GY

Radio, Television, and Network Audiovisual Industry Standard of the People's Republic of China

GY/T 363-2023

3D Audio Coding and Rendering

Issued on February 1, 2023	Implemented on February 1, 2023

Issued by the National Radio Television Administration

Contents

Foreword

This document is drafted in accordance with GB/T 1.1-2020 *Directives for Standardization - Part 1: Rules for the Structure and Drafting of Standardizing Documents*.

This document is under the jurisdiction of the National Radio, Film and Television Standardization Technical Committee (SAC/TC 239).

Drafting organizations of this document: China Media Group; the Academy of Broadcasting Planning, NRTA; the Academy of Broadcasting Science, NRTA; Tsinghua University; Huawei Technologies Co., Ltd.; HiSilicon (Shanghai) Technologies Co., Ltd.; Beijing Douyin Information Service Co., Ltd.; SineMedia (Beijing) Co., Ltd.; the UHD World Association.

Drafters of this document: Jiang Wenbo, Guan Chaoyang, Zhou Yun, Dou Weibei, Ning Jinhui, Wang Zhe, Guo Xiaoqiang, Zhang Jiandong, Jiao Jianbo, Wu Jian, Huang Chuanzeng, Pang Chao, Liu Derong, Lu Nannan, Zou Zhiming, Liu Hanyuan, Hu Xiao, Wu Qiang, Wang Rui, Zhai Nan, Gao Yuan, Shi Junjie, Zhang Nanpeng, Ye Xuzhou, Zhang Zhengpu, Wan Yupeng, Li Xiaoyu, Wang Qiannan, Li Lin, Cao Xuyang, Wang Dongfei, Li Da, Liu Shilei, Li Feng, Zhang Wengang, and Zhang Hongyu.

Introduction

The issuing body of this document calls for attention to the fact that the declaration of compliance with this document may involve the use of patents related to the technologies for audio encoding and decoding in sections 7.3 to 7.7, 7.9, 7.10, 7.13, and E.2 to E.4, patents related to the technologies for speaker rendering in section D.1, and patents related to the technologies for binaural rendering in section D.2. The patent list is as follows:

No.	Chapter No.	Patent No.	Patent name	Patent owner
1	6 and Appendix A	200710175993.6	ENCODING AND DECODING INTEGRATION SYSTEM AND METHOD	Tsinghua University
2	7.3	202110559102.7	ENCODING AND DECODING METHOD, APPARATUS, AND DEVICE, STORAGE MEDIUM, AND COMPUTER PROGRAM	Huawei Technologies Co., Ltd.
3	7.3	202110596023.3	AUDIO DATA CODING METHOD AND RELATED APPARATUS, AND COMPUTER-READABLE STORAGE MEDIUM	Huawei Technologies Co., Ltd.
4	7.4	202110865328.X	METHOD AND APPARATUS FOR AUDIO SIGNAL ENCODING AND DECODING	Huawei Technologies Co., Ltd.
5	7.5	200710135833.9	STEREO AUDIO ENCODING/DECODING METHOD AND ENCODER/DECODER	Huawei Technologies Co., Ltd. and Tsinghua University
6	7.5	200810106460.7	STEREO SIGNAL ENCODING AND DECODING METHOD AND APPARATUS, AND CODEC SYSTEM	Huawei Technologies Co., Ltd. and Tsinghua University
7	7.5	200710304486.8	AUDIO SIGNAL ENCODING AND DECODING METHOD AND APPARATUS	Tsinghua University
8	7.5	201110289391.X	DOWNMIXED SIGNAL GENERATION AND RESTORATION METHOD AND APPARATUS	Huawei Technologies Co., Ltd.
9	7.6	202210699863.7	MULTI-CHANNEL SIGNAL ENCODING AND DECODING METHOD AND DEVICE, AND TERMINAL DEVICE	Huawei Technologies Co., Ltd.
10	7.6	202010699711.8	MULTI-CHANNEL AUDIO SIGNAL ENCODING AND DECODING METHOD AND APPARATUS	Huawei Technologies Co., Ltd.
11	7.6 and 7.7	200910235713.5	MULTI-CHANNEL AUDIO ENCODING AND DECODING	Tsinghua University

No.	Chapter No.	Patent No.	Patent name	Patent owner
			METHOD, ENCODER, AND DECODER	
12	7.7	202110700570.1	THREE-DIMENSIONAL AUDIO SIGNAL PROCESSING METHOD AND APPARATUS	Huawei Technologies Co., Ltd.
13	7.7	202110602507.4	THREE-DIMENSIONAL AUDIO SIGNAL PROCESSING METHOD AND APPARATUS	Huawei Technologies Co., Ltd. and Peking University
14	7.9	200910169403.8	BANDWIDTH EXTENSION METHOD AND APPARATUS	Huawei Technologies Co., Ltd.
15	7.9	201180003043.X	SOURCE SIGNAL RECONSTRUCTION METHOD AND DECODER	Huawei Technologies Co., Ltd.
16	7.9	202110654037.6	ENCODING AND DECODING METHOD, APPARATUS, AND DEVICE, STORAGE MEDIUM, AND COMPUTER PROGRAM	Huawei Technologies Co., Ltd.
17	7.10	201610877571.2	AUDIO SIGNAL RECONSTRUCTION METHOD AND APPARATUS	Huawei Technologies Co., Ltd.
18	7.10	201010187426.4	SIGNAL PROCESSING METHOD AND SYSTEM	Huawei Technologies Co., Ltd.
19	7.13	202110247466.1	VIRTUAL SPEAKER SET DETERMINING METHOD AND APPARATUS	Huawei Technologies Co., Ltd. and Peking University
20	7.13	202110246382.6	HOA COEFFICIENT OBTAINING METHOD AND APPARATUS	Huawei Technologies Co., Ltd. and Peking University
21	7.13	202011377433.0	AUDIO ENCODING AND DECODING METHOD AND APPARATUS	Huawei Technologies Co., Ltd. and Peking University
22	7.13	202011377320.0	AUDIO ENCODING AND DECODING METHOD AND APPARATUS	Huawei Technologies Co., Ltd. and Peking University
23	D.2.1	PCT/CN2021/100076	AUDIO RENDERING SYSTEM AND METHOD, AND ELECTRONIC DEVICE	Beijing Bytedance Network Technology Co., Ltd.
24	D.2.2	PCT/CN2021/100062	AUDIO SIGNAL CODING METHOD AND APPARATUS, AND ELECTRONIC DEVICE FOR AUDIO RENDERING	Beijing Bytedance Network Technology Co., Ltd.
25	Appendix C and D.2.1	PCT/CN2021/114366	AUDIO METADATA PROCESSING METHOD AND APPARATUS	Beijing Bytedance Network Technology Co., Ltd.

No.	Chapter No.	Patent No.	Patent name	Patent owner
26	D.1.1	202110984837.4	AUDIO PRODUCTION MODEL AND GENERATION METHOD, ELECTRONIC DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
27	D.1.1	202111102045.6	AUDIO PROGRAM METADATA AND GENERATION METHOD, ELECTRONIC DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
28	D.1.1	202111100818.7	AUDIO CONTENT METADATA AND GENERATION METHOD, ELECTRONIC DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
29	D.1.1	202111102038.6	AUDIO OBJECT METADATA AND GENERATION METHOD, ELECTRONIC DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
30	D.1.1	202111205630.9	AUDIO TRACK UNIQUE IDENTIFICATION METADATA AND GENERATION METHOD, ELECTRONIC DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
31	D.1.1	202111204386.4	AUDIO TRACK METADATA AND GENERATION METHOD, ELECTRONIC EQUIPMENT, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
32	D.1.1	202111202898.7	AUDIO STREAM METADATA AND GENERATION METHOD, ELECTRONIC DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
33	D.1.1	202111308422.1	AUDIO BED–BASED AUDIO PACKET FORMAT METADATA AND GENERATION METHOD, DEVICE, AND MEDIUM	SineMedia (Beijing) Co., Ltd.
34	D.1.1	202111308430.6	OBJECT-BASED AUDIO PACKET FORMAT METADATA AND GENERATION METHOD, DEVICE, AND MEDIUM	SineMedia (Beijing) Co., Ltd.
35	D.1.4	202111306844.5	SCENE-BASED AUDIO PACKET FORMAT METADATA AND GENERATION METHOD, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
36	D.1.1	202111308421.7	BINAURAL-BASED AUDIO PACKET FORMAT METADATA AND GENERATION METHOD, DEVICE, AND MEDIUM	SineMedia (Beijing) Co., Ltd.

No.	Chapter No.	Patent No.	Patent name	Patent owner
37	D.1.1	202111021068.4	AUDIO BED–BASED AUDIO CHANNEL METADATA AND GENERATION METHOD, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
38	D.1.1	202111020417.0	OBJECT-BASED AUDIO CHANNEL METADATA AND GENERATION METHOD, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
39	D.1.1	202111021066.5	SCENE-BASED AUDIO CHANNEL METADATA AND GENERATION METHOD, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
40	D.1.1	202111021039.8	BINAURAL-BASED AUDIO CHANNEL METADATA AND GENERATION METHOD, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
41	D.1.1	202111425628.2	SERIAL AUDIO METADATA FRAME GENERATION METHOD AND APPARATUS, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
42	D.1.1	202111425590.9	TRANSMISSION AUDIO TRACK FORMAT SERIAL METADATA GENERATION METHOD AND APPARATUS, DEVICE, AND MEDIUM	SineMedia (Beijing) Co., Ltd.
43	D.1.1	202111424251.9	SERIAL AUDIO BLOCK FORMAT METADATA GENERATION METHOD AND APPARATUS, DEVICE, AND MEDIUM	SineMedia (Beijing) Co., Ltd.
44	D.1.1	202111424254.2	SERIAL AUDIO METADATA GENERATION METHOD AND APPARATUS, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
45	D.1.1	202111666346.1	BROADCAST AUDIO FORMAT FILE GENERATION METHOD AND APPARATUS, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
46	D.1.1	202111666362.0	AUDIO METADATA BLOCK GENERATION METHOD AND APPARATUS, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
47	D.1.1	202210588174.9	METHOD AND APPARATUS FOR GENERATING INTERNAL DATA	SineMedia (Beijing) Co., Ltd.

No.	Chapter No.	Patent No.	Patent name	Patent owner
			STRUCTURE OF RENDERER, DEVICE, AND STORAGE MEDIUM	
48	D.1.1 and D.1.2	202210634563.0	METHOD AND APPARATUS FOR RENDERING AUDIO BED–BASED AUDIO BY USING METADATA	SineMedia (Beijing) Co., Ltd.
49	D.1.1 and D.1.2	202210762912.7	SHARED RENDERER COMPONENT CONFIGURATION METHOD AND APPARATUS, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
50	D.1.1 and D.1.2	202210760302.3	AUDIO BED RENDERING ITEM DATA MAPPING METHOD AND APPARATUS, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
51	D.1.1 and D.1.2	202210603204.9	METHOD AND APPARATUS FOR DETERMINING RENDERING ITEM OF RENDERER, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
52	D.1.1 and D.1.2	202210600880.0	AUDIO BED OUTPUT RENDERING ITEM DETERMINING METHOD AND APPARATUS, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
53	D.1.1 and D.1.3	202210603208.7	OBJECT OUTPUT RENDERING ITEM DETERMINING METHOD AND APPARATUS, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
54	D.1.1 and D.1.4	202210603212.3	SCENE OUTPUT RENDERING ITEM DETERMINING METHOD AND APPARATUS, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
55	D.1.1	202210603184.5	METHOD AND APPARATUS FOR PROCESSING RENDERING ITEM OF AUDIO RENDERER, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
56	D.1.1 and D.1.3	202210608202.9	SCENE RENDERING ITEM DATA MAPPING METHOD AND APPARATUS, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
57	D.1.1 to D.1.4	202210782056.1	METHOD AND APPARATUS FOR CALCULATING GAIN OF AUDIO RENDERER, DEVICE, AND STORAGE MEDIUM	SineMedia (Beijing) Co., Ltd.
58	D.1.1 and	202210910129.0	METHOD AND APPARATUS FOR PARSING METADATA FOR	SineMedia (Beijing)

No.	Chapter No.	Patent No.	Patent name	Patent owner
	D.1.3		OBJECT RENDERER, DEVICE, AND MEDIUM	Co., Ltd.
59	D.1.1 and D.1.3	202210907370.8	METHOD AND APPARATUS FOR RENDERING OBJECT–BASED AUDIO BY USING METADATA	SineMedia (Beijing) Co., Ltd.
60	D.1.1 and D.1.4	202210912275.7	METHOD AND APPARATUS FOR RENDERING SCENE–BASED AUDIO BY USING METADATA	SineMedia (Beijing) Co., Ltd.
61	D.1.1 and D.1.3	202211057713.2	METHOD AND APPARATUS FOR RENDERING OBJECT–BASED AUDIO BY USING METADATA	SineMedia (Beijing) Co., Ltd.
62	D.1.1 and D.1.4	202211063746.8	METHOD AND APPARATUS FOR RENDERING SCENE–BASED AUDIO BY USING METADATA	SineMedia (Beijing) Co., Ltd.
63	E.2	201610879165.X	AUDIO SIGNAL RECONSTRUCTION METHOD AND APPARATUS	Huawei Technologies Co., Ltd.
64	E.2	201610252268.3	AUDIO SIGNAL SAMPLING AND RECONSTRUCTION METHOD, APPARATUS, AND SYSTEM	Huawei Technologies Co., Ltd.
65	E.3	202110595367.2	MULTI-CHANNEL AUDIO SIGNAL CODING METHOD AND APPARATUS	Huawei Technologies Co., Ltd.
66	E.3	200980154599.1	STEREO ENCODING METHOD AND DEVICE	Huawei Technologies Co., Ltd.
67	E.3	202010699775.8	MULTI-CHANNEL AUDIO SIGNAL CODING METHOD AND APPARATUS	Huawei Technologies Co., Ltd.
68	E.3	202010699706.7	MULTI-CHANNEL AUDIO SIGNAL ENCODING AND DECODING METHOD AND APPARATUS	Huawei Technologies Co., Ltd.
69	E.4	202110530309.1	AUDIO ENCODING AND DECODING METHOD AND APPARATUS	Huawei Technologies Co., Ltd.
70	E.4	202110536634.9	THREE-DIMENSIONAL AUDIO SIGNAL CODING METHOD AND APPARATUS, AND CODER	Huawei Technologies Co., Ltd.
71	E.4	202110680341.8	THREE-DIMENSIONAL AUDIO SIGNAL CODING METHOD AND APPARATUS, CODER, AND SYSTEM	Huawei Technologies Co., Ltd.
72	E.4	202110535832.3	THREE-DIMENSIONAL AUDIO SIGNAL CODING METHOD AND APPARATUS, AND CODER	Huawei Technologies Co., Ltd.

No.	Chapter No.	Patent No.	Patent name	Patent owner
73	E.5	202110536623.0	THREE-DIMENSIONAL AUDIO SIGNAL CODING METHOD AND APPARATUS, AND CODER	Huawei Technologies Co., Ltd. and Peking University
74	E.5	202110536631.5	THREE-DIMENSIONAL AUDIO SIGNAL CODING METHOD AND APPARATUS, AND CODER	Huawei Technologies Co., Ltd. and Peking University

The issuing body of this document has no position as to the authenticity, validity, and scope of the patents.

Patent holders have assured the issuing body of this document that they are willing to negotiate patent licenses under reasonable and non-discriminatory terms or conditions with any applicant. Patent holders' statements are registered with the issuing body of this document.

The patent information related to audio encoding and decoding technologies can be obtained using the following contact information:

Contact: Ding Haitao

Address: Huawei Base, Bantian, Longgang District, Shenzhen

Postal code: 518129

Tel: 86-18617085835

Fax: 86-0755-28429534

Contact: Dou Weibei

Address: 4-102 Department of Electrical Engineering, Tsinghua University, Haidian District, Beijing

Postal code: 100084

Tel: 86-13910663993

Fax: 86-010-62770317

The patent information related to speaker rendering technologies can be obtained using the following contact information:

Contact: Wu Jian

Address: 1906, 19F, No.12, Yabao Road, Chaowai, Chaoyang District, Beijing

Postal code: 100011

Tel: 86-13501396702

Fax: 86-010-53606805

The patent information related to binaural rendering technologies can be obtained using the following contact information:

Contact: Huang Chuanzeng

Address: Building 5, Zijin Digital Park, No.18, South Fourth Street, Zhongguancun, Haidian District, Beijing

Postal code: 100190

Tel: 86-18519188906

Fax: 86-010-58330899

Attention is drawn to the possibility that some of the elements of this document may relate to patents other than those identified. The issuing body of this document shall not be held responsible for identifying any or all such patents.

3D Audio Coding and Rendering

1 Scope

This document specifies the bitstream representation for 3D audio coding and the decoding process, and provides reference implementations for speaker rendering and binaural rendering.

This document is applicable to 3D audio, surround audio, and stereo coding and rendering in the radio, television, and network audiovisual field.

2 Normative References

The content of the following documents is normatively referenced in this document, and constitutes indispensable provisions of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced documents (including any amendments) applies.

GB/T 5271.1 Information Technology - Vocabulary - Part 1: Fundamental Terms

GB/T 5271.4 Information Technology - Vocabulary - Part 4: Organization of Data

GB/T 5271.9 Information Technology - Vocabulary - Part 9: Data Communication

GB/T 5271.34 Information Technology - Vocabulary - Part 34: Artificial Intelligence - Neural Networks

GB/T 33475.3-2018 Information Technology - High Efficiency Media Coding - Part 3: Audio

GY/T 262-2012 Algorithms to Measure Audio Program Loudness and True-peak Audio Level

GY/T 316-2018 Advanced Sound System for Program Production

ITU-R BS.2076-2 Audio Definition Model

ITU-R BS.2094-1 Common Definitions for the Audio Definition Model

3GPP TS 26.445 Codec for Enhanced Voice Services (EVS): Detailed Algorithmic Description

3 Terms and Definitions

The terms and definitions defined in GB/T 5271.1, GB/T 5271.4, GB/T 5271.9, and GB/T 5271.34 and the following terms and definitions are applicable to this document.

3.1

Encoding

Reads an audio sample value and generates a valid bitstream that conforms to this document.

3.2

Coded bitstream

A coded representation of an audio signal.

3.3

Side information

Necessary information to control decoding in a bitstream.

3.4

Object

A sound perceived as a whole or a sound independent of an environment emitted by a sound source.

3.5

Decoding

A type of data processing defined in this document, that is, a process of reading a coded bitstream and outputting an audio sample value.

3.6

Spectral coefficient

Used to analyze discrete-spectrum domain data output by a filter group.

3.7

Channel

A set of ordered audio samples for transmission to a single speaker or another replay device.

3.8

Dual-channel stereo

An audio format in which two channels carry audio signals with a specific phase relationship, and a sense of a wider sound field is usually provided for a listener through replay by two symmetric speakers located in front of the listener or by headphones.

3.9

Surround sound

An audio format in which a plurality of channels carry a plurality of audio signals that form complete audio content, and a sense of a surround sound field is provided for a listener through replay by a plurality of speakers surrounding the listener at an ear height layer of the listener.

3.10

3D sound

An audio format in which a plurality of channels carry a plurality of audio signals that form complete audio content, and a higher spatial resolution of sound image localization and a sense of an immersive sound field are provided for a listener through direct replay by a plurality of speakers surrounding the listener at different height layers, or through replay after rendering or mapping.

3.11

Metadata

Data describing information related to audio data.

3.12

Rendering

A process in which a given audio transmission format is converted into a directly replayable audio format that is applicable to configurations of terminal speakers and headphones.

3.13

Speaker rendering

Creates the presentation effect of spatial and dimensional auditory perception for audio signals through a set of speakers.

3.14

Binaural rendering binaural rendering

Creates the effect of spatial and dimensional auditory perception for multi-channel audio through two channels and presents the effect at terminal headphones.

3.15

First order ambisonics (FOA)

An ambisonic signal that is obtained through first order spherical harmonic expansion and that includes four audio channels.

Note: The ambisonic technology is a technology of three-dimensional audio recording, storage, transmission, and replay based on sound field spherical harmonic analysis. The ambisonic signal is an audio format of this technology. Spherical harmonic expansion is performed on a sound field signal including a multi-channel audio signal, and the spherical harmonic expansion coefficient is the ambisonic signal. The higher order spherical harmonic expansion indicates a higher spatial resolution of the ambisonics.

3.16

Higher order ambisonics (HOA)

An ambisonic signal obtained through second or higher order spherical harmonic expansion.

4 Acronyms and Abbreviations

The following acronyms and abbreviations apply to this document.

AASF AVS audio storage format (AVS Audio Storage Format)

AATF AVS audio transport format (AVS Audio Transport Format)

AVS audio video coding standard (Audio Video coding Standard)

BWE bandwidth extension (Bandwidth Extension)

CNN	convolutional neural network (Convolutional Neural Network)
CRC	cyclic redundancy check (Cyclic Redundancy Check)
FFT	fast fourier transform (Fast Fourier Transform)
IGDN	inverse generalized divisive normalization (Inverse Generalized Divisive Normalization)
ILD	inter-channel level difference (Inter-channel Level Difference)
IMDCT	inverse modified discrete cosine transform (Inverse Modified Discrete Cosine Transform)
LFE	low frequency effect (Low Frequency Effect)
LPC	linear prediction coefficient (Linear Prediction Coefficients)
LSF	line spectral frequencies (Line Spectral Frequencies)
LSP	line spectral pairs (Line Spectral Pairs)
MCAC	multi-channel adaptive coupling (Multi Channel Adaptive Coupling)
MCR	maximum correlation rotation (Maximum Correlation Rotation)
MDCT	modified discrete cosine transform (Modified Discrete Cosine Transform)
M/S	middle/side stereo (Middle/Side)
PAR	peak-to-average ratio (Peak-to-Average Ratio)
PCM	pulse-code modulation (Pulse-Code Modulation)
SFB	scale factor band (Scale Factor Band)
SFM	spectral flatness measure (Spectral Flatness Measure)
TNS	temporal noise shaping (Temporal Noise Shaping)
VBAP	vector-based amplitude panning (Vector-based Amplitude Panning)
VQ	vector quantization (Vector Quantization)

5 Conventions

5.1 Overview

The mathematical operators and precedences used in this document are similar to those used in the C language. However, integer division and arithmetic shift operations are specifically defined. Unless otherwise specified, numbering and counting conventions generally begin from 0.

5.2 Arithmetic Operators

Definitions of arithmetic operators are shown in Table 1.

Table 1 Definitions of arithmetic operators

Arithmetic operator	Definition
+	Addition

Arithmetic operator	Definition
–	Subtraction (as a binary operator) or negation (as a unary prefix operator)
×	Multiplication
*	Multiplication
a^b	Exponentiation, indicating the *bth* power of *a*. It may also indicate a superscript.
pow(a, b)	Exponentiation, indicating the *bth* power of *a*
/	Division without truncation or rounding
÷	Division without truncation or rounding
$\frac{a}{b}$	Division without truncation or rounding
$\sum_{i=a}^{b} f(i)$	Cumulative sum of the *f*(*i*) function when the independent variable *i* takes all integer values from *a* to *b* (including *b*).
$\lfloor \cdot \rfloor$	Rounding down
$\sqrt{a}$	Square root of *a*
$\Vert\ \Vert_2$	2-norm

5.3 Logical Operators

Definitions of logical operators are shown in Table 2.

Table 2 Definitions of logical operators

Logical operator	Definition
||	Logical OR
&&	Logical AND
!	Logical NOT

5.4 Relational Operators

Definitions of relational operators are shown in Table 3.

Table 3 Definitions of relational operators

Relational operator	Definition
>	Greater than
≥	Greater than or equal to

Relational operator	Definition
<	Less than
≤	Less than or equal to
==	Equal to
≠	Not equal to

5.5 Bitwise Operators

Definitions of bitwise operators are shown in Table 4.

Table 4 Definitions of bitwise operators

Bitwise operator	Definition
&	AND operation
|	OR operation
~	Negation operation
a >> b	Right shift of a two's complement representation of the integer *a* by *b* digits. This operation is defined only when *b* is a positive number. Bits shifted into the most significant bits (MSBs) as a result of the right shift have a value equal to the MSB of *a* prior to the shift operation.
a << b	Left shift of a two's complement representation of the integer *a* by *b* digits. This operation is defined only when *b* is a positive number. Bits shifted into the least significant bits (LSBs) as a result of the left shift have a value equal to 0.

5.6 Assignment

Definitions of assignment operators are shown in Table 5.

Table 5 Definitions of assignment operators

Assignment operation	Definition
=	Assignment operator
++	Auto-increment. *x*+1 is equivalent to *x* = *x* + 1. When the operator is used in an array subscript, a value of a variable is calculated prior to the auto-increment operation.
+=	Auto-increment by amount specified. For example, *x* += 3 is equivalent to *x* = *x* + 3, and *x* += (–3) is equivalent to *x* = *x* + (–3).
–=	Auto-decrement by amount specified. For example, x –= 3 is equivalent to *x* = *x* + (–3), and *x* –= (–3) is equivalent to *x* = *x* – (–3).

5.7 Mnemonic

Definitions of mnemonics are shown in Table 6.

Table 6 Definitions of mnemonics

Mnemonic	Definition
rpchof	Polynomial remainder, higher order first.
bslbf	A bit string, left bit first. A bit string is written as a string of 1s and 0s with single quotation marks, for example '1000 0001'. Spaces within a bit string are for ease of reading and have no significance. (Bitstream left bit first)
uimsbf	Unsigned integer. The most significant bit first. (Unsigned integer, most significant bit first)
bsmbf	A bit string is written as a string of 1s and 0s with single quotation marks, right bit first. For example, if a 5-bit value 6 and then a 3-bit value 2 are coded, the coded bit string is '010 00110'.

5.8 Mathematical Functions

For definitions of mathematical functions, refer to formula (1) and formula (2).

$$|x| = \begin{cases} x & ;\ x > 0 \\ 0 & ;\ x = 0 \\ -x & ;\ x < 0 \end{cases} \quad \text{......(1)}$$

x is an independent variable.

$$ReLu(x) = \begin{cases} x & ;\ x > 0 \\ 0 & ;\ x \leq 0 \end{cases} \quad \text{......(2)}$$

x is an independent variable.

5.9 Bitstream Syntax Rules

Each data item in the bitstream is in bold type. It is described by its name, bitwise length and type, and mnemonic of transmission order.

An operation caused by a decoded data element in a bitstream depends on a value of the data and a previously decoded data element. Unless otherwise specified, the "bit" in this document indicates a binary bit.

Note 1: The syntax described in this document is specified using "C" code. A variable or an expression of a non-zero value is equivalent to a true condition. A variable or an expression of a zero value is equivalent to a false condition.

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

If the condition is true, a data element group is generated immediately after a data stream. This process is repeated until the condition is false.

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

If the condition is true, a data element group is generated immediately after a data stream. This process is repeated until the condition is false.

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

If the condition is true, a first group of data elements is generated in a data stream. If the condition is false, a second set of data elements is generated in the data stream.

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

The expr1 is an expression that specifies an initial state of a loop, and usually specifies an initial state of a counter. The expr2 specifies a test condition before each loop. If the condition is false, the loop stops. The expr3 is an expression executed at the end of each loop, and usually is a count-up counter.

Note 2: The most common use is as follows:

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

The data element group is generated for n times. A condition structure within the data element group may depend on a value of the loop control variable i. This field is set to '0' when it appears for the first time, increases to '1' when it appears for the second time, and the like.

A corresponding data element is generated based on a value of an expression expr. When the value of expr is constcase1, the data element data_element1 is generated. When the value of expr is constcase2, the data element data_element2 is generated. The same rule applies. When the value of expr is constcasen, the data element data_elementn is generated. When the value of expr is not equal to any one of constcase1, constcase2, ..., and constcasen, the data element data_elementdefault is generated.

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

A variant of this structure is that no break occurs after a specific case. When the value of expr is constcasex, a data element is generated starting from a corresponding case constcasex until a break occurs. When the value of expr is constcase1, data elements data_element1 and data_element2 are generated. When the value of expr is constcasen, the data element data_elementn is generated.

```
switch(expr){
  case constcase1:
    data_element1
  case constcase2:
    data_element2
    break
```

```
    …
  case constcasen:
    data_elementn
    break
  default:
    data_elementdefault
    break
}
```

A data element group may contain a nested structure. For simplicity, "[ ]" is omitted when there is only one subsequent data element. The data element group is represented as follows:

- data_element[ ] is array data. The number of data elements depends on the context.

- data_element[n] is an (n+1)-th element of the array data.

- data_element[m][n] is an (m+1), (n+1)-th element of a two-dimensional array.

- data_element[l][m][n] is an (l+1), (m+1), and (n+1)-th element of a three-dimensional array.

- data_element[m...n] is all bits included between bits m and n.

Although syntax is expressed as a process item, it cannot be considered that the clause implements a reliable decoding process. It simply defines an error-free bitstream input.

Definition of **byte_alignment()** function:

If a current position is at a byte boundary, the byte_alignment() function returns '1', that is, a next bit in the bitstream is a start bit of a byte. Otherwise, '0' is returned.

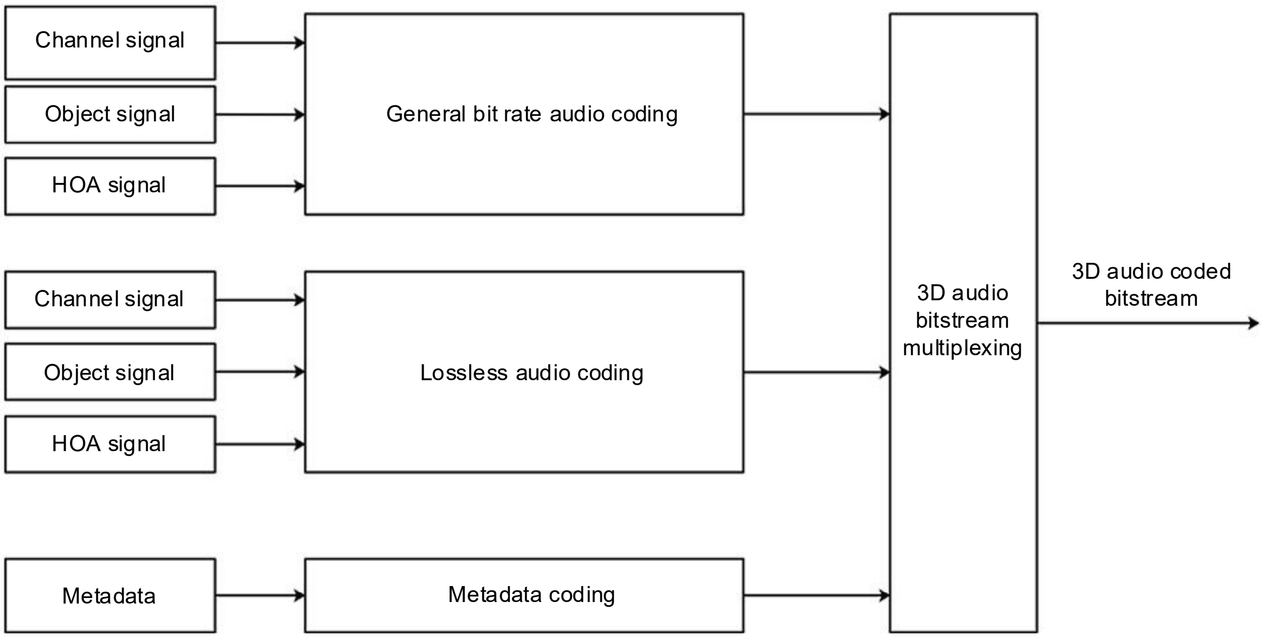
Definition of **Nextbits(n)** function:

The to-be-decoded n bits in a bit string or a bitstream are considered as a return value of Nextbits(n).

6 3D Audio Coding and Rendering Framework

3D audio coding includes general bit rate audio coding, lossless audio coding, and metadata coding. The general bit rate audio coding indicates high-efficiency lossy audio coding, the lossless audio coding indicates data lossless audio coding, and the metadata coding indicates coding of metadata. The 3D audio coding framework should be consistent with that in Figure 1. During 3D audio coding, an input audio signal is divided into a channel-based audio signal (hereinafter referred to as a "channel signal"), an object-based audio signal (hereinafter referred to as an "object signal"), and an HOA-based audio signal (including FOA, hereinafter referred to as an "HOA signal"). The channel signal is a mono signal, a dual-channel stereo signal, or a multi-channel surround/3D audio signal. The general bit rate audio coding (including a basic configuration and a low-complexity configuration) or the lossless audio coding may be selected for the channel signal, the object signal, and the HOA signal. The metadata coding is used for metadata. A 3D audio coded bitstream is obtained through 3D audio bitstream multiplexing. Neural network–based coding is used for the general bit rate audio coding.

Figure 1 3D audio coding framework



The parameters supported by the 3D audio coding are as follows:

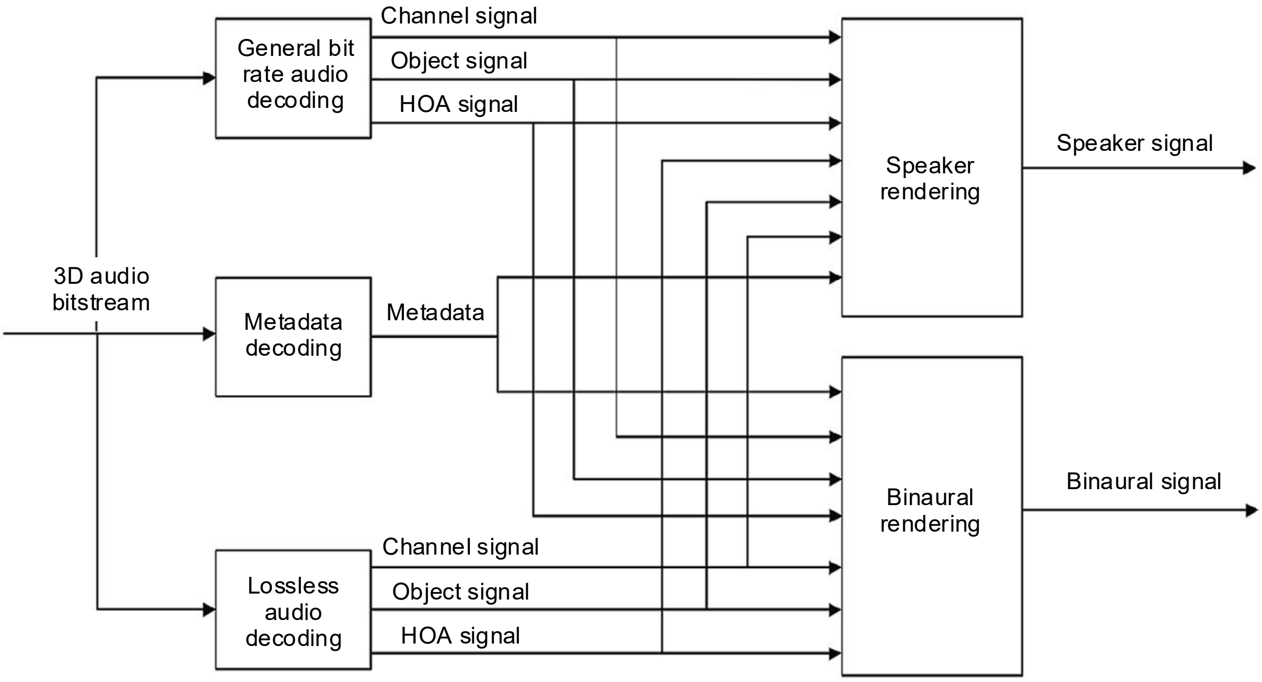
- Supported sampling rates are as follows: 44.1 kHz, 48 kHz, 96 kHz, and 192 kHz.

- Supported quantization precision is as follows: 16 bits and 24 bits.

- The lossless audio coding supports a maximum of 128 channels.

3D audio decoding is the reverse process of 3D audio coding. The 3D audio coded bitstream is processed through general bit rate audio decoding or lossless audio decoding to obtain a channel signal, an object signal, and an HOA signal, and metadata is obtained through metadata decoding. The decoded channel signal, object signal, and HOA signal may be rendered through speaker rendering to obtain a signal for speaker playing, or may be binaurally rendered to obtain a signal for playing by headphones. The 3D audio decoding and rendering framework should be consistent with that in Figure 2.

Figure 2 3D audio decoding and rendering framework



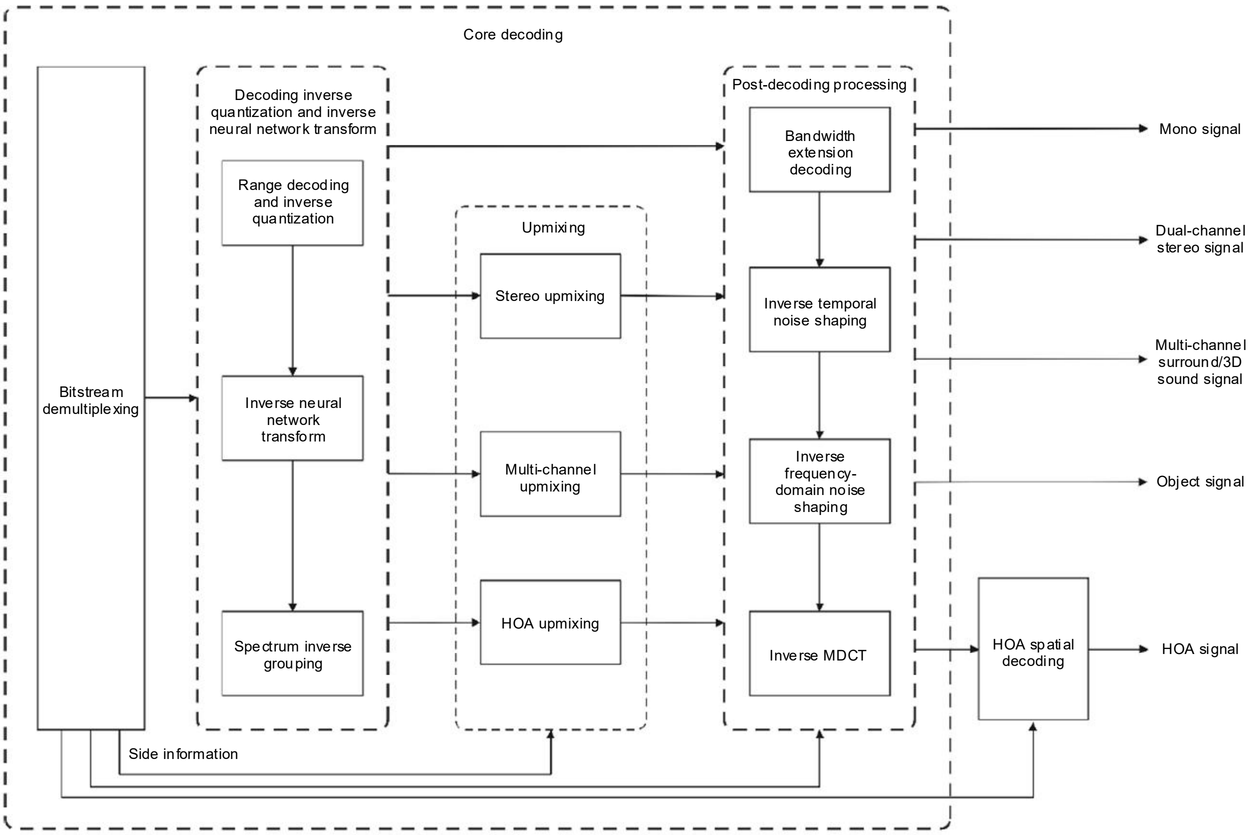
For general bit rate audio decoding, refer to chapter 7. For lossless audio decoding, refer to chapter 8. For metadata decoding, refer to chapter 9. The syntax and semantics of the 3D audio coded bitstream should comply with provisions in Annex A. The audio code table related to the general bit rate audio coding should comply with provisions in Annex B. Metadata parameters should comply with provisions in Annex C. For a speaker rendering reference implementation and a binaural rendering reference implementation, refer to Annex D. For general bit rate audio coding, refer to Annex E. For metadata coding, refer to Annex F. For correspondence between coding metadata in this document and metadata in ITU-R BS. 2076-2, refer to Annex G.

7 General Bit Rate Audio Decoding

7.1 Overview

The general bit rate audio decoding includes core decoding and HOA spatial decoding. The core decoding decodes a coded bitstream into a channel signal and an object signal. The core decoding and the HOA spatial decoding decodes a coded bitstream into an HOA signal. The general bit rate audio decoding framework should comply with Figure 3. The core decoding includes bitstream demultiplexing, decoding, inverse quantization, and inverse neural network transform (including range decoding, inverse quantization, inverse neural network transform, and spectrum inverse grouping), upmixing, and post-decoding processing. The upmixing supports stereo upmixing, multi-channel upmixing and HOA upmixing. The post-decoding processing includes bandwidth extension decoding, inverse temporal noise shaping, inverse frequency-domain noise shaping, and inverse MDCT.

Figure 3 General bit rate audio decoding framework



The mono decoding includes range decoding, inverse quantization, inverse neural network transform, spectrum inverse grouping, and post-decoding processing. The decoding framework should comply with Figure 4. In a range decoding and inverse quantization process, a transform-domain coefficient corresponding to an MDCT coefficient is obtained by parsing a bitstream. A reconstructed MDCT coefficient is obtained by performing inverse neural network transform. Finally, a time-domain mono signal is obtained through post-decoding processing. The post-decoding processing framework should comply with Figure 5.

The obtaining process is as follows.

a) Range decoding and inverse quantization: Bitstream information related to coding of an MDCT spectral coefficient is obtained from the bitstream, and range decoding and inverse quantization are performed to obtain a transform-domain coefficient corresponding to the MDCT coefficient. The transform-domain coefficient corresponding to the MDCT coefficient is used as an input for inverse neural network transform. The input and output parameters are as follows:
   - Input: the bitstream;
   - Output: the transform-domain coefficient corresponding to the MDCT coefficient.

b) Inverse neural network transform: The inverse neural network transform is performed on the transform-domain coefficient corresponding to the MDCT spectral coefficient to obtain a reconstructed MDCT coefficient. The input and output parameters are as follows:
   - Input: the transform-domain coefficient corresponding to the MDCT coefficient;
   - Output: the reconstructed MDCT coefficient.

c) Bandwidth extension decoding: A bandwidth extension decoding module performs energy adjustment and spectral detail adjustment on a low frequency part of the reconstructed MDCT coefficient based on a bandwidth extension parameter obtained by decoding the bitstream, to obtain a high-frequency spectrum component. The input and output parameters are as follows:
   - Input: the reconstructed MDCT coefficient and a bandwidth extension parameter;
   - Output: an MDCT coefficient of the mono signal.

d) Inverse temporal noise shaping: An inverse temporal noise shaping module is an inverse process at an encoder, and recovers an MDCT coefficient before temporal noise shaping. The input and output parameters are as follows:
   - Input: the MDCT coefficient of the mono signal and a temporal noise shaping parameter;
   - Output: an MDCT coefficient of the mono signal after inverse temporal noise shaping.

e) Inverse frequency-domain noise shaping: An inverse frequency-domain noise shaping module is an inverse process at an encoder, and recovers an MDCT coefficient before frequency-domain noise shaping. The input and output parameters are as follows:
   - Input: the MDCT coefficient of the mono signal after inverse temporal noise shaping and a frequency-domain noise shaping parameter;
   - Output: an MDCT coefficient of the mono signal after inverse frequency-domain noise shaping.

f) Inverse MDCT transform: An inverse MDCT module transforms the MDCT coefficient into a time-domain signal based on a windowed control parameter obtained by parsing the bitstream. The input and output parameters are as follows:
   - Input: the MDCT coefficient of the mono signal after inverse frequency-domain noise shaping and the windowed control parameter;
   - Output: a time-domain mono signal.

Figure 4 Mono decoding framework

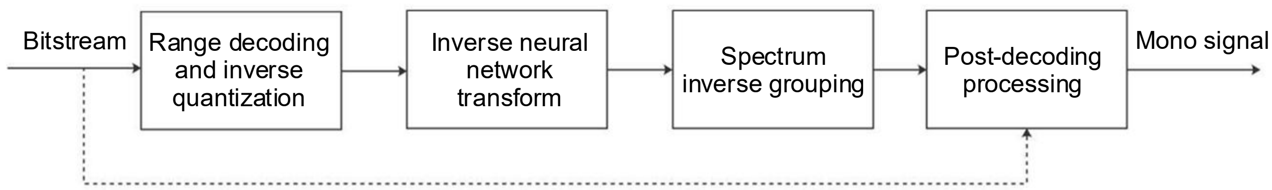
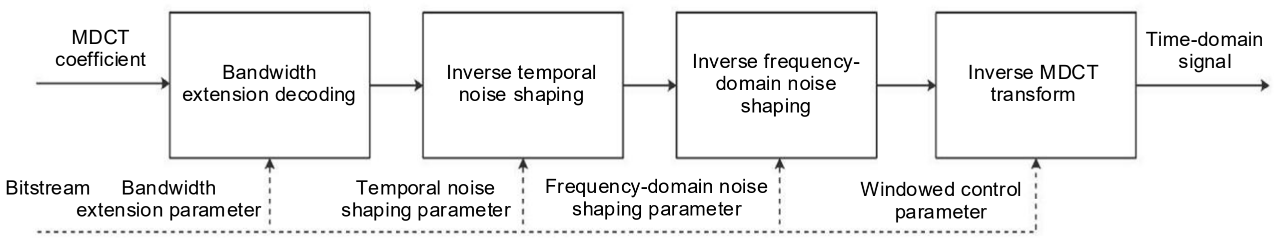


Figure 5 Post-decoding processing framework



The dual-channel stereo decoding includes range decoding, inverse quantization, inverse neural network transform, spectrum inverse grouping, dual-channel stereo upmixing, and post-decoding processing. The decoding framework should comply with Figure 6. A transform-domain coefficient corresponding to an MDCT coefficient is obtained by parsing a bitstream through range decoding and inverse quantization. A reconstructed downmixed channel MDCT coefficient is obtained through inverse neural network transform. A reconstructed left and right channel MDCT coefficient is obtained through dual-channel stereo upmixing. Finally, a time-domain dual-channel stereo signal is obtained through post-decoding processing. The post-decoding processing framework should comply with Figure 5.

The obtaining process is as follows.

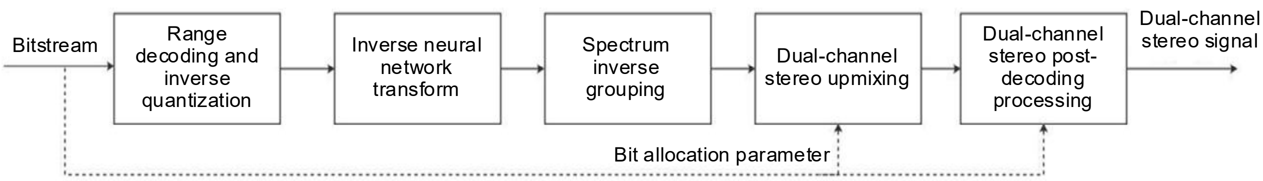
a) Range decoding and inverse quantization. The input and output parameters are as follows:
   - Input: the bitstream;
   - Output: a downmixed channel transform-domain coefficient corresponding to the MDCT coefficient.

b) Inverse neural network transform. The input and output parameters are as follows:
   - Input: the downmixed channel transform-domain coefficient corresponding to the MDCT coefficient;
   - Output: the reconstructed downmixed channel MDCT coefficient.

c) Dual-channel stereo upmixing: A dual-channel stereo upmixing module performs upmixing on the downmixed channel MDCT coefficient based on a downmixing parameter obtained by parsing the bitstream, to obtain a left and right channel signal. The input and output parameters are as follows:
   - Input: the reconstructed downmixed channel MDCT coefficient and the downmixing parameter;
   - Output: the reconstructed left and right channel MDCT coefficient.

d) Dual-channel stereo post-decoding processing: The dual-channel stereo post-decoding processing module performs post-decoding processing for each channel. The input and output parameters are as follows:
   - Input: the reconstructed left and right channel MDCT coefficient, a bandwidth extension parameter, a temporal noise shaping parameter, a frequency-domain noise shaping parameter, and a windowed control parameter;
   - Output: a time-domain dual-channel stereo signal.

Figure 6 Dual-channel stereo decoding framework

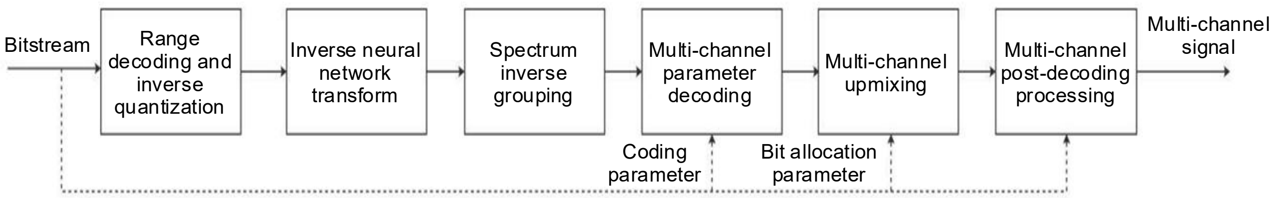


The multi-channel decoding includes range decoding, inverse quantization, inverse neural network transform, spectrum inverse grouping, multi-channel parameter decoding, multi-channel upmixing, and post-decoding processing. The decoding framework should comply with Figure 7. A transform-domain coefficient corresponding to an MDCT coefficient is obtained by parsing a bitstream through range decoding and inverse quantization. A reconstructed downmixed channel MDCT coefficient is obtained through inverse neural network transform. A multi-channel upmixing parameter is obtained by decoding a multi-channel parameter. A reconstructed multi-channel MDCT coefficient is obtained through multi-channel upmixing. Finally, a time-domain multi-channel signal is obtained through post-decoding processing. The post-decoding processing framework should comply with Figure 5.

The obtaining process is as follows.

a) Range decoding and inverse quantization. The input and output parameters are as follows:
   - Input: the bitstream;
   - Output: the downmixed multi-channel transform-domain coefficient corresponding to the MDCT coefficient.
b) Inverse neural network transform. The input and output parameters are as follows:
   - Input: the downmixed multi-channel transform-domain coefficient corresponding to the MDCT coefficient;
   - Output: a reconstructed downmixed multi-channel MDCT coefficient.
c) Multi-channel parameter decoding: A multi-channel parameter decoding module parses the bitstream to obtain a multi-channel signal coding mode parameter. The input and output parameters are as follows:
   - Input: the bitstream;
   - Output: the multi-channel signal coding mode parameter.
d) Multi-channel upmixing: A multi-channel upmixing module performs upmixing on the reconstructed downmixed multi-channel MDCT coefficient based on the multi-channel signal coding mode parameter to obtain the reconstructed multi-channel signal MDCT coefficient. The input and output parameters are as follows:
   - Input: the reconstructed downmixed multi-channel MDCT coefficient and the multi-channel signal coding mode parameter;
   - Output: the reconstructed multi-channel signal MDCT coefficient.
e) Multi-channel post-decoding processing: The multi-channel post-decoding processing module performs post-decoding processing for each channel. The input and output parameters are as follows:
   - Input: the reconstructed multi-channel signal MDCT coefficient, a bandwidth extension parameter, a temporal noise shaping parameter, a frequency-domain noise shaping parameter, and a windowed control parameter;
   - Output: a time-domain multi-channel signal.

Figure 7 Multi-channel decoding framework



HOA decoding includes range decoding, inverse quantization, inverse neural network transform, spectrum inverse grouping, transmission channel parameter decoding, HOA upmixing, transmission channel post-decoding processing, and HOA spatial decoding. The HOA decoding framework should comply with Figure 8. HOA spatial decoding includes sound field component synthesis and HOA synthesis. The HOA spatial decoding framework should comply with Figure 9. A transform-domain coefficient corresponding to an MDCT coefficient is obtained by parsing a bitstream through range decoding and inverse quantization. A reconstructed downmixed channel MDCT coefficient is obtained through inverse neural network transform. An HOA upmixing parameter is obtained through transmission channel parameter decoding. A reconstructed transmission channel signal MDCT coefficient is obtained through HOA upmixing. A time-domain transmission channel signal (the transmission channel signal includes the virtual speaker signal and the residual signal, the virtual speaker signal is a directional component in the sound field, and the residual signal is a component other than the directional component in the sound field) is obtained through transmission channel post-decoding processing. Finally, a reconstructed HOA signal is obtained through HOA spatial decoding. The post-decoding processing framework should comply with Figure 5.

The obtaining process is as follows.

a) Range decoding and inverse quantization. The input and output parameters are as follows:
   - Input: the bitstream;
   - Output: a downmixed channel transform-domain coefficient corresponding to the MDCT coefficient.

b) Inverse neural network transform:
   - Input: the downmixed channel transform-domain coefficient corresponding to the MDCT coefficient;
   - Output: the reconstructed downmixed channel MDCT coefficient.

c) Transmission channel parameter decoding: A transmission channel parameter decoding module parses the bitstream to obtain a transmission channel signal coding mode parameter. The input and output parameters are as follows:
   - Input: the bitstream;
   - Output: the transmission channel signal coding mode parameter.

d) HOA upmixing: An HOA upmixing module performs upmixing on the reconstructed downmixed channel MDCT coefficient based on the transmission channel signal coding mode parameter to obtain the reconstructed transmission channel signal MDCT coefficient. The input and output parameters are as follows:
   - Input: the reconstructed downmixed channel MDCT coefficient and the transmission channel signal coding mode parameter;
   - Output: the reconstructed transmission channel signal MDCT coefficient.

e) Transmission channel post-decoding processing: The transmission channel post-decoding processing module performs post-decoding processing for each channel. The input and output parameters are as follows:
   - Input: the reconstructed transmission channel signal MDCT coefficient, a bandwidth extension parameter, a temporal noise shaping parameter, a frequency-domain noise shaping parameter, and a windowed control parameter;
   - Output: a time-domain transmission channel signal.

f) Sound field component synthesis: A sound field component synthesis module configures a decoder based on a configuration parameter, determines a coefficient of a virtual speaker based on a sound field component parameter obtained by parsing the bitstream, and synthesizes a primary sound field signal based on a virtual speaker signal in the transmission channel signal. The input and output parameters are as follows:
   - Input: the transmission channel signal and the sound field component parameter;

- Output: the primary sound field signal.

g) HOA synthesis: An HOA synthesis module synthesizes an HOA signal based on a residual signal in the transmission channel signal, a remaining component parameter obtained by parsing the bitstream, and the primary sound field signal. The input and output parameters are as follows:
   - Input: the transmission channel signal, the primary sound field signal, and the remaining component parameter;
   - Output: the HOA signal.

Figure 8 HOA decoding framework

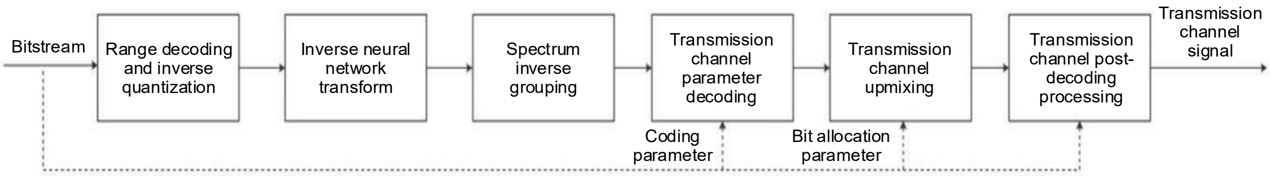
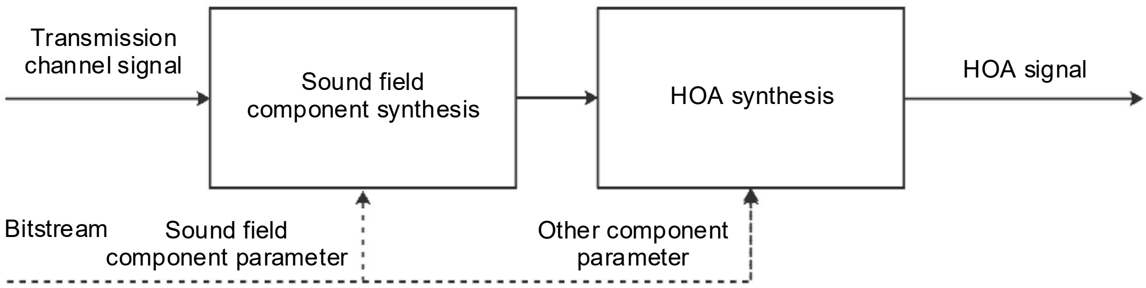


Figure 9 HOA spatial decoding framework



7.2 Syntax, Semantics, and Decoding Process of Bitstream Demultiplexing

7.2.1 Syntax

The syntax of a general bit rate audio bitstream should comply with provisions in Table 7.

Table 7 Syntax of general bit rate audio bitstream

Syntax of general bit rate audio bitstream	Number of bits	Mnemonic
ga_co_raw_data_block() {	—	—
Avs3MetadataDec()	—	—
switch(codecFormat) {	—	—
case 0x0: Avs3MonoDec()	—	—
case 0x1: Avs3StereoDec()	—	—

Syntax of general bit rate audio bitstream	Number of bits	Mnemonic
case 0x2: Avs3McDec()	—	—
case 0x3: Avs3HoaDec()	—	—
}	—	—
}	—	—

The syntax of mono decoding should comply with provisions in Table 8.

Table 8 Syntax of mono decoding

Syntax of mono decoding	Number of bits	Mnemonic
Avs3MonoDec() {	—	—
DecodeCoreSideBits()	—	—
DecodeGroupBits()	—	—

Table 8 (continued)

Syntax of mono decoding	Number of bits	Mnemonic
DecodeQcBits()	—	—
Avs3InverseQC()	—	—
Avs3PostSynthesis()	—	—
}	—	—

The syntax of dual-channel stereo decoding should comply with provisions in Table 9.

Table 9 Syntax of dual-channel stereo decoding

Syntax of dual-channel stereo decoding	Number of bits	Mnemonic
Avs3StereoDec() {	—	—
for(ch = 0; ch < 2; ch++) {	—	—
DecodeCoreSideBits()	—	—
}	—	—
for(ch = 0; ch < 2; ch++) {	—	—
DecodeGroupBits()	—	—
}	—	—
DecodeStereoSideBits()	—	—

Syntax of dual-channel stereo decoding	Number of bits	Mnemonic
StereoBitsAllocation()	—	—
for(ch = 0; ch < 2; ch++) {	—	—
DecodeQcBits()	—	—
}	—	—
Avs3InverseQC()	—	—
StereoInvMsProcess()	—	—
for(ch = 0; ch < 2; ch++) {	—	—
Avs3PostSynthesis()	—	—
}	—	—
}	—	—

The syntax of multi-channel decoding should comply with provisions in Table 10.

Table 10 Syntax of multi-channel decoding

Syntax of multi-channel decoding	Number of bits	Mnemonic
Avs3McDec() {	—	—
for(ch = 0; ch < numChans; ch++) {	—	—
DecodeCoreSideBits()	—	—
}	—	—
for(ch = 0; ch < numChans; ch++) {	—	—
DecodeGroupBits()	—	—
}	—	—
DecodeMcSideBits()	—	—
McBitsAllocation()	—	—
for(ch = 0; ch < numChans; ch++) {	—	—
DecodeQcBits()	—	—
}	—	—
Avs3InverseQC()	—	—

Table 10 (continued)

Syntax of multi-channel decoding	Number of bits	Mnemonic
Avs3McacDec()	—	—
for(ch = 0; ch < numChans; ch++) {	—	—
Avs3PostSynthesis()	—	—
}	—	—
}	—	—

The syntax of HOA decoding should comply with provisions in Table 11.

Table 11 Syntax of HOA decoding

Syntax of HOA decoding	Number of bits	Mnemonic
Avs3HoaDec() {	—	—
for(ch = 0; ch < numChans; ch++) {	—	—
DecodeCoreSideBits()	—	—
}	—	—
for(ch = 0; ch < numChans; ch++) {	—	—
DecodeGroupBits()	—	—
}	—	—
DecodeHoaSideBits()	—	—
HoaSplitBytesGroup()	—	—
for(ch = 0; ch < numChans; ch++) {	—	—
DecodeQcBits()	—	—
}	—	—
Avs3InverseQC()	—	—
Avs3HoaInverseDMX()	—	—
for(ch = 0; ch < numChans; ch++) {	—	—
Avs3PostSynthesis()	—	—
}	—	—
HoaPostSynthesisFilter()	—	—
}	—	—

The syntax of core decoder side information should comply with provisions in Table 12.

Table 12 Syntax of core decoder side information

Syntax of core decoder side information	Number of bits	Mnemonic
DecodeCoreSideBits() {	—	—
transformType	2	uimsbf
DecodeFdShapingSideBits()	—	—
DecodeTnsSideBits()	—	—
if(bwePresent == 1) {	—	—
DecodeBweSideBits()	—	—
}	—	—
}	—	—

The configuration of the windowed control parameter should comply with provisions in Table 13.

Table 13 Configuration of windowed control parameter

transformType	Windowed control parameter
0x0	Long window
0x1	Short window
0x2	Cut-in window
0x3	Cut-out window

7.2.2 Semantics

Avs3MetadataDec()

Metadata decoding.

Avs3MonoDec()

Mono decoding.

Avs3StereoDec()

Dual-channel stereo decoding.

Avs3McDec()

Multi-channel decoding.

Avs3HoaDec()

FOA/HOA decoding.

DecodeCoreSideBits()

Parsing core decoder side information.

DecodeGroupBits()

Spectrum inverse grouping.

DecodeStereoSideBits()

Parsing dual-channel stereo side information

DecodeMcSideBits()

Parsing multi-channel side information.

DecodeHoaSideBits()

Parsing FOA/HOA side information.

StereoBitsAllocation()

Dual-channel stereo bit allocation.

McBitsAllocation()

Multi-channel bit allocation.

HoaSplitBytesGroup()

FOA/HOA bit allocation.

StereoInvMsProcess()

Dual-channel stereo upmixing processing.

Avs3McacDec()

Multi-channel decoding processing.

Avs3HoaInverseDMX()

FOA/HOA decoding processing.

DecodeQcBits()

Parsing quantization coding side information.

Avs3InverseQC()

Range decoding and inverse quantization.

Avs3PostSynthesis()

Post-decoding processing.

HoaPostSynthesisFilter()

HOA spatial decoding.

codecFormat

It indicates a decoding mode, and is determined by channel_number_index (channel_number_index) in Table A.8 in Annex A. When a channel is set to mono, codecFormat is 0. When a channel is set to dual-channel stereo, codecFormat is 1. When a channel is set to a multi-channel configuration, codecFormat is 2. When a channel is set to a FOA/HOA configuration, codecFormat is 3.

numChans

The number of channels of an audio signal. When coding_profile is 0, channel_number corresponding to channel_number_index (channel_number_index) in Table A.8 in Annex A is reused.

transformType

2 bits. It indicates a windowed control parameter. The configuration of a windowed control parameter should comply with the provisions in Table 13. A short window is a sine window with a length of 256 points. A long window is a sine window with a length of 2048 points. The first 1024 points of a cut-in window are the same as the long window, and the last 1024 points are composed of 448 points of 1, a short window of 128 points, and 448 points of 0. The first 1024 points of a cut-out window are composed of 448 points of 0, a short window of 128 points, and 448 points of 1, and the last 1024 points are the same as a long window.

DecodeFdShapingSideBits()

Parsing frequency-domain noise shaping side information.

DecodeTnsSideBits()

Parsing temporal noise shaping side information.

DecodeBweSideBits().

Parsing bandwidth extension decoding side information

bwePresent

It indicates whether to enable bandwidth extension. For conditions of enabling bandwidth extension, refer to section 7.9.3.5.

7.2.3 Decoding process

A general bit rate audio bitstream (ga_co_raw_data_block) is a basic unit included in AASF and AATF coded bitstreams. A sample of one frame can be obtained by decoding ga_co_raw_data_block, and a bit rate of one frame can be obtained by decoding a frame header. For details about AASF and AATF formats, refer to Annex A. The bitstream demultiplexing determines a decoding mode (one of mono decoding, dual-channel stereo decoding, multi-channel decoding, FOA/HOA decoding, object decoding, and channel and object hybrid decoding) by parsing parameters in AASF and AATF headers, and obtains side information by decoding the bitstream ga_co_raw_data_block for subsequent core decoding and HOA spatial decoding processes. During decoding of ga_co_raw_data_block in different decoding modes, the decoder first parses metadata Avs3MetadataDec(), and then parses audio data. The audio data decoding mode is determined based on coding_profile, channel_number_index, and soundBedType parameters obtained by parsing aasf_frame_header() or aatf_frame_header(). Details are as follows.

- When coding_profile is 0 and channel_number_index is 0, the decoder selects mono decoding, and codecFormat is 0. Avs3MonoDec() first parses core decoder side information DecodeCoreSideBits(), performs spectrum inverse grouping DecodeGroupBits(), then parses quantization coding side information DecodeQcBits(), performs range decoding and inverse quantization Avs3InverseQC(), and finally performs post-decoding processing Avs3PostSynthesis(), to obtain a decoded mono signal.

- When coding_profile is 0 and channel_number_index is 1, the decoder selects dual-channel stereo decoding, and codecFormat is 1. Avs3StereoDec() first parses core decoder side information DecodeCoreSideBits() and performs spectrum inverse grouping DecodeGroupBits() on each channel, then parses dual-channel stereo side information DecodeStereoSideBits(), performs dual-channel stereo bit allocation StereoBitsAllocation(), parses quantization coding side information DecodeQcBits() for each channel, performs range decoding and inverse quantization Avs3InverseQC(), performs dual-channel stereo upmixing processing StereoInvMsProcess(), and finally performs post-decoding processing Avs3PostSynthesis() on each channel, to obtain a decoded dual-channel stereo signal.

- When coding_profile is 0 and channel_number_index is greater than 1, the decoder selects multi-channel decoding, and codecFormat is 2. Avs3McDec() first parses core decoder side information DecodeCoreSideBits() and performs spectrum inverse grouping DecodeGroupBits() on each channel, then parses multi-channel side information DecodeMcSideBits(), performs multi-channel bit allocation McBitsAllocation(), parses quantization coding side information DecodeQcBits() for each channel, performs range decoding and inverse quantization Avs3InverseQC(), performs multi-channel decoding processing Avs3McacDec(), and finally performs post-decoding processing Avs3PostSynthesis() on each channel, to obtain a decoded multi-channel signal.

- When coding_profile is 1 and soundBedType is 0, the decoder selects object decoding. When object_channel_number is 0, codecFormat is 0. When object_channel_number is 1, codecFormat is 1. When object_channel_number is greater than 1, codecFormat is 2. When the object decoding is performed according to a method the same as that for channel decoding, that is, the number of channels of an object is 1, a method the same as that for mono decoding is used. When the number of channels of an object is 2, a method the same as that for dual-channel stereo decoding is used. When the number of channels of an object is greater than 2, a method the same as that for multi-channel decoding is used.

- When coding_profile is 1 and soundBedType is 1, the decoder selects channel and object hybrid decoding, codecFormat is 2, the total number of channels for channel and object is greater than or equal to 3. The channel and object hybrid decoding is performed according to a method the same as that for multi-channel decoding.

- When coding_profile is 2, the decoder selects FOA/HOA decoding, codecFormat is 3. Only core decoding is performed for FOA. The HOA decoding includes two parts: core decoding and spatial decoding. The core decoding obtains a virtual speaker signal and a residual signal through decoding. A spatial decoder obtains an HOA signal by decoding the virtual speaker signal and the residual signal. Avs3HoaDec() first parses core decoder side information DecodeCoreSideBits() and performs spectrum inverse grouping DecodeGroupBits() on each virtual speaker signal and residual signal, then parses side information DecodeHoaSideBits() for the virtual speaker signal and residual signal, performs bit allocation HoaSplitBytesGroup() on the virtual speaker signal and residual signal, parses quantization coding side information DecodeQcBits() for each channel, performs range decoding and inverse quantization Avs3InverseQC(), performs decoding processing Avs3HoaInverseDMX() on the virtual speaker signal and residual signal, performs post-decoding processing Avs3PostSynthesis() on each virtual speaker signal and residual signal, and finally performs HOA spatial decoding HoaPostSynthesisFilter(), to obtain a decoded HOA signal.

7.3 Syntax, Semantics, and Decoding Processes of Range Decoding, Inverse Quantization, and Inverse Neural Network Transform

7.3.1 Syntax

The syntax of quantization and encoding side information should comply with provisions in Table 14.

Table 14 Syntax of quantization and encoding side information

Syntax of quantization and encoding side information	Number of bits	Mnemonic
DecodeQcBits() {	—	—
if (nn_type == 0){	—	—
isFeatAmplified	1	uimsbf
scaleQIdx	7	uimsbf
} else if (nn_type == 1) {	—	—
scaleQIdxLc	8	uimsbf
}	—	—
if(numGroups == 1) {	—	—
nfParamQIdx[0]	3	uimsbf
}else if(numGroups == 2) {	—	—
nfParamQIdx[0]	3	uimsbf
nfParamQIdx[1]	3	uimsbf
}	—	—
contextNumBytes	8	uimsbf
contextBitstream	—	uimsbf
baseBitstream	—	uimsbf
}	—	—
Note 1: nn_type indicates a neural network configuration, and is obtained from aasf_header() or aatf_header(). Note 2: The number of bytes of contextBitstream is contextNumBytes. Note 3: The number of bytes of baseBitstream is channelBytes – contextNumBytes.		

7.3.2 Semantics

isFeatAmplified

1 bit. It indicates whether a transform-domain coefficient obtained through neural network transform is scaled up or down before quantization. 0 indicates scaling down, and 1 indicates scaling up.

scaleQIdx

7 bits. It indicates a quantization index of a scale adjustment factor of a transform-domain coefficient obtained through neural network transform. A value of an adjustment factor of inverse quantization can be obtained from this variable.

scaleQIdxLc

8 bits. It indicates a quantization index of a scale adjustment factor of an MDCT spectral coefficient. A value of an adjustment factor of inverse quantization can be obtained from this variable.

nfParamQIdx

3 bits. It indicates a quantization index of a noise filling parameter corresponding to each of transform-domain coefficients of two groups. The noise filling parameter of inverse quantization can be obtained from this variable.

contextNumBytes

8 bits. It indicates the number of occupied bytes of a context part in a range coded bitstream.

contextBitstream

In a range coded bitstream, contextNumBytes indicates the number of occupied bytes of a context bitstream.

baseBitstream

In a range coded bitstream, channelBytes – contextNumBytes indicates the number of occupied bytes of a base bitstream.

numGroups

It indicates the number of groups in an MDCT spectrum. The maximum value is 2. The parsing of numGroups is shown in the syntax of DecodeGroupBits() in section 7.4.1.

channelBytes

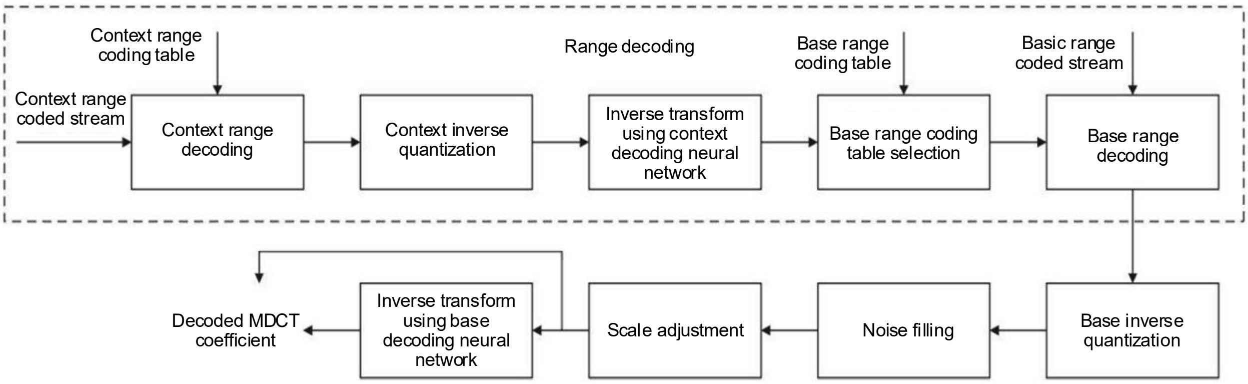
It indicates the number of bytes allocated to the current channel bitstream.

7.3.3 Decoding Process

7.3.3.1 Overview

In the basic configuration, the decoding process includes range decoding, base inverse quantization, noise filling, scale adjustment, and inverse transform using a base decoding neural network. The range decoding includes context range decoding, context inverse quantization, and inverse transform using a context decoding neural network, base range coding table selection, base range decoding. The decoding process should be consistent with that in FIG. 10. Range decoding relates to both the context and base parts. First, range decoding and inverse quantization are performed on the context part, and then after processing of a context decoding neural network, decoded context information is obtained. A base range coding table is selected based on the context information to perform range decoding and inverse quantization on the base part of the range coded bitstream and then obtain the decoded transform-domain coefficient. After the noise filling, scale adjustment, and inverse transform using the base decoding neural network, the decoded MDCT coefficient is obtained.

Figure 10 Decoding processes of range decoding, inverse quantization, and inverse neural network transform



In a low-complexity configuration, the decoding process does not include a base decoding neural network, and the decoded MDCT coefficient can be obtained after range decoding, base inverse quantization, and scale adjustment.

7.3.3.2 Context Range Decoding

The context part of the range coded bitstream is decoded based on the context range coding table to obtain a quantization index of a transform-domain coefficient of a context neural network. The context range coding table is obtained through pre-training. Refer to Table B.1.

7.3.3.3 Context inverse quantization

Inverse quantization is performed on the quantization index of the transform-domain coefficient of the context neural network to obtain a quantized transform-domain coefficient of the context neural network. Linear scalar quantization is used as the quantization method.

7.3.3.4 Inverse Transform Using Context Decoding Neural Network

An inversely quantized transform-domain coefficient of the context neural network is input into the context decoding neural network for inverse transform to obtain the decoded context information. The context decoding neural network is used to perform the inverse neural network transform on the decoded and dequantized transform-domain coefficient of the context neural network to obtain selected information of the base range coding table. The decoded and dequantized transform-domain coefficient of the context neural network is input into the context decoding neural network. A transposed CNN is a composition unit of the context decoding neural network.

The configuration of the structure of the transposed CNN should comply with provisions in Table 15.

Table 15 Configuration of structure of transposed CNN

Item	Value
Number of CNN layers	3
Convolution kernel size	3, 3, 3
Stride	2, 2, 1

Item	Value
Number of CNN channels	16, 16, 16
Activation function	ReLu, ReLu, None
HasBias	1, 1, 1

Stride is a convoluted step parameter of CNN. HasBias indicates whether there is an offset parameter at a specific layer. '1' indicates yes. '0' indicates no. A rectified linear unit (ReLu) is used as the activation function. For the definition of the ReLU mathematical function, refer to formula (2). None indicates that there is no activation function at the specific layer.

The parameter of the transposed CNN at each layer of the context decoding neural network should comply with provisions in Table B.2 to Table B.7. The input of the context decoding neural network is a transform-domain coefficient with 16 CNN channels and 16 dimensions per CNN channel. The output is the selected information of the base range coding table with 16 CNN channels and 64 dimensions per CNN channel.

7.3.3.5 Base Range Coding Table Selection

A table corresponding to base range coding is selected from the several base range coding tables based on the decoded context information. Range decoding is performed on the base range coded bitstream, to obtain a quantized transform-domain coefficient of the base neural network.

A base range coding table selection process may be represented as follows: For each dimension of the transform-domain coefficient of the base coding neural network, a minimum standard deviation greater than or equal to corresponding decoding context information is searched for in a standard deviation table corresponding to the base range coding, and an index corresponding to the standard deviation is a base range coding table index. The standard deviation table corresponding to the base range coding should comply with provisions in Table B.8. Code tables used for the base range coding are several pre-trained fixed code tables, and should comply with provisions in Table B.9.

For example, it is assumed that the value of the nth dimension in the decoding context information is 0.45, a minimum standard deviation value greater than or equal to 0.45 is searched for in a standard deviation table (that is, Table B.8) corresponding to base range coding, and an index corresponding to the standard deviation value is 13 (it is assumed that a start sequence number is 1). Then the 13th base range coding table (that is, the 13th row in Table B.9) is selected, and range decoding is performed on the nth dimension of the transform-domain coefficient of the base coding neural network.

7.3.3.6 Base Range Decoding

The base part of the range coded bitstream is decoded based on the base range coding code table selected in section 7.3.3.5. In the basic configuration, a quantization index of a transform-domain coefficient of a base neural network is obtained, and in a low-complexity configuration, a quantization index of an MDCT spectral coefficient is obtained.

7.3.3.7 Base Inverse Quantization

Inverse quantization is performed on the quantization index of the transform-domain coefficient of the base neural network in the basic configuration, to obtain a quantized transform-domain coefficient of the base neural network. Inverse quantization is performed on the quantization index of the MDCT spectral coefficient in the low-complexity configuration, to obtain a quantized MDCT spectral coefficient. The linear scalar quantization is used as the quantization method.

7.3.3.8 Noise Filling

Noise filling adds noise to the quantized transform-domain coefficient of the base neural network obtained at the decoder, to compensate for quantized noise introduced in the quantization process.

For each channel of each frame, a quantization index *nfParamQIdx* of the noise filling parameter may have one or two values. If there is one group in the MDCT spectrum, *nfParamQIdx* has one value. If there are two groups in the MDCT spectrum, *nfParamQIdx* has two values.

The following noise filling process is performed on the transform-domain coefficient of the base neural network corresponding to each group:

a) A noise component noise randomly distributed in a range of [–1, 1] is generated;

b) The noise component noise is multiplied by a noise filling parameter *nfParamQ* to obtain an adjusted noise component. The noise filling parameter *nfParamQ* is obtained by dequantizing the quantization index *nfParamQIdx*. Refer to formula (3).

$$nfParamQ = \frac{nfParamQIdx}{23.34} \quad \text{......(3)}$$

c) In the basic configuration, if the transform-domain coefficient of the base neural network is quantized to 0, it is filled with the adjusted noise component. In the low-complexity configuration, noise is added to the quantized MDCT spectral coefficient obtained at the decoder, to compensate for quantized noise introduced in the quantization process.

7.3.3.9 Scale Adjustment

In the basic configuration, scale adjustment is performed on a coefficient obtained through the base neural network transform. The basic process of scale adjustment is as follows.

a) A dequantized scale adjustment factor *featureScale* is obtained based on a feature amplification flag isFeatAmplified and a scale factor quantization index scaleQIdx. Pseudocode of this process is shown as follows:

```
if (isFeatAmplified == 0){
   featureScale = scaleQIdx / 127.0
}else {
   featureScale = pow(10.0, scaleQIdx / 86.0);
}
```

b) The transform-domain coefficient of the base neural network is divided by the scale factor *featureScale* to obtain a transform-domain coefficient after the scale adjustment. In the low-complexity configuration, scale adjustment is performed on an MDCT spectral coefficient obtained through the noise filling. The basic process of scale adjustment is as follows:

a) A dequantized scale adjustment factor *featureScale* is obtained based on a scale factor quantization index *scaleQIdxLc*. For details about *featureScale* calculation, refer to formula (4).

$$featureScale = 10^{(scaleQIdxLc - 255.0)\ /\ 31.875} \quad (4)$$

b) A decoded MDCT coefficient obtained through the noise filling is divided by the scale factor *featureScale* to obtain a decoded MDCT coefficient.

7.3.3.10 Inverse Transform Using Base Decoding Neural Network

The transform-domain coefficient of the base neural network obtained through the scale adjustment is input into the base decoding neural network to calculate the MDCT spectrum reconstructed at the decoder. The CNN is the basic structure of the base decoding neural network.

The structure parameter of the base decoding neural network should comply with provisions in Table 16.

Table 16 Structure parameter of base decoding neural network

Item	Value
Number of CNN layers	4
Convolution kernel size	5, 5, 5, 5
Stride	2, 2, 2, 2
Number of channels	8, 4, 2, 1
Activation function	IGDN, IGDN, IGDN, None
HasBias	1, 1, 1, 1

Stride is a convoluted step parameter of CNN. HasBias indicates whether there is an offset parameter at a specific layer. '1' indicates yes. '0' indicates no. IGDN[1] is used as the activation function. None indicates that there is no activation function at the specific layer.

The parameter of the transposed CNN at each layer of the base decoding neural network should comply with provisions in Table B.10 to Table B.23.

7.4 Syntax, Semantics, and Decoding Process of Spectrum Inverse Grouping

7.4.1 Syntax

The syntax of spectrum inverse grouping should comply with provisions in Table 17.

Table 17 Syntax of spectrum inverse grouping

Syntax of spectrum inverse grouping	Number of bits	Mnemonic
DecodeGroupBits() {	—	—
if(transformType == 0x1) {	—	—
numGroups	1	uimsbf
numGroups += 1	—	—
if(numGroups == 2) {	—	—
for (i = 0; i < 8; i++) {	—	—
groupIndicator[i]	1	uimsbf
}	—	—
}	—	—
} else {	—	—
numGroups = 1	—	—
for (i = 0; i < 8; i++) {	—	—
groupIndicator[i] = 0	—	—
}	—	—
}	—	—
}	—	—

7.4.2 Semantics

numGroups

1 bit. It indicates the number of groups in the MDCT spectrum of the current frame.

groupIndicator

1 bit. It indicates to which group the i-th block of the current short frame belongs. '0' indicates the transient group. '1' indicates the other group.

7.4.3 Decoding Process

The basic idea of MDCT spectrum grouping is as follows: When the windowed control parameter is a short window, the MDCT spectrum corresponding to each short window is called a short block. The MDCT spectra of eight short blocks are divided into two groups: a transient group including transient short blocks and the other group excluding transient short blocks. The MDCT spectra of the two groups are separately interleaved and concentrated into a 1024-point MDCT spectrum (consistent with the length of a long-frame spectrum). Then neural network transform, quantization, and range coding are performed on it.

A process of spectrum inverse grouping at the decoder is as follows: For a short frame, if the number of groups numGroups of the current frame is equal to 2, intra-group de-interleaving is performed on the MDCT spectra of the two groups obtained through decoding (de-interleaving is performed on MDCT coefficients of the transient group and MDCT coefficients of the other group separately), to obtain MDCT spectra of the eight short blocks obtained through intra-group de-interleaving, and inverse grouping arrangement is performed on the MDCT spectra of the eight short blocks obtained through intra-group de-interleaving based on a position indicated by groupIndicator, to obtain MDCT spectra of the eight short blocks arranged according to a sequence.

The MDCT spectra obtained through spectrum inverse grouping are used as the input of the upmixing in each mode.

Example:

The numGroups of the current short frame is equal to 2, and groupIndicator is [1, 1, 1, 0, 0, 0, 1, 1]. In other words, the fourth, the fifth, and the sixth short blocks in the eight short blocks in the current frame are transient blocks, and the first, the second, the third, the seventh, and the eighth short blocks are other blocks.

In this case, the grouped MDCT spectra obtained through decoding are arranged according to a sequence of [4, 5, 6, 1, 2, 3, 7, 8]. The spectra of the transient group (including the fourth, the fifth, and the sixth short blocks) are placed ahead of the spectra of the other group (including the first, the second, the third, the seventh, and the eighth short blocks).

Recovering the normal spectrum sequence means to arrange the spectrum of each short block according to a time sequence. Specifically, the grouped MDCT spectra obtained through decoding are rearranged according to the sequence indicated by groupIndicator, to obtain spectral coefficients arranged according to a sequence from the first to the eighth short blocks.

7.5 Syntax, Semantics, and Decoding Processes of Dual-channel Stereo Bit Allocation and Upmixing

7.5.1 Syntax

The syntax of dual-channel stereo side information should comply with provisions in Table 18.

Table 18 Syntax of dual-channel stereo side information

Syntax of dual-channel stereo side information	**Number of bits**	**Mnemonic**
DecodeStereoSideBits() {	—	—
if(useMcr == 0) {	—	—
isMs	1	uimsbf
if(isMs == 1) {	—	—
IldQIdx	4	uimsbf
}	—	—
bitsRatio	3	uimsbf
}	—	—
if(useMcr == 1) {	—	—
for(i = 0; i < vqVecNum[isShortWin]; i++) {	—	—
vqIdx[0][i]	—	uimsbf

Syntax of dual-channel stereo side information	Number of bits	Mnemonic
vqIdx[1][i]	—	uimsbf
}	—	—
}	—	—
}	—	—
Note: The numbers of bits of vqIdx[0][i] and vqIdx[1][i] are equal to that of vqNumBits[isShortWin].		

7.5.2 Semantics

useMcr

It indicates whether MCR dual-channel stereo processing is used. When the dual-channel stereo bit rate is less than or equal to 32 kb/s, useMcr is equal to 1. When the dual-channel stereo bit rate is greater than 32 kb/s, useMcr is equal to 0.

isMs

1 bit. It indicates whether M/S dual-channel stereo processing is enabled for the current frame.

IldQIdx

4 bits. It indicates an amplitude difference between the left and right channels.

bitsRatio

3 bits. It indicates a bit allocation ratio of two downmixed channels.

isShortWin

It indicates whether the left channel is a short window. If the left channel is a short window, isShortWin is equal to 1. If not, isShortWin is equal to 0.

vqVecNum[isShortWin]

It indicates the number of sub-vectors in vector quantization of the MCR rotation angle parameters, and is determined by isShortWin that indicates whether the left channel is a short window.

vqIdx[0][i]

It indicates a vector quantization index of the i-th sub-vector in the MCR rotation angle parameter vector corresponding to the even-number MDCT spectrum.

vqIdx[1][i]

It indicates a vector quantization index of the i-th sub-vector in the MCR rotation angle parameter vector corresponding to the odd-number MDCT spectrum.

7.5.3 Decoding Process

7.5.3.1 Overview

The dual-channel stereo upmixing includes the MCR and M/S upmixing modes. The MCR upmixing mode includes MCR side information decoding and MCR upmixing. The M/S upmixing mode includes M/S bit allocation and M/S upmixing. When the MCR upmixing mode is used, MDCT spectrum information transmitted in the bitstream is only MDCT spectrum information of one channel obtained through MCR rotation processing. MDCT spectra of two channels need to be recovered based on side information of the MCR dual-channel stereo mode transmitted in the bitstream. When the M/S upmixing mode is used, the side information of the dual-channel stereo mode transmitted in the bitstream is used to obtain information such as a flag that indicates whether M/S downmixing is enabled for the current frame, an amplitude difference between left and right channels, and a bit allocation ratio of a downmixed channel. After other side information in the current frame is removed, the total number of bits of the remaining information for quantization and coding as well as the bit allocation ratio information are used to calculate the respective downmixed bit allocation results channelBytes of the two channels. Range decoding, inverse quantization, and inverse neural network transform are performed on the bit allocation results to obtain the decoded MDCT coefficients of the downmixed channels. M/S upmixing is performed based on the isMs flag, and energy adjustment is performed based on ILD information, to obtain the decoded MDCT coefficients of the left and right channels.

7.5.3.2 MCR Side Information Decoding

The MCR side information is MCR rotation angle parameters of odd and even spectra of core bands. The core band should be a spectrum below a start frequency of a high-frequency spectrum in section 7.9.3.6. In the MCR upmixing mode, MCR processing is performed only on the MDCT spectra of the left and right channels. The decoding process should comply with the requirements of couple_channel_element() decoding in section 7.4 in GB/T 33475.3-2018. The MDCT spectra of the left and right channels are divided into the odd and even spectra, each of which is further divided into 18 sub-bands. Definitions of sub-band boundary and the number of sub-band frequencies should comply with provisions in Table 19 and Table 20. For example, the first sub-band starts from the sequence number 0 sub-band boundary and ends with the sequence number 1 sub-band boundary. The corresponding sub-band boundary covers from the frequency 0 to the frequency 3, and the number of sub-band frequencies is 4.

A configuration of MCR sub-band boundaries should comply with provisions in Table 19.

Table 19 Configuration of MCR sub-band boundaries

Sub-band boundary sequence number	Sub-band boundary
0	0
1	4
2	8
3	12
4	16
5	22
6	28
7	34
8	40
9	48
10	56

Sub-band boundary sequence number	Sub-band boundary
11	64
12	76
13	88
14	100
15	116
16	132
17	154
18	176

The configuration of the number of MCR sub-band frequencies should comply with provisions in Table 20.

Table 20 Configuration of the number of MCR sub-band frequencies

Sub-band sequence number	Number of sub-band frequencies
1	4
2	4
3	4
4	4
5	6
6	6
7	6
8	6
9	8
10	8
11	8
12	12
13	12
14	12
15	16

Table 20 (continued)

Sub-band sequence number	Number of sub-band frequencies
16	16
17	22
18	22

The MCR rotation angle parameters are obtained corresponding to odd and even spectra of each sub-band. In the vector quantization of the MCR rotation angle parameters, rotation angle parameters of 18 sub-bands are divided into 6 sub-vectors (that is, vqVecNum is equal to 6), and each sub-vector includes rotation angle parameters of 3 sub-bands.

The vector quantization of the rotation angle parameters has two configurations, which respectively correspond to a short window type and a non-short window type (such as a long window, cut-in window, and cut-out window) of the left channel of the current frame. In other words, isShortWin is equal to 1 or 0. When isShortWin is equal to 1, the number of vector quantization bits of each rotation angle parameter sub-vector is 8 (that is, the numbers of bits of vqIdx[0][i] and vqIdx[1][i] are 8). In other words, the size of the vector quantization code table is 256. The vector quantization code table should comply with provisions in B.154 in GB/T 33475.3-2018. When isShortWin is equal to 0, the number of vector quantization bits of each rotation angle parameter sub-vector is 9 (that is, the numbers of bits of vqIdx[0][i] and vqIdx[1][i] are 9). In other words, the size of the vector quantization code table is 512. The vector quantization code table should comply with provisions in B.155 in GB/T 33475.3-2018.

7.5.3.3 MCR Upmixing

The parameter stereo method is used for MCR upmixing, and only a core band MDCT coefficient of one channel obtained through MCR transform is transmitted. In the decoding process, upmixing needs to be performed based on a core band MDCT coefficient of one channel and MCR side information that are obtained through decoding, to obtain decoded MDCT coefficients of core bands of the left and right channels.

First, an MDCT spectrum of one channel obtained through the MCR transform obtained by decoding the bitstream is replicated, to obtain the MDCT spectra of the left and right channels obtained through the MCR transform. The inverse MCR transform should comply with the requirements of couple_channel_element() decoding in section 7.4 in GB/T 33475.3-2018, to obtain the core band decoded MDCT spectra of the left and right channels.

7.5.3.4 M/S Bit Allocation

An M/S bit allocation module is mainly used to allocate the remaining available bits after other side information is removed to the two downmixed channels based on the bit allocation ratio parameter obtained by decoding the bitstream. Then, subsequent range decoding, inverse quantization, and inverse neural network transform are performed.

First, calculate the quantity of remaining available bits after other side information is removed from the current frame. *availableBits* indicates the quantity. The following pseudocode is commonly used to calculate *availableBits*:

```
if (nn_type == 0){
    availableBits = bitsPerFrame - bitsUsed - nChans × (nbits_isFeatAmplified +
nbits_featureScale + nbits_contextNumBytes)
```

```
} else if (nn_type == 1) {
      availableBits = bitsPerFrame - bitsUsed - nChans × (nbits_featureScaleLc +
nbits_contextNumBytes)
}
for (i = 0; i < nChans; i++){
      if (numGroups[i] == 1){
            availableBits -= nbits_nfParam
      }else if (numGroups[i] == 2){
            availableBits -= 2 × nbits_nfParam
      }
}
```

bitsPerFrame indicates the total number of coding bits of the current frame. bitsUsed indicates the number of used side information bits of the current frame. nChans indicates the number of channels (in the dual-channel stereo mode, the number of channels is 2). nbits_isFeatAmplified indicates the number of bits of the isFeatAmplified parameter (fixed to 1). nbits_featureScale indicates the number of bits of the scaleQIdx parameter (fixed to 7). nbits_featureScaleLc indicates the number of bits of the scaleQIdxLc parameter (fixed to 8). nbits_contextNumBytes indicates the number of bits of the contextNumBytes parameter (fixed to 8). numGroups indicates the number of groups of the MDCT spectra of the current frame. nbits_nfParam indicates the number of bits of the noise filling parameter (fixed to 3).

A bit allocation result of the downmixed channel is obtained based on *availableBits* and *bitsRatio*:

a) Byte is the minimum unit used in the current algorithm. *availableBits* is converted into available bytes *availableBytes*. Refer to formula (5).

$$availableBytes = fl(availableBits/8) \quad (5)$$

b) In the dual-channel stereo mode, the numbers of bytes of two downmixed channels *channelBytes*[0] and *channelBytes*[1] are respectively calculated according to formula (6) and formula (7).

$$channelBytes[0] = bitsRatio \times floor(availableBytes/(1<<3)) \quad (6)$$

$$channelBytes[1] = availableBytes - channelBytes[0] \quad (7)$$

(1<<3) indicates the number of groups of allocated bits in the dual-channel stereo mode. In other words, the available bits are divided into eight groups, in which several groups are allocated to one of the two downmixed channels, and other bits are allocated to the other downmixed channel.

7.5.3.5 M/S Upmixing

If isMs that indicates whether M/S dual-channel stereo processing is enabled for the current frame is 1, M/S upmixing should be performed on the channel obtained through decoding, to obtain the decoded MDCT spectra of the left and right channels.

For the M/S upmixing, refer to formula (8) and formula (9).

$$mdctSpectrum\_L = \frac{\sqrt{2}}{2}(mdctSpectrum\_M + mdctSpectrum\_S) \quad (8)$$

$$mdctSpectrum\_R = \frac{\sqrt{2}}{2}(mdctSpectrum\_M - mdctSpectrum\_S) \qquad (9)$$

mdctSpectrum_M: one of the decoded MDCT spectra of the two downmixed channels;

mdctSpectrum_S: the other of the decoded MDCT spectra of the two downmix channels;

mdctSpectrum_L: decoded MDCT spectrum of the left channel obtained through M/S upmixing;

mdctSpectrum_R: decoded MDCT spectrum of the right channel obtained through M/S upmixing.

After M/S upmixing, inverse ILD processing should be performed on the MDCT spectra of the left and right channels to recover the amplitude difference between the left and right channels. The inverse ILD processing is performed as follows:

The amplitude ratio of the left channel to the right channel is recovered based on the *IldQIdx* obtained by decoding the bitstream. Refer to formula (10).

$$levelRatio = (1 \ll 4)/IldQIdx - 1 \qquad (10)$$

(1 << 4) indicates the maximum value range of *IldQIdx*.

The amplitude adjustment of the left and right channels is shown in the following pseudocode:

```
if (levelRatio > 1.0) {
   mdctSpectrum_R = levelRatio * mdctSpectrum_R
}else {
   mdctSpectrum_L = (1.0 / levelRatio) * mdctSpectrum_L
}
```

7.6 Syntax, Semantics, and Decoding Processes of Multi-channel Bit Allocation and Upmixing

7.6.1 Syntax

The syntax of multi-channel side information should comply with provisions in Table 21.

Table 21 Syntax of multi-channel side information

Syntax of multi-channel side information	Number of bits	Mnemonic
DecodeMcSideBits() {	—	—
HasSilFlag	1	uimsbf
if(HasSilFlag==1) {	—	—
for(i = 0; i < coupleChNum; i++) {	—	—
silFlag[i]	1	uimsbf
}	—	—
} else {	—	—

Syntax of multi-channel side information	Number of bits	Mnemonic
for(i = 0; i < coupleChNum; i++) {	—	—
silFlag[i] = 0	—	—
}	—	—
}	—	—
pairCnt	4	uimsbf
for(i = 0; i < pairCnt; i++) {	—	—
channelPairIndex	—	uimsbf
mcIld[ch1]	5	uimsbf
mcIld[ch2]	5	uimsbf
}	—	—
for (i = 0; i < coupleChNum; i++) {	—	—
if(silFlag[i] == 0) {	—	—
chBitRatios[i]	6	uimsbf
}	—	—
}	—	—
}	—	—
Note: The number of bits of channelPairIndex is determined by the number of coupled channels coupleChNum. The calculation formula is: floor(log2(coupleChNum * (coupleChNum – 1)/2 – 1)) + 1. coupleChNum is the number of all channels excluding the LFE channel.		

7.6.2 Semantics

HasSilFlag

1 bit. It indicates the silence enabling flag. '0' indicates that all channels of the multi-channel signal in the current frame are non-silent channels. '1' indicates that at least one silent channel exists in the channels of the multi-channel signal in the current frame.

silFlag[i]

1 bit. It indicates the silence flag. '0' indicates that the i-th channel of the current frame is a non-silent channel. '1' indicates that the i-th channel of the current frame is a silent channel.

pairCnt

4 bits. It indicates the number of channel pairs in the current frame.

channelPairIndex

It indicates an index of a channel pair. It can be parsed to obtain index values ch1 and ch2 of two channels in the current channel pair.

mcIld

5 bits. It indicates a parameter quantization index of an inter-channel amplitude difference ILD of the first channel and the second channel in the current channel pair, and is used for inter-channel energy adjustment.

chBitRatios

6 bits. It indicates a bit allocation ratio of each channel.

7.6.3 Decoding Process

7.6.3.1 Overview

In the multi-channel mode, the steps of bitstream demultiplexing are as follows: Perform parsing to obtain HasSilFlag, silFlag and multi-channel side information to guide multi-channel bit allocation, and then perform multi-channel bit allocation. Perform, based on the multi-channel bit allocation result, inverse neural network transform, inverse quantification, and range decoding on encoding information of each channel transmitted in the bitstream, to obtain a downmixed multi-channel signal. Multi-channel decoding processing (that is, multi-channel upmixing) is performed on the downmixed multi-channel signal based on the multi-channel side information, to obtain an upmixed multi-channel signal. Post-decoding processing (that is, bandwidth extension decoding, inverse temporal noise shaping, inverse frequency-domain noise shaping, upmixing, and inverse time-frequency transform) is performed on the upmixed multi-channel signal to obtain a multi-channel signal.

The multi-channel side information includes pairCnt, channelPairIndex, mcIld, and chBitRatios of the downmixed channel determined based on the silent channel flag.

The multi-channel mode includes a bit allocation module and a multi-channel upmixing module.

The bit allocation module is mainly used to calculate respective numbers of bit allocation bytes channelBytes of all demultiplexed channels based on silFlag, chBitRatios, and the total number of remaining bits for quantization and range coding after other side information in the current frame is removed. Range decoding, inverse quantization, and inverse neural network transform are performed based on channelBytes on a coded bitstream of a downmixed channel of a plurality of channels, to obtain a decoded MDCT coefficient of the downmixed channel of the plurality of channels.

The multi-channel upmixing module is mainly used to perform multi-channel decoding processing (M/S upmixing) based on information such as pairCnt and channelPairIndex to obtain an upmixed multi-channel MDCT spectrum. The energy adjustment is performed on the upmixed multi-channel MDCT spectrum based on the mcILD information to obtain a decoded MDCT coefficient of each channel.

The multi-channel bit allocation and channel upmixing module mainly includes bit allocation and multi-channel upmixing.

7.6.3.2 Bit Allocation

Multi-channel bit allocation McBitsAllocation() is mainly used to perform bit allocation on the downmixed multi-channel signal based on silFlag and chBitRatios that are obtained by decoding the bitstream, to obtain the number of coded bits of the downmixed multi-channel signal. The remaining available bits after other side information is removed are allocated to downmixed channels in the plurality of channels. Then, subsequent range decoding, inverse quantization, and inverse neural network transform are performed.

availableBytes indicates the number of remaining available bytes after other side information is removed from the current frame.

In the multi-channel mode, a silent channel may exist. The silent channel does not need to participate in the bit allocation process in the multi-channel mode. A fixed number of bytes are allocated in advance. The number of bytes is 8. If there is a silent channel, the number of pre-allocated bytes of the silent channel is deducted from the number of available bytes availableBytes. After the deduction, the remaining bytes are allocated to channels other than the silent channel.

In the multi-channel mode, an LFE channel may exist. Generally, there is little valid spectrum information of the LFE channel, and it does not need to participate in the bit allocation process in the multi-channel mode, as long as a fixed number of bits are allocated in advance. The number of pre-allocated bits of the LFE channel is related to the coding bit rate. cpeRate indicates an average bit rate of a channel pair. It is the result of converting a total coding bit rate into a channel pair. If cpeRate is less than 64 kb/s, 10 bytes are allocated to the LFE channel. If cpeRate is less than 96 kb/s, 15 bytes are allocated to the LFE channel. If cpeRate is greater than or equal to 96 kb/s, 20 bytes are allocated to the LFE channel. If there is an LFE channel, the number of pre-allocated bytes of the LFE channel is deducted from the number of available bytes *availableBytes*. After the deduction, the remaining bytes are allocated to channels other than the LFE channel.

The process of allocating the number of available bytes *availableBytes* to other channels is as follows.

a) The number of safe bytes safeBits is pre-allocated to each channel. The number of safe bytes is 8. The number of safe bytes is deducted from the number of available bytes *availableBytes*. After the deduction, the remaining bytes *availableBytes* are allocated in the subsequent steps.

b) Bits are allocated to each channel based on *chBitRatios*, and the number of bytes of each channel may be represented by formula (11).

$$channelBytes[i] = availableBytes * chBitRatios[i] / (1 << 6) \quad \text{……(11)}$$

(1 << 6) indicates the maximum value range of channel bit allocation ratio *chBitRatios*.

c) If some remaining bytes are not allocated in step a), they are allocated to each channel again based on the ratio indicated by *chBitRatios*[*i*].

d) If there are still remaining bits after step b), the remaining bits are allocated to the channel with the most bytes allocated in step a).

e) If the number of bytes allocated to some channels exceeds the upper limit for a single channel, the excessive bytes are allocated to other channels.

7.6.3.3 Upmixing

M/S upmixing is performed on the two coupled channels ch1 and ch2 indicated by the channel pair index channelPairIndex to obtain upmixed MDCT spectra of the coupled channels. The upmixing manner is the same as the M/S upmixing manner in the dual-channel stereo mode.

After the M/S upmixing, inverse ILD processing is performed on the upmixed MDCT spectra of the channels based on mcILD, to obtain an upmixed decoded signal for which the channel amplitude difference is recovered. To recover the channel amplitude difference, pseudocode of the inverse ILD processing is shown as follows:

```
factor = mcIldCodebook[mcIld[i]]
mdctSpectrum[i] = factor * mdctSpectrum[i]
```

factor is an amplitude adjustment factor corresponding to an ILD parameter of the i-th channel. mcIldCodebook is a quantization code table of the ILD parameter, and should comply with Table B.26. mcIld[i] indicates a quantization index corresponding to the ILD parameter of the i-th channel. mdctSpectrum[i] indicates an MDCT coefficient vector of the i-th channel.

7.7 HOA Bit Allocation and Upmixing

7.7.1 Syntax

The syntax of HOA side information should comply with provisions in Table 22.

