ICS 33.160.30
CCS M 72

团　体　标　准

T/CAIACN 00X-2023

低延迟低复杂度高清音频编解码技术规范

Low Latency Low Complexity High Resolution Audio Codec Technology Specification

（公开征求意见稿）

2023- XX - XX 发布　　2023- XX - XX 实施

中国电子音响行业协会
星闪无线短距通信联盟　发布

目　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》给出的规定起草。

中国电子音响行业协会（CHINA AUDIO INDUSTRY ASSOCIATION）自1983年成立以来就以“服务企业，献策政府”为宗旨。是我国最早成立的跨地区、跨部门、跨系统，具有社团法人资格的国家一级行业协会之一。

组织开展电子音响领域国际、国内标准化活动，制定中国电子音响行业协会团体标准（以下简称：中音协团标），满足行业需要，推动行业标准化工作，是中国电子音响行业协会的重要工作。协会的所有会员，均有权利提出制、修订中音协团标的建议并参与有关工作。

中音协团标按《中国电子音响行业协会团体标准建设管理办法》进行制定和管理。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

在本标准实施过程中，如发现需要修改或补充之处，请将意见和有关资料报送中国电子音响行业协会，以便修订时参考。

本文件由中国电子技术标准化研究院提出。

本文件由中国电子音响行业协会归口。

本文件由中国电子音响行业协会与国际星闪无线短距通信联盟共同发起成立的“无线音频编解码”联合工作组制定。本文件主要起草单位：（待补充）

本文件主要起草人：（待补充）。

本文件为首次制定。

低延迟低复杂度高清音频编解码技术规范

1 范围

本文件规定了低延迟低复杂度高清音频编解码技术标准音频编码的位流表示方式及解码过程。

本文件适用于无线音频、实时通信、广播流媒体、网络电视、虚拟现实和增强现实、视频监控等领域。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 9002-2017 音频、视频和视听设备及系统词汇

GB/T 5271.1 信息技术 词汇 第1部分：基本术语（GB/T 5271.1-2000 eqv ISO/IEC 2382-1：1993）

GB/T 5271.4 信息技术 词汇 第4部分：数据的组织（GB/T 5271.4-2000 eqv ISO/IEC 2382-4：1987）

GB/T 5271.9 信息技术 词汇 第9部分：数据通信（GB/T 5271.9-2001 eqv ISO/IEC 2382-9：1995）

3 术语和定义

GB/T 9002-2017、GB/T 5271.1、GB/T 5271.4和GB/T 5271.9界定的以及下列术语和定义适用于本文件。

3.1

编码 coding

读入音频采样值，并产生一个符合本文件的有效位流。

3.2

解码 decoding

在本文件中定义的一种数据处理，即读入编码位流并输出音频采样值的过程。

3.3

编码位流 coded bitstream

音频信号的编码表示。

3.4

边信息 side information

位流中控制解码的必要信息。

3.5

声道 channel

用于传送到单个扬声器或其他重放设备的一组有序音频样本集合。

3.6

双声道立体声 stereo audio

一种音频格式，该格式下，使用两个声道承载有一定相位关系的音频信号，通常通过位于听音者前方的两个对称的扬声器或使用耳机重放，带给听音者更宽的声场感觉。

4 缩略语

下列缩略语适用于本文件。

MDCT 改进离散余弦变换 modified discrete cosine transform

L2HC 低延迟低复杂度高清音频编解码 low latency low complexity high resolution audio Codec

CBR 恒定比特流 constant bitrate

5 约定

5.1 概述

本文件中使用的数学运算符和优先级与 C 语言使用的类似，但对整型除法和算术移位操作进行了特定的定义。除特别说明外，约定编号和计数从 0 开始。

5.2 算数运算符

算术运算符定义见表 1 表 1。

表1 算术运算符定义

算术运算符	定义
+	加法运算
-	减法运算（二元运算符）或取反（一元前缀运算符）
×	乘法运算
*	乘法运算
a^b	幂运算，表示a的b次幂。也可表示上标
pow(a，b)	幂运算，表示a的b次幂
/	除法运算，不做截断或四舍五入

表 1　算术运算符定义（续）

算术运算符	定义
÷	除法运算，不做截断或四舍五入
$\frac{a}{b}$	除法运算，不做截断或四舍五入
$\sum_{i=a}^{b} f(i)$	自变量i取由a到b（含b）的所有整数值时，函数f(i)的累加和
$\lfloor \cdot \rfloor$	下取整
$\sqrt{a}$	a的算术平方根
$\|\|\cdot\|\|_2$	2-范数

5.3　逻辑运算符

逻辑运算符定义见表 2。

表2　逻辑运算符定义

逻辑运算符	定义
||	逻辑或
&&	逻辑与
!	逻辑非

5.4　关系运算符

关系运算符定义见表 3。

表3　关系运算符定义

关系运算符	定义
>	大于
≥	大于或等于
<	小于
≤	小于或等于
==	等于
≠	不等于

5.5　位运算符

位运算符定义见表 4。

表4　位运算符定义

位运算符	定义
&	与运算
|	或运算
~	取反运算
$a >> b$	将以2的补码形式表示的整数a向右移b位。仅当b取正数时定义此运算。向右移至最高有效位时，其值与a移位运算前的最高有效位相等
$a << b$	将以2的补码形式表示的整数a向左移b位。仅当b取正数时定义此运算。向左移至最低有效位时，其值等于0

5.6　赋值

赋值运算符定义见表 5。

表5　赋值运算符定义

赋值运算	定义
=	赋值运算符
++	自加，x+1相当于$x = x$+1。当用于数组下标时，在自加运算前先求变量值
+=	自加指定值，例如，x += 3相当于$x = x$ + 3，x += (-3)相当于$x = x$+ (-3)
-=	自减指定值，例如，x -= 3相当于$x = x$+ (-3)，x -= (-3)相当于$x = x$ - (-3)

5.7　助记符

助记符定义见表 6。

表6　助记符定义

助记符	定义
rpchof	多项式余数，高阶在先
bslbf	位串，左位在前。位串是带单引号的1和0串。如‘1000 0001’。位串内的空格是便于阅读的，无特殊意义。（bitstream left bit first）
uimsbf	无符号整数，最高有效位优先。（unsigned integer, most significant bit first）
bsmbf	位串，带单引号的1和0串，右位在前，如先编码一个5bit的数值6，然后编码一个3bit的数值2，那么编码位串为‘010 00110’

5.8　数学函数

数学函数定义见如下公式。

$$|x| = \begin{cases} x & ; \ x > 0 \\ 0 & ; \ x = 0 \\ -x & ; \ x < 0 \end{cases}$$

式中：

x——自变量。

5.9 位流语法规则

位流中的每一个数据项用粗体字，通过名字、按位的长度及其类型和传输顺序的助记符来描述。

位流中被解码的数据元素所导致的操作依赖于该数据的值及以前解码的数据元素。如无特殊说明，本文件中的“位”指二进制位。

本文件语法用“C”代码规定，变量或表达式为非零值时等价于条件为真，变量或表达式为零值时等价于条件为非真。

```
while(condition){
  data_element
  …
}
```

若条件为真，则数据元素组紧接着数据流产生，如此重复直到条件为非真。

```
do{
  data_element
  …
} while(condition)
```

若条件为真，则数据元素组紧接着数据流产生，如此重复直到条件为非真。

```
if(condition){
  data_element
  …
} else{
  data_element
  …
}
```

若条件为真，在数据流中产生第一组数据元素，若条件为非真，在数据流中产生第二组数据元素。

```
for(expr1;expr2;expr3){
  data_element
  …
}
```

expr1是指定循环初始状态表达式，通常它指定了计数器的初始状态；expr2是指定的每次循环前的测试条件，条件为非真时循环终止；expr3是每次循环结束时执行的表达式，一般是增加计数器。

最通常用法为：

```
for(i=0;i<n;i++){
  data_element
  …
}
```

数据元素组产生 n 次。数据元素组内的条件结构可能依赖循环控制变量 i 的值。第一次出现时被置为‘0’，第二次增加到‘1’，如此往复。

根据表达式 expr 的值，产生对应的数据元素。expr 的值为 constcase1 时产生数据元素 data_element1，expr 的值为 constcase2 时产生数据元素 data_element2，以此类推，expr 的值为 constcasen 时产生数据元素 data_elementn。当 expr 的值不等于 constcase1，constcase2，…，constcasen 中的任何一个值时，产生数据元素 data_elementdefault。

```
switch(expr){
  case constcase1:
     data_element1
     break
  case constcase2:
     data_element2
     break
     …
  case constcasen:
     data_elementn
     break
  default:
     data_elementdefault
     break
}
```

本结构的一类变体是在 case 后不出现 break,expr 的值 constcasex 时,从对应的 case constcasex 开始产生数据元素，直到 break 出现。expr 的值为 constcase1 时产生数据元素 data_element1 和 data_element2，expr 的值为 constcasen 时产生数据元素 data_elementn。

```
switch(expr){
  case constcase1:
     data_element1
  case constcase2:
     data_element2
     break
     …
  case constcasen:
     data_elementn
     break
  default:
     data_elementdefault
     break
}
```

数据元素组中可能含有嵌套结构。为简便起见，当后面只有一个数据元素时“[]”省略。数据元素组表示方法如下：

data_element[ ]：一数组数据，数据元素的个数由上下文而定。

data_element[n]：数组数据的第 n＋1 个元素。

data_element[m][n]：二维数组的第 m＋1，n+1 个元素。

data_element[l][m][n]：三维数组的第 l+1，m+1，n+1 个元素。

data_element[m...n]：位 m 到位 n 之间包括的位。

虽然语法用过程项表示，但不能认为条款实现了可靠的解码过程。它只是定义了一个无错误的位流输入。

6 L2HC 音频编解码

6.1 概述

本文件规定了面向低延迟、低复杂度、高保真的基于 MDCT（Modified Discrete Cosine Transform）实现的频域音频编解码器，L2HC（Low Delay Low Complexity High Resolution Audio Codec）。

