INTERNATIONAL STANDARD

ISO/IEC 13818-3

Second edition
1998-04-15

Information technology — Generic coding of moving pictures and associated audio information —

Part 3:
Audio

Technologies de l'information — Codage générique des images animées et des informations sonores associées —

Partie 3: Son

Reference number
ISO/IEC 13818-3:1998(E)

Contents

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

Printed in Switzerland.

Foreword

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

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

International Standard ISO/IEC 13818-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This second edition cancels and replaces the first edition (ISO/IEC 13818-3:1995), which has been technically revised.

ISO/IEC 13818 consists of the following parts, under the general title *Information technology — Generic coding of moving pictures and associated audio information*:

– *Part 1: Systems*

– *Part 2: Video*

– *Part 3: Audio*

– *Part 4: Compliance testing*

– *Part 5: Software simulation*

– *Part 6: Extensions for DSM-CC*

– *Part 7: Advanced Audio Coding (AAC)*

– *Part 9: Extension for real time interface for systems decoders*

– *Part 10: Conformance extensions for DSM-CC*

Annexes A and B form an integral part of this part of ISO/IEC 13818. Annexes C to I are for information only.

Introduction

ISO/IEC 13818 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.

ISO/IEC 13818 is published in three 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 synchronised 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 and the decoding process required to decode audio signals.

The technical changes in this 2nd edition compared to the first publication of ISO/IEC 13818-3 (1995) are:

1. In the first publication, certain combinations of dynamic crosstalk and prediction were not prohibited but not practically implementable. In this 2nd revision, these combinations are explicitly prohibited.
2. In the first publication, a low-pass filter was to be applied to the monophonic surround signal in matrix mode 2 (analogue surround mode). This filter is omitted in this edition, greatly simplifying the decoder.
3. The description of the syntax of the LFE channel was ambiguous. This description has been clarified.

Next to these technical changes, many editorial changes have been made, improving readability and clarity.

0.1 Extension of ISO/IEC 11172-3 Audio Coding to Lower Sampling Frequencies

In order to achieve better audio quality at very low bit rates (<64 kbit/s per audio channel), in particular if compared with ITU-T (formerly CCITT) Recommendation G.722 performance, three additional sampling frequencies are provided for ISO/IEC 11172-3 layers I, II and III. The additional sampling frequencies (Fs) are 16 kHz, 22,05 kHz and 24 kHz. This allows corresponding audio bandwidths of approximately 7,5 kHz, 10,3 kHz and 11,25 kHz. The syntax, semantics, and coding techniques of ISO/IEC 11172-3 are maintained except for a new definition of the sampling frequency field, the bitrate index field, and the bit allocation tables. These new definitions are valid if the ID bit in the ISO/IEC 11172-3 header equals zero. To obtain the best audio performance, the parameters of the psychoacoustic model used in the encoder have to be changed accordingly.

With these sampling frequencies, the duration of the audio frame corresponds to:

Layer	Sampling Frequency in kHz		
	16	22,05	24
I	24 ms	17,41.. ms	16 ms
II	72 ms	52,24.. ms	48 ms
III	36 ms	26,12.. ms	24 ms

0.2 Low bitrate coding of multichannel audio

0.2.1 Universal multichannel audio system

A standard on low bit rate coding for mono or stereo audio signals was established by MPEG-1 Audio in ISO/IEC 11172-3. This standard is applicable for carrying of high quality digital audio signals associated with or without picture information on storage media or transmission channels with limited capacity.

The ISO/IEC 11172-3 audio coding standard can be used together with both MPEG-1 and MPEG-2 Video as long as only two-channel stereo is required. MPEG-2 Audio (ISO/IEC 13818-3) provides the extension up to 3/2 multichannel audio and an optional low frequency enhancement channel (LFE).

This part of ISO/IEC 13818 describes an audio subband coding system called ISO/MPEG-Audio Multichannel, which can be used to transfer high quality digital multichannel and/or multilingual audio information on storage media or transmission channels with limited capacity. One of the basic features is the backwards compatibility to ISO/IEC 11172-3 coded mono, stereo or dual channel audio programmes. It is designed for use in different applications as considered by the ISO/MPEG audio group and the specialist groups TG 10/1, 10/2 and 10/3 of the ITU-R (previously CCIR).

Multichannel audio systems provide enhanced stereo performance compared to conventional two channel audio systems. It is recognised that improved presentation performance is desirable not only for applications with accompanying picture but also for audio-only applications. A universal and compatible multichannel audio system applicable to satellite or terrestrial television broadcasting, digital audio broadcasting (terrestrial and satellite), as well as other non-broadcasting media, e.g.,

CATV	Cable TV Distribution
CDAD	Cable Digital Audio Distribution
DAB	Digital Audio Broadcast
DVD	Digital Versatile Disc
ENG	Electronic News Gathering (including Satellite News Gathering)
HDTV	High Definition Television
IPC	Interpersonal Communications (video conference, videophone, etc.)
ISM	Interactive Storage Media (optical disks, etc.)
NDB	Network Database Services (via ATM, etc.)
DSM	Digital Storage Media (digital VTR, etc.)
EC	Electronic Cinema
HTT	Home Television Theatre
ISDN	Integrated Services Digital Network

seems to be very attractive to the manufacturer, producer and consumer.

0.2.2 Representation of multichannel audio

0.2.2.1 The 3/2-stereo plus LFE format

Regarding stereophonic presentation, specialist groups of ITU-R, SMPTE, and EBU recommend the use of an additional centre loudspeaker channel C and two surround loudspeaker channels LS and RS, augmenting the front left and right loudspeaker channels L and R. This reference audio format is referred to as "3/2-stereo" (3 front / 2 surround loudspeaker channels) and requires the transmission of five appropriately formatted audio signals.

For audio accompanying picture applications (e.g. HDTV), the three front loudspeaker channels ensure sufficient directional stability and clarity of the picture related frontal images, according to the common practice in the cinema. The dominant benefit is the "stable centre", which is guaranteed at any location of the listener and important for most of the dialogue.

Additionally, for audio-only applications, the 3/2-stereo format has been found to be an improvement over two-channel stereophony. The addition of one pair of surround loudspeaker channels allows improved realism of auditory ambience.

A low frequency enhancement channel (in this part of ISO/IEC 13818 called LFE channel) can, optionally, be added to any of these configurations. The purpose of this channel is to enable listeners to extend the low frequency content of the reproduced programme in terms of both frequency and level. In this way it is the same as the LFE channel proposed by the film industry for their digital sound systems.

The LFE channel should not be used for the entire low frequency content of the multichannel sound presentation. The LFE channel is optional at the receiver, and thus should only carry low frequency sound effects, which may have a high level. The LFE channel is not included in any dematrixing operation in the decoder. The sampling frequency of the LFE channel corresponds to the sampling frequency of the main channels, divided by a factor of 96. This provides 12 LFE samples within one audio frame. The LFE channel is capable of handling signals in the range from 15 Hz to 120 Hz.

0.2.2.2 Compatibility

Extension from 2/0-stereo towards multichannel sound.

As a result of the widespread use of conventional two-channel stereo (2/0-stereo) reproduction, compatibility with existing 2/0-stereo sound reproduction systems or with existing matrixed surround sound receivers has to be maintained. This means that for many applications a basic stereo signal which contains an appropriate downmix of the audio information of the multichannel programme has to be transmitted together with the multichannel audio information. Appropriate downmix equations are given by equation pairs (1,2), (3,4), (5,6) and (7,8).

$$Lo = L + \tfrac{1}{2}\sqrt{2} * C + \tfrac{1}{2}\sqrt{2} * LS \quad (1)$$
$$Ro = R + \tfrac{1}{2}\sqrt{2} * C + \tfrac{1}{2}\sqrt{2} * RS \quad (2)$$

or

$$Lo = L + \tfrac{1}{2}\sqrt{2} * C + \tfrac{1}{2} * LS \quad (3)$$
$$Ro = R + \tfrac{1}{2}\sqrt{2} * C + \tfrac{1}{2} * RS \quad (4)$$

or

$$Lo = L \quad (5)$$
$$Ro = R \quad (6)$$

or

$$Lo = L + \tfrac{1}{2}\sqrt{2} * C - \tfrac{1}{2}\sqrt{2} * jS \quad (7)$$
$$Ro = R + \tfrac{1}{2}\sqrt{2} * C + \tfrac{1}{2}\sqrt{2} * jS \quad (8)$$

where jS is derived from LS and RS by calculation of the mono component. Then, a dynamic range compression and 90 degrees phase shift are applied to this component. The downmix (7,8) is suitable for existing matrixed surround decoders.

The format of an ISO/IEC 13818-3 bit stream is such that an ISO/IEC 11172-3 audio decoder properly decodes the basic stereo information according to one of the sets of downmix equations above (see 0.2.3.1). Compatibility with existing surround sound decoders by use of equations (7) and (8) has not been verified at the time of printing of this part of ISO/IEC 13818.

In the case of this part of ISO/IEC 13818, three different possibilities can be identified to provide to the user a basic stereo downmix together with the multichannel audio information:

1. Transmitting the 2/0-stereo sound inherently with the multichannel information in one bit stream in a backwards compatible way with ISO/IEC 11172-3, thus avoiding simulcast. This allows for the most efficient use of bit rate required for both, the 2/0-stereo and the multichannel audio signal. Additional advantages are that both programmes are strictly synchronized on a PCM audio sample basis, and that audio programme associated data carried in the ancillary data field of the MPEG-Audio bit stream have to be transmitted only once. The stereo downmix from the multichannel audio signal is handled by the ISO/IEC 13818-3 encoder. For this downmix, a number of matrix options according to equations (1) and (2) and equations (3) and (4) are provided by this part of ISO/IEC 13818 (see 2.5.2.13).

2. Simulcast of the multichannel audio signal, coded according to this part of ISO/IEC 13818, together with the 2/0-stereo signal coded according to ISO/IEC 11172-3. This solution requires two independent bit streams which can be multiplexed and transmitted by ISO/IEC 13818-1. The programme provider has to make provisions if a synchronization of both bit streams is required. Further, the simulcast option requires a significantly higher bit rate because instead of 5 channels in the case of 3/2 multichannel sound, altogether 7 audio channels have to be transmitted. However, the simulcast option allows for an individual, i.e. dynamic downmix to 2/0-stereo sound which can be controlled by a sound engineer.

3. Transmitting only the multichannel signal, by using the non-matrixed mode (downmix equation (5,6)). Each stereo decoder has then to be able to decode all the five channels, and to make a stereo downmix. Although the downmix can be applied before the filtering operation in the decoder, and the filter only needs to be done on two channels, this complicates the decoder significantly.

If compatibility with existing matrixed surround sound decoders is required, this part of ISO/IEC 13818 again provides three solutions:

1. To ensure a high efficiency regarding the bit rate required for both, the 3/2-multichannel and the matrixed surround signal, this surround signal can be transmitted in the backwards compatible stereo channel. The matrix-option '10' according to equations (7) and (8) provides an appropriate compatible signal which is transmitted in the basic stereo channels. A matrixed surround signal, suitable for existing matrixed surround decoders, can be obtained at the receiver by using an ISO/IEC 11172-3 two-channel decoder. The corresponding 3/2-channel output can be derived by using an ISO/IEC 13818-3 decoder.

2. A higher bit rate is necessary for simulcast of a matrixed surround signal using ISO/IEC 11172-3 and a 3/2-multichannel audio signal using this part of ISO/IEC 13818. This simulcast option allows for an independent mix of the matrixed surround signal which can be controlled by a sound engineer. The

drawback of this solution is the additional bit rate necessary for transmitting 7 audio channels instead of only five channels if matrix-option '10' (see 2.5.2.13) is used.

3. Transmitting only the multichannel signal, by using the non-matrixed mode. Each stereo decoder has then to be able to decode all the five channels, and to make the downmix according to equation (7,8). Although the downmix can be applied before the filtering operation in the decoder, and the filter only needs to be done on two channels, this complicates the decoder significantly.

Downwards compatibility.

A hierarchy of audio formats providing a lower number of loudspeaker channels and reduced presentation performance (down to 2/0-stereo or even mono) and a corresponding set of downwards mixing equations are recommended in ITU-R Recommendation 775: "Multichannel stereophonic audio system with and without accompanying picture", November 1992. Alternative lower level audio formats which may be used in circumstances where economic or channel capacity constraints apply, are 3/1, 3/0, 2/2, 2/1, 2/0, and 1/0. Corresponding loudspeaker arrangements are 3/2, 3/1, 3/0, 2/2, 2/1, 2/0, and 1/0.

Backwards compatibility.

For several applications, the intention is to extend the existing 2/0-stereo sound system by transmitting additional audio channels (centre, surround) without making use of simulcast operation. This provision of backwards compatibility with existing receivers implies the use of compatibility matrices: the decoder of the previous generation must reproduce the two conventional basic stereo signals L´o/R´o, and the multichannel decoder produces the complete 3/2-stereo presentation L´/C´/R´/LS´/RS´ from the basic stereo signal and the extension signals.

It is recognised that backward compatibility may not be required for all applications of MPEG-2 Audio. Therefore, nonbackward compatible (NBC) audio coding systems free of the constraints of backwards compatibility are being evaluated for optional use with this part of ISO/IEC 13818.

0.2.2.3 Multilingual capability

Particularly for HDTV applications, multichannel stereo performance and bilingual programmes or multilingual commentaries are required. This part of ISO/IEC 13818 provides for alternative audio channel configurations in the five-channel sound system, for example a bilingual 2/0 stereo programme or one 2/0, 3/0 stereo sound plus accompanying services (e.g. "clean dialogue" for the hard-of-hearing, commentary for the visually impaired, multilingual commentary etc.). An important configuration is the reproduction of commentary dialogue (e.g. via centre loudspeaker) together with the common music/effect stereo downmix (examples are documentation film, sport reports).

0.2.3 Basic Parameters of the Multichannel Audio Coding System

The transmission of the five audio signals of a 3/2 sound system requires five transmission channels (although, in the context of bitrate reduced signals, these are not necessarily independent). In order that two of the transmitted signals can provide a stereo service on their own, the source sound signals are generally combined in a linear matrix prior to encoding. These combined signals (and their transmission channels) are identified by the notation T0, T1, T2, T3 and T4.

0.2.3.1 Compatibility with ISO/IEC 11172-3

The ISO/MPEG-Audio Multichannel system provides full compatibility with ISO/IEC 11172-3. For a multichannel audio bit stream, backwards compatibility means, that an ISO/IEC 11172-3 audio decoder properly decodes the basic stereo information (see 0.2.2.2). Forwards compatibility means that an MPEG-2 multichannel audio decoder is able to decode properly an ISO/IEC 11172-3 audio bit stream.

The backwards compatibility is realised by coding the basic stereo information in conformance with ISO/IEC 11172-3 and exploiting the ancillary data field of the ISO/IEC 11172-3 audio frame (base frame, in the context of this part of ISO/IEC 13818) plus an optional extension frame for the multichannel extension.

The complete ISO/IEC 11172-3 audio frame incorporates four different types of information:

- Header information within the first 32 bits of the ISO/IEC 11172-3 audio frame.
- Cyclic Redundancy Check (CRC), consisting of 16 bits, just after the header information (optional).

- Audio data, for Layer II consisting of bit allocation (BAL), scalefactor select information (SCFSI), scalefactors (SCF), and the subband samples.
- Ancillary data. Due to the large number of different applications which will use this part of ISO/IEC 13818, the length and usage of this field are not specified.

The variable length of the ancillary data field enables packing the complete extension information of the channels T2/T3/T4 into the first part of the ancillary data field. If the MC encoder does not use all of the ancillary data field for the multichannel extension information, the remaining part of the field can be used for other ancillary data.

The bit rate required for the multichannel extension information may vary on a frame by frame basis, depending on the sound signals. The overall bit rate may be increased above that provided for in ISO/IEC 11172-3 by the use of an optional extension bit stream. The maximum bit rate, including the extension bit stream, is given by the following table:

Sampling Frequency	Layer	Maximum Total Bit Rate
32 kHz	I	903 kbit/s
32 kHz	II	839 kbit/s
32 kHz	III	775 kbit/s
44.1 kHz	I	1075 kbit/s
44.1 kHz	II	1011 kbit/s
44.1 kHz	III	947 kbit/s
48 kHz	I	1130 kbit/s
48 kHz	II	1066 kbit/s
48 kHz	III	1002 kbit/s

This part of ISO/IEC 13818 describes the combinations of the basic Lo, Ro stereo of Layer I, II and III and the multichannel extension of Layer II mc and Layer III mc. The following combinations are possible:

Basic Lo, Ro Stereo	Multichannel Extension
Layer II	Layer II mc
Layer III	Layer III mc
Layer I	Layer II mc

0.2.3.2 Audio Input/Output Format

Sampling frequencies: 48, 44.1 or 32 kHz
Quantisation: up to 24 bits/sample PCM resolution

The following combinations of audio channels can be applied as inputs to the audio encoder:

a) Five channels, using the 3/2 configuration
L, C, R plus two surround channels LS, RS

b) Four channels, using the 3/1 configuration
L, C, R plus single surround channel S

c) Three channels using the 3/0 configuration
L, C, R without surround

d) Five channels, using the 3/0 + 2/0 configuration
L, C, R of first programme plus L2, R2 of second programme

e) Four channels, using the 2/2 configuration
L, R plus two surround channels LS, RS

f) Three channels using the 2/1 configuration
L, R with single surround channel S

g) Two channels, using the 2/0 (or 1/0+1/0) configuration
Stereo (or dual channel mode) as in ISO/IEC 11172-3

h) Four channels, using the 2/0 + 2/0 (or 1/0+1/0+ 2/0) configuration
L, R (or channel I and channel II) of first programme plus L2, R2 of second programme

i) One channel, using the 1/0 configuration
Single channel mode (as in ISO/IEC 11172-3)

j) Three channels, using the 1/0 + 2/0 configuration
Single channel mode (as in ISO/IEC 11172-3) plus L2, R2 of second programme

The different combinations of audio input signals are encoded and transmitted within the up to five available transmission channels T0, T1, T2, T3 and T4, of which channels T0 and T1 are the two basic channels of ISO/IEC 11172-3 and convey the backwards compatible signals Lo and Ro. Transmission channels T2, T3 and T4 together form the multichannel extension information, which is compatibly transmitted within the ISO/IEC 11172-3 ancillary data field and an optional extension bit stream.

After multichannel decoding, the up to five audio channels are recovered and can then be presented in any convenient format at the choice of the listeners:

a) Five channels, using the 3/2 configuration
Front: Left (L) and right (R) channel plus centre channel (C)
Surround: Left surround (LS) and right surround (RS)

b) Four channels, using the 3/1 configuration
Front: Left (L) and right (R) channel plus centre channel (C)
Surround: Mono surround (S)

c) Three channels using the 3/0 configuration
Front: Left (L) and right (R) channel plus centre channel (C)
Surround: No surround

d) Four channels, using the 2/2 configuration
Front: Left (L) and right (R) channel
Surround: Left surround (LS) and right surround (RS)

e) Three channels, using the 2/1 configuration
Front: Left (L) and right (R) channel
Surround: Mono surround (S)

f) Two channels, using the 2/0 configuration
Front: Left (L) and right channel (R)
Surround: No surround

g) One channel output, using the 1/0 configuration
Front: Mono channel (Mo)
Surround: No surround

A low frequency enhancement channel can, optionally, be added to any of these configurations, except for the 1/0 configuration.

Outputs may be required to provide discrete signals, or may be combined in accordance with downward mixing, or upwards conversion equations, as defined in ITU-R Recommendation 775.

0.2.3.3 Composite Coding Modes

Dynamic Transmission Channel Switching

In order to provide a better orthogonality between the two compatible signals T0 and T1, and the three additionally transmitted signals T2, T3 and T4, it is necessary to have flexibility in the choice of the channels T2, T3 and T4. This part of ISO/IEC 13818 allows, independently for a number of frequency regions, the selection of a number of combinations of three out of the five signals L, C, R, LS, RS to be transmitted in T2, T3 and T4.

Dynamic Crosstalk

According to a binaural hearing model, it is possible to determine the portion of the stereophonic signal which is irrelevant with respect to the spatial perception of the stereophonic presentation. The stereo-irrelevant signal components are not masked, but they do not contribute to the localisation of sound sources. They are ignored in the binaural processor of the human auditory system. Therefore, stereo-irrelevant components of any stereo signal (L, C, R, LS or RS) may be reproduced via any loudspeaker, or via several loudspeakers of the arrangement, without affecting the stereophonic impression. This can be done independently for a number of frequency regions.

Adaptive Multichannel Prediction

In order to make use of the statistical inter-channel dependencies, adaptive multichannel prediction is used for redundancy reduction. Instead of transmitting the actual signals in the transmission channels T2, T3, T4, the corresponding prediction error signals are transmitted. A predictor of up to 2nd order with delay compensation is used.

Phantom Coding of Centre

Due to the fact that the human auditory system uses only intensity cues of the audio signal for localisation at higher frequencies, it is possible to transmit the high frequency part of the centre channel in the front left and right channels, constituting a phantom source at the location of the centre loudspeaker.

0.2.3.4 Encoder and Decoder Parameters

Encoding and decoding:	similar to ISO/IEC 11172-3.	
Coding modes:	3/2, 3/1, 3/0 (+ 2/0), 2/2, 2/1, 2/0 (+ 2/0), 1/0+1/0 (+ 2/0), 1/0 (+ 2/0) second stereo programme, up to 7 additional multilingual or commentary channels, associated services.	
Subband filter transforms:	Number of subbands: Sampling frequency: Bandwidth of subbands:	32 Fs/32 Fs/64
Additional decomposition by MDCT (Layer III only):	Frequency Resolution:	6 or 18 components per subband
LFE channel filter transform:	Number of LFE channels: Sampling frequency: Bandwidth of LFE channel:	1 Fs/96 125 Hz
Dynamic range:	more than 20 bits.	

Information technology — Generic coding of moving pictures and associated audio information —

Part 3:

Audio

Section 1: General

1.1 Scope

This part of ISO/IEC 13818 specifies the extension of ISO/IEC 11172-3 to lower sampling frequencies, the coded representation of multichannel and multilingual high quality audio for broadcasting, transmission and storage media, and the method for decoding of multichannel and multilingual high quality audio signals. The input of the encoder and the output of the decoder are compatible with existing PCM standards.

1.2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 13818. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 13818 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 11172-3: 1993, *Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 3: Audio.*

CCIR Recommendation 601-1: 1990, *Encoding parameters of digital television for studios*.

CCIR Recommendation 648: 1986, *Recording of audio signals*.

CCIR Recommendation 775: 1992, *Multichannel stereophonic sound system with and without accompanying picture*.

CCIR Report 955-2: 1990, *Sound broadcasting by satellite for portable and mobile receivers*, including Annex IV *Summary description of Advanced Digital System II.*

IEC 908: 1987, *Compact disc digital audio system*.

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

ITU-T Recommendation G.722: 1988, *7 kHz audio coding within 64 kbit/s*.

European Telecommunication Standard pr ETS 300 401: 1995, *Radio Broadcasting system; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers.*

ITU-T Recommendation J.52: 1995, *Digital Transmission of High Quality Sound Programme Signals Using One, Two or Three 64 kbit/s Channels per Mono Signal(and up to Six per Stereo Signal).*

Section 2: Technical elements

2.1 Definitions

For the purposes of this part of ISO/IEC 13818, the following definitions apply. If specific to a part, this is noted in square brackets.

2.1.1 16x8 prediction [video]: A prediction mode similar to field-based prediction but where the predicted block size is 16x8 luminance samples.

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

2.1.3 access unit [system]: A coded representation of a presentation unit. In the case of audio, an access unit is the coded representation of an audio frame.
In the case of video, an access unit includes all the coded data for a picture, and any stuffing that follows it, up to but not including the start of the next access unit. If a picture is not preceded by a group_start_code or a sequence_header_code, the access unit begins with the picture start code. If a picture is preceded by a group_start_code and/or a sequence_header_code, the access unit begins with the first byte of the first of these start codes. If it is the last picture preceding a sequence_end_code in the bitstream all bytes between the last byte of the coded picture and the sequence_end_code (including the sequence_end_code) belong to the access unit.

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

2.1.5 adaptive multichannel prediction [audio]: A method of multichannel data reduction exploiting statistical inter-channel dependencies.

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

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

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

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

2.1.10 ancillary data [audio]: part of the bitstream that might be used for transmission of ancillary data.

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

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

2.1.13 audio sequence [audio]: A non-interrupted series of audio frames (base frames plus optional extension frames) in which the following parameters are not changed:

- ID
- Layer
- Sampling Frequency

For Layer I and II, a decoder is not required to support a continuously variable bitrate (change in the bitrate index) of the base stream. Such a relaxation of requirements does not apply to the extension stream.

2.1.14 B-field picture [video]: A field structure B-Picture.

2.1.15 B-frame picture [video]: A frame structure B-Picture.

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

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

2.1.18 backward compatibility: A newer coding standard is backward compatible with an older coding standard if decoders designed to operate with the older coding standard are able to continue to operate by decoding all or part of a bitstream produced according to the newer coding standard.

2.1.19 backward motion vector [video]: A motion vector that is used for motion compensation from a reference frame or reference field at a later time in display order.

2.1.20 backward prediction [video]: Prediction from the future reference frame (field).

2.1.21 base bit stream [audio]: Information contained in a bit stream which consists of continuous base frames. This bit stream is decodable by an ISO/IEC 11172-3 and an ISO/IEC 13818-3 decoder. An ISO/IEC 13818-3 bit stream shall always consist of the base bit stream and optionally of an extension bit stream.

2.1.22 base frame [audio]: The part of the ISO/IEC 13818-3 encoded audio frame which can be decoded by an ISO/IEC 11172-3 decoder and contains the basic stereo signal.

2.1.23 base layer [video]: First, independently decodable layer of a scalable hierarchy.

2.1.24 big picture [video]: A coded picture that would cause VBV buffer underflow as defined in C.7 Annex C of ISO/IEC 13818-2. Big pictures can only occur in sequences where low_delay is equal to 1. "Skipped picture" is a term that is sometimes used to describe the same concept.

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

2.1.26 bitstream; stream: An ordered series of bits that forms the coded representation of the data.

2.1.27 bitstream verifier [video]: A process by which it is possible to test and verify that all the requirements specified in ISO/IEC 13818-2 are met by the bitstream.

2.1.28 block [video]: An 8-row by 8-column matrix of samples, or 64 DCT coefficients (source, quantised or dequantised).

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

2.1.30 bottom field [video]: One of two fields that comprise a frame. Each line of a bottom field is spatially located immediately below the corresponding line of the top field.

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

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

2.1.33 byte: Sequence of 8-bits.

2.1.34 centre channel [audio]: An audio presentation channel used to stabilise the central component of the frontal stereo image.

2.1.35 channel [audio]: A sequence of data representing an audio signal being transported.

2.1.36 chroma simulcast [video]: A type of scalability (which is a subset of SNR scalability) where the enhancement layer(s) contain only coded refinement data for the DC coefficients, and all the data for the AC coefficients, of the chrominance components.

2.1.37 chrominance format [video]: Defines the number of chrominance blocks in a macroblock.

2.1.38 chrominance component [video]: A matrix, block or single sample representing one of the two colour difference signals related to the primary colours in the manner defined in the bitstream. The symbols used for the chrominance signals are Cr and Cb.

2.1.39 coded audio bitstream [audio]: A coded representation of an audio signal as specified in this part of ISO/IEC 13818.

2.1.40 coded B-frame [video]: A B-frame picture or a pair of B-field pictures.

2.1.41 coded frame [video]: A coded frame is a coded I-frame, a coded P-frame or a coded B-frame.

2.1.42 coded I-frame [video]: An I-frame picture or a pair of field pictures, where the first field picture is an I-picture and the second field picture is an I-picture or a P-picture.

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

2.1.44 coded P-frame [video]: A P-frame picture or a pair of P-field pictures.

2.1.45 coded picture [video]: A coded picture is made of a picture header, the optional extensions immediately following it, and the following picture data. A coded picture may be a coded frame or a coded field.

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

2.1.47 coded video bitstream [video]: A coded representation of a series of one or more pictures as defined in ISO/IEC 13818-2.

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

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

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

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

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

2.1.53 constrained system parameter stream; CSPS [system]: A Program Stream for which the constraints defined in 2.7.9 of ISO/IEC 13818-1 apply.

2.1.54 CRC: The Cyclic Redundancy Check to verify the correctness of data.

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

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

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

2.1.58 data partitioning [video]: A method for dividing a bitstream into two separate bitstreams for error resilience purposes. The two bitstreams have to be recombined before decoding.

2.1.59 DC coefficient [video]: The DCT coefficient for which the frequency is zero in both dimensions.

2.1.60 DCT coefficient [video]: The amplitude of a specific cosine basis function.

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

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

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

2.1.64 decoder: An embodiment of a decoding process.