Table 22 Syntax of HOA side information

Syntax of HOA side information	Number of bits	Mnemonic
DecodeHoaSideBits() {	—	—
sceneType	4	uimsbf
spatialAnalysis	1	uimsbf
if(spatialAnalysis == 1) {	—	—
numVL	4	uimsbf
for(i = 0; i < numVL; i++) {	—	—
basisIdx[i]	12	uimsbf
}	—	—
}	—	—
for(groupIdx = 0; groupIdx < nTotalChanGroups; groupIdx++) {	—	—
pairIdx[groupIdx]	4	uimsbf
if(pairIdx[groupIdx] > 0) {	—	—
for(i = 0; i < pairIdx[groupIdx]; i++) {	—	—
chIdx[groupIdx][i]	—	uimsbf
dmxMode[groupIdx][i]	1	uimsbf
if(dmxMode[groupIdx][i]== 1) {	—	—
for(sfb = 0; sfb < N_SFB_HOA_LBR - 1; sfb++){	—	—
sfbMask[groupIdx][i][sfb]	1	uimsbf
}	—	—
}	—	—

Syntax of HOA side information	Number of bits	Mnemonic
}	—	—
for(i = 0; i < groupChans[groupIdx]; i++) {	—	—
groupILD[i]	5	uimsbf
}	—	—
}	—	—
groupBitsRatio[groupIdx]	4	uimsbf
for(i = 0; i < groupChans[groupIdx]; i++) {	—	—
bitsRatio[groupIdx][i]	4	uimsbf
}	—	—
}	—	—
}	—	—
Note: The number of bits of chIdx is a round-down value of $(\log_{10}(groupChans \times (groupChans - 1)/2 - 1)/\log_{10}2) + 1$.		

7.7.2 Semantics

sceneType

4 bits. It indicates the sound field type.

spatialAnalysis

1 bit. It indicates the spatial analysis type. '0' indicates that there is no HOA spatial decoding. '1' indicates that there is HOA spatial decoding.

numVL

4 bits. It indicates the number of virtual speaker signals.

basisIdx

12 bits. It indicates the virtual speaker index.

pairIdx

4 bits. It indicates the number of channel pairs.

chIdx

It indicates a channel pair index. The number of bits of chIdx is a round-down value of (log10(groupChans × (groupChans – 1)/2 – 1)/log102) + 1.

groupILD

5 bits. It indicates a quantization index of an ILD parameter in a group.

groupBitsRatio

4 bits. It indicates an inter-group bit allocation ratio parameter.

bitsRatio

4 bits. It indicates an intra-group bit allocation ratio parameter.

dmxMode

1 bit. It indicates a downmixing mode. When dmxMode is 0, it indicates full-band M/S downmixing. When dmxMode is 1, it indicates sub-band M/S downmixing.

sfbMask

1 bit. It indicates whether downmixing is performed on the current sub-band. When sfbMask is 0, it indicates that downmixing is not performed on the current sub-band. When sfbMask is 1, it indicates that downmixing is performed on the current sub-band.

nTotalChanGroups

Preset value. It indicates the number of transmission channel groups, and should comply with provisions in Table 23 to Table 25.

N_SFB_HOA_LBR

Preset value. It indicates the number of sub-bands and is defined as 22.

groupChans

Preset value. It indicates the number of channels in each group, and should comply with provisions in Table 23 to Table 25.

Definitions of values of FOA nTotalChanGroups and groupChans should comply with provisions in Table 23.

Table 23 Definitions of values of FOA nTotalChanGroups and groupChans

Bit rate **kb/s**	**Value of nTotalChanGroups**	**Value of groupChans**
96	1	4
128	1	4
192	1	4
256	1	4

Definitions of values of 2-order HOA nTotalChanGroups and groupChans should comply with provisions in Table 24.

Table 24 Definitions of values of 2-order HOA nTotalChanGroups and groupChans

Bit rate **kb/s**	**Value of nTotalChanGroups**	**Value of groupChans**
192	1	9

Bit rate kb/s	Value of nTotalChanGroups	Value of groupChans
256	1	9
320	1	9
384	1	9
480	1	9
512	1	9
640	1	9

Definitions of values of 3-order HOA nTotalChanGroups and groupChans should comply with provisions in Table 25.

Table 25 Definitions of values of 3-order HOA nTotalChanGroups and groupChans

Bit rate kb/s	Value of nTotalChanGroups	Value of groupChans
256	2	2, 6
320	2	2, 7
384	2	2, 9
512	2	2, 10
640	2	2, 12
896	1	16

7.7.3 Decoding Process

7.7.3.1 Overview

HOA bit allocation and channel upmixing are performed in the core decoder. During HOA decoding, the core decoder parses the core decoder side information from the bitstream, performs bit allocation HoaSplitBytesGroup() on the virtual speaker signal and the residual signal, and decodes the virtual speaker signal and residual signal based on the bitstream. Then, HOA decoding processing Avs3HoaInverseDMX() is performed on the virtual speaker signal and the residual signal. Finally, post-decoding processing is performed, including bandwidth extension for the virtual speaker signal and the residual signal, temporal noise shaping decoding, frequency-domain noise shaping decoding, and MDCT inverse transform, to obtain a time-domain transport channel signal. Sections 7.7.3.2 and 7.7.3.3 describe in detail the bit allocation HoaSplitBytesGroup() on the virtual speaker signal and the residual signal, and decoding processing Avs3HoaInverseDMX() on the virtual speaker signal and the residual signal.

7.7.3.2 Bit Allocation

The transport channel obtained through decoding by the core decoder consists of the virtual speaker signal and the residual signal, and is divided into *nTotalChanGroups* groups. In the decoding process, the following two parameters are obtained by parsing the bitstream: groupBitsRatio and bitsRatio. *groupBitsRatio*[*groupIdx*] occupies four bits, and indicates an inter-group bit allocation ratio parameter of the *groupIdx*-th group. The bit allocation ratio of the virtual speaker signal group and the bit allocation ratio of the residual signal group can be obtained through bit allocation. *bitsRatio*[*groupIdx*][*i*] occupies 4 bits, and indicates a bit allocation ratio parameter of the i-th channel in the *groupIdx*-th group. Therefore, a bit allocation ratio of each virtual speaker signal group in all virtual speaker signal groups and a bit allocation ratio of each residual signal group in all residual signal groups can be obtained.

Bit allocation is mainly used to allocate the remaining available bits after other side information is removed, to each transport channel based on the bit allocation ratio parameter obtained by decoding the bitstream. The coding of other side information also occupies bits.

First, the number of remaining available bits after other side information is removed from the current frame is calculated and denoted as availableBits. The general algorithm for calculating *availableBits* is expressed as *availableBits* = bitsPerFrame – bitsUsed. bitsPerFrame indicates the number of initial bits of each frame. bitsUsed indicates the number of occupied bits before bit allocation. Then, bit allocation is performed based on the number of available bits.

HOA bit allocation HoaSplitBytesGroup() is calculated as follows:

a) The number of bits *groupBytes*[*groupIdx*] of each group of channels is calculated based on the total number of available bits *availableBits* and *groupBitsRatio*[*groupIdx*]. Refer to formula (12).

$$groupBytes[groupIdx] = availableBits \times \frac{groupBitsRatio[groupIdx]}{\sum_{k=0}^{nTotalChanGroups-1} groupBitsRatio[k]} \quad \text{..........(12)}$$

$\frac{groupBitsRatio[groupIdx]}{\sum_{k=0}^{nTotalChanGroups-1} groupBitsRatio[k]}$ indicates a bit allocation ratio of a virtual speaker signal group in all transmission channel signals, or indicates a bit allocation ratio of a residual signal group in all transmission channel signals.

b) Then, the number of bits *bytesChannels*[*groupIdx*][*i*] of each channel is calculated based on *bitsRa o*[*groupIdx*][*i*]. Refer to formula (13).

$$bytesChannels[groupIdx][i] = groupBytes[groupIdx] \times \frac{bitsRatio[groupIdx][i]}{\sum_{k=0}^{groupChans[groupIdx]-1} bitsRatio[groupIdx][k]} \quad \text{....(13)}$$

$groupBytes[groupIdx]$ indicates the total number of allocated bits of the virtual speaker signal group or the total number of allocated bits of the residual signal group;

$\frac{bitsRatio[groupIdx][i]}{\sum_{k=0}^{groupChans[groupIdx]-1} bitsRatio[groupIdx][k]}$ indicates a bit allocation ratio of each virtual speaker signal in the virtual speaker signal group or a bit allocation ratio of each residual signal in the residual signal group;

$bytesChannels[groupIdx][i]$ indicates the number of bits of each virtual speaker signal or the number of bits of each residual signal.

In the HoaSplitBytesGroup() process, the numbers of bits of each group of virtual speaker signals and residual signals can be calculated, and the virtual speaker signals and the residual signals in the bitstream are decoded.

7.7.3.3 Upmixing

HOA decoding processing Avs3HoaInverseDMX() is performed by decoding the channel pair information based on *chIdx*. *chIdx* is a sequence number of an upper triangular matrix element (excluding the main diagonal element) formed by a channel pair. Therefore, *chIdx* can be used to determine the coupled channels. For example, when the number ch of channels is 4, for the *Matrix* matrix, refer to formula (14) and formula (15).

$$Matrix = \begin{bmatrix} (ch0ch1) & (ch0ch2) & (ch0ch3) & (ch0ch4) \\ & (ch1ch2) & (ch1ch3) & (ch1ch4) \\ & & (ch2ch3) & (ch2ch4) \\ & & & (ch3ch4) \end{bmatrix} \quad (14)$$

$$chIdx = \begin{bmatrix} 0 & 1 & 2 & 3 \\ & 4 & 5 & 6 \\ & & 7 & 8 \\ & & & 9 \end{bmatrix} \quad (15)$$

When *chIdx* = 0, a pair of *ch*0 and *ch*1 is obtained through parsing.

If dmxMode is full-band M/S downmixing, the value of sfbMask of each sub-band is set to 1. If dmxMode is sub-band M/S downmixing, the value of sub-band sfbMask is obtained by parsing the bitstream. If the value of sub-band sfbMask is 1, upmixing is performed to obtain upmixed channels *Y*1 and *Y*2. For upmixing, refer to formula (16) and formula (17).

$$Y1 = \frac{\sqrt{2}}{2}(ch0 + ch1) \quad (16)$$

$$Y2 = \frac{\sqrt{2}}{2}(ch0 - ch1) \quad (17)$$

Finally, the inverse ILD processing is performed on the upmixed channel. For the processing method, refer to section 7.6.3.3. An upmixed virtual speaker signal after the inverse ILD processing and an upmixed residual signal after the inverse ILD processing are obtained. Then, post-decoding processing Avs3PostSynthesis() is performed to obtain the virtual speaker signal and the residual signal. The virtual speaker signal and the residual signal are used as inputs to the HOA spatial decoder.

7.8 Syntax, Semantics, and Decoding Process of Post-decoding Processing

7.8.1 Syntax

The syntax of post-decoding processing should comply with provisions in Table 26.

Table 26 Syntax of post-decoding processing

Syntax of post-decoding processing	Number of bits	Mnemonic
Avs3PostSynthesis() {	—	—
if(bwePresent == 1) {	—	—

Syntax of post-decoding processing	Number of bits	Mnemonic
BweApplyDec()	—	—
}	—	—
TnsDec()	—	—
Avs3FdInvSpectrumShaping()	—	—
if(transformType == 0x1) {	—	—
MdctSpectrumDeinterleave()	—	—
}	—	—
Avs3InverseMdctDecoder()	—	—
}	—	—

7.8.2 Semantics

BweApplyDec()

Bandwidth extension decoding processing.

TnsDec()

Temporal noise shaping decoding processing.

Avs3FdInvSpectrumShaping()

Frequency-domain noise shaping decoding processing.

MdctSpectrumDeinterleave()

Short frame MDCT spectrum de-interleaving.

Avs3InverseMdctDecoder()

Inverse MDCT transform.

7.8.3 Decoding Process

Avs3PostSynthesis(): When bandwidth extension is enabled, bandwidth extension decoding processing is performed, and then temporal noise shaping decoding processing and frequency-domain noise shaping decoding processing are performed. When the windowed control parameter indicates a short window, MDCT spectrum de-interleaving is further performed. Finally, inverse MDCT transform is performed to obtain a time-domain reconstructed signal.

7.9 Syntax, Semantics, and Decoding Process of Bandwidth Extension Decoding

7.9.1 Syntax

The syntax of bandwidth extension decoding side information should comply with provisions in Table 27.

Table 27 Syntax of bandwidth extension decoding side information

Syntax of bandwidth extension decoding side information	Number of bits	Mnemonic
DecodeBweSideBits() {	—	—
for(i = 0; i < numSfb; i++) {	—	—
sfbEnvQIdx	7	uimsbf
}	—	—
for(i = 0; i < numTiles; i++) {	—	—
flag_whiten_ONOFF	1	uimsbf
if(flag_whiten_ONOFF == 0) {	—	—
whiteningLevel[i] = BWE_WHITENING_OFF	—	—
} else {	—	—
flag_whiten_MID_HIGH	1	uimsbf
if(flag_whiten_MID_HIGH == 0) {	—	—
whiteningLevel[i] = BWE_WHITENING_MID	—	—
} else {	—	—
whiteningLevel[i] = BWE_WHITENING_HIGH	—	—
}	—	—
}	—	—
}	—	—
}	—	—

7.9.2 Semantics

sfbEnvQIdx

7 bits. It indicates a quantization index of an envelope parameter of each SFB in a frequency band of bandwidth extension. The envelope parameter of each SFB in the frequency band of bandwidth extension can be obtained from this variable.

flag_whiten_ONOFF

1 bit. It indicates whether whitening is enabled for each frequency area in the frequency band of bandwidth extension. '0' indicates that whitening is disabled. '1' indicates that whitening is enabled.

flag_whiten_MID_HIGH

1 bit. It indicates the whitening level of each frequency area in the frequency band of bandwidth extension. '0' indicates that the whitening level is medium (MID) and '1' indicates that the whitening level is high (HIGH).

numSfb

It indicates the total number of SFBs in the frequency band of bandwidth extension. Refer to section 7.9.3.6 Bandwidth Extension Configuration Parameters.

numTiles

It indicates the total number of frequency areas in the frequency band of bandwidth extension.

whiteningLevel[i]

It indicates a whitening level parameter of the i-th frequency area in the frequency band of bandwidth extension.

7.9.3 Decoding Process

7.9.3.1 Overview

The bandwidth extension at the decoder is mainly used to recover the MDCT spectrum of the frequency band of bandwidth extension based on the core band MDCT spectrum and the bandwidth extension parameter obtained through decoding. The bandwidth extension algorithm at the decoder covers spectrum preparation for bandwidth extension, whitening, and envelope adjustment.

7.9.3.2 Spectrum Preparation for Bandwidth Extension

The core band MDCT spectrum obtained through decoding is mdctSpectrum. The MDCT spectrum of the frequency band of bandwidth extension is bweSpectrum. The start point of the source frequency area corresponding to each target frequency area in the frequency band of bandwidth extension is srcTiles[numTiles]. The boundary of each target frequency area in the spectrum of bandwidth extension is targetTiles[numTiles+1].

The basic process of spectrum preparation: Use srcTiles as the start point, replicate the core band MDCT spectrum obtained through decoding to the frequency area (the boundary of the frequency area is targetTiles) corresponding to the high frequency band of bandwidth extension.

The process of spectrum replication is shown in the following pseudocode:

```
for (tileIdx = 0; tileIdx < numTiles; tileIdx++){
    srcLineIdx = srcTiles[tileIdx]
    for (i = targetTiles[tileIdx]; i < targetTiles[tileIdx+1]; i++){
        bweSpectrum[i] = mdctSpectrum[srcLineIdx]
        srcLineIdx++
    }
}
```

7.9.3.3 Whitening

When spectrum features of the source and the target frequency areas are different, the replicated high-frequency spectrum component should be whitened to different extents, so that the spectrum features of the frequency band of bandwidth extension are more similar to those of the original high-frequency spectrum (for example, the spectrum component is closer to harmonic or noise features).

Spectrum whitening is classified into three levels: BWE_WHITENING_OFF, BWE_WHITENING_MID, and BWE_WHITENING_HIGH, respectively corresponding to the following spectrum processing methods.

- whitenedSpectrum indicates the whitened MDCT spectrum.

- BWE_WHITENING_OFF indicates whitening is not performed. That is, the replicated spectrum (bweSpectrum) is directly used as the whitened spectrum.

- BWE_WHITENING_MID: The whitening level is medium. In a basic manner of medium spectrum whitening, the moving average algorithm is used for processing on bweSpectrum to obtain the whitened MDCT spectrum.

For the moving average algorithm, refer to formula (18).

$$whitenedSpectrum[i] = \left(\frac{1}{2*AvgSize+1}\sum_{i-AvgSize}^{i+AvgSize}(bweSpectrum[i])^2\right)^{1/2} \quad \text{.........................(18)}$$

AvgSize indicates the size of the moving average processing neighborhood.

In the moving average processing on a frequency *i*, the average amplitude is calculated for the frequency in the range of [*i* – *AvgSize*, *i* + *AvgSize*].

- BWE_WHITENING_HIGH indicates that the whitening level is high. At this level, random noise of a certain amplitude is generated to replace bweSpectrum obtained through spectrum replication, so as to obtain a high-frequency spectrum component.

7.9.3.4 Envelope Adjustment

The envelope adjustment is used to adjust the amplitude of the whitened spectrum based on envelope information of each SFB in the high-frequency band obtained by decoding the bitstream, so that the energy of the high-frequency spectrum recovered by the bandwidth extension module keeps consistent with that of the original high-frequency spectrum.

For each SFB of the bandwidth extension spectrum, basic steps of the envelope adjustment algorithm are as follows:

a) Calculate the frequency band width *sfbWidth*. Refer to formula (19).

$$sfbWidth = sfbTable[sfbIdx+1] - sfbTable[sfbIdx] \quad \text{.........................(19)}$$

sfbTable indicates the frequency band division table of the bandwidth extension frequency band;

sfbIdx indicates the frequency band sequence number.

b) Calculate the energy *currEner* of the whitened spectrum of the current SFB. Refer to formula (20).

$$currEner = \frac{1}{sfbWidth}\sum_{i=sfbStart}^{sfbEnd-1}(whitenedSpectrum[i])^2 \quad \text{.........................(20)}$$

sfbStart indicates the start point of the current SFB;

sfbEnd indicates the end point of the current SFB.

$$sfbStart = sfbTable[sfbIdx] \quad \text{.........................(21)}$$

$$sfbEnd = sfbTable[sfbIdx + 1] \quad \text{……(22)}$$

c) Calculate the energy *targetEner* of the target spectrum of the current SFB based on the SFB envelope parameter transmitted in the bitstream. Refer to formula (23).

$$targetEner = 2.0^{\frac{sfbEnvQIdx[sfbIdx]}{4.24966} - 4.0} \quad \text{……(23)}$$

d) Calculate the spectrum gain of the current SFB based on the energy of the whitened and the target spectra. The pseudocode is as follows:

```
if (currEner != 0.0){
  gainSfb = sqrt(targetEner / currEner)
}else{
  gainSfb = 1.0
}
```

e) Multiply the spectrum gain of the current SFB by the whitened spectrum to obtain the high-frequency spectrum recovered according to the bandwidth extension algorithm.

7.9.3.5 Conditions for Enabling Bandwidth Extension

The conditions for enabling the bandwidth extension vary depending on mono, dual-channel stereo, multi-channel, and other signal forms.

Mono mode: enabled when the coding bit rate is less than or equal to 96 kb/s.

Dual-channel stereo mode: enabled when the coding bit rate is less than or equal to 128 kb/s.

Multi-channel mode: enabled when the equivalent dual-channel stereo coding bit rate is less than or equal to 128 kb/s. The equivalent dual-channel stereo coding bit rate is calculated by multiplying the average coding bit rate by 2. The average coding bit rate is the coding bit rate divided by the number of channels (excluding LFE channels).

FOA/HOA mode: enabled when the FOA coding rate is less than or equal to 256 kb/s, enabled when the 2-order HOA coding rate is less than or equal to 480 kb/s, and enabled when the 3-order HOA coding rate is less than or equal to 896 kb/s (disabled for the virtual speaker signal group, and enabled for the residual signal group when the 3-order HOA is less than or equal to 640 kb/s).

7.9.3.6 Bandwidth Extension Configuration Parameters

Bandwidth extension divides the high-frequency spectrum into several SFBs. One frequency area is formed by one or more SFBs. Each high-frequency area, or referred to as a target frequency area, has a corresponding low-frequency area, or referred to as a source frequency area, which is used to replicate the spectrum from the low-frequency band to the high-frequency band at the decoder.

The bandwidth extension configuration parameters are as follows:

a) The SFB division manner should comply with provisions in Tables 28 to 31. N_SFB indicates the number of SFBs. sfbStart n indicates the start point of the n-th SFB. sfbStart 1 indicates a frequency sequence number corresponding to the start frequency of the high-frequency spectrum.

b) The target frequency area division manner should comply with provisions in Tables 32 to 35. N_tT indicates the number of target frequency areas. targetTile n indicates the start point of the n-th target frequency area.

c) The source frequency area division manner should comply with provisions in Tables 36 to 39. srcTile n indicates the start point of the n-th source frequency area.

d) The manner in which the high frequency band constitutes the target frequency area should comply with provisions in Tables 40 to 43. tT_SFB n is the SFB sequence number corresponding to the start point of the n-th target frequency area.

Example:

Take the configuration of the mono bit rate of 32 kb/s or lower as an example.

In Table 28, the high-frequency spectrum is divided into six SFBs. The start point of the frequency range corresponding to the SFB 1 is 352 (that is, the 352nd MDCT frequency), and the end point is 415 (that is, 416 minus 1 in the table). The start point of the frequency range corresponding to the SFB 2 is 416, and the end point is 479 (that is, 480 minus 1 in the table). The remaining parameters are derived accordingly.

In Table 32, the high-frequency spectrum is divided into three target frequency areas. The start point of the frequency range corresponding to the TFA 1 is 352 (that is, the 352nd MDCT frequency), and the end point is 479 (that is, 480 minus 1 in the table). The start point of the frequency range corresponding to the TFA 2 is 480, and the end point is 607 (that is, 608 minus 1 in the table). The remaining parameters are derived accordingly.

In Table 36, the spectrum is also divided into three source frequency areas corresponding to the high-frequency areas. The start point of the SFA 1 is 64 (that is, the 64th MDCT frequency), and the start point of the SFA 2 is 96. The bandwidth of the source frequency area is the same as that of the corresponding target frequency area.

In Table 40, the correspondence between the target frequency area and the high frequency band is as follows: The TFA 1 includes the SFB 0 and the SFB 1, the TFA 2 includes the SFB 2 and the SFB 3, and the TFA 3 includes the SFB 4 and the SFB 5.

The configuration of the mono high-frequency SFB should comply with provisions in Table 28.

Table 28 Table of configuration of mono high-frequency SFB

Bit rate (kb/s)	**N_SFB**	**sfbStart 1**	**sfbStart 2**	**sfbStart 3**	**sfbStart 4**	**sfbStart 5**	**sfbStart 6**	**sfbStart 7**
<=32	6	352	416	480	544	608	672	768
(32, 56]	6	448	496	544	608	672	736	832
(56, 72]	4	544	608	672	736	832	—	—
(72, 96]	2	672	736	832	—	—	—	—

The configuration of the dual-channel stereo high-frequency SFB should comply with provisions in Table 29.

Table 29 Table of configuration of dual-channel stereo high-frequency SFB

Bit rate kb/s	N_SFB	sfbStart 1	sfbStart 2	sfbStart 3	sfbStart 4	sfbStart 5	sfbStart 6	sfbStart 7
<=48	6	352	416	480	544	608	672	768
(48, 64]	6	352	416	480	544	608	672	768
(64, 96]	4	544	608	672	736	832	—	—
(96, 128]	2	672	736	832	—	—	—	—

The configuration of the multi-channel high-frequency SFB should comply with provisions in Table 30.

Table 30 Configuration of multi-channel high-frequency SFB (bit rates are converted to channel pairs)

Bit rate kb/s	N_SFB	sfbStart 1	sfbStart 2	sfbStart 3	sfbStart 4	sfbStart 5	sfbStart 6	sfbStart 7
<=56	6	352	400	448	512	576	672	768
(56, 75]	5	400	448	512	576	672	768	—
(75, 108]	4	544	608	672	736	832	—	—
(108, 128]	2	672	736	832	—	—	—	—

The configuration of the HOA high-frequency SFB should comply with provisions in Table 31.

Table 31 Configuration of HOA high-frequency SFB (HOA2/HOA3 is a 2-order HOA/3-order HOA)

Bit rate kb/s	N_SFB	sfbStart 1	sfbStart 2	sfbStart 3	sfbStart 4	sfbStart 5	sfbStart 6	sfbStart 7
192(HOA2)	4	352	416	480	544	736	—	—
<=128(FOA) 256(HOA2) [256, 384](HOA3)	6	384	448	512	576	672	736	832
192(FOA) 320(HOA2) 512(HOA3)	4	544	608	672	736	832	—	—
256(FOA) [384, 480] (HOA2) [640, 896] (HOA3)	2	672	736	832	—	—	—	—

The configuration of the mono target frequency area should comply with provisions in Table 32.

Table 32 Configuration of mono target frequency area

Bit rate kb/s	**N_tT**	**targetTile 1**	**targetTile 2**	**targetTile 3**	**targetTile 4**
<=32	3	352	480	608	768
44, 56	3	448	544	672	832
64, 72	2	544	672	832	—
80, 96	1	672	832	—	—

The configuration of the dual-channel stereo target frequency area should comply with provisions in Table 33.

Table 33 Configuration of dual-channel stereo target frequency area

Bit rate kb/s	**N_tT**	**targetTile 1**	**targetTile 2**	**targetTile 3**	**targetTile 4**
<=48	3	352	480	608	768
(48, 64]	3	352	480	608	768
(64, 96]	2	544	672	832	—
(96, 128]	1	672	832	—	—

The configuration of the multi-channel target frequency area should comply with provisions in Table 34.

Table 34 Configuration of multi-channel target frequency area (bit rates are converted to channel pairs)

Bit rate kb/s	**N_tT**	**targetTile 1**	**targetTile 2**	**targetTile 3**	**targetTile 4**
<=56	3	352	448	576	768
(56, 75]	3	400	512	672	768
(75, 108]	2	544	672	832	—
(108, 128]	1	672	832	—	—

The configuration of the HOA target frequency area should comply with provisions in Table 35.

Table 35 Configuration of HOA target frequency area (HOA2/HOA3 is a 2-order HOA/3-order HOA)

Bit rate kb/s	N_tT	targetTile 1	targetTile 2	targetTile 3	targetTile 4
192(HOA2)	2	352	480	736	—
<=128(FOA) 256(HOA2) [256, 384](HOA3)	3	384	512	672	832
192(FOA) 320(HOA2) 512(HOA3)	2	544	672	832	—

Table 35 (continued)

Bit rate kb/s	N_tT	targetTile 1	targetTile 2	targetTile 3	targetTile 4
256(FOA) [384, 480] (HOA2) [640, 896] (HOA3)	1	672	832	—	—

The configuration of the mono source frequency area should comply with provisions in Table 36.

Table 36 Configuration of mono source frequency area

Bit rate kb/s	srcTile 1	srcTile 2	srcTile 3	srcTile 4
<=32	64	96	144	—
44, 56	96	144	192	—
64, 72	144	192	—	—
80, 96	192	—	—	—

The configuration of the dual-channel stereo source frequency area should comply with provisions in Table 37.

Table 37 Configuration of dual-channel stereo source frequency area

Bit rate kb/s	srcTile 1	srcTile 2	srcTile 3	srcTile 4
<=48	64	96	144	—

Bit rate kb/s	srcTile 1	srcTile 2	srcTile 3	srcTile 4
(48, 64]	64	96	144	—
(64, 96]	144	192	—	—
(96, 128]	192	—	—	—

The configuration of the multi-channel source frequency area should comply with provisions in Table 38.

Table 38 Configuration of multi-channel source frequency area (bit rates are converted to channel pairs)

Bit rate kb/s	srcTile 1	srcTile 2	srcTile 3	srcTile 4
<=56	64	96	144	—
(56, 75]	64	96	144	—
(75, 108]	144	192	—	—
(108, 128]	192	—	—	—

The configuration of the HOA source frequency area should comply with provisions in Table 39.

Table 39 Configuration of HOA source frequency area (HOA2/HOA3 is a 2-order HOA/3-order HOA)

Bit rate kb/s	srcTile 1	srcTile 2	srcTile 3	srcTile 4
192(HOA2)	64	96	—	—
<=128(FOA) 256(HOA2) [256, 384](HOA3)	96	144	192	—
192(FOA) 320(HOA2) 512(HOA3)	144	192	—	—
256(FOA) [384, 480] (HOA2) [640, 896] (HOA3)	192	—	—	—

The configuration of the SFB sequence number of the mono high frequency band constituting the target frequency area should comply with provisions in Table 40.

Table 40 Configuration of SFB sequence number of mono high frequency band constituting target frequency area

Bit rate **kb/s**	**tT_SFB 1**	**tT_SFB 2**	**tT_SFB 3**	**tT_SFB 4**
<=32	0	2	4	6
44, 56	0	2	4	6
64, 72	0	2	4	—
80, 96	0	2	—	—

The configuration of the SFB sequence number of the dual-channel stereo high frequency band constituting the target frequency area should comply with provisions in Table 41.

Table 41 Configuration of SFB sequence number of dual-channel stereo high frequency band constituting target frequency area

Bit rate **kb/s**	**tT_SFB 1**	**tT_SFB 2**	**tT_SFB 3**	**tT_SFB 4**
<=48	0	2	4	6
(48, 64]	0	2	4	6
(64, 96]	0	2	4	—
(96, 128]	0	2	—	—

The configuration of the SFB sequence number of the multi-channel high frequency band constituting the target frequency area should comply with provisions in Table 42.

Table 42 Configuration of SFB sequence number of multi-channel high frequency band constituting target frequency area (bit rates are converted to channel pairs)

Bit rate **kb/s**	**tT_SFB 1**	**tT_SFB 2**	**tT_SFB 3**	**tT_SFB 4**
<=56	0	2	4	6
(56, 75]	0	2	4	5
(75, 108]	0	2	4	—
(108, 128]	0	2	—	—

The configuration of the SFB sequence number of the HOA high frequency band constituting the target frequency area should comply with provisions in Table 43.

Table 43 Configuration of SFB sequence number of HOA high frequency band constituting target frequency area (HOA2/HOA3 is a 2-order HOA/3-order HOA)

Bit rate **kb/s**	**tT_SFB 1**	**tT_SFB 2**	**tT_SFB 3**	**tT_SFB 4**
192(HOA2)	0	2	4	—
<=128(FOA) 256(HOA2) [256, 384](HOA3)	0	2	4	6
192(FOA) 320(HOA2) 512(HOA3)	0	2	4	—
256(FOA) [384, 480] (HOA2) [640, 896] (HOA3)	0	2	—	—

7.10 Syntax, Semantics, and Decoding Process of Inverse Temporal Noise Shaping

7.10.1 Syntax

The syntax of temporal noise shaping side information should comply with provisions in Table 44.

Table 44 Syntax of temporal noise shaping side information

Syntax of temporal noise shaping side information	**Number of bits**	**Mnemonic**
DecodeTnsSideBits() {	—	—
for (i = 0; i < 2; i++) {	—	—
tnsEnable[i]	1	uimsbf
if(tnsEnable[i] == 1) {	—	—
tnsOrder[i]	3	uimsbf
tnsOrder[i] += 1	—	—
for (j = 0; j < tnsOrder[i]; j++) {	—	—
tnsHuffCode[i][j]	—	uimsbf
}	—	—
}	—	—

Syntax of temporal noise shaping side information	Number of bits	Mnemonic
}	—	—
}	—	—

7.10.2 Semantics

tnsEnable[i]

1 bit. It indicates whether the i-th TNS filter is enabled. '0' indicates disabled, and '1' indicates enabled.

tnsOrder[i]

3 bits. It indicates the order of the i-th TNS filter. The maximum order of the filter is 8.

tnsHuffCode[i][j]

It indicates the Huffman coding result of the j-order filter coefficient of the i-th TNS filter. The maximum value of the maximum order (that is, tnsOrder[i]) of the tns parameter is 8. Each order of the tns parameter has a corresponding Huffman code table, that is, the Huffman code tables (including codewords and the numbers of bits) of the tns parameters from order 0 to order 7. The numbers of bits should comply with provisions for the number of bits in Tables B.25 to B.32.

7.10.3 Decoding Process

7.10.3.1 Overview

The basic process of temporal noise shaping is as follows: First, range decoding and inverse quantization are performed on a Huffman coding result of a filter coefficient to obtain the filter coefficient. Then, the frequency band corresponding to each TNS filter is filtered to obtain the MDCT spectrum after inverse temporal noise shaping.

7.10.3.2 Decoding and Inverse Quantization of Filter Coefficient

The temporal noise shaping filter is indicated by a group of reflection coefficients. The maximum order of the filter is 8. The actual order of the filter for each frame is less than or equal to the maximum order. The Huffman coding is performed on the quantized reflection coefficients at the encoder. Each dimension of the filter coefficients has a corresponding Huffman code table, and should comply with provisions of Tables B.25 to B.32.

The Huffman decoding process of the filter coefficient quantization index is as follows: The corresponding Huffman code table is selected for the j-th filter coefficient of the i-th TNS filter of the current frame. The Huffman codeword tnsHuffCode[i][j] in the bitstream is decoded as the quantization index.

The quantized filter coefficient is obtained based on the scalar quantization code table of the filter coefficient and the quantization index obtained through decoding. The scalar quantization code table used here is a non-uniform scalar quantization code table, which should comply with provisions in Table B.33.

7.10.3.3 Inverse Temporal Noise Shaping Filtering

Temporal noise shaping uses a maximum of two groups of filters. The frequency ranges corresponding to the two groups of filters are respectively [660 Hz, 5400 Hz] and [5400 Hz, 20000 Hz].

If the enabling flag tnsEnable[i] of the i-th filter group is equal to 1, inverse temporal noise shaping filtering is performed on the frequency range corresponding to the i-th filter group.

The input of the filtering algorithm is the MDCT spectrum and the corresponding filter coefficient obtained through decoding. Frequency-domain filtering is performed on the MDCT spectrum according to the linear prediction filtering algorithm that is based on the reflection coefficient. The filtering process is shown in the following pseudocode:

```
mdct[k] -= parCoeff[tnsOrder - 1] * tnsState[tnsOrder - 1]
for (i = tnsOrder - 2; i >= 0; i--) {
    mdct[k] -= parCoeff[i] * tnsState[i]
    tnsState[i + 1] = parCoeff[i] * mdct[k] + tnsState[i]
}
tnsState[0] = mdct[k]
```

k is the frequency sequence number. mdct[k] indicates the MDCT spectral coefficient of the k-th frequency. parCoeff is the reflection coefficient. tnsOrder is the TNS filter order. tnsState is the TNS filter history.

7.11 Syntax, Semantics, and Decoding Process of Inverse Frequency-domain Noise Shaping

7.11.1 Syntax

The syntax of frequency-domain noise shaping side information should comply with provisions in Table 45.

Table 45 Syntax of frequency-domain noise shaping side information

Syntax of frequency-domain noise shaping side information	Number of bits	Mnemonic
DecodeFdShapingSideBits() {	—	—
if(lsfLbrFlag == 0) {	—	—
lsfVqIndex[0]	8	uimsbf
lsfVqIndex[1]	8	uimsbf
lsfVqIndex[2]	7	uimsbf
lsfVqIndex[3]	7	uimsbf
lsfVqIndex[4]	6	uimsbf
lsfVqIndex[5]	5	uimsbf
lsfVqIndex[6]	5	uimsbf

Syntax of frequency-domain noise shaping side information	Number of bits	Mnemonic
} else {	—	—
lsfVqIndex[0]	8	uimsbf
lsfVqIndex[1]	8	uimsbf
lsfVqIndex[2]	7	uimsbf
lsfVqIndex[3]	7	uimsbf
lsfVqIndex[4]	6	uimsbf
}	—	—
}	—	—

7.11.2 Semantics

lsfVqIndex[i]

It indicates the quantization index of the i-th sub-vector in the vector quantization of LSF parameters.

lsfLbrFlag

It indicates low-precision LSF quantization and coding. When the average bit rate of each channel is greater than 32 kb/s, lsfLbrFlag is 0, indicating that high-precision LSF quantization and coding are used. When the average bit rate of each channel is less than or equal to 32 kb/s, lsfLbrFlag is 1, indicating that low-precision LSF quantization and coding are used.

7.11.3 Decoding process

7.11.3.1 Overview

A spectral quantization noise shaping technique of an LPC-based spectral envelope is used to perform frequency-domain noise shaping. The coder calculates the LPC-based spectral envelope and shapes a to-be-coded MDCT spectrum. The decoder calculates the corresponding spectral envelope information based on the LPC parameter obtained by decoding the bitstream, and inverse shaping is performed.

Main steps of inverse frequency-domain noise shaping include LSF parameter inverse quantization, LSF parameter conversion, and inverse frequency-domain noise shaping.

7.11.3.2 LSF Parameter Inverse Quantization

Quantization and coding of an LSF parameter is performed by using the vector quantization technique. The structure of the vector quantization code table is multi-stage split vector quantization. For different coding bit rates, the high-precision and low-precision vector quantization code tables are used. When lsfLbrFlag is 0, the high-precision code table is used. When lsfLbrFlag is 1, the low-precision code table is used.

An LSF parameter has 16 dimensions.

The structure of the high-precision code table is as follows: In stage 1, the LSF vector is divided into two sub-vectors with dimensions of 9 and 7, and the numbers of bits in the code table of 8 and 8, and should comply with provisions in Tables B.34 to B.35. In stage 2, the residual vector quantized in stage 1 is divided into five sub-vectors with dimensions of 3, 3, 3, 3, and 4, and the numbers of bits in the code table of 7, 7, 6, 5, and 5, and should comply with provisions in Tables B.36 to B.40.

The structure of the low-precision code table is as follows: In stage 1, the LSF vector is divided into two sub-vectors with dimensions of 9 and 7, and the numbers of bits in the code table of 8 and 8, and should comply with provisions in Tables B.41 to B.42. In stage 2, the residual vector quantized in stage 1 is divided into three sub-vectors with dimensions of 5, 4, and 7, and the numbers of bits in the code table of 7, 7, and 6, and should comply with provisions in Tables B.43 to B.45.

The stage 1 vector and the stage 2 vector can be obtained through decoding based on the lsfVqIndex parameter obtained from the bitstream and the corresponding vector quantization code table. The stage 1 vector, the stage 2 vector, and the LSF parameter mean value vector are added together, to obtain the inverse quantized LSF parameter. The LSF parameter mean value vector should comply with provisions in Table B.46.

7.11.3.3 LSF Parameter Conversion

The LSF parameter is converted to an LSP parameter, and then the LSP parameter is converted to an LPC parameter. The LSF parameter conversion should comply with provisions in section 5.1.9 in 3GPP TS 26.445 (Release 17).

7.11.3.4 Inverse Frequency-domain Noise Shaping

The inverse frequency-domain noise shaping includes two steps: LPC spectral envelope calculation and inverse shaping.

The basic process of the LPC spectral envelope calculation is as follows:

a) Calculate the LPC spectral envelope of each frequency based on the LPC parameter. For the basic principle, refer to formula (24).

$$lpcEnv[k] = \frac{1}{\left|1-\sum_{i=1}^{p} a_i e^{\frac{2\pi jk}{N}i}\right|} \quad (24)$$

lpcEnv[*k*] indicates the LPC spectral envelope of the *k*-th frequency;

a_i indicates the *i*-th LPC coefficient.

1) The LPC coefficient is weighted to obtain the LPC weighted coefficient. The pseudocode is as follows:

```
GAMMA_LPC = 0.939999998
weightFactor[0] = 1.0
for i = 1 to lpcOrder:
    weightFactor[i] = weightFactor[i-1] * GAMMA_LPC
end
```

GAMMA_LPC is the initial value of the LPC weighting coefficient. weightFactor is the calculated LPC weighting coefficient. lpcOrder is the LPC order, and a value is 16.

The LPC weighting coefficient is multiplied by the LPC parameter to obtain a weighted LPC parameter weightedLpc.

2) A low-precision LPC spectral envelope is calculated: The weighted LPC parameter is pre-rotated. The complex FFT is performed on the pre-rotated LPC parameter rotateLpc, and the low-precision LPC spectral envelope is calculated. The pre-rotation of the LPC parameter is expressed as follows:

```
for (i = 0; i < lpcOrder + 1; i++) {
    tmp = i * PI / 512
    realPart[i] = weightedLpc[i] * cos(tmp)
    imagPart[i]= -weightedLpc[i] * sin(tmp)
}
```

realPart and imagPart are respectively a real part and an imaginary part of the pre-rotated LPC parameter *rotateLpc*.

The number of points of the complex FFT is 512. The low-precision LPC spectral envelope *rawLpcGain* is $1/\|FFT(rotateLpc)\|_2$. The number of points of the low-precision LPC spectral envelope obtained is 256.

3) An interpolated LPC spectral envelope is obtained. 4-fold linear interpolation is performed on *rawLpcGain*, to obtain an interpolated LPC spectral envelope of 1024 frequencies.

b) The sub-band LPC spectral envelope is calculated based on the frequency LPC spectral envelope: The MDCT spectrum is divided into several sub-bands with different lengths. The average value of frequency LPC spectral envelopes is calculated in each sub-band. The result is used as the sub-band LPC spectral envelope. The number of currently used sub-bands is 49, and the sub-band configuration should comply with provisions in Table B.47.

The method of inverse shaping is as follows: The calculated LPC spectral envelope is multiplied by the MDCT spectrum to obtain an MDCT spectrum obtained through inverse frequency-domain noise shaping.

7.12 Inverse MDCT Decoding Process

7.12.1 Overview

The inverse MDCT is performed on the windowed control parameter obtained by parsing the bitstream and the decoded MDCT coefficient of each channel, to obtain the reconstructed time-domain audio signal.

The inverse MDCT can be performed in different ways depending on different windowed control parameters:

- If the windowed control parameter is equal to 0x0 or 0x2, that is, the current frame is a long window or a cut-in window, the 2048-point inverse transform is performed on the MDCT spectrum to obtain a reconstructed time-domain signal of the current frame. The first half-frame of the reconstructed time-domain signal is added to the second half-frame of the reconstructed time-domain signal of the previous frame. The result is the output signal of the current frame.

- If the windowed control parameter is equal to 0x3, that is, the current frame is a cut-out window, the 2048-point inverse transform is performed on the MDCT spectrum to obtain a reconstructed time-domain signal of the current frame. The part that is in the first half-frame of the reconstructed time-domain signal and that is overlapped with the last short frame of the previous frame is added to the second half-frame of the last short frame of the previous frame. The result is the output signal of the current frame.

- If the windowed control parameter is equal to 0x1, that is, the current frame is a short window, the 256-point inverse transform is performed respectively on MDCT coefficients of eight short frames included in the de-interleaved MDCT spectrum. The obtained eight short-window time-domain signals are overlapped and added. The result is the output signal of the current frame.

7.12.2 Definition of Windowed

Forms of window types are as follows:

a) Long window: LONG_SHORT_TRANS_WINDOW, a sine window with a length of 2048 points;

b) Cut-in window: LONG_SHORT_TRANS_WINDOW, consisting of a left half sine window of 1024 points, 448 points of 1, a right half sine window of 128 points, and 448 points of 0;

c) Cut-out window: SHORT_LONG_TRANS_WINDOW, consisting of 448 points of 0, a left half sine window of 128 points, 448 points of 1, and a right half sine window of 1024 points;

d) Short window: ONLY_SHORT_WINDOW, a sine window with a length of 256 points. Eight short windows are added to each frame.

7.13 HOA Spatial Decoding

7.13.1 Syntax

The syntax of HOA spatial decoding should comply with provisions in Table 46.

Table 46 Syntax of HOA spatial decoding

Syntax of HOA spatial decoding	Number of bits	Mnemonic
HoaPostSynthesisFilter() {	—	—
if(spatialAnalysis == 1) {	—	—
HoaCoreDec()	—	—
}	—	—
}	—	—

7.13.2 Semantics

spatialAnalysis

It indicates the spatial analysis type. For the semantics of spatialAnalysis, refer to section 7.7.2.

HoaCoreDec()

It indicates sound field component synthesis and HOA synthesis decoding.

7.13.3 Decoding Process

7.13.3.1 Overview

During HOA spatial decoding, the virtual speaker signal and residual signal generated at the encoder and the target virtual speaker attribute information are reconstructed through post-decoding processing Avs3PostSynthesis(), to obtain a decoded HOA signal.

7.13.3.2 Principle of Sound Field Reconstruction by Virtual Speakers

The HOA coefficient of the virtual speaker is used to reconstruct the sound field. It is assumed that the sound field is p. For the sound field p(r, θ, φ, k) in the spherical coordinate system, refer to formula (25).

$$p(r,\theta,\varphi,k) = \sum_{m=0}^{\infty} j^m j_m^{kr}(kr) \sum_{0\le n\le m,\sigma=\pm1} B_{m,n}^{\sigma} Y_{m,n}^{\sigma}(\theta,\varphi) \quad (25)$$

r indicates the sphere radius;

θ indicates the azimuth;

φ indicates the pitch angle;

k indicates the wave velocity;

m indicates the FOA/HOA order number;

n indicates the direction parameter in each HOA order;

$j^m j_m^{kr}(kr)$ indicates the spherical Bessel function, also referred to as the radial basis function, where *j* indicates the imaginary unit;

$Y_{m,n}^{\sigma}(\theta,\varphi)$ indicates the spherical harmonic function corresponding to *θ* and *φ*;

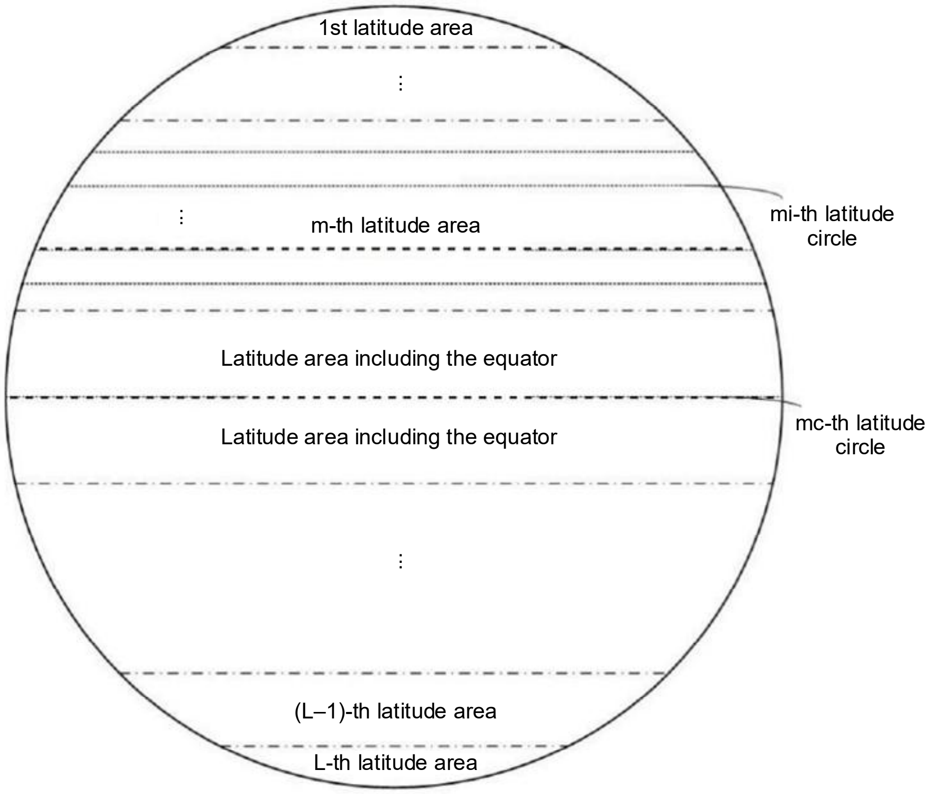
$B_{m,n}^{\sigma}$ indicates the HOA coefficient.

An *m*-order HOA coefficient is used as an approximate description of a sound field. $Y_{m,n}^{\sigma}(\theta,\varphi)$ is overlapped based on a coefficient corresponding to a sampling point of the HOA signal. Then, a spatial sound field corresponding to the sampling point can be reconstructed.

7.13.3.3 Virtual Speaker

Non-uniform virtual speaker distribution is used in HOA spatial encoding and decoding. The virtual speaker distribution includes position information of K virtual speakers. The position information includes a pitch angle index and a horizontal angle index. K is a positive integer greater than 1. K virtual speakers are distributed on a preset sphere. The preset sphere includes X latitude circles and Y longitude circles. The virtual speakers are located at intersection points of the X latitude circles and the Y longitude circles. The area division of virtual speakers should comply with Figure 11. The preset sphere includes L (L > 1) latitude areas, and the m-th latitude area includes T_m latitude circles. α_m indicates the horizontal angle difference between adjacent virtual speakers distributed on the m_i-th latitude circle in the K virtual speakers. $1 \le m \le L$. T_m is a positive integer. $1 \le m_i \le Tm$. If $T_m > 1$, α_m indicates the pitch angle difference between any two adjacent latitude circles in the m-th latitude area.

Figure 11 Division of virtual speaker area



K virtual speakers are distributed on one or more latitude circles in each latitude area. The distance between adjacent virtual speakers on a same latitude circle is indicated by a horizontal angle difference. The horizontal angle differences between all adjacent virtual speakers on the same latitude circle are equal. $\alpha_c < \alpha_m$. α_c indicates the horizontal angle difference between adjacent virtual speakers distributed on the m_c-th latitude circle in the K virtual speakers. The m_c-th latitude circle is any one of the latitude circles in the latitude area that includes the equator and that is in the L latitude areas. In the L latitude areas, the horizontal angle difference between adjacent virtual speakers in a latitude area that includes the equator is the smallest, and the distribution of virtual speakers in this area is the densest. The virtual speaker layout should comply with Figure 12.

An index indicates positions of K virtual speakers in the virtual speaker distribution table. The index can include a pitch angle index and a horizontal angle index. A horizontal angle of a virtual speaker distributed on any one of the latitude circles is set to 0. Then, the corresponding horizontal angle index is obtained through conversion according to a preset conversion formula between a horizontal angle and a horizontal angle index. Since the horizontal angle differences between any adjacent virtual speakers on the same latitude circle are equal, horizontal angles of other virtual speakers on this circle can be obtained. Therefore, the respective horizontal angle indexes of the other virtual speakers can also be obtained according to the formula (25). Similarly, respective pitch angle indexes of the virtual speakers may be obtained according to formula (24).

A pitch angle φ_k and a pitch angle index φ_k' of a k-th virtual speaker in the K virtual speakers meet the following formula (26).

$$\varphi_k' = round\left(\frac{\varphi_k}{2\pi r_{ak} \times 1024}\right) \quad (26)$$

r_{ak} indicates the radius of the longitude circle where the *k*-th virtual speaker is located;

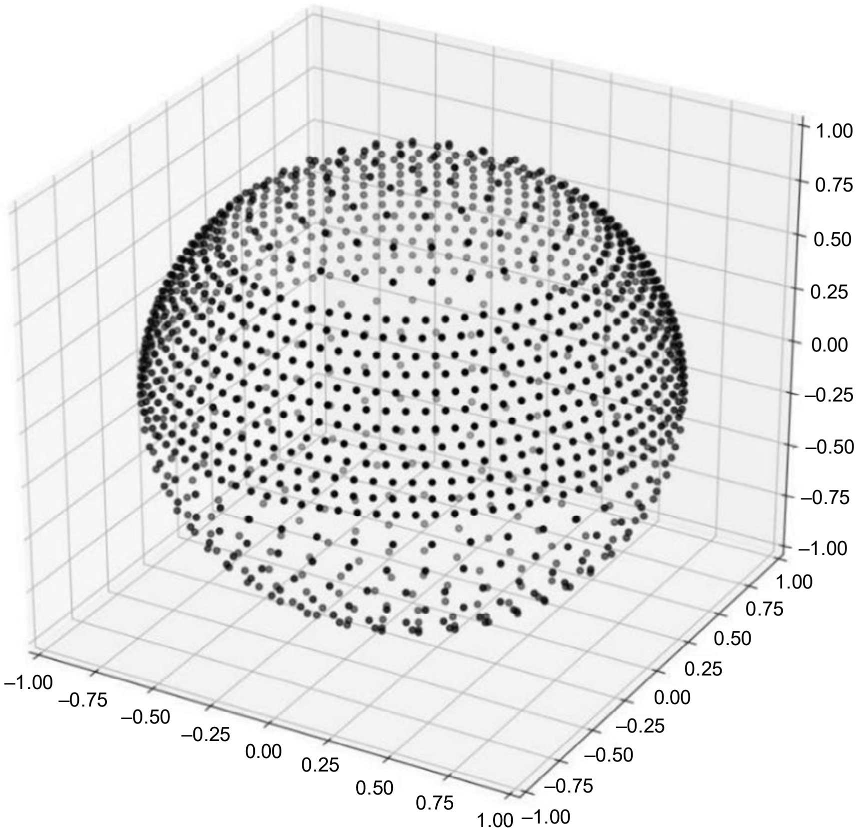
round() indicates rounding.

A horizontal angle θ_k and a horizontal angle index θ_k' of the *k*-th virtual speaker in the K virtual speakers meet the following formula (27).

$$\theta_k' = round\left(\frac{\theta_k}{2\pi r_{bk}\times 1024}\right) \qquad (27)$$

r_{bk} indicates the radius of the latitude circle where the *k*-th virtual speaker is located.

Figure 12 Distribution view of virtual speakers



7.13.3.4 HOA Coefficient Calculation of Virtual Speakers

First, the virtual speaker attribute information is parsed. The virtual speaker attribute information includes position information of the virtual speaker, and the position information includes pitch angle information and horizontal angle information. The position information of the virtual speaker may be indicated by a virtual speaker index, that is, the pitch angle information may be indicated by a pitch angle index, and the horizontal angle information may be indicated by a horizontal angle index. The HoaCoreDec() obtains the virtual speaker index basisIdx by parsing the bitstream, and searches the preset value of the virtual speaker for a virtual speaker angle index corresponding to the virtual speaker index basisIdx. The virtual speaker angle index includes a horizontal angle index θ' and a pitch angle index φ'. The preset values of virtual speakers should comply with provisions in Table B.48.

Then, HoaCoreDec() calculates the HOA coefficient of a virtual speaker based on the angle indexes of the virtual speaker. When the HOA coefficient is calculated, trigonometric function values corresponding to the horizontal angle index and the pitch angle index need to be obtained. A sine function value and a cosine function value corresponding to the pitch angle index of the virtual speaker are obtained based on the pitch trigonometric function table according to the following rules. Refer to formula (28) and formula (29).

$$sin(\varphi) = \begin{cases} sin\_table(\varphi') & \varphi' < \lfloor\frac{N}{4}\rfloor \\ sin\_table\left(\frac{N}{2} - \varphi'\right) & \lfloor\frac{N}{4}\rfloor \le \varphi' < \lfloor\frac{N}{2}\rfloor \\ -sin\_table(\varphi' - \frac{N}{2}) & \lfloor\frac{N}{2}\rfloor \le \varphi' < \lfloor\frac{3N}{4}\rfloor \\ -sin\_table(N - \varphi') & \lfloor\frac{3N}{4}\rfloor \le \varphi' < N \end{cases} \qquad (28)$$

$$cos(\varphi) = \begin{cases} sin\_table(\frac{N}{4} - \varphi') & \varphi' < \lfloor\frac{N}{4}\rfloor \\ -sin\_table(\varphi' - \frac{N}{4}) & \lfloor\frac{N}{4}\rfloor \le \varphi' < \lfloor\frac{N}{2}\rfloor \\ -sin\_table(\frac{3N}{4} - \varphi') & \lfloor\frac{N}{2}\rfloor \le \varphi' < \lfloor\frac{3N}{4}\rfloor \\ sin\_table(\varphi' - \frac{3N}{4}) & \lfloor\frac{3N}{4}\rfloor \le \varphi' < N \end{cases} \qquad (29)$$

The sine function value and the cosine function value corresponding to the horizontal angle index of the virtual speaker are obtained based on the horizontal trigonometric function table according to the following rules. Refer to formula (30) and formula (31).

$$sin(\theta) = \begin{cases} sin\_table(\theta') & \theta' < \lfloor\frac{N}{4}\rfloor \\ sin\_table\left(\frac{N}{2} - \theta'\right) & \lfloor\frac{N}{4}\rfloor \le \theta' < \lfloor\frac{N}{2}\rfloor \\ -sin\_table(\theta' - \frac{N}{2}) & \lfloor\frac{N}{2}\rfloor \le \theta' < \lfloor\frac{3N}{4}\rfloor \\ -sin\_table(N - \theta') & \lfloor\frac{3N}{4}\rfloor \le \theta' < N \end{cases} \qquad (30)$$

$$cos(\theta) = \begin{cases} sin\_table(\frac{N}{4} - \theta') & \theta' < \lfloor\frac{N}{4}\rfloor \\ -sin\_table(\theta' - \frac{N}{4}) & \lfloor\frac{N}{4}\rfloor \le \theta' < \lfloor\frac{N}{2}\rfloor \\ -sin\_table(\frac{3N}{4} - \theta') & \lfloor\frac{N}{2}\rfloor \le \theta' < \lfloor\frac{3N}{4}\rfloor \\ sin\_table(\theta' - \frac{3N}{4}) & \lfloor\frac{3N}{4}\rfloor \le \theta' < N \end{cases} \qquad (31)$$

N has a value of 1024.

The pitch trigonometric function table and the horizontal trigonometric function table should comply with provisions in Table B.49. The pitch trigonometric function table and the horizontal trigonometric function table are formed by trigonometric function values corresponding to position information of intersection points of N longitude circles and N latitude circles on the preset sphere. The pitch trigonometric function table includes trigonometric function values corresponding to the pitch angle indexes of the intersection points on the longitude circles. The horizontal trigonometric function table includes trigonometric function values corresponding to the horizontal angle indexes of the intersection points on the latitude circles.

The method for calculating an HOA coefficient of a virtual speaker based on the horizontal angle and pitch angle of the virtual speaker should comply with provisions in Table B.47. In the table, θ indicates horizontal angle information of position information of the virtual speaker on the preset sphere. φ indicates pitch angle information of position information of the virtual speaker on the preset sphere. m indicates an FOA/HOA order number. n indicates a direction parameter in each order.

The FOA/HOA coefficient expression should comply with provisions in Table 47.

Table 47 FOA/HOA coefficient expression

m	*n*	FOA/HOA coefficient expression
0	0	$\frac{1}{2\sqrt{\pi}}$
1	0	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\cos\theta$
	+1	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\sin\theta\cos\varphi$
	–1	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\sin\theta\sin\varphi$

Table 47 (continued)

m	*n*	FOA/HOA coefficient expression
2	0	$\frac{1}{4}\sqrt{\frac{5}{\pi}}(3cos^2\theta - 1)$
	+1	$\frac{1}{2}\sqrt{\frac{15}{\pi}}\sin\theta\cos\theta\cos\varphi$
	–1	$\frac{1}{2}\sqrt{\frac{15}{\pi}}\sin\theta\cos\theta\sin\varphi$
	+2	$\frac{1}{4}\sqrt{\frac{15}{\pi}}sin^2\theta\cos 2\varphi$
	–2	$\frac{1}{4}\sqrt{\frac{15}{\pi}}sin^2\theta\sin 2\varphi$
3	0	$\frac{1}{4}\sqrt{\frac{7}{\pi}}(5cos^3\theta - 3\cos\theta)$

m	n	FOA/HOA coefficient expression
	+1	$\frac{1}{4}\sqrt{\frac{21}{2\pi}}(5cos^2\theta - 1)\sin\theta\cos\varphi$
	–1	$\frac{1}{4}\sqrt{\frac{21}{2\pi}}(5cos^2\theta - 1)\sin\theta\sin\varphi$
	+2	$\frac{1}{4}\sqrt{\frac{105}{\pi}}\cos\theta\, sin^2\theta\cos 2\varphi$
	–2	$\frac{1}{4}\sqrt{\frac{105}{\pi}}\cos\theta\, sin^2\theta\sin 2\varphi$
	+3	$\frac{1}{4}\sqrt{\frac{35}{2\pi}} sin^3\theta\cos 3\varphi$
	–3	$\frac{1}{4}\sqrt{\frac{35}{2\pi}} sin^3\theta\sin 3\varphi$

7.13.3.5 Sound Field Component Synthesis and HOA Synthesis

During HOA spatial decoding, the virtual speaker signal and residual signal obtained through post-decoding processing and the calculated HOA coefficient of the virtual speaker are reconstructed, to obtain a decoded HOA signal.

HoaCoreDec() synthesizes the virtual speaker signal and the HOA coefficient of the virtual speaker into a sound field component to obtain a synthesized HOA signal *recoverySignal*. The *recoverySignal* is obtained by multiplying the HOA coefficient of the virtual speaker by the virtual speaker signal. Refer to formula (32).

$$recoverySignal = A \times w \qquad (32)$$

A indicates the virtual speaker HOA coefficient matrix with a size of (M x C);

w indicates the virtual speaker signal vector;

M indicates the number of m-order HOA coefficients;

C indicates the number of virtual speakers.

Finally, the *recoverySignal* is adjusted by using the residual signal, and the *recoverySignal* and the residual signal are added to obtain a decoded HOA signal.

8 Lossless Audio Decoding

Lossless audio decoding should comply with provisions in chapter 8 of GB/T 33475.3-2018.

9 Metadata Decoding

9.1 Syntax and Semantics of Metadata Bitstream

9.1.1 Syntax

The syntax of the metadata bitstream should comply with provisions in Table 48.

Table 48 Syntax of metadata bitstream

Syntax of metadata bitstream	Number of bits	Mnemonic
Avs3MetadataDec() {	—	—
smFlag	1	uimsbf
if(smFlag == 1) {	—	—
Avs3SmDec()	—	—
}	—	—
dmFlag	1	uimsbf
if(dmFlag == 1) {	—	—
Avs3DmDec()	—	—
}	—	—
}	—	—

9.1.2 Semantics

smFlag

1 bit. It indicates whether there is a dynamic metadata bitstream in the current frame. '0': no; '1': yes.

dmFlag

1 bit. It indicates whether there is a dynamic metadata bitstream in the current frame. '0': no; '1': yes.

Avs3SmDec()

Static metadata decoding.

Avs3DmDec()

Decoding of dynamic metadata.

9.2 Syntax and Semantics of Static Metadata Decoding

9.2.1 Syntax

The syntax of static metadata decoding should comply with provisions in Table 49.

Table 49 Syntax of static metadata decoding

Syntax of static metadata decoding	Number of bits	Mnemonic
Avs3SmDec() {	—	—
b_vrExt	1	uimsbf
basicLevel	3	uimsbf
if((basicLevel == 0) || (basicLevel == 1)) {	—	—
BasicL1()	—	—
}	—	—
}	—	—

9.2.2 Semantics

b_vrExt

1 bit. It indicates the status of extended static metadata of the current frame. '0': not existent; '1': reserved.

basicLevel

3 bits. It indicates the level of basic static metadata. '0': level 0 configuration in Table C.9; '1': level 1 configuration in Table C.9; '2' to '7': reserved.

BasicL1()

It includes basic static metadata decoding at the L0 and L1 levels.

9.3 Syntax and Semantics of Dynamic Metadata Decoding

9.3.1 Syntax

The syntax of dynamic metadata decoding should comply with provisions in Table 50.

Table 50 Syntax of dynamic metadata decoding

Syntax of dynamic metadata decoding	Number of bits	Mnemonic
Avs3DmDec () {	—	—
dmLevel	3	uimsbf
for(i = 0; i < numDmChans; i++) {	—	—
muteFlag	1	uimsbf

Table 50 (continued)

Syntax of dynamic metadata decoding	Number of bits	Mnemonic
transChRef	5	uimsbf
if(dmLevel == 0) {	—	—
Avs3DmL1Dec()	—	—
}	—	—
if(dmLevel == 1) {	—	—
Avs3DmL1Dec()	—	—
Avs3DmL2Dec()	—	—
}	—	—
}	—	—
}	—	—

9.3.2 Semantics

dmLevel

3 bits. It indicates the level of dynamic metadata. '0': Avs3DmL1Dec() corresponds to this level; '1': Avs3DmL1Dec() and Avs3DmL2Dec() correspond to this level.

muteFlag

1 bit. It indicates whether there is dynamic metadata. '0': yes; '1': no.

transChRef

5 bits. It indicates an index of a physical channel on which the dynamic metadata takes effect.

numDmChans

It indicates the number of object channels, and should comply with provisions for the semantics of object_channel_number in Table A.2.

Avs3DmL1Dec()

It indicates dynamic metadata decoding at the L1 level.

Avs3DmL2Dec()

It indicates dynamic metadata decoding at the L2 level.

9.4 Syntax and Semantics of Basic Static Metadata Decoding

9.4.1 Syntax

The syntax of basic static metadata decoding should comply with provisions in Table 51.

Table 51 Syntax of basic static metadata decoding

Syntax of basic static metadata decoding	Number of bits	Mnemonic
BasicL1() {	—	—
audioProgramme()	—	—
numOfContents	2	uimsbf
numOfContents = numOfContents + 1	—	—
for (i = 0; i < numOfContents; i++) {	—	—
audioContent()	—	—
}	—	—
numOfObjects	3	uimsbf
numOfObjects = numOfObjects + 1	—	—
for (i = 0; i < numOfObjects; i++) {	—	—
audioObject()	—	—
}	—	—
numOfPacks	3	uimsbf
numOfPacks = numOfPacks + 1	—	—
for (i = 0; i < numOfPacks; i++) {	—	—
AudioPackFormat ()	—	—
}	—	—
numOfChannels	5	uimsbf
numOfChannels = numOfChannels + 1	—	—
for (i = 0; i < numOfChannels; i++) {	—	—
AudioChannelFormat ()	—	—
}	—	—
}	—	—

Table 51 (continued)

9.4.2 Semantics

AudioProgramme()

It indicates AudioProgramme() decoding, and all AudioContent() items are combined to form complete mixing.

AudioContent()

It indicates AudioContent() decoding, and the audio content is described, including parameters such as language (if a dialogue exists) and loudness.

AudioObject()

It indicates AudioObject() decoding. The object is used to associate the audio content with the format.

AudioPackFormat()

It indicates AudioPackFormat() decoding, and packs one or more AudioChannelFormat() items that belong to each other (for example, a pair of stereos).

AudioChannelFormat()

It indicates AudioChannelFormat() decoding, and describes a single audio waveform.

numOfContents

2 bits. It indicates the number of audioContent() items included in BasicL1().

numOfObjects

3 bits. It indicates the number of audioObject() items included in BasicL1().

numOfPacks

3 bits. It indicates the number of audioPackFormat() items included in BasicL1().

numOfChannels

5 bits. It indicates the number of audioChannelFormat() items included in BasicL1().

9.5 Syntax and Semantics of Decoding Program Layer in Basic Static Metadata

9.5.1 Syntax

The syntax of decoding the program layer in the basic static metadata should comply with provisions in Table 52.

Table 52 Syntax of decoding program layer in basic static metadata

Syntax of decoding program layer in basic static metadata	Number of bits	Mnemonic
AudioProgramme() {	—	—
b_audioProgrammeLanguage	1	uimsbf
b_maxDuckingDepth	1	uimsbf
b_loudnessMetadata	1	uimsbf
b_audioProgrammeReferenceScreen	1	uimsbf
if (b_audioProgrammeLanguage) {	—	—

Syntax of decoding program layer in basic static metadata	Number of bits	Mnemonic
audioProgrammeLanguage	4	uimsbf
}	—	—
if (b_maxDuckingDepth) {	—	—
maxDuckingDepth	5	uimsbf
}	—	—
if (b_loudnessMetadata) {	—	—
LoudnessMetadata()	—	—
}	—	—
if (b_audioProgrammeReferenceScreen) {	—	—
AudioProgrammeReferenceScreen()	—	—
}	—	—
numContents	2	uimsbf
numContents = numContents + 1	—	—
for (i = 0; i < numContents; i++) {	—	—
refContentIdx	2	uimsbf
}	—	—
}	—	—

9.5.2 Semantics

b_audioProgrammeLanguage

1 bit. It indicates whether the audioProgrammeLanguage field exists in AudioProgramme(). '0': no; '1': yes.

b_maxDuckingDepth

1 bit. It indicates whether maxDuckingDepth exists in AudioProgramme(). '0': no; '1': yes.

b_loudnessMetadata

1 bit. It indicates whether LoudnessMetadata() exists in AudioProgramme(). '0': no; '1': yes.

b_audioProgrammeReferenceScreen

1 bit. It indicates whether AudioProgrammeReferenceScreen() exists in AudioProgramme(). '0': no; '1': yes.

audioProgrammeLanguage

4 bits. It indicates the dialogue language in AudioProgramme(). '0': Chinese; '1': English; '2': French; '3': Spanish; '4': Portuguese; '5': German; '6' to '15': reserved.

maxDuckingDepth

5 bits. It indicates the maximum ducking depth allowed for each AudioObject() in a program. The value range is [–62, 0].

numContents

2 bits. It indicates the number of refContentIdx items referenced in AudioProgramme().

refContentIdx

2 bits. It indicates an index of content included in AudioProgramme().

LoudnessMetadata()

It indicates a loudness decoding interface, which corrects audio according to the loudness algorithm in GY/T 262-2012.

AudioProgrammeReferenceScreen()

It indicates a screen size decoding interface, which standardizes the reference, production, and monitoring screen size of AudioProgramme().

9.6 Syntax and Semantics of Decoding Content Layer in Basic Static Metadata

9.6.1 Syntax

The syntax of decoding the content layer in the basic static metadata should comply with provisions in Table 53.

Table 53 Syntax of decoding content layer in basic static metadata

Syntax of decoding content layer in basic static metadata	Number of bits	Mnemonic
AudioContent () {	—	—
contentIdx	2	uimsbf
b_audioContentLanguage	1	uimsbf
b_loudnessMetadata	1	uimsbf
b_dialogue	1	uimsbf
b_numComplementaryObjectGroup	1	uimsbf
if (b_audioContentLanguage) {	—	—
audioContentLanguage	4	uimsbf
}	—	—
if (b_loudnessMetadata) {	—	—
loudnessMetadata()	—	—
}	—	—
if (b_dialogue) {	—	—

Syntax of decoding content layer in basic static metadata	Number of bits	Mnemonic
Dialogue()	—	—
}	—	—
if(b_numComplementaryObjectGroup) {	—	—
numComplementaryObjectGroup	2	uimsbf
numComplementaryObjectGroup=numComplementaryObjectGroup + 1	—	—
for (i=0; i < numComplementaryObjectGroup; i++) {	—	—

Table 53 (continued)

Syntax of decoding content layer in basic static metadata	Number of bits	Mnemonic
numComplementaryObject	3	uimsbf
numComplementaryObject= numComplementaryObject + 1	—	—
for (j=0; j < numComplementaryObject; j++) {	—	—
ComplementaryObjectIdx	3	uimsbf
}	—	—
}	—	—
}	—	—
numObjects	3	uimsbf
numObjects = numObjects + 1	—	—
for (i = 0; i < numObjects; i++) {	—	—
refObjectIdx	3	uimsbf
}	—	—
}	—	—

9.6.2 Semantics

contentIdx

2 bits. It indicates a unique index value of AudioContent().

b_audioContentLanguage

1 bit. It indicates whether the audioContentLanguage field exists in AudioContent(). '0': no; '1': yes.

b_loudnessMetadata

1 bit. It indicates whether LoudnessMetadata() exists in AudioContent(). '0': no; '1': yes.

b_dialogue

1 bit. It indicates whether the dialogue field exists in AudioContent(). '0': no; '1': yes.

b_numComplementaryObjectGroup

1 bit. It indicates whether the numComplementaryObjectGroup field exists in AudioContent(). '0': no; '1': yes.

audioContentLanguage

4 bits. It indicates the dialogue language in AudioContent(). '0': Chinese; '1': English; '2': French; '3': Spanish; '4': Portuguese; '5': German; '6' to '15': reserved.

numComplementaryObjectGroup

2 bits. It indicates the number of complementary object groups in AudioContent().

numComplementaryObject

2 bits. It indicates the number of complementary object indexes ComplementaryObjectIdx included in ComplementaryObjectGroup.

ComplementaryObjectIdx

3 bits. It indicates an index value of the complementary object in the specified ComplementaryObjectGroup.

numObjects

3 bits. It indicates the number of refObjectIdx items referenced in AudioContent().

refObjectIdx

3 bits. It indicates the index of the object included in AudioContent().

Dialogue()

It indicates a dialogue decoding interface, which details the content type.

LoudnessMetadata()

It indicates a loudness decoding interface, which corrects audio according to the loudness algorithm in GY/T 262-2012.

9.7 Syntax and Semantics of Decoding Object Layer in Basic Static Metadata

9.7.1 Syntax

The syntax of decoding the object layer in the basic static metadata should comply with provisions in Table 54.

Table 54 Syntax of decoding object layer in basic static metadata

Syntax of decoding object layer in basic static metadata	Number of bits	Mnemonic
AudioObject() {	—	—
objectIdx	3	uimsbf

Syntax of decoding object layer in basic static metadata	Number of bits	Mnemonic
b_audioObjectLanguage	1	uimsbf
b_dialogue	1	uimsbf
b_audioObjectImportance	1	uimsbf
b_disableDucking	1	uimsbf
b_interact	1	uimsbf
b_gain	1	uimsbf
b_headLocked	1	uimsbf
b_mute	1	uimsbf
if (b_audioObjectLanguage) {	—	—
audioObjectLanguage	4	uimsbf
}	—	—
if (b_dialogue) {	—	—
Dialogue()	—	—
}	—	—
if (b_audioObjectImportance) {	—	—
audioObjectImportance	4	uimsbf
}	—	—
if (b_interact) {	—	—
for (i = 0; i < 24; i++) {	—	—
ObjectName[i]	8	uimsbf
}	—	—
audioObjectInteraction()	—	—
}	—	—
if (b_gain) {	—	—
objectGainUnit	1	uimsbf
objectGainQFlag	1	uimsbf
objectGain	6	uimsbf
}	—	—

Table 54 (continued)

Syntax of decoding object layer in basic static metadata	Number of bits	Mnemonic
numPacks	3	uimsbf
numPacks = numPacks + 1	—	—
for (i = 0; i < numPacks; i++) {	—	—
refPackFormatIdx	3	uimsbf
}	—	—
}	—	—

9.7.2 Semantics

objectIdx

3 bits. It indicates a unique index value of AudioObject().

b_audioObjectLanguage

1 bit. It indicates whether the AudioObjectLanguage field exists in AudioObject(). '0': no; '1': yes.

b_dialogue

1 bit. It indicates whether the dialogue field exists in AudioObject(). '0': no; '1': yes.

b_audioObjectImportance

1 bit. It indicates whether the audioObjectImportance field exists in AudioObject(). '0': no; '1': yes.

b_disableDucking

1 bit. It indicates whether AudioObject() allows automatic ducking. '1': no; '0': yes.

b_interact

1 bit. It indicates whether the interact field exists in AudioObject(). '0': no; '1': yes.

b_gain

1 bit. It indicates whether the gain field exists in AudioObject(). '0': no; '1': yes.

b_headLocked

1 bit. It indicates whether the perception position of the audio element of AudioObject() is locked relative to the head. '0': yes; '1': no.

b_mute

1 bit. It indicates whether AudioObject() is in a playback state. '0': The object is being played back; '1': The object is muted.

audioObjectLanguage

4 bits. It indicates the dialogue language in AudioObject(). '0': Chinese; '1': English; '2': French; '3': Spanish; '4': Portuguese; '5': German; '6' to '15': reserved.

audioObjectImportance

4 bits. It indicates the importance level of AudioObject(). '10': highest importance; '0': lowest importance; '11' to '15': reserved.

objectName

8 bits. It indicates the name of AudioObject().

objectGainUnit

1 bit. It indicates the unit of objectGain. '0': linear; '1': dB.

objectGainQFlag

1 bit. It indicates the gain quantization interval. When gainUnit is 0 and gainQFlag is 0, the quantization interval is [0, 1]. When gainUnit is 0 and gainQFlag is 1, the quantization interval is (1, 16). When gainUnit is 1 and gainQFlag is 0, the quantization interval is [–80, 0]. When gainUnit is 1 and gainQFlag is 1, the quantization interval is (0, 24].

objectGain

6 bits. It indicates a gain value applied to all audio samples referenced by AudioObject(). The value range is linear [0, 16]/dB[–80, 24].

numPacks

3 bits. It indicates the number of refPackFormatIdx items referenced in AudioObject().

refPackFormatIdx

3 bits. It indicates an index of audioPackFormat included in AudioObject().

Dialogue()

It indicates a dialogue decoding interface, which details the content type.

9.8 Syntax and Semantics of Decoding Dialogue Field in Basic Static Metadata

9.8.1 Syntax

The syntax of decoding the dialogue field in the basic static metadata should comply with provisions in Table 55.

Table 55 Syntax of decoding dialogue field in basic static metadata

Syntax of decoding dialogue field in basic static metadata	Number of bits	Mnemonic
Dialogue() {	—	—
dialogueAttribute	2	uimsbf
dialogueType	3	uimsbf
}	—	—

9.8.2 Semantics

dialogueAttribute

2 bits. It indicates the dialogue content type, and should comply with provisions in Table 33 in ITU-R BS.2076-2.

dialogueType

3 bits. It indicates the content type contained in the dialogueAttribute, and should comply with provisions in Table 34 in ITU-R BS.2076-2.

9.9 Syntax and Semantics of Decoding Packaging Layer in Basic Static Metadata

9.9.1 Syntax

The syntax of decoding the packaging layer in the basic static metadata should comply with provisions in Table 56.

Table 56 Syntax of decoding packaging layer in basic static metadata

Syntax of decoding packaging layer in basic static metadata	Number of bits	Mnemonic
AudioPackFormat () {	—	—
packFormatIdx	3	uimsbf
b_audioPackFormatImportance	1	uimsbf
b_transChannelReuse	1	uimsbf
if (**b_audioPackFormatImportance**) {	—	—
audioPackFormatImportance	4	uimsbf
}	—	—

Table 56 (continued)

Syntax of decoding packaging layer in basic static metadata	Number of bits	Mnemonic
typeLabel	3	uimsbf
absoluteDistance	5	uimsbf
if (typeLabel == 1 || typeLabel == 2) {	—	—
packFormatID	6	uimsbf
if (typeLabel == 2) {	—	—
numMatrixOutputChannel	5	uimsbf
numMatrixOutputChannel = numMatrixOutputChannel + 1	—	—
for (i = 0; i< numMatrixOutputChannel; i++) {	—	—
DirectSpeakersPosition()	—	—
}	—	—

Syntax of decoding packaging layer in basic static metadata	Number of bits	Mnemonic
}	—	—
}	—	—
else if (typeLabel == 4) {	—	—
normalization	2	uimsbf
nfcRefDist	4	uimsbf
screenRef	1	uimsbf
hoaOrder	3	uimsbf
}	—	—
if (b_transChannelReuse == 0) {	—	—
packFormatStartIdx	5	uimsbf
}	—	—
numChannels	5	uimsbf
numChannels = numChannels + 1	—	—
for (i = 0; i < numChannels; i++) {	—	—
refChannelIdx	5	uimsbf
channelTypeLabel [refChannelIdx] = typeLabel	—	—
if (typeLabel == 1 || typeLabel == 2) {	—	—
channelPackFormatID[refChannelIdx] = packFormatID	—	—
}	—	—
	—	—
if (typeLabel == 2) {	—	—
channelNumMatrixOutputChannel[refChannelIdx] = numMatrixOutputChannel	—	—
}	—	—
if(b_transChannelReuse) {	—	—
transChRef	5	uimsbf
}	—	—
}	—	—
}	—	—

9.9.2 Semantics

packFormatIdx

3 bits. It indicates a unique index value of AudioPackFormat().

b_audioPackFormatImportance

1 bit. It indicates whether the audioPackFormatImportance field exists in AudioPackFormat(). '0': no; '1': yes.

b_transChannelReuse

1 bit. It indicates whether physical channels corresponding to all AudioChannelFormat() items in AudioPackFormat() are arranged consecutively. 0 indicates yes, and 1 indicates no. When the value of b_transChannelReuse is 0, the physical channel index corresponding to the first AudioChannelFormat() item in AudioPackFormat() is indicated by the packFormatStartIdx field. When the value of b_transChannelReuse is 1, the physical channel index corresponding to each AudioChannelFormat() item in the AudioPackFormat() is indicated by transChRef.

audioPackFormatImportance

4 bits. It indicates the importance level of audioObject. '10': highest importance; '0': lowest importance; '11' to '15': reserved.

typeLabel

4 bits. It indicates the description related to the channel type, and should comply with provisions in Table 57.

absoluteDistance

5 bits. It indicates the absolute distance. The value range is [0, 16].

packFormatID

6 bits. '0' to '31' should comply with provisions of the 5 least significant bits of AudioPackFormatID in Table 2 of ITU-R BS.2094-1. '32' to '63' are defined by users.

numMatrixOutputChannel

5 bits. It indicates the number of matrix output channels based on the matrix.

normalization

2 bits. It indicates a scene-based normalization manner. '0': SN3D; '1' to '3': reserved.

nfcRefDist

4 bits. It indicates the reference distance (in meters) that is normalized by absoluteDistance and that is used in the scene-based audio production process. The reference distance can be used for near-field compensation (NFC) audio rendering. The value range is [0, 1].

screenRef

1 bit. It indicates whether the scene-based program is related to the screen.

hoaOrder

3 bits. It indicates the scene-based maximum order.

packFormatStartIdx

5 bits. It indicates a physical channel index corresponding to AudioChannelFormat() included in AudioPackFormat() or a start value of a plurality of consecutive physical channel indexes corresponding to a plurality of AudioChannelFormat() items.

numChannels

5 bits. It indicates the number of refChannelIdx items referenced in AudioPackFormat().

refChannelIdx

5 bits. It indicates an index of AudioChannelFormat included in AudioPackFormat().

channelTypeLabel

It indicates a local variable that maps 32 TypeLabel values. It is obtained by parsing the AudioPackFormat() function and used in the AudioChannelFormat function.

channelPackFormatID

It indicates a local variable that maps 32 PackFormatID values. It is obtained by parsing the AudioPackFormat() function and used in the AudioChannelFormat function.

channelNumMatrixOutputChannel

It indicates a local variable that maps 32 channelNumMatrixOutputChannel values. It is obtained by parsing the AudioPackFormat() function and used in the AudioChannelFormat function.

transChRef

5 bits. It indicates a physical channel index corresponding to AudioChannelFormat included in AudioPackFormat(). The definition of typeLabel should comply with provisions in Table 57.

Table 57 Definition of typeLabel

Channel type	typeLabel	Description
DirectSpeakers	1	For channel-based audio, each channel feeds a speaker directly.
Matrix	2	For all other typeLabel items, signals are matrixed together, such as the middle, left, and right signals.
Objects	3	For object-based audio, channels represent audio objects (or parts of objects), including position information.
HOA	4	For scene-based audio, high-fidelity surround stereo system and HOA are used.
Binaural	5	For dual-channel audio, playback is performed over headphones.
User Custom	Other values	User custom types.

9.10 Syntax and Semantics of Decoding Channel Layer in Basic Static Metadata

9.10.1 Syntax

The syntax of decoding the channel layer in the basic static metadata should comply with provisions in Table 58.

Table 58 Syntax of decoding channel layer in basic static metadata

Syntax of decoding channel layer in basic static metadata	Number of bits	Mnemonic
AudioChannelFormat () {	—	—
channelFormatIdx	5	uimsbf
b_channelGain	1	uimsbf
if(b_channelGain) {	—	—
channelGainUnit	1	uimsbf
channelGain_QFlag	1	uimsbf
channelGain	6	uimsbf
}	—	—
if(channelTypeLabel [channelFormatIdx]== 1) {	—	—
if(channelPackFormatID[channelFormatIdx]== 0x3f) {	—	—
DirectSpeakersPosition()	—	—
}	—	—
} else if(channelTypeLabel [channelFormatIdx]== 2) {	—	—
for (i = 0; i< channelNumMatrixOutputChannel[channelFormatIdx]; i++) {	—	—
matrixCoef[i]	8	uimsbf
}	—	—
}	—	—
}	—	—

9.10.2 Semantics

channelFormatIdx

5 bits. It indicates a unique index value of AudioChannelFormat().

b_channelGain

1 bit. It indicates whether the channelGain field exists in AudioChannelFormat(). '0': no; '1': yes.

channelGainUnit

1 bit. It indicates the unit of channelGain. '0': linear; '1': dB.

channelGain_QFlag

1 bit. It indicates the channelGain quantization interval. When gainUnit is 0 and channelGainQFlag is 0, the quantization interval is [0, 1]. When gainUnit is 0 and channelGainQFlag is 1, the quantization interval is (1, 16]. When gainUnit is 1 and channelGainQFlag is 0, the quantization interval is [–80, 0]. When gainUnit is 1 and channelGainQFlag is 1, the quantization interval is (0, 24].

channelGain

6 bits. It indicates a gain value applied to all audio samples referenced by AudioChannelFormat().

matrixCoef

8 bits. It indicates the matrix coefficient based on the matrix type. The value range is [0.1, 10].

9.11 Syntax and Semantics of Decoding Object Interaction Field in Basic Static Metadata

9.11.1 Syntax

The syntax of decoding the object interaction field in the basic static metadata should comply with provisions in Table 59.

Table 59 Syntax of decoding object interaction field in basic static metadata

Syntax of decoding object interaction field in basic static metadata	Number of bits	Mnemonic
AudioObjectInteraction() {	—	—
onOffInteract	1	uimsbf
gainInteract	1	uimsbf
positionInteract	1	uimsbf
if(gainInteract) {	—	—
gainInteractionUnit	1	uimsbf
gainInteractionRange_min	7	uimsbf
gainInteractionRange_max	7	uimsbf
}	—	—
if(positionInteract) {	—	—
cartesianInteraction	1	uimsbf
if (cartesianInteraction == 1) {	—	—
positionInteractionRange_Xmin	8	uimsbf
positionInteractionRange_Xmax	8	uimsbf

Syntax of decoding object interaction field in basic static metadata	Number of bits	Mnemonic
positionInteractionRange_Ymin	6	uimsbf
positionInteractionRange_Ymax	6	uimsbf
positionInteractionRange_Zmin	4	uimsbf
positionInteractionRange_Zmax	4	uimsbf
} else {	—	—

Table 59 (continued)

Syntax of decoding object interaction field in basic static metadata	Number of bits	Mnemonic
positionInteractionRange_azimuthMin	8	uimsbf
positionInteractionRange_azimuthMax	8	uimsbf
positionInteractionRange_elevationMin	6	uimsbf
positionInteractionRange_elevationMax	6	uimsbf
positionInteractionRange_distanceMin	4	uimsbf
positionInteractionRange_distanceMax	4	uimsbf
}	—	—
}	—	—
}	—	—

9.11.2 Semantics

onOffInteract

1 bit. If a user can switch the object on or off, this item is set to 1. Otherwise, this item is set to 0.

gainInteract

1 bit. If a user can change the object gain, this item is set to 1. Otherwise, this item is set to 0.

positionInteract

1 bit. If a user can change the object position, this item is set to 1. Otherwise, this item is set to 0.

gainInteractionUnit

1 bit. It indicates the unit of gainInteractionRange. '0': linear; '1': dB.

gainInteractionRange_min

7 bits. It indicates the minimum linear gain factor or logarithmic gain offset of a possible user gain interaction. The value range is [0, 1].

gainInteractionRange_max

7 bits. It indicates the maximum linear gain factor or logarithmic gain offset of a possible user gain interaction. The value range is [1, 16].

cartesianInteraction

1 bit. It indicates whether the Cartesian coordinate system is used.

positionInteractionRange_Xmin

8 bits. It indicates the minimum X-axis normalized offset value of a possible user position interaction. The value range is [–1, 1].

positionInteractionRange_Xmax

8 bits. It indicates the maximum X-axis normalized offset value of a possible user position interaction. The value range is [–1, 1].

positionInteractionRange_Ymin

6 bits. It indicates the minimum Y-axis normalized offset value of a possible user position interaction. The value range is [–1, 1].

positionInteractionRange_Ymax

6 bits. It indicates the maximum Y-axis normalized offset value of a possible user position interaction. The value range is [–1, 1].

positionInteractionRange_Zmin

4 bits. It indicates the minimum Z-axis normalized offset value of a possible user position interaction. The value range is [–1, 1].

positionInteractionRange_Zmax

4 bits. It indicates the maximum Z-axis normalized offset value of a possible user position interaction. The value range is [–1, 1].

positionInteractionRange_azimuthMin

8 bits. It indicates the minimum azimuth offset value of a possible user position interaction. The value range is [–180, 0].

positionInteractionRange_azimuthMax

8 bits. It indicates the maximum azimuth offset value of a possible user position interaction. The value range is [0, 180].

positionInteractionRange_elevationMin

6 bits. It indicates the minimum height offset value of a possible user position interaction. The value range is [–90, 0].

positionInteractionRange_elevationMax

6 bits. It indicates the maximum height offset value of a possible user position interaction. The value range is [0, 90].

positionInteractionRange_distanceMin

4 bits. It indicates the minimum normalized distance of a possible user position interaction. The value range is [0, 1].

positionInteractionRange_distanceMax

4 bits. It indicates the maximum normalized distance of a possible user position interaction. The value range is [0, 1].

9.12 Syntax and Semantics of Decoding Speaker Position in Basic Static Metadata

9.12.1 Syntax

The syntax of decoding the speaker position in the basic static metadata should comply with provisions in Table 60.

Table 60 Syntax of decoding speaker position in basic static metadata

Syntax of decoding speaker position in basic static metadata	Number of bits	Mnemonic
DirectSpeakersPosition() {	—	—
azimuth	8	uimsbf
elevation	6	uimsbf
distance	4	uimsbf
DirectSpeakerScreenEdgeLock	2	uimsbf
}	—	—

9.12.2 Semantics

azimuth

8 bits. It indicates the exact azimuth position of a speaker. The value range is [–180, 180].

elevation

6 bits. It indicates the exact height position of a speaker. The value range is [–90, 90].

distance

4 bits. It indicates the exact normalized distance of a speaker from the start point. The value range is [0, 1].

DirectSpeakerscreenEdgeLock

2 bits. It defines a speaker position at a screen edge. It includes four values: left, right, top, and bottom. '0': left; '1': right; '2': top; '3': bottom.

9.13 Syntax and Semantics of Decoding Loudness Field in Basic Static Metadata

9.13.1 Syntax

The syntax of decoding the loudness field in the basic static metadata should comply with provisions in Table 61.

Table 61 Syntax of decoding loudness field in basic static metadata

Syntax of decoding loudness field in basic static metadata	Number of bits	Mnemonic
LoudnessMetadata() {	—	—
b_integratedLoudness	1	uimsbf
b_loudnessRange	1	uimsbf
b_maxTruePeak	1	uimsbf
b_maxMomentary	1	uimsbf
b_maxShortTerm	1	uimsbf
b_dialogueLoudness	1	uimsbf
if (b_integratedLoudness) {	—	—
integratedLoudness	5	uimsbf
}	—	—
if (b_loudnessRange) {	—	—
loudnessRange	5	uimsbf
}	—	—
if (b_maxTruePeak) {	—	—
maxTruePeak	5	uimsbf
}	—	—
if (b_maxMomentary) {	—	—
maxMomentary	5	uimsbf
}	—	—
if (b_maxShortTerm) {	—	—
maxShortTerm	5	uimsbf
}	—	—
if (b_dialogueLoudness) {	—	—
dialogueLoudness	5	uimsbf
}	—	—

Syntax of decoding loudness field in basic static metadata	Number of bits	Mnemonic
}	—	—

9.13.2 Semantics

b_integratedLoudness

1 bit. It indicates whether the integratedLoudness field exists in loudness. '0': no; '1': yes.

b_loudnessRange

1 bit. It indicates whether the loudnessRange field exists in loudness. '0': no; '1': yes.

b_maxTruePeak

1 bit. It indicates whether the maxTruePeak field exists in loudness. '0': no; '1': yes.

b_maxMomentary

1 bit. It indicates whether the maxMomentary field exists in loudness. '0': no; '1': yes.

b_maxShortTerm

1 bit. It indicates whether the maxShortTerm field exists in loudness. '0': no; '1': yes.

b_dialogueLoudness

1 bit. It indicates whether the dialogueLoudness field exists in loudness. '0': no; '1': yes.

integratedLoudness

5 bits. It indicates the integrated loudness value. The value range is [–70, 0].

loudnessRange

5 bits. It indicates the loudness range. The value range is [10, 70].

maxTruePeak

5 bits. It indicates the maximum true peak. The value range is [–70, 0].

maxMomentary

5 bits. It indicates the maximum momentary loudness. The value range is [–70, 0].

maxShortTerm

5 bits. It indicates the maximum short-term loudness. The value range is [–70, 0].

dialogueLoudness

5 bits. It indicates the loudness of the average dialogue. The value range is [–70, 0].

9.14 Syntax and Semantics of Decoding Program Reference Screen Field in Basic Static Metadata

9.14.1 Syntax

The syntax of decoding the program reference screen field in the basic static metadata should comply with provisions in Table 62.

Table 62 Syntax of decoding program reference screen field in basic static metadata

Syntax of decoding program reference screen field in basic static metadata	Number of bits	Mnemonic
AudioProgrammeReferenceScreen() {	—	—
cartesianReferenceScreen	1	uimsbf
aspectRatio	3	uimsbf
if (cartesianReferenceScreen == 0) {	—	—
screenCentrePosition_azimuth	8	uimsbf
screenCentrePosition_elevation	6	uimsbf
screenCentrePosition_distance	4	uimsbf
screenWidth_polar	7	uimsbf
} else {	—	—
screenCentrePosition_X	8	uimsbf
screenCentrePosition_Y	6	uimsbf
screenCentrePosition_Z	4	uimsbf
screenWidth_cartesian	7	uimsbf
}	—	—
}	—	—

9.14.2 Semantics

cartesianReferenceScreen

1 bit. It indicates whether the Cartesian coordinate system is used.

aspectRatio

3 bits. It indicates the mapping table of the aspect ratio of the screen. '0': 16:9; '1': 21:9; '2' to '7': reserved.

screenCentrePosition_azimuth

8 bits. It indicates the azimuth angle of the screen center. The value range is [–180, 180].

screenCentrePosition_elevation

6 bits. It indicates the height of the screen center. The value range is [0, 90].

screenCentrePosition_distance

4 bits. It indicates the normalized distance to the screen center. The value range is [0, 1].

screenWidth_polar

7 bits. It indicates the screen width in polar coordinates. The value range is [0, 180].

screenCentrePosition_X

8 bits. It indicates the normalized X coordinate of the screen center. The value range is [–1, 1].

screenCentrePosition_Y

6 bits. It indicates the normalized Y coordinate of the screen center. The value range is [–1, 1].

screenCentrePosition_Z

4 bits. It indicates the normalized Z coordinate of the screen center. The value range is [–1, 1].

screenWidth_cartesian

7 bits. It indicates the normalized screen width in the Cartesian coordinate system. The maximum value of screenWidth_cartesian should comply with the maximum value of screenWidth in the Cartesian coordinate system in ITU-R BS.2076-2. The value range is [0, 1].

9.15 Syntax and Semantics of Dynamic Metadata Level1 Decoding

9.15.1 Syntax

The syntax of dynamic metadata Level1 decoding should comply with provisions in Table 63.

Table 63 Syntax of dynamic metadata Level1 decoding

Syntax of dynamic metadata Level1 decoding	Number of bits	Mnemonic
Avs3DmL1Dec() {	—	—
if(muteFlag == 0) {	—	—
cartesianDm	1	uimsbf
b_obj_extent	1	uimsbf
b_obj_gain	1	uimsbf
b_obj_diffuse	1	uimsbf
b_obj_importance	1	uimsbf
if(cartesianDm == 0) {	—	—
obj_position_azimuth	8	uimsbf
obj_position_elevation	6	uimsbf
obj_position_distance	4	uimsbf
if(b_obj_extent) {	—	—
obj_width_horizontal	7	uimsbf

Syntax of dynamic metadata Level1 decoding	Number of bits	Mnemonic
obj_hight_vertical	5	uimsbf
obj_depth_distance	4	uimsbf
}	—	—
}	—	—

Table 63 (continued)

Syntax of dynamic metadata Level1 decoding	Number of bits	Mnemonic
else {	—	—
obj_position_x	8	uimsbf
obj_position_y	6	uimsbf
obj_position_z	4	uimsbf
if(b_obj_extent) {	—	—
obj_width_x	7	uimsbf
obj_hight_y	5	uimsbf
obj_depth_z	4	uimsbf
}	—	—
}	—	—
if(b_obj_gain) {	—	—
gain	7	uimsbf
}	—	—
if(b_obj_diffuse) {	—	—
diffuse	7	uimsbf
}	—	—
jumpPosition	1	uimsbf
if(b_obj_importance) {	—	—
importance	4	uimsbf
}	—	—
}	—	—
}	—	—

9.15.2 Semantics

cartesianDm

1 bit. It indicates the used coordinate system type. '0': polar coordinate system; '1': Cartesian coordinate system.

b_obj_extent

1 bit. It indicates whether the obj_width, obj_hight, and obj_depth fields exist in Avs3DmL1Dec(). '0': no; '1': yes.

b_obj_gain

1 bit. It indicates whether the gain field exists in Avs3DmL1Dec(). '0': no; '1': yes.

b_obj_diffuse

1 bit. It indicates whether the diffuse field exists in Avs3DmL1Dec(). '0': no; '1': yes.

b_obj_importance

1 bit. It indicates whether the importance field exists in Avs3DmL1Dec(). '0': no; '1': yes.

obj_position_azimuth

8 bits. It indicates the horizontal angle at the position of the object when the polar coordinate system is used. The value range is [–180, 180].

obj_position_elevation

6 bits. It indicates the pitch angle at the position of the object when the polar coordinate system is used. The value range is [–90, 90].

obj_position_distance

4 bits. It indicates the normalized distance at the position of the object when the polar coordinate system is used. The value range is [0, 1].

obj_position_x

8 bits. It indicates the normalized X coordinate of the left/right dimension of the object when the Cartesian coordinate system is used. The value range is [–1, 1].

obj_position_y

6 bits. It indicates the normalized Y coordinate of the front/back dimension of the object when the Cartesian coordinate system is used. The value range is [–1, 1].

obj_position_z

4 bits. It indicates the normalized Z coordinate of the upper/lower dimension of the object when the Cartesian coordinate system is used. The value range is [–1, 1].

obj_width_horizontal

7 bits. It indicates the width of the object sound source when the polar coordinate system is used. The value range is [0, 360].

obj_height_vertical

5 bits. It indicates the height of the object sound source when the polar coordinate system is used. The value range is [0, 360].

obj_depth_distance

4 bits. It indicates the normalized depth of the object sound source when the polar coordinate system is used. The value range is [0, 1].

obj_width_x

7 bits. It indicates the normalized width of the object sound source when the Cartesian coordinate system is used. The value range is [0, 1].

obj_height_y

5 bits. It indicates the normalized height of the object sound source when the Cartesian coordinate system is used. The value range is [0, 1].

obj_depth_z

4 bits. It indicates the normalized depth of the object sound source when the Cartesian coordinate system is used. The value range is [0, 1].

gain

7 bits. It indicates the object rendering gain information, which is linear. The value range is [0, 6].

diffuse

7 bits. It indicates the diffuse reflection of the object. The value range is [0, 1].

jumpPosition

1 bit. It indicates whether the object position in the current frame changes. '0': Interpolation is performed in the entire current frame; '1': The position changes.

importance

4 bits. It indicates the importance level of the object. '10': highest importance; '0': lowest importance; '11' to '15': reserved.

9.16 Syntax and Semantics of Dynamic Metadata Level2 Decoding

9.16.1 Syntax

The syntax of dynamic metadata Level2 decoding should comply with provisions in Table 64.

Table 64 Syntax of dynamic metadata Level2 decoding

Syntax of dynamic metadata Level2 decoding	Number of bits	Mnemonic
Avs3DmL2Dec() {	—	—
if(muteFlag == 0) {	—	—
hasChannelLock	**1**	**uimsbf**
if(hasChannelLock == 1) {	—	—
channelLock	**1**	**uimsbf**
if(channelLock == 1) {	—	—

Table 64 (continued)

Syntax of dynamic metadata Level2 decoding	Number of bits	Mnemonic
channelLock_maxDistance	**4**	**uimsbf**
}	—	—
}	—	—
hasObjectDivergence	**1**	**uimsbf**
if(hasObjectDivergence == 1) {	—	—
objectDivergence	**4**	**uimsbf**
if(objectDivergence != 0)	—	—
objectDivergence_azimuthRange	**6**	**uimsbf**
}	—	—
}	—	—
hasObjectScreenRef	**1**	**uimsbf**
if(hasObjectScreenRef == 1) {	—	—
obj_screenRef	**1**	**uimsbf**
}	—	—
hasScreenEdgeLock	**1**	**uimsbf**
if(hasScreenEdgeLock == 1) {	—	—
screenEdgeLock	**2**	**uimsbf**
}	—	—
}	—	—
}	—	—

9.16.2 Semantics

hasChannelLock

1 bit. It indicates whether channelLock metadata exists. '0': no; '1': yes.

channelLock

1 bit. It indicates whether the channel is locked. '0': no; '1': yes.

channelLock_maxDistance

4 bits. It indicates the maximum distance in channel locking. The value range is [0, 2].

hasObjectDivergence

1 bit. It indicates whether objectDivergence metadata exists. '0': no; '1': yes.

objectDivergence

4 bits. It indicates the object divergence. The value range is [0, 1].

objectDivergence_azimuthRange

6 bits. It indicates the relative position of the virtual object indicated by the object divergence. The value range is [0, 180].

hasObjectScreenRef

1 bit. It indicates whether screen-related metadata exists. '0': no; '1': yes.

obj_screenRef

1 bit. It indicates whether the object is related to the screen. '0': no; '1': yes.

hasScreenEdgeLock

1 bit. It indicates whether screenEdgeLock metadata exists. '0': no; '1': yes.

screenEdgeLock

2 bits. It indicates screen edge lock. '0': left; '1': right; '2': top; '3': bottom.

9.17 Decoding Process

During metadata decoding, a quantization step and a quantization offset of the metadata are set based on a value range of the metadata, to determine the quantization precision. The decoder parses a bitstream to obtain a quantization index of the metadata. The dequantized value *Qvalue* of the metadata is calculated based on the quantization index *QIdx*, the quantization step *QStep*, and the quantization offset *QSet*. In Table 65, the metadata is dequantized according to formula (33).

$$Qvalue = QIdx \times \mathrm{QStep} + \mathrm{Qset} \quad (33)$$

The metadata quantization parameter should comply with provisions in Table 65.

