INTERNATIONAL STANDARD

ISO/IEC 11172-3

First edition
1993-08-01

Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s —

Part 3:
Audio

Technologies de l'information — Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s —

Partie 3: Audio

ISO IEC

Reference number
ISO/IEC 11172-3:1993(E)

Contents

Page

ISO/IEC Copyright Office • Case Postale 56 • CH1211 Genève 20 • Switzerland

Printed in Switzerland.

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 11172-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Sub-Committee SC 29, *Coded representation of audio, picture, multimedia and hypermedia information.*

ISO/IEC 11172 consists of the following parts, under the general title *Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s*:

— *Part 1: Systems*

— *Part 2: Video*

— *Part 3: Audio*

— *Part 4: Compliance testing*

Annexes A and B form an integral part of this part of ISO/IEC 11172. Annexes C, D, E, F, G and H are for information only.

Introduction

Note: Readers interested in an overview of MPEG Audio should read this Introduction and then proceed to annex A (Diagrams) and annex C (The encoding process) before reading the normative clauses 1 and 2.

To aid in the understanding of the specification of the stored compressed bitstream and its decoding, a sequence of encoding, storage and decoding is described.

0.1 Encoding

The encoder processes the digital audio signal and produces the compressed bitstream for storage. The encoder algorithm is not standardized, and may use various means for encoding such as estimation of the auditory masking threshold, quantization, and scaling. However, the encoder output must be such that a decoder conforming to the specifications of clause 2.4 will produce audio suitable for the intended application.

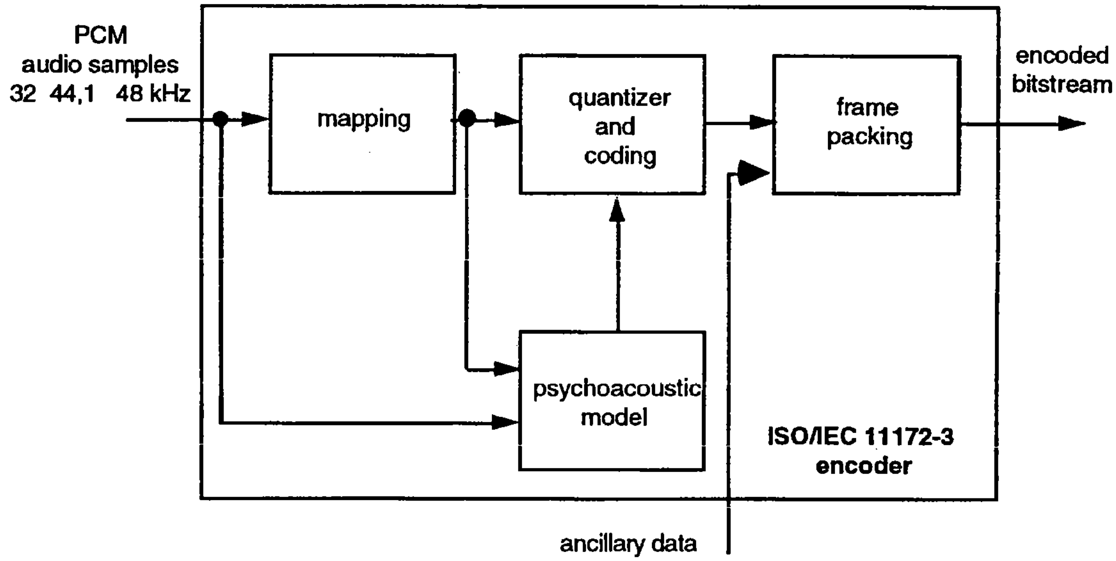


Figure 1 -- Sketch of the basic structure of an encoder

Figure 1 illustrates the basic structure of a audio encoder. Input audio samples are fed into the encoder. The mapping creates a filtered and subsampled representation of the input audio stream. The mapped samples may be called either subband samples (as in Layer I or II, see below) or transformed subband samples (as in Layer III). A psychoacoustic model creates a set of data to control the quantizer and coding. These data are different depending on the actual coder implementation. One possibility is to use an estimation of the masking threshold to do this quantizer control. The quantizer and coding block creates a set of coding symbols from the mapped input samples. Again, this block can depend on the encoding system. The block 'frame packing' assembles the actual bitstream from the output data of the other blocks, and adds other information (e.g. error correction) if necessary.

There are four different modes possible, single channel, dual channel (two independent audio signals coded within one bitstream), stereo (left and right signals of a stereo pair coded within one bitstream), and Joint Stereo (left and right signals of a stereo pair coded within one bitstream with the stereo irrelevancy and redundancy exploited).

0.2 Layers

Depending on the application, different layers of the coding system with increasing encoder complexity and performance can be used. An ISO/IEC 11172-3 Audio Layer N decoder is able to decode bitstream data which has been encoded in Layer N and all layers below N.

Layer I

This layer contains the basic mapping of the digital audio input into 32 subbands, fixed segmentation to format the data into blocks, a psychoacoustic model to determine the adaptive bit allocation, and quantization using block companding and formatting. The theoretical minimum encoding/decoding delay for Layer I is about 19 ms.

Layer II

This layer provides additional coding of bit allocation, scalefactors and samples. Different framing is used. The theoretical minimum encoding/decoding delay for Layer II is about 35 ms.

Layer III

This layer introduces increased frequency resolution based on a hybrid filterbank. It adds a different (nonuniform) quantizer, adaptive segmentation and entropy coding of the quantized values. The theoretical minimum encoding/decoding delay for Layer III is about 59 ms.

Joint Stereo coding can be added as an additional feature to any of the layers.

0.3 Storage

Various streams of encoded video, encoded audio, synchronization data, systems data and auxiliary data may be stored together on a storage medium. Editing of the audio will be easier if the edit point is constrained to coincide with an addressable point.

Access to storage may involve remote access over a communication system. Access is assumed to be controlled by a functional unit other than the audio decoder itself. This control unit accepts user commands, reads and interprets data base structure information, reads the stored information from the media, demultiplexes non-audio information and passes the stored audio bitstream to the audio decoder at the required rate.

0.4 Decoding

The decoder accepts the compressed audio bitstream in the syntax defined in 2.4.1, decodes the data elements according to 2.4.2, and uses the information to produce digital audio output according to 2.4.3.

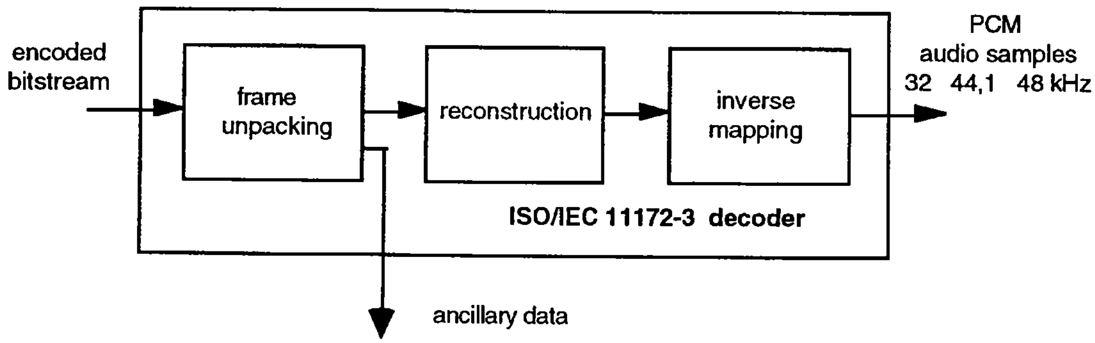


Figure 2 -- Sketch of the basic structure of a decoder

Figure 2 illustrates the basic structure of a audio decoder. Bitstream data is fed into the decoder. The bitstream unpacking and decoding block does error detection if error-check is applied in the encoder (see 2.4.2.4). The bitstream data are unpacked to recover the various pieces of information. The reconstruction block reconstructs the quantized version of the set of mapped samples. The inverse mapping transforms these mapped samples back into uniform PCM.

Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s —

Part 3: Audio

Section 1: General

1.1 Scope

This part of ISO/IEC 11172 specifies the coded representation of high quality audio for storage media and the method for decoding of high quality audio signals. The input of the encoder and the output of the decoder are compatible with existing PCM standards such as standard Compact Disc and Digital Audio Tape.

This part of the ISO/IEC 11172 is intended for application to digital storage media providing a total continuous transfer rate of about 1,5 Mbits/sec for both audio and video bitstreams, such as CD, DAT and magnetic hard disc. The storage media may either be connected directly to the decoder, or via other means such as communication lines and the ISO/IEC 11172 multiplexed stream defined in ISO/IEC 11172-1. This part of ISO/IEC 11172 is intended for sampling rates of 32 kHz, 44,1 kHz, and 48 kHz.

1.2 Normative references

The following International Standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 11172. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 11172 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 11172-1:1993 *Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 1 Systems.*

ISO/IEC 11172-2:1993 *Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2: Video.*

CCIR Recommendation 601-2 *Encoding parameters of digital television for studios.*

CCIR Report 624-4 *Characteristics of systems for monochrome and colour television.*

CCIR Recommendation 648 *Recording of audio signals.*

CCIR Report 955-2 *Sound broadcasting by satellite for portable and mobile receivers, including Annex IV Summary description of Advanced Digital System II.*

CCITT Recommendation J.17 *Pre-emphasis used on Sound-Programme Circuits.*

IEEE Draft Standard P1180/D2 1990 *Specification for the implementation of 8x 8 inverse discrete cosine transform".*

IEC publication 908:1987 *CD Digital Audio System.*

Section 2: Technical elements

2.1 Definitions

For the purposes of ISO/IEC 11172, the following definitions apply. If specific to a part, this is noted in square brackets.

2.1.1 ac coefficient [video]: Any DCT coefficient for which the frequency in one or both dimensions is non-zero.

2.1.2 access unit [system]: In the case of compressed audio an access unit is an audio access unit. In the case of compressed video an access unit is the coded representation of a picture.

2.1.3 adaptive segmentation [audio]: A subdivision of the digital representation of an audio signal in variable segments of time.

2.1.4 adaptive bit allocation [audio]: The assignment of bits to subbands in a time and frequency varying fashion according to a psychoacoustic model.

2.1.5 adaptive noise allocation [audio]: The assignment of coding noise to frequency bands in a time and frequency varying fashion according to a psychoacoustic model.

2.1.6 alias [audio]: Mirrored signal component resulting from sub-Nyquist sampling.

2.1.7 analysis filterbank [audio]: Filterbank in the encoder that transforms a broadband PCM audio signal into a set of subsampled subband samples.

2.1.8 audio access unit [audio]: For Layers I and II an audio access unit is defined as the smallest part of the encoded bitstream which can be decoded by itself, where decoded means "fully reconstructed sound". For Layer III an audio access unit is part of the bitstream that is decodable with the use of previously acquired main information.

2.1.9 audio buffer [audio]: A buffer in the system target decoder for storage of compressed audio data.

2.1.10 audio sequence [audio]: A non-interrupted series of audio frames in which the following parameters are not changed:

- ID
- Layer
- Sampling Frequency
- For Layer I and II: Bitrate index

2.1.11 backward motion vector [video]: A motion vector that is used for motion compensation from a reference picture at a later time in display order.

2.1.12 Bark [audio]: Unit of critical band rate. The Bark scale is a non-linear mapping of the frequency scale over the audio range closely corresponding with the frequency selectivity of the human ear across the band.

2.1.13 bidirectionally predictive-coded picture; B-picture [video]: A picture that is coded using motion compensated prediction from a past and/or future reference picture.

2.1.14 bitrate: The rate at which the compressed bitstream is delivered from the storage medium to the input of a decoder.

2.1.15 block companding [audio]: Normalizing of the digital representation of an audio signal within a certain time period.

2.1.16 block [video]: An 8-row by 8-column orthogonal block of pels.

2.1.17 bound [audio]: The lowest subband in which intensity stereo coding is used.

2.1.18 byte aligned: A bit in a coded bitstream is byte-aligned if its position is a multiple of 8-bits from the first bit in the stream.

2.1.19 byte: Sequence of 8-bits.

2.1.20 channel: A digital medium that stores or transports an ISO/IEC 11172 stream.

2.1.21 channel [audio]: The left and right channels of a stereo signal

2.1.22 chrominance (component) [video]: A matrix, block or single pel representing one of the two colour difference signals related to the primary colours in the manner defined in CCIR Rec 601. The symbols used for the colour difference signals are Cr and Cb.

2.1.23 coded audio bitstream [audio]: A coded representation of an audio signal as specified in this part of ISO/IEC 11172.

2.1.24 coded video bitstream [video]: A coded representation of a series of one or more pictures as specified in ISO/IEC 11172-2.

2.1.25 coded order [video]: The order in which the pictures are stored and decoded. This order is not necessarily the same as the display order.

2.1.26 coded representation: A data element as represented in its encoded form.

2.1.27 coding parameters [video]: The set of user-definable parameters that characterize a coded video bitstream. Bitstreams are characterised by coding parameters. Decoders are characterised by the bitstreams that they are capable of decoding.

2.1.28 component [video]: A matrix, block or single pel from one of the three matrices (luminance and two chrominance) that make up a picture.

2.1.29 compression: Reduction in the number of bits used to represent an item of data.

2.1.30 constant bitrate coded video [video]: A compressed video bitstream with a constant average bitrate.

2.1.31 constant bitrate: Operation where the bitrate is constant from start to finish of the compressed bitstream.

2.1.32 constrained parameters [video]: The values of the set of coding parameters defined in 2.4.3.2 of ISO/IEC 11172-2.

2.1.33 constrained system parameter stream (CSPS) [system]: An ISO/IEC 11172 multiplexed stream for which the constraints defined in 2.4.6 of ISO/IEC 11172-1 apply.

2.1.34 CRC: Cyclic redundancy code.

2.1.35 critical band rate [audio]: Psychoacoustic function of frequency. At a given audible frequency it is proportional to the number of critical bands below that frequency. The units of the critical band rate scale are Barks.

2.1.36 critical band [audio]: Psychoacoustic measure in the spectral domain which corresponds to the frequency selectivity of the human ear. This selectivity is expressed in Bark.

2.1.37 data element: An item of data as represented before encoding and after decoding.

2.1.38 dc-coefficient [video]: The DCT coefficient for which the frequency is zero in both dimensions.

2.1.39 dc-coded picture; D-picture [video]: A picture that is coded using only information from itself. Of the DCT coefficients in the coded representation, only the dc-coefficients are present.

2.1.40 DCT coefficient: The amplitude of a specific cosine basis function.

2.1.41 decoded stream: The decoded reconstruction of a compressed bitstream.

2.1.42 decoder input buffer [video]: The first-in first-out (FIFO) buffer specified in the video buffering verifier.

2.1.43 decoder input rate [video]: The data rate specified in the video buffering verifier and encoded in the coded video bitstream.

2.1.44 decoder: An embodiment of a decoding process.

2.1.45 decoding (process): The process defined in ISO/IEC 11172 that reads an input coded bitstream and produces decoded pictures or audio samples.

2.1.46 decoding time-stamp; DTS [system]: A field that may be present in a packet header that indicates the time that an access unit is decoded in the system target decoder.

2.1.47 de-emphasis [audio]: Filtering applied to an audio signal after storage or transmission to undo a linear distortion due to emphasis.

2.1.48 dequantization [video]: The process of rescaling the quantized DCT coefficients after their representation in the bitstream has been decoded and before they are presented to the inverse DCT.

2.1.49 digital storage media; DSM: A digital storage or transmission device or system.

2.1.50 discrete cosine transform; DCT [video]: Either the forward discrete cosine transform or the inverse discrete cosine transform. The DCT is an invertible, discrete orthogonal transformation. The inverse DCT is defined in annex A of ISO/IEC 11172-2.

2.1.51 display order [video]: The order in which the decoded pictures should be displayed. Normally this is the same order in which they were presented at the input of the encoder.

2.1.52 dual channel mode [audio]: A mode, where two audio channels with independent programme contents (e.g. bilingual) are encoded within one bitstream. The coding process is the same as for the stereo mode.

2.1.53 editing: The process by which one or more compressed bitstreams are manipulated to produce a new compressed bitstream. Conforming edited bitstreams must meet the requirements defined in this ISO/IEC 11172.

2.1.54 elementary stream [system]: A generic term for one of the coded video, coded audio or other coded bitstreams.

2.1.55 emphasis [audio]: Filtering applied to an audio signal before storage or transmission to improve the signal-to-noise ratio at high frequencies.

2.1.56 encoder: An embodiment of an encoding process.

2.1.57 encoding (process): A process, not specified in ISO/IEC 11172, that reads a stream of input pictures or audio samples and produces a valid coded bitstream as defined in ISO/IEC 11172.

2.1.58 entropy coding: Variable length lossless coding of the digital representation of a signal to reduce redundancy.

2.1.59 fast forward playback [video]: The process of displaying a sequence, or parts of a sequence, of pictures in display-order faster than real-time.

2.1.60 FFT: Fast Fourier Transformation. A fast algorithm for performing a discrete Fourier transform (an orthogonal transform).

2.1.61 filterbank [audio]: A set of band-pass filters covering the entire audio frequency range.

2.1.62 fixed segmentation [audio]: A subdivision of the digital representation of an audio signal into fixed segments of time.

2.1.63 forbidden: The term "forbidden" when used in the clauses defining the coded bitstream indicates that the value shall never be used. This is usually to avoid emulation of start codes.

2.1.64 forced updating [video]: The process by which macroblocks are intra-coded from time-to-time to ensure that mismatch errors between the inverse DCT processes in encoders and decoders cannot build up excessively.

2.1.65 forward motion vector [video]: A motion vector that is used for motion compensation from a reference picture at an earlier time in display order.

2.1.66 frame [audio]: A part of the audio signal that corresponds to audio PCM samples from an Audio Access Unit.

2.1.67 free format [audio]: Any bitrate other than the defined bitrates that is less than the maximum valid bitrate for each layer.

2.1.68 future reference picture [video]: The future reference picture is the reference picture that occurs at a later time than the current picture in display order.

2.1.69 granules [Layer II] [audio]: The set of 3 consecutive subband samples from all 32 subbands that are considered together before quantization. They correspond to 96 PCM samples.

2.1.70 granules [Layer III] [audio]: 576 frequency lines that carry their own side information.

2.1.71 group of pictures [video]: A series of one or more coded pictures intended to assist random access. The group of pictures is one of the layers in the coding syntax defined in ISO/IEC 11172-2.

2.1.72 Hann window [audio]: A time function applied sample-by-sample to a block of audio samples before Fourier transformation.

2.1.73 Huffman coding: A specific method for entropy coding.

2.1.74 hybrid filterbank [audio]: A serial combination of subband filterbank and MDCT.

2.1.75 IMDCT [audio]: Inverse Modified Discrete Cosine Transform.

2.1.76 intensity stereo [audio]: A method of exploiting stereo irrelevance or redundancy in stereophonic audio programmes based on retaining at high frequencies only the energy envelope of the right and left channels.

2.1.77 interlace [video]: The property of conventional television pictures where alternating lines of the picture represent different instances in time.

2.1.78 intra coding [video]: Coding of a macroblock or picture that uses information only from that macroblock or picture.

2.1.79 intra-coded picture; I-picture [video]: A picture coded using information only from itself.

2.1.80 ISO/IEC 11172 (multiplexed) stream [system]: A bitstream composed of zero or more elementary streams combined in the manner defined in ISO/IEC 11172-1.

2.1.81 joint stereo coding [audio]: Any method that exploits stereophonic irrelevance or stereophonic redundancy.

2.1.82 joint stereo mode [audio]: A mode of the audio coding algorithm using joint stereo coding.

2.1.83 layer [audio]: One of the levels in the coding hierarchy of the audio system defined in this part of ISO/IEC 11172.

2.1.84 layer [video and systems]: One of the levels in the data hierarchy of the video and system specifications defined in ISO/IEC 11172-1 and ISO/IEC 11172-2.

2.1.85 luminance (component) [video]: A matrix, block or single pel representing a monochrome representation of the signal and related to the primary colours in the manner defined in CCIR Rec 601. The symbol used for luminance is Y.

2.1.86 macroblock [video]: The four 8 by 8 blocks of luminance data and the two corresponding 8 by 8 blocks of chrominance data coming from a 16 by 16 section of the luminance component of the picture. Macroblock is sometimes used to refer to the pel data and sometimes to the coded representation of the pel values and other data elements defined in the macroblock layer of the syntax defined in ISO/IEC 11172-2. The usage is clear from the context.

2.1.87 mapping [audio]: Conversion of an audio signal from time to frequency domain by subband filtering and/or by MDCT.

2.1.88 masking [audio]: A property of the human auditory system by which an audio signal cannot be perceived in the presence of another audio signal .

2.1.89 masking threshold [audio]: A function in frequency and time below which an audio signal cannot be perceived by the human auditory system.

2.1.90 MDCT [audio]: Modified Discrete Cosine Transform.

2.1.91 motion compensation [video]: The use of motion vectors to improve the efficiency of the prediction of pel values. The prediction uses motion vectors to provide offsets into the past and/or future reference pictures containing previously decoded pel values that are used to form the prediction error signal.

2.1.92 motion estimation [video]: The process of estimating motion vectors during the encoding process.

2.1.93 motion vector [video]: A two-dimensional vector used for motion compensation that provides an offset from the coordinate position in the current picture to the coordinates in a reference picture.

2.1.94 MS stereo [audio]: A method of exploiting stereo irrelevance or redundancy in stereophonic audio programmes based on coding the sum and difference signal instead of the left and right channels.

2.1.95 non-intra coding [video]: Coding of a macroblock or picture that uses information both from itself and from macroblocks and pictures occurring at other times.

2.1.96 non-tonal component [audio]: A noise-like component of an audio signal.

2.1.97 Nyquist sampling: Sampling at or above twice the maximum bandwidth of a signal.

2.1.98 pack [system]: A pack consists of a pack header followed by one or more packets. It is a layer in the system coding syntax described in ISO/IEC 11172-1.

2.1.99 packet data [system]: Contiguous bytes of data from an elementary stream present in a packet.

2.1.100 packet header [system]: The data structure used to convey information about the elementary stream data contained in the packet data.

2.1.101 packet [system]: A packet consists of a header followed by a number of contiguous bytes from an elementary data stream. It is a layer in the system coding syntax described in ISO/IEC 11172-1.

2.1.102 padding [audio]: A method to adjust the average length in time of an audio frame to the duration of the corresponding PCM samples, by conditionally adding a slot to the audio frame.

2.1.103 past reference picture [video]: The past reference picture is the reference picture that occurs at an earlier time than the current picture in display order.

2.1.104 pel aspect ratio [video]: The ratio of the nominal vertical height of pel on the display to its nominal horizontal width.

2.1.105 pel [video]: Picture element.

2.1.106 picture period [video]: The reciprocal of the picture rate.

2.1.107 picture rate [video]: The nominal rate at which pictures should be output from the decoding process.

2.1.108 picture [video]: Source, coded or reconstructed image data. A source or reconstructed picture consists of three rectangular matrices of 8-bit numbers representing the luminance and two chrominance signals. The Picture layer is one of the layers in the coding syntax defined in ISO/IEC 11172-2. Note that the term "picture" is always used in ISO/IEC 11172 in preference to the terms field or frame.

2.1.109 polyphase filterbank [audio]: A set of equal bandwidth filters with special phase interrelationships, allowing for an efficient implementation of the filterbank.

2.1.110 prediction [video]: The use of a predictor to provide an estimate of the pel value or data element currently being decoded.

2.1.111 predictive-coded picture; P-picture [video]: A picture that is coded using motion compensated prediction from the past reference picture.

2.1.112 prediction error [video]: The difference between the actual value of a pel or data element and its predictor.

2.1.113 predictor [video]: A linear combination of previously decoded pel values or data elements.

2.1.114 presentation time-stamp; PTS [system]: A field that may be present in a packet header that indicates the time that a presentation unit is presented in the system target decoder.

2.1.115 presentation unit; PU [system]: A decoded audio access unit or a decoded picture.

2.1.116 psychoacoustic model [audio]: A mathematical model of the masking behaviour of the human auditory system.

2.1.117 quantization matrix [video]: A set of sixty-four 8-bit values used by the dequantizer.

2.1.118 quantized DCT coefficients [video]: DCT coefficients before dequantization. A variable length coded representation of quantized DCT coefficients is stored as part of the compressed video bitstream.

2.1.119 quantizer scalefactor [video]: A data element represented in the bitstream and used by the decoding process to scale the dequantization.

2.1.120 random access: The process of beginning to read and decode the coded bitstream at an arbitrary point.

2.1.121 reference picture [video]: Reference pictures are the nearest adjacent I- or P-pictures to the current picture in display order.

2.1.122 reorder buffer [video]: A buffer in the system target decoder for storage of a reconstructed I-picture or a reconstructed P-picture.

2.1.123 requantization [audio]: Decoding of coded subband samples in order to recover the original quantized values.

2.1.124 reserved: The term "reserved" when used in the clauses defining the coded bitstream indicates that the value may be used in the future for ISO/IEC defined extensions.

2.1.125 reverse playback [video]: The process of displaying the picture sequence in the reverse of display order.

2.1.126 scalefactor band [audio]: A set of frequency lines in Layer III which are scaled by one scalefactor.

2.1.127 scalefactor index [audio]: A numerical code for a scalefactor.

2.1.128 scalefactor [audio]: Factor by which a set of values is scaled before quantization.

2.1.129 sequence header [video]: A block of data in the coded bitstream containing the coded representation of a number of data elements.

2.1.130 side information: Information in the bitstream necessary for controlling the decoder.

2.1.131 skipped macroblock [video]: A macroblock for which no data are stored.

2.1.132 slice [video]: A series of macroblocks. It is one of the layers of the coding syntax defined in ISO/IEC 11172-2.

2.1.133 slot [audio]: A slot is an elementary part in the bitstream. In Layer I a slot equals four bytes, in Layers II and III one byte.

