ISO/IEC MPEG-2 Advanced Audio Coding[1]

MARINA BOSI[*2], *AES Member*, **KARLHEINZ BRANDENBURG**[**], *AES Fellow*, **SCHUYLER QUACKENBUSH**[***], **LOUIS FIELDER**[*], *AES Fellow*, **KENZO AKAGIRI**[†], **HENDRIK FUCHS**[††], **MARTIN DIETZ**[**], **JÜRGEN HERRE**[†††], *AES Fellow*, **GRANT DAVIDSON**[*], *AES Member*, **AND YOSHIAKI OIKAWA**[†]

*Dolby Laboratories, San Francisco, CA 94103, USA

**FhG-IIS, D-91058, Erlangen, Germany

***AT&T Laboratories, Murray Hill, NJ 07574, USA

†Sony Corporation, Tokyo, Japan

††University of Hanover, D-30167, Hanover, Germany

†††Lucent Technologies Bell Laboratories, Murray Hill, NJ 07574, USA

The ISO/IEC MPEG-2 advanced audio coding (AAC) system was designed to provide MPEG-2 with the best audio quality without any restrictions due to compatibility requirements. The main features of the AAC system (ISO/IEC 13818-7) are described. MPEG-2 AAC combines the coding efficiency of a high-resolution filter bank, prediction techniques, and Huffman coding with additional functionalities aimed to deliver very high audio quality at a variety of data rates.

0 INTRODUCTION

The standardization body ISO/IEC JTC1/SC29/WG11, also known as the Moving Pictures Expert Group (MPEG), was established in 1988 to specify digital video and audio coding schemes at low data rates. MPEG completed its first phase of specifications (MPEG-1) in 1992 November [1]. The MPEG-1 audio coding system, specified in ISO/IEC 11172-3 (see also [2]) operates in single-channel or two-channel stereo modes at sampling frequencies of 32, 44.1, and 48 kHz. MPEG-1 layer 2 provides very high quality at data rates of 128 kbit/s per channel [3], [4].

In its second phase of development, MPEG's goals were to define a multichannel extension to MPEG-1 audio that would be backward compatible with existing MPEG-1 systems (MPEG-2 BC) and to define an audio coding standard at lower sampling frequencies (16, 22.5, 24 kHz) than MPEG-1, MPEG-2 LSF. Both MPEG-2 BC and MPEG-2 LSF were completed in 1994 November [5]. MPEG-2 BC provides good audio quality at data rates of 640–896 kbit/s [6] for five full-bandwidth channels.

Started in 1994, another effort of the MPEG-2 audio standardization committee was to define a higher quality multichannel standard than achievable while requiring MPEG-1 backward compatibility, the so-called MPEG-2 non-backward-compatible audio standard, later renamed MPEG-2 advanced audio coding (MPEG-2 AAC) [7]. The aim of this development was to reach "indistinguishable" audio quality as specified by the International Telecommunication Union, Radiocommunication Bureau (ITU-R) [8] at data rates of 384 kbit/s or lower for five full-bandwidth channel audio signals. Tests carried out in the fall of 1996 at BBC, UK, and NHK, Japan, showed that MPEG-2 AAC satisfies the ITU-R quality requirements at 320 kbit/s per five full-bandwidth channels (or lower according to the NHK data).

MPEG-2 AAC was finalized as an international stan-

[1] Presented at the 101st Convention of the Audio Engineering Society, Los Angeles, CA, 1996 November 8–11; revised 1997 July 22.

[2] Currently with Digital Theater Systems (DTS), Los Angeles, CA, USA.

dard in 1997 April (ISO/IEC 13818-7) [9]. The MPEG-2 AAC scheme will also constitute the kernel of the forthcoming MPEG-4 audio standard [7]. The MPEG-2 AAC specifications are the result of a collaborative effort among companies around the world, each of which contributed advanced audio coding technology. AAC combines the coding efficiency of a high-resolution filter bank, prediction techniques, and Huffman coding to achieve broadcast-quality audio at very low data rates. The AAC specifications have undergone a number of revisions since the first submission of proposals (1994 November). In order to define the AAC system, the committee has selected a modular approach in which the full system is broken down into a series of self-contained modules. The AAC reference model (RM) describes the requirements of each module and how they fit together. Each aspect of the RM has been evaluated via core experiments that were carried out between 1995 January and 1996 July. In order to be incorporated in the final international standard, aspects of the RM were selected based on core experiment results. Table 1 lists the milestones of the MPEG-2 AAC project.

The following AAC modules are described in this paper:

- Gain control
- Filter bank
- Prediction
- Quantization and coding
- Noiseless coding
- Bit-stream multiplexing
- Temporal noise shaping (TNS)
- Mid/side (M/S) stereo coding and intensity stereo coding.

TNS, M/S, and intensity stereo coding are also described in detail in other publications [10], [11].

1 OVERVIEW OF MPEG-2 ADVANCED AUDIO CODING

At low bit rates, efficient audio coding removes both redundancies and irrelevancies from audio signals. Correlations between audio samples and statistics of the samples' representation are exploited in order to remove redundancies. Frequency-domain and time-domain masking properties of the human auditory system [12] are exploited in order to remove imperceptible signal content (irrelevancies). The frequency content of the audio signal is subdivided into subbands by means of a filter bank. The data-rate reduction is achieved by quantizing the spectrum of the signal according to perceptual models and includes a noiseless coding process. The steps to carry out these processes, as will be fully described in the following sections, lead to the basic structure of the MPEG-2 AAC system, as shown in Figs. 1 and 2.

Table 1. Development of ISO/IEC MPEG audio and milestones for MPEG-2 AAC.

ISO/IEC MPEG-1 audio finalized (ISO/IEC 11172-3)	1992 November
ISO/IEC MPEG-2 audio BC and LSF finalized (ISO/IEC 13818-3)	1994 November
First submission of proposals for ISO/IEC MPEG-2 AAC	1994 November
AAC first core experiment plan, starting of collaborative phase	1995 January
AAC committee draft	1996 July
AAC formal tests results, draft international standard	1996 November
ISO/IEC MPEG-2 AAC finalized (ISO/IEC 13818-7)	1997 April
ISO/IEC MPEG-4 audio finalized	1999 January

In order to allow a tradeoff between the quality and the memory and processing power requirements, the AAC system offers three profiles: main profile, low-complexity (LC) profile, and scalable sampling rate (SSR) profile.

- *Main profile*: In this configuration the AAC system provides the best audio quality at any given data rate. With the exception of the gain control tool, all parts of the AAC tools may be used. The memory and processing power required in this configuration are higher than the memory and processing power required in the LC profile configuration (see also Section 10). It should be noted that a main profile AAC decoder can decode an LC-profile encoded bit stream.
- *Low-complexity (LC) profile*: In this configuration the prediction and preprocessing tools are not employed and the TNS order is limited. While the quality performance of the LC profile is very high (see Section 9), the memory and processing power requirements are considerably reduced in this configuration.
- *Scalable sampling rate (SSR) profile*: In this configuration the gain control tool is required. The preprocessing performed by the gain control tool consists of a polyphase quadrature filter (PQF), gain detectors, and gain modifiers. The prediction module is not used in this profile, and the TNS order and bandwidth are limited. The SRS profile has lower complexity than the main and LC profiles, and it can provide a frequency scalable signal.

The AAC encoder process can be described as follows. First, a filter bank is used to decompose the input signal into subsampled spectral components (time–frequency domain). At 48 kHz the AAC system allows for a frequency resolution of 23 Hz and time resolution of 2.6 ms. Based on the input signal, an estimate of the current (time-dependent) masking threshold is computed using rules known from psychoacoustics. A perceptual model similar to the MPEG-1 psychoacoustic model II [1] is used for the AAC system. A signal-to-mask ratio, that is, an assessment of how much quantization noise can be masked by the input signal, is derived from the masking threshold. This information is utilized in the quantization stage in order to minimize the audible distortion of the quantized signal at any given data rate.

After the analysis filter bank, the TNS performs an in-place filtering operation on the spectral values, that is, it replaces the target spectral coefficients (set of spectral

coefficients to which TNS should be applied) with the prediction residual. The TNS technique permits the encoder to exercise control over the temporal fine structure of the quantization noise, even within a filter-bank window.

For multichannel signals, intensity stereo coding is also applied. In this operation only the energy envelope is transmitted. Intensity stereo coding allows for a reduction in the spatial information transmitted. This is a powerful method for reducing audible artifacts at very low data rates.

The time-domain prediction tool is used in order to take advantage of correlations between subsampled spectral components of subsequent frames resulting in an increased redundancy reduction for stationary signals.

Instead of transmitting the left and right signals, the normalized sum (M as in middle) and difference signals (S as in side) are transmitted. Enhanced M/S stereo coding is used in the multichannel AAC encoder at low data rates.

The spectral components are quantized and coded with the aim of keeping the quantization noise below the masked threshold. This step is done by employing an analysis-by-synthesis stage and using additional noiseless compression tools. A mechanism called "bit reservoir" allows for a locally variable data rate in order to satisfy the signal demands on a frame-by-frame basis.

Finally a bit-stream formatter is used to assemble the bit stream, which consists of the quantized and coded spectral coefficients and control parameters.

The MPEG-2 AAC system supports up to 48 audio channels. Default configurations include monophonic, two-channel and five-channel plus low-frequency enhancement (LFE) channel (bandwidth <200 Hz) configuration. In the default five-channel plus LFE configuration, the 3/2 loudspeaker arrangement is adopted [13]. In this configuration the channel presentation is as follows: center, left, right, left surround, right surround. In addition to the default configurations, 16 possible

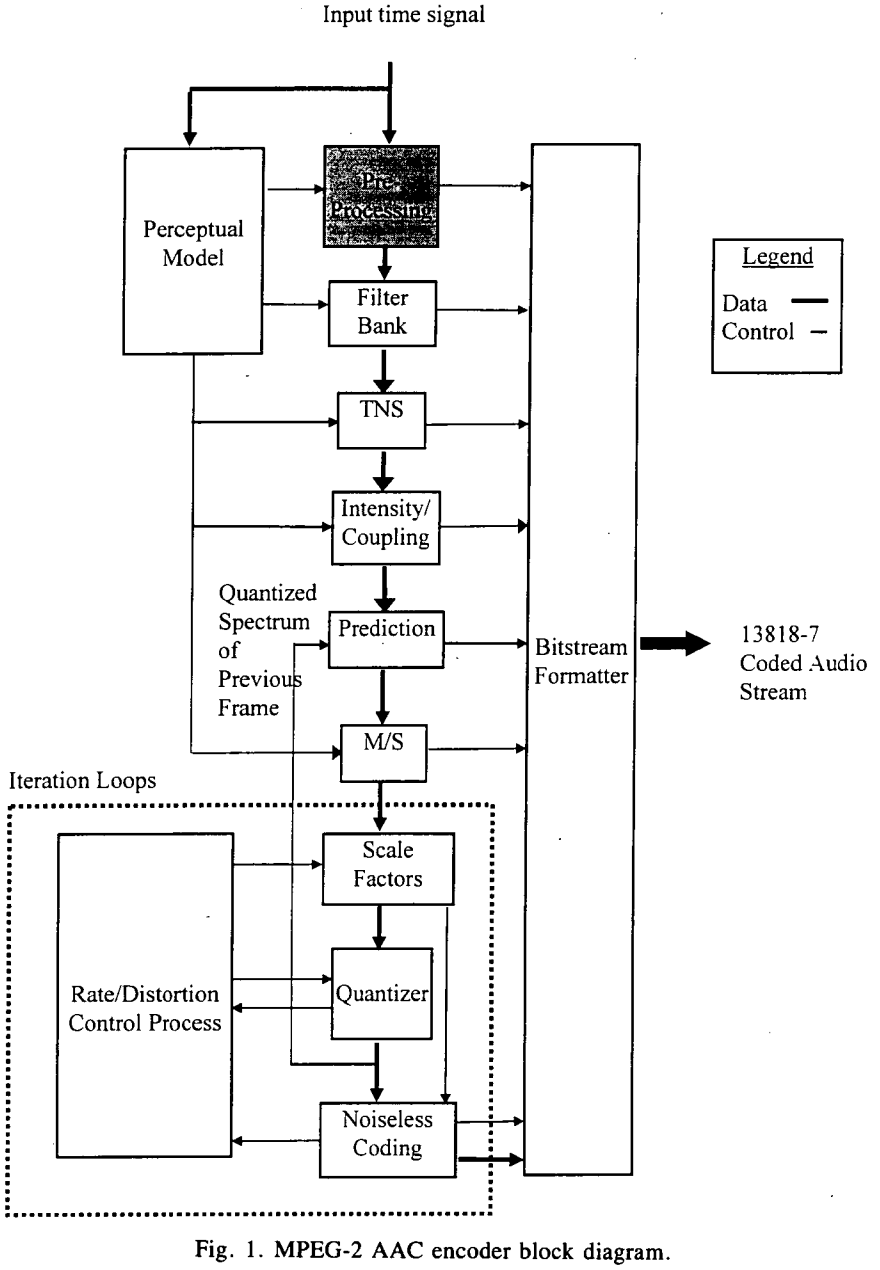


Fig. 1. MPEG-2 AAC encoder block diagram.

program configurations can be defined in the encoder. Downmix capability is also supported [14].

The sampling rates supported by the AAC system vary from 8 to 96 kHz, as shown in Table 2. Table 2 also shows the maximum data rate per channel, which depends on the sampling rate.

2 GAIN CONTROL

In the SSR profile, the gain control block is added in the input stage of the encoder. The gain control module consists of a PQF bank, gain detectors, and gain modifiers. The PQF splits each audio channel's input signal into four frequency bands of equal width, which are critically decimated. Each filter bank's output has gain modification as necessary and is processed by the modified discrete cosine transform (MDCT) tool to produce 256 spectral coefficients, for a total of 1024 coefficients. Gain control can be applied to each of the four bands independently. The step size of the gain modification is 2^n, where n is an integer.

SSR gain control in the decoder has the same components as it does in the encoder, but in an inverse arrangement. The distinctive feature of the SSR profile is that lower bandwidth output signals, and hence lower sampling rate output signals, can be obtained by neglecting the signal from the upper bands of the PQF. This leads to output bandwidths of 18, 12, and 6 kHz when one,

Table 2. MPEG-2 AAC sampling frequencies and maximum data rates.