Table 65 Metadata quantization parameters

Metadata element	**Metadata name**	**Quantization step**	**Quantization Bias**
AudioProgramme()	maxDuckingDepth	2.000000	0
AudioPackFormat ()	nfcRefDist	0.066666	0
LoudnessMetadata()	integratedLoudness	2.258065	0
	loudnessRange	1.935483	10.0
	maxTruePeak	2.258065	0
	maxMomentary	2.258065	0
	maxShortTerm	2.258065	0
	dialogueLoudness	2.258065	0
AudioProgrammeReference	screenCentrePosition_azimuth	1.411764	–180.0

Metadata element	Metadata name	Quantization step	Quantization Bias
Screen()	screenCentrePosition_elevation	1.428571	0
	screenCentrePosition_distance	0.066666	0
	screenWidth_polar	1.417322	0
	screenCentrePosition_X	0.007843	–1.0
	screenCentrePosition_Y	0.031746	–1.0
	screenCentrePosition_Z	0.133333	–1.0
	screenWidth_cartesian	0.007874	0
AudioObjectInteraction()	gainInteractionRange_min (gainInteractionUnit is 0)	0.007874	0
	gainInteractionRange_min (gainInteractionUnit is 1)	0.629921	0
	gainInteractionRange_max (gainInteractionUnit is 0)	0.118110	1.0
	gainInteractionRange_max (gainInteractionUnit is 1)	0.188976	0
	positionInteractionRange_Xmin	0.007843	–1.0
	positionInteractionRange_Xmax	0.007843	–1.0

Table 65 (continued)

Metadata element	Metadata name	Quantization step	Quantization Bias
AudioObjectInteraction()	positionInteractionRange_Ymin	0.031746	–1.0
	positionInteractionRange_Ymax	0.031746	–1.0
	positionInteractionRange_Zmin	0.133333	–1.0
	positionInteractionRange_Zmax	0.133333	–1.0
	positionInteractionRange_azimuthMin	0.705882	0
	positionInteractionRange_azimuthMax	0.705882	0
	positionInteractionRange_elevationMin	1.428571	0
	positionInteractionRange_elevationMax	1.428571	0
	positionInteractionRange_distanceMin	0.066667	0
	positionInteractionRange_distanceMax	0.066667	0

Metadata element	Metadata name	Quantization step	Quantization Bias
DirectSpeakersPosition()	azimuth	1.411764	0
	elevation	2.857140	0
	distance	0.066667	0
AudioObject()	objectGain (objectGainUnit is 0, objectGainQFlag is 0)	0.015873	0
	objectGain (objectGainUnit is 0, objectGainQFlag is 1)	0.238095	1.0
	objectGain (objectGainUnit is 1, objectGainQFlag is 0)	1.269841	0
	objectGain (objectGainUnit is 1, objectGainQFlag is 1)	0.380952	0
AudioChannelFormat()	channelGain (channelGainUnit is 0, channelGain_QFlag is 0)	0.015873	0
	channelGain (channelGainUnit is 0, channelGain_QFlag is 1)	0.238095	1.0
	channelGain (channelGainUnit is 1, channelGain_QFlag is 0)	1.269841	0
	channelGain (channelGainUnit is 1, channelGain_QFlag is 1)	0.380952	0
	MatrixCoef	0.038823	0.1
AcousticEnv()	earlyReflectionGain	0.007874	0
	lateReverbGain	0.007874	0
Surface()	absorption	0.007874	0
	scattering	0.007874	0
Vertex()	x	1.574803	–100.0
	y	1.574803	–100.0
	z	1.574803	–100.0
AudioEffect()	attackTime	6.600000	1.0
	releaseTime	16.666660	50.0
	threshold	0.708661	–80.0

Metadata element	Metadata name	Quantization step	Quantization Bias
	preGain	0.157480	–10.0

Table 65 (continued)

Metadata element	Metadata name	Quantization step	Quantization Bias
AudioEffect()	postGain	0.157480	0
	ratio	0.779527	1.0
	effectGain	0.314960	–20.0
eqEffect()	eqQ(eqQQFlag is 0)	0.014285	0.1
	eqQ(eqQQFlag is 1)	0.174603	1.0
	eqGain	0.314960	–20.0
Avs3DmL1Dec()	obj_position_azimuth	1.411764	–180
	obj_position_elevation	2.857142	–90
	obj_position_distance	0.066666	0
	obj_width_horizontal	2.834645	0
	obj_hight_vertical	11.612903	0
	obj_depth_distance	0.066666	0
	obj_position_x	0.007843	–1.0
	obj_position_y	0.031746	–1.0
	obj_position_z	0.133333	–1.0
	obj_width_x	0.007874	0
Avs3DmL1Dec()	obj_hight_y	0.032258	0
	obj_depth_z	0.066666	0
	gain	0.047244	0
	diffuse	0.007874	0
Avs3DmL2Dec()	channelLock_maxDistance	0.133333	0
	objectDivergence	0.066666	0
	objectDivergence_azimuthRange	2.857142	0

The *absoluteDistance* metadata is dequantized according to formula (34).

$$absoluteDistance = 10^{(QIdx \times 0.03969)} - 1.0 \quad (34)$$

The *eqFc* metadata is dequantized according to formula (35).

$$eqFc = 10^{((QIdx \times 0.457179 + 20 \times log10(20))/20)} \quad (35)$$

Annex A
(Normative)
Syntax and Semantics of 3D Audio Coded Bitstream

A.1 Syntax

A.1.1 Overview

According to the AASF and AATF specifications in GB/T 33475.3-2018, the syntax and semantics of the AASF and AATF are provided for the format of the 3D audio bitstream. The file extension name of the 3D audio bitstream is av3a.

AASF includes the header information of an audio sequence and the following raw data block. AASF is only applicable to systems that define the starting point without having to start decoding from the intermediate part of the audio datastream. AASF includes all the necessary information for decoding and playing audio data, which is an exchange memory format.

As a streaming format, AATF includes the synchronization word and the necessary information for decoding. The synchronization word allows the decoder to decode without a determined starting point.

A.1.2 Syntax of AASF

AASF includes the header information of an audio sequence and the following raw data block. The syntax should comply with provisions in Table A.1.

Table A.1 Syntax of aasf_sequence()

Syntax	Number of bits	Mnemonic
aasf_sequence()	—	—
{	—	—
aasf_header()	—	—
if (audio_codec_id < 2) {	—	—
if (coding_profile == 0) {	—	—
if (audio_codec_id == 1)	—	—
ll_raw_data_stream()	—	—
}	—	—
if (coding_profile == 2)	—	—
hoa_raw_data_stream()	—	—
}	—	—
else if (audio_codec_id == 2)	—	—
ga_co_raw_data_stream()	—	—
}	—	—

The AASF header describes the header information of audio storage format, including identification and a size of a data storage format of the AASF header. The syntax of aasf_header() should comply with provisions in Table A.2.

Table A.2 Syntax of aasf_header()

Syntax	Number of bits	Mnemonic
aasf_header()	—	—
{	—	—
aasf_id	32	bslbf
header_size	24	bslbf
raw_stream_length	32	bslbf
audio_codec_id	4	bslbf
resolution	2	bslbf
if(audio_codec_id==2) {	—	—
nn_type	3	uimsbf
}	—	—
coding_profile	3	bslbf
anc_data_index	1	bslbf
if(audio_codec_id==1)	—	—
channel_number	{4;8}	bslbf
if(audio_codec_id==2){	—	—
if(coding_profile ==0){	—	—
channel_number_index	7	bslbf
}	—	—
if(coding_profile ==1){	—	—
soundBedType	2	uimsbf
if (soundBedType == 0){	—	—
object_channel_number	7	uimsbf
bitrate_index_per_channel	4	uimsbf
} else if (soundBedType == 1){	—	—
channel_number_index	7	uimsbf
bitrate_index	4	uimsbf
object_channel_number	7	uimsbf
bitrate_index_per_channel	4	uimsbf

Syntax	Number of bits	Mnemonic
}	—	—
}	—	—
if(coding_profile ==2){	—	—
order	4	uimsbf
}	—	—
}	—	—
sampling_frequency_index	4	bslbf
if(audio_codec_id==1){	—	—
if(sampling_frequency_index==0xf) {	—	—
sampling_frequency	24	uimsbf

Table A.2 (continued)

Syntax	Number of bits	Mnemonic
}	—	—
}	—	—
if(audio_codec_id==2 && coding_profile != 1){	—	—
bitrate_index	4	bslbf
}	—	—
byte_alignment()	—	—
}	—	—

A.1.3 AATF Syntax

AATF includes a synchronization word and necessary information for decoding. Its syntax should comply with provisions in Table A.3.

Table A.3 Syntax of aatf_sequence()

Syntax	Number of bits	Mnemonic
aatf_sequence()	—	—
{	—	—
while (nextbits(n) == syncword) {	—	—
aatf_frame()	—	—

Syntax	Number of bits	Mnemonic
}	—	—
}	—	—

An AATF frame describes a bitstream sequence of an audio transmission frame. The sequence includes corresponding decoded header information, information header, error check, and raw data block. The syntax of aatf_frame() should comply with provisions in Table A.4.

Table A.4 Syntax of aatf_frame()

Syntax	Number of bits	Mnemonic
aatf_frame()	—	—
{	—	—
syncword	12	bslbf
audio_codec_id	4	bslbf
anc_data_index	1	bslbf
if(audio_codec_id < 3){	—	—
aatf_frame_header()	—	—
if (audio_codec_id == 2){	—	—
frame_error_check()	—	—
}	—	—
byte_alignment()	—	—
if (audio_codec_id < 2) {	—	—
if (coding_profile == 0) {	—	—

Table A.4 (continued)

Syntax	Number of bits	Mnemonic
if (audio_codec_id == 1)	—	—
ll_raw_data_block()	—	—
}	—	—
if (coding_profile == 2)	—	—
hoa_raw_data_block()	—	—
}	—	—
}	—	—

Syntax	Number of bits	Mnemonic
else if (audio_codec_id == 2)	—	—
ga_co_raw_data_block()	—	—
}	—	—
if (audio_codec_id == 1){	—	—
frame_error_check()	—	—
}	—	—
}	—	—

frame_error_check() is located at the end of an attf frame and is used to check integrity of a bitstream of each frame. All bits of a current frame enter a CRC algorithm according to a sequence in which the bits appear. The syntax of frame_error_check() should comply with provisions in Table A.5.

Table A.5 Syntax of frame_error_check()

Syntax	Number of bits	Mnemonic
frame_error_check()	—	—
{	—	—
crc_check	8	rpchof
}	—	—

An AATF decoded header describes decoded header information, and includes a synchronization word, a sample rate index, and the like. The syntax of aatf_frame_header() should comply with provisions in Table A.6.

Table A.6 Syntax of aatf_frame_header()

Syntax	Number of bits	Mnemonic
aatf_frame_header()	—	—
{	—	—
if(audio_codec_id==2) {	—	—
nn_type	3	uimsbf
}	—	—
coding_profile	3	uimsbf
sampling_frequency_index	4	uimsbf
if(audio_codec_id==1) {	—	—

Syntax	Number of bits	Mnemonic
if (sampling_frequency_index==0xf) {	—	—

Table A.6 (continued)

Syntax	Number of bits	Mnemonic
sampling_frequency	24	uimsbf
}	—	—
}	—	—
if(audio_codec_id != 2){	—	—
raw_frame_length	16	bslbf
}	—	—
aatf_error_check()	—	—
if(audio_codec_id==1)	—	—
channel_number	{4; 8}	bslbf
if(audio_codec_id==2){	—	—
if(coding_profile ==0){	—	—
channel_number_index	7	bslbf
}	—	—
if(coding_profile ==1){	—	—
soundBedType	2	uimsbf
if (soundBedType == 0){	—	—
object_channel_number	7	uimsbf
bitrate_index_per_channel	4	uimsbf
} else if (soundBedType == 1){	—	—
channel_number_index	7	uimsbf
bitrate_index	4	uimsbf
object_channel_number	7	uimsbf
bitrate_index_per_channel	4	uimsbf
}	—	—
}	—	—
if(coding_profile ==2){	—	—

Syntax	Number of bits	Mnemonic
order	4	uimsbf
}	—	—
}	—	—
resolution	2	uimsbf
if(audio_codec_id==2 && coding_profile != 1){	—	—
bitrate_index	4	uimsbf
}	—	—
}	—	—

The syntax of aatf_error_check() should comply with provisions in Table A.7.

Table A.7 Syntax of aatf_error_check()

Syntax	Number of bits	Mnemonic
aatf_error_check()	—	—
{	—	—
crc_check	8	rpchof
}	—	—

A.2 Semantics

ll_raw_data_stream()

It indicates the sequence of ll_raw_data_block(), which is generated by a lossless audio coding tool.

ga_co_raw_data_stream()

It indicates the sequence of ga_co_raw_data_block(), which is generated by a general bit rate audio coding tool.

raw_stream_length

16 bits. It indicates the length of the raw audio data stream, in bytes.

audio_codec_id

4 bits. '1': lossless audio coding data; '2': general bit rate audio coding data; '0', '3' to '15': reserved.

anc_data_index

1 bit. It indicates whether an auxiliary data block exists.

resolution

2 bits. It indicates an index of the number of quantization bits of the input signal. '1': 16 bits per sampling point; '2': 24 bits per sampling point; '0' and '3': reserved.

nn_type

3 bits. '0': a basic neural network configuration; '1': a low-complexity neural network configuration; '2' to '7': reserved.

coding_profile

3 bits. '0': basic framework; '1': object metadata coding framework; '2': FOA/HOA data coding framework; '3' to '7': reserved.

channel_number_index

7 bits. It indicates an index of the number of channels, and should comply with provisions in Table A.8. The spatial position of the speaker should comply with provisions in GY/T 316-2018.

channel_number

When audio_codec_id is 1 and the number of channels is less than 16, this field contains 4 bits. '0' to '14' indicate the number of channels, and '15' indicates extension. When audio_codec_id is 1 and the number of channels is greater than or equal to 16, this field contains 8 bits, and indicates the number of channels.

order

4 bits. (order + 1) indicates the order of the FOA/HOA signal.

sampling_frequency_index

4 bits. It indicates an index of the sampling frequency of the input signal, and should comply with provisions in Table A.9.

sampling_frequency

A 24-bit unsigned integer. It indicates the sampling frequency value for extension, in the unit of hertz (Hz).

0x000000: 0 Hz;

0x000001: 1 Hz;

...

0xFFFFFE: 16777214 Hz;

0xFFFFFF: reserved.

bitrate_index

4 bits. It indicates an index of the bit rate, and should comply with provisions in Table A.10 to Table A.20.

ll_raw_data_block()

It indicates lossless audio coded raw bitstream data, and is generated by the lossless audio coding tool.

ga_co_raw_data_block()

It indicates general bit rate audio coded raw bitstream data, and is generated by the general bit rate audio coding tool.

aasf_sequence()

It indicates a sequence in the AVS_Audio_Storage_Format format, and should comply with provisions in Table A.1.

aasf_header()

It indicates an AASF header, which is located at the beginning of aasf_sequence, and should comply with provisions in Table A.2.

aatf_sequence()

It indicates a sequence in AVS_Audio_Transport_Format format, and should comply with provisions in Table A.3.

aatf_frame()

It indicates an AATF frame, and should comply with provisions in Table A.4.

aatf_frame_header()

It indicates an AATF decoding frame header, and should comply with provisions in Table A.6.

raw_frame_length

16 bits. It indicates the total length of the current frame in the bitstream. The flag of the total length of the current frame in the bitstream is added to obtain the basic information of the bitstream.

frame_error_check()

It indicates the data generated through the CRC check. All bits in the AATF frame are checked according to the CRC algorithm and the sequence in which they appear. It should comply with provisions in Table A.5.

aatf_error_check()

It indicates the data generated during the CRC check, and should comply with provisions in Table A.7.

syncword

12 bits. It indicates the synchronization word. The bit string is '1111 1111 1111'.

soundBedType

2 bits. It indicates the sound bed type.

'0': no sound bed, object audio only; '1': one of mono audio, dual-channel stereo audio, or multi-channel audio; '2' to '3': reserved.

object_channel_number

7 bits. (object_channel_number + 1) indicates the number of channels for all the object audio.

bitrate_index_per_channel

4 bits. It indicates an index of the average bit rate per channel for all the object audio. channel_number_index is reused.

crc_check

CRC check.

Table A.8 channel_number configuration table

channel_number_index	Channel configuration	channel_number
0x0	Mono	1
0x1	Dual-channel stereo	2
0x2	5.1	6
0x3	7.1	8
0x6	FOA	4
0x7	5.1.2	8
0x8	5.1.4	10
0x9	7.1.2	10
0xa	7.1.4	12
0xb	3-order HOA	16
0xc	2-order HOA	9
0x4, 0x5, and 0xd to 0x7f	Reserved	—

Table A.9 Sampling rate mapping table

sampling_frequency_index	Sampling frequency Hz
0x0	192000
0x1	96000
0x2	48000
0x3	44100
0x4 to 0xf	Reserved

Table A.10 Mono coding bit rate index table

bitrate_index	Bit rate kb/s
0x1	32
0x2	44

bitrate_index	**Bit rate** **kb/s**
0x3	56
0x4	64
0x5	72
0x6	80
0x7	96
0x8	128
0x9	144
0xa	164
0xb	192
0x0, and 0xc to 0xf	Reserved

Table A.11 Dual-channel stereo coding bit rate index table

bitrate_index	**Bit rate** **kb/s**
0x1	32
0x2	48
0x3	64
0x4	80
0x5	96
0x6	128
0x7	144
0x8	192
0x9	256
0xa	320
0x0, and 0xb to 0xf	Reserved

Table A.12 5.1 multi-channel coding bit rate index table

bitrate_index	**Bit rate** **kb/s**
0x0	192

bitrate_index	**Bit rate** **kb/s**
0x1	256
0x2	320
0x3	384
0x4	448
0x5	512
0x6	640
0x7	720
0x8	144
0x9	96
0xa	128
0xb	160
0xc to 0xf	Reserved

Table A.13 7.1 multi-channel coding bit rate index table

bitrate_index	**Bit rate** **kb/s**
0x0	192
0x1	480
0x2	256
0x3	384
0x4	576

Table A.13 (continued)

bitrate_index	**Bit rate** **kb/s**
0x5	640
0x6	128
0x7	160
0x8 to 0xf	Reserved

Table A.14 FOA coding bit rate index table

bitrate_index	Bit rate kb/s
0x1	96
0x2	128
0x3	192
0x4	256
0x0, and 0x5 to 0xf	Reserved

Table A.15 5.1.2 multi-channel coding bit rate index table

bitrate_index	Bit rate kb/s
H	152
0x1	320
0x2	480
0x3	576
0x4 to 0xf	Reserved

Table A.16 5.1.4 multi-channel coding bit rate index table

bitrate_index	Bit rate kb/s
0x0	176
0x1	384
0x2	576
0x3	704
0x4	256
0x5	448
0x6 to 0xf	Reserved

Table A.17 7.1.2 multi-channel coding bit rate index table

bitrate_index	Bit rate kb/s
0x0	216

bitrate_index	Bit rate kb/s
0x1	480
0x2	576
0x3	384
0x4	768
0x5 to 0xf	Reserved

Table A.18 7.1.4 multi-channel coding bit rate index table

bitrate_index	Bit rate kb/s
0x0	240
0x1	608
0x2	384
0x3	512
0x4	832
0x5 to 0xf	Reserved

Table A.19 2-order HOA coding bit rate index table

bitrate_index	Bit rate kb/s
0x0	192
0x1	256
0x2	320
0x3	384
0x4	480
0x5	512
0x6	640
0x7 to 0xf	Reserved

Table A.20 3-order HOA coding bit rate index table

bitrate_index	**Bit rate** **kb/s**
0x0	256
0x1	320
0x2	384
0x3	512

Table A.20 (continued)

bitrate_index	**Bit rate** **kb/s**
0x4	640
0x5	896
0x6 to 0xf	Reserved

Annex B
(Normative)
Audio Code Table

The 3D audio code table should comply with provisions in Table B.1 to Table B.49.

Table B.1 Context range coding table

Index	Probability
1	0, 1, 65534, 65535, 65536
2	0, 1, 65534, 65535, 65536
3	0, 1, 65534, 65535, 65536
4	0, 1, 65534, 65535, 65536
5	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 16, 30, 69, 169, 405, 909, 1885, 3581, 6203, 9806, 14349, 20045, 27710, 39581, 63884, 65352, 65478, 65513, 65523, 65526, 65527, 65528, 65529, 65530, 65531, 65532, 65533, 65534, 65535, 65536
6	0, 1, 65534, 65535, 65536
7	0, 1, 65534, 65535, 65536
8	0, 1, 65534, 65535, 65536
9	0, 1, 65534, 65535, 65536
10	0, 1, 65534, 65535, 65536
11	0, 1, 65534, 65535, 65536
12	0, 1, 65534, 65535, 65536
13	0, 1, 2, 3, 4, 16, 227, 4637, 62245, 65417, 65525, 65531, 65532, 65533, 65534, 65535, 65536
14	0, 1, 65534, 65535, 65536
15	0, 1, 65534, 65535, 65536
16	0, 1, 65534, 65535, 65536

Table B.2 Layer 1 CNN (convolution kernel parameter kernel) of context decoding neural network parameter

Parameter value
–0x1.0292f20000000p–9, –0x1.366fe00000000p–11, –0x1.87d1260000000p–10,
–0x1.cc52800000000p–9, –0x1.6487740000000p–1, 0x1.828c6c0000000p–11, 0x1.0e08e00000000p–11, 0x1.b500fa0000000p–10, –0x1.2c5b220000000p–11,
0x1.1109140000000p–10, –0x1.899ed60000000p–9, –0x1.14eea20000000p–9,
–0x1.3e52fe0000000p–4, –0x1.70af560000000p–10, –0x1.2668220000000p–12,

Parameter value
0x1.c0e2fe0000000p–10,
0x1.ff18000000000p–9, 0x1.9878960000000p–10, 0x1.a7f9ec0000000p–8, 0x1.2fba2c0000000p–8, –0x1.40fa4e0000000p–2, –0x1.705e160000000p–9, –0x1.09d3900000000p–8, 0x1.c2f6460000000p–10, –0x1.5402600000000p–9, 0x1.da25b00000000p–10, –0x1.2bfaf60000000p–9, 0x1.46ced60000000p–10, –0x1.3890400000000p–1, –0x1.0021f00000000p–8, –0x1.57a5ba0000000p–10, –0x1.3e41140000000p–10,

Table B.2 (continued)

Parameter value
–0x1.6a29260000000p–7, –0x1.45bb160000000p–8, –0x1.3078d00000000p–7, –0x1.53d64e0000000p–7, –0x1.d2fd4a0000000p–2, 0x1.5911c00000000p–8, 0x1.6539ca0000000p–7, –0x1.36a2e60000000p–9, 0x1.dbac1c0000000p–9, 0x1.4fc8b40000000p–9, 0x1.2cdc1e0000000p–8, –0x1.80a7f80000000p–11, –0x1.d4c0160000000p–1, 0x1.04827c0000000p–11, –0x1.12470a0000000p–10, –0x1.24878e0000000p–9,
–0x1.eb9d460000000p–7, –0x1.d1f8600000000p–9, –0x1.7db0400000000p–7, –0x1.a270a20000000p–7, –0x1.53a65a0000000p–1, 0x1.7cc2a40000000p–8, 0x1.4d8f800000000p–7, 0x1.fabc2c0000000p–15, 0x1.b9bba80000000p–8, –0x1.0820600000000p–9, 0x1.48a9ec0000000p–8, –0x1.6501ec0000000p–10, –0x1.547f9a0000000p–1, 0x1.aed1c40000000p–9, 0x1.3f92960000000p–9, –0x1.589be20000000p–10,
0x1.55c9fc0000000p–8, 0x1.5fef600000000p–8, 0x1.5abd8a0000000p–8, 0x1.43a8580000000p–8, –0x1.a74e920000000p–2, –0x1.121cd40000000p–8, –0x1.7a86360000000p–8, –0x1.80dea20000000p–10, –0x1.631c460000000p–8, 0x1.eb8acc0000000p–10, 0x1.76a6680000000p–13, 0x1.616ffe0000000p–11, –0x1.00a68c0000000p–2, 0x1.b95ba60000000p–11, –0x1.455f8e0000000p–9, –0x1.b823ac0000000p–13,
–0x1.338c020000000p–5, –0x1.d717660000000p–7, –0x1.f4a05e0000000p–6, –0x1.06740e0000000p–5, –0x1.c5f5940000000p–1, 0x1.c7f1700000000p–7, 0x1.d753300000000p–6, –0x1.f8530e0000000p–9, 0x1.fe39c40000000p–7, –0x1.842c660000000p–10, 0x1.22700e0000000p–7, –0x1.e0d3140000000p–8, –0x1.c33e720000000p–5, 0x1.1cb7320000000p–7, 0x1.a451b80000000p–8, 0x1.226df40000000p–9,
0x1.f359840000000p–11, –0x1.104b300000000p–10, 0x1.1676a80000000p–10, 0x1.6adc2c0000000p–12, –0x1.1f921c0000000p–1, 0x1.580dc00000000p–9, 0x1.cb52920000000p–13, –0x1.6313d80000000p–12, 0x1.77a0120000000p–12, 0x1.b043e60000000p–12, 0x1.5bc0320000000p–10, –0x1.3a8bee0000000p–11,

Parameter value
–0x1.5249920000000p–5, 0x1.6a6f160000000p–10, –0x1.f626940000000p–13, –0x1.c109a80000000p–13,
0x1.4352880000000p–6, 0x1.42f98c0000000p–7, 0x1.0ff08a0000000p–6, 0x1.2a00d60000000p–6, –0x1.9dc7040000000p–1, –0x1.1645220000000p–7, –0x1.0b2c200000000p–6, 0x1.22cafc0000000p–9, –0x1.4f324e0000000p–7, –0x1.57332e0000000p–11, –0x1.c87e280000000p–8, 0x1.e999160000000p–9, 0x1.a929ce0000000p–6, –0x1.93fc580000000p–8, –0x1.31cc680000000p–8, 0x1.88ebca0000000p–10,
0x1.661a680000000p–6, 0x1.5a72120000000p–7, 0x1.51c0d40000000p–6, 0x1.4e4fba0000000p–6, –0x1.5ea56c0000000p–3, –0x1.29e23e0000000p–7, –0x1.4844440000000p–6, 0x1.c5cd260000000p–8, –0x1.1e96180000000p–7, 0x1.aa21580000000p–10, –0x1.712eba0000000p–9, 0x1.7cf6a60000000p–9, 0x1.ec5d460000000p–1, –0x1.1d7b320000000p–7, –0x1.5d53640000000p–8, –0x1.274ee20000000p–10,

Table B.2 (continued)

Parameter value
–0x1.8c5efa0000000p–7, –0x1.02360a0000000p–7, –0x1.c269b80000000p–7, –0x1.0d53ce0000000p–6, 0x1.41bf9e0000000p–2, 0x1.03e8880000000p–7, 0x1.9c7f0e0000000p–7, –0x1.2fc2980000000p–8, 0x1.df16840000000p–9, 0x1.6d733a0000000p–9, 0x1.5d40fa0000000p–8, –0x1.1da57a0000000p–8, –0x1.6854200000000p–2, 0x1.35eb9e0000000p–7, 0x1.32ec120000000p–8, –0x1.ab3cf60000000p–9,
–0x1.55a11c0000000p–4, –0x1.03bede0000000p–5, –0x1.16411e0000000p–4, –0x1.23561c0000000p–4, 0x1.c07daa0000000p–8, 0x1.a8beec0000000p–6, 0x1.0314840000000p–4, –0x1.6e54b60000000p–7, 0x1.046d800000000p–5, 0x1.8d052a0000000p–10, 0x1.59b9ae0000000p–6, –0x1.9c1e260000000p–7, 0x1.13f1800000000p–3, 0x1.0cd2600000000p–6, 0x1.05ca240000000p–6, –0x1.0264e60000000p–9,
0x1.00c44c0000000p–6, 0x1.7f93a20000000p–8, 0x1.cb3f380000000p–7, 0x1.81a6ac0000000p–7, –0x1.c6a8f00000000p–4, –0x1.c6f7c00000000p–8, –0x1.a3d0840000000p–7, 0x1.0e63900000000p–8, –0x1.46a5ca0000000p–8, –0x1.2bdabe0000000p–10, –0x1.42b33e0000000p–8, 0x1.3462000000000p–10, –0x1.f8b61e0000000p–4, –0x1.0775400000000p–11, –0x1.5552b80000000p–9, 0x1.06f3f00000000p–15,
0x1.2692aa0000000p–8, 0x1.614c400000000p–10, 0x1.4396440000000p–8, 0x1.a255740000000p–8, –0x1.0ca9720000000p–3, –0x1.6093680000000p–9, –0x1.1892180000000p–8, 0x1.b350a60000000p–12, –0x1.5841fc0000000p–8,

Parameter value
–0x1.d51fe80000000p–10, –0x1.18bb860000000p–10, –0x1.545d7a0000000p–10, –0x1.2c272a0000000p+0, –0x1.77dffa0000000p–9, 0x1.7a89d60000000p–13, 0x1.7c57220000000p–11,
–0x1.8c939a0000000p–8, 0x1.0f32440000000p–10, –0x1.dfffd20000000p–8, –0x1.7aec5a0000000p–8, 0x1.eebdaa0000000p–4, 0x1.2a379c0000000p–8, 0x1.7dc1ce0000000p–8, 0x1.e6996e0000000p–13, 0x1.8a7fde0000000p–8, –0x1.a675fc0000000p–12, 0x1.1ec7420000000p–8, –0x1.e3ed200000000p–10, –0x1.1e9b840000000p+0, 0x1.e3ee460000000p–9, 0x1.15f5ba0000000p–8, –0x1.0aa18a0000000p–11,
–0x1.edc16a0000000p–9, –0x1.6912aa0000000p–8, –0x1.ab72ee0000000p–11, –0x1.c3f94a0000000p–10, –0x1.2b77740000000p–1, 0x1.bab7f00000000p–9, 0x1.3d11d20000000p–10, –0x1.724ce80000000p–9, –0x1.3723860000000p–10, –0x1.3033f20000000p–11, 0x1.fc38020000000p–11, –0x1.93710e0000000p–10, –0x1.70a8300000000p–3, –0x1.548f100000000p–10, –0x1.970bec0000000p–10, –0x1.68d1120000000p–9,
–0x1.8e7a740000000p–8, –0x1.6c89b60000000p–9, –0x1.33e5ea0000000p–8, –0x1.01ce900000000p–9, 0x1.b8829c0000000p–3, 0x1.090e860000000p–9, 0x1.a6940e0000000p–9, –0x1.e1aaec0000000p–9, 0x1.58d3ac0000000p–11, –0x1.1bc8aa0000000p–13, 0x1.7a75ba0000000p–11, –0x1.0b5f260000000p–11, 0x1.f27c580000000p–1, 0x1.390c0c0000000p–9, 0x1.bfb8f60000000p–12, –0x1.3e35480000000p–11,

Table B.2 (continued)

Parameter value
0x1.d15b6c0000000p+1, 0x1.15310a0000000p–2, 0x1.1c4af60000000p+0, 0x1.5667520000000p+0, 0x1.1318400000000p–4, –0x1.bd89b80000000p–3, –0x1.9110580000000p–1, 0x1.66adda0000000p–4, –0x1.08f7260000000p–2, –0x1.cd21920000000p–7, –0x1.2d1f4a0000000p–3, 0x1.eef6c00000000p–4, –0x1.3abeae0000000p–5, –0x1.14d6a80000000p–3, –0x1.afdb260000000p–4, –0x1.be7ee40000000p–9,
–0x1.ce70d40000000p–8, –0x1.0b4de20000000p–8, –0x1.a4a3860000000p–8, –0x1.347d5e0000000p–8, –0x1.c099ec0000000p–2, 0x1.253d7e0000000p–9, 0x1.6fd1320000000p–8, 0x1.47f58e0000000p–11, 0x1.186d2e0000000p–8, 0x1.4ae9cc0000000p–10, 0x1.061da00000000p–9, –0x1.476cc60000000p–11, 0x1.07bab00000000p–1, 0x1.ae89e80000000p–11, 0x1.125cf00000000p–8, 0x1.99ed8c0000000p–11,
–0x1.2db3540000000p–6, –0x1.e01b1a0000000p–8, –0x1.a878460000000p–7, –0x1.38785e0000000p–6, –0x1.37a3480000000p–5, 0x1.a3ba6c0000000p–8, 0x1.dcd3740000000p–7, –0x1.7185620000000p–10, 0x1.9173980000000p–8,

Parameter value
0x1.86255c0000000p–10, 0x1.96f1ae0000000p–8, –0x1.3114340000000p–9, –0x1.6786360000000p–2, 0x1.0288e20000000p–9, 0x1.ec3e5e0000000p–15, –0x1.2a9bf40000000p–11,
0x1.469e9a0000000p–6, 0x1.7108280000000p–8, 0x1.380c340000000p–6, 0x1.1d3d340000000p–6, –0x1.0fbb9e0000000p–1, –0x1.1c55c20000000p–7, –0x1.f1f0a40000000p–7, 0x1.07f3300000000p–8, –0x1.b7fafe0000000p–8, 0x1.6255320000000p–9, –0x1.346ef00000000p–8, 0x1.a948420000000p–9, 0x1.2fb6b80000000p–2, –0x1.383ce60000000p–8, –0x1.fe50aa0000000p–9, 0x1.13ed9e0000000p–10,
0x1.edf5680000000p–11, –0x1.703e680000000p–9, 0x1.e579680000000p–16, 0x1.cdc1d00000000p–12, –0x1.2e31940000000p–2, –0x1.1b89680000000p–9, 0x1.f088ca0000000p–10, 0x1.29d3280000000p–8, –0x1.6408800000000p–8, –0x1.557a820000000p–8, –0x1.343b1a0000000p–8, 0x1.541a160000000p–10, 0x1.1db0be0000000p–1, –0x1.af98ea0000000p–10, 0x1.d4002e0000000p–12, –0x1.28a1aa0000000p–9,
–0x1.28d2080000000p–9, –0x1.8c43040000000p–10, –0x1.d78a180000000p–10, –0x1.2a1e780000000p–9, –0x1.ad8b780000000p–1, 0x1.23d41a0000000p–10, 0x1.3f14fe0000000p–9, –0x1.43261c0000000p–16, 0x1.8d18de0000000p–11, –0x1.a77a9e0000000p–14, 0x1.2dc0ba0000000p–9, –0x1.496fea0000000p–11, 0x1.ba47b40000000p–4, 0x1.0405380000000p–9, 0x1.75308e0000000p–10, –0x1.b371460000000p–12,
–0x1.dc7cc20000000p–8, –0x1.6aa8a40000000p–8, –0x1.1b499a0000000p–7, –0x1.2769c00000000p–7, 0x1.13f3440000000p–6, 0x1.9922a40000000p–8, 0x1.bff37e0000000p–8, –0x1.3d44920000000p–10, 0x1.738e6a0000000p–9, 0x1.bc8d6a0000000p–10, 0x1.24ab360000000p–8, –0x1.1316340000000p–9, –0x1.9b07ca0000000p–2, 0x1.5ab1f80000000p–9, 0x1.c169b80000000p–9, 0x1.aa58a00000000p–11,

Table B.2 (continued)

Parameter value
0x1.62cb4a0000000p–10, 0x1.26baea0000000p–8, 0x1.f935be0000000p–11, 0x1.e0f51e0000000p–10, –0x1.68083e0000000p–3, –0x1.61f0ac0000000p–10, –0x1.36c6900000000p–9, 0x1.a34f580000000p–9, –0x1.2a732e0000000p–9, 0x1.d658ce0000000p–18, –0x1.3494280000000p–10, 0x1.8a5e980000000p–10, 0x1.6f1b5c0000000p–4, –0x1.6326040000000p–9, –0x1.d2e4e20000000p–12, –0x1.376f9c0000000p–11,
–0x1.d129820000000p–7, –0x1.fb78900000000p–8, –0x1.3d01c20000000p–7, –0x1.ef6d680000000p–8, –0x1.dd48b60000000p–7, 0x1.5c15a20000000p–8, 0x1.168cf00000000p–7, –0x1.a8885c0000000p–9, 0x1.20aeda0000000p–8,

Parameter value
0x1.de284c0000000p–10, 0x1.2302a40000000p–8, –0x1.d9ddd20000000p–13, –0x1.42b7f00000000p–1, 0x1.db8a1c0000000p–11, 0x1.fd34240000000p–15, –0x1.b7712c0000000p–17,
0x1.6c16140000000p–6, 0x1.0a3dfa0000000p–7, 0x1.589ed20000000p–6, 0x1.539eb40000000p–6, –0x1.f728be0000000p–4, –0x1.1d15ae0000000p–6, –0x1.ee18d20000000p–7, 0x1.eecb620000000p–8, –0x1.845ecc0000000p–7, –0x1.5eec220000000p–9, –0x1.29163a0000000p–10, 0x1.b234f60000000p–8, 0x1.a24c6e0000000p–4, –0x1.9954ee0000000p–8, –0x1.b3c5f00000000p–9, –0x1.2b6d2e0000000p–9,
0x1.a56bf40000000p–6, 0x1.9fbfa40000000p–7, 0x1.5d2a140000000p–6, 0x1.5c38200000000p–6, 0x1.0ce4a40000000p–6, –0x1.3e65ec0000000p–7, –0x1.615b9c0000000p–6, 0x1.45b1680000000p–8, –0x1.9e2b3c0000000p–7, –0x1.bc51180000000p–10, –0x1.fd96da0000000p–8, 0x1.941d3c0000000p–8, –0x1.361bdc0000000p–2, –0x1.0d2d040000000p–7, –0x1.8e00440000000p–8, 0x1.946bea0000000p–11,
0x1.23504c0000000p–11, –0x1.d3783c0000000p–10, –0x1.37ffec0000000p–11, 0x1.6cade20000000p–13, –0x1.37f2540000000p–2, –0x1.dddff00000000p–11, 0x1.14dca40000000p–10, 0x1.42b2420000000p–10, 0x1.6f5aec0000000p–12, 0x1.97d85e0000000p–11, 0x1.84b9280000000p–11, –0x1.64918c0000000p–10, 0x1.18c7e00000000p–2, 0x1.491b0e0000000p–17, –0x1.f7f1300000000p–15, –0x1.21f60a0000000p–10,
–0x1.5044260000000p–7, –0x1.41fff80000000p–8, –0x1.447aa40000000p–7, –0x1.3c44560000000p–7, –0x1.ea45980000000p–3, 0x1.0fc69c0000000p–8, 0x1.27b4540000000p–7, 0x1.5c886a0000000p–11, 0x1.df79440000000p–9, –0x1.03639a0000000p–10, 0x1.966ed60000000p–10, 0x1.6ed2980000000p–10, 0x1.9d97de0000000p–1, 0x1.093cce0000000p–12, 0x1.c0992a0000000p–11, 0x1.75319c0000000p–9,
–0x1.4b50fc0000000p–8, –0x1.568b140000000p–10, –0x1.355d120000000p–9, –0x1.2109ee0000000p–8, –0x1.f4ddb40000000p–2, 0x1.3cebfe0000000p–8, 0x1.38d9a40000000p–8, –0x1.8642aa0000000p–10, 0x1.74fe040000000p–8, 0x1.c86f540000000p–10, –0x1.e6e05c0000000p–12, 0x1.2964780000000p–11, –0x1.4e58020000000p–6, 0x1.750fa40000000p–10, –0x1.733b860000000p–9, 0x1.5d6fbe0000000p–9,

Table B.2 (continued)

Parameter value
0x1.9891be0000000p–4, 0x1.9082100000000p–4, –0x1.084cb40000000p–4, –0x1.a86ba60000000p–3, 0x1.42737a0000000p–3, 0x1.2a2e200000000p–3,

Parameter value
–0x1.960aec0000000p–3, 0x1.ac39b80000000p–7, –0x1.7291fe0000000p–3, –0x1.5365d80000000p–4, –0x1.17fdea0000000p–5, 0x1.a6a4d60000000p–3, 0x1.fd811a0000000p–4, –0x1.293aa60000000p–5, –0x1.2a90740000000p–4, 0x1.52e80a0000000p–9,
0x1.ea70180000000p–10, –0x1.edc9fa0000000p–13, 0x1.be09b00000000p–9, 0x1.6fdbac0000000p–10, 0x1.2b09460000000p–1, –0x1.dfad280000000p–10, –0x1.9809e60000000p–10, –0x1.7310ba0000000p–10, –0x1.d8ecdc0000000p–9, –0x1.31f2d00000000p–9, –0x1.b5bffc0000000p–9, –0x1.2677300000000p–9, 0x1.c30c6e0000000p–2, –0x1.333edc0000000p–11, 0x1.8a138a0000000p–11, –0x1.7311d40000000p–10,
0x1.5f9de80000000p+2, 0x1.872ab80000000p–1, 0x1.6f57f00000000p+1, 0x1.a7c9d20000000p+1, 0x1.96798a0000000p–1, –0x1.4346520000000p–1, –0x1.1bf3820000000p+1, 0x1.c87f220000000p–3, –0x1.8641320000000p–1, –0x1.f366780000000p–6, –0x1.94dfe00000000p–2, 0x1.36e74a0000000p–2, –0x1.faf0c00000000p–4, –0x1.9a0c560000000p–2, –0x1.32b1e20000000p–2, 0x1.451ef20000000p–8,
–0x1.d1346c0000000p–7, –0x1.1887c40000000p–8, –0x1.7cca660000000p–7, –0x1.c0a9c20000000p–7, –0x1.e207b80000000p–4, 0x1.ac2da20000000p–8, 0x1.2abd920000000p–7, –0x1.9cc5580000000p–10, 0x1.b7435e0000000p–9, –0x1.3cd3760000000p–8, 0x1.b721ce0000000p–13, –0x1.a595280000000p–11, 0x1.284f380000000p–4, 0x1.3d72340000000p–12, –0x1.fd135c0000000p–10, 0x1.7637720000000p–9,
0x1.5b416e0000000p–9, 0x1.609f700000000p–10, 0x1.bb98760000000p–9, 0x1.6ffa360000000p–9, 0x1.722f200000000p–7, –0x1.2f83aa0000000p–9, –0x1.1b94d60000000p–9, 0x1.20d0840000000p–11, –0x1.4496c80000000p–8, –0x1.aa35b40000000p–13, –0x1.9ce2c00000000p–10, 0x1.9aef080000000p–10, –0x1.3c07b60000000p–10, –0x1.f8ef860000000p–9, –0x1.118a7a0000000p–10, 0x1.9d75080000000p–11,
–0x1.1ecb040000000p–5, –0x1.f10fb80000000p–7, –0x1.fd35ea0000000p–6, –0x1.e8976a0000000p–6, –0x1.02c5f40000000p–4, 0x1.abe8100000000p–7, 0x1.c9c0d40000000p–6, –0x1.bba9da0000000p–9, 0x1.e1c6100000000p–7, 0x1.fe275a0000000p–10, 0x1.bf63920000000p–8, –0x1.412e2e0000000p–8, 0x1.b4e6300000000p–5, 0x1.8ee4220000000p–7, 0x1.cf3fd60000000p–9, –0x1.0619240000000p–10,
0x1.65a3b20000000p–5, 0x1.076bd40000000p–6, 0x1.2b28f80000000p–5, 0x1.32b92a0000000p–5, –0x1.1e3ede0000000p–3, –0x1.3d14e00000000p–7, –0x1.01cd9a0000000p–5, 0x1.9187ac0000000p–8, –0x1.0654000000000p–6, 0x1.eb10400000000p–10, –0x1.85d4f60000000p–8, 0x1.0ef6620000000p–7, 0x1.0ac6e60000000p–6, –0x1.2494ac0000000p–7, –0x1.4200620000000p–8, –0x1.8b31420000000p–12,

Table B.2 (continued)

Parameter value
–0x1.3d84140000000p–5, –0x1.fab80a0000000p–7, –0x1.03eb5a0000000p–5, –0x1.0996600000000p–5, –0x1.01dde60000000p–1, 0x1.aed93c0000000p–7, 0x1.eaba620000000p–6, –0x1.378fbe0000000p–8, 0x1.f49dd60000000p–7, –0x1.eda8c60000000p–13, 0x1.0a266a0000000p–7, –0x1.e599ea0000000p–8, –0x1.f9a9e60000000p–4, 0x1.1620020000000p–7, 0x1.a953620000000p–8, 0x1.4437fa0000000p–13,
0x1.5067920000000p–7, 0x1.f5e5760000000p–9, 0x1.3b30560000000p–7, 0x1.1978ea0000000p–7, –0x1.f5830e0000000p–2, –0x1.83560a0000000p–9, –0x1.1d93000000000p–7, 0x1.ddc9fe0000000p–13, –0x1.147f400000000p–9, 0x1.98d2ec0000000p–11, –0x1.0cc0bc0000000p–9, 0x1.3fe3240000000p–9, 0x1.038dea0000000p–6, –0x1.24b3220000000p–9, –0x1.fee3e40000000p–10, 0x1.2d1d780000000p–10,
0x1.cffa960000000p–6, 0x1.9df73a0000000p–7, 0x1.98d8c80000000p–6, 0x1.a6f13a0000000p–6, 0x1.3f5b880000000p–1, –0x1.ba056a0000000p–7, –0x1.8a052a0000000p–6, 0x1.b420fa0000000p–9, –0x1.80636c0000000p–7, –0x1.e4249c0000000p–11, –0x1.c61e300000000p–8, 0x1.cbc3c00000000p–8, 0x1.a000b40000000p–5, –0x1.eb431a0000000p–8, –0x1.94f6c80000000p–8, 0x1.302db00000000p–10,
0x1.7a26620000000p–5, 0x1.d10b640000000p–7, 0x1.3d7bd60000000p–5, 0x1.51ef300000000p–5, 0x1.50961c0000000p–2, –0x1.01cd780000000p–6, –0x1.10dc640000000p–5, 0x1.2e269e0000000p–7, –0x1.0eef620000000p–6, –0x1.03ea9a0000000p–8, –0x1.59ec400000000p–7, 0x1.3e0f720000000p–7, 0x1.ccdfba0000000p–7, –0x1.5846420000000p–7, –0x1.82822c0000000p–8, 0x1.1e4cb80000000p–9,
–0x1.360ee40000000p–6, –0x1.e31c000000000p–8, –0x1.bafb980000000p–7, –0x1.e5db8c0000000p–7, –0x1.3e66c20000000p–2, 0x1.a744e60000000p–9, 0x1.662c9e0000000p–7, 0x1.99cb0c0000000p–10, 0x1.865b080000000p–7, 0x1.eb5efa0000000p–10, 0x1.0576780000000p–7, –0x1.1dffe60000000p–11, 0x1.d011860000000p–4, 0x1.4ae8240000000p–9, 0x1.9e5a7a0000000p–8, –0x1.5069ec0000000p–9,
–0x1.a09d660000000p–4, –0x1.530d420000000p–5, –0x1.63017c0000000p–4, –0x1.7213e40000000p–4, –0x1.debafe0000000p–2, 0x1.1ecff80000000p–5, 0x1.4390d40000000p–4, –0x1.cd4d280000000p–7, 0x1.4def1a0000000p–5, –0x1.a9574e0000000p–12, 0x1.54792e0000000p–6, –0x1.22805a0000000p–6, –0x1.1d847c0000000p–2, 0x1.71cd640000000p–6, 0x1.13ccec0000000p–6, –0x1.0bf1bc0000000p–10,
0x1.222f680000000p–6, 0x1.3d90e00000000p–7, 0x1.1e3e2e0000000p–6, 0x1.00114c0000000p–6, –0x1.031bd40000000p–3, –0x1.9cab920000000p–8, –0x1.d5c02a0000000p–7, 0x1.36afae0000000p–9, –0x1.5113e00000000p–7,

Parameter value
0x1.7490ba0000000p–11, –0x1.79b4380000000p–8, 0x1.91eea20000000p–9, –0x1.0fd2460000000p–5, –0x1.24c3880000000p–9, –0x1.e6d70e0000000p–9, –0x1.16d6740000000p–11,

Table B.2 (continued)

Parameter value
0x1.8c2a820000000p–7, 0x1.cb8fcc0000000p–8, 0x1.376a600000000p–7, 0x1.5501ae0000000p–7, 0x1.d782640000000p–4, –0x1.5df7820000000p–8, –0x1.5f99f20000000p–7, 0x1.898a6e0000000p–9, –0x1.1fece00000000p–7, 0x1.0b0c0e0000000p–11, –0x1.9aad440000000p–11, 0x1.9b21560000000p–8, –0x1.3339cc0000000p–3, –0x1.325f160000000p–10, –0x1.24882c0000000p–9, –0x1.d7b5b60000000p–10,
0x1.6472ba0000000p–6, 0x1.65a1d20000000p–7, 0x1.2909320000000p–6, 0x1.7bb62e0000000p–6, –0x1.e5bbcc0000000p–4, –0x1.aedd740000000p–8, –0x1.37f01a0000000p–6, 0x1.5040660000000p–9, –0x1.0bb0aa0000000p–7, 0x1.4a7af00000000p–9, –0x1.d5243e0000000p–8, 0x1.c00c460000000p–9, 0x1.8f6c500000000p–2, –0x1.1527f80000000p–7, –0x1.656eaa0000000p–8, 0x1.2c8bc00000000p–10,
0x1.dce1f20000000p–5, 0x1.1e74d40000000p–5, 0x1.d4a4f60000000p–5, 0x1.aed3d20000000p–5, 0x1.ff698e0000000p–2, –0x1.3491480000000p–6, –0x1.988d380000000p–5, 0x1.52f33c0000000p–7, –0x1.87be160000000p–6, –0x1.0a44ae0000000p–12, –0x1.8123ce0000000p–8, 0x1.1df4ac0000000p–7, 0x1.e86eb60000000p–2, –0x1.1bf5a40000000p–8, –0x1.841a220000000p–7, 0x1.a8e5200000000p–11,
–0x1.3f89c20000000p–8, –0x1.9e28d80000000p–10, –0x1.7eb31e0000000p–9, –0x1.089d0c0000000p–8, –0x1.975ac60000000p–3, 0x1.fe0e6e0000000p–13, 0x1.61e1580000000p–9, –0x1.7fed040000000p–12, 0x1.0b0cec0000000p–10, 0x1.3551980000000p–16, –0x1.9df84e0000000p–11, –0x1.817b9c0000000p–10, 0x1.3e828a0000000p–7, –0x1.3533d80000000p–12, 0x1.9fd8f60000000p–16, –0x1.ebf3800000000p–9,

Table B.3 Layer 1 CNN (bias parameter bias) of context decoding neural network parameter

Parameter value
0x1.b737080000000p–3, –0x1.635d260000000p+1, –0x1.993fd20000000p–2, –0x1.c72c020000000p+0, –0x1.63e0fa0000000p+1, 0x1.41686e0000000p–2, –0x1.f642a00000000p–3, 0x1.4067f20000000p–1, –0x1.0798e40000000p–2, –0x1.eb1dd60000000p–1, 0x1.04c7d80000000p–5, –0x1.76560e0000000p–1,

Parameter value
–0x1.b0c64a0000000p–2, –0x1.2410f60000000p–1, –0x1.0f3c6c0000000p+0, –0x1.d71c320000000p–2,

Table B.4 Layer 2 CNN (convolution kernel parameter kernel) of context decoding neural network parameter

Parameter value
0x1.2a9e6a0000000p–1, 0x1.acf3840000000p+1, 0x1.8011da0000000p–2, 0x1.a34f7a0000000p–1, 0x1.57683a0000000p+1, –0x1.40068e0000000p–2, 0x1.a4af4a0000000p–6, –0x1.c33c040000000p–6, –0x1.a70f300000000p–3, –0x1.9a64a80000000p–2, –0x1.50111e0000000p–1, 0x1.da8a060000000p–3, 0x1.a3f6d40000000p–4, 0x1.3059a40000000p–3, 0x1.beaa700000000p–2, –0x1.5a652a0000000p+0,
–0x1.3cb0220000000p–7, 0x1.94a3600000000p+0, 0x1.6fa2b40000000p–3, 0x1.41c3340000000p–3, 0x1.7b7b320000000p–5, 0x1.e327ea0000000p–3, –0x1.cc8d1a0000000p–3, –0x1.110b140000000p–3, 0x1.4a56680000000p–5, 0x1.d96b240000000p–4, –0x1.434bbe0000000p–3, 0x1.bb9c240000000p–4, 0x1.d2adb00000000p–2, –0x1.8218540000000p–6, –0x1.326f560000000p–11, –0x1.d9cfb00000000p–4,
0x1.77449c0000000p–2, 0x1.d30b460000000p–1, –0x1.83be980000000p–3, 0x1.89c0e00000000p–3, 0x1.d443b20000000p–2, 0x1.fc7a820000000p–13, –0x1.d5c4100000000p–2, –0x1.2be84a0000000p–4, –0x1.2f67540000000p–3, 0x1.f10a5e0000000p–3, –0x1.7777040000000p–2, –0x1.718d460000000p–8, 0x1.457ad20000000p–3, –0x1.38cd280000000p–5, 0x1.3530880000000p–2, –0x1.dc22f20000000p–1,
0x1.2583740000000p–1, 0x1.2a62620000000p+2, 0x1.4452900000000p–3, 0x1.92e2bc0000000p+0, 0x1.c14b3e0000000p+0, –0x1.7ce9480000000p–2, –0x1.02a5fe0000000p–1, –0x1.d6e4e60000000p–2, –0x1.3687700000000p–6, –0x1.e55ad40000000p–3, –0x1.dd72760000000p–2, 0x1.cd121e0000000p–1, 0x1.7ac87c0000000p–2, –0x1.4ca4f60000000p–2, 0x1.6763c00000000p–1, –0x1.3eb2900000000p+1,
0x1.bb17b60000000p–1, 0x1.73fba80000000p+0, 0x1.3c39a60000000p–5, 0x1.f4fd9e0000000p–3, 0x1.39a19e0000000p–1, –0x1.f54e040000000p–3, –0x1.5cb8760000000p–3, –0x1.da4c0c0000000p–4, –0x1.b9bda20000000p–2, –0x1.bf87ba0000000p–2, –0x1.bbb1200000000p–4, 0x1.81fb200000000p–2, –0x1.1bbf9c0000000p–3, 0x1.099bd20000000p–5, 0x1.b7b3960000000p–3, –0x1.b7951e0000000p–1,
–0x1.d6b2280000000p–8, 0x1.1f7f660000000p–2, 0x1.4550500000000p–4, 0x1.48d4d60000000p–3, –0x1.07163e0000000p–2, –0x1.2e35000000000p–3, 0x1.ff12200000000p–4, –0x1.f67c120000000p–10, –0x1.913bec0000000p–3,

Parameter value
0x1.9e0c420000000p–6, 0x1.075c0a0000000p–5, 0x1.eb70ec0000000p–1, 0x1.0f39060000000p–4, 0x1.af02500000000p–3, 0x1.aaa3940000000p–3, –0x1.1491200000000p–2,
0x1.53bcb40000000p–6, 0x1.efbd120000000p–1, 0x1.42dd3a0000000p–3, 0x1.7dd7be0000000p–3, 0x1.bc7f2e0000000p–2, 0x1.6d57a20000000p–3, –0x1.55391c0000000p–3, –0x1.0ff51a0000000p–3, –0x1.4aa7740000000p–2, 0x1.0bc9300000000p–2, –0x1.9300620000000p–4, 0x1.b8bfa60000000p–4, 0x1.4e46120000000p–6, 0x1.6d65fa0000000p–2, 0x1.905ee20000000p–2, –0x1.81c9f40000000p–3,

Table B.4 (continued)

Parameter value
–0x1.1b3f120000000p–11, –0x1.4fbe880000000p–2, 0x1.d5deb80000000p–4, –0x1.d0931a0000000p–5, 0x1.3efb540000000p–3, 0x1.4acc940000000p–4, –0x1.1c4dee0000000p–1, –0x1.2062660000000p–4, –0x1.d5e9600000000p–3, –0x1.fdc1580000000p–2, –0x1.c340940000000p–5, –0x1.39c68a0000000p–3, 0x1.6e35f00000000p–6, 0x1.e144700000000p–6, 0x1.3c55560000000p–1, –0x1.2bf3420000000p–1,
0x1.1000100000000p–2, –0x1.b60bfa0000000p–4, –0x1.19c0f80000000p–3, 0x1.b386740000000p–5, –0x1.3e315c0000000p–2, 0x1.091b780000000p–1, –0x1.7b52a20000000p–2, 0x1.68c9f60000000p–3, –0x1.2d392a0000000p–3, 0x1.f9b3540000000p–4, –0x1.dc8a1c0000000p–3, 0x1.ed66720000000p–2, –0x1.7c633a0000000p–2, 0x1.84b7700000000p–3, 0x1.cf38fc0000000p–2, –0x1.303da40000000p–3,
0x1.8230180000000p–3, 0x1.60d3980000000p+0, –0x1.4f231e0000000p–3, 0x1.1e03b20000000p–1, 0x1.a2d0240000000p+0, –0x1.9302480000000p–4, –0x1.1a1f120000000p–2, –0x1.8a676e0000000p–4, –0x1.78ca920000000p–1, 0x1.ba15060000000p+0, –0x1.8074740000000p–3, –0x1.14e95c0000000p–3, 0x1.460d080000000p–3, 0x1.1a6d560000000p–2, 0x1.abeaae0000000p–2, –0x1.1f10ac0000000p+1,
0x1.27f8fa0000000p–2, 0x1.b7753a0000000p–3, –0x1.61e35c0000000p–3, 0x1.5469780000000p–2, 0x1.dcc1d20000000p–1, –0x1.51dcd60000000p–3, 0x1.9e2fb80000000p–4, –0x1.e4e8160000000p–3, –0x1.a158700000000p–1, –0x1.3066340000000p–1, 0x1.05926a0000000p–5, –0x1.31349c0000000p–2, 0x1.62087c0000000p–6, 0x1.02b5c40000000p–2, 0x1.bebb1c0000000p–2, –0x1.9e570a0000000p–2,
–0x1.8c405c0000000p–2, 0x1.120a5a0000000p+0, 0x1.db47620000000p–5, 0x1.2fd7c80000000p–1, 0x1.38474c0000000p–1, –0x1.31069c0000000p–6,

Parameter value
–0x1.54876e0000000p–5, 0x1.0b473c0000000p–3, –0x1.62150a0000000p–1, –0x1.5801080000000p–1, 0x1.8e8fda0000000p–4, 0x1.8db6640000000p–3, 0x1.8acaea0000000p–3, 0x1.2979f60000000p–2, 0x1.18c40a0000000p–3, –0x1.455a1c0000000p–1,
0x1.8ea3300000000p–2, 0x1.0f30940000000p–1, 0x1.0eb5540000000p–2, 0x1.4e016c0000000p–2, –0x1.4bb0080000000p–1, –0x1.138fbe0000000p–4, –0x1.81690c0000000p–2, –0x1.b7ba600000000p–4, –0x1.d450160000000p–4, 0x1.e6eaac0000000p–3, –0x1.319b6a0000000p–3, –0x1.09c4720000000p–4, –0x1.4275720000000p–3, 0x1.88c1560000000p–4, –0x1.0f4f560000000p–1, –0x1.ee3f540000000p–3,
–0x1.0f28240000000p–3, 0x1.4869b00000000p–1, 0x1.58f72a0000000p–3, 0x1.1059460000000p–2, 0x1.d12d2e0000000p–2, 0x1.352c680000000p–3, –0x1.30f94c0000000p–4, 0x1.02b0180000000p–4, 0x1.dc1d780000000p–6, 0x1.5a10780000000p–1, 0x1.2dedfc0000000p–3, 0x1.d7e6bc0000000p–2, 0x1.85996a0000000p–3, 0x1.dcb52c0000000p–5, 0x1.5cf15a0000000p–2, –0x1.9e21580000000p–1,

Table B.4 (continued)

Parameter value
0x1.db76ce0000000p–2, 0x1.0a93420000000p–1, 0x1.54dad80000000p–2, 0x1.0d613a0000000p–2, –0x1.d70bc00000000p–1, –0x1.5637700000000p–3, –0x1.065ed60000000p–2, –0x1.6a8dac0000000p–11, –0x1.df807c0000000p–4, 0x1.3fa36a0000000p–7, 0x1.dcfb120000000p–3, 0x1.0d22f40000000p–5, 0x1.7f04300000000p–3, 0x1.679f120000000p–1, –0x1.dfba660000000p–4, –0x1.09932a0000000p+0,
–0x1.67a2fc0000000p–1, –0x1.2511400000000p–4, –0x1.7b7c040000000p–2, 0x1.7519c40000000p–5, 0x1.a6e82e0000000p–2, 0x1.1c3b8e0000000p–5, –0x1.1961280000000p–3, 0x1.f94ce00000000p–3, –0x1.06b0840000000p–1, –0x1.0ccce80000000p+0, 0x1.5e820a0000000p–2, 0x1.099b300000000p–2, –0x1.a33c5c0000000p–3, –0x1.9eb9820000000p–3, –0x1.7523d80000000p–2, –0x1.0a4a160000000p+0,
0x1.cf0f780000000p–2, 0x1.2dfe800000000p–2, –0x1.1691820000000p–1, 0x1.97642c0000000p–1, 0x1.fb20900000000p–1, –0x1.03564a0000000p–3, –0x1.dbbae60000000p–4, –0x1.9717c20000000p–6, –0x1.b79acc0000000p–2, –0x1.1815c40000000p–1, 0x1.f484c80000000p–3, 0x1.b15d580000000p–4, –0x1.3f217a0000000p–3, 0x1.f24d220000000p–7, 0x1.d44eca0000000p–2, –0x1.da581e0000000p–2,
0x1.6930940000000p–3, 0x1.e5cc000000000p–3, 0x1.0a8d9e0000000p–4,

Parameter value
0x1.44c8120000000p–4, 0x1.afc5680000000p–3, 0x1.c969dc0000000p–4, –0x1.bf5eb00000000p–4, –0x1.7489da0000000p–4, –0x1.387fd20000000p–3, –0x1.096c660000000p–1, –0x1.e962280000000p–5, 0x1.0270020000000p–5, 0x1.afdb460000000p–3, 0x1.9272dc0000000p–4, 0x1.4ae3120000000p–3, –0x1.683b820000000p–5,
–0x1.496dfa0000000p–2, –0x1.3d3b9a0000000p–3, 0x1.8db2f00000000p–8, –0x1.9cab300000000p–6, 0x1.202afa0000000p–5, 0x1.e32d720000000p–6, –0x1.e8c76c0000000p–9, –0x1.9ccfa80000000p–3, –0x1.93ddd60000000p–7, 0x1.7336780000000p–4, –0x1.088a500000000p–5, 0x1.4aaf7e0000000p–3, –0x1.22685a0000000p–2, –0x1.93e2620000000p–4, 0x1.fdf6b80000000p–4, 0x1.87d7ee0000000p–8,
–0x1.4abd900000000p–5, 0x1.3cbb060000000p–2, 0x1.39855e0000000p–5, 0x1.56a0820000000p–1, 0x1.3589520000000p+0, –0x1.0bfe980000000p–3, –0x1.8f2b360000000p–3, –0x1.48fb160000000p–3, 0x1.465d1c0000000p–4, –0x1.848c780000000p–1, –0x1.664cd20000000p–3, 0x1.e9455e0000000p–3, 0x1.5865700000000p–3, 0x1.07b5f20000000p–3, 0x1.9b14100000000p–2, –0x1.34f4460000000p–2,
0x1.87bbba0000000p–4, 0x1.3c66d40000000p–1, –0x1.6b88620000000p–3, –0x1.5480aa0000000p–3, 0x1.26af540000000p–1, 0x1.c908e80000000p–5, –0x1.4b55820000000p–3, 0x1.5522c40000000p–5, 0x1.4787aa0000000p–4, –0x1.165e0e0000000p–3, 0x1.e8e0fa0000000p–5, –0x1.77acb80000000p–2, –0x1.25b4200000000p–1, –0x1.34e84c0000000p+0, 0x1.cb5b520000000p–5, –0x1.5da8d60000000p–3,

Table B.4 (continued)

Parameter value
0x1.658d5e0000000p–1, 0x1.a245cc0000000p–3, 0x1.e4dd3e0000000p–6, –0x1.075f880000000p–6, 0x1.4604dc0000000p–2, –0x1.f3e5640000000p–6, 0x1.e5d6200000000p–6, –0x1.6f73a60000000p–1, –0x1.56278e0000000p+0, –0x1.44411c0000000p–3, 0x1.4631060000000p–4, 0x1.146eac0000000p–1, 0x1.3844200000000p–5, –0x1.2112d40000000p–4, 0x1.93d6240000000p–2, –0x1.093f660000000p–4,
0x1.3c669c0000000p–13, –0x1.6c2ea40000000p–8, 0x1.90a1c00000000p–4, 0x1.4e92cc0000000p–3, 0x1.3373a40000000p–1, 0x1.8762160000000p–3, –0x1.01714e0000000p–3, –0x1.3214d40000000p–3, –0x1.ca05b40000000p–5, –0x1.fa776a0000000p–3, –0x1.5d58f40000000p–4, –0x1.53f3ae0000000p–5, 0x1.251b820000000p–2, –0x1.e8d3dc0000000p–7, –0x1.62a20c0000000p–2, –0x1.0cef0a0000000p–6,
0x1.04e31c0000000p–2, –0x1.6d429a0000000p–3, –0x1.468df60000000p–7,

Parameter value
–0x1.8506120000000p–5, –0x1.92ade00000000p–1, 0x1.a516180000000p–5, –0x1.95fcca0000000p–5, –0x1.03e9400000000p–4, –0x1.0212180000000p–3, 0x1.2e9a300000000p–6, –0x1.3c7c860000000p–5, 0x1.4da71e0000000p–3, 0x1.5529160000000p–6, –0x1.3fc3880000000p–3, 0x1.bef81a0000000p–4, 0x1.43753c0000000p–4,
0x1.eefb640000000p–2, 0x1.1fd5d40000000p–2, 0x1.128af40000000p–3, 0x1.4ed46a0000000p–5, 0x1.1d5a220000000p–4, 0x1.4e00c60000000p–4, 0x1.6356760000000p–5, 0x1.6cf0740000000p–5, –0x1.9dcca00000000p–3, –0x1.e369800000000p–4, –0x1.78bc500000000p–6, 0x1.7650c80000000p–2, –0x1.9ab98e0000000p–4, –0x1.6d33c00000000p–3, 0x1.d3bffe0000000p–2, –0x1.91dc960000000p–4,
0x1.27079a0000000p–2, 0x1.833f940000000p–2, –0x1.8b0ba60000000p–7, 0x1.08bcc80000000p–2, 0x1.2b9f560000000p–2, 0x1.65bf040000000p–5, –0x1.00c2b00000000p–3, 0x1.8051ae0000000p–9, –0x1.2138ec0000000p–3, 0x1.84b4da0000000p–3, –0x1.943db20000000p–5, 0x1.3b5aba0000000p–5, 0x1.2234840000000p–4, –0x1.0df4180000000p–3, 0x1.39fb440000000p–3, 0x1.166f3c0000000p–1,
–0x1.3a93d60000000p–6, 0x1.d181920000000p–4, –0x1.b995140000000p–8, –0x1.3d5fae0000000p–4, –0x1.bcca740000000p–4, 0x1.0cb5340000000p–5, –0x1.96a9160000000p–3, –0x1.0dc2120000000p–2, 0x1.bc2e700000000p–3, –0x1.4804b80000000p–5, –0x1.628d960000000p–5, 0x1.4c6fc60000000p–7, –0x1.babbb60000000p–4, 0x1.76a43a0000000p–4, 0x1.2ab79e0000000p–3, 0x1.866b8e0000000p–7,
0x1.4a08d80000000p–2, 0x1.37c1920000000p–2, –0x1.0ffbf40000000p–1, 0x1.be2d780000000p–1, 0x1.7e1a3a0000000p+0, –0x1.81d22e0000000p–4, –0x1.195f820000000p–2, 0x1.d980740000000p–3, –0x1.c4c8160000000p–3, –0x1.12db9a0000000p–2, –0x1.91f4a20000000p–2, –0x1.67554c0000000p–4, –0x1.6188860000000p–5, –0x1.3d8ca80000000p–3, 0x1.ec7ba00000000p–2, 0x1.1c48720000000p–3,

Table B.4 (continued)

Parameter value
0x1.1fdffc0000000p–3, 0x1.a1f22a0000000p–2, 0x1.2170cc0000000p–2, 0x1.8056000000000p–4, –0x1.7226440000000p–3, –0x1.8bc9040000000p–4, –0x1.89a9ac0000000p–8, –0x1.fe74120000000p–4, –0x1.53ea700000000p–4, 0x1.0866e20000000p–2, –0x1.8247180000000p–5, 0x1.328ab40000000p–2, 0x1.e6be520000000p–2, –0x1.233e3c0000000p–2, 0x1.72d6860000000p–2, –0x1.5f63aa0000000p+0,

Parameter value
0x1.622e380000000p–2, 0x1.6592960000000p–3, –0x1.146bc00000000p–2, 0x1.aab2a20000000p–9, 0x1.2180580000000p+0, –0x1.9ce9360000000p–3, –0x1.dd44e40000000p–3, 0x1.26b3660000000p–5, 0x1.8e204e0000000p–3, –0x1.5060de0000000p–1, 0x1.c6b1e80000000p–4, 0x1.1c8e780000000p–1, –0x1.03575c0000000p–1, 0x1.812aec0000000p–5, 0x1.ed5e5a0000000p–3, 0x1.0589b80000000p–3,
0x1.81e0a80000000p–1, 0x1.4eb3e00000000p+0, –0x1.6de4000000000p–3, 0x1.950bf20000000p–2, 0x1.1d844e0000000p+1, –0x1.a3b9c80000000p–2, –0x1.c9fb6a0000000p–4, –0x1.0c72360000000p–3, –0x1.643a060000000p–3, –0x1.5f3d340000000p+0, –0x1.ec04380000000p–3, –0x1.fb3e480000000p–1, 0x1.0951820000000p–3, –0x1.efdbdc0000000p–3, 0x1.ec84020000000p–2, –0x1.e2024c0000000p+0,
0x1.cd519e0000000p–4, 0x1.9a9ff00000000p–3, –0x1.14ba7c0000000p–2, 0x1.5bb8be0000000p–2, 0x1.f044ea0000000p–4, –0x1.34a4380000000p–6, –0x1.34cae00000000p–3, 0x1.072a100000000p–3, –0x1.d6ea840000000p–2, 0x1.be8c620000000p–2, 0x1.d09caa0000000p–10, 0x1.c6cce00000000p–3, –0x1.46a4040000000p–2, –0x1.1dcc120000000p–3, 0x1.fbab6a0000000p–3, 0x1.d03fd40000000p–3,
0x1.40ea4e0000000p–2, –0x1.84b21a0000000p–4, –0x1.7f39b00000000p+0, –0x1.5a00d60000000p–3, –0x1.7794600000000p+1, –0x1.356e280000000p–1, –0x1.e8a7780000000p–5, –0x1.01c86c0000000p–1, –0x1.9753680000000p–3, –0x1.00776a0000000p–1, –0x1.1905a40000000p–2, –0x1.1d1c720000000p–2, –0x1.9d2c800000000p–6, –0x1.1edf1a0000000p–1, –0x1.c4bc220000000p–4, 0x1.2fe0a00000000p–3,
–0x1.8b52f00000000p–4, –0x1.25f2200000000p–2, –0x1.114e6c0000000p–4, 0x1.55bfea0000000p–6, –0x1.b32e8e0000000p–3, 0x1.52456e0000000p–4, –0x1.f3cf3a0000000p–4, –0x1.2305860000000p–5, 0x1.1ae6a60000000p–6, –0x1.bdd6aa0000000p–5, –0x1.60b7500000000p–4, –0x1.64ab1e0000000p–4, –0x1.bad3020000000p–8, 0x1.ea03bc0000000p–8, –0x1.ad39900000000p–2, 0x1.cb51960000000p–3,
0x1.0bbe400000000p–1, –0x1.e077440000000p–2, –0x1.1051e60000000p–5, –0x1.62b0120000000p–7, 0x1.5d2f6e0000000p+0, –0x1.6692da0000000p–5, –0x1.7510640000000p–2, –0x1.ad2dde0000000p–9, 0x1.eb94900000000p–4, –0x1.3c7c6a0000000p–1, 0x1.f301820000000p–5, –0x1.1334c60000000p+1, 0x1.a2d5480000000p–6, 0x1.720abc0000000p–3, 0x1.3951be0000000p–3, –0x1.78ead00000000p–3,

Table B.4 (continued)

Parameter value
–0x1.b27f7c0000000p–2, –0x1.40928c0000000p+0, –0x1.0a6eca0000000p+0, –0x1.a0e5080000000p–1, –0x1.de7686000000p–1, –0x1.5394760000000p+0, –0x1.011a600000000p+0, –0x1.fdbad20000000p–1, –0x1.f09e1c0000000p–1, 0x1.01c5e00000000p–9, –0x1.44a76c0000000p–1, –0x1.1e4eb00000000p+0, –0x1.23a9940000000p+0, –0x1.f643780000000p–1, –0x1.565a0a0000000p–2, –0x1.4a7abe0000000p–3,
–0x1.91f2dc0000000p–1, 0x1.54de640000000p–3, –0x1.07805e0000000p–2, 0x1.24f8820000000p–2, 0x1.e7dcac0000000p–2, 0x1.adc7d20000000p–6, –0x1.b1da960000000p–2, –0x1.d30af40000000p–3, –0x1.e61a5c0000000p–2, –0x1.4b8ed00000000p–2, 0x1.4f860e0000000p–7, 0x1.353c860000000p–2, 0x1.5c25440000000p–3, 0x1.a8cd0a0000000p–7, 0x1.9cb8f00000000p–2, –0x1.fe02240000000p–2,
–0x1.a214a60000000p–3, –0x1.ee45da0000000p–4, –0x1.a10fb00000000p–3, –0x1.cc80f40000000p–3, 0x1.7ccba00000000p–3, –0x1.6a49e20000000p–2, –0x1.8777f80000000p–3, –0x1.0af2aa0000000p–1, –0x1.1722ae0000000p–3, –0x1.da2d080000000p–4, –0x1.3ba1c40000000p–3, –0x1.17cc4e0000000p–3, –0x1.2573cc0000000p–6, 0x1.f01a200000000p–5, 0x1.6a8e360000000p–3, –0x1.65d0a60000000p–3,
0x1.a910d60000000p–3, 0x1.f18f1c0000000p–3, –0x1.ebe1fa0000000p–5, –0x1.79800e0000000p–4, –0x1.edf78c0000000p–4, 0x1.a289dc0000000p–5, –0x1.a6a7040000000p–3, –0x1.fbba620000000p–6, 0x1.22ccfa0000000p–4, –0x1.e14f4a0000000p–1, –0x1.dff9300000000p–6, –0x1.0c412a0000000p–3, 0x1.3e0c3e0000000p–4, 0x1.475afc0000000p–5, 0x1.f84be40000000p–3, –0x1.69b4cc0000000p–2,
–0x1.8659d40000000p–5, –0x1.2341860000000p–5, 0x1.2ca4800000000p–5, –0x1.b615a00000000p–5, 0x1.00bdd20000000p–3, 0x1.402ade0000000p–4, –0x1.fac57e0000000p–4, –0x1.827e9a0000000p–3, 0x1.08bc140000000p–2, –0x1.ee6fce0000000p–5, –0x1.a9fb780000000p–6, 0x1.86f0300000000p–3, 0x1.0c32b80000000p–3, 0x1.d1e0ca0000000p–4, 0x1.0e15320000000p–2, 0x1.0319060000000p–8,
0x1.20b3940000000p–2, –0x1.27e11a0000000p–3, –0x1.20d9e60000000p–3, –0x1.472f700000000p–6, 0x1.0ba4d20000000p–2, 0x1.8b75520000000p–3, –0x1.36cc700000000p–2, –0x1.f9acc20000000p–2, 0x1.53369c0000000p–2, 0x1.31814e0000000p–2, –0x1.4e83c00000000p–3, –0x1.19d2220000000p–4, 0x1.2dbd260000000p–5, –0x1.04273a0000000p–8, 0x1.b3e6480000000p–2, 0x1.3a78780000000p–2,
0x1.8381ac0000000p–2, 0x1.7d6a240000000p–1, –0x1.cd874e0000000p–2, 0x1.9c6c3c0000000p–2, 0x1.76afec0000000p–2, –0x1.1d61ac0000000p–3, –0x1.1a0c0a0000000p–3, –0x1.a028e20000000p–2, 0x1.5d2ec60000000p–1,

Parameter value
0x1.55cfb80000000p+0, –0x1.58ca8a0000000p–4, 0x1.3e00d80000000p–1, –0x1.025e9c0000000p–3, –0x1.86862a0000000p–2, 0x1.9147140000000p–2, –0x1.553d960000000p–1,

Table B.4 (continued)

Parameter value
0x1.6407f40000000p–1, 0x1.8349140000000p–3, –0x1.db39140000000p–3, –0x1.a666820000000p–4, 0x1.f6183c0000000p–4, 0x1.2120f40000000p–7, 0x1.00a5940000000p–2, –0x1.3f14360000000p–1, –0x1.2487f00000000p–4, –0x1.15d42c0000000p–1, 0x1.644b4c0000000p–4, 0x1.d9e87a0000000p–4, –0x1.c87fe20000000p–5, 0x1.27ac580000000p–2, 0x1.3392d40000000p–1, –0x1.1cf2b60000000p–4,
–0x1.716c940000000p–4, –0x1.1e872c0000000p–5, –0x1.43157c0000000p–2, 0x1.03161c0000000p–3, 0x1.2ee1520000000p–1, 0x1.0bae500000000p–3, –0x1.646ace0000000p–4, 0x1.af71ee0000000p–6, 0x1.a577940000000p–3, 0x1.29e12a0000000p–2, –0x1.5763400000000p–9, –0x1.d380800000000p–4, –0x1.17238e0000000p–5, –0x1.c56a060000000p–4, 0x1.4364460000000p–2, 0x1.3f1c640000000p–2,
0x1.d2b5360000000p–3, –0x1.74bb600000000p–1, 0x1.9970140000000p–6, 0x1.149fec0000000p–2, 0x1.77f11e0000000p–4, 0x1.2fc1200000000p–6, –0x1.b6d6860000000p–3, 0x1.2a68120000000p–4, 0x1.7e7dde0000000p–7, –0x1.c97b560000000p–3, –0x1.f45bc20000000p–7, 0x1.68ca3c0000000p–2, –0x1.43c6480000000p–5, 0x1.0d25040000000p–4, 0x1.0f05f60000000p–6, –0x1.ce4a380000000p–3,
0x1.89fa660000000p–4, 0x1.2e1d920000000p–1, –0x1.1028fc0000000p–2, 0x1.2810360000000p–6, –0x1.29f7ca0000000p–6, –0x1.43b4ac0000000p–2, 0x1.696b680000000p–3, –0x1.668cbc0000000p–3, 0x1.8cf46c0000000p–2, –0x1.df75ae0000000p–3, 0x1.24318c0000000p–4, –0x1.4124b80000000p–2, –0x1.8091a00000000p+0, –0x1.ae4cea0000000p–6, –0x1.4b183e0000000p–2, 0x1.10c0cc0000000p–1,
–0x1.81c6c40000000p–4, 0x1.9ca85c0000000p–5, –0x1.d1bf6a0000000p–2, –0x1.f5b52e0000000p–4, –0x1.01d3ea0000000p–3, –0x1.9e98a00000000p+0, –0x1.7e3b8a0000000p+0, –0x1.5604780000000p+1, –0x1.b641ec0000000p–1, 0x1.3ededa0000000p–3, –0x1.39d6720000000p–1, –0x1.a3f40a0000000p–4, –0x1.a58ed80000000p–2, –0x1.fc9d420000000p–2, 0x1.712f6e0000000p–5, –0x1.2460b00000000p–1,
–0x1.2222be0000000p–3, 0x1.b16d520000000p–2, –0x1.ef074e0000000p–5, 0x1.68b4160000000p–3, 0x1.d32ba80000000p–1, 0x1.d988ae0000000p–5, 0x1.f532220000000p–6, –0x1.bc20c80000000p–2, –0x1.568c240000000p–2,

Parameter value
–0x1.7b5fd40000000p–1, 0x1.cbe5560000000p–4, 0x1.afdb880000000p–1,
–0x1.3c2d6a0000000p–2, –0x1.876a120000000p–5, –0x1.091e780000000p–4, 0x1.15b0d40000000p–3,

Table B.5 Layer 2 CNN (bias parameter bias) of context decoding neural network parameter

Parameter value
–0x1.8651700000000p+0, 0x1.3d20080000000p–7, –0x1.2adf2a0000000p–6, –0x1.bd9b740000000p+0, –0x1.38cc000000000p–3, –0x1.1858e00000000p–4, 0x1.4141d00000000p–6, 0x1.0ea0580000000p–10, –0x1.0c9ba60000000p–4, –0x1.acd53e0000000p–1, –0x1.8695fc0000000p–2, –0x1.63eb6e0000000p+0, –0x1.6c770e0000000p–5, –0x1.b741960000000p–4, –0x1.b0056c0000000p–1, –0x1.5847640000000p–2,

Table B.6 Layer 3 CNN (convolution kernel parameter kernel) of context decoding neural network parameter

Parameter value
0x1.c85c500000000p–3, 0x1.a9ff920000000p–3, –0x1.216db20000000p–1, 0x1.0a1a2e0000000p–1, –0x1.9a86340000000p–1, 0x1.c22e7a0000000p–3, 0x1.ab533c0000000p–2, –0x1.fcfa9e0000000p–3, 0x1.2e4f300000000p–2, 0x1.dc9c1e0000000p–1, 0x1.9dafe80000000p–5, 0x1.354ae20000000p–2, –0x1.4881b80000000p–6, –0x1.54d7680000000p–2, 0x1.0dee8a0000000p–2, 0x1.ee0d4e0000000p–4,
0x1.79b3720000000p–2, –0x1.a945240000000p–10, –0x1.bb59e20000000p–2, 0x1.8f1b1c0000000p–1, –0x1.6894b40000000p+0, –0x1.97e0f40000000p–2, 0x1.7b5d2e0000000p–3, –0x1.c32bce0000000p–3, 0x1.039b1a0000000p–3, 0x1.78a7d00000000p–3, 0x1.0bc78c0000000p–4, 0x1.b196580000000p–5, 0x1.5cfa620000000p–2, –0x1.978bec0000000p–5, –0x1.474f400000000p–2, 0x1.9751880000000p–5,
0x1.0079e60000000p–2, –0x1.c738de0000000p–7, –0x1.e3e9780000000p–2, 0x1.b864800000000p–1, –0x1.71b1720000000p+0, –0x1.7773b60000000p–1, 0x1.16a8ec0000000p–3, –0x1.979a5c0000000p–3, 0x1.f2f1080000000p–4, 0x1.1b80c60000000p–3, 0x1.23e3de0000000p–4, 0x1.8f42fc0000000p–5, 0x1.bf29e40000000p–3, –0x1.906a9a0000000p–6, –0x1.7cffc60000000p–3, 0x1.4ce4800000000p–6,
0x1.8c49640000000p–2, –0x1.46ca0c0000000p–8, –0x1.294bd40000000p–1, 0x1.cab6860000000p–2, –0x1.5e67b80000000p+0, 0x1.6e8cae0000000p–1, 0x1.fdb9180000000p–3, –0x1.06736e0000000p–3, 0x1.1e4a1c0000000p–3,

Parameter value
0x1.1668480000000p–2, 0x1.0631500000000p–8, 0x1.239dc80000000p–5, 0x1.ebe5be0000000p–2, –0x1.3a8afe0000000p–4, –0x1.0eec320000000p–1, 0x1.022a800000000p–4,
0x1.43ac7e0000000p–1, –0x1.087bc20000000p–8, –0x1.99d48c0000000p–2, –0x1.3dc5e60000000p–1, –0x1.5bd7020000000p+0, –0x1.40ee280000000p–1, 0x1.10507e0000000p–4, –0x1.58656c0000000p–2, 0x1.ab96b00000000p–4, 0x1.b82f4a0000000p–5, –0x1.c844fa0000000p–3, 0x1.07f8880000000p–5, 0x1.a3265c0000000p–2, 0x1.fda6680000000p–7, 0x1.aecec80000000p+0, 0x1.2b29dc0000000p–4,

Table B.6 (continued)

Parameter value
0x1.38274c0000000p–2, 0x1.990c100000000p–3, –0x1.01aca00000000p–1, 0x1.2c413e0000000p–1, –0x1.08fcee0000000p+0, 0x1.83d56c0000000p–3, 0x1.8f74d00000000p–2, –0x1.4a96f80000000p–2, 0x1.21b1d80000000p–2, 0x1.97360a0000000p–1, 0x1.5314c40000000p–1, 0x1.a056500000000p–3, 0x1.9a7dde0000000p–4, –0x1.5c84b80000000p–2, 0x1.bd72420000000p–3, 0x1.c1a8b40000000p–3,
0x1.c15eda0000000p–2, 0x1.cc72f20000000p–6, –0x1.4a0d460000000p–1, 0x1.1f65660000000p–1, –0x1.73f7a60000000p+0, 0x1.3f20f00000000p–1, 0x1.23fa800000000p–2, –0x1.0c8a3a0000000p–3, 0x1.61e0b80000000p–3, 0x1.ecf7420000000p–2, 0x1.4ce1ea0000000p–5, 0x1.8269060000000p–5, 0x1.d804c60000000p–2, –0x1.14ce240000000p–3, –0x1.2ccd840000000p–1, 0x1.cf523e0000000p–4,
0x1.dc1eda0000000p–2, –0x1.241edc0000000p–6, –0x1.83de640000000p–3, –0x1.0ba9060000000p–1, –0x1.2ddebe0000000p+0, 0x1.44defe0000000p–3, 0x1.1948740000000p–4, –0x1.7869360000000p–2, 0x1.85928e0000000p–4, 0x1.08bf500000000p–5, –0x1.05b6e00000000p–3, 0x1.f2c1f80000000p–7, 0x1.f3d3060000000p–5, 0x1.8fdb520000000p–6, –0x1.48723a0000000p–3, 0x1.fd51f40000000p–5,
0x1.1c16a20000000p–2, 0x1.17ac1e0000000p–3, 0x1.13018e0000000p–3, 0x1.ca7ee60000000p–1, –0x1.226e680000000p+0, –0x1.22d4ae0000000p–1, 0x1.8325bc0000000p–2, –0x1.b0f62e0000000p–3, 0x1.0a47120000000p–2, 0x1.2c199a0000000p–1, 0x1.462ca00000000p–1, 0x1.9adb1a0000000p–4, 0x1.208b020000000p–2, –0x1.3e7c960000000p–2, 0x1.52ede20000000p–4, 0x1.f0d93a0000000p–3,
0x1.3977960000000p–1, 0x1.a649ba0000000p–2, –0x1.29258a0000000p–2,

Parameter value
0x1.10770e0000000p–1, –0x1.5ff2f80000000p–1, –0x1.f4b3580000000p–7, 0x1.9e3eea0000000p–2, –0x1.50dfe00000000p–1, 0x1.75135e0000000p–2, 0x1.8691d60000000p–1, 0x1.a8f73a0000000p–2, 0x1.965eae0000000p–2, –0x1.86e9520000000p–3, –0x1.ebc0980000000p–3, 0x1.a3426c0000000p–2, 0x1.84c5740000000p–6,
0x1.8d17e20000000p–2, 0x1.19ada00000000p–4, –0x1.777ddc0000000p–2, 0x1.afd8d40000000p–1, –0x1.55c33e0000000p+0, –0x1.e625d20000000p–2, 0x1.49b9e40000000p–2, –0x1.bf08a40000000p–4, 0x1.b796ce0000000p–3, 0x1.2a62b20000000p–1, 0x1.9178cc0000000p–3, 0x1.126dd80000000p–4, 0x1.d336340000000p–2, –0x1.c5a2e80000000p–3, –0x1.ee30d20000000p–3, 0x1.85cbd80000000p–3,
0x1.00e2a00000000p–1, –0x1.95745e0000000p–9, –0x1.4e2eec0000000p–1, –0x1.d76ef00000000p–4, –0x1.71dd900000000p+0, –0x1.44d2960000000p–2, 0x1.67309c0000000p–4, –0x1.b39e3c0000000p–3, 0x1.9d56400000000p–4, 0x1.08e6620000000p–3, –0x1.7c99360000000p–4, 0x1.6821cc0000000p–7, 0x1.fe93380000000p–2, 0x1.17e13e0000000p–7, 0x1.fca2a20000000p+0, 0x1.98d80e0000000p–7,

Table B.6 (continued)

Parameter value
0x1.31e2540000000p–2, –0x1.1f23980000000p–5, –0x1.71bb140000000p–3, –0x1.4af0fa0000000p–2, –0x1.f5be620000000p–2, –0x1.f66a1e0000000p–3, 0x1.6aa53c0000000p–4, –0x1.6c4f700000000p–3, 0x1.e997c40000000p–5, 0x1.2f5dac0000000p–3, –0x1.3bb98e0000000p–6, –0x1.3fe01c0000000p–7, –0x1.8f07900000000p–4, 0x1.be47da0000000p–6, –0x1.cac9840000000p–2, –0x1.2a5d580000000p–6,
0x1.0fad9c0000000p–1, 0x1.619b0e0000000p–2, –0x1.5c4ff20000000p–3, 0x1.5b1f380000000p–1, –0x1.b226180000000p–2, –0x1.731eca0000000p–1, 0x1.c4cdd20000000p–2, –0x1.33e28c0000000p–1, 0x1.5cb7420000000p–2, 0x1.bfd2f40000000p–1, 0x1.25b3be0000000p–4, 0x1.8333060000000p–2, –0x1.9fc0080000000p–3, –0x1.131c580000000p–2, 0x1.7bdd3e0000000p–1, 0x1.30ee200000000p–4,
0x1.8f97700000000p–2, –0x1.1b22e60000000p–8, –0x1.6ba58a0000000p–1, 0x1.5151a20000000p–2, –0x1.4f4ebc0000000p+0, 0x1.3e472e0000000p–4, 0x1.f7fe980000000p–4, –0x1.1eeeba0000000p–2, 0x1.c154380000000p–4, 0x1.8249740000000p–4, 0x1.3733400000000p–4, 0x1.6547340000000p–6, 0x1.5364600000000p–2, –0x1.7192160000000p–7, 0x1.9874f60000000p–1, –0x1.6857a00000000p–7,

Parameter value
0x1.ae8d4e0000000p–2, –0x1.88e5360000000p–7, –0x1.ea85440000000p–2, –0x1.9928840000000p–3, –0x1.44527e0000000p+0, 0x1.1efc280000000p–1, 0x1.2cbe680000000p–3, –0x1.e731ec0000000p–3, 0x1.a4cce00000000p–4, 0x1.c6a9480000000p–6, –0x1.e3495e0000000p–6, 0x1.61dd7c0000000p–10, 0x1.edb8c40000000p–2, –0x1.174c7c0000000p–7, 0x1.872ed00000000p+0, –0x1.451bb60000000p–8,
0x1.508a480000000p–2, 0x1.b1236e0000000p–3, 0x1.a700e60000000p–3, 0x1.165c620000000p–4, 0x1.5ba9f00000000p–2, –0x1.5a6dc20000000p–2, 0x1.4ba5a00000000p–7, –0x1.f2aa4c0000000p–1, 0x1.3b88f00000000p–2, 0x1.35b4980000000p–4, 0x1.29e4d80000000p–1, 0x1.4a2e080000000p–2, –0x1.53f25c0000000p–2, 0x1.e6a24c0000000p–3, –0x1.988f480000000p–2, 0x1.685e8e0000000p–3,
0x1.7029860000000p–1, 0x1.e0eb060000000p–2, –0x1.f3374a0000000p–3, 0x1.21ee340000000p+0, 0x1.e2755c0000000p–5, –0x1.7d52f80000000p–3, 0x1.4fe81a0000000p–3, –0x1.6dfb980000000p–1, 0x1.72f0c60000000p–2, 0x1.02bc800000000p–2, 0x1.77b5880000000p–2, 0x1.9ac3e40000000p–2, –0x1.6f09b20000000p–1, 0x1.1ba3540000000p–2, 0x1.8a387c0000000p–3, 0x1.ae923a0000000p–4,
0x1.7ac2d20000000p–1, 0x1.f9f19e0000000p–2, –0x1.0f844c0000000p–3, 0x1.2e1d4c0000000p+0, –0x1.d1390a0000000p–4, 0x1.7cc4040000000p–4, 0x1.6d96a60000000p–3, –0x1.64aa2c0000000p–1, 0x1.5c9ddc0000000p–2, 0x1.89801a0000000p–2, 0x1.43e9400000000p–2, 0x1.8c422e0000000p–2, –0x1.3d182e0000000p–1, 0x1.00a42e0000000p–2, 0x1.fa50940000000p–3, 0x1.5957280000000p–4,

Table B.6 (continued)

Parameter value
0x1.3c8b340000000p–1, 0x1.9950620000000p–2, 0x1.acd3620000000p–2, 0x1.2480d80000000p+0, 0x1.80d8220000000p–4, –0x1.7a4c8c0000000p–1, 0x1.66a5b00000000p–3, –0x1.41f20a0000000p–1, 0x1.6b24d80000000p–2, 0x1.33d28c0000000p–2, 0x1.2cbfd80000000p–4, 0x1.96e4d00000000p–2, –0x1.66cbd40000000p–1, 0x1.488fca0000000p–3, 0x1.81b3440000000p–3, 0x1.c080ce0000000p–5,
0x1.aa29020000000p–2, 0x1.0f06f60000000p–1, 0x1.074a720000000p–3, 0x1.ec52660000000p+0, –0x1.5c2c680000000p–2, 0x1.67e1ee0000000p–1, 0x1.ff135e0000000p–3, –0x1.e0b4040000000p–2, 0x1.6699ca0000000p–2, 0x1.5ee4260000000p–1, 0x1.50a9de0000000p–3, 0x1.a9e7e80000000p–2, –0x1.5dce7e0000000p–2, 0x1.1916bc0000000p–3, 0x1.ad58520000000p–1, 0x1.16b7400000000p–4,

Parameter value
0x1.b4c07e0000000p–3, 0x1.bc16dc0000000p–3, –0x1.4f790a0000000p–2, 0x1.3f6dc40000000p–2, 0x1.84ced80000000p–2, 0x1.d0b8980000000p–9, 0x1.286e720000000p–5, –0x1.d7dc0e0000000p–1, 0x1.34d92e0000000p–2, 0x1.731e780000000p–4, 0x1.9f49140000000p–2, 0x1.51fdbc0000000p–2, –0x1.93a95e0000000p–2, 0x1.5f13fc0000000p–3, –0x1.36c36a0000000p–2, 0x1.0c242c0000000p–3,
0x1.05cb4e0000000p–1, 0x1.4bf3180000000p–2, 0x1.b554100000000p–2, 0x1.15544a0000000p+0, 0x1.5fa0d00000000p–3, –0x1.872fc80000000p–1, 0x1.8557e80000000p–3, –0x1.d2c9560000000p–1, 0x1.65a0d20000000p–2, 0x1.9246740000000p–3, –0x1.a43c820000000p–3, 0x1.972ea60000000p–2, –0x1.9ed2b80000000p–1, 0x1.a6977c0000000p–3, –0x1.e2c05c0000000p–6, 0x1.f5a8fe0000000p–6,
0x1.be94fe0000000p–3, 0x1.09e0300000000p–1, –0x1.67b1ce0000000p–2, 0x1.780e680000000p+0, –0x1.8e78700000000p–1, 0x1.fff9180000000p–1, 0x1.1380980000000p–2, –0x1.209aec0000000p–1, 0x1.6107280000000p–2, 0x1.6325600000000p–1, 0x1.c239000000000p–2, 0x1.aa76dc0000000p–2, –0x1.17921a0000000p–2, –0x1.83956c0000000p–4, 0x1.5514720000000p–1, 0x1.2d8fa00000000p–4,
0x1.6007520000000p–2, 0x1.c11bee0000000p–3, –0x1.7d04880000000p–3, 0x1.6614dc0000000p–2, 0x1.7c07820000000p–2, 0x1.fb585e0000000p–4, 0x1.4c16c60000000p–4, –0x1.9597f40000000p–1, 0x1.430b0c0000000p–2, 0x1.622a300000000p–3, 0x1.81ed560000000p–2, 0x1.806aa80000000p–2, –0x1.bd05560000000p–2, 0x1.371e440000000p–3, –0x1.c715f00000000p–3, 0x1.ec84580000000p–5,
0x1.18b8aa0000000p–6, 0x1.b72e460000000p–4, 0x1.a888fc0000000p–4, 0x1.7a0fd20000000p–6, 0x1.38f6900000000p–4, 0x1.aaa0ec0000000p–4, –0x1.d8da000000000p–8, –0x1.a8bbc00000000p–1, 0x1.216c660000000p–2, 0x1.8855aa0000000p–3, –0x1.0436d80000000p–5, 0x1.7f1fb20000000p–2, –0x1.f674800000000p–4, 0x1.5408320000000p–3, –0x1.411e260000000p–2, 0x1.e118ca0000000p–5,

Table B.6 (continued)

Parameter value
0x1.a0fd580000000p–2, 0x1.1782fe0000000p–2, –0x1.4305260000000p–2, 0x1.800b240000000p–1, 0x1.70a2c00000000p–2, –0x1.558cb20000000p–2, 0x1.3392a60000000p–3, –0x1.da101e0000000p–1, 0x1.5a586e0000000p–2, 0x1.ad16760000000p–3, 0x1.80e4c00000000p–6, 0x1.92223c0000000p–2, –0x1.45ebd80000000p–1, 0x1.7035c40000000p–3, –0x1.2ea34c0000000p–3, 0x1.dac89a0000000p–6,

Parameter value
0x1.12cb440000000p–1, 0x1.130e4e0000000p–1, –0x1.40d6140000000p–7, 0x1.cb5bfe0000000p+0, 0x1.da584c0000000p–4, –0x1.9a421a0000000p–2, 0x1.d21efa0000000p–3, –0x1.5b7efa0000000p–1, 0x1.715e340000000p–2, 0x1.1a93600000000p–1, –0x1.2aa1ec0000000p–2, 0x1.a580de0000000p–2, –0x1.0061960000000p–1, 0x1.d67d6a0000000p–3, 0x1.2837d60000000p–1, 0x1.1490700000000p–4,
0x1.db9dc60000000p–4, 0x1.67d8fc0000000p–2, 0x1.78dc400000000p–2, 0x1.bfb1620000000p–3, –0x1.4df09e0000000p–1, –0x1.1891280000000p–1, 0x1.8668100000000p–2, –0x1.06cb8e0000000p–1, 0x1.3cbba80000000p–2, 0x1.cd35fa0000000p–1, –0x1.1cf8b00000000p–2, 0x1.a3ace60000000p–2, –0x1.f600f60000000p–3, –0x1.b85af00000000p–2, 0x1.a65e980000000p–2, 0x1.406dce0000000p–4,
0x1.4999320000000p–4, 0x1.7361a20000000p–3, –0x1.87c3c60000000p–3, –0x1.d707e80000000p–5, 0x1.0326e20000000p–1, 0x1.4243a60000000p–6, –0x1.065a420000000p–4, –0x1.04f4b00000000p+0, 0x1.3e2d040000000p–2, 0x1.0ec9c60000000p–4, –0x1.3c99920000000p–3, 0x1.8c2c240000000p–2, –0x1.6be5ec0000000p–3, 0x1.03f34c0000000p–2, –0x1.82110e0000000p–2, 0x1.3c8efc0000000p–4,
0x1.37a7ac0000000p–1, 0x1.05d0260000000p–1, –0x1.359be60000000p–3, 0x1.558e020000000p+0, 0x1.bf147c0000000p–5, –0x1.73deee0000000p–3, 0x1.35ae700000000p–3, –0x1.5614260000000p–1, 0x1.60a1520000000p–2, 0x1.9745b60000000p–2, –0x1.0d18e60000000p–5, 0x1.8ae6e00000000p–2, –0x1.1181680000000p–1, 0x1.02ac9c0000000p–2, 0x1.4020880000000p–2, 0x1.09df7e0000000p–4,
0x1.2a70d80000000p–1, 0x1.fd3da80000000p–2, 0x1.c3c37e0000000p–2, 0x1.6d9adc0000000p+0, 0x1.44c6160000000p–3, –0x1.2628500000000p–1, 0x1.89a6680000000p–3, –0x1.fe37ac0000000p–2, 0x1.7288400000000p–2, 0x1.0f6abc0000000p–1, –0x1.cf19300000000p–4, 0x1.923f4c0000000p–2, –0x1.e5725e0000000p–2, 0x1.88fbde0000000p–3, 0x1.98d5020000000p–2, 0x1.15db700000000p–4,
0x1.c414260000000p–8, 0x1.3df1b60000000p–6, 0x1.990bb20000000p–2, –0x1.ef4a3a0000000p–4, –0x1.be932e0000000p–2, 0x1.7331f20000000p–3, 0x1.37e11c0000000p–10, –0x1.7850660000000p–2, 0x1.bac3540000000p–4, 0x1.0b423c0000000p–3, –0x1.cf6eb40000000p–6, 0x1.5451fa0000000p–5, –0x1.340b340000000p–5, 0x1.c101980000000p–7, –0x1.29d21e0000000p–3, 0x1.6641a80000000p–4,

Table B.6 (continued)

Parameter value
0x1.8aa5c20000000p–6, 0x1.271a140000000p–3, 0x1.468c440000000p–7, –0x1.29fab60000000p–2, –0x1.393fee0000000p–3, 0x1.c0ba920000000p–4, –0x1.8a51160000000p–4, –0x1.5d15260000000p–1, 0x1.08a79a0000000p–2, 0x1.71e9bc0000000p–3, –0x1.1e03960000000p–2, 0x1.5da1520000000p–2, –0x1.e3e1ba0000000p–4, 0x1.c3630a0000000p–3, –0x1.a8fdc20000000p–2, 0x1.6ad2b60000000p–4,
0x1.cfbb480000000p–5, 0x1.0047aa0000000p–3, 0x1.3f45c20000000p–2, –0x1.5191d80000000p–2, –0x1.2012500000000p–2, 0x1.1b42d00000000p–4, –0x1.0cb8980000000p–4, –0x1.86ebfa0000000p–1, 0x1.0ea05e0000000p–2, 0x1.8d7de00000000p–3, –0x1.07cc0a0000000p–2, 0x1.72c8f20000000p–2, –0x1.8d15420000000p–4, 0x1.a4325e0000000p–3, –0x1.c2dbb40000000p–2, 0x1.0fa3ce0000000p–4,
0x1.ef6c080000000p–4, 0x1.f8411c0000000p–4, –0x1.ae1c6a0000000p–2, –0x1.0f15600000000p–2, –0x1.e54e100000000p–3, –0x1.8aa93a0000000p–5, –0x1.12b1520000000p–4, –0x1.64693c0000000p–1, 0x1.e0831e0000000p–3, 0x1.9461c40000000p–3, –0x1.ecc9f00000000p–5, 0x1.3ad9ea0000000p–2, –0x1.4e7e6e0000000p–4, 0x1.b8721c0000000p–3, –0x1.4fd60c0000000p–2, 0x1.9a7bae0000000p–4,
–0x1.1e621e0000000p–4, 0x1.1cc2200000000p–4, –0x1.0823920000000p–1, –0x1.af31c80000000p–3, –0x1.31de300000000p–2, 0x1.c66ff00000000p–6, 0x1.26837a0000000p–7, –0x1.aeb1ec0000000p–1, 0x1.019a7e0000000p–2, 0x1.58eaaa0000000p–6, 0x1.131b940000000p–2, 0x1.50118a0000000p–2, –0x1.3b0ea20000000p–3, 0x1.c8b7880000000p–3, –0x1.46e72e0000000p–2, 0x1.344e580000000p–4,
0x1.c6484e0000000p–7, 0x1.b0d7380000000p–9, –0x1.5ea8ce0000000p–5, –0x1.6d1c620000000p–3, –0x1.8a60ec0000000p–2, 0x1.1b96ca0000000p–3, 0x1.a1c1ba0000000p–8, –0x1.5da19e0000000p–2, 0x1.0c88540000000p–3, 0x1.7b1a2a0000000p–4, 0x1.c662d60000000p–3, 0x1.0a8f140000000p–5, –0x1.cd8c0a0000000p–6, 0x1.dd00fa0000000p–6, –0x1.952ebe0000000p–4, 0x1.13a1e20000000p–3,
0x1.04daf20000000p–3, 0x1.0787a00000000p–4, –0x1.a827920000000p–3, –0x1.4b85cc0000000p–2, –0x1.4e0cde0000000p–2, –0x1.6332220000000p–4, –0x1.0762140000000p–5, –0x1.3528d40000000p–1, 0x1.9c05680000000p–3, 0x1.210e280000000p–3, 0x1.86aca20000000p–2, 0x1.ee49120000000p–3, –0x1.0875ce0000000p–5, 0x1.b851480000000p–3, –0x1.aa11bc0000000p–3, 0x1.ddd3500000000p–4,
–0x1.c667540000000p–5, 0x1.c5a8700000000p–5, –0x1.eeb8bc0000000p–3, –0x1.1fc4e20000000p–4, –0x1.4918560000000p–3, 0x1.fd83860000000p–3, 0x1.acd87e0000000p–5, –0x1.8b25b00000000p–1, 0x1.002e620000000p–2,