2.1.65 decoder sub-loop [video]: Stages within encoder which produce numerically identical results to the decode process described in ISO/IEC 13818-2, clause 7. Encoders capable of producing more than just I-pictures embed a decoder sub-loop to create temporal predictions and to model the behaviour of downstream decoders.

2.1.66 decoding (process): The process defined in ISO/IEC 13818 parts 1, 2 and 3 that reads an input coded bitstream and outputs decoded pictures or audio samples.

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

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

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

2.1.70 discrete cosine transform; DCT: Either the forward discrete cosine transform or the inverse discrete cosine transform. The DCT is an invertible, discrete orthogonal transformation.

2.1.71 display aspect ratio [video]: The ratio height/width (in SI units) of the intended display.

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

2.1.73 display process [video]: The (non-normative) process by which reconstructed frames are displayed.

2.1.74 downmix [audio]: A matrixing of n channels to obtain less than n channels.

2.1.75 drift [video]: Accumulation of mismatch between the reconstructed output produced by the hypothetical decoder sub-loop embedded within an encoder (see definition of "decoder sub-loop") and the reconstructed outputs produced by a (downstream) decoder.

2.1.76 DSM-CC: digital storage media command and control.

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

2.1.78 dual-prime prediction [video]: A prediction mode in which two forward field-based predictions are averaged. The predicted block size is 16x16 luminance samples. Dual-prime prediction is only used in interlaced P-pictures.

2.1.79 dynamic crosstalk [audio]: A method of multichannel data reduction in which stereo-irrelevant signal components are copied to another channel.

2.1.80 dynamic transmission channel switching [audio]: A method of multichannel data reduction by allocating the most orthogonal signal components to the transmission channels.

2.1.81 editing: The process by which one or more coded bitstreams are manipulated to produce a new coded bitstream. Conforming edited bitstreams must meet the requirements defined in parts 1, 2, and 3 of ISO/IEC 13818.

2.1.82 Elementary Stream Clock Reference; ESCR [system]: A time stamp in the PES Stream from which decoders of PES streams may derive timing.

2.1.83 elementary stream; ES [system]: A generic term for one of the coded video, coded audio or other coded bitstreams in PES packets. One elementary stream is carried in a sequence of PES packets with one and only one stream_id.

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

2.1.85 encoder: An embodiment of an encoding process.

2.1.86 encoding (process): A process, not specified in ISO/IEC 13818, that reads a stream of input pictures or audio samples and produces a valid coded bitstream as defined in parts 1, 2, and 3 of ISO/IEC 13818.

2.1.87 enhancement layer [video]: A relative reference to a layer (above the base layer) in a scalable hierarchy. For all forms of scalability, its decoding process can be described by reference to the lower layer decoding process and the appropriate additional decoding process for the enhancement layer itself.

2.1.88 entitlement control message; ECM [systems]: Entitlement Control Messages are private conditional access information which specify control words and possibly other, typically stream-specific, scrambling and/or control parameters.

2.1.89 entitlement management message; EMM [systems]: Entitlement Management Messages are private conditional access information which specify the authorisation levels or the services of specific decoders. They may be addressed to single decoders or groups of decoders.

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

2.1.91 event [systems]: An event is defined as a collection of elementary streams with a common time base, an associated start time, and an associated end time.

2.1.92 evil bitstreams: bitstreams orthogonal to reality.

2.1.93 extension bitstream [audio]: Information contained in an optional additional bit stream related to the audio base bit stream at the system level, to support bit rates beyond those defined in ISO/IEC 11172-3. The optional extension bit stream contains the remainder of the multichannel and multilingual data.

2.1.94 extension frame [audio]: The part of the ISO/IEC 13818-3 encoded audio frame which can be decoded only by an ISO/IEC 13818-3 decoder. This optional frame contains the remainder of the multichannel and multilingual data, as well as optional ancillary data.

2.1.95 fast reverse playback [video]: The process of displaying the picture sequence in the reverse of display order faster than real-time.

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

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

2.1.98 field [video]: For an interlaced video signal, a “field” is the assembly of alternate lines of a frame. Therefore an interlaced frame is composed of two fields, a top field and a bottom field.

2.1.99 field period [video]: The reciprocal of twice the frame rate.

2.1.100 field picture; field structure picture [video]: A field structure picture is a coded picture with picture_structure is equal to "Top field" or "Bottom field".

2.1.101 field-based prediction [video]: A prediction mode using only one field of the reference frame. The predicted block size is 16x16 luminance samples. Field-based prediction is not used in progressive frames.

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

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

2.1.104 flag: A variable which can take one of only the two values defined in this specification.

2.1.105 FLC: Fixed Length Code.

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

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

2.1.108 forward compatibility: A newer coding standard is forward compatible with an older coding standard if decoders designed to operate with the newer coding standard are able to decode bitstreams of the older coding standard.

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

2.1.110 forward prediction [video]: Prediction from the past reference frame (field).

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

2.1.112 frame [video]: A frame contains lines of spatial information of a video signal. For progressive video, these lines contain samples starting from one time instant and continuing through successive lines to the bottom of the frame. For interlaced video a frame consists of two fields, a top field and a bottom field. One of these fields may be temporally located one field period later than the other.

2.1.113 frame period [video]: The reciprocal of the frame rate.

2.1.114 frame picture; frame structure picture [video]: A frame structure picture is a coded picture with picture_structure is equal to "Frame".

2.1.115 frame rate [video]: The rate at which frames are be output from the decoding process.

2.1.116 frame reordering [video]: The process of reordering the reconstructed frames when the coded order is different from the display order. Frame reordering occurs when B-frames are present in a bitstream. There is no frame reordering when decoding low delay bitstreams.

2.1.117 frame-based prediction [video]: A prediction mode using both fields of the reference frame.

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

2.1.119 Fs [audio]: Sampling Frequency, as defined in 2.1.219

2.1.120 future reference frame (field) [video]: A future reference frame(field) is a reference frame(field) that occurs at a later time than the current picture in display order.

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

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

2.1.123 group of pictures [video]: A notion defined only in ISO/IEC 11172-2 (MPEG-1 Video). In ISO/IEC 13818-2, a similar functionality can be achieved by the mean of inserting group of pictures headers.

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

2.1.125 header: A block of data in the coded bitstream containing the coded representation of a number of data elements pertaining to the coded data that follow the header in the bitstream.

2.1.126 Huffman coding: A specific method for entropy coding.

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

2.1.128 hybrid scalability [video]: Hybrid scalability is the combination of two (or more) types of scalability.

2.1.129 I-field picture [video]: A field structure I-Picture.

2.1.130 I-frame picture [video]: A frame structure I-Picture.

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

2.1.132 IDCT: Inverse Discrete Cosine Transform.

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

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

2.1.135 interlace [video]: The property of conventional television frames where alternating lines of the frame represent different instances in time. In an interlaced frame, one of the field is meant to be displayed first. This field is called the first field. The first field can be the top field or the bottom field of the frame.

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

2.1.137 ITU-T Rec. H.222.0 | ISO/IEC 13818 (multiplexed) stream [system]: A bitstream composed of 0 or more elementary streams combined in the manner defined in ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

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

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

2.1.140 layer [audio]: One of the levels in the coding hierarchy of the audio system defined in this part of ISO/IEC 13818.

2.1.141 layer [systems]: One of the levels in the data hierarchy of the video and system specifications defined in ISO/IEC 13818 parts 1 and 2.

2.1.142 layer [video]: In a scalable hierarchy denotes one out of the ordered set of bitstreams and (the result of) its associated decoding process (implicitly including decoding of **all** layers below this layer).

2.1.143 layer bitstream [video]: A single bitstream associated to a specific layer (always used in conjunction with layer qualifiers, e.g. "enhancement layer bitstream").

2.1.144 level [video]: A defined set of constraints on the values which may be taken by the parameters of this specification within a particular profile. A profile may contain one or more levels. In a different context, level is the absolute value of a non-zero coefficient (see “run”).

2.1.145 LFE [audio]: Low Frequency Enhancement channel. A limited bandwidth channel for low frequency audio effects in a multichannel system.

2.1.146 low frequency enhancement channel [audio]: A limited bandwidth channel for low frequency audio effects in a multichannel system.

2.1.147 lower layer [video]: A relative reference to the layer immediately below a given enhancement layer (implicitly including decoding of **all** layers below this enhancement layer).

2.1.148 luminance component [video]: A matrix, block or single sample representing a monochrome representation of the signal and related to the primary colours in the manner defined in the bitstream. The symbol used for luminance is Y.

2.1.149 macroblock [video]: The four 8 by 8 blocks of luminance data and the two (for 4:2:0 chrominance format), four (for 4:2:2 chrominance format) or eight (for 4:4:4 chrominance format) 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 sample data and sometimes to the coded representation of the sample values and other data elements defined in the macroblock header of the syntax defined in this part of this specification. The usage is clear from the context.

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

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

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

2.1.153 Mbit [video]: 1 000 000 bits.

2.1.154 MCP [video]: Motion Compensated Predictor.

2.1.155 MDCT [audio]: Modified Discrete Cosine Transform which corresponds to the Time Domain Aliasing Cancellation Filter Bank.

2.1.156 mismatch [video]: Numerical discrepancy between the data reconstructed from the same coded bitstream by two decoding processes. With the exception of IDCT, the specification of ISO/IEC 13818-2 defines the decoding process absolutely unambiguously. Therefore, if both decoding processes are implemented according the specifications ISO/IEC 13818-2, mismatch can only be caused by different implementations of IDCT.

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

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

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

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

2.1.161 multichannel [audio]: A combination of audio channels used to create a spatial sound field.

2.1.162 multilingual [audio]: A presentation of dialogue in more than one language.

2.1.163 NIT [systems]: Network Information Table as defined in table 2-23 of ISO/IEC 13818-1.

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

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

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

2.1.167 opposite parity [video]: The opposite parity of top is bottom, and vice versa.

2.1.168 P-field picture [video]: A field structure P-Picture.

2.1.169 P-frame picture [video]: A frame structure P-Picture.

2.1.170 P-picture; predictive-coded picture [video]: A picture that is coded using motion compensated prediction from past reference fields or frame.

2.1.171 pack [system]: A pack consists of a pack header followed by zero or more packets. It is a layer in the system coding syntax described in 2.5.3.3 on page 51 of ISO/IEC 13818-1.

2.1.172 packet [system]: A packet consists of a header followed by a number of contiguous bytes from an elementary data stream. It is a layer in the system coding syntax described in 2.4.3 of ISO/IEC 13818-1.

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

2.1.174 packet identifier; PID [system]: A unique integer value used to associate elementary streams of a program in a single or multi-program Transport Stream as described in 2.4.3 of ISO/IEC 13818-1.

2.1.175 padding [audio]: 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.

2.1.176 parameter: A variable within the syntax of this specification which may take one of a range of values. A variable which can take one of only two values is a flag or indicator and not a parameter.

2.1.177 parity (of field) [video]: The parity of a field can be top or bottom.

2.1.178 parser: Functional stage of a decoder which extracts from a coded bitstream series of bits representing coded elements (FLC or VLC).

2.1.179 past reference frame (field) [video]: A past reference frame(field) is a reference frame(field) that occurs at an earlier time than the current picture in display order.

2.1.180 PAT [systems]: Program Association Table as defined in 2.4.4.3 of ISO/IEC 13818-1.

2.1.181 payload [systems]: Payload refers to the bytes which follow the header bytes in a packet. For example, the payload of a Transport Stream packet includes the PES_packet_header and its PES_packet_data_bytes, or pointer_field and PSI sections, or private data; but a PES_packet_payload consists of only PES_packet_data_bytes. The Transport Stream packet header and adaptation fields are not payload.

2.1.182 PES [system]: An abbreviation for Packetized Elementary Stream.

2.1.183 PES packet [system]: The data structure used to carry elementary stream data. It consists of a PES packet header followed by PES packet payload and is described in 2.4.3.6 and 2.4.3.7 of ISO/IEC 13818-1.

2.1.184 PES packet header[system]: The leading fields in a PES packet up to and not including the PES_packet_data_byte fields, where the stream is not a padding stream. In the case of a padding stream the PES packet header is similarly defined as the leading fields in a PES packet up to and not including padding_byte fields.

2.1.185 PES Stream [system]: A PES Stream consists of PES packets, all of whose payloads consist of data from a single elementary stream, and all of which have the same stream_id. Specific semantic constraints apply.

2.1.186 picture [video]: Source, coded or reconstructed image data. A source or reconstructed picture consists of three rectangular matrices of 8-bit numbers representing the luminance and two chrominance signals. A “coded picture” is defined in ISO/IEC 13818-2. For progressive video, a picture is identical to a frame, while for interlaced video, a picture can refer to a frame, or the top field or the bottom field of the frame depending on the context.

2.1.187 picture data [video]: In the VBV operations, picture data is defined as all the bits of the coded picture, all the header(s) and user data immediately preceding it if any (including any stuffing between them) and all the stuffing following it, up to (but not including) the next start code, except in the case where the next start code is an end of sequence code, in which case it is included in the picture data.

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

2.1.189 prediction [audio]: The use of a predictor to provide an estimate of the subband sample in one channel from the subband samples in other channels.

2.1.190 prediction error: The difference between the actual value of a sample or data element and its predictor.

2.1.191 prediction: The use of a predictor to provide an estimate of the sample value or data element currently being decoded.

2.1.192 predictor: A linear combination of previously decoded sample values or data elements.

2.1.193 presentation channel [audio]: audio channels at the output of the decoder corresponding to the loudspeaker positions left, centre, right, left surround and right surround.

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

2.1.195 presentation unit; PU [system]: A decoded Audio Access Unit or a decoded picture.

2.1.196 profile [video]: A defined subset of the syntax of this specification.

2.1.197 profile-and-level combination [video]: Point of conformance for video bitstreams and decoders. Defined profile-and-level combinations are defined in clause 8 of ISO/IEC 13818-2. In the case of a bitstream, the profile-and-level combination is derived from the profile_and_level_indication. A decoder may comply with several profile-and level combinations.

2.1.198 program [system]: A program is a collection of program elements. Program elements may be elementary streams. Program elements need not have any defined time base; those that do, have a common time base and are intended for synchronised presentation.

2.1.199 Program Clock Reference; PCR [system]: A time stamp in the Transport Stream from which decoder timing is derived.

2.1.200 program element[system]: A generic term for one of the elementary streams or other data streams that may be included in a program.

2.1.201 Program Specific Information; PSI [system]: PSI consists of normative data which is necessary for the demultiplexing of Transport Streams and the successful regeneration of programs and is described in 2.4.4 of ISO/IEC 13818-1. One case of PSI, the non-mandatory network information table, is privately defined.

2.1.202 progressive [video]: The property of film frames where all the samples of the frame represent the same instances in time.

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

2.1.204 quantisation matrix [video]: A set of sixty-four 8-bit values used by the dequantiser.

2.1.205 quantised DCT coefficients [video]: DCT coefficients before dequantisation. A variable length coded representation of quantised DCT coefficients is transmitted as part of the coded video bitstream.

2.1.206 quantiser scale [video]: A scale factor coded in the bitstream and used by the decoding process to scale the dequantisation.

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

2.1.208 reconstructed frame [video]: A reconstructed frame consists of three rectangular matrices of 8-bit numbers representing the luminance and two chrominance signals. A reconstructed frame is obtained by decoding a coded frame.

2.1.209 reconstructed picture [video]: A reconstructed picture is obtained by decoding a coded picture. A reconstructed picture is either a reconstructed frame (when decoding a frame picture), or one field of a reconstructed frame (when decoding a field picture). If the coded picture is a field picture, then the reconstructed picture is the top field or the bottom field of the reconstructed frame.

2.1.210 reference decoder [video]: A decoder that implements precisely the decoding process as specified in ISO/IEC 13818-2 and uses a reference IDCT. The reference decoder is capable of decoding compliant bitstreams of any defined profile-and-level.

2.1.211 reference IDCT [video]: Embodiment of the saturated mathematical integer-number IDCT specified in Annex A of ISO/IEC 13818-2.

2.1.212 reference field [video]: A reference field is one field of a reconstructed frame. Reference fields are used for forward and backward prediction when P-pictures and B-pictures are decoded. Note that when field P-pictures are decoded, prediction of the second field P-picture of a coded frame uses the first reconstructed field of the same coded frame as a reference field.

2.1.213 reference frame [video]: A reference frame is a reconstructed frame that was coded in the form of a coded I-frame or a coded P-frame. Reference frames are used for forward and backward prediction when P-pictures and B-pictures are decoded.

2.1.214 reordering delay [video]: A delay in the decoding process that is caused by frame reordering.

2.1.215 requantisation [audio]: Decoding of coded subband samples in order to recover the original quantised values.

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

2.1.217 run [video]: The number of zero coefficients preceding a non-zero coefficient, in the scan order. The absolute value of the non-zero coefficient is called “level”.

2.1.218 sample aspect ratio [video]: (abbreviated to **SAR**). This specifies the distance between samples. It is defined (for the purposes of this specification) as the vertical displacement of the lines of luminance samples in a frame divided by the horizontal displacement of the luminance samples. Thus its units are (metres per line) ÷ (metres per sample).

2.1.219 Sampling Frequency (Fs) [audio]: Defines the rate in Hertz which is used to digitise an audio signal during the sampling process.

2.1.220 saturation [video]: Limiting a value that exceeds a defined range by setting its value to the maximum or minimum of the range as appropriate.

2.1.221 scalability [video]: Scalability is the ability of a decoder to decode an ordered set of bitstreams to produce a reconstructed sequence. Moreover, useful video is output when subsets are decoded. The minimum subset that can thus be decoded is the first bitstream in the set which is called the base layer. Each of the other bitstreams in the set is called an enhancement layer. When addressing a specific enhancement layer, "lower layer" refer to the bitstream which precedes the enhancement layer.

2.1.222 scalable hierarchy [video]: coded video data consisting of an ordered set of more than one video bitstream.

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

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

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

2.1.226 scrambling [systems]: The alteration of the characteristics of a video, audio or coded data stream in order to prevent unauthorised reception of the information in a clear form. This alteration is a specified process under the control of a conditional access system.

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

2.1.228 skipped macroblock [video]: A macroblock for which no data is encoded.

2.1.229 slice [video]: A series of consecutive macroblocks that start, in the coded bitstream, by a slice_start_code and that continue up to the next start code (or to the first stuffing bytes if the next start_code is preceded by stuffing bytes)..

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

2.1.231 SNR scalability [video]: A type of scalability where the enhancement layer (s) contain only coded refinement data for the DCT coefficients of the lower layer.

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

2.1.233 source; input [video]: Term used to describe the video material or some of its attributes before encoding.

2.1.234 spatial prediction [video]: prediction derived from a decoded frame of the lower layer decoder used in spatial scalability.

2.1.235 spatial scalability [video]: A type of scalability where an enhancement layer also uses predictions from sample data derived from a lower layer without using motion vectors. The layers can have different frame sizes, frame rates or chrominance formats.

2.1.236 splicing[system]: The concatenation, performed on the system level, of two different elementary streams. The resulting system stream conforms totally to ISO/IEC 13818-1. The splice may result in discontinuities in timebase, continuity counter, PSI, and decoding.

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

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

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

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

2.1.241 stereo-irrelevant [audio]: a portion of a stereophonic audio signal which does not contribute to spatial perception.

2.1.242 still picture [systems]: A coded still picture consists of a video sequence containing exactly one coded picture which is intra-coded. This picture has an associated PTS and the presentation time of succeeding pictures, if any, is later than that of the still picture by at least two picture periods.

2.1.243 stuffing (bits); stuffing (bytes): Code-words that may be inserted at particular locations in the coded bitstream that are discarded in the decoding process. Their purpose is to increase the bitrate of the stream which would otherwise be lower than the desired bitrate.

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

2.1.245 subband filterbank [audio]: A set of band filters covering the entire audio frequency range. In this part of ISO/IEC 13818, the subband filterbank is a polyphase filterbank.

2.1.246 subband samples [audio]: The subband filterbank within the audio encoder creates a filtered and subsampled representation of the input audio samples. The filtered samples are called subband samples. From 32 time-consecutive input audio samples, one subband samples is generated within each of the 32 subbands.

2.1.247 surround channel [audio]: An audio presentation channel added to the front channels (L and R or L, R, and C) to enhance the spatial perception.

2.1.248 syncword [audio]: A 12-bit code embedded in the audio bit stream that identifies the start of a base frame or an extension frame.

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

2.1.250 System Clock Reference; SCR [system]: A time stamp in the Program Stream from which decoder timing is derived.

2.1.251 system header [system]: The system header is a data structure defined in 2.5.3.5 of ISO/IEC 13818-1, that carries information summarising the system characteristics of the ITU-T Rec. H.222.0 | ISO/IEC 13818 multiplexed Program Stream.

2.1.252 system target decoder; STD [system]: A hypothetical reference model of a decoding process used to describe the semantics of an ITU-T Rec. H.222.0 | ISO/IEC 13818 multiplexed bitstream.

2.1.253 temporal prediction [video]: prediction derived from reference frames or fields other than those defined as spatial prediction.

2.1.254 temporal scalability [video]: A type of scalability where an enhancement layer also uses predictions from sample data derived from a lower layer using motion vectors. The layers have identical frame size, and chrominance formats, but can have different frame rates.

2.1.255 time-stamp [system]: A term that indicates the time of a specific action such as the arrival of a byte or the presentation of a Presentation Unit.

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

2.1.257 top field [video]: One of two fields that comprise a frame. Each line of a top field is spatially located immediately above the corresponding line of the bottom field.

2.1.258 top layer [video]: the topmost layer (with the highest layer_id) of a scalable hierarchy.

2.1.259 Transport Stream packet header [system]: The leading fields in a Transport Stream packet, up to and including the continuity_counter field.

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

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

2.1.262 variable length coding: A reversible procedure for coding that assigns shorter code-words to frequent symbols and longer code-words to less frequent symbols.

2.1.263 Variable Length Code: A code word assigned by variable length encoder (See variable length coding)

2.1.264 Variable Length Decoder: A procedure to obtain the symbols encoded with a variable length coding technique.

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

2.1.266 video sequence [video]: The highest syntactic structure of coded video bitstreams. It contains a series of one or more coded frames.

2.1.267 xxx profile bitstream [video]: a bitstream of a scalable hierarchy with a profile indication corresponding to xxx. Note that this bitstream is only decodable together with all its lower layer bitstreams (unless it is a base layer bitstream).

2.1.268 xxx profile decoder [video]: decoder able to decode one or a scalable hierarchy of bitstreams of which the top layer conforms to the specifications of the xxx profile (with xxx being any of the defined Profile names).

2.1.269 xxx profile scalable hierarchy [video]: set of bitstreams of which the top layer conforms to the specifications of the xxx profile.

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

2.2 Symbols and abbreviations

The mathematical operators used to describe this part of ISO/IEC 13818 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 towards –∞.

| | Absolute value.

Absolute value.	| x | = x	when x > 0
	| x | = 0	when x == 0
	| x | = –x	when x < 0

% Modulus operator. Defined only for positive numbers.

Sign( ) Sign.

Sign.	Sign(x) = 1	when x > 0
	Sign(x) = 0	when x == 0
	Sign(x) = –1	when x < 0

NINT ( ) Nearest integer operator. Returns the nearest integer value to the real-valued argument. Half-integer values are rounded away from zero.

sin Sine.

cos Cosine.

exp Exponential.

√ Square root.

$\log_{10}$ Logarithm to base ten.

$\log_e$ Logarithm to base e.

$\log_2$ Logarithm to base 2.

2.2.2 Logical operators

|| Logical OR.

&& Logical AND.

! Logical NOT.

2.2.3 Relational operators

> Greater than.

>= Greater than or equal to.

< Less than.

<= Less than or equal to.

== Equal to.

!= Not equal to.

max [,...,] the maximum value in the argument list.

min [,...,] the minimum value in the argument list.

2.2.4 Bitwise operators

A two's complement number representation is assumed where the bitwise operators are used.

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

2.2.5 Assignment

=	Assignment operator.

2.2.6 Mnemonics

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

bslbf	Bit string, left bit first, where "left" is the order in which bit strings are written in ISO/IEC 13818. 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.
centre_chan	Index of centre channel.
centre_limited	Variable which indicates whether a subband of the centre is not transmitted. It is used in the case of Phantom coding of centre channel.
ch	Channel. If ch has the value 0, the left channel of a stereo signal or the first of two independent signals is indicated.
dyn_cross	dyn_cross means that dynamic crosstalk is used for a certain transmission channel and a certain subband.
gr	Granule of 3 * 32 subband samples in audio Layer II, 18 * 32 subband samples in audio Layer III.
Lo, Ro	Compatible stereo audio signals.
L, C, R, LS, RS	Left, centre, right, left surround and right surround audio signals.
L^W, C^W, R^W, LS^W, RS^W	Weighted left, centre, right, left surround and right surround audio signals. The weighting is necessary for two reasons: 1) All signals have to be attenuated prior to encoding to avoid overload when calculating the compatible stereo signal. 2) The matrix equations contain attenuation factors and other processing like phase shifting. The weighted and processed signals are actually coded, transmitted and denormalised in the decoder.
left_surr_chan	Index of left surround channel.
main_data	The main_data portion of the bit stream contains the scalefactors, Huffman encoded data and ancillary information.
mlsblimit	Maximum used subband in multilingual part of bitstream.
mono_surr_chan	Index of the mono surround channel. This index is identical to the index of the left surround channel.
msblimit	Maximum used subband in multichannel extension part of bitstream.
nch	Number of channels; equal to 1 for single_channel mode, 2 in other modes.
nmch	Number of channels in the multichannel extension part.
nmlch	Number of multilingual channels.

npred	Number of allowed predictors according to the tables in 2.5.2.15.
npredcoeff	Number of prediction coefficients used.
part2_length	The number of main_data bits used for scalefactors.
pci	index of predictor[0, 1, 2].
px	index of predictor[0, 1, ..., npred-1].
right_surr_chan	Index of right surround channel.
rpchof	Remainder polynomial coefficients, highest order first.
sb	Subband.
sbgr	Groups of individual subbands according to subbandgroup table in 2.5.2.15.
sblimit	The number of the lowest subband for which no bits are allocated.
scfsi	Scalefactor selection information.
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.
T0, T1, T2, T3, T4	Audio transmission channels. The assignment of audio signals to transmission channels is determined by the dematrixing procedure and the transmission channel allocation information.
tc	Transmitted channel.
uimsbf	Unsigned integer, most significant bit first.
vlclbf	Variable length code, left bit first, where "left" refers to the order in which the VLC codes are written.
window	Number of the actual time slot in case of block_type==2, 0 <= window <= 2 (Layer III).

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

2.2.7 Constants

π	3,14159265358...
e	2,71828182845...

2.3 Method of describing bit stream syntax

The bit stream retrieved by the decoder is described in 2.4.1 and 2.5.1. 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 the definition of the state variables used in their decoding are described in 2.4.2, 2.4.3, 2.5.2 and 2.5.3. The following constructs are used to express the conditions when data elements are present, and are in normal type:

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

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

do { **data_element** . . . } while (condition)	The data element always occurs at least once. The data element is repeated until the condition is not true.

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

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

for (expr1; expr2; expr3) {
data_element
. . .
}
	Expr1 is an expression specifying the initialisation of the loop. Normally it specifies the initial state of the counter. Expr2 is a condition specifying a test, made before each iteration of the loop. The loop terminates when the condition is not true. Expr3 is an expression that is performed at the end of each iteration of the loop, normally it increments a counter.

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

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

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

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

While the syntax is expressed in procedural terms, it should not be assumed that 2.4.3 and 2.5.3 implement a satisfactory decoding procedure. In particular, it assumes a correct and error-free input bit stream. Actual decoders must include a means to look for start codes in order to begin decoding correctly.

Definition of bytealigned function

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

Definition of nextbits function

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

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

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

2.4 Requirements for Extension of ISO/IEC 11172-3 to Lower Sampling Frequencies

2.4.1 Specification of the Coded Audio Bit Stream Syntax

2.4.1.1 Audio Sequence

See ISO/IEC 11172-3, 2.4.1.1.

2.4.1.2 Audio Frame

See ISO/IEC 11172-3, 2.4.1.2.

2.4.1.3 Header

See ISO/IEC 11172-3, 2.4.1.3.

2.4.1.4 Error Check

See ISO/IEC 11172-3, 2.4.1.4.

2.4.1.5 Audio Data Layer I

See ISO/IEC 11172-3, 2.4.1.5.

2.4.1.6 Audio Data Layer II

See ISO/IEC 11172-3, 2.4.1.6.