Sampling Frequency (Hz)	Maximum Bit Rate per Channel (kbit/s)
96 000	576
88 200	329.2
64 000	384
48 000	288
44 100	264.6
32 000	192
24 000	144
22 050	132.3
16 000	96
12 000	72
11 025	66.25
8 000	48

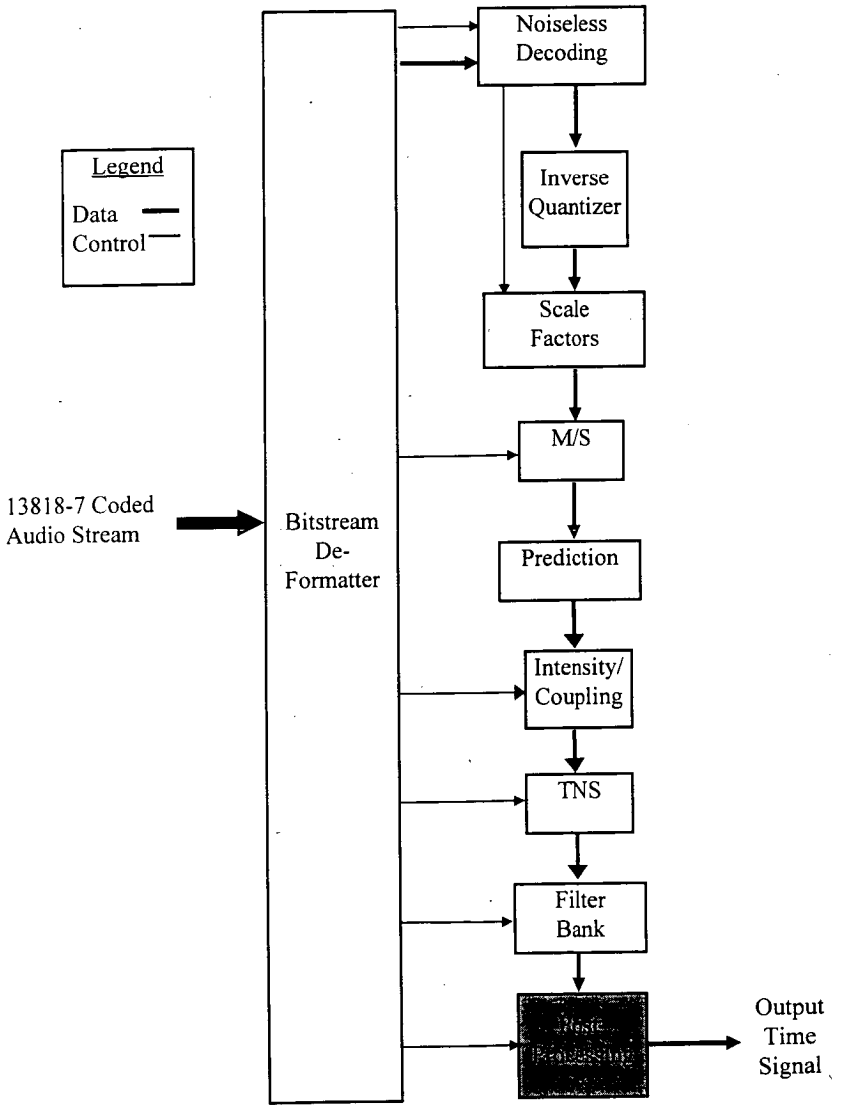


Fig. 2. MPEG-2 AAC decoder block diagram.

two, or three PQF outputs are ignored, respectively. The advantage of this scalability is that the decoder complexity can be reduced as the output bandwidth is reduced.

2.1 Encoding Process

The gain control module in the encoder receives as input the time-domain signals and produces as outputs the gain control data and a gain-modified signal whose length is equal to the length of the MDCT window (see Section 3). The block diagram of the gain control tool is shown in Fig. 3.

2.1.1 PQF

The input signal is divided by the PQF into four frequency bands of equal width. The coefficients of each band's PQF are given by

$$h_i(n) = 4\cos\left[\frac{(2i+1)(2n+5)\pi}{16}\right] Q(n), \quad 0 \leq n \leq 95, \quad 0 \leq i \leq 3 \qquad (1)$$

where

$$Q(n) = Q(95-n), \quad 48 \leq n \leq 95 \qquad (2)$$

the $Q(n)$ being the filter coefficients that are standardized in the decoder.

2.1.2 Gain Detector

The gain detector produces gain control data consisting of the number of bands receiving gain modification, and the number of modified segments and indices indicating the location and level of gain modification for each segment. Note that the output gain control data are for the signal of the previous frame, so that the gain detector has a one-frame delay.

The time resolution of the gain control is approximately 0.7 ms at a 48-kHz sampling rate. The step size of gain control is 2^n, where n is an integer between -4 and 11. Thus the signal can be amplified or attenuated by the gain control.

2.1.3 Gain Modifier

The gain modifier applies gain control to the signal in each PQF band by windowing the signals by the gain control function. The encoder gain control function is calculated from the gain control data such that it applies the inverse gain with respect to the decoder gain control function.

2.2 Decoding Process

The gain control module is added at the end of the decoding process in the SSR profile. Postprocessing performed by the gain control tool consists of applying gain compensation to the sequences produced by each of the four inverse MDCTs (IMDCTs), overlapping and adding successive sequences with appropriate time alignment and combining these sequences in the inverse polyphase quadrature filter (IPQF) bank. The block diagram of the decoder gain control is shown in Fig. 4.

2.2.1 Gain Compensator and Overlapping

Gain compensation in the decoder requires the following three steps for each of the PQF bands:

1) Decoding of gain control data
2) Calculation of gain control function
3) Windowing and overlap adding.

In decoding the gain control data, the gain modification and the position of the modification for each band are decoded from bit-stream elements. From this information the gain control function is calculated, and is

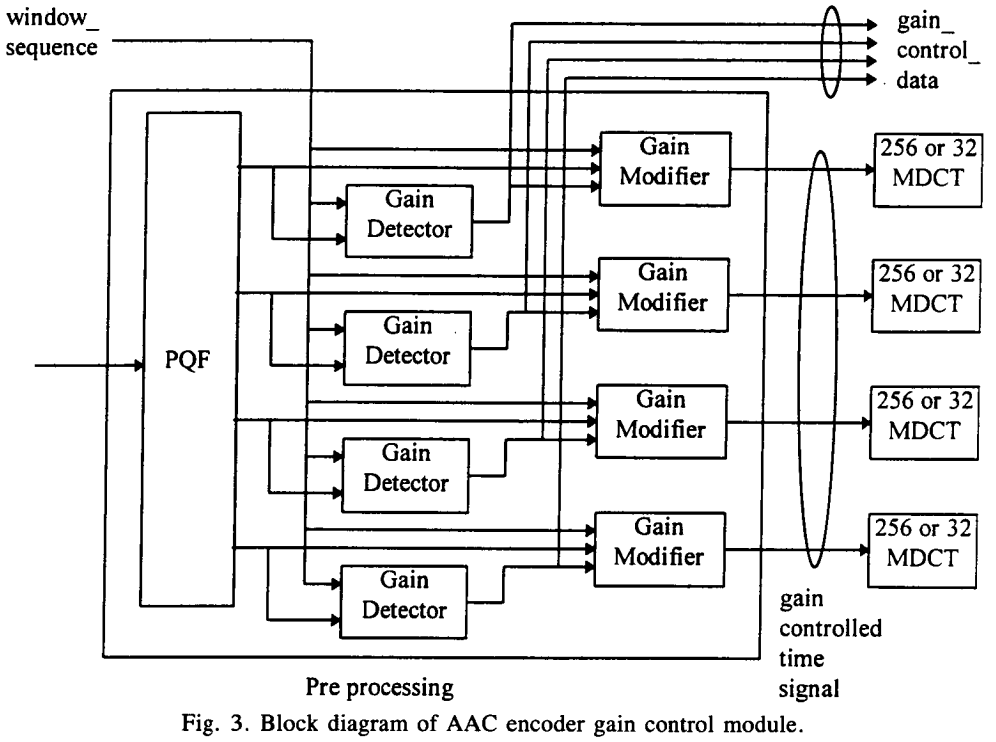


Fig. 3. Block diagram of AAC encoder gain control module.

used to window the sequence from the IMDCT. Consecutive windowed sequences are overlapped and added with appropriate time alignment.

2.2.2 IPQF

The IPQF combines the four frequency bands to synthesize the decoded signal. The aliasing components introduced by the PQF in the encoder are canceled by the IPQF.

3 FILTER BANK

A fundamental component of the MPEG-2 AAC system is the conversion of the time-domain signals at the input of the encoder into an internal time–frequency representation and the reverse process in the decoder. This conversion is done by a forward MDCT and an IMDCT in the decoder. The MDCTs and IMDCTs employ a technique called time-domain aliasing cancellation (TDAC). Additional information about the TDAC transform and the window-overlap–add process can be found in [15], [16].

The analytical expression for the MDCT is

$$X_{ik} = 2 \sum_{n=0}^{N-1} x_{\text{in}} \cos\left[\frac{2\pi}{N}(n + n_0)\left(k + \frac{1}{2}\right)\right],$$

$$k = 0, \ldots, \frac{N}{2} - 1. \quad (3)$$

Since the sequence X_{ik} is odd symmetric, the coefficients from 0 to $N/2 - 1$ uniquely specify the transform. The analytical expression of the IMDCT is

$$X_{\text{in}} = \frac{2}{N} \sum_{k=0}^{N/2-1} X_{ik} \cos\left[\frac{2\pi}{N}(n + n_0)\left(k + \frac{1}{2}\right)\right],$$

$$n = 0, \ldots, N - 1 \quad (4)$$

where

n = sample index
N = transform block length
i = block index
$n_0 = \dfrac{N/2 + 1}{2}$.

In the encoder this process takes the appropriate block of time samples, modulates them by an appropriate window function, and performs the MDCT to ensure good frequency selectivity for the filter bank. Each block of input samples is overlapped by 50% with the immediately preceding block and the following block. The transform block length N can be set to either 2048 or 256 samples.

Because the window function has a significant effect on the filter-bank frequency response, the MPEG-2 AAC filter bank has been designed to allow a change in window shape to best adapt to input signal conditions. The shape of the window is varied in the encoder and the decoder simultaneously to allow the filter bank to separate spectral components of the input efficiently for a wider variety of input signals.

The use of the 2048 time-domain sample transform improves coding efficiency for signals with complex spectra, but may create problems for transient signals. Quantization errors extending more than a few milliseconds before a transient event are not effectively masked by the transient itself. This leads to a phenomenon called preecho, in which the quantization error from one transform block is spread in time and becomes audible. Long transforms are inefficient for coding signals that are transient in nature. Transient signals are best encoded with relatively short transform lengths. Unfortunately short transforms produce inefficient coding of steady-state signals due to poorer frequency resolution. See Davidson and Bosi [17] for a more detailed explanation of this problem.

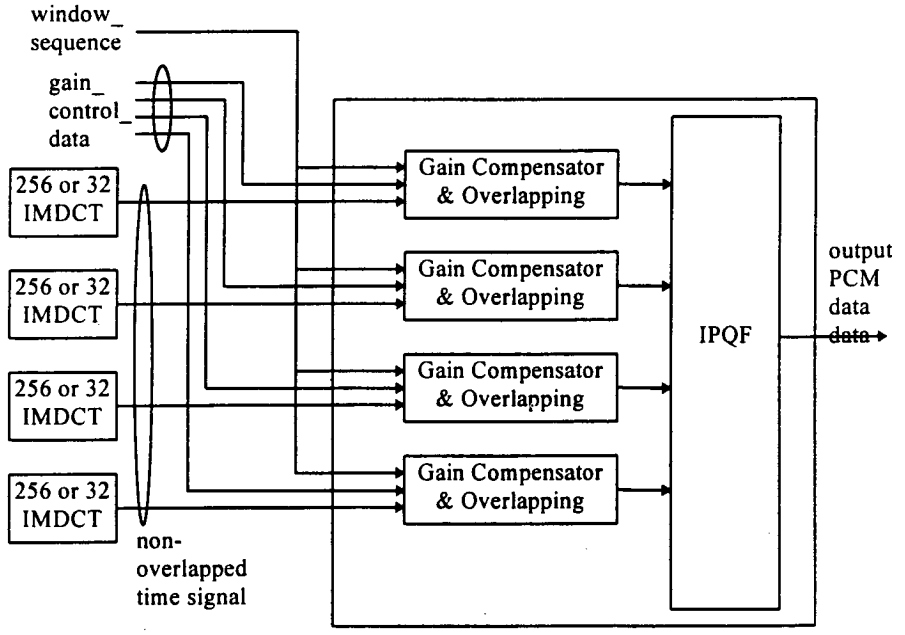


Fig. 4. Block diagram of AAC decoder gain control module.

The MPEG-2 AAC system circumvents this problem by allowing the block length of the transform to vary as a function of the signal conditions [18]. Signals that are short-term stationary are best accommodated by the long transform, whereas transient signals are generally reproduced more accurately by short transforms. The transition between long and short transforms is seamless in the sense that aliasing is completely canceled in the absence of transform coefficient quantization.

3.1 Filter-Bank Frequency Selectivity and Window Design

The frequency selectivity of an MDCT filter bank is dependent on the window function. A window commonly used in audio coding is the sine window. This window produces a filter bank with good separation of nearby spectral components, improving coding efficiency for signals with a dense harmonic content. For other types of signals, however, a window with better ultimate rejection may provide better coding efficiency.

A desirable characteristic for filter banks used in audio coding is perfect reconstruction. Perfect reconstruction is achieved in MDCT filter banks when the sum of the product of overlapped analysis and synthesis windows equals a constant [7]. The sine window is a natural choice because it satisfies this constraint. However, the sine window frequency selectivity can be improved for some classes of signals using a window with higher ultimate rejection. Such a window as designed for the AAC system using a numerical procedure which allows optimization of the transition bandwidth and the ultimate rejection of the filter bank, and simultaneously guarantees perfect reconstruction. This window is called a Kaiser–Bessel derived (KBD) window and is introduced in [16].

Fig. 5 shows the relative amplitude response or spectral leakage of one MDCT band for both sine and KBD windows at the sampling rate of 48 kHz. A comparison with a minimum masking template from [16] is also shown. The solid line represents the KBD window, the dotted line the sine function window, and the dashed line the minimum masking template. The figure shows that the KBD window leakage function lies below the minimum masking template for frequency offsets greater than 110 Hz, whereas the sine window leakage is approximately 20 dB greater than the masking template. This indicates that the KBD window is effective at perceptually isolating spectral components when the components are spaced more than 220 Hz apart. The sine window is not as effective at isolating components for most frequency separations. Note that the increased rejection of the KBD window comes at a price of poorer rejection of components for frequency offsets less than ±70 Hz when compared to the sine window. Therefore when perceptually significant frequency components are spaced more closely than 140 Hz, the filter bank employing a sine window may be more effective.

