TITLE PAGE PROVIDED BY ISO

CD 11172-3
CODING OF MOVING PICTURES AND ASSOCIATED AUDIO FOR DIGITAL STORAGE MEDIA AT UP TO ABOUT 1.5 MBIT/s
Part 3 AUDIO
CONTENTS

FOREWORD

This standard is a committee draft that was submitted for approval to ISO-IEC/JTC1 SC29 on 22 November 1991. It was prepared by SC29/WG11, also known as MPEG (Moving Pictures Expert Group). MPEG was formed in 1988 to establish a standard for the coded representation of moving pictures and associated audio stored on digital storage media.

This standard is published in four parts. Part 1 - systems - specifies the system coding layer of the standard. It defines a multiplexed structure for combining audio and video data and means of representing the timing information needed to replay synchronized sequences in real-time. Part 2 - video - specifies the coded representation of video data and the decoding process required to reconstruct pictures. Part 3 - audio - specifies the coded representation of audio data. Part 4 - conformance testing - is still in preparation. It will specify the procedures for determining the characteristics of coded bit streams and for testing compliance with the requirements stated in Parts 1, 2 and 3.

In Part 1 of this standard all annexes are informative and contain no normative requirements.

In Part 2 of this standard 2-Annex A, 2-Annex B and 2-Annex C contain normative requirements and are an integral part of this standard. 2-Annex D and 2-Annex E are informative and contain no normative requirements.

In Part 3 of this standard 3-Annex A and 3-Annex B contain normative requirements and are an integral part of this standard. All other annexes are informative and contain no normative requirements.

INTRODUCTION

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.

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.

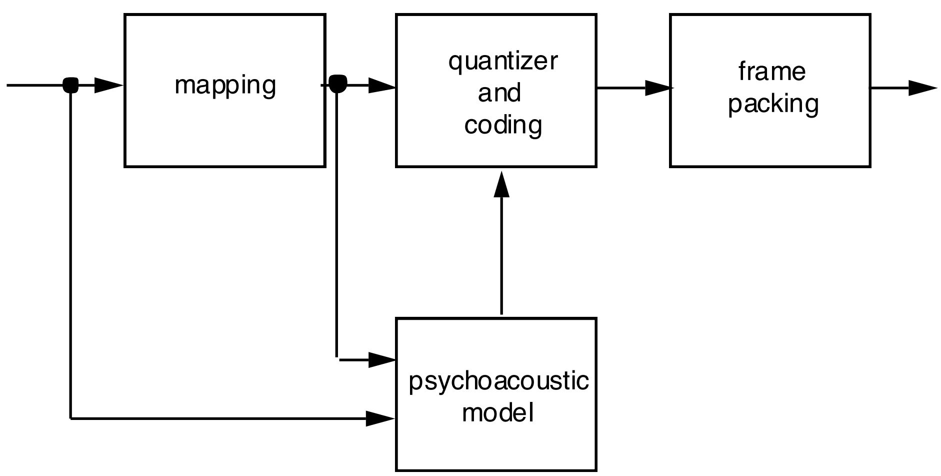


Figure I-1 Sketch of a basic encoder

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, 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.

Layers

Depending on the application, different layers of the coding system with increasing encoder complexity and performance can be used. An ISO MPEG 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.

Layer II:
This layer provides additional coding of bit allocation, scalefactors and samples. Different framing is used.

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 .

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

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.

Decoding

The decoder, subject to the application-dependent parameters of clause 2.4.1, accepts the compressed audio bitstream in the syntax defined in clause 2.4.2, decodes the data elements according to clause 2.4.3, and uses the information to produce digital audio output according to clause 2.4.4.

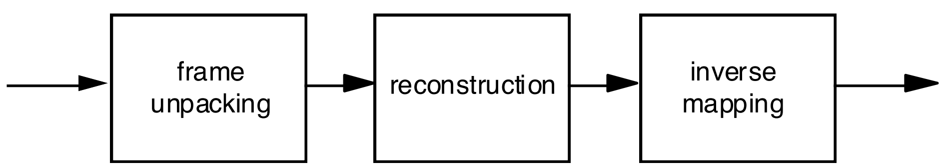


Figure I-2 Sketch of the basic structure of the decoder

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 clause 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.

1. GENERAL NORMATIVE ELEMENTS

1.1 Scope

This standard 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 standard is intended for application to digital storage media providing a total continuous transfer rate of about 1.5 Mbit/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 11172 multiplex stream defined in Part1 of this standard. This standard is intended for sampling rates of 32 kHz, 44.1 kHz, and 48 kHz.

1.2 References

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard 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.

Recommendations and reports of the CCIR, 1990
XVIIth Plenary Assembly, Dusseldorf, 1990
Volume XI - Part 1
Broadcasting Service (Television)
Rec. 601-1 "Encoding parameters of digital television for studios".

Volume X
Rec. 953 "Encoding parameters of digital audio".

IEEE Draft Standard "Specification for the implementation of 8x 8 inverse discrete cosine transform".
P1180/D2, July 18,1990

2. TECHNICAL NORMATIVE ELEMENTS

2.1 Definitions

For the purposes of this International Standard, the following definitions apply.

AC coefficient: Any DCT coefficient for which the frequency in one or both dimensions is non-zero.

access unit: 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.

Adaptive segmentation: A subdivision of the digital representation of an audio signal in variable segments of time.

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

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

Alias: Mirrored signal component resulting from sub-Nyquist sampling.

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

Audio Access Unit: 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".

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

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

Bark: Unit of critical band rate.

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

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

Block companding: Normalizing of the digital representation of an audio signal within a certain time period.

block: An 8-row by 8-column orthogonal block of pels.

Bound: The lowest subband in which intensity stereo coding is used.

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.

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

chrominance (component): A matrix, block or sample of pels 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.

coded audio bitstream: A coded representation of an audio signal as specified in this International Standard.

coded video bitstream: A coded representation of a series of one or more pictures as specified in this International Standard.

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

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

coding parameters: The set of user-definable parameters that characterise a coded video bitstream. Bit-streams are characterised by coding parameters. Decoders are characterised by the bitstreams that they are capable of decoding.

component: A matrix, block or sample of pel data from one of the three matrices (luminance and two chrominance) that make up a picture.

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

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

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

Constrained Parameters: In the case of the video specification, the values of the set of coding parameters defined in Part 2 Clause 2.4.4.4.

constrained system parameter stream (CSPS): An ISO 11172 multiplexed stream for which the constraints defined in Part 1 Clause 2.4.6 apply.

CRC: Cyclic redundancy code.

Critical Band Rate: Psychoacoustic measure in the spectral domain which corresponds to the frequency selectivity of the human ear.

Critical Band: Part of the spectral domain which corresponds to a width of one Bark.

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

DC-coefficient: The DCT coefficient for which the frequency is zero in both dimensions.

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

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

decoded stream: The decoded reconstruction of a compressed bit stream.

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

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

decoder: An embodiment of a decoding process.

decoding process: The process defined in this International Standard that reads an input coded bitstream and outputs decoded pictures or audio samples.

decoding time-stamp; DTS: 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.

Dequantization [Audio]: Decoding of coded subband samples in order to recover the original quantized values.

dequantization: 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.

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

discrete cosine transform; DCT: 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 2-Annex A of Part 2.

display order: 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.

editing: The process by which one or more compressed bitstreams are manipulated to produce a new compressed bitstream. Conforming editted bitstreams must meet the requirements defined in this International Standard.

elementary stream: A generic term for one of the coded video, coded audio or other coded bit streams.

encoder: An embodiment of an encoding process.

encoding process: A process, not specified in this International Standard, that reads a stream of input pictures or audio samples and produces a valid coded bitstream as defined in this International Standard.

Entropy coding: Variable length noiseless coding of the digital representation of a signal to reduce redundancy.

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

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

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

Fixed segmentation: A subdivision of the digital representation of an audio signal in to fixed segments of time.

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.

forced updating: 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.

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

Frame [audio]: A part of the audio signal that corresponds to a fixed number of audio PCM samples.

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

Granules [Layer II]: 96 subband samples, 3 consecutive subband samples for all 32 subbands that are considered together before quantisation

Granules [Layer III]: 576 frequency lines that carry their own side information.

group of pictures: A series of one or more pictures intended to assist random access. The group of pictures is one of the layers in the coding syntax defined in Part 2 of this International Standard.

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

Huffman coding: A specific method for entropy coding.

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

IMDCT: Inverse Modified Discrete Cosine Transform.

Intensity stereo: 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.

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

intra coding: Compression coding of a block or picture that uses information only from that block or picture.

intra-coded picture; I-picture: A picture coded using information only from itself.

ISO 11172 (multiplexed) stream: A bitstream composed of zero or more elementary streams combined in the manner defined in Part 1 of this International Standard.

Joint stereo coding: Any method that exploits stereophonic irrelevance or stereophonic redundancy.

Joint stereo mode: A mode of the audio coding algorithm using joint stereo coding.

layer [audio]: One of the levels in the coding hierarchy of the audio system defined in this International Standard.

layer [video and systems]: One of the levels in the data hierarchy of the video and system specifications defined in Parts 1 and 2 of this International Standard.

luminance (component): A matrix, block or sample of pels 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.

macroblock: 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 and other data elements defined in the macroblock layer of the syntax defined in Part 2 of this International Standard. The usage is clear from the context.

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

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

Masking: property of the human auditory system by which an audio signal cannot be perceived in the presence of another audio signal .

MDCT: Modified Discrete Cosine Transform.

motion compensation: 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 frames containing previously decoded pels that are used to form the prediction.

motion vector estimation: The process of estimating motion vectors during the encoding process.

motion vector: 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.

MS stereo: 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.

non-intra coding: Coding of a block or picture that uses information both from itself and from blocks and pictures occurring at other times.

Non-tonal component: A noise-like component of an audio signal.

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

pack: A pack consists of a pack header followed by one or more packets. It is a layer in the system coding syntax described in Part 1 of this standard.

packet data: Contiguous bytes of data from an elementary stream present in a packet.

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

packet: 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 Part 1 of this International Standard.

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

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

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

pel: An 8-bit sample of luminance or chrominance data.

picture period: The reciprocal of the picture rate.

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

picture: Source or reconstructed image data. A 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 Part 2 of this International Standard. NOTE: the term "picture" is always used in this standard in preference to the terms field or frame.

Polyphase filterbank: A set of equal bandwidth filters with special phase interrelationships, allowing for an efficient implementation of the filterbank.

prediction: The use of predictor to provide an estimate of the pel or data element currently being decoded.

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

predictor: A linear combination of previously decoded pels or data elements.

presentation time-stamp; PTS: 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.

presentation unit: A decoded audio access unit or a decoded picture.

Psychoacoustic model: A mathematical model of the masking behaviour of the human auditory system.

quantization matrix: A set of sixty-four 8-bit scaling values used by the dequantizer.

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

quantizer scale factor: A data element represented in the bitstream and used by the decoding process to scale the dequantization.

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

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

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

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 defined extensions.

reverse play: The process of displaying the picture sequence in the reverse of display order.

Scalefactor band: A set of frequency lines in Layer III which are scaled by one scalefactor.

Scalefactor index: A numerical code for a scalefactor.

Scalefactor: Factor by which a set of values is scaled before quantization.

sequence header: A block of data in the coded bitstream containing the coded representation of a number of data elements. It is one of the layers of the coding syntax defined in Part 2 of this International Standard.

Side information: Information in the bitstream necessary for controlling the decoder.

skipped macroblock: A macroblock for which no data is stored.

slice: A series of macroblocks. It is one of the layers of the coding syntax defined in Part 2 of this International Standard.

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.

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

Spreading function: A function that describes the frequency spread of masking.

start codes: 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.

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

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.

Subband [audio]: Subdivision of the audio frequency band.

Subband filterbank: A set of band filters covering the entire audio frequency range. In Part 3 of this International Standard the subband filterbank is a polyphase filterbank.

Syncword: A 12-bit code embedded in the audio bitstream that identifies the start of a frame.

Synthesis filterbank: Filterbank in the decoder that reconstructs a PCM audio signal from subband samples.

system header: The system header is a data structure defined in Part 1 of this International Standard that carries information summarising the system characteristics of the ISO 11172 multiplexed stream.

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

time-stamp: A term that indicates the time of an event.

Tonal component: A sinusoid-like component of an audio signal.

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

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.

video buffering verifier; VBV: 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.

video sequence: A series of one or more groups of pictures.

zig-zag scanning order: 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 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 two's-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 toward -8.

% Modulus operator. Defined only for positive numbers.

Sign( ) Sign(x)

$$\mathrm{Sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x == 0 \\ -1 & x < 0 \end{cases}$$

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

log10 Logarithm to base ten

loge Logarithm to base e

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

& 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 the standard. 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.

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 in Annex B.

ch channel.

gr granule of three sub-band samples in audio layer 2, twelve sub-band samples in audio layer 3.

rpchof remainder polynomial coefficients, highest order first.

sb sub-band.

scfsi scale-factor selector information.

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

2.2.7 Constants

pi 3.14159265359...
e 2.71828182846...

2.3 Method of Describing Bit Stream Syntax

The bit stream retrieved by the decoder is described in clause 2.4.2. Each data item in the bit stream 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 bit stream 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 clause 2.4.3. The following constructs are used to express the conditions when data elements are present, and are in normal type:

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

for (i = 0; i < n; i++) { The group of data elements occurs n times. Conditional constructs
data_element 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 {} are omitted when only one data element follows.

data_element [n]data_element [n] is the n+1th element of an 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 clause 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, and to identify errors, erasures or insertions while decoding. The methods to identify these situations, and the actions to be taken, are not standardized.

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 bit stream is the first bit in a byte.

Definition of nextbits function

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

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.

```
next_start_code() {
   while ( !bytealigned() )
      zero_bit     1          "0"
   while ( nextbits() != '0000 0000 0000 0000 0000 0001' )
      zero_byte   8          "00000000"
   }
```

2.4 Requirements

2.4.1 Specification of the Coded Audio Bitstream Syntax

2.4.1.1 Audio Sequence

```
audio sequence()
 {
  while (true)
  {
      frame()
  }
 }
```

2.4.1.2 Audio Frame

```
frame()
{
   header()
   error_check()
   audio_data()
   ancillary_data()
}
```

2.4.1.3 Header

```
header()
{
   syncword                                             12   bits       bslbf
   ID                                                   1    bit        bslbf
   layer                                                2    bits       bslbf
   protection_bit                                       1    bit        bslbf
   bitrate_index                                        4    bits       bslbf
   sampling_frequency                                   2    bits       bslbf
   padding_bit                                          1    bit        bslbf
   private_bit                                          1    bit        bslbf
   mode                                                 2    bits       bslbf
   mode_extension                                       2    bits       bslbf
   copyright                                            1    bit        bslbf
   original/home                                        1    bit        bslbf
   emphasis                                             2    bits       bslbf
}
```

2.4.1.4 Error check

```
error_check()
{
 if  (protection_bit==0)
    crc_check                                         16  bits       rpchof
}
```

2.4.1.5 Audio data, Layer I

```
audio_data()
{
 if (mode==single_channel)
 {
  for (sb=0; sb<32; sb++)
   allocation[sb]                                    4    bits       uimsbf
  for (sb=0; sb<32; sb++)
   if (allocation[sb]!=0)
    scalefactor[sb]                                  6    bits       uimsbf
  for (s=0; s<12; s++)
   for (sb=0; sb<32; sb++)
    if (allocation[sb]!=0)
     sample[sb][s]                                   2..15bits       uimsbf
 }
 if (mode==stereo) || (mode==dual_channel)
 {
  for (sb=0; sb<32; sb++)
   for (ch=0; ch<2; ch++)
    allocation[ch][sb]                               4    bits       bsmsbf
  for (sb=0; sb<32; sb++)
   for (ch=0; ch<2; ch++)
    if (allocation[ch][sb]!=0)
     scalefactor[ch][sb]                             6    bits       uimsbf
  for (s=0; s<12; s++)
   for (sb=0; sb<32; sb++)
    for (ch=0; ch<2; ch++)
     if (allocation[ch][sb]!=0)
      sample[ch][sb][s]                              2..15bits       uimsbf
  }
  if (mode==intensity_stereo)
  {
   for (sb=0; sb<bound; sb++)
    for (ch=0; ch<2; ch++)
     allocation[ch][sb]                              4    bits       uimsbf
   for (sb=bound; sb<32; sb++)
    allocation[sb]                                   4    bits       uimsbf
   for (sb=0; sb<bound; sb++)
    for (ch=0; ch<2; ch++)
     if (allocation[ch][sb]!=0)
      scalefactor[ch][sb]                            6    bits       uimsbf
   for (sb=bound; sb<32; sb++)
    for (ch=0; ch<2; ch++)
     if (allocation[sb]!=0)
      scalefactor[ch][sb]                            6    bits       uimsbf
      for (s=0; s<12; s++)
   {
    for (sb=0; sb<bound; sb++)
     for (ch=0; ch<2; ch++)
      if (allocation[ch][sb]!=0)
```

```
      sample[ch][sb][s]                               2..15 bits      uimsbf
    for (sb=bound; sb<32; sb++)
     if (allocation[sb]!=0)
      sample[sb][s]                                   2..15 bits      uimsbf
   }
  }
 }
```

2.4.1.6 Audio data, Layer II

```
audio_data()
{
 if (mode==single_channel)
 {
  for (sb=0; sb<sblimit; sb++)
   allocation[sb]                                     2..4  bits      uimsbf
  for (sb=0; sb<sblimit; sb++)
   if (allocation[sb]!=0)
    scfsi[sb]                                         2     bits      bslbf
  for (sb=0; sb<sblimit; sb++)
   if (allocation[sb]!=0)
   {
    if (scfsi[sb]==0)
    { scalefactor[sb][0]                              6     bits      uimsbf
      scalefactor[sb][1]                              6     bits      uimsbf
      scalefactor[sb][2] }                            6     bits      uimsbf
    if (scfsi[sb]==1) || (scfsi[sb]==3)
    { scalefactor[sb][0]                              6     bits      uimsbf
      scalefactor[sb][2] }                            6     bits      uimsbf
    if (scfsi[sb]==2)
      scalefactor[sb][0]                              6     bits      uimsbf
   }
   for (gr=0; gr<12; gr++)
    for (sb=0; sb<sblimit; sb++)
     if (allocation[sb]!=0)
     {
      if (grouping[sb])
       samplecode[sb][gr]                             5..10 bits      uimsbf
      else for (s=0; s<3; s++)
       sample[sb][3*gr+s]                             2..16 bits      uimsbf
     }
 }

if (mode==stereo) || (mode==dual_channel)
 {
  for (sb=0; sb<sblimit; sb++)
   for (ch=0; ch<2; ch++)
     allocation[ch][sb]                               2..4  bits      uimsbf
  for (sb=0; sb<sblimit; sb++)
   for (ch=0; ch<2; ch++)
    if (allocation[ch][sb]!=0)
     scfsi[ch][sb]                                    2     bits      bslbf
    for (sb=0; sb<sblimit; sb++)
   for (ch=0; ch<2; ch++)
    if (allocation[ch][sb]!=0)
    {
     if (scfsi[ch][sb]==0)
     { scalefactor[ch][sb][0]                         6     bits      uimsbf
       scalefactor[ch][sb][1]                         6     bits      uimsbf
```

```
      scalefactor[ch][sb][2] }                 6    bits       uimsbf
     if (scfsi[ch][sb]==1) || (scfsi[ch][sb]==3)
     { scalefactor[ch][sb][0]                  6    bits       uimsbf
       scalefactor[ch][sb][2] }                6    bits       uimsbf
     if (scfsi[ch][sb]==2)
      scalefactor[ch][sb][0]                   6    bits       uimsbf
    }
   for (gr=0; gr<12; gr++)
    for (sb=0; sb<sblimit; sb++)
     for (ch=0; ch<2; ch++)
     if (allocation[ch][sb]!=0)
     {
      if (grouping[ch][sb])
        samplecode[ch][sb][gr]                 5..10bits       uimsbf
      else for (s=0; s<3; s++)
        sample[ch][sb][3*gr+s]                 2..16bits       uimsbf
     }
  }

if (mode==intensity_stereo)
  {
   for (sb=0; sb<bound; sb++)
    for (ch=0; ch<2; ch++)
     allocation[ch][sb]                        2..4 bits       uimsbf
   for (sb=bound; sb<sblimit; sb++)
    allocation[sb]                             2..4 bits       uimsbf
   for (sb=0; sb<bound; sb++)
    for (ch=0; ch<2; ch++)
    if (allocation[ch][sb]!=0)
      scfsi[ch][sb]                            2    bits       bslbf
   for (sb=bound; sb<sblimit; sb++)
    for (ch=0; ch<2; ch++)
    if (allocation[sb]!=0)
      scfsi[ch][sb]                            2    bits       bslbf
   for (sb=0; sb<bound; sb++)
    for (ch=0; ch<2; ch++)
    if (allocation[ch][sb]!=0)
     {
      if (scfsi[ch][sb]==0)
      { scalefactor[ch][sb][0]                 6    bits       uimsbf
        scalefactor[ch][sb][1]                 6    bits       uimsbf
        scalefactor[ch][sb][2] }               6    bits       uimsbf
      if (scfsi[ch][sb]==1) || (scfsi[ch][sb]==3)
      { scalefactor[ch][sb][0]                 6    bits       uimsbf
        scalefactor[ch][sb][2] }               6    bits       uimsbf
      if (scfsi[ch][sb]==2)
        scalefactor[ch][sb][0]                 6    bits       uimsbf
     }
   for (sb=bound; sb<sblimit; sb++)
    for (ch=0; ch<2; ch++)
    if (allocation[sb]!=0)
     {
      if (scfsi[ch][sb]==0)
      { scalefactor[ch][sb][0]                 6    bits       uimsbf
        scalefactor[ch][sb][1]                 6    bits       uimsbf
        scalefactor[ch][sb][2] }               6    bits       uimsbf
      if (scfsi[ch][sb]==1) || (scfsi[ch][sb]==3)
      { scalefactor[ch][sb][0]                 6    bits       uimsbf
```

```
        scalefactor[ch][sb][2] }                     6    bits       uimsbf
      if (scfsi[ch][sb]==2)
        scalefactor[ch][sb][0]                       6    bits       uimsbf
     }
   for (gr=0; gr<12; gr++)
    {
     for (sb=0; sb<bound; sb++)
      for (ch=0; ch<2; ch++)
      if (allocation[ch][sb]!=0)
      {
       if (grouping[ch][sb])
         samplecode[ch][sb][gr]                      5..10bits       uimsbf
       else for (s=0; s<3; s++)
         sample[ch][sb][3*gr+s]                      2..16bits       uimsbf
      }
     for (sb=bound; sb<sblimit; sb++)
      if (allocation[sb]!=0)
      {
       if (grouping[sb])
         samplecode[sb][gr]                          5..10bits       uimsbf
       else for (s=0; s<3; s++)
         sample[sb][3*gr+s]                          2..16bits       uimsbf
      }
    }
  }
}
```

2.4.1.7 Audio data, Layer III

```
audio_data()
{
 if (mode == single_channel)
 {
  main_data_end                                          9    bits   uimsbf
  private_bits                                           5    bits   bslbf
  for (scfsi_band=0; scfsi_band<4; scfsi_band++)
   scfsi[scfsi_band]                                     1    bits   bslbf
  for (gr=0; gr<2; gr++)
   {
    part2_3_length[gr]                                   12   bits   uimsbf
    big_values[gr]                                       9    bits   uimsbf
    global_gain[gr]                                      8    bits   uimsbf
    scalefac_compress[gr]                                4    bits   bslbf
    blocksplit_flag[gr]                                  1    bit    bslbf
    if (blocksplit_flag[gr])
      {
        block_type[gr]                                   2    bits   bslbf
        switch_point[gr]                                 1    bits   uimsbf
        for (region=0; region<2; region++)
          table_select[region][gr]                       5    bits   bslbf
        for (window=0; window<3; window++)
        subblock_gain[window][gr]                        3    bits   uimsbf
      }
    else
      {
        for (region=0; region<3; region++)
          table_select[region][gr]                       5    bits   bslbf
```

```
        region_address1[gr]                                 4     bits    bslbf
        region_address2[gr]                                 3     bits    bslbf
      }
    preflag[gr]                                             1     bit     bslbf
    scalefac_scale[gr]                                      1     bit     bslbf
    count1table_select[gr]                                  1     bit     bslbf
   }

    /***
    The main_data follows.  It does not follow the above side information in the
    bitstream.  The main_data ends at a location in the main_data bitstream
    preceding the frame header of the following frame at an offset given by the
    value of main_data_end (see definition of main_data_end and 3-Annex Fig.3-
    A.7.1)
    ***/
  for (gr=0; gr<2; gr++)
    if (blocksplit_flag[gr] == 1 && block_type[gr] == 2)
    {
     for (cb=0; cb<switch_point_l[gr]; cb++)
      if (scfsi[cb]==0) || (gr==0)
       scalefac[cb][gr]                                     0..4  bits    uimsbf
     for (cb=switch_point_s[gr]; cb<cblimit_short; cb++)
      for (window=0; window<3; window++)
       if (scfsi[cb]==0) || (gr==0)
        scalefac[cb][window][gr]                            0..4  bits    uimsbf
    }
    else
     for (cb=0; cb<cblimit; cb++)
      if (scfsi[cb]==0) || (gr==0)
       scalefac[cb][gr]                                     0..4  bits    uimsbf
    Huffmancodebits                                             (part2_3_length-
part2_length)                                               bits  bslbf
    while (position != main_data_end)
    {
       ancillary_bit                                        1     bit     bslbf
    }

 }

if (mode==stereo) || (mode==dual_channel) || (mode==ms_stereo)
 {
  main_data_end                                                  9 bits  uimsbf
  private_bits                                                   3 bits  bslbf
  for (ch=0; ch<2; ch++)
   for (scfsi_band=0; scfsi_band<4; scfsi_band++)
    scfsi[scfsi_band][ch]                                        1 bits  bslbf
  for (gr=0; gr<2; gr++)
   for (ch=0; ch<2; ch++}
    part2_3_length[gr][ch]                                  12    bits    uimsbf
    big_values[gr][ch]                                      9     bits    uimsbf
    global_gain[gr][ch]                                     8     bits    uimsbf
    scalefac_compress[gr][ch]                               4     bits    bslbf
    blocksplit_flag[gr][ch]                                 1     bit     bslbf
    if (blocksplit_flag[gr][ch])
      {
       block_type[gr][ch]                                   2     bits    bslbf
       switch_point[gr][ch]                                 1     bits    uimsbf
       for (region=0; region<2; region++)
```

```
        table_select[region][gr][ch]                            5     bits   bslbf
       for (window=0; window<3; window++)
        subblock_gain[window][gr][ch]                           3     bits   uimsbf
      }
    else
      {
       for (region=0; region<3; region++)
        table_select[region][gr][ch]                            5     bits   bslbf
       region_address1[gr][ch]                                  4     bits   bslbf
       region_address2[gr][ch]                                  3     bits   bslbf
      }
    preflag[gr][ch]                                             1     bit    bslbf
    scalefac_scale[gr][ch]                                      1     bit    bslbf
    count1table_select[gr][ch]                                  1     bit    bslbf

    /***
    The main_data follows. It does not follow the above side information in the
    bitstream. The main_data ends at a location in the main_data bitstream
    preceding the frame header of the following frame at an offset given by the
    value of main_data_end.
    ***/
 for (gr=0; gr<2; gr++)
   for (ch=0; ch<2; ch++) {
    if (blocksplit_flag[gr][ch] == 1 && block_type[gr][ch] == 2)
    {
     for (cb=0; cb<switch_point_l[gr][ch]; cb++)
      if (scfsi[cb]==0) || (gr==0)
       scalefac[cb][gr][ch]                                     0..4  bits   uimsbf
     for (cb=switch_point_s[gr][ch]; cb<cblimit_short; cb++)
      for (window=0; window<3; window++)
       if (scfsi[cb]==0) || (gr==0)
        scalefac[cb][window][gr][ch]                            0..4  bits   uimsbf
    }
    else
     for (cb=0; cb<cblimit; cb++)
      if (scfsi[cb]==0) || (gr==0)
       scalefac[cb][gr][ch]                                     0..4  bits   uimsbf
    Huffmancodebits                                             (part2_3_length-
part2_length)                                                   bits  bslbf
    while (position != main_data_end)
    {
       ancillary_bit                                            1     bit    bslbf
    }

   }
 }
}
```

2.4.1.8 Ancillary data

```
if(layer == 1 || layer == 2)
{
  ancillary_data()
  {
   while (nextbits() != syncword)
```

```
   {
   ancillary_bit                                          1    bit    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 1152 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 side and main information. In Layer III it contains information for 1152 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 MPEG audio, '0' is reserved.
Layer - 2 bits to indicate which layer is used, according to the following table.

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

To change the layer, a reset of the decoder is 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.
bit_rate_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 bit_rate_index is an index to a table, which is different for the different Layers.
The bit_rate_index indicates the total bitrate irrespective of the mode (stereo, joint_stereo, dual_channel, single_channel).
For Layer II, not all combinations of total bitrate and mode are allowed. See 3-Annex B, Table 3-B.2 "LAYER II BIT ALLOCATION TABLES".

	bitrate		
bit_rate_index	Layer I	Layer II	Layer III
'0000'	free format	free format	free format
'0001'	32 kbit/s	32 kbit/s	32 kbit/s
'0010'	64 kbit/s	48 kbit/s	40 kbit/s
'0011'	96 kbit/s	56 kbit/s	48 kbit/s
'0100'	128 kbit/s	64 kbit/s	56 kbit/s

'0101'	160	kbit/s	80	kbit/s	64	kbit/s
'0110'	192	kbit/s	96	kbit/s	80	kbit/s
'0111'	224	kbit/s	112	kbit/s	96	kbit/s
'1000'	256	kbit/s	128	kbit/s	112	kbit/s
'1001'	288	kbit/s	160	kbit/s	128	kbit/s
'1010'	320	kbit/s	192	kbit/s	160	kbit/s
'1011'	352	kbit/s	224	kbit/s	192	kbit/s
'1100'	384	kbit/s	256	kbit/s	224	kbit/s
'1101'	416	kbit/s	320	kbit/s	256	kbit/s
'1110'	448	kbit/s	384	kbit/s	320	kbit/s

In order to provide the smallest possible delay and complexity, the decoder is not required to support a continuously variable bitrate when in LayerI or II. Layer III supports variable bitrate by switching the bit_rate_index. However, in free format, fixed bitrate is required.

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

'00'	44.1	kHz
'01'	48	kHz
'10'	32	kHz
'11'	reserved	

A reset of the decoder is 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 only necessary with a sampling frequency of 44.1kHz.

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

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

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.

'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.

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

copyright - if this bit equals '0' there is no copyright on the coded bitstream, '1' means copyright protected.

original/home - 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.

'00'	no emphasis
'01'	50/15 microsec. emphasis
'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[sb]` - indicates the number of bits used to code the samples in subband sb. Valid for single_channel subbands and for subbands in intensity_stereo mode. In the latter case the allocation is valid for both channels.

CODE	BITS
'0000'	0
'0001'	2
'0010'	3
'0011'	4
'0100'	5
'0101'	6
'0110'	7
'0111'	8
'1000'	9
'1001'	10
'1010'	11
'1011'	12
'1100'	13
'1101'	14
'1110'	15
'1111'	invalid

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

`allocation[ch][sb]` - same as allocation[sb] but now for the channel ch in stereo or dual_channel mode.

`scalefactor[sb]` - indicates the factor of subband sb by which the requantized samples of subband sb shall be multiplied. The six bits constitute an unsigned integer, index to 3-Annex B, Table 3-B.1 "LAYER I, II SCALEFACTORS". Valid for single_channel mode.

`scalefactor[ch][sb]` - same as scalefactor[sb] but now for one of the channels in stereo, intensity_stereo, or dual_channel mode.

`sample[sb][s]` - coded representation of the s-th sample in subband sb. Valid for single_channel subbands and for subbands in intensity_stereo mode. In the latter case the value is valid for both channels.

`sample[ch][sb][s]` - same as sample[sb][s] but for the channel ch in stereo or dual_channel mode.

2.4.2.6 Audio data, Layer II

`allocation[sb]` - contains information concerning the quantizers used for the samples in subband sb, 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 3-Annex B, Table 3-B.2 "LAYER II BIT ALLOCATION TABLES", which gives the number of levels used for quantization. 3-Annex B, Table 3-B.4 "LAYER II CLASSES OF QUANTIZATION" gives additional information concerning each possible quantizer: the requantization coefficients, whether grouping has been used, the number of samples per codeword, and the number of bits per codeword. Several tables exist for different combinations of bitrate and sampling frequency, see 3-Annex B, Table 3-B.2 "LAYER II BIT ALLOCATION TABLES". This is valid for single_channel subbands or for subbands in intensity_stereo mode. In the latter case the allocation is valid for both channels.

`allocation[ch][sb]` - same as allocation[sb] but now for channel ch in stereo or dual_channel mode.

`gr` - index to a granule. In Layer II, a granule consists of 3 consecutive samples from each of the 32 subbands.

`scfsi[sb]` - scalefactor selection information. This gives information on the number of scalefactors transferred for subband sb 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.

'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.

Valid in single_channel mode.

`scfsi[ch][sb]` - same as scfsi[sb] but now for the channel ch in stereo, intensity_stereo, or dual_channel mode.

`scalefactor[sb][p]` - indicates the factor by which the requantized samples of subband sb and of part p of the frame should be multiplied. The six bits constitute an unsigned integer, index to 3-Annex B, Table 3-B.1 "LAYER I, II SCALEFACTORS'. Valid in single_channel mode.

`scalefactor[ch][sb][p]` - same as scalefactor[sb][p] but now for one of the channels ch in stereo, intensity_stereo, or dual_channel mode.

`samplecode[sb][gr]` - coded representation of the three consecutive samples in the granule gr in subband sb. Valid for single_channel subbands and for subbands in intensity_stereo mode. In the latter case, the code is valid for both channels.

`samplecode[ch][sb][gr]` - same as samplecode[sb][gr] but now for channel ch in stereo or dual_channel mode.

`sample[sb][s]` - coded representation of the s-th sample in subband sb. Valid for single_channel subbands and for subbands in intensity_stereo mode. In the latter case the value is valid for both channels.

`sample[ch][sb][s]` - same as sample[sb][s] but now for the channel ch in stereo or dual_channel mode.

2.4.2.7 Audio data, Layer III

`gr` - the granules in Layer III consist of 18 * 32 subband samples. Each frame holds the data from 2 granules. The audio data in a frame is allocated in the following way:

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

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

main_data_end - The value of main_data_end is used to determine the location in the bitstream of the last bit of main_data for the frame. The main_data_end value specifies the location as a negative offset in bytes from the next frame's frame header location in the main_data portion of the bitstream. This is explained in 3-Annex Fig. 3- A.7.1.

private_bits - bits for private use. These bits will not be used in the future by ISO.

main_data_beg - This gives 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.

main_data - The main_data portion of the bitstream contains the scalefactors, Huffman encoded data, and ancillary information.

`scfsi[scfsi_band]` - in Layer III the scalefactor selection information works similarly to Layers I and II. The main difference is the use of the variable scfsi_band to apply scfsi to groups of scalefactors instead of single scalefactors. scfsi controls the use of scalefactors to the granules.