2.4.1.7 Audio Data Layer III

Syntax	No. of bits	Mnemonic
audio_data()		
{		
main_data_begin	8	uimsbf
if (mode==single_channel)		
private_bits	1	bslbf
else		
private_bits	2	bslbf
for (ch=0; ch<nch; ch++) {		
part2_3_length[ch]	12	uimsbf

big_values[ch]	9	uimsbf
global_gain[ch]	8	uimsbf
scalefac_compress[ch]	9	bslbf
window_switching_flag[ch]	1	bslbf
if (window_switching_flag[ch]=='1') {		
block_type[ch]	2	bslbf
mixed_block_flag[ch]	1	uimsbf
for (region=0; region<2; region++)		
table_select[ch][region]	5	bslbf
for (window=0; window<3; window++)		
subblock_gain[ch][window]	3	uimsbf
}		
else {		
for (region=0; region<3; region++)		
table_select[ch][region]	5	bslbf
region0_count[ch]	4	bslbf
region1_count[ch]	3	bslbf
}		
scalefac_scale[ch]	1	bslbf
count1table_select[ch]	1	bslbf
}		
main_data ()		
}		

The main data bit stream is defined below. The main_data field in the audio_data() syntax contains bytes from the main data bit stream. However, because of the variable nature of Huffman coding used in Layer III, the main data for a frame does not generally follow the header and side information for that frame. The main_data for a frame starts at a location in the bit stream preceding the header of the frame at a negative offset given by the value of main_data_begin. (See definition of main_data_begin in ISO/IEC 11172-3).

Syntax	No. of bits	Mnemonic
main_data()		
{		
for (ch=0; ch<nch; ch++) {		
if ((window_switching_flag[ch]=='1') &&		
(block_type[ch]=='10')) {		
if (mixed_block_flag[ch] == '1') {		
for (sfb=0; sfb<6; sfb++)		
scalefac_l[ch][sfb]	0..4	uimsbf
for (sfb=3; sfb<12; sfb++)		
for (window=0; window<3; window++)		
scalefac_s[ch][sfb][window]	0..5	uimsbf
}		
else {		
for (sfb=0; sfb<12; sfb++)		
for (window=0; window<3; window++)		
scalefac_s[ch][sfb][window]	0..5	uimsbf
}		
}		
else {		
for (sfb=0; sfb<21; sfb++)		
scalefac_l[ch][sfb]	0..5	uimsbf
}		
Huffmancodebits()		
}		
for (b=0; b<no_of_ancillary_bits; b++)		
ancillary_bit	1	bslbf
}		

Huffmancodebits - see ISO/IEC 11172-3, 2.4.1.7.

2.4.1.8 Ancillary Data

See ISO/IEC 11172-3, 2.4.1.8.

2.4.2 Semantics for the Audio Bit Stream Syntax

2.4.2.1 Audio Sequence General

See ISO/IEC 11172-3, 2.4.2.1.

A Layer III Lower Sampling Rate frame, however, only contains information for 576 samples as opposed to the 1152 samples of an ISO/IEC 11172-3 Layer III frame.

2.4.2.2 Audio Frame

See ISO/IEC 11172-3, 2.4.2.2.

2.4.2.3 Header

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

syncword - See ISO/IEC 11172-3, 2.4.2.3.

ID - One bit to indicate the ID of the algorithm. Equals '1' for ISO/IEC 11172-3, '0' for extension to lower sampling frequencies.

Layer - See ISO/IEC 11172-3, 2.4.2.3.

protection_bit - See ISO/IEC 11172-3, 2.4.2.3.

bitrate_index - Four bits to indicate 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 an audio frame contains either N or N+1 slots, depending on the value of the padding bit. The bitrate_index is an index to a table, which is the same for Layer II and III, but different for Layer I.

The bitrate_index indicates the total bitrate irrespective of the mode (stereo, joint_stereo, dual_channel, single_channel), according to the following table which is valid for ID=='0'.

bitrate_index	bitrate specified (kbit/s) for Fs = 16, 22,05, 24 kHz	
	Layer I	Layer II, Layer III
'0000'	free	free
'0001'	32	8
'0010'	48	16
'0011'	56	24
'0100'	64	32
'0101'	80	40
'0110'	96	48
'0111'	112	56
'1000'	128	64
'1001'	144	80
'1010'	160	96
'1011'	176	112
'1100'	192	128
'1101'	224	144
'1110'	256	160
'1111'	forbidden	forbidden

The decoder is not required to support bitrates higher than 256 kbit/s, 160 kbit/s, 160 kbit/s in respect to Layer I, II and III when in free format mode.

sampling_frequency - Indicates the sampling frequency for ID=='0', according to the following table.

sampling_frequency	frequency specified (kHz)
'00'	22,05
'01'	24
'10'	16
'11'	reserved

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

padding_bit - See ISO/IEC 11172-3, 2.4.2.3. Padding is necessary with a sampling frequency of 22,05 kHz. Padding may also be required in free format.

private_bit - See ISO/IEC 11172-3, 2.4.2.3.

mode - See ISO/IEC 11172-3, 2.4.2.3.

mode_extension - See ISO/IEC 11172-3, 2.4.2.3.

copyright - See ISO/IEC 11172-3, 2.4.2.3.

original/copy - See ISO/IEC 11172-3, 2.4.2.3.

emphasis - See ISO/IEC 11172-3, 2.4.2.3.

2.4.2.4 Error Check

For Layer I and Layer II, see ISO/IEC 11172-3, 2.4.2.4.

For Layer III, the bits used to calculate the error check are:

bits 16..31 of the header
bits 0..71 of audio_data for single channel mode,
bits 0..135 of audio_data for other modes.

2.4.2.5 Audio Data Layer I

See ISO/IEC 11172-3, 2.4.2.5.

2.4.2.6 Audio Data Layer II

See ISO/IEC 11172-3, 2.4.2.6.

2.4.2.7 Audio Data Layer III

See ISO/IEC 11172-3, 2.4.2.7 with the exception of a different definition of scalefac_compress.

scalefac_compress[ch] - Selects the number of bits used for the transmission of the scalefactors and sets or resets preflag. If preflag is set, the values of a table are added to the scalefactors as described in ISO/IEC 11172-3 (Table B.6 Annex B).

2.4.2.8 Ancillary Data

See ISO/IEC 11172-3, 2.4.2.8.

2.4.3 The Audio Decoding Process

2.4.3.1 Audio Decoding Layer I, II

See ISO/IEC 11172-3, 2.4.3. For Layer II, instead of tables B.2 (Layer II bit allocation tables) in ISO/IEC 11172-3, table B.1 (Possible quantisation per subband, Layer II) of this part of ISO/IEC 13818 should be used.

2.4.3.2 Audio Decoding Layer III

Decoding of Layer III low sampling frequencies is performed as for Layer III in ISO/IEC 11172-3 with the following differences:

- For low sampling frequencies a Layer III frame contains only one granule in opposite to ISO/IEC 11172, where a Layer III frame consists of two frames. The variable 'gr' does not exist any more. The number of samples per frame is 576. The constant which is used for the calculation of the frame length (ISO/IEC 11172, 2.4.3.1) and the padding algorithm (ISO/IEC 11172, 2.4.2.3) must therefore be changed for Layer III according to the following table:

Changed constants for ISO/IEC 13818-3 Layer III		
	ISO/IEC 11172-3	ISO/IEC 13818-3
slots_per_frame	144	72
frame_size	1152	576

- If intensity stereo is selected, the maximum value for intensity position will indicate an illegal intensity position. As in ISO/IEC 11172-3 scalefactor bands with an illegal intensity position have to be decoded according to the MS equations as defined in ISO/IEC 11172-3 if MS stereo is enabled, or both channels independently if MS stereo is not enabled.

- As in ISO/IEC 11172-3 the last scalefactor band which is not intensity coded is equal to the last scalefactor band in the right channel which is not completely zero, and in which the corresponding scalefactor does not indicate illegal intensity position. Like ISO/IEC 11172-3 decoding of the lower bound for intensity stereo is performed individually for each window in the case of short blocks (block_type == '10'). This means that, like in ISO/IEC 11172-3, 2.4.3.4, the calculation of the intensity bound is applied to the values of each short window and permits individual intensity stereo decoding per short window.

- Steps 4 and 5 of the described decoding process for intensity stereo decoding are modified as follows:

  4) $R_i := L_i * k_r$
  5) $L_i := L_i * k_l$

  The values k_l and k_r are calculated from the transmitted scalefactor / is_pos_{sb} value as follows:

$$
\begin{array}{lll}
\text{if } (is\_pos_{sb} == 0) & k_l = 1.0 & k_r = 1.0 \\
\text{else if } (is\_pos_{sb} \% 2 == 1) & k_l = i_0^{(is\_pos_{sb}+1)/2} & k_r = 1.0 \\
\text{else} & k_l = 1.0 & k_r = i_0^{is\_pos_{sb}/2}
\end{array}
$$

  The basic intensity stereo decoding factor i_0 is determined by intensity_scale (1/√2 for intensity_scale==1, else 1/√√2). The value of intensity_scale is derived from the value of **scalefac _compress** of the right channel according to:
  intensity_scale = **scalefac_compress** % 2

- The paragraph "Scalefactors" in ISO/IEC 11172-3, 2.4.3.4 has to be replaced by the following text:

Scalefactors

The scalefactors are decoded according to the slen1, slen2, slen3, and slen4 and nr_of_sfb1, nr_of_sfb2, nr_of_sfb3, and nr_of_sfb4 which are determined from the values of scalefac_compress.

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

part2_length = nr_of_sfb1*slen1 + nr_of_sfb2*slen2 + nr_of_sfb3*slen3 + nr_of_sfb4*slen4

The scalefactors are transmitted in four partitions. The number of scalefactors in each partition (nr_of_sfb1, nr_of_sfb2, nr_of_sfb3, and nr_of_sfb4), the length of the scalefactors in each partition (slen1, slen2, slen3, and slen4), and preflag are decoded from **scalefac_compress** according to the following procedure:

```
if (!(((mode_extension == '01') || (mode_extension == '11') ) && (ch==1) ) ){
    if ( scalefac_compress < 400 ) {
        slen1 = (scalefac_compress >> 4) / 5
        slen2 = (scalefac_compress >> 4) % 5
        slen3 = (scalefac_compress % 16) >>2
        slen4 = scalefac_compress % 4
        preflag = 0
        block_type      mixed_block_flag   nr_of_sfb1   nr_of_sfb2   nr_of_sfb3   nr_of_sfb4
        '00','01','11'  x                  6            5            5            5
        '10'            0                  9            9            9            9
        '10'            1                  6            9            9            9
    }
    if ( (400 <= scalefac_compress) && (scalefac_compress < 500) ) {
        slen1 = ((scalefac_compress-400) >> 2) / 5
        slen2 = ((scalefac_compress-400) >> 2) % 5
        slen3 = (scalefac_compress-400) % 4
        slen4 = 0
        preflag = 0
        block_type      mixed_block_flag   nr_of_sfb1   nr_of_sfb2   nr_of_sfb3   nr_of_sfb4
        '00','01','11'  x                  6            5            7            3
        '10'            0                  9            9            12           6
        '10'            1                  6            9            12           6
    }
    if ( (500 <= scalefac_compress) && (scalefac_compress < 512) )  {
        slen1 = (scalefac_compress-500) / 3
        slen2 = (scalefac_compress-500) % 3
        slen3 = 0
        slen4 = 0
        preflag = 1
        block_type      mixed_block_flag   nr_of_sfb1   nr_of_sfb2   nr_of_sfb3   nr_of_sfb4
        '00','01','11'  x                  11           10           0            0
        '10'            0                  18           18           0            0
        '10'            1                  15           18           0            0
    }
}
if ( ( (mode_extension == '01') || (mode_extension == '11') ) && (ch == 1) ) {
    intensity_scale = scalefac_compress % 2
    int_scalefac_compress = scalefac_compress >> 1
    if (int_scalefac_compress < 180 ) {
        slen1 = int_scalefac_compress / 36
        slen2 = (int_scalefac_compress % 36) / 6
        slen3 = (int_scalefac_compress % 36) % 6
        slen4 = 0
```

```
        preflag = 0
        block_type      mixed_block_flag   nr_of_sfb1   nr_of_sfb2   nr_of_sfb3   nr_of_sfb4
        '00','01','11'  x                  7            7            7            0
        '10'            0                  12           12           12           0
        '10'            1                  6            15           12           0
    }
    if ( (180 <= int_scalefac_compress) && (int_scalefac_compress < 244) ) {
        slen1 = ((int_scalefac_compress-180) % 64 ) >> 4
        slen2 = ((int_scalefac_compress-180) % 16 ) >> 2
        slen3 = (int_scalefac_compress-180) % 4
        slen4 = 0
        preflag = 0
        block_type      mixed_block_flag   nr_of_sfb1   nr_of_sfb2   nr_of_sfb3   nr_of_sfb4
        '00','01','11'  x                  6            6            6            3
        '10'            0                  12           9            9            6
        '10'            1                  6            12           9            6
    }
    if ( (244 <= int_scalefac_compress) && (int_scalefac_compress <= 255) ) {
        slen1 = (int_scalefac_compress-244) / 3
        slen2 = (int_scalefac_compress-244) % 3
        slen3 = 0
        slen4 = 0
        preflag = 0
        block_type      mixed_block_flag   nr_of_sfb1   nr_of_sfb2   nr_of_sfb3   nr_of_sfb4
        '00','01','11'  x                  8            8            5            0
        '10'            0                  15           12           9            0
        '10'            1                  6            18           9            0
    }
}
```

In scalefactor bands where slen1, slen2, slen3 or slen4 is zero and the corresponding nr_of_slen1, nr_of_slen2, nr_of_slen3 or nr_of_slen4 is not zero, the scalefactors of these bands must be set to zero, resulting in an intensity position of zero.

2.5 Requirements for Extension of ISO/IEC 11172-3 to Multichannel Audio

2.5.1 Specification of the Coded Audio Bit Stream Syntax

2.5.1.1 Audio Sequence

An audio sequence consists of a base bit stream (also decodable by an ISO/IEC 11172-3 decoder), and optionally an extension bit stream.

2.5.1.1.1 Base Bit Stream

Syntax	No. of bits	Mnemonic
base_bit_stream()		
{		
while (nextbits()==syncword) {		
base_frame()		
}		
}		

2.5.1.1.2 Extension Bit Stream

Syntax	No. of bits	Mnemonic
ext_bit_stream()		
{		
if (ext_bit_stream_present=='1') {		
ext_frame()		
}		
}		

2.5.1.2 Base Frame Layer I

Syntax	No. of bits	Mnemonic
base_frame()		
{		
mpeg1_header()		
mpeg1_error_check()		
mpeg1_audio_data()		
mc_extension_data_part1()		
continuation_bit	1	bslbf
mpeg1_header()		
mpeg1_error_check()		
mpeg1_audio_data()		
mc_extension_data_part2()		
continuation_bit	1	bslbf
mpeg1_header()		
mpeg1_error_check()		
mpeg1_audio_data()		
mc_extension_data_part3()		
mpeg1_ancillary_data()		
}		

2.5.1.3 Base Frame Layer II

Syntax	No. of bits	Mnemonic
base_frame()		
{		
mpeg1_header()		
mpeg1_error_check()		
mpeg1_audio_data()		
mc_extension_data_part1()		
mpeg1_ancillary_data()		
}		

2.5.1.4 Base Frame Layer III

```
Syntax                                                  No. of bits   Mnemonic
base_frame()
{
        mpeg1_header()
        mpeg1_error_check()
        mpeg1_audio_side_info()
        mpeg1_main_data()
}
```

```
Syntax                                                  No. of bits   Mnemonic
mpeg1_main_data()
{
        mpeg1_audio_main_data()
        mc_extension_data_part1()
        mpeg1_ancillary_data()
}
```

2.5.1.5 Extension Frame

```
Syntax                                                  No. of bits   Mnemonic
ext_frame()
{
   ext_header()
   ext_data()
   if( layer! = 3)
      ext_ancillary_data()
}
```

2.5.1.6 MPEG1 Header

See ISO/IEC 11172-3, 2.4.1.3.

2.5.1.7 MPEG1 Error Check

See ISO/IEC 11172-3, 2.4.1.4.

2.5.1.8 MPEG1 Audio Data

See ISO/IEC 11172-3, 2.4.1.5, 2.4.1.6 and 2.4.1.7.

2.5.1.9 MPEG1 Ancillary Data

If ext_bit_stream_present=='1' || layer == 3 then the following syntax is valid.

```
Syntax                                                  No. of bits   Mnemonic
mpeg1_ancillary_data()
{
        if (ext_bit_stream_present == '1' || layer == 3) {
        for (b=0; b<8*n_ad_bytes; b++)
                ancillary_bit                           1             bslbf
        }
}
```

If ext_bit_stream_present=='0' && layer != 3, see ISO/IEC 11172-3, 2.4.1.8.

2.5.1.10 Ext Header

Syntax	No. of bits	Mnemonic
ext_header()		
{		
ext_syncword	12	bslbf
ext_crc_check	16	bslbf
ext_length	11	uimsbf
ext_ID_bit	1	bslbf
}		

2.5.1.11 Ext Ancillary Data

Syntax	No. of bits	Mnemonic
ext_ancillary_data()		
{		
for (b=0; b<no_of_ext_ancillary_bits; b++)		
ext_ancillary_bit	1	bslbf
}		

2.5.1.12 MC Extension

2.5.1.12.1 MC Extension Layer I and II

Syntax	No. of bits	Mnemonic
mc_extension()		
{		
mc_header()		
mc_error_check()		
mc_composite_status_info()		
mc_audio_data()		
ml_audio_data()		
}		

2.5.1.12.2 MC Extension Layer III

Syntax	No. of bits	Mnemonic
mc_extension()		
{		
mc_header()		
mc_error_check()		
mc_composite_status_info()		
mpeg2_audio_side_info()		
while(!bytealigned())		
byte_align_bit	1	bslbf
mpeg2_audio_main_data()		
}		

Syntax	No. of bits	Mnemonic
mpeg2_audio_side_info()		
{		
mc_side_info()		

```
  if( lfe == '1' )
     lfe_side_info()
  if( no_of_multi_lingual_ch != 0 )
     ml_side_info()
}
```

Syntax	No. of bits	Mnemonic
mpeg2_audio_main_data()		
{		
mc_audio_main_data()		
if(lfe == '1')		
lfe_audio_main_data()		
if(no_of_multi_lingual_ch != 0)		
ml_audio_main_data()		
mpeg2_ancillary_data()		
}		

Syntax	No. of bits	Mnemonic
mpeg2_ancillary_data()		
{		
for(b=0; b< l3_mpeg2_ancillary_bits; b++) {		
ancillary_bit	1	bslbf
}		
}		

2.5.1.12.3 MC Extension Data Location

In Layer I, the contents of mc_extension() are subdivided between mc_extension_data_part1(), mc_extension_data_part2() and mc_extension_data_part3(), optionally followed by ext_data(), which is transmitted in the corresponding extension frame.

In Layers II and III, the contents of mc_extension() are subdivided between mc_extension_data_part1(), optionally followed by ext_data(), which is transmitted in the corresponding extension frame. This can be represented as follows:

Syntax	No. of bits	Mnemonic
mc_extension_data()		
{		
if (layer == 1) {		
mc_extension_data_part1()		
mc_extension_data_part2()		
mc_extension_data_part3()		
}		
else		
mc_extension_data_part1()		
if (ext_bit_stream_present == '1')		
ext_data()		
}		

2.5.1.13 MC Header

Syntax	No. of bits	Mnemonic
mc_header()		
{		
ext_bit_stream_present	1	bslbf
if (ext_bit_stream_present == '1' || layer == 3)		
n_ad_bytes	8	uimsbf
centre	2	bslbf

surround	2	bslbf
lfe	1	bslbf
audio_mix	1	bslbf
dematrix_procedure	2	bslbf
no_of_multi_lingual_ch	3	uimsbf
multi_lingual_fs	1	bslbf
multi_lingual_layer	1	bslbf
copyright_identification_bit	1	bslbf
copyright_identification_start	1	bslbf
}		

2.5.1.14 MC Error Check

Syntax	No. of bits	Mnemonic
mc_error_check()		
{		
mc_crc_check	16	rpchof
}		

2.5.1.15 MC Composite Status Information, Layer I, II

Syntax	No. of bits	Mnemonic
mc_composite_status_info()		
{		
tc_sbgr_select	1	bslbf
dyn_cross_on	1	bslbf
mc_prediction_on	1	bslbf
if (tc_sbgr_select == '1') {		
tc_allocation	0..3	uimsbf
for (sbgr=0; sbgr<12; sbgr++)		
tc_allocation[sbgr] = tc_allocation		
}		
else for (sbgr=0; sbgr<12; sbgr++)		
tc_allocation[sbgr]	0..3	uimsbf
if (dyn_cross_on == '1') {		
dyn_cross_LR	1	bslbf
for (sbgr=0; sbgr<12; sbgr++) {		
dyn_cross_mode[sbgr]	0..4	bslbf
if (surround=='11')		
dyn_second_stereo[sbgr]	1	bslbf
}		
}		
if (mc_prediction_on == '1') {		
for (sbgr=0; sbgr<8; sbgr++) {		
mc_prediction[sbgr]	1	bslbf
if (mc_prediction[sbgr] == '1')		
for (px=0; px<npred; px++)		
predsi[sbgr][px]	2	bslbf
}		
}		
}		

2.5.1.16 MC Composite Status Information, Layer III

Syntax	No. of bits	Mnemonic
mc_composite_status_info()		
{		
mc_data_begin	11	uimsbf

```
for(gr=0; gr<2; gr++)
    for ( ch=2; ch<4; ch++) {
        seg_list_present[gr][ch]                              1       bslbf
        tc_present[gr][ch]                                    1       bslbf
        block_type[gr][ch]                                    2       bslbf
    }
if (centre!= '00') {
    for(gr=0; gr<2; gr++) {
        seg_list_present[gr][centre_chan]                     1       bslbf
        tc_present[gr][centre_chan]                           1       bslbf
        block_type[gr][centre_chan]                           2       bslbf
    }
}
if (surround=='01') {
    for(gr=0; gr<2; gr++) {
        seg_list_present[gr][mono_surr_chan]                  1       bslbf
        tc_present[gr][mono_surr_chan]                        1       bslbf
        block_type[gr][mono_surr_chan]                        2       bslbf
    }
}
if (surround=='10'  ||  surround=='11') {
    for(gr=0; gr<2; gr++)
        for ( ch=left_surr_chan; ch<=right_surr_chan; ch++) {
            seg_list_present[gr][ch]                          1       bslbf
            tc_present[gr][ch]                                1       bslbf
            block_type[gr][ch]                                2       bslbf
        }
}
if (dematrix_procedure != '11')
    dematrix_length                                           4       bslbf
else
    dematrix_length = '0000'
for ( sbgr=0; sbgr<dematrix_length; sbgr++)
    dematrix_select[sbgr]                                     3..4    bslbf

for (gr=0; gr<2; gr++)
    for( ch=2; ch<7; ch++ ) {
        if (ch_present(ch)=='1' && seg_list_present[gr][ch] == '1') {
            seg_list_nodef[gr][ch]                            1       bslbf
            if ( seg_list_nodef[gr][ch] == '1') {
                if ( gr==1  &&  seg_list_present[gr_0][ch] == '1'
                            && seg_list_nodef[gr_0][ch] == '1' ) {
                    segment_list_repeat[ch]                   1       bslbf
                    if ( segment_list_repeat[ch] == '0') {
                        segment_list( gr,ch )
                    }
                }
                else
                    segment_list( gr,ch)
            }
        }
    }

mc_prediction_on                                              1       bslbf
if (mc_prediction_on == '1') {
    for (sbgr=0; sbgr< 15; sbgr++)
        mc_prediction[sbgr]                                   1       bslbf

    for (sbgr=0; sbgr< 15; sbgr++) {
        if (mc_prediction[sbgr] == '1') {
            for (pci=0; pci<npredcoef; pci++)
```

predsi[sbgr][pci]	1	bslbf
}		
}		
for (sbgr=0; sbgr< 15; sbgr++) {		
for (pci=0; pci<npredcoef; pci++) {		
if (predsi[sbgr][pci] == '1')		
pred_coef[sbgr][pci]	3	uimsbf
}		
}		
}		
}		

Syntax	No. of bits	Mnemonic
segment_list(gr,ch)		
{		
seg = 0		
sbgr = dematrix_length		
if (block_type[gr][ch] =='10')		
sbgr_cnt = 12		
else sbgr_cnt = 15		
attenuation_range[gr][ch]	2	uimsbf
attenuation_scale[gr][ch]	1	uimsbf
while (sbgr < sbgr_cnt) {		
seg_length[gr][ch][seg]	4	uimsbf
if (seg_length[gr][ch][seg] == 0)		
break;		
tc_select[gr][ch][seg]	3	uimsbf
if (tc_select[gr][ch][seg] != 7 && tc_select[gr][ch][seg] != ch)		
for (sbgr1=sbgr; sbgr1<sbgr+seg_length[gr][ch][seg]; sbgr1++)		
attenuation[gr][ch][seg][sbgr1]	2...5	uimsbf
sbgr += seg_length[gr][ch][seg]		
seg++		
}		
}		

2.5.1.17 MC Audio Data, Layer I, II

Syntax	No. of bits	Mnemonic
mc_audio_data()		
{		
if (lfe == '1')		
lfe_allocation	4	uimsbf
else lfe_allocation=0		
for (sb=0; sb<msblimit; sb++)		
for (mch=0; mch<nmch; mch++)		
if (!centre_limited[mch][sb] && !dyn_cross[mch][sb])		
allocation[mch][sb]	2..4	uimsbf
else if (centre_limited[mch][sb])		
allocation[mch][sb]=0		
for (sb=0; sb<msblimit; sb++)		
for (mch=0; mch<nmch; mch++)		
if (allocation[mch][sb]!=0)		
scfsi[mch][sb]	2	bslbf
if (mc_prediction_on == '1')		
for (sbgr=0; sbgr<8; sbgr++)		
if (mc_prediction [sbgr] == '1')		
for (px=0; px<npred; px++)		
if (predsi[sbgr][px] !='00') {		
delay_comp[sbgr][px]	3	uimsbf

Syntax	No. of bits	Mnemonic
for (pci=0; pci<predsi[sbgr][px]; pci++)		
pred_coef[sbgr][px][pci]	8	uimsbf
}		
if (lfe_allocation!=0)		
lf_scalefactor	6	uimsbf
for (sb=0; sb<msblimit; sb++)		
for (mch=0; mch<nmch; mch++)		
if (allocation[mch][sb]!=0) {		
if (scfsi[mch][sb]=='00') {		
scalefactor[mch][sb][0]	6	uimsbf
scalefactor[mch][sb][1]	6	uimsbf
scalefactor[mch][sb][2]	6	uimsbf
}		
if (scfsi[mch][sb]=='01' || scfsi[mch][sb]=='11') {		
scalefactor[mch][sb][0]	6	uimsbf
scalefactor[mch][sb][2]	6	uimsbf
}		
if (scfsi[mch][sb]=='10')		
scalefactor[mch][sb][0]	6	uimsbf
}		
for (gr=0; gr<12; gr++)		
{		
if (lfe_allocation!=0)		
lf_sample[gr]	2..16	uimsbf
for (sb=0; sb<msblimit; sb++)		
for (mch=0; mch<nmch; mch++)		
if (allocation[mch][sb]!=0 && !dyn_cross[mch][sb])		
{		
if (grouping[mch][sb])		
samplecode[mch][sb][gr]	5..10	uimsbf
else for (s=0; s<3; s++)		
sample[mch][sb][3*gr+s]	2..16	uimsbf
}		
}		
}		

2.5.1.18 ML Audio Data, Layer I, II

Syntax	No. of bits	Mnemonic
ml_audio_data()		
{		
for (sb=0; sb<mlsblimit; sb++)		
for (mlch=0; mlch<nmlch; mlch++)		
allocation[mlch][sb]	2..4	uimsbf
for (sb=0; sb<mlsblimit; sb++)		
for (mlch=0; mlch<nmlch; mlch++)		
if (allocation[mlch][sb]!=0)		
scfsi[mlch][sb]	2	bslbf
for (sb=0; sb<mlsblimit; sb++)		
for (mlch=0; mlch<nmlch; mlch++)		
if (allocation[mlch][sb]!=0) {		
if (scfsi[mlch][sb]=='00') {		
scalefactor[mlch][sb][0]	6	uimsbf
scalefactor[mlch][sb][1]	6	uimsbf
scalefactor[mlch][sb][2]	6	uimsbf
}		
if (scfsi[mlch][sb]=='01' || scfsi[mlch][sb]=='11') {		
scalefactor[mlch][sb][0]	6	uimsbf
scalefactor[mlch][sb][2]	6	uimsbf
}		

if (scfsi[mlch][sb]=='10')		
scalefactor[mlch][sb][0]	6	uimsbf
}		
for (gr=0; gr<ngr; gr++)		
for (sb=0; sb<mlsblimit; sb++)		
for (mlch=0; mlch<nmlch; mlch++)		
if (allocation[mlch][sb]!=0) {		
if (grouping[mlch][sb])		
samplecode[mlch][sb][gr]	5..10	uimsbf
else for (s=0; s<3; s++)		
sample[mlch][sb][3*gr+s]	2..16	uimsbf
}		
}		