3.2 Window Shape Adaptation

In Section 3.1 it has been shown that the use of only one window shape results in compromises in filter-bank efficiency for certain signals. This observation naturally leads to a coder improvement in which the window shape can be varied dynamically as a function of the signal. The AAC system allows seamless switching between KBD and sine windows. Perfect reconstruction and critical sampling are preserved in the filter bank during window shape changes. A single bit per frame is transmitted in the bit stream to indicate the window shape. Note that window shape switching is different than block switching (described in Section 3.3).

Window shape decisions made by the encoder are applicable to the second half of the window function only since the first half is constrained to use the appropriate window shape from the preceding frame. Fig. 6 shows the overlap–add sequence of blocks for the two situations. The sequence of windows labeled A–B–C employs the KBD window, whereas the sequence D–E–F

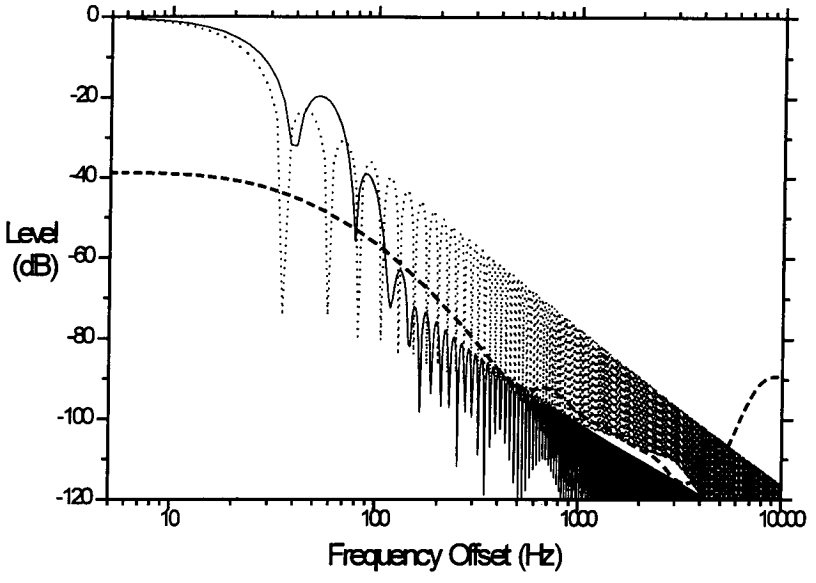


Fig. 5. Comparison of masking template with 2048-sample transform filter-bank frequency selectivity of KBD and sine-function windows at 48 kHz. —— KBD window; ··· sine-function window; – – – minimum masking template.

shows the transition to and from a single frame employing the sine function window. The window shape selector generally produces window shape run lengths greater than that shown in the figure.

3.3 Transform Block Switching

The adaptation of the time–frequency resolution of the filter bank to the characteristics of the input signal is done by shifting between transforms whose input lengths are either 2048 or 256 samples. The 256 sample length for transient signal coding was selected as the best compromise between frequency selectivity and preecho suppression at a data rate of around 64 kbit/s per channel. Transform block switching is an effective tool for adapting the time–frequency resolution of the filter bank but potentially creates a problem of block synchrony between the different channels being coded. If one channel uses a 2048 transform length during the same time interval that another channel uses three 256 transforms, the long blocks following the block switch interval will no longer be time aligned. This lack of alignment between channels is undesirable since it creates problems in combining channels during encoding and bit-stream formatting and deformatting.

This problem of maintaining block alignment between each channel of the MPEG AAC system has been solved as follows. During transitions between long and short transforms a start and stop bridge window is used that preserves the time-domain aliasing cancellation properties of the MDCT and IMDCT transforms and maintains block alignment. These bridge transforms are designated "start" and "stop" sequences, respectively. The conventional long transform with the 2048-sample length is termed a "long" sequence, while the short transforms occur in groups called "short" sequence. The short sequence is composed of eight short block transforms arranged to overlap 50% with each other and have the half transforms at the sequence boundaries to overlap with the start and stop window shapes. This overlap sequence and grouping of transform blocks into start, stop, long, and short sequences is shown in Fig. 7.

Fig. 7 displays the window overlap–add process appropriate for both steady-state and transient conditions. Curves A, B, and C represent this process when block switching is not employed and all transforms have 2048 samples and are composed of long sequences only. The windowed transform blocks A, B, and C are 50% overlapped with each other and assembled in sequential order. The lower part of the figure shows the use of block switching to smoothly transition to and from the shorter $N = 256$ time sample transforms that are present in the sample index region between sample numbers 1600 and 2496. The figure shows that short-length transforms (#2–#9) are grouped into a sequence of eight 50% overlapped transforms of length 256 samples each and employing a sine function window of the appropriate length. The start (#1) and stop (#10) sequences allow a smooth transition between short and long transforms. The first half of the window function for the start sequence, that is, time-domain samples numbered 0–1023, is the first half of either the KBD or the sine window that matches the previous long sequence window type. The next section of the window has a value of unity between sample numbers 1024 to 1471; then it is followed by KBD or a sine window. The sine window function is given by the formula

$$W = \sin\left[\frac{\pi}{256}(n - 1343.5)\right] \tag{5}$$

where $1472 \leq n < 1600$.

This region is followed by a final region with zero-valued samples to sample number 2047. The stop sequence window is the time-reversed version of the start window, and both are designed to ensure a smooth transition between transforms of both lengths and the proper time-domain aliasing cancellation properties for the transforms used. Additional information and an explana-

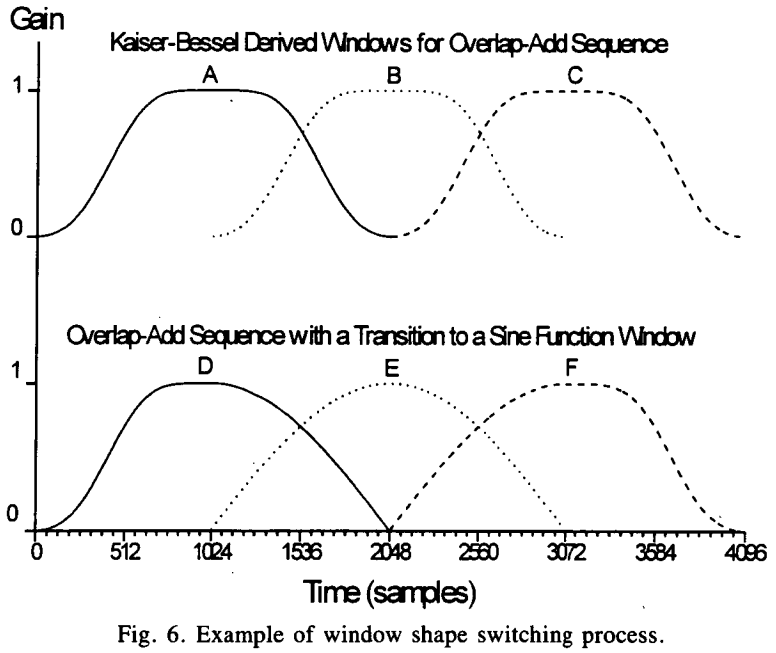


Fig. 6. Example of window shape switching process.

tion of this basic method of block switching are given in [18]. This AAC block switching method allows the flexibility of encoding transients with eight or more 256-point transforms while preserving the block alignment of the channels. For transients which are closely spaced, a single eight-short-window sequence can be extended by adding more consecutive short windows, subject to the restriction that short windows must be added in integral multiples of eight.

4 PREDICTION

4.1 Overview

Prediction is used for improved redundancy reduction and is especially effective in case of more or less stationary parts of a signal which belong to the most demanding parts in terms of the required data rate. Because the use of a short window in the filter bank indicates signal changes, that is, a nonstationary signal characteristic, prediction is only used for long windows.

For each channel, prediction is applied to the spectral components resulting from the spectral decomposition of the filter bank. For each spectral component up to 16 kHz there is one corresponding predictor, resulting in a bank of predictors, where each predictor exploits the autocorrelation between the spectral component values of consecutive frames.

Since a filter bank with high spectral resolution is used in the AAC system, backward-adaptive predictors are adopted. The predictor coefficients are calculated from preceding quantized spectral components in the encoder as well as in the decoder. In this case, no additional side information is needed for the transmission of predictor coefficients—as would be required if forward-adaptive predictors were to be used. A second-order backward-adaptive lattice structure predictor is used for each spectral component, so that each predictor is working on the spectral component values of the two preceding frames. The predictor parameters are adapted to the current signal statistics on a frame-by-frame base, using an LMS-based adaptation algorithm. If prediction is activated, the quantizer is fed with a prediction error instead of the original spectral component, resulting in a higher coding efficiency. Fig. 8 shows the block diagram of the prediction unit for one single predictor of the predictor bank. The predictor control operates on all predictors of one scale factor band (see Section 4.3). A more detailed description of the principles can be found in [19].

4.2 Predictor Processing

The following description is valid for one single predictor and has to be applied to each predictor. An estimate $x_{est}(n)$ of the current value of the spectral component $x(n)$ is calculated from preceding reconstructed values $x_{rec}(n-1)$ and $x_{rec}(n-2)$, stored in the predictor state variables, using the predictor coefficients $k_1(n)$ and $k_2(n)$. This estimate is then subtracted from the spectral component $x(n)$, resulting in the prediction error $e(n)$, which is then quantized to $e_q(n)$ and coded. In the decoder the same estimate is calculated and added to the quantized prediction error reconstructed from the transmitted data, resulting in the reconstructed value $x_{rec}(n)$ of the current spectral component $x(n)$.

Due to the realization in a lattice structure, the predictor consists of two so-called basic elements that are cascaded. In each element, the part $x_{est,m}(n)$, $m = 1, 2$, of the estimate is calculated according to

$$x_{est,m}(n) = b \cdot k_m(n) \cdot a \cdot r_{m-1}(n-1) \tag{6}$$

where

$$r_{q,m}(n) = r_{q,m-1}(n-1) - b \cdot k_m(n) \cdot e_{q,m-1}(n)$$

and

$$e_{q,m}(n) = e_{q,m-1}(n) - 1\, x_{est,m}(n)\,. \tag{7}$$

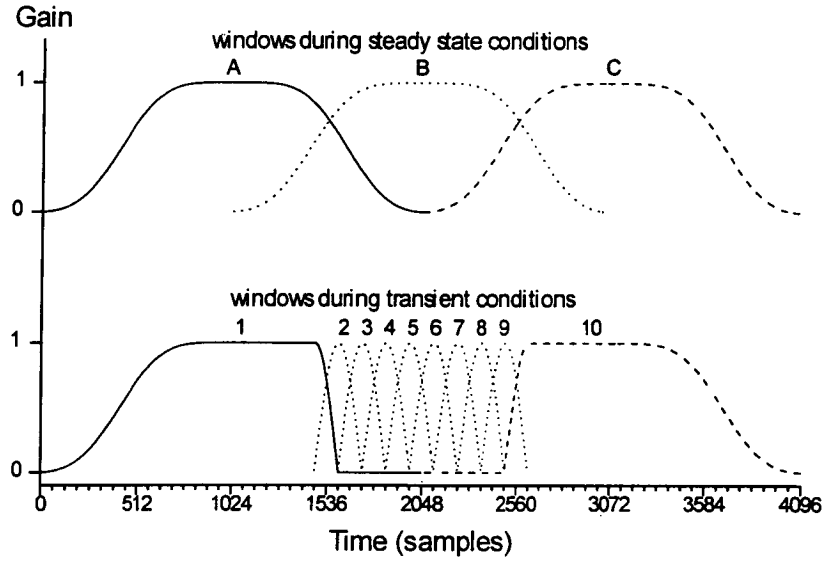


Fig. 7. Comparison of window overlap–add processes for steady-state and transient conditions.

Hence the overall estimate results in

$$x_{est}(n) = x_{est,1}(n) + x_{est,2}(n) . \quad (8)$$

The constants a and b, $0 < a, b \leq 1$, are attenuation factors that are included in each signal path, contributing to the recursivity of the structure for the purpose of stabilization. Thus possible oscillations due to transmission errors or drift between predictor coefficients on the encoder and decoder side due to numerical inaccuracy can be faded out or even prevented.

In the case of stationary signals and with $a = b = 1$, the predictor coefficient of element m is calculated by

$$k_m(n) = \frac{E[e_{q,m-1}(n) \cdot r_{q,m-1}(n-1)]}{1/2\,(E[e^2_{q,m-1}(n)] + E[r^2_{q,m-1}(n-1)])}, \quad m = 1, 2 \quad (9)$$

and

$$e_{q,0}(n) = r_{q,0}(n) = x_{rec}(n) . \quad (10)$$

In order to adapt the coefficients to the current signal properties, the expected values in Eq. (10) are substituted by time average estimates measured over a limited past signal period. A compromise has to be chosen between a good convergence against the optimum predictor setting for signal periods with quasi-stationary characteristic and the ability of fast adaptation in case of signal transitions. In this context algorithms with iterative improvement of the estimates, that is, from sample to sample, are of special interest. Here a least-mean-square (LMS) approach is used and the predictor coefficients are calculated as follows:

$$k_m(n+1) = \frac{\mathrm{COR}_m(n)}{\mathrm{VAR}_m(n)} \quad (11)$$

with

$$\mathrm{COR}_m(n) = \alpha \cdot \mathrm{COR}_m(n-1) + r_{q,m-1}(n-1) \cdot e_{q,m-1}(n)$$

$$\mathrm{VAR}_m(n) = \alpha \cdot \mathrm{VAR}_m(n-1) + 0.5 + [r^2_{q,m-1}(n-1) + e^2_{q,m-1}(n)] \quad (12)$$

where α is an adaptation time constant which determines the influence of the current sample on the estimate of the expected values. The value of α is chosen as $\alpha = 0.90625$.

The optimum values of the attenuation factors a and b have to be determined as a compromise between high prediction gain and small fade-out time. The values chosen are $a = b = 0.953125$.

Whether prediction is disabled—either totally or only for a particular scale factor band—or not, all the predictors are run all the time in order to adapt the coefficients always to the current signal statistics.