2.1.134 source stream: A single non-multiplexed stream of samples before compression coding.

2.1.135 spreading function [audio]: A function that describes the frequency spread of masking.

2.1.136 start codes [system and video]: 32-bit codes embedded in that coded bitstream that are unique. They are used for several purposes including identifying some of the layers in the coding syntax.

2.1.137 STD input buffer [system]: A first-in first-out buffer at the input of the system target decoder for storage of compressed data from elementary streams before decoding.

2.1.138 stereo mode [audio]: Mode, where two audio channels which form a stereo pair (left and right) are encoded within one bitstream. The coding process is the same as for the dual channel mode.

2.1.139 stuffing (bits); stuffing (bytes) : Code-words that may be inserted into the compressed bitstream that are discarded in the decoding process. Their purpose is to increase the bitrate of the stream.

2.1.140 subband [audio]: Subdivision of the audio frequency band.

2.1.141 subband filterbank [audio]: A set of band filters covering the entire audio frequency range. In this part of ISO/IEC 11172 the subband filterbank is a polyphase filterbank.

2.1.142 subband samples [audio]: The subband filterbank within the audio encoder creates a filtered and subsampled representation of the input audio stream. The filtered samples are called subband samples.

From 384 time-consecutive input audio samples, 12 time-consecutive subband samples are generated within each of the 32 subbands.

2.1.143 syncword [audio]: A 12-bit code embedded in the audio bitstream that identifies the start of a frame.

2.1.144 synthesis filterbank [audio]: Filterbank in the decoder that reconstructs a PCM audio signal from subband samples.

2.1.145 system header [system]: The system header is a data structure defined in ISO/IEC 11172-1 that carries information summarising the system characteristics of the ISO/IEC 11172 multiplexed stream.

2.1.146 system target decoder; STD [system]: A hypothetical reference model of a decoding process used to describe the semantics of an ISO/IEC 11172 multiplexed bitstream.

2.1.147 time-stamp [system]: A term that indicates the time of an event.

2.1.148 triplet [audio]: A set of 3 consecutive subband samples from one subband. A triplet from each of the 32 subbands forms a granule.

2.1.149 tonal component [audio]: A sinusoid-like component of an audio signal.

2.1.150 variable bitrate: Operation where the bitrate varies with time during the decoding of a compressed bitstream.

2.1.151 variable length coding; VLC: A reversible procedure for coding that assigns shorter code-words to frequent events and longer code-words to less frequent events.

2.1.152 video buffering verifier; VBV [video]: A hypothetical decoder that is conceptually connected to the output of the encoder. Its purpose is to provide a constraint on the variability of the data rate that an encoder or editing process may produce.

2.1.153 video sequence [video]: A series of one or more groups of pictures. It is one of the layers of the coding syntax defined in ISO/IEC 11172-2.

2.1.154 zig-zag scanning order [video]: A specific sequential ordering of the DCT coefficients from (approximately) the lowest spatial frequency to the highest.

2.2 Symbols and abbreviations

The mathematical operators used to describe this International Standard are similar to those used in the C programming language. However, integer division with truncation and rounding are specifically defined. The bitwise operators are defined assuming twos-complement representation of integers. Numbering and counting loops generally begin from zero.

2.2.1 Arithmetic operators

+	Addition.
-	Subtraction (as a binary operator) or negation (as a unary operator).
++	Increment.
- -	Decrement.
*	Multiplication.
^	Power.
/	Integer division with truncation of the result toward zero. For example, 7/4 and -7/-4 are truncated to 1 and -7/4 and 7/-4 are truncated to -1.
//	Integer division with rounding to the nearest integer. Half-integer values are rounded away from zero unless otherwise specified. For example 3//2 is rounded to 2, and -3//2 is rounded to -2.
DIV	Integer division with truncation of the result towards $-\infty$.
| |	Absolute value. $\lvert x \rvert = x$ when $x > 0$ $\lvert x \rvert = 0$ when $x == 0$ $\lvert x \rvert = -x$ when $x < 0$
%	Modulus operator. Defined only for positive numbers.
Sign( )	Sign(x) = 1 x > 0 0 x == 0 -1 x < 0
NINT ( )	Nearest integer operator. Returns the nearest integer value to the real-valued argument. Half-integer values are rounded away from zero.
sin	Sine.
cos	Cosine.
exp	Exponential.
√	Square root.
$\log_{10}$	Logarithm to base ten.
$\log_e$	Logarithm to base e.
$\log_2$	Logarithm to base 2.

2.2.2 Logical operators

||	Logical OR.
&&	Logical AND.

! Logical NOT

2.2.3 Relational operators

>	Greater than.
>=	Greater than or equal to.
<	Less than.
<=	Less than or equal to.
==	Equal to.
!=	Not equal to.
max [,...,]	the maximum value in the argument list.
min [,...,]	the minimum value in the argument list.

2.2.4 Bitwise operators

A twos complement number representation is assumed where the bitwise operators are used.

&	AND
|	OR
>>	Shift right with sign extension.
<<	Shift left with zero fill.

2.2.5 Assignment

= Assignment operator.

2.2.6 Mnemonics

The following mnemonics are defined to describe the different data types used in the coded bit-stream.

bslbf	Bit string, left bit first, where "left" is the order in which bit strings are written in ISO/IEC 11172. Bit strings are written as a string of 1s and 0s within single quote marks, e.g. '1000 0001'. Blanks within a bit string are for ease of reading and have no significance.
ch	Channel. If ch has the value 0, the left channel of a stereo signal or the first of two independent signals is indicated. (Audio)
nch	Number of channels; equal to 1 for single_channel mode, 2 in other modes. (Audio)
gr	Granule of 3 * 32 subband samples in audio Layer II, 18 * 32 sub-band samples in audio Layer III. (Audio)
main_data	The main_data portion of the bitstream contains the scalefactors, Huffman encoded data, and ancillary information. (Audio)
main_data_beg	The location in the bitstream of the beginning of the main_data for the frame. The location is equal to the ending location of the previous frame's main_data plus one bit. It is calculated from the main_data_end value of the previous frame. (Audio)
part2_length	The number of main_data bits used for scalefactors. (Audio)

rpchof	Remainder polynomial coefficients, highest order first. (Audio)
sb	Subband. (Audio)
sblimit	The number of the lowest sub-band for which no bits are allocated. (Audio)
scfsi	Scalefactor selection information. (Audio)
switch_point_l	Number of scalefactor band (long block scalefactor band) from which point on window switching is used. (Audio)
switch_point_s	Number of scalefactor band (short block scalefactor band) from which point on window switching is used. (Audio)
uimsbf	Unsigned integer, most significant bit first.
vlclbf	Variable length code, left bit first, where "left" refers to the order in which the VLC codes are written.
window	Number of the actual time slot in case of block_type==2, $0 \leq$ window ≤ 2. (Audio)

The byte order of multi-byte words is most significant byte first.

2.2.7 Constants

π	3,14159265358...
e	2,71828182845...

2.3 Method of describing bitstream syntax

The bitstream retrieved by the decoder is described in 2.4.1. Each data item in the bitstream is in bold type. It is described by its name, its length in bits, and a mnemonic for its type and order of transmission.

The action caused by a decoded data element in a bitstream depends on the value of that data element and on data elements previously decoded. The decoding of the data elements and definition of the state variables used in their decoding are described in 2.4.2. The following constructs are used to express the conditions when data elements are present, and are in normal type:

Note this syntax uses the 'C'-code convention that a variable or expression evaluating to a non-zero value is equivalent to a condition that is true.

while (condition) {
  data_element
  . . .
}
	If the condition is true, then the group of data elements occurs next in the data stream. This repeats until the condition is not true.

do {
  data_element
  . . .
	The data element always occurs at least once.

} while (condition)	The data element is repeated until the condition is not true.

if (condition) {
  data_element
  . . .
}
	If the condition is true, then the first group of data elements occurs next in the data stream.

else {
  data_element
  . . .
}
	If the condition is not true, then the second group of data elements occurs next in the data stream.

for (expr1; expr2; expr3) {
  data_element
  . . .
}
	expr1 is an expression specifying the initialization of the loop. Normally it specifies the initial state of the counter. expr2 is a condition specifying a test made before each iteration of the loop. The loop terminates when the condition is not true. expr3 is an expression that is performed at the end of each iteration of the loop, normally it increments a counter.

Note that the most common usage of this construct is as follows:

for (i = 0; i < n; i++) {
  data_element
  . . .
}
	The group of data elements occurs n times. Conditional constructs within the group of data elements may depend on the value of the loop control variable i, which is set to zero for the first occurrence, incremented to one for the second occurrence, and so forth.

As noted, the group of data elements may contain nested conditional constructs. For compactness, the { } may be omitted when only one data element follows.

data_element []	data_element [] is an array of data. The number of data elements is indicated by the context.
data_element [n]	data_element [n] is the n+1th element of an array of data.
data_element [m][n]	data_element [m][n] is the m+1,n+1 th element of a two-dimensional array of data.
data_element [l][m][n]	data_element [l][m][n] is the l+1,m+1,n+1 th element of a three-dimensional array of data.
data_element [m..n]	is the inclusive range of bits between bit m and bit n in the data_element.

While the syntax is expressed in procedural terms, it should not be assumed that 2.4.3 implements a satisfactory decoding procedure. In particular, it defines a correct and error-free input bitstream. Actual decoders must include a means to look for start codes in order to begin decoding correctly.

Definition of bytealigned function

The function bytealigned () returns 1 if the current position is on a byte boundary, that is the next bit in the bitstream is the first bit in a byte. Otherwise it returns 0.

Definition of nextbits function

The function nextbits () permits comparison of a bit string with the next bits to be decoded in the bitstream.

Definition of next_start_code function

The next_start_code function removes any zero bit and zero byte stuffing and locates the next start code.

Syntax	No. of bits	Mnemonic
next_start_code() {		
while (!bytealigned())		
zero_bit	1	'0'
while (nextbits() != '0000 0000 0000 0000 0000 0001')		
zero_byte	8	'00000000'
}		

This function checks whether the current position is bytealigned. If it is not, zero stuffing bits are present. After that any number of zero bytes may be present before the start-code. Therefore start-codes are always bytealigned and may be preceded by any number of zero stuffing bits.

2.4 Requirements

2.4.1 Specification of the coded audio bitstream syntax

2.4.1.1 Audio sequence

Syntax	No. of bits	Mnemonic
audio sequence()		
{		
while (nextbits()==syncword) {		
frame()		
}		
}		

2.4.1.2 Audio frame

Syntax	No. of bits	Mnemonic
frame()		
{		
header()		
error_check()		
audio_data()		
ancillary_data()		
}		

2.4.1.3 Header

Syntax	No. of bits	Mnemonic
header()		
{		
syncword	12	bslbf
ID	1	bslbf
layer	2	bslbf
protection_bit	1	bslbf
bitrate_index	4	bslbf
sampling_frequency	2	bslbf
padding_bit	1	bslbf
private_bit	1	bslbf
mode	2	bslbf
mode_extension	2	bslbf
copyright	1	bslbf
original/copy	1	bslbf
emphasis	2	bslbf
}		

2.4.1.4 Error check

Syntax	No. of bits	Mnemonic
error_check()		
{		
if (protection_bit==0)		
crc_check	16	rpchof
}		

2.4.1.5 Audio data, Layer I

Syntax	No. of bits	Mnemonic
audio_data() {		
for (sb=0; sb<bound; sb++)		
for (ch=0; ch<nch; ch++)		
allocation[ch][sb]	4	**uimsbf**
for (sb=bound; sb<32; sb++) {		
allocation[0][sb]	4	**uimsbf**
allocation[1][sb]=allocation[0][sb]		
}		
for (sb=0; sb<32; sb++)		
for (ch=0; ch<nch; ch++)		
if (allocation[ch][sb]!=0)		
scalefactor[ch][sb]	6	**uimsbf**
for (s=0; s<12; s++) {		
for (sb=0; sb<bound; sb++)		
for (ch=0; ch<nch; ch++)		
if (allocation[ch][sb]!=0)		
sample[ch][sb][s]	2..15	**uimsbf**
for (sb=bound; sb<32; sb++)		
if (allocation[0][sb]!=0)		
sample[0][sb][s]	2..15	**uimsbf**
}		
}		

2.4.1.6 Audio data, Layer II

```
Syntax                                                         No. of bits   Mnemonic
audio_data()
{
    for (sb=0; sb<bound; sb++)
        for (ch=0; ch<nch; ch++)
            allocation[ch][sb]                                    2..4        uimsbf
    for (sb=bound; sb<sblimit; sb++) {
        allocation[0][sb]                                         2..4        uimsbf
        allocation[1][sb]=allocation[0][sb]
    }
    for (sb=0; sb<sblimit; sb++)
        for (ch=0; ch<nch; ch++)
            if (allocation[ch][sb]!=0)
                scfsi[ch][sb]                                     2           bslbf
    for (sb=0; sb<sblimit; sb++)
        for (ch=0; ch<nch; ch++)
            if (allocation[ch][sb]!=0) {
                if (scfsi[ch][sb]==0) {
                    scalefactor[ch][sb][0]                        6           uimsbf
                    scalefactor[ch][sb][1]                        6           uimsbf
                    scalefactor[ch][sb][2]                        6           uimsbf
                }
                if ((scfsi[ch][sb]==1) || (scfsi[ch][sb]==3)) {
                    scalefactor[ch][sb][0]                        6           uimsbf
                    scalefactor[ch][sb][2]                        6           uimsbf
                }
                if (scfsi[ch][sb]==2)
                    scalefactor[ch][sb][0]                        6           uimsbf
            }
    for (gr=0; gr<12; gr++) {
        for (sb=0; sb<bound; sb++)
            for (ch=0; ch<nch; ch++)
                if (allocation[ch][sb]!=0) {
                    if (grouping[ch][sb])
                        samplecode[ch][sb][gr]                    5..10       uimsbf
                    else
                        for (s=0; s<3; s++)
                            sample[ch][sb][3*gr+s]                3..16       uimsbf
                }
        for (sb=bound; sb<sblimit; sb++)
            if (allocation[0][sb]!=0) {
                if (grouping[0][sb])
                    samplecode[0][sb][gr]                         5..10       uimsbf
                else
                    for (s=0; s<3; s++)
                        sample[0][sb][3*gr+s]                     3..16       uimsbf
            }
    }
}
```

2.4.1.7 Audio data, Layer III

Syntax	No. of bits	Mnemonic
audio_data()		
{		
main_data_begin	**9**	**uimsbf**
if (mode==single_channel)		
private_bits	**5**	**bslbf**
else		
private_bits	**3**	**bslbf**
for (ch=0; ch<nch; ch++)		
for (scfsi_band=0; scfsi_band<4; scfsi_band++)		
scfsi[ch][scfsi_band]	**1**	**bslbf**
for (gr=0; gr<2; gr++)		
for (ch=0; ch<nch; ch++) {		
part2_3_length[gr][ch]	**12**	**uimsbf**
big_values[gr][ch]	**9**	**uimsbf**
global_gain[gr][ch]	**8**	**uimsbf**
scalefac_compress[gr][ch]	**4**	**bslbf**
window_switching_flag[gr][ch]	**1**	**bslbf**
if (window_switching_flag[gr][ch]) {		
block_type[gr][ch]	**2**	**bslbf**
mixed_block_flag[gr][ch]	**1**	**uimsbf**
for (region=0; region<2; region++)		
table_select[gr][ch][region]	**5**	**bslbf**
for (window=0; window<3; window++)		
subblock_gain[gr][ch][window]	**3**	**uimsbf**
}		
else {		
for (region=0; region<3; region++)		
table_select[gr][ch][region]	**5**	**bslbf**
region0_count[gr][ch]	**4**	**bslbf**
region1_count[gr][ch]	**3**	**bslbf**
}		
preflag[gr][ch]	**1**	**bslbf**
scalefac_scale[gr][ch]	**1**	**bslbf**
count1table_select[gr][ch]	**1**	**bslbf**
}		
main_data ()		
}		

The main data bitstream is defined below. The **main_data** field in the audio_data() syntax contains bytes from the main data bistream. However, because of the variable nature of Huffman coding used in Layer III, the main data for a frame does not generally follow the header and side information for that frame. The main_data for a frame starts at a location in the bitstream preceeding the header of the frame at a negative offset given by the value of main_data_begin. (See definition of main_data_begin and figure A.7.a).

Syntax	No. of bits	Mnemonic
main_data()		
{		
for (gr=0; gr<2; gr++) {		
for (ch=0; ch<nch; ch++) {		
if ((window_switching_flag[gr][ch]==1)		
&& (block_type[gr][ch]==2)) {		
if (mixed_block_flag[gr][ch]) {		
for (sfb=0; sfb<8; sfb++)		
scalefac_l[gr][ch][sfb]	**0..4**	**uimsbf**
for (sfb=3; sfb<12; sfb++)		
for (window=0; window<3; window++)		
scalefac_s[gr][ch][sfb][window]	**0..4**	**uimsbf**
}		
else {		
for (sfb=0; sfb<12; sfb++)		
for (window=0; window<3; window++)		
scalefac_s[gr][ch][sfb][window]	**0..4**	**uimsbf**
}		
}		
else {		
if ((scfsi[ch][0]==0) || (gr == 0))		
for(sfb=0;sfb<6;sfb++)		
scalefac_l[gr][ch][sfb]	**0..4**	**uimsbf**
if ((scfsi[ch][1]==0) || (gr == 0))		
for(sfb=6;sfb<11;sfb++)		
scalefac_l[gr][ch][sfb]	**0..4**	**uimsbf**
if ((scfsi[ch][2]==0) || (gr == 0))		
for(sfb=11;sfb<16;sfb++)		
scalefac_l[gr][ch][sfb]	**0..3**	**uimsbf**
if ((scfsi[ch][3]==0) || (gr == 0))		
for(sfb=16;sfb<21;sfb++)		
scalefac_l[gr][ch][sfb]	**0..3**	**uimsbf**
}		
Huffmancodebits()		
}		
}		
for (b=0; b<no_of_ancillary_bits; b++)		
ancillary_bit	**1**	**bslbf**
}		

Syntax	No. of bits	Mnemonic
Huffmancodebits() {		
for (l=0; l<big_values*2; l+=2) {		
hcod[|x|][|y|]	0..19	bslbf
if (|x|==15 && linbits>0)		
linbitsx	1..13	uimsbf
if (x != 0)		
signx	1	bslbf
if (|y|==15 && linbits>0)		
linbitsy	1..13	uimsbf
if (y != 0)		
signy	1	bslbf
is[l] = x		
is[l+1] = y		
}		
for (; l<big_values*2+count1*4; l+=4) {		
hcod[|v|][|w|][|x|][|y|]	1..6	bslbf
if (v!=0)		
signv	1	bslbf
if (w!=0)		
signw	1	bslbf
if (x!=0)		
signx	1	bslbf
if (y!=0)		
signy	1	bslbf
is[l] = v		
is[l+1] = w		
is[l+2] = x		
is[l+3] = y		
}		
for (; l<576; l++)		
is[l] = 0		
}		

2.4.1.8 Ancillary data

Syntax	No. of bits	Mnemonic
ancillary_data() {		
if ((layer == 1) || (layer == 2))		
for (b=0; b<no_of_ancillary_bits; b++)		
ancillary_bit	1	bslbf
}		

2.4.2 Semantics for the audio bitstream syntax

2.4.2.1 Audio sequence general

frame -- Layer I and Layer II: Part of the bitstream that is decodable by itself. In Layer I it contains information for 384 samples and in Layer II for 1 152 samples. It starts with a syncword, and ends just before the next syncword. It consists of an integer number of slots (four bytes in Layer I, one byte in Layer II).

-- Layer III: Part of the bitstream that is decodable with the use of previously acquired main information. In Layer III it contains information for 1 152 samples. Although the distance between the start of consecutive syncwords is an integer number of slots (one byte in Layer III), the audio information belonging to one frame is generally not contained between two successive syncwords.

2.4.2.2 Audio frame

header -- Part of the bitstream containing synchronization and state information.

error_check -- Part of the bitstream containing information for error detection.

audio_data -- Part of the bitstream containing information on the audio samples.

ancillary_data -- Part of the bitstream that may be used for ancillary data.

2.4.2.3 Header

The first 32 bits (four bytes) are header information which is common to all layers.

syncword -- The bit string '1111 1111 1111'.

ID -- One bit to indicate the ID of the algorithm. Equals '1' for ISO/IEC 11172-3 audio, '0' is reserved.

Layer -- 2 bits to indicate which layer is used, according to the following.

Layer	
'11'	Layer I
'10'	Layer II
'01'	Layer III
'00'	reserved

To change the layer, a reset of the audio decoder may be required.

protection_bit -- One bit to indicate whether redundancy has been added in the audio bitstream to facilitate error detection and concealment. Equals '1' if no redundancy has been added, '0' if redundancy has been added.

bitrate_index -- Indicates the bitrate. The all zero value indicates the 'free format' condition, in which a fixed bitrate which does not need to be in the list can be used. Fixed means that a frame contains either N or N+1 slots, depending on the value of the padding bit. The bitrate_index is an index to a table, which is different for the different layers.

The bitrate_index indicates the total bitrate irrespective of the mode (stereo, joint_stereo, dual_channel, single_channel).

bitrate_index	bitrate specified (kbits/s)		
	Layer I	Layer II	Layer III
'0000'	free	free	free
'0001'	32	32	32
'0010'	64	48	40
'0011'	96	56	48
'0100'	128	64	56
'0101'	160	80	64
'0110'	192	96	80
'0111'	224	112	96
'1000'	256	128	112
'1001'	288	160	128
'1010'	320	192	160
'1011'	352	224	192
'1100'	384	256	224
'1101'	416	320	256
'1110'	448	384	320
'1111'	forbidden	forbidden	forbidden

In order to provide the smallest possible delay and complexity, the decoder is not required to support a continuously variable bitrate when in Layer I or II. Layer III supports variable bitrate by switching the bitrate_index. The switching of the bitrate_index can be used either to optimize storage requirements on DSM or to interpolate any mean data rate by switching between nearby values in the bitrate table. However, in free format, fixed bitrate is required. The decoder is also not required to support bitrates higher than 448 kbits/s, 384 kbits/s, 320 kbits/s in respect to Layer I, II and III when in free format mode.

For Layer II, not all combinations of total bitrate and mode are allowed. See the following table.

bit rate (kbits/s)	Allowed modes
free format	all modes
32	single_channel
48	single_channel
56	single_channel.
64	all modes
80	single_channel
96	all modes
112	all modes
128	all modes
160	all modes
192	all modes
224	stereo, intensity stereo, dual channel
256	stereo, intensity stereo, dual channel
320	stereo, intensity stereo, dual channel
384	stereo, intensity stereo, dual channel

sampling_frequency -- Indicates the sampling frequency, according to the following table.

sampling_frequency	frequency specified (kHz)
'00'	44,1
'01'	48
'10'	32
'11'	reserved

A reset of the audio decoder may be required to change the sampling rate.

padding_bit -- If this bit equals '1', the frame contains an additional slot to adjust the mean bitrate to the sampling frequency, otherwise this bit will be '0'. Padding is necessary with a sampling frequency of 44,1 kHz. Padding may also be required in free format.

The padding should be applied to the bitstream such that the accumulated length of the coded frames, after a certain number of audio frames does not deviate more than (+0, -1 slot) from the following computed value:

$$\text{accumulated frame length} = \sum_{\text{first frame}}^{\text{current frame}} (\text{frame\_size} * \text{bitrate} / \text{sampling frequency})$$

where frame_size = 384 for Layer I,
1 152 for Layer II or III.

The following method can be used to determine whether or not to use padding:

```
for 1st audio frame:
        rest = 0;
        padding = no;

for each subsequent audio frame:
        if (Layer == 1) dif = (12 * bitrate) % sampling_frequency;
        else dif = (144 * bitrate) % sampling_frequency;
        rest = rest - dif;
        if (rest < 0) {
                padding = yes;
                rest = rest + sampling_frequency;
        }
        else padding = no;
```

private_bit -- Bit for private use. This bit will not be used in the future by ISO/IEC.

mode -- Indicates the mode according to the following table. In Layer I and II the joint_stereo mode is intensity_stereo, in Layer III it is intensity_stereo and/or ms_stereo.

mode	mode specified
'00'	stereo
'01'	joint_stereo (intensity_stereo and/or ms_stereo)
'10'	dual_channel
'11'	single_channel

In Layer I, in all modes except joint stereo, the value of bound equals 32. In layer II, in all modes except joint_stereo, the value of bound equals sblimit. In joint_stereo mode the bound is determined by the mode_extension.

mode_extension -- These bits are used in joint_stereo mode. In Layer I and II they indicate which subbands are in intensity_stereo. All other subbands are coded in stereo.

mode_extension	
'00'	subbands 4-31 in intensity_stereo, bound=4
'01'	subbands 8-31 in intensity_stereo, bound=8
'10'	subbands 12-31 in intensity_stereo, bound=12
'11'	subbands 16-31 in intensity_stereo, bound=16

In Layer III they indicate which type of joint stereo coding method is applied. The frequency ranges over which the intensity_stereo and ms_stereo modes are applied are implicit in the algorithm. For more information see 2.4.3.4.

mode_extension	intensity_stereo	ms_stereo
'00'	off	off
'01'	on	off
'10'	off	on
'11'	on	on

Note that the mode "stereo" is used if the mode bits specify stereo or equivalently if the mode bits specify joint stereo and the mode_extension specifies intensity_stereo "off" and ms_stereo "off".

copyright -- If this bit equals '0', there is no copyright on the ISO/IEC 11172-3 bitstream, '1' means copyright protected.

original/copy -- This bit equals '0' if the bitstream is a copy, '1' if it is an original.

emphasis -- Indicates the type of de-emphasis that shall be used.

emphasis	emphasis specified
'00'	none
'01'	50/15 microseconds
'10'	reserved
'11'	CCITT J.17

2.4.2.4 Error check

crc_check -- A 16 bit parity-check word is used for optional error detection within the encoded bitstream.

