, 63$), t represents the time when the subband sample is presented to the decoder. $\%$ is the modular operator and $[x]$ represents the largest integer that is not greater than x .

For each subband, one sample is presented and
 10 multiplied by $N_r(k)$, resulting in 64 samples. The 64 samples are stored in 1024 FIFO (First In First Out) buffer, samples have been stored therein being shifted by 64. 32 PCM output samples are obtained by multiplying samples in the 1024 FIFO buffer by coefficients of the
 15 time window.

The synthesis filterbank for backward decoding according to the present invention will be described below in detail with reference to the MPEG standard synthesis filterbank for the forward decoding.

20 It should be noted that for backward decoding, subband samples are presented to the decoder in the reverse order to their playback order. For example, given N samples for

each subband, while the forward decoder decodes the samples in the increasing order ($t = 0, 1, 2, \dots, N-1$), the samples have to be decoded in the decreasing order ($t=N-1, N-2, \dots, 0$) for backward decoding.

- 5 Because MPEG standard synthesis filterbank requires past samples for synthesizing PCM audio samples, if samples are presented in the reverse order to perform backward decoding, MPEG standard synthesis filterbank cannot use the previous samples. As a result, MPEG
- 10 standard synthesis filterbank must be modified to perform backward decoding. The structure thereof is explained below.

FIG. 10 depicts a flow graph showing the synthesis filterbank for backward decoding according to the present

15 invention, which is identical to the forward decoding synthesis filterbank except that $H_r(z)$ is replaced by $B_r(z)$. Note that $x_r(mT_{s1})$ is presented to the filterbank in the decreasing order, i.e., $m = N-1, N-2, \dots, 1, 0$.

Equation (1) is changed to equation (4) in accordance

20 with the reverse of the presentation order.

$$\begin{aligned}
 S_t(nT_{s2}) &= \sum_{r=0}^{31} \sum_{k=0}^{511} x_r(((32t-511)+n+k)T_{s2}) \cdot H_r(kT_{s2}) \\
 &= \sum_{r=0}^{31} \sum_{k=0}^{511} x_r((32(t-15)+(n-31)+k)T_{s2}) \cdot h(kT_{s2}) \cdot N_r(k) \\
 &= \sum_{r=0}^{31} \sum_{k=0}^{511} x_r((32(t-15)+(n-31)+k)T_{s2}) \cdot h(kT_{s2}) \cdot \cos\left(\frac{(2r+1)(k+16)\pi}{64}\right)
 \end{aligned} \tag{4}$$

Compared to equation (1), the band-pass filter of each subband is the same as that of equation (1), but sequence of subband samples that are multiplied by the band-pass filter coefficients is reversed, i.e., $x_r(((32t+n)-k)Ts)$ is replaced by $x_r(((32t+n)-511+k)Ts)$.

The above equation is partially optimized to reduce the number of computations, resulting in the following equations.

$$S_i(n) = \sum_{i=0}^{15} h(31-n+32i) \cdot (-1)^{[i/2]} \cdot g_i(31-n+64i+32 \times (i\%2)) \dots\dots\dots (5)$$

$$= \sum_{i=0}^{15} d(31-n+32i) \cdot g_i(31-n+64i+32 \times (i\%2))$$

$$10 \quad g_i(63-k+64i) = \sum_{r=0}^{31} x_r(32(t-15)+32i) \cdot \cos\left(\frac{(2r+1)(63-k+16)\pi}{64}\right) \dots\dots\dots (6)$$

where all indices are the same as those in equations (2) and (3).

If using $j = 31 - n$ and $m = 63 - k$, the equation (5) and (6) are changed to the following equations.

$$15 \quad S_i(j) = \sum_{i=0}^{15} d(j+32i) \cdot g_i(j+64i+32 \times (i\%2)) \dots\dots\dots (7)$$

$$g_i(m+64i) = \sum_{r=0}^{31} x_r(32(t-15)+32i) \cdot \cos\left(\frac{(2r+1)(m+16)\pi}{64}\right) \dots\dots\dots (8)$$

Note that equations (7) and (8) are the same as equations (2) and (3) except the index of input samples. The flow chart of an algorithm implementing the equations (7) and (8) and block diagram thereof are shown in FIG. 11 and FIG. 12, respectively.

It should be noted that the synthesis filter for

backward decoding is similar to the synthesis filter for forward decoding, and therefore computation and memory size that are needed to implement the synthesis filter are identical. Accordingly, the backward decoding can be
5 performed with the synthesis filter for forward decoding by reversing the direction in which the samples in the FIFO buffer are shifted as well as the order in which subband samples are summed.