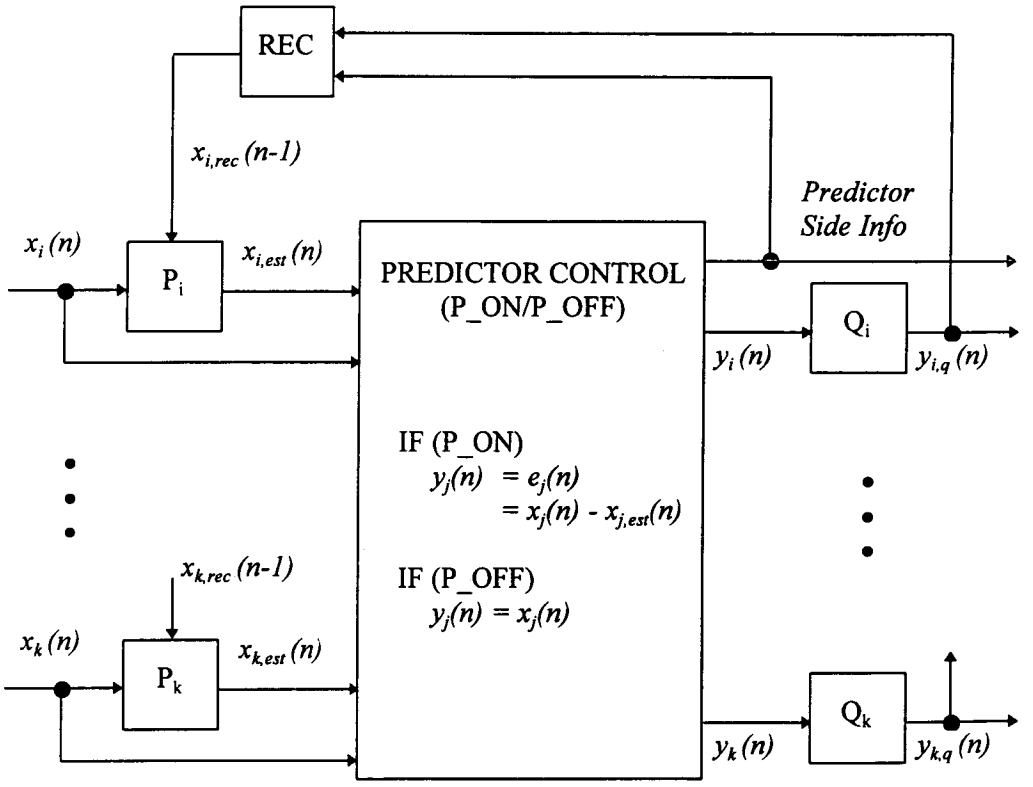


Fig. 8. Block diagram of AAC prediction unit for one scale factor band. Complete processing is only shown for predictor p_iQ—quantizer; REC—reconstruction of last quantized value. (Note that the predictor control operates on all predictors P_i, . . . , P_j, . . . , P_k of a scale factor band and is followed by a second control over all scale factor bands.)

4.3 Predictor Control

In order to guarantee that prediction is only used if this results in a coding gain, an appropriate predictor control is required, and a small amount of predictor control information has to be transmitted to the decoder. For the predictor control, the predictors are grouped into scale factor bands. The predictor control information for each frame is determined in two steps. First, for each scale factor band one determines whether or not prediction gives a coding gain, and all predictors belonging to a scale factor band are switched on or off accordingly. Then one determines whether the overall coding gain by prediction in the current frame compensates at least the additional bits needed for the predictor side information. Only in this case, prediction is activated and the side information is transmitted. Otherwise prediction is not used in the current frame, and only one signaling bit is transmitted.

In order to increase the stability of the predictors and to allow defined entry points in the bit stream, a cyclic reset mechanism is applied in the encoder and decoder, in which all predictors are initialized again during a certain time interval in an interleaved way. The whole set of predictors is subdivided into 30 so-called reset groups (group 1: P_1, P_{31}, P_{61}, . . . ; group 2: P_2, P_{32}, P_{62}, . . . ; group 30: P_{30}, P_{60} . . .), which are then periodically reset, one after the other with a certain spacing. For example, if one group is reset every eighth frame, then all predictors are reset within an interval of $8 \times 30 = 240$ frames. The reset mechanism is controlled by a reset on–off bit, which always has to be transmitted as soon as prediction is enabled, and a conditional 5-bit index specifying the group of predictors to be reset.

In case of short windows prediction is always disabled and a full reset, that is, all predictors at once, is carried out.

4.4 Coding Gain

The various listening tests during the development phase of the standard have shown that significant improvement in sound quality—up to one grade on the ITU-R five-grade impairment scale—is achieved by prediction for stationary signals, such as pitch pipe or harpsichord.

5 QUANTIZATION AND CODING

While all previous steps perform some kind of preprocessing of audio data, the real bit-rate reduction is achieved during the quantization process. The primary goal of this module is to quantize the spectral data in such a way that the quantization noise fulfills the demands of the psychoacoustic model. At the same time the number of bits needed to code this quantized spectrum must be below a certain limit, normally the average number of bits available for a block of audio data. This value depends on the sampling frequency and, of course, on the desired bit rate. In AAC a bit reservoir gives the possibility of influencing the bit distribution between consecutive audio blocks on a short-time basis. These two constraints, fulfilling the demands of the psychoacoustic model on the one hand and keeping the number of needed bits below a certain number on the other, reveal the main problem of the quantization process: What can be done when the demands cannot be fulfilled with the available number of bits? What should be done if not all bits are needed to meet the requirements?

There is no standardized strategy for gaining optimum quantization, the only requirement being that the bit stream produced be AAC compliant [9]. One possible strategy is using two nested iteration loops, as described in this paper. This technique was used for the formal AAC test (see also Section 9). Other strategies are also possible. One important issue, however, is the tuning between the psychoacoustic model and the quantization process, which may be regarded as one of the "secrets" of audio coding, since it requires a great deal of experience and know-how.

The main features of the AAC quantization process are:

- Nonuniform quantization
- Huffman coding of the spectral values using different tables
- Noise shaping by amplification of groups of spectral values, so-called scale factor bands. (The information about the amplification is stored in the scale factor values.)
- Huffman coding of differential scale factors.

5.1 Nonuniform Quantization

The nonuniform quantizer used in AAC is described as follows (see also [1]):

$$ix(i) = \text{sign}[xr(i)] \cdot \text{nint}\left\{\left[\frac{|xr(i)|}{\sqrt[4]{2}^{\text{quantizer\_stepsize}}}\right]^{0.75} - 0.0946\right\}.$$

where nint means nearest integer value.

The main advantage of the nonuniform quantizer is the built-in noise shaping depending on the coefficient amplitude. The increase of the signal-to-noise ratio with rising signal energy is much lower than in a linear quantizer. The range of quantized values is limited to ±8191. Quantizer_stepsize represents the global quantizer step size. Thus the quantizer may be changed in steps of 1.5 dB.

5.2 Coding of Quantized Spectral Values

The quantized coefficients created by the quantizer are encoded using Huffman codes. A highly flexible coding method allows the use of several Huffman tables for one spectrum. Two- and four-dimensional tables

(signed or unsigned) are available. The coding process is described in detail in Section 6. To calculate the number of bits needed to encode a spectrum of quantized data, the coding process has to be performed and the number of bits needed for the spectral data and the side information has to be accumulated.

5.3 Noise Shaping

The use of a nonuniform quantizer is, of course, not sufficient to fulfill psychoacoustic demands. An additional method for shaping the quantization noise is required. AAC uses individual amplification of groups of spectral coefficients, the so-called scale factor bands. In order to fulfill the requirements as efficiently as possible, it is desirable to be able to shape the quantization noise in units similar to the critical bands of the human auditory system. Since the AAC system offers a relatively high frequency resolution for long blocks of 23.43 Hz per line at 48-kHz sampling frequency, it is possible to build groups of spectral values which reflect the bandwidth of the critical bands very closely. Fig. 9 shows the width of the scale factor bands for long blocks. (For several reasons the width of the scale factor bands is limited to 32 coefficients except for the last scale factor band.) The total number of scale factor bands for long blocks is 49.

The AAC system now offers the possibility to amplify the scale factor bands individually in increments of 1.5 dB. The noise shaping is achieved because amplified coefficients have larger amplitudes. They will therefore generally obtain a higher signal-to-noise ratio after quantization. On the other hand, larger amplitudes normally need more bits to be encoded, that is, the bit distribution between the scale factor bands is changed.

The inverse amplification has to be applied in the decoder. For this reason the amplification information, stored in the scale factors (in units of 1.5-dB steps), must also be transmitted to the decoder.

5.4 Encoding of Scale Factors

The first scale factor represents the global quantizer step size and is encoded in a PCM value called global_gain. All following scale factors are differentially encoded using a special Huffman code. This is described in detail in Section 6.

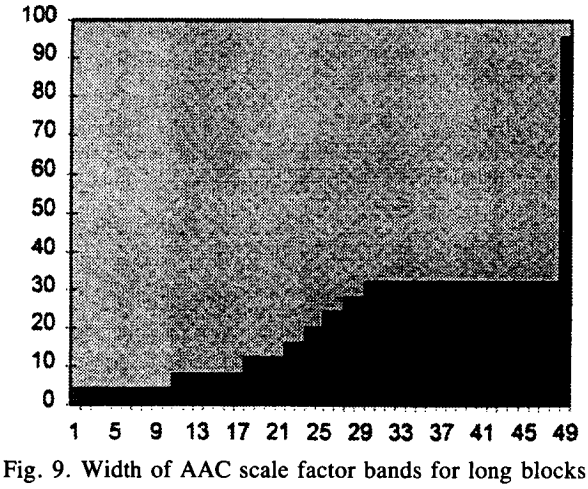


Fig. 9. Width of AAC scale factor bands for long blocks.

5.5 Iteration Process

The decision as to which scale factor band has to be amplified is, within certain limits, left up to the encoder. The thresholds calculated by the psychoacoustic model are of course the most important criteria, but not the only ones, since only a limited number of bits may be used. As mentioned, the iteration process described here is only one method for performing the noise shaping. This method is, however, known to produce very good audio quality. Two nested loops, the so-called inner and outer iteration loops, are used for determining optimum quantization. The description given here was simplified to facilitate understanding of the process.

5.5.1 *Inner Iteration Loop*

The task of the inner iteration loop is changing the quantizer step size until the given spectral data can be encoded with the number of available bits. For that purpose an initial quantizer step size is chosen, the spectral data are quantized, and the number of bits necessary to encode the quantized data is counted. If this number is higher than the number of available bits, the quantizer step size is increased, and the whole process is repeated. The inner iteration loop is shown in Fig. 10.

5.5.2 *Outer Iteration Loop*

The task of the outer iteration loop is amplifying the scale factor bands in such a way that the demands of the psychoacoustic model are fulfilled as far as possible.

1) At the beginning, no scale factor band is amplified.
2) The inner loop is called.
3) For each scale factor band, the distortion caused by the quantization is calculated (analysis by synthesis).
4) The actual distortion is compared with the permitted distortion calculated via the psychoacoustic model.
5) If this result is the best result so far, it is stored.

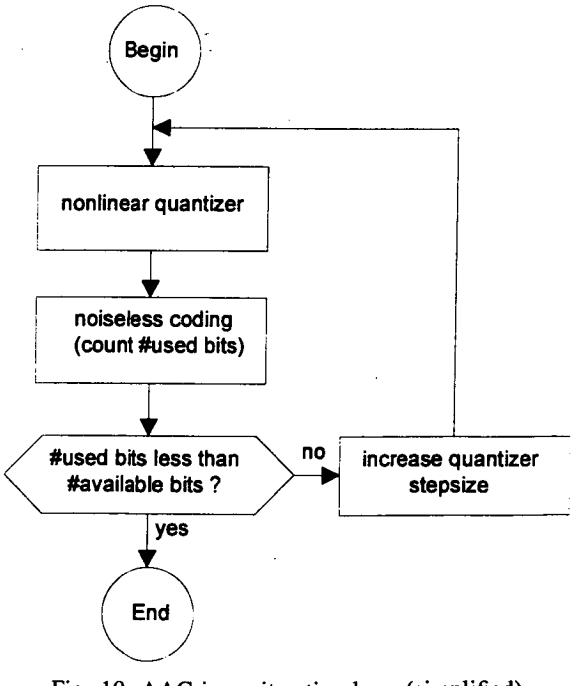


Fig. 10. AAC inner iteration loop (simplified).

This is important, since the iteration process does not necessarily converge.

6) Scale factor bands with an actual distortion higher than the permitted distortion are amplified. At this point, different methods for determining the scale factor bands that are to be amplified can be applied.

7) If all scale factor bands were amplified, the iteration process stops. The best result is restored.

8) If there is no scale factor band with an actual distortion above the permitted distortion, the iteration process will stop as well.

9) Otherwise the process will be repeated with the new amplification values.

There are some other conditions not mentioned here which cause a termination of the outer iteration loop. If the amplified parts of the spectrum need more bits for encoding while the number of available bits is constant, the quantizer step size has to be changed in the inner iteration loop to decrease the number of bits used. This mechanism shifts bits from spectral regions where they are not required to those where they are required. For the same reason the result after an amplification in the outer loop may be worse than before, so that the best result has to be restored after termination of the iteration process. The outer iteration loop is shown in Fig. 11.

The quantization and encoding process for short blocks is similar to that for long blocks, but grouping and interleaving must be taken into account. Both mechanisms are described in more detail in Section 6.

6 NOISELESS CODING

The input to the noiseless coding module is the set of 1024 quantized spectral coefficients. As a first step a method of noiseless dynamic range compression may be applied to the spectrum. Up to four coefficients can be coded separately as magnitudes in excess of 1, with a value of ± 1 left in the quantized coefficient array to carry the sign. The "clipped" coefficients are coded as integer magnitudes and an offset from the base of the coefficient array to mark their location. Since the side information for carrying the clipped coefficients costs some bits, this noiseless compression is applied only if it results in a net saving of bits.