L2HC 编解码器包括单声道编解码、立体声（双声道）编解码。L2HC 支持单声道、立体声音频编解码，编码模式为 CBR（Constant BitRate）模式，单声道编码码率范围为 64kbps 到 960kbps，立体声编码码率范围为 128kbps 到 1920kbps。在码率支持的范围内，编码码率可以在编码端任意设置，最小步长变化为 1kbps。输入/输出的音频信号采样率支持 44.1kHz、48kHz、88.2kHz 和 96kHz。位深支持 16 位定点，24 位定点和 32 位定点/浮点，编码器和解码器位深可以不同。

L2HC 编解码器支持帧长 5ms 和 10ms。10ms 帧长条件下，在 48kHz 和 96kHz 采样率下的算法时延为 12.5ms，look ahead 为 2.5ms，采样率 48kHz 和 96kHz 对应的帧长分别为 480 点和 960 采样点；在 44.1kHz 和 88.2kHz 采样率下的算法时延为 13.6ms，其中帧长为 10.88ms，look ahead 为 2.72ms，采样率 44.1kHz 和 88.2kHz 对应的帧长分别为 480 点和 960 点。5ms 帧长条件下，在 48kHz 和 96kHz 采样率下的算法时延为 7.5ms，look ahead 为 2.5ms，采样率 48kHz 和 96kHz 对应的帧长分别为 240 点和 480 采样点；在 44.1kHz 和 88.2kHz 采样率下的算法时延为 8.16ms，其中帧长为 5.44ms，look ahead 为 2.72ms，采样率 44.1kHz 和 88.2kHz 对应的帧长分别为 240 点和 480 点。

L2HC 编解码器在解码端需要配置的参数包括：解码信号位深和解码模式。在处理立体声信号时，L2HC 支持 3 种解码模式,分别为左声道输出，右声道输出和立体声输出。

6.2 节介绍 L2HC 编解码器框架，6.3 节介绍 L2HC 位流数据格式，6.4 节描述边信息解码流程，6.5 节描述频域噪声整形参数的获取方法，6.6 节描述 MDCT 系数的熵解码方法，6.7 节描述残余解码方法，6.8 节描述 MDCT 系数的逆量化方法，6.9 节描述立体声信号的上混方法，6.10 节描述逆时频变换方法，6.11 节介绍 L2HC 位流格式，6.12 节介绍 L2HC 信号输入/输出位深处理方法。附录 A 包含子带包络划分表、Huffman 码表等。

6.2 L2HC 编解码器框架

L2HC 音频编码器框架如图 1 所示，L2HC 音频编码器可分为时频变换、信号自适应分析、立体声下混、子带包络参数获取与编码、频域噪声整形参数获取、MDCT 系数量化、熵编码和残余编码等模块。

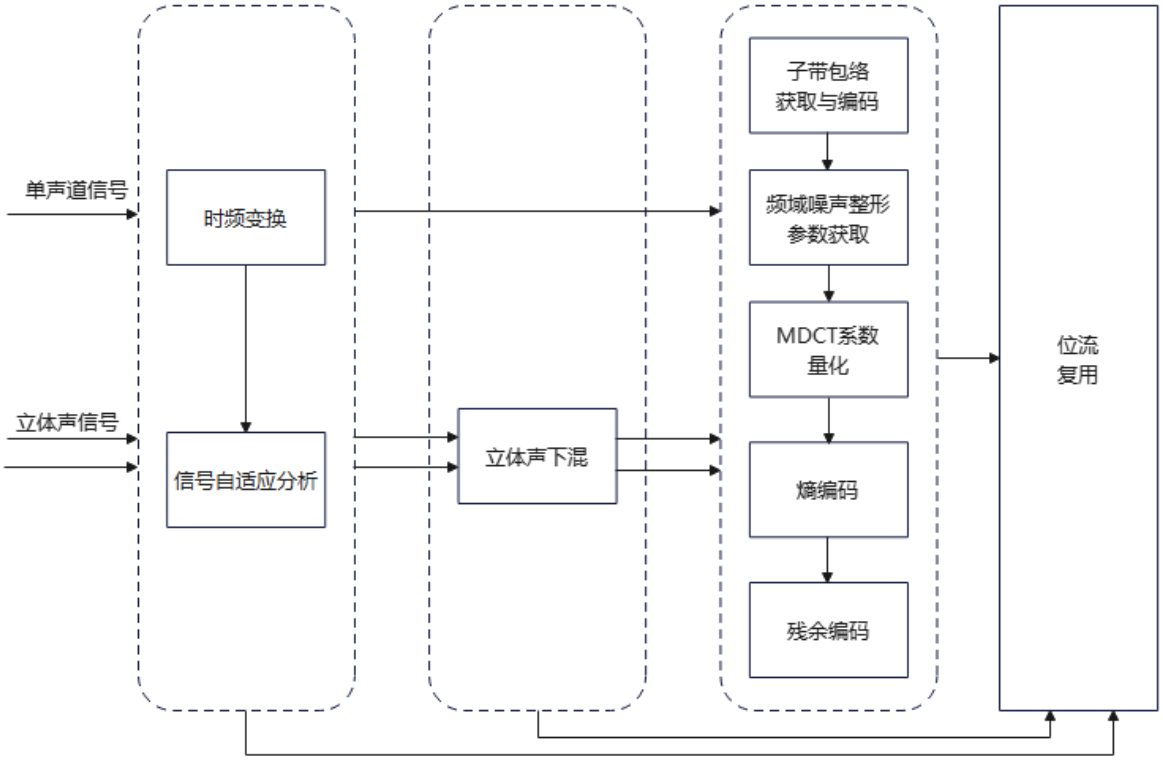

图1　L2HC 音频编码器框架

L2HC 音频解码器框架如图 2 所示，L2HC 音频解码器可分为边信息解码、频域噪声整形参数获取、熵解码、残余解码、MDCT 系数逆量化、立体声上混和逆时频变换。

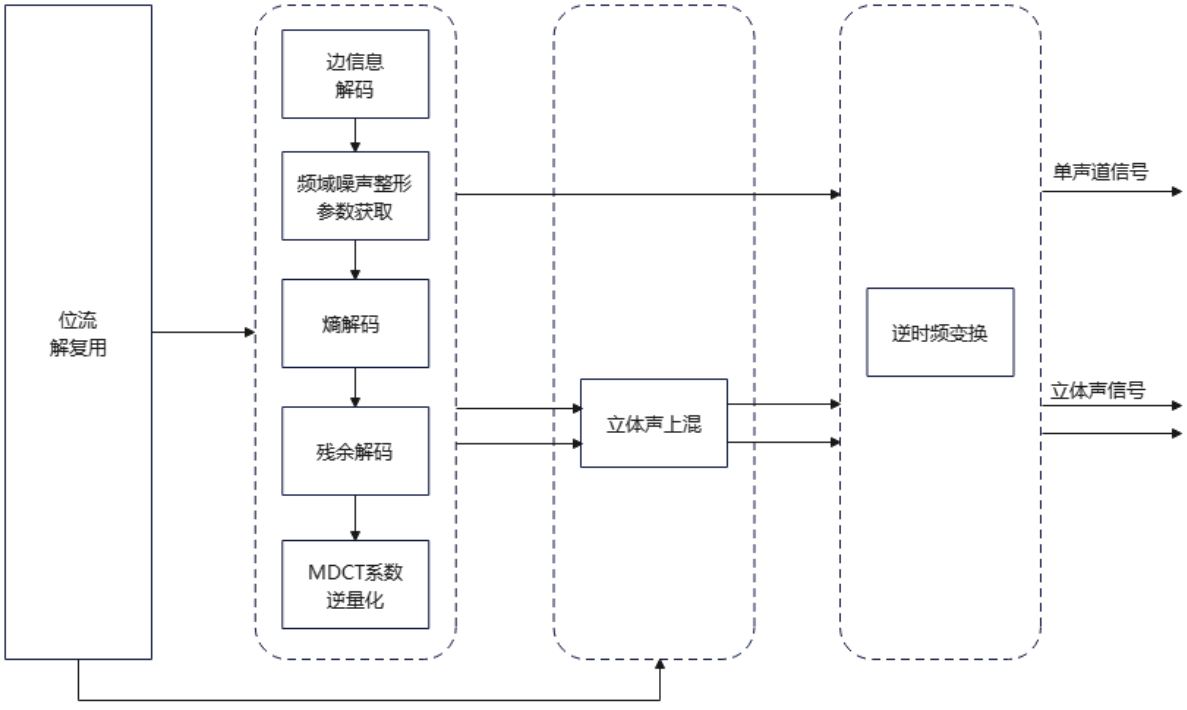

图2　L2HC 音频解码器框架

以下分别对输入信号为单声道和立体声的情况进行介绍。

6.2.1 L2HC 单声道编码

L2HC 在编码单声道信号时，信号需经过时频变换、信号自适应分析、子带包络参数获取与编码、频域噪声整形参数获取、MDCT 系数量化、熵编码和残余编码等模块处理。下面对各模块功能进行简要介绍。

6.2.1.1 时频变换

时频变换模块对输入的时域信号加窗并进行 MDCT 变换。

6.2.1.2 信号自适应分析

对输入信号进行带宽检测，结合码率确定子带划分情况，输出边信息包括：需要编码的子带个数 bandNum、子带划分方式索引 sfId、低码率标识符 lowBrFlag 等，并写入位流。

6.2.1.3 子带包络获取与编码

根据子带划分情况计算各个子带的包络信息并进行编码。编码对象可以是子带包络参数或子带包络参数的差分值。

6.2.1.4 频域噪声整形参数获取

利用子带包络和子带间掩蔽关系计算频域噪声整形参数，包括心理声学谱包络参数 psyScalefactor 和量化等级参数 quantScale。

6.2.1.5 MDCT 系数量化

利用心理声学谱包络参数 psyScalefactor 求得子带的噪声基底 noisefloor，对各子带的 MDCT 系数进行量化。其中，量化精度数值上等于噪声基底 noisefloor。

6.2.1.6 熵编码

对量化后的单声道信号的 MDCT 系数进行熵编码，并写入位流。

6.2.1.7 残余编码

利用剩余比特进行残余编码，提升 MDCT 系数的编码精度，相关信息写入位流。

6.2.2 L2HC 单声道解码

L2HC 在解码单声道信号时，位流需经过边信息解码、频域噪声整形参数获取、熵解码、残余解码、MDCT 系数逆量化和逆时频变换，最终获得解码单声道信号。下面对各模块功能进行简要介绍。