```
'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[scfsi_band][ch]` – same as scfsi[scfsi_band] but for use in stereo, joint_stereo or dual_channel mode
`scfsi_band` - scfsi_band controls the use of the scalefactor selection information for groups of scalefactors (scfsi_bands).

```
scfsi_band  scalefactor bands (see 3-Annex B, Table 3-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]` - 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 each granule and the position of ancillary information (if used).
`part2_3_length[gr][ch]` – same as part2_3_length[gr] but for use in stereo, joint_stereo or dual_channel mode
`part2_length` - this value contains the number of main_data bits used for scalefactors. Its value is given as follows:
For switch_point == 0,
part2_length = 11 * slen1 + 10 * slen2 for long blocks (block_type 0, 1 or 3), and
part2_length = 18 * slen1 + 18 * slen2 for short blocks (block_type==2).

For switch_point == 1,

part2_length = 17 * slen1 + 18 * slen2 (block_type == 2), and

for long blocks (block type 0, 1, or 3) the value of part2_length is the same as that for switch_point == 0.

`big_values[gr]` - 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 don't 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 remains. 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 figure shows the partitioning:

```
xxxxxxxxxxxxx------------------0000000000000000000000000000
|            |                  |                             |
1            bigvalues*2       bigvalues*2+count1*4         iblen
The values 000 are all zero.
The values --- are -1,0 or +1. Their number is a multiple of 4.
The values xxx are not bound.
Iblen is 576.
```

big_values[gr][ch] – same as big_values[gr] but for use in stereo, joint_stereo or dual_channel mode
global_gain[gr] - 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".

global_gain[gr][ch] – same as global_gain[gr] but for use in stereo, joint_stereo or dual_channel mode

scalefac_compress[gr] - 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 switch_point 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 switch_point is 1:
slen1: length of scalefactors for the scalefactor bands 0 to 7 (long window scalefactor band) and 4 to 5 (short window scalefactor band) Note: Scalefactor bands 0-7 are from the "long window scalefactor band" table, and scalefactor bands 4-11 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	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

scalefac_compress[gr][ch] – same as scalefac_compress[gr] but for use in stereo, joint_stereo or dual_channel mode
blocksplit_flag[gr] - signals that the block uses an other than normal (type 0) window.

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

region_address1 = 8 (in case of block_type==1 or block_type==3)
region_address1 = 9 (in case of block_type==2)
region_address2 = 0 In this case the length of region 2 is zero.

If blocksplit_flag is not set, then the value of block_type is zero.
blocksplit_flag[gr][ch] – same as blocksplit_flag[gr] but for use in stereo, joint_stereo or dual_channel mode
block_type[gr] - indicates the window type for the actual granule (see description of the filterbank, Layer III).

```
type 0      reserved
type 1      start block
type 2      3 short windows
type 3      end block
```

Block_type and switch_point give the information about assembling of values in the block and about length and count of the transforms (see 3-Annex A, Figure 3-A.4 for a schematic, 3-Annex C for an analytic description). In the case of block_type=2 the switch_point indicates whether some polyphase filter subbands are coded using long transforms even in case of block_type 2. The polyphase filterbank is described in the clause 2.4.3.2 Layer I.
- In the case of long blocks (block_type not equal to 2 or in the lower subbands of block_type 2) 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) 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.

block_type[gr][ch] – same as block_type[gr] but for use in stereo, joint_stereo or dual_channel mode

switch_point[gr] - signals the split point of short/long transforms. The following table shows the number of the scalefactor band above which window switching (i.e. block_type different from 0) is used.

switch_point	switch_point_l (No of sb)	switch_point_s (No of sb)	
´0´	0	0	; switching of the whole spectrum
´1´	8	3	; switching of higher frequencies only

switch_point[gr][ch] – same as switch_point[gr] but for use in stereo, joint_stereo or dual_channel mode

switch_point_l - Number of scalefactor band (long block scalefactor band) from which point on window switching is used.

switch_point_s - Number of scalefactor band (short block scalefactor band) from which point on window switching is used.

cb_limit - Number of scalefactor bands for long blocks (block_type != 2). This is a constant, 21, for Layer III in all modes and at all sampling frequencies.

cb_limit_short - Number of scalefactor bands for short blocks (block_type=2). This is a constant, 12, for Layer III in all modes and at all sampling frequencies.

window - Number of actual time slot in case of block_type==2, 0 = window = 2.

table_select[region][gr] - 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 3-Annex B Table 3-B.7.

table_select[region][gr][ch] – same as table_select[region][gr] but for use in stereo, joint_stereo or dual_channel mode

subblock_gain[window][gr] - 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.^subblock_gain(window) in the decoder.

subblock_gain[window][gr][ch] – same as subblock_gain[window][gr] but for use in stereo, joint_stereo or dual_channel mode

region_address1[gr] - 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. Each region is coded using a different Huffman code table depending on the maximum quantized value and the local signal statistics.
The values region_address[1,2] are used to point to the boundaries of the regions. The region boundaries are aligned with the partitioning of the spectrum into critical bands.
In case of block_type==2 (short blocks) the scalefactor bands representing the different time slots are counted separately. If switch_point==0, the total amount of scalefactor bands for the granule in this case is 12*3=36. If block_type==2 and switch_point==1, the amount of scalefactor bands is 8+9*3=35.
region_address1 counts the number of scalefactor bands until the upper edge of the first region:

```
region_address1   upper edge of region is
       upper edge of scalefactor band number:
0      0  (no first region)
1      1
2      2
...    ...
15     15
```

region_address1[gr][ch] – same as region_address1[gr] but for use in stereo, joint_stereo or dual_channel mode

region_address2[gr] - region_address2 counts the number of scalefactor bands which are partially or totally in region 3. Again if block_type==2 the scalefactor bands representing different time slots are counted separately.

region_address2[gr][ch] – same as region_address2[gr] but for use in stereo, joint_stereo or dual_channel mode

preflag[gr] - 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 3-Annex B, Table 3-B.6). This is equivalent to multiplication of the requantized scalefactors with table values. preflag is never used if block_type==2 (short blocks).

preflag[gr][ch] – same as preflag[gr] but for use in stereo, joint_stereo or dual_channel mode

scalefac_scale[gr] - the scalefactors are logarithmically quantized with a step size of 2 or ($\sqrt{2}$) depending on scalefac_scale.

```
scalefac_scale = 0       stepsize sqrt(2)
scalefac_scale = 1       stepsize 2
```

scalefac_scale[gr][ch] – same as scalefac_scale[gr] but for use in stereo, joint_stereo or dual_channel mode

count1table_select[gr] - 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 = 0	Table A of 3-Annex B.7
count1table_select = 1	Table B of 3-Annex B.7

count1table_select[gr][ch] – same as count1table_select[gr] but for use in stereo, joint_stereo or dual_channel mode

scalefac[cb][gr] - the scalefactors are used to colour the quantization noise. If the quantization noise is colored with the right shape, it is masked completely. Unlike LayersI and II, the LayerIII 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 blocks of values. In the case of Layer III, the blocks stretch over several frequency lines. These blocks are called scalefactor bands and are selected to resemble critical bands as close 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 (see clause 2.4.3.4, MS_stereo mode).

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 3-Annex Table 3-B.8).

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

scalefac[cb][window][gr] – same as scalefac[cb][gr] but for different windows if block_type==2

scalefac[cb][gr][ch] – same as scalefac[ch][gr] but for use in stereo, joint_stereo or dual_channel mode

scalefac[cb][window][gr][ch] – same as scalefac[cb][window][gr] but for use in stereo, joint_stereo or dual_channel mode

Huffman_code_bits
To get a clear picture of the Huffman code syntax some pseudo-functions and structures have to be defined:
All quantized values of absolute value 15 and less are directly coded using a Huffman code. Always pairs of values (x,y) are coded. If quantized values of magnitude greater than 15 are found, ESC-codes are used to flag these values. If one or both values of a pair is not zero, one or two sign bits are appended to the Huffman code word.

```
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
max_table_entry         is the maximum table entry index. This is a system
                        constant (15, maximum number of entries in a single
                        table is 256)
signx                   sign of the 1st. value (0 if positive, 1 if negative)
signy                   sign of the 2nd. value (0 if positive, 1 if negative)

struct coded_word {
        codeword        hcod[|x|][|y|], length is hlen[|x|][|y|]
        linbitsx        If (x=max_table_entry) this constitutes an ESC-code.
                        In this case the length of this field is linbits,
                        else zero. The unsigned integer contained in this
                        field is added to max_table_entry -1 to establish the
                        absolute value of the encoded data.
        signx           sign of x (transmitted only if x not equal 0)
```

```
          linbitsy        See linbitsx.
          signy           sign of y (transmitted only if y not equal 0)
          }
```

The ESCaped codes linbitsx or linbitsy are only used if a value greater or equal to the table maximum is actually flagged. They are never used if the selected table is one for blocks with a maximum quantized value equal or less than max_table_entry. The sign bits are transmitted only if the value of x with respect to y is different from zero.

For the higher end of the spectrum quadruples of values are coded using one of two special tables. Again magnitude values are coded using a Huffman code. One of the two codes is not really a 4-dimensional code because it is constructed from the trivial code: 0 is coded with a 1 (no sign bit needed), 1 is coded with a 0 (sign bit added). In both cases the Huffman code is assembled as follows:

```
struct quad_word {
          codeword        hcod[|v|][|w|][|x|][|y|], hlen[|v|][|w|][|x|][|y|]
          signv           only if v not equal 0
          signw           only if w not equal 0
          signx           only if x not equal 0
          signy           only if y not equal 0
          }
```

At the high end of the spectrum the number of pairs of zeroes is simply counted. As this value is implicitly known when the other values have been decoded, it is not transmitted.

Ordering of Huffman encoded data:

If the block_type is 0, 1 or 3 the Huffman encoded data are ordered in terms of increasing frequency.

If the block_type is 2 (short blocks) then the Huffman encoded data are ordered in a pattern similar to that of the scalefactor values (see clause 2.4.2.7):

The Huffman encoded data are given for successive scalefactor bands, beginning with scalefactor band 0 and ending with scalefactor band 11. Within each scalefactor band, the data is given for successive time windows, beginning with window 0 and ending with window 2. The data values within each window are arranged in order of increasing frequency.

2.4.2.8 Ancillary data

`Ancillary_bit` - user definable

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 just after the syncword : the bitstream is subdivided in slots. The distance between the start of two consecutive syncwords is constant and equals "N" slots. The value of "N" depends on the Layer.
For Layer I the following equation is valid:
N = 12 * bit_rate / sampling_frequency.
For Layers II and III the equation becomes:

N = 144 * bit_rate / 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 clause 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:

- 16 bits of header(), starting with bit_rate_index and ending with emphasis
- a number of bits of audio_data(), starting with the first bit. This number is given by 3-AnnexB,Table 3-B.5 "NUMBER OF PROTECTED AUDIO_DATA BITS".

The method is depicted in 3-Annex A, Figure 3-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 3-Annex A, Figure 3-A.9 "CRC-CHECK DIAGRAM". The outputs b15...b0 constitute a word to be compared with the CRC-check word in the bitstream. If the words are not identical, a transmission error has occured 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 clause 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 3-Annex A, Figure 3-A.1 "LAYER I AND II DECODER FLOW CHART".

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 clause 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 :

$$s'' = (2^{nb} / (2^{nb} - 1)) * (s''' + 2^{-nb+1})$$

where:

	s'''	is the fractional number,
	s''	the requantized value,
	nb	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 3-Annex B, Table 3-B.1 "LAYER I, II SCALEFACTORS". The rescaled value s' is calculated as :

$$s' = factor * s''.$$

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 3-Annex A, Figure 3-A.2 "SYNTHESIS SUBBAND FILTER FLOW CHART" show the reconstruction operation. The coefficients N_{ik} for the matrixing operation are given by

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

The coefficients D_i for the windowing operation can be found in 3-Annex B, Table 3-B.3 "COEFFICIENTS Di OF THE SYNTHESIS WINDOW". One frame contains 12 * 32 = 384 subband samples, which result, after filtering, in 384 audio samples.

2.4.3.3 Layer II

LayerII is a more efficient but more complex coding scheme than LayerI. The flowchart in 3-Annex A, Figure 3-A.1 "LAYER I AND II DECODER FLOW CHART" applies to both LayersI and II. The first step is to perform the decoding which is common to all three layers (see clause 2.4.3.1).

Bit allocation decoding

For different combinations of bitrate and sampling frequency different bit allocation tables exist (3-Annex B, Table3-B.2 "LAYER II BIT ALLOCATION TABLES"). 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 3-Annex B, Table 3-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 3-Annex B, Table 3-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'.

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.

Scalefactor decoding

For every subband with a nonzero bit allocation the coded scalefactor 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 3-Annex B, Table 3-B.1 "LAYER I, II SCALEFACTORS". This table gives the scalefactor by which the relevant subband samples should be multiplied after requantization.

Requantization of subband samples

Next the coded samples are read. As can be seen from clause 2.4.1.6, the coded samples appear as triplets, the code contains three consecutive samples at a time. From 3-Annex B, Table 3-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 3-Annex B, Table 3-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 3-Annex B, Table 3-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'' the requantized value.

The values of the constants C and D are given in 3-Annex B, Table 3-B.4 "LAYER II CLASSES OF QUANTIZATION". The requantized values have to be rescaled. The multiplication factors can be found in the 3-Annex B, Table 3-B.1 "LAYER I, II SCALEFACTORS". as described above. The rescaled value s' is calculated as:

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

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 3-Annex A, Figure 3-A.2 "SYNTHESIS SUBBAND FILTER FLOW CHART" have to be carried out. The coefficients N_{ik} for the matrixing operation are given by

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

The coefficients Di for the windowing operation can be found in 3-Annex B, Table 3-B.3 "COEFFICIENTS Di OF THE SYNTHESIS WINDOW". One frame contains 36*32=1152 subband samples, which result after filtering in 1152 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 a MDCT. The window length of the MDCT is 36. Adaptive window switching is used to control time artifacts (pre-echoes), see the description in 3-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 'switch_point' 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 3-Annex 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 3-Annex C). The size of the input buffer is the size of one frame at the bitrate of 160 kbit/s per channel for LayerIII. 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.

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 3-Annex B.8).

Side information

Decoding of the side information requires storage of the decoded parameters. The table select information is used to select the decoder table and the number of ESC-bits (linbits), according to the table in the 3-Annex B-B.7.

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 Fig. 3-Annex A.7.1 and 3-Annex A.7.2. The begin of the main data part is located by using the main_data_end pointer of the preceding 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 position from the bit_rate_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 Fig. 3-Annex A.7.2).

MS_stereo mode

This mode switch (found in the header: mode_extension) allows switching from 'independent stereo' to MS_stereo. The upper bound of the scalefactor bands decoded in ms stereo is derived from the "zero_part" of the difference (right) channel. Above this bound intensity stereo can 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 clause 2.4.2.7) to the Nyquist rate.

- MS matrix

In MS stereo mode the values of the normalized middle/side channels Mi/Si are transmitted instead of the left/right channel values Li/Ri. Thus Li/Ri are reconstructed using

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

The values Mi are transmitted in the left, values Si are transmitted in the right channel

Intensity stereo mode

This mode switch (found in the header: mode_extension) allows switching from 'normal stereo' to intensity stereo. 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 (left 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:

- the intensity stereo position is_possb is read from the scalefactor of the right channel
- if (is_possb == 7) do not perform the following steps (illegal is_pos)
- is_ratio = tan(is_possb * p/12)

- $L_i := L_i * \dfrac{\text{is\_ratio}}{1 + \text{is\_ratio}}$ for all indices i within the actual scalefactor band sb
- $R_i := L_i * \dfrac{1}{1 + \text{is\_ratio}}$ for all indices i within the actual scalefactor band sb

Scalefactors

The scalefactors are decoded according to the actual slen1 and slen2 which themselves are decoded from scalefac_select. 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.

Huffman decoding

Huffman decoding is done using a state machine. The state machine works from a ROM table, where each entry is the information for one node in the decoder tree. All necessary information including the table which realizes the Huffman code tree can be generated from the tables in 3-Annex B, Table 3-B.7. Decoding is done until all Huffmancodebits have been decoded or until quantized values representing 576 frequency lines have been decoded, whichever comes first. If there are more Huffmancodebits than necessary to decode 576 values they are regarded as stuffing bits and discarded.

Requantizer

The nonuniform quantizer uses a power law. For each output valueY from the Huffman decoder Y4/3 is calculated. This can be done either by table lookup or by explicit calculation.

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 3-Annex 3-A.8.

The following example is given for the example of a granule containing data with block_type==2 (short blocks). It can accordingly be used for other block types. 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):

$$xr(i) = is(i)^{4/3} \;*\; 2^{.25*\left(\text{global\_gain}[gr] - 64 - 8 * \text{subblock\_gain}[window][gr]\right)} *$$
$$2^{.25*\left(-2 * \left(1+\text{scalefac\_scale}[gr]\right) * \text{scalefac}[cb][window][gr] - 2 * \text{preflag}[gr] * \left(1+\text{scalefac\_scale}[gr]\right) * \text{pretab}[cb]\right)}$$

The constant 64 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. If an implementation with a different power transfer characteristic is chosen (different global scaling) then the constant has to be changed accordingly.

Synthesis filterbank

3-Annex A, Figure 3-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 3-Annex A Figure 3-A.5 and in 3-Annex B Table 3-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.

Buffer considerations

The following rule can be used to calculate the maximum number of bits used for one granule:
At the highest possible bitrate of Layer III (320 kbit/s per stereo signal) the frames must be of constant length, i.e. one buffer length is

320000 * .024 bit = 7680 bit.

This value is used as the maximum buffer per channel at the lower bitrates. At 64 kbit/s (128 kbit/s stereo) the mean granule length is 64000/48000*576 = 768 bit at 48 kHz sampling frequency. This means that there is a maximum deviation (short time buffer) of 7680 - 4*768 = 4608 bits is allowed at 64 kbit/s. The actual deviation is equal to the number of bytes denoted by the main_data_end offset pointer. The actual maximum deviation is 2**9 * 8 bit = 4096 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_end is always set to 0 in the case of bit_rate_index==14, i.e. data rate 320 kbps per stereo signal. In this case all data are allocated between adjacent header words.

IMDCT

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

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

Windowing

Depending on the block_type different shapes of windows are used.

a) block_type=0

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

b) block_type=1

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

c) block_type=3

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

d) block_type=2:

Each of the three small blocks is windowed separately.

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

The windowed small blocks must be overlapped and concatenated.

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

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:

$$x_i = y_i + s_i \qquad \text{for i=0 to 17}$$

$$s_i = y_{i+18} \qquad \text{for i=0 to 17}$$

3-ANNEX A(informative)
DIAGRAMS

Figure 3-A.1. Layer I and II decoder flow chart

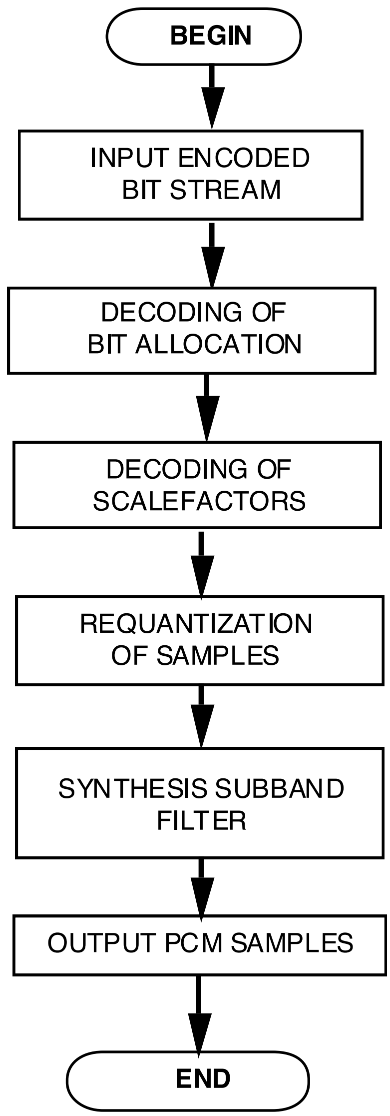


Figure 3-A.2. Synthesis subband filter flow chart

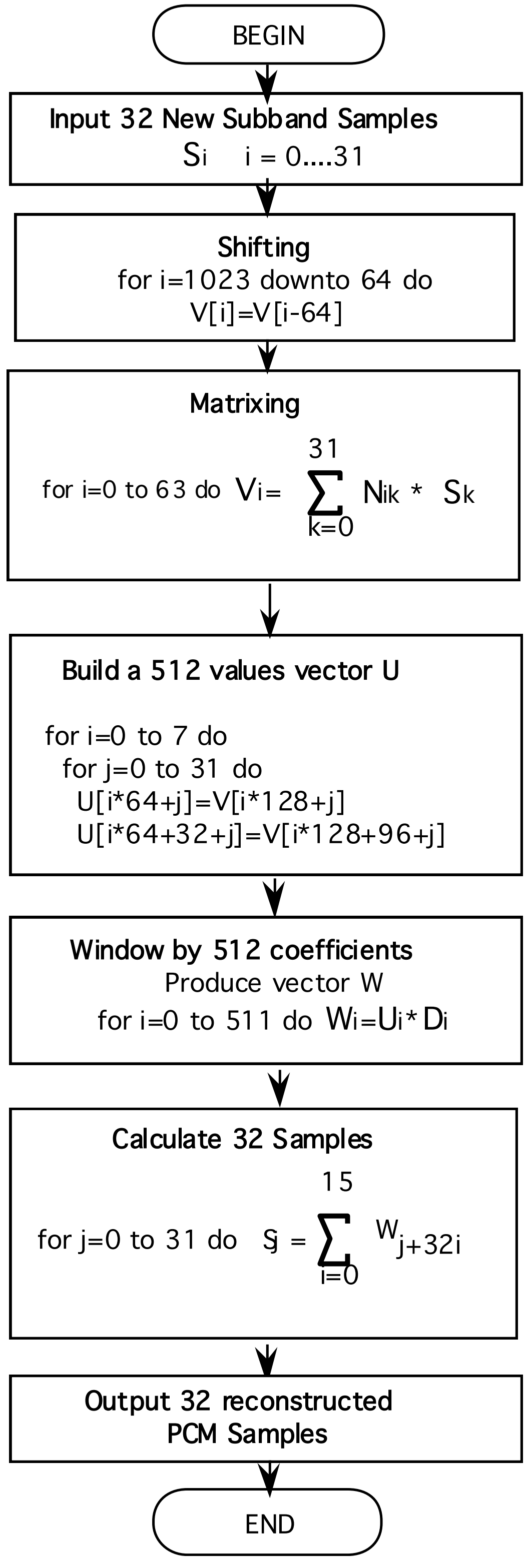


Figure 3-A.3. Layer III decoder flow chart

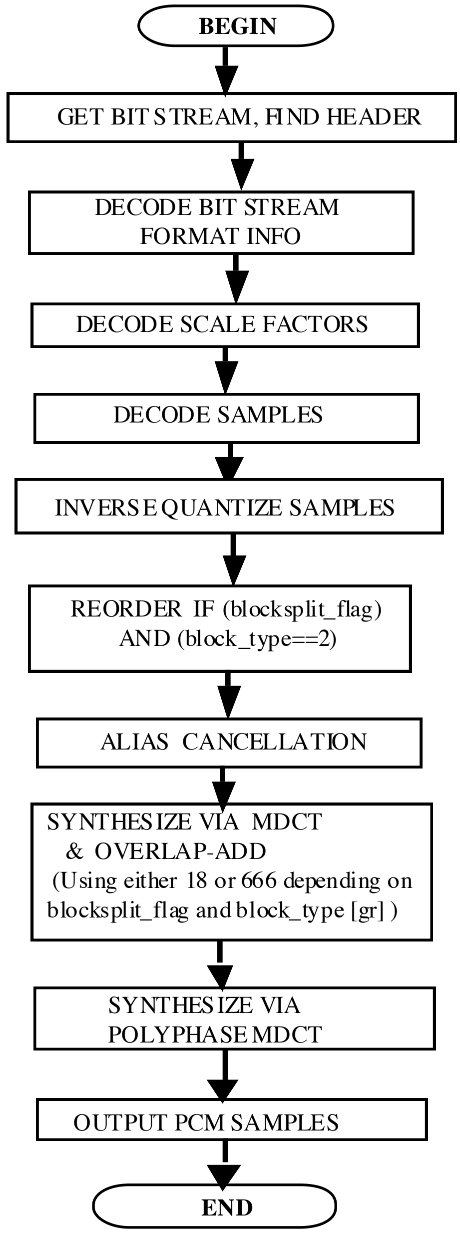


Figure 3-A.4. Layer III decoder diagram

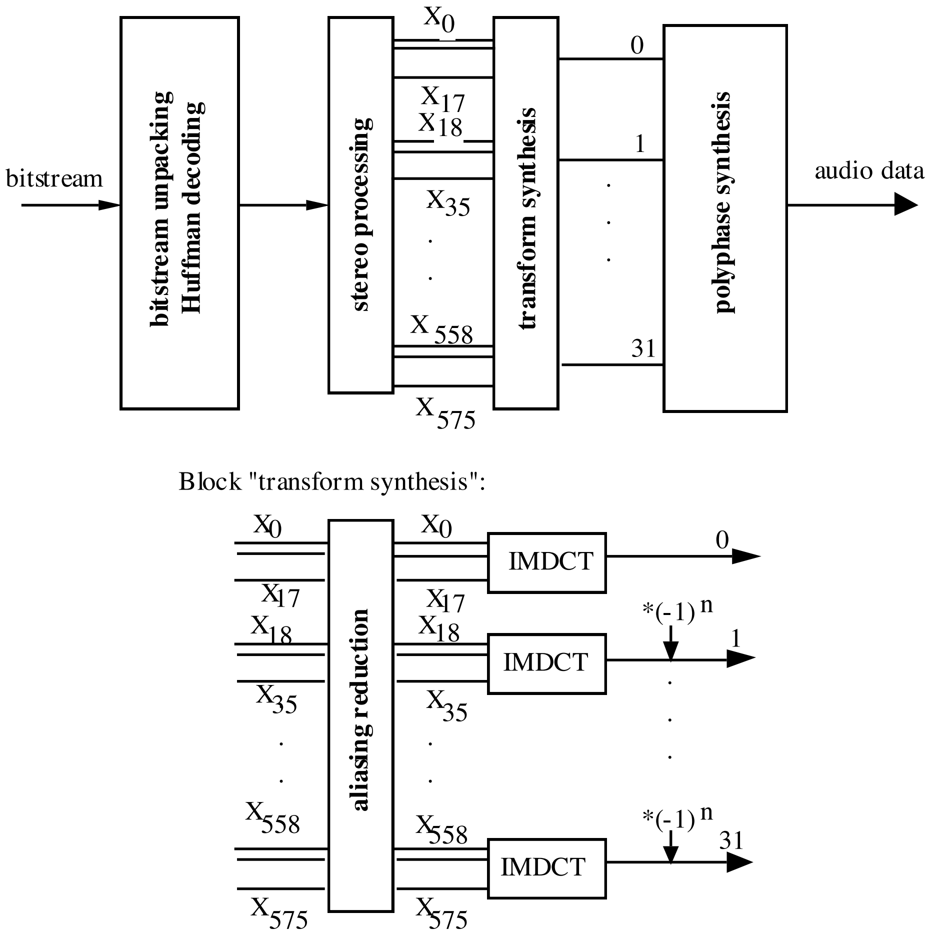


Figure 3-A.5. Layer III aliasing reduction encoder/decoder diagram

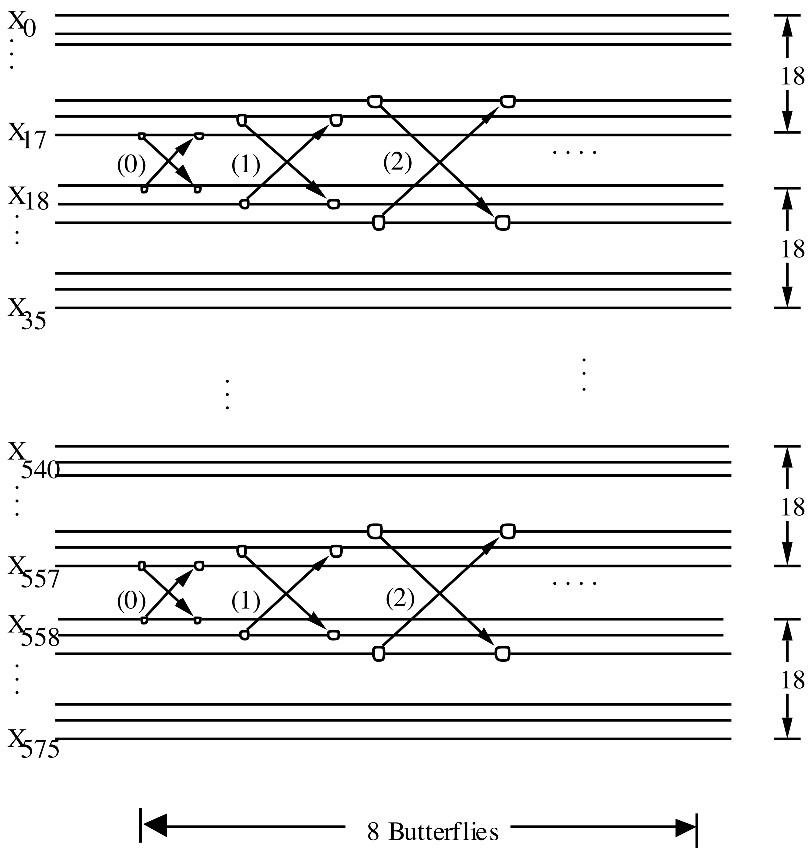


Figure 3-A.6. Layer III aliasing-butterfly, decoder

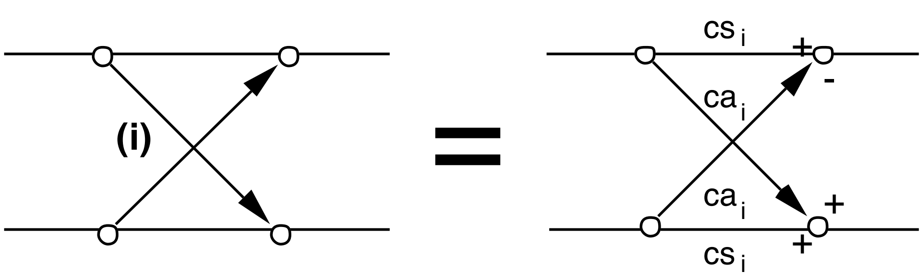


Figure 3-A.7.1. Layer III bitstream organization

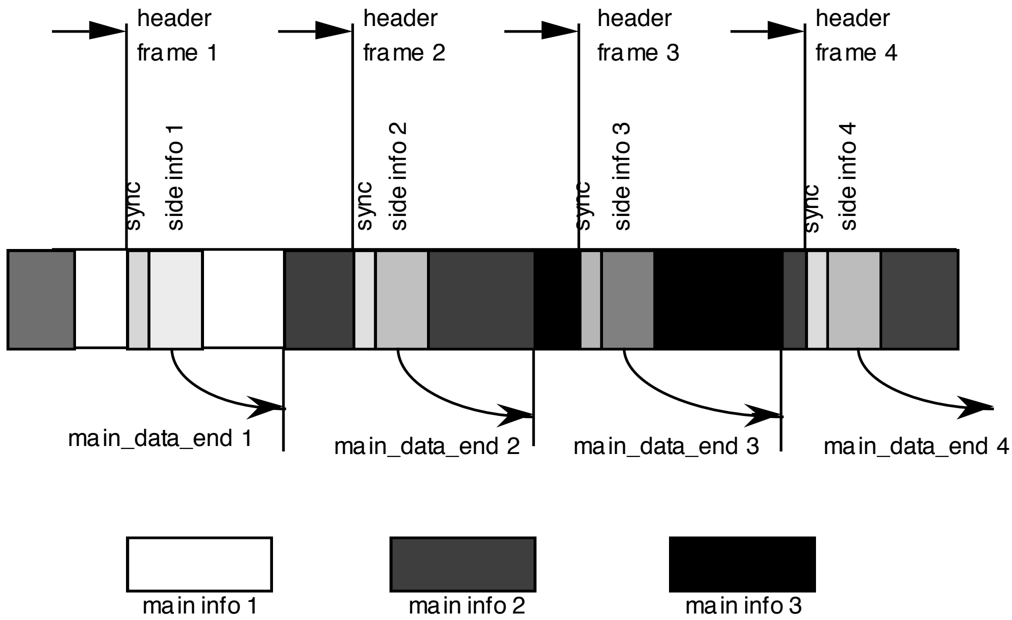
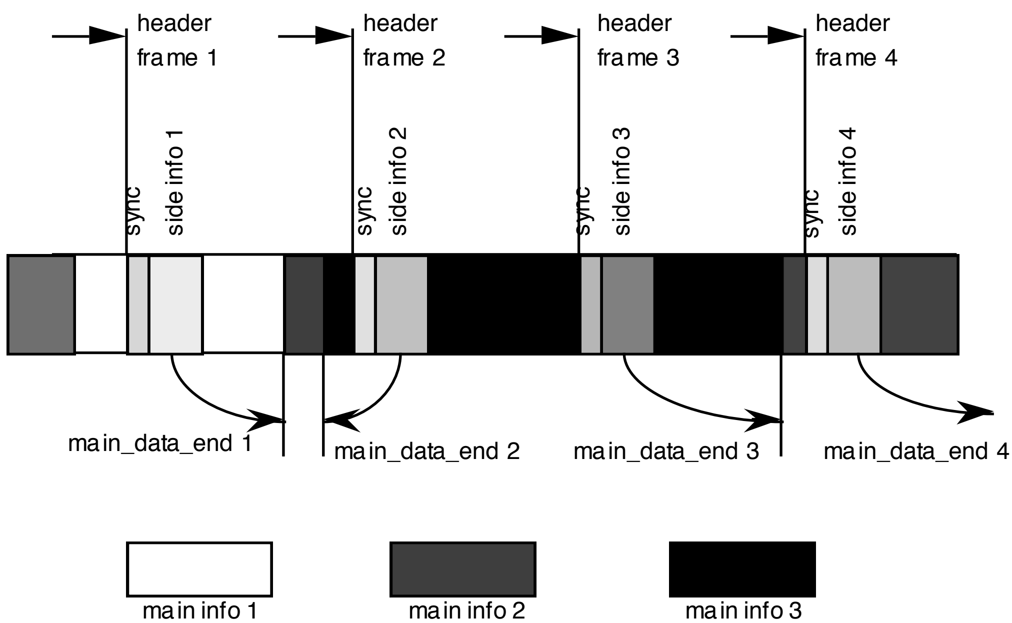


Figure 3-A.7.2. Layer III bitstream organization with peak demand at main info 3 and small demand at main info 2.



Note: 'info' means information

Figure 3-A.8. Layer III illustration of granules for frame with no block split in first granule and block split in second granule.

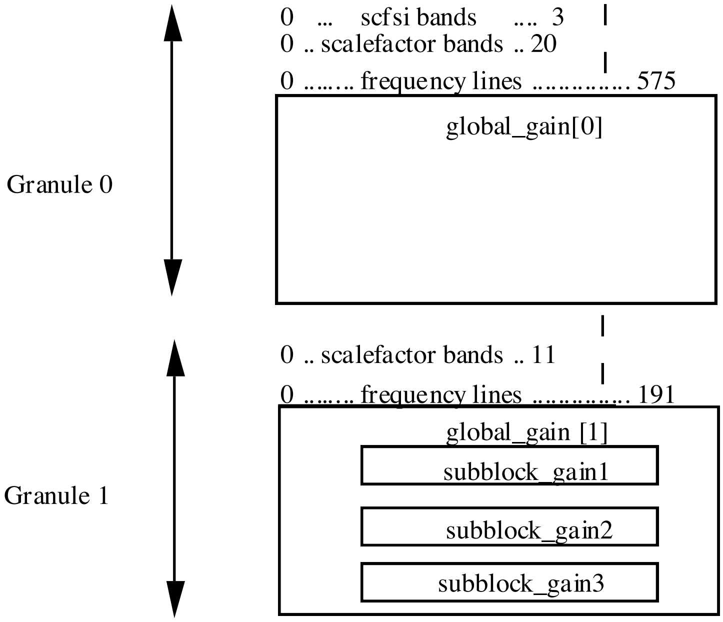
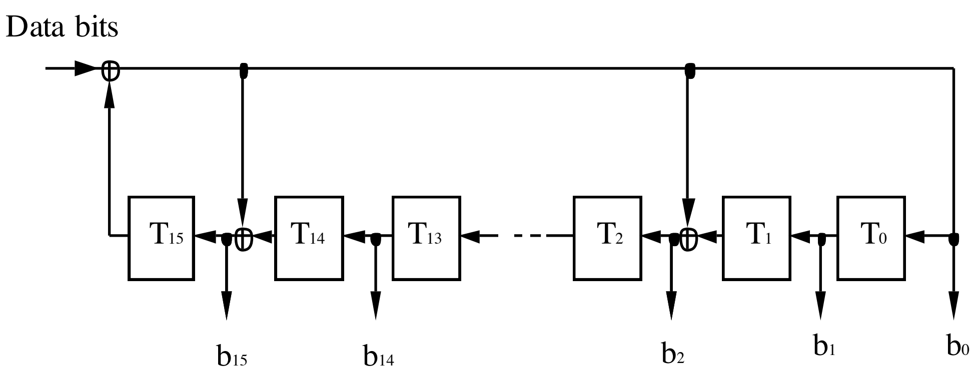


Figure 3-A.9. CRC-Check diagram



3-ANNEX B (normative)
TABLES

Table 3-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		

3-B.2. Layer II bit allocation tables

Table 3-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
SB0	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
SB1	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
SB2	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
SB3	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB4	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB5	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB6	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535

SB7	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB8	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB9	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB10	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB11	3	-	3	5	7	9	15	31	65535								
SB12	3	-	3	5	7	9	15	31	65535								
SB13	3	-	3	5	7	9	15	31	65535								
SB14	3	-	3	5	7	9	15	31	65535								
SB15	3	-	3	5	7	9	15	31	65535								
SB16	3	-	3	5	7	9	15	31	65535								
SB17	3	-	3	5	7	9	15	31	65535								
SB18	3	-	3	5	7	9	15	31	65535								
SB19	3	-	3	5	7	9	15	31	65535								
SB20	3	-	3	5	7	9	15	31	65535								
SB21	3	-	3	5	7	9	15	31	65535								
SB22	3	-	3	5	7	9	15	31	65535								
SB23	2	-	3	5	65535												
SB24	2	-	3	5	65535												
SB25	2	-	3	5	65535												
SB26	2	-	3	5	65535												
SB27	0	-															
SB28	0	-															
SB29	0	-															
SB30	0	-															
SB31	0	-															

sblimit = 27
Sum of nbal = 88

Table 3-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
SB0	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
SB1	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
SB2	4	-	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
SB3	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB4	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535

SB5	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB6	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB7	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB8	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB9	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB10	4	-	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
SB11	3	-	3	5	7	9	15	31	65535								
SB12	3	-	3	5	7	9	15	31	65535								
SB13	3	-	3	5	7	9	15	31	65535								
SB14	3	-	3	5	7	9	15	31	65535								
SB15	3	-	3	5	7	9	15	31	65535								
SB16	3	-	3	5	7	9	15	31	65535								
SB17	3	-	3	5	7	9	15	31	65535								
SB18	3	-	3	5	7	9	15	31	65535								
SB19	3	-	3	5	7	9	15	31	65535								
SB20	3	-	3	5	7	9	15	31	65535								
SB21	3	-	3	5	7	9	15	31	65535								
SB22	3	-	3	5	7	9	15	31	65535								
SB23	2	-	3	5	65535												
SB24	2	-	3	5	65535												
SB25	2	-	3	5	65535												
SB26	2	-	3	5	65535												
SB27	2	-	3	5	65535												
SB28	2	-	3	5	65535												
SB29	2	-	3	5	65535												
SB30	0	-															
SB31	0	-															

sblimit = 30
Sum of nbal = 94

Table 3-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
SB0	4	-	3	5	9	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767
SB1	4	-	3	5	9	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767
SB2	3	-	3	5	9	15	31	63	127								
SB3	3	-	3	5	9	15	31	63	127								
SB4	3	-	3	5	9	15	31	63	127								
SB5	3	-	3	5	9	15	31	63	127								

SB6	3	-	3	5	9	15	31	63	127
SB7	3	-	3	5	9	15	31	63	127
SB8	0	-							
SB9	0	-							
SB10	0	-							
SB11	0	-							
SB12	0	-							
SB13	0	-							
SB14	0	-							
SB15	0	-							
SB16	0	-							
SB17	0	-							
SB18	0	-							
SB19	0	-							
SB20	0	-							
SB21	0	-							
SB22	0	-							
SB23	0	-							
SB24	0	-							
SB25	0	-							
SB26	0	-							
SB27	0	-							
SB28	0	-							
SB29	0	-							
SB30	0	-							
SB31	0	-							

sblimit = 8
Sum of nbal = 26

Table 3-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
SB0	4	-	3	5	9	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767
SB1	4	-	3	5	9	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767
SB2	3	-	3	5	9	15	31	63	127								
SB3	3	-	3	5	9	15	31	63	127								
SB4	3	-	3	5	9	15	31	63	127								
SB5	3	-	3	5	9	15	31	63	127								
SB6	3	-	3	5	9	15	31	63	127								
SB7	3	-	3	5	9	15	31	63	127								
SB8	3	-	3	5	9	15	31	63	127								
SB9	3	-	3	5	9	15	31	63	127								
SB10	3	-	3	5	9	15	31	63	127								
SB11	3	-	3	5	9	15	31	63	127								
SB12	0	-															

SB13	0	-
SB14	0	-
SB15	0	-
SB16	0	-
SB17	0	-
SB18	0	-
SB19	0	-
SB20	0	-
SB21	0	-
SB22	0	-
SB23	0	-
SB24	0	-
SB25	0	-
SB26	0	-
SB27	0	-
SB28	0	-
SB29	0	-
SB30	0	-
SB31	0	-

Max. No. of active subbands = 12
Sum of nbal = 38

Table 3-B.3. Coefficients Di 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 3-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 3-B.5. Number of protected audio_data bits

Layer	bit alloc. table no.	no. of bits single channel mode	no. of bits other modes
I	-	128	256
I I	3-B.2a	142	284
I I	3-B.2b	154	308
I I	3-B.2c	42	84
I I	3-B.2d	62	124
I I I	-	136	256

Table 3-B.6. Layer III Preemphasis

0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 2 2 3 3 3 2

Table 3-B.7. Huffman codes for Layer III

Huffman code table for quadruples (A)

Value	hlen	hcod
0000	1	1
0001	4	0101
0010	4	0100
0011	5	00101
0100	4	0110
0101	6	000101
0110	5	00100
0111	6	000100
1000	4	0111
1001	5	00011
1010	5	00110
1011	6	000000
1100	5	00111
1101	6	000010
1110	6	000011
1111	6	000001

Huffman code table for quadruples (B)

Value	hlen	hcod
0000	4	1111
0001	4	1110
0010	4	1101
0011	4	1100
0100	4	1011
0101	4	1010
0110	4	1001
0111	4	1000
1000	4	0111
1001	4	0110
1010	4	0101
1011	4	0100
1100	4	0011
1101	4	0010
1110	4	0001
1111	4	0000

Huffman code table 0

x	y	hlen
0	0	0

Huffman code table 1

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

Huffman code table 2

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

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

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

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

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

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

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

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

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

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

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

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

ESC table, 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
ESC table, 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 3-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 3-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 3-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 3-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 3-B.9 Layer III coefficients for aliasing reduction:

(i)	ci
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 csi and cai are calculated as follows:

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

3-ANNEX C (informative)

THE ENCODING PROCESS

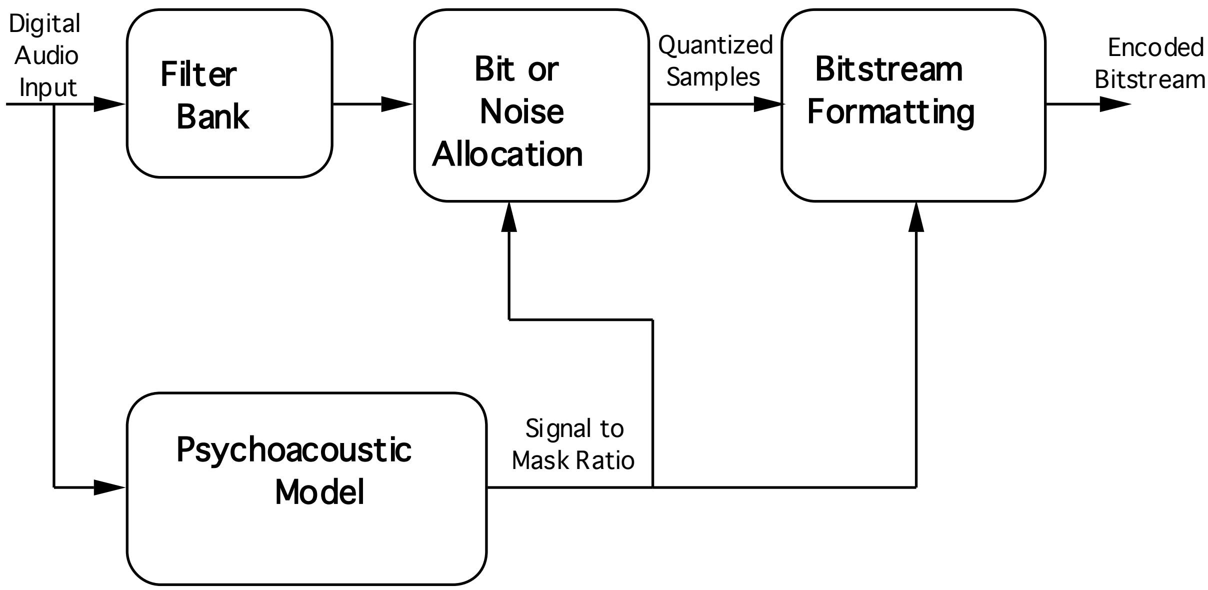
3-C.1 Encoder

3-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.

INTRODUCTION

The MPEG-Audio algorithm is a psychoacoustic algorithm. The figure below shows the primary parts of a psychoacoustic algorithm.



The four primary parts of the psychoacoustic encoder are:

1) The Filterbank:
The filterbank does a time to frequency mapping. There are two filterbanks used in the MPEG-Audio algorithm, each providing 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.

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 3-Annex D. While they can both be applied to any layer of the 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).

3) Bit or Noise Allocation:
The allocator looks at both the output samples from the   and the SMR's from the psychoacoustic model, and adjusts the bit allocation (Layers I and II) or noise allocation (Layer III) in order to simultaneously 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.

4) The bitstream formatter:
The bitstream formatter takes the quantized filterbank outputs, 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.

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-dependant 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.

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 the actually injected noise. In either case, the result is a set of quantization parameters and quantized output samples that are given to the bistream formatter.

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.

3-C.1.2 Input High-Pass Filter

The encoding algorithms provide a frequency response down to DC. 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 10Hz.

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

3-C.1.3 Analysis Subband Filter

An analysis subband filterbank is used to split the broadband signal with sampling frequency fs into 32 equally spaced subbands with sampling frequencies fs/32. The flow chart of this process with the appropriate formulas is given in Figure 3-C.1 "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 audiosamples are shifted in at positions 0 to 31, the most recent on at position 0, and the 32 oldestelements are shifted out.
- Window vector X by vector C. The coefficients are to be found in Table 3-C.1"COEFFICIENTS C_i FOR THE ANALYSIS WINDOW".
- Calculate the 64 values Y_i according to the formula given in theflow chart.
- Calculate the 32 subband samples S_i by matrixing. The coefficients for the matrix can be calculated by the following formula:

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

Table 3-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

3-C.1.4 Psychoacoustic Models

Two examples of psychoacoustic models are presented in Annex D, "PSYCHOACOUSTIC MODELS".

3-C.1.5 Encoding

3-C.1.5.1 Layer I Encoding

1. Introduction

This clause describes a possible Layer I encoding method. The description is made according to Figure 3-C.2, "LAYERI, II ENCODER FLOW CHART".

2. Psychoacoustic Model

The calculation of the psychoacoustic parameters can be done either with Psychoacoustic Model I described in Annex D, clause 3-D.1. or with Psychoacoustic Model II as described in Annex D, clause 3-D.2. The FFT shiftlength equals 384 samples. Either model provides the signal-to-mask ratio for every subband.

3. Analysis Subband Filtering

The subband analysis is described in the clause 3-C.1.3, "ANALYSIS SUBBAND FILTER".

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 next largest value in 3-Annex B, Table 3-B.1., "LAYER I, II SCALEFACTORS" is used as the scalefactor.

5. Coding of Scalefactors

The index in the 3-Annex B, Table 3-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.

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 numbers of bits needed for bit allocation "bbal", and the number of bits required for ancillary data "banc":

adb = cb - (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 clause 2.4.2.5 of the main part of the audio standard (Audio data, LayerI); 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 3-Annex C, Table 3-C.2., "LAYER I SIGNAL-TO-NOISE-RATIOS". 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-(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.

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 3-Annex C, Table 3-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.

8. Coding of Bit Allocation

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

9. Formatting

The encoded subband information is transferred in frames (See also clauses 2.4.1.2, 2.4.1.3, 2.4.1.5 and 2.4.1.8 of the clause 2.4.1 "Specification of the Coded Audio Bitstream Syntax " of the main part of the audio standard. 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 LayerI is 32 bits. The number of slots in a frame can be computed by this formula :

Number of slots/frame (N) = * 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 below:

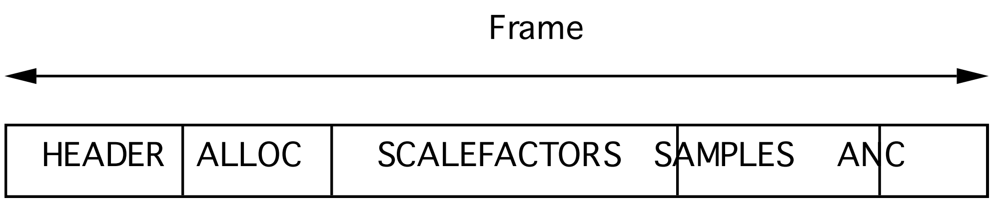


TABLE 3-C.2 LAYER I SIGNAL-TO-NOISE RATIOS

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
1023	61.96
2047	67.98
4095	74.01
8191	80.03
16383	86.05
32767	92.01

TABLE 3-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
1023	0.999023438	-0.000976563

2047	0.999511719	-0.000488281
4095	0.999755859	-0.000244141
8191	0.999877930	-0.000122070
16383	0.999938965	-0.000061035
32767	0.999969482	-0.000030518

3-C.1.5.2 Layer II Encoding

1. Introduction

This clause describes a possible Layer II encoding method. The description is made according to Figure 3-C.2, "LAYERI, II ENCODER FLOWCHART".

2. Psychoacoustic Model

The calculation of the psychoacoustic parameters can be done either with Psychoacoustic Model I described in Annex D, clause 3-D.1. or with Psychoacoustic Model II described in Annex D, clause 3-D.2. If Psychoacoustic Model I is used to calculate the psychoacoutic parameters, the FFT shiftlength is 1152 samples. If Psychoacoustic Model II 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.

3. Analysis Subband Filter

The analysis subband filter is described in clause 3-C.1.3, "ANALYSIS SUBBAND FILTER".

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 next largest value in 3-Annex B, Table 3-B.1., "LAYER I, II TABLE OF SCALEFACTORS" is the scalefactor.

5. Coding of Scalefactors

A frame corresponds to 36 subband samples and therefore contains three scalefactors per subband. Define 'scf' as the index in Annex B, Table 3-B.1., "LAYER I, II SCALEFACTORS". First, the two differences dscf1 and dscf2 of the successive scalefactor indices scf1, scf2 and scf3 are calculated:

dscf1 = scf1 - scf2
dscf2 = scf2 - scf3