6.1 Sectioning

The noiseless coding segments the set of 1024 quantized spectral coefficients into sections, such that a single

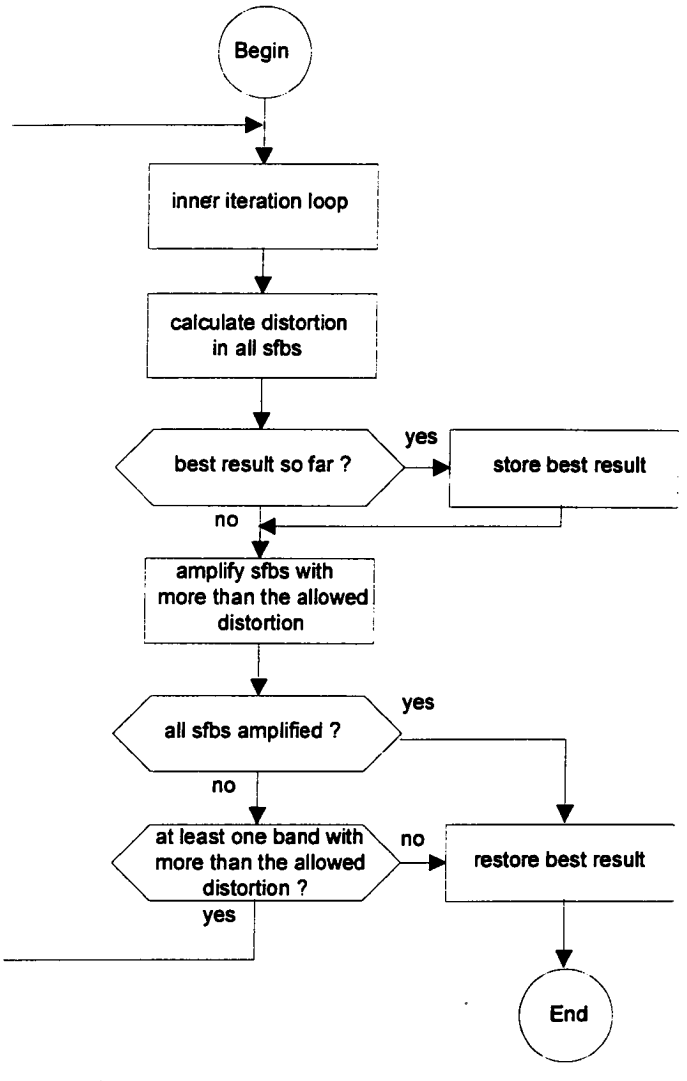


Fig. 11. AAC outer iteration loop (simplified). sfbs—scale factor bands.

Huffman codebook is used to code each section. (The method of Huffman coding is explained in a later section.) For reasons of coding efficiency, section boundaries can only be at scale factor band boundaries so that for each section of the spectrum one must transmit the length of the section, in scale factor bands, and the Huffman codebook number used for the section.

Sectioning is dynamic and typically varies from block to block, such that the number of bits needed to represent the full set of quantized spectral coefficients is minimized. This is done using a greedy merge algorithm starting with the maximum possible number of sections, each of which uses the Huffman codebook with the smallest possible index. Sections are merged if the resulting merged section results in a lower total bit count, with merges that yield the greatest bit count reduction done first. If the sections to be merged do not use the same Huffman codebook, then the codebook with the higher index must be used.

Sections often contain only coefficients whose value is zero. For example, if the audio input is band-limited to 20 kHz or lower, then the highest coefficients are zero. Such sections are coded with Huffman codebook zero, which is an escape mechanism that indicates that all coefficients are zero and it does not require that any Huffman codewords be sent for that section.

6.2 Grouping and Interleaving

If the window sequence is composed of eight short windows, then the set of 1024 coefficients is actually a matrix of 8 × 128 frequency coefficients representing the time–frequency evolution of the signal over the duration of the eight short windows. Although the sectioning mechanism is flexible enough to represent the eight zero sections efficiently, grouping and interleaving provide for greater coding efficiency. As explained earlier, the coefficients associated with contiguous short windows can be grouped such that they share scale factors among all scale factor bands within the group. In addition, the coefficients within a group are interleaved by interchanging the order of scale factor bands and windows. To be specific, assume that before interleaving the set of 1024 coefficients c is indexed as follows:

$$c[g][w][b][k] \tag{13}$$

where

g = index on groups
w = index on windows within a group
b = index on scale factor bands within a window
k = index on coefficients within a scale factor band

and the rightmost index varies most rapidly. After interleaving, the coefficients are indexed as

$$c[g][b][w][k] . \tag{14}$$

This has the advantage of combining all zero sections due to band-limiting within each group.

6.3 Scale Factors

The coded spectrum uses one quantizer per scale factor band. The step size of each of these quantizers is specified as a set of scale factors and a global gain that normalizes these scale factors. In order to increase compression, scale factors associated with scale factor bands that have only zero-valued coefficients are not transmitted. Both the global gain and the scale factors are quantized in 1.5-dB steps. The global gain is coded as an 8-bit unsigned integer, and the scale factors are differentially encoded relative to the previous scale factor (or global gain for the first scale factor) and then Huffman coded. The dynamic range of the global gain is sufficient to represent full-scale values from a 24-bit PCM audio source.

6.4 Huffman Coding

Huffman coding is used to represent n-tuples of quantized coefficients, with the Huffman code drawn from one of 12 codebooks. The spectral coefficients within n-tuples are ordered (low to high) and the n-tuple size is two or four coefficients. The maximum absolute value of the quantized coefficients that can be represented by each Huffman codebook and the number of coefficients in each n-tuple for each codebook are shown in Table 3. There are two codebooks for each maximum absolute value, with each representing a distinct probability distribution function. The best fit is always chosen. In order to save on codebook storage (an important consideration in a mass-produced decoder), most codebooks represent unsigned values. For these codebooks the magnitude of the coefficients is Huffman coded and the sign bit of each nonzero coefficient is appended to the codeword.

Two codebooks require special notes—codebooks 0 and 11. As mentioned previously, codebook 0 indicates that all coefficients within a section are zero. Codebook 11 can represent quantized coefficients that have an absolute value greater than or equal to 16. If the magnitude of one or both coefficients is greater than or equal to 16, a special escape coding mechanisms is used to represent those values. The magnitude of the coefficients is limited to no greater than 16, and the corresponding 2-tuple is Huffman coded. The sign bits are appended to the codeword as needed. For each coefficient magnitude

Table 3. Huffman codebooks.

Codebook Index	n-Tuple Size	Maximum Absolute Value	Signed Values
0		0	
1	4	1	Yes
2	4	1	Yes
3	4	2	No
4	4	2	No
5	2	4	Yes
6	2	4	Yes
7	2	7	No
8	2	7	No
9	2	12	No
10	2	12	No
11	2	16 (ESC)	No

greater than or equal to 16, an escape code is also appended, as follows:

$$\text{escape code} = \langle\text{escape\_prefix}\rangle\langle\text{escape\_separator}\rangle\langle\text{escape\_word}\rangle \quad (15)$$

where

⟨escape_prefix⟩ = sequence of N binary 1's
⟨escape_separator⟩ = binary 0
⟨escape_word⟩ = $(N + 4)$-bit unsigned integer, most significant bit first

and N is a count that is just large enough so that the magnitude of the quantized coefficient is equal to

$$2^{(N+4)} + \langle\text{escape\_word}\rangle . \quad (16)$$

7 BIT-STREAM MULTIPLEXING

The MPEG-2 AAC system has a very flexible bit-system syntax. Two layers are defined—the lower specifies the raw audio information while the higher specifies a specific audio transport mechanism. Since any one transport cannot be appropriate for all applications, the raw-data layer is designed to be parsable on its own, and in fact is entirely sufficient for applications such as compression to computer storage devices.

The composition of a bit stream is given in Table 4. Here tokens in the bit stream are indicated by angle brackets ⟨ ⟩. The bit stream is indicated by the token ⟨stream⟩ and is a series of ⟨block⟩ tokens, each containing all information necessary to decode 1024 audio-frequency samples. Furthermore each ⟨block⟩ token begins on a byte boundary relative to the start of the first ⟨block⟩ in the bit stream. Between ⟨block⟩ tokens there may be transport information, indicated by ⟨transport⟩, such as would be needed for synchronization on break-in or for error control. Braces { } indicate an optional token, and brackets [ ] indicate that the token may appear zero or more times.

Since the AAC system has a bit buffer that permits its instantaneous data rate to vary as required by the audio signal, the length of each ⟨block⟩ is not constant. In this respect the AAC bit stream uses variable-rate headers (a header being the ⟨transport⟩ token). These headers are byte-aligned so as to permit editing of bit streams at any block boundary. Tokens within a ⟨block⟩ can be as shown in Table 5.

7.1 Bit-Stream Elements

The prog_config_ele is a configuration element that specifies the audio channel to output loudspeaker assignment so that multichannel coding can be as flexible as possible. It can specify the correct voice tracks for multilingual programming and specifies the analog sampling rate.

There are three possible audio elements—single_channel_ele is a monophonic audio channel, channel_pair_ele is a stereo pair, and low_freq_effects_ele is a subwoofer channel. Each of the audio elements is named with a 4-bit tag such that up to 16 of any one element can be represented in the bit stream and assigned to a specific output channel. At least one audio element must be present.

The coupling_ele is a mechanism to code signal components common to two or more audio channels (see Section 8.2).

The data_ele is a tagged data stream that can continue over an arbitrary number of blocks. Unlike other elements, the data element contains a length count such that an audio decoder can strip it from the bit stream without knowledge of its meaning. As with the audio elements, up to 16 distinct data streams are supported.

The fill_ele is a bit-stuffing mechanism that enables an encoder to increase the instantaneous rate of the compressed audio stream such that it fills a constant-rate channel. Such mechanisms are required as, first, the encoder has a region of convergence for its target bit allocation so that the bits used may be less than the bit budget, and second, the encoder's representation of a digital zero sequence is so much less than the average

Table 4. Composition of bit stream.

⟨stream⟩	{⟨transport⟩}⟨block⟩{⟨transport⟩}⟨block⟩ . . .
⟨block⟩	[⟨prog_config_ele⟩]⟨audio_ele⟩[⟨audio_ele⟩][⟨coupling_ele⟩][⟨data_ele⟩][⟨fill_ele⟩]⟨term_ele⟩

Table 5. Tokens within a ⟨block⟩.

Token	Meaning
prog_config_ele	Program configuration element
audio_ele	Audio element, one of:
single_channel_ele	Single channel
channel_pair_ele	Stereo pair
low_freq_effects_ele	Low-frequency-effects channel
coupling_ele	Multichannel coupling
data_ele	Data element, segment of data stream
fill_ele	Fill element, adjusts data rate for constant-rate channels
term_ele	Terminator, signals end of block

coding bit budget that it must resort to bit stuffing.

The term_ele signals the end of a block. It is mandatory as this makes the bit stream parsable. Padding bits may follow term_ele such that the next ⟨block⟩ begins on a byte boundary.

An example of one ⟨block⟩ for a 5.1-channel bit stream (where the .1 indicates the LFE channel) is

⟨block⟩ ⟨single_channel_ele⟩⟨channel_pair_ele⟩⟨channel_pair_ele⟩⟨low_freq_effects_ele⟩⟨term_ele⟩.

Although discussion of the syntax of each element is beyond the scope of this paper, all elements make frequent use of conditional components. This increases flexibility while keeping bit-stream overhead to a minimum. For example, a 1-bit field indicates whether prediction is used in an audio channel in a given block. If set to 1, then the set of bits indicating which scale factor bands use prediction follows. Otherwise the bits are not sent. For additional information see [9].

8 ADDITIONAL FEATURES

In addition to the basic building blocks of perceptual audio coding, MPEG-2 AAC takes advantage of a temporal noise-masking technique and stereo and multichannel techniques.

8.1 Temporal Noise Shaping

A novel concept in perceptual audio coding is represented by the temporal noise-shaping (TNS) tool of the AAC system [10]. This tool is motivated by the fact that despite the advanced state of today's perceptual audio coders, the handling of transient and pitched input signals still presents a major challenge. This is mainly due to the problem of maintaining the masking effect in the reproduced audio signal under all conditions. In particular, coding is difficult because of the temporal mismatch between masking threshold and quantization noise (also known as the preecho problem [20]).

The TNS technique permits the coder to exercise control over the temporal fine structure of the quantization noise even within a filter-bank window. The concept of this technique can be outlined as follows:

- *Time–frequency duality considerations*: The concept of TNS uses the duality between time and frequency domains to extend the known predictive coding techniques by a new variant. It is well known that signals with an "unflat" spectrum can be coded efficiently either by directly coding spectral values (transform coding) or by applying predictive coding methods to the time signal [21]. Consequently the corresponding dual statement relates to the coding of signals with an "unflat" time structure, that is, transient signals. Efficient coding of transient signals can thus be achieved either by directly coding time-domain values or by applying predictive coding methods to the spectral data. Such a predictive coding of spectral coefficients over frequency in fact constitutes the dual concept of the intrachannel prediction tool described in Section 4. While intrachannel prediction over time increases the coder's spectral resolution, prediction over frequency enhances its temporal resolution.
- *Noise shaping by predictive coding*: If an open-loop (forward) predictive coding technique is applied to a time signal, the quantization error in the final decoded signal is known to be adapted in its power spectral density (PSD) to the PSD of the input signal [21]. Dual to this, if predictive coding is applied to spectral data over frequency, the temporal shape of the quantization error signal will appear adapted to the temporal shape of the input signal at the output of the decoder. This effectively puts the quantization noise under the actual signal and in this way avoids problems of temporal masking, in either transient or pitched signals. This type of predictive coding of spectral data is therefore referred to as the TNS method.

Since TNS processing can be applied either for the entire spectrum or for only part of the spectrum, the time-domain noise control can be applied in any necessary frequency-dependent fashion. In particular, it is possible to use several predictive filters operating on distinct frequency (coefficient) regions.

8.1.1 Implementation in AAC Encoder and Decoder

The predictive encoding and decoding process over frequency can be realized easily by adding one building block to the standard structure of a generic perceptual encoder and decoder. This is shown for the encoder in Figs. 12 and 13. Immediately after the analysis filter bank an additional block, "TNS filtering," is inserted, which performs an in-place filtering operation on the spectral values, that is, it replaces the target spectral coefficients (the set of spectral coefficients to which TNS should be applied) with the prediction residual. This is symbolized by a "rotating switch" circuitry as shown in Fig. 13. Sliding in the order of both increasing and decreasing frequency is possible.

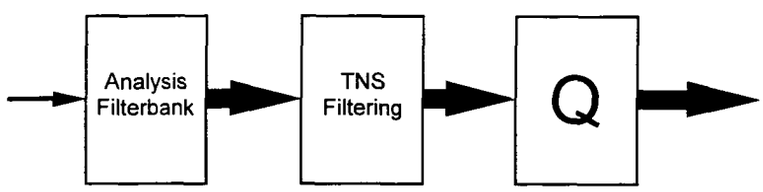


Fig. 12. TNS processing for MPEG-2 AAC encoder.

Similarly, the TNS decoding process is done by inserting an additional block, "inverse TNS filtering," immediately before the synthesis filter bank (see Figs. 14 and 15). An inverse in-place filtering operation is performed on the residual spectral values so that the target spectral coefficients are replaced with the decoded spectral coefficients by means of the inverse prediction (all-pole) filter.

