



INTERNATIONAL STANDARD ISO/IEC 23003-1:2007
TECHNICAL CORRIGENDUM 1

Published 2008-02-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — MPEG audio technologies —

Part 1: MPEG Surround

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Technologies audio MPEG —

Partie 1: Ambiance MPEG

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 23003-1:2007 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 29, *Coding of Audio, Picture, Multimedia and Hypermedia Information*.

In the following, changes in existing text and tables are highlighted by gray background.

Throughout the whole document, replace:

down-mix

by:

downmix

Throughout the whole document, replace:

down mix

by:

downmix

Throughout the whole document, replace:

Uimsbf

by:

uimsbf

Throughout the whole document, replace:

Vlclbf

by:

vlclbf

Throughout the whole document, replace:

UiMsbF

by:

uimsbf

Throughout the whole document, replace:

BsMsbf

by:

bsmsbf

Throughout the whole document, including subclause headings, replace:

signalling

by:

signaling

and replace:

Signalling

by:

Signaling

In subclause 3.1, insert at the beginning the new subclause 3.1.1 given below and increment the numbering of the following subclauses 3.1.x:

3.1.1

ATD

Arbitrary Tree Data, corresponds to CLD for arbitrary tree elements.

In subclause 3.1.19, insert a line break between "3.1.19" and "processing band".

In subclause 4.3.1.4, replace the title:

Enhanced Matrixed Mode of MPEG Surround

by:

Enhanced Matrix Mode of MPEG Surround

In subclause 4.3.1.4, in the first paragraph, replace:

includes an enhanced matrixed mode that creates

by:

includes an enhanced matrix mode that creates

In subclause 4.3.1.6, in the first paragraph, replace:

system supports **matrixed** encoded downmixes. The MPEG Surround encoder can create a stereo downmix that is **matrixed** encoded, and can thus be decode by legacy **matrixed** surround decoders. The MPEG Surround decoder, will invert the **matrixed** encoded downmix, and produce the multi-channel signal based on the inverted downmix and the spatial parameters, without any degradation in quality due to the **matrixed** encoded downmix.

by:

system supports matrix encoded downmixes. The MPEG Surround encoder can create a stereo downmix that is matrix encoded, and can thus be decode by legacy matrix surround decoders. The MPEG Surround decoder, will invert the matrix encoded downmix, and produce the multi-channel signal based on the inverted downmix and the spatial parameters, without any degradation in quality due to the matrix encoded downmix.

In subclause 4.3.1.6, replace:

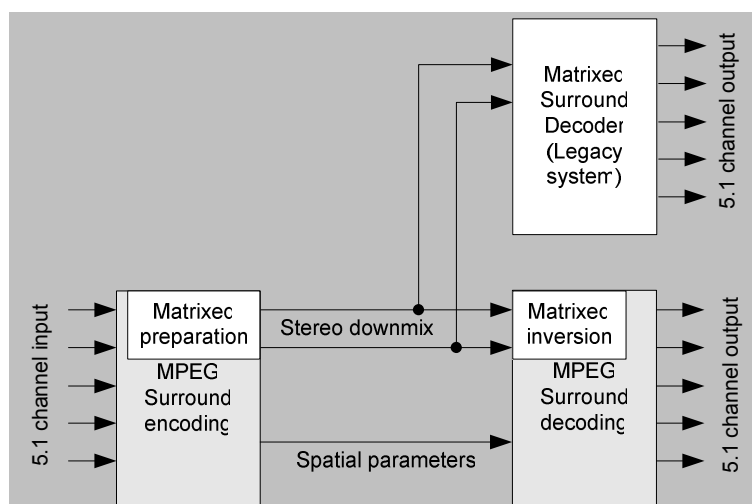


Figure 1 — Matrixed compatible MPEG Surround

by:

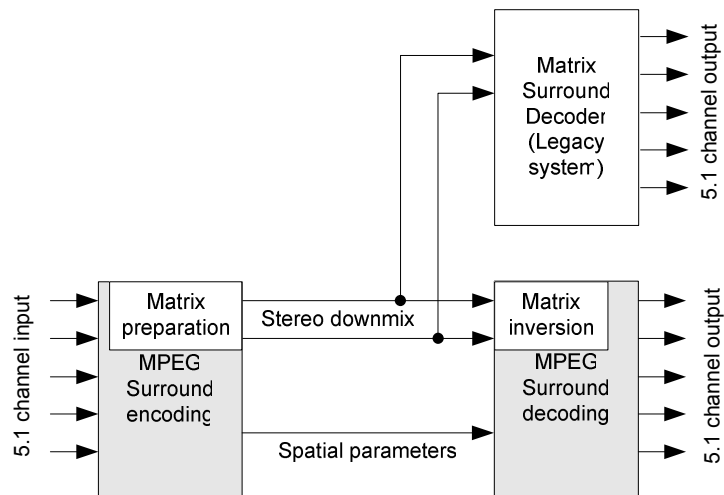


Figure 2 — Matrix compatible MPEG Surround

In subclause 4.3.1.7, in the first paragraph, replace:

the binaural encoded downmix is inverted similarly to the inversion of the matrixed compatible downmix.

by:

the binaural encoded downmix is inverted similarly to the inversion of the matrix compatible downmix.

In subclause 4.3.2, Table 2, replace:

Only supported over the complex valued part of the frequency range

by:

Only supported over a limited part of the frequency range

In subclause 4.3.2, replace the following row in Table 2:

Decorrelators		
---------------	--	--

by:

Decorrelators	Lattice decorrelators	PS decorrelators and low complexity lattice decorrelators
---------------	-----------------------	---

In subclause 4.4, replace:

The MPEG Surround decoder can be interfaced in either the time-domain or the QMF domain. In general, the MPEG Surround coder interfaces to the downmix channels by means of a 64 bands QMF domain frequency representation, identical to that standardized in ISO/IEC 14496-3 for High Quality Parametric Audio Coding. In the case the spatial coder is combined with HE-AAC, this QMF representation is directly available as an intermediate signal in the HE-AAC coder. In combination with alternative core coders, additional QMF analysis

or synthesis modules, as defined in ISO/IEC 14496-3, are required for the spatial decoder and encoder, respectively. A Low Power MPEG Surround decoder cannot be connected in the QMF domain with a High Quality HE-AAC decoder, and vice versa a High Quality MPEG Surround decoder cannot be connected in the QMF domain with a Low Power HE-AAC decoder. For both combinations they shall be combined in the **timedomain**.

by:

The MPEG Surround decoder can be interfaced in either the **time domain** or the QMF domain. In general, the MPEG Surround coder interfaces to the downmix channels by means of a 64 band QMF domain representation, identical to that standardized in ISO/IEC 14496-3 subclause 4.6.18. In the case the spatial coder is combined with HE-AAC, this QMF representation is directly available as an intermediate signal in the HE-AAC coder. In combination with alternative **downmix** coders, additional QMF analysis or synthesis modules, as defined in subclause 6.3.2.1, are required for the spatial decoder and encoder, respectively.

It is not possible to connect HE-AAC and MPEG Surround in the QMF domain if the number of QMF bands is different. HE-AAC employs a 64 band QMF domain representation for the output signal, or a 32 band QMF domain representation if downsampled SBR is used. MPEG Surround employs a 32, 64, or 128 band QMF domain representation for the input signal, depending upon the sampling frequency used. In any combination where the number of QMF bands is not the same for HE-AAC and MPEG Surround they shall be connected in the **time domain**.

A Low Power MPEG Surround decoder cannot be connected in the QMF domain with a High Quality HE-AAC decoder, and vice versa a High Quality MPEG Surround decoder cannot be connected in the QMF domain with a Low Power HE-AAC decoder. Both combinations shall be **connected** in the **time domain**.

In subclause 4.5, remove the following sentence from the first paragraph:

The direct connection in the QMF domain is not possible in case of upsampled or downsampled operation of MPEG Surround, as defined in subclause 6.3.3.

In subclause 4.5, replace the following sentence in the fifth paragraph:

Note that special consideration is required if an MPEG Surround decoder and an HE-AAC decoder are connected in the time domain while a connection in the QMF domain would have been possible.

by:

Note that special consideration is required if an MPEG Surround decoder and an HE-AAC decoder are connected in the time domain while a connection in the QMF domain would have been possible **according to subclause 4.4**.