2.4.2.5 Audio data, Layer I

allocation[ch][sb] -- Indicates the number of bits used to code the samples in subband sb of channel ch. For subbands in intensity_stereo mode the bitstream contains only one allocation data element per subband.

allocation[ch][sb]	bits per sample
0	0
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10
10	11
11	12
12	13
13	14
14	15
15	forbidden

Note: For code '0000' no samples are transferred.

scalefactor[ch][sb] -- Indicates the factor of subband sb of channel ch by which the requantized samples of subband sb in channel ch shall be multiplied. The six bits constitute an unsigned integer, index to table B.1 "Layer I, II scalefactors".

sample[ch][sb][s] -- Coded representation of the s-th sample in subband sb of channel ch. For subbands in intensity_stereo mode the coded representation of the sample is valid for both channels.

2.4.2.6 Audio data, Layer II

allocation[ch][sb] -- Contains information concerning the quantizers used for the samples in subband sb in channel ch, whether the information on three consecutive samples has been grouped to one code, and on the number of bits used to code the samples. The meaning and length of this field depends on the number of the subband, the bitrate, and the sampling frequency. The bits in this field form an unsigned integer used as an index to the relevant table in table B.2 "Layer II bit allocation tables", which gives the number of levels used for quantization. For subbands in intensity_stereo mode the bitstream contains only one allocation data element per subband.

scfsi[ch][sb] -- Scalefactor selection information. This gives information on the number of scalefactors transferred for subband sb in channel ch and for which parts of the signal in this frame they are valid. The frame is divided into three equal parts of 12 subband samples each per subband.

scfsi[sb]	
'00'	three scalefactors transmitted, for parts 0,1,2 respectively.
'01'	two scalefactors transmitted, first one valid for parts 0 and 1, second one for part 2.
'10'	one scalefactor transmitted, valid for all three parts.
'11'	two scalefactors transmitted, first one valid for part 0, the second one for parts 1 and 2.

scalefactor[ch][sb][p] -- Indicates the factor by which the requantized samples of subband sb in channel ch and of part p of the frame should be multiplied. The six bits constitute an unsigned integer, index to table B.1 "Layer I, II scalefactors'.

grouping[ch][sb] -- Is a function that determines, whether grouping is in effect for coding of samples in subband sb of channel ch. Grouping means, that three consecutive samples (a triplet) of the current subband sb in channel ch in the current granule gr are coded and transmitted using one common codeword and not using three separate codewords. Grouping[ch][sb] is true, if in the Bit Allocation table currently in use (see B.2) the value found under the sb (row) and the allocation[sb] (column) is either 3, 5, or 9. Otherwise it is false. For subbands in intensity_stereo mode the grouping is valid for both channels.

samplecode[ch][sb][gr] -- Coded representation of the three consecutive samples in the granule gr in subband sb of channel ch. For subbands in intensity_stereo mode the coded representation of the samplecode is valid for both channels.

sample[ch][sb][s] -- Coded representation of the s-th sample in subband sb of channel ch. For subbands in intensity_stereo mode the coded representation of the sample is valid for both channels.

2.4.2.7 Audio data, Layer III

main_data_begin -- The value of main_data_begin is used to determine the location of the first bit of main data of a frame. The main_data_begin value specifies the location as a negative offset in bytes from the first byte of the audio sync word. The number of bytes belonging to the header and side information is not taken into account. For example, if main_data_begin == 0, then main data starts after the side information. Examples are given in figure A.7.a and figure A.7.b.

private_bits -- Bits for private use. These bits will not be used in the future by ISO/IEC. The number of private_bits depends on the number of channels. The number of bits allocated for private_bits is determined to equalize the total number of bits used for side-information.

scfsi[ch][scfsi_band] -- In Layer III, the scalefactor selection information works similarly to audio Layer II. The main difference is the use of the variable scfsi_band to apply scfsi to groups of scalefactors instead of single scalefactors. The application of scalefactors to granules is controlled by scfsi.

scfsi[scfsi_band]	
'0'	scalefactors are transmitted for each granule
'1'	scalefactors transmitted for granule 0 are also valid for granule 1

If short windows are switched on, i.e. block_type==2 for one of the granules, then scfsi is always 0 for this frame.

scfsi_band controls the use of the scalefactor selection information for groups of scalefactors (scfsi_bands).

scfsi_band	scalefactor bands (see table B.8)
0	0,1,2,3,4,5,
1	6,7,8,9,10,
2	11 ... 15
3	16 ... 20

part2_3_length[gr][ch] -- This value contains the number of main_data bits used for scalefactors and Huffman code data. Because the length of the side information is always the same, this value can be used to calculate the beginning of the main information for the next granule or the position of the ancillary information (if used). Note that single channel audio frames contain 17 bytes of side information and dual channel audio frames contain 32 bytes of side information (see 2.4.1.7 Audio Data, Layer III - syntax for audio_data()).

big_values[gr][ch] -- The spectral values of each granule are coded with different Huffman code tables.The full frequency range from zero to the Nyquist frequency is divided into several regions, which then are coded using different tables. Partitioning is done according to the maximum quantized values. This is done with the assumption that values at higher frequencies are expected to have lower amplitudes or do not need to be coded at all. Starting at high frequencies, the pairs of quantized values equal to zero are counted. This number is named "rzero". Then, quadruples of quantized values with absolute value not exceeding 1 (i.e. only 3 possible quantization levels) are counted. This number is named "count1". Again an even number of values remain. Finally, the number of pairs of values in the region of the spectrum which extends down to zero is named "big_values". The maximum absolute value in this range is constrained to 8191. The following figure shows the partitioning:

```
xxxxxxxxxxxxxx------------------0000000000000000000000000000
|             |                 |                          |
1             bigvalues*2       bigvalues*2+count1*4       iblen
```

The values 000 are all zero. Their number is a multiple of 2.
The values --- are -1,0 or +1. Their number is a multiple of 4.
The values xxx are not bound. Their number is a multiple of 2.
Iblen is 576.

global_gain[gr][ch] -- The quantizer step size information is transmitted in the side information variable global_gain. It is logarithmically quantized. For the application of global_gain, refer to the formula in 2.4.3.4, "Formula for requantization and all scaling".

scalefac_compress[gr][ch] -- Selects the number of bits used for the transmission of the scalefactors according to the following table:

if **block_type** is 0, 1, or 3:
slen1: length of scalefactors for the scalefactor bands 0 to 10
slen2: length of scalefactors for the scalefactor bands 11 to 20

if **block_type** is 2 and mixed_block_flag is 0:
slen1: length of scalefactors for the scalefactor bands 0 to 5
slen2: length of scalefactors for the scalefactor bands 6 to 11

if **block_type** is 2 and mixed_block_flag is 1:
slen1: length of scalefactors for the scalefactor bands 0 to 7 (long window scalefactor band) and 3 to 5 (short window scalefactor band) Note: Scalefactor bands 0-7 are from the "long window

scalefactor band" table, and scalefactor bands 3 to 11 from the "short window scalefactor band" table. This combination of partitions is contiguous and spans the entire frequency spectrum.
slen2: length of scalefactors for the scalefactor bands 6 to 11

scalefac_compress[gr]	slen1	slen2
0	0	0
1	0	1
2	0	2
3	0	3
4	3	0
5	1	1
6	1	2
7	1	3
8	2	1
9	2	2
10	2	3
11	3	1
12	3	2
13	3	3
14	4	2
15	4	3

window_switching_flag[gr][ch] -- Signals that the block uses an other than normal (type 0) window.

If window_switching_flag is set, several other variables are set by default:

region0_count = 7 (in case of **block_type**==1 or **block_type**==3 or **block_type**==2 and mixed_block_flag)
region0_count = 8 (in case of **block_type**==2 and not mixed_block_flag)
region1_count = 36 Thus all remaining values in the big_value region are contained in region 1.

If window_switching_flag is not set, then the value of **block_type** is zero.

block_type[gr][ch] -- Indicates the window type for the granule (see description of the filterbank, Layer III).

block_type[gr]	
0	reserved
1	start block
2	3 short windows
3	end block

Block_type and **mixed_block_flag** give the information about assembling of values in the block and about length and count of the transforms (see figure A.4 for a schematic, annex C for an analytic description). If window_switching_flag==1, then the mixed_block_flag indicates whether lower frequency polyphase filter subbands are coded using normal window type. The polyphase filterbank is described in 2.4.3.

In the case of long blocks (block_type not equal to 2 or in the lower subbands of block_type 2 if the mixed_block_flag is set) the IMDCT generates an output of 36 values every 18 input values. The output is windowed depending on the block_type and the first half is overlapped with the second half of the block before. The resulting vector is the input of the synthesis part of the polyphase filterbank of one band.

In the case of short blocks (in the upper subbands of a type 2 block if the mixed_block_flag is set, or in all subbands of a type 2 block if mixed_block_flag is not set), three transforms are performed producing 12 output values each. The three vectors are windowed and overlapped each. Concatenating 6 zeros on both ends of the resulting vector gives a vector of length 36, which is processed like the output of a long transform.

mixed_block_flag[gr][ch] -- Indicates that lower frequencies are transformed with a window type that is different than that which is used at higher frequencies. If mixed_block_flag is zero, then all blocks are transformed as indicated by block_type[gr][ch]. If mixed_block_flag is one, then the frequency lines corresponding to the two lowest frequency polyphase subbands are transformed with normal window (block_type==0), while the remaining 30 subbands are transformed as block_type[gr][ch].

table_select[gr][ch][region] -- Different Huffman code tables are used depending on the maximum quantized value and the local statistics of the signal. There are a total of 32 possible tables given in table B.7.

subblock_gain[gr][ch][window] -- Indicates the gain offset (quantization: factor 4) from the global gain for one subblock. Used only with block type 2 (short windows). The values of the subblock have to be divided by $4^{(\text{subblock\_gain}[\text{window}])}$ in the decoder. See 2.4.3.4 - Formula for requantization and all scaling.

region0_count[gr][ch] -- A further partitioning of the spectrum is used to enhance the performance of the Huffman coder. It is a subdivision of the region which is described by big_values. The purpose of this subdivision is to get better error robustness and better coding efficiency. Three regions are used, they are named: region 0, 1 and 2. Each region is coded using a different Huffman code table depending on the maximum quantized value and the local signal statistics.

The values region0_count and region1_count are used to indicate the boundaries of the regions. The region boundaries are aligned with the partitioning of the spectrum into scale factor bands.

The field region0_count contains one less than the number of scalefactor bands in region 0. In the case of short blocks, each scale factor band is counted three times, once for each short window, so that a region0_count value of 8 indicates that region1 begins at scalefactor band number 3.

If block_type==2 and mixed_block_flag==0, the total amount of scalefactor bands for the granule in this case is 12*3=36. If block_type==2 and mixed_block_flag==1, the amount of scalefactor bands is 8+9*3=35. If block_type!=2, the amount of scalefactor bands is 21.

region1_count[gr][ch] -- region1_count counts one less than the number of scalefactor bands in region 1. Again, if block_type==2 the scalefactor bands representing different time slots are counted separately.

preflag[gr][ch] -- This is a shortcut for additional high frequency amplification of the quantized values. If preflag is set, the values of a table are added to the scalefactors (see table B.6). This is equivalent to multiplication of the requantized scalefactors with table values. If block_type==2 (short blocks) preflag is never used.

scalefac_scale[gr][ch] -- The scalefactors are logarithmically quantized with a step size of 2 or ($\sqrt{2}$) depending on scalefac_scale. The following table indicates the scale factor multiplier used in the requantization equation for each stepsize.

scalefac_scale[gr]	scalefac_multiplier
0	0,5
1	1

count1table_select[gr][ch] -- This flag selects one of two possible Huffman code tables for the region of quadruples of quantized values with magnitude not exceeding 1.

count1table_select[gr]	
0	Table B.7 - A
1	Table B.7 - B

scalefac_l[gr][ch][sfb], scalefac_s[gr][ch][sfb][window], is_pos[sfb] -- The scalefactors are used to colour the quantization noise. If the quantization noise is coloured with the right shape, it is masked completely. Unlike Layers I and II, the Layer III scalefactors say nothing about the local maximum of the quantized signal. In Layer III, scalefactors are used in the decoder to get division factors for groups of values. In the case of Layer III, the groups stretch over several frequency lines. These groups are called scalefactor bands and are selected to resemble critical bands as closely as possible.

The scalefac_compress table shows that the scalefactors 0...10 have a range of 0 to 15 (maximum length 4 bits) and the scalefactors 11...21 have a range of 0 to 7 (maximum length 3 bits).

If intensity_stereo is enabled (modebit_extension) the scalefactors of the "zero_part" of the difference (right) channel are used as intensity_stereo positions, is_pos[sfb] (see 2.4.3.4, MS_stereo mode). is_pos[sfb] is the intensity stereo position for scalefactor band sfb.

The subdivision of the spectrum into scalefactor bands is fixed for every block length and sampling frequency and stored in tables in the coder and decoder (see table B.8). The scale factor for frequency lines above the highest line in the tables is zero, which means that the actual multiplication factor is 1,0.

The scalefactors are logarithmically quantized. The quantization step is set with scalefac_scale.

huffmancodebits() -- Huffman encoded data.

The syntax for huffmancodebits() shows how quantized values are encoded. Within the big_values partition, pairs of quantized values with an absolute value less than 15 are directly coded using a Huffman code. The codes are selected from huffman tables 0 through 31 in table B.7. Always pairs of values (x,y) are coded. If quantized values of magnitude greater than or equal to 15 are coded, the values are coded with a separate field following the huffman code. If one or both values of a pair is not zero, one or two sign bits are appended to the code word.

The Huffman tables for the big_values partition are comprised of three parameters:

hcod[|x|][|y|]	is the Huffman code table entry for values x,y.
hlen[|x|][|y|]	is the Huffman length table entry for values x,y.
linbits	is the length of linbitsx or linbitsy when they are coded.

The syntax for huffmancodebits contains the following fields and parameters:

signv	is the sign of v (0 if positive, 1 if negative).
signw	is the sign of w (0 if positive, 1 if negative).
signx	is the sign of x (0 if positive, 1 if negative).
signy	is the sign of y (0 if positive, 1 if negative).
linbitsx	is used to encode the value of x if the magnitude of x is greater or equal to 15. This field is coded only if |x| in hcod is equal to 15. If linbits is zero, so that no bits are actually coded when |x|==15, then the value linbitsx is defined to be zero.
linbitsy	is the same as linbitsx but for y.
is[l]	Is the quantized value for frequency line number l.

The linbitsx or linbitsy fields are only used if a value greater or equal to 15 needs to be encoded. These fields are interpreted as unsigned integers and added to 15 to obtain the encoded value. The linbitsx and linbitsy fields are never used if the selected table is one for blocks with a maximum quantized value less than 15. Note that a value of 15 can still be encoded with a huffman table for which linbits is zero. In this case, the linbitsx or linbitsy fields are not actually coded, since linbits is zero.

Within the count1 partition, quadruples of values with magnatude less than or equal to one are coded. Again magnitude values are coded using a Huffman code from tables A or B in table B.7. Again, for each non-zero value, a sign bit is appended after the huffman code symbol.

The Huffman tables for the count1 partition are comprised of the following parameters:

hcod[|v|][|w|][|x|][|y|]	is the Huffman code table entry for values v,w,x,y.
hlen[|v|][|w|][|x|][|y|]	is the Huffman length table entry for values v,w,x,y.

Huffman code table B is not really a 4-dimensional code because it is constructed from the trivial code: 0 is coded with a 1, and 1 is coded with a 0.

Quantized values above the count1 partition are zero, so they are not encoded.

For clarity, the parameter "count1" is used in this document to indicate the number of Huffman codes in the count1 region. However, unlike the bigvalues partition, the number of values in the count1 partition is not explicitly coded by a field in the syntax. The end of the count1 partition is known only when all bits for the granule (as specified by part2_3_length), have been exhausted, and the value of count1 is known implicitly after decoding the count1 region.

The order of the Huffman data depends on the block_type of the granule. If block_type is 0, 1 or 3 the Huffman encoded data is ordered in terms of increasing frequency.

If block_type==2 (short blocks) the Huffman encoded data is ordered in the same order as the scalefactor values for that granule. The Huffman encoded data is given for successive scalefactor bands, beginning with scalefactor band 0. Within each scalefactor band, the data is given for successive time windows, beginning with window 0 and ending with window 2. Within each window, the quantized values are then arranged in order of increasing frequency.

2.4.2.8 Ancillary data

Ancillary_bit -- User definable.

The number of ancillary bits (no_of_ancillary_bits) equals the available number of bits in an audio frame minus the number of bits actually used for header, error check and audio data. In Layer I and II the no_of_ancillary_bits corresponds to the distance between the end of the audio data and the beginning of the next header. In Layer III the no_of_ancillary_bits corresponds to the distance between the end of the Huffman_code_bits and the location in the bitstream where the next frame's main_data_begin pointer points to.

2.4.3 The audio decoding process

2.4.3.1 General

The first action is synchronization of the decoder to the incoming bitstream. Just after startup this may be done by searching in the bitstream for the 12 bit syncword. In some applications the ID, layer, and protection status are already known to the decoder, and thus the first 16 bits of the header should be regarded as a 16 bit syncword, thereby allowing a more reliable synchronization. The position of consecutive syncwords can be calculated from the information provided by the seven bits after the protection_bit: the bitstream is subdivided in slots. The distance between the start of two consecutive syncwords is equal to "N" or "N+1" slots. The value of "N" depends on the layer.

For Layer I the following equation is valid:

$$N = 12 * \frac{\text{bitrate}}{\text{sampling\_frequency}}.$$

For Layers II and III the equation becomes:

$$N = 144 * \frac{\text{bitrate}}{\text{sampling\_frequency}}.$$

If this calculation does not give an integer number the result is truncated and 'padding' is required. In this case the number of slots in a frame will vary between N and N+1. The padding bit is set to '0' if the number of slots equals N, and to '1' otherwise. This knowledge of the position of consecutive syncwords greatly facilitates synchronization.

If the bitrate index equals '0000', the exact bitrate is not indicated. N can be determined from the distance between consecutive syncwords and the value of the padding bit.

The mode bits in the bitstream shall be read and if their value is '01' the mode_extension bits shall also be read. The mode_extension bits set the 'bound' as shown in 2.4.2.3 and thus indicate which subbands are coded in joint_stereo mode.

If the protection bit in the header equals '0', a CRC-check word has been inserted in the bitstream just after the header. The error detection method used is 'CRC-16' whose generator polynomial is:

$$G(X) = X^{16} + X^{15} + X^2 + 1$$

The bits included into the CRC-check are given by table B.5.

The method is depicted in figure A.9 "CRC-check diagram". The initial state of the shift register is '1111 1111 1111 1111'. Then all the bits included into the CRC-check are input to the circuit shown in figure A.9 "CRC-check diagram". After each bit is input the shift register is shifted by one bit. After the last shift operation, the outputs $b_{15}...b_0$ constitute a word to be compared with the CRC-check word in the bitstream. If the words are not identical, a transmission error has occurred in the protected field of the bitstream. To avoid annoying distortions, application of a concealment technique, such as muting of the actual frame or repetition of the previous frame, is recommended.

2.4.3.2 Layer I

After the part of the decoding which is common to all layers (see 2.4.3.1) the bit allocation information has to be read for all subbands, and the scalefactors read for all subbands with a nonzero bit allocation. The decoder flowchart is given in figure A.1 "Layer I and II decoder flow chart".

2.4.3.2.1 Requantization of subband samples

From the bit allocation the number of bits nb that has to be read for the samples in each subband is known. The order of the samples is given in 2.4.1.5 for each mode. After the bits for one sample have been gathered from the bitstream, the first bit has to be inverted. The resulting number can be considered as a two's complement fractional number, where the MSB represents the value -1. The requantized value can be obtained by applying a linear formula :

$$s'' = \frac{2^{nb}}{2^{nb} - 1} * \left(s''' + 2^{-nb+1}\right)$$

where

s'''	is the fractional number;
s''	is the requantized value;
nb	is the number of bits allocated to samples in the subband.

Samples in subbands which are in intensity_stereo mode must be copied to both channels. The requantized value has to be rescaled. The multiplication factor can be found in the table B.1 "Layer I, II scalefactors". The rescaled value s' is calculated as :

$$s' = \text{factor} * s''$$

2.4.3.2.2 Synthesis subband filter

If a subband has no bits allocated to it, the samples in that subband are set to zero. Each time the subband samples for all 32 subbands of one channel have been calculated, they can be applied to the synthesis subband filter and 32 consecutive audio samples can be calculated. The actions in flow diagram figure A.2 "Synthesis subband filter flow chart" show the reconstruction operation. The coefficients N_{ik} for the matrixing operation are given by

$$N_{ik} = \cos\left[(16 + i)\,(2k+1)\,\frac{\pi}{64}\right] \qquad 0 \le i \le 63,\ 0 \le k \le 31$$

The coefficients D_i for the windowing operation can be found in table B.3 "Coefficients D_i of the synthesis window". The coefficients have been derived by numerical optimization. One frame contains 12 * 32 = 384 subband samples, which result, after filtering, in 384 audio samples.

2.4.3.3 Layer II

Layer II is a more efficient but more complex coding scheme than Layer I. The flowchart in figure A.1 "Layer I and II decoder flow chart" applies to both Layers I and II. The first step is to perform the decoding which is common to all three layers (see 2.4.3.1).

2.4.3.3.1 Bit allocation decoding

For different combinations of bitrate and sampling frequency different bit allocation tables exist (table B.2 "Layer II bit allocation tables"). Note that the bitrates given in the table headers are per channel. If the mode is not single_channel, the bitrate should be divided by two to obtain the bitrate per channel. The decoding of the bit allocation table is done in a three-step approach. The first step consists of reading 'nbal' (2,3, or 4) bits of information for one subband from the bitstream. The value of 'nbal' is given in the second column of the relevant table B.2 "Layer II bit allocation tables". These bits shall be interpreted as an unsigned integer number. The second step uses this number and the number of the subband as indices to point to a value in the table. This value represents the number of levels 'nlevels' used to quantize the samples in the subband. As a third step, using table B.4 "Layer II classes of quantization", the number of bits used to code the quantized samples, the requantization coefficients, and whether the codes for three consecutive subband samples have been grouped to one code can be determined. It can be seen from the bit allocation tables that some of the highest subbands will never have bits allocated. The number of the lowest subband that will not have bits allocated to it is assigned to the identifier 'sblimit'.

2.4.3.3.2 Scalefactor selection information decoding

The 36 samples in one subband within a frame are divided in three equal parts of 12 subband samples. Each part can have its own scalefactor. The number of scalefactors that has to be read from the bitstream depends on scfsi[sb]. The scalefactor selection information scfsi[sb] is read from the bitstream for the subbands that have a nonzero bit allocation. If scfsi[sb] equals '00' three scalefactors are transmitted, for parts 0,1,2 respectively. If scfsi[sb] equals '01' two scalefactors are transmitted, the first one valid for parts 0 and 1, the second one for part 2. If scfsi[sb] equals '10' one scalefactor is transmitted, valid for all three parts. If scfsi[sb] equals '11' two scalefactors are transmitted, the first one valid for part 0, the second one for parts 1 and 2.

2.4.3.3.3 Scalefactor decoding

For every subband with a nonzero bit allocation the coded scalefactors for that subband are read from the bitstream. The number of coded scalefactors and the part of the subband samples they refer to is defined by scfsi[sb]. The 6 bits of a coded scalefactor should be interpreted as an unsigned integer index to table B.1 "Layer I, II scalefactors". This table gives the scalefactor by which the relevant subband samples should be multiplied after requantization.

2.4.3.3.4 Requantization of subband samples

Next the coded samples are read. As can be seen from 2.4.1.6, the coded samples appear as triplets, the code contains three consecutive samples at a time. From table B.4 "Layer II classes of quantization" it is known how many bits have to be read for one triplet from the bitstream for each subband. Also from table B.4 "Layer II classes of quantization", it is known whether this code consists of three consecutive separable codes for each sample or of one combined code for the three samples (grouping). In the last case degrouping must be performed. The combined code has to be regarded as an unsigned integer, called 'c'. The following algorithm will supply the three separate codes s[0], s[1], s[2].

```
for (i=0; i<3; i++) {
        s[i]= c % nlevels
        c  = c DIV nlevels
}
```

where nlevels is the number of steps as shown in table B.2 "Layer II bit allocation table".

The first bit of each of the three codes has to be inverted, and the resulting numbers should be regarded as two's complement fractional numbers, where the MSB represents the value -1. The requantized values can be obtained by applying a linear formula :

$$s'' = C * (s''' + D)$$

where

s''' is the fractional number;

s'' is the requantized value.

The values of the constants C and D are given in table B.4 "Layer II classes of quantization". The requantized values have to be rescaled. The multiplication factors can be found in the table B.1 "Layer I, II scalefactors". as described above. The rescaled value s' is calculated as :

$$s' = \text{factor} * s''$$

2.4.3.3.5 Synthesis subband filter

If a subband has no bits allocated to it, the samples in that subband are set to zero. Each time the subband samples for all 32 subbands of one channel have been calculated, they can be applied to the synthesis subband filter and 32 consecutive audio samples can be calculated. For that purpose, the actions in the flow diagram of figure A.2 "Synthesis subband filter flow chart" have to be carried out. The coefficients N_{ik} for the matrixing operation are given by

$$N_{ik} = \cos\left[(16 + i)(2k+1)\frac{\pi}{64}\right] \qquad 0 \le i \le 63,\ 0 \le k \le 31$$

The coefficients D_i for the windowing operation can be found in table B.3 "Coefficients D_i of the synthesis window". One frame contains 36 * 32 = 1 152 subband samples, which result after filtering in 1 152 audio samples.