2.5.1.19 MC Audio Data, Layer III

Syntax	No. of bits	Mnemonic
mc_side_info()		
{		
if (dematrix_procedure != '11')		
matrix_attenuation_present	1	bslbf
else		
matrix_attenuation_present = '0'		
if (matrix_attenuation_present == '1') {		
for (gr=0; gr<2; gr++)		
for (ch=2; ch<7; ch++)		
if (block_type[gr][ch] == '10')		
for (sbgr=dematrix_length; sbgr<12; sbgr++)		
if (js_carrier[gr][ch][sbgr]) {		
matrix_attenuation_l[gr][ch][sbgr]	3	bslbf
matrix_attenuation_r[gr][ch][sbgr]	3	bslbf
}		
else		
for (sbgr=dematrix_length; sbgr<15; sbgr++)		
if (js_carrier[gr][ch][sbgr]) {		
matrix_attenuation_l[gr][ch][sbgr]	3	bslbf
matrix_attenuation_r[gr][ch][sbgr]	3	bslbf
}		
}		
for (tc=2; tc < 7; tc++)		
for (scfsi_band=0; scfsi_band<4; scfsi_band++)		
if(tc_present[gr_0][tc]=='1' && tc_present[gr_1][tc] == '1')		
scfsi[tc][scfsi_band]	1	bslbf
else		
scfsi[tc][scfsi_band] = '0'		
for (gr=0; gr<2; gr++) {		
for (tc=2; tc<7; tc++) {		
if (tc_present[gr][tc] == '1') {		
part2_3_length[gr][tc]	12	uimsbf
big_values[gr][tc]	9	uimsbf
global_gain[gr][tc]	8	uimsbf
scalefac_compress[gr][tc]	4	bslbf
if (block_type[gr][tc] != '00') {		
for (region=0; region<2; region++)		
table_select[gr][tc][region]	5	bslbf
if (block_type[gr][tc] == '10') {		
for (window=0; window<3; window++)		
subblock_gain[gr][tc][window]	3	uimsbf
}		

}		
else {		
for (region=0; region<3; region++)		
table_select[gr][tc][region]	5	bslbf
region0_count[gr][tc]	4	bslbf
region1_count[gr][tc]	3	bslbf
}		
preflag[gr][tc]	1	bslbf
scalefac_scale[gr][tc]	1	bslbf
count1table_select[gr][tc]	1	bslbf
}		
}		
}		
}		

Syntax	No. of bits	Mnemonic
mc_audio_main_data()		
{		
for (gr=0; gr<2; gr++) {		
for (tc=2; tc<7; tc++) {		
if (tc_present[gr][tc] == '1') {		
if (block_type[gr][tc]=='10') {		
for (sfb=0; sfb<12; sfb++)		
for (window=0; window<3; window++)		
if (data_present[gr][tc][sfb][window]) {		
scalefac_s[gr][tc][sfb][window]	0..4	uimsbf
}		
}		
else {		
if ((scfsi[tc][0]=='0') || (gr == 0))		
for(sfb=0;sfb<6; sfb++)		
if (data_present[gr][tc][sfb]) {		
scalefac_l[gr][tc][sfb]	0..4	uimsbf
}		
if ((scfsi[tc][1]=='0') || (gr == 0))		
for(sfb=6; sfb<11; sfb++)		
if (data_present[gr][tc][sfb]) {		
scalefac_l[gr][tc][sfb]	0..4	uimsbf
}		
if ((scfsi[tc][2]=='0') || (gr == 0))		
for(sfb=11; sfb<16; sfb++)		
if (data_present[gr][tc][sfb]) {		
scalefac_l[gr][tc][sfb]	0..4	uimsbf
}		
if ((scfsi[tc][3]=='0') || (gr == 0))		
for(sfb=16; sfb<21; sfb++)		
if (data_present[gr][tc][sfb]) {		
scalefac_l[gr][tc][sfb]	0..4	uimsbf
}		
}		
Huffmancodebits()		
}		
}		
}		
}		

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

2.5.1.20 LFE Side Info, Layer III

<table>
<tr><th>Syntax</th><th>No. of bits</th><th>Mnemonic</th></tr>
<tr><td>lfe_side_info ()</td><td></td><td></td></tr>
<tr><td>{</td><td></td><td></td></tr>
<tr><td> lfe_hc_len</td><td>8</td><td>uimsbf</td></tr>
<tr><td> lfe_gain</td><td>8</td><td>uimsbf</td></tr>
<tr><td> lfe_table_select</td><td>5</td><td>uimsbf</td></tr>
<tr><td>}</td><td></td><td></td></tr>
</table>

2.5.1.21 LFE Audio Main Data, Layer III

<table>
<tr><th>Syntax</th><th>No. of bits</th><th>Mnemonic</th></tr>
<tr><td>lfe_audio_main_data ()</td><td></td><td></td></tr>
<tr><td>{</td><td></td><td></td></tr>
<tr><td> for (l=0; l<lfe_bigval; l++) {</td><td></td><td></td></tr>
<tr><td>   hcod[|x|][|y|]</td><td>0...19</td><td>bslbf</td></tr>
<tr><td>   if (|x|==15 && linbits>0)</td><td></td><td></td></tr>
<tr><td>   linbitsx</td><td>1...13</td><td>uimsbf</td></tr>
<tr><td>   if (x != 0)</td><td></td><td></td></tr>
<tr><td>   signx</td><td>1</td><td>bslbf</td></tr>
<tr><td>   if (|y|==15 && linbits>0)</td><td></td><td></td></tr>
<tr><td>   linbitsy</td><td>1...13</td><td>uimsbf</td></tr>
<tr><td>   if (y != 0)</td><td></td><td></td></tr>
</table>

signy	1	bslbf
is_lfe[gr_0][l] = x		
is_lfe[gr_1][l] = y		
}		
while (l<6) {		
is_lfe[gr_0][l] = 0;		
is_lfe[gr_1][l] = 0;		
l++;		
}		
}		

2.5.1.22 ML Side Info, Layer III

If multi_lingual_fs==0, see audio_data() syntax in ISO/IEC 11172-3, 2.4.2.7, but without main_data_begin, private_bits and main_data().

If multi_lingual_fs==1, see audio_data() syntax in ISO/IEC 11172-3, 2.4.1.2 of this part of ISO/IEC 13818, but without main_data_begin, private_bits and main_data().

For use as ML Side Info, nch is set to no_of_multi_lingual_ch.

2.5.1.23 ML Audio Main Data, Layer III

If multi_lingual_fs==0, see main_data syntax in ISO/IEC 11172-3, 2.4.1.7.

If multi_lingual_fs==1, see main_data syntax in 2.4.1.7 of this part of ISO/IEC 13818.

For use as ML audio main data, nch is set to no_of_multi_lingual_ch.

2.5.2 Semantics for the Audio Bit Stream Syntax

2.5.2.1 Audio Sequence General

base_frame plus optional **ext_frame** - Part of the bit stream that is decodable by itself. It contains information for 1152 audio samples for each coded audio channel, 12 samples for the LFE channel, and either 1152 or 576 samples for each multilingual channel. It starts with a syncword, and ends just before the third following syncword in Layer I and just before the next syncword in Layer II or III. It consists of an integer number of slots (four bytes in Layer I, one byte in Layer II or III).

The base frame shall contain either the backwards compatible stereo or the left and right channels, depending on matrixing information. It starts with the mpeg1_header, mpeg1_error_check, followed by the mpeg1_audio_data, the mc_extension_data_part1 and the mpeg1_ancillary_data for Layer I and II. For Layer I, the mc_extension_data_part is split into three parts, mc_extension_data_part1, mc_extension_data_part2 and mc_extension_data_part3. For Layer III, it also starts with the mpeg1_header and mpeg1_error_check, but then followed by the mpeg1_audio_side_info and mpeg1_main_data. The mpeg1_main_data consist of the mpeg1_audio_main_data, the mc_extension_data_part1 and the mpeg1_ancillary_data.

In case, the overall bitrate exceeds the bitrate of the base_frame, as specified in the mpeg1_header, the mc_extension_data_part1 shall comprise at least the mc_header. The base_frame is decodable by itself by an ISO/IEC 11172-3 decoder.

2.5.2.2 Base Frame Layer I

mpeg1_header - Part of the bit stream containing synchronisation and state information.

mpeg1_error_check - Part of the bit stream containing information for error detection in the MPEG-1 part of the bit stream.

mpeg1_audio_data - Part of the bit stream containing information on the audio samples of the MPEG-1 part of the bit stream.

mc_extension_data_part1, mc_extension_data_part2, mc_extension_data_part3 - These three parts plus optional ext_data from an extension frame form the complete multichannel extension field 'mc_extension' of one audio frame, containing the mc_header, mc_error_check, mc_composite_status_info, mc_audio_data and ml_audio_data.

continuation_bit - One bit with the value '0', to aid synchronisation.

mpeg1_ancillary_data - Part of the bit stream that may be used for ancillary data.

2.5.2.3 Base Frame Layer II

mpeg1_header - See 2.5.2.2.

mpeg1_error_check - See 2.5.2.2.

mpeg1_audio_data - See 2.5.2.2.

mc_extension_data_part1() - This part plus optional ext_data from an extension frame form the multichannel extension field, containing the mc_header, mc_error_check, mc_composite_status_info, mc_audio_data and ml_audio_data.

mpeg1_ancillary_data - See 2.5.2.2.

2.5.2.4 Base Frame Layer III

mpeg1_header - See 2.5.2.2.

mpeg1_error_check - See 2.5.2.2.

mpeg1_audio_side_info - This is the same as the syntax element audio_data() in ISO/IEC 11172-3, 2.4.1.7 but without main_data().

mpeg1_main_data - This is the same as the syntax element main_data() in ISO/IEC 11172-3, 2.4.1.7. This data is accessed using main_data_begin (see syntax element audio_data() in ISO/IEC 11172-3, section 2.4.1.7) and contains MPEG-1 audio data as well as MPEG-2 audio data (multichannel and multilingual) and ancillary data.

mpeg1_audio_main_data - This is the same as the syntax element main_data() in ISO/IEC 11172-3, 2.4.1.7 but without ancillary data.

mc_extension_data_part1 - This part plus optional ext_data from an extension frame form the multichannel extension field, containing the mc_header, mc_error_check, mc_composite_status_info, mpeg2_audio_side_info and mpeg2_audio_main_data.

mpeg1_ancillary_data - See 2.5.2.2.

2.5.2.5 Extension Frame

ext_header - part of the extension bit stream containing synchronisation and state information.

ext_data - part of the multichannel/multilingual field in the bit stream that contains those bits that cannot be transmitted in the base_frame.

ext_ancillary_data - Part of the extension bit stream that can be used for carrying ancillary data for Layer I and II. For Layer III the additional ancillary data for the multichannel/multilingual extension 'mpeg2_ancillary_data' is located in the mpeg2_audio_main_data, independent whether the extension bit stream is used or not (see 2.5.1.12.2).

2.5.2.6 MPEG1 Header

See ISO/IEC 1172-3, 2.4.2.3.

2.5.2.7 MPEG1 Error Check

See ISO/IEC 11172-3, 2.4.2.4.

2.5.2.8 MPEG1 Audio Data

See ISO/IEC 11172-3, 2.4.2.5, 2.4.2.6 and 2.4.2.7.

2.5.2.9 MPEG-1 Ancillary Data

See ISO/IEC 11172-3, 2.4.2.8.

2.5.2.10 Extension header

ext_syncword - A 12 bit string '0111 1111 1111', to synchronise the base bit stream and the extension bit stream.

ext_crc_check - Mandatory 16 bit check word. The calculation of the CRC-check begins with the first bit of the ext_length field. The number of bits included in the CRC check equals 128, or less if the end of the ext_data field is reached earlier.

ext_length - 11 bit number, indicating the total number of bytes in the extension frame.

ext_ID_bit - Reserved for future use. Should be set to '0' for an ISO/IEC 13818-3 extension frame.

2.5.2.11 Ext Ancillary Data

ext_ancillary_bit - User definable. The number of extension ancillary bits (no_of_ext_ancillary_bits) equals ext_length minus the number of bits used for ext_header and ext_data.

2.5.2.12 MC Extension

2.5.2.12.1 MC Extension Layer I and II

mc_header - Part of the bit stream containing synchronisation and state information on the multichannel and multilingual extension of the bit stream.

mc_error_check - Part of the bit stream containing information for error detection in the multichannel extension part of the bit stream.

mc_composite_status_info - Part of the bit stream containing information about the status of the composite coding mode.

mc_audio_data - Part of the bit stream containing information on the audio samples of the multichannel extension part of the bit stream.

ml_audio_data - Part of the bit stream containing information on the audio samples of the commentary extension part of the bit stream.

2.5.2.12.2 MC Extension Layer III

mc_header - see 2.5.2.12.1.

mc_error_check - see 2.5.2.12.1.

mc_composite_status_info - see 2.5.2.12.1.

mpeg2_audio_side_info - Part of the bit stream containing information needed for decoding the multichannel extension and the multilingual extension.

byte_align_bit - Private bit used to do a byte alignment of the mpeg2_audio_main_data.

mpeg2_audio_main_data - Part of the bit stream containing information on the audio samples of the multichannel and multilingual extension. This data is accessed via the mc_data_begin element in the syntax element mc_composite_status_info (see section 2.5.1.16). Because of the variable nature of Huffman coding in Layer III and the bit reservoir technique, the mpeg2_audio_main_data for a frame does not generally follow the mpeg2_audio_side_info of that frame. The mpeg2_audio_main_data for a frame starts at a location in the bit stream preceding the mc_header and mpeg2_audio_side_info of a frame at a negative offset given by the value of mc_data_begin (see definition of main_data_begin in ISO/IEC 11172-3, section 2.4.2.7). The number of bytes used for information other than mpeg2_audio_main_data is not taken into account when applying mc_data_begin.

mc_side_info - Part of the bit stream containing information needed for decoding of the full bandwidth channels.

lfe_side_info - Part of the bit stream containing information needed for decoding of the low frequency enhancement channel.

ml_side_info - Part of the bit stream containing information needed for decoding of the multilingual channels.

mc_audio_main_data - Part of the bit stream containing information on the audio samples of the full bandwidth channels.

lfe_audio_main_data - Part of the bit stream containing information on the audio samples of the low frequency enhancement channel.

ml_audio_main_data - Part of the bit stream containing information on the audio samples of the multilingual channels.

mpeg2_ancillary_data - This is the ancillary data of the multichannel/multilingual extension part. The number of ancillary data bits l3_mpeg2_ancillary_bits corresponds to the distance between the end of the multichannel/multilingual Huffman data and the location in the mpeg2_audio_main_data where the next frame's mc_data_begin pointer points to.

2.5.2.13 MC Header

ext_bit_stream_present - One bit to indicate whether an extension bit stream exists, which contains a remainder of the multichannel and multilingual audio information in case the information does not fit in one base_frame.

'0' no extension stream present
'1' extension bit stream present

If the value of ext_bit_stream_present changes, a reset of the decoder may occur. In case of a variable bitrate application using an extension bit stream, if the required number of bits for a certain audio frame already fits in the base_frame, and consequently does not require an ext_frame, the ext_frame could consist of only an ext_header, to avoid such a reset.

n_ad_bytes - 8 bits that form an unsigned integer indicating how many bytes are used for the MPEG-1 compatible ancillary data field if an extension bit stream exists (Layer I and Layer II) or if Layer III is used (with or without extension bit stream).

centre - Two bits to indicate whether a centre channel is contained in the multiplex, and to indicate its bandwidth.

'00' no centre channel present
'01' centre channel present
'10' not defined
'11' centre bandwidth limited (Phantom coding)

If the centre signal is bandwidth limited, the subbands above subband 11 are not transmitted. The decoder shall set the variable centre_limited[mch][sb] to true for these subbands, and the allocation of these subbands shall be set to zero:

```
for (sb=0; sb<12; sb++)
    centre_limited[centre][sb]=false;
if (centre=='11')
    for (sb=12; sb<msblimit; sb++)
        centre_limited[centre][sb]=true;
else
    for (sb=12; sb<msblimit; sb++)
        centre_limited[centre][sb]=false;
```

For those subbands, where centre_limited [mch][sb] is true, only transmission channel allocations that include the centre signal can be used. In the case of dynamic crosstalk which includes the centre channel, no scalefactors are transmitted for those subbands.

surround - Two bits to indicate whether surround channels are contained in the mc_extension bit stream.

'00' no surround
'01' mono surround
'10' stereo surround
'11' no surround, but second stereo programme present

lfe - One bit to indicate whether a low frequency enhancement channel is present.

'0' no low frequency enhancement channel present
'1' low frequency enhancement channel present

audio_mix - One bit to indicate whether the signal is mixed for a large listening room, like a theatre, or for a small listening room, like a living room. This bit is to be ignored by the decoder but may be exploited by the reproduction system.

'0' audio programme mixed for a large listening room
'1' audio programme mixed for a small listening room

dematrix_procedure - Two bits to indicate which dematrix procedure has to be applied in the decoder. The dematrix_procedure affects the tc_allocation decoding and the denormalisation procedure. For the procedures see 2.5.3.2.1.1 and 2.5.3.2.5.

'00' procedure 0
'01' procedure 1
'10' procedure 2
'11' procedure 3

The value '10' can only occur in combination with a 3/1 or 3/2 configuration.

no_of_multi_lingual_ch - An unsigned integer of three bits to indicate the number of multilingual or commentary channels in the mc_extension bit stream.

multi_lingual_fs - One bit to indicate whether the sampling frequencies of the multilingual and the main audio channels are the same or not. Equals '1' if the sampling frequency of the multilingual channels is chosen to be 1/2 * Fs (of the main audio channels), '0' if both sampling frequencies are the same.

multi_lingual_layer - One bit to indicate whether Layer II ml or Layer III ml is used. With Layer I, Layer II ml is always used.

ISO/IEC 11172-3 basic stereo	multi_lingual_layer	Layer
Layer I	X	Layer II ml
Layer II	'0'	Layer II ml
Layer II	'1'	Layer III ml
Layer III	'0'	Layer II ml
Layer III	'1'	Layer III ml

copyright_identification_bit - One bit which is part of a 72-bit copyright identification field. The start is indicated by the copyright_identification_start bit. The field consists of an 8-bit copyright_identifier, followed by a 64-bit copyright_number. The copyright identifier is given by a Registration Authority as designated by SC29. The copyright_number is a value which identifies uniquely the copyrighted material.

copyright_identification_start - One bit to indicate that the copyright_identification_bit in this audio frame is the first bit of the 72-bit copyright identification. If no copyright identification is transmitted, this bit should be kept '0'.

'0' no start of copyright identification in this audio frame
'1' start of copyright identification in this audio frame

2.5.2.14 MC Error Check

mc_crc_check - Mandatory 16 bits check word for error detection. Also used to detect whether multichannel or multilingual information is available. In Layer I and II, the calculation begins with the first bit of the multichannel header and ends with the last bit of the scfsi field, but excluding the mc_crc_check field itself.

In Layer III, the calculation begins with the first bit of the multichannel header and ends with the last bit of ML_header() .

2.5.2.15 MC Composite Status Info Layer I, II

tc_sbgr_select - One bit indicating whether the tc_allocation is valid for all subbands or for individual subband groups. Equals ´1´ if valid for all subbands, ´0´ if tc_allocation is valid for individual subband groups. The following table shows the assignment of subbands to the subband groups sbgr.

sbgr	subbands included in the subband group
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8...9
9	10...11
10	12...15
11	16...31

dyn_cross_on - One bit indicating whether dynamic crosstalk is used. Equals ´1´ if dynamic crosstalk is used, ´0´ otherwise.

mc_prediction_on - One bit indicating whether mc_prediction is used. Equals ´1´ if mc_prediction is used, ´0´ otherwise.

tc_allocation, tc_allocation[sbgr] - Contains information on the transmission channel allocation for all subbands or for the subbands in subband group sbgr, respectively. T0 always contains Lo, and T1 always contains Ro. The case of dematrix_procedure equals '11' implies tc_allocation[sbgr]==0. If Phantom coding is used (centre=='11'), the centre channel must be contained in the additional transmission channels for the subband groups involved, i.e. for those subband groups the value of tc_allocation must be limited to

0,3,4,5 in 3/2 mode,
0,3,4 in 3/1 mode,
0 in 3/0 and 3/0 + 2/0 mode.

A) 3/2 configuration (nmch==3, length of tc_allocation field: 3 bits):

tc_allocation	T2	T3	T4
0	C^W	LS^W	RS^W
1	L^W	LS^W	RS^W
2	R^W	LS^W	RS^W
3	C^W	L^W	RS^W
4	C^W	LS^W	R^W
5	C^W	L^W	R^W
6	R^W	L^W	RS^W
7	L^W	LS^W	R^W

B) 3/1 configuration (nmch==2, length of tc_allocation field: 3 bits):

tc_allocation	T2	T3	
0	C^W	S^W	
1	L^W	S^W	
2	R^W	S^W	
3	C^W	L^W	
4	C^W	R^W	
5	L^W	R^W	{ only possible for dematrix_procedure '10' }

C) 3/0 (+ 2/0) configuration (nmch==1 in 3/0 mode, nmch==3 in 3/0+2/0 mode, length of tc_allocation field: 2 bits):

tc_allocation	T2
0	C^W
1	L^W
2	R^W

In the case of a second stereo programme, T3 contains L2 and T4 contains R2 of the second stereo programme.

D) 2/2 configuration (nmch==2, length of tc_allocation field: 2 bits):

tc_allocation	T2	T3
0	LS^W	RS^W
1	L^W	RS^W
2	LS^W	R^W
3	L^W	R^W

E) 2/1 configuration (nmch==1, length of tc_allocation field: 2 bits):

tc_allocation	T2
0	S^W
1	L^W
2	R^W

F) 2/0 (+ 2/0) configuration (nmch==0 in 2/0 mode, nmch==2 in 2/0+2/0 mode, length of tc_allocation field: 0 bits):

In the case of a second stereo programme, T2 contains L2 and T3 contains R2 of the second stereo programme.

G) 1/0 (+ 2/0) configuration (nmch==0 in 1/0 mode, nmch==2 in 1/0+2/0 mode):

Length of tc_allocation field: 0 bits.

In the case of a second stereo programme, T1 contains L2 and T2 contains R2 of the second stereo programme.

dyn_cross_LR - One bit indicating whether C^W and/or S^W shall be copied from Lo (dyn_cross_LR=='0'), or from Ro (dyn_cross_LR=='1').

dyn_cross_mode[sbgr] - One to four bits, indicating between which transmission channels dynamic crosstalk is active for the subbands in subband group sbgr. For those subbands, the bit allocation and subband samples are missing in the bit stream. The number of bits of this field depends on the channel configuration which can be either 3/2 (A), 3/1 (B), 3/0 (C), 2/2 (D) and 2/1 (E). The following tables give the missing transmission channels for all modes. If a transmission channel Tj is missing (indicated by a '–' in the tables), the requantised but not yet re-scaled subband samples for the corresponding audio channel have to be copied according to the following rules:

- if there is a term Tij in the same row of the table, the subband samples in transmission channel j have to be copied from transmission channel i.
- if there is a term Tijk in the same row of the table, the subband samples in transmission channels j and k have to be copied from transmission channel i.
- for all other cases:
  - Lw and LSw shall be copied from Lo,
  - Rw and RSw shall be copied from Ro,
  - Cw and Sw shall be copied from Lo if dyn_cross_LR=='0', or from Ro if dyn_cross_LR=='1'.

Initially, for all subbands of all transmission channels, the variable dyn_cross[Tx][sb] has to be set to false. Then, for subbands of transmission channels of which the bit allocation and samples are not transmitted, the variable dyn_cross[mch][sb] must be set to true:

```
for ( sb = lim1; sb <= lim2; sb++)
        dyn_cross[Tx][ sb] = true;
```

where lim1 and lim2 stand for the subband group bounds (see the table in the beginning of 2.5.2.15). The bit allocation of subbands for which dyn_cross[Tx][sb] is true, has to be copied from the corresponding transmission channel. If that allocation is zero, the scalefactor select information and the scalefactors are not transmitted.

A) 3/2 configuration (length of field 'dyn_cross_mode': 4 bits):

dyn_cross_mode[sbgr]	transmission channel			Remarks
'0000'	T2	T3	T4	{ No dynamic crosstalk }
'0001'	T2	T3	–	
'0010'	T2	–	T4	
'0011'	–	T3	T4	
'0100'	T2	–	–	
'0101'	–	T3	–	
'0110'	–	–	T4	
'0111'	–	–	–	
'1000'	T2	T34	–	No prediction for T34
'1001'	T23	-	T4	No prediction for T23
'1010'	T24	T3	–	No prediction for T24
'1011'	T23	–	–	No prediction
'1100'	T24	–	–	No prediction
'1101'	–	T34	–	No prediction
'1110'	T234	–	–	No prediction
'1111'	forbidden			

B) 3/1 configuration (length of field 'dyn_cross_mode': 3 bits):

dyn_cross_mode[sbgr]	transmission channel		Remarks
'000'	T2	T3	{No dynamic crosstalk}
'001'	T2	–	
'010'	–	T3	
'011'	–	–	
'100'	T23	–	No prediction
'101'	forbidden		
'110'	forbidden		
'111'	forbidden		

C) 3/0 (+ 2/0) configuration (length of field 'dyn_cross_mode': 1 bit):

dyn_cross_mode[sbgr]	transmission channel	Remarks
'0'	T2	{ No dynamic crosstalk }
'1'	–	

D) 2/2 configuration (length of field 'dyn_cross_mode': 3 bits):

dyn_cross_mode[sbgr]	transmission channel		Remarks
'000'	T2	T3	{ No dynamic crosstalk }
'001'	T2	–	
'010'	–	T3	
'011'	–	–	
'100'	T23	–	No prediction
'101'	forbidden		
'110'	forbidden		
'111'	forbidden		

E) 2/1 configuration (length of field 'dyn_cross_mode': 1 bit):

dyn_cross_mode[sbgr]	transmission channel	Remarks
'0'	T2	{ No dynamic crosstalk }
'1'	–	

F) 2/0 (+2/0) configuration (length of field 'dyn_cross_mode': 0 bits).

G) 1/0 (+2/0) configuration (length of field 'dyn_cross_mode': 0 bits).

dyn_second_stereo[sbgr] - One bit indicating whether dynamic crosstalk is used in the second stereo programme. Equals '0' if there is no dynamic crosstalk used in the second stereo programme. If it is '1', subband samples of R2 (Transmission channel T3 in 2/0 + 2/0 configuration, T4 in 3/0 + 2/0 configuration) are copied from L2 (transmission channel T2 in 2/0 + 2/0 configuration, T3 in 3/0 + 2/0 configuration).

mc_prediction[sbgr] - One bit indicating whether or not multichannel redundancy reduction by prediction is used in subband group sbgr. The use of mc_prediction is limited to subband groups 0 to 7. Equals '1' if redundancy reduction is used, '0' if no redundancy reduction is used.

predsi[sbgr][px] - Predictor select information. This indicates whether the predictor indexed by px in subband group sbgr is used and if yes, how many coefficients are transferred.

'00'	predictor is not used
'01'	1 coefficient is transferred
'10'	2 coefficients are transferred
'11'	3 coefficients are transferred

The maximum number of used predictors npred depends on the dynamic crosstalk (dyn_cross_mode). The values of npred are as follows:

configuration	dynamic crosstalk															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3/2	6	4	4	4	2	2	2	0	2	2	2	0	0	0	0	-
3/1	4	2	2	0	0	-	-	-								
3/0	2	0														
2/2	4	2	2	0	0	-	-	-								
2/1	2	0														

2.5.2.16 MC Composite Status Information Layer III

mc_data_begin - 11 bits indicating the negative offset in bytes from the first byte of the actual frame. The number of bytes belonging to the MPEG-1 part of the frame, the mc_header, mc_error_check and mc_composite_status_info is not taken into account. This means if mc_data_begin == 0, the mc_main_data starts after the last byte_aligned_bit.

seg_list_present[gr][ch] - Is only transmitted if the channel is flagged as present in mc_header(). If seg_list_present is not flagged (which is only legal for at maximum two channels), the respective channel is reconstructed by dematrixing of the left/right compatible and the transmitted channels.

seg_list_nodef[gr][ch] - Indicates if the segment list is transmitted or if the default is used. The default segment list indicates that the channel is transmitted completely in the specified channel.

segment_list_repeat[ch] - Indicates if the segment list of the second granule is identical to the segment list of the first one. This variable is only transmitted if the segment list of the first granule is transmitted is not the default type.

tc_present[gr][ch] - Indicates whether information about a (transmitted channel tc) can be found in the bit stream. The difference between seg_list_present and tc_present is that there can be fewer transmitted channels than output channels even if considering channels which are reconstructed by dematrixing. A channel which does have a segment list, but no corresponding tc must be reconstructed via intensity stereo. A channel which has tc_present set can be referenced by tc_select. tc_present =='1' means the channel is present. For audio channels which are flagged in the mc_header() as not present, a tc_present value of '0' is assumed.

ch_present(ch) - Is a function indicating whether audio channel ch is present, as indicated in mc_header().