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 3-C.4., "LAYER II SCALEFACTOR TRANSMISSION PATTERNS". The "adjusted scalefactor pattern" 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 must be transmitted for a certain subband within one frame. Only the scalefactors indicated in the "transmission pattern" are transmitted. The information describing the number and the position of the scalefactors in each subband is called "scalefactor select information".

6. Coding of Scalefactor Select Information

The "scalefactor select information" (scfsi) is coded by a two bit word, which is also to be found in 3-ANNEX C, Table 3-C.4., "LAYER II SCALEFACTOR TRANSMISSION PATTERNS". Only the scfsi for the subbands which will get a nonzero bit allocation are transmitted.

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 available bits "cb", the number of bits needed for bit allocation "bbal", and the number of bits "banc" required for ancillary data:

adb = cb - (bbal + banc)

The resulting number 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. Use is made of 3-Annex B, Table 3-B.2., "LAYER II POSSIBLE QUANTIZATIONS PER SUBBAND" that indicates for every subband the number of steps that may be used to quantize the samples. The number of bits required to represent these quantized samples can be derived from 3-Annex B, Table 3-B.4., "LAYER II CLASSES OF QUANTIZATION".

The allocation procedure is an iterative procedure where, in each iteration step the number of levels of the subband that has the 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 table 3-C.5. "LAYER II SIGNAL-TO-NOISE-RATIOS". 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 entry in the relevant Annex B, Table 3-B.2., "LAYER II POSSIBLE QUANTIZATIONS PER SUBBAND".

- 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, bsel has to be updated, and bscf has to be updated according to the number of scalefactors required for this subband. Then adb is calculated again using the formula :

adb=cb-(bbal + bsel + bscf + bspl +banc)

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

8. Quantization and Encoding of Subband Samples

Each of the 12 subband samples is normalized by dividing its value by the scale factor to obtain X and quantized using the following formula:

- Calculate A * X + B
- Take the N most significant bits.
- Invert the MSB

A and B can be found in the 3-ANNEX C, TABLE 3-C.6., "LAYER II QUANTIZATION COEFFICIENTS". Nrepresents the necessary number of bits to encode the number of steps. The inversion of the MSB is done in order to avoid the all '1' code that is used for the synchronization word.

Given the number of steps that the samples will be quantized to, 3-Annex B, Table 3-B.4., "LAYER II CLASSES OF QUANTIZATION" shows whether grouping will be used. If grouping is not required, the three samples are coded with individual codewords.

If grouping is required, three consecutive samples are coded as one codeword. Only one value v_m, MSB first, is transmitted for this triplet. The relationships between the coded value v_m (m=3,5,9) and the three consecutive subband samples x, y, z are:

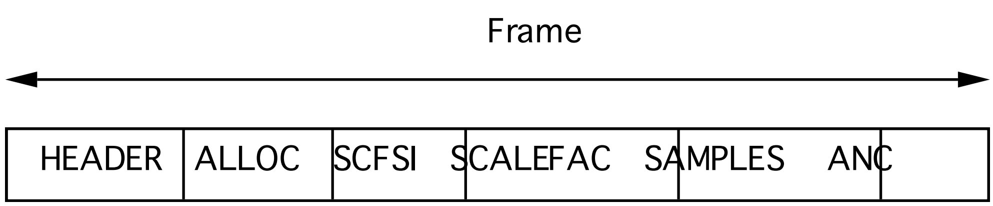
$v3 = 9z + 3y + x$ (v3 in 0... 26)
$v5 = 25z + 5y + x$ (v5 in 0...124)
$v9 = 81z + 9y + x$ (v9 in 0...728)

9. Coding of Bit Allocation

For the purpose of a more efficient coding, only a limited number of possible quantizations, which may be different for each subband, are allowed. Only the index with wordlength "nbal" in the relevant Annex B, Table 3-B.2., "LAYER II POSSIBLE QUANTZATIONS PER SUBBAND" is transmitted, MSB first.

10. Formatting

An overview of the Layer II format can be seen as follows:



The differences compared to the Layer I format are:
- The length of a slot equals 8 bits.
- A new block scfsi containing the scalefactor select information has been introduced.
- The bit allocation information, scalefactors and samples have been subject to further coding (see the related clauses).

The details can be found in the clause 2.4.1 of the main part of this audio standard, "SPECIFICATION OF THE CODED AUDIO BITSTREAM SYNTAX " .

TABLE 3-C.4: LAYER II Scalefactor transmission patterns

Class1	Class2	Transmission pattern	Select Information
1	1	1 2 3	0
1	2	1 2 2	3
1	3	1 2 2	3
1	4	1 3 3	3
1	5	1 2 3	0
2	1	1 1 3	1
2	2	1 1 1	2
2	3	1 1 1	2
2	4	4 4 4	2
2	5	1 1 3	1
3	1	1 1 1	2
3	2	1 1 1	2
3	3	1 1 1	2
3	4	3 3 3	2
3	5	1 1 3	1
4	1	2 2 2	2
4	2	2 2 2	2
4	3	2 2 2	2
4	4	3 3 3	2
4	5	1 2 3	0
5	1	1 2 3	0
5	2	1 2 2	3
5	3	1 2 2	3
5	4	1 3 3	3
5	5	1 2 3	0

TABLE 3-C.5: LAYER II SIGNAL-TO-NOISE RATIOS

No. of steps	SNR (dB)
0	0.00
3	7.00
5	11.00
7	16.00
9	20.84
15	25.28
31	31.59
63	37.75
127	43.84
255	49.89
511	55.93
1023	61.96
2047	67.98
4095	74.01
8191	80.03
16383	86.05
32767	92.01
65535	98.01

TABLE 3-C.6: LAYER II QUANTIZATION COEFFICIENTS

No. of steps	A	B
3	0.750000000	-0.250000000
5	0.625000000	-0.375000000
7	0.875000000	-0.125000000
9	0.562500000	-0.437500000
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
1023	0.999023438	-0.000976563
2047	0.999511719	-0.000488281
4095	0.999755859	-0.000244141
8191	0.999877930	-0.000122070
16383	0.999938965	-0.000061035
32767	0.999969482	-0.000030518
65535	0.999984741	-0.000015259

FIGURE 3-C.1 Analysis subband filter flow chart

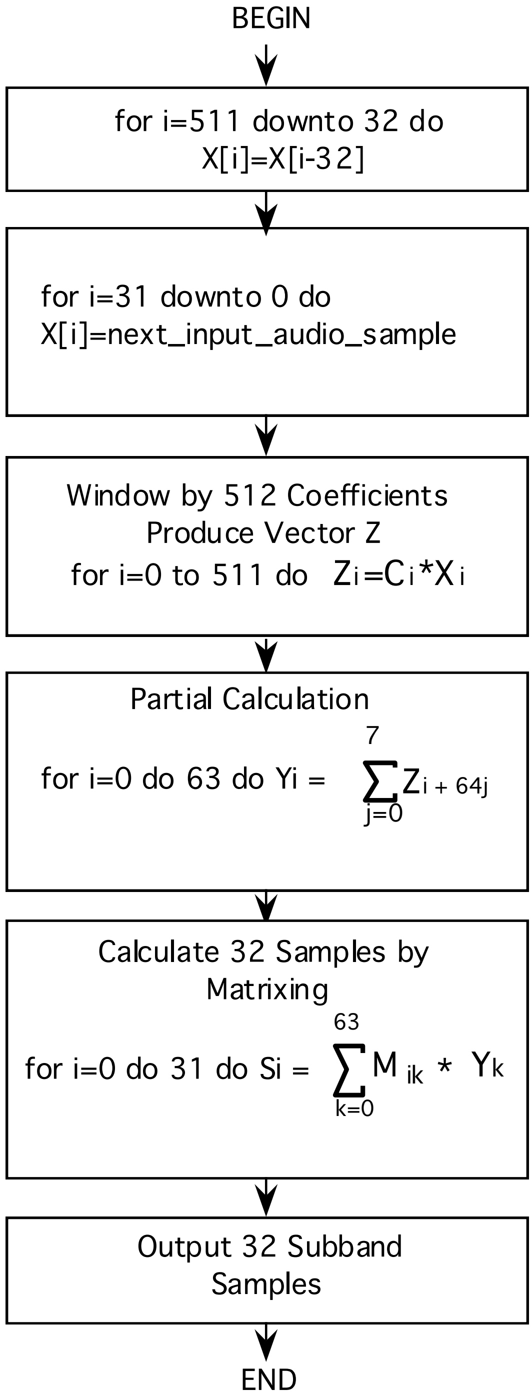
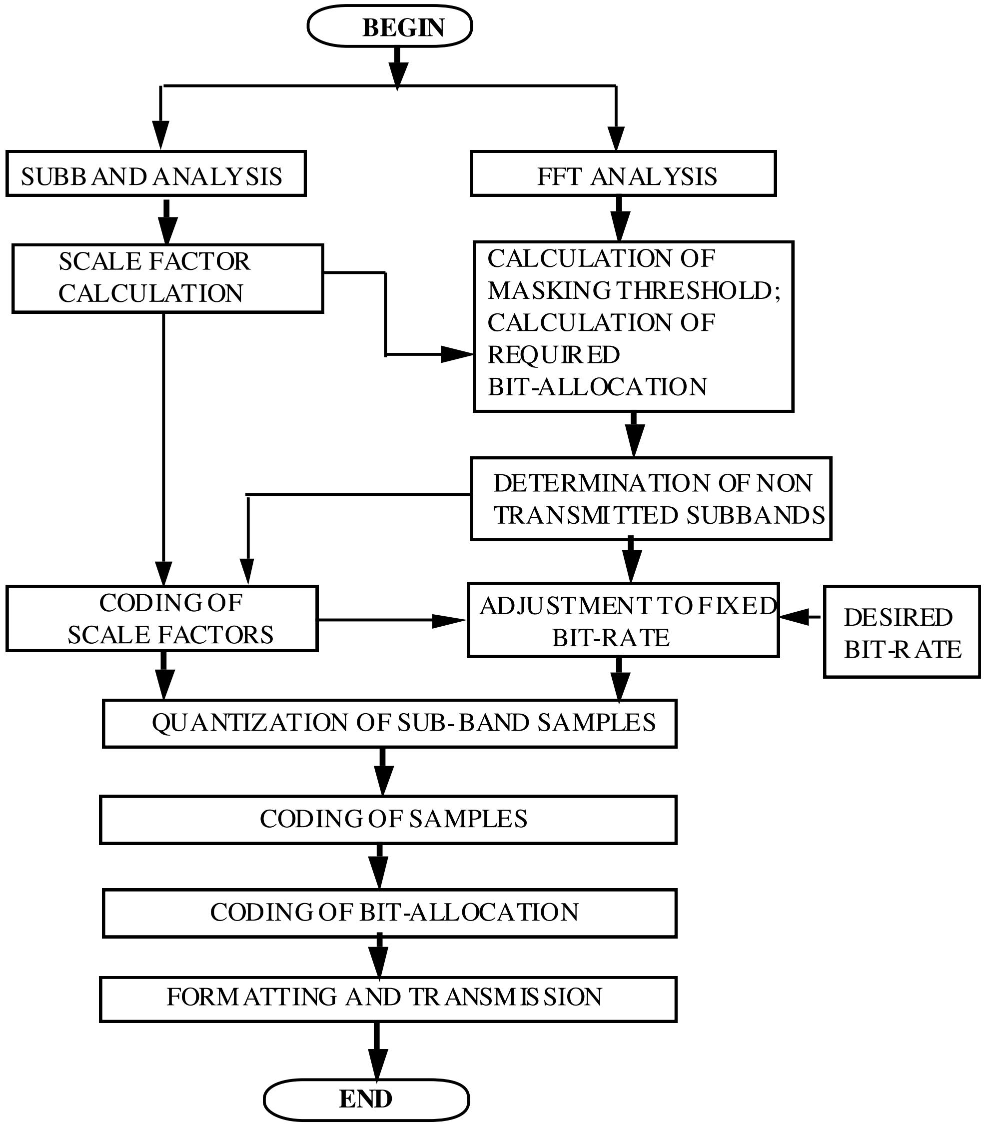


FIGURE 3-C.2 Layer I, II encoder flow chart



3-C.1.5.3 Layer III Encoding

1. Introduction

This clause describes a possible Layer III encoding method. The basic data flow is described by the general psychoacoustic coder block diagram. The basic blocks are described in more detail and below.

2. Psychoacoustic Model

The calculation of the psychoacoustic parameters can be done either with Psychoacoustic Model I described in Annex D, clause 3-D.1. or with Psychoacoustic Model II described in Annex D, clause 3-D.2. A description of modifications to Psychoacoustic Model II for use with Layer III can be found below. The model is run twice per block, using a shiftlength of 576 samples. A signal-to-mask-ratio is provided for every scale factor band

2.1. Adaptation of Psychoacoustic Model II for Layer III

For the use with Layer III encoding the psychoacoustic model 2 (Annex D, clause 3-D.2.) is modified as described below.

General Considerations:

The model is calculated twice in parallel. One computation is done with a shift length **iblen** of 192 samples (to be used with short blocks), the other is done with a shift length of 576 samples. For the shift length of 192 samples the block length of the FFT is changed to 256, and the parameters changed accordingly.

Change to Unpredictability Calculation:

The calculation of the unpredictability metric in Psychoacoustic Model II is changed.

- Calculation of the unpredictability
  The unpredictability cw is calculated for the first 206 spectral lines. For the other spectral lines, the unpredicatblility is set to 0.4.
  The unpredictability for the first 6 lines is calculated from the long FFT (window length = 1024, shiftlen = 576).
  For the spectral lines 6 upto 205 the unpredictability is calculated from the short FFT (window length 256, shiftlen = 192):

$$cw(w) = \begin{cases} cw\_l(w) & \text{for} \quad 0 = w < 6 \\ cw\_s(w/4) & \text{for} \quad 6 = w < 206,\ w=6,10,14,\ldots \\ 0.4 & \text{for} \quad w = 206 \end{cases}$$

  cw_l is the unpredictability calculated from the long FFT, cw_s is the unpredictability calculated from the second short block out of three short blocks within one granule.

- The spreading function has been replaced:

  If j = i tmpy= 3.0 (j -i)
  else tmpy= 1.5(j-i) is used.

  Only values of the spreading function greater than 10-6 are used. All other values are set to zero.

- For converting the unpredictability the parameters

  conv1 = -0.299
  conv2 = - 0.43

  are used.

- The parameter NMT (noise masking tone) is set to 6.0 db for all threshold calculation partions. The parameter TMN (tone masking noise) is set to 29.0 db for all partitions.
  For minval see table "threshold calculation partitions"

- The psychoacoustic entropy is estimated from the ratio thr/eb, where thr is the threshold and eb is the energy:

  pe = - ? (cbwidthk • log(thrk/(ebk+1.)))

  where k indexes the threshold calculation partitions and cbwidth is the width of the threshold calculation partition (see tables).

- pre-echo control

  The following constants are used for the control of pre-echo's (see block diagram):

rpelev = 2.
rpelev2 = 16.

- The threshold is not spread over the FFT lines. The threshold calculation partitions are converted directly to scalefactor bands. The first partition which is added to the scalefactor band is weighted with w1, the last with w2 (see table 3-Annex 3-C.8 "converting threshold calculation partitions to scalefactor bands"). The table contains also the number of partitions (cbw) converted to one scalefactor band (excluding the first and the last partition). The parameters bo and bu are shown in the table 3-Annex 3-C.8 used for converting threshold calculation partitions to scalefactor bands.

- For short blocks a simplified version of the threshold calculation (constant signal to noise ratio) is used. The constants can be found in the columns "SNR (dB)" in table 3-Annex 3-C.7. below.

Tables:

Table 3-C.7: Threshold calculation partitions with following parameters: width, minval, threshold in quiet, norm and bval:

Table 3-C.7a: Sampling_frequency = 48 kHz
long blocks

no.	FFT-lines		minval	qthr	norm	bval
0	1	24.5	4.532	0.970	0.000	
1	1	24.5	4.532	0.755	0.469	
2	1	24.5	4.532	0.738	0.937	
3	1	24.5	0.904	0.730	1.406	
4	1	24.5	0.904	0.724	1.875	
5	1	20	0.090	0.723	2.344	
6	1	20	0.090	0.723	2.812	
7	1	20	0.029	0.723	3.281	
8	1	20	0.029	0.718	3.750	
9	1	20	0.009	0.690	4.199	
10	1	20	0.009	0.660	4.625	
11	1	18	0.009	0.641	5.047	
12	1	18	0.009	0.600	5.437	
13	1	18	0.009	0.584	5.828	
14	1	12	0.009	0.531	6.187	
15	1	12	0.009	0.537	6.522	
16	2	6	0.018	0.857	7.174	
17	2	6	0.018	0.858	7.800	
18	2	3	0.018	0.853	8.402	
19	2	3	0.018	0.824	8.966	
20	2	3	0.018	0.778	9.483	
21	2	3	0.018	0.740	9.966	
22	2	0	0.018	0.709	10.426	
23	2	0	0.018	0.676	10.866	
24	2	0	0.018	0.632	11.279	
25	2	0	0.018	0.592	11.669	
26	2	0	0.018	0.553	12.042	
27	2	0	0.018	0.510	12.386	
28	2	0	0.018	0.513	12.721	
29	3	0	0.027	0.608	13.115	
30	3	0	0.027	0.673	13.561	
31	3	0	0.027	0.636	13.983	

32	3	0	0.027	0.586	14.371
33	3	0	0.027	0.571	14.741
34	4	0	0.036	0.616	15.140
35	4	0	0.036	0.640	15.562
36	4	0	0.036	0.597	15.962
37	4	0	0.036	0.538	16.324
38	4	0	0.036	0.512	16.665
39	5	0	0.045	0.528	17.020
40	5	0	0.045	0.516	17.373
41	5	0	0.045	0.493	17.708
42	6	0	0.054	0.499	18.045
43	7	0	0.063	0.525	18.398
44	7	0	0.063	0.541	18.762
45	8	0	0.072	0.528	19.120
46	8	0	0.072	0.510	19.466
47	8	0	0.072	0.506	19.807
48	10	0	0.180	0.525	20.159
49	10	0	0.180	0.536	20.522
50	10	0	0.180	0.518	20.873
51	13	0	0.372	0.501	21.214
52	13	0	0.372	0.496	21.553
53	14	0	0.400	0.497	21.892
54	18	0	1.628	0.495	22.231
55	18	0	1.628	0.494	22.569
56	20	0	1.808	0.497	22.909
57	25	0	22.607	0.494	23.248
58	25	0	22.607	0.487	23.583
59	35	0	31.650	0.483	23.915
60	67	0	605.867	0.482	24.246
61	67	0	605.867	0.524	24.576

Table 3.-C.7b: Sampling_frequency = 44.1 kHz
long blocks

no.	FFT-lines	minval	qthr	norm	bval
0	1	24.5	4.532	0.951	0.000
1	1	24.5	4.532	0.700	0.431
2	1	24.5	4.532	0.681	0.861
3	1	24.5	0.904	0.675	1.292
4	1	24.5	0.904	0.667	1.723
5	1	20	0.090	0.665	2.153
6	1	20	0.090	0.664	2.584
7	1	20	0.029	0.664	3.015
8	1	20	0.029	0.664	3.445
9	1	20	0.029	0.655	3.876
10	1	20	0.009	0.616	4.279
11	1	20	0.009	0.597	4.670
12	1	18	0.009	0.578	5.057
13	1	18	0.009	0.541	5.415
14	1	18	0.009	0.575	5.774
15	2	12	0.018	0.856	6.422
16	2	6	0.018	0.846	7.026
17	2	6	0.018	0.840	7.609
18	2	3	0.018	0.822	8.168
19	2	3	0.018	0.800	8.710
20	2	3	0.018	0.753	9.207

21	2	3	0.018	0.704	9.662
22	2	0	0.018	0.674	10.099
23	2	0	0.018	0.640	10.515
24	2	0	0.018	0.609	10.917
25	2	0	0.018	0.566	11.293
26	2	0	0.018	0.535	11.652
27	2	0	0.018	0.531	11.997
28	3	0	0.027	0.615	12.394
29	3	0	0.027	0.686	12.850
30	3	0	0.027	0.650	13.277
31	3	0	0.027	0.611	13.681
32	3	0	0.027	0.567	14.062
33	3	0	0.027	0.520	14.411
34	3	0	0.027	0.513	14.751
35	4	0	0.036	0.557	15.119
36	4	0	0.036	0.584	15.508
37	4	0	0.036	0.570	15.883
38	5	0	0.045	0.579	16.263
39	5	0	0.045	0.585	16.654
40	5	0	0.045	0.548	17.020
41	6	0	0.054	0.536	17.374
42	6	0	0.054	0.550	17.744
43	7	0	0.063	0.532	18.104
44	7	0	0.063	0.504	18.447
45	7	0	0.063	0.496	18.781
46	9	0	0.081	0.516	19.130
47	9	0	0.081	0.527	19.487
48	9	0	0.081	0.516	19.838
49	10	0	0.180	0.497	20.179
50	10	0	0.180	0.489	20.510
51	11	0	0.198	0.502	20.852
52	14	0	0.400	0.502	21.196
53	14	0	0.400	0.491	21.531
54	15	0	0.429	0.497	21.870
55	20	0	1.808	0.504	22.214
56	20	0	1.808	0.504	22.558
57	21	0	1.899	0.495	22.898
58	27	0	24.415	0.486	23.232
59	27	0	24.415	0.484	23.564
60	36	0	32.554	0.483	23.897
61	73	0	660.124	0.475	24.229
62	18	0	162.770	0.515	24.542

Table 3-C.7c: Sampling_frequency = 32 kHz
long blocks

no.	FFT-lines		minval	qthr	norm	bval
0	2	24.5	9.064	0.997	0.312	
1	2	24.5	9.064	0.893	0.937	
2	2	24.5	1.808	0.881	1.562	
3	2	20	0.181	0.873	2.187	
4	2	20	0.181	0.872	2.812	
5	2	20	0.057	0.871	3.437	
6	2	20	0.018	0.860	4.045	

7	2	20	0.018	0.839	4.625
8	2	18	0.018	0.812	5.173
9	2	18	0.018	0.784	5.698
10	2	12	0.018	0.741	6.184
11	2	12	0.018	0.697	6.634
12	2	6	0.018	0.674	7.070
13	2	6	0.018	0.651	7.492
14	2	6	0.018	0.633	7.905
15	2	3	0.018	0.611	8.305
16	2	3	0.018	0.589	8.695
17	2	3	0.018	0.575	9.064
18	3	3	0.027	0.654	9.483
19	3	3	0.027	0.724	9.966
20	3	0	0.027	0.701	10.425
21	3	0	0.027	0.673	10.866
22	3	0	0.027	0.631	11.279
23	3	0	0.027	0.592	11.669
24	3	0	0.027	0.553	12.042
25	3	0	0.027	0.510	12.386
26	3	0	0.027	0.505	12.721
27	4	0	0.036	0.562	13.091
28	4	0	0.036	0.598	13.488
29	4	0	0.036	0.589	13.873
30	5	0	0.045	0.607	14.268
31	5	0	0.045	0.620	14.679
32	5	0	0.045	0.580	15.067
33	5	0	0.045	0.532	15.424
34	5	0	0.045	0.517	15.771
35	6	0	0.054	0.517	16.120
36	6	0	0.054	0.509	16.466
37	6	0	0.054	0.506	16.807
38	8	0	0.072	0.522	17.158
39	8	0	0.072	0.531	17.518
40	8	0	0.072	0.519	17.869
41	10	0	0.090	0.512	18.215
42	10	0	0.090	0.509	18.562
43	10	0	0.090	0.497	18.902
44	12	0	0.108	0.494	19.239
45	12	0	0.108	0.501	19.579
46	13	0	0.117	0.507	19.925
47	14	0	0.252	0.502	20.269
48	14	0	0.252	0.493	20.606
49	16	0	0.289	0.497	20.944
50	20	0	0.572	0.506	21.288
51	20	0	0.572	0.510	21.635
52	23	0	0.658	0.504	21.979
53	27	0	2.441	0.496	22.319
54	27	0	2.441	0.493	22.656
55	32	0	2.894	0.490	22.993
56	37	0	33.458	0.483	23.326
57	37	0	33.458	0.458	23.656
58	12	0	10.851	0.500	23.937

Table 3-C.7d: Sampling_frequency = 48 kHz
short blocks

no.	FFT-lines		qthr	norm	SNR (db)	bval
0	1	4.532	0.970	-8.240	0.000	
1	1	0.904	0.755	-8.240	1.875	
2	1	0.029	0.738	-8.240	3.750	
3	1	0.009	0.730	-8.240	5.437	
4	1	0.009	0.724	-8.240	6.857	
5	1	0.009	0.723	-8.240	8.109	
6	1	0.009	0.723	-8.240	9.237	
7	1	0.009	0.723	-8.240	10.202	
8	1	0.009	0.718	-8.240	11.083	
9	1	0.009	0.690	-8.240	11.864	
10	1	0.009	0.660	-7.447	12.553	
11	1	0.009	0.641	-7.447	13.195	
12	1	0.009	0.600	-7.447	13.781	
13	1	0.009	0.584	-7.447	14.309	
14	1	0.009	0.532	-7.447	14.803	
15	1	0.009	0.537	-7.447	15.250	
16	1	0.009	0.857	-7.447	15.667	
17	1	0.009	0.858	-7.447	16.068	
18	1	0.009	0.853	-7.447	16.409	
19	2	0.018	0.824	-7.447	17.044	
20	2	0.018	0.778	-6.990	17.607	
21	2	0.018	0.740	-6.990	18.097	
22	2	0.018	0.709	-6.990	18.528	
23	2	0.018	0.676	-6.990	18.930	
24	2	0.018	0.632	-6.990	19.295	
25	2	0.018	0.592	-6.990	19.636	
26	3	0.054	0.553	-6.990	20.038	
27	3	0.054	0.510	-6.990	20.486	
28	3	0.054	0.513	-6.990	20.900	
29	4	0.114	0.608	-6.990	21.305	
30	4	0.114	0.673	-6.020	21.722	
31	5	0.452	0.637	-6.020	22.128	
32	5	0.452	0.586	-6.020	22.512	
33	5	0.452	0.571	-6.020	22.877	
34	7	6.330	0.616	-5.229	23.241	
35	7	6.330	0.640	-5.229	23.616	
36	11	9.947	0.597	-5.229	23.974	
37	17	153.727	0.538	-5.229	24.312	

Table 3-C.7e: Sampling_frequency = 44.1 kHz
short blocks

no.	FFT-lines		qthr	norm	SNR (db)	bval
0	1	4.532	0.952	-8.240	0.000	
1	1	0.904	0.700	-8.240	1.723	
2	1	0.029	0.681	-8.240	3.445	
3	1	0.009	0.675	-8.240	5.057	
4	1	0.009	0.667	-8.240	6.422	
5	1	0.009	0.665	-8.240	7.609	
6	1	0.009	0.664	-8.240	8.710	

7	1	0.009	0.664	-8.240	9.662
8	1	0.009	0.664	-8.240	10.515
9	1	0.009	0.655	-8.240	11.293
10	1	0.009	0.616	-7.447	12.009
11	1	0.009	0.597	-7.447	12.625
12	1	0.009	0.578	-7.447	13.210
13	1	0.009	0.541	-7.447	13.748
14	1	0.009	0.575	-7.447	14.241
15	1	0.009	0.856	-7.447	14.695
16	1	0.009	0.846	-7.447	15.125
17	1	0.009	0.840	-7.447	15.508
18	1	0.009	0.822	-7.447	15.891
19	2	0.018	0.800	-7.447	16.537
20	2	0.018	0.753	-6.990	17.112
21	2	0.018	0.704	-6.990	17.620
22	2	0.018	0.674	-6.990	18.073
23	2	0.018	0.640	-6.990	18.470
24	2	0.018	0.609	-6.990	18.849
25	3	0.027	0.566	-6.990	19.271
26	3	0.027	0.535	-6.990	19.741
27	3	0.054	0.531	-6.990	20.177
28	3	0.054	0.615	-6.990	20.576
29	3	0.054	0.686	-6.990	20.950
30	4	0.114	0.650	-6.020	21.316
31	4	0.114	0.612	-6.020	21.699
32	5	0.452	0.567	-6.020	22.078
33	5	0.452	0.520	-6.020	22.438
34	5	0.452	0.513	-5.229	22.782
35	7	6.330	0.557	-5.229	23.133
36	7	6.330	0.584	-5.229	23.484
37	7	6.330	0.570	-5.229	23.828
38	19	171.813	0.578	-4.559	24.173

Table 3-C.7f: Sampling_frequency = 32 kHz
short blocks

no.	FFT-lines		qthr	norm	SNR (db)	bval
0	1	4.532	0.997	-8.240	0.000	
1	1	0.904	0.893	-8.240	1.250	
2	1	0.090	0.881	-8.240	2.500	
3	1	0.029	0.873	-8.240	3.750	
4	1	0.009	0.872	-8.240	4.909	
5	1	0.009	0.871	-8.240	5.958	
6	1	0.009	0.860	-8.240	6.857	
7	1	0.009	0.839	-8.240	7.700	
8	1	0.009	0.812	-8.240	8.500	
9	1	0.009	0.784	-8.240	9.237	
10	1	0.009	0.741	-7.447	9.895	
11	1	0.009	0.697	-7.447	10.500	
12	1	0.009	0.674	-7.447	11.083	
13	1	0.009	0.651	-7.447	11.604	
14	1	0.009	0.633	-7.447	12.107	
15	1	0.009	0.611	-7.447	12.554	
16	1	0.009	0.589	-7.447	13.000	

17	1	0.009	0.575	-7.447	13.391
18	1	0.009	0.654	-7.447	13.781
19	2	0.018	0.724	-7.447	14.474
20	2	0.018	0.701	-6.990	15.096
21	2	0.018	0.673	-6.990	15.667
22	2	0.018	0.631	-6.990	16.177
23	2	0.018	0.592	-6.990	16.636
24	2	0.018	0.553	-6.990	17.057
25	2	0.018	0.510	-6.990	17.429
26	2	0.018	0.506	-6.990	17.786
27	3	0.027	0.562	-6.990	18.177
28	3	0.027	0.598	-6.990	18.597
29	3	0.027	0.589	-6.990	18.994
30	3	0.027	0.607	-6.020	19.352
31	3	0.027	0.620	-6.020	19.693
32	4	0.072	0.580	-6.020	20.066
33	4	0.072	0.532	-6.020	20.461
34	4	0.072	0.517	-5.229	20.841
35	5	0.143	0.517	-5.229	21.201
36	5	0.143	0.509	-5.229	21.549
37	6	0.172	0.506	-5.229	21.911
38	7	0.633	0.522	-4.559	22.275
39	7	0.633	0.531	-4.559	22.625
40	8	0.723	0.519	-3.980	22.971
41	10	9.043	0.512	-3.980	23.321

Table 3-C.8: Tables for converting threshold calculation partitions to scalefactor bands

Table 3-C.8a: Sampling_frequency = 48 kHz
long blocks

no. sb	**cbw**	**bu**	**bo**	**w1**	**w2**
0	3	0	4	1.000	0.056
1	3	4	7	0.944	0.611
2	4	7	11	0.389	0.167
3	3	11	14	0.833	0.722
4	3	14	17	0.278	0.639
5	2	17	19	0.361	0.417
6	3	19	22	0.583	0.083
7	2	22	24	0.917	0.750
8	3	24	27	0.250	0.417
9	3	27	30	0.583	0.648
10	3	30	33	0.352	0.611
11	3	33	36	0.389	0.625
12	4	36	40	0.375	0.144
13	3	40	43	0.856	0.389
14	3	43	46	0.611	0.160
15	3	46	49	0.840	0.217
16	3	49	52	0.783	0.184
17	2	52	54	0.816	0.886
18	3	54	57	0.114	0.313
19	2	57	59	0.687	0.452
20	1	59	60	0.548	0.908

Table 3-C.8b: Sampling_frequency = 44.1 kHz long blocks

no. sb	cbw	bu	bo	w1	w2
0	3	0	4	1.000	0.056
1	3	4	7	0.944	0.611
2	4	7	11	0.389	0.167
3	3	11	14	0.833	0.722
4	3	14	17	0.278	0.139
5	1	17	18	0.861	0.917
6	3	18	21	0.083	0.583
7	3	21	24	0.417	0.250
8	3	24	27	0.750	0.805
9	3	27	30	0.194	0.574
10	3	30	33	0.426	0.537
11	3	33	36	0.463	0.819
12	4	36	40	0.180	0.100
13	3	40	43	0.900	0.468
14	3	43	46	0.532	0.623
15	3	46	49	0.376	0.450
16	3	49	52	0.550	0.552
17	3	52	55	0.448	0.403
18	2	55	57	0.597	0.643
19	2	57	59	0.357	0.722
20	2	59	61	0.278	0.960

Table 3-C.8c: Sampling_frequency = 32 kHz long blocks

no. sb	cbw	bu	bo	w1	w2
0	1	0	2	1.000	0.528
1	2	2	4	0.472	0.305
2	2	4	6	0.694	0.083
3	1	6	7	0.917	0.861
4	2	7	9	0.139	0.639
5	2	9	11	0.361	0.417
6	3	11	14	0.583	0.083
7	2	14	16	0.917	0.750
8	3	16	19	0.250	0.870
9	3	19	22	0.130	0.833
10	4	22	26	0.167	0.389
11	4	26	30	0.611	0.478
12	4	30	34	0.522	0.033
13	3	34	37	0.967	0.917
14	4	37	41	0.083	0.617
15	3	41	44	0.383	0.995
16	4	44	48	0.005	0.274
17	3	48	51	0.726	0.480
18	3	51	54	0.519	0.261
19	2	54	56	0.739	0.884
20	2	56	58	0.116	1.000

Table 3-C.8d: Sampling_frequency = 48 kHz short blocks

no. sb	cbw	bu	bo	w1	w2

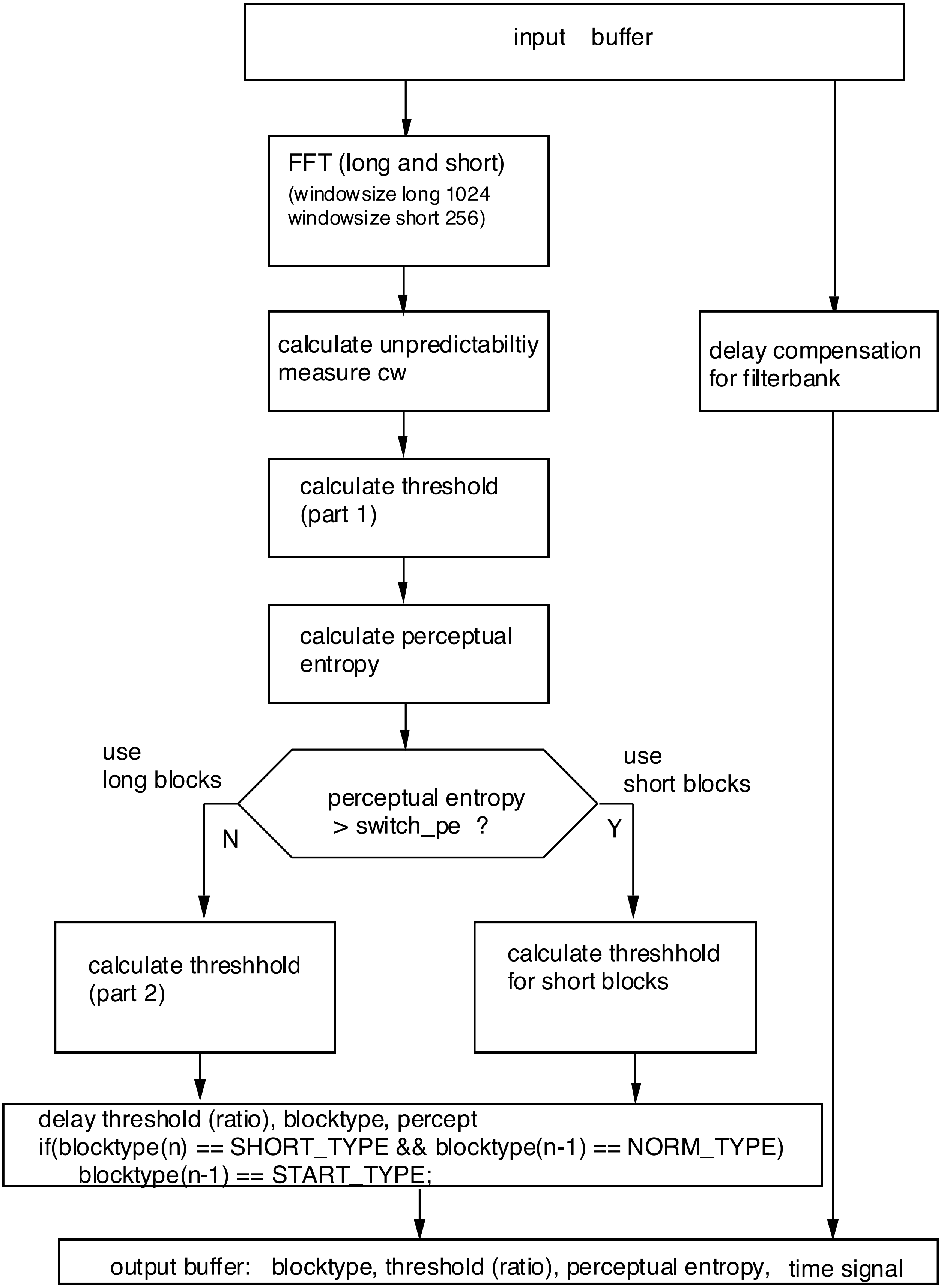
0	2	0	3	1.000	0.167
1	2	3	5	0.833	0.833
2	3	5	8	0.167	0.500
3	3	8	11	0.500	0.167
4	4	11	15	0.833	0.167
5	4	15	19	0.833	0.583
6	3	19	22	0.417	0.917
7	4	22	26	0.083	0.944
8	4	26	30	0.055	0.042
9	2	30	32	0.958	0.567
10	3	32	35	0.433	0.167
11	2	35	37	0.833	0.618

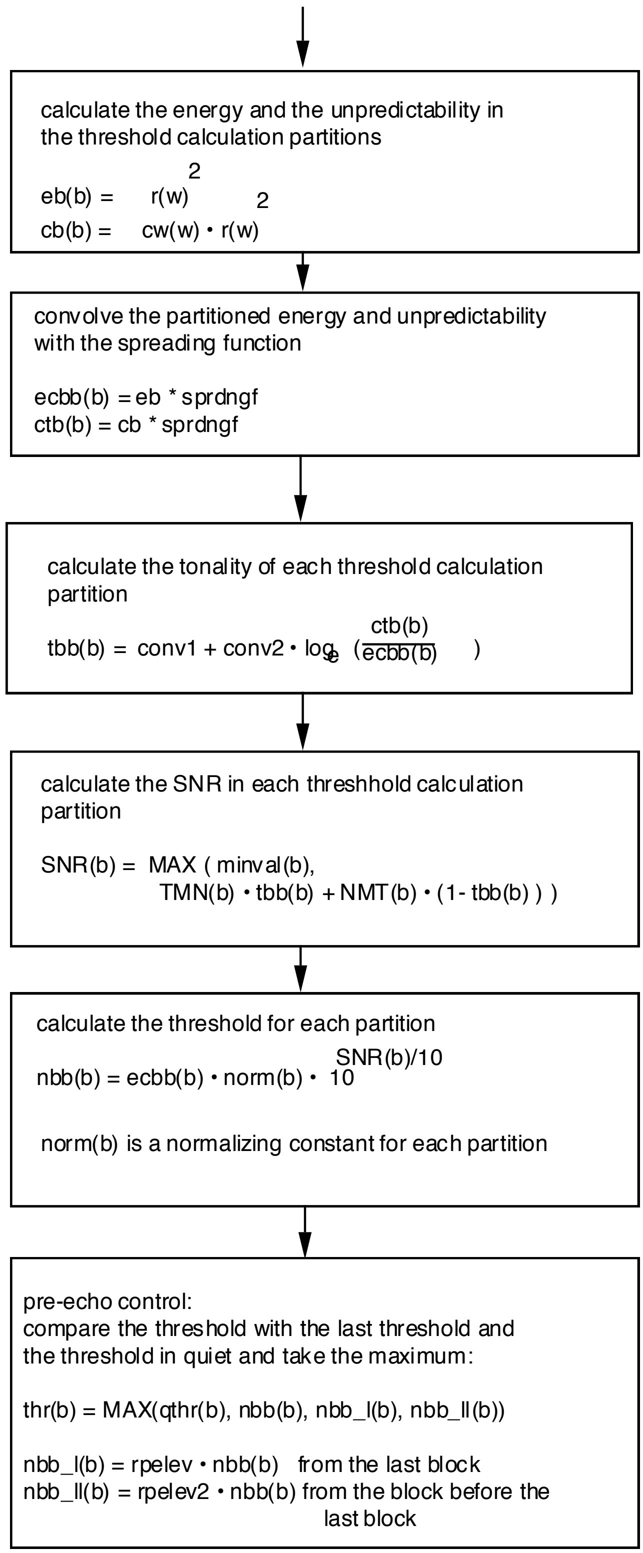
Table 3-C.8e: Sampling_frequency = 44.1 kHz
short blocks

no. sb	cbw	bu	bo	w1	w2
0	2	0	3	1.000	0.167
1	2	3	5	0.833	0.833
2	3	5	8	0.167	0.500
3	3	8	11	0.500	0.167
4	4	11	15	0.833	0.167
5	5	15	20	0.833	0.250
6	3	20	23	0.750	0.583
7	4	23	27	0.417	0.055
8	3	27	30	0.944	0.375
9	3	30	33	0.625	0.300
10	3	33	36	0.700	0.167
11	2	36	38	0.833	1.000

Table 3-C.8f: Sampling_frequency = 32 kHz
short blocks

no. sb	cbw	bu	bo	w1	w2
0	2	0	3	1.000	0.167
1	2	3	5	0.833	0.833
2	3	5	8	0.167	0.500
3	3	8	11	0.500	0.167
4	4	11	15	0.833	0.167
5	5	15	20	0.833	0.250
6	4	20	24	0.750	0.250
7	5	24	29	0.750	0.055
8	4	29	33	0.944	0.375
9	4	33	37	0.625	0.472
10	3	37	40	0.528	0.937
11	1	40	41	0.062	1.000


input buffer
FFT (long and short)
(windowsize long 1024
windowsize short 256)
calculate unpredictabiltiy
measure cw
calculate threshold
(part 1)
calculate perceptual
entropy
delay compensation
for filterbank
use
long blocks
perceptual entropy
> switch_pe ?
use
short blocks
N
Y
calculate threshhold
(part 2)
calculate threshhold
for short blocks
delay threshold (ratio), blocktype, percept
if(blocktype(n) == SHORT_TYPE && blocktype(n-1) == NORM_TYPE)
blocktype(n-1) == START_TYPE;
output buffer: blocktype, threshold (ratio), perceptual entropy, time signal



Block diagram psychoacoustic model II, layer III: calculate threshold (part 1) convert from threshold calculation partitions to codec partitions (scalefactor bands):

energy in each scalefactor band:

$$en(sb) = w1 \cdot eb(bu) + \sum_{b=bu+1}^{b=bo-1} eb(b) + w2 \cdot eb(bo)$$

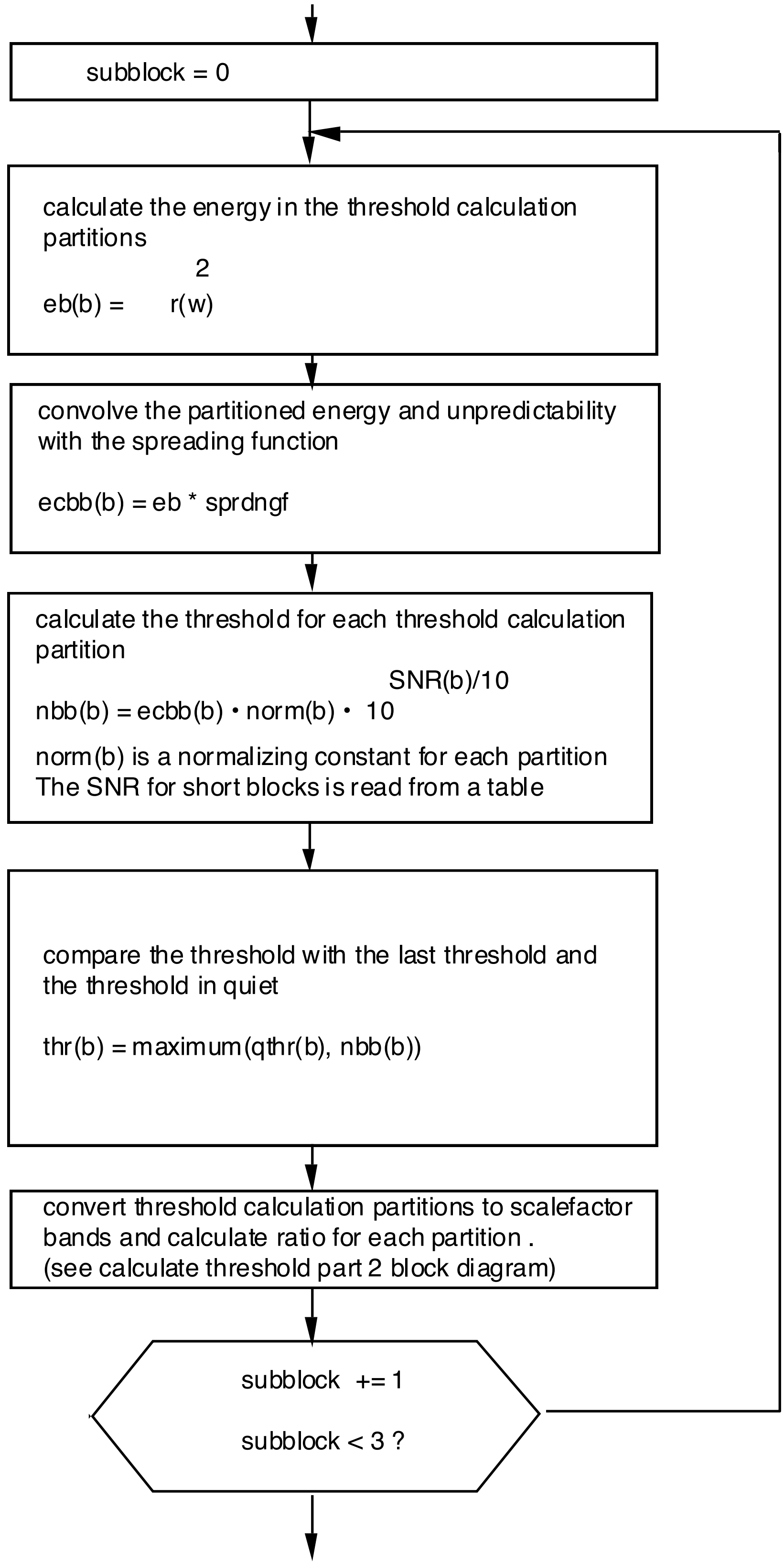
threshold in each scalefactor band:

$$thrn(sb) = w1 \cdot thr(bu) + \sum_{b=bu+1}^{b=bo-1} thr(b) + w2 \cdot thr(bo)$$

Calculate ratios for each scalefactor band:

$$ratio(sb) = \frac{thrn(sb)}{en(sb)}$$

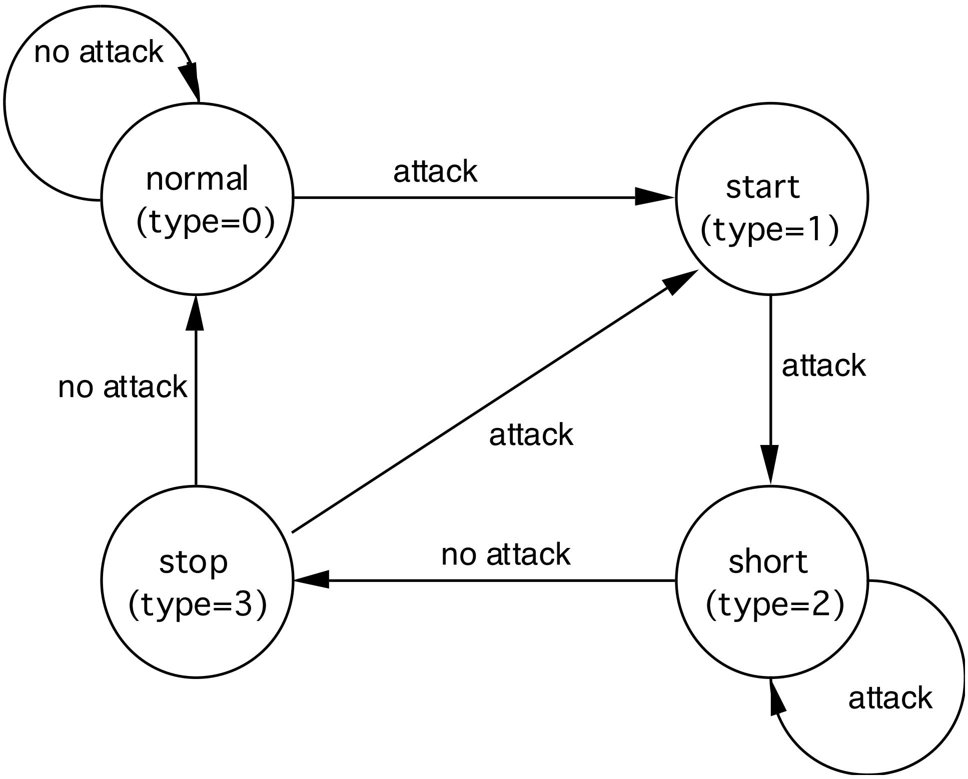
Block diagram psychoacoustic model II, layer III: calculate threshold (part 2)



Block diagram psychoacoustic model II, layer III: calculate threshold for short blocks

Window switching decision:

The decision whether the filterbank should be switched to short windows is derived from the calculation of the masking threshold by calculating the estimate of the psychoacoustic entropy (PE) and switching when the PE exceeds the value 1800. If this condition is met, the sequence start (block_type=1), short (block_type=2), short, stop (block_type=3) is started. The figure below shows the possible state changes for the window switching logic.



3. Analysis Part of the Hybrid Filterbank

The subband analysis of the polyphase filterbank is described in clause 3-C.1.3, "SUBBAND ANALYSIS FILTER". The output of the polyphase filterbank is the input to the subdivision using the MDCT. According to the output of the psychoacoustic model (variables **blocksplit_flag** and **block_type**) the window and transform types **normal**, **start**, **short** or **stop** are used.

18 consecutive output values of one granule and 18 output values of the granule before are assembled to one block of 36 samples.

Block type "**normal**"

$$z_i = x'_i \sin\left(\frac{\pi}{36}\left(i + \frac{1}{2}\right)\right) \qquad \text{for i=0 to 35}$$

Block type "**start**"

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

Block type "**stop**"

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

Block type "**short"**

The block of 36 samples is divided into three overlapping blocks:

$$y_i^{(0)} = x'_{i+6} \quad \text{for } i=0 \text{ to } 11$$

$$y_i^{(1)} = x'_{i+12} \quad \text{for } i=0 \text{ to } 11$$

$$y_i^{(2)} = x'_{i+18} \quad \text{for } i=0 \text{ to } 11$$

Each of the three small blocks is windowed separately:

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

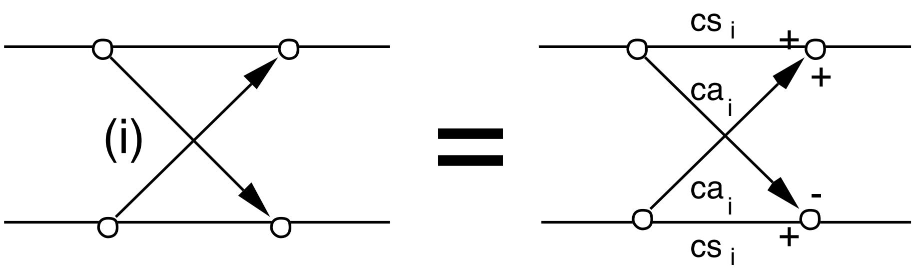
MDCT:

In the following n is the number of windowed samples. For short blocks n is 12, for long blocks n is 36. The analytical expression of the MDCT is:

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

Aliasing-Butterfly, Encoder:

The calculation of aliasing reduction in the encoder is performed as in the decoder. The general procedure is shown in Fig. 3-Annex 3-A.5. The butterfly definition to be used in the encoder is shown below. The coefficients cai and csi can be found in 3-Annex Table 3-B.9



4. Calculation of average available bits

The average number of bits per granule is calculated from the frame size. The bitrate 64 kb/second is used as an example. At bitrate 64 kb/second at 48000 samples per second,

(64000 * 0.24 bits per frame) / (2 granules per frame) = 768 bits per granule

As the header takes 32 bits and the side information takes 17 bytes (136 bits) in single_channel mode, the average amount of available bits for the main_data for a granule is given by

mean_bits = 768 bits per granule - (32+136 bits per frame)/(2 granules per frame) = 684 bits per granule

Bit reservoir:

The bit reservoir can provide additional bits which may be used for the granule. The number of bits which are provided is determined within the iteration loops.

5. Quantization and Encoding of Frequency Domain Samples

The frequency domain data are quantized and coded within two nested iteration loops. Chapter 3-C1.5.4 contains a detailed description of these iteration loops.

6. Formatting

The details about the Layer III bitstream format can be found in the clause 2.4.4 of the main part of this audio standard, "SPECIFICATION OF THE CODED AUDIO BITSTREAM SYNTAX " . The formatting of the Huffman code words is described below:

The Huffman codewords are in sequence from low to high frequencies. In the iteration loops the following variables have been calculated and are used in encoding the Huffman codewords:

is(i), i=0...575	quantized frequency domain values
table_select[region]	Huffman code table used for regions (region = 0, 1, 2)
region_adress1	defines the border between region 0 and 1
region_adress2	defines the border between region 1 and 2
max_value[region]	maximum absolute value of quantized data in regions (region = 0, 1, 2)

The data are written to the bitstream according to the Huffman code syntax described in clause 2.4.2.7

The actual assembly of the Huffman code for the big_values part is described in a pseudo high level language:

```
for region number from 0 to 2
  if table_select for this region is 0
```

```
  nothing to do, all values in region are zero
 else
  if table_select for this region is > 15
   an ESC-table is used: look up linbits value connected to the table used
   for i = begin of region to end of region, count in pairs
    x = is(i), y = is(i+1)
    if x > 14
     linbitsx = x - 15, x = 15
    end if
    signx = sign(x), x = abs(x)
    if y > 14
     linbitsy = y - 15, y = 15
    end if
    signy = sign(y), y = abs(y)
    look for codeword = hcod([x][y]]) in table table_seletct
    write hcod([x][y]), beginning with the leftmost bit, number of bits is hlen([x][y])
    if x > 14
     write linbitsx to the bitstream, number of bits is linbits
    end if
    if x != 0
     write signx to bitstream
    end if
    if y > 14
     write linbitsy to the bitstream, number of bits is linbits
    end if
    if y != 0
     write signy to bitstream
    end if
   end do
  else
   no ESC-words are used in this region:
   for i = beginning of region to end of region, count in pairs
    x = is(i), y = is(i+1)
   signx = sign(x), x = abs(x)
   signy = sign(y), y = abs(y)
    look for codeword = hcod([x][y]) in table table_seletct
    write hcod([x][y]), beginning with the leftmost bit, number of bits is hlen([x][y])
   if x != 0
     write signx to bitstream
    end if
    if y != 0
     write signy to bitstream
    end if
   end do
  end if
 end if
end for
```

A possible application for the private_bits is to use them as frame counter.

3-C.1.5.4 Layer III Iteration Loops

1. Introduction

The description of the Layer III loop module is subdivided into three levels. The top level is called "loops frame program". The loops frame program calls a subroutine named "outer iteraton loop" which calls the subroutine "inner iteration loop". For each level a corresponding flow diagram is shown.

The loops module quantizes an input vector of spectral data in an iterative process according to several demands. The inner loop quantizes the input vector and increases the quantizer step size until the output vector can be coded with the available amount of bit. After completion of the inner loop an outer loop checks the distortion of each scalefactor band and, if the allowed distortion is exceeded, amplifies the scalefactor band and calls the inner loop again.

Layer III loops module input:
(1) vector of the magnitudes of the spectral values xr(0..575)
(2) xmin(cb), the allowed distortion of the scalefactor bands
(3) blocksplit_flag which in conjunction with switch_point determines the number of scalefactor bands
(4) mean_bits (bit available for the Huffman coding and the coding of the scalefactors)
(5) more_bits, the number of bits in addition to the average number of bits, as demanded by the value of the psychoacoustic entropy for the granule:
more_bits = 3.1 * PE - (average number of bits)

Layer III loops module output:
(1) vector of quantized values ix(0..575)
(2) ifq(cb), the scalefactors
(3) qquant (quantizer step size information)
(4) number of unused bit available for later use
(5) preflag (loops preemphasis on/off)
(6) Huffman code related side information
- big_values (number of pairs of Huffman coded values, excluding "count1")
- count1table_select (Huffman code table of absolut values <= 1 at the upper end of the spectrum
- table_select[0..2](Huffman code table of regions)
- region_address1,2 (used to calculate boundaries between regions)
- part2_3_length

2. Preparatory Steps

2.1 Reset of all iteration variables

The scalefactors of the coder partitions scalefac[cb] are set to zero.

The counter qquant for the quantizer step size is reset to zero.

Preflag is reset to zero.

Scalefac_scale is reset to zero.

The inital value of quantanf is set as follows:

quantanf = system_const * loge(sfm),

where sfm is the spectral flatness measure and quantanf depends on the computational implementation of the encoder.

The spectral flatness measure sfm is given by

$$sfm = \frac{e^{\frac{1}{n}\left(\sum_{i=0}^{n-1} \log xr(i)^2\right)}}{\frac{1}{n}\sum_{i=0}^{n-1} xr(i)^2}$$

The value of system_const is chosen so that for all signals the first iteration of the inner loop for all signals comes out with a bit sum higher than the desired bitsum. By that it is ensured that the first call of the inner loop results in the solution which uses as many of the available bits as possible. In order to spare computing time it is desirable to minimize the number of iterations by adapting the value of quantanf to the bitrate and the signal statistics.

2.2 Bit reservoir control

Bits are saved to the reservoir when fewer than the mean_bits are used to code one granule. If bits are saved for a frame, the value of main_data_end is increased accordingly. See diagram 3-Annex 3-A.7.1.

The number of bits which are made available for the main_data (called "max_bits") is derived from the actual estimated threshold (the PE as calculated by the psychoacoustic model), the average number of bits (mean_bits) and the actual content of the bit reservoir. The number of bytes in the bit reservoir is given by main_data_end.

The actual rules for the control of the bit reservoir are given below:

- If a number of bytes available to the inner iteration loop is not used for the Huffman encoding or other main_data, the number is added to the bit reservoir.
- If the bit reservoir contains more than 0.8 times the maximum allowed content of the bit reservoir, all bytes exceeding this number are made available for main_data (in addition to mean_bits)
- If more_bits is greater than 100 bits, then max(more_bits/8, 0.6*main_data_end) bytes are taken from the bit reservoir and made available for main_data (in addition to mean_bits).
- After the actual loops computations have been completed, the number of bytes not used for main_data is added to the bit reservoir.
- If after the step above the number of bytes in the bit reservoir exceeds the maximum allowed content, stuffing bits are written to the bitstream and the content of the bit reservoir is adjusted accordingly.

2.3 Calculation of the scalefactor select information (scfsi)

The scfsi contains the information, which scalefactors (grouped in the scfsi_bands) of the first granule can also be used for the second granule. These scalefactors are therefore not transmitted, the gained bits can be used for the huffman coding.
To determine the usage of the scfsi, the following information of each granule must be stored:

1. The block type
2. The total energy of the granule:

$$en\_tot = int\left\{ \log 2\left(\sum_{i=1}^{n} | xr(i)|^2 \right) \right\}$$

where n is the total number of spectral values

3. The energy of each scalefactor band:

$$en(cb) = int\left\{ \log 2\left(\sum_{i=lbl(cb)}^{lbl(cb)+bw(cb)-1} |xr(i)|^2 \right) \right\}$$

where lbl(cb) is the number of the first coefficient belonging to scalefactor band cb and bw(cb) is the number of coefficients within scalefactor band cb

4. The allowed distortion of each scalefactor band:

$$xm(cr.bd) = int\{log2(xmin(i))\}$$

xmin(cb) is calculated by the psychoacoustic model.

The scalefactors of the first granule are always transmitted. When coding the second granule, the information of the two granules is compared. There are four criteria to determine if the scfsi can be used in general. If one of the four is not fulfilled, the scfsi is disabled (that means it is set to 0 in all scfsi_bands). The criteria are (index *0* means first, index *1* second granule):

1. The spectral values are not all zero
2. None of the granules contains short blocks
3. $$\left| en\_tot_0 - en\_tot_1 \right| < en\_tot_{krit}$$
4. $$\sum_{\text{all scalefactor bands}} \left| en(cb)_0 - en(cb)_1 \right| < en\_dif_{krit}$$

If the scfsi is not disabled after the tests above, there are two criterias for each scfsi_band, which have both to be fulfilled to enable scfsi (that means to set it to 1 in this scfsi_band):

1. $$\sum_{\text{all cr.bds in scfsi\_band}} \left| en(cr.bd)_0 - en(cr.bd)_1 \right| < en(scfsi\_band)_{krit}$$

2. $$\sum_{\text{all cr.bds in scfsi\_band}} \left| xm(cr.bd)_0 - xm(cr.bd)_1 \right| < xm(scfsi\_band)_{krit}$$

The constants (with the index *krit*) have to be chosen so, that the scfsi is only enabled in case of similar energy/distortion.

Suggested values are:

en_tot	=	10	
en_dif	=	100	
en(scfsi_band)	=	10	for each scfsi_band
xm(scfsi_band)	=	10	for each scfsi_band

3. Outer Iteration Loop (distortion control loop)

The outer iteration loop controls the quantization noise which is produced by the quantization of the frequency domain lines within the inner iteration loop. The colouration of the noise is done by multiplikation of the lines within scalefactor bands with the actual scalefactors before doing the quantization. The following pseudo-code illustrates the multiplication.

```
do for each scalefactor band:
  do from lower index to upper index of scale factor band
    xr(i) = xr(i) * sqrt(2) ^ ((1 + scalefac_scale) * ifq(scalefactor band))
  end do
end do
```

In the actual system the multiplication is done incrementally with just the increase of the scalefactors applied in each distortion control loop. This is described in clause 3.5 below.

The distortion loop is always starting with scalefac_scale = 0. If after some iterations the maximum length of the scalefactors would be exceeded (see scalefac_compress table in 2.4.2.7 and 3.5 below), then scalefac_scale is increased to the value 1 thus increasing the possible dynamic range of the scalefactors. In this case the actual scalefactors and frequency lines have to be corrected accordingly.

3.1 Saving of the scalefactors

The scalefactors of all scalefactor bands ifq(cb) as well as the quantizer step size qquant are saved. If the computation of the outer loop is cancelled without having reached a proper result this values together with the quantized spectrum give an approximation and can be transmitted.

3.2 Call of inner iteration loop

For each outer iteration loop (distortion control loop) the inner iteration loop (rate control loop) is called. The parameters are the frequency domain values (hybrid filterbank output) with the scalefactors applied to the values within the scalefactor bands and the number of bits which are available to the rate control loop. The result is the number of bits actually used and the quantized frequency lines ix(i).

3.3 Calculation of the distortion of the scalefactor bands

For each scalefactor band the actual distortion is calculated according to:

$$xfsf(cr.bd.) = \sum_{lbl(cr.bd.)}^{i=lbl(cr.bd.)+bw(cr.bd.)-1} \frac{\left(|xr(i)| - ix(i)^{4/3} * \sqrt[4]{2}^{\,qquant+quantanf}\right)^2}{bandwidth(cr.bd.)}$$

where lbl(cb) is the number of the coefficient representing the lowest frequency in a scalefactor band and bw(cb) is the number of coefficients within this band.

3.4 Preemphasis

The preemphasis option (switched on by setting preflag to a value of 1) provides the possibility to amplifiy the upper part of the spectrum according to the preemphasis tables, B.6 in the annex.

```
if preflag==1
{
        xmin(j) = xmin(j) *ifqstep2*prefact(j)
        for (i=lower limit of scalefactor band j; i <=upper limit of scalefactor band j; i++)  {
                xr(i) = xr(i) * ifqstepprefact(j)
        }
}
```

The condition to switch on the preemphasis is up to the implementation. For example preemphasis could be switched on if in all of the upper 4 scalefactor bands the actual distortion exceeds the threshold after the first call of the inner loop.
If the second granule is being coded and scfsi is active in at least one scfsi_band, the preemphasis in the second granule is set equal to the setting in first granule.

3.5 Amplification of scalefactor bands which violate the masking threshold

All spectral values of the scalefactor bands which have a distortion that exceeds the allowed distortion are amplified by a factor of ifqstep. The value of ifqstep is transmitted by scalefac_scale.