In subclause 4.6, replace the title:

Downmix gain

by:

Pre- and post-gains

In subclause 4.7.2, replace:

Five different hierarchical levels are defined, which allow for different numbers of input and output channels, for different ranges of sampling rates, and for a different bandwidth of the residual signal decoding. The level of the decoder must be equal to or larger than the level of the bitstream in order to ensure proper decoding. In addition, decoders of level 1 and 2 are capable of decoding all bitstreams of level 2 and 3, though at a possibly slightly reduced quality due to the limitations of the decoder. The quality and format of the output of an MPEG Surround decoder furthermore depends on the specific decoder configuration, as detailed below. However, these decoder configuration aspects are completely orthogonal to the different levels of this profile.

by:

Five different hierarchical levels are defined, which allow for different numbers of input and output channels, for different ranges of sampling rates, and for a different bandwidth of the residual signal decoding. The level of the decoder must be equal to or larger than the level of the bitstream in order to ensure proper decoding. In addition, decoders of level 1, 2 and 3 are capable of decoding all bitstreams of level 2, 3 and 4, though at a possibly slightly reduced quality due to the limitations of the decoder. The quality and format of the output of an MPEG Surround decoder furthermore depends on the specific decoder configuration, as detailed below. However, these decoder configuration aspects are completely orthogonal to the different levels of this profile.

In subclause 4.7.2, in Table 3, replace:

Residual coding based External Downmix compensation (subclause 6.5.2.3.4)

by:

Residual coding based External Downmix compensation (subclause 6.1.6)

In subclause 4.7.2, at the penultimate bullet in the third bullet list, replace:

Stereo output (specified in subclause 6.4.7.3).

by:

Stereo output (specified in subclause 6.4.7).

In subclause 4.7.2, at the last bullet in the third bullet list, replace:

Binaural output (specified in clause 7).

by:

Binaural output (specified in clause 6.11).

In subclause 4.7.2, replace:

Table 4 — Levels of the Baseline MPEG Surround profile

Level	Tree configurations	Max. number output channels	Max. sampling rate [kHz]	Max. bandwidth residual coding [QMF bands]	Max. PCU High Quality decoder	Max. RCU High Quality decoder	Max. PCU Low Power decoder	Max. RCU Low Power decoder
1	515, 525 (Note 1) (Note 4)	2.0	48	0 (Note 2)	12	5	6	4
2	515, 525 (Note 4)	5.1	48	0 (Note 2)	25	15	12	11
3	515, 525 (Note 4)	5.1	48	64 (Note 3)	25	15	12	11
4	515, 525, 757, 727 (Note 4)	7.1	48	64 (Note 3)	34	21	17	15
5	515, 525, 757, 727, plus arbitrary tree extension	32 incl. LFE	96	64 (Note 3)	123 (max. 70 at 48 kHz sampling)	61 (max. 38 at 48 kHz sampling)	80 (max. 44 at 48 kHz sampling)	53 (max. 32 at 48 kHz sampling)

Note 1: This level provides a 2-channel stereo output.

Note 2: Residual coding data, if present in the bitstream, is not utilized, hence the residual decoding tool is not required.

Note 3: A low power decoder utilizes only residual coding data for the first 8 QMF bands, corresponding to approximately 2.7 kHz bandwidth.

Note 4: Arbitrary tree extension data, if present, is not utilized.

by:

Table 4 — Levels of the Baseline MPEG Surround profile

Level	Tree configurations	Max. number output channels	Max. sampling rate [kHz]	Max. bandwidth residual coding [QMF bands]	Max. PCU High Quality decoder	Max. RCU High Quality decoder	Max. PCU Low Power decoder	Max. RCU Low Power decoder
1	515, 525, 727 (Note 1) (Note 4)	2.0	48	0 (Note 2)	12	5	6	4
2	515, 525, 727 (Note 4)	5.1	48	0 (Note 2)	25	15	12	11
3	515, 525, 727 (Note 4)	5.1	48	64 (Note 3)	25	15	12	11
4	515, 525, 727 (Note 4)	7.1	48	64 (Note 3)	34	21	17	15

5	515, 525, 757, 727 (Note 4)	7.1	48	64 (Note 3)	34	21	17	15
6	515, 525, 757, 727, plus arbitrary tree extension	32 incl. LFE	96	64 (Note 3)	123 (max. 70 at 48 kHz sampling)	61 (max. 38 at 48 kHz sampling)	80 (max. 44 at 48 kHz sampling)	53 (max. 32 at 48 kHz sampling)

Note 1: This level provides a 2-channel stereo output.