6.2.2.1 边信息解码

从位流中获取与边信息相关的位流信息，进行解码得到低码率标识符 lowBrFlag、心理声学谱包络系数动态调节因子 dr、辅助调节因子 drQuater、子带划分方式索引 sfId、编码子带个数 bandNum、子带包络差分编码开启标志 diffFlag、子带包络参数和 Huffman 解码码表索引等。

6.2.2.2 频域噪声整形参数获取

利用解码获取的子带包络信息和子带间掩蔽关系得到各子带的频域噪声整形因子 dr_{adjust}。结合心理声学谱包络系数动态调节因子 dr 和辅助调节因子 drQuater 求得频域噪声整形参数，包括心理声学谱包络参数 psyScalefactor 和量化等级参数 quantScale。

6.2.2.3 熵解码

从位流中获取与 MDCT 系数编码相关的位流信息，结合 Huffman 码表进行熵解码得到量化后的 MDCT 系数。

6.2.2.4 残余解码

从位流中获取残余编码相关的位流信息，进行残余解码。

6.2.2.5 MDCT 系数逆量化

MDCT 系数逆量化模块是由心理声学谱包络参数 psyScalefactor 获取噪声基底 noisefloor。然后利用 noisefloor 参数对量化的 MDCT 系数进行逆量化得到解码的 MDCT 系数。

6.2.2.6 逆时频变换

逆时频变换模块将 MDCT 系数变换到时域，并进行加窗和叠接相加操作，得到解码的时域音频信号。

6.2.3 L2HC 立体声编码

L2HC 在编码立体声信号时，信号需经过时频变换、信号自适应分析、立体声下混、子带包络参数获取与编码、频域噪声整形参数获取、MDCT 系数量化、熵编码和残余编码等模块处理。

与 L2HC 单声道编码相比增加了立体声下混模块，立体声下混模块根据立体声信号特征对左右声道 MDCT 系数进行 M/S 下混，以去除信号间冗余信息，提升编码效率。

6.2.4 L2HC 立体声信号解码

L2HC 在解码立体声信号时，位流需经过边信息解码、频域噪声整形参数获取、熵解码、残余解码、MDCT 系数逆量化、M/S 上混和逆时频变换等模块处理，最终获得解码立体声信号。

与 L2HC 单声道解码流程相比，立体声解码增加了立体声上混模块，立体声上混模块是编码端立体声下混的逆过程，对解码所得下混声道的 MDCT 系数进行 M/S 上混，得到立体声的 MDCT 系数。

6.3 L2HC 位流数据

6.3.1 语法

L2HC 位流数据语法见表 7。

表7 L2hcDecodeRawData()语法

L2hcDecodeRawData()语法	比特数	助记符
L2hcDecodeRawData(){	—	—
L2hcHeaderUnpack()	—	—
switch(codecFormat) {	—	—
case 0x0: L2hcCommonDec(chNumDec=chNum)	—	—

表 7　L2hcDecodeRawData()语法（续）

L2hcDecodeRawData()语法	比特数	助记符
case 0x1: L2hcHighCostDec()	—	—
}	—	—
PcmCopyChannel()	—	—
}	—	—

包头信息解码见表 8。

表8　L2hcHeaderUnpack()语法

L2hcHeaderUnpack()语法	比特数	助记符
L2hcHeaderUnpack(){	—	—
codecType	2	uimsbf
sampleRate	2	uimsbf
chNum	1	uimsbf
chNum = chNum + 1	—	—
frameLength	2	uimsbf
}	—	—

单声道/立体声普通解码语法见表 9。

表9　L2hcCommonDec()语法

L2hcCommonDec()语法	比特数	助记符
L2hcCommonDec(chNumDec) {	—	—
DecodeCoreSideBits(chNumDec)	—	—
GetMdctSnsParam(chNumDec)	—	—
for (ch = 0; ch < chNumDec; ch++) {	—	—
DecodeMdctSpec()	—	—
}	—	—
if (bitLeft) {	—	—
DecodeMdctRes()	—	—
}	—	—
chBegin, chEnd = GetDecodeChRange(chNum, decMode)	—	—
if (msFlag == 1) {		
for (ch = 0; ch < chNumDec; ch++) {		
QuantInvMdct()		
}		
MdctInvMS()		
} else {		

表 9　L2hcCommonDec()语法（续）

L2hcCommonDec()语法	比特数	助记符
for (ch = chBegin; ch < chEnd; ch++) {	—	—
QuantInvMdct()	—	—
}	—	—
}	—	—
for (ch = chBegin; ch < chEnd; ch++) {	—	—
InverseMdct()	—	—
}	—	—
}	—	—

立体声高清低功耗解码模式语法见表 10。

表10　L2hcHighCostDec()语法

L2hcHighCostDec()语法	比特数	助记符
L2hcHighCostDec() {	—	—
MovPosByDecMode()	—	—
chNumDec = 1	—	—
if (decMode == 0 || decMode == 1) {	—	—
L2hcCommonDec(chNumDec)	—	—
}else{	—	—
L2hcCommonDec(chNumDec)	—	—
L2hcCommonDec(chNumDec)	—	—
}	—	—
}	—	—

解码通道复制语法见表 1 表 11。

表11　PcmCopyChannel()语法

PcmCopyChannel()语法	比特数	助记符
PcmCopyChannel() {	—	—
if (chNum == 2) {	—	—
if (decMode == 0){	—	—
CopyPcmL2R()	—	—
}	—	—
if (decMode == 1){	—	—
CopyPcmR2L()	—	—
}	—	—
}	—	—
}		

解码通道索引范围获取语法见表 1 表 12。

表12　GetDecodeChRange()语法

GetDecodeChRange()语法	比特数	助记符
GetDecodeChRange() {	—	—
chBegin = 0	—	—
chEnd = chNum	—	—
if (chNum == 2){	—	—
if (decMode == 0){	—	—
chBegin = 0	—	—
chEnd = 1	—	—
}	—	—
if (decMode == 1){	—	—
chBegin = 1	—	—
}	—	—
}	—	—
}	—	—

6.3.2　语义

L2hcHeaderUnpack()

解析包头信息，包括帧类型、采样率、声道数、帧长。

L2hcCommonDec()

单声道/立体声普通解码。

L2hcHighCostDec()

立体声高清低功耗解码。

PcmCopyChannel()

解码通道复制，用于仅解码 L 通道或 R 通道的情况。

DecodeCoreSideBits()

解析边信息。

GetMdctSnsParam()

频域噪声整形参数获取。

DecodeMdctSpec()

MDCT 系数解码。

DecodeMdctRes()

残余解码。

GetDecodeChRange()

获取解码通道索引范围。

QuantInvMdct()

MDCT 系数逆量化。

MdctInvMS()

M/S 上混。

InverseMdct()

MDCT 逆变换。

MovPosByDecMode()

位流读取位置跳转，decMode 等于 0 或者 2 时位流读取位置不跳转，decMode 等于 1 时位流读取位置跳转至右声道位流起始位置。

CopyPcmL2R()

将左声道信号复制到右声道。

CopyPcmR2L()

将右声道信号复制到左声道。

codecFormat

编码格式，0 代表单/立体声普通格式，1 代表立体声高清低功耗格式。

codecType

2 比特,代表编码器类型；当前版本设为 1。0,2,3 作为保留字段。

sampleRate

2 比特，用于表示音频信号的采样率，取值范围为 0、1、2、3，对应的采样率依次为 44.1kH、48kHz、88.2kHz、96kHz。

chNum

1 比特，用于表示音频信号的声道数，0 代表单声道，1 代表立体声。

chNumDec

用于表示实际解码的声道数，取值小于或等于 chNum。

frameLength

2 比特，用于表示音频信号的帧长,取值范围为 0、1、2、3，对应的帧长依次为 120 采样点、240 采样点、480 采样点、960 采样点。

decMode

立体声解码模式，通过解码器命令行配置，取值范围为 0、1、2,分别对应左声道输出，右声道输出和立体声输出。

bitLeft

是否有比特剩余标识符，当标识符为 0 时，表示比特已全部解码，当标识符为 1 时，表示比特有剩余，需要进行残余解码。

chBegin

解码通道索引范围的起始值。

chEnd

解码通道索引范围的终止值。

6.3.3 解码过程

L2HC 解码器获取一帧音频位流信息并进行解码，解码过程可参照 L2hcDecodeRawData()，基本过程描述如下：

解码器首先解析包头信息，获得编码器类型、编码采样率、编码通道数和帧长等信息。

codecFormat 参数需要根据编码通道数、编码采样率和码率综合判断获得，判断条件为：当同时满足编码通道数为 2（即立体声信号编码），编码码率大于等于 300kbps，编码采样率大于等于 88.2kHz 三

个条件时，codecFormat 为 1，否则 codecFormat 为 0。其中，编码码率由通讯协议上层获得的接收帧的包大小计算得到。

codecFormat 为 0 时，编码器选择单声道/立体声普通格式，解码过程表示为 L2hcCommonDec()。首先解码核心编码器边信息；而后参照 GetMdctSnsParam()获取频域噪声整形参数；然后参照 DecodeMdctSpec()解码 MDCT 系数；如果仍然有比特剩余，则参照 DecodeMdctRes()进行残余解码；根据 chNum 和 decMode 获取解码通道索引的范围。当 msFlag 等于 1 时，参照 QuantInvMdct()依次对 chNumDec 个解码通道进行 MDCT 系数逆量化操作，后进行立体声上混操作；当 msFlag 不等于 1 时，根据解码通道索引范围起始值 chBegin 和终止值 chEnd，依次对解码通道进行 MDCT 系数逆量化操作；最后参照 InverseMdct()进行逆时频变换，得到解码单声道/立体声信号。

codecFormat 为 1 时，编码器选择立体声高清低功耗格式，解码过程可参照 L2hcHighCostDec()。首先参照函数 MovPosByDecMode()进行位流读取位置跳转操作，并将通道数 chNumDec 设为 1。而后根据解码器命令行配置的解码模式 decMode 参数进行不同的解码处理。当 decMode 为 0 时，参照 L2hcCommonDec(chNumDec)进行单声道普通解码来获得左声道的解码信号，并将左声道的 PCM 值赋值给右声道。当 decMode 为 1 时，参照 L2hcCommonDec(chNumDec)进行单声道普通解码来获得右声道的 PCM 值，并将右声道的解码信号赋值给左声道。当 decMode 为 2 时，参照 L2hcCommonDec(chNumDec)进行单声道普通解码来分别获取左声道和右声道的解码信号。