2.4.3.4 Layer III

Additional frequency resolution is provided by the use of an hybrid filterbank. Every band is split into 18 frequency lines by use of an MDCT. The window length of the MDCT is 36. Adaptive window switching is used to control time artifacts (pre-echoes), see the description in annex C. The frequency above which shorter blocks (better time resolution) are used can be selected. Parts of the signal below a frequency depending on "mixed_block_flag" are coded with better frequency resolution, parts of the signal above are coded with better time resolution.

The frequency components are quantized using a nonuniform quantizer and coded using a Huffman encoder. The Huffman coder uses one of 18 different tables (see table B.7). A buffer is used to help enhance the coding efficiency of the Huffman coder and to help in the case of pre-echo conditions (see the description in annex C). The size of the input buffer is the size of one frame at the bitrate of 160 kbits/s per channel for Layer III. The short term buffer technique used is called "bit reservoir" because it uses short-term variable bitrate with a maximum integral offset from the mean bitrate.

Each frame holds the data from 2 granules. The audio data in a frame is allocated in the following way:

- main_data_begin pointer
- side info for both granules (scfsi)
- side info granule 1
- side info granule 2

The header and this part of the audio data constitute the side information stream.

- scalefactors and Huffman code data granule 1
- scalefactors and Huffman code data granule 2
- ancillary data

These data constitute the main data stream. The main_data_begin pointer specifies a negative offset from the position of the first byte of the header.

2.4.3.4.1 Decoding

The first action is the synchronization of the decoder to the incoming bitstream. This is done as in the other layers. The header information (first 32 bits including syncword) is read in just as in the other layers. The information about sampling frequency is used to select the scalefactor_band table (see annex B.8).

2.4.3.4.2 Side information

The side information must be extracted from the bitstream and stored for use during the decoding of the associated frame. The table select information is used to select the Huffman decoder table and the number of ESC-bits (linbits), according to table B.7.

2.4.3.4.3 Start of main_data

The main_data (scalefactors, Huffman coded data and ancillary information) are not necessarily located adjacent to the side information. This is described in figure A.7.a and figure A.7.b. The beginning of the main data part is located by using the main_data_begin pointer of the current frame. The allocation of the main data is done in a way that all main data are resident in the input buffer when the header of the next frame is arriving in the input buffer. The decoder has to skip Header and side information when decoding the main data. It knows their positions from the bitrate_index and padding_bit. The length of the Header is always 4 bytes, the length of the side information is 17 bytes in mode single_channel and 32 bytes in the other modes. Main data can span more than one block of header and side information (see figure A.7.b).

2.4.3.4.4 Buffer considerations

The following rule can be used to calculate the maximum number of bits used for one granule:

The buffer length is 7 680 bits. This value is used as the maximum buffer at every bitrate. At the highest possible bitrate of Layer III (320 kbits/s per stereo signal) and sampling frequency 48 kHz the mean frame length is (320 000/48 000)*1 152 = 7 680 bits. Therefore the frames must be of constant length at this bitrate and sampling frequency. At 64 kbits/s (128 kbits/s stereo) the mean granule length is

(64 000/48 000)*576 = 768 bit at 48 kHz sampling frequency. This means that there is a maximum deviation (short time buffer) of 7 680 - 4*768 = 4 608 bits is allowed at 64 kbits/s. The actual deviation is equal to the number of bytes denoted by the main_data_begin offset pointer. The actual maximum deviation is 2**9 * 8 bit = 4 096 bits. For intermediate bitrates the delay and buffer length can be calculated accordingly. The exchange of buffer between the left and right channel in a stereo bitstream is allowed without restrictions. Because of the constraint on the buffer size main_data_begin is always set to 0 in the case of bitrate_index==14, i.e. data rate 320 kbits/s per stereo signal. In this case all data are allocated between adjacent header words.

At sampling frequencies lower than 48 kHz the buffer should be constrained such that the same physical buffer size is sufficient as the one calculated for the 48 kHz case above.

2.4.3.4.5 Scalefactors

The scalefactors are decoded according to the slen1 and slen2 which themselves are determined from the values of scalefac_compress. The decoded values can be used as entries into a table or used to calculate the factors for each scalefactor band directly. When decoding the second granule, the scfsi has to be considered. For the bands in which the corresponding scfsi is set to 1, the scalefactors of the first granule are also used for the second granule, therefore they are not transmitted for the second granule.

The number of bits used to encode scalefactors is called part2_length, and is calculated as follows.

For block_type == 0, 1, or 3 (long blocks):

$$\text{part2\_length} = 11 * \text{slen1} + 10 * \text{slen2}.$$

For block_type==2 (short blocks) and mixed_block_flag == 0:

$$\text{part2\_length} = 18 * \text{slen1} + 18 * \text{slen2}.$$

For block_type==2 (short blocks) and mixed_block_flag == 1:

$$\text{part2\_length} = 17 * \text{slen1} + 18 * \text{slen2}.$$

These formulas are valid if gr==0 or if gr==1 and scfsi[ch][scfsi_band]==0 for all scfsi_bands, i.e. scalefactor selection information is not used.

2.4.3.4.6 Huffman decoding

All necessary information including the table which realizes the Huffman code tree can be generated from the tables in table B.7. First the big_values data are decoded, using the tables with the number table_select[gr][ch][region]. The frequency lines in region 0, region 1 and region 2 are Huffman decoded in pairs until big_values number of line-pairs have been decoded. The remaining Huffmancodebits are decoded using the table according to count1table_select[gr][ch]. Decoding is done until all Huffman code bits have been decoded or until quantized values representing 576 frequency lines have been decoded, whichever comes first. If there are more Huffman code bits than necessary to decode 576 values they are regarded as stuffing bits and discarded. The variable count1 is implicitely derived as the number of quadruples of decoded values using count1table_select.

2.4.3.4.7 Requantizer

The nonuniform quantizer uses a power law. For each output value, "is", from the Huffman decoder, "$|is|^{4/3}$" is calculated. This can be done either by table lookup or by explicit calculation.

2.4.3.4.7.1 Formula for requantization and all scaling

One complete formula describes all the processing from the Huffman decoded values to the input of the synthesis filterbank. All necessary scaling factors are contained within this formula. The output data are reconstructed from requantized samples. Global gain and subblock gain values affect all values within one time window (in the case of block_type==2). Scalefactors and preflag further adjust the gain within each scalefactor band. An illustration can be found in figure A.8.

The following is the requantization equation for short windows. The Huffman decoded value at buffer index i is called is_i, the input to the synthesis filterbank at index i is called xr_i:

$$xr_i = \mathrm{sign}(is_i) * |is_i|^{\frac{4}{3}} * 2^{\frac{1}{4}(\mathrm{global\_gain[gr]} - 210 - 8*\mathrm{subblock\_gain[window][gr]})}$$
$$* 2^{-(\mathrm{scalefac\_multiplier} * \mathrm{scalefac\_s[gr][ch][sfb][window]})}$$

For long blocks, the formula is:

$$xr_i = \mathrm{sign}(is_i) * |is_i|^{\frac{4}{3}} * 2^{\frac{1}{4}(\mathrm{global\_gain[gr]} - 210)}$$
$$* 2^{-(\mathrm{scalefac\_multiplier} * (\mathrm{scalefac\_l[sfb][ch][gr]} + \mathrm{preflag[gr]} * \mathrm{pretab[sfb]}))}$$

Pretab[sfb] is a value given in the preemphasis table B.6. The constant 210 in this formula is needed to scale the output appropriately. It is a system constant. The synthesis filterbank is assumed to be implemented according to the formulas below. The range of the output values of the decoder (PCM samples) is between - 1,0 and + 1,0.

2.4.3.4.8 Reordering

If short blocks are used (block_type==2), the rescaled data xr[scf_band][window][freq_line] (as described in huffmancodebits() in 2.4.1.7) shall be reordered in subband order, xr[subband[window][freq_line], prior to the IMDCT operation.

2.4.3.4.9 Stereo Processing

After requantization, the reconstructed values are processed for MS or intensity stereo modes or both, before going to the synthesis filterbank. In MS_stereo mode, both channels of a granule must have the same block_type.

2.4.3.4.9.1 MS_stereo mode

This mode switch (found in the header: mode_extension) allows switching from "independent stereo" to MS_stereo. If MS_stereo is enabled but intensity stereo is not enabled the entire spectrum is decoded in MS_stereo. If both MS_Stereo and intensity stereo are enabled, the upper bound of the scalefactor bands decoded in MS_stereo is derived from the "zero_part" of the difference (right) channel. In this case the scalefactor band in which the last non-zero (right channel) frequency line occurs is the last scalefactor band to which the MS_stereo equations apply. Above this bound intensity stereo may be applied if enabled in the header. The "zero_part" of the difference channel is the part of the spectrum from "bigvalues * 2 + count1 * 4" (see 2.4.2.7) to the Nyquist rate.

2.4.3.4.9.2 MS matrix

In MS stereo mode the values of the normalized middle/side channels M_i/S_i are transmitted instead of the left/right channel values L_i/R_i. Thus L_i/R_i are reconstructed using

$$L_i = \frac{M_i + S_i}{\sqrt{2}} \text{ and } R_i = \frac{M_i - S_i}{\sqrt{2}}.$$

The values M_i are transmitted in the left, values S_i are transmitted in the right channel.

If window switching occurs, then the M and S channels must switch synchronously.

2.4.3.4.9.3 Intensity stereo mode

This mode switch (found in the header: mode_extension) allows switching from 'normal stereo' to intensity stereo. In Layer III, intenisty stereo is not done using a pair of scalefactor s as in Layers I and II, but by specifying the magnitude (via the scalefactors of the right channel as normal) and a stereo position is_pos_{sb}[sfb]. is_pos_{sb}[sfb] is transmitted instead of scalefactors for the right channels. The stereo position is used to derive the left and right channel signals according to the formulas below. The lower bound of the scalefactor bands decoded in intensity stereo is derived from the "zero_part" of the right channel. Above this bound decoding of intensity stereo is applied using the scalefactors of the right channel as intensity stereo positions. An intensity stereo position of 7 in one scalefactor band indicates that this scalefactor band is not decoded as intensity stereo.

```
Scalefactor bands :
|     |     |     |     |     |     |     |     |     |     | . . .    |    |
|<--- nonzero_part of spectrum (right chan) --->|<----- zero_part of spectrum ------>|
|<------- m/s or l/r stereo coded part ---------->|<- intensity stereo coded part ->|
```

For each scalefactor band (sb) coded in intensity stereo, the following steps are executed:

1) the intensity stereo position is_pos_{sb} is read from the scalefactor of the right channel.

2) if (is_pos_{sb} == 7) do not perform the following steps (illegal is_pos).

3) $is\_ratio = \tan\left(is\_pos_{sb} * \frac{\pi}{12}\right)$.

4) $L_i := L_i * \frac{is\_ratio}{1 + is\_ratio}$ for all indices i within the actual scalefactor band sb.

5) $R_i := L_i * \frac{1}{1 + is\_ratio}$ for all indices i within the actual scalefactor band sb.

2.4.3.4.10 Synthesis filterbank

Figure A.4. shows a block diagram including the synthesis filterbank. The frequency lines are preprocessed by the "alias reduction" scheme (see the block diagrams in in figure A.5 and in table B.9. for the coefficients) and fed into the IMDCT matrix, each 18 into one transform block. The first half of the output values are added to the stored overlap values from the last block. These values are new output values and are input values for the polyphase filterbank. The second half of the output values is stored for overlap with the next data granule. For every second subband of the polyphase filterbank every second input value is multiplied by -1 to correct for the frequency inversion of the polyphase filterbank.

2.4.3.4.10.1 Alias reduction

For long block_type granules (block_type != 2) the input to the synthesis filterbank is processed for alias reduction before processing by the IMDCT. The following pseudo code describes the alias reduction computation:

```
for (sb=1; sb<32; sb++)
    for (i=0; i<8; i++) {
        xar[18*sb-1-i] = xr[18*sb-1-i]Cs[i] -  xr[18*sb+i]Ca[i]
        xar[18*sb+i] = xr[18*sb+i]Cs[i] + xr[18*sb-1-i]Ca[i]
    }
```

The indices of arrays xar[] and xr[] label the frequency lines in a granule, arranged in order of lowest frequency to highest frequency, with zero being the index of the lowest frequency line, and 575 being the index of the highest. The coefficients: Cs[i] and Ca[i] can be found in table B.9. Figures A.5 and A.6 illustrate the alias reduction computation.

Alias reduction is not applied for granules with block_type == 2 (short block).

2.4.3.4.10.2 IMDCT

In the following, n is the number of windowed samples (for short blocks n is 12, for long blocks n is 36). In the case of a block of type "short", each of the three short blocks is transformed separately. n/2 values X_k are transformed to n values x_i. The analytical expression of the IMDCT is:

$$x_i = \sum_{k=0}^{\frac{n}{2}-1} X_k \cos\left(\frac{\pi}{2n}\left(2i+1+\frac{n}{2}\right)(2k+1)\right) \quad \text{for i = 0 to n - 1}$$

2.4.3.4.10.3 Windowing

Depending on the block_type different shapes of windows are used.

a) *block_type=0 (normal window)*

$$z_i = x_i \sin\left(\frac{\pi}{36}\left(i+\frac{1}{2}\right)\right) \quad \text{for i = 0 to 35}$$

b) *block_type=1 (start block)*

$$z_i = \begin{cases} x_i \sin\left(\frac{\pi}{36}\left(i+\frac{1}{2}\right)\right) & \text{for i = 0 to 17} \\ x_i & \text{for i = 18 to 23} \\ x_i \sin\left(\frac{\pi}{12}\left(i-18+\frac{1}{2}\right)\right) & \text{for i = 24 to 29} \\ 0 & \text{for i = 30 to 35} \end{cases}$$

c) *block_type=3 (stop block)*

$$z_i = \begin{cases} 0 & \text{for i = 0 to 5} \\ x_i \sin\left(\frac{\pi}{12}\left(i-6+\frac{1}{2}\right)\right) & \text{for i = 6 to 11} \\ x_i & \text{for i = 12 to 17} \\ x_i \sin\left(\frac{\pi}{36}\left(i+\frac{1}{2}\right)\right) & \text{for i = 18 to 35} \end{cases}$$

d) *block_type=2 (short block)*

Each of the three short blocks is windowed separately.

$$y_i^{(j)} = x_i^{(j)} \sin\left(\frac{\pi}{12}\left(i+\frac{1}{2}\right)\right) \quad \text{for i = 0 to 11, j = 0 to 2}$$

The windowed short blocks must be overlapped and concatenated.

$$z_i = \begin{cases} 0 & \text{for i = 0 to 5} \\ y_{i-6}^{(1)} & \text{for i = 6 to 11} \\ y_{i-6}^{(1)} + y_{i-12}^{(2)} & \text{for i = 12 to 17} \\ y_{i-12}^{(2)} + y_{i-18}^{(3)} & \text{for i = 18 to 23} \\ y_{i-18}^{(3)} & \text{for i = 24 to 29} \\ 0 & \text{for i = 30 to 35} \end{cases}$$

2.4.3.4.10.4 Overlapping and adding with previous block

The first half of the block of 36 values is overlapped with the second half of the previous block. The second half of the actual block is stored to be used in the next block:

$$\text{result}_i = z_i + s_i \quad \text{for i = 0 to 17}$$
$$s_i = z_{i+18} \quad \text{for i = 0 to 17}$$

2.4.3.4.10.5 Compensation for frequency inversion of polyphase filterbank

The output of the overlap add consists of 18 time samples for each of the 32 polyphase subbands. If the time samples are labeled 0 through 17, with 0 being the earliest time sample, and subbands are labeled 0 through 31, with 0 being the lowest subband, then every odd time sample of every odd subband is multiplied by -1 before processing by the polyphase filter bank.

Annex A
(normative)