Output samples are produced in the reverse order to
10 their playback order in units of 32 samples, but each of 32 samples is arranged in the playback order. Accordingly, the backward decoder outputs the 32 samples per a frame in the reverse order. This is repeated frame by frame until the first frame of the MPEG audio data. Note that
15 the synthesis filter for backward decoding can be used in MPEG audio layer-1, layer-2, and layer-3.

Meanwhile, when it is asked that MPEG audio data are recorded on a magnetic tape at a high speed, it is desirable that the decoded audio data are recorded on the
20 forward track on the magnetic tape, and at the same time another decoded audio data are recorded on the backward track while the magnetic tape travels in predetermined direction. To do this, MPEG audio data that will be recorded on the backward track on the magnetic tape must

be decoded backward in real time by using the foregoing embodiment according to the present invention. Besides the foregoing embodiment, it is possible to use the conventional forward decoding algorithm for simultaneous
5 record of MPEG audio data on both tracks on magnetic tape, which will be described below.

MPEG audio data that will be recorded on the backward track are first decoded into PCM samples and stored in a buffer, retrieved from the end in the backward order, and
10 converted into analog audio signals. This method is very simple, but it needs a large-sized buffer for temporarily storing the decoded audio data. Moreover, because the required buffer size depends on the length of MPEG audio clip being decoded, it is difficult to fix the maximum
15 size of the buffer in advance.

It also gives another method to apply forward decoding algorithm to an MPEG audio bitstream in units of a predetermined number of frames to record the decoded audio segment to the backward track in the reverse order.
20 For example, if an MPEG audio clip of N frames is to be decoded in units of M frames, blocks from the (N-M)-th frame to the N-th frame are decoded and then blocks from the (N-M+1)-th frame to the N-th frame are recorded in the reverse order. It should be noted that the (N-M)-th

frame is not recorded because the precedent frame, i.e., the (N-M-1)-th frame is needed to obtain its complete decoded samples. Decoding of the current requires only one precedent frame. For the reason, data blocks from the
5 N-th frame to the (N-M+1)-th frame are valid except the (N-M)-th frame.

And then, blocks from the (N-2M)-th frame to the (N-M)-th frame are decoded and then blocks from the (N-2M+1)-th frame to the (N-M)-th frame are recorded in the reverse
10 order. Note that the (N-M)-th frame is included again in the second decoding. At this time, the (N-M)-th frame is decoded perfectly because it is decoded with its precedent frame. The decoding-and-recording operation is repeated till all frames are decoded.

15 The first frame that is included in the block to be decoded last is just decoded and recorded because it has no precedent frame, as in the forward decoding.

Compared to the forgoing method, this method has advantages that small-sized buffer enough to store M
20 samples is sufficient and that the buffer size is fixed in advance.

fast recording of MPEG audio data on both of

The backward decoding algorithm needs more memory than the forward decoding algorithm, but the number of

computations thereof is the same as that of the forward decoding algorithm. The size of memory needed is two times larger than that of memory in the forward decoding because a frame must be Huffman-decoded at a time unlike
5 the forward decoding algorithm, where it is possible that two blocks consisting of a frame are Huffman-decoded sequentially. Thus, the memory size results in 1152×2 words. The backward decoding that is performed by applying forward decoding algorithm to every
10 predetermined number of frames requires a buffer in which forward decoded data are temporarily stored, but is easy to implement.

The foregoing is provided only for the purpose of illustration and explanation of the preferred embodiments
15 of the present invention, so changes, variations and modifications may be made without departing from the spirit and scope of the invention.

C L A I M S

1. A method of backward decoding compressed digital audio data consisting of a plurality of frames, comprising the steps of:

A. locating a header of a last frame of the compressed digital audio data;

B. dequantizing a plurality of data blocks constructing the frame based on information contained in the located header;

C. extracting time signals of each frequency subband from the dequantized data blocks, reducing discontinuities between the dequantized data blocks; and

D. synthesizing the extracted time signals of all subbands backward into real audio signal reversed in time.

2. A method according to claim 1, wherein said step A locates the header of the last frame based on information contained in the header of a first frame of the digital audio data.

3. A method according to claim 1, wherein said step A comprises the steps of:

A1. obtaining frame size from information contained in a header of a first frame of the compressed digital audio data;

A2. estimating location of the header of the last frame on the basis of the obtained frame size; and

A3. locating the header of the last frame around the estimated location depending on whether padding bit is present or absent.