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

block_type[gr]	window type
'00'	normal block
'01'	start block
'10'	3 short windows
'11'	end block

The block_type gives the information about assembling of values in the block and about length and count of the transforms (see figure A.4 for a schematic, annex C for an analytic description). The polyphase filterbank is described in ISO/IEC 11172-3, 2.4.3.

In the case of long blocks (block_type not equal to '10'), the IMDCT generates an output of 36 values for 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 preceding block. The resulting vector is the input of the synthesis part of the polyphase filterbank of one band.

In the case of short blocks (block_type is '10'), three transforms are performed producing 12 output values each. The three vectors are windowed and overlapped. 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.

If block_type is not '00', several other variables are set by default:

region0_count = 7 (in case of block_type=='01' or block_type=='11')
region0_count = 8 (in case of block_type=='10')
region1_count = 36 Thus all remaining values in the big_value region are contained in region 1.

dematrix_length - Number of scalefactorband_groups where the dematrixed channels are explicitly transmitted. The first dematrix_length scalefactorband_groups have no joint stereo information (tc_select) transmitted. If dematrix_length =='0000', the channels to be reconstructed by dematrixing are determined by seg_list_present.

dematrix_select[sbgr] - Information given for the first dematrix_length scalefactorband_groups. It tells which of the output channels should be reconstructed by dematrixing using the formula of the compatibility matrix. The following table shows the mapping of transmitted value in dematrix_select to the channels which have to be reconstructed by dematrixing. x means that this channel has to reconstructed by dematrixing and '0' means no dematrixing of this channel.

3/2, 3/1 and 2/2 configuration (4 bits)

dematrix_select:	L	R	C	LS / S	RS	valid in 3/2	valid in 3/1	valid in 2/2
'0000'	0	0	0	0	0	y	y	y
'0001'	x	0	0	0	0	y	y	y
'0010'	0	x	0	0	0	y	y	y
'0011'	x	x	0	0	0	y	y	y
'0100'	0	0	x	0	0	y	y	n
'0101'	x	0	x	0	0	y	y	n
'0110'	0	x	x	0	0	y	y	n
'0111'	0	0	0	x	0	y	y	y
'1000'	0	x	0	x	0	y	y	y
'1001'	0	0	x	x	0	y	n	n
'1010'	0	0	0	0	x	y	n	y
'1011'	x	0	0	0	x	y	n	y
'1100'	0	0	x	0	x	y	n	n
'1101'	0	0	0	x	x	y	n	y
'1110'	x	0	0	x	0	n	y	n
'1111'	-	-	-	-	-	n	n	n

3/0 and 2/1 configuration (3 bits)

dematrix_select:	L	R	C / S	valid in 3/0	valid in 2/1
'000'	0	0	0	y	y
'001'	x	0	0	y	y
'010'	x	x	0	y	y
'011'	0	0	x	y	y
'100'	x	0	x	y	y
'101'	0	x	x	y	y
'110'	-	-	-	n	n
'111'	-	-	-	n	n

scalefactorband_group - For transmitting of the dematrix_length and the segment list, the scalefactorbands are grouped together. The following two tables show the grouping for long (block_type =='00', '01', '11') and short blocks (block_type == '10'). For short blocks the scalefactorband_group includes the respective values of all three subblocks.

Width and begin of each scalefactorband_group (sbgr) in scalefactorbands:

sbgr #	Long blocks (block_type =='00', '01', '11')		Short block (block_type == '10')	
	sbgr width	sbgr begin	sbgr width	sbgr begin
0	3	0	1	0
1	3	3	1	1
2	3	6	1	2
3	1	9	1	3
4	1	10	1	4
5	1	11	1	5
6	1	12	1	6
7	1	13	1	7
8	1	14	1	8
9	1	15	1	9
10	1	16	1	10
11	1	17	2	11
12	1	18	–	13
13	1	19	–	–
14	2	20	–	–
15	–	22	–	–

attenuation_range[gr][ch] - The attenuation of the segment list has four different ranges. The following table indicates the range of the attenuation:

attenuation_range:	number of bits for attenuation
0	2
1	3
2	4
3	5

attenuation_scale[gr][ch] - Determines the attenuation step size. For attenuation_scale == 0, the step size is $1/\sqrt{\sqrt{2}}$. For attenuation_scale == 1, the step size is $1/\sqrt{2}$.

seg_length[gr][ch][seg] - Is the number of scalefactorband groups which are multiplied with attenuation from tc_select and are copied into the channel (ch). Seg_length == 0 stops the transmitting of tc_select and attenuation immediately. The unselected scalefactorband_groups are set to zero.

tc_select[gr][ch][seg] - Indicates the number of the transmitted channel which is the source for segment list processing. tc_select == 7 indicates that the values in this segment are reconstructed by dematrixing.

attenuation[gr][ch][seg][sbgr] - For each scalefactorband_group, one attenuation is transmitted to assemble the channel. The width of attenuation can vary from 2 to 5 bits. This is indicated by attenuation_range. The step size of attenuation is determined by attenuation_scale and can vary between $\sqrt{2}$ and $\sqrt{\sqrt{2}}$. If tc_select == 7, this means dematrixing of the channel and no attenuations are transmitted. If tc_select == ch, this means that the transmitted channel is the selected channel and no attenuations are transmitted.

mc_prediction_on - One bit indicating whether mc_prediction is used. Equals ´1´ if mc_prediction is used, ´0´ if not.

mc_prediction[sbgr] - One bit indicating whether or not multichannel redundancy reduction by prediction is used in subband group sbgr. Equals '1' if redundancy reduction is used, '0' if no redundancy reduction is used.

predsi[sbgr][pci] - Predictor select information, indicating whether the predictor coefficient indexed by pci in subband group sbgr is transferred. Equals '1' if coefficient is transmitted, '0' otherwise.

pred_coef[sbgr][pci] - Actual prediction coefficient used for the subbands in subband group sbgr and index pci.

2.5.2.17 MC Audio Data Layer I, II

lfe_allocation - Contains information on the quantiser used for the samples in the low frequency enhancement channel. The 4 bits in this field form an unsigned integer used as an index to the following table, which gives the number of bits per sample as well as the number of levels used for quantisation. Thus, lfe_allocation indicates the number of bits used to code the samples in the low frequency enhancement channel. The following table is valid for all sampling frequencies.

lfe_allocation	bits per sample	number of levels
0	0	-
1	2	3
2	3	7
3	4	15
4	5	31
5	6	63
6	7	127
7	8	255
8	9	511
9	10	1023
10	11	2047
11	12	4095
12	13	8191
13	14	16383
14	15	32767
15	16	65535

allocation[mch][sb] - Contains information on the quantiser used for the samples in subband sb of the multichannel extension channel mch. Whether this allocation field exists for a certain subband and channel depends on the composite_status_info. The bits in this field form an unsigned integer used as an index to the relevant table in Table B.2 "Layer II bit allocation table" of ISO/IEC 11172-3, which gives the number of levels used for quantisation. Table B.2.a shall be used if Fs equals 48 kHz, table B.2.b shall be used if Fs equals 44,1 kHz or 32 kHz, regardless of the bitrate. The value of msblimit should be set to sblimit of the relevant table.

scfsi[mch][sb] - Scalefactor select information, indicating the number of scalefactors transmitted for subband sb of the multichannel extension channel mch. The audio 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, second one for parts 1 and 2.

delay_comp[sbgr][px] - Three bits specifying a shift of 0, 1, 2, ... , 7 subband samples for delay compensation in subband group sbgr and predictor index px.

pred_coef[sbgr][px][pci] - Actual coefficient of predictor with up to second order in subband group sbgr and predictor index px.

lf_scalefactor - Indicates the factor by which the requantised samples of the low frequency enhancement channel should be multiplied. The six bits constitute an unsigned integer, index to table B.1 "Layer I, II scalefactors" of ISO/IEC 11172-3.

scalefactor[mch][sb][p] - Indicates the factor by which the requantised samples of subband sb of part p of the audio frame of multichannel extension channel mch should be multiplied. The six bits constitute an unsigned integer, index to table B.1 "Layer I, II scalefactors" of ISO/IEC 11172-3.

lf_sample[gr] - Coded representation of the single sample in granule gr of the low frequency enhancement channel.

samplecode[mch][sb][gr] - Coded representation of three consecutive samples in granule gr of subband sb of multichannel extension channel mch.

sample[mch][sb][s] - Coded representation of the sample s of subband sb of multichannel extension channel mch.

2.5.2.18 ML Audio Data Layer I, II

allocation[mlch][sb] - Contains information on the quantiser used for the samples in subband sb of the multilingual extension channel mlch. The bits in this field form an unsigned integer used as an index to the relevant table in Table B.2 "Layer II bit allocation table" of ISO/IEC 11172-3, which gives the number of levels used for quantisation. Table B.2.a shall be used if Fs equals 48 kHz, table B.2.b shall be used if Fs equals 44,1 kHz or 32 kHz, regardless of the bitrate. If the half sampling frequency is used for the multilingual channels (multi_lingual_fs=='1'), table B.1 of this part of ISO/IEC 13818 shall be used. The value of mlsblimit should be set to sblimit of the relevant table.

scfsi[mlch][sb] - Scalefactor select information, indicating the number of scalefactors transferred for subband sb of the multilingual extension channel mlch. The audio frame is divided into three equal parts of 12 (if multi_lingual_fs equals '0', full sampling frequency) or 6 (if multi_lingual_fs equals '1', half sampling frequency) 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, second one for parts 1 and 2.

scalefactor[mlch][sb][p] - Indicates the factor by which the requantised samples of subband sb of part p of the audio frame of multilingual extension channel mlch should be multiplied. The six bits constitute an unsigned integer, index to table B.1 "Layer I, II scalefactors" of ISO/IEC 11172-3.

samplecode[mlch][sb][gr] - Coded representation of three consecutive samples in granule gr of subband sb of multilingual extension channel mlch. The number of granules ngr equals 12 if multi_lingual_fs equals '0' (full sampling frequency) and equals 6 if multi_lingual_fs equals '1' (half sampling frequency).

sample[mlch][sb][s] - Coded representation of the sample s of subband sb of multilingual extension channel mlch.

2.5.2.19 MC Audio Data Layer III

data_present[gr][tc][sfb] - Is a map describing which data (dependent on granule, transmitted channel and scalefactorband) are actually transmitted. This map is not transmitted but recovered in the decoder by determining the scalefactorbands which are referenced by dematrix_select or the segment_lists.

js_carrier[gr][tc][sbgr] - Is a map describing which scalefactorband_group data (dependent on granule, transmitted channel and scalefactorband_group) are used as a carrier for joint stereo transmission. This map is not transmitted but recovered in the decoder by determining the scalefactorband_groups which are referenced with a tc_select != ch.

matrix_attenuation_present - Denotes whether or not the matrix_attenuation is transmitted.
The matrix_attenuation_present equals '1' if matrix_attenuation is transmitted.

matrix_attenuation_l/r[gr][ch][sbgr] - In the case of joint stereo coding, correction values are needed to get energy preservation in the compatible downmixed signals Lo and Ro. In the decoder an attenuation is applied to get correct dematrixing.

The actual attenuation factors are calculated as:

attenuation = 1 / (√√2 ** matrix_attenuation_l/r)

For the dematrix procedure using the Lo (Ro) channel, matrix_attenuation_l (matrix_attenuation_r) is used. The modification of the dematrixing operation is described in the decoding process.

scfsi[tc][scfsi_band] - In Layer III, the scalefactor selection information works similarly to audio Layer II. The main difference is the use of the variable scfsi_band to apply scfsi to groups of scalefactors instead of single scalefactors. The application of scalefactors to granules is controlled by scfsi. The scalefactor selection information is only transmitted if the channel is transmitted in both granules. The others are set to zero.

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

If short windows are switched on, i.e. block_type=='10' for one of the granules, then scfsi is always '0' for this frame.

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

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

part2_3_length[gr][tc] - Contains the number of main_data bits used for scalefactors and Huffman code data.

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

```
xxxxxxxxxxxxxxxxxxxxxxx-------------------00000000000000000000000000
|                     |                  |                         |
1                 bigvalues*2     bigvalues*2+count1*4          iblen
```

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

global_gain[gr][tc] - The quantiser step size information is transmitted in the side information variable global_gain. It is logarithmically quantised. For the application of global_gain, refer to the formula in ISO/IEC 11172-3, 2.4.3.4, "Formula for requantisation and all scaling".

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

if block_type is '00', '01', '11':
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 '10':
slen1: length of scalefactors for the scalefactor bands 0 to 5
slen2: length of scalefactors for the scalefactor bands 6 to 11

scalefac_compress[gr]	slen1	slen2
'0000'	0	0
'0001'	0	1
'0010'	0	2
'0011'	0	3
'0100'	3	0
'0101'	1	1
'0110'	1	2
'0111'	1	3

'1000'	2	1
'1001'	2	2
'1010'	2	3
'1011'	3	1
'1100'	3	2
'1101'	3	3
'1110'	4	2
'1111'	4	3

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

subblock_gain[gr][tc][window] - Indicates the gain offset (quantisation: factor 4) from the global gain for one subblock. Transmitted only with block type 2 (short windows). The values of the subblock have to be divided by $4^{\text{(subblock\_gain[window])}}$ in the decoder. See ISO/IEC 11172-3, 2.4.3.4 "Formula for requantisation and all scaling".

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

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

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

If block_type=='10', the total amount of scalefactor bands for the granule in this case is 12*3=36. If block_type!='10', the amount of scalefactor bands is 21.

region1_count[gr][tc] - Region1_count counts one less than the number of scalefactor bands in region 1. Again, if block_type=='10' the scalefactor bands representing different time slots are counted separately.

preflag[gr][tc] - This is a shortcut for additional high frequency amplification of the quantised values. If preflag is set, the values of a table are added to the scalefactors (see ISO/IEC 11172-3, table B.6). This is equivalent to multiplication of the requantised scalefactors with table values. If block_type=='10' (short blocks) preflag is never used.

scalefac_scale[gr][tc] - The scalefactors are logarithmically quantised with a step size of 2 or √2 depending on scalefac_scale. The following table indicates the scalefactor multiplier used in the requantisation equation for each stepsize.

scalefac_scale[gr]	scalefac_multiplier
'0'	0,5
'1'	1

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

count1table_select[gr]	
'0'	ISO/IEC 11172-3, table B.7 - A
'1'	ISO/IEC 11172-3, table B.7 - B

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

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

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 ISO/IEC 11172-3, table B.8). The scale factor for frequency lines above the highest line in the tables is zero, which means that the actual multiplication factor is 1,0.

The scalefactors are logarithmically quantised. The quantisation step is set with scalefac_scale.

Scalefactors of scalefactor bands which are not selected by a transmitted channel are not transmitted. This means the scalefactors will be packed together for transmission and have to be unpacked for decoding or dematrixing.

Huffmancodebits() - Huffman encoded data.

The syntax for Huffmancodebits() shows how quantised values are encoded. Within the big_values partition, pairs of quantised values with an absolute value less than 15 are directly coded using a Huffman code. The codes are selected from Huffman tables 0 through 31 in ISO/IEC 11172-3, table B.7. Values (x,y) are always coded in pairs. If quantised values of magnitude greater than or equal to 15 are coded, these values are coded with a separate field following the Huffman code. If one or both values of a pair are not zero, one or two sign bits are appended to the code word.

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

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

The syntax for Huffmancodebits() contains the following fields and parameters:

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

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

Within the count1 partition, quadruples of values with magnitude less than or equal to one are coded. Magnitude values are coded using a Huffman code from tables A or B in ISO/IEC 11172-3, table B.7. For each non-zero value, a sign bit is appended after the Huffman code symbol.

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

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

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

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

For clarity, the parameter "count1" is used in this part of ISO/IEC 13818 to indicate the number of Huffman codes in the count1 region. However, unlike the bigvalues partition, the number of values in the count1 partition is not explicitly coded by a field in the syntax. The end of the count1 partition is known only when all bits for the granule (as specified by part2_3_length), have been exhausted, and the value of count1 is known explicitly after decoding the count1 region.

The order of the Huffman data depends on the block_type of the granule. If block_type is ‘00', '01' or '11', the Huffman encoded data is ordered in terms of increasing frequency.

If block_type=='10' (short blocks) the Huffman encoded data is ordered in the same order as the scalefactor values for that granule. The Huffman encoded data is given for successive scalefactor bands, beginning with scalefactor band 0. Within each scalefactor band, the data is given for successive time windows, beginning with window 0 and ending with window 2. Within each window, the quantised values are then arranged in order of increasing frequency.

2.5.2.20 LFE Side Info, Layer III

lfe_table_select - Determines the Huffman code table that is used to decode the spectral values of the low frequency enhancement channel. The interpretation is the same as for table_select.

lfe_hc_len - Determines the total length of the Huffman coded spectral values of the low frequency enhancement channel for both granules.

lfe_gain - Determines the quantiser step size of the low frequency enhancement channel. The interpretation is the same as for global_gain.

2.5.2.21 LFE Audio Main Data, Layer III

lfe_audio_main_data() - Contains the Huffman coded spectral values for the low frequency enhancement channel in both granules. lfe_main_data() is be interpreted just like a Huffmancodebits() structure that consists of big_values and zero_values only. Similarly to count1 in Huffmancodebits(), the number of Huffman codes in lfe_main_data() (i.e. lfe_bigval) is not transmitted explicitly. Instead, it is recovered by Huffman decoding until all bits indicated in lfe_hc_len have been exhausted. Unlike the Huffmancodebits() structure, the decoded values x and y denote the values of a spectral coefficient for granule 0 and 1 respectively.

2.5.2.22 ML Side Info, Layer III

If multi_lingual_fs==0, see audio_data() syntax in ISO/IEC 11172-3, 2.4.2.7, but without main_data_begin, private_bits and main_data().

If multi_lingual_fs==1, see audio_data() syntax in 2.4.2.7 of this document, but without main_data_begin, private_bits and main_data().

2.5.2.23 ML Audio Data Layer III

See ISO/IEC 11172-3, 2.4.2.7, or this part of ISO/IEC 13818, 2.4.2.7, depending on multi_lingual_fs.

2.5.3 The Audio Decoding Process

2.5.3.1 General

The general decoding process closely resembles ISO/IEC 11172-3, 2.4.3. This includes bit allocation decoding, scalefactor select info decoding, scalefactor decoding, requantisation of subband samples in case of Layer I or II, and side information decoding, scalefactor decoding, Huffman decoding, requantisation, reordering, synthesis filterbank and alias reduction in case of Layer III.

The first action is decoding of the backwards compatible signal Lo, Ro according to ISO/IEC 11172-3, 2.4.3. The MPEG-1 ancillary data field is initially assumed to contain the coded multichannel extension. If the mandatory CRC-check yields a valid result, then multichannel decoding will be started. Only one out of each three consecutive ISO/IEC 11172-3 Layer I frames contains a multichannel header. The first 16 or 24 bits of the multichannel extension constitute the multichannel header, providing information on the presence of a centre channel, surround channels, LFE channel, the dematrixing procedure to be followed, the number of multilingual channels contained in the multichannel extension bit stream, the sampling frequency of the multilingual channels, the coding layer which has been applied to the multilingual channels, and a copyright identification.

This part of ISO/IEC 13818 provides the possibility to extend the bit rate beyond the bit rates defined in ISO/IEC 11172-3 for the three Layers, while preserving backwards compatibility with that standard. This is achieved by using an extension bit stream that contains the remainder of the data of the multichannel/multilingual data. A typical example of the structure of this bit stream for Layer II is depicted in Figure A.2 of Annex A. Within the MPEG-2 bit stream, the base bit stream contains at least the MPEG-1 Audio Data and MC header. The corresponding structure for a Layer III bit stream is shown in Figure A.3 of Annex A.

The error detection method of the mandatory CRC-check word which follows directly the mc_header is identical to the one used in ISO/IEC 11172-3, and is described in ISO/IEC 11172-3, 2.4.3.1.

2.5.3.2 Layer I, II decoding

2.5.3.2.1 Composite Coding Modes

2.5.3.2.1.1 Transmission Channel Switching

The allocation of the audio channels to the transmission channels (tc_allocation) can be valid for the whole bandwidth or for individual subband groups depending on the value of tc_sbgr_select. The tc_allocation field determines, depending on the configuration, which audio channels are contained in the transmission channels. For each possibility, a decoding matrix exists that has to be applied in the subband domain to all the transmitted channels in order to obtain the output channels. The matrices are given below. The resulting signals still have to be de-normalised (see 2.5.3.2.5). If dematrix_procedure '11' (see 2.5.2.13) is chosen, all signals can be directly derived from the transmission channels, and no dematrixing is needed. In this case the default value tc_allocation '0' applies. If dematrix_procedure=='10', the following processing is needed on the surround channels:

1. In the 3/2 configuration, calculate the monophonic surround signal:

   $jS^W = 0{,}5*(jLS^W + jRS^W);$

2. The resulting signal jS^W has to be used for the dematrixing.

The following processing may be done on the signals jLS^W and jRS^W in the 3/2 configuration or jS^W in the 3/1 configuration before output (these operations may not be done before dematrixing):

3a –90 degrees phase shift;

3b Dynamic expansion.

Decoding matrices:

The following dematrix equations are valid for the different multichannel configurations. The dematrixing equations do not affect a second stereo programme.

3/2 configuration, dematrixing procedure equals '00' or '01':

tc_allocation	decoding matrix
0	$L^W = Lo - T2 - T3$
	$R^W = Ro - T2 - T4$
	$C^W = T2$
	$LS^W = T3$
	$RS^W = T4$

tc_allocation	decoding matrix
1	$C^W = Lo - T2 - T3$
	$R^W = Ro - C^W - T4$
	$L^W = T2$
	$LS^W = T3$
	$RS^W = T4$

tc_allocation	decoding matrix
2	$C^W = Ro - T2 - T4$
	$L^W = Lo - C^W - T3$
	$R^W = T2$
	$LS^W = T3$
	$RS^W = T4$

tc_allocation	decoding matrix
3	$LS^W = Lo - T3 - T2$
	$R^W = Ro - T2 - T4$
	$C^W = T2$
	$L^W = T3$
	$RS^W = T4$

tc_allocation	decoding matrix
4	L^W = Lo – T2 – T3
	RS^W = Ro – T4 – T2
	C^W = T2
	LS^W = T3
	R^W = T4

tc_allocation	decoding matrix
5	LS^W = Lo – T3 – T2
	RS^W = Ro – T4 – T2
	C^W = T2
	L^W = T3
	R^W = T4

tc_allocation	decoding matrix
6	C^W = Ro – T2 – T4
	LS^W = Lo – T3 – C^W
	R^W = T2
	L^W = T3
	RS^W = T4

tc_allocation	decoding matrix
7	C^W = Lo – T2 – T3
	RS^W = Ro – T4 – C^W
	L^W = T2
	LS^W = T3
	R^W = T4

3/2 configuration, dematrixing procedure equals '10':

tc_allocation	decoding matrix
0	L^W = Lo – T2 + jS^W
	R^W = Ro – T2 – jS^W
	C^W = T2
	jLS^W = T3
	jRS^W = T4

tc_allocation	decoding matrix
1	C^W = Lo – T2 + jS^W
	R^W = Ro – C^W – jS^W
	L^W = T2
	jLS^W = T3
	jRS^W = T4

tc_allocation	decoding matrix
2	C^W = Ro – T2 – jS^W
	L^W = Lo – C^W + jS^W
	R^W = T2
	jLS^W = T3
	jRS^W = T4

tc_allocation	decoding matrix
3	R^W = Lo + Ro – 2*T2 – T3
	jLS^W = –2*(Lo – T2 – T3) – T4
	C^W = T2
	L^W = T3
	jRS^W = T4

tc_allocation	decoding matrix
4	L^W = Lo + Ro – 2*T2 – T4
	jRS^W = 2*Ro – 2*(T2 +T4) – T3
	C^W = T2
	jLS^W = T3
	R^W = T4

tc_allocation	decoding matrix
5	jLS^W = 0.5*(Ro – Lo + T3 - T4)
	jRS^W = jLS^W
	C^W = T2
	L^W = T3
	R^W = T4

tc_allocation	decoding matrix
6	C^W = 0.5*(Ro + Lo – T2 – T3)
	jLS^W = Ro – Lo – T2 + T3 – T4
	R^W = T2
	L^W = T3
	RS^W = T4

tc_allocation	decoding matrix
7	C^W = 0.5*(Lo + Ro – T2 – T4)
	jRS^W = Ro – Lo + T2 –T3 – T4
	L^W = T2
	jLS^W = T3
	R^W = T4

3/1 configuration, dematrixing procedure equals '00' or '01:

tc_allocation	decoding matrix
0	L^W = Lo – T2 – T3
	R^W = Ro – T2 – T3
	C^W = T2
	S^W = T3

tc_allocation	decoding matrix
1	C^W = Lo – T2 – T3
	R^W = Ro – C^W – T3
	L^W = T2
	S^W = T3

tc_allocation	decoding matrix
2	C^W = Ro – T2 – T3
	L^W = Lo – C^W – T3
	R^W = T2
	S^W = T3

tc_allocation	decoding matrix
3	S^W = Lo – T2 – T3
	R^W = Ro – T2 – S^W
	C^W = T2
	L^W = T3

tc_allocation	decoding matrix
4	S^W = Ro – T2 – T3
	L^W = Lo – T2 – S^W
	C^W = T2
	R^W = T3

3/1 configuration, dematrixing procedure equals '10':

tc_allocation	decoding matrix
0	L^W = Lo – T2 + T3
	R^W = Ro – T2 – T3
	C^W = T2
	jS^W = T3

tc_allocation	decoding matrix
1	C^W = Lo – T2 + T3
	R^W = Ro – C^W – T3
	L^W = T2
	jS^W = T3

tc_allocation	decoding matrix
2	C^W = Ro –T2 – T3
	L^W = Lo – C^W + T3
	R^W = T2
	jS^W = T3

tc_allocation	decoding matrix
3	jS^W = – Lo + T2 + T3
	R^W = Ro –T2 – jS
	C^W = T2
	L^W = T3

tc_allocation	decoding matrix
4	jS^W = Ro –T2 – T3
	L^W = Lo – T2 + jS
	C^W = T2
	R^W = T3

tc_allocation	decoding matrix
5	C^W = 0.5*(Ro + Lo – T2 – T3)
	jS^W = 0.5*(Ro – Lo + T2 – T3)
	L^W = T2
	R^W = T3

3/0 configuration, dematrixing procedure equals '00' or '01':

tc_allocation	decoding matrix
0	$L^W = Lo - T2$
	$R^W = Ro - T2$
	$C^W = T2$

tc_allocation	decoding matrix
1	$C^W = Lo - T2$
	$R^W = Ro - C^W$
	$L^W = T2$

tc_allocation	decoding matrix
2	$C^W = Ro - T2$
	$L^W = Lo - C^W$
	$R^W = T2$

2/2 configuration, dematrixing procedure equals '00' or '01':

tc_allocation	decoding matrix
0	$L^W = Lo - T2$
	$R^W = Ro - T3$
	$LS^W = T2$
	$RS^W = T3$

tc_allocation	decoding matrix
1	$R^W = Ro - T3$
	$LS^W = Lo - T2$
	$L^W = T2$
	$RS^W = T3$

tc_allocation	decoding matrix
2	$L^W = Lo - T2$
	$RS^W = Ro - T3$
	$LS^W = T2$
	$R^W = T3$

tc_allocation	decoding matrix
3	$LS^W = Lo - T2$
	$RS^W = Ro - T3$
	$L^W = T2$
	$R^W = T3$

2/1 configuration, dematrixing procedure equals '00' or '01':

tc_allocation	decoding matrix
0	$L^W = Lo - T2$
	$R^W = Ro - T2$
	$S^W = T2$

tc_allocation	decoding matrix
1	$S^W = Lo - T2$
	$R^W = Ro - S^W$
	$L^W = T2$

tc_allocation	decoding matrix
2	$S^W = Ro - T2$
	$L^W = Lo - S^W$
	$R^W = T2$

2.5.3.2.1.2 Dynamic Crosstalk

If dynamic crosstalk is enabled for a channel for a certain subband group, i.e. dyn_cross[Tx][Sb] is true, the bit allocation for each subband of this subband group, and the coded subband samples are not transmitted. The bit allocation and decoded subband samples shall be copied from the corresponding transmission channel. The 'dyn_cross_mode' field in the bit stream indicates from which channel, and to which channel the subband samples have to be copied. The scalefactor select information and the scalefactors which shall be used for the re-scaling of the subband samples are however contained in the bit stream.