Diagrams

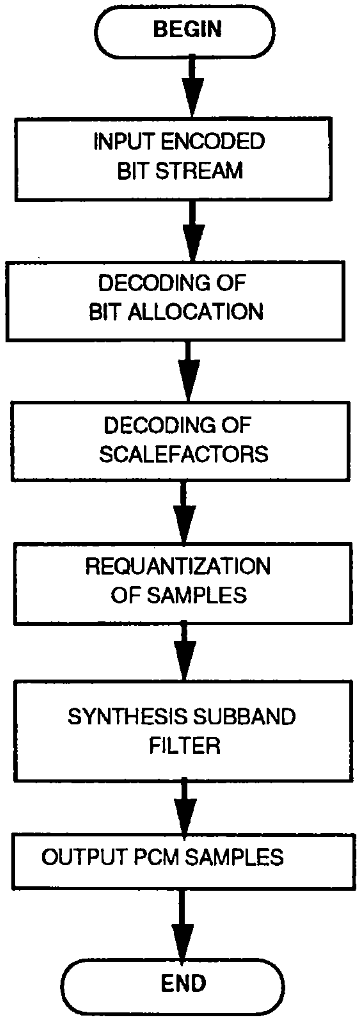


Figure A.1 -- Layer I and II decoder flow chart

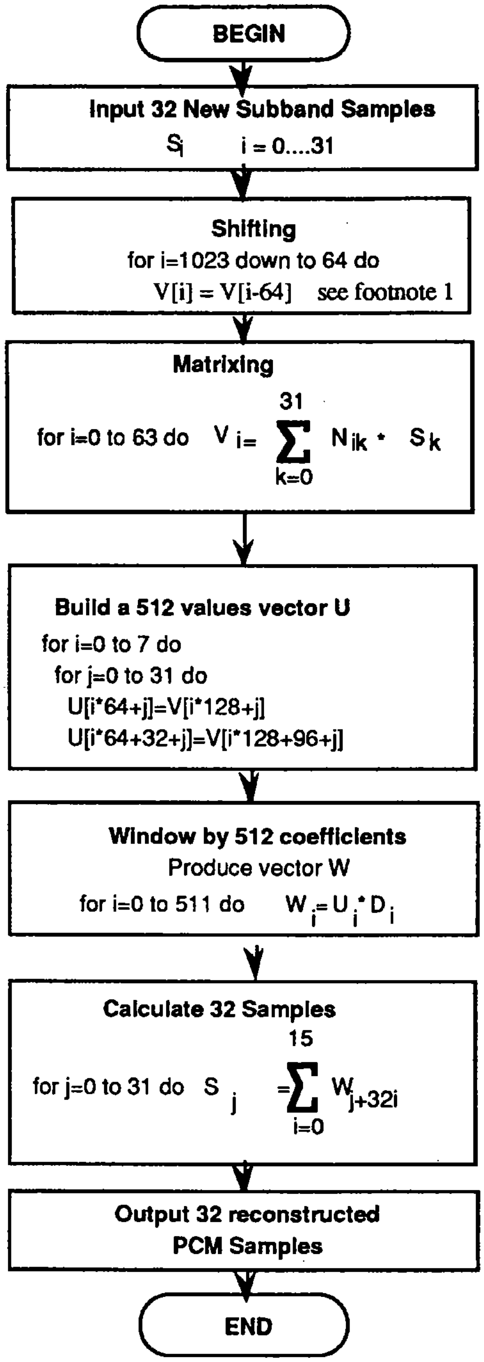


Figure A.2 -- Synthesis subband filter flow chart

---

1 V to be initialized with zeroes during startup.

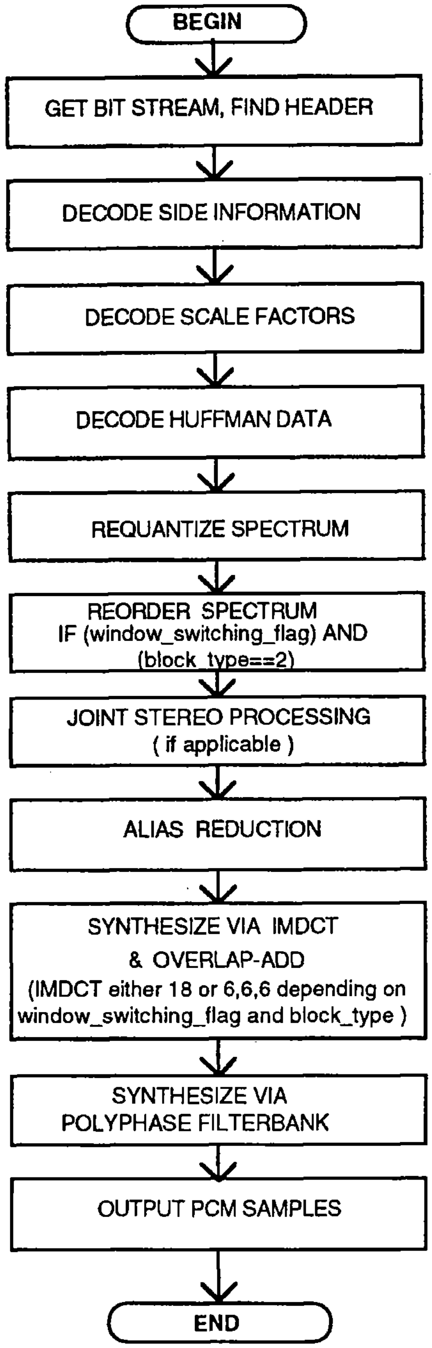


Figure A.3 -- Layer III decoder flow chart

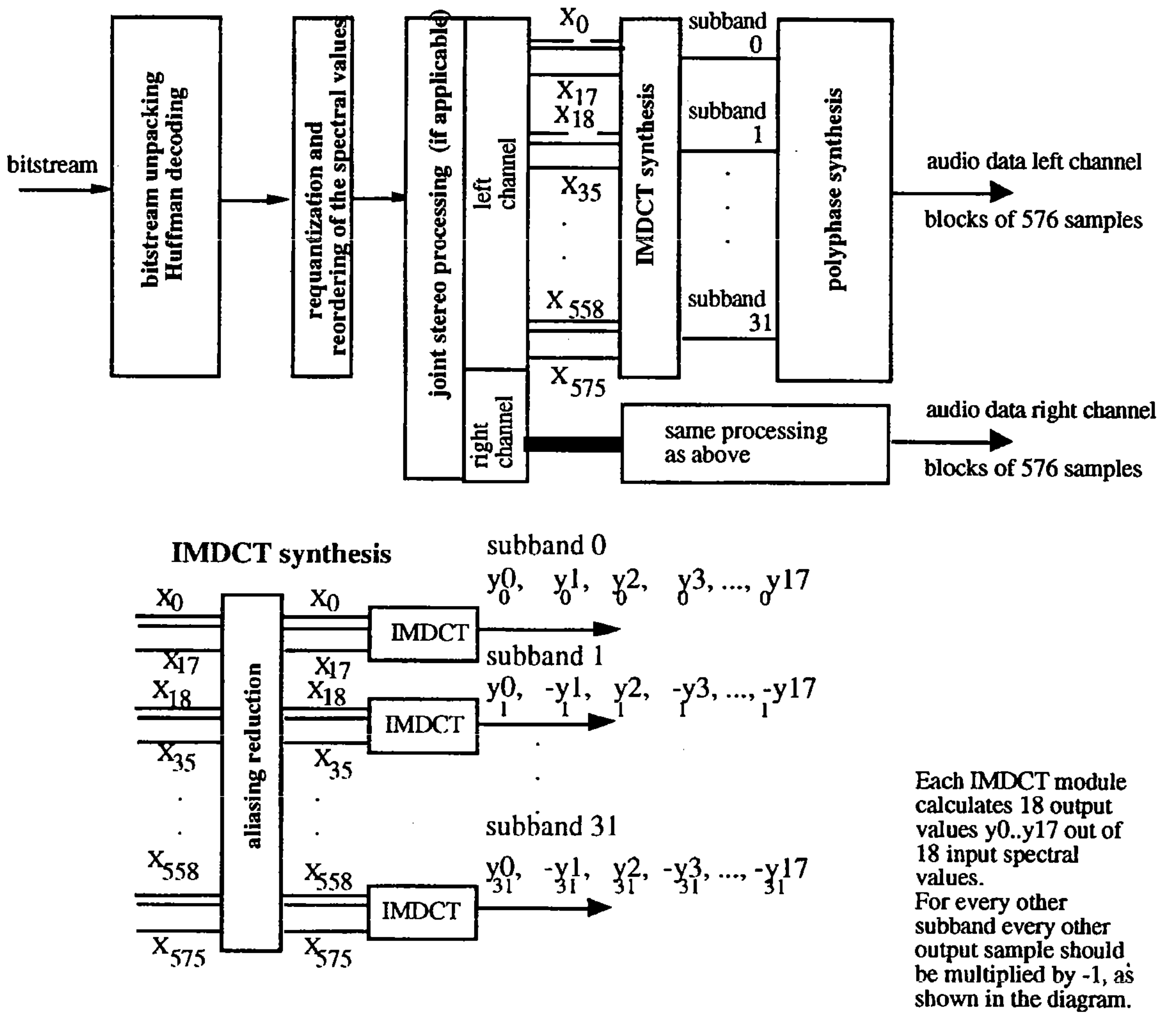


Figure A.4 -- Layer III decoder diagram

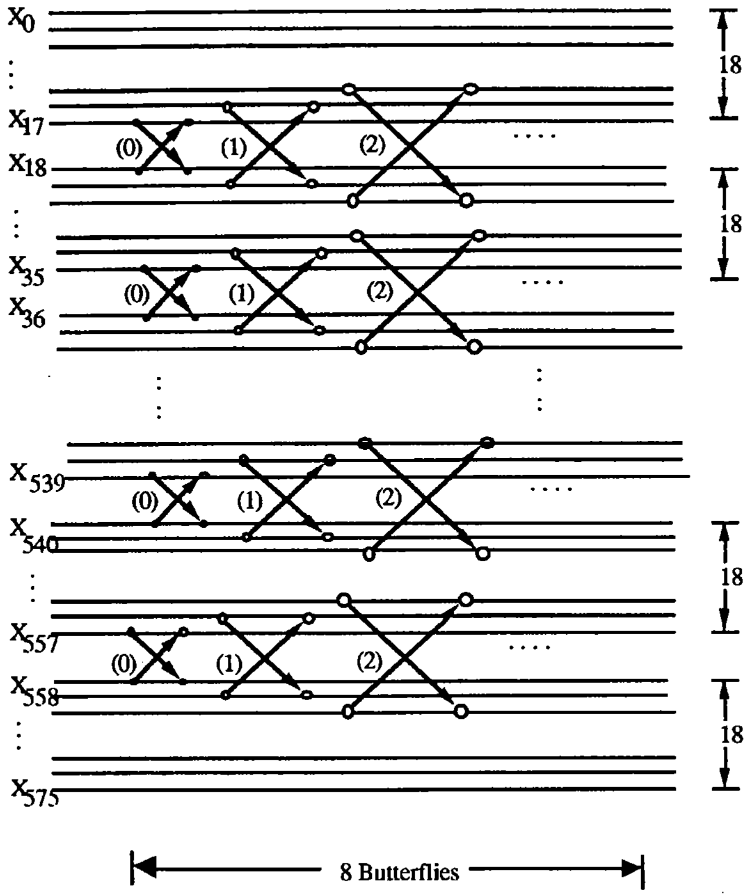


Figure A.5 -- Layer III aliasing reduction decoder diagram

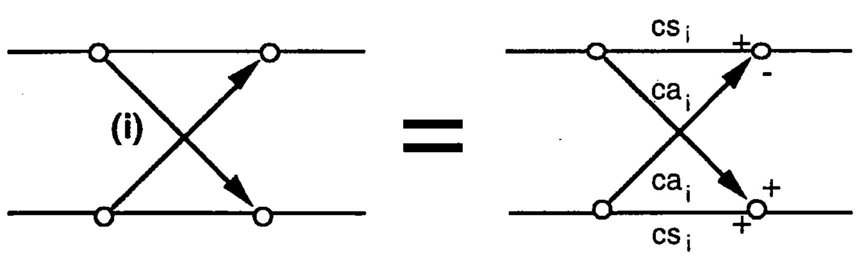


Figure A.6 -- Layer III aliasing-butterfly, decoder

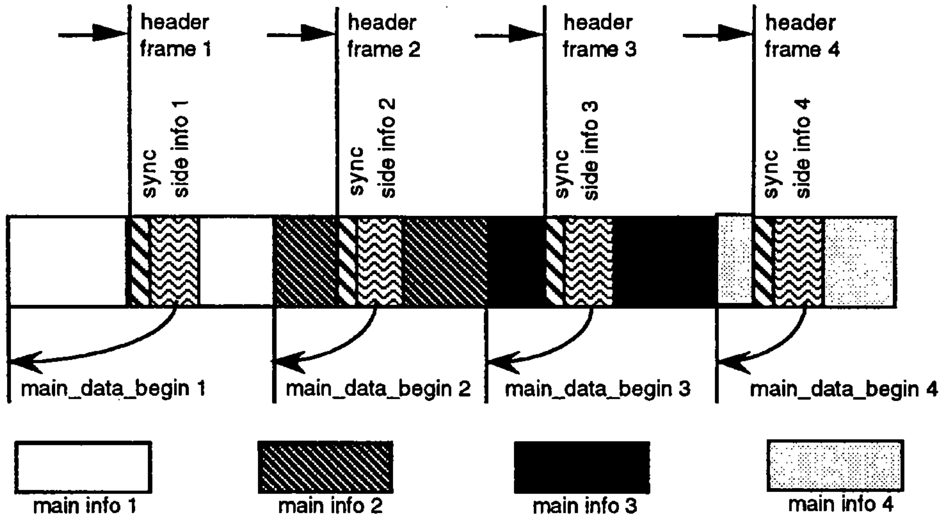
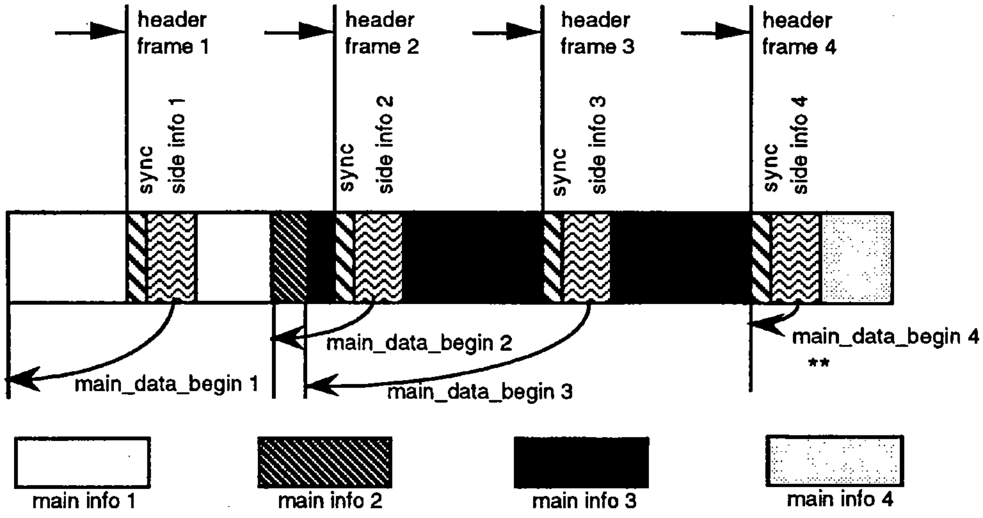


Figure A.7.a -- Layer III bitstream organization



**) main_data_begin 4 == 0 : This signifies that main data starts directly after the side information for frame 4. This is the lower limit for main_data_begin; main_data cannot start later than this point. Note that data bytes used by "sync" and "side info" are not counted by the main_data_begin pointer.

Note: 'info' means information

Figure A.7.b -- Layer III bitstream organization with peak demand at main info 3 and small demand at main info 2.

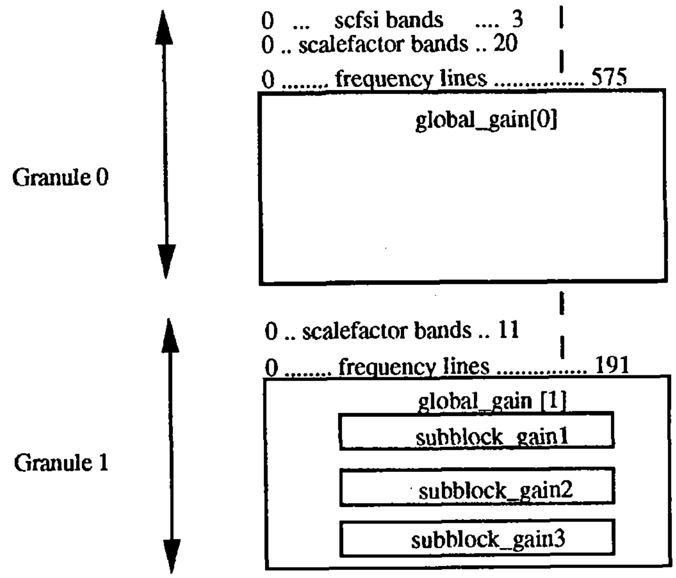


Figure A.8 -- Layer III illustration of granules for frame with block_type == 0 in first granule and block_type == 2 in second granule.

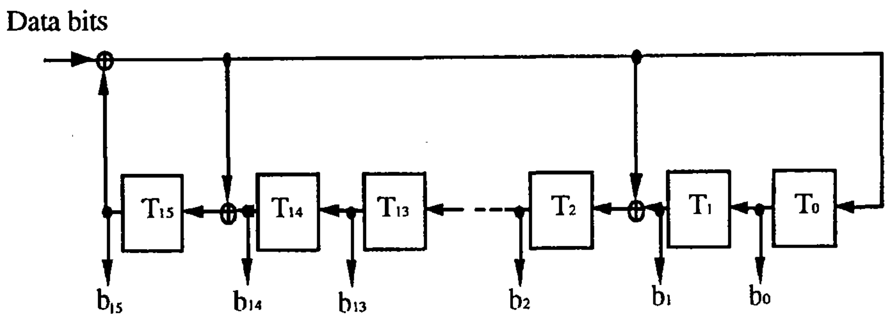


Figure A.9 -- CRC-check diagram

Annex B

(normative)

Tables

Table B.1 -- Layer I,II scalefactors

index	scalefactor	index	scalefactor
0	2,00000000000000	32	0,00123039165029
1	1,58740105196820	33	0,00097656250000
2	1,25992104989487	34	0,00077509816991
3	1,00000000000000	35	0,00061519582514
4	0,79370052598410	36	0,00048828125000
5	0,62996052494744	37	0,00038754908495
6	0,50000000000000	38	0,00030759791257
7	0,39685026299205	39	0,00024414062500
8	0,31498026247372	40	0,00019377454248
9	0,25000000000000	41	0,00015379895629
10	0,19842513149602	42	0,00012207031250
11	0,15749013123686	43	0,00009688727124
12	0,12500000000000	44	0,00007689947814
13	0,09921256574801	45	0,00006103515625
14	0,07874506561843	46	0,00004844363562
15	0,06250000000000	47	0,00003844973907
16	0,04960628287401	48	0,00003051757813
17	0,03937253280921	49	0,00002422181781
18	0,03125000000000	50	0,00001922486954
19	0,02480314143700	51	0,00001525878906
20	0,01968626640461	52	0,00001211090890
21	0,01562500000000	53	0,00000961243477
22	0,01240157071850	54	0,00000762939453
23	0,00984313320230	55	0,00000605545445
24	0,00781250000000	56	0,00000480621738
25	0,00620078535925	57	0,00000381469727
26	0,00492156660115	58	0,00000302772723
27	0,00390625000000	59	0,00000240310869
28	0,00310039267963	60	0,00000190734863
29	0,00246078330058	61	0,00000151386361
30	0,00195312500000	62	0,00000120155435
31	0,00155019633981		

Table B.2 -- Layer II bit allocation tables

Table B.2a -- Possible quantization per subband

Fs = 48 kHz Bit rates per channel = 56, 64, 80, 96, 112, 128, 160, 192 kbits/s, and free format.
Fs = 44,1 kHz Bit rates per channel = 56, 64, 80 kbits/s.
Fs = 32 kHz Bit rates per channel = 56, 64, 80 kbits/s.

		index															
sb	nbal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
1	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
2	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
3	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
4	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
5	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
6	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
7	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
8	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
9	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
10	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
11	3	-	3	5	7	9	15	31	65535								
12	3	-	3	5	7	9	15	31	65535								
13	3	-	3	5	7	9	15	31	65535								
14	3	-	3	5	7	9	15	31	65535								
15	3	-	3	5	7	9	15	31	65535								
16	3	-	3	5	7	9	15	31	65535								
17	3	-	3	5	7	9	15	31	65535								
18	3	-	3	5	7	9	15	31	65535								
19	3	-	3	5	7	9	15	31	65535								
20	3	-	3	5	7	9	15	31	65535								
21	3	-	3	5	7	9	15	31	65535								
22	3	-	3	5	7	9	15	31	65535								
23	2	-	3	5	65535												
24	2	-	3	5	65535												
25	2	-	3	5	65535												
26	2	-	3	5	65535												
27	0	-															
28	0	-															
29	0	-															
30	0	-															
31	0	-															

sblimit = 27
Sum of nbal = 88

Table B.2b -- Possible quantization per subband

Fs = 48 kHz ------------- not relevant -------------
Fs = 44,1 kHz Bitrates per channel = 96, 112, 128, 160, 192 kbits/s and free format
Fs = 32 kHz Bitrates per channel = 96, 112, 128, 160, 192 kbits/s and free format

index

sb	nbal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
1	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
2	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
3	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
4	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
5	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
6	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
7	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
8	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
9	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
10	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
11	3	-	3	5	7	9	15	31	65535								
12	3	-	3	5	7	9	15	31	65535								
13	3	-	3	5	7	9	15	31	65535								
14	3	-	3	5	7	9	15	31	65535								
15	3	-	3	5	7	9	15	31	65535								
16	3	-	3	5	7	9	15	31	65535								
17	3	-	3	5	7	9	15	31	65535								
18	3	-	3	5	7	9	15	31	65535								
19	3	-	3	5	7	9	15	31	65535								
20	3	-	3	5	7	9	15	31	65535								
21	3	-	3	5	7	9	15	31	65535								
22	3	-	3	5	7	9	15	31	65535								
23	2	-	3	5	65535												
24	2	-	3	5	65535												
25	2	-	3	5	65535												
26	2	-	3	5	65535												
27	2	-	3	5	65535												
28	2	-	3	5	65535												
29	2	-	3	5	65535												
30	0	-															
31	0	-															

sblimit = 30
Sum of nbal = 94

Table B.2c -- Possible quantization per subband

Fs = 48 kHz Bitrates per channel = 32, 48 kbits/s
Fs = 44,1 kHz Bitrates per channel = 32, 48 kbits/s
Fs = 32 kHz -------- not relevant --------

		index															
sb	nbal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	-	3	5	9	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767
1	4	-	3	5	9	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767
2	3	-	3	5	9	15	31	63	127								
3	3	-	3	5	9	15	31	63	127								
4	3	-	3	5	9	15	31	63	127								
5	3	-	3	5	9	15	31	63	127								
6	3	-	3	5	9	15	31	63	127								
7	3	-	3	5	9	15	31	63	127								
8	0	-															
9	0	-															
10	0	-															
11	0	-															
12	0	-															
13	0	-															
14	0	-															
15	0	-															
16	0	-															
17	0	-															
18	0	-															
19	0	-															
20	0	-															
21	0	-															
22	0	-															
23	0	-															
24	0	-															
25	0	-															
26	0	-															
27	0	-															
28	0	-															
29	0	-															
30	0	-															
31	0	-															

sblimit = 8
Sum of nbal = 26

Table B.2d. -- Possible quantization per subband

Fs = 48 kHz ------- not relevant -------
Fs = 44,1kHz ------- not relevant -------
Fs = 32 kHz Bitrates per channel = 32, 48 kbits/s.

		index															
sb	nbal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	-	3	5	9	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767
1	4	-	3	5	9	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767
2	3	-	3	5	9	15	31	63	127								
3	3	-	3	5	9	15	31	63	127								
4	3	-	3	5	9	15	31	63	127								
5	3	-	3	5	9	15	31	63	127								
6	3	-	3	5	9	15	31	63	127								
7	3	-	3	5	9	15	31	63	127								
8	3	-	3	5	9	15	31	63	127								
9	3	-	3	5	9	15	31	63	127								
10	3	-	3	5	9	15	31	63	127								
11	3	-	3	5	9	15	31	63	127								
12	0	-															
13	0	-															
14	0	-															
15	0	-															
16	0	-															
17	0	-															
18	0	-															
19	0	-															
20	0	-															
21	0	-															
22	0	-															
23	0	-															
24	0	-															
25	0	-															
26	0	-															
27	0	-															
28	0	-															
29	0	-															
30	0	-															
31	0	-															

sblimit = 12
Sum of nbal = 38

Table B.3 -- Coefficients D_i of the synthesis window

D[0]= 0,000000000	D[1]=-0,000015259	D[2]=-0,000015259	D[3]=-0,000015259
D[4]=-0,000015259	D[5]=-0,000015259	D[6]=-0,000015259	D[7]=-0,000030518
D[8]=-0,000030518	D[9]=-0,000030518	D[10]=-0,000030518	D[11]=-0,000045776
D[12]=-0,000045776	D[13]=-0,000061035	D[14]=-0,000061035	D[15]=-0,000076294
D[16]=-0,000076294	D[17]=-0,000091553	D[18]=-0,000106812	D[19]=-0,000106812
D[20]=-0,000122070	D[21]=-0,000137329	D[22]=-0,000152588	D[23]=-0,000167847
D[24]=-0,000198364	D[25]=-0,000213623	D[26]=-0,000244141	D[27]=-0,000259399
D[28]=-0,000289917	D[29]=-0,000320435	D[30]=-0,000366211	D[31]=-0,000396729
D[32]=-0,000442505	D[33]=-0,000473022	D[34]=-0,000534058	D[35]=-0,000579834
D[36]=-0,000625610	D[37]=-0,000686646	D[38]=-0,000747681	D[39]=-0,000808716
D[40]=-0,000885010	D[41]=-0,000961304	D[42]=-0,001037598	D[43]=-0,001113892
D[44]=-0,001205444	D[45]=-0,001296997	D[46]=-0,001388550	D[47]=-0,001480103
D[48]=-0,001586914	D[49]=-0,001693726	D[50]=-0,001785278	D[51]=-0,001907349
D[52]=-0,002014160	D[53]=-0,002120972	D[54]=-0,002243042	D[55]=-0,002349854
D[56]=-0,002456665	D[57]=-0,002578735	D[58]=-0,002685547	D[59]=-0,002792358
D[60]=-0,002899170	D[61]=-0,002990723	D[62]=-0,003082275	D[63]=-0,003173828
D[64]= 0,003250122	D[65]= 0,003326416	D[66]= 0,003387451	D[67]= 0,003433228
D[68]= 0,003463745	D[69]= 0,003479004	D[70]= 0,003479004	D[71]= 0,003463745
D[72]= 0,003417969	D[73]= 0,003372192	D[74]= 0,003280640	D[75]= 0,003173828
D[76]= 0,003051758	D[77]= 0,002883911	D[78]= 0,002700806	D[79]= 0,002487183
D[80]= 0,002227783	D[81]= 0,001937866	D[82]= 0,001617432	D[83]= 0,001266479
D[84]= 0,000869751	D[85]= 0,000442505	D[86]=-0,000030518	D[87]=-0,000549316
D[88]=-0,001098633	D[89]=-0,001693726	D[90]=-0,002334595	D[91]=-0,003005981
D[92]=-0,003723145	D[93]=-0,004486084	D[94]=-0,005294800	D[95]=-0,006118774
D[96]=-0,007003784	D[97]=-0,007919312	D[98]=-0,008865356	D[99]=-0,009841919
D[100]=-0,010848999	D[101]=-0,011886597	D[102]=-0,012939453	D[103]=-0,014022827
D[104]=-0,015121460	D[105]=-0,016235352	D[106]=-0,017349243	D[107]=-0,018463135
D[108]=-0,019577026	D[109]=-0,020690918	D[110]=-0,021789551	D[111]=-0,022857666
D[112]=-0,023910522	D[113]=-0,024932861	D[114]=-0,025909424	D[115]=-0,026840210
D[116]=-0,027725220	D[117]=-0,028533936	D[118]=-0,029281616	D[119]=-0,029937744
D[120]=-0,030532837	D[121]=-0,031005859	D[122]=-0,031387329	D[123]=-0,031661987
D[124]=-0,031814575	D[125]=-0,031845093	D[126]=-0,031738281	D[127]=-0,031478882
D[128]= 0,031082153	D[129]= 0,030517578	D[130]= 0,029785156	D[131]= 0,028884888
D[132]= 0,027801514	D[133]= 0,026535034	D[134]= 0,025085449	D[135]= 0,023422241
D[136]= 0,021575928	D[137]= 0,019531250	D[138]= 0,017257690	D[139]= 0,014801025
D[140]= 0,012115479	D[141]= 0,009231567	D[142]= 0,006134033	D[143]= 0,002822876
D[144]=-0,000686646	D[145]=-0,004394531	D[146]=-0,008316040	D[147]=-0,012420654
D[148]=-0,016708374	D[149]=-0,021179199	D[150]=-0,025817871	D[151]=-0,030609131
D[152]=-0,035552979	D[153]=-0,040634155	D[154]=-0,045837402	D[155]=-0,051132202
D[156]=-0,056533813	D[157]=-0,061996460	D[158]=-0,067520142	D[159]=-0,073059082
D[160]=-0,078628540	D[161]=-0,084182739	D[162]=-0,089706421	D[163]=-0,095169067
D[164]=-0,100540161	D[165]=-0,105819702	D[166]=-0,110946655	D[167]=-0,115921021
D[168]=-0,120697021	D[169]=-0,125259399	D[170]=-0,129562378	D[171]=-0,133590698
D[172]=-0,137298584	D[173]=-0,140670776	D[174]=-0,143676758	D[175]=-0,146255493
D[176]=-0,148422241	D[177]=-0,150115967	D[178]=-0,151306152	D[179]=-0,151962280
D[180]=-0,152069092	D[181]=-0,151596069	D[182]=-0,150497437	D[183]=-0,148773193
D[184]=-0,146362305	D[185]=-0,143264771	D[186]=-0,139450073	D[187]=-0,134887695
D[188]=-0,129577637	D[189]=-0,123474121	D[190]=-0,116577148	D[191]=-0,108856201
D[192]= 0,100311279	D[193]= 0,090927124	D[194]= 0,080688477	D[195]= 0,069595337
D[196]= 0,057617187	D[197]= 0,044784546	D[198]= 0,031082153	D[199]= 0,016510010
D[200]= 0,001068115	D[201]=-0,015228271	D[202]=-0,032379150	D[203]=-0,050354004
D[204]=-0,069168091	D[205]=-0,088775635	D[206]=-0,109161377	D[207]=-0,130310059
D[208]=-0,152206421	D[209]=-0,174789429	D[210]=-0,198059082	D[211]=-0,221984863
D[212]=-0,246505737	D[213]=-0,271591187	D[214]=-0,297210693	D[215]=-0,323318481
D[216]=-0,349868774	D[217]=-0,376800537	D[218]=-0,404083252	D[219]=-0,431655884
D[220]=-0,459472656	D[221]=-0,487472534	D[222]=-0,515609741	D[223]=-0,543823242

D[224]=-0,572036743	D[225]=-0,600219727	D[226]=-0,628295898	D[227]=-0,656219482
D[228]=-0,683914185	D[229]=-0,711318970	D[230]=-0,738372803	D[231]=-0,765029907
D[232]=-0,791213989	D[233]=-0,816864014	D[234]=-0,841949463	D[235]=-0,866363525
D[236]=-0,890090942	D[237]=-0,913055420	D[238]=-0,935195923	D[239]=-0,956481934
D[240]=-0,976852417	D[241]=-0,996246338	D[242]=-1,014617920	D[243]=-1,031936646
D[244]=-1,048156738	D[245]=-1,063217163	D[246]=-1,077117920	D[247]=-1,089782715
D[248]=-1,101211548	D[249]=-1,111373901	D[250]=-1,120223999	D[251]=-1,127746582
D[252]=-1,133926392	D[253]=-1,138763428	D[254]=-1,142211914	D[255]=-1,144287109
D[256]= 1,144989014	D[257]= 1,144287109	D[258]= 1,142211914	D[259]= 1,138763428
D[260]= 1,133926392	D[261]= 1,127746582	D[262]= 1,120223999	D[263]= 1,111373901
D[264]= 1,101211548	D[265]= 1,089782715	D[266]= 1,077117920	D[267]= 1,063217163
D[268]= 1,048156738	D[269]= 1,031936646	D[270]= 1,014617920	D[271]= 0,996246338
D[272]= 0,976852417	D[273]= 0,956481934	D[274]= 0,935195923	D[275]= 0,913055420
D[276]= 0,890090942	D[277]= 0,866363525	D[278]= 0,841949463	D[279]= 0,816864014
D[280]= 0,791213989	D[281]= 0,765029907	D[282]= 0,738372803	D[283]= 0,711318970
D[284]= 0,683914185	D[285]= 0,656219482	D[286]= 0,628295898	D[287]= 0,600219727
D[288]= 0,572036743	D[289]= 0,543823242	D[290]= 0,515609741	D[291]= 0,487472534
D[292]= 0,459472656	D[293]= 0,431655884	D[294]= 0,404083252	D[295]= 0,376800537
D[296]= 0,349868774	D[297]= 0,323318481	D[298]= 0,297210693	D[299]= 0,271591187
D[300]= 0,246505737	D[301]= 0,221984863	D[302]= 0,198059082	D[303]= 0,174789429
D[304]= 0,152206421	D[305]= 0,130310059	D[306]= 0,109161377	D[307]= 0,088775635
D[308]= 0,069168091	D[309]= 0,050354004	D[310]= 0,032379150	D[311]= 0,015228271
D[312]=-0,001068115	D[313]=-0,016510010	D[314]=-0,031082153	D[315]=-0,044784546
D[316]=-0,057617187	D[317]=-0,069595337	D[318]=-0,080688477	D[319]=-0,090927124
D[320]= 0,100311279	D[321]= 0,108856201	D[322]= 0,116577148	D[323]= 0,123474121
D[324]= 0,129577637	D[325]= 0,134887695	D[326]= 0,139450073	D[327]= 0,143264771
D[328]= 0,146362305	D[329]= 0,148773193	D[330]= 0,150497437	D[331]= 0,151596069
D[332]= 0,152069092	D[333]= 0,151962280	D[334]= 0,151306152	D[335]= 0,150115967
D[336]= 0,148422241	D[337]= 0,146255493	D[338]= 0,143676758	D[339]= 0,140670776
D[340]= 0,137298584	D[341]= 0,133590698	D[342]= 0,129562378	D[343]= 0,125259399
D[344]= 0,120697021	D[345]= 0,115921021	D[346]= 0,110946655	D[347]= 0,105819702
D[348]= 0,100540161	D[349]= 0,095169067	D[350]= 0,089706421	D[351]= 0,084182739
D[352]= 0,078628540	D[353]= 0,073059082	D[354]= 0,067520142	D[355]= 0,061996460
D[356]= 0,056533813	D[357]= 0,051132202	D[358]= 0,045837402	D[359]= 0,040634155
D[360]= 0,035552979	D[361]= 0,030609131	D[362]= 0,025817871	D[363]= 0,021179199
D[364]= 0,016708374	D[365]= 0,012420654	D[366]= 0,008316040	D[367]= 0,004394531
D[368]= 0,000686646	D[369]=-0,002822876	D[370]=-0,006134033	D[371]=-0,009231567
D[372]=-0,012115479	D[373]=-0,014801025	D[374]=-0,017257690	D[375]=-0,019531250
D[376]=-0,021575928	D[377]=-0,023422241	D[378]=-0,025085449	D[379]=-0,026535034
D[380]=-0,027801514	D[381]=-0,028884888	D[382]=-0,029785156	D[383]=-0,030517578
D[384]= 0,031082153	D[385]= 0,031478882	D[386]= 0,031738281	D[387]= 0,031845093
D[388]= 0,031814575	D[389]= 0,031661987	D[390]= 0,031387329	D[391]= 0,031005859
D[392]= 0,030532837	D[393]= 0,029937744	D[394]= 0,029281616	D[395]= 0,028533936
D[396]= 0,027725220	D[397]= 0,026840210	D[398]= 0,025909424	D[399]= 0,024932861
D[400]= 0,023910522	D[401]= 0,022857666	D[402]= 0,021789551	D[403]= 0,020690918
D[404]= 0,019577026	D[405]= 0,018463135	D[406]= 0,017349243	D[407]= 0,016235352
D[408]= 0,015121460	D[409]= 0,014022827	D[410]= 0,012939453	D[411]= 0,011886597
D[412]= 0,010848999	D[413]= 0,009841919	D[414]= 0,008865356	D[415]= 0,007919312
D[416]= 0,007003784	D[417]= 0,006118774	D[418]= 0,005294800	D[419]= 0,004486084
D[420]= 0,003723145	D[421]= 0,003005981	D[422]= 0,002334595	D[423]= 0,001693726
D[424]= 0,001098633	D[425]= 0,000549316	D[426]= 0,000030518	D[427]=-0,000442505
D[428]=-0,000869751	D[429]=-0,001266479	D[430]=-0,001617432	D[431]=-0,001937866
D[432]=-0,002227783	D[433]=-0,002487183	D[434]=-0,002700806	D[435]=-0,002883911
D[436]=-0,003051758	D[437]=-0,003173828	D[438]=-0,003280640	D[439]=-0,003372192
D[440]=-0,003417969	D[441]=-0,003463745	D[442]=-0,003479004	D[443]=-0,003479004
D[444]=-0,003463745	D[445]=-0,003433228	D[446]=-0,003387451	D[447]=-0,003326416
D[448]= 0,003250122	D[449]= 0,003173828	D[450]= 0,003082275	D[451]= 0,002990723
D[452]= 0,002899170	D[453]= 0,002792358	D[454]= 0,002685547	D[455]= 0,002578735