Note 2: Residual coding data, if present in the bitstream, is not utilized, hence the residual decoding tool is not required.

Note 3: A low power decoder utilizes only residual coding data for the first 8 QMF bands, corresponding to approximately 2.7 kHz bandwidth.

Note 4: Arbitrary tree extension data, if present, is not utilized.

In subclause 4.7.2, after the bullet “Support of stereo output for 515...” insert the following new bullet:

- Support of 5.1 output for 727 (subclause 6.4.8) is mandatory for levels 2 and 3.

In subclause 4.7.2 replace:

- Support for “Multi-slots HRTF convolution approach” (subclause 6.11.4.2.3) is mandatory for all levels of a high quality decoder (since this mode is not applicable in case of a low power decoder), and the complexity figures assume that a set of HRTF filters with a length of 128 samples (level 1) or 512 samples (levels 2 to 5) in the time domain is used.

by:

- Support for “Multi-slots HRTF convolution approach” (subclause 6.11.4.2.3) is mandatory for all levels of a high quality decoder (since this mode is not applicable in case of a low power decoder), and the complexity figures assume that a set of HRTF filters with a length of 128 samples (level 1) or 512 samples (levels 2 to 6) in the time domain is used.

In subclause 5.1, in all syntax tables, replace:

Else

by:

else

In subclause 5.1, replace Table 8:

Syntax	No. of bits	Mnemonic
ParamHRTFset() {		
bsHRTFfreqRes;	3	uimsbf
bsHRTFasymmetric;	1	uimsbf
for (hc=0; hc<HRTFnumChan; hc++) {		Note 1
for (hb =0; hb <HRTFnumBands; hb ++){		Note 2
bsHRTFlevelLeft [hc][hb];	6	uimsbf
}		
if (bsHRTFasymmetric) {		
for (hb =0; hb <HRTFnumBands; hb ++){		Note 2
bsHRTFlevelRight [hc][hb];	6	uimsbf
}		
}		
bsHRTFphase [hc];	1	uimsbf
if (bsHRTFphase[hc]) {		
for (hb =0; hb <HRTFnumPhase; hb ++){		Note 3
bsHRTFphaseLR [hc][hb];	6	uimsbf
}		
}		
bsHRTFicc [hc];	1	uimsbf
if (bsHRTFicc[hc]) {		
for (hb =0; hb <HRTFnumBands; hb ++){		Note 2
bsHRTFiccLR [hc][hb];	3	uimsbf
}		
}		
}		
}		
Note 1: HRTFnumChan= 5. Note 2: HRTFnumBands is defined in Table 53 and depends on bsHRTFfreqRes. Note 3: HRTFnumPhase is defined in Table 53 and depends on bsHRTFfreqRes.		

by:

Syntax	No. of bits	Mnemonic
ParamHRTFset() { bsHRTFfreqRes ; bsHRTFasymmetric ; for (hc=0; hc<HRTFnumChan; hc++) { for (hb =0; hb < HRTFnumBands; hb ++) { bsHRTFlevelLeft [hc][hb]; } if (bsHRTFasymmetric) { for (hb =0; hb <HRTFnumBands; hb ++) { bsHRTFlevelRight [hc][hb]; } } bsHRTFphase [hc]; if (bsHRTFphase[hc]) { for (hb =0; hb <HRTFnumPhase; hb ++) { bsHRTFphaseLR [hc][hb]; } } } }	3 1 6 6 1 6	uimsbf uimsbf Note 1 Note 2 uimsbf Note 2 uimsbf uimsbf Note 3 uimsbf
Note 1: HRTFnumChan = 5. Note 2: HRTFnumBands is defined in Table 53 and depends on bsHRTFfreqRes. Note 3: HRTFnumPhase is defined in Table 53 and depends on bsHRTFfreqRes.		