The TNS operation is signaled to the decoder via a TNS on–off flag, the number (maximum 3) and the frequency range of the TNS filters applied in each transform window, the order of the prediction filter (maximum 12 or 20, depending on the profile), and the filter data itself.

8.1.2 Properties of TNS Processing

The properties of the TNS technique can be described as follows:

- The combination of filter bank and adaptive prediction filter can be interpreted as a continuously signal adaptive filter bank, as opposed to the classic "switched-filter-bank" approach. In fact, this type of adaptive filter bank dynamically provides a continuum in its behavior between a high-resolution filter bank (for stationary signals) and a low-resolution filter bank (for transient signals), thus approaching the optimum filter-bank structure for a given signal [22].
- The TNS approach permits a more efficient use of masking effects by adapting the temporal fine structure of the quantization noise to that of the masker (signal). In particular, it enables a better encoding of "pitch-based" signals such as speech, which consist of a pseudostationary series of impulselike events where traditional transform block switching schemes do not offer an efficient solution.
- The method reduces the peak bit demand of the coder for transient signal segments by exploiting irrelevancy. As a side effect, a coder using transform block switching can more often stay in the preferred "long-block" mode so that use of the critical "short-block" mode is minimized.
- The technique can be applied in combination with other methods addressing the TNS problem, such as transform block switching and preecho control.

During the standardization process of the MPEG-2 AAC system, the TNS tool demonstrated a significant increase in performance for speech stimuli. In particular, an improvement in quality of approximately 0.9 in the five-point

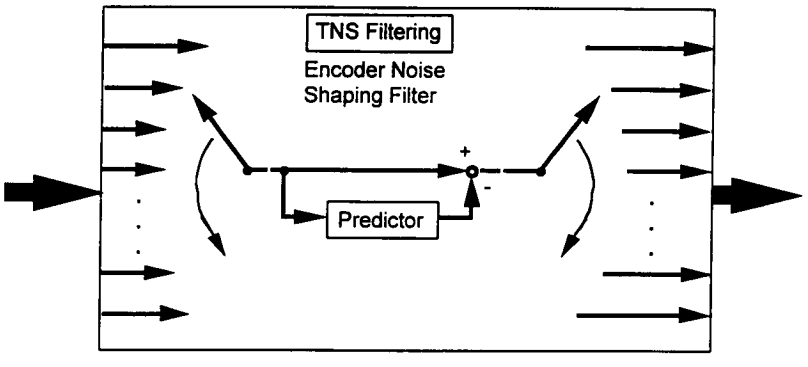


Fig. 13. Diagram of MPEG-2 AAC encoder TNS filtering stage.

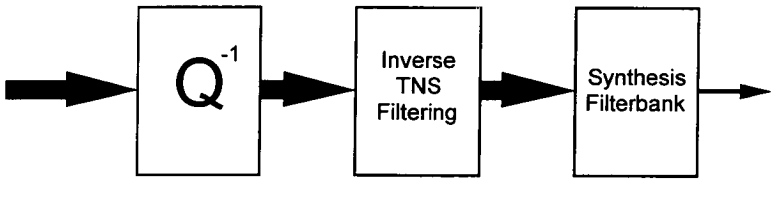


Fig. 14. TNS processing for MPEG-2 AAC decoder.

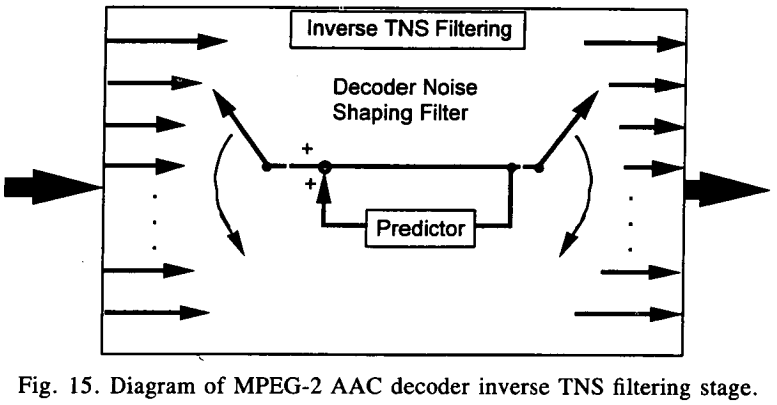


Fig. 15. Diagram of MPEG-2 AAC decoder inverse TNS filtering stage.

ITU-R impairment scale for the most critical speech item "German male speech" was shown during the AAC core experiments. Advantages were also shown for other transient signals (such as in the glockenspiel item).

8.2 Joint Stereo Coding

For further enhancement of its capabilities, the MPEG-2 AAC system includes two techniques for stereo coding of signals—mid/side (M/S) stereo coding (also known as sum–difference coding) and intensity stereo coding. Both stereo coding strategies can be combined by selectively applying them to different frequency regions. By using M/S stereo coding, intensity stereo coding, and L/R (independent) coding as appropriate, it is possible to avoid the expensive overcoding due to binaural masking level depression, account for noise imaging, and very frequently achieve a significant saving in data rate. The concept of joint stereo coding in the MPEG-2 AAC system is discussed in greater detail in [11].

8.2.1 M/S Stereo Coding

M/S stereo coding is used to control the imaging of coding noise, as compared to the imaging of the original signal. In particular, this technique is capable of addressing the issue of binaural masking level depression (BMLD) [23], [24], where a signal at lower frequencies (below 2 kHz) can show up to 20-dB difference in masking thresholds, depending on the phase of the signal and noise present (or lack of correlation in the case of noise). A second important issue is that of high-frequency time-domain imaging on transient or pitched signals. In either case, the properly coded stereo signal can require more bits than two transparently coded monophonic signals.

In the MPEG-2 AAC system, M/S stereo coding is applied within each channel pair of the multichannel signal, that is, between a pair of channels that are arranged symmetrically on the left–right listener axis. In this way imaging problems due to spatial unmasking are avoided to a large degree.

M/S stereo coding can be used in a flexible way by selectively switching in time (on a block-by-block basis), as well as in frequency (on a scale-factor-band by scale-factor-band basis); see [25]. The switching state (M/S stereo coding on or off) is transmitted to the decoder as an array of signaling bits (ms_used). This can accommodate short-time delays between the L and R channels, and still accomplish both image control and some signal-processing gain. While the amount of time delay that it allows is limited, the time delay is greater than the interaural time delay, and allows for control of the most critical imaging issues [25].

8.2.2 Intensity Stereo Coding

The second important stereo coding strategy for exploiting interchannel irrelevance is the well-known concept of intensity stereo coding [26], [27]. This idea has been widely utilized in the past for stereophonic and multichannel coding under various names (such as dynamic crosstalk [28] or channel coupling [29]). Intensity stereo coding exploits the fact that the perception of high-frequency sound components mainly relies on the analysis of their energy–time envelopes [23]. Thus it is possible for certain types of signals to transmit a single set of spectral values that is shared among several audio channels with virtually no loss in sound quality. The original energy–time envelopes of the coded channels are preserved approximately by means of a scaling operation such that each channel signal is reconstructed with its original level after decoding.

The MPEG-2 AAC system provides two mechanisms for applying intensity stereo coding:

1) The first is based on the channel-pair concept as used for M/S stereo coding and implements an easy-to-use straightforward coding concept that covers most of the normal needs without introducing noticeable signaling overhead into the bit stream. For simplicity, this mechanism is referred to as the AAC intensity stereo coding tool. While the intensity stereo coding tool only implements joint coding within each channel pair, it may be used for coding of two-channel as well as multichannel signals.

2) In addition, a second, more sophisticated, mechanism is available that is not restricted by the channel-pair concept and allows better control over the coding parameters. This mechanism is called the AAC coupling channel element and provides two functionalities. First, coupling channels may be used to implement generalized intensity stereo coding, where channel spectra can be shared across channel boundaries (including sharing among different channel pairs). The second functionality of the coupling channel element is to perform a downmix of additional sound objects into the stereo image so that, for example, a commentary channel can be added to an existing multichannel program (voice-over). Depending on the profile, certain restrictions apply regarding consistency between coupling channel and target channels in terms of window sequence and window shape parameters [9].

Thus the MPEG-2 AAC system provides appropriate coding tools for many types of stereophonic program material from traditional two-channel recordings to five- or seven-channel surround-sound material.

9 TEST RESULTS

Since the first submission of AAC proposals in 1994 November, a number of core experiments were planned and carried out to select the best performing tools to be incorporated in the AAC RM. The final MPEG-2 AAC system was tested according to the ITU-R BS.1116 specifications in 1996 September in the five-channel, full-bandwidth configuration and compared to the MPEG-2 BC layer 2 in the same configuration [30]. The formal subjective tests were carried out at BBC, UK, and NHK, Japan. A total of 23 reliable[3] expert listeners at BBC

[3] Due to the very rigorous test method adopted, only statistically reliable expert listeners were taken into consideration in the final data analysis. A total of 32 listeners at BBC and 24 listeners at NHK originally participated in the tests. After postscreening of the subjects, nine listeners at BBC and eight listeners at NHK were removed.

and 16 reliable expert listeners at NHK participated in the listening tests. As specified by ITU-R BS.1116, the tests were conducted according to the triple-stimulus/hidden-reference/double-blind method using the ITU-R five-point impairment scale (see Table 6).

From the 94 submitted critical excerpts, a selection panel selected the 10 most critical items (see Table 7).

The following systems were tested in the 1996 September formal tests:

1) MPEG-2 AAC main profile at 256 kbit/s per five full-bandwidth channels

2) MPEG-2 AAC main profile at 320 kbit/s per five full-bandwidth channels

3) MPEG-2 AAC low-complexity profile at 320 kbit/s per five full-bandwidth channels

4) MPEG-2 layer 2 BC at 640 kbit/s per five full-bandwidth channels.

The test results, in terms of nonoverlapping 95% confidence intervals for diffgrades as per ITU-R BS.1116 specifications [31] are shown in Figs. 16–19.[4]

The MPEG-2 AAC test results show that the AAC system at 320 kbit/s per five full-bandwidth channels fulfills the ITU-R requirements for indistinguishable quality [8] in a BS.1116 compliant test. Furthermore, according to the NHK data, AAC can achieve indistinguishable quality also at 256 kbit/s per five full-bandwidth channels.

Table 6. ITU-R five-grade impairment scale and corresponding diffgrades [31].

Impairment Description	ITU-R Grade	Diffgrade
Imperceptible	5.0	0.0
Perceptible, but not annoying	4.0	−1.0
Slightly annoying	3.0	−2.0
Annoying	2.0	−3.0
Very annoying	1.0	−4.0

Table 7. Critical items.

No.	Name	Description
1	Cast	Castanets panned across front, noise in surround
2	Clarinet	Clarinet in center front, theater foyer ambience, rain on windows in surround
3	Eliot	Female and male speech in a restaurant, chamber music
4	Glock	Glockenspiel and timpani
5	Harp	Harpsichord
6	Manc	Orchestra—strings, cymbals, drums, horns
7	Pipe	Pitch pipe
8	Station	Male voice with steam-locomotive effects
9	Thal	Piano front left, sax front right, female voice center
10	Tria	Triangle

[4] In Figs. 16–18 the vertical axes show AAC coded diffgrades minus reference signal grades. In Fig. 19 the vertical axis shows AAC coded diffgrades minus layer 2 BC coded diffgrades. A positive value indicates that the AAC codec was awarded a better diffgrade than the layer 2 BC codec, and vice versa. (See also Table 6.)

The AAC multichannel system at 320 kbit/s overall ranks higher than MPEG-2 layer 2 BC at 640 kbit/s (see Fig. 19). In particular, the difference between the two systems' mean scores for the pitch pipe excerpt is more than 1.5 points in the ITU-R five-point impairment scale according to the BBC data. It should be noted that the test data for MPEG-2 layer 2 BC at 640 kbit/s were consistent with data obtained in previously conducted subjective tests [6].

10 DECODER COMPLEXITY EVALUATION

In this section a complexity evaluation of the decoding process in its main and LC profile configuration is presented. In order to quantify the complexity of the AAC decoder, the number of machine instructions, read–write storage locations (RAM), and read-only storage locations (ROM) will be specified for each module (see also [32]). For simplicity, the assumption is made that the audio signal is sampled at 48-kHz, 16 bit per sample, the data rate is 64 kbit/s per channel, and there are 1024 frequency values per block.

Two categories of the AAC decoder implementation are considered: software decoders running on general-purpose processors, and hardware decoder running on single-chip ASICs. For these two categories, a summary of the AAC decoder complexity is shown in Table 8.

10.1 Input–Output Buffers

Considering the bit reservoir encoder structure and the maximum data rate per channel, the minimum decoder input buffer size is 6144 bit. The decoder output, assuming a 16-bit PCM double-buffer system, requires a 1024, 16-bit word buffer. The total number of 16-bit words for the decoder input–output buffer (RAM) is

$$384 + 1024 = 1408 . \quad (17)$$

10.2 Huffman Coding

In order to decode a Huffman codeword, the decoder must traverse a Huffman code tree from root node to leaf node. Approximately 10 instructions per bit are required for the Huffman decoding. Given the average of 1365 bits per blocks, the number of instructions per block is 13 653. Huffman decoding requires the storage of the tree and the value corresponding to the codeword. The total buffer required is a 995 16-bit-word buffer (ROM).

10.3 Inverse Quantization

The inverse quantization can be done by table lookup. Assuming that only 854 spectral coefficients (20-kHz bandwidth) must be inverse quantized and scaled by a scale factor, the 16-bit-word ROM buffer is 256 words, and the total number of instructions is 1708.