完成上述流程后可参照 PcmCopyChannel()， 根据解码声道数 chNum 和立体声解码模式 decMode 进行解码通道复制，最后输出解码的立体声信号。

6.4 边信息解码

6.4.1 语法

边信息解码语法见表 13。

表13 DecodeCoreSideBits()语法

DecodeCoreSideBits()语法	比特数	助记符
DecodeCoreSideBits(chNumDec) {	—	—
lowBrFlag	1	uimsbf
if (chNumDec == 2) {	—	—
msFlag	1	uimsbf
}else{	—	—
msFlag = 0	—	—
}	—	—
dr	5	uimsbf
drQuater	3	uimsbf
sfId	3	uimsbf
if (lowBrFlag) {	—	—
bandNum	4	uimsbf
bandNum += 16	—	—
}else{	—	—
bandNum = 32	—	—
}	—	—

表 13　DecodeCoreSideBits()语法（续）

DecodeCoreSideBits()语法	比特数	助记符
diffFlag	1	uimsbf
for (ch = 0; ch < chNumDec; ch++) {	—	—
DecodeScaleFactor()	—	—
}	—	—
for (ch = 0; ch < chNumDec; ch++) {	—	—
hufTupId_4tuple[ch]	2	uimsbf
hufTupId_2tuple[ch]	2	uimsbf
hufTupId_1tuple[ch]	2	uimsbf
}	—	—
}	—	—

子带包络解码语法见表 14。

表14　DecodeScaleFactor()语法

DecodeScaleFactor()语法	比特数	助记符
DecodeScaleFactor() {	—	—
if (diffFlag == 0) {	—	—
for (b = 0; b < bandNum; b++) {	—	—
sfValueNonDiff	5	uimsbf
sf[b] = sfValueNonDiff - 8	—	—
}	—	—
}else{	—	—
sfValueNonDiff	5	
sf[0] = sfValueNonDiff - 8	—	—
for (b = 1; b <bandNum; b++) {	—	—
sfValueDiff	注1	uimsbf
if (sfValueDiff != 0){	—	—
sfSign	1	uimsbf
if (sfSign == 0){	—	—
sf[b] = sf[b-1] - sfValueDiff	—	—
} else {	—	—
sf[b] = sf[b-1] + sfValueDiff	—	—
}	—	—
}else{	—	—
sf[b] = sf[b-1]	—	—
}	—	—
}	—	—
}	—	—
注： sfValueDiff采用Huffman编码，比特数不固定		

6.4.2 语义

lowBrFlag

1 比特，低码率模式标识符，0 代表高码率模式，1 代表低码率模式。

msFlag

1 比特，用于表示当前帧是否开启 M/S 立体声处理，0 代表不开启，1 代表开启。

dr

5 比特，心理声学谱包络系数动态调节因子，范围为[-8,23]，用于调节心理声学谱包络参数。

drQuater

3 比特，辅助调节因子，用于进一步调节心理声学谱包络参数和量化等级参数。

sfId

3 比特，用于获取子带划分方式。

bandNum

4 比特，编码子带个数。

diffFlag

1 比特，子带包络参数的差分编码标志，0 代表使用非差分编码，1 代表使用差分编码。

DecodeScaleFactor()

解析子带包络信息。

hufTupId_4tuple[ch]

2 比特，第 ch 个声道 MDCT 系数编码所用 4 tuple 编码方式对应的 Huffman 码表索引，N tuple 编码方式参见 6.6.3。

hufTupId_2tuple[ch]

2 比特，第 ch 个声道 MDCT 系数编码所用 2 tuple 编码方式对应的 Huffman 码表索引。

hufTupId_1tuple[ch]

2 比特，第 ch 个声道 MDCT 系数编码所用 1 tuple 编码方式对应的 Huffman 码表索引。

sfValueNonDiff

5 比特，子带包络参数编码值（非差分编码）。

sfValueDiff

子带包络参数差分值的绝对值，采用 Huffman 编码。

sfSign

1 比特，子带包络参数差分值的符号位，符号位为 0 表示差分值为负值，为 1 表示差分值为正值。

6.4.3 解码过程

6.4.3.1 概述

DecodeCoreSideBits()从位流中解析核心解码器边信息,包括低码率标识符、立体声下混开启标志、动态调节因子和辅助调节因子、编码子带个数、子带包络差分编码开启标志、子带包络和 Huffman 解码码表索引，用于后续的解码模块。

6.4.3.2 子带划分方式获取

子带划分方式的获取由帧长、采样率、低码率标识符 lowBrFlag 和子带划分方式索引 sfId 共同决定。

当帧长为 10ms，采样率为 88.2kHz/96kHz，lowBrFlag 为 0 时，选择 960 点帧长高码率子带划分表，见表 A.1。

当帧长为 10ms，采样率为 88.2kHz/96kHz，lowBrFlag 为 1 时，选择 960 点帧长低码率子带划分表，见表 A.2。

当帧长为 10ms，采样率为 44.1kHz/48kHz，lowBrFlag 为 0 时，选择 480 点帧长高码率子带划分表，见表 A.3。

当帧长为 10ms，采样率为 44.1kHz/48kHz，lowBrFlag 为 1 时，选择 480 点帧长低码率子带划分表，见表 A.4。

当帧长为 5ms，采样率为 88.2kHz/96kHz，lowBrFlag 为 0 时，选择 480 点帧长高码率子带划分表，见表 A.3。

当帧长为 5ms，采样率为 88.2kHz/96kHz，lowBrFlag 为 1 时，选择 480 点帧长高码率子带划分表，见表 A.4。

当帧长为 5ms，采样率为 44.1kHz/48kHz，lowBrFlag 为 0 时，选择 240 点帧长高码率子带划分表，见表 A.5。

当帧长为 5ms，采样率为 44.1kHz/48kHz，lowBrFlag 为 1 时，选择 240 点帧长低码率子带划分表，见表 A.6。

选定子带划分表后，进一步根据 sfId 选择对应的子带划分方式。例如，选定表 A.3 为当前帧子带划分表时，若 sfId 为 2，则选定表 A.3 中索引为 2 的子带划分表，作为当前帧的子带划分方式。

上述定义的子带划分方式中，最大子带数量为 32 个。实际编码子带个数 bandNum 小于或等于 32，bandNum 的最小值为 16。

将子带索引标记为 b，b = 0, 1, 2, …, bandNum-1，则第 b 个子带的起始和终止频点序号分别记为 bandstart[b]、bandend[b]，即当前子带的频点范围为[bandstart[b],bandend[b])。例如，对表 A.3 中索引为 0 的子带划分方式，第 0 个子带的起始频点序号为 0，终止频点序号为 1，表明第 0 个子带仅包括频点 0；第 31 个子带的起始频点序号为 160，终止频点序号为 480，表明第 31 个子带共包含 320 个频点。

6.4.3.3 子带包络解码

子带包络的编码方式分为差分编码和非差分编码，可由子带包络参数的差分编码标志 diffFlag 进行判断，若 diffFlag 为 0 代表使用非差分编码，1 代表使用差分编码。子带包络解码的基本过程如下：

1）使用非差分编码时，子带包络解码过程如下：

每个子带包络参数的编码值占 5 比特，记为 sfValueNonDiff，将此值减 8 得到子带包络参数解码值：

$$sf[b] = sfValueNonDiff - 8$$

2）使用差分编码时，子带包络解码过程如下：

首先解码第 0 个子带的子带包络参数 sf(0)，方式与非差分编码时相同。

而后，对后续子带（即 b>0），利用 Huffman 解码来获取子带包络参数差分值的绝对值 sfDiffValue。解码子带包络使用的 Huffman 码表参见附录 A 表 A.7。解码过程为将位流中的码字与 Huffman 码表中码字的取值和长度进行匹配，从而得到解码结果。

解码符号位：当 sfDiffValue 不为 0 时，则读取 1 比特作为子带包络参数差分值的符号位（sfSign），若符号位为 1 则子带包络参数的差分值为正值，符号位为 0 则差分值为负值；当 sfDiffValue 为 0 时，则不需要解码符号位。

根据子带包络参数差分值的绝对值 sfDiffValue、子带包络参数差分值的符号位 sfSign，以及上一个子带的子带包络参数 sf [b-1]，可以得到当前子带的子带包络参数 sf [b]，过程如下：

若差分值的绝对值 sfDiffValue 不为零，差分值的符号位 sfSign 为 0，则：

$$sf[b] = sf[b-1] - sfDiffValue, b = 1,2 \ldots bandNum - 1$$

若差分值的绝对值 sfDiffValue 不为零，差分值的符号位 sfSign 为 1，则：

$$sf[b] = sf[b-1] + sfDiffValue, b = 1,2 \ldots bandNum - 1$$

若差分值的绝对值 sfDiffValue 为零，则：

$$sf[b] = sf[b-1], b = 1,2 \ldots bandNum - 1$$

6.5 频域噪声整形参数获取

6.5.1 语法

频域噪声整形参数获取语法见表15。

表15 GetMdctSnsParam()语法

GetMdctSnsParam()语法	比特数	助记符
GetMdctSnsParam(chNumDec){	—	—
if(chNumDec == 2){	—	—
MdctSns1()	—	—
}else{	—	—
MdctSns2()	—	—
}	—	—
GetPsyScale()	—	—
}	—	—

6.5.2 语义

MdctSns1()

用于立体声解码的频域噪声整形因子获取。

MdctSns2()

用于单声道解码的频域噪声整形因子获取。

GetPsyScale()