In subclause 5.1, in Table 20 and Table 21, replace:

numTempShapeChan is defined by Table 40 and

by:

numTempShapeChan is defined by Table 67 and

In subclause 5.1, Table 26, replace:

Pilot = 1Dhuff_dec(hcodPilot_XXX, **bsCodeW**);

by:

Pilot = 1Dhuff_dec(hcodFirstBand_XXX, **bsCodeW**);

In subclause 5.1, Table 26, replace:

Note 1: XXX is to be replaced by the value of dataType (CLD, ICC or CPC, ATD) .

by:

Note 1: XXX is to be replaced by the value of dataType (CLD, ICC, CPC, ATD) .

In subclause 5.1, replace Table 34:

Table 34 — Syntax of ResidualData()

Syntax	No. of bits	Mnemonic
<pre> ResidualData() { for (i=0; i<numOttBoxes+numTttBoxes; i++) { if (bsResidualPresent[i]) { if (i<numOttBoxes) { for (ps=0; ps<numParamSets; ps++) { bslccDiffPresent[i][ps]; if (bslccDiffPresent[i][ps]) { for (pb=0; pb<bsResidualBands[i]; pb++) { lccDiff[i][ps][pb] = 1Dhuff_dec(hcod1D_ICC_Diff,bsCodeW); } } } } tempExtraFrame=numSlots/(bsResidualFramesPerSpatialFrame+1); for (rf=0; rf<bsResidualFramesPerSpatialFrame; rf++) individual_channel_stream(0); if (window_sequence == EIGHT_SHORT_SEQUENCE) && ((tempExtraFrame == 18) (tempExtraFrame == 24) (tempExtraFrame == 30)) { individual_channel_stream(0); } } } } </pre>	<p>1</p> <p>1..7</p>	<p>Note 2 Uimsbf</p> <p>vlclbf Note 3</p> <p>Note 4</p> <p>Note 1 Note 5</p> <p>Note 1</p>
<p>Note1: individual_channel_stream(0) according to MPEG-2 AAC Low Complexity profile bitstream syntax described in subclause 6.3 of ISO/IEC 13818-7.</p> <p>Note 2: numParamSets is defined by numParamSets = bsNumParamSets + 1.</p> <p>Note 3: 1Dhuff_dec() is defined in Annex A.1.</p> <p>Note 4: numSlots is defined by numSlots = bsFrameLength +1. Furthermore the division shall be interpreted as ANSI C integer division.</p> <p>Note 5: individual_channel_stream(0) determines the value of window_sequence.</p>		

by:

Table 34 — Syntax of ResidualData()

Syntax	No. of bits	Mnemonic
<pre> ResidualData() { for (k=0; k<numOttBoxes+numTttBoxes; k++) { if (bsTreeConfig==3 bsTreeConfig==4) { i = resReorder[k]; } else i = k; if (bsResidualPresent[i]) { if (i<numOttBoxes) { for (ps=0; ps<numParamSets; ps++) { bslccDiffPresent[i][ps]; if (bslccDiffPresent[i][ps]) { for (pb=0; pb<bsResidualBands[i]; pb++) { lccDiff[i][ps][pb] = 1Dhuff_dec(hcod1D_ICC_Diff,bsCodeW); } } } } tempExtraFrame=numSlots/(bsResidualFramesPerSpatialFrame+1); for (rf=0; rf<bsResidualFramesPerSpatialFrame; rf++) individual_channel_stream(0); if (window_sequence == EIGHT_SHORT_SEQUENCE) && ((tempExtraFrame == 18) (tempExtraFrame == 24) (tempExtraFrame == 30)) { individual_channel_stream(0); } } } } </pre>	<p>1</p> <p>1..7</p>	<p>Note 6</p> <p>Note 2 Uimbsf</p> <p>vlclbf Note 3</p> <p>Note 4</p> <p>Note 1 Note 5</p> <p>Note 1</p>
<p>Note 1: individual_channel_stream(0) according to MPEG-2 AAC Low Complexity profile bitstream syntax described in subclause 6.3 of ISO/IEC 13818-7.</p> <p>Note 2: numParamSets is defined by numParamSets = bsNumParamSets + 1.</p> <p>Note 3: 1Dhuff_dec() is defined in Annex A.1.</p> <p>Note 4: numSlots is defined by numSlots = bsFrameLength + 1. Furthermore the division shall be interpreted as ANSI C integer division.</p> <p>Note 5: individual_channel_stream(0) determines the value of window_sequence.</p> <p>Note 6: resReorder[k] for 0 ≤ k < 6 is the following constant integer array: {0, 1, 2, 5, 3, 4}.</p>		