Parameter value
0x1.a15e640000000p–4, 0x1.9c2ffe0000000p–6, 0x1.74ab5a0000000p–2, –0x1.a084180000000p–3, 0x1.7ca24e0000000p–3, –0x1.cfd3160000000p–2, 0x1.786b1a0000000p–5,

Table B.6 (continued)

Parameter value
0x1.e7ea560000000p–5, –0x1.83f7240000000p–7, –0x1.b042ba0000000p–2, –0x1.1331e20000000p–2, –0x1.c671440000000p–2, –0x1.1cbc6a0000000p–5, 0x1.99ed020000000p–9, –0x1.1b183a0000000p–2, 0x1.4865960000000p–3, 0x1.9be30a0000000p–4, 0x1.d94b700000000p–2, 0x1.0a45040000000p–4, –0x1.2862200000000p–4, 0x1.3f0b8e0000000p–4, –0x1.ef22320000000p–6, 0x1.8e9c8e0000000p–3,
0x1.3871780000000p–5, –0x1.c192480000000p–8, –0x1.8f6fa20000000p–4, –0x1.abd61a0000000p–4, –0x1.f10d300000000p–3, –0x1.e6887a0000000p–4, 0x1.6c66f80000000p–6, –0x1.02b93e0000000p–2, 0x1.2f1e1a0000000p–4, 0x1.2189a40000000p–4, 0x1.a13b9e0000000p–3, 0x1.1213240000000p–7, 0x1.58261c0000000p–7, –0x1.3b53d20000000p–6, –0x1.713f1e0000000p–5, 0x1.0ab6bc0000000p–6,
0x1.410b900000000p–4, –0x1.0170660000000p–8, –0x1.35d91a0000000p–2, –0x1.6cd5800000000p–2, –0x1.d474b80000000p–2, –0x1.8c2b140000000p–5, –0x1.3b2f020000000p–6, –0x1.e898d20000000p–2, 0x1.878b080000000p–3, 0x1.d472c20000000p–5, 0x1.681cca0000000p–1, 0x1.dddbbc0000000p–4, –0x1.67be7e0000000p–5, 0x1.324aca0000000p–3, –0x1.17c5120000000p–4, 0x1.9b021c0000000p–3,
0x1.15ebd00000000p–4, 0x1.7655400000000p–4, –0x1.ad45da0000000p–3, –0x1.196c620000000p–2, –0x1.bfeaa20000000p–2, –0x1.9f7c440000000p–4, –0x1.83b1180000000p–6, –0x1.baf1800000000p–1, 0x1.08b7cc0000000p–2, 0x1.dcfb040000000p–6, 0x1.acac080000000p–2, 0x1.5c43420000000p–2, –0x1.efc7e60000000p–4, 0x1.11859c0000000p–2, –0x1.813ebc0000000p–2, 0x1.d385d60000000p–4,
0x1.f1375e0000000p–3, 0x1.54fa780000000p–3, 0x1.32194c0000000p–3, –0x1.af0da80000000p–7, 0x1.c4b5860000000p–3, –0x1.56d6b80000000p–4, 0x1.2892b60000000p–5, –0x1.f685160000000p–1, 0x1.2441820000000p–2, 0x1.c4f3b80000000p–4, 0x1.63c2900000000p–2, 0x1.6dc7d80000000p–2, –0x1.2f4cac0000000p–2, 0x1.1dfa940000000p–2, –0x1.3f4a860000000p–1, 0x1.14d0300000000p–3,
–0x1.c625660000000p–6, 0x1.7f73120000000p–7, –0x1.8594c40000000p–3, –0x1.8f6a280000000p–5, –0x1.7be0b00000000p–3, 0x1.cc34380000000p–9, 0x1.e9d1ac0000000p–7, –0x1.2d674c0000000p–3, 0x1.757c4e0000000p–5,

Parameter value
0x1.f5b5280000000p–6, 0x1.8de20a0000000p–4, –0x1.b369a20000000p–8, –0x1.79cf9c0000000p–6, 0x1.3603be0000000p–6, 0x1.9ab88e0000000p–7, 0x1.a898900000000p–5,
0x1.8e0c700000000p–4, 0x1.3ce9980000000p–3, –0x1.335c940000000p–2, –0x1.5840900000000p–3, –0x1.f401200000000p–3, 0x1.5416120000000p–5, –0x1.8bc9460000000p–5, –0x1.8fdee80000000p–1, 0x1.0e92500000000p–2, 0x1.114b360000000p–3, –0x1.b339780000000p–4, 0x1.6e64b80000000p–2, –0x1.09320e0000000p–3, 0x1.123e160000000p–2, –0x1.d53e240000000p–2, 0x1.08569c0000000p–4,

Table B.6 (continued)

Parameter value
0x1.46007a0000000p–3, 0x1.e455460000000p–4, –0x1.bfde1c0000000p–2, –0x1.a388000000000p–3, –0x1.0ef9c60000000p–2, –0x1.3f9b920000000p–3, –0x1.a9e4d80000000p–6, –0x1.99d9ce0000000p–1, 0x1.0d9c780000000p–2, 0x1.c9be5e0000000p–4, 0x1.4c55340000000p–3, 0x1.638db00000000p–2, –0x1.0dd39a0000000p–3, 0x1.09975a0000000p–2, –0x1.ceb3e00000000p–2, 0x1.d2f1ec0000000p–4,

Table B.7 Layer 3 CNN (bias parameter bias) of context decoding neural network parameter

Parameter value
0x1.c296300000000p–4, 0x1.c2940c0000000p–4, 0x1.c2930c0000000p–4, 0x1.c28c660000000p–4, 0x1.c28fb60000000p–4, 0x1.c287860000000p–4, 0x1.c27ef40000000p–4, 0x1.c2965a0000000p–4, 0x1.c285ec0000000p–4, 0x1.c286260000000p–4, 0x1.c297720000000p–4, 0x1.c292580000000p–4, 0x1.c27c780000000p–4, 0x1.c298400000000p–4, 0x1.c28c100000000p–4, 0x1.c2999a0000000p–4,

Table B.8 Standard deviation corresponding to base range coding

Parameter value
0x1.c28f5c0000000p–4, 0x1.fd8f280000000p–4, 0x1.20245e0000000p–3, 0x1.45df8c0000000p–3, 0x1.708b900000000p–3, 0x1.a0ce0a0000000p–3, 0x1.d762480000000p–3, 0x1.0a8e120000000p–2, 0x1.2d759a0000000p–2, 0x1.54ef340000000p–2, 0x1.81941a0000000p–2, 0x1.b411940000000p–2, 0x1.ed2b960000000p–2, 0x1.16dfe40000000p–1, 0x1.3b64660000000p–1,

Parameter value
0x1.64b1100000000p–1,
0x1.93662e0000000p–1, 0x1.c8390a0000000p–1, 0x1.01fb560000000p+0, 0x1.23c3780000000p+0,
0x1.49f80c0000000p+0, 0x1.752d5c0000000p+0, 0x1.a60b1c0000000p+0, 0x1.dd4ef60000000p+0,
0x1.0de7b60000000p+1, 0x1.313f8c0000000p+1, 0x1.59382a0000000p+1, 0x1.866cb40000000p+1,
0x1.b98ca00000000p+1, 0x1.f35e5e0000000p+1, 0x1.1a612c0000000p+2, 0x1.3f5b2c0000000p+2,
0x1.692cb80000000p+2, 0x1.98781e0000000p+2, 0x1.cdf4f00000000p+2, 0x1.0539640000000p+3,
0x1.276e380000000p+3, 0x1.4e1dba0000000p+3, 0x1.79de0e0000000p+3, 0x1.ab59080000000p+3,
0x1.e34eb40000000p+3, 0x1.114c220000000p+4, 0x1.3515b00000000p+4, 0x1.5d8ee80000000p+4,
0x1.8b54e60000000p+4, 0x1.bf19500000000p+4, 0x1.f9a5160000000p+4, 0x1.1dedba0000000p+5,
0x1.435eb40000000p+5, 0x1.6db6ce0000000p+5, 0x1.9d9a5e0000000p+5, 0x1.d3c3480000000p+5,
0x1.0881e20000000p+6, 0x1.2b24c40000000p+6, 0x1.5250be0000000p+6, 0x1.7e9dd80000000p+6,
0x1.b0b8060000000p+6, 0x1.e961c00000000p+6, 0x1.14bb780000000p+7, 0x1.38f82a0000000p+7,
0x1.61f39e0000000p+7, 0x1.904ce00000000p+7, 0x1.c4b7dc0000000p+7, 0x1.0000000000000p+8,

Table B.9 Base range coding

Index	Probability
1	0, 1, 65534, 65535, 65536
2	0, 2, 65533, 65535, 65536
3	0, 12, 65523, 65535, 65536
4	0, 55, 65480, 65535, 65536
5	0, 179, 65356, 65535, 65536
6	0, 459, 65076, 65535, 65536
7	0, 977, 64558, 65535, 65536
8	0, 1794, 63741, 65535, 65536
9	0, 2930, 62605, 65535, 65536
10	0, 4363, 61172, 65535, 65536

Index	Probability
11	0, 2, 6036, 59499, 65533, 65535, 65536
12	0, 14, 7875, 57659, 65521, 65535, 65536
13	0, 60, 9803, 55732, 65475, 65535, 65536
14	0, 193, 11751, 53783, 65342, 65535, 65536
15	0, 486, 13662, 51872, 65047, 65533, 65536
16	0, 11, 1026, 15497, 50038, 64509, 65524, 65535, 65536
17	0, 49, 1865, 17225, 48310, 63670, 65486, 65535, 65536
18	0, 162, 3022, 18829, 46701, 62508, 65368, 65530, 65536
19	0, 413, 4460, 20292, 45210, 61042, 65089, 65502, 65536
20	0, 67, 923, 6162, 21653, 43879, 59370, 64608, 65464, 65531, 65536
21	0, 201, 1702, 7997, 22859, 42645, 57507, 63802, 65303, 65504, 65536
22	0, 61, 530, 2824, 9939, 23967, 41557, 55585, 62700, 64994, 65464, 65525, 65536
23	0, 180, 1078, 4213, 11864, 24931, 40550, 53617, 61268, 64402, 65300, 65480, 65536
24	0, 88, 502, 1967, 5882, 13783, 25824, 39680, 51721, 59622, 63537, 65002, 65416, 65504, 65536
25	0, 55, 286, 1064, 3165, 7714, 15614, 26614, 38897, 49897, 57797, 62346, 64447, 65225, 65456 65511, 65536
26	0, 156, 637, 1884, 4605, 9595, 17291, 27272, 38156, 48136, 55832, 60822, 63543, 64790, 65271 65427, 65536
27	0, 124, 469, 1304, 3067, 6316, 11545, 18890, 27895, 37534, 46540, 53885, 59114, 62363, 64126 64961, 65306, 65430, 65536
28	0, 114, 396, 1024, 2278, 4531, 8170, 13454, 20350, 28441, 36974, 45065, 51961, 57245, 60884 63137, 64391, 65019, 65301, 65415, 65536
29	0, 116, 373, 896, 1874, 3556, 6217, 10091, 15279, 21671, 28915, 36468, 43712, 50104, 55292 59166, 61827, 63509, 64487, 65010, 65267, 65383, 65536
30	0, 60, 188, 443, 917, 1743, 3091, 5153, 8106, 12068, 17049, 22914, 29382, 36065, 42533, 48398 53379, 57341, 60294, 62356, 63704, 64530, 65004, 65259, 65387, 65447, 65536
31	0, 78, 227, 495, 953, 1698, 2849, 4538, 6893, 10013, 13940, 18637, 23975, 29739, 35652, 41416 46754, 51451, 55378, 58498, 60853, 62542, 63693, 64438, 64896, 65164, 65313,

Index	Probability
	65391, 65536

Table B.9 (continued)

Index	Probability
32	0, 58, 162, 340, 633, 1097, 1804, 2838, 4291, 6253, 8798, 11970, 15767, 20136, 24964, 30091 35322, 40449, 45277, 49646, 53443, 56614, 59159, 61121, 62574, 63608, 64315, 64779, 65072, 65250 65354, 65412, 65536
33	0, 50, 134, 271, 486, 814, 1299, 1995, 2962, 4263, 5961, 8109, 10742, 13870, 17471, 21490, 25836 30391, 35018, 39573, 43919, 47938, 51539, 54667, 57300, 59448, 61146, 62448, 63415, 64111, 64596 64924, 65139, 65276, 65360, 65410, 65536
34	0, 49, 126, 245, 423, 683, 1054, 1570, 2271, 3201, 4403, 5920, 7789, 10034, 12667, 15680, 19044 22709, 26606, 30648, 34740, 38783, 42680, 46345, 49709, 52722, 55355, 57600, 59469, 60986, 62188 63118, 63819, 64335, 64706, 64966, 65144, 65263, 65340, 65389, 65536
35	0, 53, 131, 245, 407, 634, 945, 1364, 1917, 2634, 3545, 4680, 6068, 7734, 9694, 11957, 14521 17370, 20475, 23795, 27278, 30863, 34483, 38068, 41551, 44871, 47976, 50825, 53389, 55652, 57612 59278, 60666, 61801, 62712, 63429, 63982, 64401, 64712, 64939, 65101, 65215, 65293, 65346, 65536
36	0, 43, 104, 189, 306, 466, 680, 962, 1329, 1799, 2392, 3129, 4031, 5119, 6412, 7925, 9669, 11650 13867, 16311, 18965, 21804, 24796, 27902, 31079, 34280, 37457, 40563, 43555, 46394, 49048, 51492 53709, 55690, 57434, 58947, 60240, 61328, 62230, 62967, 63560, 64030, 64397, 64679, 64893, 65053 65171, 65256, 65317, 65360, 65536
37	0, 39, 93, 166, 263, 390, 556, 770, 1042, 1383, 1807, 2327, 2958, 3715, 4613, 5664, 6881, 8274 9849, 11610, 13555, 15679, 17971, 20416, 22993, 25678, 28443, 31257, 34088, 36902, 39667, 42352 44929, 47374, 49666, 51790, 53735, 55496, 57071, 58464, 59681, 60732, 61630, 62387, 63018, 63538 63962, 64303, 64575, 64789, 64955, 65083, 65180, 65253, 65307, 65346, 65536
38	0, 31, 71, 124, 193, 282, 395, 538, 717, 939, 1211, 1543, 1943, 2422, 2989, 3655, 4429, 5322

Index	Probability
	6341, 7495, 8789, 10227, 11810, 13537, 15404, 17403, 19525, 21757, 24083, 26485, 28942, 31433 33936, 36427, 38884, 41286, 43612, 45844, 47966, 49965, 51832, 53559, 55142, 56580, 57874, 59028 60047, 60940, 61714, 62380, 62947, 63426, 63826, 64158, 64430, 64652, 64831, 64974, 65087, 65176 65245, 65298, 65338, 65369, 65536
39	0, 27, 62, 107, 163, 234, 322, 431, 564, 726, 922, 1158, 1439, 1771, 2162, 2618, 3146, 3753 4446, 5232, 6116, 7104, 8201, 9409, 10730, 12165, 13712, 15368, 17128, 18985, 20931, 22955 25045, 27188, 29370, 31575, 33788, 35993, 38175, 40318, 42408, 44432, 46378, 48235, 49995, 51651 53198, 54633, 55954, 57162, 58259, 59247, 60131, 60917, 61610, 62217, 62745, 63201, 63592, 63924 64205, 64441, 64637, 64799, 64932, 65041, 65129, 65200, 65256, 65301, 65336, 65363, 65536
40	0, 28, 62, 104, 156, 219, 296, 389, 500, 632, 789, 974, 1192, 1446, 1740, 2080, 2470, 2914, 3418 3987, 4625, 5337, 6127, 6998, 7953, 8995, 10125, 11344, 12651, 14045, 15524, 17084, 18720, 20426 22195, 24020, 25891, 27800, 29736, 31688, 33645, 35597, 37533, 39442, 41313, 43138, 44907, 46613 48249, 49809, 51288, 52682, 53989, 55208, 56338, 57380, 58335, 59206, 59996, 60708, 61346, 61915 62419, 62863, 63253, 63593, 63887, 64141, 64359, 64544, 64701, 64833, 64944, 65037, 65114, 65177 65229, 65271, 65305, 65333, 65536

Table B.9 (continued)

Index	Probability
41	0, 25, 55, 91, 135, 187, 249, 322, 408, 509, 627, 764, 923, 1107, 1318, 1559, 1833, 2144, 2494 2888, 3328, 3818, 4361, 4960, 5619, 6339, 7124, 7975, 8894, 9882, 10939, 12065, 13260, 14522 15850, 17240, 18689, 20193, 21748, 23347, 24986, 26657, 28354, 30070, 31797, 33528, 35255, 36971 38668, 40339, 41978, 43577, 45132, 46636, 48085, 49475, 50803, 52065, 53260, 54386, 55443, 56431 57350, 58201, 58986, 59706, 60365, 60964, 61507, 61997, 62437, 62831, 63181, 63492, 63766, 64007 64218, 64402, 64561, 64698, 64816, 64917, 65003, 65076, 65138, 65190, 65234, 65270, 65300, 65325

Index	Probability
	65536
42	0, 21, 46, 76, 111, 152, 200, 256, 320, 395, 481, 580, 693, 822, 969, 1135, 1323, 1534, 1771 2036, 2331, 2658, 3020, 3419, 3858, 4339, 4864, 5434, 6052, 6720, 7439, 8210, 9034, 9912, 10845 11832, 12873, 13967, 15113, 16309, 17553, 18842, 20174, 21545, 22952, 24391, 25857, 27346, 28853 30373, 31901, 33431, 34959, 36479, 37986, 39475, 40941, 42380, 43787, 45158, 46490, 47779, 49023 50219, 51365, 52459, 53500, 54487, 55420, 56298, 57122, 57893, 58612, 59280, 59898, 60468, 60993 61474, 61913, 62312, 62674, 63001, 63296, 63561, 63798, 64009, 64197, 64363, 64510, 64639, 64752 64851, 64937, 65012, 65076, 65132, 65180, 65221, 65256, 65286, 65311, 65332, 65536
43	0, 20, 44, 71, 102, 138, 180, 228, 282, 344, 414, 494, 584, 685, 799, 926, 1068, 1227, 1403 1599, 1815, 2053, 2315, 2603, 2918, 3261, 3635, 4040, 4479, 4952, 5462, 6009, 6595, 7221, 7887 8595, 9345, 10137, 10971, 11848, 12767, 13727, 14728, 15769, 16848, 17964, 19115, 20299, 21513 22755, 24022, 25311, 26620, 27945, 29282, 30628, 31979, 33332, 34683, 36029, 37366, 38691, 40000 41289, 42556, 43798, 45012, 46196, 47347, 48463, 49542, 50583, 51584, 52544, 53463, 54340, 55174 55966, 56716, 57424, 58090, 58716, 59302, 59849, 60359, 60832, 61271, 61676, 62050, 62393, 62708 62996, 63258, 63496, 63712, 63908, 64084, 64243, 64385, 64512, 64626, 64727, 64817, 64896, 64966 65028, 65082, 65130, 65172, 65208, 65239, 65266, 65290, 65310, 65536
44	0, 16, 35, 56, 80, 108, 139, 174, 214, 259, 309, 365, 428, 498, 577, 664, 761, 868, 986, 1116 1260, 1418, 1591, 1780, 1986, 2210, 2453, 2717, 3002, 3310, 3642, 3999, 4381, 4790, 5227, 5693 6189, 6715, 7273, 7862, 8484, 9139, 9827, 10548, 11302, 12089, 12909, 13761, 14645, 15560, 16505 17480, 18483, 19512, 20566, 21644, 22743, 23862, 24999, 26151, 27317, 28494, 29679, 30871, 32066 33263, 34458, 35650, 36835, 38012, 39178, 40330, 41467, 42586, 43685, 44763, 45817, 46846, 47849 48824, 49769, 50684, 51568, 52420, 53240, 54027, 54781, 55502, 56190, 56845, 57467, 58056, 58614 59140, 59636, 60102, 60539, 60948, 61330, 61687, 62019, 62327, 62612, 62876, 63119, 63343, 63549 63738, 63911, 64069, 64213, 64343, 64461, 64568, 64665, 64752, 64831, 64901,

Index	Probability
	64964, 65020, 65070
	65115, 65155, 65190, 65221, 65249, 65273, 65294, 65313, 65329, 65536
45	0, 15, 32, 51, 72, 96, 123, 153, 186, 223, 264, 309, 359, 414, 475, 542, 616, 697, 786, 883, 989
	1105, 1231, 1368, 1516, 1676, 1849, 2036, 2238, 2455, 2688, 2938, 3205, 3490, 3795, 4119, 4464
	4830, 5218, 5629, 6063, 6521, 7003, 7509, 8040, 8597, 9180, 9788, 10422, 11082, 11768, 12480
	13217, 13980, 14767, 15579, 16414, 17272, 18152, 19053, 19974, 20915, 21873, 22848, 23838, 24842
	25859, 26886, 27923, 28967, 30017, 31072, 32129, 33187, 34244, 35299, 36349, 37393, 38430, 39458
	40475, 41479, 42469, 43444, 44402, 45343, 46264, 47165, 48045, 48903, 49738, 50550, 51337, 52100
	52837, 53549, 54235, 54895, 55529, 56137, 56720, 57277, 57808, 58314, 58796, 59254, 59688, 60099
	60487, 60853, 61198, 61522, 61827, 62112, 62379, 62629, 62862, 63079, 63281, 63468, 63641, 63801
	63949, 64086, 64212, 64328, 64434, 64531, 64620, 64701, 64775, 64842, 64903, 64958, 65008, 65053
	65094, 65131, 65164, 65194, 65221, 65245, 65266, 65285, 65302, 65317, 65536

Table B.9 (continued)

Index	Probability
46	0, 14, 30, 47, 66, 87, 110, 136, 164, 195, 229, 266, 307, 351, 399, 452, 510, 573, 641, 715, 795
	881, 974, 1075, 1184, 1301, 1427, 1562, 1707, 1862, 2028, 2205, 2394, 2595, 2809, 3036, 3277
	3533, 3804, 4090, 4392, 4711, 5047, 5400, 5771, 6160, 6568, 6995, 7441, 7907, 8393, 8899, 9425
	9971, 10537, 11124, 11731, 12358, 13005, 13672, 14358, 15063, 15787, 16529, 17289, 18066, 18860
	19670, 20495, 21335, 22188, 23054, 23932, 24820, 25718, 26625, 27539, 28460, 29386, 30316, 31249
	32184, 33120, 34055, 34988, 35918, 36844, 37765, 38679, 39586, 40484, 41372, 42250, 43116, 43969
	44809, 45634, 46444, 47238, 48015, 48775, 49517, 50241, 50946, 51632, 52299, 52946, 53573, 54180
	54767, 55333, 55879, 56405, 56911, 57397, 57863, 58309, 58736, 59144, 59533, 59904, 60257, 60593
	60912, 61214, 61500, 61771, 62027, 62268, 62495, 62709, 62910, 63099, 63276, 63442, 63597, 63742
	63877, 64003, 64120, 64229, 64330, 64423, 64509, 64589, 64663, 64731, 64794,

Index	Probability
	64852, 64905, 64953 64997, 65038, 65075, 65109, 65140, 65168, 65194, 65217, 65238, 65257, 65274, 65290, 65304, 65536
47	0, 12, 25, 39, 55, 72, 91, 111, 133, 157, 183, 212, 243, 277, 313, 352, 395, 441, 491, 544, 601 663, 729, 800, 876, 958, 1045, 1138, 1238, 1344, 1457, 1578, 1706, 1842, 1987, 2141, 2304, 2476 2658, 2850, 3053, 3267, 3492, 3729, 3978, 4239, 4513, 4800, 5100, 5414, 5742, 6084, 6441, 6812 7198, 7600, 8017, 8449, 8897, 9361, 9841, 10337, 10848, 11375, 11918, 12477, 13051, 13641, 14246 14866, 15501, 16150, 16813, 17490, 18181, 18884, 19600, 20328, 21067, 21817, 22577, 23347, 24126 24913, 25707, 26508, 27315, 28128, 28945, 29766, 30590, 31416, 32243, 33070, 33897, 34723, 35547 36368, 37185, 37998, 38805, 39606, 40400, 41187, 41966, 42736, 43496, 44246, 44985, 45713, 46429 47132, 47823, 48500, 49163, 49812, 50447, 51067, 51672, 52262, 52836, 53395, 53938, 54465, 54976 55471, 55951, 56415, 56863, 57295, 57712, 58114, 58500, 58871, 59228, 59570, 59898, 60212, 60512 60799, 61073, 61334, 61583, 61820, 62045, 62259, 62462, 62654, 62836, 63008, 63171, 63325, 63470 63606, 63734, 63855, 63968, 64074, 64174, 64267, 64354, 64436, 64512, 64583, 64649, 64711, 64768 64821, 64871, 64917, 64960, 64999, 65035, 65069, 65100, 65129, 65155, 65179, 65201, 65221, 65240 65257, 65273, 65287, 65300, 65312, 65536
48	0, 11, 23, 35, 49, 64, 80, 97, 115, 135, 156, 179, 204, 231, 260, 291, 324, 359, 397, 437, 480 526, 575, 628, 684, 744, 808, 876, 948, 1024, 1105, 1191, 1282, 1378, 1480, 1587, 1700, 1820 1946, 2079, 2219, 2366, 2521, 2683, 2854, 3033, 3220, 3416, 3621, 3835, 4059, 4293, 4537, 4791 5055, 5330, 5616, 5913, 6221, 6541, 6872, 7215, 7570, 7937, 8316, 8707, 9110, 9526, 9954, 10394 10847, 11312, 11790, 12280, 12782, 13296, 13822, 14360, 14910, 15471, 16044, 16628, 17223, 17828 18444, 19069, 19704, 20348, 21001, 21663, 22333, 23010, 23695, 24386, 25084, 25787, 26496, 27209 27927, 28648, 29372, 30099, 30828, 31558, 32289, 33020, 33751, 34481, 35210, 35937, 36661, 37382 38100, 38813, 39522, 40225, 40923, 41614, 42299, 42976, 43646, 44308, 44961, 45605, 46240, 46865 47481, 48086, 48681, 49265, 49838, 50399, 50949, 51487, 52013, 52527, 53029,

Index	Probability
	53519, 53997, 54462
	54915, 55355, 55783, 56199, 56602, 56993, 57372, 57739, 58094, 58437, 58768, 59088, 59396, 59693
	59979, 60254, 60518, 60772, 61016, 61250, 61474, 61688, 61893, 62089, 62276, 62455, 62626, 62788
	62943, 63090, 63230, 63363, 63489, 63609, 63722, 63829, 63931, 64027, 64118, 64204, 64285, 64361
	64433, 64501, 64565, 64625, 64681, 64734, 64783, 64829, 64872, 64912, 64950, 64985, 65018, 65049
	65078, 65105, 65130, 65153, 65174, 65194, 65212, 65229, 65245, 65260, 65274, 65286, 65298, 65309
	65536

Table B.9 (continued)

Index	Probability
49	0, 10, 21, 32, 44, 57, 71, 86, 102, 119, 137, 156, 177, 199, 223, 248, 275, 304, 334, 366, 400
	436, 475, 516, 559, 605, 654, 705, 759, 816, 876, 940, 1007, 1078, 1153, 1232, 1315, 1402, 1493
	1589, 1690, 1795, 1905, 2021, 2142, 2269, 2401, 2539, 2683, 2834, 2991, 3155, 3326, 3504, 3689
	3881, 4080, 4287, 4502, 4725, 4956, 5195, 5443, 5699, 5964, 6238, 6521, 6813, 7114, 7424, 7744
	8073, 8412, 8760, 9118, 9485, 9862, 10249, 10645, 11051, 11467, 11892, 12327, 12772, 13226
	13689, 14162, 14644, 15135, 15635, 16144, 16661, 17187, 17721, 18263, 18813, 19371, 19936, 20508
	21087, 21673, 22265, 22863, 23467, 24076, 24690, 25309, 25932, 26559, 27190, 27824, 28461, 29101
	29743, 30387, 31032, 31678, 32325, 32972, 33619, 34265, 34910, 35554, 36196, 36836, 37473, 38107
	38738, 39365, 39988, 40607, 41221, 41830, 42434, 43032, 43624, 44210, 44789, 45361, 45926, 46484
	47034, 47576, 48110, 48636, 49153, 49662, 50162, 50653, 51135, 51608, 52071, 52525, 52970, 53405
	53830, 54246, 54652, 55048, 55435, 55812, 56179, 56537, 56885, 57224, 57553, 57873, 58183, 58484
	58776, 59059, 59333, 59598, 59854, 60102, 60341, 60572, 60795, 61010, 61217, 61416, 61608, 61793
	61971, 62142, 62306, 62463, 62614, 62758, 62896, 63028, 63155, 63276, 63392, 63502, 63607, 63708
	63804, 63895, 63982, 64065, 64144, 64219, 64290, 64357, 64421, 64481, 64538, 64592, 64643, 64692

Index	Probability
	64738, 64781, 64822, 64861, 64897, 64931, 64963, 64993, 65022, 65049, 65074, 65098, 65120, 65141 65160, 65178, 65195, 65211, 65226, 65240, 65253, 65265, 65276, 65287, 65297, 65536
50	0, 9, 18, 28, 39, 50, 62, 75, 89, 103, 118, 134, 151, 169, 188, 208, 230, 253, 277, 303, 330 358, 388, 420, 453, 488, 525, 564, 605, 648, 693, 740, 790, 842, 897, 954, 1014, 1077, 1143 1212, 1284, 1359, 1438, 1520, 1606, 1696, 1790, 1887, 1989, 2095, 2205, 2319, 2438, 2562, 2691 2824, 2962, 3106, 3255, 3409, 3569, 3734, 3905, 4082, 4265, 4454, 4649, 4851, 5059, 5274, 5495 5723, 5958, 6200, 6449, 6705, 6968, 7238, 7515, 7800, 8092, 8392, 8699, 9013, 9335, 9665, 10002 10347, 10699, 11059, 11426, 11801, 12184, 12574, 12971, 13376, 13788, 14207, 14634, 15068, 15509 15957, 16411, 16872, 17340, 17814, 18294, 18781, 19274, 19772, 20276, 20785, 21300, 21820, 22345 22874, 23408, 23946, 24488, 25034, 25583, 26136, 26692, 27250, 27811, 28374, 28939, 29506, 30075 30645, 31216, 31787, 32359, 32931, 33503, 34074, 34645, 35215, 35784, 36351, 36916, 37479, 38040 38598, 39154, 39707, 40256, 40802, 41344, 41882, 42416, 42945, 43470, 43990, 44505, 45014, 45518 46016, 46509, 46996, 47476, 47950, 48418, 48879, 49333, 49781, 50222, 50656, 51083, 51502, 51914 52319, 52716, 53106, 53489, 53864, 54231, 54591, 54943, 55288, 55625, 55955, 56277, 56591, 56898 57198, 57490, 57775, 58052, 58322, 58585, 58841, 59090, 59332, 59567, 59795, 60016, 60231, 60439 60641, 60836, 61025, 61208, 61385, 61556, 61721, 61881, 62035, 62184, 62328, 62466, 62599, 62728 62852, 62971, 63085, 63195, 63301, 63403, 63500, 63594, 63684, 63770, 63852, 63931, 64006, 64078 64147, 64213, 64276, 64336, 64393, 64448, 64500, 64550, 64597, 64642, 64685, 64726, 64765, 64802 64837, 64870, 64902, 64932, 64960, 64987, 65013, 65037, 65060, 65082, 65102, 65121, 65139, 65156 65172, 65187, 65201, 65215, 65228, 65240, 65251, 65262, 65272, 65281, 65290, 65536

Table B.9 (continued)

Index	Probability
51	0, 8, 16, 24, 33, 42, 52, 62, 73, 85, 97, 110, 124, 138, 153, 169, 186, 204, 222, 241, 261, 282 304, 328, 353, 379, 406, 435, 465, 496, 529, 563, 599, 636, 675, 716, 759, 803, 849, 897, 947 1000, 1055, 1112, 1171, 1233, 1297, 1364, 1434, 1506, 1581, 1659, 1740, 1824, 1911, 2001, 2094 2191, 2291, 2395, 2502, 2613, 2728, 2847, 2970, 3097, 3228, 3363, 3502, 3646, 3794, 3947, 4104 4266, 4433, 4605, 4782, 4964, 5151, 5343, 5540, 5742, 5950, 6163, 6381, 6605, 6834, 7069, 7310 7556, 7808, 8066, 8330, 8600, 8875, 9156, 9443, 9736, 10035, 10340, 10651, 10968, 11291, 11620 11955, 12295, 12641, 12993, 13351, 13715, 14084, 14459, 14839, 15225, 15616, 16013, 16415, 16822 17235, 17653, 18076, 18503, 18935, 19372, 19813, 20259, 20709, 21163, 21621, 22083, 22549, 23018 23491, 23967, 24446, 24928, 25413, 25901, 26391, 26883, 27377, 27873, 28371, 28871, 29372, 29874 30377, 30881, 31386, 31891, 32397, 32903, 33409, 33914, 34419, 34923, 35426, 35928, 36429, 36929 37427, 37923, 38417, 38909, 39399, 39887, 40372, 40854, 41333, 41809, 42282, 42751, 43217, 43679 44137, 44591, 45041, 45487, 45928, 46365, 46797, 47224, 47647, 48065, 48478, 48885, 49287, 49684 50075, 50461, 50841, 51216, 51585, 51949, 52307, 52659, 53005, 53345, 53680, 54009, 54332, 54649 54960, 55265, 55564, 55857, 56144, 56425, 56700, 56970, 57234, 57492, 57744, 57990, 58231, 58466 58695, 58919, 59137, 59350, 59558, 59760, 59957, 60149, 60336, 60518, 60695, 60867, 61034, 61196 61353, 61506, 61654, 61798, 61937, 62072, 62203, 62330, 62453, 62572, 62687, 62798, 62905, 63009 63109, 63206, 63299, 63389, 63476, 63560, 63641, 63719, 63794, 63866, 63936, 64003, 64067, 64129 64188, 64245, 64300, 64353, 64403, 64451, 64497, 64541, 64584, 64625, 64664, 64701, 64737, 64771 64804, 64835, 64865, 64894, 64921, 64947, 64972, 64996, 65018, 65039, 65059, 65078, 65096, 65114 65131, 65147, 65162, 65176, 65190, 65203, 65215, 65227, 65238, 65248, 65258, 65267, 65276, 65284 65292, 65300, 65536
52	0, 7, 14, 22, 30, 38, 47, 56, 66, 76, 87, 98, 110, 122, 135, 148, 162, 177, 192, 208, 225, 242 260, 279, 299, 320, 342, 364, 387, 411, 436, 463, 491, 520, 550, 581, 613, 647, 682,

Index	Probability
	718, 756
	795, 836, 878, 922, 968, 1015, 1064, 1115, 1168, 1222, 1278, 1336, 1396, 1458, 1523, 1590, 1659
	1730, 1804, 1880, 1959, 2040, 2124, 2210, 2299, 2391, 2486, 2584, 2685, 2789, 2896, 3006, 3119
	3235, 3354, 3477, 3603, 3733, 3866, 4003, 4143, 4287, 4435, 4587, 4742, 4901, 5064, 5231, 5402
	5577, 5756, 5940, 6128, 6320, 6516, 6717, 6922, 7131, 7345, 7563, 7786, 8013, 8245, 8481, 8722
	8968, 9218, 9473, 9732, 9996, 10265, 10538, 10816, 11099, 11386, 11678, 11975, 12276, 12582
	12892, 13207, 13526, 13850, 14178, 14511, 14848, 15189, 15534, 15884, 16238, 16596, 16958, 17324
	17694, 18068, 18446, 18827, 19212, 19601, 19993, 20388, 20787, 21189, 21594, 22002, 22413, 22827
	23244, 23663, 24085, 24509, 24935, 25364, 25795, 26228, 26662, 27098, 27536, 27975, 28416, 28858
	29301, 29745, 30190, 30635, 31081, 31528, 31975, 32422, 32869, 33316, 33763, 34210, 34656, 35101
	35546, 35990, 36433, 36875, 37316, 37755, 38193, 38629, 39063, 39496, 39927, 40356, 40782, 41206
	41628, 42047, 42464, 42878, 43289, 43697, 44102, 44504, 44903, 45298, 45690, 46079, 46464, 46845
	47223, 47597, 47967, 48333, 48695, 49053, 49407, 49757, 50102, 50443, 50780, 51113, 51441, 51765
	52084, 52399, 52709, 53015, 53316, 53613, 53905, 54192, 54475, 54753, 55026, 55295, 55559, 55818
	56073, 56323, 56569, 56810, 57046, 57278, 57505, 57728, 57946, 58160, 58369, 58574, 58775, 58971
	59163, 59351, 59535, 59714, 59889, 60060, 60227, 60390, 60549, 60704, 60856, 61004, 61148, 61288
	61425, 61558, 61688, 61814, 61937, 62056, 62172, 62285, 62395, 62502, 62606, 62707, 62805, 62900
	62992, 63081, 63167, 63251, 63332, 63411, 63487, 63561, 63632, 63701, 63768, 63833, 63895, 63955
	64013, 64069, 64123, 64176, 64227, 64276, 64323, 64369, 64413, 64455, 64496, 64535, 64573, 64609
	64644, 64678, 64710, 64741, 64771, 64800, 64828, 64855, 64880, 64904, 64927, 64949, 64971, 64992
	65012, 65031, 65049, 65066, 65083, 65099, 65114, 65129, 65143, 65156, 65169, 65181, 65193, 65204
	65215, 65225, 65235, 65244, 65253, 65261, 65269, 65277, 65284, 65291, 65536

Table B.9 (continued)

Index	Probability
53	0, 6, 12, 19, 26, 33, 41, 49, 57, 66, 75, 84, 94, 104, 115, 126, 137, 149, 161, 174, 187, 201
	216, 231, 247, 263, 280, 298, 316, 335, 355, 375, 396, 418, 441, 465, 489, 514, 540, 567, 595
	624, 654, 685, 717, 750, 785, 821, 858, 896, 935, 976, 1018, 1061, 1106, 1152, 1200, 1249, 1300
	1352, 1406, 1462, 1519, 1578, 1639, 1702, 1766, 1832, 1900, 1970, 2042, 2116, 2192, 2270, 2350
	2433, 2518, 2605, 2694, 2786, 2880, 2977, 3076, 3178, 3282, 3389, 3498, 3610, 3725, 3843, 3963
	4086, 4212, 4341, 4473, 4608, 4746, 4887, 5031, 5178, 5328, 5481, 5638, 5798, 5961, 6127, 6297
	6470, 6646, 6826, 7009, 7196, 7386, 7580, 7777, 7978, 8182, 8390, 8601, 8816, 9035, 9257, 9483
	9712, 9945, 10182, 10422, 10666, 10914, 11165, 11420, 11678, 11940, 12206, 12475, 12748, 13024
	13304, 13587, 13874, 14164, 14458, 14755, 15056, 15360, 15667, 15977, 16291, 16608, 16928, 17251
	17577, 17906, 18238, 18573, 18911, 19252, 19596, 19943, 20292, 20644, 20998, 21355, 21714, 22076
	22440, 22806, 23174, 23544, 23916, 24290, 24666, 25044, 25423, 25804, 26187, 26571, 26956, 27343
	27731, 28120, 28510, 28901, 29293, 29686, 30079, 30473, 30867, 31262, 31657, 32052, 32447, 32842
	33237, 33632, 34027, 34422, 34816, 35210, 35603, 35996, 36388, 36779, 37169, 37558, 37946, 38333
	38718, 39102, 39485, 39866, 40245, 40623, 40999, 41373, 41745, 42115, 42483, 42849, 43213, 43575
	43934, 44291, 44645, 44997, 45346, 45692, 46036, 46377, 46715, 47050, 47382, 47711, 48037, 48360
	48680, 48997, 49311, 49621, 49928, 50232, 50533, 50830, 51124, 51414, 51701, 51984, 52264, 52540
	52813, 53082, 53348, 53610, 53868, 54123, 54374, 54622, 54866, 55106, 55343, 55576, 55805, 56031
	56253, 56472, 56687, 56898, 57106, 57310, 57511, 57708, 57902, 58092, 58279, 58462, 58642, 58818
	58991, 59161, 59327, 59490, 59650, 59807, 59960, 60110, 60257, 60401, 60542, 60680, 60815, 60947
	61076, 61202, 61325, 61445, 61563, 61678, 61790, 61899, 62006, 62110, 62212, 62311, 62408, 62502
	62594, 62683, 62770, 62855, 62938, 63018, 63096, 63172, 63246, 63318, 63388, 63456, 63522, 63586
	63649, 63710, 63769, 63826, 63882, 63936, 63988, 64039, 64088, 64136, 64182, 64227, 64270, 64312

Index	Probability
	64353, 64392, 64430, 64467, 64503, 64538, 64571, 64603, 64634, 64664, 64693, 64721, 64748, 64774
	64799, 64823, 64847, 64870, 64892, 64913, 64933, 64953, 64972, 64990, 65008, 65025, 65041, 65057
	65072, 65087, 65101, 65114, 65127, 65139, 65151, 65162, 65173, 65184, 65194, 65204, 65213, 65222
	65231, 65239, 65247, 65255, 65262, 65269, 65276, 65282, 65288, 65536

Table B.9 (continued)

Index	Probability
54	0, 5, 11, 17, 23, 29, 36, 43, 50, 57, 65, 73, 81, 89, 98, 107, 116, 126, 136, 147, 158, 169, 181
	193, 206, 219, 232, 246, 260, 275, 290, 306, 322, 339, 357, 375, 394, 413, 433, 454, 475, 497
	520, 543, 567, 592, 618, 644, 671, 699, 728, 758, 789, 821, 854, 887, 921, 956, 992, 1030, 1069
	1109, 1150, 1192, 1235, 1279, 1325, 1372, 1420, 1469, 1520, 1572, 1625, 1680, 1736, 1794, 1853
	1914, 1976, 2040, 2105, 2172, 2241, 2311, 2383, 2457, 2532, 2609, 2688, 2769, 2852, 2937, 3023
	3111, 3201, 3293, 3387, 3483, 3582, 3683, 3786, 3891, 3998, 4107, 4219, 4333, 4449, 4568, 4689
	4812, 4938, 5066, 5196, 5329, 5464, 5602, 5742, 5885, 6031, 6179, 6330, 6483, 6639, 6798, 6959
	7123, 7290, 7459, 7631, 7806, 7984, 8164, 8347, 8533, 8722, 8914, 9108, 9305, 9505, 9708, 9914
	10123, 10334, 10548, 10765, 10985, 11208, 11434, 11662, 11893, 12127, 12364, 12604, 12846, 13091
	13339, 13590, 13843, 14099, 14358, 14619, 14883, 15150, 15419, 15691, 15966, 16243, 16523, 16805
	17090, 17377, 17666, 17958, 18252, 18548, 18847, 19148, 19451, 19756, 20063, 20372, 20683, 20996
	21311, 21628, 21947, 22268, 22591, 22915, 23241, 23569, 23898, 24229, 24561, 24894, 25229, 25565
	25902, 26240, 26580, 26921, 27263, 27606, 27950, 28294, 28639, 28985, 29331, 29678, 30026, 30374
	30722, 31071, 31420, 31769, 32118, 32468, 32818, 33168, 33517, 33866, 34215, 34564, 34912, 35260
	35608, 35955, 36301, 36647, 36992, 37336, 37680, 38023, 38365, 38706, 39046, 39385, 39722, 40058
	40393, 40726, 41058, 41389, 41718, 42046, 42372, 42696, 43019, 43340, 43659, 43976, 44291, 44604
	44915, 45224, 45531, 45836, 46139, 46440, 46739, 47035, 47329, 47621, 47910,

Index	Probability
	48197, 48482, 48764
	49044, 49321, 49596, 49868, 50137, 50404, 50668, 50929, 51188, 51444, 51697, 51948, 52196, 52441
	52683, 52923, 53160, 53394, 53625, 53853, 54079, 54302, 54522, 54739, 54953, 55164, 55373, 55579
	55782, 55982, 56179, 56373, 56565, 56754, 56940, 57123, 57303, 57481, 57656, 57828, 57997, 58164
	58328, 58489, 58648, 58804, 58957, 59108, 59256, 59402, 59545, 59685, 59823, 59958, 60091, 60221
	60349, 60475, 60598, 60719, 60838, 60954, 61068, 61180, 61289, 61396, 61501, 61604, 61705, 61804
	61900, 61994, 62086, 62176, 62264, 62350, 62435, 62518, 62599, 62678, 62755, 62830, 62904, 62976
	63046, 63115, 63182, 63247, 63311, 63373, 63434, 63493, 63551, 63607, 63662, 63715, 63767, 63818
	63867, 63915, 63962, 64008, 64052, 64095, 64137, 64178, 64218, 64257, 64295, 64331, 64366, 64400
	64433, 64466, 64498, 64529, 64559, 64588, 64616, 64643, 64669, 64695, 64720, 64744, 64767, 64790
	64812, 64833, 64854, 64874, 64893, 64912, 64930, 64948, 64965, 64981, 64997, 65012, 65027, 65041
	65055, 65068, 65081, 65094, 65106, 65118, 65129, 65140, 65151, 65161, 65171, 65180, 65189, 65198
	65206, 65214, 65222, 65230, 65237, 65244, 65251, 65258, 65264, 65270, 65276, 65282, 65287, 65536

Table B.9 (continued)

Index	Probability
55	0, 5, 10, 15, 20, 25, 31, 37, 43, 49, 55, 62, 69, 76, 83, 90, 98, 106, 114, 122, 131, 140, 149
	159, 169, 179, 189, 200, 211, 223, 235, 247, 260, 273, 286, 300, 314, 329, 344, 359, 375, 391
	408, 425, 443, 461, 480, 499, 519, 540, 561, 583, 605, 628, 652, 676, 701, 726, 752, 779, 807
	835, 864, 894, 925, 956, 988, 1021, 1055, 1090, 1125, 1161, 1198, 1236, 1275, 1315, 1356, 1398
	1441, 1485, 1530, 1576, 1623, 1671, 1720, 1771, 1823, 1876, 1930, 1985, 2041, 2099, 2158, 2218
	2279, 2342, 2406, 2471, 2538, 2606, 2676, 2747, 2820, 2894, 2970, 3047, 3126, 3206, 3288, 3371
	3456, 3542, 3630, 3720, 3811, 3904, 3999, 4096, 4194, 4294, 4396, 4500, 4606, 4713, 4822, 4933
	5046, 5161, 5278, 5397, 5518, 5641, 5766, 5893, 6022, 6153, 6286, 6421, 6558, 6697,

Index	Probability
	6838, 6981
	7126, 7273, 7422, 7574, 7728, 7884, 8042, 8202, 8364, 8528, 8695, 8864, 9035, 9208, 9383, 9561
	9741, 9923, 10107, 10294, 10483, 10674, 10867, 11062, 11260, 11460, 11662, 11866, 12072, 12281
	12492, 12705, 12920, 13137, 13356, 13578, 13802, 14028, 14256, 14486, 14718, 14952, 15188, 15426
	15666, 15908, 16152, 16398, 16646, 16896, 17148, 17402, 17658, 17916, 18176, 18437, 18700, 18965
	19232, 19500, 19770, 20042, 20315, 20590, 20866, 21144, 21423, 21704, 21986, 22270, 22555, 22841
	23129, 23418, 23708, 23999, 24292, 24586, 24881, 25177, 25474, 25772, 26071, 26371, 26672, 26973
	27275, 27578, 27882, 28186, 28491, 28796, 29102, 29409, 29716, 30023, 30331, 30639, 30947, 31256
	31565, 31874, 32183, 32492, 32801, 33110, 33419, 33728, 34037, 34346, 34654, 34962, 35270, 35577
	35884, 36191, 36497, 36802, 37107, 37411, 37715, 38018, 38320, 38621, 38922, 39222, 39521, 39819
	40116, 40412, 40707, 41001, 41294, 41585, 41875, 42164, 42452, 42738, 43023, 43307, 43589, 43870
	44149, 44427, 44703, 44978, 45251, 45523, 45793, 46061, 46328, 46593, 46856, 47117, 47377, 47635
	47891, 48145, 48397, 48647, 48895, 49141, 49385, 49627, 49867, 50105, 50341, 50575, 50807, 51037
	51265, 51491, 51715, 51937, 52156, 52373, 52588, 52801, 53012, 53221, 53427, 53631, 53833, 54033
	54231, 54426, 54619, 54810, 54999, 55186, 55370, 55552, 55732, 55910, 56085, 56258, 56429, 56598
	56765, 56929, 57091, 57251, 57409, 57565, 57719, 57871, 58020, 58167, 58312, 58455, 58596, 58735
	58872, 59007, 59140, 59271, 59400, 59527, 59652, 59775, 59896, 60015, 60132, 60247, 60360, 60471
	60580, 60687, 60793, 60897, 60999, 61099, 61197, 61294, 61389, 61482, 61573, 61663, 61751, 61837
	61922, 62005, 62087, 62167, 62246, 62323, 62399, 62473, 62546, 62617, 62687, 62755, 62822, 62887
	62951, 63014, 63075, 63135, 63194, 63252, 63308, 63363, 63417, 63470, 63522, 63573, 63622, 63670
	63717, 63763, 63808, 63852, 63895, 63937, 63978, 64018, 64057, 64095, 64132, 64168, 64203, 64238
	64272, 64305, 64337, 64368, 64399, 64429, 64458, 64486, 64514, 64541, 64567, 64592, 64617, 64641
	64665, 64688, 64710, 64732, 64753, 64774, 64794, 64813, 64832, 64850, 64868, 64885, 64902, 64918
	64934, 64949, 64964, 64979, 64993, 65007, 65020, 65033, 65046, 65058, 65070,

Index	Probability
	65082, 65093, 65104 65114, 65124, 65134, 65144, 65153, 65162, 65171, 65179, 65187, 65195, 65203, 65210, 65217, 65224 65231, 65238, 65244, 65250, 65256, 65262, 65268, 65273, 65278, 65283, 65288, 65293, 65536

Table B.9 (continued)

Index	Probability
56	0, 4, 8, 12, 17, 22, 27, 32, 37, 42, 47, 53, 59, 65, 71, 77, 83, 90, 97, 104, 111, 118, 126, 134 142, 150, 159, 168, 177, 186, 195, 205, 215, 225, 236, 247, 258, 269, 281, 293, 305, 318, 331 344, 358, 372, 386, 401, 416, 432, 448, 464, 481, 498, 516, 534, 553, 572, 591, 611, 631, 652 673, 695, 717, 740, 763, 787, 812, 837, 863, 889, 916, 944, 972, 1001, 1030, 1060, 1091, 1122 1154, 1187, 1221, 1255, 1290, 1326, 1362, 1399, 1437, 1476, 1516, 1556, 1597, 1639, 1682, 1726 1771, 1817, 1864, 1911, 1959, 2008, 2058, 2109, 2161, 2214, 2268, 2323, 2379, 2436, 2494, 2554 2615, 2677, 2740, 2804, 2869, 2935, 3002, 3071, 3141, 3212, 3284, 3358, 3433, 3509, 3586, 3665 3745, 3826, 3909, 3993, 4078, 4165, 4253, 4342, 4433, 4525, 4619, 4714, 4811, 4909, 5008, 5109 5211, 5315, 5420, 5527, 5636, 5746, 5858, 5971, 6086, 6202, 6320, 6440, 6561, 6684, 6808, 6934 7062, 7191, 7322, 7455, 7589, 7725, 7863, 8002, 8143, 8286, 8431, 8577, 8725, 8875, 9026, 9179 9334, 9491, 9649, 9809, 9971, 10134, 10299, 10466, 10635, 10805, 10977, 11151, 11327, 11504 11683, 11864, 12046, 12230, 12416, 12604, 12793, 12984, 13177, 13371, 13567, 13765, 13964, 14165 14368, 14572, 14778, 14986, 15195, 15406, 15618, 15832, 16047, 16264, 16483, 16703, 16925, 17148 17373, 17599, 17826, 18055, 18285, 18517, 18750, 18984, 19220, 19457, 19695, 19935, 20176, 20418 20662, 20907, 21153, 21400, 21648, 21897, 22147, 22399, 22652, 22906, 23161, 23417, 23674, 23932 24190, 24449, 24709, 24970, 25232, 25495, 25758, 26022, 26287, 26553, 26819, 27086, 27353, 27621 27890, 28159, 28429, 28699, 28969, 29240, 29511, 29783, 30055, 30327, 30599, 30872, 31145, 31418 31691, 31964, 32237, 32510, 32783, 33056, 33329, 33602, 33875, 34148, 34421,

Index	Probability
	34694, 34966, 35238
	35510, 35782, 36053, 36324, 36594, 36864, 37134, 37403, 37672, 37940, 38207, 38474, 38740, 39006
	39271, 39535, 39798, 40061, 40323, 40584, 40844, 41103, 41361, 41619, 41876, 42132, 42387, 42641
	42894, 43146, 43396, 43645, 43893, 44140, 44386, 44631, 44875, 45117, 45358, 45598, 45836, 46073
	46309, 46543, 46776, 47008, 47238, 47467, 47694, 47920, 48145, 48368, 48590, 48810, 49029, 49246
	49461, 49675, 49887, 50098, 50307, 50515, 50721, 50925, 51128, 51329, 51528, 51726, 51922, 52116
	52309, 52500, 52689, 52877, 53063, 53247, 53429, 53610, 53789, 53966, 54142, 54316, 54488, 54658
	54827, 54994, 55159, 55322, 55484, 55644, 55802, 55959, 56114, 56267, 56418, 56568, 56716, 56862
	57007, 57150, 57291, 57430, 57568, 57704, 57838, 57971, 58102, 58231, 58359, 58485, 58609, 58732
	58853, 58973, 59091, 59207, 59322, 59435, 59547, 59657, 59766, 59873, 59978, 60082, 60184, 60285
	60384, 60482, 60579, 60674, 60768, 60860, 60951, 61040, 61128, 61215, 61300, 61384, 61467, 61548
	61628, 61707, 61784, 61860, 61935, 62009, 62081, 62152, 62222, 62291, 62358, 62424, 62489, 62553
	62616, 62678, 62739, 62799, 62857, 62914, 62970, 63025, 63079, 63132, 63184, 63235, 63285, 63334
	63382, 63429, 63476, 63522, 63567, 63611, 63654, 63696, 63737, 63777, 63817, 63856, 63894, 63931
	63967, 64003, 64038, 64072, 64105, 64138, 64170, 64201, 64232, 64262, 64291, 64320, 64348, 64376
	64403, 64429, 64455, 64480, 64505, 64529, 64552, 64575, 64597, 64619, 64640, 64661, 64681, 64701
	64720, 64739, 64758, 64776, 64794, 64811, 64828, 64844, 64860, 64876, 64891, 64906, 64920, 64934
	64948, 64961, 64974, 64987, 64999, 65011, 65023, 65034, 65045, 65056, 65067, 65077, 65087, 65097
	65106, 65115, 65124, 65133, 65142, 65150, 65158, 65166, 65174, 65181, 65188, 65195, 65202, 65209
	65215, 65221, 65227, 65233, 65239, 65245, 65250, 65255, 65260, 65265, 65270, 65275, 65280, 65284
	65288, 65292, 65536

Table B.9 (continued)

Index	Probability

Index	Probability
57	0, 4, 8, 12, 16, 20, 24, 28, 32, 37, 42, 47, 52, 57, 62, 67, 72, 78, 84, 90, 96, 102, 108, 115
	122, 129, 136, 143, 150, 158, 166, 174, 182, 190, 198, 207, 216, 225, 234, 244, 254, 264, 274
	284, 295, 306, 317, 329, 341, 353, 365, 378, 391, 404, 417, 431, 445, 459, 474, 489, 504, 520
	536, 552, 569, 586, 603, 621, 639, 658, 677, 696, 716, 736, 757, 778, 799, 821, 843, 866, 889
	913, 937, 962, 987, 1013, 1039, 1066, 1093, 1121, 1149, 1178, 1207, 1237, 1268, 1299, 1331, 1363
	1396, 1430, 1464, 1499, 1534, 1570, 1607, 1644, 1682, 1721, 1760, 1800, 1841, 1883, 1925, 1968
	2012, 2057, 2102, 2148, 2195, 2243, 2291, 2340, 2390, 2441, 2493, 2546, 2599, 2653, 2708, 2764
	2821, 2879, 2938, 2998, 3059, 3120, 3182, 3245, 3309, 3374, 3440, 3507, 3575, 3644, 3714, 3785
	3857, 3930, 4004, 4080, 4157, 4235, 4314, 4394, 4475, 4557, 4640, 4724, 4809, 4896, 4984, 5073
	5163, 5254, 5346, 5440, 5535, 5631, 5728, 5826, 5926, 6027, 6129, 6232, 6337, 6443, 6550, 6658
	6768, 6879, 6991, 7105, 7220, 7336, 7453, 7572, 7692, 7813, 7936, 8060, 8185, 8312, 8440, 8569
	8700, 8832, 8965, 9100, 9236, 9373, 9512, 9652, 9793, 9936, 10080, 10226, 10373, 10521, 10671
	10822, 10974, 11128, 11283, 11439, 11597, 11756, 11916, 12078, 12241, 12405, 12571, 12738, 12906
	13076, 13247, 13419, 13593, 13768, 13944, 14122, 14301, 14481, 14662, 14845, 15029, 15214, 15400
	15588, 15777, 15967, 16158, 16351, 16545, 16740, 16936, 17133, 17331, 17531, 17732, 17934, 18137
	18341, 18546, 18752, 18959, 19167, 19376, 19586, 19797, 20009, 20222, 20436, 20651, 20867, 21084
	21302, 21521, 21741, 21962, 22184, 22407, 22630, 22854, 23079, 23305, 23532, 23759, 23987, 24216
	24445, 24675, 24906, 25137, 25369, 25602, 25835, 26069, 26303, 26538, 26773, 27009, 27245, 27482
	27719, 27957, 28195, 28433, 28672, 28911, 29150, 29390, 29630, 29870, 30110, 30351, 30592, 30833
	31074, 31315, 31556, 31798, 32040, 32282, 32524, 32766, 33008, 33250, 33492, 33734, 33975, 34216
	34457, 34698, 34939, 35180, 35420, 35660, 35900, 36140, 36379, 36618, 36857, 37095, 37333, 37571
	37808, 38045, 38281, 38517, 38752, 38987, 39221, 39455, 39688, 39921, 40153, 40384, 40615, 40845
	41074, 41303, 41531, 41758, 41985, 42211, 42436, 42660, 42883, 43106, 43328, 43549, 43769, 43988

Index	Probability
	44206, 44423, 44639, 44854, 45068, 45281, 45493, 45704, 45914, 46123, 46331, 46538, 46744, 46949
	47153, 47356, 47558, 47759, 47959, 48157, 48354, 48550, 48745, 48939, 49132, 49323, 49513, 49702
	49890, 50076, 50261, 50445, 50628, 50809, 50989, 51168, 51346, 51522, 51697, 51871, 52043, 52214
	52384, 52552, 52719, 52885, 53049, 53212, 53374, 53534, 53693, 53851, 54007, 54162, 54316, 54468
	54619, 54769, 54917, 55064, 55210, 55354, 55497, 55638, 55778, 55917, 56054, 56190, 56325, 56458
	56590, 56721, 56850, 56978, 57105, 57230, 57354, 57477, 57598, 57718, 57837, 57954, 58070, 58185
	58299, 58411, 58522, 58632, 58740, 58847, 58953, 59058, 59161, 59263, 59364, 59464, 59562, 59659
	59755, 59850, 59944, 60036, 60127, 60217, 60306, 60394, 60481, 60566, 60650, 60733, 60815, 60896
	60976, 61055, 61133, 61210, 61286, 61360, 61433, 61505, 61576, 61646, 61715, 61783, 61850, 61916
	61981, 62045, 62108, 62170, 62231, 62292, 62352, 62411, 62469, 62526, 62582, 62637, 62691, 62744
	62797, 62849, 62900, 62950, 62999, 63047, 63095, 63142, 63188, 63233, 63278, 63322, 63365, 63407
	63449, 63490, 63530, 63569, 63608, 63646, 63683, 63720, 63756, 63791, 63826, 63860, 63894, 63927
	63959, 63991, 64022, 64053, 64083, 64112, 64141, 64169, 64197, 64224, 64251, 64277, 64303, 64328
	64353, 64377, 64401, 64424, 64447, 64469, 64491, 64512, 64533, 64554, 64574, 64594, 64613, 64632
	64651, 64669, 64687, 64704, 64721, 64738, 64754, 64770, 64786, 64801, 64816, 64831, 64845, 64859
	64873, 64886, 64899, 64912, 64925, 64937, 64949, 64961, 64973, 64984, 64995, 65006, 65016, 65026
	65036, 65046, 65056, 65065, 65074, 65083, 65092, 65100, 65108, 65116, 65124, 65132, 65140, 65147
	65154, 65161, 65168, 65175, 65182, 65188, 65194, 65200, 65206, 65212, 65218, 65223, 65228, 65233
	65238, 65243, 65248, 65253, 65258, 65262, 65266, 65270, 65274, 65278, 65282, 65286, 65290, 65536

Table B.9 (continued)

Index	Probability
58	0, 3, 6, 9, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 61, 66, 71, 76, 81, 86, 91, 96, 102
	108, 114, 120, 126, 132, 138, 144, 151, 158, 165, 172, 179, 186, 193, 201, 209, 217,

Index	Probability
	225, 233
	241, 250, 259, 268, 277, 286, 296, 306, 316, 326, 336, 347, 358, 369, 380, 391, 403, 415, 427
	439, 452, 465, 478, 491, 505, 519, 533, 547, 562, 577, 592, 608, 624, 640, 656, 673, 690, 707
	725, 743, 761, 780, 799, 818, 838, 858, 878, 899, 920, 942, 964, 986, 1009, 1032, 1056, 1080
	1104, 1129, 1154, 1180, 1206, 1232, 1259, 1286, 1314, 1342, 1371, 1400, 1430, 1460, 1491, 1522
	1554, 1586, 1619, 1652, 1686, 1720, 1755, 1790, 1826, 1863, 1900, 1938, 1976, 2015, 2054, 2094
	2135, 2176, 2218, 2260, 2303, 2347, 2391, 2436, 2482, 2528, 2575, 2623, 2671, 2720, 2770, 2820
	2871, 2923, 2975, 3028, 3082, 3137, 3192, 3248, 3305, 3363, 3421, 3480, 3540, 3601, 3663, 3725
	3788, 3852, 3917, 3983, 4049, 4116, 4184, 4253, 4323, 4394, 4466, 4538, 4611, 4685, 4760, 4836
	4913, 4991, 5070, 5150, 5230, 5311, 5393, 5476, 5560, 5645, 5731, 5818, 5906, 5995, 6085, 6176
	6268, 6361, 6455, 6550, 6646, 6743, 6841, 6940, 7040, 7141, 7243, 7346, 7450, 7555, 7661, 7768
	7876, 7985, 8095, 8206, 8318, 8431, 8545, 8660, 8776, 8893, 9011, 9130, 9250, 9372, 9495, 9619
	9744, 9870, 9997, 10125, 10254, 10384, 10515, 10647, 10780, 10914, 11049, 11185, 11322, 11460
	11599, 11740, 11882, 12025, 12169, 12314, 12460, 12607, 12755, 12904, 13054, 13205, 13357, 13510
	13664, 13819, 13975, 14132, 14290, 14449, 14609, 14770, 14932, 15095, 15259, 15424, 15590, 15757
	15925, 16094, 16264, 16435, 16607, 16780, 16953, 17127, 17302, 17478, 17655, 17833, 18012, 18192
	18373, 18554, 18736, 18919, 19103, 19288, 19474, 19660, 19847, 20035, 20224, 20413, 20603, 20794
	20986, 21178, 21371, 21565, 21760, 21955, 22151, 22348, 22545, 22743, 22941, 23140, 23340, 23540
	23741, 23942, 24144, 24347, 24550, 24754, 24958, 25163, 25368, 25574, 25780, 25986, 26193, 26400
	26608, 26816, 27025, 27234, 27443, 27653, 27863, 28073, 28284, 28495, 28706, 28917, 29129, 29341
	29553, 29765, 29977, 30190, 30403, 30616, 30829, 31042, 31255, 31468, 31682, 31896, 32110, 32324
	32538, 32752, 32966, 33180, 33394, 33608, 33822, 34035, 34248, 34461, 34674, 34887, 35100, 35313
	35525, 35737, 35949, 36161, 36373, 36584, 36795, 37006, 37217, 37427, 37637, 37847, 38056, 38265
	38474, 38682, 38890, 39097, 39304, 39510, 39716, 39922, 40127, 40332, 40536,

Index	Probability
	40740, 40943, 41146
	41348, 41549, 41750, 41950, 42150, 42349, 42547, 42745, 42942, 43139, 43335, 43530, 43725, 43919
	44112, 44304, 44496, 44687, 44877, 45066, 45255, 45443, 45630, 45816, 46002, 46187, 46371, 46554
	46736, 46917, 47098, 47278, 47457, 47635, 47812, 47988, 48163, 48337, 48510, 48683, 48855, 49026
	49196, 49365, 49533, 49700, 49866, 50031, 50195, 50358, 50520, 50681, 50841, 51000, 51158, 51315
	51471, 51626, 51780, 51933, 52085, 52236, 52386, 52535, 52683, 52830, 52976, 53121, 53265, 53408
	53550, 53691, 53830, 53968, 54105, 54241, 54376, 54510, 54643, 54775, 54906, 55036, 55165, 55293
	55420, 55546, 55671, 55795, 55918, 56040, 56160, 56279, 56397, 56514, 56630, 56745, 56859, 56972
	57084, 57195, 57305, 57414, 57522, 57629, 57735, 57840, 57944, 58047, 58149, 58250, 58350, 58449
	58547, 58644, 58740, 58835, 58929, 59022, 59114, 59205, 59295, 59384, 59472, 59559, 59645, 59730
	59814, 59897, 59979, 60060, 60140, 60220, 60299, 60377, 60454, 60530, 60605, 60679, 60752, 60824
	60896, 60967, 61037, 61106, 61174, 61241, 61307, 61373, 61438, 61502, 61565, 61627, 61689, 61750
	61810, 61869, 61927, 61985, 62042, 62098, 62153, 62208, 62262, 62315, 62367, 62419, 62470, 62520
	62570, 62619, 62667, 62715, 62762, 62808, 62854, 62899, 62943, 62987, 63030, 63072, 63114, 63155
	63196, 63236, 63275, 63314, 63352, 63390, 63427, 63464, 63500, 63535, 63570, 63604, 63638, 63671
	63704, 63736, 63768, 63799, 63830, 63860, 63890, 63919, 63948, 63976, 64004, 64031, 64058, 64084
	64110, 64136, 64161, 64186, 64210, 64234, 64258, 64281, 64304, 64326, 64348, 64370, 64391, 64412
	64432, 64452, 64472, 64491, 64510, 64529, 64547, 64565, 64583, 64600, 64617, 64634, 64650, 64666
	64682, 64698, 64713, 64728, 64743, 64757, 64771, 64785, 64799, 64812, 64825, 64838, 64851, 64863
	64875, 64887, 64899, 64910, 64921, 64932, 64943, 64954, 64964, 64974, 64984, 64994, 65004, 65013
	65022, 65031, 65040, 65049, 65057, 65065, 65073, 65081, 65089, 65097, 65104, 65111, 65118, 65125
	65132, 65139, 65146, 65152, 65158, 65164, 65170, 65176, 65182, 65188, 65194, 65199, 65204, 65209
	65214, 65219, 65224, 65229, 65234, 65238, 65242, 65246, 65250, 65254, 65258, 65262, 65266, 65270
	65274, 65278, 65281, 65284, 65287, 65290, 65536

Table B.9 (continued)

Index	Probability
59	0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 34, 38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78, 83, 88
	93, 98, 103, 108, 113, 118, 123, 128, 134, 140, 146, 152, 158, 164, 170, 176, 182, 189, 196, 203
	210, 217, 224, 231, 238, 246, 254, 262, 270, 278, 286, 294, 303, 312, 321, 330, 339, 348, 358
	368, 378, 388, 398, 408, 419, 430, 441, 452, 463, 475, 487, 499, 511, 523, 536, 549, 562, 575
	588, 602, 616, 630, 644, 659, 674, 689, 704, 720, 736, 752, 768, 785, 802, 819, 836, 854, 872
	890, 909, 928, 947, 966, 986, 1006, 1026, 1047, 1068, 1089, 1111, 1133, 1155, 1178, 1201, 1224
	1248, 1272, 1296, 1321, 1346, 1371, 1397, 1423, 1450, 1477, 1504, 1532, 1560, 1589, 1618, 1647
	1677, 1707, 1738, 1769, 1800, 1832, 1864, 1897, 1930, 1964, 1998, 2033, 2068, 2104, 2140, 2176
	2213, 2250, 2288, 2326, 2365, 2404, 2444, 2484, 2525, 2566, 2608, 2651, 2694, 2738, 2782, 2827
	2872, 2918, 2964, 3011, 3058, 3106, 3155, 3204, 3254, 3304, 3355, 3407, 3459, 3512, 3565, 3619
	3674, 3729, 3785, 3842, 3899, 3957, 4015, 4074, 4134, 4194, 4255, 4317, 4379, 4442, 4506, 4570
	4635, 4701, 4767, 4834, 4902, 4971, 5040, 5110, 5181, 5252, 5324, 5397, 5471, 5545, 5620, 5696
	5773, 5850, 5928, 6007, 6087, 6167, 6248, 6330, 6413, 6496, 6580, 6665, 6751, 6837, 6924, 7012
	7101, 7191, 7281, 7372, 7464, 7557, 7651, 7745, 7840, 7936, 8033, 8131, 8229, 8328, 8428, 8529
	8631, 8734, 8837, 8941, 9046, 9152, 9259, 9366, 9474, 9583, 9693, 9804, 9916, 10028, 10141
	10255, 10370, 10486, 10603, 10720, 10838, 10957, 11077, 11198, 11320, 11442, 11565, 11689, 11814
	11940, 12066, 12193, 12321, 12450, 12580, 12711, 12842, 12974, 13107, 13241, 13376, 13511, 13647
	13784, 13922, 14061, 14200, 14340, 14481, 14623, 14765, 14908, 15052, 15197, 15343, 15489, 15636
	15784, 15933, 16082, 16232, 16383, 16534, 16686, 16839, 16993, 17147, 17302, 17458, 17614, 17771
	17929, 18088, 18247, 18407, 18568, 18729, 18891, 19053, 19216, 19380, 19544, 19709, 19875, 20041
	20208, 20375, 20543, 20712, 20881, 21051, 21221, 21392, 21563, 21735, 21908, 22081, 22255, 22429
	22604, 22779, 22955, 23131, 23308, 23485, 23662, 23840, 24018, 24197, 24376, 24556, 24736, 24916

Index	Probability
	25097, 25278, 25460, 25642, 25824, 26007, 26190, 26373, 26557, 26741, 26925, 27110, 27295, 27480
	27665, 27851, 28037, 28223, 28409, 28596, 28783, 28970, 29157, 29344, 29532, 29720, 29908, 30096
	30284, 30472, 30660, 30848, 31037, 31226, 31415, 31604, 31793, 31982, 32171, 32360, 32549, 32738
	32927, 33116, 33305, 33494, 33683, 33872, 34061, 34250, 34439, 34627, 34815, 35003, 35191, 35379
	35567, 35755, 35943, 36130, 36317, 36504, 36691, 36878, 37064, 37250, 37436, 37622, 37807, 37992
	38177, 38362, 38546, 38730, 38914, 39097, 39280, 39463, 39645, 39827, 40009, 40190, 40371, 40551
	40731, 40911, 41090, 41269, 41447, 41625, 41802, 41979, 42156, 42332, 42508, 42683, 42858, 43032
	43206, 43379, 43552, 43724, 43895, 44066, 44236, 44406, 44575, 44744, 44912, 45079, 45246, 45412
	45578, 45743, 45907, 46071, 46234, 46396, 46558, 46719, 46880, 47040, 47199, 47358, 47516, 47673
	47829, 47985, 48140, 48294, 48448, 48601, 48753, 48904, 49055, 49205, 49354, 49503, 49651, 49798
	49944, 50090, 50235, 50379, 50522, 50664, 50806, 50947, 51087, 51226, 51365, 51503, 51640, 51776
	51911, 52046, 52180, 52313, 52445, 52576, 52707, 52837, 52966, 53094, 53221, 53347, 53473, 53598
	53722, 53845, 53967, 54089, 54210, 54330, 54449, 54567, 54684, 54801, 54917, 55032, 55146, 55259
	55371, 55483, 55594, 55704, 55813, 55921, 56028, 56135, 56241, 56346, 56450, 56553, 56656, 56758
	56859, 56959, 57058, 57156, 57254, 57351, 57447, 57542, 57636, 57730, 57823, 57915, 58006, 58096
	58186, 58275, 58363, 58450, 58536, 58622, 58707, 58791, 58874, 58957, 59039, 59120, 59200, 59280
	59359, 59437, 59514, 59591, 59667, 59742, 59816, 59890, 59963, 60035, 60106, 60177, 60247, 60316
	60385, 60453, 60520, 60586, 60652, 60717, 60781, 60845, 60908, 60970, 61032, 61093, 61153, 61213
	61272, 61330, 61388, 61445, 61502, 61558, 61613, 61668, 61722, 61775, 61828, 61880, 61932, 61983
	62033, 62083, 62132, 62181, 62229, 62276, 62323, 62369, 62415, 62460, 62505, 62549, 62593, 62636
	62679, 62721, 62762, 62803, 62843, 62883, 62922, 62961, 62999, 63037, 63074, 63111, 63147, 63183
	63219, 63254, 63289, 63323, 63357, 63390, 63423, 63455, 63487, 63518, 63549, 63580, 63610, 63640
	63669, 63698, 63727, 63755, 63783, 63810, 63837, 63864, 63890, 63916, 63941, 63966, 63991, 64015

Index	Probability
	64039, 64063, 64086, 64109, 64132, 64154, 64176, 64198, 64219, 64240, 64261, 64281, 64301, 64321
	64340, 64359, 64378, 64397, 64415, 64433, 64451, 64468, 64485, 64502, 64519, 64535, 64551, 64567
	64583, 64598, 64613, 64628, 64643, 64657, 64671, 64685, 64699, 64712, 64725, 64738, 64751, 64764
	64776, 64788, 64800, 64812, 64824, 64835, 64846, 64857, 64868, 64879, 64889, 64899, 64909, 64919
	64929, 64939, 64948, 64957, 64966, 64975, 64984, 64993, 65001, 65009, 65017, 65025, 65033, 65041
	65049, 65056, 65063, 65070, 65077, 65084, 65091, 65098, 65105, 65111, 65117, 65123, 65129, 65135
	65141, 65147, 65153, 65159, 65164, 65169, 65174, 65179, 65184, 65189, 65194, 65199, 65204, 65209
	65213, 65217, 65221, 65225, 65229, 65233, 65237, 65241, 65245, 65249, 65253, 65257, 65260, 65263
	65266, 65269, 65272, 65275, 65278, 65281, 65284, 65287, 65536

Table B.9 (continued)

Index	Probability
60	0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 55, 59, 63, 67, 71, 75, 79
	83, 87, 91, 95, 99, 103, 107, 111, 116, 121, 126, 131, 136, 141, 146, 151, 156, 161, 166, 171
	177, 183, 189, 195, 201, 207, 213, 219, 225, 231, 238, 245, 252, 259, 266, 273, 280, 287, 294
	302, 310, 318, 326, 334, 342, 350, 358, 367, 376, 385, 394, 403, 412, 421, 431, 441, 451, 461
	471, 481, 492, 503, 514, 525, 536, 547, 559, 571, 583, 595, 607, 619, 632, 645, 658, 671, 684
	698, 712, 726, 740, 754, 768, 783, 798, 813, 828, 844, 860, 876, 892, 908, 925, 942, 959, 976
	994, 1012, 1030, 1048, 1067, 1086, 1105, 1124, 1144, 1164, 1184, 1204, 1225, 1246, 1267, 1288
	1310, 1332, 1354, 1377, 1400, 1423, 1446, 1470, 1494, 1518, 1543, 1568, 1593, 1619, 1645, 1671
	1698, 1725, 1752, 1780, 1808, 1836, 1865, 1894, 1923, 1953, 1983, 2013, 2044, 2075, 2106, 2138
	2170, 2203, 2236, 2269, 2303, 2337, 2371, 2406, 2441, 2477, 2513, 2549, 2586, 2623, 2661, 2699
	2738, 2777, 2816, 2856, 2896, 2937, 2978, 3020, 3062, 3104, 3147, 3190, 3234, 3278, 3323, 3368
	3414, 3460, 3507, 3554, 3602, 3650, 3699, 3748, 3798, 3848, 3899, 3950, 4002, 4054, 4107, 4160

Index	Probability
	4214, 4268, 4323, 4378, 4434, 4490, 4547, 4604, 4662, 4720, 4779, 4838, 4898, 4959, 5020, 5082
	5144, 5207, 5270, 5334, 5398, 5463, 5529, 5595, 5662, 5729, 5797, 5865, 5934, 6004, 6074, 6145
	6216, 6288, 6361, 6434, 6508, 6582, 6657, 6733, 6809, 6886, 6963, 7041, 7120, 7199, 7279, 7360
	7441, 7523, 7605, 7688, 7772, 7856, 7941, 8027, 8113, 8200, 8287, 8375, 8464, 8553, 8643, 8734
	8825, 8917, 9010, 9103, 9197, 9292, 9387, 9483, 9579, 9676, 9774, 9872, 9971, 10071, 10171
	10272, 10374, 10476, 10579, 10683, 10787, 10892, 10998, 11104, 11211, 11318, 11426, 11535, 11644
	11754, 11865, 11976, 12088, 12201, 12314, 12428, 12543, 12658, 12774, 12890, 13007, 13125, 13243
	13362, 13482, 13602, 13723, 13844, 13966, 14089, 14212, 14336, 14460, 14585, 14711, 14837, 14964
	15092, 15220, 15349, 15478, 15608, 15738, 15869, 16001, 16133, 16266, 16399, 16533, 16668, 16803
	16939, 17075, 17212, 17349, 17487, 17625, 17764, 17903, 18043, 18184, 18325, 18467, 18609, 18752
	18895, 19039, 19183, 19328, 19473, 19619, 19765, 19912, 20059, 20207, 20355, 20503, 20652, 20801
	20951, 21101, 21252, 21403, 21555, 21707, 21859, 22012, 22165, 22319, 22473, 22627, 22782, 22937
	23093, 23249, 23405, 23562, 23719, 23876, 24034, 24192, 24350, 24509, 24668, 24827, 24987, 25147
	25307, 25468, 25629, 25790, 25951, 26113, 26275, 26437, 26599, 26762, 26925, 27088, 27251, 27415
	27579, 27743, 27907, 28071, 28236, 28401, 28566, 28731, 28896, 29061, 29227, 29393, 29559, 29725
	29891, 30057, 30223, 30389, 30555, 30722, 30889, 31056, 31223, 31390, 31557, 31724, 31891, 32058
	32225, 32392, 32559, 32726, 32893, 33060, 33227, 33394, 33561, 33728, 33895, 34062, 34229, 34396
	34563, 34730, 34896, 35062, 35228, 35394, 35560, 35726, 35892, 36058, 36224, 36389, 36554, 36719
	36884, 37049, 37214, 37378, 37542, 37706, 37870, 38034, 38197, 38360, 38523, 38686, 38848, 39010
	39172, 39334, 39495, 39656, 39817, 39978, 40138, 40298, 40458, 40617, 40776, 40935, 41093, 41251
	41409, 41566, 41723, 41880, 42036, 42192, 42348, 42503, 42658, 42812, 42966, 43120, 43273, 43426
	43578, 43730, 43882, 44033, 44184, 44334, 44484, 44633, 44782, 44930, 45078, 45226, 45373, 45520
	45666, 45812, 45957, 46102, 46246, 46390, 46533, 46676, 46818, 46960, 47101, 47242, 47382, 47521

Index	Probability
	47660, 47798, 47936, 48073, 48210, 48346, 48482, 48617, 48752, 48886, 49019, 49152, 49284, 49416
	49547, 49677, 49807, 49936, 50065, 50193, 50321, 50448, 50574, 50700, 50825, 50949, 51073, 51196
	51319, 51441, 51562, 51683, 51803, 51923, 52042, 52160, 52278, 52395, 52511, 52627, 52742, 52857
	52971, 53084, 53197, 53309, 53420, 53531, 53641, 53750, 53859, 53967, 54074, 54181, 54287, 54393
	54498, 54602, 54706, 54809, 54911, 55013, 55114, 55214, 55314, 55413, 55511, 55609, 55706, 55802
	55898, 55993, 56088, 56182, 56275, 56368, 56460, 56551, 56642, 56732, 56821, 56910, 56998, 57085
	57172, 57258, 57344, 57429, 57513, 57597, 57680, 57762, 57844, 57925, 58006, 58086, 58165, 58244

Table B.9 (continued)

Index	Probability
60	58322, 58399, 58476, 58552, 58628, 58703, 58777, 58851, 58924, 58997, 59069, 59140, 59211, 59281
	59351, 59420, 59488, 59556, 59623, 59690, 59756, 59822, 59887, 59951, 60015, 60078, 60141, 60203
	60265, 60326, 60387, 60447, 60506, 60565, 60623, 60681, 60738, 60795, 60851, 60907, 60962, 61017
	61071, 61125, 61178, 61231, 61283, 61335, 61386, 61437, 61487, 61537, 61586, 61635, 61683, 61731
	61778, 61825, 61871, 61917, 61962, 62007, 62051, 62095, 62138, 62181, 62223, 62265, 62307, 62348
	62389, 62429, 62469, 62508, 62547, 62586, 62624, 62662, 62699, 62736, 62772, 62808, 62844, 62879
	62914, 62948, 62982, 63016, 63049, 63082, 63115, 63147, 63179, 63210, 63241, 63272, 63302, 63332
	63362, 63391, 63420, 63449, 63477, 63505, 63533, 63560, 63587, 63614, 63640, 63666, 63692, 63717
	63742, 63767, 63791, 63815, 63839, 63862, 63885, 63908, 63931, 63953, 63975, 63997, 64018, 64039
	64060, 64081, 64101, 64121, 64141, 64161, 64180, 64199, 64218, 64237, 64255, 64273, 64291, 64309
	64326, 64343, 64360, 64377, 64393, 64409, 64425, 64441, 64457, 64472, 64487, 64502, 64517, 64531
	64545, 64559, 64573, 64587, 64601, 64614, 64627, 64640, 64653, 64666, 64678, 64690, 64702, 64714
	64726, 64738, 64749, 64760, 64771, 64782, 64793, 64804, 64814, 64824, 64834, 64844, 64854, 64864
	64873, 64882, 64891, 64900, 64909, 64918, 64927, 64935, 64943, 64951, 64959,

Index	Probability
	64967, 64975, 64983
	64991, 64998, 65005, 65012, 65019, 65026, 65033, 65040, 65047, 65054, 65060, 65066, 65072, 65078
	65084, 65090, 65096, 65102, 65108, 65114, 65119, 65124, 65129, 65134, 65139, 65144, 65149, 65154
	65159, 65164, 65169, 65174, 65178, 65182, 65186, 65190, 65194, 65198, 65202, 65206, 65210, 65214
	65218, 65222, 65226, 65230, 65234, 65237, 65240, 65243, 65246, 65249, 65252, 65255, 65258, 65261
	65264, 65267, 65270, 65273, 65276, 65279, 65282, 65285, 65536
61	0, 2, 4, 6, 8, 10, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66
	69, 72, 75, 79, 83, 87, 91, 95, 99, 103, 107, 111, 115, 119, 123, 127, 131, 135, 139, 143, 148
	153, 158, 163, 168, 173, 178, 183, 188, 193, 198, 203, 208, 214, 220, 226, 232, 238, 244, 250
	256, 262, 268, 274, 280, 287, 294, 301, 308, 315, 322, 329, 336, 343, 350, 358, 366, 374, 382
	390, 398, 406, 414, 422, 430, 439, 448, 457, 466, 475, 484, 493, 502, 512, 522, 532, 542, 552
	562, 572, 582, 593, 604, 615, 626, 637, 648, 659, 671, 683, 695, 707, 719, 731, 744, 757, 770
	783, 796, 809, 823, 837, 851, 865, 879, 893, 908, 923, 938, 953, 968, 983, 999, 1015, 1031, 1047
	1063, 1080, 1097, 1114, 1131, 1148, 1166, 1184, 1202, 1220, 1238, 1257, 1276, 1295, 1314, 1334
	1354, 1374, 1394, 1414, 1435, 1456, 1477, 1498, 1520, 1542, 1564, 1586, 1609, 1632, 1655, 1678
	1702, 1726, 1750, 1774, 1799, 1824, 1849, 1874, 1900, 1926, 1952, 1979, 2006, 2033, 2060, 2088
	2116, 2144, 2173, 2202, 2231, 2260, 2290, 2320, 2350, 2381, 2412, 2443, 2475, 2507, 2539, 2571
	2604, 2637, 2670, 2704, 2738, 2772, 2807, 2842, 2877, 2913, 2949, 2985, 3022, 3059, 3097, 3135
	3173, 3212, 3251, 3290, 3330, 3370, 3410, 3451, 3492, 3533, 3575, 3617, 3660, 3703, 3746, 3790
	3834, 3879, 3924, 3969, 4015, 4061, 4108, 4155, 4202, 4250, 4298, 4347, 4396, 4445, 4495, 4545
	4596, 4647, 4698, 4750, 4802, 4855, 4908, 4962, 5016, 5070, 5125, 5180, 5236, 5292, 5349, 5406
	5464, 5522, 5580, 5639, 5698, 5758, 5818, 5879, 5940, 6002, 6064, 6127, 6190, 6253, 6317, 6381
	6446, 6511, 6577, 6643, 6710, 6777, 6845, 6913, 6982, 7051, 7121, 7191, 7262, 7333, 7405, 7477
	7550, 7623, 7697, 7771, 7846, 7921, 7997, 8073, 8150, 8227, 8304, 8382, 8460, 8539, 8618, 8698

Index	Probability
	8778, 8859, 8940, 9022, 9104, 9187, 9271, 9355, 9440, 9525, 9611, 9697, 9784, 9871, 9959, 10047
	10136, 10225, 10315, 10405, 10496, 10587, 10679, 10771, 10864, 10957, 11051, 11145, 11240, 11335
	11431, 11527, 11624, 11721, 11819, 11917, 12016, 12115, 12215, 12315, 12416, 12517, 12619, 12721
	12824, 12927, 13031, 13135, 13240, 13345, 13451, 13557, 13664, 13771, 13879, 13987, 14095, 14204

Table B.9 (continued)

Index	Probability
61	14313, 14423, 14533, 14644, 14755, 14867, 14979, 15092, 15205, 15319, 15433, 15548, 15663, 15779
	15895, 16011, 16128, 16245, 16363, 16481, 16600, 16719, 16839, 16959, 17079, 17200, 17321, 17443
	17565, 17688, 17811, 17934, 18058, 18182, 18307, 18432, 18558, 18684, 18810, 18937, 19064, 19191
	19319, 19447, 19576, 19705, 19834, 19964, 20094, 20225, 20356, 20487, 20619, 20751, 20883, 21016
	21149, 21282, 21416, 21550, 21684, 21819, 21954, 22089, 22225, 22361, 22497, 22634, 22771, 22908
	23045, 23183, 23321, 23459, 23598, 23737, 23876, 24015, 24155, 24295, 24435, 24575, 24716, 24857
	24998, 25139, 25281, 25423, 25565, 25707, 25850, 25993, 26136, 26279, 26422, 26566, 26710, 26854
	26998, 27142, 27287, 27432, 27577, 27722, 27867, 28012, 28157, 28303, 28449, 28595, 28741, 28887
	29033, 29179, 29325, 29472, 29619, 29766, 29913, 30060, 30207, 30354, 30501, 30648, 30795, 30942
	31089, 31236, 31384, 31532, 31680, 31828, 31976, 32124, 32272, 32420, 32568, 32716, 32864, 33012
	33160, 33308, 33456, 33604, 33752, 33900, 34048, 34195, 34342, 34489, 34636, 34783, 34930, 35077
	35224, 35371, 35518, 35665, 35812, 35959, 36105, 36251, 36397, 36543, 36689, 36835, 36981, 37127
	37272, 37417, 37562, 37707, 37852, 37997, 38142, 38286, 38430, 38574, 38718, 38862, 39005, 39148
	39291, 39434, 39577, 39719, 39861, 40003, 40145, 40286, 40427, 40568, 40709, 40849, 40989, 41129
	41269, 41408, 41547, 41686, 41825, 41963, 42101, 42239, 42376, 42513, 42650, 42787, 42923, 43059
	43195, 43330, 43465, 43600, 43734, 43868, 44002, 44135, 44268, 44401, 44533, 44665, 44797, 44928
	45059, 45190, 45320, 45450, 45579, 45708, 45837, 45965, 46093, 46220, 46347,

Index	Probability
	46474, 46600, 46726
	46852, 46977, 47102, 47226, 47350, 47473, 47596, 47719, 47841, 47963, 48084, 48205, 48325, 48445
	48565, 48684, 48803, 48921, 49039, 49156, 49273, 49389, 49505, 49621, 49736, 49851, 49965, 50079
	50192, 50305, 50417, 50529, 50640, 50751, 50861, 50971, 51080, 51189, 51297, 51405, 51513, 51620
	51727, 51833, 51939, 52044, 52149, 52253, 52357, 52460, 52563, 52665, 52767, 52868, 52969, 53069
	53169, 53268, 53367, 53465, 53563, 53660, 53757, 53853, 53949, 54044, 54139, 54233, 54327, 54420
	54513, 54605, 54697, 54788, 54879, 54969, 55059, 55148, 55237, 55325, 55413, 55500, 55587, 55673
	55759, 55844, 55929, 56013, 56097, 56180, 56262, 56344, 56425, 56506, 56586, 56666, 56745, 56824
	56902, 56980, 57057, 57134, 57211, 57287, 57363, 57438, 57513, 57587, 57661, 57734, 57807, 57879
	57951, 58022, 58093, 58163, 58233, 58302, 58371, 58439, 58507, 58574, 58641, 58707, 58773, 58838
	58903, 58967, 59031, 59094, 59157, 59220, 59282, 59344, 59405, 59466, 59526, 59586, 59645, 59704
	59762, 59820, 59878, 59935, 59992, 60048, 60104, 60159, 60214, 60268, 60322, 60376, 60429, 60482
	60534, 60586, 60637, 60688, 60739, 60789, 60839, 60888, 60937, 60986, 61034, 61082, 61129, 61176
	61223, 61269, 61315, 61360, 61405, 61450, 61494, 61538, 61581, 61624, 61667, 61709, 61751, 61792
	61833, 61874, 61914, 61954, 61994, 62033, 62072, 62111, 62149, 62187, 62225, 62262, 62299, 62335
	62371, 62407, 62442, 62477, 62512, 62546, 62580, 62614, 62647, 62680, 62713, 62745, 62777, 62809
	62841, 62872, 62903, 62934, 62964, 62994, 63024, 63053, 63082, 63111, 63140, 63168, 63196, 63224
	63251, 63278, 63305, 63332, 63358, 63384, 63410, 63435, 63460, 63485, 63510, 63534, 63558, 63582
	63606, 63629, 63652, 63675, 63698, 63720, 63742, 63764, 63786, 63807, 63828, 63849, 63870, 63890
	63910, 63930, 63950, 63970, 63989, 64008, 64027, 64046, 64064, 64082, 64100, 64118, 64136, 64153
	64170, 64187, 64204, 64221, 64237, 64253, 64269, 64285, 64301, 64316, 64331, 64346, 64361, 64376
	64391, 64405, 64419, 64433, 64447, 64461, 64475, 64488, 64501, 64514, 64527, 64540, 64553, 64565
	64577, 64589, 64601, 64613, 64625, 64636, 64647, 64658, 64669, 64680, 64691, 64702, 64712, 64722
	64732, 64742, 64752, 64762, 64772, 64782, 64791, 64800, 64809, 64818, 64827,

Index	Probability
	64836, 64845, 64854
	64862, 64870, 64878, 64886, 64894, 64902, 64910, 64918, 64926, 64934, 64941, 64948, 64955, 64962
	64969, 64976, 64983, 64990, 64997, 65004, 65010, 65016, 65022, 65028, 65034, 65040, 65046, 65052

Table B.9 (continued)

Index	Probability
61	65058, 65064, 65070, 65076, 65081, 65086, 65091, 65096, 65101, 65106, 65111, 65116, 65121, 65126
	65131, 65136, 65141, 65145, 65149, 65153, 65157, 65161, 65165, 65169, 65173, 65177, 65181, 65185
	65189, 65193, 65197, 65201, 65205, 65209, 65212, 65215, 65218, 65221, 65224, 65227, 65230, 65233
	65236, 65239, 65242, 65245, 65248, 65251, 65254, 65257, 65260, 65263, 65266, 65269, 65272, 65274
	65276, 65278, 65280, 65282, 65284, 65536
62	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 35, 38, 41, 44, 47, 50, 53, 56
	59, 62, 65, 68, 71, 74, 77, 80, 83, 86, 89, 92, 95, 98, 101, 104, 108, 112, 116, 120, 124, 128
	132, 136, 140, 144, 148, 152, 156, 160, 164, 168, 172, 176, 180, 185, 190, 195, 200, 205, 210
	215, 220, 225, 230, 235, 240, 245, 250, 255, 260, 266, 272, 278, 284, 290, 296, 302, 308, 314
	320, 326, 332, 338, 345, 352, 359, 366, 373, 380, 387, 394, 401, 408, 415, 422, 430, 438, 446
	454, 462, 470, 478, 486, 494, 502, 510, 519, 528, 537, 546, 555, 564, 573, 582, 591, 601, 611
	621, 631, 641, 651, 661, 671, 681, 692, 703, 714, 725, 736, 747, 758, 769, 781, 793, 805, 817
	829, 841, 853, 865, 878, 891, 904, 917, 930, 943, 956, 970, 984, 998, 1012, 1026, 1040, 1054
	1069, 1084, 1099, 1114, 1129, 1144, 1160, 1176, 1192, 1208, 1224, 1240, 1257, 1274, 1291, 1308
	1325, 1342, 1360, 1378, 1396, 1414, 1432, 1450, 1469, 1488, 1507, 1526, 1545, 1565, 1585, 1605
	1625, 1645, 1666, 1687, 1708, 1729, 1750, 1772, 1794, 1816, 1838, 1860, 1883, 1906, 1929, 1952
	1975, 1999, 2023, 2047, 2071, 2096, 2121, 2146, 2171, 2196, 2222, 2248, 2274, 2300, 2327, 2354
	2381, 2408, 2436, 2464, 2492, 2520, 2549, 2578, 2607, 2636, 2666, 2696, 2726, 2756, 2787, 2818
	2849, 2880, 2912, 2944, 2976, 3008, 3041, 3074, 3107, 3141, 3175, 3209, 3243, 3278,

Index	Probability
	3313, 3348
	3384, 3420, 3456, 3492, 3529, 3566, 3603, 3641, 3679, 3717, 3755, 3794, 3833, 3872, 3912, 3952
	3992, 4033, 4074, 4115, 4157, 4199, 4241, 4283, 4326, 4369, 4412, 4456, 4500, 4544, 4589, 4634
	4679, 4725, 4771, 4817, 4864, 4911, 4958, 5006, 5054, 5102, 5151, 5200, 5249, 5299, 5349, 5399
	5450, 5501, 5553, 5605, 5657, 5710, 5763, 5816, 5870, 5924, 5978, 6033, 6088, 6143, 6199, 6255
	6312, 6369, 6426, 6484, 6542, 6600, 6659, 6718, 6778, 6838, 6898, 6959, 7020, 7081, 7143, 7205
	7268, 7331, 7394, 7458, 7522, 7586, 7651, 7716, 7782, 7848, 7914, 7981, 8048, 8115, 8183, 8251
	8320, 8389, 8458, 8528, 8598, 8669, 8740, 8811, 8883, 8955, 9028, 9101, 9174, 9248, 9322, 9397
	9472, 9547, 9623, 9699, 9776, 9853, 9930, 10008, 10086, 10164, 10243, 10322, 10402, 10482, 10562
	10643, 10724, 10806, 10888, 10970, 11053, 11136, 11220, 11304, 11388, 11473, 11558, 11644, 11730
	11816, 11903, 11990, 12078, 12166, 12254, 12343, 12432, 12522, 12612, 12702, 12793, 12884, 12975
	13067, 13159, 13252, 13345, 13438, 13532, 13626, 13721, 13816, 13911, 14007, 14103, 14199, 14296
	14393, 14491, 14589, 14687, 14786, 14885, 14984, 15084, 15184, 15284, 15385, 15486, 15588, 15690
	15792, 15895, 15998, 16101, 16205, 16309, 16413, 16518, 16623, 16728, 16834, 16940, 17046, 17153
	17260, 17367, 17475, 17583, 17691, 17800, 17909, 18018, 18128, 18238, 18348, 18459, 18570, 18681
	18793, 18905, 19017, 19130, 19243, 19356, 19470, 19584, 19698, 19812, 19927, 20042, 20157, 20273
	20389, 20505, 20621, 20738, 20855, 20972, 21090, 21208, 21326, 21444, 21563, 21682, 21801, 21920
	22040, 22160, 22280, 22400, 22521, 22642, 22763, 22884, 23006, 23128, 23250, 23372, 23494, 23617
	23740, 23863, 23986, 24110, 24234, 24358, 24482, 24606, 24731, 24856, 24981, 25106, 25231, 25356
	25482, 25608, 25734, 25860, 25986, 26112, 26239, 26366, 26493, 26620, 26747, 26874, 27002, 27130
	27258, 27386, 27514, 27642, 27770, 27898, 28027, 28156, 28285, 28414, 28543, 28672, 28801, 28930
	29059, 29188, 29318, 29448, 29578, 29708, 29838, 29968, 30098, 30228, 30358, 30488, 30618, 30748
	30878, 31008, 31138, 31268, 31398, 31528, 31659, 31790, 31921, 32052, 32183, 32314, 32445, 32576

Table B.9 (continued)

Index	Probability
62	32707, 32838, 32969, 33100, 33231, 33362, 33493, 33624, 33755, 33885, 34015, 34145, 34275, 34405
	34535, 34665, 34795, 34925, 35055, 35185, 35315, 35445, 35575, 35705, 35835, 35965, 36095, 36224
	36353, 36482, 36611, 36740, 36869, 36998, 37127, 37256, 37385, 37513, 37641, 37769, 37897, 38025
	38153, 38281, 38409, 38536, 38663, 38790, 38917, 39044, 39171, 39297, 39423, 39549, 39675, 39801
	39927, 40052, 40177, 40302, 40427, 40552, 40677, 40801, 40925, 41049, 41173, 41297, 41420, 41543
	41666, 41789, 41911, 42033, 42155, 42277, 42399, 42520, 42641, 42762, 42883, 43003, 43123, 43243
	43363, 43482, 43601, 43720, 43839, 43957, 44075, 44193, 44311, 44428, 44545, 44662, 44778, 44894
	45010, 45126, 45241, 45356, 45471, 45585, 45699, 45813, 45927, 46040, 46153, 46266, 46378, 46490
	46602, 46713, 46824, 46935, 47045, 47155, 47265, 47374, 47483, 47592, 47700, 47808, 47916, 48023
	48130, 48237, 48343, 48449, 48555, 48660, 48765, 48870, 48974, 49078, 49182, 49285, 49388, 49491
	49593, 49695, 49797, 49898, 49999, 50099, 50199, 50299, 50398, 50497, 50596, 50694, 50792, 50890
	50987, 51084, 51180, 51276, 51372, 51467, 51562, 51657, 51751, 51845, 51938, 52031, 52124, 52216
	52308, 52399, 52490, 52581, 52671, 52761, 52851, 52940, 53029, 53117, 53205, 53293, 53380, 53467
	53553, 53639, 53725, 53810, 53895, 53979, 54063, 54147, 54230, 54313, 54395, 54477, 54559, 54640
	54721, 54801, 54881, 54961, 55040, 55119, 55197, 55275, 55353, 55430, 55507, 55584, 55660, 55736
	55811, 55886, 55961, 56035, 56109, 56182, 56255, 56328, 56400, 56472, 56543, 56614, 56685, 56755
	56825, 56894, 56963, 57032, 57100, 57168, 57235, 57302, 57369, 57435, 57501, 57567, 57632, 57697
	57761, 57825, 57889, 57952, 58015, 58078, 58140, 58202, 58263, 58324, 58385, 58445, 58505, 58565
	58624, 58683, 58741, 58799, 58857, 58914, 58971, 59028, 59084, 59140, 59195, 59250, 59305, 59359
	59413, 59467, 59520, 59573, 59626, 59678, 59730, 59782, 59833, 59884, 59934, 59984, 60034, 60083
	60132, 60181, 60229, 60277, 60325, 60372, 60419, 60466, 60512, 60558, 60604, 60649, 60694, 60739
	60783, 60827, 60871, 60914, 60957, 61000, 61042, 61084, 61126, 61168, 61209,