```
if (xmin - xfsf) of scalefactor band j  <  0
{
        xmin(j) = xmin(j) * ifqstep2
        ifq(j) = ifq(j) + 1
        for (i=lower limit of scalefactor band; i <=upper limit of scalefactor band; i++)  {
                xr(i) = xr(i) * ifqstep
        }
}
```

If the second granule is being coded and scfsi is active in at least one scfsi_band, the following steps have to be done:

1. ifqstep has to be set similar to the first granule
2. If it is the first iteration, the scalefactors of scalefactor bands in which scfsi is enabled have to be taken over from the first granule. The corresponding spectral values have to be amplified:
   if (scfsi according to scalefactor band j = 1)
   {
   ifq(j) = ifq(j)first granule
   for (i=lower limit of scalefactor band; i <=upper limit of scalefactor band; i++)
   { xr(i) = xr(i) * ifqstepifq(j) }
   }
3. If it is not the first iteration, the amplification must be prevented for scalefactor bands in which scfsi is enabled.

3.5 Conditions for the termination of the loops processing

Normally the loops processing terminates if there is no scalefactor band with more than the allowed distortion. However this is not always possible to obtain. In this case there are other conditions to terminate the outer loop. If

a) all scalefactor bands are already amplified
b) the amplification of at least one band exceeds the upper limit which is determined by the transmission format of the scalefactors. The upper limit is a scalefactor of 15 for scalefactor bands 0 through 10 and 7 for scalefactors 11 through 20.

the loops processing stops and by restoring the saved ifq(cb.) a useful output is available. For realtime implementation there might be a third condition added which terminates the loops in case of a lack of computing time.

4. Inner Iteration Loop (rate control loop)

The inner iteration loop does the actual quantization of the frequency domain data and prepares the formatting. The table selection, subdivision of the big_values range into regions and the selection of the quantizer step size takes place here.

4.1 Quantization

The quantization of the complete vector of spectral values is done according to

$$ix(i) = \mathrm{nint}\left(\left(\frac{|xr(i)|}{\sqrt[4]{2}^{\,qquant+quantanf}}\right)^{0.75} - 0.0946\right)$$

4.2 Test of the maximum of the quantized values

The maximum allowed quantized value is limited. This limit is set to constraint the table size if a table-lookup is used to requantize the quantized frequency lines. The limit is given by the possible values of the length identifier, "linbits", of values flagged with an ESC-code. Therefore before any bit counting is done the quantizer stepsize is increased by

qquant = qquant+1

until the maximum of the quantized values is within the range of the largest Huffman code table.

4.3 Calculation of the run length of zeros

The run length rzero of pairs of spectral coefficients quantized to zero on the upper end of the spectrum is counted and called "rzero".

4.4 Calculation of the run length of values less or equal one

The run length of quadrupels of spectral coefficients quantized to one or zero, following the rzero pairs of zeros, is calculated and called "count1"

4.5 Counting the bit necessary to code the values less or equal one

One Huffman code word is used to code one of the "count1" quadrupels. There are two different Huffman code books with corresponding code length tables (table A and table B in 3-Annex 3-B.7). The number of bits to code all the count1 quadrupels is given by:

bitsum_count1 = min(bitsum_table0 , bitsum_table1)

where
count1table_0 is used to point to table A

$$\text{bitsum\_table0} = \sum_{k=\text{firstcount1}}^{k=\text{firstcount1}+\text{count1}-1} \text{count1table\_0}\big(ix(4k)+2*ix(4k+1)+4*ix(4k+2)+8*ix(4k+3)\big)$$

and
count1table_1 is used to point to table B

$$\text{bitsum\_table1} = \sum_{k=\text{firstcount1}}^{k=\text{firstcount1}+\text{count1}-1} \text{count1table\_1}\big(ix(4k)+2*ix(4k+1)+4*ix(4k+2)+8*ix(4k+3)\big)$$

The information which table is used is transmitted by count1table_select, which is "0" for table A or "1" for table B, respectively.

4.6 Call of subroutine SUBDIVIDE

The number of pairs of quantized values not counted in "count1" or "rzero" is called bigvalues. SUBDIVIDE splits the scalefactor bands corresponding to this values into three groups. The last one, incomplete generally, counts as a complete one. Region_adress1/2 contains the number of scalefactor bands in the first and the last region, respectively. The number of scalefactor bands in the second region can be calculated using bigvalues. If bigvalues comprises only two scalefactor bands region_adress2 is set to zero. If there are less than two also region_adress1 is zero. The split strategy is up to the implementation. A very simple one for instance is to assign 1/3 of the scalefactor bands to the first and 1/4 to the last region.

Subdivide in case of blocksplit is done analoguous but only two subregions. Region_address 1 is set to a default in this case. This default is 8 in the case of split_point=0 and 9 in the case of split_point=1. Both this values point to the same absolute frequency.

4.7 Calculation of the code book for each subregion

There are 32 different Huffman code tables for the coding of pairs of quantized values available. They differ from each other in the maximum value that can be coded and in the signal statistic they are optimized for. Only codes for values < 16 are in the table. For values >=16 there are two tables provided where the largest value 15 is an escape character. In this case the value 15 is coded in an additional word using a linear PCM code with a word length called linbits. linbits can be calculated by taking the base 2 logarithm of the PCM code, which is x - max_table_entry (see clause 2.4.2.7).

A simple way to choose a table is to use the maximum of the quantized values in a subregion. Tables which have the same size are optimized for different signal statistics. Therefore additional coding gain can be achieved for example by trying all of this tables.

4.8 Counting of the bit necessary to code the values in the subregions

The number of bits necessary to code the quantized values of one subregion is given by:

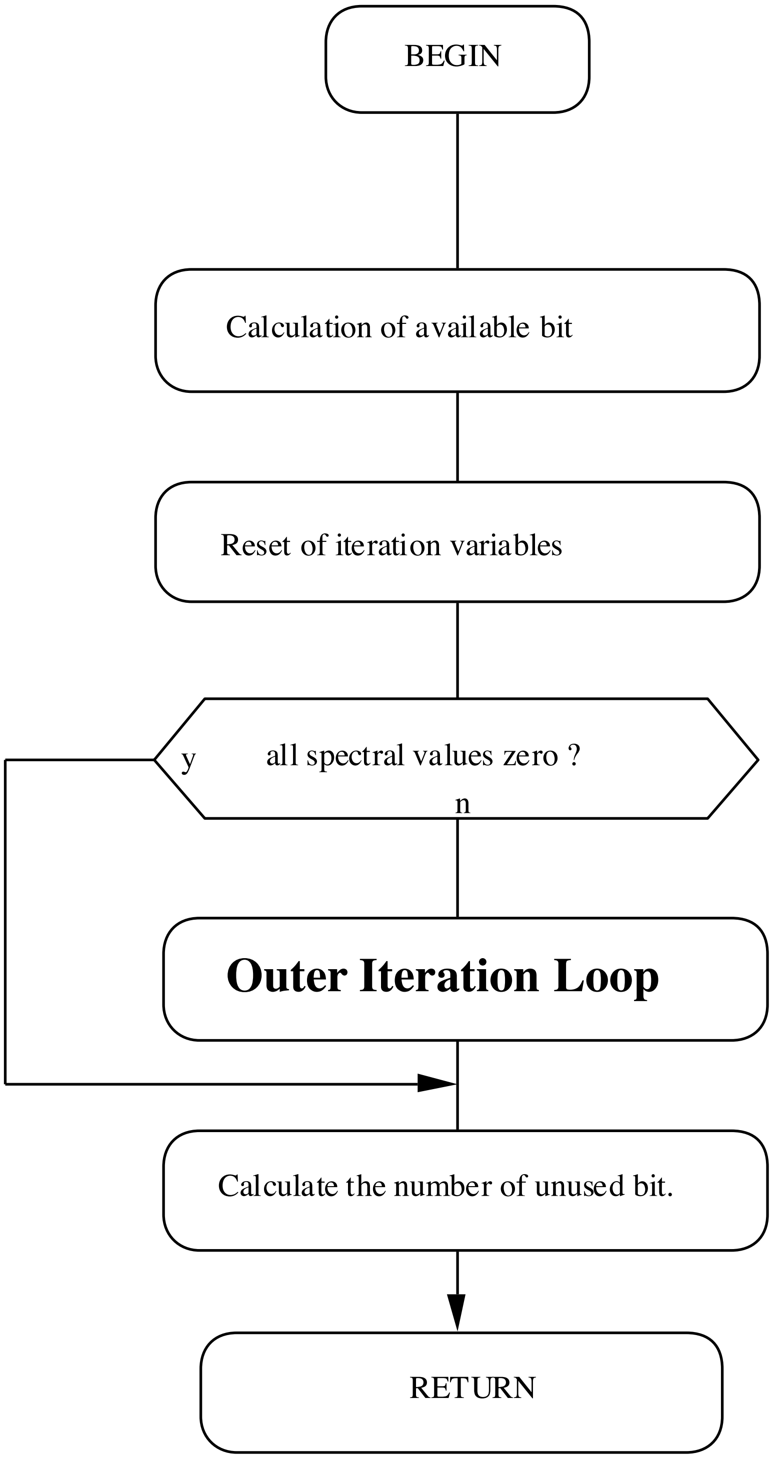
$$\mathrm{bitsum}(j) = \sum_{k=0}^{k=np(j)-1} \mathrm{bitz}\big(\mathrm{tableselect}(j),\ \min(15,\ ix(2k+fe(j))),\ \min(15,\ ix(2k+fe(j)+1))\big)$$

$$+ \sum_{k=0}^{k=np(j)-1} \big(s(\ ix(2k+fe(j)) - 15) + s(\ ix(2k+fe(j)+1) - 15)\big) * \mathrm{linbits}(j)$$

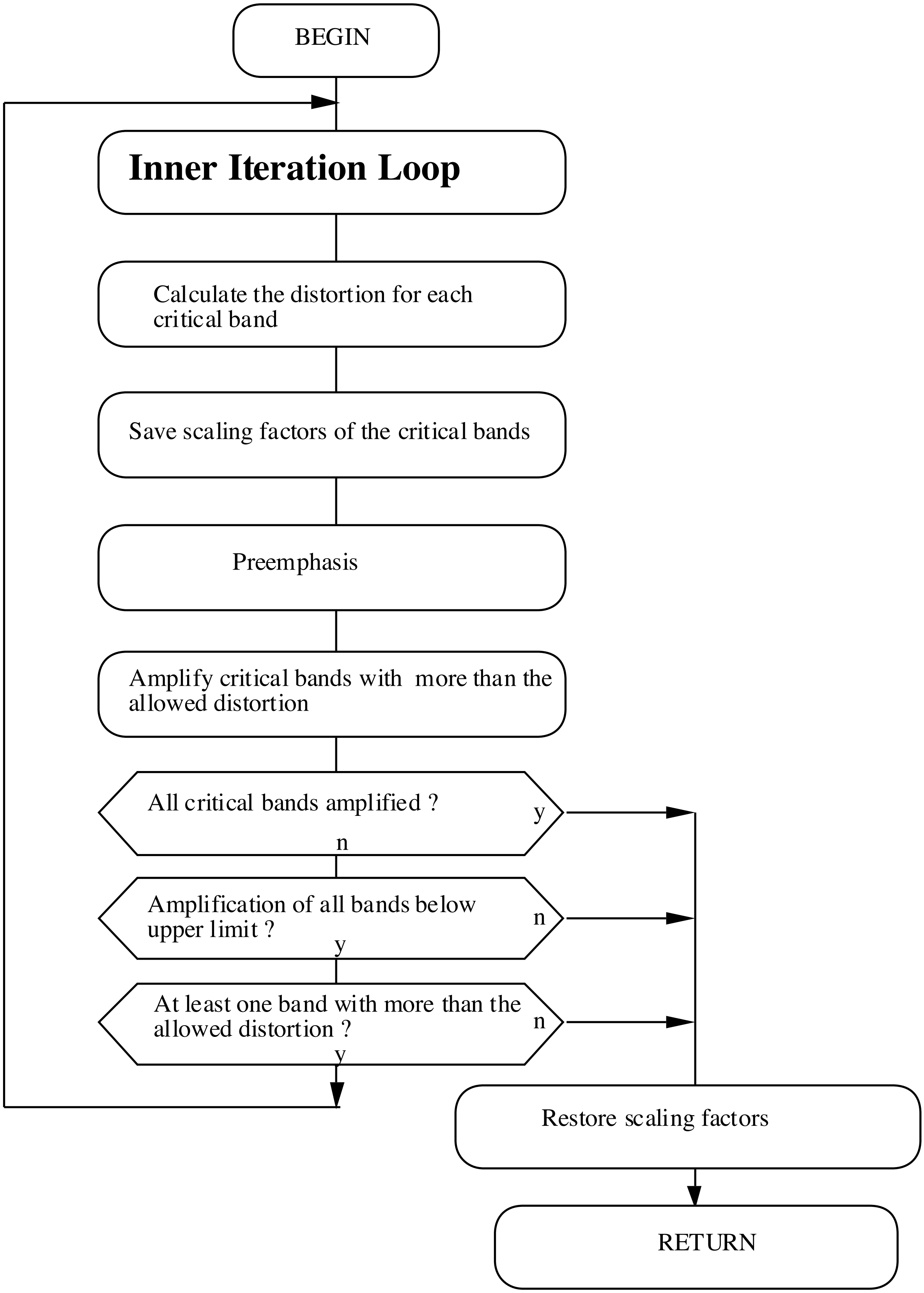
np(j): number of pairs in a sub region
fe(j): number of the first quantized value in a sub-region
bitz: table with Huffman code length

s(...) step function: if x >= 0 s(x) = 1
if x < 0 s(x) = 0

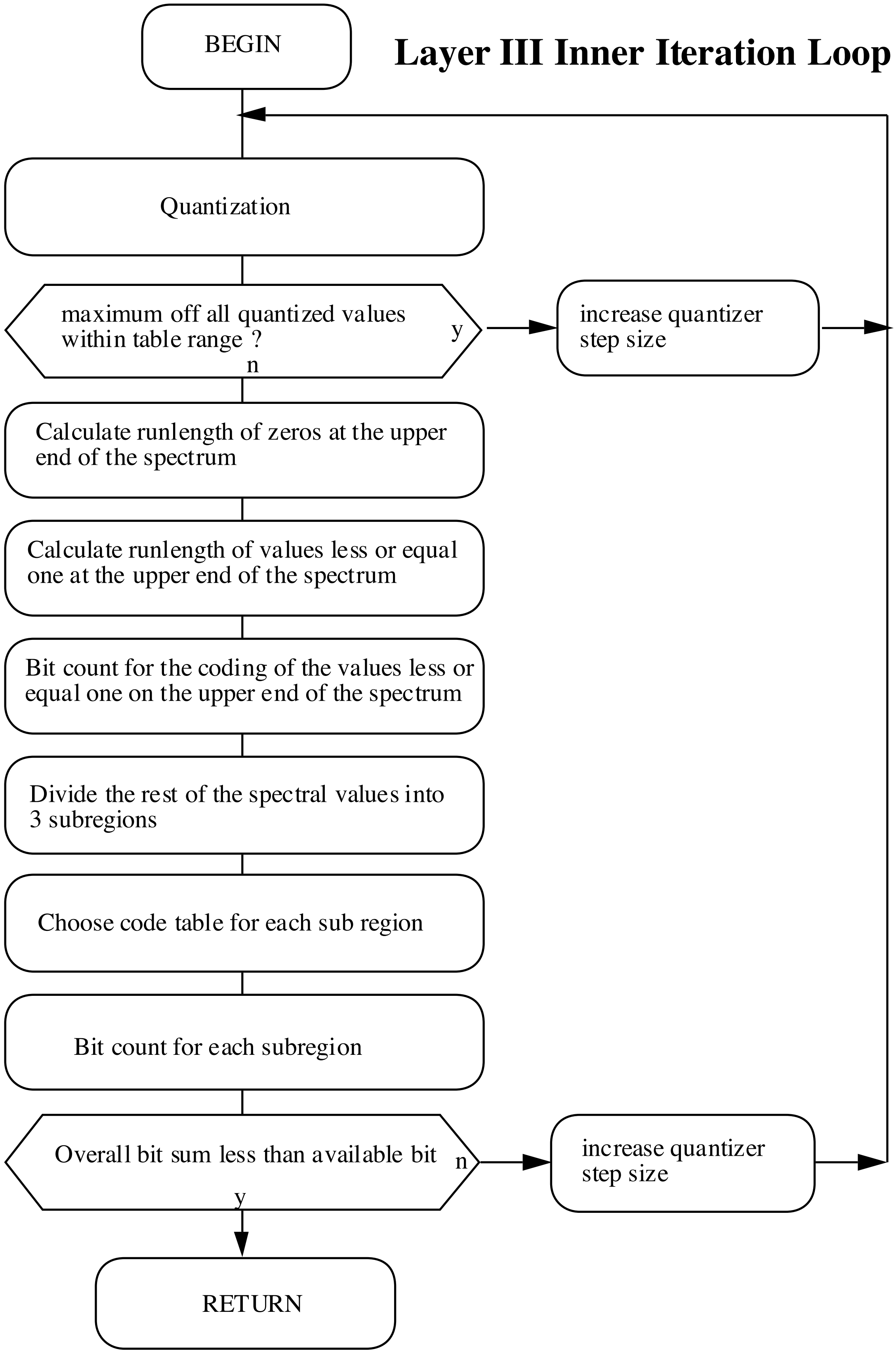
Layer III Iteration loops



Layer III Outer Iteration Loop



Layer III Inner Iteration Loop



3-ANNEX D (informative)

PSYCHOACOUSTIC MODELS

3-D.1. Psychoacoustic Model I

The calculation of the psychoacoustic model has to be adapted to the corresponding layer. This example is valid for Layers I and II. The model can be adapted to LayerIII.

There is no principal difference in the application of psychoacoustic model 1 to Layer I or II.

Layer I: A new bit allocation is calculated for each block of 12 subband or 384 input PCM samples.

Layer II: A new bit allocation is calculated for three blocks totaling 36 subband samples corresponding to 3*384 (1152) input PCM samples.

The bit allocation of the 32 subbands is calculated on the basis of the signal-to-mask ratios of all the subbands. Therefore it is necessary to determine, for each subband the maximum signal level and the minimum masking threshold. The minimum masking threshold is derived from an FFT of the input PCM signal, followed by a psychoacoustic model calculation.

The FFT in parallel with the subband filter compensates for the lack of spectral selectivity obtained at low frequencies by the subband filterbank. This technique provides both a sufficient time resolution for the coded audio signal (Polyphase filter with optimized window for minimal pre-echoes) and a sufficient spectral resolution for the calculation of the masking thresholds.

The frequencies and levels of aliasing distortions can be calculated. This is necessary for calculating a minimum bit rate for those subbands which need some bits to cancel the aliasing components in the decoder. The additional complexity to calculate the better frequency resolution is necessary only in the encoder, and introduces no additional delay or complexity in the decoder.

The calculation of the signal-to-mask-ratio is based on the following steps:

Step 1
- Calculation of the FFT for time to frequency conversion.

Step 2
- Determination of the sound pressure level in each subband.

Step 3
- Determination of the threshold in quiet (absolute threshold).

Step 4
- Finding of the tonal (more sinusoid-like) and non-tonal (more noise-like) components of the audio signal.

Step 5
- Decimation of the maskers, to obtain only the relevant maskers.

Step 6
- Calculation of the individual masking thresholds.

Step 7
- Determination of the global masking threshold.

Step 8

- Determination of the minimum masking threshold in each subband.

Step 9
- Calculation of the signal-to-mask ratio in each subband.

These steps will be further discussed. A sampling frequency of 48kHz is assumed. For the other two sampling frequencies all frequencies mentioned should be scaled accordingly.

Step 1: FFT Analysis

The masking threshold is derived from an estimate of the power density spectrum that is calculated by a 512-point FFT for Layer I, or by a 1024-point FFT for Layers II and III. The FFT is calculated directly from the input PCM signal, windowed by a Hann window.

For a coincidence in time between the bit-allocation and the corresponding subband samples, the PCM-samples entering the FFT have to be delayed:

1. The delay of the analysis subband filter is 256 samples, corresponding to 5.3ms at the 48kHz sampling rate. This corresponds to a window shift of 256 samples.

2. The Hann window must coincide with the subband samples of the frame. For Layer I this amounts to an additional window shift of 64 samples, for Layer II an additional window shift of minus 64 samples.

Technical data of the FFT:

		Layer I	Layer II
-	transform length	512 samples	1024 samples
	Window size if fs = 48 kHz	10.67 ms	21.3 ms
	Window size if fs = 44.1 kHz	11.6 ms	23.2 ms
	Window size if fs = 32 kHz	16 ms	32 ms
-	Frequency resolution	fs/512	fs/1024

- Hann window, h(i):
  h(i) = * 0.5 * {1 - cos[2 * p * (i)/(N-1)]}   0 <= i <= N-1

- power density spectrum X(k):
  X(k) = 10 * log |1/Nh(l) * s(l) * e(-j*k*l*2*p/N)|2 dB   k = 0...N/2

A normalization to the reference level of 96 dB SPL (Sound Pressure Level) has to be done in such a way that the maximum value corresponds to 96dB.

Step 2: Determination of the sound pressure level

The sound pressure level L_{sb} in subband n is computed by:

$$L_{sb}(n) = \underset{X(k) \text{ in subband } n}{\mathrm{MAX}}[\, X(k), 20 \cdot \log(scf_{max}(n) \cdot 32768) - 10 \,] \text{ dB}$$

where X(k) is the sound pressure level of the spectral line with index k of the FFT with the maximum amplitude in the frequency range corresponding to subband n. The expression $scf_{max}(n)$ is in Layer I the scalefactor, and in Layer II the maximum of the three scalefactors of subband n within a frame. The "-10 dB" term corrects for the difference between peak and RMS level. The sound pressure level $L_{sb}(n)$ is computed for every subband n.

Step 3: Considering the threshold in quiet

The threshold in quiet LTq(k), also called absolute threshold, is available in the tables "Frequencies, Critical Band Rates and Absolute Threshold" (Tables 3-D.1a, 3-D.1b, 3-D.1c for LayerI; Tables 3-D.1d, 3-D.1e, 3-D.1f for LayerII). These tables depend on the sampling rate of the input PCM signal. Values are available for each sample in the frequency domain where the masking threshold is calculated.

An offset depending on the overall bit rate is used for the absolute threshold. This offset is -12 dB for bit rates >= 96 kbit/s and 0 dB for bit rates < 96 kbit/s per channel.

Step 4: Finding of tonal and non-tonal components

The tonality of a masking component has an influence on the masking threshold. For this reason, it is worthwhile to discriminate between tonal and non-tonal components. For calculating the global masking threshold it is necessary to derive the tonal and the non-tonal components from the FFT spectrum.

This step starts with the determination of local maxima, then extracts tonal components (sinusoids) and calculates the intensity of the non-tonal components within a bandwidth of a critical band. The boundaries of the critical bands are given in the tables "CRITICAL BAND BOUNDARIES" (Tables 3-D.2a, 3-D.2b, 3-D.2c for LayerI; Tables 3-D.2d, 3-D.2e, 3-D.2f for LayerII).

The bandwidth of the critical bands varies with the center frequency with a bandwidth of about only 0.1 kHz at low frequencies and with a bandwidth of about 4 kHz at high frequencies. It is known from psychoacoustic experiments that the ear has a better frequency resolution in the lower than in the higher frequency region. To determine if a local maximum may be a tonal component a frequency range df around the local maximum is examined. The frequency range df is given by:

Sampling rate: 32 kHz

Layer I:	df = 125 Hz	0 kHz	< f <=	4.0kHz
	df = 187.5 Hz	4.0 kHz	< f <=	8.0 kHz
	df = 375 Hz	8.0 kHz	< f <=	15.0kHz
Layer II:	df = 62.5 Hz	0 kHz	< f <=	3.0 kHz
	df = 93.75 Hz	3.0 kHz	< f <=	6.0 kHz
	df = 187.5 Hz	6.0 kHz	< f <=	12.0 kHz
	df = 375 Hz	12.0 kHz	< f <=	24.0 kHz

Sampling rate: 44.1kHz

Layer I:	df = 172.266 Hz	0 kHz	< f <=	5.512kHz
	df = 281.25 Hz	5.512 kHz	< f <=	11.024 kHz
	df = 562.50 Hz	11.024 kHz	< f <=	19.982kHz
Layer II:	df = 86.133 Hz	0 kHz	< f <=	2.756 kHz
	df = 129.199 Hz	2.756 kHz	< f <=	5.512kHz
	df = 258.398 Hz	5.512 kHz	< f <=	11.024 kHz
	df = 516.797 Hz	11.024 kHz	< f <=	19.982kHz

Sampling rate: 48 kHz

Layer I:	df = 187.5 Hz	0 kHz	< f <=	6.0 kHz
	df = 281.25 Hz	6.0 kHz	< f <=	12.0 kHz
	df = 562.50 Hz	12.0 kHz	< f <=	24.0 kHz
Layer II:	df = 93.750 Hz	0 kHz	< f <=	3.0 kHz
	df = 140.63 Hz	3.0 kHz	< f <=	6.0 kHz
	df = 281.25 Hz	6.0 kHz	< f <=	12.0 kHz
	df = 562.50 Hz	12.0 kHz	< f <=	24.0 kHz

To make lists of the spectral lines X(k) that are tonal or non-tonal, the following three operations are performed:

(i) Labelling of local maxima

A spectral line X(k) is labelled as a local maximum if

$$X(k) > X(k-1) \quad \text{and} \quad X(k) >= X(k+1)$$

(ii) Listing of tonal components and calculation of the sound pressure level
A local maximum is put in the list of tonal components if

$$X(k) - X(k+j) >= 7 \text{ dB},$$

where j is chosen according to

Layer I:

j = -2, +2	for 2 < k < 63
j = -3, -2, +2, +3	for 63 <= k < 127
j = -6,..., -2, +2,..., +6	for 127<= k <= 250

Layer II:

j = -2, +2	for 2 < k < 63
j = -3, -2, +2, +3	for 63 <= k < 127
j = -6,..., -2, +2,..., +6	for 127 <= k < 255
j = -12,..., -2, +2,..., +12	for 255<= k <= 500

If X(k) is found to be a tonal component, then the following parameters are listed:
- Index number k of the spectral line.
- Sound pressure level $X_{tm}(k)=X(k-1)+X(k)+X(k+1)$, in dB
- Tonal flag.

Next, all spectral lines within the examined frequency range are set to -8 dB.

(iii) Listing of non-tonal components and calculation of the power

The non-tonal (noise) components are calculated from the remaining spectral lines. To calculate the non-tonal components from these spectral lines X(k), the critical bands z(k) are determined using the tables, "Critical Band Boundaries" (Tables 3-D.2a, 3-D.2b, 3-D.2c for LayerI; Tables 3-D.2d, 3-D.2e, 3-D.2f for LayerII). In LayerI, 23 critical bands are used for the sampling rate of 32kHz, 24 critical bands for 44.1kHz and 25 critical bands are used for 48kHz. In LayerII, 24 critical bands are used for 32kHz sampling rate, and 26 critical bands are used for 44.1kHz and 48kHz sampling rate. Within each critical band, the power of the spectral lines are summed to form the sound pressure level of the new non-tonal component corresponding to that critical band.

The following parameters are listed:
- Index number k of the spectral line nearest to the geometric mean of the critical band.
- Sound pressure level $X_{nm}(k)$ in dB.
- Non-tonal flag.

Step 5: Decimation of tonal and non-tonal masking components

Decimation is a procedure that is used to reduce the number of maskers which are considered for the calculation of the global masking threshold.

(i) Tonal $X_{tm}(k)$ or non-tonal components $X_{nm}(k)$ are considered for the calculation of the masking threshold only if:

$$Xtm(k) >= LTq(k) \quad \text{or} \quad Xnm(k) >= LTq(k)$$

In this expression, LTq(k) is the absolute threshold (or threshold in quiet) at the frequency of index k. These values are given in the Tables 3-D.1a, 3-D.1b, 3-D.1c for LayerI; Tables 3-D.1d, 3-D.1e, 3-D.1f for LayerII.

(ii) Decimation of two or more tonal components within a distance of less then 0.5 Bark: Keep the component with the highest power, and remove the smaller component(s) from the list of tonal components. For this operation, a sliding window in the critical band domain is used with a width of 0.5 Bark.

In the following, the index j is used to indicate the relevant tonal or non-tonal masking components from the combined decimated list.

Step 6: Calculation of individual masking thresholds

Of the original N/2 frequency domain samples, indexed by k, only a subset of the samples, indexed by i, are considered for the global masking threshold calculation. The samples used are shown in Tables 3-D.1a, 3-D.1b, 3-D.1c for LayerI; Tables 3-D.1d, 3-D.1e, 3-D.1f for LayerII.

Layer I:
For the frequency lines corresponding to the frequency region which is covered by the first six subbands no subsampling is used. For the frequency region corresponding to the next six subbands every second spectral line is considered. Finally, in the case of 44.1 and 48 kHz sampling rates, in the frequency region corresponding to the remaining subbands, every fourth spectral line is considered up to 20 kHz. In the case of 32 kHz sampling rate, in the frequency region corresponding to the remaining subbands, every fourth spectral line is considered up to 15 kHz (See also Tables 3-D.1a, 3-D.1b, 3-D.1c "Frequencies, Critical Band Rates and Absolute Threshold" for LayerI.)

Layer II:
For the frequency lines corresponding to the frequency region which is covered by the first three subbands no subsampling is used. For the frequency region which is covered by next three subbands every second spectral line is considered. For the frequency region corresponding to the next six subbands every fourth spectral line is considered. Finally, in the case of 44.1 and 48 kHz sampling rates, in the remaining subbands every eighth spectral line is considered up to 20 kHz. In the case of 32 kHz sampling rate, in the frequency region corresponding to the remaining subbands, every eighth spectral line is considered up to 15 kHz. (See also Tables 3-D.1d, 3-D.1e, 3-D.1f "Frequencies, Critical Band Rates and Absolute Threshold" for LayerII.)

The number of samples, i, in the subsampled frequency domain is different depending on the sampling rates and layers.

32 kHz sampling rate:	i = 108 for Layer I	and	i = 132 for Layer II
44.1 kHz sampling rate:	i = 106 for Layer I	and	i = 130 for Layer II
48 kHz sampling rate:	i = 102 for Layer I	and	i = 126 for Layer II

To every tonal and non-tonal component the index i in the subsampled frequency domain is assigned, which is closest in frequency to the original spectral line X(k). This index i is given in Tables 3-D.1a, 3-D.1b, 3-D.1c for LayerI; Tables 3-D.1d, 3-D.1e, 3-D.1f for LayerII, "Frequencies, Critical Band Rates and Absolute Threshold".

The individual masking thresholds of both tonal and non-tonal components are given by the following expression:

$$LT_{tm}[z(j),z(i)] = X_{tm}[z(j)] + av_{tm}[z(j)] + vf[z(j),z(i)] \quad dB$$
$$LT_{nm}[z(j),z(i)] = X_{nm}[z(j)] + av_{nm}[z(j)] + vf[z(j),z(i)] \quad dB$$

In this formula LT_{tm} and LT_{nm} are the individual masking thresholds at critical band ratez in Bark of the masking component at the critical band rate z_m in Bark. The values in dB can be either positive or negative. The term $X_{tm}[z(j)]$ is the sound pressure level of the masking component with the index number j at the corresponding critical band rate z(j). The term av is called the masking index and vf the masking function of the masking component $X_{tm}[z(j)]$. The masking index av is different for tonal and non-tonal masker (av_{tm} and av_{nm}).

For tonal maskers it is given by

$av_{tm} = - 1.525 - 0.275 * z(j) - 4.5$ dB,

and for non-tonal maskers

$av_{nm} = - 1.525 - 0.175 * z(j) - 0.5$ dB.

The masking function vf of a masker is characterized by different lower and upper slopes, which depend on the distance in Bark dz = z(i)- z(j) to the masker. In this expression i is the index of the spectral line at which the masking function is calculated and j that of the masker. The critical band rates z(j) and z(i) can be found in Tables 3-D.1a, 3-D.1b, 3-D.1c for LayerI; Tables 3-D.1d, 3-D.1e, 3-D.1f for LayerII, "Frequencies, Critical Band Rates and Absolute Threshold". The masking function, which is the same for tonal and non-tonal maskers, is given by:

$vf = 17 * (dz + 1) - (0.4 * X[z(j)] + 6)$ dB for $-3 <= dz < -1$ Bark

$vf = (0.4 * X[z(j)] + 6) * dz$ dB for $-1 <= dz < 0$ Bark

$vf = - 17 * dz$ dB for $0 <= dz < 1$ Bark

$vf = - (dz -1) * (17 - 0.15 * X[z(j)]) - 17$ dB for $1 <= dz < 8$ Bark

In these expressions X[z(j)] is the sound pressure level of the j'th masking component in dB.
If dz < -3 Bark, or dz >= 8 Bark, the masking is no longer considered (LT_{tm} and LT_{nm} are set to -8dB outside this range).

Step 7: Calculation of the global masking threshold LTg

The global masking threshold $LT_g(i)$ at the i'th frequency sample is derived from the upper and lower slopes of the individual masking threshold of each of the j tonal and non-tonal maskers, and in addition from the threshold in quiet $LT_q(i)$. This is also given in Tables 3-D.1a, 3-D.1b, 3-D.1c for LayerI; Tables 3-D.1d, 3-D.1e, 3-D.1f for LayerII "Frequencies, Critical Band Rates and Absolute Threshold". The global masking threshold is found by summing the powers corresponding to the individual masking thresholds and the threshold in quiet.

$LT_g(i) = 10 \log (10^{LT_q(i)/10} + +)$

The total number of tonal maskers is given by m, and the total number of non-tonal maskers is given by n. For a given i, the range of j can be reduced to just encompass those masking components that are within -8 to +3 Bark from i. Outside of this range LT_{tm} and LT_{nm} are -8 dB.

Step 8: Determination of the minimum masking threshold

The minimum masking level $LT_{min}(n)$ in subband n is determined by the following expression:

$$LT_{min}(n) = \underset{f(i) \text{ in subband } n}{\mathrm{MIN}}[LT_g(i)] \text{ dB}$$

where f(i) is the frequency of the i'th frequency sample. The f(i) are tabulated in the Tables 3-D.1a, 3-D.1b, 3-D.1c for LayerI; Tables 3-D.1d, 3-D.1e, 3-D.1f for LayerII of "Frequencies, Critical Band Rates and Absolute Threshold". A minimum masking level $LT_{min}(n)$ is computed for every subband.

Step 9: Calculation of the signal-to-mask-ratio

The signal-to-mask ratio

$$SMR_{sb}(n) = L_{sb}(n) - LT_{min}(n) \quad dB$$

is computed for every subband n.

Table 3-D.1a.: Frequencies, Critical Band Rates and Absolute Threshold

Table is valid for Layer I at a sampling rate of 32.0 kHz.

Index Number i	Frequency [Hz]	Crit.Band Rate [z]	Absolute Thresh. [dB]
1	62.50	.617	33.44
2	125.00	1.232	19.20
3	187.50	1.842	13.87
4	250.00	2.445	11.01
5	312.50	3.037	9.20
6	375.00	3.618	7.94
7	437.50	4.185	7.00
8	500.00	4.736	6.28
9	562.50	5.272	5.70
10	625.00	5.789	5.21
11	687.50	6.289	4.80
12	750.00	6.770	4.45
13	812.50	7.233	4.14
14	875.00	7.677	3.86
15	937.50	8.103	3.61
16	1000.00	8.511	3.37
17	1062.50	8.901	3.15
18	1125.00	9.275	2.93
19	1187.50	9.632	2.73
20	1250.00	9.974	2.53
21	1312.50	10.301	2.32
22	1375.00	10.614	2.12
23	1437.50	10.913	1.92
24	1500.00	11.199	1.71
25	1562.50	11.474	1.49
26	1625.00	11.736	1.27
27	1687.50	11.988	1.04
28	1750.00	12.230	.80
29	1812.50	12.461	.55
30	1875.00	12.684	.29
31	1937.50	12.898	.02
32	2000.00	13.104	-.25
33	2062.50	13.302	-.54
34	2125.00	13.493	-.83
35	2187.50	13.678	-1.12
36	2250.00	13.855	-1.43
37	2312.50	14.027	-1.73
38	2375.00	14.193	-2.04
39	2437.50	14.354	-2.34
40	2500.00	14.509	-2.64
41	2562.50	14.660	-2.93
42	2625.00	14.807	-3.22
43	2687.50	14.949	-3.49
44	2750.00	15.087	-3.74
45	2812.50	15.221	-3.98

46	2875.00	15.351	-4.20	
47	2937.50	15.478	-4.40	
48	3000.00	15.602	-4.57	
49	3125.00	15.841	-4.82	
50	3250.00	16.069	-4.96	
51	3375.00	16.287	-4.97	
52	3500.00	16.496	-4.86	
53	3625.00	16.697	-4.63	
54	3750.00	16.891	-4.29	