心理声学谱包络参数和量化等级参数获取。

6.5.3 解码过程

6.5.3.1 概述

频域噪声整形参数根据信号子带包络信息，结合心理声学模型的子带间掩蔽效应、子带内对基底噪声的掩蔽效应计算得到，用于 MDCT 系数解码和 MDCT 系数逆量化。

频域噪声整形参数包括：心理声学谱包络参数 psyScalefactor 和量化等级参数 quantScale。

频域噪声整形参数获取的主要步骤包括：频域噪声整形因子（dr_{adjust}）获取、心理声学谱包络参数（psyScalefactor）和量化等级参数（quantScale）获取。以下分别进行介绍。

6.5.3.2 频域噪声整形因子获取

频域噪声整形因子的获取根据编码码率分为两种情况：高码率情况下，设置 dradjust 参数为 0，在低码率模式下（lowBrFlag=1），按下述逻辑确定 dr_{adjust} 参数。

频域噪声整形分为两种实现策略，包括保守平稳的整形策略，记为 MdctSns1()，以及激进锐化的整形策略，记为 MdctSns2()。以下分别进行描述。

1）针对立体声编码，采用保守平稳的整形策略，即 MdctSns1()。

首先，根据解码所得的各子带包络参数求出子带包络均值 sf_{avg}（以下描述省略声道标号 ch，下同）：

$$sf_{avg} = \frac{1}{bandNum} \sum_{b=0}^{bandNum-1} sf[b] + 1$$

其中，bandNum 为编码子带个数。

考虑到各子带包络的掩蔽关系，重新计算获取考虑掩蔽效应的子带包络参数 sf1：

$$sf1[b] = sf[b] + c * MAX(sf[b-1] - sf[b], 0) + c * MAX(sf[b+1] - sf[b], 0)$$

其中，c 为掩蔽系数，获取方式如下：

$$c = \begin{cases} 0.375, sf[b] > sf_{avg} \\ 0.25, sf[b] \le sf_{avg} \end{cases}$$

特别地，对第 0 个子带和第 bandNum-1 个子带：

$$sf1[0] = sf[0] + c * MAX(sf[1] - sf[0], 0)$$

$$sf1[bandNum-1] = sf[bandNum-1] + c * MAX(sf[bandNum-2] - sf[bandNum-1], 0)$$

根据 sf1 计算考虑掩蔽效应的子带包络均值 $sf1_{avg}$：

$$sf1_{avg} = \frac{1}{bandNum} \sum_{b=0}^{bandNum-1} sf1[b] + 1$$

根据 $sf1_{avg}$ 得到调整后的子带包络参数 sf_{new}[b]：

$$sf_{new}[b] = \begin{cases} sf1_{avg} + 0.375 * \left(sf1[b] - sf1_{avg}\right), sf1[b] > sf1_{avg} \\ sf1_{avg} + 0.5 * \left(sf1[b] - sf1_{avg}\right), sf1[b] \le sf1_{avg} \end{cases}$$

最后，根据调整后的子带包络参数 sf_{new}[b]和子带包络参数 sf[b]得到频域噪声整形因子 dr_{adjust}[b]：

$$dr_{adjust}[b] = sf[b] - sf_{new}[b]$$

2）针对单声道编码，采用激进锐化的整形策略，即 MdctSns2()。

首先，基于心理声学模型对解码所得的各子带包络参数 sf[b]进行平滑，得到调整后的子带谱包络系数 sf_{new}[b]：

$$sf_{new}[b] = sf[b] + c1 * max(sf[b-1] - sf[b], 0) + c2 * max(sf[b+1] - sf[b], 0)$$

特别地，第 0 个子带的 sf_{new} 表示为：

$$sf_{new}[0] = sf[0] + c2 * MAX(sf[1] - sf[0], 0)$$

最后一个子带（第 bandNum-1 个子带）的 sf_{new} 表示为：

$$sf_{new}[bandNum-1] = sf[bandNum-1] + c1 * max(sf[bandNum-2] - sf[bandum-1], 0)$$

其中，c1，c2 是掩蔽系数，取值为 c1=0.175，c2=0.125。

然后，对调整后的谱包络参数进行能量增强，以子带为单元，累进增加 18dB：

$$sf_{inc}[b] = sf_{new}[b] + 3b/(bandNum-1), b = 0,1,2,..,bandNum-1$$

当前帧的增强谱包络参数最大值记为 sf_{max}，各增强子带谱包络系数与 sf_{max} 做差，得到噪声整形因子 $dr_{adjust}[b]$：

$$dr_{adjust}[b] = sf_{inc}[b] - sf_{max}$$

6.5.3.3 心理声学谱包络参数和量化等级参数获取

根据各子带包络 sf[b]、心理声学谱包络系数动态调节因子 dr，以及噪声整形因子 $dr_{adjust}[b]$进行以下操作，获取心理声学谱包络参数（记为 psyScalefactor）：

$$psyScalefactor[b] = \max(min(sf[b], dr + dr_{adjust}), sfmin), b = 0,1,2,..,bandNum-1$$

其中，sfmin 的取值为-8。

根据子带包络参数和心理声学谱包络参数可获取 MDCT 系数的量化等级参数 quantScale：

$$quantScale[b] = max(sf[b] - psyScalefactor[b], 0), b = 0,1,2,..,bandNum-1$$

而后根据解码边信息获取的辅助调节因子 drQuater 进一步调节心理声学谱包络参数 psyScalefactor 和量化等级参数 quantScale。

若 drQuater 等于 0，则无需调整。若 drQuater 不为 0 时，调节流程如下：

1）计算总的有效频点个数 freqNum：即所有 quantScale[b]>0 的子带的频点总个数。

2）计算压缩区间 freqNumScale，等于 freqNum*drQuater/8。

3）若低码率标识符 lowBrFlag=1，即为低码率模式，调节过程为：

按从高频到低频的顺序对各个子带进行处理(即子带序号 b 从 bandNum-1 到 0)。当 quantScale[b]>0 时，将 quantScale[b]减 1，psyScaleFactor[b]加 1，freqNumScale 减去 bandWidth[b]。重复上述操作直到遍历完所有子带或者 freqNumScale 小于 0 时退出。

其中，bandWidth[b]表示第 b 个子带带宽，可通过边信息 sfId 查找表 A.1，表 A.3， 表 A.5（对应高码率模式）或者表 A.2，表 A.4，表 A.6（对应低码率模式）获得。

4）若低码率标识符 lowBrFlag=0，即高码率模式，调节过程为：

首先根据 quantScale 对 sf 做预处理，得到权重数组 w。伪代码如下：

```
GetWeight(sfLeft, sfCenter, sfRight, quantScale[b]) {
    w = ((sfCenter << 1) - (sfLeft + sfRight)) << 2
    w += (min(max(sfCenter - 10, 0), 10) << 2)
    w -= (quantScale[b] << 2) + (quantScale[b] << 1)
    w *= (quantScale[b] + 2)
}
```

其中，第 b 个子带的 sfLeft, sfCenter, sfRight 参数的获取方式如下：

```
sfLeft = (b == 0) ? sf[0] - 1 : sf[b - 1]
sfCenter = sf[b]
sfRight = (b == bandNum-1) ? sf[b] - 1 : sf[b + 1]
```

而后进行以下处理：获取权重数组 w 中最小值所对应的子带索引 b；将 quantScale[b]减 1，psyscalefactor[b]加 1，freqNumScale 减去 bandWidth[b]，bandWidth[b]表示第 b 个子带带宽；结合新的 quantScale[b]对 sf[b]做预处理，更新权重数组 w，更新方式如 GetWeight()所示；重复上述操作，直到 freqNumScale 小于 0 时退出。

6.6 MDCT 系数熵解码

6.6.1 语法

MDCT 系数熵解码语法见表 16。

表16 DecodeMdctSpec()语法

DecodeMdctSpec()语法	比特数	助记符
DecodeMdctSpec(){	—	—
if (lowBrFlag == 1){	—	—
UnPack1TupThreeSign()	—	—
UnPack2TupTwoSign()	—	—
UnPack4TupOneSign()	—	—
} else {	—	—
UnPack1TupThree()	—	—
UnPack2TupTwo()	—	—
UnPack4TupOne()	—	—
UnPackSign()	—	—
}	—	—
}	—	—

6.6.2 语义

UnPack1TupThreeSign()

解码按1 tuple方式编码的子带的MDCT系数，包括MDCT系数的幅值和符号位，适用于quantScale = 3和quantScale > 3的子带。

UnPack2TupTwoSign()

解码按2 tuple方式编码的子带的MDCT系数，包括MDCT系数的幅值和符号位，适用于quantScale = 2的子带。

UnPack4TupOneSign()

解码按4 tuple方式编码的子带的MDCT系数，包括MDCT系数的幅值和符号位，适用于quantScale = 1的子带。

UnPack1TupThree()

解码按1 tuple方式编码的子带的MDCT系数的幅值，适用于quantScale = 3和quantScale > 3的子带。

UnPack2TupTwo()

解码按2 tuple方式编码的子带的MDCT系数的幅值，适用于quantScale = 2的子带。

UnPack4TupOne()

解码按4 tuple方式编码的子带的MDCT系数的幅值，适用于quantScale = 1的子带。

UnPackSign()

解码MDCT系数的符号位。

6.6.3 解码过程

在MDCT系数编码过程中，每个子带的编码方式由子带的量化等级参数quantScale的值来决定，一共有三种编码方式，即quantScale>=3对应1 tuple编码方式、quantScale等于2对应2 tuple编码方式、quantScale等于1对应4 tuple编码方式。

1 tuple编码方式为将一个MDCT系数的幅值或者MDCT系数的幅值的高三位打包成一个符号，并使用Huffman码表进行编码，不足3bit的值高位补0。2 tuple编码方式为将二个MDCT系数的幅值打包成一个符号，并使用Huffman码表进行编码，不足2bit的值高位补0。4 tuple编码方式为将四个MDCT系数的幅值打包成一个符号，并使用Huffman码字进行编码。

对MDCT系数的符号位编码时，将非0的MDCT系数的符号位写入位流，每个符号位使用1比特，符号位为0代表负值，符号位为1代表正值。

6.6.3.1 Huffman 解码码表获取

由边信息解码所得的hufTupId_1tup作为索引来选择1 tuple的Huffman码表，索引变量的取值为[0,1,2,3]，1 tuple码表索引和Huffman码表的对应关系如表17所示。