Index	Probability
	61250, 61291, 61331
	61371, 61411, 61450, 61489, 61528, 61566, 61604, 61642, 61680, 61717, 61754, 61791, 61827, 61863
	61899, 61935, 61970, 62005, 62040, 62074, 62108, 62142, 62176, 62209, 62242, 62275, 62307, 62339
	62371, 62403, 62434, 62465, 62496, 62527, 62557, 62587, 62617, 62647, 62676, 62705, 62734, 62763
	62791, 62819, 62847, 62875, 62902, 62929, 62956, 62983, 63009, 63035, 63061, 63087, 63112, 63137
	63162, 63187, 63212, 63236, 63260, 63284, 63308, 63331, 63354, 63377, 63400, 63423, 63445, 63467
	63489, 63511, 63533, 63554, 63575, 63596, 63617, 63638, 63658, 63678, 63698, 63718, 63738, 63757
	63776, 63795, 63814, 63833, 63851, 63869, 63887, 63905, 63923, 63941, 63958, 63975, 63992, 64009
	64026, 64043, 64059, 64075, 64091, 64107, 64123, 64139, 64154, 64169, 64184, 64199, 64214, 64229
	64243, 64257, 64271, 64285, 64299, 64313, 64327, 64340, 64353, 64366, 64379, 64392, 64405, 64418
	64430, 64442, 64454, 64466, 64478, 64490, 64502, 64514, 64525, 64536, 64547, 64558, 64569, 64580
	64591, 64602, 64612, 64622, 64632, 64642, 64652, 64662, 64672, 64682, 64692, 64701, 64710, 64719
	64728, 64737, 64746, 64755, 64764, 64773, 64781, 64789, 64797, 64805, 64813, 64821, 64829, 64837
	64845, 64853, 64861, 64868, 64875, 64882, 64889, 64896, 64903, 64910, 64917, 64924, 64931, 64938
	64945, 64951, 64957, 64963, 64969, 64975, 64981, 64987, 64993, 64999, 65005, 65011, 65017, 65023
	65028, 65033, 65038, 65043, 65048, 65053, 65058, 65063, 65068, 65073, 65078, 65083, 65088, 65093
	65098, 65103, 65107, 65111, 65115, 65119, 65123, 65127, 65131, 65135, 65139, 65143, 65147, 65151
	65155, 65159, 65163, 65167, 65171, 65175, 65179, 65182, 65185, 65188, 65191, 65194, 65197, 65200
	65203, 65206, 65209, 65212, 65215, 65218, 65221, 65224, 65227, 65230, 65233, 65236, 65239, 65242
	65245, 65248, 65251, 65254, 65256, 65258, 65260, 65262, 65264, 65266, 65268, 65270, 65272, 65274
	65276, 65278, 65280, 65282, 65284, 65536

Table B.9 (continued)

Index	Probability
63	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 50, 52, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105, 108, 111, 114 117, 120, 123, 126, 129, 132, 135, 138, 142, 146, 150, 154, 158, 162, 166, 170, 174, 178, 182 186, 190, 194, 198, 202, 206, 210, 214, 218, 222, 226, 231, 236, 241, 246, 251, 256, 261, 266 271, 276, 281, 286, 291, 296, 301, 306, 311, 316, 321, 327, 333, 339, 345, 351, 357, 363, 369 375, 381, 387, 393, 399, 405, 411, 418, 425, 432, 439, 446, 453, 460, 467, 474, 481, 488, 495 502, 509, 517, 525, 533, 541, 549, 557, 565, 573, 581, 589, 597, 605, 614, 623, 632, 641, 650 659, 668, 677, 686, 695, 704, 714, 724, 734, 744, 754, 764, 774, 784, 794, 804, 814, 825, 836 847, 858, 869, 880, 891, 902, 913, 925, 937, 949, 961, 973, 985, 997, 1009, 1021, 1034, 1047 1060, 1073, 1086, 1099, 1112, 1125, 1139, 1153, 1167, 1181, 1195, 1209, 1223, 1237, 1252, 1267 1282, 1297, 1312, 1327, 1342, 1357, 1373, 1389, 1405, 1421, 1437, 1453, 1469, 1486, 1503, 1520 1537, 1554, 1571, 1588, 1606, 1624, 1642, 1660, 1678, 1696, 1715, 1734, 1753, 1772, 1791, 1810 1829, 1849, 1869, 1889, 1909, 1929, 1949, 1970, 1991, 2012, 2033, 2054, 2076, 2098, 2120, 2142 2164, 2186, 2209, 2232, 2255, 2278, 2301, 2324, 2348, 2372, 2396, 2420, 2444, 2469, 2494, 2519 2544, 2569, 2595, 2621, 2647, 2673, 2699, 2726, 2753, 2780, 2807, 2834, 2862, 2890, 2918, 2946 2974, 3003, 3032, 3061, 3090, 3120, 3150, 3180, 3210, 3240, 3271, 3302, 3333, 3364, 3395, 3427 3459, 3491, 3523, 3556, 3589, 3622, 3655, 3689, 3723, 3757, 3791, 3825, 3860, 3895, 3930, 3965 4001, 4037, 4073, 4109, 4146, 4183, 4220, 4257, 4295, 4333, 4371, 4409, 4448, 4487, 4526, 4565 4605, 4645, 4685, 4725, 4766, 4807, 4848, 4889, 4931, 4973, 5015, 5058, 5101, 5144, 5187, 5231 5275, 5319, 5363, 5408, 5453, 5498, 5543, 5589, 5635, 5681, 5728, 5775, 5822, 5869, 5917, 5965 6013, 6062, 6111, 6160, 6209, 6259, 6309, 6359, 6410, 6461, 6512, 6563, 6615, 6667, 6719, 6772 6825, 6878, 6931, 6985, 7039, 7093, 7148, 7203, 7258, 7314, 7370, 7426, 7482, 7539, 7596, 7653 7711, 7769, 7827, 7886, 7945, 8004, 8063, 8123, 8183, 8243, 8304, 8365, 8426, 8488, 8550, 8612 8674, 8737, 8800, 8863, 8927, 8991, 9055, 9120, 9185, 9250, 9316, 9382, 9448, 9514,

Index	Probability
	9581, 9648
	9715, 9783, 9851, 9919, 9988, 10057, 10126, 10196, 10266, 10336, 10406, 10477, 10548, 10619
	10691, 10763, 10835, 10908, 10981, 11054, 11128, 11202, 11276, 11350, 11425, 11500, 11575, 11651
	11727, 11803, 11880, 11957, 12034, 12112, 12190, 12268, 12346, 12425, 12504, 12583, 12663, 12743
	12823, 12904, 12985, 13066, 13148, 13230, 13312, 13394, 13477, 13560, 13643, 13727, 13811, 13895
	13979, 14064, 14149, 14234, 14320, 14406, 14492, 14579, 14666, 14753, 14840, 14928, 15016, 15104
	15193, 15282, 15371, 15460, 15550, 15640, 15730, 15821, 15912, 16003, 16094, 16186, 16278, 16370
	16462, 16555, 16648, 16741, 16835, 16929, 17023, 17117, 17212, 17307, 17402, 17497, 17593, 17689
	17785, 17881, 17978, 18075, 18172, 18269, 18367, 18465, 18563, 18661, 18760, 18859, 18958, 19057
	19157, 19257, 19357, 19457, 19558, 19659, 19760, 19861, 19963, 20065, 20167, 20269, 20371, 20474
	20577, 20680, 20783, 20887, 20991, 21095, 21199, 21303, 21408, 21513, 21618, 21723, 21828, 21934
	22040, 22146, 22252, 22358, 22465, 22572, 22679, 22786, 22893, 23000, 23108, 23216, 23324, 23432
	23540, 23649, 23758, 23867, 23976, 24085, 24194, 24303, 24413, 24523, 24633, 24743, 24853, 24963
	25074, 25185, 25296, 25407, 25518, 25629, 25740, 25852, 25964, 26076, 26188, 26300, 26412, 26524
	26636, 26749, 26862, 26975, 27088, 27201, 27314, 27427, 27540, 27653, 27766, 27880, 27994, 28108
	28222, 28336, 28450, 28564, 28678, 28792, 28906, 29020, 29134, 29248, 29363, 29478, 29593, 29708
	29823, 29938, 30053, 30168, 30283, 30398, 30513, 30628, 30743, 30858, 30973, 31088, 31203, 31318
	31433, 31548, 31663, 31778, 31893, 32008, 32123, 32238, 32353, 32469, 32585, 32701, 32817, 32933
	33048, 33163, 33278, 33393, 33508, 33623, 33738, 33853, 33968, 34083, 34198, 34313, 34428, 34543

Table B.9 (continued)

Index	Probability
63	34658, 34773, 34888, 35003, 35118, 35233, 35348, 35463, 35578, 35693, 35808, 35923, 36038, 36152
	36266, 36380, 36494, 36608, 36722, 36836, 36950, 37064, 37178, 37292, 37406,

Index	Probability
	37520, 37633, 37746
	37859, 37972, 38085, 38198, 38311, 38424, 38537, 38649, 38761, 38873, 38985, 39097, 39209, 39321
	39433, 39545, 39656, 39767, 39878, 39989, 40100, 40211, 40322, 40432, 40542, 40652, 40762, 40872
	40982, 41091, 41200, 41309, 41418, 41527, 41636, 41745, 41853, 41961, 42069, 42177, 42285, 42392
	42499, 42606, 42713, 42820, 42927, 43033, 43139, 43245, 43351, 43457, 43562, 43667, 43772, 43877
	43982, 44086, 44190, 44294, 44398, 44502, 44605, 44708, 44811, 44914, 45016, 45118, 45220, 45322
	45424, 45525, 45626, 45727, 45828, 45928, 46028, 46128, 46228, 46327, 46426, 46525, 46624, 46722
	46820, 46918, 47016, 47113, 47210, 47307, 47404, 47500, 47596, 47692, 47788, 47883, 47978, 48073
	48168, 48262, 48356, 48450, 48544, 48637, 48730, 48823, 48915, 49007, 49099, 49191, 49282, 49373
	49464, 49555, 49645, 49735, 49825, 49914, 50003, 50092, 50181, 50269, 50357, 50445, 50532, 50619
	50706, 50793, 50879, 50965, 51051, 51136, 51221, 51306, 51390, 51474, 51558, 51642, 51725, 51808
	51891, 51973, 52055, 52137, 52219, 52300, 52381, 52462, 52542, 52622, 52702, 52781, 52860, 52939
	53017, 53095, 53173, 53251, 53328, 53405, 53482, 53558, 53634, 53710, 53785, 53860, 53935, 54009
	54083, 54157, 54231, 54304, 54377, 54450, 54522, 54594, 54666, 54737, 54808, 54879, 54949, 55019
	55089, 55159, 55228, 55297, 55366, 55434, 55502, 55570, 55637, 55704, 55771, 55837, 55903, 55969
	56035, 56100, 56165, 56230, 56294, 56358, 56422, 56485, 56548, 56611, 56673, 56735, 56797, 56859
	56920, 56981, 57042, 57102, 57162, 57222, 57281, 57340, 57399, 57458, 57516, 57574, 57632, 57689
	57746, 57803, 57859, 57915, 57971, 58027, 58082, 58137, 58192, 58246, 58300, 58354, 58407, 58460
	58513, 58566, 58618, 58670, 58722, 58773, 58824, 58875, 58926, 58976, 59026, 59076, 59125, 59174
	59223, 59272, 59320, 59368, 59416, 59463, 59510, 59557, 59604, 59650, 59696, 59742, 59787, 59832
	59877, 59922, 59966, 60010, 60054, 60098, 60141, 60184, 60227, 60270, 60312, 60354, 60396, 60437
	60478, 60519, 60560, 60600, 60640, 60680, 60720, 60759, 60798, 60837, 60876, 60914, 60952, 60990
	61028, 61065, 61102, 61139, 61176, 61212, 61248, 61284, 61320, 61355, 61390, 61425, 61460, 61494
	61528, 61562, 61596, 61630, 61663, 61696, 61729, 61762, 61794, 61826, 61858,

Index	Probability
	61890, 61921, 61952
	61983, 62014, 62045, 62075, 62105, 62135, 62165, 62195, 62224, 62253, 62282, 62311, 62339, 62367
	62395, 62423, 62451, 62478, 62505, 62532, 62559, 62586, 62612, 62638, 62664, 62690, 62716, 62741
	62766, 62791, 62816, 62841, 62865, 62889, 62913, 62937, 62961, 62984, 63007, 63030, 63053, 63076
	63099, 63121, 63143, 63165, 63187, 63209, 63231, 63252, 63273, 63294, 63315, 63336, 63356, 63376
	63396, 63416, 63436, 63456, 63475, 63494, 63513, 63532, 63551, 63570, 63589, 63607, 63625, 63643
	63661, 63679, 63697, 63714, 63731, 63748, 63765, 63782, 63799, 63816, 63832, 63848, 63864, 63880
	63896, 63912, 63928, 63943, 63958, 63973, 63988, 64003, 64018, 64033, 64048, 64062, 64076, 64090
	64104, 64118, 64132, 64146, 64160, 64173, 64186, 64199, 64212, 64225, 64238, 64251, 64264, 64276
	64288, 64300, 64312, 64324, 64336, 64348, 64360, 64372, 64383, 64394, 64405, 64416, 64427, 64438
	64449, 64460, 64471, 64481, 64491, 64501, 64511, 64521, 64531, 64541, 64551, 64561, 64571, 64581
	64590, 64599, 64608, 64617, 64626, 64635, 64644, 64653, 64662, 64671, 64680, 64688, 64696, 64704
	64712, 64720, 64728, 64736, 64744, 64752, 64760, 64768, 64776, 64783, 64790, 64797, 64804, 64811
	64818, 64825, 64832, 64839, 64846, 64853, 64860, 64867, 64874, 64880, 64886, 64892, 64898, 64904
	64910, 64916, 64922, 64928, 64934, 64940, 64946, 64952, 64958, 64964, 64969, 64974, 64979, 64984
	64989, 64994, 64999, 65004, 65009, 65014, 65019, 65024, 65029, 65034, 65039, 65044, 65049, 65054
	65059, 65063, 65067, 65071, 65075, 65079, 65083, 65087, 65091, 65095, 65099, 65103, 65107, 65111
	65115, 65119, 65123, 65127, 65131, 65135, 65139, 65143, 65147, 65150, 65153, 65156, 65159, 65162

Table B.9 (continued)

Index	Probability
63	65165, 65168, 65171, 65174, 65177, 65180, 65183, 65186, 65189, 65192, 65195, 65198, 65201, 65204
	65207, 65210, 65213, 65216, 65219, 65222, 65225, 65228, 65231, 65233, 65235, 65237, 65239, 65241
	65243, 65245, 65247, 65249, 65251, 65253, 65255, 65257, 65259, 65261, 65263,

Index	Probability
	65265, 65267, 65269 65271, 65273, 65275, 65277, 65279, 65281, 65283, 65285, 65536
64	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 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 85, 88, 91, 94, 97, 100, 103 106, 109, 112, 115, 118, 121, 124, 127, 130, 133, 136, 139, 142, 145, 148, 151, 154, 157, 160 163, 166, 169, 172, 175, 178, 181, 184, 188, 192, 196, 200, 204, 208, 212, 216, 220, 224, 228 232, 236, 240, 244, 248, 252, 256, 260, 264, 268, 272, 276, 280, 284, 289, 294, 299, 304, 309 314, 319, 324, 329, 334, 339, 344, 349, 354, 359, 364, 369, 374, 379, 384, 389, 395, 401, 407 413, 419, 425, 431, 437, 443, 449, 455, 461, 467, 473, 479, 485, 491, 497, 504, 511, 518, 525 532, 539, 546, 553, 560, 567, 574, 581, 588, 595, 602, 610, 618, 626, 634, 642, 650, 658, 666 674, 682, 690, 698, 706, 714, 722, 731, 740, 749, 758, 767, 776, 785, 794, 803, 812, 821, 830 839, 849, 859, 869, 879, 889, 899, 909, 919, 929, 939, 949, 960, 971, 982, 993, 1004, 1015, 1026 1037, 1048, 1059, 1070, 1082, 1094, 1106, 1118, 1130, 1142, 1154, 1166, 1178, 1190, 1202, 1215 1228, 1241, 1254, 1267, 1280, 1293, 1306, 1319, 1333, 1347, 1361, 1375, 1389, 1403, 1417, 1431 1445, 1459, 1474, 1489, 1504, 1519, 1534, 1549, 1564, 1579, 1595, 1611, 1627, 1643, 1659, 1675 1691, 1707, 1723, 1740, 1757, 1774, 1791, 1808, 1825, 1842, 1859, 1877, 1895, 1913, 1931, 1949 1967, 1985, 2004, 2023, 2042, 2061, 2080, 2099, 2118, 2137, 2157, 2177, 2197, 2217, 2237, 2257 2277, 2298, 2319, 2340, 2361, 2382, 2403, 2424, 2446, 2468, 2490, 2512, 2534, 2556, 2579, 2602 2625, 2648, 2671, 2694, 2717, 2741, 2765, 2789, 2813, 2837, 2861, 2886, 2911, 2936, 2961, 2986 3011, 3037, 3063, 3089, 3115, 3141, 3167, 3194, 3221, 3248, 3275, 3302, 3329, 3357, 3385, 3413 3441, 3469, 3498, 3527, 3556, 3585, 3614, 3643, 3673, 3703, 3733, 3763, 3793, 3823, 3854, 3885 3916, 3947, 3978, 4010, 4042, 4074, 4106, 4138, 4171, 4204, 4237, 4270, 4303, 4337, 4371, 4405 4439, 4473, 4508, 4543, 4578, 4613, 4648, 4684, 4720, 4756, 4792, 4828, 4865, 4902, 4939, 4976 5013, 5051, 5089, 5127, 5165, 5203, 5242, 5281, 5320, 5359, 5398, 5438, 5478, 5518, 5558, 5599

Index	Probability
	5640, 5681, 5722, 5763, 5805, 5847, 5889, 5931, 5973, 6016, 6059, 6102, 6145, 6189, 6233, 6277
	6321, 6365, 6410, 6455, 6500, 6545, 6591, 6637, 6683, 6729, 6776, 6823, 6870, 6917, 6964, 7012
	7060, 7108, 7156, 7205, 7254, 7303, 7352, 7402, 7452, 7502, 7552, 7602, 7653, 7704, 7755, 7806
	7858, 7910, 7962, 8014, 8067, 8120, 8173, 8226, 8280, 8334, 8388, 8442, 8496, 8551, 8606, 8661
	8716, 8772, 8828, 8884, 8940, 8997, 9054, 9111, 9168, 9226, 9284, 9342, 9400, 9459, 9518, 9577
	9636, 9696, 9756, 9816, 9876, 9936, 9997, 10058, 10119, 10180, 10242, 10304, 10366, 10428, 10491
	10554, 10617, 10680, 10744, 10808, 10872, 10936, 11001, 11066, 11131, 11196, 11262, 11328, 11394
	11460, 11527, 11594, 11661, 11728, 11795, 11863, 11931, 11999, 12067, 12136, 12205, 12274, 12343
	12413, 12483, 12553, 12623, 12694, 12765, 12836, 12907, 12979, 13051, 13123, 13195, 13267, 13340
	13413, 13486, 13559, 13633, 13707, 13781, 13855, 13930, 14005, 14080, 14155, 14230, 14306, 14382
	14458, 14534, 14611, 14688, 14765, 14842, 14920, 14998, 15076, 15154, 15232, 15311, 15390, 15469
	15548, 15628, 15708, 15788, 15868, 15948, 16029, 16110, 16191, 16272, 16353, 16435, 16517, 16599
	16681, 16764, 16847, 16930, 17013, 17096, 17180, 17264, 17348, 17432, 17516, 17601, 17686, 17771
	17856, 17941, 18027, 18113, 18199, 18285, 18371, 18458, 18545, 18632, 18719, 18806, 18894, 18982
	19070, 19158, 19246, 19335, 19424, 19513, 19602, 19691, 19780, 19870, 19960, 20050, 20140, 20230

Table B.9 (continued)

Index	Probability
64	20320, 20411, 20502, 20593, 20684, 20775, 20867, 20959, 21051, 21143, 21235, 21327, 21419, 21512
	21605, 21698, 21791, 21884, 21977, 22071, 22165, 22259, 22353, 22447, 22541, 22635, 22730, 22825
	22920, 23015, 23110, 23205, 23300, 23396, 23492, 23588, 23684, 23780, 23876, 23972, 24069, 24166
	24263, 24360, 24457, 24554, 24651, 24748, 24845, 24943, 25041, 25139, 25237, 25335, 25433, 25531
	25629, 25727, 25826, 25925, 26024, 26123, 26222, 26321, 26420, 26519, 26618, 26717, 26817, 26917
	27017, 27117, 27217, 27317, 27417, 27517, 27617, 27717, 27817, 27917, 28017,

Index	Probability
	28118, 28219, 28320
	28421, 28522, 28623, 28724, 28825, 28926, 29027, 29128, 29229, 29330, 29431, 29532, 29633, 29734
	29836, 29938, 30040, 30142, 30244, 30346, 30448, 30550, 30652, 30754, 30856, 30958, 31060, 31162
	31264, 31366, 31468, 31570, 31672, 31774, 31876, 31978, 32080, 32182, 32284, 32386, 32488, 32590
	32692, 32794, 32896, 32998, 33100, 33202, 33304, 33406, 33508, 33610, 33712, 33814, 33916, 34018
	34120, 34222, 34324, 34426, 34528, 34630, 34732, 34834, 34936, 35038, 35140, 35242, 35344, 35446
	35548, 35649, 35750, 35851, 35952, 36053, 36154, 36255, 36356, 36457, 36558, 36659, 36760, 36861
	36962, 37063, 37164, 37265, 37365, 37465, 37565, 37665, 37765, 37865, 37965, 38065, 38165, 38265
	38365, 38465, 38565, 38664, 38763, 38862, 38961, 39060, 39159, 39258, 39357, 39456, 39555, 39653
	39751, 39849, 39947, 40045, 40143, 40241, 40339, 40437, 40534, 40631, 40728, 40825, 40922, 41019
	41116, 41213, 41310, 41406, 41502, 41598, 41694, 41790, 41886, 41982, 42077, 42172, 42267, 42362
	42457, 42552, 42647, 42741, 42835, 42929, 43023, 43117, 43211, 43305, 43398, 43491, 43584, 43677
	43770, 43863, 43955, 44047, 44139, 44231, 44323, 44415, 44507, 44598, 44689, 44780, 44871, 44962
	45052, 45142, 45232, 45322, 45412, 45502, 45591, 45680, 45769, 45858, 45947, 46036, 46124, 46212
	46300, 46388, 46476, 46563, 46650, 46737, 46824, 46911, 46997, 47083, 47169, 47255, 47341, 47426
	47511, 47596, 47681, 47766, 47850, 47934, 48018, 48102, 48186, 48269, 48352, 48435, 48518, 48601
	48683, 48765, 48847, 48929, 49010, 49091, 49172, 49253, 49334, 49414, 49494, 49574, 49654, 49734
	49813, 49892, 49971, 50050, 50128, 50206, 50284, 50362, 50440, 50517, 50594, 50671, 50748, 50824
	50900, 50976, 51052, 51127, 51202, 51277, 51352, 51427, 51501, 51575, 51649, 51723, 51796, 51869
	51942, 52015, 52087, 52159, 52231, 52303, 52375, 52446, 52517, 52588, 52659, 52729, 52799, 52869
	52939, 53008, 53077, 53146, 53215, 53283, 53351, 53419, 53487, 53554, 53621, 53688, 53755, 53822
	53888, 53954, 54020, 54086, 54151, 54216, 54281, 54346, 54410, 54474, 54538, 54602, 54665, 54728
	54791, 54854, 54916, 54978, 55040, 55102, 55163, 55224, 55285, 55346, 55406, 55466, 55526, 55586
	55646, 55705, 55764, 55823, 55882, 55940, 55998, 56056, 56114, 56171, 56228,

Index	Probability
	56285, 56342, 56398
	56454, 56510, 56566, 56621, 56676, 56731, 56786, 56840, 56894, 56948, 57002, 57056, 57109, 57162
	57215, 57268, 57320, 57372, 57424, 57476, 57527, 57578, 57629, 57680, 57730, 57780, 57830, 57880
	57930, 57979, 58028, 58077, 58126, 58174, 58222, 58270, 58318, 58365, 58412, 58459, 58506, 58553
	58599, 58645, 58691, 58737, 58782, 58827, 58872, 58917, 58961, 59005, 59049, 59093, 59137, 59180
	59223, 59266, 59309, 59351, 59393, 59435, 59477, 59519, 59560, 59601, 59642, 59683, 59724, 59764
	59804, 59844, 59884, 59923, 59962, 60001, 60040, 60079, 60117, 60155, 60193, 60231, 60269, 60306
	60343, 60380, 60417, 60454, 60490, 60526, 60562, 60598, 60634, 60669, 60704, 60739, 60774, 60809
	60843, 60877, 60911, 60945, 60979, 61012, 61045, 61078, 61111, 61144, 61176, 61208, 61240, 61272
	61304, 61335, 61366, 61397, 61428, 61459, 61489, 61519, 61549, 61579, 61609, 61639, 61668, 61697
	61726, 61755, 61784, 61813, 61841, 61869, 61897, 61925, 61953, 61980, 62007, 62034, 62061, 62088
	62115, 62141, 62167, 62193, 62219, 62245, 62271, 62296, 62321, 62346, 62371, 62396, 62421, 62445
	62469, 62493, 62517, 62541, 62565, 62588, 62611, 62634, 62657, 62680, 62703, 62726, 62748, 62770
	62792, 62814, 62836, 62858, 62879, 62900, 62921, 62942, 62963, 62984, 63005, 63025, 63045, 63065

Table B.9 (continued)

Index	Probability
64	63085, 63105, 63125, 63145, 63164, 63183, 63202, 63221, 63240, 63259, 63278, 63297, 63315, 63333
	63351, 63369, 63387, 63405, 63423, 63440, 63457, 63474, 63491, 63508, 63525, 63542, 63559, 63575
	63591, 63607, 63623, 63639, 63655, 63671, 63687, 63703, 63718, 63733, 63748, 63763, 63778, 63793
	63808, 63823, 63837, 63851, 63865, 63879, 63893, 63907, 63921, 63935, 63949, 63963, 63976, 63989
	64002, 64015, 64028, 64041, 64054, 64067, 64080, 64092, 64104, 64116, 64128, 64140, 64152, 64164
	64176, 64188, 64200, 64212, 64223, 64234, 64245, 64256, 64267, 64278, 64289, 64300, 64311, 64322
	64333, 64343, 64353, 64363, 64373, 64383, 64393, 64403, 64413, 64423, 64433,

Index	Probability
	64443, 64452, 64461
	64470, 64479, 64488, 64497, 64506, 64515, 64524, 64533, 64542, 64551, 64560, 64568, 64576, 64584
	64592, 64600, 64608, 64616, 64624, 64632, 64640, 64648, 64656, 64664, 64672, 64680, 64687, 64694
	64701, 64708, 64715, 64722, 64729, 64736, 64743, 64750, 64757, 64764, 64771, 64778, 64785, 64791
	64797, 64803, 64809, 64815, 64821, 64827, 64833, 64839, 64845, 64851, 64857, 64863, 64869, 64875
	64881, 64887, 64893, 64898, 64903, 64908, 64913, 64918, 64923, 64928, 64933, 64938, 64943, 64948
	64953, 64958, 64963, 64968, 64973, 64978, 64983, 64988, 64993, 64998, 65002, 65006, 65010, 65014
	65018, 65022, 65026, 65030, 65034, 65038, 65042, 65046, 65050, 65054, 65058, 65062, 65066, 65070
	65074, 65078, 65082, 65086, 65090, 65094, 65098, 65101, 65104, 65107, 65110, 65113, 65116, 65119
	65122, 65125, 65128, 65131, 65134, 65137, 65140, 65143, 65146, 65149, 65152, 65155, 65158, 65161
	65164, 65167, 65170, 65173, 65176, 65179, 65182, 65185, 65188, 65191, 65194, 65197, 65200, 65202
	65204, 65206, 65208, 65210, 65212, 65214, 65216, 65218, 65220, 65222, 65224, 65226, 65228, 65230
	65232, 65234, 65236, 65238, 65240, 65242, 65244, 65246, 65248, 65250, 65252, 65254, 65256, 65258
	65260, 65262, 65264, 65266, 65268, 65270, 65272, 65274, 65276, 65278, 65280, 65282, 65536

Table B.10 Layer 1 CNN (convolution kernel parameter kernel) of base decoding neural network parameter

Parameter value
0x1.59e8d00000000p–11, –0x1.303e2e0000000p–6, 0x1.f808ac0000000p–8, 0x1.d4da920000000p–7, –0x1.19255a0000000p–6, 0x1.a818920000000p–15, 0x1.7a2b560000000p–9, 0x1.15338c0000000p–5, 0x1.7ff9e80000000p–11, –0x1.1db39a0000000p–14, –0x1.b576c40000000p–10, –0x1.32a3780000000p–7, 0x1.076a0a0000000p–4, 0x1.2927b60000000p–10, –0x1.a75d660000000p–8, 0x1.503bf00000000p–5,
0x1.0ec3560000000p–10, 0x1.b544240000000p–7, 0x1.74d22a0000000p–6, –0x1.f3dc660000000p–7, 0x1.9fc23e0000000p–6, –0x1.536ec40000000p–11, –0x1.6218060000000p–9, 0x1.e1a0a60000000p–7, –0x1.884efa0000000p–10, –0x1.ac084c0000000p–10, –0x1.d28be60000000p–10, 0x1.40c1ba0000000p–6, –0x1.77cdb80000000p–1, 0x1.8542020000000p–12, –0x1.44e1de0000000p–5,

Parameter value
–0x1.dce52c0000000p–7,
0x1.2bb7900000000p–10, 0x1.19f6360000000p–4, –0x1.a6dfaa0000000p–5, –0x1.6664d40000000p–5, 0x1.2117500000000p–4, 0x1.3a65380000000p–12, –0x1.0562320000000p–7, –0x1.6691ce0000000p–3, –0x1.a74f300000000p–9, 0x1.27394e0000000p–12, 0x1.1631760000000p–8, 0x1.ca4b820000000p–7, 0x1.1731260000000p–3, –0x1.229d140000000p–8, 0x1.5bd2480000000p–4, –0x1.dc34380000000p–4,

Table B.10 (continued)

Parameter value
0x1.4c296c0000000p–10, –0x1.2356aa0000000p–6, 0x1.b709080000000p–7, 0x1.29d4dc0000000p–7, –0x1.333e580000000p–6, –0x1.3b4c180000000p–11, 0x1.009dcc0000000p–9, 0x1.a80dec0000000p–5, –0x1.efade20000000p–12, –0x1.b7aca60000000p–12, –0x1.b43cba0000000p–12, 0x1.68897a0000000p–7, –0x1.66d2320000000p–5, –0x1.6f6ca80000000p–11, –0x1.772ab80000000p–7, 0x1.724c820000000p–5,
0x1.42e0c20000000p–11, 0x1.163a640000000p–6, –0x1.85e8060000000p–6, –0x1.7c34180000000p–7, 0x1.774b600000000p–7, 0x1.e969a80000000p–10, 0x1.3cbab60000000p–9, –0x1.f24c1a0000000p–5, –0x1.0250100000000p–9, –0x1.5c73840000000p–11, –0x1.c3f3480000000p–11, –0x1.53a6cc0000000p–6, 0x1.3ce0760000000p–16, 0x1.1f97fc0000000p–10, 0x1.66122c0000000p–6, –0x1.0567e00000000p–6,
0x1.f09be60000000p–11, 0x1.e149060000000p–6, –0x1.23bf360000000p–5, –0x1.21e62e0000000p–6, 0x1.bd17b80000000p–8, 0x1.88a32c0000000p–9, 0x1.5c065e0000000p–13, –0x1.7b61600000000p–4, –0x1.bd8f7c0000000p–9, –0x1.0fe26c0000000p–12, 0x1.6490360000000p–10, –0x1.9a43800000000p–6, 0x1.5f6e080000000p–2, –0x1.de432c0000000p–12, 0x1.e4ad080000000p–5, –0x1.4fac560000000p–6,
–0x1.cd39b00000000p–10, –0x1.e16f580000000p–8, 0x1.abe0640000000p–7, 0x1.7f62a00000000p–8, 0x1.21d3660000000p–6, –0x1.2241fa0000000p–8, –0x1.d583ee0000000p–11, 0x1.3ce7480000000p–5, 0x1.0fa3120000000p–9, –0x1.6282d20000000p–11, –0x1.8c524a0000000p–10, 0x1.912d4e0000000p–6, –0x1.f7aee00000000p–2, 0x1.e4505e0000000p–11, –0x1.8708360000000p–5, –0x1.7e5ace0000000p–5,
0x1.355c680000000p–13, 0x1.5522a60000000p–5, –0x1.06f7b00000000p–6, –0x1.bf1a9a0000000p–6, 0x1.ecaeac0000000p–8, 0x1.605c700000000p–8, –0x1.a0040c0000000p–10, –0x1.3c05b40000000p–4, –0x1.41cb120000000p–9, 0x1.6bd0c20000000p–12, 0x1.ffd6a20000000p–10, –0x1.e283600000000p–7, 0x1.2473880000000p–3, 0x1.1c161a0000000p–12, 0x1.45797e0000000p–5,

Parameter value
–0x1.0235900000000p–5,
0x1.1343040000000p–7, –0x1.474ab40000000p–3, –0x1.72bd740000000p–5, –0x1.5625d20000000p–5, 0x1.2a6b000000000p–5, –0x1.94492c0000000p–5, –0x1.6469a40000000p–5, –0x1.beeb820000000p–4, 0x1.dc5b440000000p–7, –0x1.7b40020000000p–7, 0x1.91742e0000000p–6, 0x1.4603940000000p–3, –0x1.22f7140000000p–3, 0x1.1b42460000000p–6, –0x1.4c612e0000000p–2, –0x1.3658be0000000p–5,
0x1.da2c6e0000000p–7, 0x1.3c2a900000000p–3, 0x1.8121480000000p–1, 0x1.dede740000000p–2, –0x1.d6ae0c0000000p–4, 0x1.83cb8c0000000p–8, –0x1.a9e4a40000000p–7, 0x1.ab9e660000000p–3, –0x1.d9e9fc0000000p–7, –0x1.a8fb320000000p–8, –0x1.f48fd60000000p–8, 0x1.a520ba0000000p–2, –0x1.31f3880000000p–2, 0x1.4b74e40000000p–9, 0x1.8a7d540000000p–6, –0x1.52c7680000000p–2,

Table B.10 (continued)

Parameter value
–0x1.1468960000000p–4, 0x1.77f6520000000p–3, –0x1.2ef9680000000p–3, 0x1.177af20000000p–4, –0x1.8ba6980000000p–2, 0x1.fd51000000000p–4, 0x1.848b840000000p–3, 0x1.f84fe00000000p–2, –0x1.03f5ca0000000p–7, 0x1.66fc6a0000000p–5, –0x1.6592ce0000000p–4, –0x1.5e32080000000p–2, 0x1.309e0e0000000p–4, –0x1.4bd0380000000p–4, 0x1.16ed760000000p–1, –0x1.acd88c0000000p–4,
0x1.ee11ee0000000p–7, –0x1.e4f7720000000p–4, 0x1.76eb5a0000000p–5, 0x1.0d6ca40000000p–6, 0x1.04805c0000000p–2, –0x1.261a440000000p–5, –0x1.514d700000000p–5, 0x1.7549c00000000p–1, 0x1.307a000000000p–8, –0x1.a21c920000000p–7, 0x1.7dcb500000000p–6, 0x1.8dae920000000p–3, 0x1.1af0700000000p–2, 0x1.3d308a0000000p–6, –0x1.a4f4880000000p–3, 0x1.32d5780000000p–3,
–0x1.ab05320000000p–6, –0x1.2e86be0000000p–2, 0x1.52c3da0000000p–5, 0x1.720a380000000p–3, –0x1.5803de0000000p–1, 0x1.25f59a0000000p–9, 0x1.dd26c60000000p–5, –0x1.1d495e0000000p–3, 0x1.61892a0000000p–6, 0x1.5df5860000000p–7, –0x1.e6c5b60000000p–8, 0x1.36e0340000000p–5, 0x1.3086fa0000000p–1, –0x1.3606ea0000000p–6, –0x1.551fd20000000p–2, 0x1.b2ce540000000p–6,
–0x1.91011c0000000p–5, –0x1.3be3760000000p–3, –0x1.5f919c0000000p–2, –0x1.170c220000000p–4, –0x1.11cec00000000p–2, 0x1.d384340000000p–7, 0x1.4e515e0000000p–4, 0x1.6ed4900000000p–4, 0x1.f3f3e20000000p–6, 0x1.588fce0000000p–6, –0x1.2f0c780000000p–6, –0x1.aa2b960000000p–8,

Parameter value
–0x1.31315a0000000p–1, –0x1.23a2720000000p–5, –0x1.38940a0000000p–3, –0x1.b95d240000000p–3,
0x1.1fff4e0000000p–5, 0x1.de7e2c0000000p–3, 0x1.d96dec0000000p–2, 0x1.865a1a0000000p–3, –0x1.9c60840000000p–5, 0x1.4641600000000p–5, –0x1.01b9560000000p–6, –0x1.a008180000000p–2, –0x1.7070c20000000p–5, –0x1.00f29c0000000p–7, –0x1.6bd2ec0000000p–7, –0x1.b9a4e00000000p–3, 0x1.8f2e040000000p–4, 0x1.906a940000000p–7, 0x1.8288900000000p–2, 0x1.155f9a0000000p–2,
–0x1.49d4ce0000000p–5, 0x1.ccf8100000000p–3, –0x1.3c0b7a0000000p–3, –0x1.d2c6ec0000000p–5, 0x1.0ee7680000000p–3, 0x1.97b0c80000000p–6, 0x1.a15dfc0000000p–5, –0x1.6a5cca0000000p–4, 0x1.e6f8d40000000p–7, 0x1.2395280000000p–6, –0x1.dbe67c0000000p–6, 0x1.0d2a000000000p–3, 0x1.02eff00000000p–1, –0x1.0aecec0000000p–5, 0x1.7681c00000000p–3, –0x1.52a75c0000000p–1,
0x1.39c67a0000000p–2, 0x1.99a6c60000000p–2, 0x1.677eca0000000p–3, –0x1.210f5a0000000p–7, –0x1.7dc66c0000000p–7, 0x1.56a8240000000p–5, 0x1.93cd7a0000000p–4, 0x1.c68bb00000000p–6, –0x1.58f8de0000000p–3, 0x1.61164a0000000p–5, –0x1.cc95c60000000p–6, –0x1.3238b60000000p–1, 0x1.729dea0000000p–5, 0x1.435e9e0000000p–3, –0x1.574bf60000000p–1, –0x1.ff5c220000000p–4,

Table B.10 (continued)

Parameter value
0x1.8dd84a0000000p–6, 0x1.f8e1c00000000p–8, 0x1.2cb7d40000000p–2, –0x1.c7bd5c0000000p–6, –0x1.abe2cc0000000p–6, 0x1.540ba40000000p–2, –0x1.dbd9e20000000p–3, –0x1.0951f40000000p–6, –0x1.a9e5f80000000p–2, –0x1.ed7d000000000p–2, 0x1.1739e20000000p–3, 0x1.11220c0000000p–8, 0x1.44b9a60000000p–5, –0x1.14f44c0000000p–3, 0x1.08c0dc0000000p–3, –0x1.7ca5240000000p–4,
–0x1.19e74e0000000p–1, –0x1.6199140000000p–5, –0x1.1b37120000000p–4, 0x1.c101700000000p–4, –0x1.790a200000000p–4, 0x1.8c918a0000000p–4, –0x1.f7ca9c0000000p–2, 0x1.eb0af20000000p–6, 0x1.6a11a40000000p–2, –0x1.3f8b2e0000000p–4, 0x1.853cbc0000000p–2, –0x1.31116a0000000p–2, –0x1.5e73d60000000p–8, –0x1.89e9040000000p–3, –0x1.00a79e0000000p–2, –0x1.05c19e0000000p–5,
0x1.9eac9e0000000p–3, 0x1.67defa0000000p–3, –0x1.083a4a0000000p–2, 0x1.86b6ba0000000p–5, –0x1.802d760000000p–2, –0x1.32cc600000000p–3, –0x1.6baad00000000p–1, 0x1.5ef5d80000000p–4, –0x1.6d150c0000000p–3,

Parameter value
–0x1.076e540000000p–7, –0x1.0511e60000000p–2, 0x1.cb45820000000p–3, –0x1.b1a08a0000000p–4, 0x1.dfc21e0000000p–4, –0x1.867ada0000000p–3, 0x1.59579c0000000p–2,
0x1.4be16e0000000p–2, –0x1.3813860000000p–5, –0x1.3323f80000000p–1, 0x1.81acb80000000p–3, 0x1.8f04240000000p–4, –0x1.81e8ce0000000p–5, 0x1.300bf00000000p–3, –0x1.4c7f420000000p–5, –0x1.b738a20000000p–6, –0x1.6640a60000000p–3, 0x1.54b7920000000p–1, 0x1.4848940000000p–3, –0x1.233cd00000000p–4, 0x1.356d420000000p–2, –0x1.9e7de80000000p–3, 0x1.754c0c0000000p–3,
0x1.2078780000000p–3, 0x1.a2bf820000000p–2, 0x1.24cf1c0000000p–1, –0x1.a50eb60000000p–2, –0x1.5ab4700000000p–2, 0x1.c5951e0000000p–3, –0x1.a748f80000000p–4, 0x1.4472ec0000000p–6, –0x1.7669a00000000p–7, 0x1.2c32f80000000p–4, 0x1.09aad60000000p–2, 0x1.b8fa340000000p–3, 0x1.c642b40000000p–4, 0x1.37af460000000p–3, 0x1.0d3e5e0000000p–3, –0x1.8375d60000000p–5,
–0x1.75dc4a0000000p–2, 0x1.cc3bc00000000p–2, –0x1.3100700000000p–4, –0x1.0b73de0000000p–2, –0x1.e2111a0000000p–3, –0x1.0612ec0000000p–2, 0x1.9338380000000p–2, 0x1.254d2c0000000p–5, 0x1.9132e20000000p–3, –0x1.9574160000000p–3, 0x1.2c819c0000000p–4, 0x1.1dd2b20000000p–2, 0x1.5b4d300000000p–9, –0x1.e655480000000p–3, –0x1.273e560000000p–2, 0x1.a0512a0000000p–3,
–0x1.9726500000000p–3, 0x1.15dc980000000p–2, –0x1.07ed280000000p–1, –0x1.0686120000000p–10, 0x1.4a5fcc0000000p–4, 0x1.4ce72a0000000p–1, 0x1.8f08380000000p–4, –0x1.34d79a0000000p–5, –0x1.1454340000000p–3, 0x1.d163120000000p–5, –0x1.0d35440000000p–3, 0x1.d399860000000p–3, –0x1.442b6c0000000p–6, –0x1.ca9b040000000p–3, –0x1.e1a5620000000p–4, 0x1.4ed30e0000000p–3,

Table B.10 (continued)

Parameter value
0x1.55cdca0000000p–1, 0x1.a038300000000p–5, –0x1.dbb8f00000000p–6, 0x1.db47260000000p–6, –0x1.82f09c0000000p–6, 0x1.ead8a00000000p–4, –0x1.1947a40000000p–6, –0x1.026a5a0000000p–5, 0x1.2fae0a0000000p–1, 0x1.a209040000000p–7, 0x1.bf47a00000000p–7, 0x1.69331a0000000p–5, –0x1.9cf3a60000000p–8, –0x1.95c1f80000000p–2, –0x1.22086c0000000p–5, 0x1.de2d7e0000000p–5,
–0x1.e5fbf60000000p–4, 0x1.5a7aac0000000p–5, –0x1.29f7b60000000p–5, –0x1.3ec9cc0000000p–6, –0x1.fa24140000000p–7, 0x1.c4ab820000000p–4, 0x1.82e2b20000000p–10, –0x1.ac17fe0000000p–9, 0x1.47e5e80000000p–8, 0x1.c2a7d00000000p–5, 0x1.223a260000000p–4, 0x1.456c8c0000000p–4,

Parameter value
–0x1.f213ec0000000p–10, –0x1.3a72960000000p–7, 0x1.236af20000000p–7, 0x1.1872600000000p–5,
0x1.181c7c0000000p–2, –0x1.869c840000000p–9, 0x1.3d2e280000000p–8, 0x1.2ef4140000000p–7, –0x1.7a23960000000p–7, 0x1.4d9e760000000p–5, –0x1.47ba5c0000000p–6, –0x1.a791d00000000p–8, 0x1.24b8f00000000p–2, –0x1.e7bd920000000p–4, 0x1.3d99140000000p–4, –0x1.2f5f420000000p–6, –0x1.662e3a0000000p–10, 0x1.1c0c520000000p–5, –0x1.de8e480000000p–9, 0x1.34843c0000000p–8,
0x1.7a98ca0000000p–3, –0x1.feaf640000000p–4, 0x1.931a9c0000000p–4, 0x1.0595760000000p–4, 0x1.390eec0000000p–10, –0x1.33f9360000000p–2, –0x1.44fb8c0000000p–4, 0x1.178a7a0000000p–6, –0x1.c7e0380000000p–3, –0x1.583d140000000p–9, 0x1.7f94020000000p–2, –0x1.27719a0000000p–3, –0x1.f3c76a0000000p–11, –0x1.4701ce0000000p–3, –0x1.2ba7920000000p–9, –0x1.31c0340000000p–4,
0x1.7036e00000000p–3, –0x1.25467a0000000p–5, 0x1.e412c00000000p–5, 0x1.1750a40000000p–8, 0x1.9c7bce0000000p–6, –0x1.9328e40000000p–3, 0x1.760b340000000p–5, 0x1.b8e8200000000p–7, –0x1.58c8460000000p–3, –0x1.7816f80000000p–3, –0x1.d8ef200000000p–4, –0x1.c533920000000p–4, 0x1.017fc20000000p–8, 0x1.4930740000000p–5, –0x1.1c4fc60000000p–8, –0x1.c3a7120000000p–5,
–0x1.888c7a0000000p–4, 0x1.5ffae00000000p–4, –0x1.bc7c780000000p–4, –0x1.2fb71a0000000p–6, –0x1.edbd5a0000000p–5, 0x1.5fa2d20000000p–5, –0x1.dd639a0000000p–6, –0x1.764c980000000p–8, –0x1.bf212c0000000p–3, 0x1.9f69460000000p–2, 0x1.53fc460000000p–2, 0x1.a838c40000000p–3, –0x1.01993e0000000p–7, –0x1.98d9180000000p–2, 0x1.0c22b00000000p–7, 0x1.612eba0000000p–4,
0x1.1f1ed60000000p–2, –0x1.3e483a0000000p–7, –0x1.4447000000000p–9, 0x1.31a5ec0000000p–5, –0x1.9899000000000p–6, –0x1.b246900000000p–3, –0x1.918f480000000p–5, 0x1.07bb740000000p–8, –0x1.1d63a00000000p–2, 0x1.e700460000000p–3, 0x1.070bac0000000p–2, 0x1.68bfac0000000p–7, –0x1.4235b80000000p–8, –0x1.ccebb60000000p–2, –0x1.b6a7180000000p–7, 0x1.7a2f320000000p–8,

Table B.10 (continued)

Parameter value
0x1.79271c0000000p–4, –0x1.5c37a80000000p–9, 0x1.a0e4840000000p–7, –0x1.e0532e0000000p–8, 0x1.35e3500000000p–7, –0x1.696bca0000000p–3, 0x1.19a6a00000000p–5, 0x1.7c86460000000p–7, –0x1.ce37860000000p–3,

Parameter value
0x1.813cda0000000p–6, –0x1.9d4bbe0000000p–4, –0x1.2b4f560000000p–5, 0x1.3444160000000p–9, –0x1.21c4fc0000000p–2, 0x1.1324de0000000p–8, –0x1.6fc7a60000000p–6,
0x1.1770880000000p–6, 0x1.abcd320000000p–15, 0x1.ab025a0000000p–9, –0x1.69ca560000000p–9, –0x1.17e9c60000000p–10, –0x1.87b26e0000000p–5, 0x1.26d7b00000000p–9, 0x1.7967780000000p–9, –0x1.dddfa60000000p–6, –0x1.31d51e0000000p–11, 0x1.28dbc20000000p–5, –0x1.f0a34c0000000p–9, –0x1.fe5e600000000p–14, –0x1.8e34840000000p–6, –0x1.a6da4c0000000p–10, –0x1.47fd440000000p–8,
–0x1.3232c20000000p–7, –0x1.de6f460000000p–17, 0x1.ccd8c60000000p–15, –0x1.1e01be0000000p–13, –0x1.65c4560000000p–10, –0x1.1382320000000p–5, –0x1.85048c0000000p–10, 0x1.b096dc0000000p–9, –0x1.36d4b20000000p–4, 0x1.bda6060000000p–6, 0x1.a820ac0000000p–6, –0x1.f54d6c0000000p–13, 0x1.1cfe700000000p–13, –0x1.5555780000000p–5, 0x1.cf99660000000p–10, –0x1.15cde80000000p–9,
0x1.6d70020000000p–8, –0x1.f393d00000000p–15, 0x1.5ca03a0000000p–12, –0x1.c293ca0000000p–13, 0x1.8272300000000p–16, –0x1.2a73c80000000p–8, –0x1.b3f7720000000p–12, 0x1.6bb13e0000000p–12, 0x1.f3d3de0000000p–7, –0x1.c03a620000000p–8, 0x1.4560ce0000000p–8, –0x1.0574280000000p–9, 0x1.c25e980000000p–13, 0x1.2c5b5c0000000p–7, –0x1.e850700000000p–14, –0x1.6952960000000p–10,
0x1.05436a0000000p–6, 0x1.229a720000000p–9, 0x1.5c89680000000p–10, –0x1.03d07a0000000p–13, 0x1.7565c40000000p–14, 0x1.1a7bea0000000p–4, –0x1.bf93500000000p–10, –0x1.1d9d140000000p–8, 0x1.24db940000000p–3, –0x1.183e580000000p–4, –0x1.5dd3e00000000p–8, –0x1.ba19fe0000000p–9, 0x1.76c2ec0000000p–13, 0x1.0074620000000p–3, –0x1.d24c100000000p–10, 0x1.48cd580000000p–9,
0x1.57b3e80000000p–7, –0x1.ae09c60000000p–12, –0x1.2703400000000p–10, 0x1.9652900000000p–10, 0x1.9918e80000000p–9, 0x1.72f5060000000p–5, 0x1.0e55ca0000000p–8, –0x1.5e03820000000p–8, 0x1.9d14ac0000000p–4, –0x1.5da8d00000000p–6, –0x1.c008200000000p–5, 0x1.899aec0000000p–10, 0x1.5c1c8e0000000p–13, 0x1.10c86a0000000p–5, –0x1.cea7a80000000p–9, 0x1.05f70e0000000p–8,
–0x1.d84bc20000000p–7, –0x1.72bda80000000p–12, 0x1.7028a20000000p–9, –0x1.2af6440000000p–9, –0x1.9034ae0000000p–8, –0x1.958a300000000p–4, –0x1.fe77280000000p–11, 0x1.28d2e80000000p–7, –0x1.95f5860000000p–3, 0x1.922b940000000p–5, 0x1.76d3040000000p–4, –0x1.ebff3e0000000p–9, –0x1.7154380000000p–12, –0x1.6660300000000p–4, 0x1.4a0cfe0000000p–8, –0x1.3059420000000p–7,

Table B.10 (continued)

Parameter value
0x1.d5885a0000000p–7, 0x1.0a90780000000p–11, 0x1.76068e0000000p–9, –0x1.8f2d620000000p–10, –0x1.bd5bd60000000p–10, –0x1.e41c7c0000000p–7, 0x1.b683640000000p–9, 0x1.e6cb860000000p–11, –0x1.0cef940000000p–8, –0x1.417d000000000p–6, 0x1.e5d1820000000p–6, –0x1.8a17720000000p–9, –0x1.218aee0000000p–12, 0x1.48fd140000000p–6, –0x1.6b766a0000000p–11, –0x1.582b660000000p–9,
0x1.e08a3a0000000p–10, –0x1.6f23560000000p–11, –0x1.4ed4540000000p–10, 0x1.7043c80000000p–13, 0x1.3213a60000000p–10, 0x1.6bc5540000000p–7, 0x1.7e4b0e0000000p–10, –0x1.03ee660000000p–9, 0x1.37c9600000000p–5, 0x1.0110ac0000000p–10, –0x1.cc9fee0000000p–6, 0x1.4a8fd40000000p–11, 0x1.7581960000000p–13, –0x1.70e2de0000000p–9, –0x1.d36e520000000p–10, 0x1.505c960000000p–10,

Table B.11 Layer 1 CNN (bias parameter bias) of base decoding neural network parameter

Parameter value
–0x1.eb5fae0000000p–9, –0x1.9feb3e0000000p–10, 0x1.adfcfe0000000p–7, 0x1.09e3760000000p–10, –0x1.4e5c400000000p–8, 0x1.3030800000000p–9, 0x1.587aa60000000p–10, 0x1.1136a60000000p–9,

Table B.12 Layer 1 CNN (IGDN activation function beta parameter) of base decoding neural network parameter

Parameter value
0x1.75b7a60000000p+1, 0x1.f8ee3e0000000p+1, 0x1.c093e20000000p+1, 0x1.b223300000000p+1, 0x1.bfa5ca0000000p+1, 0x1.c8cf660000000p+1, 0x1.e3f8880000000p+1, 0x1.b99eba0000000p+1,

Table B.13 Layer 1 CNN (IGDN activation function gamma parameter) of base decoding neural network parameter

Parameter value
0x1.423ea80000000p–23, 0x1.0fbbb80000000p–20, 0x1.323ad60000000p–26, 0x1.2b173c0000000p–20, 0x1.d4079e0000000p–27, 0x1.71b1280000000p–20, 0x0.0p+0, 0x1.c2c1e60000000p–22,
0x1.e4bdec0000000p–20, 0x1.8649d60000000p–20, 0x1.0ee6240000000p–21, 0x1.b89fb80000000p–21, 0x1.54ae100000000p–21, 0x1.fe03300000000p–20,

Parameter value
0x1.844cf60000000p–22, 0x1.330ee00000000p–22,
0x1.539eae0000000p–21, 0x1.2e78960000000p–21, 0x1.89f0cc0000000p–20, 0x1.1b027a0000000p–20, 0x1.b696820000000p–24, 0x1.e2a72c0000000p–21, 0x1.d5564a0000000p–21, 0x1.46a55c0000000p–25,
0x0.0p+0, 0x1.569d220000000p–19, 0x1.1ea8720000000p–19, 0x1.c135700000000p–20, 0x0.0p+0, 0x1.3b9fe20000000p–19, 0x1.2a67e80000000p–20, 0x1.f72f720000000p–23,

Table B.13 (continued)

Parameter value
0x1.6286ca0000000p–24, 0x1.1e50920000000p–21, 0x1.09ff880000000p–20, 0x1.743d620000000p–21, 0x1.a07fac0000000p–21, 0x1.1cd0fe0000000p–23, 0x1.371da20000000p–20, 0x1.dff2c00000000p–24,
0x1.60e7c80000000p–19, 0x1.43dd440000000p–19, 0x1.1aafba0000000p–19, 0x1.9caf300000000p–20, 0x1.76cdca0000000p–20, 0x1.37c2820000000p–20, 0x1.1b7b5a0000000p–22, 0x1.60ce4a0000000p–19,
0x1.b27f140000000p–30, 0x1.5b16c60000000p–20, 0x1.3ace560000000p–25, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x1.e3ffb00000000p–20, 0x1.d07f2a0000000p–33,
0x0.0p+0, 0x0.0p+0, 0x1.e0580a0000000p–21, 0x0.0p+0, 0x1.f4b2e20000000p–22, 0x1.093efc0000000p–20, 0x1.ad02020000000p–20, 0x1.ad60fa0000000p–20,

Table B.14 Layer 2 CNN (convolution kernel parameter kernel) of base decoding neural network parameter

Parameter value
–0x1.8080960000000p–1, –0x1.af34300000000p–7, –0x1.1dc0200000000p–2, 0x1.0db8120000000p–5, 0x1.3e01f40000000p–4, 0x1.3237c80000000p–3, –0x1.4519c00000000p–5, 0x1.8875fc0000000p–4,
–0x1.68d4780000000p–1, –0x1.06cbb60000000p–3, –0x1.1ac34e0000000p–2, 0x1.ef616a0000000p–3, 0x1.32e3e60000000p–3, 0x1.da977e0000000p–3, 0x1.d3c9a20000000p–3, 0x1.aa7b2e0000000p–3,
–0x1.207ae60000000p–4, 0x1.2219e20000000p–4, 0x1.acb9760000000p–4, –0x1.a4f0000000000p–2, –0x1.94c1820000000p–4, –0x1.7183d60000000p–2, –0x1.8e0bd00000000p–2, –0x1.cbbc5e0000000p–3,

Parameter value
0x1.c1c5a20000000p–3, –0x1.680e680000000p–3, –0x1.4dc3900000000p–5, –0x1.6eb7d80000000p–5, 0x1.234d1a0000000p–4, –0x1.fcd0940000000p–3, 0x1.8e84180000000p–3, 0x1.cd8d1c0000000p–2,
–0x1.ee98ba0000000p–2, 0x1.74d2a00000000p–2, 0x1.0517440000000p–8, –0x1.b1489c0000000p–3, –0x1.3056d20000000p–1, 0x1.0e99300000000p–2, –0x1.2dd36c0000000p–1, –0x1.0e6a2e0000000p–1,
–0x1.2291c60000000p–3, –0x1.2e55160000000p–3, 0x1.d3da260000000p–3, 0x1.01296a0000000p–1, 0x1.76677e0000000p–2, –0x1.d4a8200000000p–1, –0x1.54a83a0000000p–2, 0x1.31d0a00000000p–2,
0x1.1d3ec00000000p–2, 0x1.1a7bfe0000000p–4, 0x1.3e4ba60000000p–2, 0x1.58cb4c0000000p–1, –0x1.be652c0000000p–2, 0x1.03497c0000000p–2, 0x1.0c41860000000p–4, –0x1.42167a0000000p–3,

Table B.14 (continued)

Parameter value
0x1.181dac0000000p–2, –0x1.2f9db60000000p–4, –0x1.1ee1f40000000p–3, –0x1.aa883c0000000p–1, 0x1.2a88e00000000p–3, –0x1.a4ad000000000p–3, 0x1.c424ea0000000p–6, 0x1.2cd3160000000p–3,
0x1.ebcf2a0000000p–4, 0x1.09e1d20000000p–1, –0x1.2fb1bc0000000p–2, 0x1.4d2f420000000p–5, –0x1.7cb27c0000000p–2, 0x1.bda4be0000000p–4, –0x1.9ffd380000000p–3, 0x1.b32a620000000p–2,
–0x1.5ba4ec0000000p–3, –0x1.43d97a0000000p–4, 0x1.1ee4fe0000000p–5, –0x1.dac2b20000000p–4, 0x1.8a33e80000000p–2, –0x1.13d6740000000p–2, 0x1.e9bf800000000p–2, –0x1.79d25e0000000p–1,
–0x1.d6520e0000000p–3, –0x1.095d6e0000000p–2, 0x1.32106e0000000p–4, 0x1.3ce10c0000000p–4, –0x1.7468e00000000p–1, –0x1.0463500000000p–1, 0x1.63b5e60000000p–2, –0x1.53df8e0000000p–5,
0x1.df30d40000000p–3, –0x1.0aa7b60000000p–3, –0x1.c6243e0000000p–1, 0x1.5628e00000000p–4, –0x1.606d3e0000000p–2, –0x1.72e9500000000p–2, 0x1.e2d9b00000000p–4, –0x1.895dba0000000p–3,
0x1.6e2d2a0000000p–2, –0x1.a606800000000p–2, –0x1.74de1c0000000p–2, 0x1.601aec0000000p–3, 0x1.b70ca20000000p–3, 0x1.1e1d5c0000000p–2, –0x1.196e640000000p–1, –0x1.ea98820000000p–3,
–0x1.0401800000000p–3, 0x1.1a0c260000000p–1, –0x1.016c140000000p–3, 0x1.d58bb20000000p–4, 0x1.9967ca0000000p–2, 0x1.a8e3540000000p–4, 0x1.fd6cc80000000p–3, 0x1.76fb920000000p–3,
0x1.0c56f20000000p–4, 0x1.e9de5c0000000p–2, –0x1.45d4aa0000000p–3,

Parameter value
–0x1.30c3500000000p–8, –0x1.37da420000000p–2, –0x1.5d21980000000p–2, 0x1.cf72860000000p–3, 0x1.77bb6c0000000p–3,
0x1.915e400000000p–3, 0x1.74ecf80000000p–1, –0x1.1f2a360000000p–2, 0x1.2aff840000000p–4, 0x1.cd33e40000000p–2, –0x1.438fee0000000p–3, 0x1.5486b20000000p–4, –0x1.eec7a20000000p–2,
0x1.373a040000000p–4, –0x1.ae6ab40000000p–3, –0x1.25897e0000000p–5, –0x1.5437fe0000000p–7, 0x1.c309020000000p–6, 0x1.b0fcfc0000000p–5, –0x1.0f23200000000p–3, –0x1.2679520000000p–3,
–0x1.bdc3220000000p–6, 0x1.a39eb40000000p–17, –0x1.986b960000000p–5, 0x1.fdee220000000p–9, –0x1.3891ec0000000p–8, 0x1.931a540000000p–7, 0x1.cb77380000000p–5, 0x1.2271300000000p–4,
–0x1.1e5d600000000p–6, –0x1.f977400000000p–4, –0x1.ce4b0a0000000p–5, 0x1.5189860000000p–7, 0x1.c9fd7c0000000p–7, 0x1.3289980000000p–4, –0x1.0e84da0000000p–8, 0x1.2ff7920000000p–4,
0x1.29f4480000000p–5, –0x1.ac52340000000p–5, –0x1.dd0e2a0000000p–4, 0x1.4dba520000000p–6, 0x1.b02ef20000000p–4, 0x1.fe13f00000000p–5, –0x1.54f1560000000p–6, –0x1.b37cf80000000p–5,

Table B.15 Layer 2 CNN (bias parameter bias) of base decoding neural network parameter

Parameter value
0x1.7c47e60000000p–8, –0x1.7c8f440000000p–8, 0x1.3c2ee60000000p–7, 0x1.5192020000000p–8,

Table B.16 Layer 2 CNN (IGDN activation function beta parameter) of base decoding neural network parameter

Parameter value
0x1.f377ac0000000p+3, 0x1.33e7f40000000p+4, 0x1.116c040000000p+4, 0x1.33c9640000000p+4,

Table B.17 Layer 2 CNN (IGDN activation function gamma parameter) of base decoding neural network parameter

Parameter value
0x1.8d38dc0000000p–24, 0x0.0p+0, 0x1.87aa380000000p–21, 0x1.86341c0000000p–34,
0x0.0p+0, 0x1.aff7280000000p–24, 0x0.0p+0, 0x0.0p+0,

Parameter value
0x1.b5a80c0000000p–29, 0x0.0p+0, 0x0.0p+0, 0x1.14e17c0000000p–30,
0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x1.40acb60000000p–22,

Table B.18 Layer 3 CNN (convolution kernel parameter kernel) of base decoding neural network parameter

Parameter value
–0x1.b8e81e0000000p–3, –0x1.5958ce0000000p–5, –0x1.1584300000000p–2, –0x1.8a80b40000000p–2,
0x1.0494b00000000p–1, –0x1.f36b9e0000000p–2, 0x1.dd89840000000p–1, 0x1.ff75620000000p–1,
–0x1.5c91720000000p+0, –0x1.05b4b40000000p–2, 0x1.3ad1ee0000000p–1, –0x1.36be5e0000000p–2,
0x1.a9fa840000000p–1, –0x1.9be53c0000000p+0, 0x1.c2e5120000000p–7, –0x1.e7ea940000000p–2,
0x1.6053300000000p–1, 0x1.283cd00000000p–1, 0x1.78c36a0000000p–1, –0x1.973f980000000p–1,
–0x1.a0ce1c0000000p–7, –0x1.9b70f60000000p–3, 0x1.a71f340000000p–2, –0x1.88e55a0000000p–2,
0x1.57e9a40000000p–5, –0x1.63deb60000000p–3, –0x1.a95f460000000p–5, 0x1.0bfc700000000p–3,
0x1.a008a20000000p–4, –0x1.5de1f80000000p–7, –0x1.2492880000000p–2, 0x1.2f429e0000000p–2,
–0x1.629b440000000p–3, 0x1.57a2a40000000p–8, 0x1.c4069e0000000p–5, 0x1.19454e0000000p–2,
0x1.2414120000000p–4, –0x1.87f62e0000000p–5, –0x1.53d83c0000000p–4, 0x1.38ac100000000p–5,

Table B.19 Layer 3 CNN (bias parameter bias) of base decoding neural network parameter

Parameter value
–0x1.49a5f80000000p–4, –0x1.6ea38e0000000p–5,

Table B.20 Layer 3 CNN (IGDN activation function beta parameter) of base decoding neural network parameter

Parameter value
0x1.03ca7e0000000p+8, 0x1.38580e0000000p+8,

Table B.21 Layer 3 CNN (IGDN activation function gamma parameter) of base decoding neural network parameter

Parameter value
0x1.ebbe640000000p–34, 0x1.5e6fec0000000p–36,
0x0.0p+0, 0x0.0p+0,

Table B.22 Layer 4 CNN (convolution kernel parameter kernel) of base decoding neural network parameter

Parameter value
–0x1.04527a0000000p+1, –0x1.4c0b340000000p–4,
0x1.a286c20000000p–4, –0x1.872ccc0000000p+0,
–0x1.3a54b80000000p–3, 0x1.295ea60000000p–7,
0x1.2f833a0000000p–2, 0x1.dae1dc0000000p–2,
0x1.ebc0880000000p–9, –0x1.cf3be20000000p–4,

Table B.23 Layer 4 CNN (bias parameter bias) of base decoding neural network parameter

Parameter value
0x1.a12a9e0000000p–2,

Table B.24 mcILD code table

Index	Index value
0	1.777777778
1	0.750000000
2	0.562500000
3	3.200000000
4	5.333333333
5	0.812500000
6	1.066666667
7	4.000000000

Index	Index value
8	0.187500000
9	1.142857143
10	0.437500000

Table B.24 (continued)

Index	Index value
11	1.454545455
12	0.125000000
13	0.625000000
14	2.285714286
15	0.500000000
16	16.00000000
17	2.000000000
18	0.875000000
19	0.250000000
20	1.333333333
21	0.375000000
22	1.600000000
23	8.000000000
24	0.687500000
25	0.062500000
26	1.230769231
27	0.312500000
28	0.937500000
29	2.666666667

Table B.25 Dimension 1 Huffman code table tnsCodingTable0 for TNS reflection coefficient quantization indexes

Index	Codeword	Number of bits
1	4053	12
2	1012	10

Index	Codeword	Number of bits
3	507	9
4	127	7
5	30	5
6	0	3
7	1	3
8	2	3
9	2	2
10	3	3
11	6	3
12	14	4
13	62	6
14	252	8
15	2027	11
16	8105	13

Table B.26 Dimension 2 Huffman code table tnsCodingTable1 for TNS reflection coefficient quantization indexes

Index	Codeword	Number of bits
1	15360	15
2	7681	14
3	3841	13
4	961	11
5	241	9
6	61	7
7	14	5
8	2	3
9	2	2
10	3	2
11	0	2
12	6	4
13	31	6

Index	Codeword	Number of bits
14	121	8
15	481	10
16	1921	12

Table B.27 Dimension 3 Huffman code table tnsCodingTable2 for TNS reflection coefficient quantization indexes

Index	Codeword	Number of bits
1	27136	15
2	27137	15
3	3393	12
4	425	9
5	107	7
6	52	6
7	12	4
8	7	3
9	0	1
10	2	2
11	27	5
12	213	8
13	849	10
14	1697	11
15	6785	13
16	27138	15

Table B.28 Dimension 4 Huffman code table tnsCodingTable3 for TNS reflection coefficient quantization indexes

Index	Codeword	Number of bits
1	8708	14
2	8709	14
3	8710	14
4	1089	11

Index	Codeword	Number of bits
5	273	9
6	137	8
7	35	6
8	5	3
9	0	1
10	3	2
11	9	4
12	16	5
13	69	7
14	545	10
15	8711	14
16	4352	13

Table B.29 Dimension 5 Huffman code table tnsCodingTable4 for TNS reflection coefficient quantization indexes

Index	Codeword	Number of bits
1	4100	14
2	4101	14
3	4102	14
4	257	10
5	65	8
6	17	6
7	5	4
8	0	2
9	1	1
10	3	3
11	9	5
12	33	7
13	129	9
14	513	11
15	4103	14

Index	Codeword	Number of bits
16	2048	13

Table B.30 Dimension 6 Huffman code table tnsCodingTable5 for TNS reflection coefficient quantization indexes

Index	Codeword	Number of bits
1	8272	14
2	8273	14
3	2069	12
4	516	10
5	128	8
6	65	7
7	17	5
8	5	3
9	0	1
10	3	2
11	9	4
12	33	6
13	259	9
14	1035	11
15	8274	14
16	8275	14

Table B.31 Dimension 7 Huffman code table tnsCodingTable6 for TNS reflection coefficient quantization indexes

Index	Codeword	Number of bits
1	13312	14
2	13313	14
3	3329	12
4	833	10
5	209	8
6	53	6

Index	Codeword	Number of bits
7	12	4
8	2	2
9	0	1
10	7	3
11	27	5
12	105	7
13	417	9
14	1665	11
15	13314	14
16	13315	14

Table B.32 Dimension 8 Huffman code table tnsCodingTable7 for TNS reflection coefficient quantization indexes

Index	Codeword	Number of bits
1	10490	14
2	2625	12
3	657	10
4	165	8
5	83	7
6	21	5
7	4	3
8	3	2
9	10497	14
10	0	1
11	11	4
12	40	6
13	329	9
14	1313	11
15	10498	14
16	10499	14

Table B.33 TNS reflection coefficient scalar quantization codebook tnsCoeff4

Index	Codeword
1	–0.9957341763
2	–0.9618256432
3	–0.8951632914
4	–0.7980172227
5	–0.6736956436
6	–0.5264321629
7	–0.3612416661
8	–0.1837495178
9	0.0000000000
10	0.2079116908
11	0.4067366431
12	0.5877852523
13	0.7431448255
14	0.8660254038
15	0.9510565163
16	0.9945218954

Table B.34 Stage 1 first sub-vector codebook lsf_stage1_CB1_hbr of high-precision LSF vector quantization codebook

Index	Codeword
1	12.532774, –294.914795, –839.115295, –1283.263306, –1366.740234 –1291.976929, –889.037109, –662.501465, –394.010132
2	7.659530, 64.449364, –429.962616, –159.504883, –127.081932 –528.241577, –898.911499, –1261.207764, –1585.741333
3	–265.316223, –1177.764160, –1719.669189, –2321.858154, –2539.730713 –1376.201050, –393.688843, –135.207047, 153.807602
4	–429.040771, 1138.646240, 1062.909912, 1183.278198, 1069.893311 1066.192871, 987.967041, 938.215393, 848.846802
5	–259.962219, –841.635864, –792.817505, –1015.578674, –1039.584106 –1190.869629, –1251.823975, –1344.401611, –1400.721069
6	–255.383240, –340.604065, –225.726089, –194.386597, –82.405243

Index	Codeword
	–87.643135, 83.903320, 127.325462, 214.289505
7	–292.835663, –1187.095581, –1540.663330, –946.716309, –604.317749 –340.563690, –148.200073, –110.725548, –88.781868
8	–66.158363, 288.254822, 52.301933, –176.605835, –372.656982 –638.140747, –765.024902, –1038.674561, –1158.282227
9	–294.390991, –419.236633, 164.241470, 72.962151, 184.964706 –5.999948, 18.262583, –294.669708, –360.836609
10	–354.790070, –980.296875, –664.455078, 154.531082, 347.748291 –81.605064, –166.168503, –575.064392, –619.221313
11	–51.074921, –590.091919, –995.163879, –1449.023193, –1636.229980 –1833.939575, –2052.215088, –2393.046387, –2492.594238
12	–756.212219, 607.650330, 1362.533325, 1448.326538, 1708.366821 1743.075806, 1864.355957, 1851.665405, 1904.776489
13	–268.823273, –670.716125, –729.266479, –1178.513184, –717.594543 1188.218018, 973.420044, 669.331726, 253.003632
14	–380.711853, –880.759583, 437.412231, 377.531342, 279.782745 –164.422226, –350.319489, –689.558228, –759.321899
15	199.626038, 259.191193, –255.783997, –712.039185, –1172.695801 –1557.458618, –1920.221191, –2091.199219, –2199.324707
16	–45.761089, 121.817169, –221.829346, –605.307129, 50.221458 –352.458374, –718.818115, –1095.817383, –1017.988525
17	–324.111664, –1289.777954, –1839.569458, –2347.638672, –1012.049316 –69.315178, 187.488708, 379.120422, 501.824005
18	–297.901764, –1236.970581, –1812.303833, –2200.946045, –986.018372 –547.798462, –307.574707, –199.468140, –167.143997
19	–107.839973, –595.928772, –649.595825, –1203.271484, –417.419617 –843.091003, –1060.114380, –1251.820312, –790.997131

Table B.34 (continued)

Index	Codeword
20	–202.438538, –1077.976929 –1395.944458 –1834.929443 –1851.974121 –475.270538, 147.567062, 405.381622, 628.118896
21	–140.992111, –243.299820, –374.084686, –761.737305, 379.897888 146.736359, –218.489792, –693.225525, –1122.441040

Index	Codeword
22	298.091644, 165.684982, –229.728912, –676.614502, –41.712887 –241.386246, –313.544312, –564.628601, –634.669128
23	–339.094910, –932.599060, –810.994873, –1159.873413 –139.622528 –477.156525, –791.750977, –613.249939, 193.185974
24	192.596329, –448.119476, –199.818085, –711.967346, –599.650146 –906.385376, –844.235535, –1146.109009 –1127.517456
25	–153.114761, –477.945557, –842.755737, –1287.280884 –868.664062 –659.642517, –409.744415, –252.595947, –137.149399
26	468.480865, 188.345016, –232.100403, –407.251648, –446.181030 –654.295837, –737.947449, –982.220154, –1033.285034
27	–127.511642, –746.018372, –1140.810181 –1449.175049 –1522.306641 –1604.072266 –1617.172607 –1747.286011 –1815.891724
28	–670.408997, 573.691101, 818.152100, 1135.519531, 1336.889160 1511.981445, 1608.480469, 1687.401001, 1774.689819
29	167.956970, 304.036682, –168.396362, –614.536133, –1020.216125 –1307.400024 –1464.742065 –1539.064941 –1679.836670
30	–286.643921, –820.866394, –919.530457, –1296.737915 –321.350098 –247.957581, –74.895782, –46.720711, –16.407169
31	177.128052, –597.696228, –1405.832397 –661.515259, –1250.172363 –1315.480957 –1228.537109 –1394.990479 –1568.131226
32	–110.358360, 228.341812, –54.777168, –180.577560, 17.660156 –109.016502, –293.451660, –733.721802, –1050.250854
33	–155.538376, –889.169373, –1357.671143 –1947.136353 –2360.618164 –2616.920898 –2678.006104 –2436.539795 –1598.749878
34	–243.543640, –406.297119, –323.730835, –427.441620, –504.955658 –723.262573, –574.874329, –697.197388, –441.316406
35	–248.952698, –476.071716, –109.808342, –350.722839, 1045.941162 772.961060, 494.033112, –5.088210, –161.959808
36	151.197754, –438.898071, –938.110779, –479.895996, –675.861450 –705.350769, –794.681152, –874.259277, –977.368958
37	228.726776, –288.248199, –45.990913, –358.880219, –273.076996 –574.781677, –553.036743, –901.941895, –933.009277
38	76.154396, –272.780579, –790.993469, –1419.796875 –1771.096313 –1205.924927 –1421.059082 –1131.960205 –634.587891

Table B.34 (continued)

Index	Codeword
39	–176.957001 –998.468262 –1355.086548 –1801.816040 –1972.916626 –2021.075684 –1891.074829 –1003.528870 –271.320404
40	–318.367218 –1027.510376 –1053.401367 –1400.671631 –688.487488 –59.555279 173.477051 293.360443 413.468353
41	–31.048168 –539.577820 –779.418274 –1198.360352 –1667.825073 –1355.676392 –248.139374 –578.224731 –848.542908
42	243.113541 –271.529938 75.644211 –120.757767 –23.166691 –275.094727 –270.071869 –566.301392 –527.317078
43	209.083649 –86.944138 –171.801407 –309.664154 –601.843628 –1093.211426 –1161.305298 –629.750732 –553.044739
44	50.228077 260.481262 –176.737579 –563.293579 –836.673584 –983.849121 –900.034058 –1054.167847 –1024.612671
45	–217.254745 –1059.019897 –1551.903564 –2159.453125 –2555.874512 –2660.438965 –1700.472412 –939.193237 –491.305817
46	–109.810867 –139.434006 –212.173462 189.489395 433.778625 328.265594 316.352631 29.588856 –62.139759
47	–122.472839 –77.221489 –139.844727 –493.818024 –217.068741 444.526581 186.679306 –248.982697 –697.615417
48	–283.886810 –634.963989 –187.824463 –160.884415 225.083527 –196.268967 –448.745483 –961.819641 –1293.330566
49	–205.688416 –818.517212 –1049.939453 –1471.114380 –1637.467896 569.867554 289.272095 33.614040 –107.433998
50	–470.968445 74.234940 –189.621552 93.866119 712.355530 1110.288940 1314.786499 1475.519287 1557.706787
51	–258.598328 –1029.964111 –1337.363525 –1035.258911 –921.234985 –896.027100 –776.560547 –744.058167 –680.859985
52	–304.905701 –1248.585083 –1799.832031 –2462.894287 –2100.090576 –516.581787 –119.507423 130.499710 273.351562
53	234.289627 1074.261353 614.698914 125.832085 –265.399841 –442.323151 –487.567566 –416.854156 –340.315491
54	125.100708 –360.461273 –764.823303 –1421.218750 –967.333435 –1095.811401 –1196.823486 –1400.642334 –1383.499878
55	–133.548233 –617.994385 –312.143127 –710.831909 79.074684 –298.849487 –592.688049 –1062.510376 –562.840149

Index	Codeword
56	–468.169861 1446.408569 1413.416870 1576.199707 1347.820801 1305.200928 1179.695923 1049.632812 898.322266
57	–274.492950 –345.295959 –346.863892 –393.565643 –159.954437 –288.083282 –280.803223 –615.538269 –798.523987

Table B.34 (continued)

Index	Codeword
58	414.343719, –33.507385, –560.052307, –877.893921, –936.769043 –889.978210, –848.494446, –983.392456, –1004.623169
59	–127.857430, 10.724673, –503.393402, –1047.103760, –1597.348145 23.368412, 119.567314, –27.045931, –122.189880
60	50.129940, –513.716614, –175.773712, –814.073059, –183.152222 –398.482635, –736.121338, –1233.094971, –1505.156860
61	–192.924240, –1000.330994, –1506.831055, –2159.660400, –2665.731934 –3122.868652, –3047.991943, –1299.960693, –665.220581
62	–229.910477, 371.929138, 198.593094, 147.880417, 147.134155 59.440495, 74.496574, –130.838730, –153.672745
63	31.351290, –454.428986, –421.578094, –1073.841797, –293.511871 –674.466675, –667.398499, –510.883423, –513.513489
64	1165.111938, 1601.962036, 1369.250000, 1091.164673, 806.724060 592.549011, 437.339203, 274.545990, 130.017639
65	–391.330902, –1148.816284, –703.030945, –919.529114, –682.469421 –552.763611, –389.571045, –372.423248, –269.741028
66	–415.906036, –1370.143433, –467.918060, 400.263123, 538.698425 626.269958, 628.958679, 609.006653, 555.666260
67	–211.888199, –304.512207, –574.377808, –763.588745, –798.061340 –969.718689, –1008.344788, –1133.811157, –1208.669189
68	387.929260, –384.363983, –491.873169, –1177.873779, –2040.045410 –2299.358643, –879.417908, –600.336487, –538.996033
69	–194.664413, –432.293304, –354.330627, –721.107178, 220.619720 –28.813951, –13.943194, –279.962128, –211.170929
70	–254.518509, –1089.525269, –1553.118164, –1953.478149, –1708.065063 –1163.100708, –875.388916, –675.993652, –554.956543
71	45.468208, –177.159286, –486.094971, –971.243835, –326.891663

Index	Codeword
	–774.877563, –1253.826904, –1702.123291, –1697.666016
72	–617.304138, 1302.192261, 1183.291016, 1409.274414, 1356.658203 1509.198608, 1565.556396, 1796.875854, 2312.282715
73	–318.085754, –294.912109, –243.064804, 294.789520, 587.472717 685.794067, 779.710144, 827.043884, 850.107422
74	–675.562988, 1130.372803, 1039.005249, 1216.456055, 1123.337158 1227.838257, 1186.562622, 1320.220581, 1376.252319
75	–155.287216, –653.789795, –979.381104, –1584.975708, –907.041931 –791.481995, –893.957764, –690.019653, –588.572449
76	1085.995117, 433.405426, 23.847609, –363.684875, –394.045135 –519.063965, –459.017670, –605.080383, –566.073059

Table B.34 (continued)

Index	Codeword
77	–39.306290, –267.335541, –399.840454, –361.051941, –369.583893 –813.095459, –1069.663818, –1484.840576, –729.816772
78	–157.645782, –441.672302, 196.467331, –31.353685, –266.225250 –728.974548, –1055.533203, –1360.301636, –1394.537109
79	–302.830658, –981.803101, –1199.857910, –1471.296997, –406.638672 –282.785950, –374.713959, –592.583313, –744.692444
80	–414.440247, –1089.211548, –650.316528, –1022.616089, –497.021057 257.376678, 508.734467, 967.452759, 1139.718262
81	629.385315, 46.781452, –475.762695, –1036.605469, –1335.133911 –1327.513062, –1000.840942, –768.826721, –775.389404
82	–14.132797, 483.183777, 1040.778809, 889.231567, 552.954224 70.982803, –318.060852, –793.120911, –1011.159302
83	135.842896, 114.402382, –402.932617, –912.733643, –456.571472 –654.064758, –907.594116, –1166.712891, –1294.101074
84	–405.660950, 650.800964, 390.291229, 402.028564, 397.279541 643.243591, 782.320801, 996.435913, 1037.156860
85	739.563538, 198.752533, –274.114716, –756.031982, –1016.008545 –397.001160, –507.254028, –710.850891, –777.544189
86	–505.712128, –1306.856934, –302.960205, –50.899620, 724.131470 892.069763, 1116.432861, 1241.490356, 1347.699341

Index	Codeword
87	–185.285233, 637.581787, 570.410583, 573.242920, 520.846680 458.776489, 506.174042, 430.875732, 434.773529
88	1638.158081, 1885.557617, 1226.245117, 752.522339, 329.570221 –7.930009, –429.244537, –698.395142, –824.893433
89	67.108162, 145.657013, –310.819153, –690.611633, –588.102966 –149.363922, –405.786072, –809.160034, –1272.879028
90	–585.281555, 595.188293, 708.387085, 891.753052, 958.253723 1118.726318, 1189.508423, 1320.166870, 1432.868286
91	–149.615555, –240.052521, –648.237000, –988.918823, –466.479462 –76.178154, –229.023560, –486.511444, –608.023254
92	–430.823914, –1171.099731, –536.025940, –441.452942, 255.522385 338.995361, 481.303986, 464.989960, 568.602966
93	–324.765167, –1047.405151, –1045.352539, –363.335541, –84.914948 –66.833885, 30.027208, –15.760809, 12.066683
94	–175.676392, –408.857758, –583.354431, –648.645386, –479.757019 –418.775604, –267.904999, –147.864532, –25.590437
95	–208.584457, –939.634705, –1357.809814, –1543.731812, –1245.787964 –1208.176758, –1149.390625, –1191.954224, –1206.649658

Table B.34 (continued)

Index	Codeword
96	–104.840843, 131.496292, –209.161987, –429.285919, –443.612244 190.636688, 472.597717, 768.328491, 861.614197
97	–407.797485, –1443.926514, –1661.918823, 207.416931, 534.250854 731.394470, 835.406738, 833.161926, 767.225098
98	1845.772217, 1309.853271, 668.536926, 343.226593, 143.572464 –49.394199, –113.645187, –29.705999, 13.073053
99	56.131977, –152.148804, –464.899872, –1033.450439, –1184.909790 –460.081299, –810.854675, –1229.766235, –1627.935913
100	–452.430084, –1235.425537, –176.435577, –54.552139, 201.717514 89.107384, 104.389801, –15.340947, 55.963818
101	–294.794617, –982.590759, –1157.254761, –1552.222290, –1173.347168 –776.471863, –508.820099, –345.842194, –264.661591
102	422.903564, –77.037788, –533.205566, –1148.150024, –1583.096802

Index	Codeword
	–1563.172607, –1482.618774, –1418.075928, –1232.485107
103	349.948730, –169.888306, –473.498138, –799.142944, –1174.529297 –1534.228882, –856.638062, –1243.882568, –1346.036865
104	331.781677, –83.041771, –525.813416, –1026.352173, –1185.363159 –1167.713379, –1430.438721, –1703.381348, –1894.847290
105	–163.386047, –482.613098, –367.271667, –620.273438, –823.262390 –1089.952515, –1272.664673, –1561.849487, –1797.645264
106	–94.470627, –55.125443, –547.598450, –970.332947, –1201.722412 –1190.345947, –1261.533813, –1256.466187, –1336.214600
107	–280.255554, –122.187737, 222.988007, –0.161418, –149.800110 –457.114594, –539.179016, –763.223145, –744.701721
108	–226.694626, 146.819641, 99.686066, 156.132889, 264.517853 255.099182, 390.046783, 413.388489, 484.449554
109	–193.396912, –668.117432, –935.539307, –1391.381470, –1675.302490 257.404022, 755.294250, 780.717102, 659.994995
110	–351.291901, –1336.534912, –1761.683838, –812.096375, 147.030319 253.470673, 430.642029, 409.021210, 470.170197
111	–334.503387, –847.869446, –543.447876, –818.843140, –453.180328 –681.399109, –369.549103, –800.736877, –985.739807
112	–169.888916, 188.977081, 38.719604, –193.778885, 663.089172 407.570740, 102.907455, –395.730255, –696.857788
113	–597.549011, 11.597784, 173.840820, 871.852783, 1169.863525 1430.254272, 1538.062988, 1649.076782, 1709.838135
114	110.810623, –260.302307, –717.694092, –1216.236084, –1650.935913 –1598.782959, –1049.980835, –991.954529, –953.349915

Table B.34 (continued)

Index	Codeword
115	–149.068253, –724.761475, –1071.322144, –1674.264038, –2245.992676 –310.163513, –453.569458, –402.124115, –212.454956
116	–53.550934, –631.069824, –720.459900, –1071.667725, –1272.118164 –1691.381958, –1747.413452, –887.452759, –563.786804
117	93.887642, –169.664261, –540.245117, –1219.269531, –823.180481 –1033.660400, –1287.411987, –699.180481, –743.583679

Index	Codeword
118	106.536568, 131.629593, –247.396606, –506.335999, –527.198425 –326.019135, 14.460602, 319.439056, 400.146210
119	345.794647, –128.985657, –553.267151, –1045.432983, –1151.686646 –834.284668, –627.922363, –414.898682, –315.911713
120	16.523335, –81.917313, –481.277618, –28.254915, 41.456417 –76.575104, –110.086128, –321.329346, –374.198547
121	25.006420, 2178.584473, 2119.460693, 1804.463623, 1368.770874 973.360657, 580.718323, 271.132294, 10.037239
122	–207.521225, –389.666595, –658.753906, –1088.745239, –147.729385 –409.515961, –591.869751, –895.262207, –1240.875854
123	431.983582, 279.939575, –40.320492, –317.882599, –521.578796 –840.770447, –1032.591431, –1336.981812, –1490.085083
124	–6.341250, 951.111572, 1478.494995, 1295.491821, 970.222351 504.117615, 163.589752, –330.766846, –542.744446
125	–275.387726, –907.883240, –428.596405, –844.338684, –750.426392 520.789246, 191.068069, –101.187233, –342.247009
126	–66.128357, –356.209076, –771.646179, –1201.168701, –840.965698 –289.156830, 20.764372, 132.631866, 253.929382
127	566.312805, 529.282288, 382.089203, 246.211899, 92.596085 –190.611786, –343.605133, –657.513916, –751.330872
128	–89.353828, –615.056641, –164.835449, –727.441833, –718.769043 95.425873, –281.069977, –664.238403, –1059.275757
129	–246.166534, –842.018738, –1040.614258, –406.870605, –277.347656 –449.857819, –463.342133, –622.325012, –565.881592
130	–326.831329, –787.752136, –173.408951, –412.883850, 580.736755 336.680817, 40.913807, –444.342743, –772.719482
131	9.320523, 332.287872, 385.677032, 785.866394, 891.927979 525.718018, 197.978317, –313.102997, –582.461182
132	929.624023, 297.050354, –158.697464, –666.646606, –876.853760 –1016.439880, –1018.763245, –1223.214111, –1140.760864
133	140.714645, –163.534546, –663.564331, –1197.529541, –1644.409424 –1826.487671, –1873.009277, –1903.410522, –1918.154663

Table B.34 (continued)

Index	Codeword
134	129.709396, 1713.367920, 1161.276245, 707.854980, 299.398163 162.641678, –26.087206, –77.141838, –171.849213
135	–205.154602, 110.982040, –142.183319, –365.562744, –301.546692 –279.877960, –196.408676, –350.420135, –323.925201
136	–69.882439, 99.225761, –316.386963, –783.445007, –774.641357 –866.159729, –460.005524, –366.746399, –196.000290
137	–161.529800, –873.052002, –820.817627, –1312.994629, –1011.838318 –781.103821, –1038.130737, 533.575745, 471.384552
138	–374.898529, –888.264526, –167.955841, –273.180542, –202.054932 –368.927612, –237.626297, –390.566956, –269.131531
139	194.377274, 436.584747, 307.389587, 374.059601, 419.120728 301.247375, 322.779480, 77.648521, 34.339844
140	–77.034248, 685.388672, 415.089294, 122.450302, –161.383041 –452.515381, –613.104004, –873.833557, –951.395569
141	–386.433350, –1145.266479, –1259.791016, –1136.691406, 545.907166 747.837524, 923.854492, 916.237183, 975.617310
142	–247.674072, –890.098083, –707.935608, –1260.291138, –1387.769531 –469.448212, –107.897461, 29.525518, 187.592911
143	75.124016, –594.743408, –319.550049, –1086.576904, –1389.873901 –1009.383911, –1102.857422, –1238.913208, –1034.670898
144	–213.455551, –1097.307861, –1620.846313, –2178.167725, –2369.833252 –1891.464478, –1046.413208, –597.500854, –269.395081
145	–120.505707, –545.911987, –856.895325, –1281.255371, –1845.679321 –1432.887329, 619.328125, 417.883484, 233.810257
146	–15.592245, –329.802032, –900.705811, –1522.858643, –2115.699707 –2749.846924, –3296.389648, –3688.250977, –3953.603027
147	–82.807320, –749.621643, –1257.475830, –1907.140503, –2538.343750 –3129.432129, –3701.928955, –3864.471191, –1477.994141
148	–452.950867, –441.102600, 869.980042, 831.790283, 1031.825684 875.656311, 909.180115, 737.284363, 744.969055
149	–148.392014, –888.583252, –1215.410522, –1655.602417, –2103.372803 –2140.581055, –394.873322, –189.024429, 99.190369
150	668.062195, 1458.963623, 1275.840820, 1355.906494, 1220.712769 1246.279053, 1182.607300, 1140.741455, 993.256104
151	–256.370544, –1082.097290, –1411.395996, –1818.622192, –1726.187256

Index	Codeword
	–1003.165771, –337.975769, –137.042953, 38.859200
152	–366.040039, –1343.038208, –1263.489624, –12.731862, 325.425690 409.669739, 441.824127, 415.941864, 374.911804

Table B.34 (continued)

Index	Codeword
153	–112.508049, 54.832523, 231.882675, 534.124878, 808.231689 746.702209, 716.196777, 473.386963, 386.079987
154	–375.002533, –1194.725464, –1038.355469, –1037.824829, –11.491217 117.387360, 292.925751, 285.436829, 373.991669
155	–110.031464, 402.007935, 218.209946, 103.379486, 8.490447 –179.653946, –258.198151, –527.337402, –586.289978
156	97.160622, 1709.108643, 1261.806885, 869.773987, 438.982056 32.777740, –373.052063, –746.220032, –967.538818
157	–47.745808, 137.209152, 168.656677, 88.941284, –78.959335 –387.421143, –421.709259, –261.284119, –19.217987
158	525.404297, 244.304626, –157.482239, –522.035645, –503.885803 –472.361633, –211.423141, –210.994324, –100.604553
159	–64.922203, –577.785645, –692.792053, –977.476746, –1072.638306 –1360.974854, –1721.664429, –2073.228027, –2290.486816
160	–250.663101, –1146.197388, –1699.822632, –2325.511963, –2795.845459 –2657.724609, –835.075684, –357.637390, 10.896358
161	–171.140884, –654.061401, –598.868103, –1201.529785, –1382.656372 –849.022766, –794.400818, –573.129028, –378.523499
162	1585.175171, 1531.839478, 1551.516235, 1452.569946, 1442.305908 1338.680786, 1311.629150, 1214.936279, 1141.125244
163	–15.817514, –411.485168, –394.904633, –753.658813, –597.352112 –668.474182, 116.694031, –271.499939, –593.399780
164	68.322708, –326.461426, –433.361847, –998.881042, –920.705383 –390.451447, –750.696716, –1122.629150, –548.098206
165	–308.922272, –1033.517578, –1245.763550, –1659.946777, –963.066772 –433.620239, –130.026184, –19.373592, 55.147762
166	396.746063, 360.242676, 2.003803, –101.434189, –69.522614 –177.138092, –174.745331, –394.647644, –427.269562

Index	Codeword
167	–199.146637, –455.867065, –788.567322, –982.193665, –794.411926 –802.438049, –647.405396, –676.345154, –567.256836
168	–431.786865, 62.282070, 216.347824, 678.388000, 795.189697 954.403870, 1031.630981, 1167.656860, 1262.549561
169	708.548950, 192.753342, –206.201889, –693.366821, –838.465393 –1190.832153, –358.575958, –655.519409, –667.919006
170	–24.659691, 646.923035, 357.575531, 42.665451, –318.174011 –703.991577, –1065.087036, –1389.942993, –1617.854614
171	–457.514343, 523.834534, 812.696045, 919.779358, 959.375000 926.008667, 893.313049, 813.890991, 772.358887

Table B.34 (continued)

Index	Codeword
172	–252.229111, –205.577118, –99.335968, –261.883484, 477.315582 0.361599, –358.287689, –819.807617, –474.220642
173	–53.019951, –151.925522, –92.089211, 110.928047, –146.320724 –725.104370, –78.699577, –230.563110, –788.255493
174	–165.916931, –780.054993, –682.217590, –808.404785, –1022.207703 –1302.715820, –362.507385, –233.079498, –63.115982
175	–293.424652, –879.620972, –768.047363, –469.037689, –368.898315 –681.270874, –910.819763, –1210.304688, –1444.320190
176	–70.139114, –411.455475, –836.225708, –1157.311279, –1184.458862 –1145.269043, –1037.999146, –1040.423218, –917.642273
177	458.907898, 239.671219, –274.590546, –997.059631, –456.334076 –879.237061, –834.715515, –929.808228, –453.293610
178	–201.639328, –373.349304, –552.528625, –328.857635, 297.073822 401.934570, 530.450989, 488.733856, 495.095428
179	368.688141, 1257.926880, 814.361023, 495.116058, 46.022434 –361.370819, –752.548462, –1097.315186, –1236.272583
180	85.106880, 208.105011, –91.348747, –327.705048, –615.689270 –1000.071167, –1401.193359, –1803.255371, –2100.476807
181	–14.701283, –503.161835, –825.721863, –1459.594971, –2056.221924 –1266.375000, –921.267029, –558.592896, –308.952728

Index	Codeword
182	2364.062988, 2268.497803, 1553.899414, 1021.092712, 683.231995 391.977448, 151.044174, 16.674553, –95.852577
183	–118.223747, –668.176819, –1049.796265, –1510.749268, –1862.077271 –1833.617554, –1126.837524, –915.571106, –646.829102
184	81.282196, –137.213165, –536.142334, –1273.106079, –744.081238 –1123.636353, –557.924438, –875.544250, –944.424561
185	–146.028366, 965.887390, 669.985779, 441.832611, 227.827728 –48.675701, –282.892731, –611.419556, –752.563721
186	–215.103439, –891.118774, –1128.932861, –1478.110840, –1518.857300 –1391.129761, –784.654297, –464.452179, –218.385284
187	–104.318695, –755.709900, –1227.947144, –1774.801636, –2108.293213 –2460.826416, –2636.740479, –2869.123779, –2697.159424
188	374.387817, 305.442200, 17.710152, –78.148804, –160.953735 –425.886536, –515.567261, –755.588318, –765.299072
189	–25.625576, 344.318176, –43.530491, –324.637817, –482.693909 –559.479553, –535.363708, –679.792480, –645.745972
190	–722.385986, 1108.328003, 1432.110596, 1615.616211, 1747.338501 1839.730713, 1887.988037, 1897.838745, 1882.956177

Table B.34 (continued)

Index	Codeword
191	–216.053711, –774.596313, –807.091064, –1286.393799, –868.596130 –644.140198, –719.721497, –999.943665, –1312.422852
192	347.996399, –73.063454, –430.095886, –405.626465, –424.762238 –505.689728, –474.230194, –602.869690, –516.047668
193	–121.138519, 599.921997, 1637.786133, 1673.657104, 1515.356201 1099.747314, 782.899292, 290.842285, 50.356499
194	–479.236481, –1292.459961, 486.444305, 617.295349, 807.077637 742.250183, 786.319092, 690.602112, 686.478699
195	–384.315399, –1041.603760, –388.681671, –720.672424, –491.007507 –261.076233, 70.605843, 193.263412, 359.707428
196	726.636963, 1068.222656, 835.229492, 646.974976, 372.526855 81.769661, –155.821640, –467.406036, –609.194092
197	1011.585083, 1131.065430, 551.508789, 84.949585, –427.427521

Index	Codeword
	–899.412659, –1345.255493, –1725.048218, –1932.867798
198	–104.705009, –570.926758, –903.287659, –1329.082520, –1503.404053 –1593.681519, –1433.338745, –1354.490723, –1103.630737
199	684.434998, 2677.031738, 2640.076416, 2477.602783, 2097.601562 1757.809448, 1344.368896, 984.018860, 607.138794
200	–275.864197, 29.051449, –274.191254, –358.941620, 46.813194 657.108032, 913.113220, 1191.818359, 1285.592896
201	617.544373, 809.540405, 751.557190, 656.415466, 539.906250 344.674286, 311.402069, 104.217766, 43.276100
202	–570.533142, –1292.932617, 781.957703, 861.536133, 1279.650024 1209.266235, 1395.982300, 1345.671509, 1472.285889
203	613.478943, 498.496368, 221.305801, –10.624701, –228.434753 –553.930908, –738.420349, –1059.372314, –1161.117065
204	–99.197083, –364.513275, –658.022156, –1038.899536, –1296.954346 –1541.067017, –1558.542969, –1601.610840, –1636.576050
205	1902.935669, 2657.081543, 2283.650879, 1829.887695, 1418.237427 1029.569336, 635.649658, 348.315186, 57.349442
206	–169.365402, –779.464294, –807.007690, –1272.415161, –1381.448120 –31.502048, –271.029419, –526.351501, –776.372559
207	–372.196869, –999.805115, –67.133698, –185.357285, –272.310028 –589.583923, –690.671143, –871.234436, –813.840027
208	780.040894, 540.443909, 272.137726, 127.595924, 103.035805 –7.213268, 33.308098, –92.559692, –86.642143
209	–43.860863, –75.046074, –498.833160, 79.453178, –48.209049 –427.222076, –640.230713, –834.722412, –843.463928

Table B.34 (continued)

Index	Codeword
210	–84.002121, –803.222839, –903.731995, –1555.773071, –1820.684082 –645.273804, –843.073975, –983.579224, –1107.764282
211	287.774628, –39.120178, –597.361511, –1199.857666, –1640.720093 –873.398071, –894.083130, –927.508362, –891.872498
212	–189.896698, 782.072388, 781.725647, 674.643738, 497.708771 241.687012, 148.634705, –93.176682, –177.620422

Index	Codeword
213	–121.669731, –873.069214, –1265.429932, –1776.589233, –2045.400513 –2201.425537, –2075.754150, –2074.478760, –1888.381592
214	–169.272018, 30.012857, –109.015106, 465.255615, 453.288849 40.586933, –249.114258, –631.611816, –837.331116
215	–362.530945, –801.328735, 282.536194, 243.578491, 390.871216 297.866333, 479.047913, 369.696472, 405.338074
216	–238.607315, –852.011536, –964.665527, –1365.129395, –1391.019897 –1305.390503, –1077.980591, –937.738281, –683.551453
217	–85.789406, 182.220551, 330.619476, 318.162048, 156.505722 –267.389893, –595.257080, –1019.441650, –1273.395508
218	–187.219238, –403.721039, –729.755493, –965.228516, –133.767776 126.393250, 296.895660, 303.930176, 363.149719
219	310.347870, –47.092258, –511.695251, –1076.267456, –986.932312 –1167.762695, –1244.994751, –1354.601929, –820.385315
220	–97.799355, 442.941376, 811.982605, 957.969604, 918.645081 678.298462, 573.526001, 214.357635, 77.677307
221	–179.466064, –953.375244, –1399.735840, –1847.744751, –1956.635498 –1822.329590, –1526.700073, –1380.293213, –1160.773560
222	–653.149902, 1254.798218, 1124.465698, 1332.852051, 1239.890015 1379.581177, 1360.235107, 1535.566772, 1697.100464
223	–264.660797, 86.838188, 638.347168, 474.387848, 295.338104 –62.242229, –152.140320, –451.995605, –545.072205
224	–294.116180, –926.886108, –564.997986, –810.562195, –823.729614 –1049.912598, –951.003418, –904.211121, –781.321716
225	23.488888, –412.369110, –441.759216, –691.951904, –884.265442 –1368.422485, –1694.430542, –1727.004883, –1126.412109
226	–34.125507, –201.101593, –432.403564, –877.453430, –1181.488647 –619.693298, –423.591156, –702.601868, –1005.619995
227	–100.312309, –673.506653, –981.401489, –1410.046509, –1695.960571 –1886.820923, –2109.124023, –1959.750244, –1201.863892
228	–346.761017, –884.866821, –900.544067, –1077.662720, 619.997192 543.461304, 242.865036, –100.780441, –286.818024

Table B.34 (continued)