The following rules shall apply for the different configurations:

3/2 configuration

If transmission channel T2 is missing, and the corresponding presentation channel is L, this channel is derived by multiplying the subband samples transmitted for Lo with the scalefactors transmitted in T2. If the missing presentation channel in T2 is C and dyn_cross_LR is '0', this channel is derived by multiplying the subband samples transmitted for Lo with the scalefactors transmitted in T2. If dyn_cross_LR is '1', this channel is derived by multiplying the subband samples transmitted for Ro with the scalefactors transmitted in T2. If the missing presentation channel is R, this channel is derived by multiplying the subband samples transmitted for Ro with the scalefactors transmitted in T2. If transmission channel T3 is missing, which contains either L or LS, the presentation channels L or LS are derived by multiplying the subband samples of Lo with the scalefactors transmitted in T3. If transmission channel T4 is missing, which contains either R or RS, these channels are derived by multiplying the subband samples of Ro with the scalefactors transmitted in T4. A term TS_{ij} in the table means that the subband samples in transmission channel j have to be copied from transmission channel i. The input samples for the synthesis filter of transmission channel T_i are derived by multiplying the subband samples TS_{ij} by scalefactors scf_i. The input samples for the synthesis filter of transmission channel T_j are derived by multiplying the subband samples TS_{ij} by scalefactors scf_j. The rest of the decoding is the same as the situation without dynamic crosstalk.

3/1 configuration

If transmission channel T2 is missing, and the corresponding presentation channel is L, this channel is derived by multiplying the subband samples transmitted for Lo with the scalefactors transmitted in T2. If the missing presentation channel in T2 is C and dyn_cross_LR is '0', this channel is derived by multiplying the subband samples transmitted for Lo with the scalefactors transmitted in T2. If dyn_cross_LR is '1', this channel is derived by multiplying the subband samples transmitted for Ro with the scalefactors transmitted in T2. If the missing presentation channel is R, this channel is derived by multiplying the subband samples transmitted for Ro with the scalefactors transmitted in T2. If transmission channel T3 is missing, and the corresponding presentation channel is S and dyn_cross_LR is '0', this channel is derived by multiplying the subband samples transmitted for Lo with the scalefactors transmitted in T3. If transmission channel T3 is missing, and the corresponding presentation channel is S and dyn_cross_LR is '1', this channel is derived by multiplying the subband samples transmitted for Ro with the scalefactors transmitted in T3.

A term TS_{ij} in the table means that the subband samples in transmission channel j have to be copied from transmission channel i. The input samples for the synthesis filter of transmission channel T_i are derived by multiplying the subband samples TS_{ij} by scalefactors scf_i. The input samples for the synthesis filter of transmission channel T_j are derived by multiplying the subband samples TS_{ij} by scalefactors scf_j. The rest of the decoding is the same as the situation without dynamic crosstalk.

3/0 configuration

If transmission channel T2 is missing, and the corresponding presentation channel is L, this channel is derived by multiplying the subband samples transmitted for Lo with the scalefactors transmitted in T2. If the missing presentation channel in T2 is C and dyn_cross_LR is '0', this channel is derived by multiplying the subband samples transmitted for Lo with the scalefactors transmitted in T2. If dyn_cross_LR is '1', this channel is derived by multiplying the subband samples transmitted for Ro with the scalefactors transmitted in T2.. If the missing presentation channel is R, this channel is derived by multiplying the subband samples transmitted for Ro with the scalefactors transmitted in T2. The rest of the decoding is the same as the situation without dynamic crosstalk.

2/2 configuration

If transmission channel T2 is missing, the presentation channels L or LS, which may be allocated to this transmission channel, are derived by multiplying the subband samples transmitted for Lo with the scalefactors transmitted in T2. If transmission channel T3 is missing, the presentation channels R or RS, which may be allocated to this transmission channel, are derived by multiplying the subband samples transmitted for Ro with the scalefactors transmitted in T3.

A term TS_{ij} in the table means that the subband samples in transmission channel j have to be copied from transmission channel i. The input samples for the synthesis filter of transmission channel T_i are derived by multiplying the subband samples TS_{ij} by scalefactors scf_i. The input samples for the synthesis filter of transmission channel T_j are derived by multiplying the subband samples TS_{ij} by scalefactors scf_j. The rest of the decoding is the same as the situation without dynamic crosstalk.

2/1 configuration

If transmission channel T2 is missing, the presentation channels L, which may be allocated to this transmission channel, is derived by multiplying the subband samples transmitted for Lo with the scalefactors transmitted in T2. If the missing presentation channel is R, this channel is derived by multiplying the subband samples transmitted for Ro with the scalefactors transmitted in T2. If transmission channel T2 is missing, and the corresponding presentation channel is S and dyn_cross_LR is '0', this channel is derived by multiplying the subband samples transmitted for Lo with the scalefactors transmitted in T2. If transmission channel T2 is missing, and the corresponding presentation channel is S and dyn_cross_LR is '1', this channel is derived by multiplying the subband samples transmitted for Ro with the scalefactors transmitted in T2.

2.5.3.2.1.3 MC_Prediction

If the mc_prediction_on bit is set and if the mc_prediction[sbgr] bit is set, the predsi[sbgr][px] bits determine, which predictor is used and how many coefficients pred_coef[sbgr] [px] [pci] are transmitted for each subband group sbgr. If predsi[sbgr][px] is '01', '10' or '11', the delay compensation delay_comp[sbgr][px] and the next 1, 2 or 3 predictor coefficients have to be read from the bit stream. The predictor coefficients are transmitted as 8 bit uimsbf values and have to be dequantised according to the following equation:

pred_coef[sbgr][px][pci] = (pred_coef[sbgr][px][pci] - 127)/32.

If less than three coefficients are transmitted, the remaining pred_coef[sbgr][px] [pci] are set to zero. If predsi[sbgr][px] is '00', all corresponding pred_coef[sbgr][px][pci] are set to zero.

For subband groups with "no dynamic crosstalk" in the 3/2 configuration (dyn_cross_mode[sbgr]=='0000'), the correspondence of the predictor coefficients stored in pred_coef[sbgr][px][pci] to the transmission channels T2, T3 and T4 is as follows (npred=6):

T2: px=0 and px=1, i.e.

pred_coef_T2_0[sbgr][pci] = pred_coef[sbgr][px=0][pci]
pred_coef_T2_1[sbgr][pci] = pred_coef[sbgr][px=1][pci]

T3: px=2 and px=3, i.e.

pred_coef_T3_0[sbgr][pci] = pred_coef[sbgr][px=2][pci]
pred_coef_T3_1[sbgr][pci] = pred_coef[sbgr][px=3][pci]

T4: px=4 and px=5, i.e.

pred_coef_T4_0[sbgr][pci] = pred_coef[sbgr][px=4][pci]
pred_coef_T4_1[sbgr][pci] = pred_coef[sbgr][px=5][pci]

For other configurations and the different dynamic crosstalk modes, the correspondence of the predictor coefficients to the transmission channels has to be adapted to the dynamic crosstalk tables (see 2.5.2.15).

For example:

3/2 configuration, dyn_cross_mode[sbgr]='0010', npred=4

T2: px=0 and px=1, i.e.

pred_coef_T2_0[sbgr][pci] = pred_coef[sbgr][px=0][pci]
pred_coef_T2_1[sbgr][pci] = pred_coef[sbgr][px=1][pci]

T3: not transmitted => no prediction

T4: px=2 and px=3, i.e.

pred_coef_T4_0[sbgr][pci] = pred_coef[sbgr][px=2][pci]
pred_coef_T4_1[sbgr][pci] = pred_coef[sbgr][px=3][pci]

{ end of example }

For each of the up to three signals transmitted in the transmission channels T2, T3 and T4 the prediction signals in each subband group sbgr are calculated as follows:

$$\hat{T}2(n) = \sum_{pci=0}^{2} pred\_coef\_T2\_0[sbgr][pci] * T0(n - delay\_comp - pci) + \sum_{pci=0}^{2} pred\_coef\_T2\_1[sbgr][pci] * T1(n - delay\_comp - pci)$$

$$\hat{T}3(n) = \sum_{pci=0}^{2} pred\_coef\_T3\_0[sbgr][pci] * T0(n - delay\_comp - pci) + \sum_{pci=0}^{2} pred\_coef\_T3\_1[sbgr][pci] * T1(n - delay\_comp - pci)$$

$$\hat{T}4(n) = \sum_{pci=0}^{2} pred\_coef\_T4\_0[sbgr][pci] * T0(n - delay\_comp - pci) + \sum_{pci=0}^{2} pred\_coef\_T4\_1[sbgr][pci] * T1(n - delay\_comp - pci)$$

where T0(n) and T1(n) refer to the subband samples of T0 and T1 after requantisation and application of the scalefactors.

By adding the transmitted prediction error signals to the prediction signals, the signals in the subband group sbgr are reconstructed using the corresponding three, two or one of the following equations:

$$T2(n) = \hat{T}2(n) + \varepsilon_{T2}(n)$$

$$T3(n) = \hat{T}3(n) + \varepsilon_{T3}(n)$$

$$T4(n) = \hat{T}4(n) + \varepsilon_{T4}(n)$$

In those cases of the dynamic crosstalk modes, where combined signals - indicated by Txy or Txyz - are transmitted in one of the transmission channels T2, T3 or T4, no prediction can be applied.

2.5.3.2.2 Requantisation Procedure

See ISO/IEC 11172-3, 2.4.3.1 and 2.4.3.3.4.

2.5.3.2.3 Decoding of Scalefactors

See ISO/IEC 11172-3, 2.4.3.3.3.

2.5.3.2.4 Decoding of Low Frequency Enhancement Channel

The low frequency enhancement channel is transmitted as block companded linear PCM coded samples, at a sampling frequency that is 96 times lower than the sampling frequency of the other channels. The requantisation of the transmitted samples and application of the scalefactors are as in ISO/IEC 11172-3 for Layer I (no grouping). See also ISO/IEC 11172-3, 2.4.3.2.1. As the bandwidth of the LFE channel is specified to be 125 Hz, it is recommended to apply a 125 Hz low-pass filter before reproduction, in order to minimise reproduction of coding noise outside the LFE frequency band.

2.5.3.2.5 De-normalisation procedure

In the decoder, the weighted signals L^W, C^W, R^W, LS^W, RS^W first have to be inverse weighted by multiplying the signals with the inverse weighting factors. Next, these signals can be multiplied by the de-normalisation factor to undo the attenuation, done at the encoder site to avoid overload when calculating the compatible signals.

dematrix_procedure	signals	inverse weighting factor	de-normalisation factor
'00', '10'	L^W, R^W	1	$1 + \sqrt{2}$
	C^W, LS^W, RS^W	$\sqrt{2}$	
'01'	L^W, R^W	1	$1{,}5 + 0{,}5 * \sqrt{2}$
	LS^W, RS^W	2	
	C^W	$\sqrt{2}$	
'11'	L^W, R^W, C^W, LS^W, RS^W	1	1

2.5.3.2.6 Synthesis Subband Filter

See ISO/IEC 11172-3, 2.4.3.2.2

2.5.3.3 Layer III Decoding

2.5.3.3.1 Layer III Segment Lists

The segment list syntax allows flexible joint stereo coding of multichannel signals while using only a few bits in the minimum case. The main idea is to construct each output audio channel from a pool of spectral data in the transmitted channels (TCs). This may vary for different parts of the channel spectrum (segments). For each segment, the length and the number of the source TC are transmitted (seg_length, in units of scalefactorband_groups and tc_select, respectively). For Layer III, the following TC numbers are defined:

TC #	Channel	Symbolic
0	Lo	left_comp_chan
1	Ro	right_comp_chan
2	L	left_chan
3	R	right_chan
4	C	centre_chan
5	LS / S	left_surr_chan, mono_surr_chan
6	RS	right_surr_chan
7	„dematrixing“	-

In case a second stereo programme is transmitted (surround=='11') TCs 5 and 6 are used for the left and right channel respectively. If dematrix_procedure '11' is used (no matrixing), the left and right channel signals are transmitted in TCs 0 and 1 respectively instead of TCs 2 and 3.

For each of the TCs there exists a similar data structure as for MPEG-1 Layer III audio channels, i.e. side information and Huffman coded spectral values. The tc_present flags are used to indicate which TCs are transmitted, i.e. how many sets of side information and main information are contained in the mc_audio bit stream. In case of MPEG-2, the amount of side information for each channel is variable. Apart from this difference, decoding of the Huffman coded values works just as in an MPEG-1 decoder.

Each segment of an audio output channel, ch, has a default mapping to the corresponding TC (tc_select == ch), but is assigned to a different TC for composite coding. In this case, an attenuation value is transmitted and applied to the TC spectral data to recover the audio output channel spectral data. In the special case of tc_select==7, the respective segments are reconstructed by dematrixing.

For several segment list types, shortcuts have been defined:

- seglist_present == 0 indicates a segment list in which the data of all covered scalefactorband_groups are reconstructed by dematrixing.
  (i.e. maximum segment length, tc_select=7)
- seglist_nodef == 0 indicates a simple "default" segment list in which the data of all covered scalefactorband_groups are transmitted within the corresponding TC.
  (i.e. maximum segment length, tc_select=ch)
- seglist_repeat == 1 indicates that for granule 1 the same segment list is used as it has been transmitted for granule 0.

Segment lists can be either valid for only one of the granules or, as denoted by segment_list_repeat, be valid for both granules within one frame. A seg_length of zero indicates that the segment list terminates and that the remaining part of the channel spectrum is set to zero.

For frequencies above the scalefactorband_group border (denoted by dematrix_length), the segment lists are used to denote channels which may be compositely coded. For scalefactorband_groups lower than dematrix_length, a less flexible method of assigning the actual transmitted channels is used which does not allow for composite coding.

dematrix_select is a 3..4 bit value with 14 possible assignments (for 3/2 configuration). It is used to find which channels have to be dematrixed and which are transmitted. Two, one or even no channels are reconstructed by means of dematrixing. While segment lists are transmitted for each granule, dematrix_select is valid for both granules.

2.5.3.3.2 Decoding Process for Layer III

If an extension bit stream is available, its access units may contain parts of mc_composite_status_info and mc_audio_data. Their contents are concatenated to the mc_composite_status_info and/or mc_audio_data in the main data part of the MPEG-1 compatible bit stream. The target of the mc_data_begin pointer is calculated in the buffer containing the concatenated bit stream. The structure of a Layer III multichannel /multilingual bit stream can be found in Annex A.3. A possible ext_data (indicated by the ext_bit_stream_present flag in mc_header) must be inserted between mpeg2_main_data and mpeg1_ancillary data.

The decoding process consists of the following steps:

- **Expansion of Default Segment List Types**
  This is done by evaluating seg_list_present, seg_list_nodef and seg_list_repeat. If these syntax elements indicate that a shortcut is used, then a full segment_list representation is expanded according to the shortcut definitions stated in 2.5.3.3.1.

- **Construction of Decoding Maps**
  Construct a map data_present[gr][tc][sfb] describing which spectral TC data (dependent on granule, transmitted channel and scalefactorband) are actually transmitted. This is done by determining the scalefactorbands which are referenced by dematrix_select or the segment_lists (as part of a scalefactorband_group).
  In addition, construct a map js_carrier[gr][tc][sbgr] describing which spectral TC data (dependent on granule, transmitted channel and scalefactorband_group) are used as a carrier for joint stereo transmission. This is done for each audio channel ch by determining the scalefactorband_groups which are referenced with a tc_select != ch.

- **Decoding of TC Information**
  Requantise the TC data of all channels as indicated by tc_present. This works just as in a Layer III MPEG-1 decoder using information in the elements: block_type, scalefac_l, scalefac_s, scfsi, part2_3_length, big_values, global_gain, scalefac_compress, table_select, subblock_gain, region0_count, region1_count, preflag, scalefac_scale, count1table_select. The requantisation operation is described in ISO/IEC 11172-3, 2.4.3.4. The decoded data is the raw spectral information of the respective audio output channel, where all coefficients belonging to scalefactorbands with data_present[gr][tc][sfb] == 0 have been left out.

- **Decoding of MultiChannel Prediction**
  The decoding of multi channel prediction is done similar to Layers I and II independently for each scalefactorband_group sbgr. If mc_prediction_on is off, no decoding of prediction is required for any scalefactorband_group. If the mc_prediction_sbgr[sbgr] flag is off, no prediction is used in the respective scalefactorband_group and no further predictor information is transmitted. Prediction information is transmitted once for each frame and applies to both granules.

  <u>Calculation of possible prediction combinations and number of predictor coefficients:</u>

  For each scalefactorband_group sbgr the possible prediction combinations are calculated according to the following rules:

  - Each channel can be a possible destination channel for multi channel prediction if (1) data is transmitted for one of the granules (data_present[gr_0][ch][sfb(sbgr)]!=0 || data_present[gr_1][ch][sfb(sbgr)]!=0) and (2) source and destination channel have the same block_type.

  - For each possible destination channel one or two source channels (and predictor coefficients) are possible:

Destination channel	Number of source channels	Source channel(s)
L	1	Lo
R	1	Ro
C, S	2	Lo, Ro
LS	1	Lo
RS	1	Ro

  In case of joint stereo coding (js_carrier[gr][ch][sbgr] != 0), both source channels Lo and Ro are regarded as possible source channels. The value npredcoef denotes the total number of possible predictor coefficients in one scalefactorband_group. For short blocks (block_type == '10'), npredcoef is defined as zero for scalefactorband_groups above 11 (i.e. above the number of defined scalefactorband_groups).

- For each possible coefficient, one bit in the predictor select information predsi[sbgr][ ] is transmitted. The bits for the possible coefficients are ordered according to the destination channel using the standard channel assignment order, i.e. L, R, C, LS, RS. If two source channels are possible for the destination channel, the first bit corresponds to the Lo and the second to the Ro source channel.

- If predsi[sbgr][pci] is '0', the corresponding coefficient pred_coef[sbgr][pci] is set to 0. Otherwise a coefficient is transmitted. The ordering of the coefficients is the same as for the predsi information, i.e. the coefficients are ordered according to the destination channel (coarse ordering) and to the source channel (fine ordering). The coefficients are requantised according to the following table:

Transmitted value	Requantised value
0	-0.61199
1	-0.24565
2	0.24565
3	0.61199
4	1.15831
5	1.97304
6	3.18805
7	5

<u>Calculation of prediction signals:</u>

In each of the referenced destination channels, the prediction signals are calculated and added to the transmitted prediction error signals by:

```
L  += pred_coef_L[sbgr]  * Lo
R  += pred_coef_R[sbgr]  * Ro
C  += pred_coef_C1[sbgr] * Lo  + pred_coef_C2[sbgr] * Ro
LS += pred_coef_LS[sbgr] * Lo
RS += pred_coef_RS[sbgr] * Ro
```

and for the case of joint stereo coding

```
JS += pred_coef_JS1[sbgr] * Lo + pred_coef_JS2[sbgr] * Ro
```

The addition of predicted signals is performed only for granules in which data is transmitted for the respective channels (data_present[gr][ch][sbgr]!=0).

- **Decoding of Channel Data**
  Each output audio channel is assembled from the decoded TC data according to its segment list and dematrix_select configuration. All scalefactorband_groups that are reconstructed by dematrixing are to be omitted. The data_present map is used to direct the coded spectral values from the TC data to the correct scalefactorband_group positions in the spectrum buffer of the destination channels.

  For composite coded segments (i.e. tc_select != ch && tc_select != 7), a scaling operation is applied to the spectral data using the transmitted attenuation values as follows:

  - Determine the basic attenuation factor a_0 ($1/\sqrt{2}$ for attenuation_scale==1, else $1/\sqrt{\sqrt{2}}$)
  - Apply scaling using the actual attenuation factor a:

$$a = \begin{cases} a_0^{attenuation} & \text{for } attenuation < 0.75 \cdot max\_attenuation \\ a_0^{attenuation-max\_attenuation} & \text{for } attenuation \geq 0.75 \cdot max\_attenuation \end{cases}$$

  with $max\_attenuation = 2^{attenuation\_range+2}$

- **Dematrixing**
  Dematrixing is used to reconstruct the missing scalefactorband_groups (only for dematrix_procedure !='11', not for second stereo programme, i.e. surround == '11').
  For the first dematrix_length number of scalefactorband_groups, the dematrixed parts are determined by the transmitted dematrix_select values for the whole frame. Above this border, they are defined by the segment

list segments with tc_select==7. Dematrixing is done by recovering zero, one or two channels from the downmix equations for the 3/2 stereo configuration:

Lo = α*(L+β*C+γ*LS) and Ro = α*(R+β*C+γ*RS)

or, in the case of 3/1 stereo configuration:

Lo = α*(L+β*C+γ*S) and Ro = α*(R+β*C+γ*S)

where α is an overall attenuation for all channels and β and γ are the attenuation factors of the centre and surround signals. For other stereo configurations, the downmixing equations can be derived from one of these by regarding the absent audio channels as zero. In the case of dematrix_procedure == '10', the dematrixing equations are modified as is described in 2.5.3.2.1.

The attenuation factor values are specified for each dematrixing procedure:

dematrix_procedure	α	β	γ
'00'	1/(1 + √2)	1/√2	1/√2
'01'	1/(1,5 + 0,5*√2)	1/√2	0,5
'10'	1/(1 + √2)	1/√2	1/√2

<u>Phantom coded centre channel:</u>
In the case of Phantom coding of the centre channel (centre=='11'), the appearance of coding noise in a dematrixed centre channel is suppressed by limiting the bandwidth of the dematrixed centre channel as shown:

Sampling frequency [Hz]	Number of valid lines in centre channel
48000	230
44100	238
32000	296

This step is carried out *before* a second channel is dematrixed.

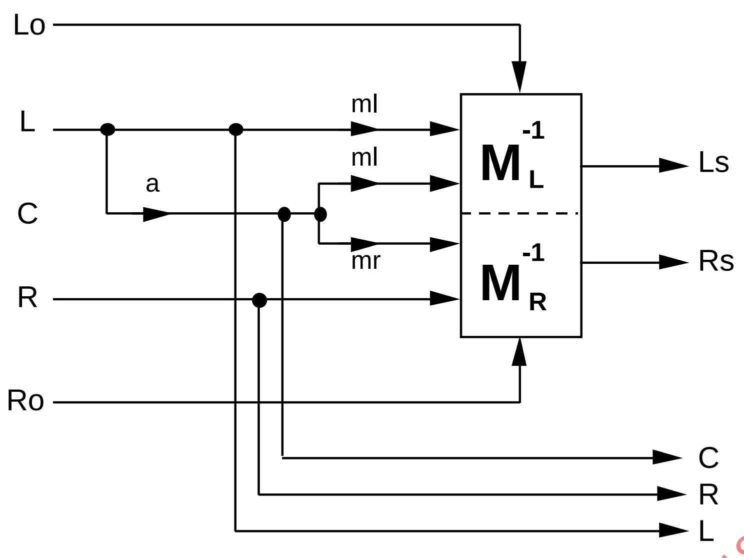
<u>Correction of joint stereo dematrixing:</u>
If the matrix_attenuation_present flag is on, the standard procedure for channel dematrixing is modified. For the dematrixing operation, all joint stereo coded scalefactorband_group data is previously scaled by an attenuation factor. This scaling is performed independently for both halves of the dematrixing equations involving Lo and Ro.

The scaling factors ml and mr are determined by the transmitted matrix_attenuation values:

$$ml = 2^{-0,25 \cdot matrix\_attenuation\_l[js\_ch][sbgr]}$$
$$mr = 2^{-0,25 \cdot matrix\_attenuation\_r[js\_ch][sbgr]}$$

Here, js_ch denotes the TC where the actual spectral data for the joint stereo coded signals has been transmitted and sbgr denotes the scalefactorband_group index.

This procedure is illustrated below for the case of joint stereo coding of L and C. The spectral data is transmitted in the TC of the L channel (i.e. TC 2). Thus, C is constructed from the same data using the corresponding attenuation value. Prior to dematrixing, L and C are scaled by the factors ml and mr. This scaling operation is *not* applied to the final output channel data.



- **Synthesis Filterbank**
  Apply the synthesis filterbank (see ISO/IEC 11172-3, 2.4.3.4.10).

2.5.3.3.3 Decoding of LFE for Layer III

The LFE values are decoded from a simplified Layer III-type bit stream.

- The decoding of the Huffman coded values is done using the Huffman code table indicated by lfe_table_select.
- The decoding of transmitted Huffman codes is continued until all bits indicated by lfe_hc_len have been exhausted. After this process, the value of lfe_bigval is known. For clarity, this parameter is introduced to indicate the number of Huffman code words used to transmit the spectral data of the low frequency channel. The decoded components, x and y, are interpreted as values of the respective spectral coefficients for granules 0 and 1.
- Subsequently, the requantisation is performed in a manner similar to the requantisation of TC data. For this purpose, lfe_gain is used and a scalefactor and subblock gain of zero is assumed.
- As a synthesis filterbank for the low frequency enhancement channel, the inverse MDCT (IMDCT) for the reconstruction of data in short blocks (block_type == '10') is used that is described as part of the synthesis hybrid filterbank in ISO/IEC 11172-3, 2.4.3.4.10. Thus, the window type which is described under ISO/IEC 11172-3, 2.4.3.4.10 / Windowing (d) is applied to the 12 IMDCT output samples of each granule. Because there is only one window per granule, the "overlap add" process simplifies to:

$$\text{result}_i = y_i + s_i \qquad \text{for i=0 to 5}$$
$$s_i = y_{i+6} \qquad \text{for i=0 to 5}$$

2.5.3.3.4 Decoding of ML Data for Layer III

If multilingual_fs==0, see ISO/IEC 11172-3, 2.4.3.4

If multilingual_fs==1, see 2.4.3.2 of this part of ISO/IEC 13818.

For use as ML main data, nch is set to no_of_multi_lingual_ch.

2.6 Registration of Copyright Identifiers

2.6.1 General

Parts 1, 2, and 3 of ISO/IEC 13818 provide support for the management of audio-visual works copyrighting. In ITU-T Recommendation H.222.0|ISO/IEC 13818-1 this is by means of a copyright descriptor, while ITU-T Recommendation H.262|ISO/IEC 13818-2 and this part of ISO/IEC 13818 contain fields for identifying copyright holders through syntax fields in the elementary stream syntax. Subclause 2.6.2 presents the method of obtaining and registering copyright identifiers in this part of ISO/IEC 13818.

This part of ISO/IEC 13818 specifies a 72-bit copyright identification field, formed by 72 consecutive copyright_identification_bit's (2.5.2.13). The copyright identification field consists of an 8-bit copyright_identifier followed by a 64-bit copyright_number. The unique 8 bit copyright_identifier is a work type code identifier (such as ISBN, ISSN, ISRC, etc...) carried in the copyright descriptor. The copyright_identifier enables identification of a wide number of Copyright Registration Authorities. Each Copyright Registration Authority may specify a syntax and semantic for identifying the audio-visual works or other copyrighted works within that particular copyright organisation through appropriate use of the 64-bit copyright_number.

In 2.6.2 and annexes G,H and I, the benefits and responsibilities of all parties to the registration of copyright_identifier are outlined.

2.6.2 Implementation of a Registration Authority (RA)

ISO/IEC JTC 1 shall call for nominations for an international organisation which will serve as the Registration Authority for the **copyright_identifier** as defined in this part of ISO/IEC 13818. The selected organisation shall serve as the Registration Authority. The so-named Registration Authority shall execute its duties in compliance with Annex H of the JTC 1 directives. The registered copyright_identifier is hereafter referred to as the Registered Identifier (RID).

Upon selection of the Registration Authority, JTC 1 shall require the creation of a Registration Management Group (RMG) which will review appeals filed by organisations whose request for a RID to be used in conjunction with this part of ISO/IEC 13818 has been denied by the Registration Authority.

Annexes G, H and I provide information on the procedures for registering a unique copyright identifier.