表17 1 tuple 码表索引和 Huffman 码表的对应关系

hufTupId_1tup	0	1	2	3
Huffman码表	表A.16	表A.17	表A.18	表A.19

由边信息解码所得的 hufTupId_2tup 作为索引来选择 2 tuple 的 Huffman 码表，索引变量的取值为[0,1,2,3]，2 tuple 码表索引和 Huffman 码表的对应关系如表 18 所示。

表18 2 tuple 码表索引和 Huffman 码表的对应关系

hufTupId_2tup	0	1	2	3
Huffman码表	表A.12	表A.13	表A.14	表A.15

由边信息解码所得的 hufTupId_4tup 作为索引来选择 4 tuple 的 Huffman 码表，索引变量的取值为[0,1,2,3]，4 tuple 码表索引和 Huffman 码表的对应关系如表 19 所示。

表19 4 tuple 码表索引和 Huffman 码表的对应关系

hufTupId_4tup	0	1	2	3
Huffman码表	表A.8	表A.9	表A.10	表A.11

6.6.3.2 MDCT 系数解码

MDCT 系数的熵解码按每个子带 quantScale 的取值分为四组，依次进行，解码顺序为：所有 quantScale 为 3 的子带、所有 quantScale 大于 3 的子带、所有 quantScale 为 2 的子带、所有 quantScale 为 1 的子带。

1）量化等级参数 quantScale 为 3 的 MDCT 系数值解码

对 quantScale = 3 的子带，获得 1 tuple Huffman 码表后，依次从位流中读取比特并和码表中码字的值和长度进行匹配，值和长度均匹配的码字的索引即为解码结果，解码结果的取值范围为[0, 7]。

2）量化等级参数 quantScale 大于 3 的 MDCT 系数值解码

对 quantScale>3 的子带，获得 1 tuple Huffman 码表后，依次从位流中读取比特并和码表中的码字的值和长度进行匹配，值和长度均匹配的码字的索引作为解码结果的高 3 位，继续从位流中读取 quantScale-3 个比特，作为解码结果的低位，上述高 3 位和低 quantScale-3 位构成当前 MDCT 系数的解码结果。

3）量化等级参数 quantScale 为 2 的 MDCT 系数值解码

对 quantScale = 2 的子带，获得 2 tuple Huffman 码表后，依次从位流中读取比特并和码表中码字的值和长度进行匹配，值和长度均匹配的码字的索引即为解码结果。解码结果为 4 比特的值，将其拆分为 2 个 2 比特的值，拆分方式为高 2 位作为当前 2 tuple 中的第一个解码值，低 2 位作为第二个解码值。

4）量化等级参数 quantScale 为 1 的 MDCT 系数值解码

对 quantScale = 1 的子带，获得 4 tuple Huffman 码表后，依次从位流中读取比特并和码表中码字的值和长度进行匹配，值和长度均匹配的码字的索引即为解码结果。解码结果为 4 比特的值，将其从高位到低位依次拆分为 4 个 1 比特的值，作为当前 4 tuple 中第一个、第二个、第三个、第四个解码值。

5）MDCT 系数的符号位解码

MDCT 系数的符号位信息在不同码率下对应不同的解码方式。

在低码率模式标识符 lowBrFlag 为 1 时，即低码率模式下，符号位信息是在每个 tuple 后获取的，得到 tuple 对应的解码值后，统计解码值中不为 0 的个数，然后依次从位流中获取符号位，每个符号位占用 1 比特。若符号位为 0，则对应的 MDCT 系数为负值，若符号位为 1，则对应的 MDCT 系数为正值。以 4 tuple 为例，假设四个解码值均不为 0，则从位流中获取四个符号位进行判断与赋值。

在低码率模式标识符 lowBrFlag 为 0 时，即在高码率模式下，符号位信息是在解完所有的 MDCT 系数后获取的（即所有 tuple 解码完成后）。根据获取的 MDCT 系数的解码值是否为 0 进行判断，若 MDCT 系数的解码值为 0，则不需解码符号位。统计出 MDCT 系数的解码值不为 0 的个数，即需要解码的符号位个数，依次从位流中获取符号位，每个符号位占用 1 比特。若符号位为 0，则对应的 MDCT 系数为负值，若符号位为 1，则对应的 MDCT 系数为正值。

6.7 残余解码

6.7.1 语法

残余解码语法见表 20。

表20 DecodeMdctRes()语法

DecodeMdctRes()语法	比特数	助记符
DecodeMdctRes(){	—	—
for (b = 0; b < bandNum; b++) {	—	—
for (ch = 0; ch < chNumDec; ch++){	—	—
if (quantScale[ch][b] == 0) {	—	—
for (freq = bandStart[b]; freq <= bandEnd[b]; freq++){	—	—
resValue1	1	uimsbf
mdct[ch][freq] = 0.5 * resValue1	—	—
if (mdct[ch][freq] !=0){	—	—
resSign	1	uimsbf

表 20　DecodeMdctRes()语法（续）

DecodeMdctRes()语法	比特数	助记符
mdct[ch][freq] *= (resSign * 2 - 1)	—	—
}	—	—
}	—	—
} else if (quantScale[ch][b] > 0){	—	—
if (quantScale[ch][b] == 1) {	—	—
frac = 0.125	—	—
}else{	—	—
frac = 0	—	—
}	—	—
for (freq = bandStart[b]; freq <= bandEnd[b]; freq++){	—	—
resValue2	1	uimsbf
mdct[ch][freq] += (frac + resValue2 * 0.5 - 0.25)	—	—
}	—	—
}	—	—
}	—	—
}	—	—
}	—	—

6.7.2　语义

RankBandMdctRes()

对子带包络及索引进行降序排序操作。

resValue1

1 比特,当 quantScale 等于 0 时的残余解码值。

resSign

1 比特,当 quantScale 等于 0 时残余解码符号位。

resValue2

1 比特,当 quantScale 大于 0 时的残余解码值。

bandStart[b]

第 b 个子带的起始频点序号。

bandEnd[b]

第 b 个子带的终止频点序号。

freq

MDCT 频点序号。

6.7.3　解码过程

根据 bitLeft 的值来决定是否进行残余解码，bitLeft 的获取方式为从协议上层获得的包大小减去已经解码的比特数，如果还有剩余，则 bitLeft 为 1，否则为 0。bitLeft 为 1 时需要进行残余解码直到解码比特数用完。残余解位流程如下。

首先，查找子带包络的最大值和对应的子带索引，分别记为 peakSf、peakBand；对子带包络及索引进行降序排列，得到排序 rank 数组，过程如下；

计算获取每个声道各个子带和子带包络最大值所在子带的距离，计算公式为

$$band[i]=\begin{cases}bandNum-\ abs(b-peakBand), if\ quantScale[ch][b]\le 0\\ -abs(b-peakBand), if\ quantScale[ch][b]>0\end{cases}$$

其中，i = b*chNumDec + ch，ch 取值为 0 或 1，b 取值范围为[0, bandNum-1]。

将 band[i]和索引 i 配对，以 band[i]为参考进行降序排序，得到 rank 数据结构。rank[i].value 对应排序后的 band 数组的值，rank[i].index 对应 band 数组的索引。

根据排序结果对应的索引值,从前往后重新进行残余解码,重复以下步骤,直至残余比特解码完毕。

从前往后读取 rank[i].index，得到声道索引 ch 和子带索引 b 为

$$ch=rank[i].index\%chNumDec$$

$$b=rank[i].index/chNumDec$$

当 quantScale[ch][b]=0 时，对当前子带内的 MDCT 系数依次进行残余解码。解码方式为对子带内的每个 MDCT 系数，从位流中读取 1 比特（记为 resValue1），如果 resValue1 为 1，则 MDCT 系数的残余解码的幅值为 0.5，如果 resValue1=0，MDCT 系数的残余解码的幅值为 0。由于 quantScale[ch][b]为 0 的子带 MDCT 系数未编码，故 MDCT 系数的残余解码的幅值即作为解码 MDCT 系数的幅值，即 mdct[ch][freq]。

若 mdct[ch][freq]不为 0，则继续从位流中读取 1 比特作为符号位 resSign，若 resSign 为 0，则 mdct[ch][freq]为负值，若 resSign 为 1，则 mdct[ch][freq]为正值。若 mdct[ch][b]为 0 时，则不需要解码符号位。

当 quantScale[ch][b]>0 时，对当前子带内的 MDCT 系数依次进行残余解码，解码方式为对子带内的每个 MDCT 系数从位流中读取 1 比特（记为 resValue2），得到 MDCT 系数的残余解码值：

$$mdctRes[ch][freq]=frac+resValue2*0.5-0.25$$

上式中，frac 的值由所在子带的量化等级参数 quantScale 确定，方式为

$$frac=\begin{cases}0.125, if\ quantScale[ch][b]=1\\ 0, if\ quantScale[ch][b]\ne 1\ and\ quantScale[ch][b]>0\end{cases}$$

将残余解码值与 MDCT 系数解码值相加得到 mdct[ch][freq]为

$$mdct[ch][freq]\ +=mdctRes[ch][freq]$$

6.8 MDCT 系数逆量化

6.8.1 解码过程

由心理声学包络系数 psyScalefactor 获取子带的底噪（以下描述省略声道标号 ch）：

$$NoiseFloor[b]=\ 2^{psyScalefactor[b]}, b=0,1,2,..,bandNum-1$$

其中，bandNum 为编码子带数。

熵解码和残余解码步骤获取的 MDCT 系数，经 MDCT 系数逆量化处理可得到解码的 MDCT 系数：

$$mdct[freq]*=NoiseFloor[b]$$

上式中， b 为子带包络索引，b 取值范围为[0, bandNum-1]，freq 为每个子带内 MDCT 系数的频点序号，取值范围为[bandstart[b], bandend[b])，其中 bandstart[b]、bandend[b]分别为每个子带的起始和终止频点序号。