Index	Codeword
229	–200.410614, 109.605659, –170.653717, –421.138000, –651.408691 –912.773987, –1131.077881, –1341.235229, –1498.216187
230	572.233276, 755.728027, 730.481140, 826.995178, 843.460083 827.285400, 823.618469, 699.820312, 607.336975
231	–19.294380, –195.121170, –463.009827, –795.654175, –981.652588 –1232.025391, –832.356018, –925.299133, –594.547485
232	–406.959015, –1069.484375, –700.927673, –869.769348, 138.809525 –27.963032, –146.023773, –592.793091, –904.695923
233	358.704681, 2203.979248, 1786.071411, 1356.908936, 891.832825 512.293335, 67.515404, –327.043518, –622.445557
234	300.532776, 129.682983, 149.795502, 261.271667, 280.571259 67.060669, 35.455791, –271.818146, –376.685242
235	–201.779694, –71.861748, 512.022095, 501.403564, 545.430664 269.734314, 290.257141, 20.506559, –9.238894
236	220.352310, 646.220276, 485.153656, 378.431824, 239.117081 30.119192, –48.385849, –281.666260, –321.418243
237	–37.013084, –12.237608, –445.684906, –694.351868, –519.398560 –689.213684, –619.480469, –789.566101, –831.354065
238	–311.697540, –1214.498779, –1629.167725, –1667.692383, –252.691345 –41.316105, 138.973297, 128.725128, 171.948929
239	–177.504532, –999.752441, –1437.215332, –2003.558228, –2320.895508 –2459.510742, –2212.731934, –1705.895630, –941.731689
240	–64.721283, 798.213867, 379.698395, 88.914154, –168.081284 –58.058834, 73.794998, 323.422882, 401.970367
241	124.400887, –86.299744, –713.425903, –1270.068115, –1901.768066 –2410.684326, –2521.028076, –2378.131104, –2131.198730
242	–246.546921, –936.140015, –666.568726, –711.366394, –604.071777 –503.898346, 1039.701782, 805.238953, 313.919495
243	–288.440033, –241.610382, –246.760681, –401.584534, –492.142914 –692.678162, –740.253235, –969.517273, –1111.669189
244	–357.615021, –505.923218, 1234.285034, 1028.074219, 764.083801 298.424377, 180.949036, –154.882401, –236.654190
245	–159.627060, 1235.245239, 1178.883301, 1088.014160, 860.808533 636.503967, 566.731140, 383.741333, 251.244156

Index	Codeword
246	–54.214455, –417.550842, –859.441040, –1392.187866, –1611.388672 –927.907532, –386.700897, –194.817215, 14.651560
247	–742.290100, –15.748692, 1266.636108, 1328.531128, 1668.744019 1669.059204, 1832.402100, 1788.834961, 1862.470825

Table B.34 (continued)

Index	Codeword
248	–343.283539, –1302.701172, –1851.272339, –1961.326782, –130.270493 248.502701, 551.546631, 607.094482, 766.398682
249	107.124557, –148.659027, –554.413086, –982.619141, –1348.362915 –1730.620483, –2168.555664, –2596.479980, –2861.052490
250	–418.114899, –961.129211, –17.794352, 661.979614, 851.078003 437.627960, 205.163864, –273.702789, –388.516815
251	275.978882, 184.001373, –100.555817, –225.438446, 74.304443 22.188887, 179.736740, 0.251966, 10.995980
252	1615.943481, 1123.951782, 539.817871, 175.317184, –110.204140 –465.569305, –681.310852, –827.931396, –798.271057
253	580.269958, 370.878906, 110.667450, 134.644119, 306.276093 418.459473, 557.146606, 516.662964, 507.707855
254	36.891468, –515.519226, –762.948547, –1508.485718, –1802.767700 –1083.640381, –1172.074219, –1403.558838, –1596.932861
255	383.148956, –26.410944, –467.441589, –667.710754, –780.453552 –992.135803, –1154.880615, –1385.698730, –1454.032837
256	153.504349, –389.179504, –746.059998, –1376.398071, –2047.961426 –2373.300781, –1687.155884, –1366.466553, –1077.949463

Table B.35 Stage 1 second sub-vector codebook lsf_stage1_CB2_hbr of high-precision LSF vector quantization codebook

Index	Codeword
1	–2.419283, 160.314499, –65.667160, 145.171967, –234.788620 58.263897, –857.876892
2	–1078.523315, –1171.671143, –1471.139771, –1721.603516, –2394.023438 –3054.640137, –371.735413

Index	Codeword
3	–1384.490234, –722.422668, –1254.738892, –1515.970703, –1980.408691 –1571.614868, –2312.044922
4	–1211.196167, –1535.960938, –1990.369141, –1961.203735, –1703.231079 –1067.977661, –1775.004883
5	–1028.398438, –939.567017, –1238.632568, –1016.474548, –1560.545044 –1644.496826, –2429.453613
6	–1490.962769, –1297.616699, –1220.265747, –988.459229, –1302.738037 –1197.546387, –1958.636597
7	463.516296, 493.455872, 55.139450, 115.217438, –464.511414 –580.085999, –1564.193848
8	–1785.097778, –1858.684814, –2109.224365, –2145.851074, –2486.418213 –2603.061768, –3070.082275
9	–1086.836304, –1152.029175, –1538.223999, –1379.222656, –1861.346191 –1678.717773, –2455.295898

Table B.35 (continued)

Index	Codeword
10	–1128.577026, –1054.297852, –974.278442, –911.481689, –875.174438 –844.964905, –1078.336792
11	–469.022614, –829.216492, –1263.203735, –1220.989136, –1111.808350 –834.006348, –1571.638306
12	–1552.077637, –1593.502808, –1913.150513, –1876.412720, –2260.737793 –1906.382080, –2539.385986
13	–46.287777, 76.654182, –390.476898, –316.844574, –910.179382 –908.117615, –1883.781006
14	–352.165497, –324.392731, –374.813843, –377.579254, –412.347839 –616.016418, –1052.164185
15	2562.200684, 2434.767578, 2081.610840, 1673.590088, 1283.249878 942.633057, 707.649902
16	–503.909973, –505.138428, –966.220215, –990.389893, –1561.319458 –1457.232422, –2308.474365
17	–1655.097412, –1696.792847, –1498.722900, –1659.318237, –1431.109619 –1221.678345, –1568.875122
18	–721.078979, –461.355804, –801.716370, –720.517578, –1222.670288

Index	Codeword
	–1142.029541, –2031.396606
19	–1293.315674, –873.878784, –679.288696, 31.862846, –228.983414 –440.117798, –1383.703369
20	595.191284, 568.244507, 557.998779, 649.209656, 584.837952 555.886047, 474.990448
21	811.444946, 775.415100, 772.300293, 550.483521, 392.688019 –140.597336, –225.270554
22	–887.068237, –1424.254639, –2049.020752, –1434.494263, –1748.213257 –1317.314209, –2176.650635
23	–616.794067, –1045.691406, –1647.098145, –1025.717041, –1460.346680 –1325.690430, –2155.200195
24	–635.036682, –836.133301, 446.604218, 574.536316, 437.461243 360.389191, 355.061340
25	1443.341187, 1918.654541, 1750.556519, 1440.371948, 1099.758789 820.523560, 634.274231
26	–372.092194, –290.193817, –417.522156, –456.368469, –676.214355 –1011.395630, –1729.396240
27	–1407.802124, –1410.329712, –1326.036133, –1297.428345, –1112.815430 –1066.590942, –1198.377930
28	–1155.173584, –470.017883, –875.280945, –991.236267, –1473.512573 –1466.203979, –2238.273926

Table B.35 (continued)

Index	Codeword
29	–1100.306641, –265.063721, –716.102051, –924.100891, –1284.598145 –869.538757, –1762.164185
30	–1201.742798, –1308.445068, –1072.893188, –1381.481323, –1818.513550 –1411.241333, –2124.507324
31	–1093.597290, –1401.546509, –1534.675781, –1167.730591, –1508.403564 –1272.788940, –2043.262451
32	–754.049805, –706.918457, –1074.583252, –1259.205933, –2180.911133 –1706.644531, 1.634244
33	–1318.558105, –1511.099365, –1847.709717, –1602.843018, –1996.738037 –1653.422607, –2384.769775

Index	Codeword
34	–1110.535645, –1422.926270, –1907.578003, –2039.082153, –2278.691650 –1397.641602, –2057.705566
35	–297.689423, –405.208588, –590.064941, –701.502441, –636.377380 –546.131653, –241.727249
36	–473.099213, –404.824402, –347.584503, –261.104828, –228.279114 –188.513779, –20.013456
37	–1784.105713, –1652.451172, –1898.372681, –1465.228760, –1634.671509 –1010.233643, –1834.144043
38	–1227.201172, –1219.217651, –1632.298462, –1706.958496, –2169.429443 –1836.036255, –2525.359375
39	–706.488342, –746.910095, –1253.204346, –1326.531860, –1862.745117 –1672.360107, –2445.018555
40	854.645203, 292.497955, –411.134827, –850.383911, –1575.365845 –1757.057251, –2767.714600
41	–1319.187622, –1640.014404, –1963.474976, –1162.507080, –1304.337158 –958.915283, –1782.987671
42	–1232.779175, –1079.236206, –1447.466187, –1258.913452, –1644.954712 –1100.258301, –1906.573486
43	–847.552856, –713.231628, –752.985596, –389.162506, –754.285461 –648.719849, –1566.578369
44	–1708.502441, –1424.723267, –1653.111572, –1522.201782, –1966.691406 –2173.961670, –2931.311035
45	–1751.297852, –1319.098145, –1734.377686, –1953.311157, –2216.190430 –1501.015137, –2159.846436
46	–858.159241, –429.482086, –517.704590, –187.014236, –571.912231 –561.262939, –1495.169434
47	–1599.613403, –1549.241089, –1734.210693, –1790.577881, –2356.241211 –1012.164307, 8.991084

Table B.35 (continued)

Index	Codeword
48	–1529.388672, –1403.281006, –1647.208618, –1530.277222, –2019.098633 –204.429886, –1181.068848
49	–739.415649, –794.005127, –1411.415649, –1707.639160, –2181.610107

Index	Codeword
	–2057.284912, –2670.123291
50	–670.362915, 142.910156, –360.664429, –494.088745, –1075.005737 –949.166016, –1895.671509
51	–1492.212524, –1310.221436, –1528.393799, –1207.413574, –1402.620728 –877.879089, –1732.826782
52	–1899.612915, –1946.405640, –2189.999756, –2077.627197, –2337.402832 –1881.988037, –2417.630859
53	–443.188019, –565.652710, –618.054932, –1095.713745, –1577.799927 –258.012268, 195.130524
54	–1586.906616, –919.452026, –1106.189697, –755.043091, –1071.426147 –833.499817, –1684.316284
55	–1143.235718, –936.594666, –1197.897095, –1033.911987, –1462.376709 –1257.573853, –2066.716797
56	–536.623352, –206.564285, –343.659088, –62.936752, –504.874664 –529.995789, –1492.665894
57	–635.089966, –626.647522, –1009.990417, –1018.424805, –1485.962158 –1090.072388, –1937.496460
58	–620.463623, –633.232483, –1289.380615, –1406.137085, –2240.552490 –2669.604736, –3655.253418
59	385.163025, 235.010010, 109.594536, –140.336746, –343.256104 –936.745605, –1449.994995
60	–1792.394043, –1997.911011, –2452.838379, –2765.314941, –3405.402344 –3936.408936, –4873.289551
61	–1154.483887, –1268.832031, –1734.370239, –1847.437744, –2347.112793 –2373.519043, –3009.969238
62	–627.695007, –288.654572, –112.204475, 34.272385, 176.012314 283.529846, 305.059326
63	–1217.558350, –1120.316528, –1583.846558, –784.825684, –1247.993896 –1176.566162, –2005.552856
64	–910.307007, –771.879333, –654.502563, –518.989929, –393.384796 –309.178009, –177.891571
65	–2529.651855, –2346.246338, –2178.039307, –1941.845825, –2070.227295 –800.921936, –31.894379
66	–3132.030029, –3499.226318, –2046.768066, –1019.603027, –549.786011 –104.957031, 167.858231

Table B.35 (continued)

Index	Codeword
67	–474.505066, –509.255920, –905.583496, –815.027100, –1272.156982 –819.950623, –1736.453125
68	–172.418259, –211.089874, –379.782471, –747.982300, –954.582886 –1373.502441, –363.987579
69	–2398.576416, –2511.631104, –2776.724854, –2771.916748, –3270.471924 –3033.739258, –508.376343
70	–1303.360962, –1687.757080, –1746.616455, –2382.338379, –2609.013184 –3633.147461, –1998.025391
71	–952.940369, –987.538635, –1416.463867, –1309.199341, –1750.525146 –1389.113770, –2180.002197
72	–574.681458, –562.496338, –678.858276, –728.855408, –818.664185 –756.459534, –1134.977417
73	–412.590485, 448.910034, –602.397278, –842.126648, –489.682098 –746.656067, –1191.718628
74	152.910690, 247.706787, 214.242065, 260.911499, 232.637787 236.632812, 182.894653
75	–1539.218384, –1439.939575, –1750.878662, –1712.403442, –2418.214600 –2109.415771, –92.771507
76	–269.296265, –194.720230, –660.341675, –704.296448, –1362.437866 –1454.097656, –2336.554688
77	–717.084900, –772.887573, –1189.133179, –1135.438721, –1627.112549 –1386.582886, –2218.122070
78	–105.389801, –78.868919, –62.281616, –27.662807, –34.128387 –25.283138, 105.283974
79	–103.275963, –120.688072, –213.694702, –342.597443, –543.487488 –995.674866, –1635.099976
80	1009.061951, 609.200684, –465.664246, 30.097935, –210.724548 –536.370544, –807.965881
81	–1385.402832, –1392.083252, –1377.516113, –832.042542, –1012.094482 –767.382751, –1628.946655
82	–1500.500488, –846.375916, –895.946228, –1546.125366, –1017.199158 –895.583435, –1452.323242

Index	Codeword
83	1122.770508, 1670.907715, 1631.485596, 1352.043579, 1042.138550 780.370483, 613.057800
84	531.805969, 18.100105, –1.253166, 522.595703, 462.182434 442.523926, 430.177887
85	–670.851379, –733.118835, –1088.392700, –611.241272, –1116.670410 –934.079407, –1845.358521

Table B.35 (continued)

Index	Codeword
86	82.398796, 412.937378, –409.110931, –275.867157, –172.803696 –533.976257, –666.394287
87	543.545166, 607.136230, 301.981262, 172.075729, 113.461922 –367.959656, –589.366028
88	–46.647766, –158.918213, –854.853210, –1275.315552, –1905.142456 –1610.375366, –2407.119385
89	521.322937, 241.396042, 453.540802, 341.791687, 266.615082 –259.359222, –238.219513
90	–3024.762939, –3473.895752, –3734.945068, –4403.958984, –2253.765869 –701.315063, –197.920502
91	–2045.555664, –2459.894775, –2769.943604, –3419.300781, –3497.661865 –956.672546, –174.634003
92	108.707832, 252.321487, –208.273285, –93.021729, –641.800171 –682.776855, –1663.273682
93	–1423.165039, –1088.585571, –1452.920044, –1146.858765, –1597.936279 –1454.332642, –2243.208496
94	–2298.474854, –1807.991699, –1828.818726, –1534.023926, –1638.536865 –1291.564331, –1974.960815
95	96.640167, 112.615158, 138.253311, 75.827415, 79.389519 –317.291626, –423.935577
96	–2190.903564, –2466.402344, –2937.284424, –3436.668213, –1297.438965 –498.876495, –27.772251
97	–243.354324, –515.297729, –1009.140625, –799.371338, –1319.890137 –1174.466919, –2037.242920
98	–1041.807251, –1027.113037, –1163.923096, –1349.724854, –1977.282715

Index	Codeword
	–652.251343, 115.361565
99	–1354.297119, –562.692932, –753.652527, –413.559052, –754.919495 –635.021240, –1544.279419
100	–1668.864868, –1325.623413, –1609.677124, –1448.219360, –1860.879028 –1637.513184, –2361.399902
101	–347.324158, –75.992462, –131.846283, 291.220245, 162.089828 388.920471, –653.307129
102	–2050.590332, –2310.230957, –2291.951904, –2675.141113, –2778.385986 –3633.289307, –1146.127441
103	–1614.307007, –663.231628, –1097.481689, –1164.423218, –1516.326782 –1110.013550, –1895.555176
104	–319.394135, –225.083298, –154.588684, –149.742538, –115.347519 –371.147858, –522.308655

Table B.35 (continued)

Index	Codeword
105	–859.840576, –938.111267, –1020.927307, –1169.248535, –1277.663086 –1379.227295, –1146.013184
106	–1326.813721, –1601.993652, –1752.636230, –2242.349609, –2609.572266 –3539.369629, –913.157288
107	–2168.114014, –2284.954102, –2477.869629, –2073.338623, –2052.058105 –1417.045776, –2028.703491
108	–1867.457886, –1809.243896, –1990.184448, –1686.044189, –1962.137573 –1634.547119, –2286.456299
109	–1081.580444, –196.722565, –694.187744, –1347.482910, –605.154907 –826.458252, –1013.666016
110	–919.857666, –1200.682861, –1102.610352, –927.558167, –1131.861206 –1148.240112, –1908.182495
111	–440.644104, 597.595337, 1034.048462, 968.113770, 761.715088 594.234375, 512.320557
112	391.910309, 563.187866, 992.584900, 989.100708, 791.781616 635.883789, 524.414001
113	–1439.674683, –432.677399, 805.322449, 1533.820435, 1271.657715 1052.136719, 766.054626

Index	Codeword
114	743.810669, 1176.505127, 1493.295532, 1480.138062, 1338.440552 1121.954712, 834.426086
115	–1309.833740, –1543.746216, –2007.929199, –2380.663330, –2985.001221 –3658.058105, –4591.415039
116	–1922.901245, –1399.050415, –1030.848633, –619.000916, –646.067261 –641.641907, –1374.606079
117	–1196.555542, –973.870911, –1342.077759, –1283.749268, –1889.687012 –2061.299561, –2908.992432
118	–980.406311, –728.974548, –885.837646, –752.803345, –1042.341309 –745.296265, –1563.498047
119	–1263.167725, –1426.673706, –1675.168213, –1466.998169, –1760.544067 –1011.527344, –1834.742432
120	–751.732178, –818.037720, –1138.700195, –342.462280, –658.276855 –1265.891968, –2143.573486
121	–1062.747192, –1031.713623, –1468.429077, –1521.497925, –1117.125610 –1381.837769, –2174.170410
122	–727.855774, –761.253967, –860.535095, –1033.114380, –965.472717 –817.884644, –371.561340
123	2878.902344, 2563.296875, 2195.727051, 1742.656860, 1348.161011 980.504700, 733.287659

Table B.35 (continued)

Index	Codeword
124	541.978516, 672.371216, 416.751801, 434.956940, –161.895126 –387.416565, –1383.375610
125	273.264313, 421.542664, 200.308136, 299.106689, –225.245010 –408.481689, –1372.519531
126	1209.753784, 1304.036499, 1377.753174, 1216.187012, 969.355408 773.580505, 594.077393
127	21.863016, 1291.028809, 1398.661255, 1895.399658, 2431.829102 2305.669922, 1364.550049
128	1303.329468, 1405.020874, 1302.691284, 1044.241333, 482.370239 –99.826935, –742.828918
129	–243.456650, 19.171192, –169.918671, 41.986885, –427.325165

Index	Codeword
	–493.024017, –1455.053223
130	–1885.273560, –1884.482178, –1712.416626, –1577.402344, –1217.507446 –1040.416504, –771.315369
131	–353.943054, –561.155701, –838.340637, –301.564575, –768.600586 –660.318481, –1599.372559
132	–596.970764, –412.483307, –497.547699, –143.599548, –386.428436 59.043865, –814.205750
133	–224.299942, –247.268860, 146.163208, 623.048340, 533.660522 471.134277, 421.878693
134	–1553.117188, –1283.806396, –911.148193, –677.186279, –274.994110 51.233665, 148.706848
135	–2115.568604, –2261.006104, –2324.882812, –1577.229004, –1470.652100 –1025.168945, –1738.755493
136	–167.383331, –12.103893, 73.544403, 159.821869, 283.711426 364.061676, 391.078552
137	–1912.710205, –1896.303101, –1644.200806, –1082.829590, –1104.169800 –869.738037, –1651.942749
138	–1340.277466, –1241.133667, –1657.185791, –1511.284302, –2178.502197 1093.690063, 45.818890
139	–1570.591797, –1679.194092, –1796.852661, –2047.197021, –2285.650879 –2350.982910, –1454.589722
140	–1231.343994, –824.596130, –996.815735, –746.322632, –1122.684814 –1111.156372, –1945.990845
141	–591.877258, 516.155334, –489.766815, –64.764053, 152.845764 333.220337, 62.063614
142	15.653727, 224.633698, 17.151478, 178.893738, –337.240814 –462.331940, –1436.735352

Table B.35 (continued)

Index	Codeword
143	–880.164734, –1290.051636, –1623.288086, –1078.399658, –1245.619141 –837.299866, –1683.115723
144	–381.960052, –284.937103, –677.927490, –490.750854, –1093.306274 –1053.094482, –2001.423950

Index	Codeword
145	539.515076, 387.047943, 101.081024, –7.919919, –134.106995 –150.895966, 135.475677
146	1593.414062, 1659.969238, 1600.840820, 1375.510620, 1092.070435 869.114746, 646.936401
147	–787.252808, –1061.453613, –1590.851074, –1673.327759, –2047.322632 –1546.440063, –2271.757812
148	–876.749084, –45.474106, 644.229065, 755.632751, 620.382935 499.973053, 440.885681
149	–1919.868652, –2080.363770, –1926.503418, –2014.912842, –1927.411865 –1671.540894, –1158.850708
150	–1218.003174, –1076.153198, –932.393799, –782.836609, –592.775208 –526.451233, –449.310181
151	–2234.837158, –2223.677734, –2711.080322, –1939.675659, –578.922241 –171.547806, 191.678024
152	–515.611633, 583.618469, 401.767639, 306.167664, 320.778778 322.591461, 361.566620
153	–184.567017, –207.550415, –637.447937, –573.346252, –1081.125610 –787.249268, –1735.852539
154	227.247940, 444.299164, 297.879089, 608.798340, 334.543396 427.894012, –635.143921
155	–1812.030884, –1838.233154, –2065.830566, –1949.772827, –2039.627563 –1210.980957, –1901.968018
156	1436.209717, 735.195618, –322.993958, 678.544128, 432.665771 592.310730, 166.796082
157	–2169.908936, –2414.579346, –2538.250488, –2509.250000, –2578.761475 –2044.640625, –2299.136719
158	–1557.501587, –1493.935425, –1315.934204, –1132.046143, –782.128296 –679.114502, –560.047363
159	15.841063, 1116.713013, 1351.467651, 1158.964600, 898.157715 685.430969, 553.459656
160	–1616.942993, –617.771545, –350.499329, –60.721684, 76.723930 237.614853, 156.796127
161	–876.453308, 166.194077, –586.573547, –950.025879, –162.685486 –11.021299, –74.673676

Table B.35 (continued)

Index	Codeword
162	–933.958252, –792.638428, –681.551208, –596.130432, –585.906128 –594.729431, –936.187195
163	–1208.862915, –1227.308350, –1181.368286, –1337.841309, –948.802124 –172.173843, 113.793404
164	47.086929, 9.638669, –28.048338, –108.225067, –177.868164 –537.585083, –921.853699
165	2227.242676, 2299.043213, 1983.756958, 1611.156250, 1230.987183 910.737427, 687.558899
166	–2634.895264, –1341.407104, –655.939148, –435.204132, –138.849350 109.223824, 175.581787
167	381.176819, 172.363068, –181.324799, –255.549347, 88.474831 314.105194, 349.051849
168	–942.328003, –727.934448, –1051.050659, –895.382385, –1331.618530 –949.222717, –1827.845581
169	1050.846069, 960.810059, 899.902222, 918.711304, 741.007629 642.515076, 427.510681
170	–1923.227295, –2332.062744, –2334.375244, –2903.227539, –2950.154785 –3899.762695, –2456.197754
171	–1321.074463, –1645.838745, –1858.400757, –2214.028320, –1537.176392 –342.966125, 61.078449
172	794.880249, 930.161804, 695.314148, 628.398499, 40.544762 –332.564636, –1304.538208
173	–1196.832886, –1046.309692, –974.942627, –507.224152, –811.209106 –686.430359, –1591.003418
174	–1766.451294, –1068.532959, –1442.078491, –1631.367676, –1790.646729 –1115.237061, –1888.490234
175	–2597.554443, –3010.818359, –3857.105469, –4409.147461, –5543.684570 –5077.316406, –969.496399
176	686.251831, 616.379700, 427.633118, 373.425873, 278.641876 251.032150, 277.012299
177	–938.053894, –423.530273, –804.657959, –1640.446411, –1215.777222 –1003.299316, –1671.776978
178	222.797958, 344.567383, 353.334686, 417.572205, 452.120239 471.539337, 452.700836

Index	Codeword
179	–193.654877, 486.405945, 1418.244995, 1514.918335, 1254.049561 1046.550415, 773.944763
180	–1542.704712, –1538.845825, –1861.758179, –1613.740601, –1869.317993 –1350.750854, –2065.242920

Table B.35 (continued)

Index	Codeword
181	–1929.660034, –1931.505859, –2094.259277, –2162.314209, –2643.336426 –3044.777344, –388.269257
182	2023.152832, 2168.609131, 1901.014771, 1566.093262, 1199.623291 905.672363, 677.364563
183	–1454.306519, –1817.152954, –2259.534668, –1921.817383, –2116.793457 –1604.087891, –2239.123047
184	–2059.189697, –1293.733521, –1435.723511, –1025.210571, –1308.423828 –1063.895020, –1869.822754
185	1757.312622, 2066.028564, 1840.750732, 1509.043823, 1156.838501 859.446899, 657.098877
186	–1604.242920, –1782.429321, –2060.613525, –989.568481, –304.592560 52.363155, 261.436890
187	–790.060547, –677.193604, –854.483887, –1002.869568, –1361.044922 187.528214, –758.687378
188	–850.720215, –815.171509, –782.120972, –750.712524, –930.394592 –1100.443970, –1842.617432
189	–1013.528564, –619.302734, –1039.494873, –1345.197998, –1822.371460 –1228.573486, –1966.074951
190	–890.626709, –1002.113586, –1763.994995, 44.655560, –635.783936 –1296.306763, –1941.137939
191	–2550.192383, –3048.042969, –3690.282471, –4373.286133, –5208.394531 –1917.806274, –581.258240
192	–46.051090, –444.285278, –621.605347, 218.354416, 294.350952 309.784790, 355.443756
193	–832.117493, –659.459045, –481.507568, –341.173279, –48.068172 158.933624, 259.165100
194	–723.183533, –566.399109, –422.477692, –387.357666, –312.310120

Index	Codeword
	–495.552063, –642.440918
195	–3213.293945, –3731.129150, –3566.625977, –1970.124390, –969.271667 –349.031250, 88.375351
196	–1493.730713, –1925.979980, –2232.583740, –1535.923706, –1746.921387 –1267.045776, –1990.320190
197	–931.412964, –793.034607, –1133.670776, –1261.493530, –1737.755371 –699.421814, –1597.601440
198	–802.901489, –780.312927, –1156.061279, –814.251770, –1303.202515 –1265.482178, –2118.627930
199	1859.657349, 1899.714355, 1752.933350, 1489.179077, 1168.854858 923.944519, 675.960144

Table B.35 (continued)

Index	Codeword
200	–758.564758, –1219.479126, –1183.540894, –239.990875, –2.090081 238.982742, 254.410858
201	–654.688660, –1015.827209, –1190.445923, –605.556641, –856.632019 –642.030579, –1519.150391
202	1119.471191, 858.251709, 249.507751, 471.759491, 265.540253 –206.793472, –294.015991
203	–1211.920776, –1029.572876, –1262.090942, –1054.197144, –1343.196411 –919.689575, –1749.811035
204	–610.228638, –286.711029, –544.461487, –354.503784, –793.421814 –698.029663, –1636.272949
205	–1988.942139, –2270.791504, –2623.452881, –3106.659668, –3696.352295 –4359.229980, –1231.581909
206	–769.379578, –1117.632202, –1622.923950, –1560.120117, –1645.681274 –1032.027710, –1828.624878
207	–214.166290, 389.766052, 490.709106, 845.222534, 281.663879 –253.058563, –1178.845459
208	–1189.373657, –1266.664551, –1796.728516, –1872.993408, –2566.956055 –2864.288330, –3873.072998
209	–752.549072, –213.618332, –171.188766, 526.129272, 59.229225 –384.757385, –1381.908691

Index	Codeword
210	589.618896, 958.856995, 1043.165405, 1143.152832, 1853.732544 2285.578369, 1410.694824
211	–1040.660400, –1191.703125, –654.014648, –941.981873, –1474.148315 –921.510010, –1640.542358
212	–1563.276001, –1639.329346, –2228.906738, –2306.195312, –2921.983398 –2888.887939, –3908.613770
213	–1378.559082, –1430.042114, –1700.228394, –53.977154, –540.871887 –542.380432, –1553.821655
214	–1160.307495, –1163.990601, –1185.203003, –1398.706299, –1303.736572 –987.148376, –471.440216
215	–1030.117188, –972.643860, –1276.948486, –738.906311, –1100.516724 –795.453491, –1693.452637
216	–468.792480, –450.322815, –974.565002, –1008.784973, –1696.910156 –1940.522095, –2817.710938
217	–838.831787, 274.059998, –663.343933, –1405.616699, –877.030273 –879.753784, –1595.256836
218	137.608704, 69.000343, –47.363461, –233.584244, –445.194702 –970.103699, –1565.977539

Table B.35 (continued)

Index	Codeword
219	–1042.776245, –477.652252, –729.477844, –454.572510, –946.314819 –931.773499, –1823.560181
220	42.125374, 35.638535, 740.415527, 843.886780, 668.383301 543.146301, 472.985168
221	392.935059, 653.371216, –448.405304, 375.040161, 295.584198 471.666870, 103.572754
222	–154.301132, –306.378052, –466.394440, –551.980957, –192.136795 165.342453, 298.238892
223	–2925.972656, –2523.248535, –1111.652954, –685.163635, –275.557953 25.750113, 194.467148
224	–1574.326416, –1417.313110, –1626.486328, –1278.805542, –1589.582886 –1240.964600, –2012.671753
225	594.715515, 1486.845581, 1531.019653, 1274.003052, 970.901367

Index	Codeword
	732.011169, 579.845337
226	137.167023, 36.624592, –517.013245, –589.374573, –1265.440552 –1145.184204, –2051.384277
227	–26.435886, –56.422054, –404.378052, –394.194000, –813.856628 –314.063995, –1064.395996
228	–1260.744629, –1497.010498, –2024.455566, –2299.268555, –2770.762207 –2045.283081, –2527.341797
229	–359.603333, –176.854752, –683.117859, –981.453491, –1492.026001 –1031.153564, –1905.748535
230	–1651.876099, –1859.713501, –2263.955811, –2451.694580, –2612.706787 –1484.526978, –2041.182739
231	–597.484070, –370.993713, –597.899780, 472.524628, –203.220230 –1188.975708, –2064.148193
232	–626.137695, –534.148499, –583.699036, –581.131592, –782.097595 –1054.995117, –1791.960815
233	–1330.559326, –1746.493286, –1845.468750, –2474.347168, –2675.653564 –3675.258301, –3224.888184
234	–350.680878, –649.529663, –1278.257690, –1429.337891, –1821.152344 –1156.966309, –1977.541138
235	–1108.990601, –980.478210, –1450.680420, –1764.365967, –2155.622559 –1117.615234, –1904.928223
236	–731.516907, –960.150146, –368.916779, 237.534698, 298.949432 354.164429, 369.252014
237	726.680481, 970.058655, 1240.952026, 1122.956055, 901.566833 712.591248, 579.026184

Table B.35 (continued)

Index	Codeword
238	–1509.388062, –1746.989380, –1382.398804, –1719.351562, –2064.650879 –1278.447510, –1975.246460
239	414.816895, 535.429321, –51.128113, –234.962631, –1016.078186 –1202.653809, –2175.594971
240	–759.597778, –960.860474, –1123.675293, –1550.068359, –1919.764771 –2633.365967, –1350.687866

Index	Codeword
241	–1151.824707, –1196.646729, –1087.567993, –1698.596680, –1496.082275 –1052.537354, –1673.215332
242	–1385.504883, –1422.370239, –1487.047119, –1680.568115, –1802.940796 –1530.545532, –793.454041
243	1013.024414, 923.160645, 798.228455, 699.183228, 541.279236 341.881500, 412.349915
244	–2169.254883, –1868.476440, –1482.921021, –1427.902588, –1178.953979 –290.343079, 48.260975
245	–2174.610840, –2151.178711, –2372.573730, –2247.478760, –2823.552979 –1877.511353, –206.024292
246	1082.607422, 2064.338379, 2365.753174, 2626.614014, 2299.045410 2062.203125, 1200.084229
247	144.305130, –24.135992, –270.347900, –433.131073, –458.793701 –345.861237, 27.411316
248	–1911.949707, –2278.694824, –2423.027588, –2859.584961, –3014.995117 –3781.610840, –3822.277100
249	–836.404602, –933.741699, –1307.806152, –1098.735352, –1511.847290 –1059.338257, –1894.064087
250	–640.583130, –440.947906, –751.173767, –560.251831, –1045.903076 –798.479858, –1751.981079
251	–2256.432617, –1674.571045, –700.385254, 756.211304, 1299.046753 1050.701660, 726.589233
252	–722.514709, –774.690674, –832.236450, –947.848145, –522.118652 7.550344, 214.319077
253	–1925.266357, 238.258560, –597.768127, –720.570435, 15.188728 186.199814, 8.823078
254	–1493.605347, –1635.037720, –1872.133667, –1141.085938, –1540.600220 –1550.948486, –2279.032471
255	–250.391434, –80.652458, –379.481018, –217.784698, –719.979736 –681.776978, –1644.389404
256	–1256.465210, –1254.223389, –1621.097778, –1516.847778, –1851.374390 –1392.915283, –2135.959473

Table B.36 Stage 2 first sub-vector codebook lsf_stage2_CB1_hbr of high-precision LSF vector quantization codebook

Index	Codeword
1	–713.785767, 168.407120, 153.748795
2	36.340725, 90.616714, –32.429771
3	52.624313, –38.090992, –364.945282
4	154.185974, 109.762367, 355.092072
5	–375.530365, –62.235508, –134.981415
6	115.415070, –699.178345, 276.090454
7	–363.505493, –245.460114, –404.869843
8	–102.644508, 121.746376, –52.143639
9	250.990021, –180.223267, –60.538208
10	2.304316, 254.452957, –118.709351
11	–64.000519, –172.352188, –549.504944
12	145.008041, –198.819794, 74.863930
13	–319.988373, 62.626167, –325.089386
14	–288.144287, –715.401611, –396.285156
15	–211.637512, –4.421244, –14.446482
16	82.578644, 123.950401, –213.957626
17	–164.570007, 844.651917, 255.872421
18	–247.887390, 55.306465, 200.785889
19	–215.868759, –609.106079, –63.238937
20	392.302124, 42.328953, 249.753616
21	–184.973984, –69.296509, 118.628799
22	51.923347, –142.208099, 177.060547
23	53.448772, –305.917572, –103.890076
24	229.675934, 0.439482, 122.373756
25	–277.801453, 166.968170, 733.307129
26	170.546982, 339.845001, 31.960520
27	64.136490, –25.794970, 65.980400
28	–43.597557, –34.969471, –37.739056
29	–212.885864, 507.455505, 437.958435

30	–493.030060, –311.711365, 187.651596
31	–354.960724, 144.233612, 368.872223
32	81.155983, 198.582993, –361.475037
33	–57.561897, –190.277390, –40.531013
34	225.483200, 184.535919, 145.773300
35	37.873772, –300.896942, 273.377380
36	–161.762527, –104.531548, –353.909149
37	437.577271, –248.396957, –118.537544
38	95.924393, –536.217834, –123.140144
39	–116.965042, 137.023346, –432.474640

Table B.36 (continued)

Index	Codeword
40	–401.308441, –316.035950, –124.605537
41	265.316132, 7.975056, –193.403076
42	–10.286757, –256.780396, –308.450928
43	286.242706, 221.018967, –186.312241
44	231.734329, –155.717773, 278.019806
45	–344.505493, –185.656815, 515.490845
46	–210.659317, –371.243408, 298.229218
47	–55.369648, –502.107391, 166.864746
48	–237.739120, 449.408295, 71.104065
49	–197.362518, 292.102844, 232.818176
50	–244.392792, 154.775345, 58.205173
51	266.715729, 694.912415, 70.328987
52	–204.886414, –186.953674, –33.269745
53	119.856064, –148.999146, –212.473984
54	–96.027031, –133.978561, 283.992157
55	–277.930481, 14.714230, –619.926331
56	108.560051, 381.440857, –174.843597
57	–532.884521, 158.514374, –200.185989

Index	Codeword
58	–14.112489, 326.978088, 130.464569
59	–101.656219, 31.647610, –269.940765
60	–144.576035, –422.618805, –323.873138
61	65.474930, –420.618225, –505.687775
62	209.829102, –387.511017, 94.998833
63	–172.908279, –60.086086, –150.022415
64	–46.326584, 260.807922, 452.773834
65	279.548889, –141.048431, –322.010864
66	393.468262, 8.009486, –42.706268
67	–81.231201, 45.355511, 65.719444
68	222.150711, –141.217514, –564.965637
69	346.164856, 434.299622, –66.598831
70	568.707031, –159.705505, 544.176575
71	–211.341019, –216.253036, –211.727448
72	224.405930, 357.485809, 257.797882
73	553.497620, 31.112984, –678.450806
74	82.964523, 135.422699, –656.731628
75	175.658020, –128.771088, 503.778290
76	14.064637, 492.906281, 239.662155
77	472.926147, –403.081818, –391.692749
78	–292.928436, –699.444397, 353.090057

Table B.36 (continued)

Index	Codeword
79	72.990135, –133.231140, –53.157795
80	–52.832638, –117.261017, 69.052734
81	426.745056, 201.403336, 31.993965
82	223.471848, 449.586548, –409.097839
83	–74.437065, 170.063431, 109.249542
84	94.061584, –8.890752, –155.487183

Index	Codeword
85	181.220139, –12.836661, –26.633982
86	–257.568481, 404.635712, –478.268524
87	–118.702698, –1.801430, 436.133667
88	–328.992798, –148.799011, 49.605450
89	685.787781, –5.360787, 84.299042
90	–92.548859, –405.149048, –942.782654
91	545.499023, 378.187408, 237.816772
92	223.737640, 689.972900, 549.611572
93	–402.027008, 256.815186, 47.899891
94	198.374649, –338.206146, –228.664978
95	–74.177917, –279.765411, 485.423126
96	592.158936, 327.568512, –262.079163
97	–97.092148, 110.176552, 270.185944
98	–134.869324, 293.152802, –12.102485
99	196.897675, 149.523315, –40.585079
100	974.516663, 442.782013, 130.876434
101	–119.678520, 183.199692, –210.154816
102	520.430542, –462.481323, 137.886856
103	64.144768, –24.995043, 303.479828
104	–65.387932, 380.154449, –306.889679
105	312.868439, 105.024971, –392.744720
106	–42.009869, –144.065552, –184.797028
107	–286.622528, –143.909866, 256.411652
108	361.120209, 194.673004, 488.534027
109	608.696533, –6.981796, –274.527039
110	45.004539, 218.194778, 28.713619
111	–256.952240, –384.646057, 64.113510
112	–488.355804, 532.116638, 224.472809
113	–128.912506, –241.478180, 128.511200
114	–272.429382, 330.666290, –179.248230
115	33.182674, 226.608612, 254.291412

Index	Codeword
116	–439.983124, 8.855278, 102.383743
117	53.941254, 37.994209, 697.273926

Table B.36 (continued)

Index	Codeword
118	–128.799225, –350.602203, –105.367256
119	–11.603866, –330.001129, 57.193336
120	–109.075554, –486.121399, 795.247375
121	–183.717377, 632.973999, –138.407562
122	–6.666337, 486.127014, –17.339613
123	389.025299, –181.146561, 137.332169
124	–271.502441, 115.163071, –122.910507
125	–39.242790, 21.079741, –150.635345
126	257.078369, –403.075104, 437.620300
127	–45.516216, –8.866961, 180.737320
128	70.438759, 98.982910, 139.766769

Table B.37 Stage 2 second sub-vector codebook lsf_stage2_CB2_hbr of high-precision LSF vector quantization codebook

Index	Codeword
1	20.507162, 117.254471, 308.534973
2	167.627548, –284.581360, 35.337910
3	–50.533733, 262.051270, –222.744385
4	213.973526, –192.300507, 227.454147
5	324.752869, 154.691422, 142.521759
6	–302.892273, 22.365881, –268.279053
7	384.219574, –46.759758, 295.384735
8	–326.932983, –142.875214, 123.286591
9	462.440582, 30.680933, –644.826782
10	75.870522, 100.746597, –229.091095

Index	Codeword
11	–184.132187, –518.547424, –105.319809
12	–128.916153, –172.195633, 83.659973
13	63.237682, –497.342957, 72.141266
14	274.313446, 167.584351, –33.896770
15	–596.630493, 8.026925, –321.141144
16	50.223522, –16.749166, –83.381676
17	44.889523, –767.325989, –139.284637
18	224.780106, 352.491333, 75.805626
19	–33.128654, 305.753296, –601.859070
20	115.771713, –93.239197, 107.058525
21	–148.837738, 183.316940, 413.980072
22	–182.227646, –697.383789, 278.016632
23	445.581207, 256.918182, –61.204651
24	707.401611, 599.105469, 466.282776

Table B.37 (continued)

Index	Codeword
25	172.835205, –255.287506, –399.850555
26	–182.281845, 185.917496, 198.393707
27	–354.348358, 77.655319, 121.351845
28	165.057236, –250.012955, –153.211365
29	15.198963, –101.545570, 337.992859
30	635.096924, –22.253139, –276.124939
31	–174.264359, 306.408936, –30.304029
32	63.197758, 65.521347, 148.829025
33	–565.097168, –720.727173, 95.707863
34	173.535858, 4.769509, 226.110519
35	–302.163788, –364.092468, 40.338612
36	322.009796, –130.186035, –257.399811
37	–143.584061, 432.809937, 187.272064

Index	Codeword
38	–8.131482, 172.260559, 72.100441
39	–69.128052, –15.661174, –246.318619
40	141.381821, 678.433350, 463.580078
41	205.150635, 60.560955, –127.768204
42	–396.496399, 355.075836, 271.316498
43	–16.061943, 261.906860, 182.079605
44	119.859367, –86.526634, –181.600311
45	–278.325073, 273.584839, –346.330780
46	–848.170715, –77.039818, 206.621765
47	–20.108572, –224.001541, –248.247284
48	–152.951675, 146.029510, 4.365578
49	62.817871, 353.723999, 343.701416
50	–436.317230, –275.409271, 341.552063
51	147.565918, 230.680481, 166.706573
52	–34.370865, –308.244354, –41.257004
53	–241.203552, –430.650146, 813.072693
54	368.040619, –410.771362, 232.966476
55	51.651077, –146.025299, –22.975538
56	–282.722992, –620.837402, –850.400146
57	–165.943176, –55.919006, 572.439575
58	–103.814415, –227.512253, –490.515442
59	124.234863, –193.956253, –844.806152
60	–505.521362, 82.886086, 466.838531
61	148.491516, –239.119812, 546.069153
62	63.785629, –80.139122, –321.356995
63	564.061768, 129.677856, 165.186554

Table B.37 (continued)

Index	Codeword
64	158.311157, 128.742462, 70.707832

Index	Codeword
65	316.238434, –27.633970, 86.694901
66	–175.450073, –93.772957, –166.443024
67	–142.252975, –157.993195, 255.044876
68	–313.525330, 13.380783, –652.654724
69	–241.896759, 28.034353, –57.149029
70	–190.798920, 0.189457, 116.813072
71	844.387390, 337.000793, –19.824232
72	–564.077698, 500.165619, –415.747192
73	213.419327, –91.672638, –57.745274
74	–99.660248, 74.968384, –407.527557
75	–29.572922, –96.862167, –131.641983
76	–135.936050, –355.414703, 151.864426
77	222.614548, –673.049561, 517.544067
78	111.716286, 204.121689, –51.537071
79	80.887352, 35.771717, 2.501737
80	–370.751129, –221.242981, –170.055420
81	–549.659729, –200.956406, 12.992573
82	59.403713, –418.178558, –191.717911
83	–52.472393, 56.039600, 70.462021
84	–88.299973, –178.931824, –61.182095
85	125.889633, 580.681763, –326.664459
86	–592.087524, 160.197723, 37.150238
87	362.081543, 337.822754, 311.010651
88	186.879440, 331.649445, –169.341782
89	–238.267609, 783.910889, 138.339188
90	–5.536943, –490.727203, –455.820465
91	349.078400, –449.050262, –113.857185
92	524.526978, –149.489349, 603.087891
93	470.803162, 318.161041, –352.185669
94	–166.528366, –349.740906, 415.440033
95	159.741531, 26.906466, 458.308716

Index	Codeword
96	123.212921, –21.127138, –508.872711
97	–239.674759, –150.437744, –346.729095
98	300.417145, 81.030838, –299.241089
99	–400.569550, –31.321560, –64.911179
100	–329.600861, 264.643341, 72.342133
101	–76.098831, 23.414692, 223.034027
102	410.300262, 545.022339, 14.200037

Table B.37 (continued)

Index	Codeword
103	64.472435, –380.021576, 299.929443
104	–276.012024, –14.363681, 307.793549
105	458.341339, –364.150879, –473.549255
106	–134.677811, –57.834167, –0.576738
107	–138.862488, 111.922058, –166.296036
108	–390.244507, 475.571930, –62.043774
109	146.384476, 230.729202, –393.689453
110	216.011246, 162.604935, 315.091980
111	371.276337, –198.172241, –7.094508
112	–412.132812, –448.155182, –344.452454
113	21.202076, –217.792053, 148.690628
114	–239.708435, –178.662338, –46.387856
115	–372.993713, 190.326294, –134.271698
116	78.791969, 546.357361, 89.262733
117	108.147072, 201.503693, 691.316589
118	–295.153564, 468.795502, 587.713989
119	11.740393, 334.976410, –1.581412
120	–170.412354, –303.384949, –200.288986
121	3.908800, 121.635567, –68.972488
122	183.778122, 8.971548, 47.254635

Index	Codeword
123	–62.340725, 12.960299, –53.699871
124	422.356262, 21.269390, –79.914871
125	–10.798393, –58.968788, 82.892532
126	–110.142456, 493.066437, –173.040512
127	–762.578003, 593.424316, 203.118790
128	648.862854, –213.159912, 89.717255

Table B.38 Stage 2 third sub-vector codebook lsf_stage2_CB3_hbr of high-precision LSF vector quantization codebook

Index	Codeword
1	61.442013, 88.865410, –6.914186
2	–189.400803, 183.880447, 453.544067
3	119.694290, 122.304260, –185.256332
4	185.325165, 257.642426, 21.324890
5	176.223907, 376.925690, 276.048035
6	44.387566, –336.891266, –287.661438
7	364.234314, –399.658417, 453.150360
8	309.771759, –396.966858, –246.510818
9	–98.734177, –268.355042, 117.649399

Table B.38 (continued)

Index	Codeword
10	68.921753, 181.501190, 315.960663
11	–16.736830, –112.308594, 57.227196
12	–191.615204, –370.536560, –89.005638
13	–635.830200, –111.513008, 426.530273
14	133.312378, –378.453064, 44.851124
15	–247.431152, 145.292679, 33.881779
16	702.903503, 405.354767, –185.583740
17	55.931442, –52.922546, 225.309769

Index	Codeword
18	–329.518463, 23.220282, 207.375168
19	–81.797279, 61.733788, 214.160568
20	–333.189178, –366.966156, 253.709396
21	300.709991, 191.407745, 190.362869
22	–54.259617, 292.960663, –166.641479
23	207.949539, –137.050339, 90.344116
24	292.658539, 413.935455, –165.154205
25	43.520733, –285.075592, 348.921387
26	–173.507782, –407.853882, –702.097595
27	576.308167, 110.584236, –7.069339
28	583.669678, 2.836300, –559.254211
29	331.974945, 23.614412, –271.937256
30	275.383270, –28.569506, 282.474884
31	311.056641, 79.100693, –21.822622
32	–192.355347, –147.341660, –184.034576
33	–439.608124, 448.283508, 603.143738
34	–201.011368, 21.700256, –159.099167
35	135.237228, 2.610675, 35.422089
36	–19.653358, –37.952515, –95.742783
37	–74.458069, 109.526535, –83.227531
38	–335.823456, 744.129639, 38.045818
39	466.931183, –218.274597, 4.676877
40	–13.563426, 208.090698, 87.632927
41	119.996140, 111.237968, 156.117310
42	–259.981445, 466.053467, –437.298828
43	–84.826576, 11.426362, 51.927937
44	–158.319885, 331.992126, 210.548325
45	494.739502, 174.590012, 389.298523
46	–27.986452, –202.765732, –108.582108
47	–81.897018, –700.229736, 150.311615
48	–441.722595, –63.444637, –332.879852

Table B.38 (continued)

Index	Codeword
49	87.043427, 43.144299, 555.890503
50	–174.406799, –471.973938, 753.418884
51	154.363052, –138.182922, –528.208679
52	–610.850464, 358.816376, 115.910812
53	–172.365845, –105.489944, 361.526398
54	–500.688629, –436.756714, –99.204269
55	156.351166, 236.082245, –466.519012
56	–179.466690, –272.355408, –352.192108
57	383.152679, 591.679932, 542.802246
58	164.616272, –115.712784, –131.835144
59	–373.097534, 293.575073, –173.105423
60	–251.338821, –126.143494, 19.707720
61	–147.272232, 61.723743, –422.298248
62	3.450175, 514.963806, 16.579840
63	–531.877258, –14.696597, –27.526365
64	12.894713, –64.106720, –285.631409

Table B.39 Stage 2 fourth sub-vector codebook lsf_stage2_CB4_hbr of high-precision LSF vector quantization codebook

Index	Codeword
1	92.195709, –173.186340, –76.764999
2	–42.733780, 402.076965, –167.067413
3	216.264893, 102.677452, 168.312927
4	–180.997543, 162.147827, –134.432327
5	67.712898, 258.969391, 239.192825
6	–170.611481, 256.868103, 173.602570
7	–8.513791, –111.140099, 92.899986
8	327.391144, 205.558334, –102.128365
9	244.509750, –137.128418, –470.007751

Index	Codeword
10	–305.208832, 85.310158, 56.496731
11	180.332489, –404.626892, 35.443607
12	–89.915100, –413.847076, 440.065216
13	–49.306343, 63.195087, 62.291702
14	–56.507740, –124.494896, –224.872910
15	–117.042595, –206.133224, –17.508348
16	–199.800507, –28.671963, –490.060699
17	57.166000, 181.844971, 14.875186
18	–11.494995, –12.210985, –83.709663
19	83.800140, 98.593323, –225.477493

Table B.39 (continued)

Index	Codeword
20	130.287781, 5.693187, 39.476902
21	128.140930, –135.807297, 296.940247
22	–112.687485, –389.502289, –221.875732
23	–157.652573, –37.295250, 11.366231
24	–496.884033, –539.886902, –120.588898
25	–78.960876, 34.301903, 287.433350
26	269.951935, –62.663803, –131.213440
27	–538.176331, 470.946381, 245.755112
28	–265.331299, –83.310402, –141.434708
29	436.499817, 440.873291, 263.916779
30	–631.545959, –29.409035, –72.509338
31	–294.762054, –158.774612, 187.861343
32	453.967224, –94.971153, 122.421051

Table B.40 Stage 2 fifth sub-vector codebook lsf_stage2_CB5_hbr of high-precision LSF vector quantization codebook

Index	Codeword
1	–446.630402, 640.495911, 268.249786, –67.830261
2	34.200180, 144.016098, 43.309258, 28.019665
3	122.397560, 413.036896, 108.406776, 21.556137
4	–291.235809, –590.999878, –4.219362, –29.149132
5	–182.353165, 247.858917, –82.613281, –248.262726
6	11.704876, –19.809439, 175.695541, 109.505432
7	–199.381882, –162.686661, 116.547264, 96.769501
8	85.434540, –28.285826, 18.597948, –23.235142
9	112.726555, –284.508789, –29.389679, 257.062927
10	–253.303757, –54.471107, –98.781433, –31.290049
11	–28.797440, –6.548668, –64.319229, 103.122360
12	102.159714, –280.140747, 276.293243, –38.849163
13	–184.768158, –482.613373, –769.700989, –47.989521
14	–223.921066, 166.102005, 120.577484, 120.780533
15	58.940697, 148.444000, 49.055420, 347.937683
16	–194.453476, –55.881226, –203.431305, 440.613892
17	271.730316, 37.378754, –396.253754, 157.584000
18	–125.173019, 240.714401, –231.325714, 77.097366
19	245.718994, 72.768524, 112.223206, 59.880192
20	–58.394215, –204.128754, –21.877651, –69.457428
21	198.198151, 152.518585, –76.095955, –92.793144
22	5.645205, –2.743056, –126.321007, –129.334778

Table B.40 (continued)

Index	Codeword
23	140.627350, 39.890167, 114.086823, –290.992889
24	129.491241, 165.633606, 406.660248, 150.717422
25	–606.185791, –12.697894, 7.033066, –27.042786
26	585.682068, 260.057220, 63.060730, –9.845518

Index	Codeword
27	254.377533, –187.517456, –100.691185, –109.688454
28	–81.884468, –167.102570, –333.533386, –11.585288
29	216.725479, 93.283714, –303.676849, –367.260162
30	–95.634460, –174.039597, –142.232117, –356.129242
31	–83.721230, –3.592948, 37.293602, –44.416279
32	–173.826035, –209.999924, 505.608887, 329.540344

Table B.41 Stage 1 first sub-vector codebook lsf_stage1_CB1_lbr of low-precision LSF vector quantization codebook

Index	Codeword
1	–350.459351, –1110.657715 –1039.387939 –307.705811, –81.258949 –67.539551, 27.608225, –15.766386, 26.909994
2	830.518982, 1316.991455, 969.974976, 701.664673, 395.667511 140.679199, –86.249687, –326.514038, –441.620605
3	875.874268, 300.750061, –135.191727, –749.033386, –624.504944 –977.145691, –270.945770, –528.141113, –484.400909
4	–37.185486, –438.739746, –198.407791, –716.461426, 25.968819 –322.178650, –642.945679, –1081.934448 –619.731140
5	8.619967, –325.081543, –537.529907, –906.260193, –1086.351807 –1269.749512 –990.259827, –1082.904907 –659.676147
6	1975.103882, 1380.655762, 714.073425, 323.243988, 40.096863 –340.828369, –585.161987, –748.583984, –674.141174
7	–149.657196, –842.284973, –991.205566, –1085.922729 –1127.279663 –1483.629272 –1023.704834 –461.994324, –253.976318
8	–18.673189, 189.549789, –168.876450, –405.123322, –453.465790 –77.701363, 268.827789, 600.122742, 675.741089
9	–481.714874, –1312.720947 497.238983, 617.127869, 809.172363 738.139771, 787.788086, 698.119446, 695.123840
10	165.517990, 463.347443, 161.873764, –94.733566, –408.583618 –801.933167, –1183.599121 –1549.846436 –1829.587524
11	–54.419785, –532.895569, –852.639648, –1328.288696 –1952.080444 –2071.530518 –92.382095, –203.139877, –186.850891
12	183.861481, –367.273651, –851.810547, –411.512238, –523.118713 –562.440063, –624.123779, –710.631714, –784.563538

Index	Codeword
13	–177.176773, –521.651245, –243.776535, –638.196167, –178.797272 619.959656, 324.879456, –91.975945, –505.510773

Table B.41 (continued)

Index	Codeword
14	–257.685272, –815.271667, –835.617249, –1264.782471 –1022.726257 1117.330078, 928.825928, 663.468262, 264.610596
15	25.521532, 100.323082, –382.839996, –812.957520, –1234.972046 –1564.814453 –1898.064453 –1977.380615 –2026.754517
16	–176.444000, –438.630615, –730.473633, –1083.432129 –1262.495728 –1234.185547 –1185.764893 –1210.682739 –1209.831421
17	–360.419434, –785.268372, –865.262695, –1157.854126 766.852356 969.479614, 737.747498, 532.021301, 313.538544
18	73.917870, –175.426498, –555.667419, –1068.706665 –1075.443237 –587.874329, –348.189636, –173.672394, –40.342525
19	–171.413345, 57.096062, –278.747833, –399.241699, –545.361328 –765.097778, –964.719727, –1209.839111 –1428.539917
20	–357.214722, –896.660034, –37.925079, –239.189606, –322.952301 –631.056213, –714.007874, –913.038513, –878.441833
21	1577.066040, 1529.908569, 1551.067139, 1455.275879, 1443.303467 1338.591675, 1316.248779, 1216.262451, 1138.473267
22	–409.347382, –896.647339, 645.859741, 732.300842, 705.857483 302.651703, 108.818642, –286.890961, –332.548187
23	–245.251160, –851.509521, –998.825500, –1403.856812 –688.652405 –72.345734, –200.590012, –436.256897, –716.309570
24	–330.863983, –905.250244, –375.980499, –700.450745, –352.954773 –106.922882, –215.295853, –600.505005, –879.973511
25	–414.539246, –976.347595, 90.056046, 24.310310, 12.638400 –256.423431, –215.814499, –414.811096, –375.642853
26	–407.677155, 67.193260, –239.487579, –109.491417, 505.456665 979.732239, 1202.223755, 1395.712646, 1486.746216
27	–103.741013, 419.907928, 320.453217, 215.451691, 83.301155 –252.383514, –505.506042, –914.186646, –1144.382202
28	22.720533, –186.878723, –341.957092, –498.801239, –536.332214

Index	Codeword
	–952.614136, –1357.851318 –1808.392944 –2053.646729
29	–26.351635, –744.136047, –797.139587, –1550.414185 –1788.882568 –811.625061, –984.366882, –1143.760254 –1279.114746
30	–107.477005, –773.448425, –1269.981323 –1817.958374 –2156.203369 –2516.463135 –2683.684570 –2863.591309 –2623.000488
31	–95.678566, 204.781097, –189.688644, –453.370697, –564.583740 –524.080322, –323.315308, –217.640259, –82.427811
32	304.838196, 198.058044, 26.587915, 227.075500, 279.514038 90.428169, 25.509212, –293.474701, –421.430298

Table B.41 (continued)

Index	Codeword
33	705.637146, 193.382828, –315.379425, –720.216431, –1010.554382 –368.707733, –478.114960, –653.081177, –744.303284
34	–124.040001, –617.344604, –750.567200, –1259.906250, –1586.850342 –129.100784, –301.769806, –535.021301, –797.296326
35	–120.237625, 945.004333, 1023.360596, 1063.863770, 923.643372 722.150696, 610.020325, 326.225830, 204.115326
36	–242.126205, –1014.856384, –1121.360840, –1530.979004, –1499.600098 2.104447, 449.181427, 711.969604, 828.493713
37	–87.722557, –762.384583, –1226.589966, –839.319092, –951.323608 –971.707947, –973.453064, –954.515076, –996.160034
38	–168.498444, –970.490906, –1430.185791, –1907.497803, –2064.904053 –2020.493530, –1702.496704, –1505.148682, –1141.392456
39	235.255157, –214.254913, –735.741943, –1327.962769, –1079.209229 –1155.385376, –1207.560913, –1325.105225, –1136.770874
40	489.040497, 379.885406, 126.575279, 166.232132, 351.813141 480.143921, 609.508545, 543.443359, 531.942444
41	–153.286011, –209.206512, –534.360840, –915.646362, 70.271286 –16.743921, –169.522858, –503.006714, –716.120972
42	–345.934845, –1309.356201, –1890.006348, –2248.306396, –439.151978 92.394310, 370.706940, 471.791962, 618.747803
43	–310.524963, –842.749207, –696.589539, –105.589561, 71.578148 –348.991364, –368.559540, –730.835938, –645.298462

Index	Codeword
44	–306.158356, –1102.411255, –1307.940552, –1490.519287, –1121.813599 –795.747864, –604.703613, –509.754211, –467.392792
45	–279.305969, –1188.364014, –1767.922607, –2403.655518, –2683.665283 –1444.281616, –463.252106, –212.524490, 61.606976
46	647.597961, 8.382929, –398.628387, –858.689392, –1196.260742 –1258.485107, –1150.217529, –1349.348267, –1085.179932
47	–449.228638, 1142.992554, 1064.484375, 1191.123779, 1070.723755 1070.619873, 986.951904, 948.657349, 863.388184
48	–254.184570, 69.529388, 958.512634, 839.154419, 698.573792 330.158081, 380.084473, 116.397308, 39.122948
49	–247.474976, –291.702637, –560.705872, –726.362122, –712.323059 –847.740417, –819.774109, –972.264526, –1000.929565
50	–102.649529, –744.442200, –1258.676880, –1895.340088, –2534.861816 –3116.099365, –3649.456787, –3755.880127, –1480.863159
51	–224.878769, –868.444153, –1144.696777, –1584.141479, –1576.102783 –1216.113770, –629.454285, –408.229340, –209.635284

Table B.41 (continued)

Index	Codeword
52	–222.695526, 203.857239, 149.654663, 180.832245, 239.630402 195.768661, 296.441437, 316.698608, 426.070404
53	121.947586, –357.326324, –682.028442, –1382.699463, –1620.812744 –1094.719849, –1241.071045, –1532.304321, –1752.286865
54	–403.487427, –1423.419800, –1591.178345, 245.310394, 599.750122 789.179199, 882.052734, 882.112976, 816.936096
55	–36.153755, –279.553741, –632.521973, –8.765594, 35.625114 –69.890617, 16.580078, –204.663498, –340.181366
56	–199.492142, 81.513809, 32.626942, –178.653061, –278.078857 –547.297485, –676.165894, –952.851868, –1025.475952
57	237.930984, –243.357269, –671.658203, –733.092896, –758.623291 –951.332397, –1087.989868, –1268.829590, –1280.351929
58	–227.537750, –287.850098, –292.606018, –429.307404, –253.741806 –301.787720, –203.673462, –411.002045, –375.994415
59	–231.314774, 6.343965, 1509.542236, 1552.771118, 1414.968018

Index	Codeword
	1010.981384, 781.095398, 288.140961, 64.403107
60	–273.534241, –1185.445679, –1731.143311, –2246.872070, –1664.682007 –876.096619, –482.111023, –280.437042, –194.066895
61	–59.669891, –437.954163, –952.351013, –1369.370361, –1366.131226 –1311.555176, –1082.967407, –927.321472, –682.334717
62	684.509644, 105.415596, –391.989441, –928.875854, –1156.915527 –1280.749268, –705.104919, –712.426208, –747.950562
63	21.285646, 850.948059, 1489.332886, 1323.255493, 997.157410 536.696960, 184.663040, –327.162109, –538.481995
64	–672.908691, 1205.853271, 1076.858276, 1270.047241, 1163.228638 1278.610229, 1231.399780, 1384.312500, 1458.875610
65	1123.680298, 1195.730835, 596.091187, 120.459366, –367.943024 –844.443298, –1278.164551, –1673.333130, –1885.995850
66	272.514954, –301.388336, –37.983318, –253.615463, –214.135483 –513.741882, –462.207184, –767.507812, –831.534363
67	87.529411, –284.147980, –737.057129, –1398.583252, –1831.413574 –1200.515869, –1424.156372, –1247.542358, –695.322876
68	190.618988, –83.082245, –404.415741, –902.791565, –341.542328 –531.072021, –452.109833, –526.579529, –438.453522
69	–356.549683, –1249.736206, –1552.698242, –1509.339844, 96.601120 463.938782, 772.470825, 876.086670, 1027.148193
70	–251.656082, –741.916626, –800.877991, –1204.293213, –898.387207 –536.743469, –540.424072, –951.805664, –1371.296997

Table B.41 (continued)

Index	Codeword
71	–16.392891, –341.709778, –903.738708, –1518.366821, –2100.664795 –2723.664795, –3262.568848, –3655.256592, –3920.665039
72	–284.557129, –626.414917, –170.010101, –351.521912, 993.011292 707.154114, 430.032684, –56.580963, –208.748627
73	–63.585968, –276.151489, –456.599518, –256.501343, –333.562897 –755.818237, –1020.620972, –1435.192871, –740.306030
74	1629.834961, 1146.042969, 539.183167, 225.645462, 43.176285 –56.714531, –61.315525, 51.049088, 105.315742

Index	Codeword
75	–257.347992, –949.083496, –514.112061, –946.301208, –1061.262451 423.505493, 64.353943, –184.519806, –310.060455
76	601.338196, 730.051941, 529.544189, 350.136139, 123.053047 –167.247162, –385.363373, –718.114075, –827.812866
77	957.477234, 1673.412354, 1102.213501, 726.657349, 193.804886 –166.666946, –647.636536, –933.063171, –1035.739258
78	–177.455978, –666.090027, –713.341553, –1062.997437, –483.726715 –690.400879, –933.267517, –1214.673828, –318.606110
79	–189.500244, –405.898071, –668.867981, –764.180420, –19.830473 181.539169, 350.049530, 341.817993, 381.419586
80	58.198463, 87.016228, –470.353455, –1019.402344, –505.580994 –476.027191, –643.724976, –953.289673, –1218.301025
81	–9.731637, 35.897629, –398.662964, 108.671082, –30.824179 –403.612305, –511.957275, –623.195251, –685.248840
82	–157.306366, –553.577942, –975.876953, –1460.082764, –992.205627 –776.317871, –511.261353, –326.743317, –198.921204
83	–662.229797, 337.577576, 714.611816, 1115.122437, 1346.081543 1525.260742, 1630.367676, 1702.934814, 1783.298096
84	–96.698639, 476.073669, 356.679108, 307.013367, 275.333221 144.351913, 131.806870, –89.757820, –125.505257
85	96.625595, –234.501587, –735.064636, –1204.421265, –1373.773438 –1125.901245, –666.270874, –459.149506, –250.652649
86	169.706879, –187.591034, –705.413025, –1239.362549, –1701.577881 –1862.408569, –1851.633179, –1883.478271, –1916.869873
87	–64.554306, –62.929207, –407.237640, 103.524368, 40.249016 –404.697784, –763.795471, –1124.062500, –1390.584351
88	–141.843674, –793.938049, –1194.776245, –1549.655762, –1645.277832 –1719.506592, –1687.905640, –1854.041382, –1926.495483
89	–194.186035, –903.843506, –790.955811, –1279.267334, –942.369080 –715.308228, –985.681213, 525.467102, 483.847260

Table B.41 (continued)

Index	Codeword
90	2144.286133, 2143.108154, 1424.848999, 907.611938, 586.665710

Index	Codeword
	307.282593, 24.629150, –128.423553, –254.069794
91	–168.323730, –977.983582, –1317.796509, –1770.269409, –1939.598022 –2061.373291, –1970.316895, –1014.373840, –266.364838
92	–14.177239, –456.114960, –698.976685, –1179.255615, –1664.940552 –1250.903931, –484.642120, –708.259399, –900.153687
93	–243.909988, –627.637512, –352.178253, –584.734863, 317.760437 –84.799057, –425.512421, –952.116028, –1349.741211
94	84.997139, –326.942902, –436.166351, –822.967712, –793.385132 –1216.504883, –1579.634766, –1824.714111, –1103.522095
95	–78.882149, 557.116882, 243.507629, –59.426495, –326.818054 –656.480164, –882.566284, –1162.840576, –1292.122192
96	472.399078, 324.246429, 123.760132, 107.621017, 172.215866 100.703049, 220.447205, 70.769211, 93.025391
97	–279.016632, 319.207550, –34.997967, –125.365486, –3.664740 454.022797, 673.832275, 971.419556, 1070.634033
98	–86.031052, –99.652267, –551.247314, –880.790283, –1066.362305 –1154.501709, –1332.679810, –1483.808594, –1698.237671
99	139.571533, –74.816330, –730.456421, –1296.350708, –1943.886841 –2471.267334, –2555.464111, –2359.309814, –2118.012939
100	–232.477631, –323.773041, –457.001678, –708.959167, –360.985809 484.481323, 753.548096, 1113.769897, 1170.534302
101	–96.813431, –683.746643, –784.213257, –1114.412476, –1279.278809 –1705.722290, –1768.958984, –1196.400635, –768.627014
102	–234.763580, –1016.336548, –1436.150757, –1801.835938, –1697.735352 –1363.577026, –1055.104492, –852.366211, –672.977417
103	423.673950, 204.145523, –80.427010, –237.875229, –397.674561 –692.070435, –832.088562, –1137.829102, –1263.773438
104	504.952789, 140.112259, –276.739960, –574.314697, –427.731293 –264.691650, –6.429366, 59.627628, 128.354279
105	181.932892, –406.816956, –736.782471, –1377.915894, –2054.307373 –2382.816895, –1719.648071, –1395.232666, –1084.625366
106	89.986191, 1808.005493, 1425.416870, 1018.296204, 603.047913 200.996918, –203.648102, –614.505920, –859.780212
107	25.575348, –491.275757, –169.827484, –756.985229, –510.820648 –271.569244, –627.316101, –1105.351074, –1497.432861

Index	Codeword
108	–0.721510, 752.166748, 412.833252, 60.089832, –264.969238 –507.899292, –573.338013, –668.694214, –629.476257

Table B.41 (continued)

Index	Codeword
109	–157.224030, –924.664978, –1350.100952, –1907.713013, –2248.231201 –2433.884766, –2367.341064, –2260.742920, –1727.894043
110	–312.066467, –1067.247803, –1253.844849, –1673.790649, –1243.493896 –606.165710, –238.625320, –74.536827, 29.102182
111	188.922134, –387.703766, –181.650040, –684.412659, –465.589478 –816.664978, –761.582275, –1098.897217, –1035.031494
112	–502.310791, –1291.180786, –377.128448, –356.472809, 549.371277 798.475464, 1053.768799, 1193.387329, 1321.974854
113	120.875381, –86.850357, –507.376892, –934.800537, –1352.639282 –1755.437500, –2217.682129, –2608.354004, –2838.226074
114	–28.809593, –518.338684, –698.639709, –1113.990356, –1242.462402 –1523.905029, –1635.930420, –1677.925537, –1646.178345
115	–2.076617, 351.878967, 417.553589, 814.712280, 924.905518 541.083313, 210.906433, –301.686249, –554.606995
116	–174.367935, –101.753105, 506.573730, 213.724670, –62.353058 –467.419281, –549.319031, –683.690430, –647.043274
117	359.021729, 415.796295, 73.191391, –83.700104, –112.212723 –194.968842, –109.225288, –253.853516, –208.237640
118	–610.554260, 678.981201, 786.243042, 980.827698, 1033.276733 1177.275757, 1230.203247, 1353.269897, 1464.978394
119	–422.542664, –1154.585815, –685.516357, –561.854980, 237.946915 312.140289, 475.985229, 467.722260, 577.558716
120	–303.906586, –299.817566, –38.045284, –111.681908, –18.812908 –235.245544, –267.668060, –654.276001, –863.731628
121	–307.668579, –1263.498169, –1795.801514, –2345.257080, –1378.010864 –224.626175, 76.730064, 291.228607, 412.261261
122	–159.790817, –357.482483, 245.199310, –26.278240, –233.085129 –694.296326, –1010.374084, –1348.147705, –1435.401978
123	–341.661224, –972.107300, –596.526306, –812.299072, –721.850769

Index	Codeword
	–892.498962, –719.133484, –799.860718, –747.034363
124	208.944168, –571.758667, –1403.062866, –698.768188, –1295.974854 –1408.635620, –1278.894287, –1461.294312, –1620.020142
125	2016.535767, 2583.699951, 2156.421387, 1682.253540, 1306.288452 930.015930, 575.535400, 332.289062, 81.306969
126	–481.642883, 767.079956, 478.750885, 549.838379, 567.944458 842.335510, 934.838013, 1140.607788, 1189.669922
127	–651.013611, 1213.169678, 1114.427612, 1324.416138, 1282.206177 1429.387817, 1446.570801, 1614.566772, 1816.776367

Table B.41 (continued)

Index	Codeword
128	30.436600, –268.822021, –520.411560, –1239.242188, –502.846497 –964.426086, –957.118896, –648.420837, –727.841003
129	–67.992126, –398.622040, –443.123474, –723.682251, –735.579407 –836.367432, –9.895487, –451.035370, –848.732605
130	270.295380, –93.148148, –561.977783, –1183.561768, –1104.187744 –1177.850220, –1397.009399, –728.114258, –721.127014
131	–568.377502, 1.897134, –48.158859, 682.458374, 1064.579224 1367.183472, 1499.239258, 1623.660278, 1682.907227
132	–332.755524, –1277.148193, –1741.189209, –1430.306396, –47.963100 90.906227, 290.168091, 258.427002, 330.425507
133	–157.008728, –29.765123, –517.562866, –1025.533936, –1501.759155 172.473587, 378.613739, 263.816620, 157.555740
134	373.106750, 553.056580, 50.987843, –386.420197, –904.769958 –1344.794800, –1750.855347, –2064.066650, –2277.853027
135	–425.244843, –1345.192139, –378.066376, 387.963837, 549.967407 625.941589, 644.349915, 621.692932, 584.571045
136	–16.541977, 138.275986, –180.281464, –352.161438, 132.832703 87.394142, 201.175293, –36.061161, –62.357914
137	–618.062744, 1298.624023, 1185.361084, 1413.948364, 1361.496704 1514.361084, 1570.251099, 1806.322876, 2337.903564
138	–357.054291, –1350.821655, –1671.045654, –585.366516, 200.789124 301.339294, 449.095581, 433.317352, 471.774445

Index	Codeword
139	–108.307884, –546.904785, –848.436340, –1438.375000, –1839.243286 –789.302368, –314.564728, –132.880920, 70.787682
140	–308.546326, –210.140747, –150.658768, 375.302368, 643.934448 749.838379, 835.551147, 891.345764, 908.518005
141	–467.672546, 66.437386, 352.558075, 733.662537, 851.549316 1012.505676, 1108.903809, 1238.738037, 1367.354126
142	–225.670258, –176.692368, –139.687531, –297.688843, 484.590149 –12.994805, –385.974274, –824.333435, –420.143768
143	–321.209778, –609.134277, 326.301758, 326.337311, 475.917419 347.087677, 517.783081, 389.418243, 412.928894
144	304.724731, 317.969849, 168.403549, 88.447319, 16.958170 –236.259430, –315.584869, –580.212830, –623.322693
145	–224.797653, –70.260223, 609.323975, 491.570801, 339.155457 –11.061495, –53.120930, –337.209778, –425.730225
146	1121.619385, 1424.637451, 1250.940796, 1042.490601, 812.568176 620.814148, 503.227112, 336.917236, 223.762650

Table B.41 (continued)

Index	Codeword
147	–301.173950, –982.761230, –689.736267, –1041.371948, –1144.856323 –886.875671, –277.517120, –220.478470, –55.952248
148	1.571693, 303.673279, –95.214012, –438.626190, –701.699219 –824.135132, –750.147827, –931.409912, –964.330688
149	360.211334, 2345.139160, 1970.203003, 1550.722778, 1044.960815 669.019226, 233.912216, –111.696175, –383.211975
150	–198.679718, –459.207184, –647.677734, –856.236145, –497.603973 –371.051819, –88.504814, –52.162579, –3.863596
151	–234.505722, –1106.629150, –1637.704224, –2245.764404, –2716.826416 –2709.702393, –860.633789, –364.690216, –3.835027
152	433.529449, 360.604156, 53.111431, –97.453865, –204.377487 –441.845123, –556.891479, –850.518005, –914.695190
153	–94.201759, –682.459534, –1009.064270, –1450.390747, –1754.623047 –1931.054321, –2161.267090, –2073.351562, –1264.456177
154	158.497513, –39.482288, –491.640747, –1033.636963, –1495.640747

Index	Codeword
	–1644.056763, –1503.796021, –1429.203735, –1311.148682
155	–37.986122, 49.700111, 122.811394, 363.070068, 539.050903 389.771210, 358.130951, 42.615650, –54.583321
156	–190.028824, 35.819481, –72.171303, 416.761719, 468.406982 48.947430, –217.165985, –629.450134, –855.864807
157	690.035156, 678.191345, 374.738464, 116.511391, –177.495270 –559.347290, –781.556396, –1086.437988, –1198.095215
158	982.496399, 394.303375, –71.543907, –523.927917, –677.260681 –894.055725, –894.495422, –1078.243774, –1019.578369
159	–59.720707, –615.157593, –1011.128235, –1471.507812, –1689.347046 –1918.099365, –2118.354004, –2456.670654, –2522.051758
160	–19.764742, –471.573608, –823.959961, –1426.643921, –1940.420166 –1354.637817, –1006.084229, –600.211182, –315.964661
161	451.760864, –386.723450, –443.880493, –1157.029419, –2024.143433 –2307.281494, –988.382568, –643.842834, –581.822632
162	29.281801, –474.824768, –717.323547, –1104.237671, –1173.256714 –1403.408569, –1757.094482, –2137.989014, –2409.224121
163	–126.811966, –660.853577, –1009.586609, –1468.573242, –1837.390503 –1873.889648, –1122.959351, –944.855286, –682.774719
164	–201.716019, –376.240814, –457.417786, –95.987808, 325.228302 398.688049, 527.887512, 476.375580, 496.663727
165	–243.958618, –349.740448, –196.670685, –205.858994, –178.919907 –210.137161, –34.842102, 96.661407, 234.190201

Table B.41 (continued)

Index	Codeword
166	–22.390350, 489.585999, 985.910645, 861.128479, 541.654785 74.230301, –314.271606, –788.963013, –999.877136
167	222.074677, –287.362183, 74.757889, –117.507614, 15.699092 –264.202545, –159.531601, –392.734039, –338.009735
168	–181.277679, –638.428589, –680.598938, –1230.816284, –1306.731812 –876.499939, –893.502136, –641.560120, –465.425415
169	45.706028, –187.878601, –227.202896, –348.698608, –655.258301 –1177.003418, –1363.528564, –834.438538, –787.763733

Index	Codeword
170	538.307739, 95.790123, –343.766388, –791.648621, –1077.554932 –1263.006348, –1397.224121, –1605.978027, –1821.339966
171	–284.588654, –921.890015, –737.540161, –840.388123, –948.210571 –1147.289307, –1362.922241, –1606.361816, –1775.419800
172	–8.925940, 1316.169312, 874.871948, 478.259766, 134.867432 –152.284210, –396.394501, –642.910522, –747.362061
173	–570.521057, –1320.476807, 680.706299, 816.224182, 1261.479004 1220.774414, 1413.046021, 1383.548584, 1500.537842
174	–376.266327, –1358.371704, –1142.299683, 117.106003, 372.370209 457.582367, 465.587280, 445.678497, 394.158661
175	–171.539291, –748.739136, –1096.384766, –1554.841553, –1886.605469 336.762848, 160.452591, 4.648007, –53.059750
176	57.858696, –230.532944, –734.336426, –1199.881348, –1048.929321 –844.434143, –756.553528, –898.697021, –814.052979
177	102.989395, –343.258972, –396.959747, –646.421448, –774.568970 –1126.143433, –563.837280, –424.341248, –188.013916
178	–146.991638, –804.075195, –1119.743408, –1683.964355, –2263.950439 –503.006744, –721.959473, –673.123291, –445.826538
179	–395.988495, –1070.903198, –590.672363, 382.879913, 644.958984 241.903732, 46.419594, –411.099579, –550.546265
180	–157.615616, –459.796967, –879.255615, –1276.781128, –463.483490 –34.213669, 160.186905, 225.185394, 315.109528
181	–193.233856, –983.809692, –1439.684448, –2042.781250, –2454.115234 –2739.922119, –2691.471436, –1722.092529, –884.103638
182	–209.261917, –1071.798706, –1602.855591, –2159.484375, –2374.191406 –1981.226807, –1148.103149, –682.483826, –329.949249
183	–112.721451, 202.234741, 2.389480, –264.000092, –27.009350 202.602722, 26.050808, –430.921570, –765.662109
184	397.861053, 182.528809, –235.648392, –405.027954, –136.935242 –235.972595, –246.528183, –481.237274, –535.715332

Table B.41 (continued)

Index	Codeword
185	–278.234009, –827.781128, –752.883911, –466.989594, –320.336670

Index	Codeword
	–672.032898, –863.731323, –1161.255981, –1400.910889
186	–40.973869, –131.660767, –299.178619, –742.490723, –771.245972 113.764877, –196.912857, –561.841675, –982.662415
187	–404.340363, –1134.885864, –1034.914307, –997.058594, 267.367828 173.992477, –1.694684, –464.673004, –770.912231
188	–253.682114, –217.055069, –450.508270, –521.870178, –282.631500 –517.459839, –495.103638, –834.461365, –1062.342529
189	–249.738586, –937.397705, –640.001892, –675.438660, –570.580078 –407.106415, 1070.533203, 836.401245, 326.148163
190	527.339905, 719.609558, 720.028137, 822.540894, 849.751709 825.033508, 828.729248, 717.622009, 636.672119
191	–456.071869, –1250.948975, –104.695229, –75.758865, 139.109131 132.780731, 210.528854, 168.443298, 271.893005
192	–217.989136, –1060.941772, –1543.470825, –2165.294434, –2583.753662 –2728.703125, –1856.940430, –1027.727783, –544.276062
193	–474.045319, 458.974854, 837.952820, 943.188660, 1000.440247 950.523865, 926.026123, 833.391296, 796.308044
194	–90.405281, –689.172913, –543.798401, –1191.090942, –779.600403 –1011.817810, –1103.439209, –1140.771606, –1069.512573
195	654.475708, 1470.965454, 1283.830566, 1355.302246, 1215.090698 1242.490967, 1179.114990, 1123.327759, 975.381287
196	221.189468, –193.926590, –671.781799, –1185.121460, –1651.171631 –1605.885986, –1099.926270, –1008.106689, –976.504089
197	302.228058, 166.080658, –291.596680, –514.976562, –334.636627 –519.004272, –632.970520, –862.890137, –927.529724
198	–304.122864, –994.066772, –893.892944, –1265.938599, –1306.123413 –1279.007935, –1051.650269, –1009.602112, –776.189697
199	398.439453, –20.591383, –555.842163, –1159.980835, –1624.051880 –894.125793, –949.559753, –885.922180, –814.496460
200	92.801064, 1164.293945, 802.599792, 479.609894, 40.354275 –409.047546, –837.928162, –1215.925171, –1403.282349
201	–103.158302, 900.035217, 838.126953, 700.942932, 490.099823 215.066879, 54.649097, –221.317123, –332.887360
202	–316.185150, –795.790527, –465.281952, –823.125916, 279.398102 4.969502, –111.705475, –293.041595, –71.397591

Index	Codeword
203	–208.605103, –930.010925, –1317.236084, –1524.382568, –1237.802368 –1231.182373, –1223.147705, –1303.267090, –1338.722656

Table B.41 (continued)

Index	Codeword
204	–306.984497, 524.299561, 496.007812, 554.238037, 526.116821 552.875793, 621.609985, 647.903809, 648.063293
205	–191.399063, 106.350975, –10.322931, –235.823715, 781.303772 475.559570, 138.095871, –381.311462, –703.843323
206	427.002899, 193.730301, –249.266571, –576.741150, –672.449402 –905.434448, –1067.168945, –1354.593628, –1497.148926
207	–470.544312, 1433.881592, 1415.007812, 1581.021729, 1347.515015 1305.963989, 1181.518433, 1046.481812, 899.044922
208	–280.197632, –426.098328, 67.108482, 72.767601, 313.018188 100.961258, 144.139343, –174.612595, –228.310028
209	11.646732, 964.122498, 523.617188, 150.000824, –173.893707 –157.686981, –103.815186, 156.810150, 247.870422
210	–332.143280, –1160.115356, –1280.343750, –767.366089, –498.519501 –533.037537, –494.996948, –569.730042, –518.864563
211	433.381531, 15.187217, –449.491577, –703.293030, –803.316833 –835.974365, –792.097107, –951.533813, –945.431335
212	–749.471924, 13.263083, 1330.948242, 1355.199341, 1704.749146 1687.224976, 1855.235474, 1803.945068, 1879.865356
213	–329.422394, –1097.037354, –1218.800781, –1606.691040, –640.537842 –167.328995, 50.598976, 132.925644, 204.857117
214	–76.864571, 1566.287720, 1979.153564, 1816.987061, 1480.883545 1051.511841, 680.155212, 283.731506, 49.960445
215	–398.984741, –1099.112671, –441.023163, –865.695312, –596.991150 –165.189606, 145.491440, 313.867889, 534.415955
216	–344.061218, –787.391846, 314.467285, 335.746521, 330.660370 –155.917572, –427.322876, –874.026428, –1033.887695
217	–218.076859, –471.678925, –740.102173, –963.206055, –691.495605 –631.507080, –526.847839, –547.032471, –472.381805
218	–65.633423, 1715.546143, 1273.253662, 891.733337, 491.938904

Index	Codeword
	317.231781, 236.129700, 258.489990, 122.087456
219	–182.897858, –635.679382, –839.284912, –1327.341187, –1097.753052 –407.501709, –31.804829, 98.519104, 220.063461
220	–272.900055, –1190.943359, –1664.515991, –2272.859375, –2254.459961 –658.264221, –92.030678, 162.460190, 371.485138
221	110.549355, –97.581825, –448.147552, –1003.613464, –1313.954224 –490.750549, –771.622314, –1146.729614, –1488.419556
222	–15.722897, –343.917999, –623.659790, –1222.161865, –525.122131 –746.976440, –1098.472656, –1486.665894, –1646.623291

Table B.41 (continued)

Index	Codeword
223	–27.274996, 110.961189, –360.191864, –839.653381, –959.202759 –1052.568970, –1121.710815, –1161.902222, –1138.799072
224	2.914193, 183.097900, –369.737396, –906.104187, –865.116028 –992.003296, –587.679138, –629.010742, –564.785950
225	–199.768204, –522.689270, –259.972870, –581.607056, –790.009460 –1049.283936, –1102.648071, –1337.240479, –1395.463989
226	66.927528, –428.277649, –187.231018, –836.323792, –1125.244873 –458.102875, –647.304199, –886.505554, –638.468018
227	–94.328537, 261.429749, –97.315033, –452.707825, –832.687073 –1155.034424, –1376.725220, –1519.680664, –1655.947266
228	403.557434, 222.425934, –116.075737, –299.235687, –451.025604 –742.190613, –672.085266, –684.950989, –390.703125
229	–77.066376, 77.830589, 277.074188, 636.396851, 923.212097 808.036011, 707.707886, 389.819733, 273.664062
230	–233.448212, –258.093109, –270.512939, –414.142883, –485.810974 –690.496765, –622.794006, –712.351135, –551.438660
231	571.873657, 673.336548, 634.134155, 580.569336, 501.235016 329.346832, 329.521454, 114.622795, 51.585625
232	–101.469490, 666.245056, 543.401917, 582.935486, 546.431091 420.524933, 404.157104, 195.331772, 161.434845
233	–176.970886, –737.287842, –1058.790771, –1613.152466, –855.213684 –747.337891, –887.670715, –818.685181, –779.836792

Index	Codeword
234	–374.160767, –992.195374, –447.817047, –538.550659, –421.040131 –537.537476, –373.479462, –352.113800, –141.020050
235	–754.974487, 635.823425, 1339.891724, 1442.483521, 1692.018677 1734.460083, 1852.739746, 1843.592163, 1893.666260
236	–158.565567, 610.178833, 436.004883, 295.670258, 169.214249 –54.113548, –182.365295, –487.801361, –587.581665
237	–238.826248, 200.013123, 110.359848, 71.252274, –9.610732 –190.478989, –175.310699, –294.996704, –249.046890
238	156.508057, –156.490112, –465.772675, –913.607361, –1022.254211 –1363.447388, –615.717407, –1028.598267, –1266.942139
239	–356.656403, –1146.559814, –1022.871277, –1108.460815, –90.339676 72.129295, 232.148758, 237.532684, 313.416077
240	–297.429688, –711.554810, –216.735733, –505.421326, 551.118713 250.093857, –38.674564, –525.157959, –849.864563
241	–295.815125, –1209.349854, –1776.149536, –1998.250122, –592.952271 –384.668030, –185.576477, –164.222015, –136.108200

Table B.41 (continued)

Index	Codeword
242	–132.807831, –638.202820, –917.321838, –1318.292114, –1810.082275 –1220.721802, 685.909424, 556.612610, 370.619812
243	–197.379715, –1072.747314, –1440.673584, –1876.071411, –2089.344238 –1721.490356, –494.098083, –140.939713, 217.494400
244	508.745697, 620.538208, 498.007721, 371.627930, 241.868118 6.471931, –64.585320, –294.206207, –329.333954
245	–461.752197, –504.665894, 834.920532, 809.721191, 1020.176880 871.390442, 903.949707, 739.781250, 752.165588
246	68.185127, –25.366402, –289.149689, –775.257812, –47.512859 –551.142700, –992.717712, –1406.638184, –1423.804077
247	1074.732666, 523.538391, 143.826492, –187.822311, –245.405121 –337.596924, –392.422150, –526.609680, –543.828674
248	–121.817963, 286.786957, –32.153019, –268.764130, –310.116547 –382.739136, –381.658600, –611.214905, –667.152649
249	368.183838, 129.406418, –327.912201, –1042.670898, –555.348694

Index	Codeword
	–921.933716, –976.043030, –1177.621948, –504.788055
250	–243.511139, –747.707458, –671.397400, –1178.495239, –75.772865 –466.295319, –582.133057, –871.793884, –1050.080688
251	–106.184364, –608.038757, –968.961609, –1434.189453, –1641.399658 –1630.353271, –1443.817383, –1404.703247, –1230.599854
252	718.401611, 2704.733887, 2630.791748, 2413.644775, 2020.989502 1672.132812, 1268.989136, 930.174011, 549.528198
253	–280.844727, –837.520813, –893.299866, –1385.495483, –334.480225 –482.172333, –454.969421, –286.658356, –111.712273
254	–294.975677, –1109.926758, –1316.428345, –987.775269, –629.906128 –331.063782, –88.151436, –42.998192, –9.167728
255	–720.846436, 1115.457275, 1445.276978, 1626.742798, 1758.635132 1845.389893, 1893.553589, 1901.911011, 1885.918091
256	–10.630882, 173.645126, –194.461044, –520.689392, 145.682388 –76.744125, –411.265472, –869.935303, –1199.078491

Table B.42 Stage 1 second sub-vector codebook lsf_stage1_CB2_lbr of low-precision LSF vector quantization codebook

Index	Codeword
1	–1025.445557, –1558.926147, –2144.454346, –1791.063965, –1952.295288 –1405.597290, –2150.003418
2	–954.523376, –818.689392, –1113.715698, –887.345154, –1365.084351 –1268.915771, –2110.815186
3	–619.806885, –685.579651, –746.767395, –691.805176, –616.305969 –647.728027, –512.941956

Table B.42 (continued)

Index	Codeword
4	–59.775761, 22.873430, –414.587616, –322.859131, –903.828430 –888.348755, –1863.307739
5	–1225.429932, –1281.655762, –1576.963623, –1762.252441, –2420.828369 –2961.331543, –318.643402
6	–847.398743, –1271.069458, –1061.816650, –168.675690, 50.615589 259.891388, 311.872589

Index	Codeword
7	1678.143066, 1741.549927, 1650.775269, 1412.182495, 1118.326782 889.213501, 657.603821
8	–1893.017090, –2061.548584, –1934.615967, –2040.489502, –1998.668701 –1757.438721, –1278.423218
9	–1981.158325, –2000.392334, –1779.247070, –1688.792236, –1332.873413 –1121.153931, –940.593201
10	–1115.811279, –1152.664795, –1552.353027, –1439.754517, –1908.486694 –1673.207275, –2451.110107
11	–379.387024, –26.146738, 41.422009, 164.160645, 123.430641 175.891678, 62.278313
12	–2213.671631, –1141.692993, –609.784058, –353.606750, –85.206703 148.116455, 181.632812
13	–1154.319214, –1227.523071, –1489.456421, –1601.720459, –2058.670654 –814.973206, –1702.502808
14	–918.795044, –1343.674194, –1183.565918, –950.312866, –1186.982422 –1103.976562, –1854.349609
15	227.948730, 1136.904053, 1219.321167, 1506.619507, 2232.180908 2348.839111, 1403.756592
16	–2236.907715, –2289.355713, –2560.764893, –2482.412842, –3092.014160 –2176.867676, –317.259705
17	–1555.807617, –501.324615, –1023.675842, –1040.626343, –1486.283691 –1266.904053, –2026.519897
18	–1669.823242, –1464.875244, –1696.568970, –1326.180298, –1452.729004 –840.359619, –1714.870850
19	–1532.789062, –1729.420166, –2049.086426, –1779.276611, –1770.308472 –1014.531860, –1772.082520
20	–1528.967285, –1655.654541, –1899.280884, –1819.910645, –2133.318115 –1590.653442, –2254.575684
21	–1144.653076, –948.210571, –870.286560, –501.640564, –793.606384 –671.439697, –1547.295288
22	–461.364410, 682.552795, 461.194458, 359.740967, 362.942017 339.776062, 376.410309

Table B.42 (continued)

Index	Codeword
23	–1149.120239, –1575.110596, –1927.880615, –1335.437256, –1518.165649 –1044.578857, –1843.897095
24	2135.116699, 2256.741455, 1956.162842, 1594.127441, 1218.817749 904.284485, 682.448364
25	–1041.697021, –1173.235107, –633.627747, –966.614746, –1525.846680 –983.940491, –1708.915771
26	–1837.993042, –2091.420898, –2478.027588, –2809.757080, –3339.826416 –3897.479736, –4707.816406
27	–445.076660, –157.336899, 0.462463, 312.914917, 378.175018 418.287964, 437.912292
28	–439.980408, –449.660858, –1115.516235, –1478.943726, –2040.065186 –1754.303345, –2473.741211
29	183.339447, 101.839279, –21.999655, –217.910965, –431.363098 –963.968079, –1546.019775
30	–1504.246826, –1618.718872, –1747.502930, –2033.336304, –2296.113037 –2512.910645, –1610.453125
31	–917.428162, –986.813293, –1067.608032, –1200.509155, –1362.467285 –1514.312012, –1158.368652
32	1395.400391, 1494.203003, 1504.630981, 1298.671509, 1035.527832 821.061829, 624.906738
33	–448.509827, –513.075562, –838.923401, –349.190704, –993.107971 –1118.479248, –2049.759033
34	–612.845703, –1007.675659, –1485.587158, –1165.256592, –1222.697632 –840.847656, –1658.234619
35	–1450.915405, –1349.162354, –1323.501709, 74.000221, –380.690308 –498.892334, –1499.937378
36	–2156.536133, –2291.489014, –2427.126465, –1819.702637, –1754.847168 –1190.418701, –1890.954834
37	–1499.903687, –444.774536, 768.410950, 1536.175415, 1272.848511 1048.861816, 772.685303
38	–887.627563, –822.545837, –998.152100, –562.538330, –786.750061 –142.196915, –868.241516
39	1133.573730, 1022.729675, 933.447632, 929.575195, 746.131165 643.459045, 421.243073

Index	Codeword
40	–594.492798, 448.852051, 1012.362915, 956.231873, 752.287537 587.962708, 512.185059
41	–1410.218628, –1474.849854, –1449.908081, –1601.166992, –1158.019897 –255.972961, 72.685135

Table B.42 (continued)

Index	Codeword
42	–1266.662720, –1197.473999, –1101.059326, –1019.961609, –935.622437 –897.695129, –1165.877075
43	–1161.196167, –1358.582031, –1438.956055, –1154.027466, –1487.140015 –1287.429932, –2068.938477
44	–1951.491943, –2276.684082, –2605.463135, –3121.375732, –3703.884521 –4444.492188, –1389.477295
45	513.975647, 12.077785, –122.189568, 475.613800, 431.415344 426.029419, 416.788300
46	–847.774170, –475.462006, –815.791504, –800.997925, –1365.104858 –1477.232422, –2320.281250
47	264.021484, 246.813354, 64.843918, 40.896435, –145.363602 –482.314117, –1039.475708
48	–855.705627, –193.679138, 587.207397, 741.479309, 611.131592 490.605042, 419.768677
49	–904.709534, –993.312683, –997.528381, –1657.625854, –1188.919189 –997.200623, –1495.383057
50	–931.204895, –1047.634766, –1584.752808, –1791.148193, –2180.735107 –1455.360474, –2169.037109
51	–1015.164124, –1246.681641, –1685.658813, –1876.436157, –1603.031738 –1116.275635, –1803.888672
52	–851.682434, –633.694946, –451.774994, –389.725128, –313.773590 –480.187469, –692.924622
53	–712.982483, –691.948669, –657.795349, –648.190430, –834.174927 –1056.077148, –1796.616211
54	1951.557739, 2124.905762, 1876.367065, 1546.324097, 1187.195190 894.609985, 670.870239
55	73.852234, 65.499107, 665.957153, 829.415833, 664.944092

Index	Codeword
	553.647522, 477.308777
56	–225.363541, 470.226196, 1420.814209, 1524.770996, 1257.211914 1054.867676, 779.712891
57	–3066.772217, –3320.171875, –1755.439575, –913.461670, –465.759613 –57.672546, 181.598816
58	394.747803, 543.978943, 301.814484, 373.118195, –203.041214 –410.210175, –1401.083496
59	–254.991013, 6.490180, –86.547020, 334.854797, 192.452530 419.970276, –669.483948
60	–853.813477, –937.095398, –1324.438965, –1171.276611, –1614.367432 –1207.909180, –2010.941772

Table B.42 (continued)

Index	Codeword
61	–79.464409, –308.357544, –878.415833, –1053.853882, –1480.095215 –1042.867920, –1868.481201
62	1119.224121, 2373.954834, 2685.428467, 2891.306641, 2396.660889 2056.481934, 1200.541382
63	714.623291, 438.046265, 73.563614, –58.022953, –9.994848 127.793358, 296.390991
64	–672.743042, –884.477173, –1117.022827, –1600.467896, –2110.626709 –2974.947021, –1047.465210
65	–1300.258667, –1228.864258, –1260.032227, –648.009460, –942.031982 –747.062378, –1653.133667
66	2883.623047, 2563.917480, 2196.644043, 1742.524658, 1349.104370 980.756714, 733.060547
67	–1098.610962, –1080.493530, –1496.681274, –1381.791016, –1795.001709 –1327.081787, –2094.705811
68	198.491592, 526.795959, 1006.147949, 980.373901, 778.986816 607.859619, 511.755981
69	–1.237812, 160.743134, 234.740402, 367.119659, 406.874481 440.592163, 422.110687
70	–1550.515137, –851.167358, –1311.250366, –1574.810425, –1835.150879 –1107.781616, –1898.006836

Index	Codeword
71	–284.272217, –433.876221, –428.981812, –981.543457, –1270.818359 –40.644142, 220.977402
72	1028.936279, 1198.602417, 1347.095581, 1191.527466, 947.432373 748.339233, 588.323792
73	–527.319885, –372.557434, –520.435181, –530.666870, –745.792786 –1036.270386, –1784.524292
74	–1261.415161, –1271.482300, –1285.284912, –1491.556152, –1492.168945 –1121.455566, –542.928772
75	–1051.774536, –575.885498, –880.983459, –893.896545, –1280.546509 –1003.587708, –1856.981445
76	–1484.438599, –1497.676758, –1417.937378, –1478.414917, –1270.368530 –1210.548706, –1377.488403
77	–2284.488281, –2475.568848, –2549.678711, –2380.119385, –2334.494873 –1691.260498, –2079.139648
78	–849.849915, –808.925293, –1187.607544, –1324.695801, –2218.496094 –1798.911377, –7.794962
79	–1739.335327, –1621.678223, –1923.265503, –1880.257202, –2534.694092 –1862.062500, –86.207451

Table B.42 (continued)

Index	Codeword
80	–2400.431152, –2023.654663, –1612.980591, –1507.007935, –1438.144409 –414.243439, 18.127321
81	1379.186890, 734.271179, –343.965485, 666.540344, 418.169128 586.979736, 159.339020
82	–1327.495117, –1011.947632, –1257.065674, –1106.393188, –1519.401245 –1203.832642, –1988.876465
83	–2900.826904, –2071.155273, –931.973999, –620.055481, –238.298386 51.354084, 185.485703
84	1416.857788, 1443.416748, 1396.016602, 1061.671143, 539.757629 –57.588993, –529.872070
85	–1381.020996, –1011.073425, –1117.733521, –799.515869, –1151.570068 –1151.280762, –1968.326904
86	–664.578979, –340.874969, –581.869263, –385.350647, –855.842957

Index	Codeword
	–723.654907, –1668.381348
87	–239.082840, –358.530243, –462.077606, –755.325012, –1069.406006 –1435.389771, –362.738647
88	–2198.786377, –2450.488281, –2962.151123, –3407.308594, –1153.890381 –478.794434, –34.776772
89	–1491.307373, –1808.472168, –2021.277832, –2369.602783, –2064.542236 –511.397156, –23.106594
90	1012.056763, 930.190796, 809.964172, 712.060120, 556.017761 356.482849, 418.835175
91	–223.605896, –162.689224, –623.514648, –634.457458, –1285.401611 –1279.381470, –2197.202148
92	–2013.476807, –1931.635864, –2086.846191, –1884.910889, –2132.437012 –1781.376221, –2355.531250
93	–1478.611694, –1502.686279, –1579.426392, –1776.563843, –1922.639282 –1700.766479, –863.249451
94	399.884247, 1336.809937, 1467.834839, 1231.440308, 944.319824 714.730347, 566.275085
95	–1909.513794, –1941.199341, –2085.629395, –2205.229248, –2642.804443 –3147.793457, –475.752045
96	649.063477, 491.458344, –353.614777, –123.980179, –189.024261 –512.889526, –374.362457
97	–255.760773, –218.515747, –651.565857, –579.347717, –1121.250610 –875.253906, –1822.844360
98	–356.398987, –322.802887, –341.252716, –337.389313, –368.309387 –564.473328, –961.036682

Table B.42 (continued)

Index	Codeword
99	–628.832153, –520.285034, –588.919617, –597.740173, –655.028503 –689.380981, –1164.055420
100	–715.569885, –1016.947021, –1246.564453, –613.046509, –899.559570 –700.425415, –1602.584961
101	–495.605225, –498.591705, –890.112549, –794.450500, –1308.796021 –1155.602417, –2031.677734

Index	Codeword
102	–875.373779, –971.543274, –1524.570923, –1737.737549, –2204.225342 –2113.908936, –2749.268311
103	–1709.141968, –1925.645508, –2331.198242, –2283.702637, –2462.921387 –1692.611084, –2252.493164
104	720.386292, 1149.667725, 1467.922485, 1450.620361, 1317.223022 1116.553223, 841.229187
105	885.497620, 1587.753906, 1888.884277, 2303.653809, 2150.790527 2051.243652, 1198.554810
106	–13.404860, –174.803833, –402.282806, –536.339722, –521.674011 –470.419830, –92.618591
107	212.016647, 135.590973, –435.995789, –544.691711, –1228.437378 –1122.961548, –2034.787964
108	–125.149994, –583.985229, –662.346069, 246.242020, 307.632996 329.847107, 357.718689
109	751.541626, 577.168945, –540.336487, 58.593849, –367.797974 –486.880280, –1335.964600
110	–771.941772, –908.973206, –1379.661743, –1369.286865, –1839.978516 –1582.273926, –2354.780518
111	2588.802734, 2444.061523, 2091.616943, 1677.830322, 1289.599731 942.811218, 707.786316
112	483.501984, 217.483292, 386.499664, 296.597534, 220.394058 –279.699463, –232.500107
113	750.680847, 671.408386, 492.794983, 466.070251, 331.014160 265.037689, 262.980225
114	–931.252686, –739.344910, –479.008026, –292.997284, –39.842056 155.129913, 252.702530
115	–699.944885, –320.508942, –448.116058, –140.793747, –555.594910 –550.571533, –1499.539185
116	–3182.762207, –3706.974365, –3368.149658, –1724.299561, –899.661133 –304.383423, 102.869270
117	–2198.285400, –2661.550781, –2924.803711, –3695.177734, –3522.947998 –893.561279, –200.487762