In subclause 5.1, in Table 36, Note 5, replace:

profile bitstream syntax

by:

profile bitstream syntax

In subclause 5.2, in Table 41, in the second row, replace:

default fine quantization for CLD, ICC, CPC

by:

default fine quantization for CLD, ICC, CPC, **ATD**

In subclause 5.2, replace:

bsTttBandsHigh Same as bsTttBandsLow but for high band range. The high band range is $\text{bsTttBandsLow} \leq \text{pb} < \text{bsTttBandsHigh}$.

by:

bsTttBandsHigh Helper variable, same as bsTttBandsLow but for high band range. The high band range is $\text{bsTttBandsLow} \leq \text{pb} < \text{bsTttBandsHigh}$.

In subclause 5.2, remove the definition of bsHRTFicc and bsHRTFiccLR.

In subclause 5.2, Table 54 in the heading, replace:

bsSacExtTyp

by:

bsSacExtType

In subclause 5.2, Table 71, replace:

XXX (dataType)	numQuantStepsXXXCoarse	numQuantStepsXXXFine
CLD	16	31
ICC	4	8
CPC	26	51

by:

XXX (dataType)	numQuantStepsXXXCoarse	numQuantStepsXXXFine
CLD, ATD	15	31
ICC	4	8
CPC	26	51

In subclause 5.2, Table 72, replace:

numQuantSteps	maxGrpLen
3	5
6	5
7	6
11	2
13	4
19	4
25	3
51	4
any other value	1

by:

numQuantSteps	maxGrpLen
3	5
7	6
11	2
13	4
19	4
25	3
51	4
any other value	1

In subclause 5.2, Table 76, replace:

LavIdx	lavTabCLD [LavIdx]	lavTabICC [LavIdx]	lavTabCPC [LavIdx]
0	3	1	3
1	5	3	6
2	7	5	9
3	9	7	12

by:

LavIdx	lavTabCLD [LavIdx], lavTabATD [LavIdx]	lavTabICC [LavIdx]	lavTabCPC [LavIdx]
0	3	1	3
1	5	3	6
2	7	5	9
3	9	7	12

In subclause 5.2, remove the definition of *hcodPilot_XXX*

In subclause 5.2, replace definition of *hcodFirstBand_XXX*:

One-dimensional Huffman code (Table A.2, Table A.3, and Table A.4) used for coding of data the data type of which is determined by the value of XXX. It is applied for coding of the lowest frequency band whenever differential coding in frequency direction is applied.

by:

One-dimensional Huffman code (Table A.2, Table A.3, and Table A.4) used for coding of data the data type of which is determined by the value of XXX. It is applied for coding of the lowest frequency band whenever **pilot-based coding** or differential coding in frequency direction is applied.

In subclause 5.2, replace:

bsXXXpcm[pi][ps][pb] PCM coded indices where XXX is to be replaced by CLD, ICC, or CPC. The indices are offset so that they cannot be negative.

bsXXXmsbDiff[pi][ps][pb] Differentially coded most significant bits of a quantization index of data type XXX (where XXX can be either CLD or ICC or CPC) belonging to parameter index pi, parameter set ps and parameter band pb.

bsXXXlsb[pi][ps][pb] Least significant bit of a quantization index of data type XXX [...]. May only be 1 in case of data type **CLD** and fine quantization; otherwise always 0.

idxXXX[pi][ps][pb] Quantized spatial parameter (as index, can be negative) for the pi:th XXX parameter for the ps:th parameter set ($0 \leq ps < \text{numParamSets}$) and the pb:th parameter band ($0 \leq pb < \text{numBands}$). XXX is to be replaced by **CLD**, **ICC**, or **CPC**.

by:

bsXXXpcm[pi][ps][pb] PCM coded indices where XXX is to be replaced by **CLD**, **ICC**, **CPC**, or **ATD**. The indices are offset so that they cannot be negative.

bsXXXmsbDiff[pi][ps][pb] Differentially coded most significant bits of a quantization index of data type XXX (where XXX can be **CLD**, **ICC**, **CPC**, or **ATD**) belonging to parameter index pi, parameter set ps and parameter band pb.

bsXXXlsb[pi][ps][pb] Least significant bit of a quantization index of data type XXX (where XXX can be **CLD**, **ICC**, **CPC**, or **ATD**). May only be 1 in case of data type **CPC** and fine quantization; otherwise always 0.

idxXXX[pi][ps][pb] Quantized spatial parameter (as index, can be negative) for the pith XXX parameter for the ps:th parameter set ($0 \leq ps < \text{numParamSets}$) and the pbth parameter band ($0 \leq pb < \text{numBands}$). XXX is to be replaced by **CLD**, **ICC**, **CPC**, or **ATD**.