Annex A

(normative)

Diagrams

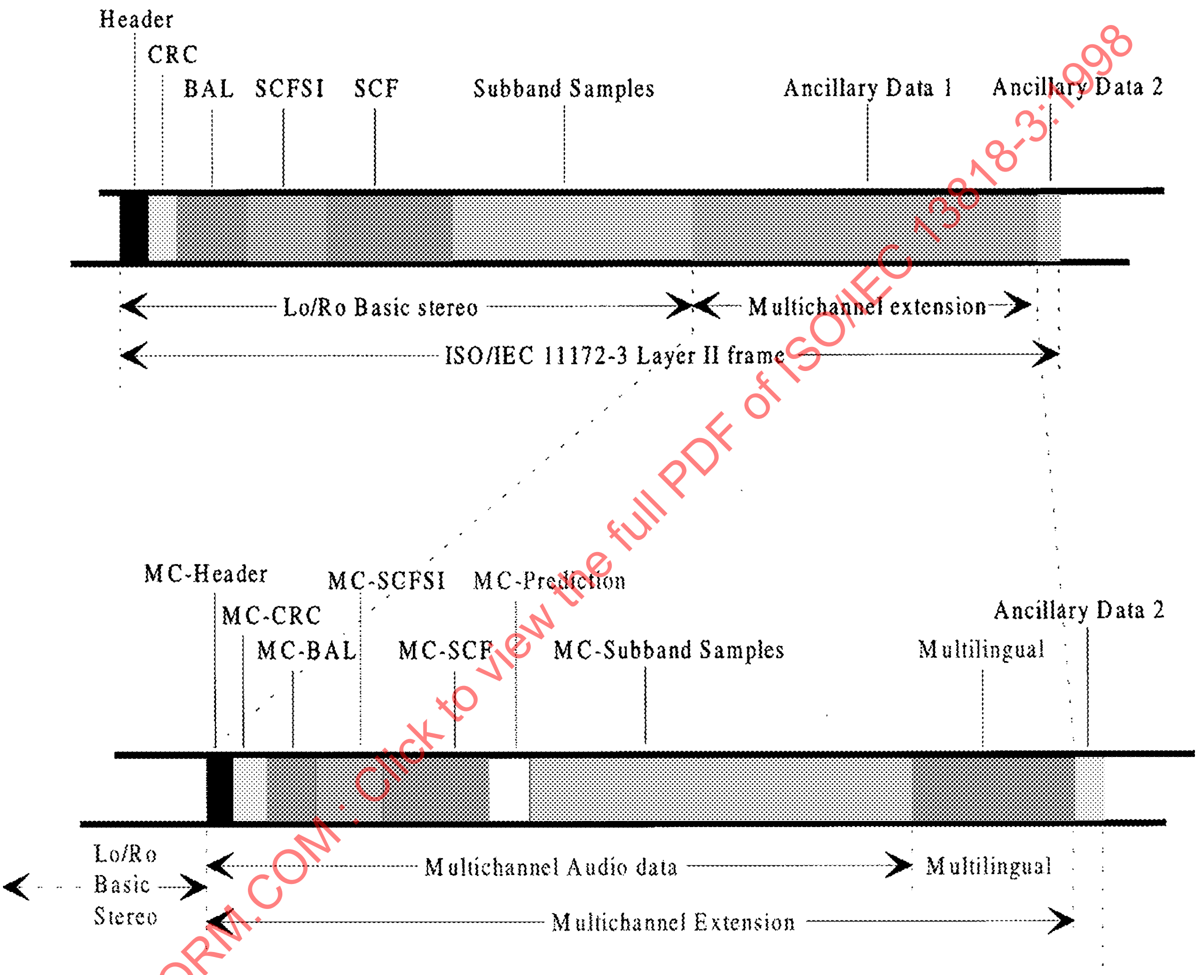


Figure A.1 — Structure of the ISO/IEC 13818-3 Layer II multichannel extension, backwards compatible with ISO 11172-3 Layer II

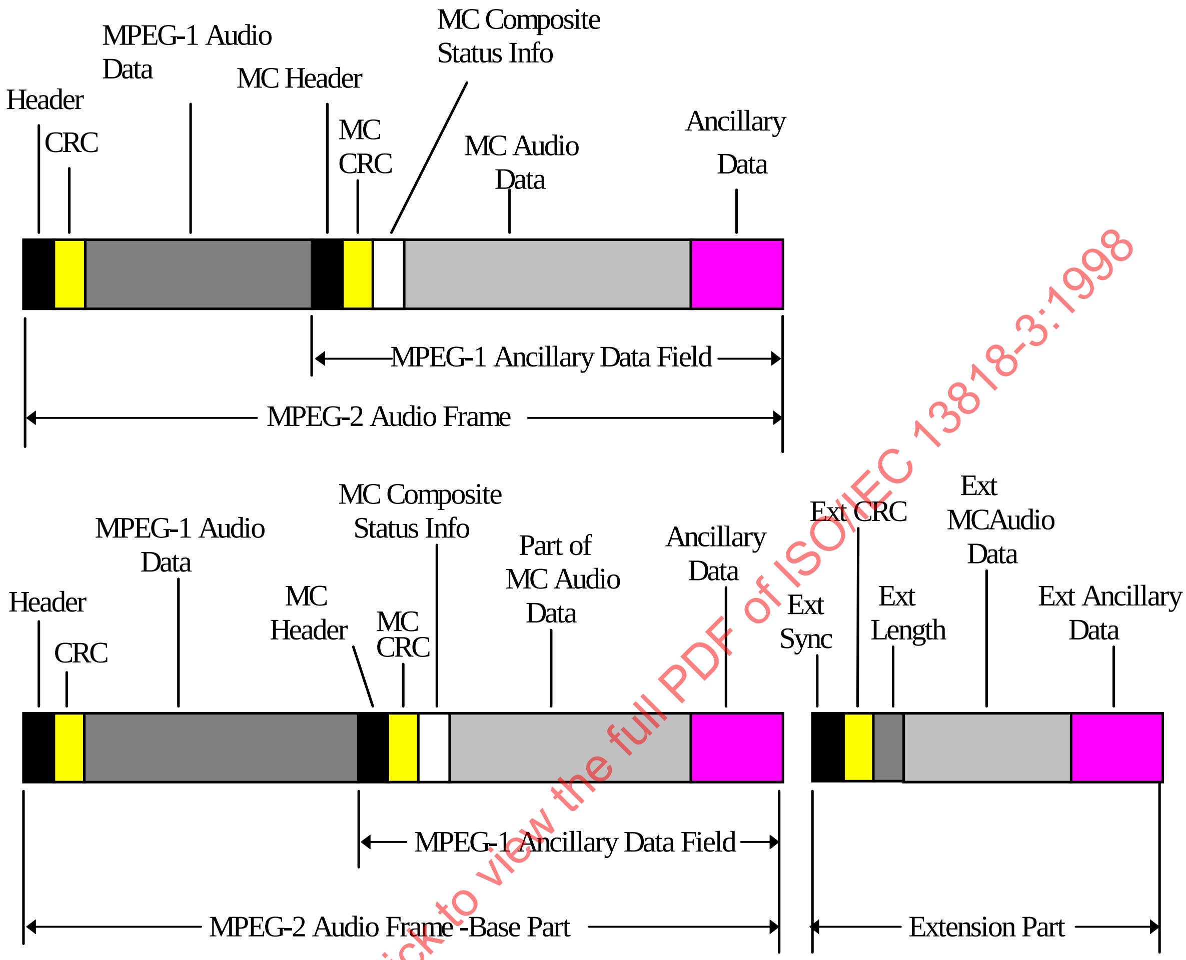
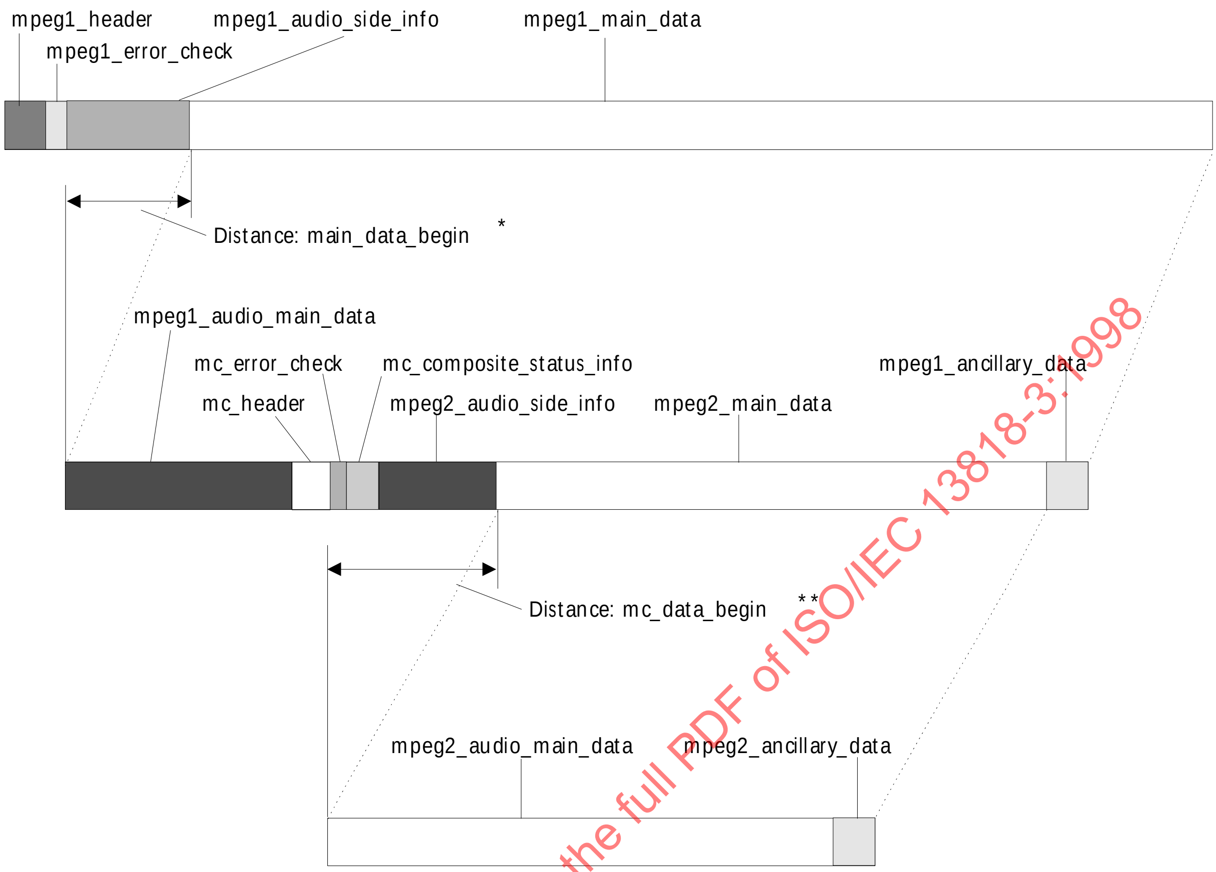


Figure A.2 — An example of a typical structure of the ISO/IEC 13818-3 Layer II multichannel extension, using the (ISO/IEC 11172-3 compatible) base bit stream, as well as the extension bit stream



* not counted are: mpeg1_header, mpeg1_error_check, mpeg1_audio_side_info

** not counted are: mpeg1_header, mpeg1_error_check, mpeg1_audio_side_info, mpeg1_audio_main_data, mc_header, mc_error_check, mc_composite_status_info, mc_audio_side_info, mpeg1_ancillary_data

Figure A.3 — An example of a typical structure of the ISO/IEC 13818-3 Layer III multichannel extension. A possible ext_data must be inserted between mpeg2_main_data and mpeg1_ancillary data.

Annex B

(normative)

Tables

Table B.1. Possible quantisation per subband, Layer II

Sampling frequencies 16; 22,05; 24 kHz.

index

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

sblimit = 30

Sum of nbal = 75

Table B.2. Layer III scalefactor bands

These tables list the width of each scalefactor band. There are 22 scalefactor bands for long windows (type 0, 1 or 3) and 13 scalefactor bands for short windows at each sampling frequency. Since the scalefactor for the last scalefactor band is set to a fixed value and is not transmitted, the number of scalefactors is 21 for long and 12 for short windows.

16 kHz sampling rate, long blocks, number of lines 576

scalefactor band	width of band	index of start	index of end
0	6	0	5
1	6	6	11
2	6	12	17
3	6	18	23
4	6	24	29
5	6	30	35
6	8	36	43
7	10	44	53
8	12	54	65
9	14	66	79
10	16	80	95
11	20	96	115
12	24	116	139
13	28	140	167
14	32	168	199
15	38	200	237
16	46	238	283
17	52	284	335
18	60	336	395
19	68	396	463
20	58	464	521
21	54	522	575

16 kHz sampling rate, short blocks, number of lines 192

scalefactor band	width of band	index of start	index of end
0	4	0	3
1	4	4	7
2	4	8	11
3	6	12	17
4	8	18	25
5	10	26	35
6	12	36	47
7	14	48	61
8	18	62	79
9	24	80	103
10	30	104	133
11	40	134	173
12	18	174	191

22,05 kHz sampling rate, long blocks, number of lines 576

scalefactor band	width of band	index of start	index of end
0	6	0	5
1	6	6	11
2	6	12	17
3	6	18	23
4	6	24	29
5	6	30	35
6	8	36	43
7	10	44	53
8	12	54	65
9	14	66	79
10	16	80	95
11	20	96	115
12	24	116	139
13	28	140	167
14	32	168	199
15	38	200	237
16	46	238	283
17	52	284	335
18	60	336	395
19	68	396	463
20	58	464	521
21	54	522	575

22,05 kHz sampling rate, short blocks, number of lines 192

scalefactor band	width of band	index of start	index of end
0	4	0	3
1	4	4	7
2	4	8	11
3	6	12	17
4	6	18	23
5	8	24	31
6	10	32	41
7	14	42	55
8	18	56	73
9	26	74	99
10	32	100	131
11	42	132	173
12	18	174	191

24 kHz sampling rate, long blocks, number of lines 576

scalefactor band	width of band	index of start	index of end
0	6	0	5
1	6	6	11
2	6	12	17
3	6	18	23
4	6	24	29
5	6	30	35
6	8	36	43
7	10	44	53
8	12	54	65
9	14	66	79
10	16	80	95
11	18	96	113
12	22	114	135
13	26	136	161
14	32	162	193
15	38	194	231
16	46	232	277
17	54	278	331
18	62	332	393
19	70	394	463
20	76	464	539
21	36	540	575

24 kHz sampling rate, short blocks, number of lines 192

scalefactor band	width of band	index of start	index of end
0	4	0	4
1	4	4	8
2	4	8	12
3	6	12	18
4	8	18	26
5	10	26	36
6	12	36	48
7	14	48	62
8	18	62	80
9	24	80	104
10	32	104	136
11	44	136	180
12	12	180	192

Annex C

(informative)

The encoding process

C.1 Extension to lower sampling frequencies

In this part of the annex, the differences from ISO/IEC 11172-3 encoders are described.

C.1.1 Lower sampling frequencies, Layer I

The only differences from the encoder described in ISO/IEC 11172-3 are the formatting and the psychoacoustic model. The encoded subband information is transferred in frames, consisting of slots. In Layer I, a slot consists of 32 bits. The number of slots in a frame depends on the sampling frequency and the bit rate. Each audio frame contains information on 384 samples for each channel of the original input signal.

Fs (kHz)	Frame size (ms)
24 22,05 16	16 17,415.. 24

The number of slots in a frame can be calculated by the formula:

Number of slots per frame (N) = bitrate * 12 / Fs

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. The same padding procedure, as described in ISO/IEC 11172-3, 2.4.2.3 has to be applied to determine when the padding bit has to be added.

The psychoacoustic model requires modification for the lower sampling frequency. See Annex D.1.

C.1.2 Lower sampling frequencies, Layer II

The differences from the encoder described in ISO/IEC 11172-3 are the formatting, the possible quantisations, and the psychoacoustic model. The encoded subband information is transferred in frames, consisting of slots. In Layer II, a slot consists of 8 bits. The number of slots in a frame depends on the sampling frequency and the bit rate. Each audio frame contains information on 1152 samples for each channel of the original input signal.

Fs (kHz)	Frame size (ms)
24 22,05 16	48 52,245.. 72

The number of slots in a frame can be calculated by the formula:

Number of slots per frame (N) = bitrate * 144 / Fs

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. The same padding procedure, as described in ISO/IEC 11172-3, 2.4.2.3 has to be applied to determine when the padding bit has to be added.

Instead of the tables B.2. "Layer II bit allocation tables, Possible Quantisation per Subband" of ISO/IEC 11172-3, the table B.1 "Possible quantisation per subband, Layer II" of this part of ISO/IEC 13818 shall be used.

The psychoacoustic model requires modification for the lower sampling frequency. See Annex D.1

C.1.3 Lower sampling frequencies, Layer III

The differences from the encoding process described in ISO/IEC 11172-3 Layer III are the changed scalefactor band tables, the omission of some side information due to the changed frame layout, and some changed tables in the psychoacoustic model. All the basic steps described in ISO/IEC 11172-3 apply, with the exception of the calculation of the scalefactor select information.

C.2 Multichannel extension

In this part of the annex, two examples of suitable multichannel encoders are described, one for layers I and II, and one for Layer III. The examples are valid for a 5^{+1} channel configuration, (i.e. Left, Centre, Right, Left Surround, Right Surround, and a low frequency enhancement channel), and for the multilingual extension with the same layer as for the multichannel extension.

C.2.1 Multichannel extension Layer I, II

C.2.1.1 The filterbank

The filterbanks used are the same as in ISO/IEC 11172-3, i.e. a 32-band polyphase filterbank for all layers, followed by an MDCT on the subband signals for Layer III only. The subband filter has to be applied to all five channels.

C.2.1.2 Calculation of scalefactors

The calculation of scalefactors and, for Layer II, scalefactor select info, is done in exactly the same way as in ISO/IEC 11172-3.

C.2.1.3 Psychoacoustic models

The two psychoacoustic models as described in ISO/IEC 11172-3 are applied here as well. For all five channels, the signal-to-mask ratios of all subbands are calculated.

C.2.1.4 Predistortion

Predistortion (or prequantisation) is used to prevent unmasked and unexpected noise from audio channels when dematrixing is performed in the decoder. This noise can appear because dematrixing in the decoder is performed with different multichannel extension signals than used for the matrixing process at the encoder. Only quantised samples are available in the decoder. The audible artefacts can be avoided by prequantisation of these samples in the encoder, prior to the matrixing. The following procedure can be used.

For each subband group:

step 1: transmission channel switching procedure; choice of the multichannel extension signals T2, T3, T4 and associated tc_allocation;

If tc_allocation[sbgr] equals 1 or 7:

- step 2: coding and decoding of T2 and T3 according to the masking thresholds calculation,
- step 3: matrixing using the predistorted versions of T2 and T3 in order to obtain Lo,
- step 4: calculation of the predistorted centre signal as it will be derived at the decoder side after encoding and decoding of Lo,
- step 5: matrixing using the predistorted centre and the predistorted version of T4 to obtain Ro.

If tc_allocation[sbgr] equals 2 or 6:

- step 2: coding and decoding of T2 and T4 according to the masking thresholds calculation, step 3: matrixing using the predistorted versions of T2 and T4 in order to obtain Ro,

step 4: calculation of the predistorted centre signal as it will be derived at the decoder side after encoding and decoding of Ro,

step 5: matrixing using the predistorted centre and the predistorted version of T3 to obtain Lo.

If tc_allocation[sbgr] equals 0,3,4 or 5

step 2: coding and decoding of T2, T3, T4 according to the masking thresholds calculation,

step 3: matrixing using the predistorted versions of T2, T3, T4 in order to get the compatible pair (Lo, Ro).

If the centre signal is dominant in a certain subband group, it is recommended to use only tc_allocations that do not contain the centre signal in one of the additional transmission channels.

C.2.1.5 Matrixing

First, all signals have to be attenuated to avoid overload when calculating the compatible stereo signal. The attenuation factor depends on the chosen matrix procedure.

Procedure 0, 2:	$1/(1 + \sqrt{2})$
Procedure 1:	$1/(1,5 + 0,5*\sqrt{2})$
Procedure 3:	1

Next, the Centre, Left Surround and Right Surround signals have to be attenuated before the compatible stereo signal is calculated. These attenuation factors are

Procedure 0, 2:	C, LS, RS	$1/\sqrt{2}$
Procedure 1:	C	$1/\sqrt{2}$
	LS, RS	0,5
Procedure 3:	C, LS, RS	1

The signals after this attenuation are called C^W,LS^W,RS^W.

Next, the compatible signal has to be calculated according to

Procedure 0, 1:

$$Lo = L^W + C^W + LS^W$$
$$Ro = R^W + C^W + RS^W$$

Procedure 2:

$$Lo = L^W + C^W - jS^W$$
$$Ro = R^W + C^W + jS^W$$

The signals to be transmitted in T3 and T4 are derived from LS^W and RS^W which optionally may be processed by dynamic range compression and 90 degrees phase shifting. jS^W is derived from jLS^W and jRS^W by calculation of the mono component (jLS^W and jRS^W)/2.

Procedure 3:

$$Lo = L^W$$
$$Ro = R^W$$

C.2.1.6 Dynamic transmission channel switching

To avoid audible artefacts due to the dematrixing process, it is necessary to choose the correct transmission channel allocation. This applies for matrix procedures 0, 1, and 2. A simple, but effective approach is to choose for the transmission channels T2, T3, T4 those channels that have the lowest scalefactors in the subband group under consideration. In subband groups that consist of more than one subband, first the maximum of the scalefactors of all subbands in a subband group has to be determined for each of the signals. Then the three signals with the lowest of the maximum scalefactor (highest scalefactor index) are allocated to the transmission channels T2, T3, and T4. If the transmission channel allocation is the same or almost the same for all subband groups, the tc_sbgr_select bit can be set to '0', in which case only one tc_allocation has to be sent for all subband groups.

C.2.1.7 Dynamic Crosstalk

According to a binaural model of the human ear, those components of two- or multichannel stereophonic signals can be determined to a large extend, which are irrelevant with respect to the spatial perception of the stereophonic presentation. The stereo-irrelevant signal components are not masked, however, on the other hand, they do not contribute to the localisation of sound sources. Therefore, not all transmission channels, in particular those containing stereo-irrelevant components, have to be transmitted during all the time. In such a case, any channel of a multichannel stereo signal (L, C, R, LS or RS) may be substituted by any other channel. This may happen either in subband groups, whereby up to 12 those groups are available, or even for the whole audio channel. On the decoding side, this channel, or part of the channel, has to be reproduced via any presentation channel, or via several presentation channels, without effecting the stereophonic impression.

The dynamic crosstalk method used in Layer I and II is based on the concept of intensity stereo coding, described in Annex G of ISO/IEC 11172-3, but allows much more flexibility between the different channels and gives a much higher resolution in terms of frequency bands. Dynamic crosstalk can be used to increase the audio quality at a given bitrate, and/or reduces the bitrate for multichannel audio signals for a certain level of quality. This method requires negligible additional decoder complexity, and does not affect the encoder and decoder delay.

Dynamic crosstalk is based on known psychoacoustic effects. On one side, this method uses, like the intensity stereo coding, the effect, that at high frequencies the localisation is mainly based on the temporal envelope and not by the temporal fine structure of the audio signal. On the other side, dynamic crosstalk is based on the fact, that only fast changes in the temporal envelope of the audio signal are important for the localisation. However, the more stationary parts, in particular after attacks, have a much weaker effect on localisation. This means that for certain time intervals in certain regions of the spectrum, crosstalk is permissible. Those signals have to be identified by means of a signal analysis in the encoder. which can be set to "mono" and transmitted in only one channel. The signals can be identified on the basis of subband groups. Up to three transmission channels of the multichannel extension part can be substituted.

Only the corresponding scalefactors and scfsi, but no bit allocation and subband samples are transmitted for those channels which will be substituted in the decoder by dynamic crosstalk. As a result, the so-called "Gestalt" information of the image is completely available in the basic channels Lo/Ro, and only the relevant stereophonic information is transmitted in the extension channels.

A Txy in the dynamic crosstalk tables in 2.5.2.15 means, that the subband samples of the representation channels indicated in the tc-allocation table (also given in 2.5.2.15) are added, as described in Annex G of ISO/IEC 11172-3. The bit allocation and subband samples are transmitted in the transmission channel Tx. The scalefactors and scfsi of the representation channels, corresponding to Tx and Ty, have to be transmitted in the transmission channels Tx and Ty. This allows a transmission of the level control information for both channels to reproduce the temporal slope of both representation channels corresponding to Tx and Ty. The entries in the dynamic crosstalk table allow a very flexible use of intensity stereo coding.

C.2.1.8 Adaptive Multichannel Prediction

Adaptive multichannel prediction is used to reduce the inter-channel redundancy. When using multichannel prediction, the signals in the transmission channels T2..T4 are predicted from the compatible stereo signal in the base bit stream (Lo, Ro). Instead of the actual signals in a subband group, the prediction error is transmitted, together with predictor coefficients and delay compensation.

The possible prediction equations are (all calculations are done on a frame by frame basis):

$$\hat{T}2(n) = \sum_{pci=0}^{2} pred\_coef\_T2\_0[sbgr][pci] * T0(n - delay\_comp - pci) + \sum_{pci=0}^{2} pred\_coef\_T2\_1[sbgr][pci] * T1(n - delay\_comp - pci)$$

$$\hat{T}3(n) = \sum_{pci=0}^{2} pred\_coef\_T3\_0[sbgr][pci] * T0(n - delay\_comp - pci) + \sum_{pci=0}^{2} pred\_coef\_T3\_1[sbgr][pci] * T1(n - delay\_comp - pci)$$

$$\hat{T}4(n) = \sum_{pci=0}^{2} pred\_coef\_T4\_0[sbgr][pci] * T0(n - delay\_comp - pci) + \sum_{pci=0}^{2} pred\_coef\_T4\_1[sbgr][pci] * T1(n - delay\_comp - pci)$$

Instead of T2, T3 and T4, the prediction error signals

$$\varepsilon_{T2}(n) = T2(n) - \hat{T}2(n)$$

$$\varepsilon_{T3}(n) = T3(n) - \hat{T}3(n)$$

$\varepsilon_{T4}(n) = T4(n) - \hat{T}4(n)$

are transmitted.

The predictor coefficients pred_coef[sbgr,px,pci] are calculated such that the power of the prediction error signals is minimised resulting in the optimum prediction gain. The prediction gain is the ratio of the energies of the original signals and the corresponding prediction error signals, expressed in dB. A detailed description of these calculations is given below.

By comparing the actual prediction gain with the amount of side information required to code the predictor coefficients, it is decided for which subband groups and for which signals (L^W, R^W, C^W, LS^W , RS^W and S^W) prediction is used in each audio frame. The 8 bits needed to code one predictor coefficient correspond to a prediction gain of 1,34 dB.

If the prediction error signal is transmitted instead of the original signal, the SMR-values used for the bit allocation procedure have to be reduced by the calculated prediction gain. To provide the SCFSI information necessary for the bit allocation, "preliminary" versions of the transmitted prediction error signals have to be calculated.

To avoid the summing of different quantisation errors, it is recommended to quantise and requantise the signals Lo, Ro and the predictor coefficients before the "final" prediction error signals are calculated. Thus, the prediction error signals will be identical in the encoder and decoder.

The coding of the transmitted signals T0, T1, T2, T3, T4 is performed as usual using "allocation", "SCFSI", "scalefactor" and "sample".

Encoding of one audio frame

{

- subband filtering;
- matrixing;
- scalefactor calculation;
- transmission pattern calculation (SCFSI);
- calculation of the SMR values by the psychoacoustic model;
- transmission channel allocation
- dynamic cross-talk
- calculation of delay compensation, predictor coefficients and prediction gain;
- calculation of the predictor select information (predsi);
- calculation of the modified SMR values;
- quantisation of the predictor coefficients;
- calculation of the preliminary prediction error signals;
- scalefactor calculation;
- transmission pattern calculation (SCFSI);
- bit allocation (using modified SMR values);
- quantisation of the subband samples;
- dequantisation of the subband samples;
- calculation of the final prediction error signals (using the dequantised subband samples)
- scalefactor calculation;
- transmission pattern calculation (SCFSI);
- quantisation of the subband samples;
- bit stream formatting;

}

Calculation of the predictor coefficients, prediction gain and predictor select information

The following C-like description is a simple example of the prediction calculation for the case that the transmission channels T2, T3, T4 contain C, LS and RS respectively. No use is made of dynamic cross-talk, and only zero order predictor without delay compensation is used. The output of the procedure consists of the coefficients coef_0, coef_1, coef_2, coef_3 and the corresponding predictor select information predsi[0..3].

In the current example, sqr() represents the square value of the argument, and sqrt() the squareroot of the argument.