4. A method according to claim 1, wherein said step B comprises the steps of:

B1. obtaining side-information from the located frame header;

B2. identifying locations of data blocks belonging to the frame based on the obtained side-information; and

B3. dequantizing all of the identified data blocks belonging to the frame simultaneously.

5. A method according to claim 4, wherein said step B3 comprises the steps of:

Huffman-decoding all of the identified data blocks belonging to the frame; and

dequantizing and descaling the Huffman-decoded data blocks.

6. A method according to claim 1, wherein said step C reduces discontinuities between inverse modified discrete cosine transformed data blocks of a frame by overlapping a second inverse-modified-discrete-cosine-transformed data block and a first inverse-modified-discrete-cosine-

transformed data block, the first data block being inputted just after the second data block.

7. A method according to claim 6, wherein the second data block of the frame is overlapped with a block filled with zeros.

8. A method of backward decoding compressed digital audio data consisting of a plurality of frames, comprising the steps of:

A. reading a header of a first frame of the compressed digital audio data;

B. obtaining information on the size of frames from information contained in the read header;

C. searching for a frame ahead of a just-decoded frame based on the obtained size information; and

D. decoding the frame discovered in said step C backward into real audio signal reversed in time.

9. A method according to claim 8, wherein said step D comprising the steps of:

D1. dequantizing a plurality of data blocks constructing the discovered frame;

D2. extracting time signals of each frequency subband from the dequantized data blocks, reducing discontinuities between the dequantized data blocks; and

D3. synthesizing the extracted time signals of all

frequency subbands into real audio data whose output sequence is reverse to normal playback.

10. A method of backward decoding compressed digital audio data consisting of a plurality of frames, comprising the steps of:

A. searching for a frame ahead of a just-decoded frame based on pre-obtained frame size information;

B. decoding the frame discovered in said step A backward into real audio signal reversed in time; and

C. repeating said steps A and B while the frame decoded in said step B is not a start one of the compressed digital audio data.

11. A method of backward decoding compressed digital audio data consisting of a plurality of frames, comprising the steps of:

A. picking up a predetermined number of frames from the end to the beginning of the compressed digital audio data, wherein the predetermined number is smaller than the number of total frames;

B. decoding forwardly the picked up frames; and

C. outputting the decoded data in the reverse to the decoding direction.

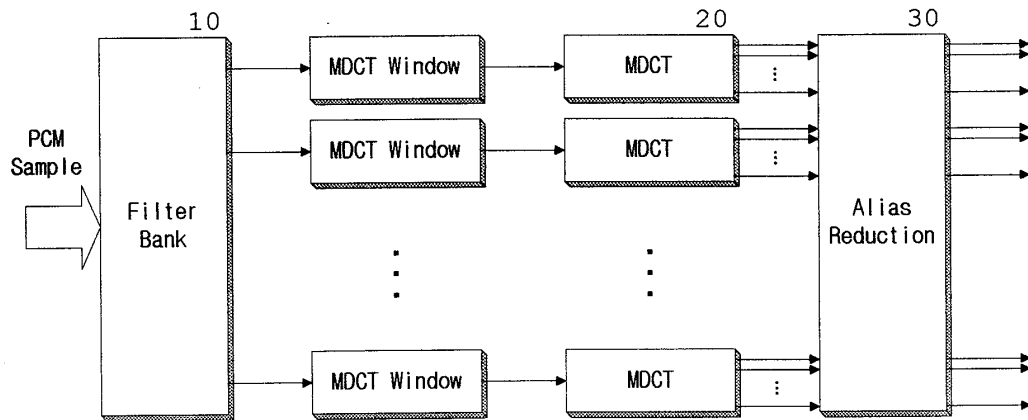
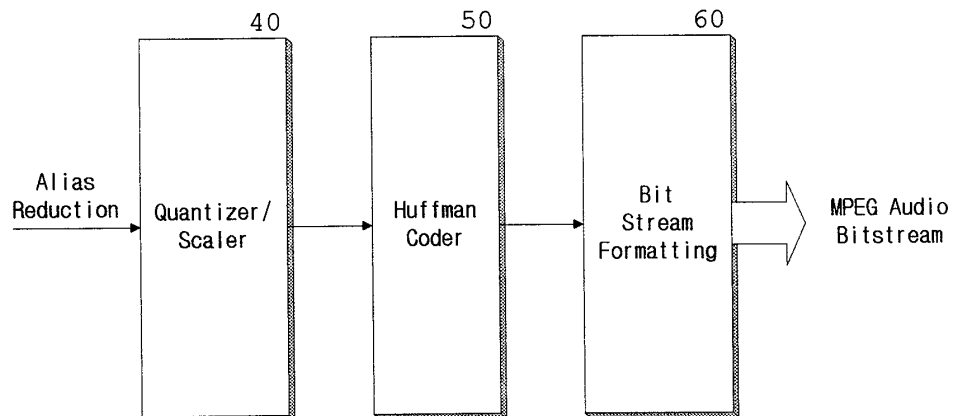
FIG. 1**FIG. 2**