D[456]= 0,002456665	D[457]= 0,002349854	D[458]= 0,002243042	D[459]= 0,002120972
D[460]= 0,002014160	D[461]= 0,001907349	D[462]= 0,001785278	D[463]= 0,001693726
D[464]= 0,001586914	D[465]= 0,001480103	D[466]= 0,001388550	D[467]= 0,001296997
D[468]= 0,001205444	D[469]= 0,001113892	D[470]= 0,001037598	D[471]= 0,000961304
D[472]= 0,000885010	D[473]= 0,000808716	D[474]= 0,000747681	D[475]= 0,000686646
D[476]= 0,000625610	D[477]= 0,000579834	D[478]= 0,000534058	D[479]= 0,000473022
D[480]= 0,000442505	D[481]= 0,000396729	D[482]= 0,000366211	D[483]= 0,000320435
D[484]= 0,000289917	D[485]= 0,000259399	D[486]= 0,000244141	D[487]= 0,000213623
D[488]= 0,000198364	D[489]= 0,000167847	D[490]= 0,000152588	D[491]= 0,000137329
D[492]= 0,000122070	D[493]= 0,000106812	D[494]= 0,000106812	D[495]= 0,000091553
D[496]= 0,000076294	D[497]= 0,000076294	D[498]= 0,000061035	D[499]= 0,000061035
D[500]= 0,000045776	D[501]= 0,000045776	D[502]= 0,000030518	D[503]= 0,000030518
D[504]= 0,000030518	D[505]= 0,000030518	D[506]= 0,000015259	D[507]= 0,000015259
D[508]= 0,000015259	D[509]= 0,000015259	D[510]= 0,000015259	D[511]= 0,000015259

Table B.4 -- Layer II classes of quantization

Number of steps	C	D	grouping	Samples per codeword	Bits per codeword
3	1,33333333333	0,50000000000	yes	3	5
5	1,60000000000	0,50000000000	yes	3	7
7	1,14285714286	0,25000000000	no	1	3
9	1,77777777777	0,50000000000	yes	3	10
15	1,06666666666	0,12500000000	no	1	4
31	1,03225806452	0,06250000000	no	1	5
63	1,01587301587	0,03125000000	no	1	6
127	1,00787401575	0,01562500000	no	1	7
255	1,00392156863	0,00781250000	no	1	8
511	1,00195694716	0,00390625000	no	1	9
1023	1,00097751711	0,00195312500	no	1	10
2047	1,00048851979	0,00097656250	no	1	11
4095	1,00024420024	0,00048828125	no	1	12
8191	1,00012208522	0,00024414063	no	1	13
16383	1,00006103888	0,00012207031	no	1	14
32767	1,00003051851	0,00006103516	no	1	15
65535	1,00001525902	0,00003051758	no	1	16

Table B.5 -- Number of protected audio_data bits

Layer	Protected Fields
I	bits 16...31 of header bit allocation
II	bits 16...31 of header bit allocation scalefactor selection information
III	bits 16...31 of header side information: - bits 0...135 of audio_data in single_channel mode - bits 0...255 of audio_data in other modes

Table B.6 -- Layer III preemphasis (pretab)

scalefactor band (cb)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
pretab[cb]	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	2	3	3	3	2

Table B.7 -- Huffman codes for Layer III

Huffman code table for quadruples (A)

v w x y	hlen	hcod
0 0 0 0	1	1
0 0 0 1	4	0101
0 0 1 0	4	0100
0 0 1 1	5	00101
0 1 0 0	4	0110
0 1 0 1	6	000101
0 1 1 0	5	00100
0 1 1 1	6	000100
1 0 0 0	4	0111
1 0 0 1	5	00011
1 0 1 0	5	00110
1 0 1 1	6	000000
1 1 0 0	5	00111
1 1 0 1	6	000010
1 1 1 0	6	000011
1 1 1 1	6	000001

Huffman code table for quadruples (B)

v w x y	hlen	hcod
0 0 0 0	4	1111
0 0 0 1	4	1110
0 0 1 0	4	1101
0 0 1 1	4	1100
0 1 0 0	4	1011
0 1 0 1	4	1010
0 1 1 0	4	1001
0 1 1 1	4	1000
1 0 0 0	4	0111
1 0 0 1	4	0110
1 0 1 0	4	0101
1 0 1 1	4	0100
1 1 0 0	4	0011
1 1 0 1	4	0010
1 1 1 0	4	0001
1 1 1 1	4	0000

Huffman code table 0

linbits=0

x	y	hlen
0	0	0

Huffman code table 1

linbits=0

x	y	hlen	hcod
0	0	1	1
0	1	3	001
1	0	2	01
1	1	3	000

Huffman code table 2

linbits=0

x	y	hlen	hcod
0	0	1	1
0	1	3	010
0	2	6	000001
1	0	3	011
1	1	3	001
1	2	5	00001
2	0	5	00011
2	1	5	00010
2	2	6	000000

Huffman code table 3

linbits=0

x	y	hlen	hcod
0	0	2	11
0	1	2	10
0	2	6	000001
1	0	3	001
1	1	2	01
1	2	5	00001
2	0	5	00011
2	1	5	00010
2	2	6	000000

Huffman code table 4

not used

Huffman code table 5

linbits=0

x	y	hlen	hcod
0	0	1	1
0	1	3	010
0	2	6	000110
0	3	7	0000101
1	0	3	011
1	1	3	001
1	2	6	000100
1	3	7	0000100
2	0	6	000111
2	1	6	000101
2	2	7	0000111
2	3	8	00000001
3	0	7	0000110
3	1	6	000001
3	2	7	0000001
3	3	8	00000000

Huffman code table 6

linbits=0

x	y	hlen	hcod
0	0	3	111
0	1	3	011
0	2	5	00101
0	3	7	0000001
1	0	3	110
1	1	2	10
1	2	4	0011
1	3	5	00010
2	0	4	0101
2	1	4	0100
2	2	5	00100
2	3	6	000001
3	0	6	000011
3	1	5	00011
3	2	6	000010
3	3	7	0000000

Huffman code table 7

linbits=0

x	y	hlen	hcod
0	0	1	1
0	1	3	010
0	2	6	001010
0	3	8	00010011
0	4	8	00010000
0	5	9	000001010
1	0	3	011
1	1	4	0011
1	2	6	000111
1	3	7	0001010
1	4	7	0000101
1	5	8	00000011
2	0	6	001011
2	1	5	00100
2	2	7	0001101
2	3	8	00010001
2	4	8	00001000
2	5	9	000000100
3	0	7	0001100
3	1	7	0001011
3	2	8	00010010
3	3	9	000001111
3	4	9	000001011
3	5	9	000000010
4	0	7	0000111
4	1	7	0000110
4	2	8	00001001
4	3	9	000001110
4	4	9	000000011
4	5	10	0000000001
5	0	8	00000110
5	1	8	00000100
5	2	9	000000101
5	3	10	0000000011
5	4	10	0000000010
5	5	10	0000000000

Huffman code table 8

linbits=0

x	y	hlen	hcod
0	0	2	11
0	1	3	100
0	2	6	000110
0	3	8	00010010
0	4	8	00001100
0	5	9	000000101
1	0	3	101
1	1	2	01
1	2	4	0010
1	3	8	00010000
1	4	8	00001001
1	5	8	00000011
2	0	6	000111
2	1	4	0011
2	2	6	000101
2	3	8	00001110
2	4	8	00000111
2	5	9	000000011
3	0	8	00010011
3	1	8	00010001
3	2	8	00001111
3	3	9	000001101
3	4	9	000001010
3	5	10	0000000100
4	0	8	00001101
4	1	7	0000101
4	2	8	00001000
4	3	9	000001011
4	4	10	0000000101
4	5	10	0000000001
5	0	9	000001100
5	1	8	00000100
5	2	9	000000100
5	3	9	000000001
5	4	11	00000000001
5	5	11	00000000000

Huffman code table 9

linbits=0

x	y	hlen	hcod
0	0	3	111
0	1	3	101
0	2	5	01001
0	3	6	001110
0	4	8	00001111
0	5	9	000000111
1	0	3	110
1	1	3	100
1	2	4	0101
1	3	5	00101
1	4	6	000110
1	5	8	00000111
2	0	4	0111
2	1	4	0110
2	2	5	01000
2	3	6	001000
2	4	7	0001000
2	5	8	00000101
3	0	6	001111
3	1	5	00110
3	2	6	001001
3	3	7	0001010
3	4	7	0000101
3	5	8	00000001
4	0	7	0001011
4	1	6	000111
4	2	7	0001001
4	3	7	0000110
4	4	8	00000100
4	5	9	000000001
5	0	8	00001110
5	1	7	0000100
5	2	8	00000110
5	3	8	00000010
5	4	9	000000110
5	5	9	000000000

Huffman code table 10

linbits=0

x	y	hlen	hcod
0	0	1	1
0	1	3	010
0	2	6	001010
0	3	8	00010111
0	4	9	000100011
0	5	9	000011110
0	6	9	000001100
0	7	10	0000010001
1	0	3	011
1	1	4	0011
1	2	6	001000
1	3	7	0001100
1	4	8	00010010
1	5	9	000010101
1	6	8	00001100
1	7	8	00000111
2	0	6	001011
2	1	6	001001
2	2	7	0001111
2	3	8	00010101
2	4	9	000100000
2	5	10	0000101000
2	6	9	000010011
2	7	9	000000110
3	0	7	0001110
3	1	7	0001101
3	2	8	00010110
3	3	9	000100010
3	4	10	0000101110
3	5	10	0000010111
3	6	9	000010010
3	7	10	0000000111
4	0	8	00010100
4	1	8	00010011
4	2	9	000100001
4	3	10	0000101111
4	4	10	0000011011
4	5	10	0000010110
4	6	10	0000001001
4	7	10	0000000011
5	0	9	000011111
5	1	9	000010110
5	2	10	0000101001
5	3	10	0000011010
5	4	11	00000010101
5	5	11	00000010100
5	6	10	0000000101
5	7	11	00000000011
6	0	8	00001110
6	1	8	00001101
6	2	9	000001010
6	3	10	0000001011
6	4	10	0000010000
6	5	10	0000000110
6	6	11	00000000101
6	7	11	00000000001
7	0	9	000001001
7	1	8	00001000
7	2	9	000000111
7	3	10	0000001000
7	4	10	0000000100
7	5	11	00000000100
7	6	11	00000000010
7	7	11	00000000000

Huffman code table 11

linbits=0

x	y	hlen	hcod
0	0	2	11
0	1	3	100
0	2	5	01010
0	3	7	0011000
0	4	8	00100010
0	5	9	000100001
0	6	8	00010101
0	7	9	000001111
1	0	3	101
1	1	3	011
1	2	4	0100
1	3	6	001010
1	4	8	00100000
1	5	8	00010001
1	6	7	0001011
1	7	8	00001010
2	0	5	01011
2	1	5	00111
2	2	6	001101
2	3	7	0010010
2	4	8	00011110
2	5	9	000011111
2	6	8	00010100
2	7	8	00000101
3	0	7	0011001
3	1	6	001011
3	2	7	0010011
3	3	9	000111011
3	4	8	00011011
3	5	10	0000010010
3	6	8	00001100
3	7	9	000000101
4	0	8	00100011
4	1	8	00100001
4	2	8	00011111
4	3	9	000111010
4	4	9	000011110
4	5	10	0000010000
4	6	9	000000111
4	7	10	0000000101
5	0	8	00011100
5	1	8	00011010
5	2	9	000100000
5	3	10	0000010011
5	4	10	0000010001
5	5	11	00000001111
5	6	10	0000001000
5	7	11	00000001110
6	0	8	00001110
6	1	7	0001100
6	2	7	0001001
6	3	8	00001101
6	4	9	000001110
6	5	10	0000001001
6	6	10	0000000100
6	7	10	0000000001
7	0	8	00001011
7	1	7	0000100
7	2	8	00000110
7	3	9	000000110
7	4	10	0000000110
7	5	10	0000000011
7	6	10	0000000010
7	7	10	0000000000

Huffman code table 12

linbits=0

x	y	hlen	hcod
0	0	4	1001
0	1	3	110
0	2	5	10000
0	3	7	0100001
0	4	8	00101001
0	5	9	000100111
0	6	9	000100110
0	7	9	000011010
1	0	3	111
1	1	3	101
1	2	4	0110
1	3	5	01001
1	4	7	0010111
1	5	7	0010000
1	6	8	00011010
1	7	8	00001011
2	0	5	10001
2	1	4	0111
2	2	5	01011
2	3	6	001110
2	4	7	0010101
2	5	8	00011110
2	6	7	0001010
2	7	8	00000111
3	0	6	010001
3	1	5	01010
3	2	6	001111
3	3	6	001100
3	4	7	0010010
3	5	8	00011100
3	6	8	00001110
3	7	8	00000101
4	0	7	0100000
4	1	6	001101
4	2	7	0010110
4	3	7	0010011
4	4	8	00010010
4	5	8	00010000
4	6	8	00001001
4	7	9	000000101
5	0	8	00101000
5	1	7	0010001
5	2	8	00011111
5	3	8	00011101
5	4	8	00010001
5	5	9	000001101
5	6	8	00000100
5	7	9	000000010
6	0	8	00011011
6	1	7	0001100
6	2	7	0001011
6	3	8	00001111
6	4	8	00001010
6	5	9	000000111
6	6	9	000000100
6	7	10	0000000001
7	0	9	000011011
7	1	8	00001100
7	2	8	00001000
7	3	9	000001100
7	4	9	000000110
7	5	9	000000011
7	6	9	000000001
7	7	10	0000000000

Huffman code table 13

linbits=0

x	y	hlen	hcod
0	0	1	1
0	1	4	0101
0	2	6	001110
0	3	7	0010101
0	4	8	00100010
0	5	9	000110011
0	6	9	000101110
0	7	10	0001000111
0	8	9	000101010
0	9	10	0000110100
0	10	11	00001000100
0	11	11	00000110100
0	12	12	000001000011
0	13	12	000000101100
0	14	13	0000000101011
0	15	13	0000000010011
1	0	3	011
1	1	4	0100
1	2	6	001100
1	3	7	0010011
1	4	8	00011111
1	5	8	00011010
1	6	9	000101100
1	7	9	000100001
1	8	9	000011111
1	9	9	000011000
1	10	10	0000100000
1	11	10	0000011000
1	12	11	00000011111
1	13	12	000000100011
1	14	12	000000010110
1	15	12	000000001110
2	0	6	001111
2	1	6	001101
2	2	7	0010111
2	3	8	00100100
2	4	9	000111011
2	5	9	000110001
2	6	10	0001001101
2	7	10	0001000001
2	8	9	000011101
2	9	10	0000101000
2	10	10	0000011110
2	11	11	00000101000
2	12	11	00000011011
2	13	12	000000100001
2	14	13	0000000101010
2	15	13	0000000010000
3	0	7	0010110
3	1	7	0010100
3	2	8	00100101
3	3	9	000111101
3	4	9	000111000
3	5	10	0001001111
3	6	10	0001001001
3	7	10	0001000000
3	8	10	0000101011
3	9	11	00001001100
3	10	11	00000111000
3	11	11	00000100101
3	12	11	00000011010
3	13	12	000000011111
3	14	13	0000000011001
3	15	13	0000000001110
4	0	8	00100011
4	1	7	0010000
4	2	9	000111100
4	3	9	000111001
4	4	10	0001100001
4	5	10	0001001011
4	6	11	00001110010

4	7	11	00001011011
4	8	10	0000110110
4	9	11	00001001001
4	10	11	00000110111
4	11	12	000000101001
4	12	12	000000110000
4	13	13	0000000110101
4	14	13	0000000010111
4	15	14	00000000011000
5	0	9	000111010
5	1	8	00011011
5	2	9	000110010
5	3	10	0001100000
5	4	10	0001001100
5	5	10	0001000110
5	6	11	00001011101
5	7	11	00001010100
5	8	11	00001001101
5	9	11	00000111010
5	10	12	000001001111
5	11	11	00000011101
5	12	13	0000001001010
5	13	13	0000000110001
5	14	14	00000000101001
5	15	14	00000000010001
6	0	9	000101111
6	1	9	000101101
6	2	10	0001001110
6	3	10	0001001010
6	4	11	00001110011
6	5	11	00001011110
6	6	11	00001011010
6	7	11	00001001111
6	8	11	00001000101
6	9	12	000001010011
6	10	12	000001000111
6	11	12	000000110010
6	12	13	0000000111011
6	13	13	0000000100110
6	14	14	00000000100100
6	15	14	00000000001111
7	0	10	0001001000
7	1	9	000100010
7	2	10	0000111000
7	3	11	00001011111
7	4	11	00001011100
7	5	11	00001010101
7	6	12	000001011011
7	7	12	000001011010
7	8	12	000001010110
7	9	12	000001001001
7	10	13	0000001001101
7	11	13	0000001000001
7	12	13	0000000110011
7	13	14	00000000101100
7	14	16	0000000000101011
7	15	16	0000000000101010
8	0	9	000101011
8	1	8	00010100
8	2	9	000011110
8	3	10	0000101100
8	4	10	0000110111
8	5	11	00001001110
8	6	11	00001001000
8	7	12	000001010111
8	8	12	000001001110
8	9	12	000000111101
8	10	12	000000101110
8	11	13	0000000110110
8	12	13	0000000100101
8	13	14	00000000011110
8	14	15	000000000010100
8	15	15	000000000010000
9	0	10	0000110101
9	1	9	000011001
9	2	10	0000101001
9	3	10	0000100101
9	4	11	00000101100
9	5	11	00000111011
9	6	11	00000110110
9	7	13	0000001010001
9	8	12	000001000010
9	9	13	0000001001100
9	10	13	0000000111001
9	11	14	00000000110110
9	12	14	00000000100101
9	13	14	00000000010010
9	14	16	0000000000100111
9	15	15	000000000001011
10	0	10	0000100011
10	1	10	0000100001
10	2	10	0000011111
10	3	11	00000111001
10	4	11	00000101010
10	5	12	000001010010
10	6	12	000001001000
10	7	13	0000001010000
10	8	12	000000101111
10	9	13	0000000111010
10	10	14	00000000110111
10	11	13	0000000010101
10	12	14	00000000010110
10	13	15	000000000011010
10	14	16	0000000000100110
10	15	17	00000000000010110
11	0	11	00000110101
11	1	10	0000011001
11	2	10	0000010111
11	3	11	00000100110
11	4	12	000001000110
11	5	12	000000111100
11	6	12	000000110011
11	7	12	000000100100
11	8	13	0000000110111
11	9	13	0000000011010
11	10	13	0000000100010
11	11	14	00000000010111
11	12	15	000000000011011
11	13	15	000000000001110
11	14	15	000000000001001
11	15	16	0000000000000111
12	0	11	00000100010
12	1	11	00000100000
12	2	11	00000011100
12	3	12	000000100111
12	4	12	000000110001
12	5	13	0000001001011
12	6	12	000000011110
12	7	13	0000000110100
12	8	14	00000000110000
12	9	14	00000000101000
12	10	15	000000000110100
12	11	15	000000000011100
12	12	15	000000000010010
12	13	16	0000000000010001
12	14	16	0000000000001001
12	15	16	0000000000000101
13	0	12	000000101101
13	1	11	00000010101
13	2	12	000000100010
13	3	13	0000001000000
13	4	13	0000000111000
13	5	13	0000000110010
13	6	14	00000000110001
13	7	14	00000000101101
13	8	14	00000000011111
13	9	14	00000000010011
13	10	14	00000000001100
13	11	15	000000000001111
13	12	16	0000000000001010
13	13	15	000000000000111
13	14	16	0000000000000110
13	15	16	0000000000000011
14	0	13	0000000110000
14	1	12	000000010111
14	2	12	000000010100
14	3	13	0000000100111
14	4	13	0000000100100
14	5	13	0000000100011
14	6	15	000000000110101
14	7	14	00000000010101
14	8	14	00000000010000
14	9	17	00000000000010111
14	10	15	000000000001101
14	11	15	000000000001010
14	12	15	000000000000110
14	13	17	00000000000000001
14	14	16	0000000000000100
14	15	16	0000000000000010
15	0	12	000000010000
15	1	12	000000001111
15	2	13	0000000010001
15	3	14	00000000011011
15	4	14	00000000011001
15	5	14	00000000010100
15	6	15	000000000011101
15	7	14	00000000001011
15	8	15	000000000010001
15	9	15	000000000001100
15	10	16	0000000000010000
15	11	16	0000000000001000
15	12	19	0000000000000000001
15	13	18	000000000000000001
15	14	19	0000000000000000000
15	15	16	0000000000000001

Huffman code table 14

not used

Huffman code table 15

linbits=0

x	y	hlen	hcod
0	0	3	111
0	1	4	1100
0	2	5	10010
0	3	7	0110101
0	4	7	0101111
0	5	8	01001100
0	6	9	001111100
0	7	9	001101100
0	8	9	001011001
0	9	10	0001111011
0	10	10	0001101100
0	11	11	00001110111
0	12	11	00001101011
0	13	11	00001010001
0	14	12	000001111010
0	15	13	0000000111111
1	0	4	1101
1	1	3	101
1	2	5	10000
1	3	6	011011
1	4	7	0101110
1	5	7	0100100
1	6	8	00111101
1	7	8	00110011
1	8	8	00101010
1	9	9	001000110
1	10	9	000110100
1	11	10	0001010011
1	12	10	0001000001
1	13	10	0000101001
1	14	11	00000111011
1	15	11	00000100100
2	0	5	10011