6.9 立体声上混

6.9.1 解码过程

立体声上混是编码端立体声下混的逆过程，当编码信号为立体声时，根据位流中获取的下混标志msFlag来决定当前帧是否进行上混操作。

若当前帧msFlag为1，则需要对解码得到的下混声道的MDCT系数进行M/S上混处理，以获得解码的左右声道MDCT系数。

M/S上混处理如下式：

$$mdct\_L = \frac{1}{2}(mdct\_M + mdct\_S)$$

$$mdct\_R = \frac{1}{2}(mdct\_M - mdct\_S)$$

其中，mdct_M和mdct_S分别为两个下混声道的解码MDCT系数，mdct_L和mdct_R分别为M/S上混后得到的左右声道的解码MDCT系数。

6.10 逆时频变换

6.10.1 解码过程

逆时频变换模块的作用是对解码得到的各声道的MDCT系数，进行逆MDCT变换，将输出的分帧信号进行加窗与叠接相加，获得重建的时域音频信号。

帧长为10ms时，当输入信号的采样率为44.1kHz和48kHz时，窗长为960采样点。当输入信号的采样率为88.2kHz和96kHz时，窗长为480采样点。

帧长为5ms时，当输入当输入信号的采样率为44.1kHz和48kHz时，窗长为240采样点。当输入信号的采样率为88.2kHz和96kHz时，窗长为480采样点。

6.11 位流格式

6.11.1 概述

编码器处理单声道信号时，按照单声道位流格式进行打包。位流结构如表21所示。

单声道位流格式包含三部分。第一部分为包头位流，如表22所示。第二部分为边信息位流，如表23所示。其中子带包络(sf)编码比特数不固定，记为nBit_sf（参考6.5.3.2 节）。第三部分为Payload位流，如表24所示。其中量化的MDCT系数使用Huffman编码，所用比特数不固定，记为nBits_Payload。

表23中标灰值bandNum仅在低码率模式（lowBrFlag=1）下传递。

表21 单声道位流格式

Header	Side Info	Payload

表22 包头位流格式

Header				
字段	Codec Type	Sample Rate	Channel Number	Frame Length
比特数	2	2	1	2

表23　边信息位流格式

Side Info				
字段	lowBrFlag	dr	drQuater	sfId
比特数	1	5	3	3
Side Info				
字段	bandNum	diffFlag	sf	hufTupID
比特数	4	1	nBits_sf	6

表24　Payload 位流格式

Payload		
字段	mdctQ	mdctRes
比特数	nBits_Payload	

编码器处理立体声信号时，涉及两种位流打包格式：立体声交织位流和立体声非交织位流。

当 codecFormat 等于 0 时，使用立体声交织位流，位流结构如表 25 所示。立体声交织位流格式包含三部分，第一部分为包头位流，复用表 22，第二部分为交织格式的边信息位流，如表 26 所示，第三部分为交织格式的 Payload 位流，如表 27 示。表 26 中标灰值 bandNum 仅在低码率模式（lowBrFlag=1）下传递。

当 codecFormat 等于 1 时，使用立体声非交织位流，位流结构如表 28 所示。立体声非交织位流格式其包含五部分，第一部分为包头位流，复用表 22，第二部分为左声道的边信息位流，复用表 23，第三部分为左声道的 Payload 位流，复用表 24，第四部分为右声道的边信息位流，复用表 23，第五部分为右声道的 Payload 位流，复用表 24。

表25　立体声交织位流格式

Header	Side Info(common)	Payload(L,R)

表26　立体声交织格式边信息位流

Side Info(common)					
字段	lowBrFlag	msFlag	dr	drQuater	sfId
比特数	1	1	5	3	3
Side Info(common)					
字段	bandNum	diffFlag	sf		hufTupID
比特数	4	1	nBits_sf(L, R)		12

表27　立体声交织格式 Payload 位流

Payload(L,R)		
字段	mdctQ(L,R)	mdctRes(L,R)
比特数	nBits_Payload(L, R)	

表28　立体声非交织位流格式

Header	Side Info(L)	Payload(L)	Side Info(R)	Payload(R)

6.12　输入/输出位深处理方法

L2HC 编解码器输入/输出的位深处理方法为：

编码侧，统一将整型 16/24/32 比特位深和浮点 32 比特位深转换成 24 比特位深数据进行编码压缩。

解码侧，根据解码器输出位深配置，将解码获得的 24 比特位深音频数据，转换为整型 16/24/32 比特位深和浮点 32 比特位深的音频数据。

设解码获得的音频信号为 sigDec，则解码器输出音频信号 sigOut 的位深转换方式描述如下：

若解码器输出位深设定为整型 16 比特，则解码器输出音频信号 sigOut 表示为：

$$sigOut[i] = min(max\left(\left\lfloor\frac{sigDec[i]}{256} + 0.5 * sign[i]\right\rfloor, -32767\right), 32767)$$

其中，$\lfloor*\rfloor$表示向下取整，sigOut[i]为解码器输出音频信号第 i 个采样点的值，sigDec[i]为解码获得音频信号第 i 个采样点的值，sign[i]为第 i 个采样点的符号位，获取方式如下：

$$sign[i] = \begin{cases} 1, & if\ sigDec[i] > 0 \\ -1, & if\ sigDec[i] \leq 0 \end{cases}$$

若解码器输出位深设定为整型 24 比特，则解码器输出音频信号 sigOut 表示为：

$$sigOut[i] = \min(\max(\lfloor sigDec + 0.5 * sign[i]\rfloor, -8388607), 8388607)$$

若解码器输出位深设定为整型 32 比特，则解码器输出音频信号 sigOut 表示为：

$$sigOut[i] = \min(\max(\lfloor 256 * sigDec[i] + 0.5 * sign[i]\rfloor, -2147483647), 2147483647)$$

若解码器输出位深设定为浮点 32 比特，则解码器输出音频信号 sigOut 表示为：

$$sigOut[i] = \min(\max\left(\frac{sigDec[i]}{8388607}, -1\right), 1)$$

附 录 A
（规范性）
音频编码表

表A.1 960 点帧长高码率子带划分表

索引	子带划分
0	0, 2, 4, 6, 8, 10, 12, 14, 16, 20, 24, 28, 32, 38, 44, 52, 60, 70, 80, 90, 100, 114, 128, 146, 164, 184, 204, 224, 248, 272, 296, 320, 960
1	0, 2, 4, 6, 8, 10, 14, 18, 22, 26, 30, 36, 42, 48, 56, 66, 76, 88, 100, 114, 128, 146, 164, 186, 208, 232, 256, 280, 310, 340, 370, 400, 960
2	0, 2, 4, 6, 8, 12, 16, 20, 26, 32, 40, 48, 56, 66, 76, 90, 104, 122, 140, 158, 176, 200, 224, 254, 284, 320, 356, 392, 434, 476, 518, 560, 960
3	0, 2, 4, 8, 12, 20, 28, 36, 44, 52, 60, 68, 84, 100, 116, 132, 148, 168, 192, 216, 240, 272, 304, 336, 384, 432, 480, 544, 608, 672, 752, 848, 960
4	0, 2, 4, 8, 12, 20, 28, 36, 52, 68, 84, 100, 124, 148, 172, 196, 224, 256, 288, 320, 352, 392, 432, 472, 512, 560, 608, 656, 704, 768, 832, 896, 960
5	0, 160, 184, 208, 224, 240, 256, 272, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360, 368, 376, 384, 392, 400, 416, 432, 448, 464, 480, 496, 512, 536, 560, 960
6	0, 400, 424, 448, 464, 480, 496, 512, 528, 536, 544, 552, 560, 568, 576, 584, 592, 600, 608, 616, 624, 632, 640, 656, 672, 688, 704, 720, 736, 752, 776, 800, 960
7	0, 640, 664, 688, 712, 728, 744, 760, 768, 776, 784, 792, 800, 808, 816, 824, 832, 840, 848, 856, 864, 872, 880, 888, 896, 904, 912, 920, 928, 936, 944, 952, 960

表A.2 960 点帧长低码率子带划分表

索引	子带划分
0	0, 1, 2, 3, 4, 6, 8, 10, 13, 16, 20, 24, 28, 33, 38, 45, 52, 61, 70, 79, 88, 100, 112, 127, 142, 160, 178, 196, 217, 238, 259, 280, 480
1	0, 1, 2, 3, 5, 7, 9, 12, 15, 18, 22, 26, 30, 35, 41, 48, 56, 65, 74, 84, 94, 106, 118, 134, 150, 166, 184, 202, 220, 240, 260, 280, 480
2	0, 1, 2, 3, 4, 5, 7, 9, 11, 14, 17, 21, 25, 29, 34, 40, 46, 52, 60, 68, 76, 86, 98, 110, 126, 144, 162, 180, 200, 224, 250, 280, 480
3	0, 2, 4, 6, 8, 12, 16, 21, 26, 31, 36, 41, 46, 51, 56, 61, 66, 71, 77, 83, 89, 95, 103, 111, 121, 131, 147, 163, 179, 203, 240, 280, 480
4	0, 1, 2, 3, 5, 7, 9, 12, 15, 19, 23, 27, 32, 37, 43, 49, 57, 66, 76, 86, 98, 110, 125, 140, 158, 176, 194, 216, 238, 264, 290, 320, 480
5	0, 1, 2, 3, 5, 7, 10, 13, 17, 21, 25, 30, 35, 41, 47, 54, 62, 70, 80, 90, 102, 114, 130, 146, 162, 180, 198, 218, 240, 264, 290, 320, 480
6	0, 1, 2, 4, 6, 8, 11, 14, 18, 22, 26, 30, 36, 42, 50, 58, 66, 76, 88, 100, 112, 128, 144, 160, 182, 204, 226, 256, 286, 316, 352, 400, 480

表 A.2　960 点帧长低码率子带划分表（续）

索引	子带划分
7	0, 1, 2, 4, 6, 8, 11, 14, 18, 22, 26, 30, 36, 42, 50, 58, 68, 78, 90, 102, 116, 132, 148, 166, 186, 208, 234, 262, 292, 324, 360, 400, 480