FIG. 3

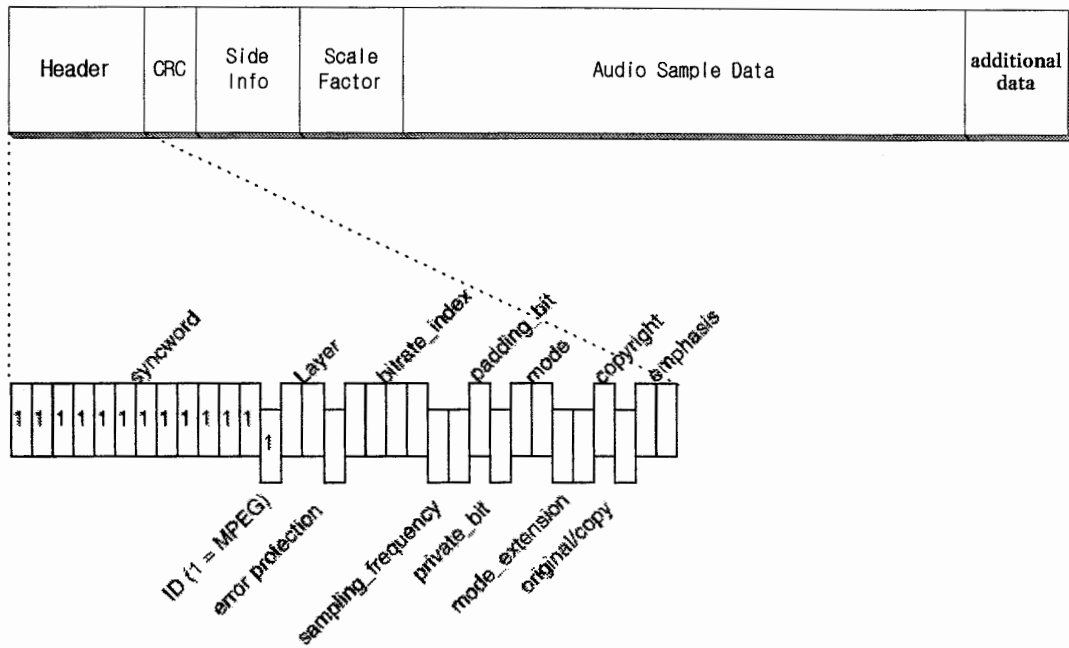
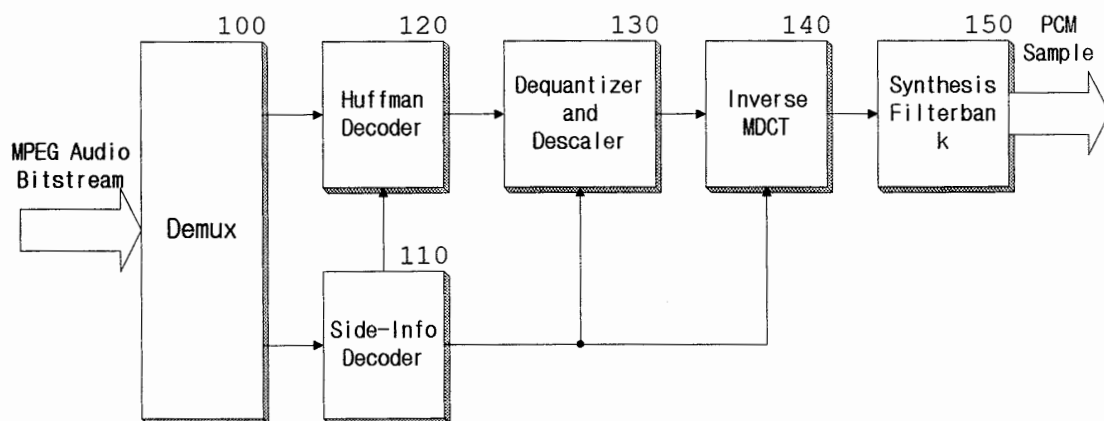