10.4 Prediction

Assuming that only the first 672 spectral coefficients will use prediction and the predictor used is a second-order predictor, the number of instructions for the pre-

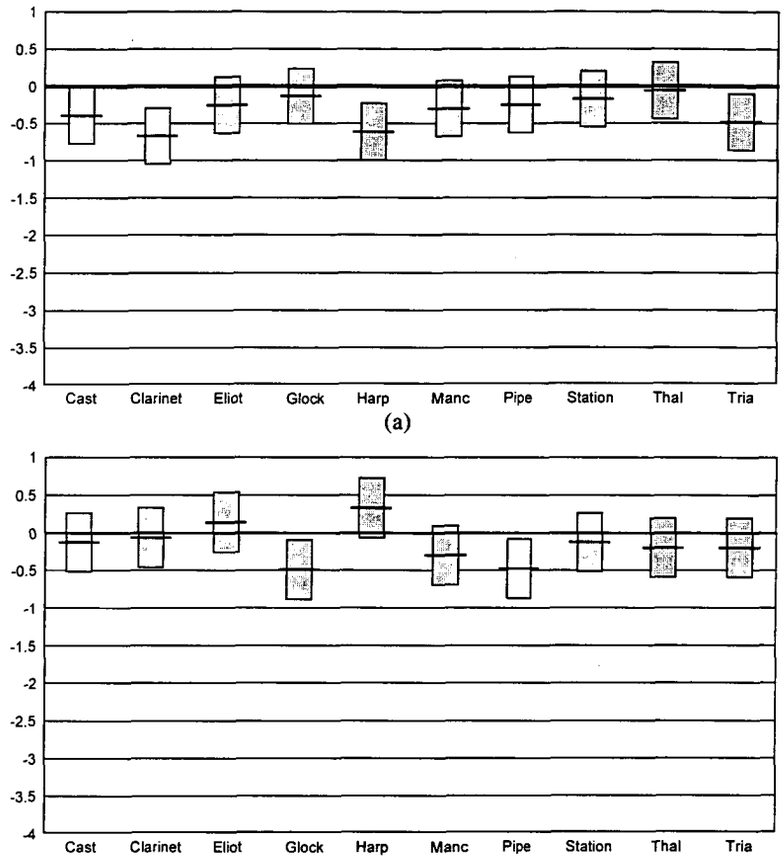


Fig. 16. Results of formal tests for MPEG-2 AAC main profile at 320 kbit/s, five-channel configuration, from [30]. (a) BBC results. (b) NHK results.

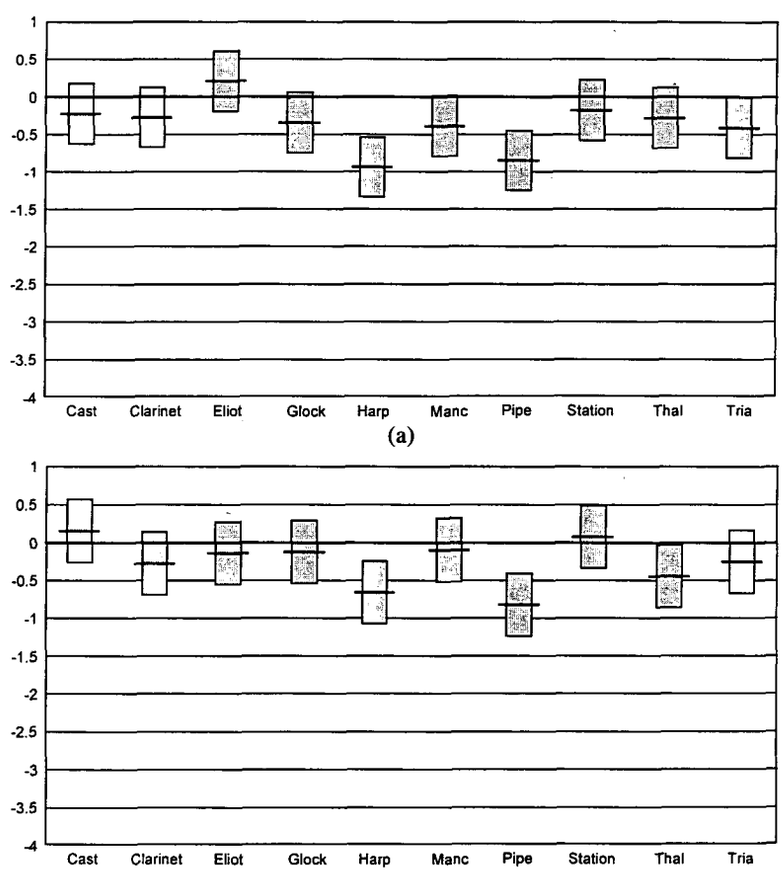


Fig. 17. Results of formal tests for MPEG-2 AAC LC profile at 320 kbit/s, five-channel configuration, from [30]. (a) BBC results. (b) NHK results.

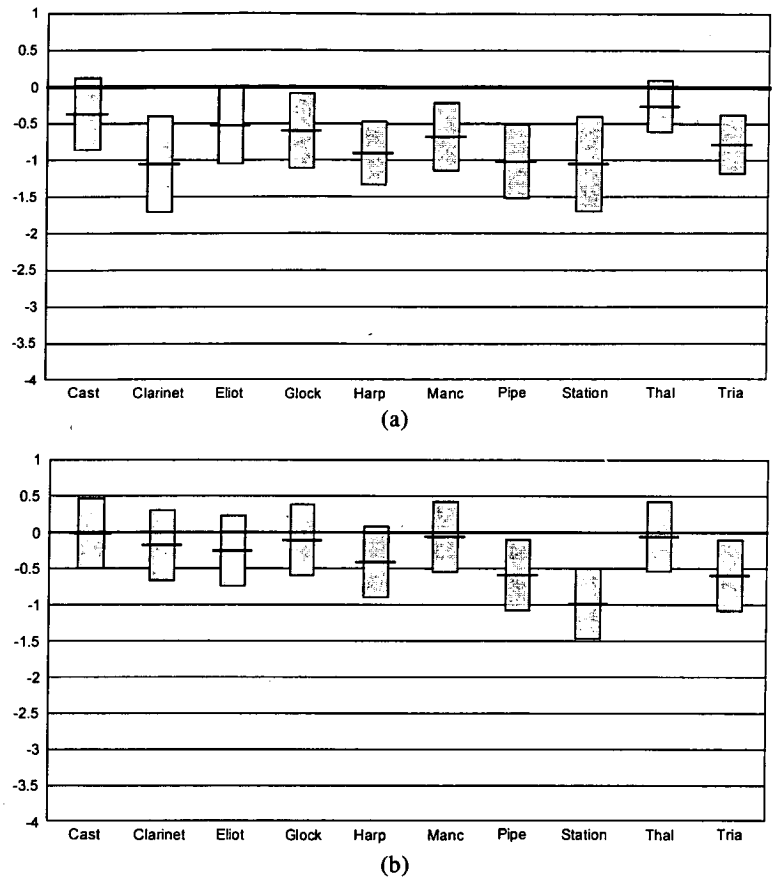


Fig. 18. Results of formal tests for MPEG-2 AAC main profile at 256 kbit/s, five-channel configuration, from [30]. (a) BBC results. (b) NHK results.

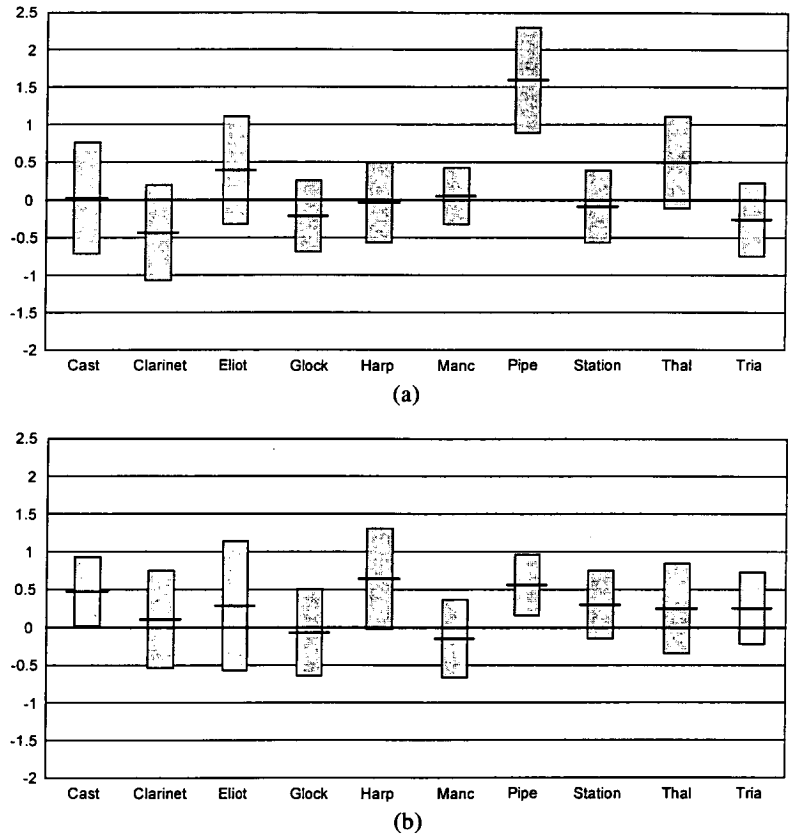


Fig. 19. Comparison between MPEG-2 AAC at 320 kbit/s and MPEG-2 BC layer 2 at 640 kbit/s, five-channel configuration, from [30]. (a) BBC results. (b) NHK results.

dictor is 66, and the total number of instructions per block is 44 352. Calculations can be made in both IEEE floating-point and fixed arithmetic, and variables are truncated to 16 bit prior to storage (see also [33], [34]). The required storage buffer is 4032, 16-bit word.

10.5 TNS

In the main profile configuration, the TNS process employs a filter on the order of 20 operating on 672 spectral coefficients. The number of instructions per block is 13 630. In the LC profile configuration the TNS process employs a filter of reduced order 12 with a total number of instructions per block of 8130. TNS requires negligible storage buffers.

10.6 M/S Stereo

This is a very simple module, which performs matrixing on two channels of a stereo pair element. Since the computation is done in place, no additional storage is required. Assuming that only a 20-kHz bandwidth needs the M/S computation, the total number of instructions per stereo pair is 854.

10.7 Intensity Stereo

Intensity stereo coding does not use any additional read-only or read–write storage. The net complexity of intensity stereo coding produces a saving of one inverse quantization per intensity stereo coded coefficient.

10.8 Inverse Filter Bank

The IMDCT of length 1024 requires about 20 000 instructions per block, while for the 128-point IMDCT the total number of instructions per block is 24 576. The total RAM for the filter bank is 1536 words, the total ROM, including window coefficients, and so on, is 2270 words. The storage requirement employs a word length of between 16 and 24 bit, depending on the stage of the filter bank.

10.9 Summary

Tables 9–11 summarize the complexity of each decoder module based on the number of instructions per block (Table 9), the amount of read–write storage, and the amount of read-only (Tables 10 and 11) in 16-bit words. The tables list complexity on a per-channel basis and for a five-channel coder. Table 12 shows a complexity comparison between the MPEG-2 AAC main profile and the LC profile.

11 CONCLUSIONS

The ISO/IEC MPEG-2 AAC (ISO/IEC 13818-7) system was designed to provide MPEG-2 with the best audio quality without any restrictions due to compatibility requirements. The AAC tools provide high coding efficiency through the use of a high-resolution filter bank, prediction techniques, noiseless coding, and added functionalities. ITU-R BS.1116 compliant tests have clearly shown that the AAC system achieves indistinguishable audio quality at data rates of 320 kbit/s for five full-bandwidth channels. While MPEG-4 audio will be the future multimedia standard, AAC will play an important role in this context. According to the current

Table 9. Summary of MPEG-2 AAC main profile decoder complexity—number of instructions per block.

AAC Tool	Single Channel	Five Channels
Huffman coding	13 657	68 285
Inverse quantization	1 708	8 540
Prediction	44 352	221 760
TNS	13 850	69 250
M/S		1 708
IMDCT	24 576	122 880
Total	98 143	492 423

Table 10. Summary of MPEG-2 AAC main profile decoder complexity—RAM (in 16-bit words).

	Single Channel	Five Channels
Input buffer	384	1 920
Output buffer	1024	5 120
Working buffer	2048	10 240
Prediction	4032	20 160
IMDCT	1024	5 120
Total	8512	42 560

Table 11. Summary of MPEG-2 AAC main profile decoder complexity—ROM.

	Single Channel	Five Channels
Huffman coding	—	995
Inverse quantization	—	256
TNS	—	24
Prediction	—	0
IMDCT		2270
Total		3545

Table 12. Summary of MPEG-2 AAC main profile and LC profile decoder complexity (five-channel configuration only).

	Main Profile	LC Profile
Instructions per block	492 423	242 063
RAM	42 560	22 400
ROM	3 545	3 545

Table 8. Summary of MPEG-2 AAC main profile decoder complexity.

MPEG-2 AAC Decoder	Complexity
2-channel main profile software decoder	40% of 133-MHz Pentium
2-channel LC profile software decoder	25% of 133-MHz Pentium
5-channel main profile hardware decoder	90-mm^2 die, 0.5-μm CMOS
5-channel LC profile hardware decoder	60-mm^2 die, 0.5-μm CMOS

plans of ISO/IEC MPEG, AAC will be the last, very high-quality audio standard for the foreseeable future. We anticipate that the MPEG-2 AAC standard will become the audio coding system of choice in applications where high performance at the lowest possible data rate is critical to the success of an application.

12 ACKNOWLEDGMENT

A great many people made this project a reality. The authors would like to express their gratitude to the MPEG Audio Committee and Leonardo Chiariglione, the Convenor of MPEG, for the vision and the support provided. Thanks are also due to J. Johnston, M. Davis, C. Robinson, S. Forshay, U. Gbur, B. Teichmann, O. Kunz, J. Hilpert, S. Kölbl, J. Koller, N. Jayant, and the many other contributors to the standards effort.

13 REFERENCES

[1] ISO/IEC 11172-3, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s, Part 3: Audio" (1992).

[2] K. Brandenburg and G. Stoll, "The ISO/MPEG-1 Audio: A Generic Standard for Coding of High-Quality Digital Audio," *J. Audio Eng. Soc.*, vol. 42, pp. 780–792 (1994 Oct.).

[3] ISO/IEC JTC1/SC29/WG11 MPEG 91/010, "The MPEG/AUDIO Subjective Listening Test," Stockholm, Sweden (1991 Apr./May).

[4] ITU-R Recommendation BS.1115, "Low Bitrate Audio Coding," Geneva, Switzerland (1994).

[5] ISO/IEC 13818-3, "Information Technology—Generic Coding of Moving Pictures and Associated Audio, Part 3: Audio" (1994–1997).

[6] ISO/IEC JTC1/SC29/WG11 N1229, "MPEG-2 Backwards Compatible CODECS Layer II and III: RACE dTTb Listening Test Report," Florence, Italy (1996 Mar.).

[7] K. Brandenburg and M. Bosi, "Overview of MPEG Audio: Current and Future Standards for Low-Bit-Rate Audio Coding," *J. Audio Eng. Soc.*, vol. 45, pp. 4–21 (1997 Jan./Feb.).