2	1	5	10001
2	2	5	01111
2	3	6	011000
2	4	7	0101001
2	5	7	0100010
2	6	8	00111011
2	7	8	00110000
2	8	8	00101000
2	9	9	001000000
2	10	9	000110010
2	11	10	0001001110
2	12	10	0000111110
2	13	11	00001010000
2	14	11	00000111000
2	15	11	00000100001
3	0	6	011101
3	1	6	011100
3	2	6	011001
3	3	7	0101011
3	4	7	0100111
3	5	8	00111111
3	6	8	00110111
3	7	9	001011101
3	8	9	001001100
3	9	9	000111011
3	10	10	0001011101
3	11	10	0001001000
3	12	10	0000110110
3	13	11	00001001011
3	14	11	00000110010
3	15	11	00000011101
4	0	7	0110100
4	1	6	010110
4	2	7	0101010
4	3	7	0101000
4	4	8	01000011
4	5	8	00111001
4	6	9	001011111
4	7	9	001001111
4	8	9	001001000
4	9	9	000111001
4	10	10	0001011001
4	11	10	0001000101
4	12	10	0000110001
4	13	11	00001000010
4	14	11	00000101110
4	15	11	00000011011
5	0	8	01001101
5	1	7	0100101
5	2	7	0100011
5	3	8	01000010
5	4	8	00111010
5	5	8	00110100
5	6	9	001011011
5	7	9	001001010
5	8	9	000111110
5	9	9	000110000
5	10	10	0001001111
5	11	10	0000111111
5	12	11	00001011010
5	13	11	00000111110
5	14	11	00000101000
5	15	12	000000100110
6	0	9	001111101
6	1	7	0100000
6	2	8	00111100
6	3	8	00111000
6	4	8	00110010
6	5	9	001011100
6	6	9	001001110
6	7	9	001000001
6	8	9	000110111
6	9	10	0001010111
6	10	10	0001000111
6	11	10	0000110011
6	12	11	00001001001
6	13	11	00000110011
6	14	12	000001000110
6	15	12	000000011110
7	0	9	001101101
7	1	8	00110101
7	2	8	00110001
7	3	9	001011110
7	4	9	001011000
7	5	9	001001011
7	6	9	001000010
7	7	10	0001111010
7	8	10	0001011011
7	9	10	0001001001
7	10	10	0000111000
7	11	10	0000101010
7	12	11	00001000000
7	13	11	00000101100
7	14	11	00000010101
7	15	12	000000011001
8	0	9	001011010
8	1	8	00101011
8	2	8	00101001
8	3	9	001001101
8	4	9	001001001
8	5	9	000111111
8	6	9	000111000
8	7	10	0001011100
8	8	10	0001001101
8	9	10	0001000010
8	10	10	0000101111
8	11	11	00001000011
8	12	11	00000110000
8	13	12	000000110101
8	14	12	000000100100
8	15	12	000000010100
9	0	9	001000111
9	1	8	00100010
9	2	9	001000011
9	3	9	000111100
9	4	9	000111010
9	5	9	000110001
9	6	10	0001011000
9	7	10	0001001100
9	8	10	0001000011
9	9	11	00001101010
9	10	11	00001000111
9	11	11	00000110110
9	12	11	00000100110
9	13	12	000000100111
9	14	12	000000010111
9	15	12	000000001111
10	0	10	0001101101
10	1	9	000110101
10	2	9	000110011
10	3	9	000101111
10	4	10	0001011010
10	5	10	0001010010
10	6	10	0000111010
10	7	10	0000111001
10	8	10	0000110000
10	9	11	00001001000
10	10	11	00000111001
10	11	11	00000101001
10	12	11	00000010111
10	13	12	000000011011
10	14	13	0000000111110
10	15	12	000000001001
11	0	10	0001010110
11	1	9	000101010
11	2	9	000101000
11	3	9	000100101
11	4	10	0001000110
11	5	10	0001000000
11	6	10	0000110100
11	7	10	0000101011
11	8	11	00001000110
11	9	11	00000110111
11	10	11	00000101010
11	11	11	00000011001
11	12	12	000000011101
11	13	12	000000010010
11	14	12	000000001011
11	15	13	0000000001011
12	0	11	00001110110
12	1	10	0001000100
12	2	9	000011110
12	3	10	0000110111
12	4	10	0000110010
12	5	10	0000101110
12	6	11	00001001010
12	7	11	00001000001
12	8	11	00000110001
12	9	11	00000100111
12	10	11	00000011000
12	11	11	00000010000
12	12	12	000000010110
12	13	12	000000001101
12	14	13	0000000001110
12	15	13	0000000000111
13	0	11	00001011011
13	1	10	0000101100
13	2	10	0000100111
13	3	10	0000100110
13	4	10	0000100010
13	5	11	00000111111
13	6	11	00000110100
13	7	11	00000101101
13	8	11	00000011111
13	9	12	000000110100
13	10	12	000000011100
13	11	12	000000010011
13	12	12	000000001110
13	13	12	000000001000
13	14	13	0000000001001
13	15	13	0000000000011
14	0	12	000001111011
14	1	11	00000111100
14	2	11	00000111010
14	3	11	00000110101
14	4	11	00000101111
14	5	11	00000101011
14	6	11	00000100000
14	7	11	00000010110
14	8	12	000000100101
14	9	12	000000011000
14	10	12	000000010001
14	11	12	000000001100
14	12	13	0000000001111
14	13	13	0000000001010
14	14	12	000000000010
14	15	13	0000000000001
15	0	12	000001000111
15	1	11	00000100101
15	2	11	00000100010
15	3	11	00000011110
15	4	11	00000011100
15	5	11	00000010100
15	6	11	00000010001
15	7	12	000000011010
15	8	12	000000010101
15	9	12	000000010000
15	10	12	000000001010
15	11	12	000000000110
15	12	13	0000000001000
15	13	13	0000000000110
15	14	13	0000000000010
15	15	13	0000000000000

Huffman code table 16

linbits=1

x	y	hlen	hcod
0	0	1	1
0	1	4	0101
0	2	6	001110
0	3	8	00101100
0	4	9	001001010
0	5	9	000111111
0	6	10	0001101110
0	7	10	0001011101
0	8	11	00010101100
0	9	11	00010010101
0	10	11	00010001010
0	11	12	000011110010
0	12	12	000011100001
0	13	12	000011000011
0	14	13	0000101111000
0	15	9	000010001
1	0	3	011
1	1	4	0100
1	2	6	001100
1	3	7	0010100
1	4	8	00100011
1	5	9	000111110
1	6	9	000110101
1	7	9	000101111
1	8	10	0001010011
1	9	10	0001001011
1	10	10	0001000100
1	11	11	00001110111
1	12	12	000011001001
1	13	11	00001101011
1	14	12	000011001111
1	15	8	00001001
2	0	6	001111
2	1	6	001101
2	2	7	0010111
2	3	8	00100110
2	4	9	001000011
2	5	9	000111010
2	6	10	0001100111
2	7	10	0001011010
2	8	11	00010100001
2	9	10	0001001000
2	10	11	00001111111
2	11	11	00001110101
2	12	11	00001101110
2	13	12	000011010001
2	14	12	000011001110
2	15	9	000010000
3	0	8	00101101
3	1	7	0010101
3	2	8	00100111
3	3	9	001000101
3	4	9	001000000
3	5	10	0001110010
3	6	10	0001100011
3	7	10	0001010111
3	8	11	00010011110
3	9	11	00010001100
3	10	12	000011111100
3	11	12	000011010100
3	12	12	000011000111
3	13	13	0000110000011
3	14	13	0000101101101
3	15	10	0000011010
4	0	9	001001011
4	1	8	00100100
4	2	9	001000100
4	3	9	001000001
4	4	10	0001110011
4	5	10	0001100101
4	6	11	00010110011
4	7	11	00010100100
4	8	11	00010011011
4	9	12	000100001000
4	10	12	000011110110
4	11	12	000011100010
4	12	13	0000110001011
4	13	13	0000101111110
4	14	13	0000101101010
4	15	9	000001001
5	0	9	001000010
5	1	8	00011110
5	2	9	000111011
5	3	9	000111000
5	4	10	0001100110
5	5	11	00010111001
5	6	11	00010101101
5	7	12	000100001001
5	8	11	00010001110
5	9	12	000011111101
5	10	12	000011101000
5	11	13	0000110010000
5	12	13	0000110000100
5	13	13	0000101111010
5	14	14	00000110111101
5	15	10	0000010000
6	0	10	0001101111
6	1	9	000110110
6	2	9	000110100
6	3	10	0001100100
6	4	11	00010111000
6	5	11	00010110010
6	6	11	00010100000
6	7	11	00010000101
6	8	12	000100000001
6	9	12	000011110100
6	10	12	000011100100
6	11	12	000011011001
6	12	13	0000110000001
6	13	13	0000101101110
6	14	14	00001011001011
6	15	10	0000001010
7	0	10	0001100010
7	1	9	000110000
7	2	10	0001011011
7	3	10	0001011000
7	4	11	00010100101
7	5	11	00010011101
7	6	11	00010010100
7	7	12	000100000101
7	8	12	000011111000
7	9	13	0000110010111
7	10	13	0000110001101
7	11	13	0000101110100
7	12	13	0000101111100
7	13	15	000001101111001
7	14	15	000001101110100
7	15	10	0000001000
8	0	10	0001010101
8	1	10	0001010100
8	2	10	0001010001
8	3	11	00010011111
8	4	11	00010011100
8	5	11	00010001111
8	6	12	000100000100
8	7	12	000011111001
8	8	13	0000110101011
8	9	13	0000110010001
8	10	13	0000110001000
8	11	13	0000101111111
8	12	14	00001011010111
8	13	14	00001011001001
8	14	14	00001011000100
8	15	10	0000000111
9	0	11	00010011010
9	1	10	0001001100
9	2	10	0001001001
9	3	11	00010001101
9	4	11	00010000011
9	5	12	000100000000
9	6	12	000011110101
9	7	13	0000110101010
9	8	13	0000110010110
9	9	13	0000110001010
9	10	13	0000110000000
9	11	14	00001011011111
9	12	13	0000101100111
9	13	14	00001011000110
9	14	13	0000101100000
9	15	11	00000001011
10	0	11	00010001011
10	1	11	00010000001
10	2	10	0001000011
10	3	11	00001111101
10	4	12	000011110111
10	5	12	000011101001
10	6	12	000011100101
10	7	12	000011011011
10	8	13	0000110001001
10	9	14	00001011100111
10	10	14	00001011100001
10	11	14	00001011010000
10	12	15	000001101110101
10	13	15	000001101110010
10	14	14	00000110110111
10	15	10	0000000100
11	0	12	000011110011
11	1	11	00001111000
11	2	11	00001110110
11	3	11	00001110011
11	4	12	000011100011
11	5	12	000011011111
11	6	13	0000110001100
11	7	14	00001011101010
11	8	14	00001011100110
11	9	14	00001011100000
11	10	14	00001011010001
11	11	14	00001011001000
11	12	14	00001011000010
11	13	13	0000011011111
11	14	14	00000110110100
11	15	11	00000000110
12	0	12	000011001010
12	1	12	000011100000
12	2	12	000011011110
12	3	12	000011011010
12	4	12	000011011000
12	5	13	0000110000101
12	6	13	0000110000010
12	7	13	0000101111101
12	8	13	0000101101100
12	9	15	000001101111000
12	10	14	00000110111011
12	11	14	00001011000011
12	12	14	00000110111000
12	13	14	00000110110101
12	14	16	0000011011000000
12	15	11	00000000100
13	0	14	00001011101011
13	1	12	000011010011
13	2	12	000011010010
13	3	12	000011010000
13	4	13	0000101110010
13	5	13	0000101111011
13	6	14	00001011011110
13	7	14	00001011010011
13	8	14	00001011001010
13	9	16	0000011011000111
13	10	15	000001101110011
13	11	15	000001101101101
13	12	15	000001101101100
13	13	17	00000110110000011
13	14	15	000001101100001
13	15	11	00000000010
14	0	13	0000101111001

14	1	13	0000101110001
14	2	11	00001100110
14	3	12	000010111011
14	4	14	00001011010110
14	5	14	00001011010010
14	6	13	0000101100110
14	7	14	00001011000111
14	8	14	00001011000101
14	9	15	000001101100010
14	10	16	0000011011000110
14	11	15	000001101100111
14	12	17	00000110110000010
14	13	15	000001101100110
14	14	14	00000110110010
14	15	11	00000000000
15	0	9	000001100
15	1	8	00001010
15	2	8	00000111
15	3	9	000001011
15	4	9	000001010
15	5	10	0000010001
15	6	10	0000001011
15	7	10	0000001001
15	8	11	00000001101
15	9	11	00000001100
15	10	11	00000001010
15	11	11	00000000111
15	12	11	00000000101
15	13	11	00000000011
15	14	11	00000000001
15	15	8	00000011

Huffman code table 17

same as table 16, but linbits=2

Huffman code table 18

same as table 16, but linbits=3

Huffman code table 19

same as table 16, but linbits=4

Huffman code table 20

same as table 16, but linbits=6

Huffman code table 21

same as table 16, but linbits=8

Huffman code table 22

same as table 16, but linbits=10

Huffman code table 23

same as table 16, but linbits=13

Huffman code table 24

linbits=4

x	y	hlen	hcod
0	0	4	1111
0	1	4	1101
0	2	6	101110
0	3	7	1010000
0	4	8	10010010
0	5	9	100000110
0	6	9	011111000
0	7	10	0110110010
0	8	10	0110101010
0	9	11	01010011101
0	10	11	01010001101
0	11	11	01010001001
0	12	11	01001101101
0	13	11	01000000101
0	14	12	010000001000
0	15	9	001011000
1	0	4	1110
1	1	4	1100
1	2	5	10101
1	3	6	100110
1	4	7	1000111
1	5	8	10000010
1	6	8	01111010
1	7	9	011011000
1	8	9	011010001
1	9	9	011000110
1	10	10	0101000111
1	11	10	0101011001
1	12	10	0100111111
1	13	10	0100101001
1	14	10	0100010111
1	15	8	00101010
2	0	6	101111
2	1	5	10110
2	2	6	101001
2	3	7	1001010
2	4	7	1000100
2	5	8	10000000
2	6	8	01111000
2	7	9	011011101
2	8	9	011001111
2	9	9	011000010
2	10	9	010110110
2	11	10	0101010100
2	12	10	0100111011
2	13	10	0100100111
2	14	11	01000011101
2	15	7	0010010
3	0	7	1010001
3	1	6	100111
3	2	7	1001011
3	3	7	1000110
3	4	8	10000110
3	5	8	01111101
3	6	8	01110100
3	7	9	011011100
3	8	9	011001100
3	9	9	010111110
3	10	9	010110010
3	11	10	0101000101
3	12	10	0100110111
3	13	10	0100100101
3	14	10	0100001111
3	15	7	0010000
4	0	8	10010011
4	1	7	1001000
4	2	7	1000101
4	3	8	10000111
4	4	8	01111111
4	5	8	01110110
4	6	8	01110000
4	7	9	011010010
4	8	9	011001000
4	9	9	010111100
4	10	10	0101100000
4	11	10	0101000011
4	12	10	0100110010
4	13	10	0100011101
4	14	11	01000011100
4	15	7	0001110
5	0	9	100000111
5	1	7	1000010
5	2	8	10000001
5	3	8	01111110
5	4	8	01110111
5	5	8	01110010
5	6	9	011010110
5	7	9	011001010
5	8	9	011000000
5	9	9	010110100
5	10	10	0101010101
5	11	10	0100111101
5	12	10	0100101101
5	13	10	0100011001
5	14	10	0100000110
5	15	7	0001100
6	0	9	011111001
6	1	8	01111011
6	2	8	01111001
6	3	8	01110101
6	4	8	01110001
6	5	9	011010111
6	6	9	011001110
6	7	9	011000011
6	8	9	010111001
6	9	10	0101011011
6	10	10	0101001010
6	11	10	0100110100
6	12	10	0100100011
6	13	10	0100010000
6	14	11	01000001000
6	15	7	0001010
7	0	10	0110110011
7	1	8	01110011
7	2	8	01101111
7	3	8	01101101
7	4	9	011010011
7	5	9	011001011
7	6	9	011000100
7	7	9	010111011
7	8	10	0101100001
7	9	10	0101001100
7	10	10	0100111001
7	11	10	0100101010
7	12	10	0100011011
7	13	11	01000010011
7	14	11	00101111101
7	15	8	00010001
8	0	10	0110101011
8	1	9	011010100
8	2	9	011010000
8	3	9	011001101
8	4	9	011001001
8	5	9	011000001
8	6	9	010111010
8	7	9	010110001
8	8	9	010101001
8	9	10	0101000000
8	10	10	0100101111
8	11	10	0100011110
8	12	10	0100001100
8	13	11	01000000010
8	14	11	00101111001
8	15	8	00010000
9	0	10	0101001111
9	1	9	011000111
9	2	9	011000101
9	3	9	010111111

9	4	9	010111101
9	5	9	010110101
9	6	9	010101110
9	7	10	0101001101
9	8	10	0101000001
9	9	10	0100110001
9	10	10	0100100001
9	11	10	0100010011
9	12	11	01000001001
9	13	11	00101111011
9	14	11	00101110011
9	15	8	00001011
10	0	11	01010011100
10	1	9	010111000
10	2	9	010110111
10	3	9	010110011
10	4	9	010101111
10	5	10	0101011000
10	6	10	0101001011
10	7	10	0100111010
10	8	10	0100110000
10	9	10	0100100010
10	10	10	0100010101
10	11	11	01000010010
10	12	11	00101111111
10	13	11	00101110101
10	14	11	00101101110
10	15	8	00001010
11	0	11	01010001100
11	1	10	0101011010
11	2	9	010101011
11	3	9	010101000
11	4	9	010100100
11	5	10	0100111110
11	6	10	0100110101
11	7	10	0100101011
11	8	10	0100011111
11	9	10	0100010100
11	10	10	0100000111
11	11	11	01000000001
11	12	11	00101110111
11	13	11	00101110000
11	14	11	00101101010
11	15	8	00000110
12	0	11	01010001000
12	1	10	0101000010
12	2	10	0100111100
12	3	10	0100111000
12	4	10	0100110011
12	5	10	0100101110
12	6	10	0100100100
12	7	10	0100011100
12	8	10	0100001101
12	9	10	0100000101
12	10	11	01000000000
12	11	11	00101111000
12	12	11	00101110010
12	13	11	00101101100
12	14	11	00101100111
12	15	8	00000100
13	0	11	01001101100
13	1	10	0100101100
13	2	10	0100101000
13	3	10	0100100110
13	4	10	0100100000
13	5	10	0100011010
13	6	10	0100010001
13	7	10	0100001010
13	8	11	01000000011
13	9	11	00101111100
13	10	11	00101110110
13	11	11	00101110001
13	12	11	00101101101
13	13	11	00101101001
13	14	11	00101100101
13	15	8	00000010
14	0	12	010000001001
14	1	10	0100011000
14	2	10	0100010110
14	3	10	0100010010
14	4	10	0100001011
14	5	10	0100001000
14	6	10	0100000011
14	7	11	00101111110
14	8	11	00101111010
14	9	11	00101110100
14	10	11	00101101111
14	11	11	00101101011
14	12	11	00101101000
14	13	11	00101100110
14	14	11	00101100100
14	15	8	00000000
15	0	8	00101011
15	1	7	0010100
15	2	7	0010011
15	3	7	0010001
15	4	7	0001111
15	5	7	0001101
15	6	7	0001011
15	7	7	0001001
15	8	7	0000111
15	9	7	0000110
15	10	7	0000100
15	11	8	00000111
15	12	8	00000101
15	13	8	00000011
15	14	8	00000001
15	15	4	0011

Huffman code table 25

same as table 24, but linbits=5

Huffman code table 26

same as table 24, but linbits=6

Huffman code table 27

same as table 24, but linbits=7

Huffman code table 28

same as table 24, but linbits=8

Huffman code table 29

same as table 24, but linbits=9

Huffman code table 30

same as table 24, but linbits=11

Huffman code table 31

same as table 24, but linbits=13

Table B.8 -- Layer III scalefactor bands

These tables list the width of each scalefactor band. There are 21 bands at each sampling frequency for long (type 0,1 or 3) windows and 12 bands each for short windows.

Table B.8a. -- 32kHz sampling rate

long blocks:

scale factor band	width of band	index of start	index of end
0	4	0	3
1	4	4	7
2	4	8	11
3	4	12	15
4	4	16	19
5	4	20	23
6	6	24	29
7	6	30	35
8	8	36	43
9	10	44	53
10	12	54	65
11	16	66	81
12	20	82	101
13	24	102	125
14	30	126	155
15	38	156	193
16	46	194	239
17	56	240	295
18	68	296	363
19	84	364	447
20	102	448	549

short blocks:

scale factor band	width of band	index of start	index of end
0	4	0	3
1	4	4	7
2	4	8	11
3	4	12	15
4	6	16	21
5	8	22	29
6	12	30	41
7	16	42	57
8	20	58	77
9	26	78	103
10	34	104	137
11	42	138	179

Table B.8b. -- 44,1kHz sampling rate

long blocks:

scale factor band	width of band	index of start	index of end
0	4	0	3
1	4	4	7
2	4	8	11
3	4	12	15
4	4	16	19
5	4	20	23
6	6	24	29
7	6	30	35
8	8	36	43
9	8	44	51
10	10	52	61
11	12	62	73
12	16	74	89
13	20	90	109
14	24	110	133
15	28	134	161
16	34	162	195
17	42	196	237
18	50	238	287
19	54	288	341
20	76	342	417

short blocks:

scale factor band	width of band	index of start	index of end
0	4	0	3
1	4	4	7
2	4	8	11
3	4	12	15
4	6	16	21
5	8	22	29
6	10	30	39
7	12	40	51
8	14	52	65
9	18	66	83
10	22	84	105
11	30	106	135

Table B.8c. -- 48 kHz sampling rate

long blocks:

scale factor band	width of band	index of start	index of end
0	4	0	3
1	4	4	7
2	4	8	11
3	4	12	15
4	4	16	19
5	4	20	23
6	6	24	29
7	6	30	35
8	6	36	41
9	8	42	49
10	10	50	59
11	12	60	71
12	16	72	87
13	18	88	105
14	22	106	127
15	28	128	155
16	34	156	189
17	40	190	229
18	46	230	275
19	54	276	329
20	54	330	383

short blocks:

scale factor band	width of band	index of start	index of end
0	4	0	3
1	4	4	7
2	4	8	11
3	4	12	15
4	6	16	21
5	6	22	27
6	10	28	37
7	12	38	49
8	14	50	63
9	16	64	79
10	20	80	99
11	26	100	125

Table B.9 -- Layer III coefficients for aliasing reduction:

(i)	c_i
0	-0,6
1	-0,535
2	-0,33
3	-0,185
4	-0,095
5	-0,041
6	-0,0142
7	-0,0037

The butterfly coefficients cs_i and ca_i are calculated as follows:

$$cs_i = \frac{1}{\sqrt{1+c_i^2}}, \quad ca_i = \frac{c_i}{\sqrt{1+c_i^2}}$$

Annex C

(infomative)

The encoding process

C.1 Encoder

C.1.1 Overview

For each of the layers, an example of one suitable encoder with the corresponding flow-diagram is given in this annex. In subsequent clauses the analysis subband filter and the layer-specific encoding techniques are described. In annex D two examples of psychoacoustic models, which are common to all layers, are described. A short introduction describes the overall philosophy.

C.1.1.1 Introduction

The ISO/IEC 11172-3 (MPEG-Audio) algorithm is a psychoacoustic algorithm. The figure C.1 shows the primary parts of a psychoacoustic algorithm.

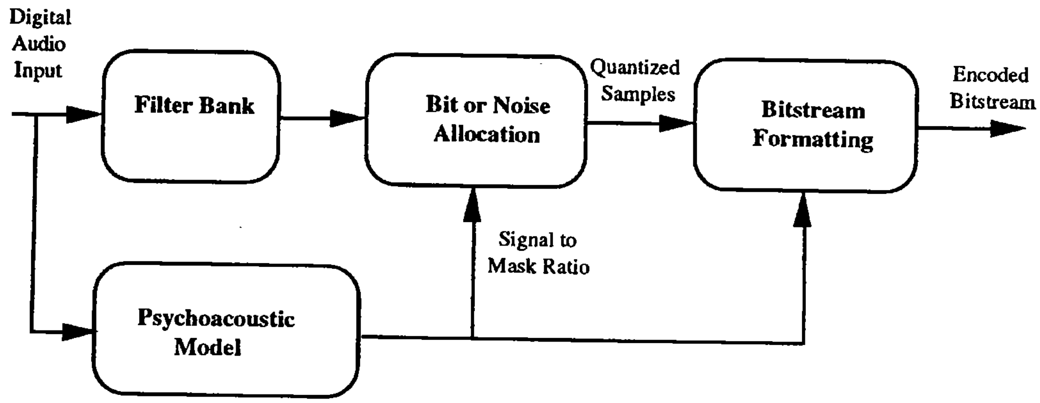


Figure C.1 -- ISO/IEC 11172-3 (MPEG-audio) encoder block diagram

The four primary parts of the psychoacoustic encoder are:

C.1.1.1.1 The filterbank

The filterbank does a time to frequency mapping. There are two filterbanks used in the ISO/IEC 11172-3 (MPEG-Audio) algorithm, a polyphase filterbank and a hybrid polyphase/MDCT filterbank. Each provides a specific mapping in time and frequency. These filterbanks are critically sampled (i.e. there are as many samples in the analyzed domain as there are in the time domain). These filterbanks provide the primary frequency separation for the encoder, and the reconstruction filters for the decoder. The output samples of the filterbank are quantized.

C.1.1.1.2 The psychoacoustic model

The psychoacoustic model calculates a just noticable noise-level for each band in the filterbank. This noise level is used in the bit or noise allocation to determine the actual quantizers and quantizer levels. There are two psychoacoustic models presented in annex D. While they can both be applied to any layer of the ISO/IEC 11172-3 (MPEG-Audio) algorithm, in practice Model 1 has been used for Layers I and II, and Model 2 for Layer III. In both psychoacoustic models, the final output of the model is a signal-to-mask ratio (SMR) for each band (Layers I and II) or group of bands (Layer III).

C.1.1.1.3 Bit or noise Allocation

The allocator looks at both the output samples from the filterbank and the SMR's from the psychoacoustic model, and adjusts the bit allocation (Layers I and II) or noise allocation (Layer III) in order simultaneously to meet both the bitrate requirements and the masking requirements. At low bitrates, these methods attempt to spend bits in a fashion that is psychoacousticly inoffensive when they cannot meet the psychoacoustic demand at the required bitrate.

C.1.1.1.4 The bitstream formatter

The bitstream formatter takes the quantized filterbank outputs, together with the bit allocation (Layers I and II) or noise allocation (Layer III) and other required side information, and encodes and formats that information in an efficient fashion. In the case of Layer III, the Huffman codes are also inserted at this point.

C.1.1.2 The filterbank

In Layers I and II, a filterbank with 32 subbands is used. In each subband, 12 or 36 samples are grouped for processing. In Layer III, the filterbank has a signal-dependent resolution, where there are either 6x32 or 18x32 frequency bands. In the case where there are 6x32 frequency samples, the 3 sets of each frequency are quantized separately.

C.1.1.3 Bit or noise allocation method

There are two different bitrate control methods explained in this annex. In Layers I and II this method is a bit allocation process, i.e. a number of bits is assigned to each sample (or group of samples) in each subband. The method for Layer III is a noise-allocation loop, where the quantizers are varied in an organized fashion, and the variable to be controlled is actually the injected noise. In either case, the result is a set of quantization parameters and quantized output samples that are given to the bitstream formatter.