Table B.42 (continued)

Index	Codeword
118	–1914.781372, –2263.327148, –2204.333252, –2708.312744, –2737.251709 –3691.747559, –1746.551514
119	–1176.847168, –1399.460693, –1944.353516, –2252.302734, –2727.296387 –1979.486816, –2509.687744
120	–617.079163, –400.202545, –169.513504, –64.215126, 141.336349 273.932709, 350.984192
121	–146.712021, 29.761538, –185.886520, 36.444355, –317.681915 –9.635806, –882.568787
122	–853.681030, –618.070862, –848.745422, –567.171326, –964.919678 –700.323547, –1625.511230
123	–1866.847534, –1855.031616, –2043.105835, –1959.685913, –2075.854004 –1290.863892, –1966.472046
124	–1240.036987, –1385.001587, –1153.131470, –1487.872192, –1927.276245 –1474.927979, –2165.958252
125	–1160.221924, –1394.681763, –1689.331177, –963.545776, –1117.765137 –785.958801, –1655.615601
126	–2506.828857, –2412.121826, –2366.885010, –2094.813232, –2321.790039 –1103.879517, –68.931801
127	116.747658, 309.537476, 102.929665, 244.603302, –300.030518 –443.569061, –1433.695190
128	–311.663879, 326.289520, 452.395294, 832.174316, 276.240356 –265.085205, –1203.591797
129	–312.455383, –534.556335, –898.104919, –421.191711, –870.713501 –686.010254, –1610.913574
130	–2223.857178, –1749.410156, –732.078979, 776.504517, 1279.655273 1057.796753, 730.522949
131	–1278.438965, –1337.653198, –1735.267212, –1699.908691, –2139.395508 –1795.072876, –2496.341064
132	1878.921997, 1922.607544, 1767.357666, 1498.063232, 1173.502563 926.899414, 677.734863
133	–860.766418, –959.988586, –1574.841675, –130.846207, –605.896606 –1274.775879, –2029.413574
134	672.132080, 784.561218, 542.281128, 505.435944, –81.187279 –364.267731, –1346.734863

Index	Codeword
135	–679.825806, –396.701569, –747.309204, –612.425232, –1114.258667 –951.520203, –1883.500000
136	273.454254, 505.299377, 380.777863, 720.950867, 446.219635 470.014709, –593.205750

Table B.42 (continued)

Index	Codeword
137	–552.414307, –614.149353, –771.000549, –820.348267, –315.271545 76.449310, 226.295868
138	–1588.281372, –1638.489868, –1975.302124, –1913.472290, –2280.515137 –2005.639771, –2615.034180
139	–1591.169922, –903.552734, –947.961975, –1538.241333, –1246.838745 –920.043213, –1608.986206
140	933.789246, 1132.058594, 909.458252, 843.237366, 217.935349 –259.025299, –1224.878418
141	–819.524170, –796.189087, –839.038879, –872.878601, –965.350586 –881.141785, –1175.653076
142	–985.351685, –982.313110, –1018.421204, –1146.665039, –1041.144531 –862.111694, –429.493561
143	–2147.105469, –2150.295898, –2634.779297, –1851.966553, –574.587769 –173.353073, 201.615067
144	808.396484, 711.239441, 766.214478, 544.480957, 387.048828 –159.360245, –213.043732
145	–835.451721, –901.315674, –1511.963867, –1617.422241, –2408.831055 –2809.031738, –3807.552979
146	–681.278564, –663.108459, –1050.592529, –1039.863647, –1489.105103 –1050.743896, –1903.991455
147	–1231.333252, –1653.905273, –1746.697510, –2400.019775, –2611.556152 –3640.648193, –2618.129395
148	–823.225830, –994.271667, 34.345436, 465.466858, 403.638428 371.257141, 349.227234
149	–308.299561, –348.854034, –380.349731, –274.439667, –135.211472 29.159761, 168.604477
150	189.439529, –19.790514, –354.154694, –546.065247, –146.602478

Index	Codeword
	230.009827, 295.738068
151	461.968658, 626.050415, 47.854416, –136.378067, –974.916626 –1239.504639, –2210.014404
152	–146.661850, 514.339966, –594.032837, –584.835510, –429.045593 –687.339600, –1210.203613
153	–36.429043, –63.248245, –169.160934, –306.868805, –515.846008 –979.940063, –1618.352295
154	–1450.773315, –1292.733765, –1625.638428, –1334.478027, –1652.320068 –1156.253296, –1948.120972
155	–1609.528809, –1748.195923, –1849.325317, –902.303650, –317.236542 59.964432, 245.630280

Table B.42 (continued)

Index	Codeword
156	–478.842133, –551.974731, –931.206116, –845.584106, –1238.124756 –782.011475, –1684.930542
157	–1868.412720, –2019.276978, –2302.041748, –2320.242676, –2630.770020 –2504.333252, –2883.652832
158	–1334.282593, –1380.305420, –1712.473755, –1599.545898, –1922.840820 –1364.126953, –2102.723389
159	674.909668, 593.001526, 632.620483, 739.557922, 647.171631 598.846130, 504.963593
160	–1293.485718, –1619.509521, –1743.204712, –2296.322266, –2625.175537 –3608.674561, –1111.967285
161	–1551.716309, –1261.823730, –1537.043457, –1439.098389, –1946.865112 –2143.446289, –2936.749756
162	–1865.563477, –2266.884277, –2327.579102, –2886.662109, –2986.587402 –3896.322510, –3199.838867
163	–1297.908936, –1507.185547, –2026.442505, –2387.808105, –3053.241699 –3704.563477, –4686.282227
164	–1578.633423, –1579.114258, –1440.448486, –1311.780762, –939.062073 –843.284424, –714.103027
165	1687.707886, 2044.316162, 1825.541992, 1497.671753, 1145.952026 849.316101, 651.482178

Index	Codeword
166	–1009.696899, –222.117264, –771.609741, –1619.151855, –1208.036255 –1003.212463, –1714.239868
167	–1810.316895, –1231.567627, –1636.892090, –1864.545288, –2074.797607 –1385.536133, –2063.006836
168	–710.491760, 492.150574, –503.805817, –229.187119, 110.906113 282.697723, 55.507629
169	–1055.189941, –1082.506836, –1631.291016, –832.915344, –1290.385132 –1288.410645, –2099.172119
170	–677.468689, –1213.786987, –1832.875732, –1311.399902, –1655.118896 –1257.517700, –2133.393066
171	439.705566, 475.338715, 70.628548, 94.696815, –488.101837 –589.449707, –1577.500610
172	–1090.651245, –966.298645, –1281.871216, –1071.899048, –1602.906494 –1546.179932, –2356.489746
173	–176.466660, –259.544586, –842.928284, –904.868774, –1627.998169 –1901.671753, –2790.327637
174	–465.968689, –549.631042, –734.263306, –1057.186890, –976.384277 –689.167419, –222.912125

Table B.42 (continued)

Index	Codeword
175	–1158.937134, –119.719742, –646.479370, –644.535522, –933.463501 –792.865967, –1670.845459
176	–750.801086, –706.406555, –713.155640, –272.636993, –658.693359 –627.088867, –1560.627197
177	–1711.215698, –1495.088745, –1775.548950, –1505.336792, –1877.358032 –1675.769531, –2388.443604
178	–1407.030518, –739.424866, –1274.515137, –1506.798096, –1997.677368 –1659.833252, –2381.872070
179	–1128.046753, –615.983215, –803.808105, –489.734253, –953.664368 –980.357361, –1857.874634
180	1321.423096, 1831.520020, 1710.042847, 1410.556519, 1080.634644 804.950562, 626.732788
181	–1208.621948, –1512.455566, –1852.088623, –1282.296753, –1691.754395

Index	Codeword
	–1564.541748, –2293.062256
182	–130.666595, 106.064713, –89.260017, 103.645401, –379.526367 –480.569031, –1442.557129
183	–846.357117, –764.076843, –1203.165894, –1197.817749, –1811.429077 –1983.140259, –2796.902344
184	–46.024593, –110.736679, –447.505859, –469.773193, –874.538391 –402.937805, –1133.668335
185	–1509.996582, –1677.197876, –2105.592041, –2329.006592, –2809.195801 –3156.773926, –3939.870605
186	–1452.266235, –1189.537476, –1349.120117, –1023.163513, –1286.727051 –908.575500, –1733.368286
187	53.716125, 98.533348, 124.961006, 58.117867, 82.433022 –320.471771, –455.971619
188	–1663.746948, –1640.456543, –1816.988037, –1481.121338, –1737.681274 –1288.965454, –2014.165039
189	–1192.195312, –1001.089722, –831.286682, –678.389160, –372.874237 –121.623199, 6.853171
190	–593.864868, 172.604919, –302.387390, –386.411194, –1001.882935 –940.392395, –1903.361816
191	–2012.852295, –1209.255737, –1386.427612, –1058.992920, –1368.542480 –1097.503174, –1899.640991
192	–1302.134644, –1512.447388, –1917.826904, –2187.393311, –2412.989014 –1257.125000, –1899.332275
193	–662.509155, –142.376114, –597.944763, –952.603516, –1486.605835 –941.209778, –1875.773926

Table B.42 (continued)

Index	Codeword
194	–766.835876, –813.700745, –965.409729, –1245.255371, –1896.465210 –555.912415, 139.788239
195	–53.709496, –24.457977, –27.835449, –9.611705, 202.381714 296.835175, 376.950104
196	–1525.039551, –1650.326904, –1216.550537, –1621.969482, –1727.015625 –1065.248413, –1784.243774

Index	Codeword
197	–813.587952, –693.967896, –843.175720, –1081.308472, –1468.924072 210.317993, –761.495422
198	105.552757, 71.800415, –42.032394, –72.381523, –146.560120 –183.135468, 85.448723
199	265.414490, 642.176758, –456.374176, 320.947174, 278.549408 459.789032, 103.696960
200	220.106293, 364.429077, 130.036377, 291.826294, –106.766251 113.313423, –802.879395
201	–3134.160400, –3552.298096, –3854.004883, –4320.645508, –1937.848022 –665.023315, –176.597946
202	–1004.107605, –1031.160889, –1016.486206, –1210.363159, –849.197205 –107.791985, 161.588196
203	–693.122864, 325.670166, –643.304443, –1307.331299, –789.680725 –839.081665, –1511.067627
204	–391.568237, –99.078201, –259.886230, –22.610729, –472.509125 –513.719727, –1472.088257
205	–1575.965454, –1609.202637, –1605.129883, –1001.017700, –1327.703369 –1221.082031, –2007.426270
206	–354.735535, –381.940735, –894.146545, –998.646545, –1582.024902 –1441.162598, –2270.900391
207	–750.359741, –745.182007, –1095.111572, –693.184387, –1186.341431 –954.111694, –1859.068237
208	250.793350, 308.514526, 257.658905, 259.137177, 223.083923 209.800003, 187.700897
209	–398.730713, –282.424469, –167.861862, –148.836700, –112.436821 –333.931335, –371.274994
210	1144.176147, 854.793945, 247.951218, 476.229004, 262.055573 –210.412064, –311.474915
211	549.641846, 657.539246, 324.278534, 175.408035, 147.500793 –372.890747, –527.594421
212	–1552.106812, –1189.027100, –1516.479858, –1235.234619, –1642.160400 –1497.831909, –2275.606689

Table B.42 (continued)

Index	Codeword
213	–1900.223633, –1642.089966, –1345.290283, –844.898010, –877.631287 –742.815796, –1521.429077
214	–1037.835571, –656.155457, –1014.410461, –1315.275879, –1811.045044 –1302.612793, –2036.575439
215	–1133.035889, –952.475220, –746.429382, –676.345276, –574.932373 –664.157715, –847.545410
216	–277.608704, –261.225098, –345.695862, –430.000031, –623.311157 –1011.414001, –1672.150635
217	–624.576355, –378.763763, –600.394714, 450.444885, –208.235550 –1186.607300, –2060.245605
218	–1379.375122, –1278.502930, –1671.380737, –1521.290283, –2184.201172 1036.537231, 14.498406
219	830.555298, 273.704865, –430.866608, –879.656616, –1596.880859 –1735.605835, –2732.404785
220	–862.293884, –338.837738, –256.978302, 453.634369, 21.184637 –388.895355, –1367.653931
221	–1692.275391, –837.354309, –996.297546, –650.886047, –921.391785 –735.284485, –1599.094971
222	–1439.049561, –403.948120, –264.716797, 34.880165, 133.721527 270.342712, 130.799301
223	–1024.433105, –954.402954, –1288.265259, –1510.096436, –1241.741699 –1407.657593, –2138.514893
224	–451.752625, –810.942505, –1316.806030, –927.019775, –1375.180420 –1263.507812, –2100.465088
225	–1834.794678, –2015.035034, –2055.714844, –1205.373779, –1243.619385 –921.998596, –1739.512573
226	–653.094360, 321.666351, –575.164124, –1054.779297, –423.627563 –749.007141, –649.139587
227	–1019.843872, –932.372925, –1270.553223, –954.567322, –1360.970947 –982.806152, –1841.252563
228	656.237000, 825.421204, 1135.276611, 1071.027466, 868.291321 698.187012, 569.556580
229	–164.890488, 1052.856323, 1300.618042, 1116.814819, 866.946289 658.868591, 540.566589

Index	Codeword
230	406.070679, 248.316833, 124.479065, –133.411728, –337.961334 –940.607117, –1446.928711
231	–2312.506348, –2472.395020, –2646.736816, –2790.495605, –3112.582520 –3451.777344, –701.623169

Table B.42 (continued)

Index	Codeword
232	–1843.943604, 176.875397, –623.110596, –896.002991, –45.831070 140.156006, –0.054655
233	–932.922363, –900.730530, –913.869141, –806.471497, –986.032654 –1149.484009, –1910.793213
234	–133.844696, –514.168762, 390.051361, 641.783997, 484.533081 417.025787, 398.798676
235	–459.463898, –717.394775, –1327.284668, –1522.364502, –1866.756714 –1091.245850, –1919.993530
236	–1220.196655, –560.840759, –629.177979, –229.373505, –581.638794 –572.932739, –1513.422974
237	2318.115723, 2330.241455, 2003.659424, 1627.261719, 1242.322510 922.694458, 694.189758
238	–2558.335205, –3061.049561, –3702.576660, –4377.168945, –5217.957520 –1919.715210, –578.802551
239	–1606.604858, –1942.337036, –2280.985107, –1590.558960, –1840.323120 –1495.487305, –2178.358887
240	47.630093, 198.778152, –192.687698, –91.508957, –656.806152 –685.909973, –1680.821411
241	–1399.591064, –1436.676147, –1885.677856, –1912.264404, –2472.140137 –2558.057861, –3365.300781
242	–1179.167236, –329.757782, –726.824158, –1418.054688, –688.230408 –842.688477, –1137.900513
243	–1112.851807, –863.788086, –1080.140869, –825.756531, –1136.570923 –786.616638, –1648.041382
244	–662.255981, –671.808655, –1132.820679, –1111.957642, –1657.906616 –1473.291992, –2316.884277
245	–58.572517, –21.614277, –155.525177, –194.482086, –169.745590

Index	Codeword
	–534.487610, –758.021912
246	–2220.477051, –1790.342407, –1814.036011, –1498.163818, –1578.185303 –1200.352905, –1899.312378
247	–996.253418, –764.938782, –1122.369385, –1230.562012, –1659.632324 –761.955811, –1652.573120
248	897.895508, 1630.975220, 1602.244385, 1327.910034, 1011.544312 757.971924, 595.457275
249	–1431.946777, –1347.628540, –1499.096924, –1541.816772, –2171.504150 –898.291260, 66.963020
250	–312.534912, –152.314102, –438.497650, –254.288010, –734.177368 –682.732117, –1636.434814

Table B.42 (continued)

Index	Codeword
251	–1506.427002, –1305.687256, –1081.984497, –917.416138, –629.043396 –482.883453, –376.623718
252	–672.532898, –546.122559, –512.605530, –442.461212, –371.147522 –294.251678, –130.954529
253	–616.184143, –447.972778, –481.961853, –102.225723, –291.078766 145.353958, –757.344055
254	383.866699, 442.064697, 416.969147, 467.443085, 480.903381 487.482300, 455.388672
255	–1118.679810, –1169.146851, –1470.901733, –1265.922729, –1537.622925 –955.582031, –1795.824219
256	–2596.727295, –3005.560547, –3863.063232, –4400.398926, –5535.787598 –5088.401367, –931.938293

Table B.43 Stage 2 first sub-vector codebook lsf_stage2_CB1_lbr of low-precision LSF vector quantization codebook

Index	Codeword
1	125.176781, 97.578979, –336.364807, 137.200378, 596.080078
2	–65.084419, 60.324146, 216.620468, 217.536072, –1.865530
3	–271.406830, –56.965103, –308.899261, –197.953033, –211.553497

Index	Codeword
4	–53.075191, 110.288887, –125.435249, –134.959381, 106.253815
5	–208.034668, –478.204865, –54.979763, –170.765671, 56.116074
6	–351.061218, –100.707382, 4.522793, –203.290604, –63.801029
7	–15.212132, –196.821518, –594.479004, –396.369141, 340.373474
8	233.424438, –0.900044, –146.513824, 45.203060, 199.356506
9	–175.750717, –504.816986, –123.821861, 272.267731, 17.651171
10	56.609001, –164.182388, 282.474365, –167.583649, –611.753967
11	–115.305000, –410.567474, –245.159256, 32.249302, 472.335541
12	–395.985168, –152.604843, –179.300735, 98.311966, 40.804863
13	–58.728832, –182.516876, –255.976562, –256.143646, 22.885551
14	316.159882, 151.374481, –191.210144, 7.382407, –124.037529
15	267.551727, –143.205856, –556.498352, 10.446942, –139.945602
16	–59.872601, –218.755066, –135.179184, 17.986374, 126.528053
17	–167.446899, 404.004425, 89.689842, 215.293472, 99.381721
18	23.880964, 436.191742, –66.432892, 294.669983, –230.850113
19	–97.411232, –77.451683, –39.728432, 151.893845, 400.719330
20	–189.941666, –360.634338, 305.063385, –129.827194, –229.681717
21	147.476151, –121.121750, 123.489090, 198.466431, 133.996948
22	168.695541, –282.628235, 57.746059, –110.763184, 256.777191
23	466.524017, 48.549519, 3.054227, –309.880829, 455.355286

Table B.43 (continued)

Index	Codeword
24	–96.707848, –226.300293, –348.300476, 437.114319, –237.053360
25	–159.994415, 266.700043, 385.393860, –41.254078, 155.542038
26	124.722549, 68.716454, 374.974396, –59.638531, –5.237550
27	765.235229, 351.720734, 289.651245, 234.858963, 174.149719
28	241.708282, –199.616089, 212.995590, –351.008759, –104.327087
29	1.665737, –141.787033, –396.943237, 249.341492, 187.570786
30	–58.401596, –344.893921, 223.540710, 343.499237, 307.385132

Index	Codeword
31	–53.936222, 375.296906, –315.383698, 102.585754, 150.957932
32	21.016300, 155.413589, –21.852629, 207.617325, 200.649414
33	10.351093, 92.124763, –347.863464, –212.420975, –146.219879
34	3.016294, 200.229645, 119.715279, –4.462936, 58.314323
35	–63.068295, –357.047424, 646.101807, 171.314423, –429.449341
36	177.134491, 163.830048, 123.773827, –95.167198, –304.159729
37	46.081257, 204.553513, –387.171906, 633.753296, 87.968590
38	–169.514145, 252.454620, –264.691772, –435.691376, 153.001617
39	120.640579, 405.465973, –192.274933, –221.965805, –202.758636
40	100.166435, –400.800293, –40.715820, –107.465210, –118.487724
41	45.016270, 65.268402, 171.456894, 302.890778, –417.200684
42	347.150360, –242.992111, –209.205505, –246.726746, 38.139465
43	194.945053, 351.289307, –99.387756, –158.804657, 213.550888
44	–129.357178, –340.776123, –39.346504, –359.011292, –283.851349
45	31.403564, –107.484909, 242.595520, 21.619331, –228.102585
46	–262.826965, –117.488831, –73.219154, –197.742828, 284.209808
47	–60.193638, –38.268238, 159.151459, –35.138390, 194.001358
48	–317.376831, –240.991592, 144.735809, 76.405945, 161.191895
49	–194.797623, –231.669067, –88.555389, 1.277639, –171.221588
50	–252.915497, 141.291977, –215.599533, 349.899689, 233.891418
51	–216.871674, 19.676451, –397.275146, –56.477779, 232.499542
52	–85.279602, 84.387878, –18.888111, –115.487076, –158.464539
53	245.224274, –426.197693, 303.512543, 63.986469, –74.643738
54	–236.141190, 103.960800, –59.016613, 284.933868, –190.134613
55	34.146214, 243.138809, –90.727913, 53.836540, –75.016983
56	–270.712738, –726.983887, 315.456696, 119.656967, –48.197346
57	–31.662683, 5.897125, 3.455140, 53.694481, –2.451202
58	729.026001, 175.908218, –257.285645, –153.161713, 41.570004
59	–80.404160, 261.090393, 192.832947, 51.316048, –202.484619
60	–158.006607, 145.411636, 565.739990, 175.046204, –235.165543
61	310.425751, –37.598225, 47.160507, 402.864685, 373.127075

Index	Codeword
62	50.817093, –32.576046, –161.691788, –209.549042, 349.112793

Table B.43 (continued)

Index	Codeword
63	–279.980255, 181.749924, 117.845192, –292.787567, 156.011307
64	237.036133, 177.224197, 174.540649, 30.305742, 297.236847
65	358.010071, –172.392334, –15.699339, 31.589804, –312.457031
66	539.975342, 455.723785, 92.737114, –77.091827, –409.111603
67	–45.966713, –147.063065, 34.420399, –154.320145, –20.449684
68	429.404053, –49.871758, 462.804718, 47.656757, –296.449646
69	486.055054, –2.306022, –436.431915, 346.355042, 134.472702
70	–93.373451, –316.287323, –203.803635, 731.280029, 355.487061
71	–309.597534, 214.576782, –197.971817, –42.248268, –97.342346
72	315.710754, 365.675140, –113.759560, 257.418457, 151.386566
73	–54.277527, 575.995667, 116.280495, –313.935638, 326.100006
74	–2.422986, 157.679749, –178.431274, 21.119404, –412.958069
75	265.604034, 144.857193, 114.580612, 191.633926, –72.416084
76	45.641788, 312.367981, 157.248138, –349.953979, –87.191002
77	–65.918320, 150.779114, –131.731110, –531.736023, 662.466064
78	–159.697388, –246.035385, 244.156631, 381.940033, –146.819397
79	230.185669, –366.364471, –172.252045, 174.231308, 92.061836
80	–76.539986, –14.540029, –51.299156, –457.502167, –38.845375
81	–118.276863, –387.540985, –433.180511, –33.551891, –79.331772
82	–61.141945, 508.815582, 467.931915, 527.722839, 420.139313
83	–131.588577, –167.306580, 485.843109, 63.184635, 89.237114
84	94.761421, 97.921158, 179.863037, –344.325256, 229.774002
85	520.248596, –109.993576, 51.674213, 101.569778, 76.834473
86	24.268560, –203.155319, –323.396484, –30.644506, –457.271484
87	–111.438065, –399.305115, –865.804260, 309.888092, 138.058884
88	–82.622826, –54.742317, 254.196701, –139.716690, 566.830811

Index	Codeword
89	29.335989, –128.033279, –40.821259, 172.080994, –200.548294
90	–233.433609, –12.658308, 46.226269, –44.890251, –430.050598
91	40.063583, –70.127205, –63.214340, –248.680176, –306.242096
92	–40.049488, –286.559692, 141.962418, 62.360870, 34.843357
93	189.553772, 74.619942, –83.881447, –261.958984, –27.939610
94	403.209381, 68.722786, –173.985245, –335.583771, –369.518097
95	457.780670, 167.100967, 149.813095, –182.236130, –8.315154
96	203.499512, 111.717392, –456.987671, –102.090157, 163.226791
97	–101.386177, –25.736250, –280.759399, 72.861252, –55.210712
98	–103.929649, –491.284943, 291.325806, –111.859535, 334.475250
99	–144.490921, 281.077942, –23.609737, –27.741974, 417.479828
100	264.139496, 186.067474, –222.433640, –646.080383, 103.886513
101	377.383392, 131.654785, –307.225342, 329.011566, –396.618927

Table B.43 (continued)

Index	Codeword
102	–173.476822, 472.666779, –5.357996, –132.365311, –48.892284
103	245.283829, –84.957413, 457.927094, 321.619659, 42.520683
104	–131.979996, –138.226730, –30.602535, 320.700684, 101.862259
105	–6.157870, 97.928215, –209.740662, –431.855286, –817.070190
106	–3.690514, –179.861206, –476.954803, –632.629822, –250.939194
107	–66.380966, 170.722198, –530.042786, 177.100327, –181.131012
108	–128.159744, 55.080872, 308.354065, –254.569229, –221.356033
109	–201.901703, 242.691742, –86.607887, –396.883148, –314.213501
110	205.063766, 462.954620, 155.144760, –16.457201, –29.848045
111	327.753082, –196.080566, 433.168854, –82.308937, 279.670837
112	–25.943148, 122.535789, 106.418846, 524.088684, 64.240387
113	126.005486, 57.947956, –185.083878, 293.347504, –32.417767
114	204.149628, –38.046509, 58.536194, –31.194141, –16.386127
115	–86.873398, –414.818848, 21.891113, 142.845123, –476.217682

Index	Codeword
116	117.195274, –132.432159, –209.501282, –48.684799, –105.391930
117	–282.344055, –42.621487, 202.447464, 49.988632, –101.511032
118	–149.238113, –321.226227, 742.572449, 467.619019, 257.926819
119	–55.879326, 65.658493, 287.207520, 217.585556, 307.118652
120	243.938828, –203.829315, 10.253199, 480.966309, –107.561493
121	–268.738800, 92.606926, 8.622302, 64.133263, 114.917183
122	–90.888931, 497.001007, 161.134674, –130.725739, –558.346558
123	–467.650482, 288.785370, 160.522598, –0.405135, –159.286697
124	–399.703735, 89.198013, 259.508423, 295.884003, 167.580093
125	–146.541641, –195.395798, 298.586151, –316.311310, 100.641792
126	–148.101120, 743.981140, 366.386505, 121.370277, –161.657700
127	134.723831, 320.460175, 349.293152, 238.967743, 33.517555
128	–35.735043, –292.962158, –26.472088, –518.628357, 271.610901

Table B.44 Stage 2 second sub-vector codebook lsf_stage2_CB2_lbr of low-precision LSF vector quantization codebook

Index	Codeword
1	–243.409409, –116.480858, 389.801025, 181.432846
2	–696.340149, –287.063507, 446.367493, 236.489014
3	–393.174835, 117.744064, –62.679790, –412.519104
4	–74.848740, 27.573660, 285.320068, –59.181179
5	116.752983, –1.356940, –109.929138, 113.496178
6	–225.366104, –189.137146, –44.427967, –286.845917
7	242.165924, 287.599548, –181.383621, –330.778229
8	–179.347458, 66.509262, 109.463470, 87.632050

Table B.44 (continued)

Index	Codeword
9	–253.711777, 323.794830, 338.500244, –337.053680
10	160.067596, –163.665115, 6.405066, –264.836487
11	–60.688374, –185.087067, –191.619583, –197.485458

Index	Codeword
12	21.429399, –370.080780, –42.411572, –108.452370
13	–52.254684, –23.956074, –313.852173, –7.787478
14	12.098438, 230.482193, –478.094971, –100.934822
15	–37.458061, 288.877838, 10.201681, 475.521515
16	2.881128, 343.488373, 450.687256, –47.208767
17	–83.777214, 151.401154, 322.103760, 253.128464
18	–316.890259, 28.818686, –616.391968, 192.355240
19	511.084839, –133.957230, 361.529755, 198.506775
20	72.080055, 199.831650, –108.111671, –84.174294
21	–98.012787, –39.761772, –68.415688, 178.787384
22	–265.631104, 164.766037, –440.029663, 823.557129
23	–168.251541, 702.906860, 207.797424, –79.284523
24	305.986115, 272.766815, 292.596832, –235.731766
25	–788.239197, 239.276917, 113.054337, –212.844467
26	258.902313, –333.383759, –243.586655, –307.619171
27	–121.466507, 174.642380, –293.016357, 272.726562
28	39.102711, –165.410400, –337.241791, 313.644287
29	–10.405048, –185.301270, –512.177002, –270.207581
30	101.790581, 11.252160, –62.398804, 392.750671
31	–239.751587, –471.905090, –310.683197, –300.936493
32	–155.268570, 119.035522, 79.837418, –192.250656
33	131.482040, 86.621811, 93.051872, –150.859161
34	200.261765, 223.326935, –433.248749, 433.448395
35	–94.155045, –43.422302, 131.647949, 330.752167
36	311.755371, 50.656456, 49.977379, –428.572540
37	–312.415009, –121.180763, –232.275406, 138.695938
38	209.342743, 687.162781, 184.681885, –401.803741
39	114.690254, 171.625977, 181.704712, 260.692871
40	–381.737579, –85.716354, 179.320496, –116.409515
41	90.773277, –43.356819, 633.900757, 6.079812
42	671.235840, 336.139923, –112.772232, –600.057617

Index	Codeword
43	371.775330, –1.344244, 163.382965, –95.445984
44	–193.736588, –71.884460, –77.847572, –49.722527
45	–141.227814, –446.522247, 236.406113, 184.287979
46	–268.798889, 352.709106, 39.056961, –68.754677
47	–221.551651, –119.258545, –121.718758, 479.259888

Table B.44 (continued)

Index	Codeword
48	676.735718, 194.002365, –362.131866, 175.913193
49	50.821251, –591.166016, 616.756592, –97.901886
50	382.693909, –181.652420, –680.908813, 676.854553
51	–80.215393, –282.486908, –509.045990, –791.799927
52	53.088028, 499.772308, –195.334656, 77.658592
53	17.863188, 292.266754, 71.577980, –618.104736
54	120.337517, –7.530244, –264.265137, –287.812256
55	–394.066956, 219.687286, 295.375671, 72.552650
56	179.996841, –192.677536, –261.940826, –25.612436
57	58.660305, 336.429840, 100.187775, –220.980881
58	42.412331, 13.519630, 112.433083, 138.155304
59	266.368958, 21.970327, –107.287674, –144.994812
60	78.719635, –430.507874, 168.567307, –456.712982
61	–411.224457, 74.529732, –57.504768, –21.765814
62	196.090912, 137.380432, 57.631134, 50.901695
63	–123.902313, –284.127167, 288.213501, –152.852875
64	–302.251831, –356.747955, 204.819641, 551.967896
65	–393.012970, –43.004662, 636.058228, –192.525909
66	–127.849014, 434.849792, –159.721878, –301.210571
67	21.238861, –452.225281, –153.344681, 734.341736
68	10.870827, 185.948196, –6.938921, 175.038879
69	–432.968109, –610.992554, –137.377045, 243.536591

Index	Codeword
70	180.154404, 423.220062, 163.796448, 72.456429
71	–452.403168, 175.101013, 299.354309, 512.337463
72	38.909637, –7.877390, 262.278534, 613.609680
73	–47.382046, –109.140800, 42.966789, –158.798386
74	–425.121552, –512.733276, 249.457809, –226.062881
75	182.071060, –206.442108, –688.257446, 108.594162
76	2.062111, 224.044220, 155.120468, 19.599276
77	38.487358, –31.999537, –109.859627, –119.361366
78	162.551880, 67.492264, 290.018524, 45.786263
79	436.423676, –43.198994, –77.049904, 569.098450
80	363.950043, 281.895386, –25.208874, –89.876564
81	–248.373856, –321.788391, –56.249428, –39.337593
82	425.436371, –16.661833, –59.625412, 116.543587
83	–82.381096, –163.117569, 114.577087, 70.018883
84	253.342102, –523.466125, 201.246658, 378.887482
85	–131.377869, 66.285393, –167.358292, –217.760086
86	–566.463745, –226.344803, –157.164017, –86.413437

Table B.44 (continued)

Index	Codeword
87	390.181671, –371.196930, 13.734019, –36.191559
88	446.689575, –241.156555, –321.963715, 233.878906
89	77.666321, –131.525879, 320.331818, 256.978943
90	367.632355, 216.511093, 243.643234, 123.523415
91	–657.771362, 41.172512, –83.689575, 283.212250
92	–460.604584, 638.931152, 35.756577, –559.100403
93	–226.586746, –80.584587, 233.980896, –493.453033
94	417.722504, –1.672927, –464.180389, –184.742569
95	167.272629, –281.216431, 276.785431, –22.157265
96	–284.334778, –108.795586, –299.789734, –209.318359

Index	Codeword
97	–132.045181, 395.118500, 158.724670, 211.790054
98	301.987091, –0.899439, –280.349152, –661.502075
99	578.751221, 626.476074, 518.695251, 378.556610
100	257.378601, –25.404896, 135.438293, 242.261215
101	132.840515, –251.884521, –2.454656, 241.139893
102	22.451145, 63.981472, –25.862547, –355.621521
103	–132.091568, –329.954651, –77.058594, 239.968750
104	–34.870598, –171.217773, –174.181763, –465.647217
105	449.525177, –237.710693, 438.640564, –306.627014
106	–20.192402, –701.768494, 79.301132, –45.662354
107	–29.979303, –361.198029, 703.003723, 443.174042
108	591.450317, –28.069363, –93.527534, –223.932785
109	–472.069946, –989.679260, 528.279358, 340.102905
110	114.742538, –549.570740, –281.955597, 145.624893
111	–292.186951, 138.453705, 13.866644, 271.594849
112	307.743439, 269.438049, –34.875519, 267.321228
113	–33.110497, –172.138855, –109.525803, 26.296240
114	246.703857, 123.510437, –305.353668, 83.845627
115	–102.935699, 587.943848, 636.555054, 524.072510
116	–145.242615, –315.349030, –352.720276, 21.209959
117	142.747467, –102.298622, 50.486584, –9.182832
118	234.382019, 261.100342, 365.964935, 399.349396
119	684.732361, 396.310455, 181.284805, –108.301056
120	–241.690918, –467.369324, –675.536865, 491.443207
121	–100.938423, 147.491653, –313.448547, –511.904236
122	–698.902588, 653.364136, 431.230652, –23.328299
123	–15.995284, 31.354193, 3.822088, –5.892229
124	–432.100098, 260.371490, –357.567444, –122.324028
125	–354.313080, –164.696030, 75.725403, 181.991409

Table B.44 (continued)

Index	Codeword
126	–447.009247, 496.912720, –161.698349, 245.110306
127	77.673477, –7.057706, 335.725311, –319.795929
128	–162.106491, 166.707016, –136.834305, 26.008549

Table B.45 Stage 2 third sub-vector codebook lsf_stage2_CB3_lbr of low-precision LSF vector quantization codebook

Index	Codeword
1	–120.858948, –66.756027, 54.656181, 64.874451, 124.259438 20.630352, 458.284790
2	54.381855, 279.556610, –216.506119, –67.174606, 1.014068 17.655251, 6.228757
3	–28.685160, 154.050797, –48.445076, –277.829315, 285.749939 195.247452, 132.268265
4	–141.355896, –64.183372, –85.978325, 75.814301, 73.956757 249.112961, 100.836151
5	–395.490082, 105.968681, –137.138885, –185.565430, –4.196878 9.040785, 9.331719
6	297.102386, –264.661957, –128.039383, –27.582905, –86.045662 –90.526764, –60.119183
7	174.266235, –103.852974, –37.978733, 31.690355, 302.096497 135.048508, 114.836746
8	–101.046234, –23.895374, –147.228592, 31.045322, –344.268677 149.099213, –37.257492
9	–61.336628, 37.914291, 10.385083, 105.648895, 6.490855 109.133385, –364.720398
10	–57.522472, –73.206619, –200.797119, –141.121109, –115.150620 650.165649, 423.394958
11	–73.722351, –140.067551, 140.425919, –391.595581, –5.604278 –10.986897, –15.726768
12	–108.406639, –6.762465, –37.700020, –6.168937, –11.448340 0.249742, 0.123609
13	152.156693, 28.136551, –168.495224, 65.982376, –84.628181 –83.325134, –141.856110

Index	Codeword
14	486.470673, 77.363564, –187.013657, 136.576614, 96.876122 25.852430, 7.105801
15	–983.513672, 399.680298, 283.328339, 145.344055, 54.366695 –53.572399, 39.212997
16	–220.661285, –47.986832, 179.123917, 401.111053, 340.343872 159.414078, 69.568336

Table B.45 (continued)

Index	Codeword
17	–105.138153, 51.941540, 56.188911, –144.363174, –119.869408 141.778366, 128.125656
18	–263.076965, 254.405991, 333.602570, –42.307457, –451.265350 –73.605690, 22.996130
19	–29.157436, 69.872826, 346.867462, –87.169640, 26.287106 –3.849262, 4.497209
20	–24.435114, –195.265182, –99.388878, 206.267960, 83.080887 –0.377817, –45.230667
21	–187.647476, 181.213882, –273.944153, 255.762146, –67.340736 23.068657, 10.287804
22	–172.619736, –140.707123, –190.551239, –6.177309, –41.383221 –111.614410, –169.822540
23	27.315220, –127.491776, –76.529587, –380.704071, 712.483704 168.066162, –282.497314
24	144.456070, 74.449486, 73.299232, –89.281288, –230.136108 –50.344418, –61.720192
25	71.039864, 16.484489, –81.132729, –146.860245, –239.453995 –276.474915, –495.753235
26	–6.601482, –6.596558, –135.847580, –200.303955, –149.754517 –46.271084, –42.693810
27	114.541161, 96.925179, –62.422073, 267.284790, 92.719643 138.007202, 0.382021
28	39.763325, –115.202148, –418.437439, –42.130184, 56.940830 87.226730, 38.288715
29	21.468893, –166.728149, –67.645279, –66.803993, –17.396999

Index	Codeword
	36.986813, 13.436744
30	4.862719, –45.827774, –68.863770, –164.614624, –305.732239 –554.339478, 19.752028
31	6.885021, –95.595200, –104.018166, –184.963974, 31.702078 –84.178284, 256.128448
32	–192.003845, –394.069550, –488.481445, –438.397797, –190.934586 –44.327446, –26.086609
33	–125.266785, 137.131226, 39.229233, –10.948144, –105.625397 –123.229942, –139.748444
34	184.016785, 264.055878, 123.471939, –34.399540, –232.265396 397.860321, 90.333214
35	80.903946, –43.319683, 180.592911, –59.142773, 138.443069 –390.676544, 205.455032

Table B.45 (continued)

Index	Codeword
36	–132.122040, –139.576950, 106.337486, –21.377464, 174.604553 16.855715, 61.933849
37	88.327904, –498.666840, 137.356384, 33.944885, 67.351425 20.773945, 24.755754
38	–345.485718, –27.660803, 67.980339, 83.968163, 23.501289 –11.475863, –47.268600
39	178.483002, –24.934628, –96.635849, –13.439441, –88.224274 176.098663, 124.763161
40	–26.483379, 21.979994, –20.658539, 330.616333, –43.313259 –384.421143, 48.723835
41	62.266487, –44.714783, –115.789307, –188.385208, 183.125153 –185.326904, –155.693787
42	53.820332, 81.532997, 98.294876, 40.696548, 98.454292 119.502266, 147.300720
43	254.267975, 70.164246, 47.383190, –142.257462, 37.144730 –19.954918, 38.457348
44	–82.691055, –20.630274, 185.650543, 180.139053, –90.897530 109.889763, 58.255798

Index	Codeword
45	244.058670, –122.199280, 189.862457, 132.944275, 46.770733 22.384722, –2.599370
46	–236.754898, –243.830826, –54.952366, –73.321976, –94.426300 –12.912512, 69.587036
47	91.892441, –262.573792, 314.933350, –58.419174, –324.563995 88.949524, 50.423519
48	417.443909, 69.775948, –251.358078, –512.123352, 4.932464 –0.540782, 29.553900
49	–150.129364, –145.058929, 114.207970, 211.513779, 194.664597 –268.673645, –373.467438
50	66.813667, 16.556433, 26.886477, 15.414439, 15.461973 –7.944495, –0.863844
51	302.242676, 324.145844, 329.172577, 244.085281, 176.245163 87.874321, 56.691006
52	–36.683731, –139.869324, 80.175735, –27.114138, –77.579376 –157.658203, –124.447487
53	108.132103, 97.285461, 7.401071, –2.834207, –210.823380 –171.585175, 446.059509
54	–244.690735, –307.984222, 598.741272, 230.457870, –28.186348 –101.380302, –8.740450

Table B.45 (continued)

Index	Codeword
55	–70.319702, 127.248039, –167.226410, 85.655022, 303.577606 –59.247124, –66.811310
56	–215.468704, 348.959717, 98.189171, 90.742363, 79.830116 84.431984, 36.955971
57	–36.783245, 69.040504, 133.008469, 157.655502, 128.536774 –34.705322, –65.879082
58	–73.992271, 343.438080, 48.778431, –304.844391, 57.568405 –77.251152, –63.522530
59	200.285736, 120.431190, 115.817978, 93.692749, 40.653152 –164.645035, –240.228683
60	110.406593, 221.153152, 92.144363, 56.062359, –36.240662

Index	Codeword
	27.389029, –19.072336
61	–588.968384, –505.493042, –121.743179, 46.453411, 157.052521 44.954498, 34.388012
62	526.978088, 425.036041, 189.527222, –62.570236, –241.602249 –253.744858, –42.391441
63	239.768356, 44.924934, –128.932877, –433.974670, –694.861633 37.230545, 13.897585
64	32.482452, –241.068497, –579.445068, 523.904236, 267.373016 4.412733, –53.777256

Table B.46 LSF parameter mean value vector table

Index	Index value
1	816.510986
2	2231.826660
3	3647.142334
4	5062.458008
5	6477.773682
6	7893.089355
7	9308.405029
8	10723.720703
9	12139.036377
10	13554.352051
11	14969.667725
12	16384.983398
13	17800.299072
14	19215.614746
15	20630.930420
16	22046.246094

Table B.47 Frequency-domain noise shaping sub-band division table

Start/end point of sub-band division
0, 4, 8, 12, 16, 20, 24, 28, 32, 36,
40, 48, 56, 64, 72, 80, 88, 96, 108, 120,
132, 144, 160, 176, 196, 216, 240, 264, 292, 320,
352, 384, 416, 448, 480, 512, 544, 576, 608, 640,
672, 704, 736, 768, 800, 832, 864, 896, 928, 1024

Table B.48 Table of preset values of virtual speakers

Index	Index value	Index	Index value
0	2, 768	672	372, 35
1	2, 791	673	384, 35
2	146, 791	674	396, 35
3	293, 791	675	407, 35
4	439, 791	676	419, 35
5	585, 791	677	431, 35
6	731, 791	678	442, 35
7	878, 791	679	454, 35
8	2, 815	680	465, 35
9	79, 815	681	477, 35
10	158, 815	682	489, 35
11	236, 815	683	500, 35
12	315, 815	684	512, 35
13	394, 815	685	524, 35
14	473, 815	686	535, 35
15	551, 815	687	547, 35
16	630, 815	688	559, 35
17	709, 815	689	570, 35
18	788, 815	690	582, 35
19	866, 815	691	593, 35
20	945, 815	692	605, 35
21	2, 838	693	617, 35

Index	Index value	Index	Index value
22	54, 838	694	628, 35
23	108, 838	695	640, 35
24	162, 838	696	652, 35
25	216, 838	697	663, 35
26	269, 838	698	675, 35
27	323, 838	699	687, 35
28	377, 838	700	698, 35
29	431, 838	701	710, 35

Table B.48 (continued)

Index	Index value	Index	Index value
30	485, 838	702	721, 35
31	539, 838	703	733, 35
32	593, 838	704	745, 35
33	647, 838	705	756, 35
34	701, 838	706	768, 35
35	755, 838	707	780, 35
36	808, 838	708	791, 35
37	862, 838	709	803, 35
38	916, 838	710	815, 35
39	970, 838	711	826, 35
40	2, 861	712	838, 35
41	41, 861	713	849, 35
42	82, 861	714	861, 35
43	123, 861	715	873, 35
44	164, 861	716	884, 35
45	205, 861	717	896, 35
46	246, 861	718	908, 35
47	287, 861	719	919, 35
48	328, 861	720	931, 35

Index	Index value	Index	Index value
49	369, 861	721	943, 35
50	410, 861	722	954, 35
51	451, 861	723	966, 35
52	492, 861	724	977, 35
53	532, 861	725	989, 35
54	573, 861	726	1001, 35
55	614, 861	727	1012, 35
56	655, 861	728	2, 47
57	696, 861	729	12, 47
58	737, 861	730	24, 47
59	778, 861	731	35, 47
60	819, 861	732	47, 47
61	860, 861	733	59, 47
62	901, 861	734	71, 47
63	942, 861	735	82, 47
64	983, 861	736	94, 47
65	2, 884	737	106, 47
66	34, 884	738	118, 47
67	68, 884	739	129, 47
68	102, 884	740	141, 47

Table B.48 (continued)

Index	Index value	Index	Index value
69	137, 884	741	153, 47
70	171, 884	742	165, 47
71	205, 884	743	177, 47
72	239, 884	744	188, 47
73	273, 884	745	200, 47
74	307, 884	746	212, 47
75	341, 884	747	224, 47

Index	Index value	Index	Index value
76	375, 884	748	235, 47
77	410, 884	749	247, 47
78	444, 884	750	259, 47
79	478, 884	751	271, 47
80	512, 884	752	282, 47
81	546, 884	753	294, 47
82	580, 884	754	306, 47
83	614, 884	755	318, 47
84	649, 884	756	330, 47
85	683, 884	757	341, 47
86	717, 884	758	353, 47
87	751, 884	759	365, 47
88	785, 884	760	377, 47
89	819, 884	761	388, 47
90	853, 884	762	400, 47
91	887, 884	763	412, 47
92	922, 884	764	424, 47
93	956, 884	765	435, 47
94	990, 884	766	447, 47
95	2, 908	767	459, 47
96	29, 908	768	471, 47
97	59, 908	769	483, 47
98	88, 908	770	494, 47
99	117, 908	771	506, 47
100	146, 908	772	518, 47
101	176, 908	773	530, 47
102	205, 908	774	541, 47
103	234, 908	775	553, 47
104	263, 908	776	565, 47
105	293, 908	777	577, 47
106	322, 908	778	589, 47

Index	Index value	Index	Index value
107	351, 908	779	600, 47

Table B.48 (continued)

Index	Index value	Index	Index value
108	380, 908	780	612, 47
109	410, 908	781	624, 47
110	439, 908	782	636, 47
111	468, 908	783	647, 47
112	497, 908	784	659, 47
113	527, 908	785	671, 47
114	556, 908	786	683, 47
115	585, 908	787	694, 47
116	614, 908	788	706, 47
117	644, 908	789	718, 47
118	673, 908	790	730, 47
119	702, 908	791	742, 47
120	731, 908	792	753, 47
121	761, 908	793	765, 47
122	790, 908	794	777, 47
123	819, 908	795	789, 47
124	848, 908	796	800, 47
125	878, 908	797	812, 47
126	907, 908	798	824, 47
127	936, 908	799	836, 47
128	965, 908	800	847, 47
129	995, 908	801	859, 47
130	2, 931	802	871, 47
131	27, 931	803	883, 47
132	54, 931	804	895, 47
133	81, 931	805	906, 47

Index	Index value	Index	Index value
134	108, 931	806	918, 47
135	135, 931	807	930, 47
136	162, 931	808	942, 47
137	189, 931	809	953, 47
138	216, 931	810	965, 47
139	243, 931	811	977, 47
140	269, 931	812	989, 47
141	296, 931	813	1000, 47
142	323, 931	814	1012, 47
143	350, 931	815	2, 58
144	377, 931	816	12, 58
145	404, 931	817	24, 58
146	431, 931	818	36, 58

Table B.48 (continued)

Index	Index value	Index	Index value
147	458, 931	819	48, 58
148	485, 931	820	60, 58
149	512, 931	821	72, 58
150	539, 931	822	84, 58
151	566, 931	823	96, 58
152	593, 931	824	108, 58
153	620, 931	825	120, 58
154	647, 931	826	133, 58
155	674, 931	827	145, 58
156	701, 931	828	157, 58
157	728, 931	829	169, 58
158	755, 931	830	181, 58
159	781, 931	831	193, 58
160	808, 931	832	205, 58

Index	Index value	Index	Index value
161	835, 931	833	217, 58
162	862, 931	834	229, 58
163	889, 931	835	241, 58
164	916, 931	836	253, 58
165	943, 931	837	265, 58
166	970, 931	838	277, 58
167	997, 931	839	289, 58
168	2, 954	840	301, 58
169	25, 954	841	313, 58
170	50, 954	842	325, 58
171	75, 954	843	337, 58
172	100, 954	844	349, 58
173	125, 954	845	361, 58
174	150, 954	846	373, 58
175	175, 954	847	386, 58
176	200, 954	848	398, 58
177	225, 954	849	410, 58
178	250, 954	850	422, 58
179	275, 954	851	434, 58
180	300, 954	852	446, 58
181	325, 954	853	458, 58
182	350, 954	854	470, 58
183	375, 954	855	482, 58
184	400, 954	856	494, 58
185	425, 954	857	506, 58

Table B.48 (continued)

Index	Index value	Index	Index value
186	450, 954	858	518, 58
187	475, 954	859	530, 58

Index	Index value	Index	Index value
188	500, 954	860	542, 58
189	524, 954	861	554, 58
190	549, 954	862	566, 58
191	574, 954	863	578, 58
192	599, 954	864	590, 58
193	624, 954	865	602, 58
194	649, 954	866	614, 58
195	674, 954	867	626, 58
196	699, 954	868	638, 58
197	724, 954	869	651, 58
198	749, 954	870	663, 58
199	774, 954	871	675, 58
200	799, 954	872	687, 58
201	824, 954	873	699, 58
202	849, 954	874	711, 58
203	874, 954	875	723, 58
204	899, 954	876	735, 58
205	924, 954	877	747, 58
206	949, 954	878	759, 58
207	974, 954	879	771, 58
208	999, 954	880	783, 58
209	2, 977	881	795, 58
210	23, 977	882	807, 58
211	47, 977	883	819, 58
212	70, 977	884	831, 58
213	93, 977	885	843, 58
214	116, 977	886	855, 58
215	140, 977	887	867, 58
216	163, 977	888	879, 58
217	186, 977	889	891, 58
218	209, 977	890	904, 58

Index	Index value	Index	Index value
219	233, 977	891	916, 58
220	256, 977	892	928, 58
221	279, 977	893	940, 58
222	303, 977	894	952, 58
223	326, 977	895	964, 58
224	349, 977	896	976, 58

Table B.48 (continued)

Index	Index value	Index	Index value
225	372, 977	897	988, 58
226	396, 977	898	1000, 58
227	419, 977	899	1012, 58
228	442, 977	900	2, 70
229	465, 977	901	12, 70
230	489, 977	902	25, 70
231	512, 977	903	37, 70
232	535, 977	904	50, 70
233	559, 977	905	62, 70
234	582, 977	906	75, 70
235	605, 977	907	87, 70
236	628, 977	908	100, 70
237	652, 977	909	112, 70
238	675, 977	910	125, 70
239	698, 977	911	137, 70
240	721, 977	912	150, 70
241	745, 977	913	162, 70
242	768, 977	914	175, 70
243	791, 977	915	187, 70
244	815, 977	916	200, 70
245	838, 977	917	212, 70

Index	Index value	Index	Index value
246	861, 977	918	225, 70
247	884, 977	919	237, 70
248	908, 977	920	250, 70
249	931, 977	921	262, 70
250	954, 977	922	275, 70
251	977, 977	923	287, 70
252	1001, 977	924	300, 70
253	2, 1001	925	312, 70
254	23, 1001	926	325, 70
255	46, 1001	927	337, 70
256	68, 1001	928	350, 70
257	91, 1001	929	362, 70
258	114, 1001	930	375, 70
259	137, 1001	931	387, 70
260	159, 1001	932	400, 70
261	182, 1001	933	412, 70
262	205, 1001	934	425, 70
263	228, 1001	935	437, 70

Table B.48 (continued)

Index	Index value	Index	Index value
264	250, 1001	936	450, 70
265	273, 1001	937	462, 70
266	296, 1001	938	475, 70
267	319, 1001	939	487, 70
268	341, 1001	940	500, 70
269	364, 1001	941	512, 70
270	387, 1001	942	524, 70
271	410, 1001	943	537, 70
272	432, 1001	944	549, 70

Index	Index value	Index	Index value
273	455, 1001	945	562, 70
274	478, 1001	946	574, 70
275	501, 1001	947	587, 70
276	523, 1001	948	599, 70
277	546, 1001	949	612, 70
278	569, 1001	950	624, 70
279	592, 1001	951	637, 70
280	614, 1001	952	649, 70
281	637, 1001	953	662, 70
282	660, 1001	954	674, 70
283	683, 1001	955	687, 70
284	705, 1001	956	699, 70
285	728, 1001	957	712, 70
286	751, 1001	958	724, 70
287	774, 1001	959	737, 70
288	796, 1001	960	749, 70
289	819, 1001	961	762, 70
290	842, 1001	962	774, 70
291	865, 1001	963	787, 70
292	887, 1001	964	799, 70
293	910, 1001	965	812, 70
294	933, 1001	966	824, 70
295	956, 1001	967	837, 70
296	978, 1001	968	849, 70
297	1001, 1001	969	862, 70
298	2, 256	970	874, 70
299	2, 230	971	887, 70
300	128, 230	972	899, 70
301	256, 230	973	912, 70
302	384, 230	974	924, 70

Table B.48 (continued)

Index	Index value	Index	Index value
303	512, 230	975	937, 70
304	640, 230	976	949, 70
305	768, 230	977	962, 70
306	896, 230	978	974, 70
307	2, 205	979	987, 70
308	73, 205	980	999, 70
309	146, 205	981	1012, 70
310	219, 205	982	2, 81
311	293, 205	983	13, 81
312	366, 205	984	26, 81
313	439, 205	985	39, 81
314	512, 205	986	52, 81
315	585, 205	987	65, 81
316	658, 205	988	78, 81
317	731, 205	989	91, 81
318	805, 205	990	104, 81
319	878, 205	991	117, 81
320	951, 205	992	130, 81
321	2, 179	993	143, 81
322	49, 179	994	156, 81
323	98, 179	995	169, 81
324	146, 179	996	181, 81
325	195, 179	997	194, 81
326	244, 179	998	207, 81
327	293, 179	999	220, 81
328	341, 179	1000	233, 81
329	390, 179	1001	246, 81
330	439, 179	1002	259, 81
331	488, 179	1003	272, 81
332	536, 179	1004	285, 81

Index	Index value	Index	Index value
333	585, 179	1005	298, 81
334	634, 179	1006	311, 81
335	683, 179	1007	324, 81
336	731, 179	1008	337, 81
337	780, 179	1009	350, 81
338	829, 179	1010	363, 81
339	878, 179	1011	376, 81
340	926, 179	1012	389, 81
341	975, 179	1013	402, 81

Table B.48 (continued)

Index	Index value	Index	Index value
342	2, 154	1014	415, 81
343	38, 154	1015	428, 81
344	76, 154	1016	441, 81
345	114, 154	1017	454, 81
346	152, 154	1018	467, 81
347	190, 154	1019	480, 81
348	228, 154	1020	493, 81
349	265, 154	1021	506, 81
350	303, 154	1022	518, 81
351	341, 154	1023	531, 81
352	379, 154	1024	544, 81
353	417, 154	1025	557, 81
354	455, 154	1026	570, 81
355	493, 154	1027	583, 81
356	531, 154	1028	596, 81
357	569, 154	1029	609, 81
358	607, 154	1030	622, 81
359	645, 154	1031	635, 81

Index	Index value	Index	Index value
360	683, 154	1032	648, 81
361	721, 154	1033	661, 81
362	759, 154	1034	674, 81
363	796, 154	1035	687, 81
364	834, 154	1036	700, 81
365	872, 154	1037	713, 81
366	910, 154	1038	726, 81
367	948, 154	1039	739, 81
368	986, 154	1040	752, 81
369	2, 2	1041	765, 81
370	11, 2	1042	778, 81
371	23, 2	1043	791, 81
372	34, 2	1044	804, 81
373	45, 2	1045	817, 81
374	56, 2	1046	830, 81
375	68, 2	1047	843, 81
376	79, 2	1048	855, 81
377	90, 2	1049	868, 81
378	101, 2	1050	881, 81
379	113, 2	1051	894, 81
380	124, 2	1052	907, 81

Table B.48 (continued)

Index	Index value	Index	Index value
381	135, 2	1053	920, 81
382	146, 2	1054	933, 81
383	158, 2	1055	946, 81
384	169, 2	1056	959, 81
385	180, 2	1057	972, 81
386	191, 2	1058	985, 81

Index	Index value	Index	Index value
387	203, 2	1059	998, 81
388	214, 2	1060	1011, 81
389	225, 2	1061	2, 93
390	236, 2	1062	13, 93
391	248, 2	1063	27, 93
392	259, 2	1064	40, 93
393	270, 2	1065	54, 93
394	281, 2	1066	67, 93
395	293, 2	1067	81, 93
396	304, 2	1068	94, 93
397	315, 2	1069	108, 93
398	326, 2	1070	121, 93
399	338, 2	1071	135, 93
400	349, 2	1072	148, 93
401	360, 2	1073	162, 93
402	371, 2	1074	175, 93
403	383, 2	1075	189, 93
404	394, 2	1076	202, 93
405	405, 2	1077	216, 93
406	416, 2	1078	229, 93
407	428, 2	1079	243, 93
408	439, 2	1080	256, 93
409	450, 2	1081	269, 93
410	461, 2	1082	283, 93
411	473, 2	1083	296, 93
412	484, 2	1084	310, 93
413	495, 2	1085	323, 93
414	506, 2	1086	337, 93
415	518, 2	1087	350, 93
416	529, 2	1088	364, 93
417	540, 2	1089	377, 93

Index	Index value	Index	Index value
418	551, 2	1090	391, 93
419	563, 2	1091	404, 93

Table B.48 (continued)

Index	Index value	Index	Index value
420	574, 2	1092	418, 93
421	585, 2	1093	431, 93
422	596, 2	1094	445, 93
423	608, 2	1095	458, 93
424	619, 2	1096	472, 93
425	630, 2	1097	485, 93
426	641, 2	1098	499, 93
427	653, 2	1099	512, 93
428	664, 2	1100	525, 93
429	675, 2	1101	539, 93
430	686, 2	1102	552, 93
431	698, 2	1103	566, 93
432	709, 2	1104	579, 93
433	720, 2	1105	593, 93
434	731, 2	1106	606, 93
435	743, 2	1107	620, 93
436	754, 2	1108	633, 93
437	765, 2	1109	647, 93
438	776, 2	1110	660, 93
439	788, 2	1111	674, 93
440	799, 2	1112	687, 93
441	810, 2	1113	701, 93
442	821, 2	1114	714, 93
443	833, 2	1115	728, 93
444	844, 2	1116	741, 93

Index	Index value	Index	Index value
445	855, 2	1117	755, 93
446	866, 2	1118	768, 93
447	878, 2	1119	781, 93
448	889, 2	1120	795, 93
449	900, 2	1121	808, 93
450	911, 2	1122	822, 93
451	923, 2	1123	835, 93
452	934, 2	1124	849, 93
453	945, 2	1125	862, 93
454	956, 2	1126	876, 93
455	968, 2	1127	889, 93
456	979, 2	1128	903, 93
457	990, 2	1129	916, 93
458	1001, 2	1130	930, 93

Table B.48 (continued)

Index	Index value	Index	Index value
459	1013, 2	1131	943, 93
460	2, 12	1132	957, 93
461	11, 12	1133	970, 93
462	23, 12	1134	984, 93
463	34, 12	1135	997, 93
464	46, 12	1136	1011, 93
465	57, 12	1137	2, 105
466	68, 12	1138	14, 105
467	80, 12	1139	28, 105
468	91, 12	1140	42, 105
469	102, 12	1141	56, 105
470	114, 12	1142	70, 105
471	125, 12	1143	84, 105

Index	Index value	Index	Index value
472	137, 12	1144	98, 105
473	148, 12	1145	112, 105
474	159, 12	1146	126, 105
475	171, 12	1147	140, 105
476	182, 12	1148	154, 105
477	193, 12	1149	168, 105
478	205, 12	1150	182, 105
479	216, 12	1151	196, 105
480	228, 12	1152	210, 105
481	239, 12	1153	224, 105
482	250, 12	1154	238, 105
483	262, 12	1155	252, 105
484	273, 12	1156	267, 105
485	284, 12	1157	281, 105
486	296, 12	1158	295, 105
487	307, 12	1159	309, 105
488	319, 12	1160	323, 105
489	330, 12	1161	337, 105
490	341, 12	1162	351, 105
491	353, 12	1163	365, 105
492	364, 12	1164	379, 105
493	375, 12	1165	393, 105
494	387, 12	1166	407, 105
495	398, 12	1167	421, 105
496	410, 12	1168	435, 105
497	421, 12	1169	449, 105

Table B.48 (continued)

Index	Index value	Index	Index value
498	432, 12	1170	463, 105
499	444, 12	1171	477, 105

Index	Index value	Index	Index value
500	455, 12	1172	491, 105
501	466, 12	1173	505, 105
502	478, 12	1174	519, 105
503	489, 12	1175	533, 105
504	501, 12	1176	547, 105
505	512, 12	1177	561, 105
506	523, 12	1178	575, 105
507	535, 12	1179	589, 105
508	546, 12	1180	603, 105
509	558, 12	1181	617, 105
510	569, 12	1182	631, 105
511	580, 12	1183	645, 105
512	592, 12	1184	659, 105
513	603, 12	1185	673, 105
514	614, 12	1186	687, 105
515	626, 12	1187	701, 105
516	637, 12	1188	715, 105
517	649, 12	1189	729, 105
518	660, 12	1190	743, 105
519	671, 12	1191	757, 105
520	683, 12	1192	772, 105
521	694, 12	1193	786, 105
522	705, 12	1194	800, 105
523	717, 12	1195	814, 105
524	728, 12	1196	828, 105
525	740, 12	1197	842, 105
526	751, 12	1198	856, 105
527	762, 12	1199	870, 105
528	774, 12	1200	884, 105
529	785, 12	1201	898, 105
530	796, 12	1202	912, 105

Index	Index value	Index	Index value
531	808, 12	1203	926, 105
532	819, 12	1204	940, 105
533	831, 12	1205	954, 105
534	842, 12	1206	968, 105
535	853, 12	1207	982, 105
536	865, 12	1208	996, 105

Table B.48 (continued)

Index	Index value	Index	Index value
537	876, 12	1209	1010, 105
538	887, 12	1210	2, 116
539	899, 12	1211	15, 116
540	910, 12	1212	30, 116
541	922, 12	1213	45, 116
542	933, 12	1214	59, 116
543	944, 12	1215	74, 116
544	956, 12	1216	89, 116
545	967, 12	1217	104, 116
546	978, 12	1218	119, 116
547	990, 12	1219	134, 116
548	1001, 12	1220	148, 116
549	1013, 12	1221	163, 116
550	2, 23	1222	178, 116
551	11, 23	1223	193, 116
552	23, 23	1224	208, 116
553	34, 23	1225	223, 116
554	46, 23	1226	237, 116
555	57, 23	1227	252, 116
556	68, 23	1228	267, 116
557	80, 23	1229	282, 116

Index	Index value	Index	Index value
558	91, 23	1230	297, 116
559	102, 23	1231	312, 116
560	114, 23	1232	326, 116
561	125, 23	1233	341, 116
562	137, 23	1234	356, 116
563	148, 23	1235	371, 116
564	159, 23	1236	386, 116
565	171, 23	1237	401, 116
566	182, 23	1238	416, 116
567	193, 23	1239	430, 116
568	205, 23	1240	445, 116
569	216, 23	1241	460, 116
570	228, 23	1242	475, 116
571	239, 23	1243	490, 116
572	250, 23	1244	505, 116
573	262, 23	1245	519, 116
574	273, 23	1246	534, 116
575	284, 23	1247	549, 116

Table B.48 (continued)

Index	Index value	Index	Index value
576	296, 23	1248	564, 116
577	307, 23	1249	579, 116
578	319, 23	1250	594, 116
579	330, 23	1251	608, 116
580	341, 23	1252	623, 116
581	353, 23	1253	638, 116
582	364, 23	1254	653, 116
583	375, 23	1255	668, 116
584	387, 23	1256	683, 116

Index	Index value	Index	Index value
585	398, 23	1257	698, 116
586	410, 23	1258	712, 116
587	421, 23	1259	727, 116
588	432, 23	1260	742, 116
589	444, 23	1261	757, 116
590	455, 23	1262	772, 116
591	466, 23	1263	787, 116
592	478, 23	1264	801, 116
593	489, 23	1265	816, 116
594	501, 23	1266	831, 116
595	512, 23	1267	846, 116
596	523, 23	1268	861, 116
597	535, 23	1269	876, 116
598	546, 23	1270	890, 116
599	558, 23	1271	905, 116
600	569, 23	1272	920, 116
601	580, 23	1273	935, 116
602	592, 23	1274	950, 116
603	603, 23	1275	965, 116
604	614, 23	1276	979, 116
605	626, 23	1277	994, 116
606	637, 23	1278	1009, 116
607	649, 23	1279	2, 128
608	660, 23	1280	16, 128
609	671, 23	1281	32, 128
610	683, 23	1282	48, 128
611	694, 23	1283	64, 128
612	705, 23	1284	80, 128
613	717, 23	1285	96, 128
614	728, 23	1286	112, 128

Table B.48 (continued)

Index	Index value	Index	Index value
615	740, 23	1287	128, 128
616	751, 23	1288	144, 128
617	762, 23	1289	160, 128
618	774, 23	1290	176, 128
619	785, 23	1291	192, 128
620	796, 23	1292	208, 128
621	808, 23	1293	224, 128
622	819, 23	1294	240, 128
623	831, 23	1295	256, 128
624	842, 23	1296	272, 128
625	853, 23	1297	288, 128
626	865, 23	1298	304, 128
627	876, 23	1299	320, 128
628	887, 23	1300	336, 128
629	899, 23	1301	352, 128
630	910, 23	1302	368, 128
631	922, 23	1303	384, 128
632	933, 23	1304	400, 128
633	944, 23	1305	416, 128
634	956, 23	1306	432, 128
635	967, 23	1307	448, 128
636	978, 23	1308	464, 128
637	990, 23	1309	480, 128
638	1001, 23	1310	496, 128
639	1013, 23	1311	512, 128
640	2, 35	1312	528, 128
641	12, 35	1313	544, 128
642	23, 35	1314	560, 128
643	35, 35	1315	576, 128
644	47, 35	1316	592, 128

Index	Index value	Index	Index value
645	58, 35	1317	608, 128
646	70, 35	1318	624, 128
647	81, 35	1319	640, 128
648	93, 35	1320	656, 128
649	105, 35	1321	672, 128
650	116, 35	1322	688, 128
651	128, 35	1323	704, 128
652	140, 35	1324	720, 128
653	151, 35	1325	736, 128

Table B.48 (continued)

Index	Index value	Index	Index value
654	163, 35	1326	752, 128
655	175, 35	1327	768, 128
656	186, 35	1328	784, 128
657	198, 35	1329	800, 128
658	209, 35	1330	816, 128
659	221, 35	1331	832, 128
660	233, 35	1332	848, 128
661	244, 35	1333	864, 128
662	256, 35	1334	880, 128
663	268, 35	1335	896, 128
664	279, 35	1336	912, 128
665	291, 35	1337	928, 128
666	303, 35	1338	944, 128
667	314, 35	1339	960, 128
668	326, 35	1340	976, 128
669	337, 35	1341	992, 128
670	349, 35	1342	1008, 128
671	361, 35	—	—

Table B.49 Pitch trigonometric function table and horizontal trigonometric function table

Index	Index value	Index	Index value
0	0.000000	129	0.711432
1	0.006136	130	0.715731
2	0.012272	131	0.720003
3	0.018407	132	0.724247
4	0.024541	133	0.728464
5	0.030675	134	0.732654
6	0.036807	135	0.736817
7	0.042938	136	0.740951
8	0.049068	137	0.745058
9	0.055195	138	0.749136
10	0.061321	139	0.753187
11	0.067444	140	0.757209
12	0.073565	141	0.761202
13	0.079682	142	0.765167
14	0.085797	143	0.769103
15	0.091909	144	0.773010
16	0.098017	145	0.776888

Table B.49 (continued)

Index	Index value	Index	Index value
17	0.104122	146	0.780737
18	0.110222	147	0.784557
19	0.116319	148	0.788346
20	0.122411	149	0.792107
21	0.128498	150	0.795837
22	0.134581	151	0.799537
23	0.140658	152	0.803208
24	0.146730	153	0.806848
25	0.152797	154	0.810457
26	0.158858	155	0.814036

Index	Index value	Index	Index value
27	0.164913	156	0.817585
28	0.170962	157	0.821102
29	0.177004	158	0.824589
30	0.183040	159	0.828045
31	0.189069	160	0.831470,
32	0.195090,	161	0.834863
33	0.201105	162	0.838225
34	0.207111	163	0.841555
35	0.213110	164	0.844854
36	0.219101	165	0.848120
37	0.225084	166	0.851355
38	0.231058	167	0.854558
39	0.237024	168	0.857729
40	0.242980	169	0.860867
41	0.248928	170	0.863973
42	0.254866	171	0.867046
43	0.260794	172	0.870087
44	0.266713	173	0.873095
45	0.272621	174	0.876070
46	0.278520	175	0.879012
47	0.284408	176	0.881921
48	0.290285	177	0.884797
49	0.296151	178	0.887640
50	0.302006	179	0.890449
51	0.307850	180	0.893224
52	0.313682	181	0.895966
53	0.319502	182	0.898674
54	0.325310	183	0.901349
55	0.331106	184	0.903989

Table B.49 (continued)

Index	Index value	Index	Index value
56	0.336890	185	0.906596
57	0.342661	186	0.909168
58	0.348419	187	0.911706
59	0.354164	188	0.914210
60	0.359895	189	0.916679
61	0.365613	190	0.919114
62	0.371317	191	0.921514
63	0.377007	192	0.923880,
64	0.382683,	193	0.926210
65	0.388345	194	0.928506
66	0.393992	195	0.930767
67	0.399624	196	0.932993
68	0.405241	197	0.935184
69	0.410843	198	0.937339
70	0.416430	199	0.939459
71	0.422000	200	0.941544
72	0.427555	201	0.943593
73	0.433094	202	0.945607
74	0.438616	203	0.947586
75	0.444122	204	0.949528
76	0.449611	205	0.951435
77	0.455084	206	0.953306
78	0.460539	207	0.955141
79	0.465977	208	0.956940
80	0.471397	209	0.958703
81	0.476799	210	0.960431
82	0.482184	211	0.962121
83	0.487550	212	0.963776
84	0.492898	213	0.965394
85	0.498228	214	0.966976
86	0.503538	215	0.968522

Index	Index value	Index	Index value
87	0.508830	216	0.970031
88	0.514103	217	0.971504
89	0.519356	218	0.972940
90	0.524590	219	0.974339
91	0.529804	220	0.975702
92	0.534998	221	0.977028
93	0.540171	222	0.978317
94	0.545325	223	0.979570

Table B.49 (continued)

Index	Index value	Index	Index value
95	0.550458	224	0.980785,
96	0.555570,	225	0.981964
97	0.560662	226	0.983105
98	0.565732	227	0.984210
99	0.570781	228	0.985278
100	0.575808	229	0.986308
101	0.580814	230	0.987301
102	0.585798	231	0.988258
103	0.590760	232	0.989177
104	0.595699	233	0.990058
105	0.600616	234	0.990903
106	0.605511	235	0.991710
107	0.610383	236	0.992480
108	0.615232	237	0.993212
109	0.620057	238	0.993907
110	0.624860	239	0.994565
111	0.629638	240	0.995185
112	0.634393	241	0.995767
113	0.639124	242	0.996313

Index	Index value	Index	Index value
114	0.643832	243	0.996820
115	0.648514	244	0.997290
116	0.653173	245	0.997723
117	0.657807	246	0.998118
118	0.662416	247	0.998476
119	0.667000	248	0.998795
120	0.671559	249	0.999078
121	0.676093	250	0.999322
122	0.680601	251	0.999529
123	0.685084	252	0.999699
124	0.689541	253	0.999831
125	0.693971	254	0.999925
126	0.698376	255	0.999981
127	0.702755	256	1.000000
128	0.707107,	—	—

Annex C
(Normative)
Metadata Parameters

For the metadata used in this Annex, the attributes and elements defined in ITU-R BS.2076-2 for content and format are used, and some attributes or sub-elements defined in ITU-R BS.2076-2 are specified as follows.

Provisions on audioProgramme should comply with Table C.1.

Table C.1 Provisions on audioProgramme

Attribute/Sub-element	Provisions	Mandatory/Optional
audioProgrammeName	The value contains a maximum of 32 bytes. Otherwise, the value is truncated.	Mandatory
start	The duration defined by the start and end fields should be the same as that of the audio file. Otherwise, the duration of the audio file is used.	Optional
end		Optional

Provisions on audioContent should comply with Table C.2.

Table C.2 Provisions on audioContent

Attribute/Sub-element	Restrictions	Mandatory/Optional
audioContentName	The value contains a maximum of 32 bytes. Otherwise, the value is truncated.	Mandatory

Provisions on audioObject should comply with Table C.3.

Table C.3 Provisions on audioObject

Attribute/Sub-element	Provisions	Mandatory/Optional
audioObjectName	The value contains a maximum of 24 bytes. Otherwise, the value is truncated.	Mandatory
audioObjectIDRef	It indicates the ID of another nested audioObject. A maximum of four layers are supported.	Optional

Provisions on audioStreamFormat should comply with Table C.4.

Table C.4 Provisions on audioStreamFormat

Attribute/Sub-element	Provisions	Mandatory/Optional
audioStreamFormatName	The value contains a maximum of 32	Mandatory

Attribute/Sub-element	Provisions	Mandatory/Optional
	bytes. Otherwise, the value is truncated.	
formatLabel	0001: PCM stream	Mandatory

Provisions on audioTrackFormat should comply with Table C.5.

Table C.5 Provisions on audioTrackFormat

Attribute/Sub-element	Provisions	Mandatory/Optional
audioTrackFormatName	The value contains a maximum of 32 bytes. Otherwise, the value is truncated.	Mandatory
formatLabel	0001: PCM stream	Mandatory

Provisions on audioPackFormat should comply with Table C.6.

Table C.6 Provisions on audioPackFormat

Attribute/Sub-element	Provisions	Mandatory/Optional
audioPackFormatName	The value contains a maximum of 32 bytes. Otherwise, the value is truncated.	Mandatory

Provisions on audioChannelFormat should comply with Table C.7.

Table C.7 Provisions on audioChannelFormat

Attribute/Sub-element	Provisions	Mandatory/Optional
audioChannelFormatName	The value contains a maximum of 32 bytes. Otherwise, the value is truncated.	Mandatory

Provisions (HOA) on audioBlockFormat should comply with Table C.8.

Table C.8 Provisions on audioBlockFormat

Attribute/Sub-element	Provisions	Mandatory/Optional
order	A maximum of seven orders are supported.	Mandatory

To ensure device compatibility, content interoperability, and controllable complexity of encoding, decoding, and rendering systems, a level-based control mechanism should be used for the number and combination of metadata. Five levels are defined: level 0 to level 4. Level 0 is used to maintain interoperability with existing audio content. This level supports "typeDefinitions = DirectSpeakers" and "SpeakerLabel = M + 000/M + 022/M – 022". Level 1 to level 3 limit the maximum number of audio content elements. Level 4 supports an unlimited number of audio content elements.

Provisions on level classification should comply with Table C.9.

Table C.9 Level classification

<audioformatExtended> Element	**Description**	**Level**				
		0	**1**	**2**	**3**	**4**
audioProgramme	Number of audio programs in an audio file or an audio stream	1	1	4	8	Unlimited
audioContent	Number of program audio content items in a file or a stream	2	4	8	16	Unlimited
audioObject	An audio source in a file or a stream	2	8	64	128	Unlimited
concurrentAudioObject	Number of concurrent audio sources in a time segment	2	8	16	32	Unlimited
audioPackFormat	Number of audio format groups in a file (excluding audio streams)	1	8	32	64	Unlimited
audioChannelFormat	Number of audio formats in a file (excluding audio streams)	2	32	64	128	Unlimited
audioStreamFormat	Number of track group formats in a file (excluding audio streams)	2	32	64	128	Unlimited
audioTrackFormat	Number of audio track formats in a file (excluding audio streams)	2	32	64	128	Unlimited
audioTrackUID	Number of unique identifiers in an audio file or an audio stream	2	32	64	128	Unlimited

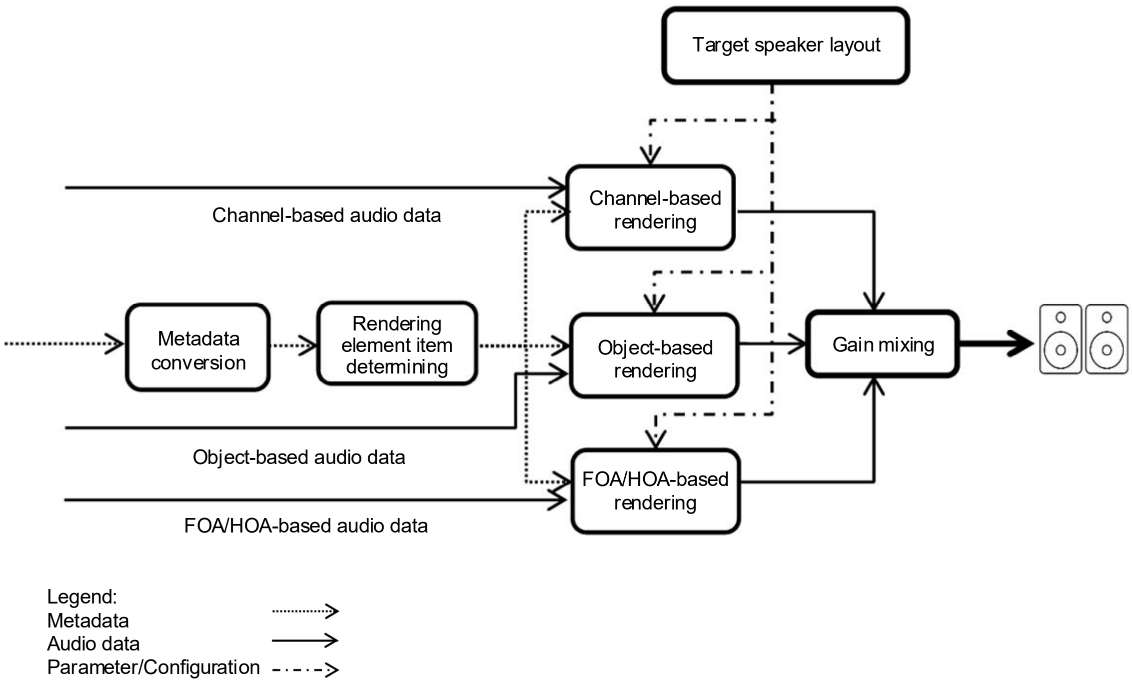
Annex D
(Informative)
3D Audio Rendering

D.1 Speaker Rendering

D.1.1 Speaker Rendering System Framework

The speaker rendering system renders input metadata and audio data based on a specified replay configuration to obtain an audio signal for replaying. Speaker rendering is classified into channel-based rendering, object-based rendering, and FOA/HOA-based rendering. Channel-based rendering converts an input channel signal into a signal required by the target speaker layout, and object-based rendering and FOA/HOA-based rendering use metadata and replay configurations to reproduce object-based and FOA/HOA-based audio data. The three rendering manners may be used in a combination or separately based on actual application requirements. For the speaker rendering system framework, refer to Figure D.1. The metadata is converted based on typeDefinition in the metadata to obtain metadata required by different speaker rendering types. Then, a rendering element item determining module is used to obtain paths of rendering items. Corresponding metadata and audio data are sent to different rendering modules, and then rendered based on a target speaker layout. Finally, a rendered signal is processed by a gain mixing module to obtain a signal finally used for replaying.

Figure D.1 Overall architecture of speaker rendering system



The metadata required for speaker rendering is saved as metadata type items through metadata conversion to correspond to audio types. The metadata types are classified into three types: channel type metadata, object type metadata, and FOA/HOA type metadata. According to different audio types, the track index information track index corresponding to the channel in the audio data is reserved, and is combined with the metadata type item to generate a rendering element item for rendering different types of audio in a next phase.

Channel type metadata contains an audioBlockFormat item. If there is general data, it further contains general data collected from external data, and is constructed as channel type metadata DirectSpeakersTypeMetadata in the metadata.

Each piece of channel type metadata can be independently processed. Therefore, the rendering item contains only one piece of track index information, and is constructed as the channel type rendering item DirectSpeakersRenderingItem.

The object type metadata ObjectTypeMetadata contains one audioBlockFormat item. If there is general data, it further contains one piece of general data collected from external data.

Similar to the channel type metadata, each piece of object type metadata can be independently processed. Therefore, the rendering item also contains only one piece of track index information, and is constructed as the object type rendering item ObjectRenderingItem.

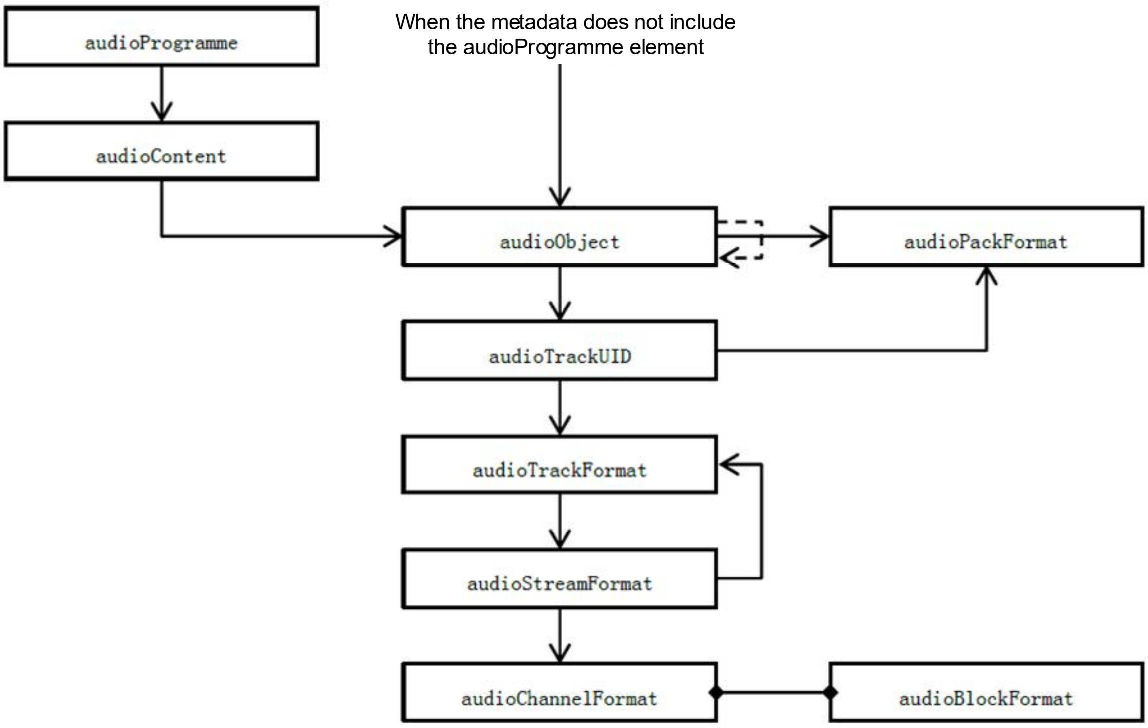
The FOA/HOA type metadata is different from the channel type metadata and the object type metadata. A group of audioChannelFormats are processed at the same time (for example, in the case of order 1, the system needs to process a group of four channels). The metadata does not contain audioBlockFormat or external data. Necessary information (such as the start time, end time, order, and degree) is extracted from audioBlockFormats and directly stored in the FOA/HOA type metadata FOA/HOATypeMetadata.

The rendering element item RenderingItem indicates a to-be-rendered metadata element item, and also includes all information required for performing this operation. The rendering element item is a single audioChannelFormat or a group of audioChannelFormats. Because each audio type has different requirements, different metadata types need to be used to meet rendering requirements of corresponding types.

The FOA/HOA rendering element item FOA/HOARenderingItem constructed by the system is different from that of the other two types, and includes one piece of track index information and a vector including the track.

To determine and construct various types of rendering element items, a structure analysis needs to be performed on the metadata to finally determine a rendering path, as shown in Figure D.2.

Figure D.2 Path of rendering item determining



The start point of rendering item determining is generally an audioProgramme element. If data includes a plurality of audioProgramme elements, the program with the lowest ID is used by default. The audioProgramme element can also be selected based on audioProgrammeID. If there is no audioProgramme element, a set of all audioObject items are used as the start point, and other audioObject do not reference these audioObject items.

During the determining process, the audioPackFormats referenced by each audioObject item and the audioTrackUID referenced by each audioObject item are cross-checked to verify the consistency and integrity of metadata elements.

When a reference loop is detected, audioObjects can be nested.

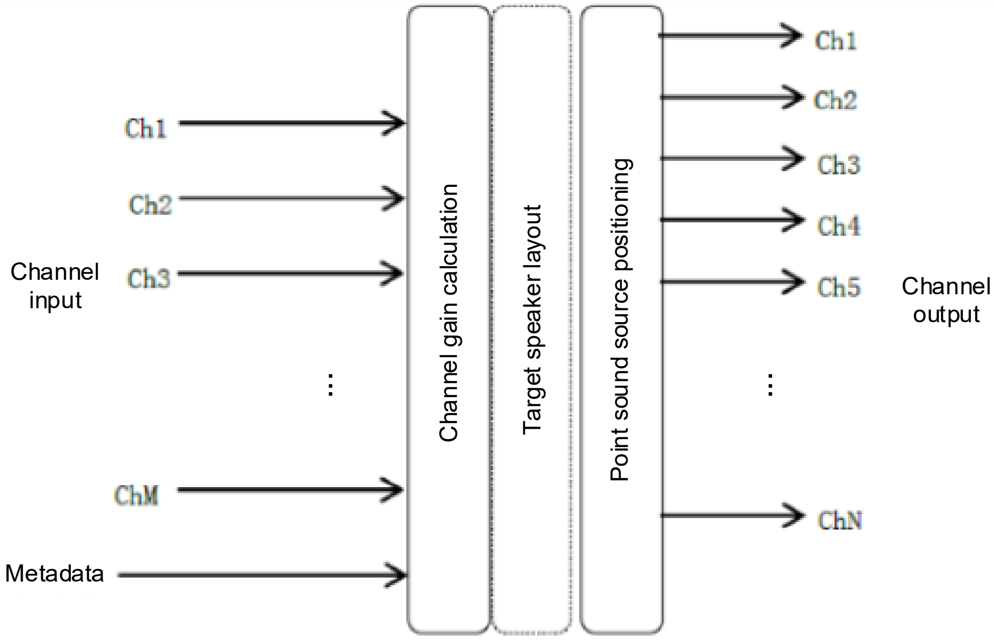
As reference from audioTrackFormats back to audioStreamFormat is optional, the mapping from audioTrackFormat to audioStreamFormats is done reversely from audio StreamFormation.

D.1.2 Channel-based rendering

D.1.2.1 Overview

Figure D.3 is a block diagram of channel-based rendering.

Figure D.3 Block diagram of channel-based rendering



The speaker position corresponding to the input channel signal is determined based on metadata speakerLable. The gain of each channel signal is determined according to the target speaker layout. During gain calculation, when the number of input channels is less than the number of channels in the target speaker layout, the upmixing gain is calculated, and then the output signal is obtained through point sound source positioning. When the number of input channels is greater than the number of channels in the target speaker layout, the downmixing gain is calculated, and then, the output signal is obtained through point sound source positioning. When the number of input channels is the same as the number of channels in the target speaker layout, the channel signal is directly sent to the corresponding speaker.

The screen position corresponding to the audio signal when the terminal replays the audio signal is determined based on the metadata screenEdgeLock screen edge lock. If the speaker coordinates corresponding to the input channel signal are Cartesian coordinates (x, y, z), the speaker coordinates are first converted into polar coordinates (d, θ, φ). Refer to formula (D.1) to formula (D.3).

$$d = \sqrt{x^2 + y^2 + z^2} \quad \text{(D.1)}$$

$$\varphi = \cos^{-1}\frac{z}{d} \quad \text{(D.2)}$$

$$\theta = \tan^{-1}\frac{y}{x} \quad \text{(D.3)}$$

The screen edge lock is divided into lock in the horizontal direction and lock in the vertical direction.

- Horizontal direction: When the left is locked, the left is set as a replay sound source point for the azimuth. When the right is locked, the right is set as a sound source point for the azimuth. If the lock is disabled, the azimuth of the sound source point remains unchanged.

- Vertical direction: When the top is locked, the top is set as a replay sound source point for the azimuth. When the bottom is locked, the bottom is set as a sound source point for the azimuth. If the lock is disabled, the azimuth of the sound source point remains unchanged.

D.1.2.2 Gain Calculation

When the number of input channels does not match the number of sound boxes in the target speaker layout, the gain of each input channel is calculated according to section D.1.2.3. When the input is 5.1 channel and the number of sound boxes in the target speaker layout is less than 3, the 5.1 channel signal is downmixed into a stereo signal in the stereo downmixing manner. The steps are as follows:

a) A sound source position of 0+5+0 is used as an input direction, and a sequence is M+030, M–030, M+000, M+110, and M–110. A corresponding gain vector is *g′*, and a range of each gain value is 0 to 1.

b) Stereo gains *g″* corresponding to M+030 and M–030 are calculated according to formula (D.4).

$$g'' = \begin{bmatrix} 1 & 0 & \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{2}} & 0 \\ 0 & 1 & \sqrt{\frac{1}{3}} & 0 & \sqrt{\frac{1}{2}} \end{bmatrix} \cdot g' \quad \text{……(D.4)}$$

c) The *g″* power is normalized according to formula (D.5) to a value determined by balancing the front and rear speakers a_{front} and a_{rear} in *g′*, so that the source between M+030 and M–030 is not attenuated, and the source between M–110 and M+110 is attenuated by 3 dB.

$$\begin{aligned} a_{\text{front}} &= \max\{g'_1, g'_2, g'_3\} \\ a_{\text{rear}} &= \max\{g'_4, g'_5\} \\ r &= \frac{a_{\text{rear}}}{a_{\text{front}}+a_{\text{rear}}} \\ g &= g''\frac{\left(\frac{1}{2}\right)^{\frac{r}{2}}}{\|g''\|_2} \end{aligned} \quad \text{……(D.5)}$$

$\max\{g'_1, g'_2, g'_3\}$ indicates the maximum value of g'_1, g'_2, and g'_3;

$\max\{g'_4, g'_5\}$ indicates the maximum value of g'_4 and g'_5.

D.1.2.3 Point Sound Source Positioning

If the number of input channels M is not equal to the number of output channels N, it means that the position of the input signal speaker may not match the position of the actual replay speaker. In this case, point sound source positioning needs to be performed, and the actual speaker virtualizes positions corresponding to the N outputs. Point sound source positioning is performed by using a triangular area method. Basic VBAP is implemented in a spherical triangular area formed by three speakers to obtain a position of a virtual speaker. A direction of the virtual speaker is defined as a three-dimensional unit vector shown in formula (D.6).

$$P = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad \text{(D.6)}$$

P indicates the virtual speaker position;

p_1, p_2, p_3 indicate positions of three target speakers closest to P.

The three positions with the minimum distance d between the target speaker and P are calculated. If the nearest speakers cannot be determined from d, the positions are compared according to the sequence of azimuth and pitch angle $\{|\theta|, \theta, |\varphi|, \varphi\}$.

The speakers p_1, p_2, p_3 are disposed on the surface of the sphere. The three-dimensional vector $l_1 = \begin{bmatrix} l_{11} \\ l_{12} \\ l_{13} \end{bmatrix}$. Its origin is the center of the sphere, and the vector points to the direction of the speaker p_1. The three-dimensional vector $l_2 = \begin{bmatrix} l_{21} \\ l_{22} \\ l_{23} \end{bmatrix}$. Its origin is the center of the sphere, and the vector points to the direction of the speaker p_2. The three-dimensional vector $l_3 = \begin{bmatrix} l_{31} \\ l_{32} \\ l_{33} \end{bmatrix}$. Its origin is the center of the sphere, and the vector points to the direction of the speaker p_3. The virtual speaker vector P is represented as a linear combination of three speaker vectors l_1, l_2, l_3, and is represented in a matrix form. Refer to formula (D.7).

$$P^T = g_1 l_1 + g_2 l_2 + g_3 l_3 = g L_{123} \quad \text{(D.7)}$$

g_1, g_2, g_3 indicates gain factors;

g indicates [g1 g2 g3];

L_{123} indicates $\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$;

The virtual speaker P corresponds to the gain vector g. Refer to formula (D.8).

$$g = P^T L_{123}^{-1} = [p_1 \quad p_2 \quad p_3] \begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix}^{-1} \quad \text{(D.8)}$$

For example, the input is 5.1.4, the LFE channel is not considered, and the three-dimensional vectors are defined for nine channels: p_{i1}, p_{i2}, ..., p_{i9}. The target speaker layout is 5.1. The three-dimensional vectors for 5 actual channels are defined as p_{r1}, p_{r2}, ..., p_{r5}. In this case, each virtual gain vector g_{i9} needs to be virtualized for the nine input channels one by one by using an actual three-dimensional vector in a point sound source positioning manner. Refer to formula (D.9).

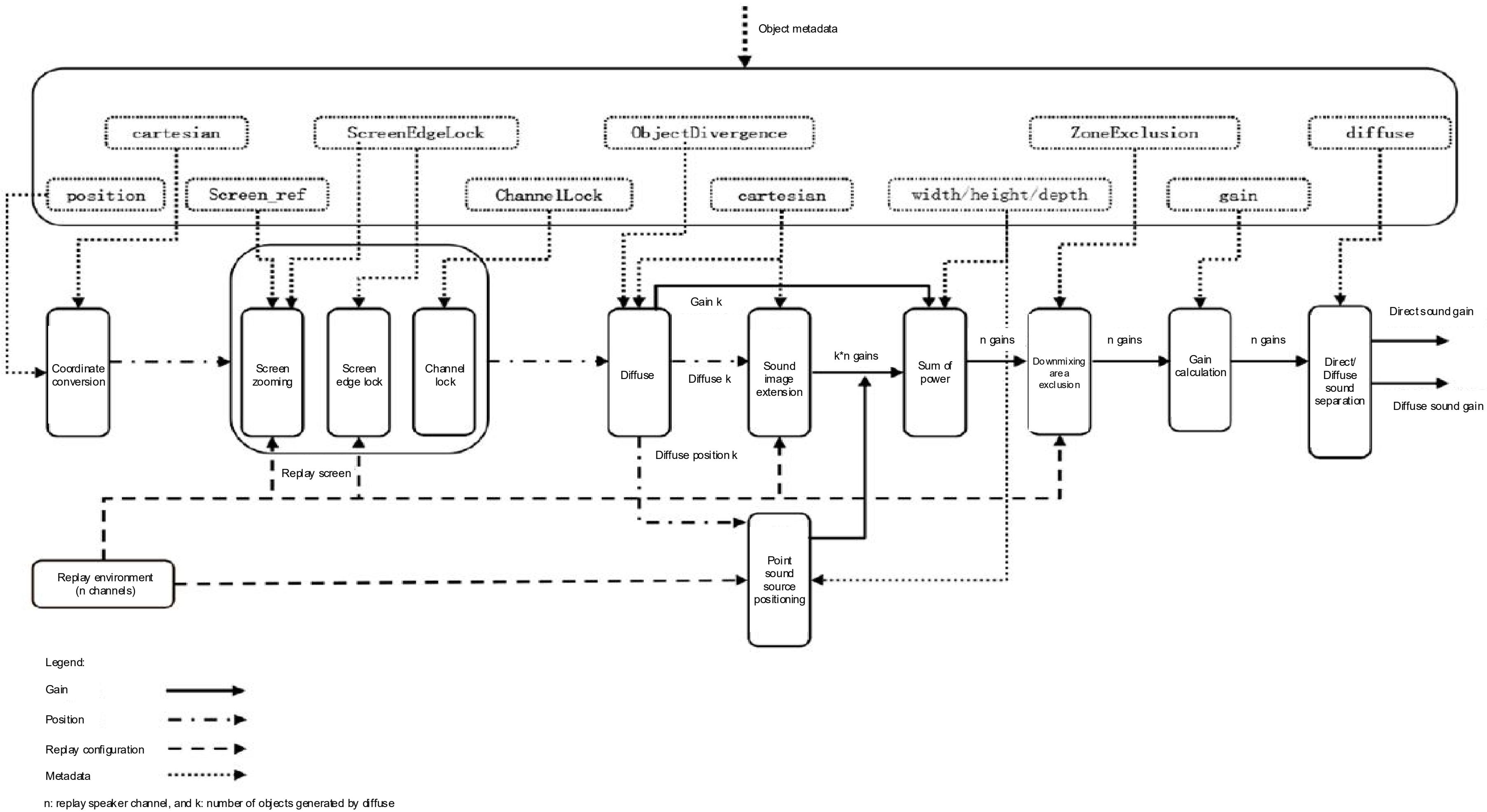
$$g_{i1} = P_{i1}^T L_{r123}^{-1} = [p_{r1} \quad p_{r2} \quad p_{r3}] \begin{bmatrix} l_{r11} & l_{r12} & l_{r13} \\ l_{r21} & l_{r22} & l_{r23} \\ l_{r31} & l_{r32} & l_{r33} \end{bmatrix}^{-1} \quad \text{(D.9)}$$

D.1.3 Object-based Rendering

D.1.3.1 Overview

Different objects of object-based audio content data are rendered. typeDefinition provides the input of metadata and audio data. The metadata enters the renderer in an input form of an object-based metadata type. An output of metadata and audio data based on object content is obtained through processing by the metadata pre-processing module and object gain calculation. For the signal flow process of gain calculation between metadata elements of non-LFE content, refer to Figure D.4.

Figure D.4 Signal flow process of gain calculation between metadata elements



D.1.3.2 Coordinate Conversion

If the position coordinates of the input object are spherical coordinates, formula (D.10) to formula (D.12) are used to convert spherical coordinates (d, θ, φ) to Cartesian coordinates (x, y, z).

$$x = \sin\left(-\frac{\pi}{180}\varphi\right)\cos\left(\frac{\pi}{180}\theta\right)d \qquad \text{(D.10)}$$

$$y = \cos\left(-\frac{\pi}{180}\varphi\right)\cos\left(\frac{\pi}{180}\theta\right)d \qquad \text{(D.11)}$$

$$z = \sin\left(-\frac{\pi}{180}\varphi\right)d \qquad \text{(D.12)}$$

D.1.3.3 Screen Zooming

If Sreen_ref exists in the metadata, screen zooming is used to convert the screen into a Cartesian central position and two vectors (in the x and y directions). The position is updated according to formula (D.13) to formula (D.15).

$$centre = cart(\varphi, \theta, d) \qquad \text{(D.13)}$$

$$\mathtt{width} = \mathrm{w}/2 \qquad \text{(D.14)}$$

$$\mathrm{height} = \mathrm{width}/a \qquad \text{(D.15)}$$

w is the width of the display screen;

a is the width-to-height ratio of the display screen. The default value of w is 3840 x 2160, and the default value of a is 1.78.

For calculation of x and z vectors of the screen, refer to formula (D.16).

$$\begin{aligned} v_x &= \{\mathtt{width}, 0{,}0\} \\ v_z &= \{0{,}0, \mathtt{height}\} \end{aligned} \qquad \text{(D.16)}$$

D.1.3.4 Screen Edge Lock

If ScreenEdgeLock exists in the metadata, screen edge lock is used, and the object position is updated based on the screen edge parameter position.

D.1.3.5 Channel Lock

If ChannelLock exists in the metadata, channel lock is used to update the position. When the Cartesian coordinate position or the width, height, and depth of the spherical coordinates of the metadata is or are received, and the channel lock is enabled, the speakers in a group of possible speakers are compared and calculated, and the speaker closest to the position is locked as the output channel. If the nearest speaker cannot be obtained from the distance, the positions are compared according to the sequence {$|\theta|$, θ, $|\varphi|$, φ} based on the azimuth and pitch angle.

D.1.3.6 Diffuse

The azimuthRange or positionRange are used to obtain a diffuse value. This is implemented by adding two additional source positions p_l and p_r to the left and right sides of the original sound source position p_c. Each position is associated with gains g_c, g_l, and g_r. For calculation of g_c, refer to formula (D.17). For calculation of g_l and g_r, refer to formula (D.18).

$$g_c = \frac{1-x}{x+1} g \quad \text{(D.17)}$$

$$g_l = g_r = \frac{x}{x+1} \quad \text{(D.18)}$$

x is a defuse value, $0 \leq x \leq 1$.

D.1.3.7 Sound Image Extension

The sound image extension is applicable to each position p corresponding to diffuse. A gain vector *gains_for_each_pos* of each speaker is generated for each position p. The gain is mixed with power determined based on the diffuse gain *diverged_gains* to form the sum of power. For the sum of power, refer to formula (D.19).

$$gains[i] = \sqrt{\sum_j diverged\_gains\,[j] \times gains\_for\_each\_pos[j,i]^2} \quad \text{(D.19)}$$

The point sound source positioning used in the channel-based rendering may be used for the sound image extension. The virtual speaker gain of the corresponding object is calculated based on the position, width, height, and depth of the object metadata.

D.1.3.8 Downmixing Area Exclusion

Downmixing area are determined by excluding a speaker. Only the nominal position of the speaker is considered in the calculation. A slight change of the position of the speaker does not affect the behavior of the area exclusion.

For the CartesianZone object, formula (D.20) is used to determine whether the speaker is in the area. {*x*, *y*, *z*} is the nominal position of the speaker, and is converted from polar coordinates with a radius of 1.

$$\begin{aligned} \mathrm{minX} - \epsilon < \quad & x < \mathrm{maxX} + \epsilon \\ \mathrm{minY} - \epsilon < \quad & y < \mathrm{maxY} + \epsilon \\ \mathrm{minZ} - \epsilon < \quad & z < \mathrm{maxZ} + \epsilon \end{aligned} \quad \text{(D.20)}$$

ϵ indicates 10^{-6} and is a safety margin that allows a rounding error during conversion between polar coordinates and Cartesian coordinates;

minX, maxY, and maxZ indicate corner vertices of a cuboid in the three-dimensional space. These corner vertices are excluded from rendering of Cartesian coordinates. A value of the attribute of each corner vertex is a floating point value, and ranges from –1.0 to 1.0. If coordinate values of a speaker are within this range, the speaker is excluded from this area.