FIG. 4



3/8

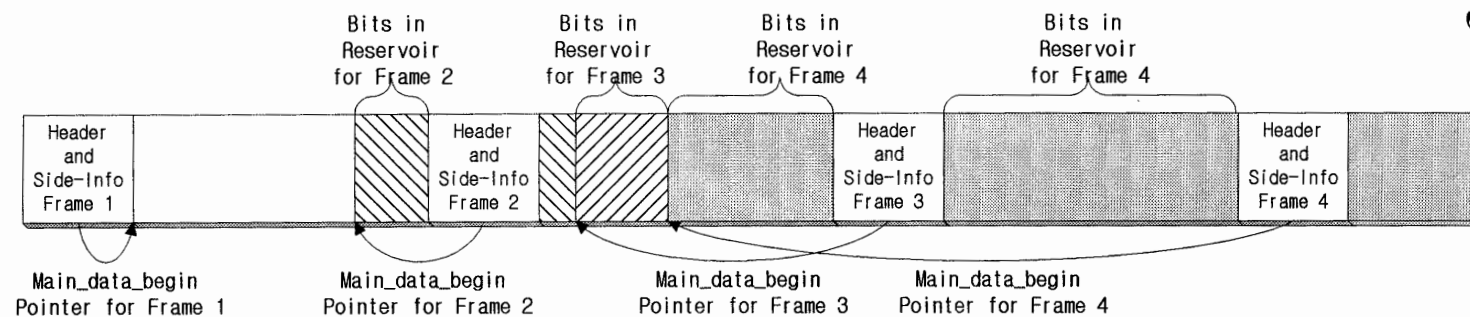


FIG. 5

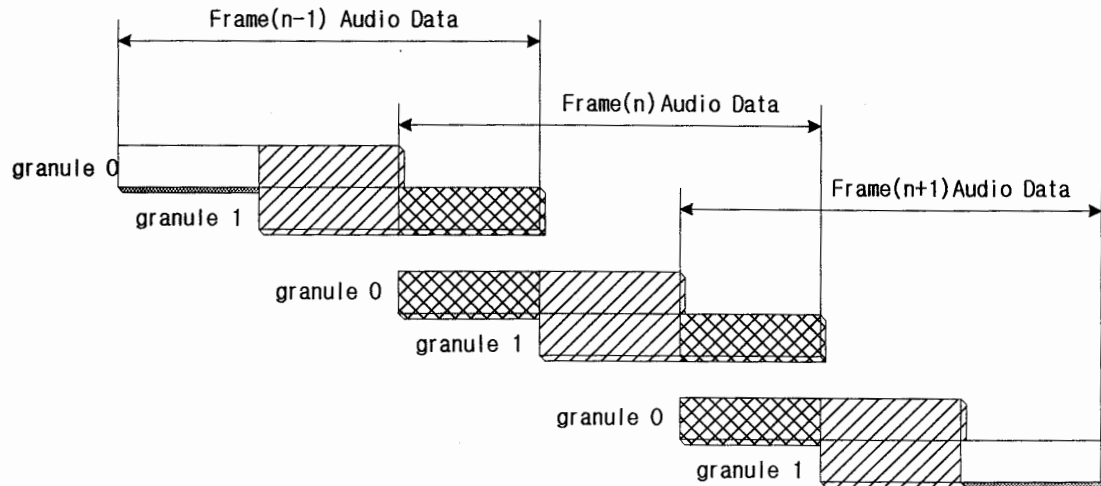
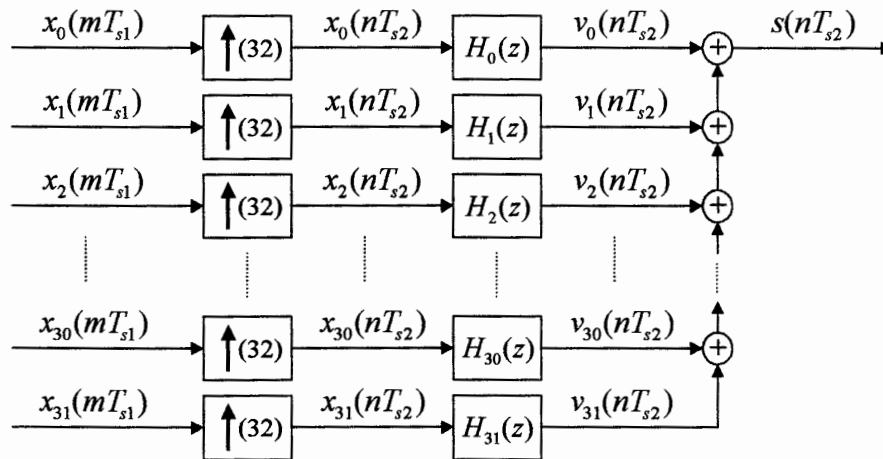
FIG. 6**FIG. 7**