表A.3　480 点帧长高码率子带划分表

索引	子带划分
0	0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, 19, 22, 26, 30, 35, 40, 45, 50, 57, 64, 73, 82, 92, 102, 112, 124, 136, 148, 160, 480
1	0, 1, 2, 3, 4, 5, 7, 9, 11, 13, 15, 18, 21, 24, 28, 33, 38, 44, 50, 57, 64, 73, 82, 93, 104, 116, 128, 140, 155, 170, 185, 200, 480
2	0, 1, 2, 3, 4, 6, 8, 10, 13, 16, 20, 24, 28, 33, 38, 45, 52, 61, 70, 79, 88, 100, 112, 127, 142, 160, 178, 196, 217, 238, 259, 280, 480
3	0, 1, 2, 4, 6, 10, 14, 18, 22, 26, 30, 34, 42, 50, 58, 66, 74, 84, 96, 108, 120, 136, 152, 168, 192, 216, 240, 272, 304, 336, 376, 424, 480
4	0, 1, 2, 4, 6, 10, 14, 18, 26, 34, 42, 50, 62, 74, 86, 98, 112, 128, 144, 160, 176, 196, 216, 236, 256, 280, 304, 328, 352, 384, 416, 448, 480
5	0, 80, 92, 104, 112, 120, 128, 136, 144, 148, 152, 156, 160, 164, 168, 172, 176, 180, 184, 188, 192, 196, 200, 208, 216, 224, 232, 240, 248, 256, 268, 280, 480
6	0, 200, 212, 224, 232, 240, 248, 256, 264, 268, 272, 276, 280, 284, 288, 292, 296, 300, 304, 308, 312, 316, 320, 328, 336, 344, 352, 360, 368, 376, 388, 400, 480
7	0, 320, 332, 344, 356, 364, 372, 380, 384, 388, 392, 396, 400, 404, 408, 412, 416, 420, 424, 428, 432, 436, 440, 444, 448, 452, 456, 460, 464, 468, 472, 476, 480

表A.4　480 点帧长低码率子带划分表

索引	子带划分
0	0, 1, 2, 3, 4, 6, 8, 10, 13, 16, 20, 24, 28, 33, 38, 45, 52, 61, 70, 79, 88, 100, 112, 127, 142, 160, 178, 196, 217, 238, 259, 280, 480
1	0, 1, 2, 3, 5, 7, 9, 12, 15, 18, 22, 26, 30, 35, 41, 48, 56, 65, 74, 84, 94, 106, 118, 134, 150, 166, 184, 202, 220, 240, 260, 280, 480
2	0, 1, 2, 3, 4, 5, 7, 9, 11, 14, 17, 21, 25, 29, 34, 40, 46, 52, 60, 68, 76, 86, 98, 110, 126, 144, 162, 180, 200, 224, 250, 280, 480
3	0, 2, 4, 6, 8, 12, 16, 21, 26, 31, 36, 41, 46, 51, 56, 61, 66, 71, 77, 83, 89, 95, 103, 111, 121, 131, 147, 163, 179, 203, 240, 280, 480
4	0, 1, 2, 3, 5, 7, 9, 12, 15, 19, 23, 27, 32, 37, 43, 49, 57, 66, 76, 86, 98, 110, 125, 140, 158, 176, 194, 216, 238, 264, 290, 320, 480
5	0, 1, 2, 3, 5, 7, 10, 13, 17, 21, 25, 30, 35, 41, 47, 54, 62, 70, 80, 90, 102, 114, 130, 146, 162, 180, 198, 218, 240, 264, 290, 320, 480
6	0, 1, 2, 4, 6, 8, 11, 14, 18, 22, 26, 30, 36, 42, 50, 58, 66, 76, 88, 100, 112, 128, 144, 160, 182, 204, 226, 256, 286, 316, 352, 400, 480

表 A.4　480 点帧长低码率子带划分表（续）

索引	子带划分
7	0, 1, 2, 4, 6, 8, 11, 14, 18, 22, 26, 30, 36, 42, 50, 58, 68, 78, 90, 102, 116, 132, 148, 166, 186, 208, 234, 262, 292, 324, 360, 400, 480

表A.5　240 点帧长高码率子带划分表

索引	子带划分
0	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 52, 56, 60, 64, 68, 74, 80, 240
1	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 82, 88, 94, 100, 240
2	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 28, 32, 36, 40, 44, 48, 52, 56, 62, 68, 74, 80, 88, 96, 104, 112, 120, 130, 140, 240
3	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 30, 34, 38, 42, 48, 54, 60, 68, 76, 84, 96, 108, 120, 136, 152, 168, 188, 212, 240
4	0, 2, 4, 6, 8, 10, 12, 14, 16, 20, 24, 28, 32, 38, 44, 50, 56, 64, 72, 80, 88, 98, 108, 118, 128, 140, 152, 164, 176, 192, 208, 224, 240
5	0, 40, 46, 52, 56, 60, 64, 68, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 104, 108, 112, 116, 120, 124, 128, 134, 140, 240
6	0, 100, 106, 112, 116, 120, 124, 128, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 164, 168, 172, 176, 180, 184, 188, 194, 200, 240
7	0, 160, 166, 172, 178, 182, 186, 190, 192, 194, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238, 240

表A.6　240 点帧长低码率子带划分表

索引	子带划分
0	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 19, 22, 26, 30, 35, 39, 44, 50, 56, 63, 71, 80, 89, 98, 108, 119, 129, 140, 240
1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13, 15, 17, 20, 24, 28, 32, 37, 42, 47, 53, 59, 67, 75, 83, 92, 101, 110, 120, 130, 140, 240
2	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 17, 20, 23, 26, 30, 34, 38, 43, 49, 55, 63, 72, 81, 90, 100, 112, 125, 140, 240
3	0, 1, 2, 3, 4, 6, 8, 10, 13, 15, 18, 20, 23, 25, 28, 30, 33, 35, 38, 41, 44, 47, 51, 55, 60, 65, 73, 81, 89, 101, 120, 140, 240
4	0, 1, 2, 3, 4, 5, 6, 7, 9, 11, 13, 14, 16, 18, 21, 24, 28, 33, 38, 43, 49, 55, 62, 70, 79, 88, 97, 108, 119, 132, 145, 160, 240
5	0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 17, 20, 23, 27, 31, 35, 40, 45, 51, 57, 65, 73, 81, 90, 99, 109, 120, 132, 145, 160, 240
6	0, 1, 2, 3, 4, 5, 6, 7, 9, 11, 13, 15, 18, 21, 25, 29, 33, 38, 44, 50, 56, 64, 72, 80, 91, 102, 113, 128, 143, 158, 176, 200, 240

表 A.6　240 点帧长低码率子带划分表（续）

索引	子带划分
7	0，1，2，3，4，5，6，7，9，11，13，15，18，21，25，29，34，39，45，51，58，66，74，83，93，104，117，131，146，162，180，200，240

表A.7　子带包络 Huffman 码表

索引	码字	比特数
0	3	2
1	2	2
2	1	2
3	1	3
4	0	4
5	2	5
6	3	5

表A.8　4 tuple Huffman 码表 1

索引	码字	比特数
0	0	4
1	1	4
2	2	4
3	3	4
4	4	4
5	5	4
6	6	4
7	7	4
8	8	4
9	9	4
10	10	4
11	11	4
12	12	4
13	13	4
14	14	4
15	15	4

表A.9　4 tuple Huffman 码表 2

索引	码字	比特数
0	1	3
1	13	4
2	12	4
3	9	4

表 A.9　4 tuple Huffman 码表 2（续）

索引	码字	比特数
4	15	4
5	5	4
6	8	4
7	7	4
8	14	4
9	4	4
10	10	4
11	0	5
12	11	4
13	1	4
14	6	4
15	1	5

表A.10　4 tuple Huffman 码表 3

索引	码字	比特数
0	1	2
1	1	4
2	0	4
3	12	4
4	3	4
5	13	4
6	11	4
7	18	5
8	2	4
9	15	4
10	14	4
11	32	6
12	10	4
13	19	5
14	17	5
15	33	6

表A.11　4 tuple Huffman 码表 4

索引	码字	比特数
0	1	1
1	5	4
2	4	4
3	4	5

表 A.11　4 tuple Huffman 码表 4（续）

索引	码字	比特数
4	7	4
5	5	5
6	3	5
7	2	6
8	6	4
9	7	5
10	6	5
11	0	7
12	2	5
13	3	6
14	1	6
15	1	7

表A.12　2 tuple Huffman 码表 1

索引	码字	比特数
0	1	3
1	13	4
2	14	4
3	8	4
4	12	4
5	15	4
6	10	4
7	4	4
8	9	4
9	5	4
10	6	4
11	0	5
12	11	4
13	7	4
14	1	4
15	1	5

表A.13　2 tuple Huffman 码表 2

索引	码字	比特数
0	1	3
1	7	3
2	0	4
3	7	4

表 A.13　2 tuple Huffman 码表 2（续）

索引	码字	比特数
4	6	3
5	1	4
6	10	4
7	10	5
8	8	4
9	9	4
10	8	5
11	22	5
12	6	4
13	9	5
14	11	5
15	23	5

表A.14　2 tuple Huffman 码表 3

索引	码字	比特数
0	1	2
1	1	4
2	2	4
3	11	4
4	0	4
5	3	4
6	14	4
7	18	5
8	12	4
9	13	4
10	16	5
11	30	5
12	10	4
13	17	5
14	19	5
15	31	5

表A.15　2 tuple Huffman 码表 4

索引	码字	比特数
0	1	1
1	5	4
2	6	4
3	3	5

表 A.15　2 tuple Huffman 码表 4（续）

索引	码字	比特数
4	4	4
5	7	4
6	6	5
7	2	6
8	4	5
9	5	5
10	0	6
11	14	6
12	2	5
13	1	6
14	3	6
15	15	6

表A.16　1 tuple Huffman 码表 1

索引	码字	比特数
0	3	2
1	2	2
2	1	2
3	1	3
4	0	4
5	3	5
6	4	6
7	5	6

表A.17　1 tuple Huffman 码表 2

索引	码字	比特数
0	3	2
1	2	2
2	0	3
3	1	3
4	3	3
5	4	4
6	10	5
7	11	5

表A.18　1 tuple Huffman 码表 3

索引	码字	比特数
0	0	1

表 A.18　1 tuple Huffman 码表 3（续）

索引	码字	比特数
1	5	3
2	6	3
3	7	3
4	9	4
5	16	5
6	34	6
7	35	6

表A.19　1 tuple Huffman 码表 4

索引	码字	比特数
0	0	3
1	1	3
2	2	3
3	3	3
4	4	3
5	5	3
6	6	3
7	7	3