C.1.1.4 Bitstream formatting

The bitstream formatter varies from layer to layer. In Layers I and II, a fixed PCM code is used for each subband sample, with the exception that in Layer II quantized samples may be grouped. In Layer III, Huffman codes are used to represent the quantized frequency samples. These Huffman codes are variable-length codes that allow for more efficient bitstream representation of the quantized samples at the cost of additional complexity.

C.1.2 Input high-pass filter

The encoding algorithms provide a frequency response down to d.c. However, in applications where this is not a requirement, it is recommended that a high-pass filter be included at the input of the encoder. The cut-off frequency should be in the range of 2 to 10 Hz.

The application of such a high-pass filter avoids an unneccessarily high bitrate requirement for the lowest subband and increases the overall audio quality.

C.1.3 Analysis subband filter

An analysis subband filterbank is used to split the broadband signal with sampling frequency f_s into 32 equally spaced subbands with sampling frequencies $f_s/32$. The flow chart of this process with the appropriate formulas is given in figure C.4 "Analysis Subband Filter Flow Chart". The analysis subband filtering includes the following steps:

- Input 32 audio samples.
- Build an input sample vector X of 512 elements. The 32 audio samples are shifted in at positions 0 to 31, the most recent one at position 0, and the 32 oldest elements are shifted out.
- Window vector X by vector C. The coefficients are to be found in table C.1.
- Calculate the 64 values Y_i according to the formula given in the flow chart.
- Calculate the 32 subband samples S_i by matrixing. The coefficients for the matrix can be calculated by the following formula:

$$M_{ik} = \cos\,[(2i + 1)(k - 16)\pi/64]\,, \quad \text{for } i = 0 \text{ to } 31, \text{ and } k = 0 \text{ to } 63.$$

Table C.1 -- Coefficients C_i of the Analysis Window

C[0]= 0,000000000	C[1]=-0,000000477	C[2]=-0,000000477	C[3]=-0,000000477
C[4]=-0,000000477	C[5]=-0,000000477	C[6]=-0,000000477	C[7]=-0,000000954
C[8]=-0,000000954	C[9]=-0,000000954	C[10]=-0,000000954	C[11]=-0,000001431
C[12]=-0,000001431	C[13]=-0,000001907	C[14]=-0,000001907	C[15]=-0,000002384
C[16]=-0,000002384	C[17]=-0,000002861	C[18]=-0,000003338	C[19]=-0,000003338
C[20]=-0,000003815	C[21]=-0,000004292	C[22]=-0,000004768	C[23]=-0,000005245
C[24]=-0,000006199	C[25]=-0,000006676	C[26]=-0,000007629	C[27]=-0,000008106
C[28]=-0,000009060	C[29]=-0,000010014	C[30]=-0,000011444	C[31]=-0,000012398
C[32]=-0,000013828	C[33]=-0,000014782	C[34]=-0,000016689	C[35]=-0,000018120
C[36]=-0,000019550	C[37]=-0,000021458	C[38]=-0,000023365	C[39]=-0,000025272
C[40]=-0,000027657	C[41]=-0,000030041	C[42]=-0,000032425	C[43]=-0,000034809
C[44]=-0,000037670	C[45]=-0,000040531	C[46]=-0,000043392	C[47]=-0,000046253
C[48]=-0,000049591	C[49]=-0,000052929	C[50]=-0,000055790	C[51]=-0,000059605
C[52]=-0,000062943	C[53]=-0,000066280	C[54]=-0,000070095	C[55]=-0,000073433
C[56]=-0,000076771	C[57]=-0,000080585	C[58]=-0,000083923	C[59]=-0,000087261
C[60]=-0,000090599	C[61]=-0,000093460	C[62]=-0,000096321	C[63]=-0,000099182
C[64]= 0,000101566	C[65]= 0,000103951	C[66]= 0,000105858	C[67]= 0,000107288
C[68]= 0,000108242	C[69]= 0,000108719	C[70]= 0,000108719	C[71]= 0,000108242
C[72]= 0,000106812	C[73]= 0,000105381	C[74]= 0,000102520	C[75]= 0,000099182
C[76]= 0,000095367	C[77]= 0,000090122	C[78]= 0,000084400	C[79]= 0,000077724
C[80]= 0,000069618	C[81]= 0,000060558	C[82]= 0,000050545	C[83]= 0,000039577
C[84]= 0,000027180	C[85]= 0,000013828	C[86]=-0,000000954	C[87]=-0,000017166
C[88]=-0,000034332	C[89]=-0,000052929	C[90]=-0,000072956	C[91]=-0,000093937
C[92]=-0,000116348	C[93]=-0,000140190	C[94]=-0,000165462	C[95]=-0,000191212
C[96]=-0,000218868	C[97]=-0,000247478	C[98]=-0,000277042	C[99]=-0,000307560
C[100]=-0,000339031	C[101]=-0,000371456	C[102]=-0,000404358	C[103]=-0,000438213
C[104]=-0,000472546	C[105]=-0,000507355	C[106]=-0,000542164	C[107]=-0,000576973
C[108]=-0,000611782	C[109]=-0,000646591	C[110]=-0,000680923	C[111]=-0,000714302
C[112]=-0,000747204	C[113]=-0,000779152	C[114]=-0,000809669	C[115]=-0,000838757
C[116]=-0,000866413	C[117]=-0,000891685	C[118]=-0,000915051	C[119]=-0,000935555
C[120]=-0,000954151	C[121]=-0,000968933	C[122]=-0,000980854	C[123]=-0,000989437
C[124]=-0,000994205	C[125]=-0,000995159	C[126]=-0,000991821	C[127]=-0,000983715
C[128]= 0,000971317	C[129]= 0,000953674	C[130]= 0,000930786	C[131]= 0,000902653
C[132]= 0,000868797	C[133]= 0,000829220	C[134]= 0,000783920	C[135]= 0,000731945
C[136]= 0,000674248	C[137]= 0,000610352	C[138]= 0,000539303	C[139]= 0,000462532
C[140]= 0,000378609	C[141]= 0,000288486	C[142]= 0,000191689	C[143]= 0,000088215
C[144]=-0,000021458	C[145]=-0,000137329	C[146]=-0,000259876	C[147]=-0,000388145
C[148]=-0,000522137	C[149]=-0,000661850	C[150]=-0,000806808	C[151]=-0,000956535
C[152]=-0,001111031	C[153]=-0,001269817	C[154]=-0,001432419	C[155]=-0,001597881
C[156]=-0,001766682	C[157]=-0,001937389	C[158]=-0,002110004	C[159]=-0,002283096
C[160]=-0,002457142	C[161]=-0,002630711	C[162]=-0,002803326	C[163]=-0,002974033
C[164]=-0,003141880	C[165]=-0,003306866	C[166]=-0,003467083	C[167]=-0,003622532
C[168]=-0,003771782	C[169]=-0,003914356	C[170]=-0,004048824	C[171]=-0,004174709
C[172]=-0,004290581	C[173]=-0,004395962	C[174]=-0,004489899	C[175]=-0,004570484
C[176]=-0,004638195	C[177]=-0,004691124	C[178]=-0,004728317	C[179]=-0,004748821
C[180]=-0,004752159	C[181]=-0,004737377	C[182]=-0,004703045	C[183]=-0,004649162
C[184]=-0,004573822	C[185]=-0,004477024	C[186]=-0,004357815	C[187]=-0,004215240
C[188]=-0,004049301	C[189]=-0,003858566	C[190]=-0,003643036	C[191]=-0,003401756
C[192]= 0,003134727	C[193]= 0,002841473	C[194]= 0,002521515	C[195]= 0,002174854
C[196]= 0,001800537	C[197]= 0,001399517	C[198]= 0,000971317	C[199]= 0,000515938
C[200]= 0,000033379	C[201]=-0,000475883	C[202]=-0,001011848	C[203]=-0,001573563
C[204]=-0,002161503	C[205]=-0,002774239	C[206]=-0,003411293	C[207]=-0,004072189
C[208]=-0,004756451	C[209]=-0,005462170	C[210]=-0,006189346	C[211]=-0,006937027
C[212]=-0,007703304	C[213]=-0,008487225	C[214]=-0,009287834	C[215]=-0,010103703
C[216]=-0,010933399	C[217]=-0,011775017	C[218]=-0,012627602	C[219]=-0,013489246
C[220]=-0,014358521	C[221]=-0,015233517	C[222]=-0,016112804	C[223]=-0,016994476
C[224]=-0,017876148	C[225]=-0,018756866	C[226]=-0,019634247	C[227]=-0,020506859
C[228]=-0,021372318	C[229]=-0,022228718	C[230]=-0,023074150	C[231]=-0,023907185
C[232]=-0,024725437	C[233]=-0,025527000	C[234]=-0,026310921	C[235]=-0,027073860
C[236]=-0,027815342	C[237]=-0,028532982	C[238]=-0,029224873	C[239]=-0,029890060
C[240]=-0,030526638	C[241]=-0,031132698	C[242]=-0,031706810	C[243]=-0,032248020
C[244]=-0,032754898	C[245]=-0,033225536	C[246]=-0,033659935	C[247]=-0,034055710
C[248]=-0,034412861	C[249]=-0,034730434	C[250]=-0,035007000	C[251]=-0,035242081
C[252]=-0,035435200	C[253]=-0,035586357	C[254]=-0,035694122	C[255]=-0,035758972
C[256]= 0,035780907	C[257]= 0,035758972	C[258]= 0,035694122	C[259]= 0,035586357
C[260]= 0,035435200	C[261]= 0,035242081	C[262]= 0,035007000	C[263]= 0,034730434
C[264]= 0,034412861	C[265]= 0,034055710	C[266]= 0,033659935	C[267]= 0,033225536
C[268]= 0,032754898	C[269]= 0,032248020	C[270]= 0,031706810	C[271]= 0,031132698
C[272]= 0,030526638	C[273]= 0,029890060	C[274]= 0,029224873	C[275]= 0,028532982
C[276]= 0,027815342	C[277]= 0,027073860	C[278]= 0,026310921	C[279]= 0,025527000
C[280]= 0,024725437	C[281]= 0,023907185	C[282]= 0,023074150	C[283]= 0,022228718
C[284]= 0,021372318	C[285]= 0,020506859	C[286]= 0,019634247	C[287]= 0,018756866
C[288]= 0,017876148	C[289]= 0,016994476	C[290]= 0,016112804	C[291]= 0,015233517

C[292]= 0,014358521	C[293]= 0,013489246	C[294]= 0,012627602	C[295]= 0,011775017
C[296]= 0,010933399	C[297]= 0,010103703	C[298]= 0,009287834	C[299]= 0,008487225
C[300]= 0,007703304	C[301]= 0,006937027	C[302]= 0,006189346	C[303]= 0,005462170
C[304]= 0,004756451	C[305]= 0,004072189	C[306]= 0,003411293	C[307]= 0,002774239
C[308]= 0,002161503	C[309]= 0,001573563	C[310]= 0,001011848	C[311]= 0,000475883
C[312]=-0,000033379	C[313]=-0,000515938	C[314]=-0,000971317	C[315]=-0,001399517
C[316]=-0,001800537	C[317]=-0,002174854	C[318]=-0,002521515	C[319]=-0,002841473
C[320]= 0,003134727	C[321]= 0,003401756	C[322]= 0,003643036	C[323]= 0,003858566
C[324]= 0,004049301	C[325]= 0,004215240	C[326]= 0,004357815	C[327]= 0,004477024
C[328]= 0,004573822	C[329]= 0,004649162	C[330]= 0,004703045	C[331]= 0,004737377
C[332]= 0,004752159	C[333]= 0,004748821	C[334]= 0,004728317	C[335]= 0,004691124
C[336]= 0,004638195	C[337]= 0,004570484	C[338]= 0,004489899	C[339]= 0,004395962
C[340]= 0,004290581	C[341]= 0,004174709	C[342]= 0,004048824	C[343]= 0,003914356
C[344]= 0,003771782	C[345]= 0,003622532	C[346]= 0,003467083	C[347]= 0,003306866
C[348]= 0,003141880	C[349]= 0,002974033	C[350]= 0,002803326	C[351]= 0,002630711
C[352]= 0,002457142	C[353]= 0,002283096	C[354]= 0,002110004	C[355]= 0,001937389
C[356]= 0,001766682	C[357]= 0,001597881	C[358]= 0,001432419	C[359]= 0,001269817
C[360]= 0,001111031	C[361]= 0,000956535	C[362]= 0,000806808	C[363]= 0,000661850
C[364]= 0,000522137	C[365]= 0,000388145	C[366]= 0,000259876	C[367]= 0,000137329
C[368]= 0,000021458	C[369]=-0,000088215	C[370]=-0,000191689	C[371]=-0,000288486
C[372]=-0,000378609	C[373]=-0,000462532	C[374]=-0,000539303	C[375]=-0,000610352
C[376]=-0,000674248	C[377]=-0,000731945	C[378]=-0,000783920	C[379]=-0,000829220
C[380]=-0,000868797	C[381]=-0,000902653	C[382]=-0,000930786	C[383]=-0,000953674
C[384]= 0,000971317	C[385]= 0,000983715	C[386]= 0,000991821	C[387]= 0,000995159
C[388]= 0,000994205	C[389]= 0,000989437	C[390]= 0,000980854	C[391]= 0,000968933
C[392]= 0,000954151	C[393]= 0,000935555	C[394]= 0,000915051	C[395]= 0,000891685
C[396]= 0,000866413	C[397]= 0,000838757	C[398]= 0,000809669	C[399]= 0,000779152
C[400]= 0,000747204	C[401]= 0,000714302	C[402]= 0,000680923	C[403]= 0,000646591
C[404]= 0,000611782	C[405]= 0,000576973	C[406]= 0,000542164	C[407]= 0,000507355
C[408]= 0,000472546	C[409]= 0,000438213	C[410]= 0,000404358	C[411]= 0,000371456
C[412]= 0,000339031	C[413]= 0,000307560	C[414]= 0,000277042	C[415]= 0,000247478
C[416]= 0,000218868	C[417]= 0,000191212	C[418]= 0,000165462	C[419]= 0,000140190
C[420]= 0,000116348	C[421]= 0,000093937	C[422]= 0,000072956	C[423]= 0,000052929
C[424]= 0,000034332	C[425]= 0,000017166	C[426]= 0,000000954	C[427]=-0,000013828
C[428]=-0,000027180	C[429]=-0,000039577	C[430]=-0,000050545	C[431]=-0,000060558
C[432]=-0,000069618	C[433]=-0,000077724	C[434]=-0,000084400	C[435]=-0,000090122
C[436]=-0,000095367	C[437]=-0,000099182	C[438]=-0,000102520	C[439]=-0,000105381
C[440]=-0,000106812	C[441]=-0,000108242	C[442]=-0,000108719	C[443]=-0,000108719
C[444]=-0,000108242	C[445]=-0,000107288	C[446]=-0,000105858	C[447]=-0,000103951
C[448]= 0,000101566	C[449]= 0,000099182	C[450]= 0,000096321	C[451]= 0,000093460
C[452]= 0,000090599	C[453]= 0,000087261	C[454]= 0,000083923	C[455]= 0,000080585
C[456]= 0,000076771	C[457]= 0,000073433	C[458]= 0,000070095	C[459]= 0,000066280
C[460]= 0,000062943	C[461]= 0,000059605	C[462]= 0,000055790	C[463]= 0,000052929
C[464]= 0,000049591	C[465]= 0,000046253	C[466]= 0,000043392	C[467]= 0,000040531
C[468]= 0,000037670	C[469]= 0,000034809	C[470]= 0,000032425	C[471]= 0,000030041
C[472]= 0,000027657	C[473]= 0,000025272	C[474]= 0,000023365	C[475]= 0,000021458
C[476]= 0,000019550	C[477]= 0,000018120	C[478]= 0,000016689	C[479]= 0,000014782
C[480]= 0,000013828	C[481]= 0,000012398	C[482]= 0,000011444	C[483]= 0,000010014
C[484]= 0,000009060	C[485]= 0,000008106	C[486]= 0,000007629	C[487]= 0,000006676
C[488]= 0,000006199	C[489]= 0,000005245	C[490]= 0,000004768	C[491]= 0,000004292
C[492]= 0,000003815	C[493]= 0,000003338	C[494]= 0,000003338	C[495]= 0,000002861
C[496]= 0,000002384	C[497]= 0,000002384	C[498]= 0,000001907	C[499]= 0,000001907
C[500]= 0,000001431	C[501]= 0,000001431	C[502]= 0,000000954	C[503]= 0,000000954
C[504]= 0,000000954	C[505]= 0,000000954	C[506]= 0,000000477	C[507]= 0,000000477
C[508]= 0,000000477	C[509]= 0,000000477	C[510]= 0,000000477	C[511]= 0,000000477

C.1.4 Psychoacoustic models

Two examples of psychoacoustic models are presented in annex D, "Psychoacoustic models".

C.1.5 Encoding

C.1.5.1 Layer I encoding

C.1.5.1.1 Introduction

This clause describes a possible Layer I encoding method. The description is made with reference to figure C.5, "Layer I, II Encoder Flow Chart".

C.1.5.1.2 Psychoacoustic model

The calculation of the psychoacoustic parameters can be done either with Psychoacoustic Model 1 described in clause D.1. or with Psychoacoustic Model 2 as described in D.2. The FFT shiftlength equals 384 samples. Either model provides the signal-to-mask ratio for every subband.

C.1.5.1.3 Analysis subband filtering

The subband analysis is described in the clause C.1.3, "Analysis subband filter".

C.1.5.1.4 Scalefactor calculation

The calculation of the scalefactor for each subband is performed every 12 subband samples. The maximum of the absolute value of these 12 samples is determined. The lowest value in table B.1, "Layer I, II Scalefactors", which is larger than this maximum is used as the scalefactor.

C.1.5.1.5 Coding of scalefactors

The index in the table B.1, "Layer I, II Scalefactors" is represented by 6 bits, MSB first. The scalefactor is transmitted only if a non-zero number of bits has been allocated to the subband.

C.1.5.1.6 Bit allocation

Before adjustment to a fixed bitrate, the number of bits that are available for coding the samples and the scalefactors must be determined. This number can be obtained by subtracting from the total number of bits available "cb", the number of bits needed for the header "bhdr" (32 bits), the CRC checkword "bcrc" if used (16 bits), the bit allocation "bbal", and the number of bits required for ancillary data "banc":

adb = cb - (bhdr + bcrc + bbal + banc)

The resulting number of bits can be used to code the subband samples and the scalefactors. The principle used in the allocation procedure is minimization of the total noise-to-mask ratio over the frame with the constraint that the number of bits used does not exceed the number of bits available for that frame. The possible number of bits allocated to one sample can be found in the table in 2.4.2.5 of the main part of the audio standard (Audio data, Layer I); the range is 0...15 bits, excluding an allocation of 1 bit.

The allocation procedure is an iterative procedure, where in each iteration step the number of levels of the subband samples of greatest benefit is increased.

First the mask-to-noise ratio "MNR" for each subband is calculated by subtracting from the signal-to-noise-ratio "SNR" the signal-to-mask-ratio "SMR":

MNR = SNR - SMR

The signal-to-noise-ratio can be found in the table C.2, "Layer I Signal-to-Noise Ratio". The signal-to-mask-ratio is the output of the psychoacoustic model.

Then zero bits are allocated to the samples and the scalefactors. The number of bits for the samples "bspl" and the number of bits for the scalefactors "bscf" are set to zero. Next an iterative procedure is started. Each iteration loop contains the following steps :

- Determination of the minimal MNR of all subbands.
- The accuracy of the quantization of the subband with the minimal MNR is increased by using the next higher number of bits.
- The new MNR of this subband is calculated.
- bspl is updated according to the additional number of bits required. If a non-zero number of bits is assigned to a subband for the first time, bscf has to be incremented by 6 bits. Then adb is calculated again using the formula:

  adb = cb - (bhdr + bcrc + bbal + bscf + bspl + banc)

The iterative procedure is repeated as long as adb is not less than any possible increase of bspl and bscf within one loop.

C.1.5.1.7 Quantization and encoding of subband samples

A linear quantizer with a symmetric zero representation is used to quantize the subband samples. This representation prevents small value changes around zero from quantizing to different levels. Each of the subband samples is normalized by dividing its value by the scalefactor to obtain X, and quantized using the following formula :

- Calculate AX+B
- Take the N most significant bits.
- Invert the MSB.

A and B can be found in table C.3, "Layer I Quantization Coefficients". N represents the necessary number of bits to encode the number of steps. The inversion of the most significant bit (MSB) is done in order to avoid the all '1' representation of the code, because the all '1' code is used for the synchronization word.

C.1.5.1.8 Coding of bit allocation

The 4-bit code for the allocation is given in 2.4.2.5, "Audio data Layer I", of the main part of the audio standard.

C.1.5.1.9 Ancillary data

The Audio standard provides a number of bits for the inclusion and transmission of variable length ancillary data with the audio bitstream. The ancillary data will reduce the number of bits available for audio, which may result in a degradation of audio quality.

The presence of a bit pattern in the ancillary data matching the syncword may hamper synchronization. This problem is more likely to occur when the free format is used.

C.1.5.1.10 Formatting

The encoded subband information is transferred in frames (See also 2.4.1.2, 2.4.1.3, 2.4.1.5 and 2.4.1.8). The number of slots in a frame varies with the sample frequency (Fs) and bitrate. Each frame contains information on 384 samples of the original input signal, so the frame rate is Fs/384.

Fs (kHz)	Frame size (ms)
48	8
44,1	8,7074...
32	12

A frame may carry audio information from one or two channels.

The length of a slot in Layer I is 32 bits. The number of slots in a frame can be computed by this formula:

$$\text{Number of slots/frame (N)} = \frac{\text{bitrate}}{\text{Fs}} * 12$$

If this does not give an integer number the result is truncated and 'padding' is required. This means that the number of slots may vary between N and N + 1.

An overview of the Layer I format is given in figure C.2:

Frame

HEADER	ALLOC	SCALEFACTORS	SAMPLES	ANC

Figure C.2 -- Layer I Format

Table C.2 -- Layer I Signal-to-Noise Ratios

No. of steps	SNR (dB)
0	0,00
3	7,00
7	16,00
15	25,28
31	31,59
63	37,75
127	43,84
255	49,89
511	55,93
1 023	61,96
2 047	67,98
4 095	74,01
8 191	80,03
16 383	86,05
32 767	92,01

Table -- C.3 Layer I Quantization Coefficients

No. of steps	A	B
3	0,750000000	-0,250000000
7	0,875000000	-0,125000000
15	0,937500000	-0,062500000
31	0,968750000	-0,031250000
63	0,984375000	-0,015625000
127	0,992187500	-0,007812500
255	0,996093750	-0,003906250
511	0,998046875	-0,001953125
1 023	0,999023438	-0,000976563
2 047	0,999511719	-0,000488281
4 095	0,999755859	-0,000244141
8 191	0,999877930	-0,000122070
16 383	0,999938965	-0,000061035
32 767	0,999969482	-0,000030518

C.1.5.2 Layer II encoding

C.1.5.2.1 Introduction

This clause describes a possible Layer II encoding method. The description is made according to figure C.5, "Layer I, II encoder flow chart".

C.1.5.2.2 Psychoacoustic model

The calculation of the psychoacoustic parameters can be done either with Psychoacoustic Model 1 described in clause D.1. or with Psychoacoustic Model 2 described in clause D.2. If Psychoacoustic Model 1 is used to calculate the psychoacoutic parameters, the FFT shiftlength is 1152 samples. If Psychoacoustic Model 2 is used, the calculation is performed twice with a shiftlength of 576 samples and the largest of each pair of signal to mask ratios is used. Either model provides the signal-to-mask ratio for every subband.

C.1.5.2.3 Analysis subband filter

The analysis subband filter is described in clause C.1.3, "Analysis subband filter".

C.1.5.2.4 Scalefactor calculation

The calculation of the scalefactor for each subband is performed every 12 subband samples. The maximum of the absolute value of these 12 samples is determined. The lowest value in table B.1, "Layer I, II Scalefactors", which is larger than this maximum is used as the scalefactor.

C.1.5.2.5 Coding of scalefactors

A frame corresponds to 36 subband samples and therefore contains three scalefactors per subband. Define 'scf' as the index in table B.1, "Layer I, II Scalefactors". First, the two differences $dscf_1$ and $dscf_2$ of the successive scalefactor indices scf_1, scf_2 and scf_3 are calculated:

$$dscf_1 = scf_1 - scf_2$$
$$dscf_2 = scf_2 - scf_3$$

The class of each of the differences is determined as follows:

class.	dscf
1	dscf <= -3
2	-3 < dscf < 0
3	dscf = 0
4	0 < dscf < 3
5	dscf >= 3

The pair of classes of differences indicate the entry point in table C.4, "Layer II Scalefactors Transmission Patterns". The column labelled "scalefactor used in encoder" gives the three scalefactors which are actually used. "1", "2" and "3" mean respectively the first, second and third scalefactor within a frame, "4" means the maximum of the three scalefactors. If, after this adjusting of scalefactors two or three are the same, not all scalefactors need to be transmitted for a certain subband within one frame. Only the scalefactors indicated in the "transmission pattern" column are transmitted. The information describing the number and the position of the scalefactors in each subband is called "scalefactor selection information".

C.1.5.2.6 Coding of scalefactor selection information

The "scalefactor selection information" (scfsi) is coded by a two bit word, which is also to be found in table C.4, "Layer II scalefactor transmission patterns". Only the scfsi for the subbands which will get a nonzero bit allocation are transmitted.

C.1.5.2.7 Bit allocation

Before adjustment to a fixed bitrate, the number of bits ,"adb", that are available for coding the samples and the scalefactors must be determined. This number can be obtained by subtracting from the total number of