In subclause 5.2, directly after the description of 'individual_channel_stream()' add a description for 'channel_pair_element()', including a new Table 79A:

MPEG-2 AAC Low Complexity profile channel_pair_element() according to the syntax defined in Table 14 (and related tables) of subclause 6.3 of ISO/IEC 13818-7. Decoding of a channel_pair_element() also determines the value of **window_sequence**, according to ISO/IEC 13818-7.

A second channel_pair_element() is present when **window_sequence** (determined by the first channel_pair_element()) equals **EIGHT_SHORT_SEQUENCE** and **tempExtraFrame** equals 18, 24 or 30. In this case, the value of **window_sequence** determined by the second channel_pair_element() shall equal **EIGHT_SHORT_SEQUENCE**.

Restrictions apply to the elements of the channel_pair_element() syntax. The restriction applied to the elements of the channel_pair_element() syntax is given in Table 79A.

Table 79A – Restrictions in syntax of channel_pair_element()

Tool	Allowed value
common_window	1

Furthermore, the restrictions applied to the elements ics_info(), defined in Table 15 of subclause 6.3 of ISO/IEC 13818-7, and individual_channel_stream(), defined in Table 16 of subclause 6.3 of ISO/IEC 13818-7, as sub-elements of the channel_pair-element(), are given in Table 77 and Table 78. In addition to the existing restriction on the maximum number of scalefactor bands to which Temporal Noise Shaping is applied, (the constant **TNS_MAX_BANDS** defined in Table 33 of subclause 7.1.6 of ISO/IEC 13818-7), the lowest scalefactor band where Temporal Noise Shaping is applied is restricted and depends on the sampling rate (derived from **bsResidualSamplingFrequencyIndex**) and whether **window_sequence** indicates a long or short window, as specified in Table 79.

In subclause 5.2, Table 81, replace:

numInChan					
1		2		6	
numAacEl	AacEl	numAacEl	AacEl	numAacEl	AacEl
1	'0'	1	'1'	2	'111'

by:

numInChan					
1		2		6	
numAacEl	AacEl	numAacEl	AacEl	numAacEl	AacEl
1	'0'	1	'1'	3	'111'

In subclause 6.1, replace the heading:

6.1 Compressed data stream decoding and de-quantization

by:

6.1 Bitstream decoding and de-quantization

In subclause 6.1.2, replace the heading:

6.1.2 Decoding of CLD, ATD, ICC, CPC, and arbitrary downmix gain parameters

by:

6.1.2 Decoding of CLD, ATD, ICC, CPC, and arbitrary downmix gain parameters

In subclause 6.1.2.1, replace:

pi = parameter instance having values in the range which, for CLD, ICC and CPC have the range 0 .. numOttBoxes+4*numTttBoxes+numInChan-1, and for ATD have the range 0 .. numOttBoxesAT-1

by:

pi = parameter instance having values in the following ranges:

- CLD: 0..numOttBoxes+4*numTttBoxes+numInChan-1
- ICC: 0..numOttBoxes+2*numTttBoxes-1
- CPC: 0..4*numTttBoxes-1
- ATD: 0..numOttBoxesAT-1

In subclause 6.1.2.1, last paragraph, replace:

The following process described in the subclauses below is carried out for all instances pi of all parameter types CLD, ATD, ICC, CPC, where XXX denotes the actual parameter type.

by:

The following process described in the subclauses below is carried out for all instances pi of all parameter types CLD, ATD, ICC, CPC, where XXX denotes the actual parameter type. Note that, if the bsOneIcc flag is set to one, one coherence information (ICC) of the original audio signal applicable to all OTT boxes is contained within the parametric representation of the decoder tree; in particular, for 5-1-5 configurations only one coherence information of the original audio signal is transmitted within the bitstream.