FIG. 8

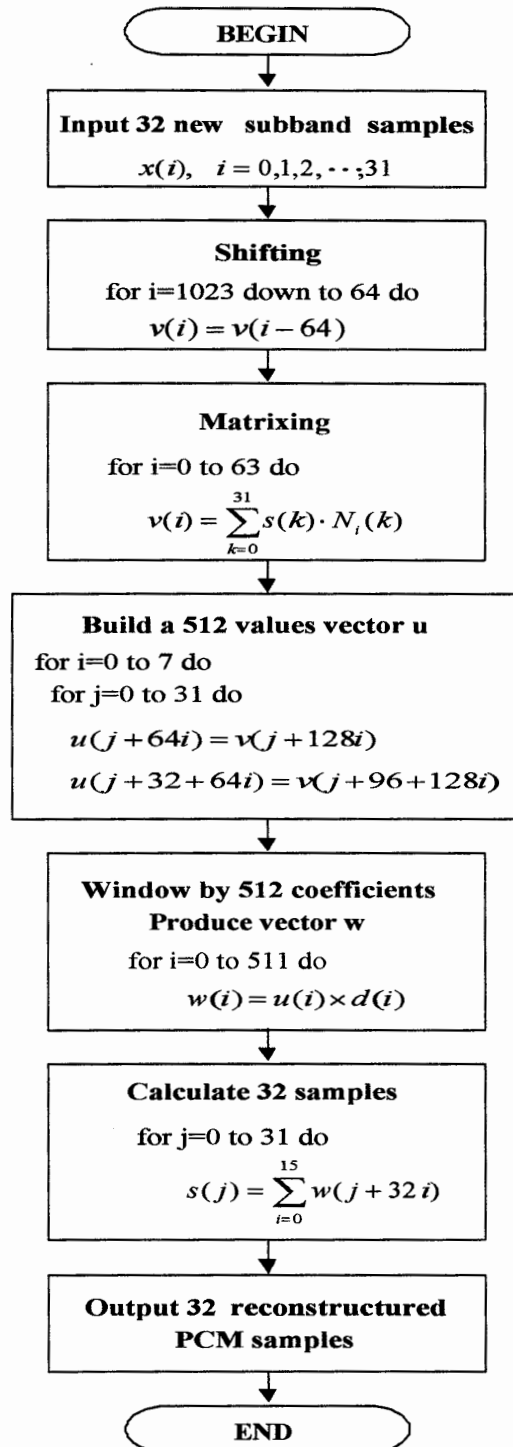


FIG. 9

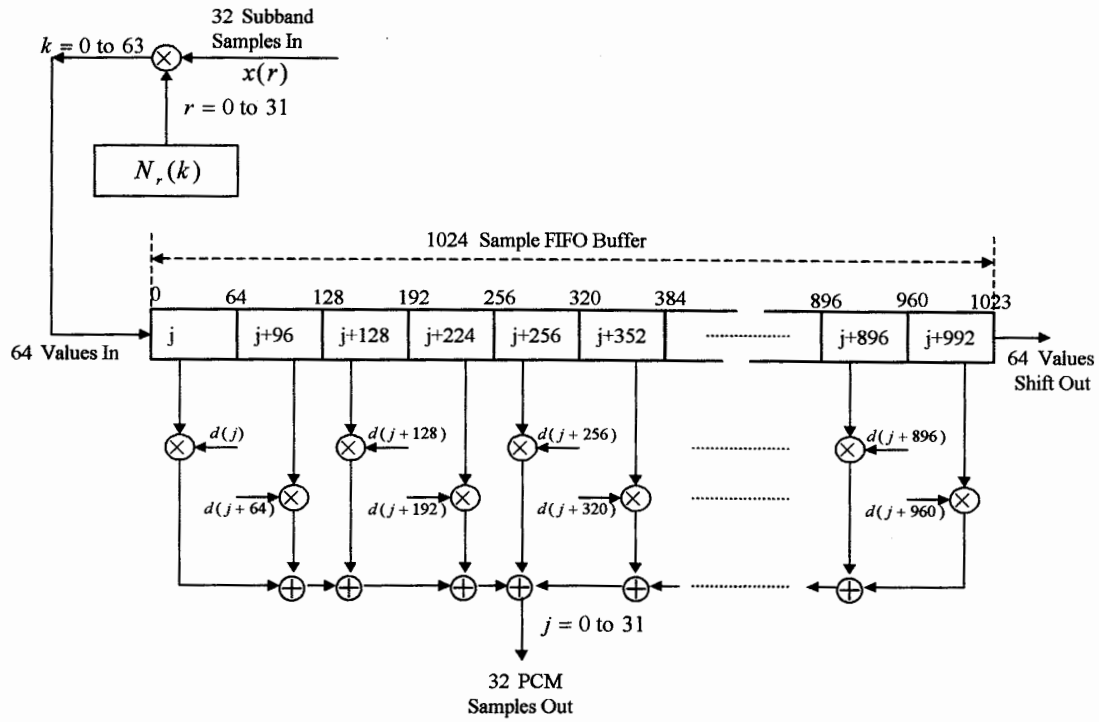


FIG. 10

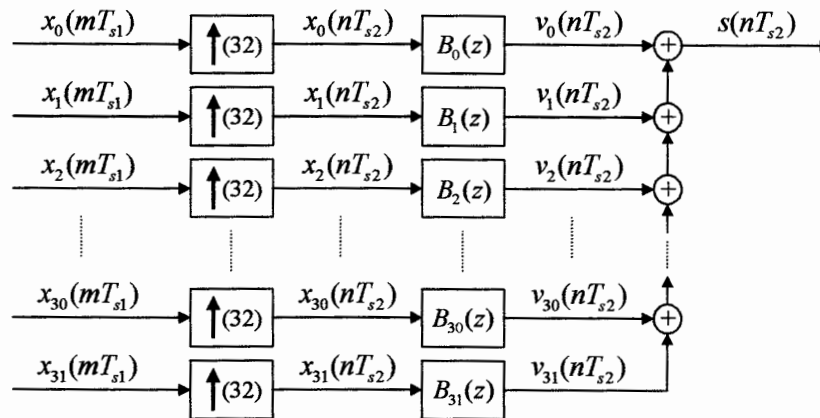


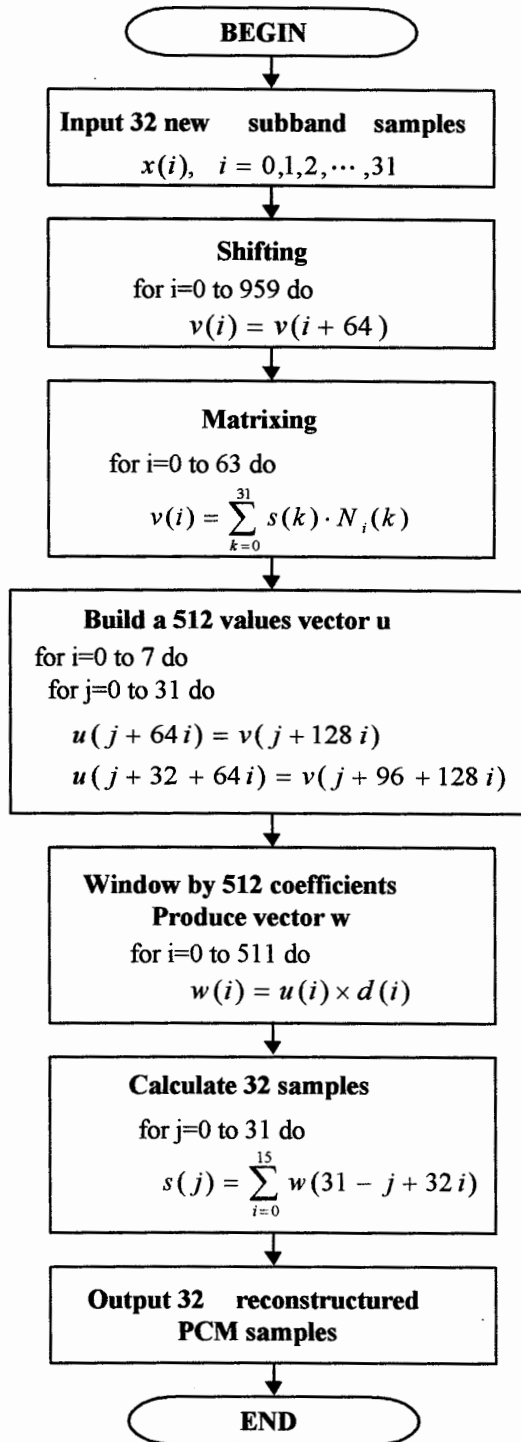
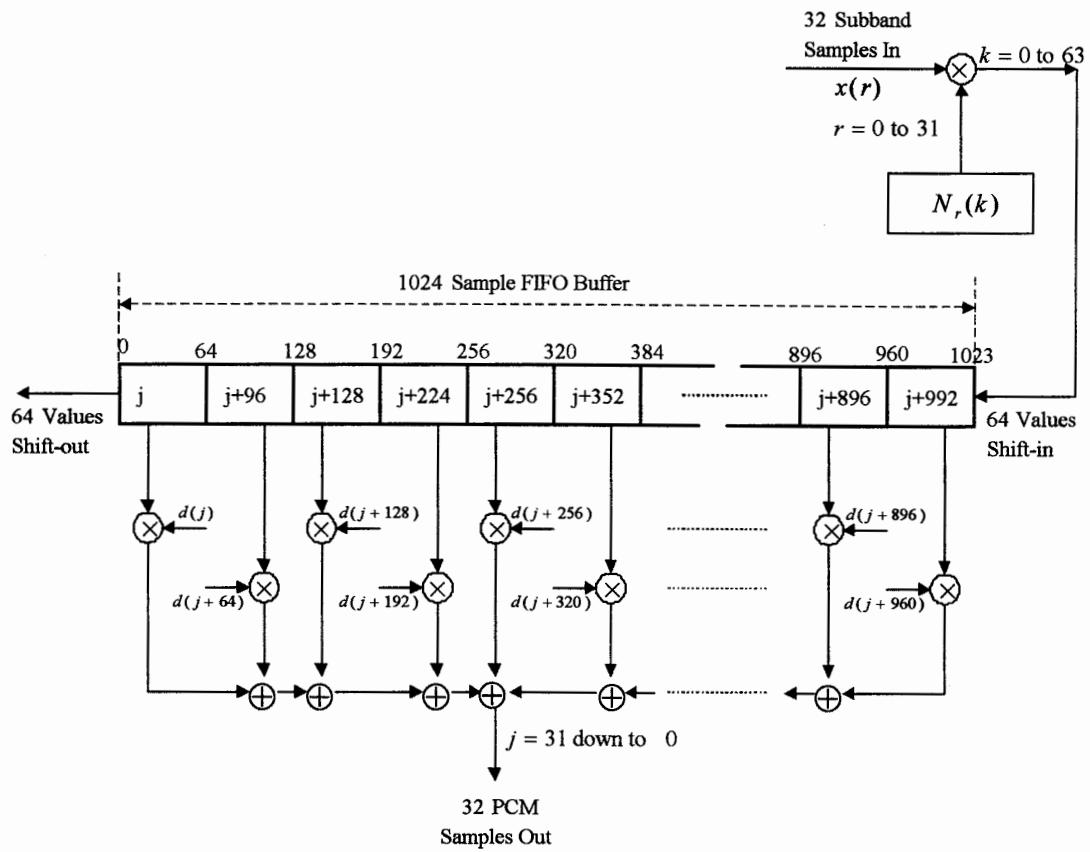
FIG. 11

FIG. 12



INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR 99/00764

A. CLASSIFICATION OF SUBJECT MATTER		
IPC ⁷ : H 03 M 7/40, H 04 N 7/50		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
IPC ⁷ : G 10 L 19/00, G 11 C 7/00, H 03 M 7/40, H 04 N 7/24, 7/50, 7/52		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
WPI		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0687111 A2 (SICAN GMBH.) 13 December 1995 (13.12.95), fig. 3, claims 1-9.	1,5,6
A	JP 10-178349 A (MATSUSHITA). Patent abstracts of Japan, Vol. 199, No. 811, 1998.09.30 (abstract).	1

<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: „A“ document defining the general state of the art which is not considered to be of particular relevance „E“ earlier application or patent but published on or after the international filing date „L“ document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) „O“ document referring to an oral disclosure, use, exhibition or other means „P“ document published prior to the international filing date but later than the priority date claimed „T“ later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention „X“ document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone „Y“ document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art „&“ document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
21 March 2000 (21.03.00)		19 July 2000 (19.07.00)
Name and mailing address of the ISA/AT Austrian Patent Office Kohlmarkt 8-10; A-1014 Vienna Facsimile No. 1/53424/200		Authorized officer Fussy Telephone No. 1/53424/328

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR 99/00764

Patent document cited in search report			Publication date	Patent family member(s)			Publication date
EP	A2	687111	13-12-1995	CA	AA	2151023	07-12-1995
EP	A3	687111	16-12-1998	DE	A1	4419678	07-12-1995
				DE	A1	4436956	18-04-1996
				DE	C2	4436956	12-03-1998
JP	A2	10178349	30-06-1998	none			