55	3875.00	17.078	-3.87	
56	4000.00	17.259	-3.39	
57	4125.00	17.434	-2.86	
58	4250.00	17.605	-2.31	
59	4375.00	17.770	-1.77	
60	4500.00	17.932	-1.24	
61	4625.00	18.089	-.74	
62	4750.00	18.242	-.29	
63	4875.00	18.392	.12	
64	5000.00	18.539	.48	
65	5125.00	18.682	.79	
66	5250.00	18.823	1.06	
67	5375.00	18.960	1.29	
68	5500.00	19.095	1.49	
69	5625.00	19.226	1.66	
70	5750.00	19.356	1.81	
71	5875.00	19.482	1.95	
72	6000.00	19.606	2.08	
73	6250.00	19.847	2.33	
74	6500.00	20.079	2.59	
75	6750.00	20.300	2.86	
76	7000.00	20.513	3.17	
77	7250.00	20.717	3.51	
78	7500.00	20.912	3.89	
79	7750.00	21.098	4.31	
80	8000.00	21.275	4.79	
81	8250.00	21.445	5.31	
82	8500.00	21.606	5.88	
83	8750.00	21.760	6.50	
84	9000.00	21.906	7.19	
85	9250.00	22.046	7.93	
86	9500.00	22.178	8.75	
87	9750.00	22.304	9.63	
88	10000.00		22.424	10.58
89	10250.00		22.538	11.60
90	10500.00		22.646	12.71
91	10750.00		22.749	13.90
92	11000.00		22.847	15.18
93	11250.00		22.941	16.54
94	11500.00		23.030	18.01
95	11750.00		23.114	19.57
96	12000.00		23.195	21.23
97	12250.00		23.272	23.01
98	12500.00		23.345	24.90
99	12750.00		23.415	26.90
100	13000.00		23.482	29.03
101	13250.00		23.546	31.28

102	13500.00	23.607	33.67
103	13750.00	23.666	36.19
104	14000.00	23.722	38.86
105	14250.00	23.775	41.67
106	14500.00	23.827	44.63
107	14750.00	23.876	47.76
108	15000.00	23.923	51.04

Table 3-D.1b.: Frequencies, Critical Band Rates and Absolute Threshold

Table is valid for Layer I at a sampling rate of 44.1 kHz.

Index Number	Frequency	Crit.Band Rate	Absolute Thresh.
i	[Hz]	[z]	[dB]
1	86.13	.850	25.87
2	172.27	1.694	14.85
3	258.40	2.525	10.72
4	344.53	3.337	8.50
5	430.66	4.124	7.10
6	516.80	4.882	6.11
7	602.93	5.608	5.37
8	689.06	6.301	4.79
9	775.20	6.959	4.32
10	861.33	7.581	3.92
11	947.46	8.169	3.57
12	1033.59	8.723	3.25
13	1119.73	9.244	2.95
14	1205.86	9.734	2.67
15	1291.99	10.195	2.39
16	1378.13	10.629	2.11
17	1464.26	11.037	1.83
18	1550.39	11.421	1.53
19	1636.52	11.783	1.23
20	1722.66	12.125	.90
21	1808.79	12.448	.56
22	1894.92	12.753	.21
23	1981.05	13.042	-.17
24	2067.19	13.317	-.56
25	2153.32	13.578	-.96
26	2239.45	13.826	-1.38
27	2325.59	14.062	-1.79
28	2411.72	14.288	-2.21
29	2497.85	14.504	-2.63
30	2583.98	14.711	-3.03
31	2670.12	14.909	-3.41
32	2756.25	15.100	-3.77
33	2842.38	15.284	-4.09
34	2928.52	15.460	-4.37
35	3014.65	15.631	-4.60
36	3100.78	15.796	-4.78
37	3186.91	15.955	-4.91
38	3273.05	16.110	-4.97
39	3359.18	16.260	-4.98
40	3445.31	16.406	-4.92
41	3531.45	16.547	-4.81
42	3617.58	16.685	-4.65
43	3703.71	16.820	-4.43

44	3789.84	16.951	-4.17	
45	3875.98	17.079	-3.87	
46	3962.11	17.205	-3.54	
47	4048.24	17.327	-3.19	
48	4134.38	17.447	-2.82	
49	4306.64	17.680	-2.06	
50	4478.91	17.905	-1.32	
51	4651.17	18.121	-.64	
52	4823.44	18.331	-.04	
53	4995.70	18.534	.47	
54	5167.97	18.731	.89	
55	5340.23	18.922	1.23	
56	5512.50	19.108	1.51	
57	5684.77	19.289	1.74	
58	5857.03	19.464	1.93	
59	6029.30	19.635	2.11	
60	6201.56	19.801	2.28	
61	6373.83	19.963	2.46	
62	6546.09	20.120	2.63	
63	6718.36	20.273	2.82	
64	6890.63	20.421	3.03	
65	7062.89	20.565	3.25	
66	7235.16	20.705	3.49	
67	7407.42	20.840	3.74	
68	7579.69	20.972	4.02	
69	7751.95	21.099	4.32	
70	7924.22	21.222	4.64	
71	8096.48	21.342	4.98	
72	8268.75	21.457	5.35	
73	8613.28	21.677	6.15	
74	8957.81	21.882	7.07	
75	9302.34	22.074	8.10	
76	9646.88	22.253	9.25	
77	9991.41	22.420	10.54	
78	10335.94		22.576	11.97
79	10680.47		22.721	13.56
80	11025.00		22.857	15.31
81	11369.53		22.984	17.23
82	11714.06		23.102	19.34
83	12058.59		23.213	21.64
84	12403.13		23.317	24.15
85	12747.66		23.415	26.88
86	13092.19		23.506	29.84
87	13436.72		23.592	33.05
88	13781.25		23.673	36.52
89	14125.78		23.749	40.25
90	14470.31		23.821	44.27
91	14814.84		23.888	48.59
92	15159.38		23.952	53.22
93	15503.91		24.013	58.18
94	15848.44		24.070	63.49
95	16192.97		24.125	68.00
96	16537.50		24.176	68.00
97	16882.03		24.225	68.00
98	17226.56		24.271	68.00
99	17571.09		24.316	68.00

100	17915.63	24.358	68.00
101	18260.16	24.398	68.00
102	18604.69	24.436	68.00
103	18949.22	24.473	68.00
104	19293.75	24.508	68.00
105	19638.28	24.542	68.00
106	19982.81	24.574	68.00

Table 3-D.1c. Frequencies, Critical Band Rates and Absolute Threshold

Table is valid for Layer I at a sampling rate of 48 kHz.

Index Number	Frequency	Crit.Band Rate	Absolute Thresh.
i	[Hz]	[z]	[dB]
1	93.75	.925	24.17
2	187.50	1.842	13.87
3	281.25	2.742	10.01
4	375.00	3.618	7.94
5	468.75	4.463	6.62
6	562.50	5.272	5.70
7	656.25	6.041	5.00
8	750.00	6.770	4.45
9	843.75	7.457	4.00
10	937.50	8.103	3.61
11	1031.25	8.708	3.26
12	1125.00	9.275	2.93
13	1218.75	9.805	2.63
14	1312.50	10.301	2.32
15	1406.25	10.765	2.02
16	1500.00	11.199	1.71
17	1593.75	11.606	1.38
18	1687.50	11.988	1.04
19	1781.25	12.347	.67
20	1875.00	12.684	.29
21	1968.75	13.002	-.11
22	2062.50	13.302	-.54
23	2156.25	13.586	-.97
24	2250.00	13.855	-1.43
25	2343.75	14.111	-1.88
26	2437.50	14.354	-2.34
27	2531.25	14.585	-2.79
28	2625.00	14.807	-3.22
29	2718.75	15.018	-3.62
30	2812.50	15.221	-3.98
31	2906.25	15.415	-4.30
32	3000.00	15.602	-4.57
33	3093.75	15.783	-4.77
34	3187.50	15.956	-4.91
35	3281.25	16.124	-4.98
36	3375.00	16.287	-4.97
37	3468.75	16.445	-4.90
38	3562.50	16.598	-4.76
39	3656.25	16.746	-4.55
40	3750.00	16.891	-4.29
41	3843.75	17.032	-3.99
42	3937.50	17.169	-3.64
43	4031.25	17.303	-3.26

44	4125.00	17.434	-2.86	
45	4218.75	17.563	-2.45	
46	4312.50	17.688	-2.04	
47	4406.25	17.811	-1.63	
48	4500.00	17.932	-1.24	
49	4687.50	18.166	-.51	
50	4875.00	18.392	.12	
51	5062.50	18.611	.64	
52	5250.00	18.823	1.06	
53	5437.50	19.028	1.39	
54	5625.00	19.226	1.66	
55	5812.50	19.419	1.88	
56	6000.00	19.606	2.08	
57	6187.50	19.788	2.27	
58	6375.00	19.964	2.46	
59	6562.50	20.135	2.65	
60	6750.00	20.300	2.86	
61	6937.50	20.461	3.09	
62	7125.00	20.616	3.33	
63	7312.50	20.766	3.60	
64	7500.00	20.912	3.89	
65	7687.50	21.052	4.20	
66	7875.00	21.188	4.54	
67	8062.50	21.318	4.91	
68	8250.00	21.445	5.31	
69	8437.50	21.567	5.73	
70	8625.00	21.684	6.18	
71	8812.50	21.797	6.67	
72	9000.00	21.906	7.19	
73	9375.00	22.113	8.33	
74	9750.00	22.304	9.63	
75	10125.00		22.482	11.08
76	10500.00		22.646	12.71
77	10875.00		22.799	14.53
78	11250.00		22.941	16.54
79	11625.00		23.072	18.77
80	12000.00		23.195	21.23
81	12375.00		23.309	23.94
82	12750.00		23.415	26.90
83	13125.00		23.515	30.14
84	13500.00		23.607	33.67
85	13875.00		23.694	37.51
86	14250.00		23.775	41.67
87	14625.00		23.852	46.17
88	15000.00		23.923	51.04
89	15375.00		23.991	56.29
90	15750.00		24.054	61.94
91	16125.00		24.114	68.00
92	16500.00		24.171	68.00
93	16875.00		24.224	68.00
94	17250.00		24.275	68.00
95	17625.00		24.322	68.00
96	18000.00		24.368	68.00
97	18375.00		24.411	68.00
98	18750.00		24.452	68.00
99	19125.00		24.491	68.00

100	19500.00	24.528	68.00
101	19875.00	24.564	68.00
102	20250.00	24.597	68.00

Table 3-D.1d.: Frequencies, Critical Band Rates and Absolute Threshold

Table is valid for Layer II at a sampling rate of 32.0 kHz.

Index Number	**Frequency**	**Crit.Band Rate**	**Absolute Thresh.**
i	**[Hz]**	**[z]**	**[dB]**
1	31.25	.309	58.23
2	62.50	.617	33.44
3	93.75	.925	24.17
4	125.00	1.232	19.20
5	156.25	1.538	16.05
6	187.50	1.842	13.87
7	218.75	2.145	12.26
8	250.00	2.445	11.01
9	281.25	2.742	10.01
10	312.50	3.037	9.20
11	343.75	3.329	8.52
12	375.00	3.618	7.94
13	406.25	3.903	7.44
14	437.50	4.185	7.00
15	468.75	4.463	6.62
16	500.00	4.736	6.28
17	531.25	5.006	5.97
18	562.50	5.272	5.70
19	593.75	5.533	5.44
20	625.00	5.789	5.21
21	656.25	6.041	5.00
22	687.50	6.289	4.80
23	718.75	6.532	4.62
24	750.00	6.770	4.45
25	781.25	7.004	4.29
26	812.50	7.233	4.14
27	843.75	7.457	4.00
28	875.00	7.677	3.86
29	906.25	7.892	3.73
30	937.50	8.103	3.61
31	968.75	8.309	3.49
32	1000.00	8.511	3.37
33	1031.25	8.708	3.26
34	1062.50	8.901	3.15
35	1093.75	9.090	3.04
36	1125.00	9.275	2.93
37	1156.25	9.456	2.83
38	1187.50	9.632	2.73
39	1218.75	9.805	2.63
40	1250.00	9.974	2.53
41	1281.25	10.139	2.42
42	1312.50	10.301	2.32
43	1343.75	10.459	2.22
44	1375.00	10.614	2.12
45	1406.25	10.765	2.02
46	1437.50	10.913	1.92
47	1468.75	11.058	1.81

48	1500.00	11.199	1.71
49	1562.50	11.474	1.49
50	1625.00	11.736	1.27
51	1687.50	11.988	1.04
52	1750.00	12.230	.80
53	1812.50	12.461	.55
54	1875.00	12.684	.29
55	1937.50	12.898	.02
56	2000.00	13.104	-.25
57	2062.50	13.302	-.54
58	2125.00	13.493	-.83
59	2187.50	13.678	-1.12
60	2250.00	13.855	-1.43
61	2312.50	14.027	-1.73
62	2375.00	14.193	-2.04
63	2437.50	14.354	-2.34
64	2500.00	14.509	-2.64
65	2562.50	14.660	-2.93
66	2625.00	14.807	-3.22
67	2687.50	14.949	-3.49
68	2750.00	15.087	-3.74
69	2812.50	15.221	-3.98
70	2875.00	15.351	-4.20
71	2937.50	15.478	-4.40
72	3000.00	15.602	-4.57
73	3125.00	15.841	-4.82
74	3250.00	16.069	-4.96
75	3375.00	16.287	-4.97
76	3500.00	16.496	-4.86
77	3625.00	16.697	-4.63
78	3750.00	16.891	-4.29
79	3875.00	17.078	-3.87
80	4000.00	17.259	-3.39
81	4125.00	17.434	-2.86
82	4250.00	17.605	-2.31
83	4375.00	17.770	-1.77
84	4500.00	17.932	-1.24
85	4625.00	18.089	-.74
86	4750.00	18.242	-.29
87	4875.00	18.392	.12
88	5000.00	18.539	.48
89	5125.00	18.682	.79
90	5250.00	18.823	1.06
91	5375.00	18.960	1.29
92	5500.00	19.095	1.49
93	5625.00	19.226	1.66
94	5750.00	19.356	1.81
95	5875.00	19.482	1.95
96	6000.00	19.606	2.08
97	6250.00	19.847	2.33
98	6500.00	20.079	2.59
99	6750.00	20.300	2.86
100	7000.00	20.513	3.17
101	7250.00	20.717	3.51
102	7500.00	20.912	3.89
103	7750.00	21.098	4.31

104	8000.00	21.275	4.79
105	8250.00	21.445	5.31
106	8500.00	21.606	5.88
107	8750.00	21.760	6.50
108	9000.00	21.906	7.19
109	9250.00	22.046	7.93
110	9500.00	22.178	8.75
111	9750.00	22.304	9.63
112	10000.00	22.424	10.58
113	10250.00	22.538	11.60
114	10500.00	22.646	12.71
115	10750.00	22.749	13.90
116	11000.00	22.847	15.18
117	11250.00	22.941	16.54
118	11500.00	23.030	18.01
119	11750.00	23.114	19.57
120	12000.00	23.195	21.23
121	12250.00	23.272	23.01
122	12500.00	23.345	24.90
123	12750.00	23.415	26.90
124	13000.00	23.482	29.03
125	13250.00	23.546	31.28
126	13500.00	23.607	33.67
127	13750.00	23.666	36.19
128	14000.00	23.722	38.86
129	14250.00	23.775	41.67
130	14500.00	23.827	44.63
131	14750.00	23.876	47.76
132	15000.00	23.923	51.04

Table 3-D.1e.: Frequencies, Critical Band Rates and Absolute Threshold

Table is valid for Layer II at a sampling rate of 44.1 kHz.

Index Number	**Frequency**	**Crit.Band Rate**	**Absolute Thresh.**
i	**[Hz]**	**[z]**	**[dB]**
1	43.07	.425	45.05
2	86.13	.850	25.87
3	129.20	1.273	18.70
4	172.27	1.694	14.85
5	215.33	2.112	12.41
6	258.40	2.525	10.72
7	301.46	2.934	9.47
8	344.53	3.337	8.50
9	387.60	3.733	7.73
10	430.66	4.124	7.10
11	473.73	4.507	6.56
12	516.80	4.882	6.11
13	559.86	5.249	5.72
14	602.93	5.608	5.37
15	646.00	5.959	5.07
16	689.06	6.301	4.79
17	732.13	6.634	4.55
18	775.20	6.959	4.32
19	818.26	7.274	4.11

20	861.33	7.581	3.92
21	904.39	7.879	3.74
22	947.46	8.169	3.57
23	990.53	8.450	3.40
24	1033.59	8.723	3.25
25	1076.66	8.987	3.10
26	1119.73	9.244	2.95
27	1162.79	9.493	2.81
28	1205.86	9.734	2.67
29	1248.93	9.968	2.53
30	1291.99	10.195	2.39
31	1335.06	10.416	2.25
32	1378.13	10.629	2.11
33	1421.19	10.836	1.97
34	1464.26	11.037	1.83
35	1507.32	11.232	1.68
36	1550.39	11.421	1.53
37	1593.46	11.605	1.38
38	1636.52	11.783	1.23
39	1679.59	11.957	1.07
40	1722.66	12.125	.90
41	1765.72	12.289	.74
42	1808.79	12.448	.56
43	1851.86	12.603	.39
44	1894.92	12.753	.21
45	1937.99	12.900	.02
46	1981.05	13.042	-.17
47	2024.12	13.181	-.36
48	2067.19	13.317	-.56
49	2153.32	13.578	-.96
50	2239.45	13.826	-1.38
51	2325.59	14.062	-1.79
52	2411.72	14.288	-2.21
53	2497.85	14.504	-2.63
54	2583.98	14.711	-3.03
55	2670.12	14.909	-3.41
56	2756.25	15.100	-3.77
57	2842.38	15.284	-4.09
58	2928.52	15.460	-4.37
59	3014.65	15.631	-4.60
60	3100.78	15.796	-4.78
61	3186.91	15.955	-4.91
62	3273.05	16.110	-4.97
63	3359.18	16.260	-4.98
64	3445.31	16.406	-4.92
65	3531.45	16.547	-4.81
66	3617.58	16.685	-4.65
67	3703.71	16.820	-4.43
68	3789.84	16.951	-4.17
69	3875.98	17.079	-3.87
70	3962.11	17.205	-3.54
71	4048.24	17.327	-3.19
72	4134.38	17.447	-2.82
73	4306.64	17.680	-2.06
74	4478.91	17.905	-1.32
75	4651.17	18.121	-.64

76	4823.44	18.331	-.04
77	4995.70	18.534	.47
78	5167.97	18.731	.89
79	5340.23	18.922	1.23
80	5512.50	19.108	1.51
81	5684.77	19.289	1.74
82	5857.03	19.464	1.93
83	6029.30	19.635	2.11
84	6201.56	19.801	2.28
85	6373.83	19.963	2.46
86	6546.09	20.120	2.63
87	6718.36	20.273	2.82
88	6890.63	20.421	3.03
89	7062.89	20.565	3.25
90	7235.16	20.705	3.49
91	7407.42	20.840	3.74
92	7579.69	20.972	4.02
93	7751.95	21.099	4.32
94	7924.22	21.222	4.64
95	8096.48	21.342	4.98
96	8268.75	21.457	5.35
97	8613.28	21.677	6.15
98	8957.81	21.882	7.07
99	9302.34	22.074	8.10
100	9646.88	22.253	9.25
101	9991.41	22.420	10.54
102	10335.94	22.576	11.97
103	10680.47	22.721	13.56
104	11025.00	22.857	15.31
105	11369.53	22.984	17.23
106	11714.06	23.102	19.34
107	12058.59	23.213	21.64
108	12403.13	23.317	24.15
109	12747.66	23.415	26.88
110	13092.19	23.506	29.84
111	13436.72	23.592	33.05
112	13781.25	23.673	36.52
113	14125.78	23.749	40.25
114	14470.31	23.821	44.27
115	14814.84	23.888	48.59
116	15159.38	23.952	53.22
117	15503.91	24.013	58.18
118	15848.44	24.070	63.49
119	16192.97	24.125	68.00
120	16537.50	24.176	68.00
121	16882.03	24.225	68.00
122	17226.56	24.271	68.00
123	17571.09	24.316	68.00
124	17915.63	24.358	68.00
125	18260.16	24.398	68.00
126	18604.69	24.436	68.00
127	18949.22	24.473	68.00
128	19293.75	24.508	68.00
129	19638.28	24.542	68.00
130	19982.81	24.574	68.00

Table 3-D.1f.: Frequencies, Critical Band Rates and Absolute Threshold

Table is valid for Layer II at a sampling rate of 48.0 kHz

Index Number i	**Frequency** [Hz]	**Crit.Band Rate** [z]	**Absolute Thresh.** [dB]
1	46.88	.463	42.10
2	93.75	.925	24.17
3	140.63	1.385	17.47
4	187.50	1.842	13.87
5	234.38	2.295	11.60
6	281.25	2.742	10.01
7	328.13	3.184	8.84
8	375.00	3.618	7.94
9	421.88	4.045	7.22
10	468.75	4.463	6.62
11	515.63	4.872	6.12
12	562.50	5.272	5.70
13	609.38	5.661	5.33
14	656.25	6.041	5.00
15	703.13	6.411	4.71
16	750.00	6.770	4.45
17	796.88	7.119	4.21
18	843.75	7.457	4.00
19	890.63	7.785	3.79
20	937.50	8.103	3.61
21	984.38	8.410	3.43
22	1031.25	8.708	3.26
23	1078.13	8.996	3.09
24	1125.00	9.275	2.93
25	1171.88	9.544	2.78
26	1218.75	9.805	2.63
27	1265.63	10.057	2.47
28	1312.50	10.301	2.32
29	1359.38	10.537	2.17
30	1406.25	10.765	2.02
31	1453.13	10.986	1.86
32	1500.00	11.199	1.71
33	1546.88	11.406	1.55
34	1593.75	11.606	1.38
35	1640.63	11.800	1.21
36	1687.50	11.988	1.04
37	1734.38	12.170	.86
38	1781.25	12.347	.67
39	1828.13	12.518	.49
40	1875.00	12.684	.29
41	1921.88	12.845	.09
42	1968.75	13.002	-.11
43	2015.63	13.154	-.32
44	2062.50	13.302	-.54
45	2109.38	13.446	-.75
46	2156.25	13.586	-.97
47	2203.13	13.723	-1.20
48	2250.00	13.855	-1.43
49	2343.75	14.111	-1.88
50	2437.50	14.354	-2.34
51	2531.25	14.585	-2.79
52	2625.00	14.807	-3.22

53	2718.75	15.018	-3.62	
54	2812.50	15.221	-3.98	
55	2906.25	15.415	-4.30	
56	3000.00	15.602	-4.57	
57	3093.75	15.783	-4.77	
58	3187.50	15.956	-4.91	
59	3281.25	16.124	-4.98	
60	3375.00	16.287	-4.97	
61	3468.75	16.445	-4.90	
62	3562.50	16.598	-4.76	
63	3656.25	16.746	-4.55	
64	3750.00	16.891	-4.29	
65	3843.75	17.032	-3.99	
66	3937.50	17.169	-3.64	
67	4031.25	17.303	-3.26	
68	4125.00	17.434	-2.86	
69	4218.75	17.563	-2.45	
70	4312.50	17.688	-2.04	
71	4406.25	17.811	-1.63	
72	4500.00	17.932	-1.24	
73	4687.50	18.166	-.51	
74	4875.00	18.392	.12	
75	5062.50	18.611	.64	
76	5250.00	18.823	1.06	
77	5437.50	19.028	1.39	
78	5625.00	19.226	1.66	
79	5812.50	19.419	1.88	
80	6000.00	19.606	2.08	
81	6187.50	19.788	2.27	
82	6375.00	19.964	2.46	
83	6562.50	20.135	2.65	
84	6750.00	20.300	2.86	
85	6937.50	20.461	3.09	
86	7125.00	20.616	3.33	
87	7312.50	20.766	3.60	
88	7500.00	20.912	3.89	
89	7687.50	21.052	4.20	
90	7875.00	21.188	4.54	
91	8062.50	21.318	4.91	
92	8250.00	21.445	5.31	
93	8437.50	21.567	5.73	
94	8625.00	21.684	6.18	
95	8812.50	21.797	6.67	
96	9000.00	21.906	7.19	
97	9375.00	22.113	8.33	
98	9750.00	22.304	9.63	
99	10125.00		22.482	11.08
100	10500.00		22.646	12.71
101	10875.00		22.799	14.53
102	11250.00		22.941	16.54
103	11625.00		23.072	18.77
104	12000.00		23.195	21.23
105	12375.00		23.309	23.94
106	12750.00		23.415	26.90
107	13125.00		23.515	30.14
108	13500.00		23.607	33.67

109	13875.00	23.694	37.51
110	14250.00	23.775	41.67
111	14625.00	23.852	46.17
112	15000.00	23.923	51.04
113	15375.00	23.991	56.29
114	15750.00	24.054	61.94
115	16125.00	24.114	68.00
116	16500.00	24.171	68.00
117	16875.00	24.224	68.00
118	17250.00	24.275	68.00
119	17625.00	24.322	68.00
120	18000.00	24.368	68.00
121	18375.00	24.411	68.00
122	18750.00	24.452	68.00
123	19125.00	24.491	68.00
124	19500.00	24.528	68.00
125	19875.00	24.564	68.00
126	20250.00	24.597	68.00

Table 3-D.2a. Critical Band Boundaries

This table is valid for Layer I at a sampling rate of 32.0 kHz.
The frequencies represent the top end of each critical band.

no	index of table F&CB	frequency [Hz]	Bark [z]
0	1	62.500	.617
1	3	187.500	1.842
2	5	312.500	3.037
3	7	437.500	4.185
4	9	562.500	5.272
5	11	687.500	6.289
6	13	812.500	7.233
7	15	937.500	8.103
8	18	1125.000	9.275
9	21	1312.500	10.301
10	24	1500.000	11.199
11	27	1687.500	11.988
12	32	2000.000	13.104
13	37	2312.500	14.027
14	44	2750.000	15.087
15	50	3250.000	16.069
16	55	3875.000	17.078
17	61	4625.000	18.089
18	68	5500.000	19.095
19	74	6500.000	20.079
20	79	7750.000	21.098
21	85	9250.000	22.046
22	94	11500.000	23.030
23	108	15000.000	23.923

Table 3-D.2b. Critical Band Boundaries

This table is valid for Layer I at a sampling rate of 44.1 kHz.
The frequencies represent the top end of each critical band.

no	index of table F&CB	frequency [Hz]	Bark [z]

0	1	86.133	.850
1	2	172.266	1.694
2	3	258.398	2.525
3	5	430.664	4.124
4	6	516.797	4.882
5	8	689.063	6.301
6	9	775.195	6.959
7	11	947.461	8.169
8	13	1119.727	9.244
9	15	1291.992	10.195
10	17	1464.258	11.037
11	20	1722.656	12.125
12	23	1981.055	13.042
13	27	2325.586	14.062
14	32	2756.250	15.100
15	37	3186.914	15.955
16	45	3875.977	17.079
17	50	4478.906	17.904
18	55	5340.234	18.922
19	61	6373.828	19.963
20	68	7579.688	20.971
21	75	9302.344	22.074
22	81	11369.531	22.984
23	93	15503.906	24.013
24	106	19982.813	24.573

Table 3-D.2c. Critical Band Boundaries

This table is valid for Layer I at a sampling rate of 48.0 kHz.
The frequencies represent the top end of each critical band.

no	**indexof tableF&CB**	**frequency[Hz]**	**Bark[z]**
0	1	93.750	.925
1	2	187.500	1.842
2	3	281.250	2.742
3	4	375.000	3.618
4	5	468.750	4.463
5	6	562.500	5.272
6	7	656.250	6.041
7	9	843.750	7.457
8	10	937.500	8.103
9	12	1125.000	9.275
10	14	1312.500	10.301
11	16	1500.000	11.199
12	19	1781.250	12.347
13	21	1968.750	13.002
14	25	2343.750	14.111
15	29	2718.750	15.018
16	35	3281.250	16.124
17	41	3843.750	17.032
18	49	4687.500	18.166
19	53	5437.500	19.028
20	58	6375.000	19.964
21	65	7687.500	21.052
22	73	9375.000	22.113

23	79	11625.000	23.072
24	89	15375.000	23.991
25	102	20250.000	24.597

Table 3-D.2d. Critical Band Boundaries

This table is valid for Layer II at a sampling rate of 32.0 kHz.
The frequencies represent the top end of each critical band.

no	**indexof tableF&CB**	**frequency[Hz]**	**Bark[z]**
0	1	31.250	.309
1	3	93.750	.925
2	6	187.500	1.842
3	10	312.500	3.037
4	13	406.250	3.903
5	17	531.250	5.006
6	21	656.250	6.041
7	25	781.250	7.004
8	30	937.500	8.103
9	35	1093.750	9.090
10	41	1281.250	10.139
11	47	1468.750	11.058
12	51	1687.500	11.988
13	56	2000.000	13.104
14	61	2312.500	14.027
15	68	2750.000	15.087
16	74	3250.000	16.069
17	79	3875.000	17.078
18	85	4625.000	18.089
19	92	5500.000	19.095
20	98	6500.000	20.079
21	103	7750.000	21.098
22	109	9250.000	22.046
23	118	11500.000	23.030
24	132	15000.000	23.923

Table 3-D.2e. Critical Band Boundaries

This table is valid for Layer II at a sampling rate of 44.1 kHz.
The frequencies represent the top end of each critical band.

no	**indexof tableF&CB**	**frequency[Hz]**	**Bark[z]**
0	1	43.066	.425
1	2	86.133	.850
2	3	129.199	1.273
3	5	215.332	2.112
4	7	301.465	2.934
5	10	430.664	4.124
6	13	559.863	5.249
7	16	689.063	6.301
8	19	818.262	7.274
9	22	947.461	8.169
10	26	1119.727	9.244
11	30	1291.992	10.195
12	35	1507.324	11.232

13	40	1722.656	12.125
14	46	1981.055	13.042
15	51	2325.586	14.062
16	56	2756.250	15.100
17	62	3273.047	16.11
18	69	3875.977	17.079
19	74	4478.906	17.904
20	79	5340.234	18.922
21	85	6373.828	19.963
22	92	7579.688	20.971
23	99	9302.344	22.074
24	105	11369.531	22.984
25	117	15503.906	24.013
26	130	19982.813	24.573

Table 3-D.2f. Critical Band Boundaries

This table is valid for Layer II at a sampling rate of 48.0 kHz.
The frequencies represent the top end of each critical band.

no	**index of table F&CB**	**frequency [Hz]**	**Bark [z]**
0	1	46.875	.463
1	2	93.750	.925
2	3	140.625	1.385
3	5	234.375	2.295
4	7	328.125	3.184
5	9	421.875	4.045
6	12	562.500	5.272
7	14	656.250	6.041
8	17	796.875	7.119
9	20	937.500	8.103
10	24	1125.000	9.275
11	27	1265.625	10.057
12	32	1500.000	11.199
13	37	1734.375	12.170
14	42	1968.750	13.002
15	49	2343.750	14.111
16	53	2718.750	15.018
17	59	3281.250	16.124
18	65	3843.750	17.032
19	73	4687.500	18.166
20	77	5437.500	19.028
21	82	6375.000	19.964
22	89	7687.500	21.052
23	97	9375.000	22.113
24	103	11625.000	23.072
25	113	15375.000	23.991
26	126	20250.000	24.597

3-D.2 Psychoacoustic Model II

Psychoacoustic Model II is an independent psychoacoustic model that can be adjusted and adapted to any ISO-MPEG-Audio layer. This annex presents the general Psychoacoustic Model II, and provides sufficient information for implementation of Model II with Layers I and II. The Layer III Psychoacoustic Model is based on this implementation, with adaptations as described in the Layer III encoder clause.

The threshold generation process has three inputs. They are:

1. The shift length for the threshold calculation process, *iblen*, where $384 < iblen < 640$. This *iblen* must remain constant over any particular application of the threshold calculation process. If (as in Layer III), it is necessary to calculate thresholds for two different shift lengths, two processes, each running with a fixed shift length, will be necessary. In the case of *iblen* outside the range of 384 to 640 it may be necessary to calculate the psychoacoustic thresholds with a different window length as well as shift length. There are two ways to do this:

   - Use a different length transform, and recalculate the startup coefficients for the model, or

   - Use the same length transform, but a substantially shorter Hann window, appropriate to the data and problem at hand.

   The choice of these is left to the implementation.

2. The newest *iblen* samples of the signal, with the samples delayed (either in the filter bank or psychoacoustic calculation) such that the window of the psychoacoustic calculation is centered in the time-window of application.

3. The sampling rate. There are sets of tables provided for the standard sampling rates. Sampling rate, like *iblen*, must necessarily remain constant over one implementation of the threshold calculation process.

There is one output from Psychoacoustic Model II, a set of Signal to Masking Ratios, *SMRn*, which are adapted to the layers as described below.

Before running the Model initially, the array used to hold the preceding FFT source data window and the arrays used to hold r and f should be zeroed to provide a known starting point.

In Layer II, the psychoacoustic masking ratios must be calculated twice during each coder frame. The more stringent of each pair of ratios is used for bit allocation as shown in the software simulation model for Layers I and II with Psychoacoustic Model II.

Comments on Notation

Throughout this threshold calculation process, three indices for data values are used. These are:

w - indicates that the calculation is indexed by frequency in the FFT spectral line domain. An index of 1 corresponds to the DC term and an index of 513 corresponds to the spectral line at the Nyquest frequency.

b - indicates that the calculation is indexed in the threshold calculation partition domain. In the case where the calculation includes a convolution or sum in the threshold calculation partition domain, *bb* will be used as the summation variable. Partition numbering starts at 1.

n - indicates that the calculation is indexed in the coder bit (or codebook) allocation domain. An index of 1 corresponds to the lowest band in the subband filter bank.

The "Spreading Function"

Several points in the following description refer to the "spreading function". It is calculated by the following method:

$tmpx = 1.05\ (j-i)$,

Where i is the bark value of the signal being spread, j is the bark value of the band being spread into, and $tmpx$ is a temporary variable.

x=8 *minimum* $((tmpx-0.5)^2-2(tmpx-0.5),0)$

Where x is a temporary variable, and minimum (a,b) is a function returning the more negative of a or b.

$tmpy=15.811389+7.5(tmpx+0.474)-17.5(1.0+(tmpx+0.474)^2)^{0.5}$

where $tmpy$ is another temporary variable.

if ($tmpy$<-100) then {$sprdngf\,(i,j)$=0} else {$sprdngf\,(i,j)=10^{(x+tmpy)/10}$ }

Steps in Threshold Calculation

The following are the necessary steps for calculation of the *SMRn* used in the coder.

1. Reconstruct 1024 samples of the input signal.