The meaning of coef_0..coef_3 is:

```
pred_coef_C0 = coef_0;
pred_coef_C1 = coef_1;
pred_coef_LS = coef_2;
pred_coef_RS = coef_3;
```

The procedure to be followed in other cases is similar.

```
for (sbgr=0; sbgr<12; sbgr++){
/* calculation of variances and correlation functions
                    using short-term estimates */
st1 = st2 = sc = sls = srs = 0;
ct1c = ct2c = ct1t2 = ct1ls = ct2rs = 0;
numsb=((sbgr==11)?sblimit:sbgr_min[sbgr+1]) – sbgr_min[sbgr];
for (sb=sbgr_min[sbgr]; sb<sbgr_min[sbgr]+numsb; sb++)
          for (gr=0; gr<3; gr++)
                    for (i=0; i<12; i++){
                              st1 += sqr(sb_sample[0][gr][i][sb]);
                              st2 += sqr(sb_sample[1][gr][i][sb]);
                              sc  += sqr(sb_sample[2][gr][i][sb]);
                              sls += sqr(sb_sample[3][gr][i][sb]);
                              srs += sqr(sb_sample[4][gr][i][sb]);
                              ct1c+= sb_sample[0][gr][i][sb]*sb_sample[2][gr][i][sb];
                              ct2c+= sb_sample[1][gr][i][sb] * sb_sample[2][gr][i][sb];
                              ct1t2+= sb_sample[0][gr][i][sb] * sb_sample[1][gr][i][sb];
                              ct1ls+=sb_sample[0][gr][i][sb] * sb_sample[3][gr][i][sb];
                              ct2rs+=sb_sample[1][gr][i][sb] * sb_sample[4][gr][i][sb];
                    }
                    st1 = sqrt (st1/(3*12*numsb));
                    st2 = sqrt (st2/(3*12*numsb));
                    sc= sqrt (sc/(3*12*numsb));
                    sls = sqrt (sls/(3*12*numsb));
                    srs = sqrt (srs/(3*12*numsb));
                    st1 = (st1>MIN_S)?st1:MIN_S;  /* to avoid division by 0 */
                    st2 = (st2>MIN_S)?st2:MIN_S;
                    sc = (sc>MIN_S)?sc:MIN_S;
                    sls = (sls>MIN_S)?sls:MIN_S;
                    srs = (srs>MIN_S)?srs:MIN_S;
                    ct1c  = ct1c / (st1*sc);
                    ct2c  = ct2c / (st2*sc);
                    ct1t2 = ct1t2 / (st1*st2);
                    ct1ls = ct1ls / (st1*sls);
                    ct2rs = ct2rs / (st2*srs);
                    /* calculation of predictor coefficients */
                    coef_0 = sc / st1 * ct1c;
                    coef_1 = sc / st2 * ct2c;
                    coef_x0 = sc / st1 * (ct1c – ct2c*ct1t2) / (1– sqr(ct1t2));
                    coef_x1 = sc / st2 * (ct2c – ct1c*ct1t2) / (1– sqr(ct1t2));
                    coef_2 = sls / st1 * ct1ls;
                    coef_3 = srs / st2 * ct2rs;
                    /* calculation of prediction gains */
                    /* problem: if sbgr contains more than one subband the */
                    /* prediction gain can be different in the subbands!!! */
                    gain_0 = 10 * log (1/(1– sqr(ct1c)));
                    gain_1 = 10 * log (1/(1– sqr(ct2c)));
                    gain_2 = 10 * log (1/(1– sqr(ct1ls)));
                    gain_3 = 10 * log (1/(1– sqr(ct2rs)));
                    temp = sqr(sc) - 2*(coef_x0*ct1c*st1*sc) – 2*(coef_x1*ct2c*st2*sc)
                              + 2*(coef_x0*coef_x1*ct1t2*st1*st2) + sqr(coef_x0*st1)
                              + sqr(coef_x1*st2);
                    gain_01 = 10 * log (sqr(sc) / temp);
```

```
/* calculation of predictor select information */
maxgain = 0;
maxmode = 0;
if (gain_0 – SI_COEF/numsb > maxgain) {
        maxgain = gain_0 – SI_COEF/numsb;
        maxmode = 1;
}
if (gain_1 – SI_COEF/numsb > maxgain) {
        maxgain = gain_1 – SI_COEF/numsb;
        maxmode = 2;
}
if (gain_01 – 2*SI_COEF/numsb > maxgain) {
        maxgain = gain_01 – 2*SI_COEF/numsb;
        maxmode = 3;
}
switch (maxmode){
case 0 :
        temp_pred_gain[0] = 0;
        predsi[0] = '0';
        predsi[1] = '0';
        break;
case 1 :
        temp_pred_gain[0] = gain_0;
        predsi[0] = '1';
        predsi[1] = '0';
        pred_coef[sbgr][0] = coef_0;
        break;
case 2 :
        temp_pred_gain[0] = gain_1;
        predsi[0] = '0';
        predsi[1] = '1';
        pred_coef[sbgr][1] = coef_1;
        break;
case 3 :
        temp_pred_gain[0] = gain_01;
        predsi[0] = '1';
        predsi[1] = '1';
        pred_coef[sbgr][0] = coef_x0;
        pred_coef[sbgr][1] = coef_x1;
        break;
}
if (gain_2 > SI_COEF/numsb){
        temp_pred_gain[1] = gain_2;
        predsi[2] = '1';
        pred_coef[sbgr][2] = coef_2;
}
else{
        temp_pred_gain[1] = 0;
        predsi[2] = '0';
}
if (gain_3 > SI_COEF/numsb){
        temp_pred_gain[2] = gain_3;
        predsi[3] = '1';
        pred_coef[sbgr][3] = coef_3;
}
else{
        temp_pred_gain[2] = 0;
        predsi[3] = '0';
}
/* simplifying assumption: prediction gain is the same in */
/*                  all subbands of one subband group */
```

```
        for (sb=sbgr_min[sbgr]; sb<sbgr_min[sbgr]+numsb; sb++)
            for (i=0; i<3; i++)
                pred_gain[i][sb] = temp_pred_gain[i];
        /* modification of the SMR values according to the prediction gain */
        /* i.e.: SMR is reduced by the prediction gain */
        for (sb=sbgr_min[sbgr]; sb<sbgr_min[sbgr]+numsb; sb++)
            for (i=0; i<3; i++)
                smr[i+2][sb] –= pred_gain[i][sb];
}  /* for (sbgr=0; sbgr<12; sbgr++) */
```

C.2.1.9 Phantom coding of centre channel

If there is a shortage of bits, the use of Phantom coding of the centre channel can provide a significant gain in an unobtrusive way. The centre signal is low-pass and high-pass filtered to obtain a low and a high frequency part. The high frequency part of the centre channel is attenuated by 3 dB, and added to the left and right channels. The filtering and summation should be done in the PCM domain, to avoid aliasing problems at the subband bound above which the Phantom coding is done. The centre bits in the multichannel bit stream have to be set to '11'. Only the bit allocation, scalefactor select information, scalefactors, and sample data of the low frequency part of the centre signal are actually transmitted.

C.2.1.10 Bit Allocation

The bit allocation procedure is similar to that used in ISO/IEC 11172-3, but now applies to 5 channels and, optionally, a low frequency enhancement channel. In Layer I, the procedure is slightly different because the compatible part requires three bit allocations, while the multichannel extension part requires only one bit allocation. A simple way to approach this is to use the same bit allocation for each three consecutive Layer I base frames, and to triple the number of bits required for side information and samples of this part. After this, it can be treated the same as in Layer II. From the total number of available bits, 2 bits have to be subtracted, because one bit, which is set to zero, has to be inserted at the end of the first two of each three consecutive base frames. This is for synchronisation purposes in the case of a bit oriented channel without further framing.

C.2.1.11 Multilingual

The encoding of multilingual channels can be done at the same sampling frequency as that of the compatible and multichannel data in the bit stream, or at half that sampling frequency. In the latter case, a significant gain in coding efficiency is obtained at the expense of a reduction in bandwidth. If the bandwidth of the input signal is already limited, as in case of speech signals, this bandwidth limitation is no real drawback.

If the full sampling frequency is used, the encoding is done according to ISO/IEC 11172-3, with the exception that no intensity stereo coding is possible and up to seven channels can be multiplexed. If the half sampling frequency is used, the encoding is done according to the extension to lower sampling frequencies as described in C.1.2, with the exception that no intensity stereo coding is possible, that up to seven channels can be multiplexed, and that the frames contain half the number of subband samples.

C.2.1.12 Formatting

The coded audio bit stream has to be formatted according to the syntax in 2.5.1. In Layer II, the multichannel extension data has to be inserted in the base bit stream directly after the audio data of the backwards compatible signal. The remaining bits in the base frame can be used for ancillary data. In Layer I, the multichannel extension data consists basically of three parts, distributed over three Layer I base frames. Part 1 has to start directly after the audio data of the backwards compatible signal, and ends one bit before the next syncword. The last bit of the base frame is set to zero. Part 2 starts directly after the audio data of the backwards compatible signal of the next base frame, and ends 1 bit before the end of that base frame. Again, the last bit is set to zero. Part 3 starts directly after the audio data of the backwards compatible signal of the next base frame, and ends before the end of that base frame. The remaining bits can be used for ancillary data. If a larger frame size than that provided by the base frames is desired, an optional extension frame can be used to allocate the bits that do not fit in the base frame(s).

C.2.2 Multichannel extension Layer III

C.2.2.1 Psychoacoustic models

The two psychoacoustic models as described in ISO/IEC 11172-3 are also suitable here. For all five channels and for the compatible channels, the threshold levels for all scalefactor bands are calculated. If encoding is done with matrix procedures (i.e. dematrix_procedure != '11'), the block_types of all channels should be the same for optimum system operation. This is achieved by applying the window switching sequence described in ISO/IEC 11172-3, C.1.5.3 / 2 to all channels when the condition for window switching is fulfilled in at least one of the channels.

C.2.2.2 The filterbank

The filterbank used is the same as in ISO/IEC 11172-3, i.e. a 32-band polyphase filterbank, followed by an MDCT on the subband signals and some processing for aliasing reduction (see ISO/IEC 11172-3, C.1.5.3 / 3). The filterbank is applied to all five channels according to the block_type values which have been calculated by the psychoacoustic model.

C.2.2.3 Segment list processing

Segment lists are a general way to introduce joint stereo coding where the output to one channel is derived as a scaled version of the data in a different channel. A requirement for application of segment_list processing is that all channels use the same block_type. This is recommended for coding of multichannel signals except for dematrix_procedure=='11'. In this case, all channels which are grouped together by composite coding should have the same block_type.

While the syntax allows for several segments containing different joint stereo modes within one block, it is possible to restrict the use of segment_lists to one segment at high frequencies. This is the recommended practice for the encoder described here.

The application of joint stereo coding is done in a controlled way by using a joint stereo detection procedure in order to determine the best joint stereo combinations between the channels. The variable dematrix_length indicates the separation point between adaptive dematrixing and joint stereo processing.

Joint stereo detection is done for all possible values of dematrix_length, from 0 to 14. Dematrix_length is set equal to the lowest index of dematrix_length where joint stereo detection shows an anticipated gain from joint stereo coding while also meeting the requirements for a virtually unimpaired image impression.

Joint stereo detection is accomplished using a search for the best joint stereo combination. Using all reasonable combinations of channels (like L+LS, R+RS, L+C+LS, R+C+RS, LS+RS etc.), the simulated joint stereo combination and the original are compared. This comparison is done by evaluating the short time energies of original and simulated joint stereo signals. If the relative energy deviation is greater than 0.03, joint stereo is not viable for this combination. In parallel, the reduction in bitrate possible from joint stereo coding is estimated using the Perceptual Entropy (PE). The combination of channels offering minimum quality loss, as indicated by the short time energy ratio, and, at the same time, the most gain in terms of PE is selected.

For the transmission of the selected joint stereo combination, one channel is used as the "carrier" for this combination. This carrier channel contains the spectral information of the joint stereo combination. The carrier channel is chosen from all combination channels as the combination channel with the highest energy.

C.2.2.4 Dynamic transmission channel switching

In order to avoid audible artefacts due to the dematrixing process, it is necessary to choose the right transmission channel allocation. This can be done in several possible ways:

- Selection of a whole channel for transmission can be done with a few bits of side information using the "seglist_present" syntax. For a valid Layer III bit stream, the encoder may select up to two channels for dematrixing by setting seglist_present[ ] to zero. In this case, the corresponding tc_present[ ] can be set to zero indicating that no further side information for the corresponding TC is transmitted.

- To have better control of the dematrixing configuration, the selection of the transmitted channels can be done on a scalefactorband group by scalefactorband group basis. This can be achieved by using the dematrix_select syntax. For scalefactorband groups above dematrix_length, the same effect can be reached by selecting a tc_select value of 7 for the respective segment.

The selection process can be based on the following criterion: For each channel, its masking ability (masked threshold, xmin) is calculated by the psychoacoustic model as for MPEG-1 Layer III encoding (see ISO/IEC 11172-3, C.1.5.3). From all channels, the two channels with the strongest masking ability are chosen for reconstruction by dematrixing and thus do not need to be transmitted. In case one of the channels is the centre channel and the computed masked thresholds differ by more than 6dB, only the channel with the strongest masking ability is selected for dematrixing.

C.2.2.5 Matrixing

The compatible stereo signals Lo / Ro are calculated from the multichannel signals as follows:

Procedure 0, 1, 3: $Lo = \alpha * (L + \beta*C + \gamma*LS)$

$Ro = \alpha * (R + \beta*C + \gamma*RS)$

and

Procedure 2: $Lo = \alpha * (L + \beta*C - \gamma*jS)$

$Ro = \alpha * (R + \beta*C + \gamma*jS)$

where jS is derived from LS and RS by calculation of the mono component, bandwidth limitation to the range 100-7000 Hz, dynamic range compression, and 90 degrees phase shifting.

In the above equations, α is an overall attenuation for all channels and β and γ are the attenuation factors of the centre and surround signals. The attenuation factor values are specific to each dematrixing procedure:

dematrix_procedure	α	β	γ
'00'	$1/(1 + \sqrt{2})$	$1/\sqrt{2}$	$1/\sqrt{2}$
'01'	$1/(1,5 + 0,5*\sqrt{2})$	$1/\sqrt{2}$	0,5
'10'	$1/(1 + \sqrt{2})$	$1/\sqrt{2}$	$1/\sqrt{2}$
'11'	1	0	0

Please note that, unlike in Layer I and II, all handling of the multichannel stereo signals L, R, C, LS, RS, S is done without applying a weighting procedure.

C.2.2.6 Adaptive multichannel prediction

Adaptive multichannel prediction can be used in Layer III multichannel coding in the same way as in Layer I and II except that the prediction procedure is applied to the output values of the hybrid filterbank.

C.2.2.7 Quantisation and coding

For subsequent coding, the output data of all five input channels and the two compatible channels are converted to a TC representation. This is done by removing all spectral parts from the output channel spectra of the filterbank which do not have to be transmitted. There are two cases where spectral parts are excluded from transmission:

- Spectral data which is reconstructed in the decoder by dematrixing will be excluded from transmission in the TCs. This is done according to the result of the dynamic transmission channel switching.

- In the case of joint stereo coding, only the carrier portion of the involved channel data is transmitted in the TCs. All other involved channel data is reconstructed in the decoder by joint stereo processing via the segment list syntax.

After the assembling of the TC data, all TCs are quantised in the same manner as the channel spectra of a Layer III stereo encoder using the iteration strategy described in ISO/IEC 11172-3, C.1.5.4. The threshold values for the respective channel and scalefactorband which have been calculated by the psychoacoustic model are used as the iteration target (i.e. the maximum allowed distortion for each scalefactorband, xmin). More sophisticated encoding strategies may involve the modification of the iteration targets according to the calculated threshold levels of other channels.

The allocation of bits among the coded TCs is done according to their relative contribution in terms of perceptual entropy (PE) as follows:

$$tc\_bits_{ch} = \frac{pe_{ch}}{\sum_i pe_i} \cdot total\_bits$$

where tc_bits denotes the allocated bits for TC #ch, pe_i denotes the total perceptual entropy of channel number i, and total_bits is the total available number of bits for this granule depending on bit-rate and sampling frequency. For the definition of the perceptual entropy see ISO/IEC 11172-3, C.1.5.3 / 2.1.

C.2.2.8 Multilingual extensions

The encoding is done, dependent on the multi_lingual_fs selected, as described in ISO/IEC 11172-3 or with the modifications described in C.1.3.

Annex D

(informative)

Psychoacoustic models

D.1 Psychoacoustic Model 1 for Lower Sampling Frequencies

A description of the psychoacoustic model 1 is repeated here, with the necessary adaptations in respect to the lower sampling frequencies.

The calculation of the psychoacoustic model has to be adapted to the corresponding layer. The example presented here is valid for Layers I and II. The model can be adapted to Layer III.

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 totalling 36 subband samples corresponding to 3*384 (1 152) 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 performed in parallel with the subband filter operation 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 optimised 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 bitrate 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 24 kHz is assumed, unless stated otherwise. For the other two sampling frequencies all frequencies mentioned should be scaled accordingly.

Step 1 Calculation of spectrum

The FFT is in principle the same as in ISO/IEC 11172-3, but due to the different sampling frequency the length when expressed in ms is different.

Technical data of the FFT:

		Layer I	Layer II
-	transform length N	512 samples	1024 samples
	Window size if Fs = 24 kHz	21,33 ms	42,67 ms
	Window size if Fs = 22,05 kHz	23,22 ms	46,44 ms
	Window size if Fs = 16 kHz	32 ms	64 ms
-	Frequency resolution	Fs / 512	Fs / 1024

- Hann window, h(i):

$$h(i) = \sqrt{8/3} * 0,5 * \{1 - \cos[2\pi\,(i)/N]\} \qquad 0 <= i <= N-1$$

- power density spectrum X(k):

$$X(k) = 10 * \log_{10} \left| \frac{1}{N} \sum_{l=0}^{N-1} h(l) * s(l) * e^{(-jkl*2\pi/N)} \right|^2 \ \mathrm{dB} \qquad k = 0...N/2,$$

where s(l) is the input signal.

A normalisation 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 96 dB.

Step 2 Determination of the sound pressure level

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

$$L_{sb}(n) = \mathrm{MAX}[\ X(k),\ 20*\log(scf_{max}(n)*32\ 768)-10\ ]\ \mathrm{dB}$$

X(k) in subband n

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.

The following alternative method of calculating $L_{sb}(n)$ offers a potential for better encoder performance, but this technique has not been subjected to a formal audio quality test.

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

$$L_{sb}(n) = \mathrm{MAX}[\ X_{spl}(n),\ 20*\log(scf_{max}(n)*32\ 768)-10\ ]\ \mathrm{dB}$$

with

$$X_{spl}(n) = 10*\log_{10}\Big(\sum_{k} 10^{X(k)/10}\Big)\ \mathrm{dB}$$

k in subband n

where $X_{spl}(n)$ is the alternative sound pressure level corresponding to subband n.

Step 3 Considering the threshold in quiet

The threshold in quiet $LT_q(k)$, also called absolute threshold, is available in the tables "Frequencies, critical band rates and absolute threshold" (tables D.1a, D.1b, D.1c for Layer I; tables D.1d, D.1e, D.1f for Layer II). 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.

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 D.2a, D.2b, D.2c for Layer I; tables D.2d, D.2e, D.2f for Layer II).

The bandwidth of the critical bands varies with the centre 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: 16 kHz

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 <=	7,5 kHz

Sampling rate: 22,05 kHz

df = 86,133 Hz	0 kHz	< f <=	2,756 kHz
df = 129,199 Hz	2,756 kHz	< f <=	5,512 kHz
df = 258,398 Hz	5,512 kHz	< f <=	10,336 kHz

Sampling rate: 24 kHz

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 <=	11,250 kHz

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

a) Labelling of local maxima

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

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

b) 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, Fs=16 kHz:

j = –2, +2	for	2 < k < 96
j = –3,–2, +2,+3	for	96 <= k < 192
j = –6,...,–2,+2,...,+6	for	192<= k < 250

Layer II, Fs=16 kHz:

j = –4, +4	for	4 < k < 192
j = –6,...,–2, +2,...,+6	for	192 <= k < 384
j = –12,...,–2, +2,..., +12	for	384 <= k < 500

Layer I, Fs=22,05, 24 kHz:

j = –2, +2	for	2 < k < 64
j = –3,–2, +2,+3	for	64 <= k < 128
j = –6,...,–2,+2,...,+6	for	128<= k < 250

Layer II, Fs=22,05, 24 kHz:

j = –4, +4	for	4 < k < 128
j = –6,...,–2, +2,...,+6	for	128 <= k < 256
j = –12,...,–2, +2,..., +12	for	256 <= 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 $X_{tm}(k) = 10 * \log_{10} \{ 10^{\frac{X(k-1)}{10}} + 10^{\frac{X(k)}{10}} + 10^{\frac{X(k+1)}{10}} \}$, in dB.
- Tonal flag.

Next, all spectral lines within the examined frequency range are set to –∞ dB.

c) 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 D.2a, D.2b, D.2c for Layer I; tables D.2d, D.2e, D.2f for Layer II). 21 critical bands are used for the sampling rate of 16 kHz, 23 critical bands are used for 22,05 kHz and 24 kHz. Within each critical band, the power of the spectral lines (remaining after the tonal components have been zeroed) are summed to form the sound pressure level of the new non-tonal component $X_{nm}(k)$ 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.

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

$$X_{tm}(k) >= LT_q(k) \quad \text{or} \quad X_{nm}(k) >= LT_q(k)$$

In this expression, $LT_q(k)$ is the absolute threshold (or threshold in quiet) at the frequency of index k. These values are given in tables D.1a, D.1b, D.1c for Layer I; tables D.1d, D.1e, D.1f for Layer II.

b) 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 D.1a, D.1b, D.1c for Layer I; tables D.1d, D.1e, D.1f for Layer II.

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, every fourth spectral line is considered for the next 18 subbands (see also tables D.1a, D.1b, D.1c for Layer I).

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, every eighth spectral line is considered for the next 18 subbands (See also tables D.1d, D.1e, D.1f for Layer II).

The number of samples, n, in the subsampled frequency domain depends on the layer. For Layer I, n equals 108, for Layer II, n equals 132.

Every tonal and non-tonal component is assigned the value of the index i that most closely corresponds to the frequency of the original spectral line X(k). This index i is given in tables D.1a, D.1b, D.1c for Layer I; tables D.1d, D.1e, D.1f for Layer II.

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

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

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

In this formula, LT_{tm} and LT_{nm} are the individual masking thresholds at critical band rate z in Bark of the masking component at the critical band rate of the masker 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 maskers (av_{tm} and av_{nm}).

For tonal maskers, it is given by

$$av_{tm} = -1{,}525 - 0{,}275 * z(j) - 4{,}5 \text{ dB},$$

and for non-tonal maskers

$$av_{nm} = -1{,}525 - 0{,}175 * z(j) - 0{,}5 \text{ dB}.$$

The masking function vf of a masker is characterised 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 D.1a, D.1b, D.1c for Layer I; tables D.1d, D.1e, D.1f for Layer II. The masking function, which is the same for tonal and non-tonal maskers, is given by:

$$vf = 17 * (dz + 1) - (0{,}4 * X[z(j)] + 6) \text{ dB} \qquad \text{for } -3 <= dz < -1 \text{ Bark}$$

$$vf = (0{,}4 * X[z(j)] + 6) * dz \text{ dB} \qquad \text{for } -1 <= dz < 0 \text{ Bark}$$

$$vf = -17 * dz \text{ dB} \qquad \text{for } 0 <= dz < 1 \text{ Bark}$$

$$vf = -(dz - 1) * (17 - 0{,}15 * X[z(j)]) - 17 \text{ dB} \qquad \text{for } 1 <= dz < 8 \text{ Bark}$$

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

Step 7 Calculation of the global masking threshold LT_g

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 thresholds of each of the j tonal and non-tonal maskers and from the threshold in quiet $LT_q(i)$. This is also given in tables D.1a, D.1b, D.1c for Layer I; tables D.1d, D.1e, D.1f for Layer II. The global masking threshold is found by summing the powers corresponding to the individual masking thresholds and the threshold in quiet.

$$LT_g(i) = 10 \log_{10} \left(10^{LT_q(i)/10} + \sum_{j=1}^{m} 10^{LT_{tm}(z(j),z(i))/10} + \sum_{j=1}^{n} 10^{LT_{nm}(z(j),z(i))/10} \right)$$

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 $-\infty$ dB.

Step 8 Determination of the minimum masking threshold

The minimum masking level LTmin(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 tables D.1a, D.1b, D.1c for Layer I; tables D.1d, D.1e, D.1f for Layer II. 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

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

is computed for every subband n.

Table D.1a. — Frequencies, critical band rates and absolute threshold

Table is valid for Layer I at a sampling rate of 16 kHz.

Index Number i	Frequency [Hz]	Crit.Band Rate [z]	Absolute Thresh. [dB]	Index Number i	Frequency [Hz]	Crit.Band Rate [z]	Absolute Thresh. [dB]
1	31,25	0,309	58,23	55	1937,50	12,898	0,02
2	62,50	0,617	33,44	56	2000,00	13,104	-0,25
3	93,75	0,925	24,17	57	2062,50	13,302	-0,54
4	125,00	1,232	19,20	58	2125,00	13,493	-0,83
5	156,25	1,538	16,05	59	2187,50	13,678	-1,12
6	187,50	1,842	13,87	60	2250,00	13,855	-1,43
7	218,75	2,145	12,26	61	2312,50	14,027	-1,73
8	250,00	2,445	11,01	62	2375,00	14,193	-2,04
9	281,25	2,742	10,01	63	2437,50	14,354	-2,34
10	312,50	3,037	9,20	64	2500,00	14,509	-2,64
11	343,75	3,329	8,52	65	2562,50	14,660	-2,93
12	375,00	3,618	7,94	66	2625,00	14,807	-3,22
13	406,25	3,903	7,44	67	2687,50	14,949	-3,49
14	437,50	4,185	7,00	68	2750,00	15,087	-3,74
15	468,75	4,463	6,62	69	2812,50	15,221	-3,98
16	500,00	4,736	6,28	70	2875,00	15,351	-4,20
17	531,25	5,006	5,97	71	2937,50	15,478	-4,40
18	562,50	5,272	5,70	72	3000,00	15,602	-4,57
19	593,75	5,533	5,44	73	3125,00	15,841	-4,82
20	625,00	5,789	5,21	74	3250,00	16,069	-4,96
21	656,25	6,041	5,00	75	3375,00	16,287	-4,98
22	687,50	6,289	4,80	76	3500,00	16,496	-4,90
23	718,75	6,532	4,62	77	3625,00	16,697	-4,70
24	750,00	6,770	4,45	78	3750,00	16,891	-4,39
25	781,25	7,004	4,29	79	3875,00	17,078	-3,99
26	812,50	7,233	4,14	80	4000,00	17,259	-3,51
27	843,75	7,457	4,00	81	4125,00	17,434	-2,99
28	875,00	7,677	3,86	82	4250,00	17,605	-2,45
29	906,25	7,892	3,73	83	4375,00	17,770	-1,90
30	937,50	8,103	3,61	84	4500,00	17,932	-1,37
31	968,75	8,309	3,49	85	4625,00	18,089	-0,86
32	1000,00	8,511	3,37	86	4750,00	18,242	-0,39
33	1031,25	8,708	3,26	87	4875,00	18,392	0,03
34	1062,50	8,901	3,15	88	5000,00	18,539	0,40
35	1093,75	9,090	3,04	89	5125,00	18,682	0,72
36	1125,00	9,275	2,93	90	5250,00	18,823	1,00
37	1156,25	9,456	2,83	91	5375,00	18,960	1,24
38	1187,50	9,632	2,73	92	5500,00	19,095	1,44
39	1218,75	9,805	2,63	93	5625,00	19,226	1,62
40	1250,00	9,974	2,53	94	5750,00	19,356	1,78
41	1281,25	10,139	2,42	95	5875,00	19,482	1,92
42	1312,50	10,301	2,32	96	6000,00	19,606	2,05
43	1343,75	10,459	2,22	97	6125,00	19,728	2,18
44	1375,00	10,614	2,12	98	6250,00	19,847	2,30
45	1406,25	10,765	2,02	99	6375,00	19,964	2,42
46	1437,50	10,913	1,92	100	6500,00	20,079	2,55
47	1468,75	11,058	1,81	101	6625,00	20,191	2,69
48	1500,00	11,199	1,71	102	6750,00	20,300	2,82
49	1562,50	11,474	1,49	103	6875,00	20,408	2,97
50	1625,00	11,736	1,27	104	7000,00	20,513	3,13
51	1687,50	11,988	1,04	105	7125,00	20,616	3,29
52	1750,00	12,230	0,80	106	7250,00	20,717	3,46
53	1812,50	12,461	0,55	107	7375,00	20,815	3,65
54	1875,00	12,684	0,29	108	7500,00	20,912	3,84