[8] ITU-R Document TG10-2/3- E only, "Basic Audio Quality Requirements for Digital Audio Bit-Rate Reduction Systems for Broadcast Emission and Primary Distribution" (1991 Oct. 28).

[9] ISO/IEC 13818-7, "Information Technology—Generic Coding of Moving Pictures and Associated Audio, Part 7: Advanced Audio Coding" (1997).

[10] J. Herre and J. D. Johnston, "Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS)," presented at the 101st Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 44, p. 1175 (1996 Dec.), preprint 4384.

[11] J. D. Johnston, J. Herre, M. Davis, and U. Gbur, "MPEG-2 NBC Audio—Stereo and Multichannel Coding Methods," presented at the 101st Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 44, p. 1175 (1996 Dec.), preprint 4383.

[12] E. Zwicker and H. Fastl, *Psychoacoustic, Facts and Models* (Springer, New York, 1990).

[13] M. Bosi, C. Todd, and T. Holman, "Aspects of Current Standardization Activities for High-Quality, Low Rate Multichannel Audio Coding," in *Proc. 1993 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (New Paltz, NY, 1993 Oct.), paper 2a.3.

[14] ISO/IEC JTC1/SC29/WG11 N1623, "Informal Assessment of AAC Downmix Stereo Performance," Bristol, UK (1997 Apr.).

[15] J. P. Princen, A. W. Johnson, and A. B. Bradley, "Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing* (Dallas, TX, 1987), pp. 2161–2164.

[16] L. Fielder, M. Bosi, G. Davidson, M. Davis, C. Todd, and S. Vernon, "AC-2 and AC-3: Low-Complexity Transform-Based Audio Coding," in *Collected Papers on Digital Audio Bit-Rate Reduction*, N. Gilchrist and C. Grewin, Eds. (Audio Engineering Society, New York, 1996), pp. 54–72.

[17] G. A. Davidson and M. Bosi, "AC-2: High Quality Audio Coding for Broadcasting and Storage," in *Proc. 46th Ann. Broadcast Eng. Conf.* (Las Vegas, NV, 1992 Apr.), pp. 98–105.

[18] B. Edler, "Coding of Audio Signals with Overlapping Block Transform and Adaptive Window Functions" (in German), *Frequenz*, vol. 43, pp. 252–256 (1989).

[19] H. Fuchs, "Improving MPEG Audio Coding by Backward Adaptive Linear Stereo Prediction," presented at the 99th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 43, p. 1087 (1995 Dec.), preprint 4086.

[20] J. D. Johnston and K. Brandenburg, "Wideband Coding Perceptual Considerations for Speech and Music," in *Advances in Speech Signal Processing*, S. Furui and M. M. Sondhi, Eds. (Marcel Dekker, New York, 1992).

[21] N. Jayant and P. Noll, *Digital Coding of Waveforms* (Prentice-Hall, Englewood Cliffs, NJ, 1984).

[22] J. Herre and J. Johnston, "A Continuously Signal-Adaptive Filterbank for High-Quality Perceptual Audio Coding," presented at the IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, Mohonk, New Paltz, NY (1997).

[23] J. Blauert, *Spatial Hearing* (MIT Press, Cambridge, MA, 1983).

[24] B. C. J. Moore, *Introduction to the Psychology of Hearing*, 3rd ed. (Academic Press, New York, 1989).

[25] J. D. Johnston and A. J. Ferreira, "Sum-Difference Stereo Transform Coding," in *Proc. IEEE ICASSP* (1992), pp. 569–571.

[26] R. G. v.d. Waal and R. N. J. Veldhuis, "Subband Coding of Sterephonic Digital Audio Signals," in *Proc. IEEE ICASSP* (1991), pp. 3601–3604.

[27] J. Herre, K. Brandenburg, and D. Lederer, "Intensity Stereo Coding," presented at the 96th Convention

of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 42, p. 394 (1994 May), preprint 3799.

[28] G. Stoll, G. Theile, S. Nielsen, A. Silzle, M. Link, R. Sedlmayer, and A. Breford, "Extension of ISO/MPEG-Audio Layer II to Multi-Channel Coding: The Future Standard for Broadcasting, Telecommunication, and Multimedia Applications," presented at the 94th Convention of the Audio Engineering Society, Berlin, Germany, 1993 March 16–19, preprint 3550.

[29] M. F. Davis, "The AC-3 Multichannel Coder," presented at the 95th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 41, p. 1071 (1993 Dec.), preprint 3774.

[30] ISO/IEC JTC1/SC29/WG11 N1420, "Overview of the Report on the Formal Subjective Listening Tests of MPEG-2 AAC Multichannel Audio Coding," Maceió, Brazil (1996 Nov.).

[31] ITU-R BS.1116, "Methods for the Subjective Assessment of Small Impairments in Audio Systems Including Multichannel Sound Systems," Geneva, Switzerland (1994).

[32] ISO/IEC JTC1/SC29/WG11 N1712, "Report on Complexity of MPEG-2 AAC Tools," Bristol, UK (1997 Apr.).

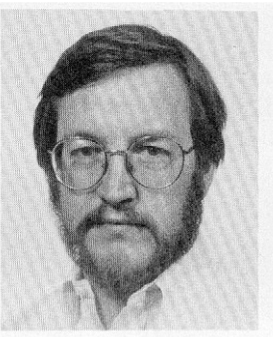
[33] ISO/IEC JTC1/SC29/WG11 N1628, "Report on Reduction of Complexity in the AAC Prediction Tool," Bristol, UK (1997 Apr.).

[34] ISO/IEC JTC1/SC29/WG11 N1629, "Results of the Brief Assessments on AAC Reduction of Prediction Complexity," Bristol, UK (1997 Apr.).

THE AUTHORS

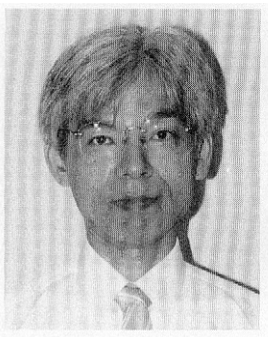
M. Bosi

K. Brandenburg



S. Quackenbush

L. Fielder

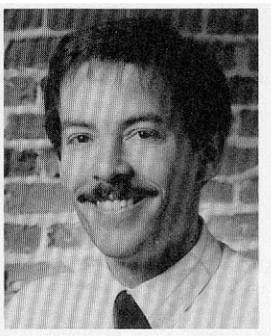


K. Akagiri

H. Fuchs

M. Dietz

J. Herre



G. A. Davidson

Y. Oikawa

Marina Bosi currently works for Digital Theater Systems (DTS) as vice president, technology standards and strategy. She is also a member of the ANSI, ISO/MPEG, DAVIC, and ITU-R standardization committees setting up international standards for low bit-rate audio coding. In addition, she is a staff member at Stanford University's Center for Computer Research in Music and Acoustics (CCRMA). She is the editor of the new MPEG-2 Advanced Audio Coding standard (ISO/IEC 13818-7) and the author of a number of publications on source coding for transmission and storage. Her current area of interest is low bit-rate coding with applications in music.

Dr. Bosi graduated from the National Conservatory of Music in Florence. She then received her doctorate in physics from the University of Florence, having completed her dissertation in Paris at the Institut de Rechèrche et Coordination Acoustique/Musique (IRCAM). She worked for Dolby Laboratories in the R&D and Business Development Department where she developed and commercialized new low bit-rate audio coders. She has also worked for Digidesign where she developed audio digital signal processing technology including dynamic range controller and music analysis/synthesis algorithms.

Dr. Bosi has served the AES San Francisco Section as committeeperson, vice-chairman, and chairman. She served as a member of the AES Board of Governors and is currently vice president of the Western Region, USA/Canada. She was cochairman of the AES 97th Convention, for which she received the AES Board of Governors Award, and papers chairman of the AES 101st Convention. She is also cochairman of the AES Conference Policy Committee. In addition to AES, Dr. Bosi is a member of the technical committee on Audio and Electroacoustics of the IEEE Signal Processing Society and a member of the Acoustical Society of America (ASA).

•

Karlheinz Brandenburg was born in Erlangen, Germany, in 1954. He received M.S. (Diplom) degrees in electrical engineering in 1980 and in mathematics in 1982 from Erlangen University. In 1989 he earned a Ph.D. in electrical engineering, also from Erlangen University, for work on digital audio coding and perceptual measurement techniques.

From 1989 to 1990 he was with AT&T Bell Laboratories in Murray Hill, NJ, USA. He worked on the ASPEC perceptual coding technique and on the definition of the ISO/IEC MPEG/Audio Layer 3 system. In 1990 he returned to Erlangen University to continue research on audio coding and to teach a course on digital audio technology. Since 1993 he has been head of the Audio/Multimedia Department at the Fraunhofer Institut for Integrated Circuits (FhG-IIs).

Dr. Brandenburg has presented numerous papers at AES conventions. In 1994 he received an AES Fellowship for his work on perceptual audio coding and psychoacoustics. He is a member of the AES and of the technical committee on Audio and Electroacoustics of the IEEE Signal Processing Society. He is an active member of the ISO MPEG standardization committee, working on advanced audio coding systems. He has been granted 12 patents and has several more pending.

•

Schuyler Quackenbush received a B.S. degree from Princeton University in 1975. After four years in industry as a design engineer, he entered the Georgia Institute of Technology, from which he received an M.S. degree and Ph.D. degree in electrical engineering in 1980 and 1985, respectively. For the latter half of 1985, he was a staff research associate at the Georgia Institute of Technology. In 1986 he joined the Digital Signal Processing Research Department of AT&T Bell Laboratories as a member of the technical staff. In 1996 he joined the newly created AT&T Laboratories.

His current research interests are speech and music coding algorithms, and real-time signal processing hardware. Dr. Quackenbush is a member of the Institute of Electrical and Electronics Engineers.

•

Louis Fielder received a B.S. degree in electrical engineering from the California Institute of Technology in 1974 and an M.S. degree in acoustics from the University of California in Los Angeles in 1976. During the period from 1976 to 1978 he worked on electronic component design for custom sound reinforcement systems at Paul Veneklasen and Associates. From 1978 to 1984 he was involved in digital audio and magnetic recording research at Ampex Corporation.

Since 1984 he has worked at Dolby Laboratories and has been involved in the application of psychoacoustics to the development of audio systems. He has written a number of papers on the determination of the limits of performance for digital audio and low-frequency loudspeaker systems. His current area of interest is the development of low-bit-rate audio coders for music transmission and storage applications. Since 1985 he has been involved in the AES Digital Audio Standards Committee subgroup on digital audio performance measurements. He is past president, a current Member of the Board of Governors, and a Fellow of the AES.

•

Kenzo Akagiri graduated from the Electrical Engineering Faculty of the National Akashi Technical College, Japan, in 1969. He joined Sony Corporation and since 1974 has been engaged in the development of analog audio noise reduction systems, digital signal processing (DSP), and audio bit-rate reduction. He was also involved in the development of the Beta Hi-Fi, 8-mm video, CD-ROM XA, and the Mini Disc audio formats. He is currently general manager, A&V Processing Laboratory, Media Processing Laboratories, Sony Corporation and is engaged in developing audio and video signal processing technologies.

Mr. Akagiri received the 1985 Eduard Rhein prize for "DAV Multi-Track PCM Sound Cassette Receiver System" and was awarded second place for the outstanding paper presented at the 1982 International Conference on Consumer Electronics of the Institute of Electrical and Electronics Engineers (IEEE).

•

Hendrik Fuchs studied electrical engineering at the University of Hanover, Germany. He received the Dipl.-Ing. degree in 1987. From 1987 to 1997, he worked as a research assistant and later as senior engineer at the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung at the University of Hanover. His main research interest has been source coding of sound signals.

Since 1988, he has been a member of the MPEG audio group where he contributed to the development of the ISO Audio Coding Standards. In 1997, he joined Robert Bosch GmbH, where he currently is involved in the development of mobile multimedia systems.

•

Martin Dietz, was born in Nuernberg, Germany, in 1965 and studied electrical engineering at Erlanger University. After receiving the M.S. (Diplom) degree in 1992 he joined the Fraunhofer Institut Integrierte Schal-

tungen (IIS) working on real-time and simulation software for ISO/MPEG Layer 3, including the implementation of a single-chip decoder. Since July 1995 he has been head of the algorithm development group of the IIS, mainly working on simulation and implementation of high-quality low bit-rate audio coding schemes, such as, MPEG Layer 3, MPEG-2 AAC, and MPEG-4 audio.

•

Jürgen Herre studied electrical engineering from 1982 to 1989 at Erlangen/Nürnberg University, Germany, with telecommunications, digital signal processing, and applied electronics as his main topics. After receiving his Diplom Ingenieur degree in 1989 he joined the Fraunhofer Institute for Integrated Circuits (IIS) in Erlangen. His work included algorithmic development as well as real-time implementation of perceptual coding systems for high-quality audio. Specifically, he participated in the development of the well-known coding algorithms ASPEC and ISO/MPEG-Audio Layer III. After receiving his Ph.D. in 1995, Dr. Herre joined Bell Laboratories for a postdoctorate term where he worked on the new MPEG-2 Advanced Audio Coding (AAC) scheme. Currently he is again at Fraunhofer involved with MPEG-4 audio work.

Special interests include perceptual measurement, joint stereo and multichannel perceptual coding, and error concealment technology for low bit-rate coders. He is author and coauthor of a large number of papers and patents in these fields. Dr. Herre is a fellow of the Audio Engineering Society and a member of the IEEE Signal Processing Society.

•

Grant A. Davidson was born in San Francisco, CA, in 1958 March. He received a B.S. degree in physics from the California Polytechnic State University in 1980, and M.S. and Ph.D. degrees in electrical engineering from the University of California, Santa Barbara, in 1984 and 1987, respectively.

From 1983 to 1987 he was a member of the Communications Research Laboratory at UC Santa Barbara, where he studied low bit-rate speech compression techniques using vector quantization and linear predictive coding. He has also studied special-purpose VLSI processor architectures for real-time DSP applications.

Since 1987, Dr. Davidson has been a member of the Research and Development Department of Dolby Laboratories. His research activities include digital speech and audio processing, with emphasis on psychoacoustically based low bit-rate audio coding algorithms such as Dolby AC-3 and MPEG-2 Advanced Audio Coding.

•

Yoshiaki Oikawa received the B.S. and M.S. degrees in electrical engineering from Nagaoka University of Technology, Niigata, Japan, in 1984 and 1986 respectively.

He joined Sony Corporation in 1986. He has been engaged in research on digital signal processing for audio and speech signals at the Media Processing Laboratory of Sony Corporation.