   iblen new samples are made available at every call to the threshold generator. The threshold generator must store 1024-*iblen* samples, and concatenate those samples to accurately reconstruct 1024 consecutive samples of the input signal, *si,* where i represents the index, $1 < i < 1024$ of the current input stream.

2. Calculate the complex spectrum of the input signal.

   First, *si* is windowed by a 1024 point Hann window, i.e. *swi=si* * (0.5-0.5cos()). Note that in Layer III, a shorter window may be used when window switching is active, with appropriate centering of the window, per the Layer III encoder description.

   Second, a standard forward FFT of *swi* is calculated.

   Third, the polar representation of the transform is calculated. *rw* and *fw* represent the magnitude and phase components of the transformed *swi,* respectively.

3. Calculate a predicted r and f.

   A predicted magnitude, *^rw*, and phase, *^fw* are calculated from the preceding two threshold calculation blocks' r and f:

   ^rw =2.0rw (t-1)-rw (t-2)

   ^fw =2.0fw (t-1)-fw (t-2)

   where t represents the current block number, *t-1* indexes the previous block's data, and *t-2* indexes the data from the threshold calculation block before that.

4. Calculate the unpredictability measure *cw*

   cw, the unpredictability measure, is:

   $cw=(((rw\cos fw - \hat{}rw\cos \hat{}fw)^2+(rw\sin fw - \hat{}rw\sin \hat{}fw)^2)^{0.5}) / (rw+ abs(\hat{}rw))$

   By sacrificing performance, this measure can be calculated on only a lower portion of the frequency lines. Calculations should be done from DC to at least 3kHz and preferably to7kHz. An upper limit of less than 5.5kHz may considerably reduce performance from that obtained during the subjective testing of the audio algorithm. The *cw* values above this limit should be set to 0.3. Best results will be obtained by calculating cw up to 20kHz.

5. Calculate the energy and unpredictability in the threshold calculation partitions.

The energy in each partition, *eb*, is:

$eb= {}_{w=wlowb}{}^{whighb} \sum rw^2$

and the weighted unpredictability, *cb*, is:

$cb= {}_{w=wlowb}{}^{whighb} \sum rw^2 \, cw$

The threshold calculation partitions provide a resolution of approximately either one FFT line or 1/3 critical band, whichever is wider. At low frequencies, a single line of the FFT will constitute a calculation partition. At high frequencies, many lines will be combined into one calculation partition. A set of partition values is provided for each of the three sampling rates in Table 3-D.3."Calculation Partition Tables". These table elements will be used in the threshold calculation process. There are several elements in each table entry:

1. The index of the calculation partition, *b*.

2. The lowest frequency line in the partition, *wlowb*.

3. The highest frequency line in the partition, *whighb*.

4. The median bark value of the partition, *bvalb*.

5. A lower limit for the SNR in the partition that controls stereo unmasking effects, *minvalb*.

6. The value for tone masking noise (in dB) for the partition, *TMNb*.

A largest value of *b,bmax*, equal to the largest index, exists for each sampling rate.

6. Convolve the partitioned energy and unpredictability with the spreading function.

$ecbb= {}_{bb=1}{}^{bmax} \sum ebb * sprdngf(bvalbb,bvalb)$

$ctb= {}_{bb=1}{}^{bmax} \sum cbb * sprdngf(bvalbb,bvalb)$

Because *ctb*. is weighted by the signal energy, it must be renormalized to *cbb*.

$cbb=ctb/ecbb$

At the same time, due to the non-normalized nature of the spreading function, *ecbb* should be renormalized and the normalized energy *enb*, calculated.

$enb=ecbb * rnormb$

The normalization coefficient, *rnormb*. is:

$rnormb=1/ ({}_{bb=0}{}^{bmax} \sum \ sprdngf(bvalbb,bvalb))$

7. Convert *cbb* to *tbb* .

tbb = -0.299-0.43loge (cbb)

Each tbb is limited to the range of $0<tbb<1$.

8. Calculate the required SNR in each partition.

$NMTb$ = 5.5dB for all b. $NMTb$ is the value for noise masking tone (in dB) for the partition. The required signal to noise ratio, $SNRb$, is:

$SNRb$ =maximum($minvalb$, $tbb * TMNb + (1-tbb) NMTb$)

Where maximum (a,b) is a function returning the least negative of a or b.

9. Calculate the power ratio.

The power ratio, bcb, is:

$bcb = 10^{-SNRb/10}$

10. Calculation of actual energy threshold, nbb.

$nbb = enb\ bcb$

11. Spread the threshold energy over FFT lines, yielding nbw.

$nbw = nbb / (whighb - wlowb + 1)$

12. Include absolute thresholds, yielding the final energy threshold of audibility, $thrw$

$thrw$ = max(nbw, $absthrw$)

The dB values of *absthr* shown in Tables 3-D.4. "Absolute Threshold Tables" are relative to the level that a sine wave of ± ½ lsb has in the FFT used for threshold calculation. The dB values must be converted into the energy domain after considering the FFT normalization actually used.

13. Pre-echo control

For Layer III, pre-echo control occurs at this point. The actual control is described as part of the Layer III encoder specification. This step is omitted for Layers I and II.

14. Calculate the signal-to-mask ratios, $SMRn$.

Table 3-D.5. "Layer I and II Coder Partition Table" shows:

1. The index, n, of the coder partition.

2. The lower index $wlown$, of the coder partition.

3. The upper index, $whighn$ of the coder partition.

4. The width index, $widthn$, where $widthn$=1 for a psychoacoustically narrow scalefactor band, and $widthn$=0 for a psychoacoustically wide scalefactor band. A psychoacoustically narrow scalefactor band is one whose width is less than approximately 1/3 critical band.

The energy in the scalefactor band, $epartn$, is:

$epartn = \sum_{w=wlown}^{whighn} rw^2$

Then, if (*widthn* = 1), the noise level in the scalefactor band, *npartn* is calculated as:

$npartn = \sum_{w=wlown}^{whighn} thrw$

else,

$npartn = \text{minimum}(thr_{wlown}, \ldots, thr_{whighn}) * (whighn - wlown + 1)$

Where, in this case, minimum (a,...,z) is a function returning the smallest positive argument of the arguments a...z.

The ratios to be sent to the coder, *SMRn* , are calculated as:

$SMRn = 10 \log_{10} (epart_n/npart_n)$

Table 3-D.3a. Calculation Partition Table

This table is valid at a sampling rate of 32.0 kHz.

Index	wlow	whigh	bval	minval	TMN
1	1	1	0.00	0.0	24.5
2	2	4	0.63	0.0	24.5
3	5	7	1.56	20.0	24.5
4	8	10	2.50	20.0	24.5
5	11	13	3.44	20.0	24.5
6	14	16	4.34	20.0	24.5
7	17	19	5.17	20.0	24.5
8	20	22	5.94	20.0	24.5
9	23	25	6.63	17.0	24.5
10	26	28	7.28	15.0	24.5
11	29	31	7.90	15.0	24.5
12	32	34	8.50	10.0	24.5
13	35	37	9.06	7.0	24.5
14	38	41	9.65	7.0	24.5
15	42	45	10.28	4.4	24.8
16	46	49	10.87	4.4	25.4
17	50	53	11.41	4.5	25.9
18	54	57	11.92	4.5	26.4
19	58	61	12.39	4.5	26.9
20	62	65	12.83	4.5	27.3
21	66	70	13.29	4.5	27.8
22	71	75	13.78	4.5	28.3
23	76	81	14.27	4.5	28.8
24	82	87	14.76	4.5	29.3
25	88	93	15.22	4.5	29.7
26	94	99	15.63	4.5	30.1
27	100	106	16.06	4.5	30.6
28	107	113	16.47	4.5	31.0
29	114	120	16.86	4.5	31.4
30	121	129	17.25	4.5	31.8
31	130	138	17.65	4.5	32.2
32	139	148	18.05	4.5	32.5
33	149	159	18.42	4.5	32.9
34	160	170	18.81	4.5	33.3

35	171	183	19.18	4.5	33.7
36	184	196	19.55	4.5	34.1
37	197	210	19.93	4.5	34.4
38	211	225	20.29	4.5	34.8
39	226	240	20.65	4.5	35.2
40	241	258	21.02	4.5	35.5
41	259	279	21.38	4.5	35.9
42	280	300	21.74	4.5	36.2
43	301	326	22.10	4.5	36.6
44	327	354	22.44	4.5	36.9
45	355	382	22.79	4.5	37.3
46	383	420	23.14	4.5	37.6
47	421	458	23.49	4.5	38.0
48	459	496	23.83	4.5	38.3
49	497	513	24.07	4.5	38.6

Table 3-D.3b. Calculation Partition Table

This table is valid at a sampling rate of 44.1.0 kHz.

Index	wlow	whigh	bval	minval	TMN
1	1	1	0.00	0.0	24.5
2	2	2	0.43	0.0	24.5
3	3	3	0.86	0.0	24.5
4	4	4	1.29	20.0	24.5
5	5	5	1.72	20.0	24.5
6	6	6	2.15	20.0	24.5
7	7	7	2.58	20.0	24.5
8	8	8	3.01	20.0	24.5
9	9	9	3.45	20.0	24.5
10	10	10	3.88	20.0	24.5
11	11	11	4.28	20.0	24.5
12	12	12	4.67	20.0	24.5
13	13	13	5.06	20.0	24.5
14	14	14	5.42	20.0	24.5
15	15	15	5.77	20.0	24.5
16	16	16	6.11	17.0	24.5
17	17	19	6.73	17.0	24.5
18	20	22	7.61	15.0	24.5
19	23	25	8.44	10.0	24.5
20	26	28	9.21	7.0	24.5
21	29	31	9.88	7.0	24.5
22	32	34	10.51	4.4	25.0
23	35	37	11.11	4.5	25.6
24	38	40	11.65	4.5	26.2
25	41	44	12.24	4.5	26.7
26	45	48	12.85	4.5	27.4
27	49	52	13.41	4.5	27.9
28	53	56	13.94	4.5	28.4
29	57	60	14.42	4.5	28.9
30	61	64	14.86	4.5	29.4
31	65	69	15.32	4.5	29.8
32	70	74	15.79	4.5	30.3
33	75	80	16.26	4.5	30.8
34	81	86	16.73	4.5	31.2
35	87	93	17.19	4.5	31.7

36	94	100	17.62	4.5	32.1
37	101	108	18.05	4.5	32.5
38	109	116	18.45	4.5	32.9
39	117	124	18.83	4.5	33.3
40	125	134	19.21	4.5	33.7
41	135	144	19.60	4.5	34.1
42	145	155	20.00	4.5	34.5
43	156	166	20.38	4.5	34.9
44	167	177	20.74	4.5	35.2
45	178	192	21.12	4.5	35.6
46	193	207	21.48	4.5	36.0
47	208	222	21.84	4.5	36.3
48	223	243	22.20	4.5	36.7
49	244	264	22.56	4.5	37.1
50	265	286	22.91	4.5	37.4
51	287	314	23.26	4.5	37.8
52	315	342	23.60	4.5	38.1
53	343	371	23.95	4.5	38.4
54	372	401	24.30	4.5	38.8
55	402	431	24.65	4.5	39.1
56	432	469	25.00	4.5	39.5
57	470	513	25.33	3.5	39.8

Table 3-D.3c. Calculation Partition Table

This table is valid at a sampling rate of 48.0 kHz.

Index	wlow	whigh	bval	minval	TMN
1	1	1	0.00	0.0	24.5
2	2	2	0.47	0.0	24.5
3	3	3	0.94	0.0	24.5
4	4	4	1.41	20.0	24.5
5	5	5	1.88	20.0	24.5
6	6	6	2.34	20.0	24.5
7	7	7	2.81	20.0	24.5
8	8	8	3.28	20.0	24.5
9	9	9	3.75	20.0	24.5
10	10	10	4.20	20.0	24.5
11	11	11	4.63	20.0	24.5
12	12	12	5.05	20.0	24.5
13	13	13	5.44	20.0	24.5
14	14	14	5.83	20.0	24.5
15	15	15	6.19	20.0	24.5
16	16	16	6.52	17.0	24.5
17	17	17	6.86	17.0	24.5
18	18	20	7.49	15.0	24.5
19	21	23	8.40	10.0	24.5
20	24	26	9.24	7.0	24.5
21	27	29	9.97	7.0	24.5
22	30	32	10.65	4.4	25.1
23	33	35	11.28	4.5	25.8
24	36	38	11.86	4.5	26.4
25	39	41	12.39	4.5	26.9
26	42	45	12.96	4.5	27.5
27	46	49	13.56	4.5	28.1
28	50	53	14.12	4.5	28.6
29	54	57	14.62	4.5	29.1

30	58	62	15.14	4.5	29.6
31	63	67	15.67	4.5	30.2
32	68	72	16.15	4.5	30.7
33	73	77	16.58	4.5	31.1
34	78	83	17.02	4.5	31.5
35	84	89	17.44	4.5	31.9
36	90	95	17.84	4.5	32.3
37	96	103	18.24	4.5	32.7
38	104	111	18.66	4.5	33.2
39	112	120	19.07	4.5	33.6
40	121	129	19.47	4.5	34.0
41	130	138	19.85	4.5	34.3
42	139	149	20.23	4.5	34.7
43	150	160	20.63	4.5	35.1
44	161	173	21.02	4.5	35.5
45	174	187	21.40	4.5	35.9
46	188	201	21.76	4.5	36.3
47	202	219	22.12	4.5	36.6
48	220	238	22.47	4.5	37.0
49	239	257	22.83	4.5	37.3
50	258	283	23.18	4.5	37.7
51	284	309	23.53	4.5	38.0
52	310	335	23.88	4.5	38.4
53	336	363	24.23	4.5	38.7
54	364	391	24.58	4.5	39.1
55	392	423	24.93	4.5	39.4
56	424	465	25.27	4.5	39.8
57	466	507	25.61	3.5	40.1
58	508	513	25.81	3.5	40.3

Table 3-D.4a. Absolute Threshold Table

This table is valid at a sampling rate of 32.0 kHz.

A value of 0 dB represents a level in the absolute threshold calculation of 96 dB below the energy of a sine wave of amplitude +-32760.

index (line) lower	higher	absthr (dB)
1	1	58.23
2	2	33.44
3	3	24.17
4	4	19.20
5	5	16.05
6	6	13.87
7	7	12.26
8	8	11.01
9	9	10.01
10	10	9.20
11	11	8.52
12	12	7.94

13	13	7.44
14	14	7.00
15	15	6.62
16	16	6.28
17	17	5.97
18	18	5.70
19	19	5.44
20	20	5.21
21	21	5.00
22	22	4.80
23	23	4.62
24	24	4.45
25	25	4.29
26	26	4.14
27	27	4.00
28	28	3.86
29	29	3.73
30	30	3.61
31	31	3.49
32	32	3.37
33	33	3.26
34	34	3.15
35	35	3.04
36	36	2.93
37	37	2.83
38	38	2.73
39	39	2.63
40	40	2.53
41	41	2.42
42	42	2.32
43	43	2.22
44	44	2.12
45	45	2.02
46	46	1.92
47	47	1.81
48	48	1.71
49	50	1.49
51	52	1.27
53	54	1.04
55	56	.80
57	57	.55
59	60	.29
61	62	.02
63	64	-.25
65	66	-.54
67	68	-.83
69	70	-1.12
71	72	-1.43
73	74	-1.73
75	76	-2.04
77	78	-2.34
79	80	-2.64
81	82	-2.93
83	84	-3.22
85	86	-3.49
87	88	-3.74

89	90	-3.98
91	92	-4.20
93	94	-4.40
95	96	-4.57
97	100	-4.82
101	104	-4.96
105	108	-4.97
109	112	-4.86
113	116	-4.63
117	120	-4.29
121	124	-3.87
125	128	-3.39
129	132	-2.86
133	136	-2.31
137	140	-1.77
141	144	-1.24
145	148	-.74
149	152	-.29
153	156	.12
157	160	.48
161	164	.79
165	168	1.06
169	172	1.29
173	176	1.49
177	180	1.66
181	184	1.81
185	188	1.95
189	192	2.08
193	200	2.33
201	208	2.59
209	216	2.86
217	224	3.17
225	232	3.51
233	240	3.89
241	248	4.31
249	256	4.79
257	264	5.31
265	272	5.88
273	280	6.50
281	288	7.19
289	296	7.93
297	304	8.75
305	312	9.63
313	320	10.58
321	328	11.60
329	336	12.71
337	344	13.90
345	352	15.18
353	360	16.54
361	368	18.01
369	376	19.57
377	384	21.23
385	392	23.01
393	400	24.90
401	408	26.90
409	416	29.03

417	424	31.28
425	432	33.67
433	440	36.19
441	448	38.86
449	456	41.67
457	464	44.63
465	472	47.76
473	480	51.03

Table 3-D.4b. Absolute Threshold Table

This table is valid at a sampling rate of 44.1 kHz.

A value of 0 dB represents a level in the absolute threshold calculation of 96 dB below the energy of a sine wave of amplitude +-32760.

index lower	(line) higher	absthr dB
1	1	45.05
2	2	25.87
3	3	18.70
4	4	14.85
5	5	12.41
6	6	10.72
7	7	9.47
8	8	8.50
9	9	7.73
10	10	7.10
11	11	6.56
12	12	6.11
13	13	5.72
14	14	5.37
15	15	5.07
16	16	4.79
17	17	4.55
18	18	4.32
19	19	4.11
20	20	3.92
21	21	3.74
22	22	3.57
23	23	3.40
24	24	3.25
25	25	3.10
26	26	2.95
27	27	2.81
28	28	2.67
29	29	2.53
30	30	2.39
31	31	2.25
32	32	2.11
33	33	1.97

34	34	1.83
35	35	1.68
36	36	1.53
37	37	1.38
38	38	1.23
39	39	1.07
40	40	.90
41	41	.74
42	42	.56
43	43	.39
44	44	.21
45	45	.02
46	46	-.17
47	47	-.36
48	48	-.56
49	50	-.96
51	52	-1.37
53	54	-1.79
55	56	-2.21
57	58	-2.63
59	60	-3.03
61	62	-3.41
63	64	-3.77
65	66	-4.09
67	68	-4.37
69	70	-4.60
71	72	-4.78
73	74	-4.91
75	76	-4.97
77	78	-4.98
79	80	-4.92
81	82	-4.81
83	84	-4.65
85	86	-4.43
87	88	-4.17
89	90	-3.87
91	92	-3.54
93	94	-3.19
95	96	-2.82
97	100	-2.06
101	104	-1.33
105	108	-.64
109	112	-.04
113	116	.47
117	120	.89
121	124	1.23
125	128	1.51
129	132	1.74
133	136	1.93
137	140	2.11
141	144	2.28
145	148	2.45
149	152	2.63
153	156	2.82
157	160	3.03
161	164	3.25

165	168	3.49
169	172	3.74
173	176	4.02
177	180	4.32
181	184	4.64
185	188	4.98
189	192	5.35
193	200	6.15
201	208	7.07
209	216	8.10
217	224	9.25
225	232	10.54
233	240	11.97
241	248	13.56
249	256	15.30
257	264	17.23
265	272	19.33
273	280	21.64
281	288	24.15
289	296	26.88
297	304	29.84
305	312	33.04
313	320	36.51
321	328	40.24
329	336	44.26
337	344	48.58
345	352	53.21
353	360	58.17
361	368	63.48
369	376	69.13
377	384	69.13
385	392	69.13
393	400	69.13
401	408	69.13
409	416	69.13
417	424	69.13
425	432	69.13
433	440	69.13
441	448	69.13
449	456	69.13
457	464	69.13

Table 3-D.4c. Absolute Threshold Table

This table is valid at a sampling rate of 48.0 kHz.

A value of 0 dB represents a level in the absolute threshold calculation of 96 dB below the energy of a sine wave of amplitude +-32760.

index (line) lower	higher	absthr. dB
1	1	42.10
2	2	24.17
3	3	17.47

4	4	13.87
5	5	11.60
6	6	10.01
7	7	8.84
8	8	7.94
9	9	7.22
10	10	6.62
11	11	6.12
12	12	5.70
13	13	5.33
14	14	5.00
15	15	4.71
16	16	4.45
17	17	4.21
18	18	4.00
19	19	3.79
20	20	3.61
21	21	3.43
22	22	3.26
23	23	3.09
24	24	2.93
25	25	2.78
26	26	2.63
27	27	2.47
28	28	2.32
29	29	2.17
30	30	2.02
31	31	1.86
32	32	1.71
33	33	1.55
34	34	1.38
35	35	1.21
36	36	1.04
37	37	.86
38	38	.67
39	39	.49
40	40	.29
41	41	.09
42	42	-.11
43	43	-.32
44	44	-.54
45	45	-.75
46	46	-.97
47	47	-1.20
48	48	-1.43
49	50	-1.88
51	52	-2.34
53	54	-2.79
55	56	-3.22
57	58	-3.62
59	60	-3.98
61	62	-4.30
63	64	-4.57
65	66	-4.77
67	68	-4.91
69	70	-4.98

71	72	-4.97
73	74	-4.90
75	76	-4.76
77	78	-4.55
79	80	-4.29
81	82	-3.99
83	84	-3.64
85	86	-3.26
87	88	-2.86
89	90	-2.45
91	92	-2.04
93	94	-1.63
95	96	-1.24
97	100	-.51
101	104	.12
105	108	.64
109	112	1.06
113	116	1.39
117	120	1.66
121	124	1.88
125	128	2.08
129	132	2.27
133	136	2.46
137	140	2.65
141	144	2.86
145	148	3.09
149	152	3.33
153	156	3.60
157	160	3.89
161	164	4.20
165	168	4.54
169	172	4.91
173	176	5.31
177	180	5.73
181	184	6.18
185	188	6.67
189	192	7.19
193	200	8.33
201	208	9.63
209	216	11.08
217	224	12.71
225	232	14.53
233	240	16.54
241	248	18.77
249	256	21.23
257	264	23.94
265	272	26.90
273	280	30.14
281	288	33.67
289	296	37.51
297	304	41.67
305	312	46.17
313	320	51.04
321	328	56.29
329	332	61.94
333	340	68.00

341	348	68.00
349	356	68.00
357	364	68.00
365	372	68.00
373	380	68.00
381	388	68.00
389	396	68.00
397	404	68.00
405	412	68.00
413	420	68.00
421	428	68.00

Table 3-D.5. Layer I and Layer II Coder Partition Table

Index	ωlown+1 ωhighn	widthn
0	1	0
1	17	0
2	33	0
3	49	0
4	65	0
5	81	0
6	97	0
7	113	0
8	129	0
9	145	0
10	161	0
11	177	0
12	193	0
13	209	1
14	225	1
15	241	1
16	257	1
17	273	1
18	289	1
19	305	1
20	321	1
21	337	1
22	353	1
23	369	1
24	385	1
25	401	1
26	417	1
27	433	1
28	449	1
29	465	1
30	481	1
31	497	1
32	513	1

3-ANNEX E (informative)
BIT SENSITIVITY TO ERRORS
3-E.1. General

This paragraph indicates the sensitivity of individual bits to random errors if application specific error protection is needed.This sensitivity is given for each bit by a value from 0 to 5, indicating the amount of degradation resulting from one isolated error :

5	catastrophic
4	very annoying
3	annoying
2	slightly annoying
1	audible
0	insensitive

The values are not the results of precise measurements, rather they rely upon knowledge of the codec. They assume the error detection scheme is not in use.

Some fields in the bit stream do not have a fixed length. All bits in this fields are rated for error sensitivity, even if not in use.

(For all layers, the header information is considered to have the highest sensitivity).

3-E.2. Layers I and II

Parameters	**#bit**	**sensitivity**
Bit allocation	all bits	5
Scalefactors select information	all bits	5
Scalefactors	5 (msb)	4
	4	4
	3	4
	2	3
	1	2
	0 (lsb)	1
Subband samples (*)	8-16(msb)	3
	5-7	2
	3,4	1
	(lsb)0-2	0

(*) according to the bit allocation

3-E.3. Layer III

Parameters	**#bit**	**sensitivity**
Scf_si	all bits	5
Part2/3 _length	all bits	4
Big_values	all bits	3
Global_gain	all bits	5
Scalefactor_select	all bits	5
Blocksplit_flag	all bits	5
Block_type	all bits	4
Switch_frequency	all bits	4
Table_select	all bits	5
Region_adress1	all bits	3

Region_adress2	all bits	3
extension_bits (if present)	all bits	0
Preflag	0	2
Scalefac_scale	0	2
Count1table_select	0	3
Subblock_gain	2 (msb)	4
	1	3
	0 (lsb)	2
Scalefac (**)	3 (msb)	3(2)
	2	3(2)
	1	2(1)
	0 (lsb)	2(1)
Huffman codes (***)	0...n-1	3 - 0

(**) the scalefac length depends on scalefac_select.
The bit sensitivity values refer to the scalefac_scale value 1 (if 0 the value is in parenthesis).
(***) If n is the number of bits for Huffman coding in one block the bit sensivity decreases linearly from 3 to 0 as the bit number varies from 0 up to n, (from low to high frequency).

Note:
Rearrangement of the Huffman coded values:
To get better implicit error robustness for the low frequency part of the spectrum the Huffman coded values can be transmitted not in their logical order, but in an interleaved fashion.

If max_hlen is the maximum length of a Huffman codeword over the tables which are used to code the particular block and n is the number of bits used for Huffman coding of data in the block (not frame), then int(n/max_hlen) slots are filled with the first codewords, beginning from low frequencies. The remaining codewords are filled into the remaining place, again arranged from low to high frequencies.

After bit interleaving, the bit sensivity of bit k+i*int(n/max_hlen) decreases linearly from 3 to 0 as k varies from 0 up to int(n/max_hlen)-1, where i=0,...,max_hlen-1, and n is the number of bits for Huffman coding in one block.

This is the recommended practice for Layer III data for all channels where error robustness is important.

3-ANNEX F (informative)
ERROR CONCEALMENT

An optional feature of the coded bit stream is the CRC word which provides some error detection facility to the decoder. The Hamming distance of this error detection code is d=4, which allows for the detection of up to 3 single bit errors or for the detection of one error burst of up to 16 bit length. The amount and the position of the protected bits within one encoded audio frame generally depends on the layer, the mode, data rate, and sampling frequency.

This can be used to control an error concealment strategy in order to avoid severe impairments of the reconstructed signal due to errors in the most sensitive information.

Some basic techniques can be used for concealment, for instance information substitution, or muting. A simple substitution technique consists, when an erroneous frame occurs, of replacing it by the previous one (if error free).

3-ANNEX G (informative)
JOINT STEREO CODING
3-G.1. Intensity Stereo Coding Layer I, II

An optional joint stereo coding method used in Layers I and II is intensity stereo coding. Intensity stereo coding can be used to increase the audio quality and/or reduce the bitrate for stereophonic signals. The gain in bitrate is typically about 10 to 30 kbit/s. It requires negligible additional decoder complexity. The increase of encoder complexity is small. The encoder and decoder delay is not affected.

Psychoacoustic results indicate that at high frequencies (above about 2 kHz) the localization of the stereophonic image within a critical band is determined by the temporal envelope and not by the temporal fine structure of the audio signal.

The basic idea for intensity stereo coding is that for some subbands, instead of transmitting separate left and right subband samples only the sum-signal is transmitted, but with scalefactors for both the left and right channels, thus preserving the stereophonic image.

Flow diagrams of a stereo encoder and decoder, including intensity stereo mode, are shown in Figure 3-G.1 "GENERAL STEREO ENCODER FLOW-CHART" and Figure 3-G.2 "GENERAL STEREO DECODER FLOW-CHART". First, an estimation is made of the required bitrate for both left and right channel. If the required bitrate exceeds the available bitrate, the required bitrate can be decreased by setting a number of subbands to intensity stereo mode. Depending on the bitrate needed, subbands

16 to 31,
12 to 31,
8 to 31, or
4 to 31

can be set to intensity stereo mode. For the quantization of such combined subbands, the higher of the bit allocations for left and right channel is used.

The left and right subband signals of the subbands in joint stereo mode are added. These new subband signals are scaled in the normal way, but the originally determined scalefactors of the left and right subband signals are transmitted according to the bitstream syntax. Quantization of common subband samples, coding of common samples, and coding of common bit allocation are performed in the same way as in independent coding.

3-G.2. MS_Stereo and Intensity Stereo Coding Layer III

In Layer III a combination of ms_stereo mode (sum/difference) and intensity stereo mode can be used.

1) MS_stereo switching

MS_stereo mode is switched on if in joint stereo mode condition

< 0.8 *

is true. The values rli and rri correspond to the energies of the FFT line spectrum of the left and right channel calculated within the psychoacoustic model.

2) MS_stereo processing

- MS matrix

In MS_stereo mode the values of the normalized middle/side channel Mi/Si are transmitted instead of the left/right channel values Li/Ri:

Mi = and Si =

- Limitation of Si channel bandwith

All Si values above the highest scalefactor band are set to zero.

- Sparsing of Si channel

In every scalefactor band sb all pairs of small values (S_i, S_{i+1}) are set to zero:

$$\text{if } (S_i^2 + S_{i+1}^2) < s_{sb} * (L_i^2 + L_{i+1}^2 + R_i^2 + R_{i+1}^2) \ \{$$

$$S_i = 0; \quad S_{i+1} = 0;$$

$$\}$$

The following difference channel threshold coefficients apply to the scalefactor bands for block type != 2 (´long MDCT transforms´):

sb	0	1	2	3	4	5	6	7	8	9
ssb	0.0	0.0	0.0	0.0	0.0	0.10	0.10	0.10	0.10	0.10

sb	10	11	12	13	14	15	16	17	18	19	20
ssb	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00	1.50

3) Intensity stereo processing

- Calculation of intensity stereo position

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

- is_pos_{sb} = NINT(* arctan( ))
- $L_i = L_i + R_i$ for all indices i within the actual scalefactor band sb
- $R_i = 0$ for all indices i within the actual scalefactor band sb
- the intensity stereo position is_pos_{sb} is transmitted instead of the scalefactor of the right channel (3 bits always, stereo positions 0..6, 7=illegal stereo position)

where $L\_Energy_{sb}/R\_Energy_{sb}$ denote the signal energies of the left/right channel within the actual scalefactor band and L_i/R_i are the transformed values.

Scalefactor bands of the right/difference channel containing only zeros after coding which do not belong to the intensity coded part should be transmitted with the scalefactor ´7´ to prevent intensity stereo decoding.

FIGURE 3-G.1 General Stereo Encoder Flow Chart

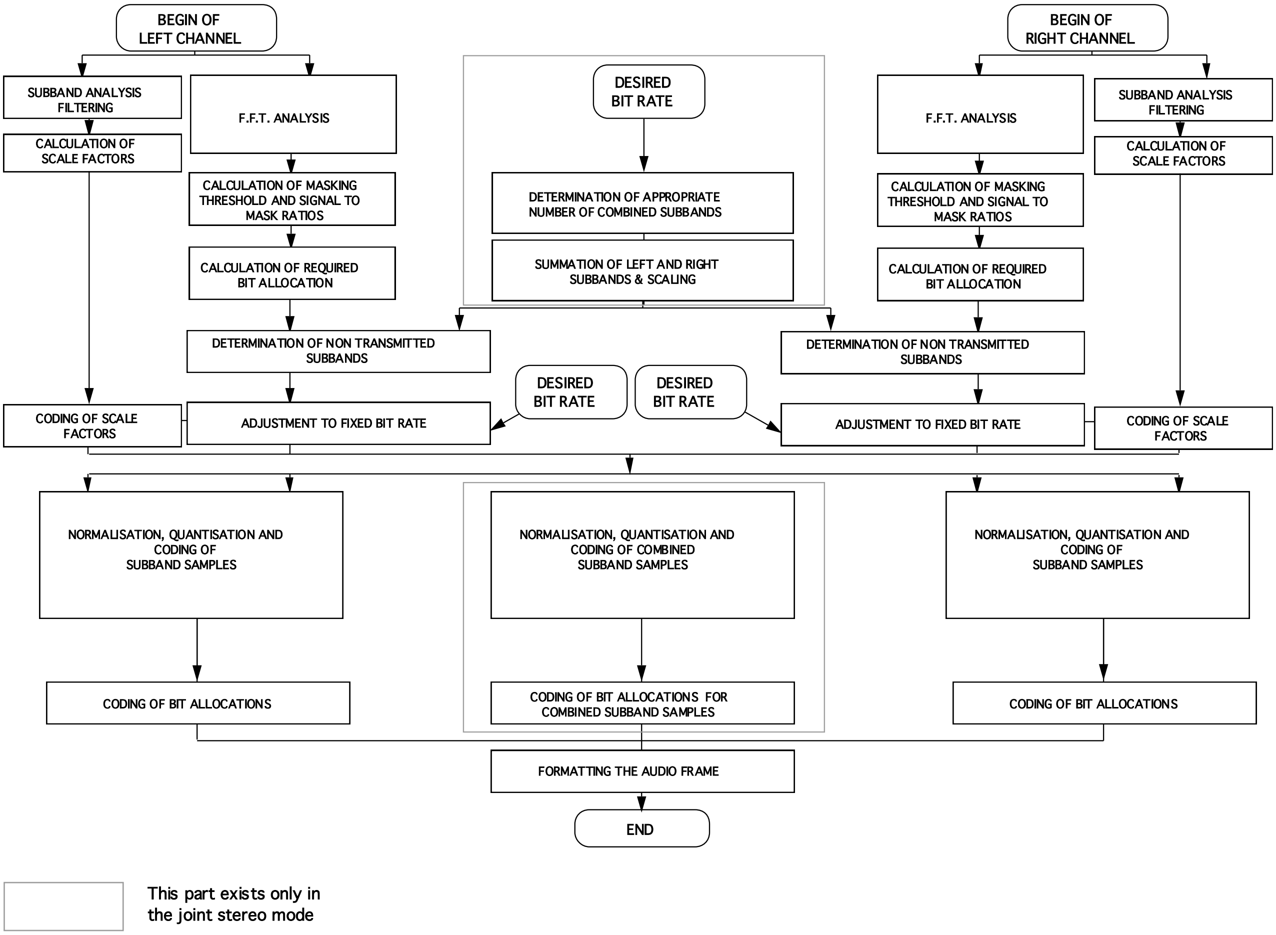
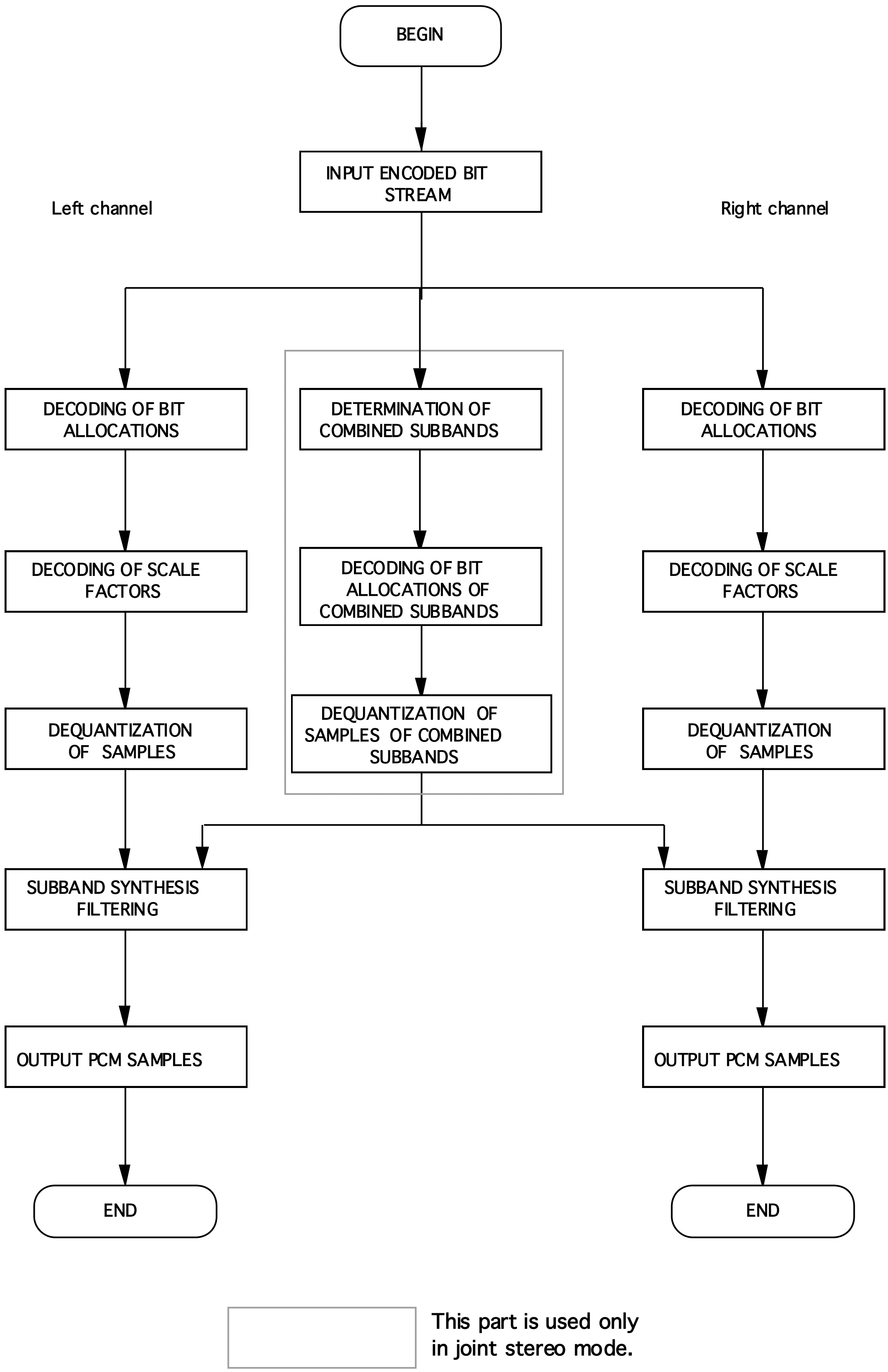


FIGURE 3-G.2 General Stereo Decoder Flow Chart


BEGIN
INPUT ENCODED BIT STREAM
Left channel
Right channel
DECODING OF BIT ALLOCATIONS
DETERMINATION OF COMBINED SUBBANDS
DECODING OF BIT ALLOCATIONS
DECODING OF SCALE FACTORS
DECODING OF BIT ALLOCATIONS OF COMBINED SUBBANDS
DECODING OF SCALE FACTORS
DEQUANTIZATION OF SAMPLES
DEQUANTIZATION OF SAMPLES OF COMBINED SUBBANDS
DEQUANTIZATION OF SAMPLES
SUBBAND SYNTHESIS FILTERING
SUBBAND SYNTHESIS FILTERING
OUTPUT PCM SAMPLES
OUTPUT PCM SAMPLES
END
END
This part is used only in joint stereo mode.