The gain vector generated after the area exclusion is extended by adding the LFE channel gain with the value of 0. After the processing in Figure D.4, all gains *gains_full* are generated. One gain value is obtained for each speaker.

D.1.3.9 Separation of Direct Sound and Diffuse Sound

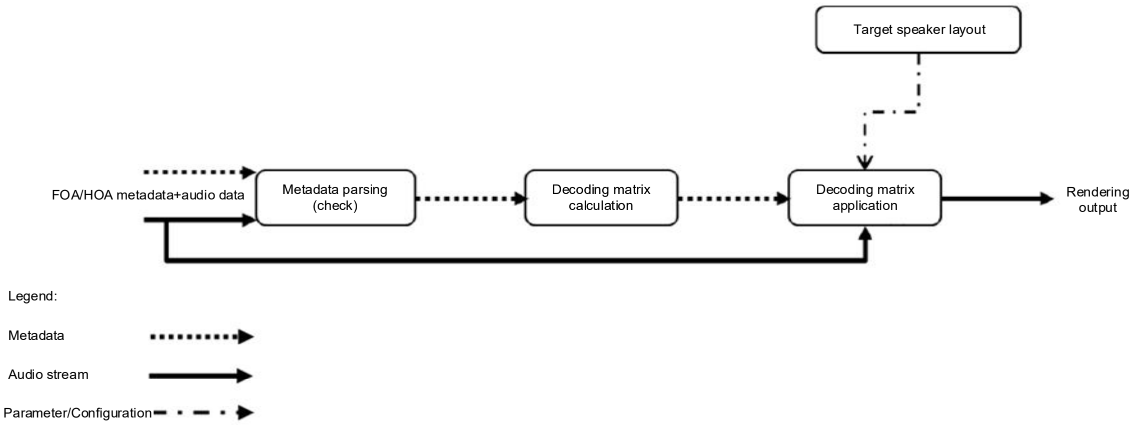
All gains are divided into a direct sound vector *direct* and a diffuse sound vector *diffuse* to control the direct and diffuse paths, depending on the diffuse metadata *diffuse* parameter. For the calculation process, refer to formula (D.21).

$$\begin{aligned} direct \quad &= gains\_full \times \sqrt{1 - diffuse} \\ diffuse \quad &= gains\_full \times \sqrt{diffuse} \end{aligned} \quad \text{(D.21)}$$

D.1.4 FOA/HOA-based Rendering

Figure D.5 is a block diagram of FOA/HOA-based rendering.

Figure D.5 FOA/HOA-based rendering



In the FOA/HOA-based rendering system, the HOA signal can be rendered at a maximum of six orders. The HOA signal is represented by a normalized FOA/HOA type sub-element. The FOA/HOA-based rendering system supports three normalization manners: N3D, SN3D, and FuMa. The same normalization manner is used for all FOA/HOA channels in audioBlockFormat. The FuMa supports only 3-order normalization.

The FOA/HOA rendering process is as follows: The FOA/HOA object format is determined by parsing the metadata, and whether the signal can be rendered is checked. It is determined that the same normalization and screenRef sub-element values are used by all FOA/HOA channels in audioBlockFormat. Then, the speaker decoding matrix is calculated, and the HOA signal rendering output is obtained according to the formula (D.22).

$$S_{\mathrm{spk}} = DS_{\mathrm{FOA/HOA}} \qquad \text{(D.22)}$$

S_{spk} is a target speaker signal matrix, and has $N_{\mathrm{spk}} \times N_{\mathrm{samp}}$ dimensions;

$S_{\mathrm{FOA/HOA}}$ is an HOA audio signal matrix, and has $N_{\mathrm{FOA/HOA}} \times N_{\mathrm{samp}}$ dimensions;

D is an FOA/HOA decoding matrix, and has $N_{\mathrm{spk}} \times N_{\mathrm{FOA/HOA}}$ dimensions.

$N_{\mathrm{FOA/HOA}}$, N_{spk}, and N_{samp} respectively indicates the HOA signal, the speaker signal, and the number of samples.

The AllRAD-based FOA/HOA decoding method is used for rendering. The decoder D of the FOA/HOA calculates the gain value HOAnspN_gain of each FOA/HOA track for each sound box, and outputs the multi-track HOA signal to an independent sound box. The decoding matrix D calculated for the FOA/HOA channel through the FOA/HOA gain calculation is directly applied to the input audio channel to produce the output audio channel.

AllRAD describes the Ambisonic gain = $g_{AllRAD}(\theta)=D_{y_N}(\theta)$ by using the decoder D. The result of D is the most matching gain $g_{\mathrm{VBAP}}(\theta)$ of VBAP-based point sound source positioning. If there is no maximum weight, D is used to define AllRAD for rendering. The integral of D indicates the minimum mean square error of the integral in all directions θ. For specific calculation, refer to formula (D.23).

$$\min_D \int_{S^2} \left\| g_{VBAP}(\theta) - D_{y_N}(\theta) \right\|^2 d\theta \qquad \text{(D.23)}$$

When the sampling function $g_{AMBI}(\theta) = y_N(\theta) diag\{a_N\} y_N(\theta_s)$ at the optimal layout of the virtual speaker is used as a plurality of virtual source inputs, AllRAD may be defined as a VBAP synthesis on a physical speaker. The VBAP synthesis *g* is an integral on an infinite plurality of virtual speakers θ. For specific calculation, refer to formula (D.24).

$$g = \int g_{VBAP}(\theta) g_{AMBI}(\theta)\, d\theta = \int g_{VBAP}(\theta) y_N(\theta) diag\{a_N\} y_N(\theta_s)\, d\theta$$
$$= \underbrace{\int g_{VBAP}(\theta) y_N^T(\theta)\, d\theta}_{D}\, diag\{a_N\} y_N(\theta_s) \qquad \text{(D.24)}$$

The item $diag\{a_N\} y_N(\theta_s)$ is extracted from the integral. The remaining integral defines the matrix D of AllRAD, which is used as the transform of the VBAP speaker gain function $g_{\mathrm{VBAP}}(\theta)$ to the polar coordinate coefficient. This method may provide the most matching FOA/HOA decoding on an irregular speaker layout.

The HOA signal may be decoded based on the matrix D of AllRAD to virtual speakers evenly distributed on the sphere, and virtual speaker signals can be generated on actual speakers through point sound source positioning.

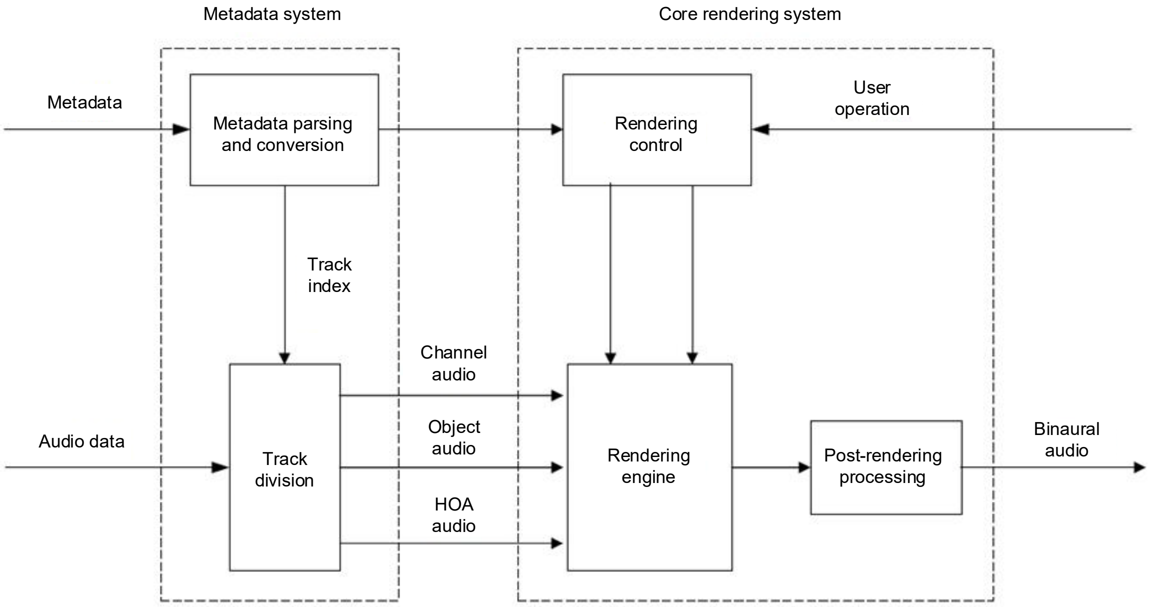
D.2 Binaural Rendering

D.2.1 Binaural Rendering System Framework

D.2.1.1 Overview

The binaural rendering system mainly includes a metadata system and a core rendering system. The metadata system parses and converts the metadata. The metadata includes control information that describes audio content and the rendering system, for example, whether an input audio format is channel audio, object audio, or FOA/HOA, and sound source position information of the object audio. If the input audio is in the interleaved format, the audio track further needs to be divided based on the audio track index in the metadata. The core rendering system performs binauralizing rendering on different audio signals and corresponding metadata to output binaural audio. For the system framework, refer to figure D.6.

Figure D.6 Binaural rendering system framework



D.2.1.2 Binaural Rendering System Input

The input of the rendering system includes audio data and metadata. The audio data supports channel audio, object audio, and FOA/HOA audio. The audio data and the metadata may be read from a local file, or may be read from a decoded bitstream. The former is applicable to an offline production scenario, and the latter is applicable to a streaming rendering scenario.

The metadata includes audioProgramme, audioContent, audioObject, audioTrackUID, audioPackFormat, audioTrackFormat, audioStreamFormat, audioChannelFormat, and audioBlockFormat. The first four are content-related metadata, and the rest are format-related metadata.

When a file is input for binaural rendering, the file needs to include both audio data and metadata. Generally, a file in a BW64 format is used as a carrier, the metadata is stored in axmlChunk, and a correspondence between an audio track index and the metadata is stored in chnaChunk. For details about how to parse a file in the BW64 format, refer to ITU-R BS.2088.

When a real-time decoded bitstream is input for binaural rendering, the decoded metadata is required to indicate the correspondence between the audio track and the metadata.

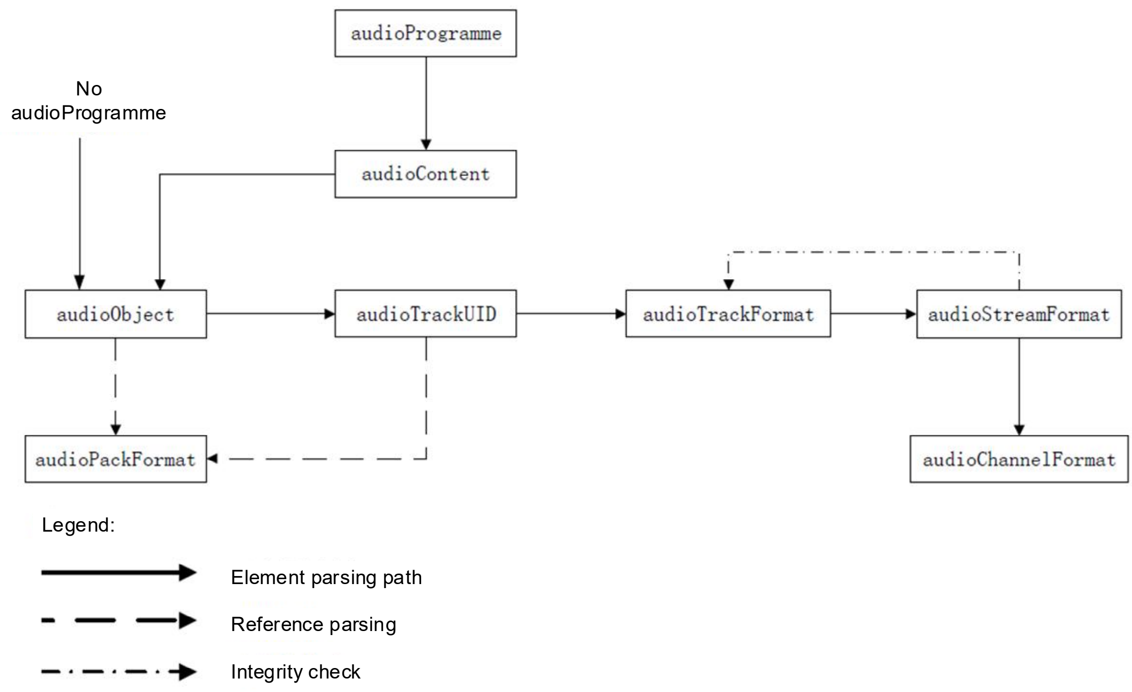
D.2.1.3 Metadata Parsing and Conversion

After the metadata is read from the BW64 file or the decoded bitstream, the metadata needs to be further parsed. To adapt to use of the renderer, metadata conversion usually needs to be further performed, to convert the metadata obtained through parsing into a format that can be supported by the renderer. The metadata obtained from the decoded bitstream does not need to be parsed additionally, but metadata conversion also needs to be performed to meet requirements in different scenarios.

The BW64 file or decoding bitstream is parsed into an element consistent with the metadata definition. Some metadata that cannot be readily used by the renderer, such as speakerLabel, needs to be converted into metadata that can be readily used by the renderer. For dynamic metadata, such as position information of the object audio, interpolation needs to be further performed to prevent audio jump or break after rendering.

Elements of the metadata have a mutual inclusion relationship and a mutual dependency relationship. During parsing, the zeroth audioProgramme element is used as an entry by default. For a metadata parsing process, refer to figure D.7.

Figure D.7 Metadata parsing process



The following should be specially noted during metadata parsing:

a) The default value of audioProgramme is allowed. In this case, audioObject is used as the metadata parsing entry.

b) Check whether the reference of audioPackFormat in audioObject is the same as that in audioTrackUID.

c) The audioTrackFormats referenced in audioStreamFormat is optional. The mutual reference between audioTrackFormat and audioStreamFormat is reverse to each other.

If audioChannelFormat in the channel audio includes only speakerLabel but does not include the speaker layout position of the corresponding channel, the corresponding sound source position parameter needs to be found based on speakerLabel for rendering. For the conversion method, refer to GY/T 316-2018. The following is an example of the conversion method:

```
convert_label_to_positiion() {
    switch speaker_label:
        case FrontLeft:
            return {30, 0, 1};
        case Center:
         return {0, 0 , 1};
         ......
}
```

When a file is input, interpolation needs to be performed for dynamic metadata, for example, object audio position parameters, to ensure the rendering effect. This is because the frame length of the renderer is different from the duration of audioBlockFormat. When the duration of two adjacent blocks is greater than the frame length T_d, a break or jump may occur in the auditory sense. The frame length is determined according to section D.2.2 Core Rendering Engine. To obtain the continuous auditory effect, interpolation needs to be performed on the dynamic metadata. The linear interpolation method is used, and the main steps are divided into framing and interpolation.

a) Framing

Each frame of data includes N sampling points, and the sampling rate is f_s. For the number T_d of milliseconds corresponding to the duration of each frame of data, refer to the formula (D.25).

$$T_d = 1000\frac{N}{F_s} \qquad \text{(D.25)}$$

The block 1 and the block 2 are two consecutive blocks. A relative start time and duration of the block 1 are T_1 and D_1, and a relative start time and duration of the block 2 are T_2 and D_2.

The number of frames in block 1 is $\frac{D_1}{T_d}$. If T_d cannot be exactly divided by D_1, the block 1 is split into two segments. The duration of the first segment can be exactly divided by T_d to obtain the largest value. For the corresponding number N_1 of frames, refer to formula (D.26).

$$N_1 = \lfloor\frac{D_1}{T_d}\rfloor \qquad \text{(D.26)}$$

The remaining part and the block 2 are combined into a new block, and this framing operation is repeated. For a relative start time T_2' of the new block, refer to formula (D.27).

$$T_2' = T_2 - (D_1 - T_d * N_1) \qquad \text{(D.27)}$$

For calculation of the duration D_2', refer to formula (D.28).

$$D_2' = D_2 + (D_1 - T_d * N_1) \qquad \text{(D.28)}$$

Each block is framed according to a sequence until all blocks are processed.

b) Interpolation

The position parameter of audioBlockFormat1 is p_1, and the position parameter of audioBlockFormat2 is p_2. The start point position of interpolation is the block1 (x_1, y_1, z_1), and the end point position is the block2 (x_2, y_2, z_2). The start point and end point of interpolation are determined by the values of jumpPosition and interpolationLength in the metadata.

When jumpPosition is 0, interpolation is performed on the entire data block. The start point block 1 of interpolation is the same as that of p_1, and the end point block 2 is the same as that of p_2. If the length of interpolationLength is 0, interpolation is not performed. The duration of p_1 is the same as the duration of audioBlockFormat1. When interpolationLength is greater than 0, interpolation is performed only on the first interpolationLength samples of audioBlockFormat1, and interpolation is not performed on the remaining part. It can be learned from framing that the block 1 is divided into N1 frames. For an x-coordinate linear interpolation step Δx from the block 1 to the block 2, refer to formula (D.29).

$$\Delta x = \frac{x_2 - x_1}{N_1} \qquad \text{(D.29)}$$

It is assumed that the x coordinate of the block 1 is x_0. For the coordinates $x_1, x_2, x_3 ... x_n$ after interpolation, refer to formula (D.30).

$$x_{n+1} = x_n + \Delta x \qquad \text{(D.30)}$$

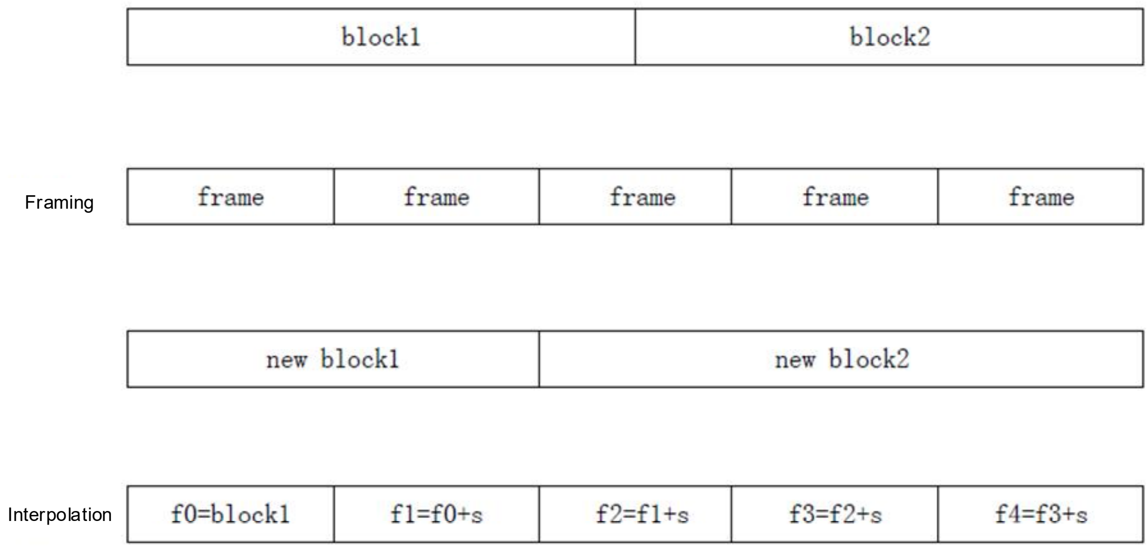
$n \in [0, N_1)$.

Coordinates of the block 1 after interpolation are (x_n, y_n, z_n), and $n \in [0, N_1)$.

The method for calculating the step of the *y* coordinate and the *z* coordinate is the same as that for calculating the step of the *x* coordinate.

For the process of framing and interpolation, refer to figure D.8.

Figure D.8 Flowchart of dynamic metadata framing and interpolation



The value range of the horizontal azimuth is [0, ±180]. There is a critical point of ±180. When the horizontal azimuth increases continuously, the horizontal azimuth returns to the origin. The interpolation calculation uses the proximity principle. For example, the interpolation is performed from azimuth 1 = 170 to azimuth 2 = –170. If counterclockwise interpolation is performed, the absolute difference between two points is 20. If clockwise interpolation is performed, the absolute difference between two points is 340. Therefore, the counterclockwise difference is selected, and the interpolation is performed according to a sequence of 170–180–(–170).

D.2.1.4 Track Division

When the input audio data is in the interleaved format, the audio data of the channels is aliased together. Because metadata needs to be associated and rendered by channel during rendering, the audio in the interleaved format needs to be separated into audio in the tiled form. In this way, each channel has a one-to-one correspondence to an audio track index in the metadata.

The dual-channel audio is used as an example. Figure D.9 shows the principle of audio separation. The upper part of the figure is the audio in the interleaved format. The odd-numbered unit and even-numbered unit respectively indicate the sampling points of the two channels. The lower part is the separated audio in the tiled format. After the separation, the sampling points of each channel are stored in the same audio data block.

Figure D.9 Audio separation principle

The following is an example of the track division code:

```
convert_interleave_to_plannar(){
  for (int i = 0; i < channel; ++i) {
    for (int j = 0; j < sample; ++j) {
      plannar_audio[i][j] = interleave_audio[j * channel + i];
    }
  }
}
```

After track division, the channel audio, the object audio, and the FOA/HOA audio can be separated based on the track index in the metadata. An example of the method is as follows:

```
get_channel_base_audio() {
        return pannar_audio[channels_index];
}
get_object_base_audio() {
        return pannar_audio[objects_index];
}
get_hoa_base_audio() {
        return pannar_audio[hoa_index];
}
```

D.2.1.5 Rendering Control

Rendering control controls the renderer behavior based on the parameters provided in the metadata, adds sound sources based on the audio track index in the metadata, audio type, and position parameters of channels of the channel audio and object audio, and adds the FOA/HOA audio based on the type, order, and degree of the FOA/HOA audio.

The following is an example of adding various types of sound sources:

```
    if (type == DirectSpeaker) {
      add_channel_base_audio_by_index()
    } else if (type == Object) {
      add_object_base_audio_by_index()
    } else if (type == HOA) {
      add_hoa_base_audio_by_index()
}
```

Rendering control updates the position information of the object audio in real time based on the dynamic metadata, and obtains the audio effect of the motion in the space through rendering. An example of the method for setting the real-time position parameter of the object audio is as follows:

```
set_object_base_audio_position() {
      update_source_position_by_index()
}
```

Other functions of rendering control, such as controlling program switching, are implemented by using a similar method.

D.2.1.6 Rendering Engine

The rendering engine implements binaural rendering on the input audio based on the metadata, and outputs the binaural audio. Content of the core rendering engine is described in detail in section D.2.2.

D.2.1.7 Post-rendering Processing

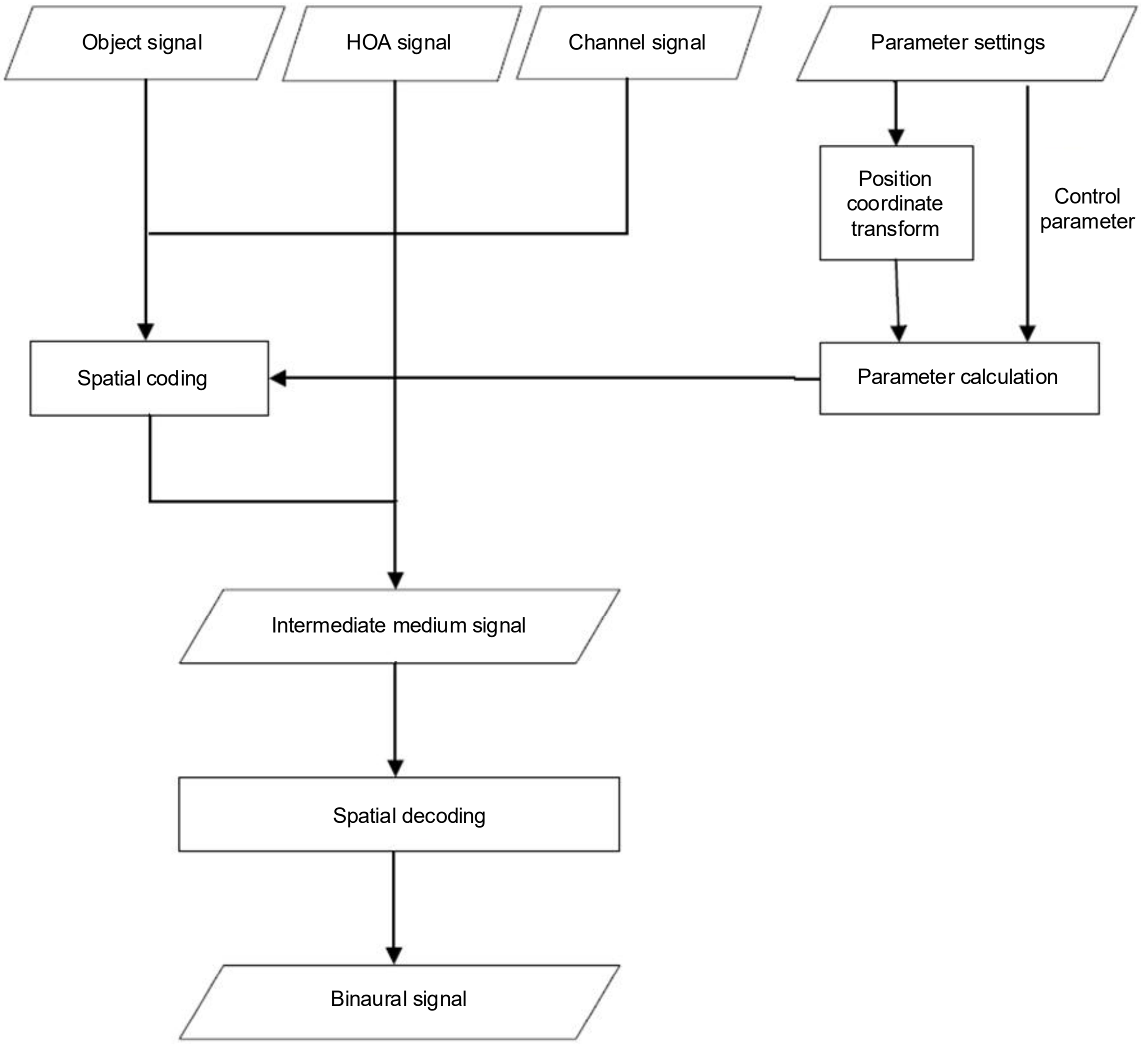
During post-rendering processing, the output binaural audio is corrected, for example, dynamic range control and loudness control. After binauralizing rendering, the amplitude and loudness of the binaural audio obtained through rendering change due to the gain of the header transmission function. When the dynamic and loudness differences between the output audio and the input audio are great, dynamic control and loudness control should be used for adjustment to accurately recover the expression intention of the rendering sequence. Otherwise, extreme cases such as crackle and crack sound may occur. For the loudness control, refer to GY/T 262-2012. Dynamic control uses a limiter to ensure the dynamic range of the output audio.

D.2.2 Principle of Binaural Rendering Engine

D.2.2.1 Overview

The binaural rendering engine uses the sound field reconstruction technology based on the Ambisonic kernel, uses the spherical harmonic function to code input audio to the spherical harmonic domain based on metadata, and stores the input audio in the Ambisonic format as an intermediate medium signal. Spatial encoding includes spatial coding of channel audio and spatial coding of object audio. Because the FOA/HOA audio is in the Ambisonic format, it only needs to be superimposed on the intermediate medium signal without spatial coding. During spatial coding, coordinate transform needs to be performed on the sound source position parameter before the parameter is used. In addition, spatial coding also needs to respond to the control parameter in the metadata and the user interaction. Finally, the binaural audio is output after spatial decoding is performed on the Ambisonic signal. Parameter calculation indicates calculation and conversion of control parameters, for example, coordinate system conversion of position parameters and object audio position update. For the flow of the binaural rendering engine, refer to Figure D.10.

Figure D.10 Principle diagram of binaural rendering engine



D.2.2.2 Position Parameter Coordinate Conversion

The coordinate system used by metadata supports the Cartesian coordinate system and the spherical coordinate system. The metadata spherical coordinate system is shown in a) in figure D.11, and the metadata Cartesian coordinate system is shown in b) in figure D.11. The audio coordinate system shown in d) in figure D.11 is used during spatial encoding. The external interface of the renderer provides the function of setting the position information of multi-channel audio and object audio. This function uses the world coordinate system shown in c) in figure D.11. The coordinate systems are described as follows:

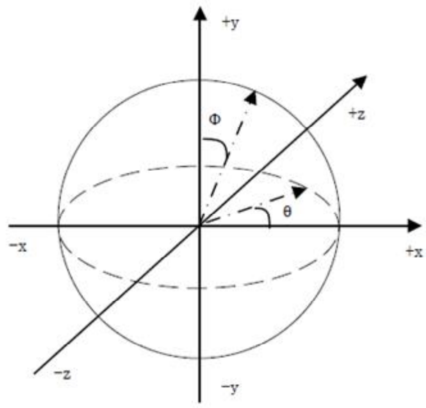
- Metadata Cartesian coordinate system: X points to the right, Y points to the inside of the screen, Z points to the sky, and a human view direction is the positive direction of the Y axis.

- Metadata spherical coordinate system: θ is an angle between a projection of the target point on the *xy* plane and a positive direction of the *x* axis. φ is an angle between a connection line between the target point and the origin and a positive direction of the *z* axis. *r* is a distance from the target point to the origin.

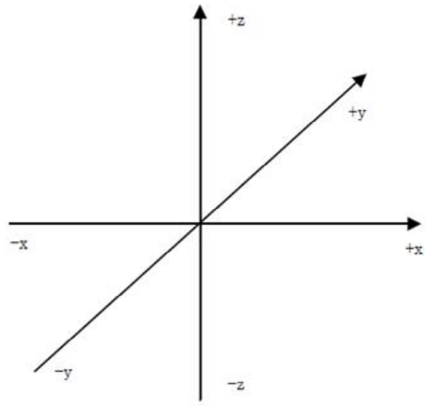
- World coordinate system: X points to the right, Y points to the sky, Z points to the outside of the screen, and a human view direction is the positive direction of the Z axis.

- Audio coordinate system: X points to the inside of the screen, Y points to the left, Z points to the sky, and a human view direction is the positive direction of the X axis.

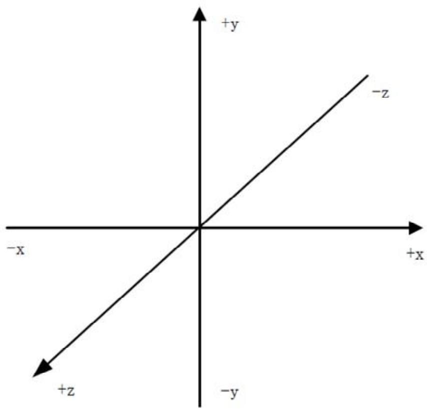
Figure D.11 Definitions of coordinate systems



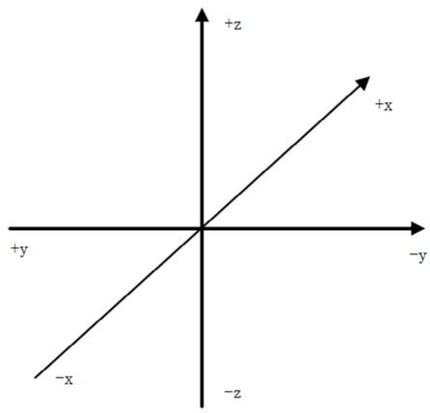
a) Metadata spherical coordinate system



b) Metadata Cartesian coordinate system



c) World coordinate system



d) Audio coordinate system

The method for converting the metadata spherical coordinates (r, azimuth, elevation) into the metadata Cartesian coordinates (x, y, z) is as follows:

```
polar_to_cartisian() {
        x = r * sin(-azimuth) * cos(elevation)
        y = r * cos(-azimuth) * sin(elevation)
        z = r * sin(elevation)
        return {x, y, z}
}
```

The method for converting the metadata Cartesian coordinates (x, y, z) into the world coordinates (x', y', z') is as follows:

```
cartisian_to_world() {
        x' = -x
        y' = z
        z'= y
        return { x', y' ,z' }
}
```

The method for converting the world coordinates (x', y', z') into the audio coordinates (x'', y'', z'') is as follows:

```
world_to_audio() {
        x'' = z'
        y'' = x'
        z'' = y'
        return { x'', y'', z''}
}
```

D.2.2.3 Spatial Coding

The Ambisonic technology is used for spatial coding. The spherical harmonic function is used to perform spatial coding on the channel audio and object audio. The coded intermediate media are uniformly in the ACN SN3D format. The FOA/HOA audio is in the Ambisonic format. It only needs to be superimposed on the intermediate media signals obtained through spatial coding. The input FOA/HOA audio supports two formats: ACN SN3D and ACN N3D. The format of the input FOA/HOA audio is determined through parameter setting. For the spherical harmonic function Y_n^m, refer to formula (D.31).

$$Y_n^m = (-1)^m \sqrt{\frac{(2n+1)(n-|m|)!}{4\pi(n+|m|)!}} P_n^m(cos\theta)e^{im\varphi} \qquad \text{(D.31)}$$

n is an order of the spherical harmonic function, and *m* is a degree of the spherical harmonic function. In section D.2, *n* is used as an order, and *m* is used as a degree by default. *θ* and *φ* describe the horizontal azimuth and pitch angle of the spatial position information of the sound source. In an actual application, a real number form of the spherical harmonic function is usually used, and $e^{im\varphi}$ is replaced with $\Phi_m(\varphi)$. Refer to formula (D.32).

$$\Phi_m(\varphi) = \begin{cases} \sqrt{2}sin(|m|\varphi) & m < 0 \\ 1 & m = 0 \\ \sqrt{2}cos(|m|\varphi) & m > 0 \end{cases} \qquad \text{(D.32)}$$

P_n^m is an accompanying Legendre polynomial. Refer to formula (D.33).

$$P_n^m(x) = (1 - x^2)^{|m|/2} \frac{d^{|m|}}{dx^{|m|}} P_n(x) \qquad \text{(D.33)}$$

$P_n(x)$ is *n*-order Legendre polynomial. Refer to formula (D.34).

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)^n \quad \text{(D.34)}$$

Y_n^m supports the order 1 to order 7, that is, $N \in [1,7]$. For the N-order spherical harmonic function, $n \in [0, \mathrm{N}]$, and $m \in [-N, N]$. The higher order of the spherical harmonic function indicates the higher precision and a better sense of space and surrounding after binauralization.

During spatial coding, an audio signal $\omega(\theta,\varphi)$ is coded to a spherical harmonic domain according to a spherical harmonic function Y_n^m, to generate an n-order Ambisonic signal $S_{\mathrm{FOA/HOA}}$. Refer to formula (D.35).

$$S_{\mathrm{FOA/HOA}}(\theta,\varphi) = Y_n^m \cdot \omega(\theta,\varphi) \quad \text{(D.35)}$$

$\omega(\theta,\varphi)$ indicates input audio at a certain position of the space, and includes audio data and metadata.

After spatial coding is performed by channel, all input signals are superimposed on $S_{\mathrm{FOA/HOA}}$ for storage as intermediate medium signals $S_{\mathrm{FOA/HOA}}(n)$. Refer to formula (D.36).

$$S_{\mathrm{FOA/HOA}}(n) = \sum_{i=0}^{L-1} S_{\mathrm{FOA/HOA}}(\theta,\varphi)_i \quad \text{(D.36)}$$

L is the number of sound sources participating in spatial coding.

D.2.2.4 Spatial Decoding

The intermediate medium signal $S_{\mathrm{FOA/HOA}}(n)$ obtained through spatial coding indicates the Ambisonic signal reconstructed by the 3D sound field of the input audio. Based on this, the $S_{\mathrm{FOA/HOA}}(n)$ signal is filtered according to the spherical harmonic domain header transmission function h_s to obtain the binaural output signal b_ω. Refer to formula (D.37).

$$b_\omega(n) = \sum_{m=0}^{M-1} S_{\mathrm{FOA/HOA}}(n) \cdot h_s(n-m) \quad \text{(D.37)}$$

M is a length of the header transmission function h_s, and is fixed to 256.

n is a length of binauralized data.

The spherical harmonic domain header transmission function h_s is obtained by converting the time-domain header transmission function *h*. The least square method is used to obtain the error of h_s and *h*. When the error reaches the minimum, the rendering effect of h_s and *h* is considered to be the same. For the definition of the error *e*, refer to formula (D.38).

$$e = min \sum_{\Omega \in S^2} |h_s \cdot Y_n^m(\Omega) - h(\Omega)|^2 \quad \text{(D.38)}$$

Ω is the angle at any position on the sphere.

h_s calculated at the minimum error *e* is the header transmission function in spherical harmonic domain.

Because the length of the header transmission function is fixed, the length of the data for each binauralization is also fixed. Therefore, the lengths of the input and output data of each frame of the renderer should be determined during renderer initialization. Typical frame lengths (numbers of sampling points) supported by the renderer are 128, 256, 512, 1024, and 2048. The frame length is also used for framing and interpolation of dynamic metadata.

D.2.2.5 LFE Channel Processing

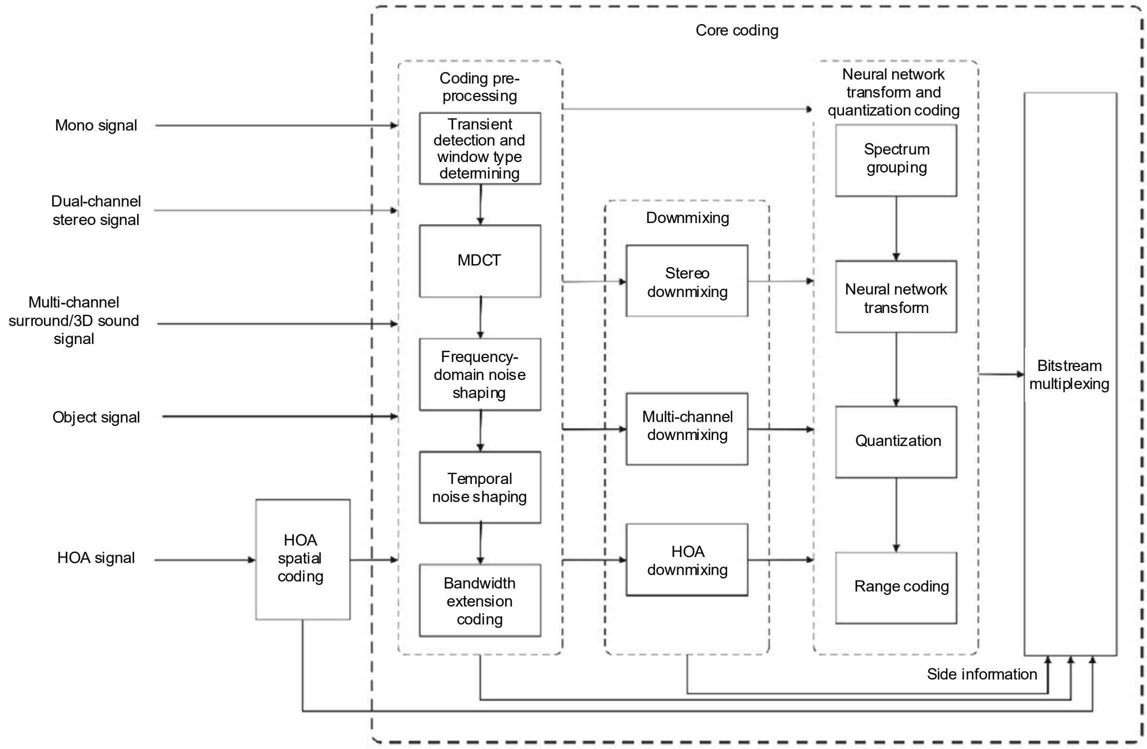
The multi-channel audio usually has an LFE bass channel. During speaker rendering, the bass channel can be rendered by using a high-power subwoofer to obtain the stunning effect. The binaural rendering uses headphones as a playback device. Because the power of the headphones is low, and the low-frequency effect is not obvious, the bass LFE channel may be ignored during binaural rendering.

Annex E
(Informative)
General Bit Rate Audio Coding

E.1 Coding Framework

The general bit rate audio coding includes core coding and HOA spatial coding. The core coding includes coding pre-processing, downmixing, neural network transform, and quantization coding. The coding pre-processing includes transient detection, window type determining, MDCT, frequency-domain noise shaping, temporal noise shaping, bandwidth extension coding. The neural network transform and quantization coding include spectrum grouping, neural network transform, quantization, and range coding. A channel signal and an object signal are coded into bitstreams through core coding, and an HOA signal is coded into a bitstream through HOA spatial coding and the core coding. Figure E.1 is a schematic diagram of the coding framework.

Figure E.1 General bit rate audio coding framework



Each channel signal is transformed from time domain to frequency domain and pre-processed through coding pre-processing. The pre-processed frequency-domain signal is downmixed through signal downmixing in different coding modes to remove the correlation between channels. During neural network transform, quantization, and range coding, a neural network is used to transform and code each downmixed channel. The HOA signal is converted through HOA spatial coding into a transmission channel signal. The following describes each mode coding module.

a) General bit rate audio mono coding

Figure E.2 and Figure E.3 are schematic diagrams of basic structures of the general bit rate audio mono coding. The mono coder performs coding pre-processing on the time-domain mono signal to obtain a processed MDCT coefficient, performs neural network transform to obtain a transform-domain coefficient, and finally performs quantization and range coding to obtain a bitstream. The coding pre-processing module includes a transient detection and window type determining module, a frequency-domain noise shaping module, a temporal noise shaping module, and a bandwidth extension coding module.

1) Transient detection and window type determining

The transient detection module determines, based on the energy of the input time-domain signal, whether there is a transient signal in the current frame. The window type determining module obtains a windowed control parameter of the current frame based on the transient signal detection result. The windowed control parameter includes a 2048-point sine window, a 256-point sine window, and a 2048-point switching window. The input and output parameters are as follows:

- Input: the time-domain mono signal;

- Output: the windowed control parameter.

2) MDCT

The MDCT module performs windowing and MDCT on the input time-domain signal based on the windowed control parameter. The input and output parameters are as follows:

- Input: the mono time-domain signal and the windowed control parameter;

- Output: an MDCT coefficient of the mono signal.

3) Frequency-domain noise shaping

The frequency-domain noise shaping module obtains quantized spectral envelope information based on the input time-domain signal, and adjusts the MDCT coefficient based on the quantized spectral envelope information to implement MDCT spectrum shaping, thereby controlling distribution of quantized noise in frequency domain. The input and output parameters are as follows:

- Input: the windowed time-domain signal and the MDCT coefficient of the mono signal;

- Output: an MDCT coefficient of the mono signal after frequency-domain noise shaping and a frequency-domain noise shaping parameter.

4) Temporal noise shaping

The temporal noise shaping module obtains a temporal noise shaping parameter based on the MDCT coefficient after frequency-domain noise shaping, and shapes the MDCT coefficient based on the temporal noise shaping parameter, to control distribution of quantized noise in time domain. The input and output parameters are as follows:

- Input: the MDCT coefficient of the mono signal after frequency-domain noise shaping;

- Output: an MDCT coefficient of the mono signal after temporal noise shaping and the temporal noise shaping parameter.

5) Bandwidth extension coding

The bandwidth extension coding module obtains a bandwidth extension parameter based on the MDCT coefficient obtained after temporal noise shaping, to indicate a correlation between high and low frequencies of the MDCT spectrum of the signal, so as to assist the decoder in recovering the high frequency component. The input and output parameters are as follows:

- Input: the MDCT coefficient of the mono signal after temporal noise shaping;

- Output: a bandwidth extension parameter.

6) Neural network transform

The neural network transform module transforms, by using a neural network, the MDCT coefficient after coding pre-processing, to further remove information redundancy in the spectral coefficient. An output of the neural network is referred to as a transform-domain coefficient, and the transform-domain coefficient is used for quantization and range coding. The input and output parameters are as follows:

- Input: the pre-processed MDCT coefficient of the mono signal;

- Output: the transform-domain coefficient.

7) Quantization and range coding

The quantization module performs linear scalar quantization on the transform-domain coefficient obtained through neural network transform, and the coding module performs range coding on a quantization result to obtain a bitstream. The input and output parameters are as follows:

- Input: the transform-domain coefficient;

- Output: a bitstream.

Figure E.2 General bit rate audio mono coder

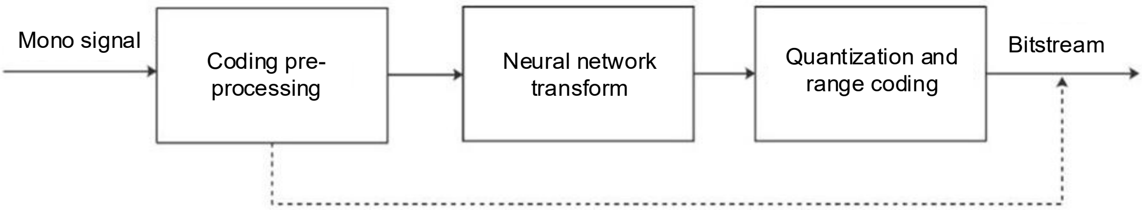
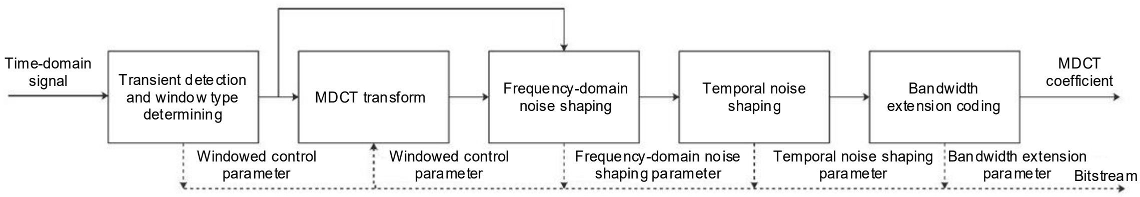


Figure E.3 General bit rate audio coding pre-processing



b) General bit rate audio dual-channel stereo coding

Figure E.4 is a schematic diagram of a basic structure of the general bit rate audio dual-channel stereo coding. The dual-channel stereo coder performs dual-channel stereo coding pre-processing on a time-domain dual-channel stereo signal to obtain a processed dual-channel stereo MDCT coefficient, obtains a downmixed dual-channel stereo MDCT coefficient through a dual-channel stereo downmixing and bit allocation module, obtains a transform-domain coefficient through neural network transform, and finally obtains a bitstream through quantization and range coding. The dual-channel stereo coding pre-processing module includes a transient detection and window type determining module, a frequency-domain noise shaping module, a temporal noise shaping module, and a bandwidth extension coding module.

1) Dual-channel stereo coding pre-processing

The dual-channel stereo coding pre-processing module performs coding pre-processing on each channel of the dual-channel stereo, and includes a transient detection and window type determining module, a frequency-domain noise shaping module, a temporal noise shaping module, and a bandwidth extension coding module. The input and output parameters are as follows:

- Input: a time-domain dual-channel stereo signal;

- Output: an MDCT coefficient of the pre-processed dual-channel stereo signal, a windowed control parameter, a frequency-domain noise shaping parameter, a temporal noise shaping parameter, and a bandwidth extension parameter.

2) Dual-channel stereo downmixing and bit allocation

The dual-channel stereo downmixing and bit allocation module performs MCR or M/S downmixing and bit allocation on the left and right channel MDCT coefficient based on the dual-channel stereo signal feature. The input and output parameters are as follows:

- Input: the MDCT coefficient of the pre-processed dual-channel stereo signal;

- Output: a downmixed channel MDCT coefficient and a bit allocation parameter.

3) Neural network transform. The input and output parameters are as follows:

- Input: the downmixed channel MDCT coefficient;

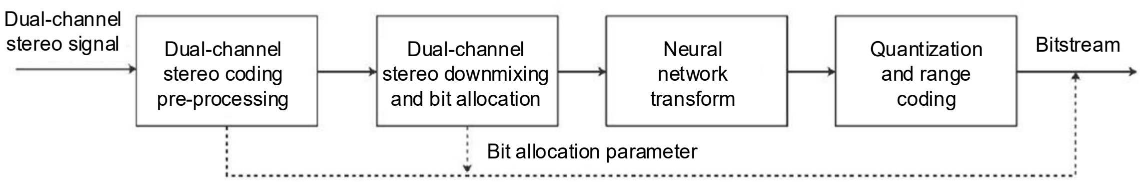
- Output: a downmixed channel transform-domain coefficient.

4) Quantization and range coding. The input and output parameters are as follows:

- Input: the downmixed channel transform-domain coefficient;

- Output: a bitstream.

Figure E.4 General bit rate audio dual-channel stereo coder



c) General bit rate audio multi-channel coding

Figure E.5 is a schematic diagram of a basic structure of the general bit rate audio multi-channel coding. The multi-channel coder performs coding pre-processing on a time-domain multi-channel signal to obtain a processed MDCT coefficient, obtains a multi-channel downmixing parameter through the multi-channel mode determining module, obtains a downmixed multi-channel MDCT coefficient through a multi-channel downmixing and bit allocation module, obtains a transform-domain coefficient through neural network transform, and finally obtains a bitstream through quantization and range coding. The coding pre-processing module includes a transient detection and window type determining module, a frequency-domain noise shaping module, a temporal noise shaping module, and a bandwidth extension coding module.

1) Multi-channel coding pre-processing

The multi-channel coding pre-processing module performs coding pre-processing on each channel. The input and output parameters are as follows:

- Input: the time-domain multi-channel signal;

- Output: an MDCT coefficient of the pre-processed multi-channel signal, a windowed control parameter, a frequency-domain noise shaping parameter, a temporal noise shaping parameter, and a bandwidth extension parameter.

2) Multi-channel mode determining

The multi-channel mode determining module determines a multi-channel signal coding mode parameter based on the multi-channel signal features and the signal correlation. The input and output parameters are as follows:

- Input: the MDCT coefficient of the pre-processed multi-channel signal;

- Output: the multi-channel signal coding mode parameter.

3) Multi-channel downmixing and bit allocation

The multi-channel signal downmixing module performs multi-channel downmixing on the MDCT coefficient of the pre-processed multi-channel signal based on the multi-channel signal coding mode parameter, and performs M/S downmixing on MDCT coefficients of two channels in each group based on a pair parameter. The bit allocation module is configured to determine a bit allocation parameter. The input and output parameters are as follows:

- Input: the MDCT coefficient of the pre-processed multi-channel signal and the multi-channel signal coding mode parameter;

- Output: a downmixed multi-channel MDCT coefficient and the bit allocation parameter.

4) Neural network transform. The input and output parameters are as follows:

- Input: the downmixed multi-channel MDCT coefficient;

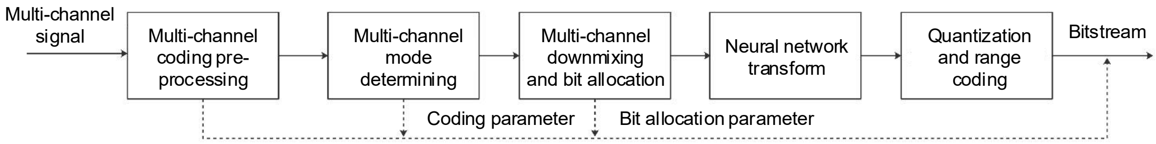
- Output: a downmixed multi-channel transform-domain coefficient.

5) Quantization and range coding. The input and output parameters are as follows:

- Input: the downmixed multi-channel transform-domain coefficient;

- Output: a bitstream.

Figure E.5 General bit rate audio multi-channel coder



d) General bit rate audio HOA coding

Figure E.6 and Figure E.7 are schematic diagrams of basic structures of the general bit rate audio HOA coding. HOA spatial coding is performed on the HOA signal to obtain a transmission channel signal, and then transmission channel coding pre-processing is performed to obtain a processed MDCT coefficient. The HOA downmixing and bit allocation module obtains a downmixed transmission channel MDCT coefficient. Neural network transform is performed to obtain a transform-domain coefficient. Finally, quantization and coding are performed to obtain a bitstream. HOA spatial coding includes a sound field component analysis module, a sound field component synthesis module, and an other component calculation module. The coding pre-processing module includes a transient detection and window type determining module, a frequency-domain noise shaping module, a temporal noise shaping module, and a bandwidth extension coding module.

1) Sound field component analysis

The sound field component analysis module performs sound field component analysis on an HOA signal through linear decomposition to determine the sound field distribution features, such as the number of sound sources, directivity of sound sources, and dispersion of sound sources. The input and output parameters are as follows:

- Input: the HOA signal;

- Output: a sound field analysis parameter.

2) Sound field component synthesis

The sound field component synthesis module configures the coder based on the sound field analysis parameters and other configuration parameters, determines a sound field component parameter based on the main sound field component of the HOA signal, and synthesizes a transmission channel signal through linear reversible conversion. The sound field component parameter includes virtual speaker attribute information, and the transmission channel signal includes a virtual speaker signal. The input and output parameters are as follows:

- Input: the HOA signal and the sound field analysis parameter;

- Output: a sound field component parameter and a transmission channel signal.

3) Other component calculation

The remaining component calculation module configures the encoder based on the sound field analysis parameter and other configuration parameters, and determines a transmission channel signal and an other component parameter based on the HOA signal, the virtual speaker signal, and the sound field component parameter. The transmission channel signal includes a residual signal. The input and output parameters are as follows:

- Input: the HOA signal, the sound field analysis parameter, and the sound field component parameter;

- Output: the transmission channel signal and the other component parameter.

4) Transmission channel coding pre-processing

The transmission channel coding pre-processing module performs coding pre-processing on each channel, including a transient detection and window type determining module, a frequency-domain noise shaping module, a temporal noise shaping module, and a bandwidth extension coding module. The input and output parameters are as follows:

- Input: the time-domain transmission channel signal;

- Output: an MDCT coefficient of the pre-processed transmission channel signal, a windowed control parameter, a frequency-domain noise shaping parameter, a temporal noise shaping parameter, and a bandwidth extension parameter.

5) Transmission channel mode determining

The transmission channel mode determining module determines the transmission channel signal coding mode parameter based on the transmission channel signal features and the sound field analysis parameter, for example, the grouping parameter, the bandwidth parameter, the downmixing parameter, and the initial bit allocation parameter. The input and output parameters are as follows:

- Input: the MDCT coefficient of the pre-processed transmission channel signal and the sound field analysis parameter;

- Output: a transmission channel coding mode parameter.

6) HOA downmixing and bit allocation

The HOA downmixing and bit allocation modules perform downmixing and bit allocation on the transmission channel MDCT coefficient. The input and output parameters are as follows:

- Input: the MDCT coefficient of the pre-processed transmission channel signal and the transmission channel coding mode parameter;

- Output: a downmixed channel MDCT coefficient and a bit allocation parameter.

7) Neural network transform. The input and output parameters are as follows:

- Input: the downmixed channel MDCT coefficient;

- Output: a downmixed channel transform-domain coefficient.

8) Quantization and coding. The input and output parameters are as follows:

- Input: the downmixed channel transform-domain coefficient;

- Output: a bitstream.

Figure E.6 General bit rate audio HOA spatial coding

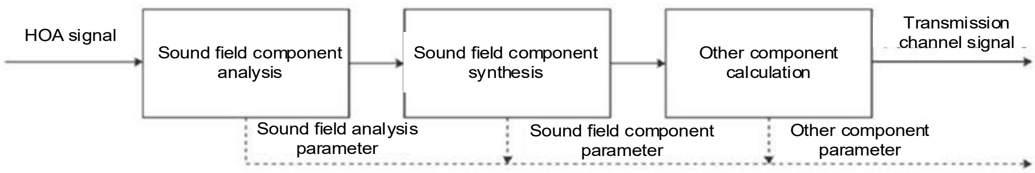
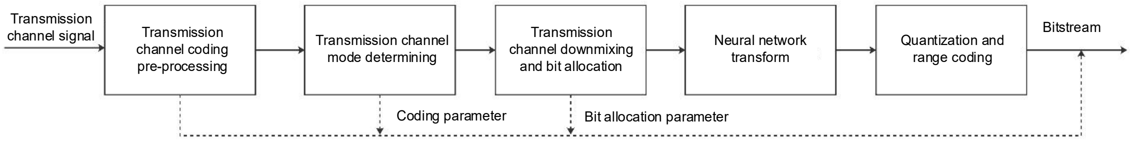


Figure E.7 General bit rate audio HOA coder



E.2 Coding Pre-processing

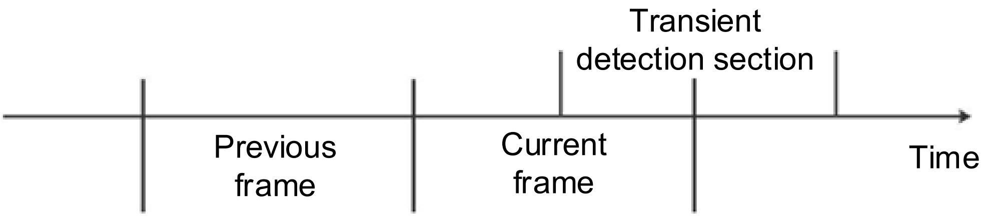
E.2.1 Transient Detection and Window Type Determining

To better code audio signals with different characteristics, a transform-domain coder needs to add a short window to a transient signal to ensure a good time-domain resolution, and needs to add a long window to a steady signal to ensure a good frequency-domain resolution.

Therefore, before the window type determining is performed, transient detection needs to be performed on the input audio signal first. The transient detection determines whether the current frame is a transient signal.

Figure E.8 is a schematic diagram of signal framing in the transient detection algorithm.

Figure E.8 Schematic diagram of signal framing in the transient detection algorithm



The length of the input audio signal of the previous frame is 1024 sampling points. The length of the input audio signal of the current frame is also 1024 sampling points. The length of the transient detection section is 1024 sampling points. The first 512 sampling points are the last 512 sampling points of the input audio signal of the current frame, and the last 512 sampling points are the first 512 sampling points of the future frame, that is, the lookahead length is 512 points. The transient detection of the current frame is performed in the transient detection section. The basic process is as follows:

a) High-pass filtering is performed on a signal in the transient detection section.

   The energy of an audio signal is usually concentrated in a low-frequency part, and a signal whose high-frequency energy changes suddenly may be missed when transient detection is performed directly on the input audio signal.

   Therefore, before the transient detection, high-pass filtering may be performed on the audio signal in the transient detection section, to suppress a low-frequency signal component to some extent, thereby facilitating detection of the transient signal.

b) The signal in the transient detection section is divided into blocks, and block energy and a corresponding energy threshold are calculated.

   The signal in the transient detection section is divided into eight equal-length blocks. A length of the signal in the transient detection section is 1024 sampling points, and a length of a block is 128 sampling points. After the signal is divided into blocks, the energy blockEner of the signal block may be calculated, and is represented by formula (E.1).

$$blockEner[i] = \sum_{n=blockStart}^{blockEnd} signal^2[n] \quad \text{(E.1)}$$

i is a sequence number of a signal block;

n is a sampling point sequence number;

blockStart is a sequence number of a start point of the signal block;

blockEnd is a sequence number of an end point of the signal block;

signal[*n*] is an audio signal of the *n*-th sampling point.

The transient detection requires a signal energy threshold parameter corresponding to each signal block, and a calculation process of the signal energy threshold parameter is described as follows:

It is assumed that signal energy of eight blocks is *blockEner*[8], a corresponding signal energy threshold is enerThresh[8], and a signal energy threshold history is enerThreshHist (initialized to a preset minimum energy value 103.37).

A manner of calculating the signal energy threshold is as follows:

```
for i = 0 to 7:
  enerThresh[i] = enerThreshHist
  enerThreshHist *= 0.8125
  if blockEner[i] > enerThreshHist:
    enerThreshHist = blockEner[i]
  end
end
```

That is, for each signal block, the signal energy threshold history enerThreshHist is used as a signal energy threshold of the current i-th block. Then, the signal energy threshold is attenuated. An attenuated signal energy threshold is compared with energy of the current block. If the energy of the current block is large, the signal energy threshold is updated.

c) Transient detection is performed on the current frame.

Based on the signal energy blockEner of each block in the current frame and the corresponding signal energy threshold enerThresh, logic of determining the transient identifier curIsTransient of the current frame is as follows:

```
if initFrame:
 curIsTransient = 0
else:
 for i = 0 to 7:
  if blockEner[i] > 8.0 * enerThresh[i]:
   curIsTransient = 1
   break
  end
 end
end
```

That is, the transient detection result of the first frame is 0, and if a block exists in subsequent frames, and the signal energy of the block is greater than several times the signal energy threshold, it is considered that the transient signal exists, and the transient identifier is set to 1.

The window type of the current frame may be determined based on the transient detection results of the current frame and the previous two frames.

If the current frame is the i-th frame, a transient detection result of the (i–2)-th frame is denoted as preIsTransient[0], and a transient detection result of the (i–1)-th frame is denoted as preIsTransient[1].

There are four windows types windowType: a long window ONLY_LONG_WINDOW, a cut-in window LONG_SHORT_TRANS_WINDOW, a cut-out window SHORT_LONG_TRANS_WINDOW, and a short window ONLY_SHORT_WINDOW. The pseudocode of the window type determining process is as follows:

```
if preIsTransient[0] == 0 && preIsTransient[1] == 0 && curIsTransient == 0:
  windowType = ONLY_LONG_WINDOW
else if preIsTransient[0] == 0 && preIsTransient[1] == 0 && curIsTransient == 1:
  windowType = LONG_SHORT_TRANS_WINDOW
else if preIsTransient[0] == 1 && preIsTransient[1] == 0 && curIsTransient == 0:
  windowType = SHORT_LONG_TRANS_WINDOW
else:
  windowType = ONLY_SHORT_WINDOW
end
```

E.2.2 MDCT

Windowing and MDCT may be performed on the input audio signal of the current frame based on the window type. The definition of each window type is as follows:

a) Long window: LONG_SHORT_TRANS_WINDOW, a sine window with a length of 2048 points;

b) Cut-in window: LONG_SHORT_TRANS_WINDOW, consisting of a left half sine window of 1024 points, 448 points of 1, a right half sine window of 128 points, and 448 points of 0;

c) Cut-out window: SHORT_LONG_TRANS_WINDOW, consisting of 448 points of 0, a left half sine window of 128 points, 448 points of 1, and a right half sine window of 1024 points;

d) Short window: ONLY_SHORT_WINDOW, a sine window with a length of 256 points. Eight short windows are added to each frame.

The object of windowing and MDCT processing is a 2048-point audio signal constituted by the previous frame input signal frame_past and the current frame input signal frame_curr. The time-frequency transform is MDCT transform.

Short frame MDCT spectrum interleaving: If the window type of the current frame is ONLY_SHORT_WINDOW, interleaving needs to be performed on the MDCT spectra of eight subframes of the current frame (the length of the MDCT spectrum of each subframe is 128 frequencies) to perform subsequent coding pre-processing.

Figure E.9 is a schematic diagram of spectrum interleaving.

Figure E.9 Schematic diagram of spectrum interleaving

Subframe 0 Frequency 0	Subframe 1 Frequency 0	Subframe 2 Frequency 0	Subframe 3 Frequency 0	Subframe 4 Frequency 0	Subframe 5 Frequency 0	Subframe 6 Frequency 0	Subframe 7 Frequency 0	Subframe 0 Frequency 1	Subframe 1 Frequency 1	Subframe 2 Frequency 1	Subframe 3 Frequency 1	Subframe 4 Frequency 1	Subframe 5 Frequency 1	Subframe 6 Frequency 1	Subframe 7 Frequency 1	Subframe 0 Frequency 2	... ...	Subframe 0 Frequency 3	... ...

E.2.3 Frequency-domain Noise Shaping

Frequency-domain noise shaping is used to control distribution of quantized noise introduced in the coding process in frequency domain, reduce perceptible quantized noise based on human auditory masking effect, and improve coding quality.

The frequency-domain noise shaping algorithm is an MDCT spectrum shaping technology based on the LPC parameter, and may include: signal pre-emphasis, autocorrelation coefficient calculation, LPC parameter solving, conversion from LPC to LSF, LSF parameter quantization coding, spectrum shaping, and the like.

a) The signal is pre-emphasized:

   The pre-emphasis coefficient is 0.9.

   The length of the windowed audio signal is 2048. The pseudocode of the pre-emphasis process is as follows:

```
for i = 2047 to 1
  preemphWinSignal[i] = winSignal[i] – preemphFactor * winSignal[i-1]
end
preemphWinSignal[0] = winSignal[0]
```

   winSignal is the windowed audio signal, preemphWinSignal is the pre-emphasis audio signal, and preemphFactor is the pre-emphasis coefficient.

b) The autocorrelation coefficient is calculated: The LPC parameter order lpcOrder is 16, and the autocorrelation coefficient of the input signal whose length is 17 needs to be calculated. The signal used for the autocorrelation calculation is the audio signal obtained through the pre-emphasis processing.

c) The LPC parameter is solved: The Wiener-Levinson-Durbin iterative algorithm is used to calculate the LPC parameter with the autocorrelation coefficient as the input. The parameter order is 16.

d) LPC is converted to LSF: The LSP parameter is solved by using Chebyshev polynomial method with 16-order LPC parameter as the input.

   1) The LSP parameter is converted to the LSF parameter. The value range of the LSF parameter is [0, samplingRate/2]. samplingRate is fixed to 48 kHz.
   2) For the conversion from the LPC parameter to the LSF parameter, refer to provisions in section 5.1.9 in 3GPP TS 26.445 (Release 17).

e) LSF parameter quantization coding:

   The LSF parameter quantization coding uses a vector quantization technology, and the LSF parameter has 16 dimensions.

   The structure of the VQ code table is multi-stage split vector quantization. For different coding rates, two sets of vector quantization codebooks are used: high-rate vector quantization codebook and low-rate vector quantization codebook.

   The method for determining whether to use the high-rate or low-rate vector quantization codebook is as follows: If the average coding rate of each channel is greater than 32 kb/s, the high-rate VQ code table is used; otherwise, the low-rate VQ code table is used. The average coding rate of the channels is equal to the total coding rate divided by the number of channels of the input signal.

For the structure of the high-rate codebook and the low-rate codebook, refer to section 7.11.3.

The LSF VQ process is as follows:

1) Mean subtraction of LSF vector

   The LSF mean value is subtracted from the LSF parameter calculated for the current frame to obtain a residual LSF vector, which is used as an input vector of the VQ algorithm.

2) Stage 1 vector quantization

   The LSF vector is divided through stage 1 vector quantization into two sub-vectors respectively having 9 and 7 dimensions. VQ code tables respectively corresponding to the two sub-vectors are searched for to obtain several candidate stage 1 quantization results with the minimum error.

   In addition, there are four candidate stage 1 quantization results.

3) Stage 2 vector quantization

   The high rate mode is used as an example, in two stage 1 VQ sub-vectors, a sub-vector having 9 dimensions is further divided into three stage 2 sub-vectors (respectively have 3, 3, and 3 dimensions).

   In the stage 2 VQ, a stage 2 residual vector is first separately calculated for the four candidate quantization results of the stage 1 VQ. Then, split vector quantization is performed on the stage 2 residual vectors to obtain stage 2 VQ errors corresponding to the four candidate quantization results. A stage 1 VQ index with a minimum stage 2 VQ error and three stage 2 VQ indexes are used as a final vector quantization result.

f) Spectrum shaping:

1) Conversion from LSF to LPC

   The quantized LSF parameter obtained through vector quantization is converted to obtain the quantized LPC parameter.

   The conversion from the LSF parameter to the LPC parameter includes two steps: conversion from the LSF parameter to the LSP parameter and conversion from the LSP parameter to the LPC parameter.

2) Spectrum shaping gain calculation

   First, the quantized LPC parameter is weighted to obtain the LPC weighted coefficient. The pseudocode is as follows:

```
GAMMA_LPC = 0.939999998
weightFactor[0] = 1.0
for i = 1 to lpcOrder:
  weightFactor[i] = weightFactor[i-1] * GAMMA_LPC
end
```

   GAMMA_LPC is the initial value of the LPC weighting coefficient, and weightFactor[lpcOrder+1] is the calculated LPC weighting coefficient.

   The LPC weighting coefficient is multiplied by the quantized LPC parameter to obtain a weighted LPC parameter weightedLpc. Second, the low-precision LPC spectrum shaping gain is calculated.

   Pre-rotation is performed on the weighted LPC parameter, complex FFT is performed on the pre-rotated LPC parameter rotateLpc, and the low-precision spectrum shaping gain is calculated. The pre-rotation of the LPC parameter is the same as that in the FFT-based MDCT implementation. The pseudocode is as follows:

```
for (i = 0; i < lpcOrder + 1; i++) {
    tmp = i * PI / 512
    realPart[i] = weightedLpc[i] * cos(tmp)
    imagPart[i]= -weightedLpc[i] * sin(tmp)
}
```

The number of complex FFT points is 512. For the low-precision spectrum shaping gain calculation, refer to formula (E.2).

$$rawLpcGain = \frac{1}{\|FFT(rotateLpc)\|_2} \quad \text{(E.2)}$$

The number of points of the obtained low-precision spectrum shaping gain is 256. Thirdly, a spectrum shaping gain of interpolation is obtained.

4-fold linear interpolation is performed on *rawLpcGain* to obtain the 1024-point interpolation spectrum shaping gain, which is denoted as interpretLpcGain.

Finally, intra-subband averaging is performed on the interpolation spectrum shaping gain to obtain the final LPC shaping gain.

3) Spectrum shaping

The MDCT spectrum is divided by the LPC spectrum shaping gain lpcGain to obtain the shaped MDCT spectrum.

E.2.4 Temporal Noise Shaping

Temporal noise shaping is used to control distribution of quantized noise introduced in the coding process in time domain, mitigate pre-echo and post-echo phenomena, and improve coding quality.

The input of the temporal noise shaping module is the MDCT spectrum after the frequency-domain noise shaping. The temporal noise shaping reduces a dynamic range of a to-be-coded MDCT spectrum through linear prediction processing in the MDCT domain, thereby improving quantization coding efficiency. The process of temporal noise shaping is as follows:

a) TNS filter configuration

TNS processing is performed on the MDCT spectrum. The entire MDCT spectrum is divided into two filters, respectively covering [660 Hz, 5400 Hz] and [5400 Hz, 20000 Hz].

Both the two TNS filters can be disabled or enabled, or only one of the two TNS filters is enabled. This is determined according to the TNS filter enabling algorithm.

b) TNS parameter calculation

The parameter used by the TNS filter is the reflection coefficient parcor (the maximum order is 8), which is an equivalent representation of parameters such as LPC and LSF. The reflection coefficient is separately calculated for the two TNS filters.

The basic calculation process is as follows:

1) Calculate a band-based normalized frequency-domain autocorrelation coefficient:

A frequency range corresponding to each TNS filter is divided into three frequency bands with an equal width, and spectrum energy in the three frequency bands is separately calculated.

Autocorrelation coefficients in the three frequency bands are separately calculated, and corresponding spectrum energy is used to perform normalization.

The sum of the normalized autocorrelation functions in the three frequency bands is used as the autocorrelation coefficient of the frequency range of the current filter.

2) Calculate the reflection coefficient:

The autocorrelation coefficients corresponding to the frequency ranges of the two TNS filters are used as the inputs, and the LeRoux-Gueguen algorithm is used to calculate the reflection coefficient parcor.

c) TNS filter enabling determining

The TNS filter is enabled based on the following conditions:

1) The filter order is greater than 0;
2) The mean square value of the Parcor parameter is greater than a threshold (set to 0.06), and the predicted gain of the filter is greater than a threshold (set to 1.35);
3) The current frame is a short frame.

When condition 1 is met and either condition 2 or 3 is met, the TNS filter is enabled.

d) TNS parameter quantization coding

The bitstream parameters of the TNS module include the TNS filter enabling flag, filter order, and reflection coefficient.

The non-uniform scalar quantization is used as the reflection coefficient. The number of quantization orders is 16. For details, refer to Table B.33.

Huffman coding is performed on the quantization index of the reflection coefficient, and each dimension of the reflection coefficient has a corresponding Huffman code table.

e) TNS filtering

In a frequency range corresponding to the TNS filter, if the TNS filter is enabled, linear prediction analysis filtering based on a reflection coefficient is performed on the MDCT spectrum within the frequency range, to obtain the MDCT spectrum after the TNS processing.

Short frame TNS processing: If the frame type of the current frame is ONLY_SHORT_WINDOW, inverse interleaving processing needs to be performed before the foregoing TNS processing, and interleaving processing is performed again after the TNS processing.

E.2.5 Bandwidth Extension Coding

At the encoder, the bandwidth extension algorithm is used to calculate a bandwidth extension parameter based on the correlation between high and low frequencies of an MDCT spectrum, so as to reconstruct a high-frequency spectrum at the decoder based on a core band spectrum obtained through decoding.

The enabling conditions and configuration parameters of bandwidth extension are determined by the input signal format and coding rate.

The parameters extracted through bandwidth extension coding include the high-frequency band envelope and whitening level. For details about the bitstream syntax, refer to section 7.9.1.

Calculation of high-frequency envelope parameters:

High-frequency band (SFB) division manner is used to calculate the MDCT spectrum energy of each SFB in the high frequency band, which is used as the high-frequency sub-band envelope parameter of bandwidth extension.

For the calculation method, refer to formula (E.3).

$$sfbEnvelope = \frac{1}{sfbWidth}\sum_{i=sfbStart}^{sfbEnd-1}(mdctSpectrum[i])^2 \quad \text{(E.3)}$$

sfbStart is the start frequency sequence number of the current sfb;

sfbEnd is the end frequency sequence number of the current sfb;

sfbWidth is the width of the current sfb (that is, the number of frequencies), $sfbWidth = sfbEnd - sfbStart + 1$.

A non-uniform scalar quantization manner is used for quantization of the high-frequency envelope parameter. For the quantization manner, refer to formula (E.4).

$$sfbEnvQIdx = floor\big(0.5 + 4.24966 * (log2(sfbEnvelope) + 4.0)\big) \quad \text{(E.4)}$$

floor() indicates rounding down.

The value of the envelope parameter quantization index *sfbEnvQIdx* is limited within the range of [0, 127], and 7-bit transmission is used.

High-frequency spectrum whitening level determining:

The whitening level of the high-frequency spectrum is determined within each target frequency area (target tile).

The whitening level indicates a type of post-processing that needs to be performed after the decoder copies the spectrum from the source frequency area (source tile). Whitening levels are classified into BWE_WHITENING_OFF, BWE_WHITENING_MID, and BWE_WHITENING_HIGH, corresponding to different post-processing manners. For details, refer to section 7.9.3.3.

The high-frequency spectrum whitening level is determined based on spectrum features of the source frequency area and the target frequency area, and the spectrum features include the SFM parameter and the PAR parameter.

The method for calculating the spectrum flatness measure SFM parameter is described as follows:

First, an average value of the log energy spectrum logEnerSpec and the energy spectrum enerSpec in the frequency band is calculated. A value of each frequency of the energy spectrum is energy of each frequency of the MDCT spectrum, and a value of each frequency of the log energy spectrum is a logarithm of energy of each frequency of the MDCT spectrum.

An average value of the log energy spectrum is denoted as avgLogEnerSpec, and an average value of the energy spectrum is denoted as avgEnerSpec. For calculation of the SFM parameter, refer to formula (E.5).

$$sfm = min(1.0, pow(2.0, avgLogEnerSpec + 0.5)/avgEnerSpec) \quad \text{(E.5)}$$

PAR parameter is defined as a ratio of the maximum value *maxLogEnerSpec* to the average value *avgLogEnerSpec* of the log energy spectrum logEnerSpec in the current frequency band. Refer to formula (E.6).

$$par = \frac{maxLogEnerSpec}{avgLogEnerSpec} \qquad (E.6)$$

The SFM parameter and the PAR parameter are separately calculated for each frequency band (SFB) of the source frequency area and the target frequency area to obtain the following parameters:

- SFM and PAR parameters of each frequency band in the source frequency area, that is, sfmSrcSfb and parSrcSfb;

- SFM and PAR parameters of each frequency band in the target frequency area, that is, sfmTarSfb and parTarSfb.

The SFM and PAR parameters of each frequency band are averaged in each frequency band of the frequency area to obtain:

- SFM and PAR parameters of the source frequency area, that is, sfmSrcTile and parSrcTile;

- SFM and PAR parameters of the target frequency area, that is, sfmTarTile and parTarTile.

The normalized SFM parameter used for determining the final whitening level is expressed as follows:

- Source frequency area: sfmSrcTileNorm = sfmSrcTile/parSrcTile;

- Target frequency area: sfmTarTileNorm = sfmTarTile/parTarTile.

The condition for determining a whitening level of a frequency area is as follows:

- BWE_WHITENING_OFF, that is, the condition for no whitening is as follows:

sfmTarTileNorm < sfmSrcTileNorm or sfmTarTileNorm < 0.19;

- BWE_WHITENING_MID, that is, the condition for the medium whitening level is as follows:

sfmTarTileNorm >= sfmSrcTileNorm && sfmTarTileNorm < (sfmSrcTileNorm + 0.15) or

sfmTarTileNorm >= 0.19&& sfmTarTileNorm < 0.3;

- BWE_WHITENING_HIGH, that is, the condition for the high whitening level is as follows:

sfmTarTileNorm >= (sfmSrcTileNorm + 0.15) or sfmTarTileNorm > 0.4.

E.3 Downmixing

E.3.1 Stereo Downmixing and Bit Allocation

E.3.1.1 M/S Downmixing Determining

The M/S downmixing manner is used for the stereo mode. Before downmixing, whether to perform downmixing is determined based on the correlation between the MDCT spectra of the left and right channels.

M/S downmixing is determined based on the normalized cross-correlation coefficient *crossCorr* of the MDCT spectra of the left and right channels to the ratio *lrRatio* of the energy of the MDCT spectra of the left and right channels.

For a method for calculating the normalized cross-correlation coefficient, refer to formula (E.7).

$$crossCorr = \frac{<mdctSpecL, mdctSpecR>}{\sqrt{\|mdctSpecL\|_2 * \|mdctSpecR\|_2}} \qquad (E.7)$$

$< mdctSpecL, mdctSpecR >$ indicates the inner product of *mdctSpecL* and *mdctSpecR*;

mdctSpecL indicates the MDCT spectrum of the left channel;

mdctSpecR indicates the MDCT spectrum of the right channel.

For the method for calculating the ratio of the energy of the MDCT spectra of the left and right channels, refer to formula (E.8).

$$lrRatio = \sqrt{\frac{\|mdctSpecL\|_2}{\|mdctSpecR\|_2}} \quad \text{(E.8)}$$

The M/S downmixing is determined based on the window types of the left and right channels, the normalized cross-correlation coefficient, and the energy ratio, and is represented as the following pseudocode:

```
isMs = 0
if transformTypeL == transformTypeR && crossCorr > 0.3:
  if lrRatio < 3.0 && lrRatio > 1/3:
    isMs = 1
  end
end
```

transformTypeL and transformTypeR are the window types of the left and right channels, or referred to as transform types.

E.3.1.2 M/S Downmixing

When the M/S flag isMs is 1, the M/S downmixing is required.

Before M/S downmixing, energy equalization needs to be performed on the MDCT spectra of the left and right channels. The energy balancing is implemented based on the ILD parameter. For the ILD parameter calculation expression, refer to formula (E.9).

$$energyRatio = \frac{\sqrt{\|mdctSpecL\|_2}}{\sqrt{\|mdctSpecL\|_2} + \sqrt{\|mdctSpecR\|_2}} \quad \text{(E.9)}$$

4-bit uniform quantization is performed on the ILD parameter. For the quantization manner, see formula (E.10).

$$energyRatioQIdx = max(1, min(15, 16 * energyRatio + 0.5)) \quad \text{(E.10)}$$

For calculation of the quantized ILD parameter, refer to formula (E.11).

$$energyRatioQ = \frac{16}{energyRatioQIdx} - 1 \quad \text{(E.11)}$$

Energy equalization is performed based on the quantized ILD parameter. The pseudocode is as follows:

```
if energyRatioQ > 1.0:
  mdctSpecR *= 1.0/energyRatioQ
else:
  mdctSpecL *= energyRatioQ
end
```

M/S downmixing is performed on the MDCT spectra of the left and right channels after the energy equalization. For a method, refer to formula (E.12) and formula (E.13).

$$mdctSpecM = \frac{\sqrt{2}}{2}(mdctSpecL + mdctSpecR) \quad \text{(E.12)}$$

$$mdctSpecS = \frac{\sqrt{2}}{2}(mdctSpecL - mdctSpecR) \quad \text{(E.13)}$$

mdctSpecM indicates the spectrum of the M channel obtained through downmixing;

mdctSpecS indicates the spectrum of the S channel obtained through downmixing.

Before bits are allocated to the MDCT spectra of the two downmixed channels, a bit allocation ratio parameter bitsRatio needs to be determined. The bit allocation ratio is calculated based on the energy ratio energyRatioDownMix of the MDCT spectra of the two downmixed channels. The 3-bit uniform quantization is performed on the bitsRatio. The calculation pseudocode is as follows:

```
bitsRatioQIdx = 4
if energyRatioDownMix>= 0:
   bitsRatioQIdx = floor(8 * bitsRatio + 0.5)
   if isMs == 0:
     if bitsRatioQIdx <= 3:
       bitsRatioQIdx += 1
     else if bitsRatioQIdx >= 5:
       bitsRatioQIdx -= 1
     end
   end
end
```

When the M/S downmixing is not performed, if the bit allocation ratio is too large or too small, the bit allocation ratio is adjusted to be more even. For the quantized bit allocation ratio parameter *bitsRatioQ*, refer to formula (E.14).

$$bitsRatioQ = bitsRatioQIdx\ /\ 8 \quad \text{(E.14)}$$

The remaining bits can be allocated based on the quantized bit allocation ratio parameter *bitsRatioQ* to the two downmixed channels. The allocation ratio of the downmixed channel 1 is *bitsRatioQ*, and the allocation ratio of the downmixed channel 2 is (1 – *bitsRatioQ*).

E.3.2 Multi-channel Downmixing and Bit Allocation

E.3.2.1 Channel Coupling Algorithm

The channel coupling algorithm used in the multi-channel mode is referred to as the MCAC algorithm. The MCAC algorithm is used to determine the multi-channel signal coding mode parameter based on the signal correlation, including the pair parameter and the downmixing parameter, to effectively reduce the redundancy between channels and improve the multi-channel coding efficiency.

A multi-channel signal usually includes a low-frequency effect channel (that is, an LFE channel), and the low-frequency effect channel has a valid audio component only at a low frequency. The LFE channel does not participate in the queuing algorithm and the bit allocation algorithm in the multi-channel mode.

The basic principle of the multi-channel coupling algorithm is to maximize the sum of the normalized cross-correlation of the coupled channels. The basic steps of the coupling algorithm are as follows:

a) Calculate a normalized cross-correlation matrix of all channels participating in coupling.

   In a 5.1 multi-channel example, there are five channels participating in coupling, and a size of a normalized cross-correlation matrix is 5 × 5.

   The symmetry of the cross-correlation matrix is considered, and the autocorrelation values on the diagonal are excluded. The effective cross-correlation information is the upper triangular part of the cross-correlation matrix.

   Each value of the normalized cross-correlation matrix corresponds to a channel pair, and a channel pair that meets a certain condition may participate in a subsequent channel coupling algorithm.

   The filtering condition includes:

   - The normalized cross-correlation coefficient of the channel pair is greater than 0.5;
   - The energy difference between the MDCT spectra of two channels in a channel pair is less than a threshold. The default threshold is 2.

b) Determine several coupled start channel pairs.

   It is assumed that the number of channel pairs with the largest cross-correlation coefficient used in the search is *numPairCondidate*, and the value of *numPairCondidate* is related to the number of channels. Refer to formula (E.15).

   $$numPairCondidate = (numChannels - 1)/2 \quad \text{(E.15)}$$

   numChannels indicates the total number of channels. *numChannels* – 1 indicates that the LFE channel is excluded.

   numPairCondidate channel pairs with a largest cross-correlation coefficient are selected from the normalized cross-correlation matrix obtained after screening, and are used as the coupled start channel pairs.

c) Determine the coupling manner.

   A channel pair with the largest correlation in the *numPairCondidate* coupled start channel pairs is used as an example. For example, in the 5.1 mode, the channel pair with the largest correlation is (0, 2). In a subsequent search process, all matrix elements related to the channel 0 and the channel 2 are excluded, and then a channel pair with the largest correlation value is found and used as a second channel pair, for example, (1, 4). The entire search process is completed by analogy, and the first group coupling manner is obtained.

It is assumed that the channel pair with the second largest cross-correlation is (0, 1). All matrix elements related to the channel 0 and the channel 1 are excluded in the subsequent search process, and then a channel pair with a largest correlation value is found as a second channel pair, for example, (2, 4), and so on, the entire search process is completed, and the second group of pair manners are obtained.

This search process is repeated, and *numPairCondidate* coupling manners may be obtained. A coupling manner with a largest cross-correlation sum is a finally determined channel coupling manner.

The content of object signals differs greatly between channels. Therefore, MCAC coupling is not performed for object signals.

E.3.2.2 Channel Pair Downmixing

In the multi-channel mode, intra–channel pair equalization is performed for energy equalization. The target amplitude value of the intra–channel pair equalization algorithm is an average value of amplitudes of two channels. The intra–channel pair equalization algorithm is implemented by performing ILD processing on the MDCT spectrum of the downmixed channel. In the ILD processing, the ratio of the MDCT spectrum amplitude to the equalization amplitude of the downmixed channel is used as an input to search for the ILD parameter in the quantization code table. The index mcIld[i] corresponding to the matched codeword is found based on the principle of minimum distance. Then, the spectrum is equalized. The pseudocode for equalization is as follows:

```
factor = mcIldCodebook[mcIld[i]]
mdctSpectrum[i] = factor * mdctSpectrum[i]
```

factor is the amplitude adjustment factor corresponding to the ILD parameter of the i-th channel, mcIldCodebook is the quantization code table of the ILD parameter, mcIld[i] indicates the quantization index corresponding to the ILD parameter of the i-th channel, and mdctSpectrum[i] indicates the MDCT coefficient vector of the i-th channel.

E.3.2.3 Silencing

E.3.2.3.1 Silent Channel Detection

Steps of silent channel detection are as follows:

a) The average energy of each channel signal of the current frame is calculated.

It is assumed that the frame length is FRAME_LEN, and the energy of the ch-th channel of the current frame is shown in formula (E.16).

$$energy(ch) = \frac{1}{FRAME\_LEN} \cdot \sum_{i=0}^{FRAME\_LEN-1} (orig_{ch}(i))^2 \quad \text{(E.16)}$$

$orig_{ch}$ is the input signal of the ch-th channel of the current frame;

energy(*ch*) is the energy of the ch-th channel of the current frame.

b) A silence detection parameter of each channel of the current frame is determined based on energy of each channel signal of the current frame.

The silence detection parameter of each channel of the current frame is calculated, and the silence detection parameter of each channel of the current frame meets formula (E.17).

$$energyDB[ch] = 10 * log10(energy[ch] / Bit\_Depth / Bit\_Depth) \quad \text{(E.17)}$$

energyDB[*ch*] is the silence detection parameter of the ch-th channel of the current frame;

energy(*ch*) is the energy of the ch-th channel of the current frame;

Bit_Depth is the full offset value of the bit width. For example, if the sampling bit depth is 16 bits, the full offset value of the bit width is $2^{16} = 65536$.

c) The silence flag of each channel of the current frame is determined based on the silence detection parameter and the silence detection threshold g_MuteThrehold of each channel of the current frame.

The silence detection parameter of each channel of the current frame is compared with the silence detection threshold. If the silence detection parameter of the ch-th channel of the current frame is less than the silence detection threshold, the ch-th channel of the current frame is a silent frame, that is, the ch-th channel is a silent channel at the current moment, and the silence flag silFlag [ch] of the ch-th channel of the current frame is 1. If the silence detection parameter of the ch-th channel of the current frame is greater than or equal to the silence detection threshold, the ch-th channel of the current frame is a non-silent frame, that is, the ch-th channel of the current frame is a non-silent frame, and the silence flag silFlag [ch] of the ch-th channel of the current frame is 0.

The pseudocode for determining the silence flag of the ch-th channel of the current frame based on the silence detection parameter and the silence detection threshold of the ch-th channel of the current frame is as follows:

```
silFlag [ch] = 0;
        if (energyDB[ch] < g_MuteThrehold)
{ silFlag [ch] = 1; }
```

E.3.2.3.2 Silence Side Information Coding

The side information of the silent frame includes silence enabling flag HasSilFlag and the silence flag silFlag[ch]:

- When the silent channel processing function is disabled, only the silence enabling flag HasSilFlag needs to be coded and the value of HasSilFlag is 0. The silence flag silFlag[ch] does not need to be coded.

- When the silent channel processing function is enabled and the silence flags silFlag[ch] of all channels participating in coupling are 0, it indicates that all the channels participating in coupling are non-silent channels. In this case, only the silence enabling flag HasSilFlag needs to be coded and the value of HasSilFlag is 0, and the silence flag silFlag[ch] does not need to be coded.

- When the silent channel processing function is enabled and the silence flags silFlag[ch] of all channels participating in coupling are not all 0, it indicates that there is a non-silent channel in the channels participating in coupling. In this case, the silence enabling flag HasSilFlag needs to be coded and the value of HasSilFlag is 1, and the silence flag silFlag[ch] needs to be coded.

E.3.2.3.3 Bit Allocation of Silence Channel

Silent frames are not involved in the bit allocation algorithm. Eight bytes are fixedly allocated to the channel of the silent frames.

The bit allocation algorithm supports three types of input: a sound bed signal, an object signal, and a sound bed and object mixed signal. In the bit allocation algorithms for the sound bed signal and the object signal, only some steps of the bit allocation algorithm for the sound bed and object mixed signal need to be performed.

a) Obtain the ratio factor bedBitsRatio of the sound bed to the total number of bits (only for the mixed signal):

1) When the total bit rate is 384 kb/s, the value of *bedBitsRatio* is fixed to 0.583335280.
2) When the total bit rate is 768 kb/s, the value of *bedBitsRatio* is fixed to 0.507143021.
3) Except for case 1) and case 2), the value of the ratio factor *bedBitsRatio* of the sound bed to the total number of bits is calculated based on the number of sound bed signal channels *bedChNum* and the number of object signal channels *objChNum*. Refer to formula (D.18).

$$bedBitsRatio = bedChNum \,/\, (bedChNum + objChNum) \quad \text{..................(E.18)}$$

b) Update the allocation strategy mixAllocStrategy for mixed signal (only for the mixed signal).

mixAllocStrategy is used to identify the allocation direction of the number of bits reduced by the silent channel of the object signal. There are two allocation strategies for the mixed signal: MIX_ALLOC_STRATEGY_INTERNAL indicates that the number of bits is allocated to another object signal, and MIX_ALLOC_STRATEGY_OBJ2BED indicates that the number of bits is allocated to the sound bed signal.

When the MIX_ALLOC_STRATEGY_INTERNAL strategy is used:

1) The silent frame processing function is disabled;
2) The input is a non-mixed signal, that is, there is only a sound bed signal or only an object signal;
3) The number numBedNoSil of non-silent channels of the sound bed signal is less than the number safetyChNum of safety channels.

When the MIX_ALLOC_STRATEGY_OBJ2BED strategy is used: The number numBedNoSil of non-silent channels of the sound bed signal is greater than or equal to the number safetyChNum of safety channels. The number numBedNoSil of non-silent channels of the sound bed signal is the number of channels whose silent flag silFlag[ch] is 0 in the channels participating in coupling. The number safetyChNum of safety channels is calculated based on the total bit rate totalBitrate, the frame length FRAME_LEN, and the sampling rate AVS3_SAMPLING_48KHZ.

c) Pre–bit allocation processing:

The pre–bit allocation processing is used to calculate the bit allocation ratio factor chBitRatios of the mixed signal and the non-mixed signal. The following steps are performed.

1) The energy ratio chBitRatiosFloat[ch] of each channel is calculated based on the channel energy.
2) The total number objBits of bits of the object is calculated based on the total number totalBits of bits and the ratio factor bedBitsRatio of the sound bed to the total number of bits (only for the mixed signal).
3) The estimated number objAvgBytes[ch] of bits of the object signal is calculated based on the total number objBits of bits of the object, the number objNum of object signals, the allocation strategy mixAllocStrategy, and the silence flag silFlag[ch] (for the mixed signal and the object signal).
4) The estimated number bedChBytes[ch] of bits of the sound bed signal is calculated based on the total number totalBits of bits, the total number objBits of bits of the object,

the channel energy ratio chBitRatiosFloat[ch], and the silence flag silFlag[ch] (for the mixed signal and the sound bed signal).

5) The bit allocation ratio factor chBitRatios is calculated based on the ratios of the estimated number bedChBytes[ch] of bits of the sound bed signal and the estimated number objAvgBytes[ch] of bits of the object signal to the total number of bits. The value of the bit allocation ratio parameter ranges from 0 to 1. The 6-bit even quantization is performed on the parameter.

E.3.3 HOA Downmixing and Bit Allocation

The coder determines the HOA downmixing configuration based on the HOA signal order and the bit rate. For related information, refer to Table 23 to Table 25. For example, when the coded signal is a 3-order HOA and the bit rate is 512 kb/s, the value of nTotalChanGroups is 2, indicating that the downmixed HOA channel is divided into two groups, and the value of groupChans is 2, 10, indicating that the number of channels in the downmixed channels of the virtual speaker signal group is 2, and the number of channels in the downmixed channels of the residual signal group is 10. The inter-group bit allocation ratio parameter and intra-group bit allocation ratio parameter of the downmixed channel may be calculated by using a method similar to that of multi-channel bit allocation.

E.4 Neural Network Transform and Quantization Coding

E.4.1 MDCT Quantization Coding based on Neural Network

The MDCT quantization coding based on the neural network includes spectrum grouping, neural network transform, entropy coding, noise filling parameter extraction, and quantization.

Figure E.10 is a schematic principle diagram of MDCT quantization coding based on the neural network.

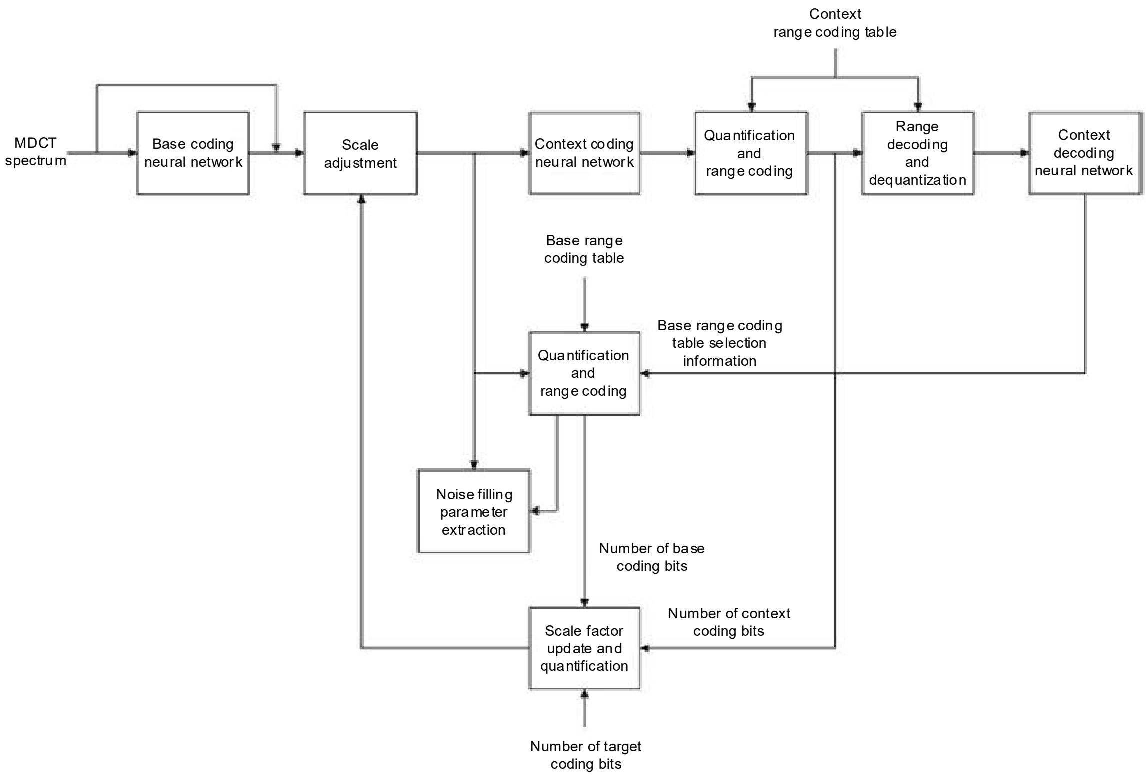
The basic process is as follows:

a) In the basic configuration, the MDCT spectrum after spectrum grouping is input to the base coding neural network to obtain a transform-domain parameter of the base coding neural network, and scale adjustment is performed on the transform-domain parameter. The transform-domain parameter after scale adjustment is input to the context coding neural network. Context information quantization and range coding are performed on the transform-domain coefficient output by the context coding neural network to obtain a context range coded bitstream. After the context range coded bitstream is decoded and dequantized, code table selection information of base range coding is obtained through context decoding neural network transform. Quantization and range coding are performed on the transform-domain parameter after scale adjustment based on the code table selection information and the base range coding table, to obtain a base range coded bitstream.

b) In a low-complexity configuration, scale adjustment is performed on an MDCT spectrum after spectrum grouping. An MDCT spectrum obtained through scale adjustment is input to a context coding neural network. Context information quantization and range coding are performed on a transform-domain coefficient output by the context coding neural network, to obtain a context range coded bitstream. After the context range coded bitstream is decoded and dequantized, code table selection information of base range coding is obtained through context decoding neural network transform. Quantization and range coding are performed on the MDCT spectrum after scale adjustment based on the code table selection information and the base range coding table, to obtain a base range coded bitstream.

In an MDCT quantization coding process based on the neural network, a scale factor needs to be updated based on the target number of coding bits, the number of base range coding bits, and the number of context range coding bits. This coding process is repeated until the number of coding bits is less than or equal to the target number of coding bits.

Figure E.10 Schematic principle diagram of MDCT quantization coding based on neural network



E.4.2 Spectrum Grouping

A basic processing manner of spectrum grouping is: If the window type of the current frame is a short window, grouping is determined on MDCT spectra of eight short frames of the current frame. If grouping is required, the MDCT spectra are divided into a transient group and a non-transient group. Spectrum interleaving is performed in each group, and an MDCT spectrum with a length of 1024 is obtained through concatenation and input into the neural network transform algorithm.

The spectrum grouping can effectively improve the coding efficiency of the spectra of several subframes belonging to the transient group in the eight subframes of the short frame, and can respectively extract the noise filling parameters from the transient group and the non-transient group to improve the noise filling effect.

The coding parameter of the spectrum grouping includes two parts: the number of groups and the grouping flag information. The number of groups indicates the number of spectrum groups of the current frame, and the grouping flag information indicates the groups to which the respective spectra of the eight subframes of the current frame belong.

a) Method for obtaining spectrum grouping information

If the window type of the current frame is a long window, a cut-in window, or a cut-out window, the group number parameter numGroups is set to 1, and the grouping flag information of all the eight subframes is 0.

If the window type of the current frame is a short window, the number of groups and the grouping flag parameter need to be determined based on the MDCT spectra of the eight subframes. The obtaining method is described as follows:

The MDCT spectra of the eight interleaved short frames of the current frame are de-interleaved, to obtain de-interleaved MDCT spectra.

Whether the MDCT spectra of the eight subframes should be divided into two groups, and the group to which each subframe belongs are determined based on the de-interleaved MDCT spectra.

First, spectrum energy of the de-interleaved MDCT spectra of eight subframes is calculated, and is denoted as blockEner[8];

Secondly, the sum *sumBlockEner* of spectrum energy and the maximum value *maxBlockEner* of each subframe are calculated, and the average spectrum energy avgBlockEner with the maximum subframe energy removed is obtained accordingly, that is, the subframe with the maximum energy is excluded from the calculation of the average spectrum energy, which is expressed as formula (E.19).

$$avgBlockEner = \frac{sumBlockEner - maxBlockEner}{numBlock - 1} \qquad \text{(E.19)}$$

numBlock is the number of subframes (or referred to as sub-blocks) in a short frame, and is set to 8.

Thirdly, the spectrum energy of each subframe is compared with the average spectrum energy, and the subframe whose spectrum energy is more than several times greater than the average spectrum energy belongs to the transient group; otherwise, the subframe belongs to the non-transient group.

if all subframes belong to the transient group or all subframes belong to the non-transient group, a group number parameter numGroups is set to 1, and grouping flag parameters of all eight subframes are set to 0.

Otherwise, the group number parameter numGroups is set to 2, the grouping flag parameter of the subframe belonging to the transient group is set to 0, and the grouping flag parameter of the subframe belonging to the non-transient group is set to 1.

b) Short frame MDCT spectrum grouping

If the current frame is a short frame and the group number parameter numGroups = 2, the MDCT spectra of the eight subframes in the short frame need to be grouped.

Step 1: Group and arrange the de-interleaved MDCT spectra to obtain the grouped and arranged MDCT spectrum with the length of 1024 frequencies. The MDCT spectrum of the subframe belonging to the transient group is placed before the MDCT spectrum of the subframe belonging to the non-transient group.

For example, it is assumed that the grouping flag information of the eight subframes is 1 0 0 0 1 1 0 0, that is, spectra of subframes 1, 5, and 6 belong to the non-transient group, and spectra of subframes 2, 3, 4, 7, and 8 belong to the transient group. Therefore, the MDCT spectra after grouping and arrangement may be represented as 2 3 4 7 8 1 5 6, that is, the spectra of subframes 2, 3, 4, 7, and 8 are placed in the front part of the spectra after grouping and arrangement, and the spectra of subframes 1, 5, and 6 are placed in the rear part of the spectra after grouping and arrangement.

Step 2: Perform intra-group interleaving on the grouped and arranged MDCT spectra. In the example in section E.2.2, interleaving needs to be performed on spectra of subframes

2, 3, 4, 7, and 8, and interleaving needs to be performed on spectra of subframes 1, 5, and 6.

E.4.3 Base Coding Neural Network

The base coding neural network (BaseEncoder) performs the neural network transform on the input MDCT spectral coefficient to obtain the transform-domain coefficient for subsequent quantization and coding.

The basic composition unit of the base coding neural network is a CNN. For information related to the network structure configuration, refer to Table E.1.

Table E.1 Structure configuration of base coding neural network

Item	Value
Number of CNN layers	4
Convolution kernel size	5, 5, 5, 5
Stride	2, 2, 2, 2
Number of channels	2, 4, 8, 16
Activation function	GDN, GDN, GDN, None
HasBias	1, 1, 1, 0

In the table, Stride is the CNN convolutional step parameter, HasBias indicates whether there is an offset parameter, and None in the activation function indicates that there is no activation function at the current layer.

For related information of parameters of each CNN layer of the base coding neural network, refer to Table E.3 to Table E.15. The 1024-point MDCT spectrum is input into the base coding neural network. The dimension of the output transform coefficient obtained through neural network transform is 16 channels, and each channel has 64 coefficients.

E.4.4 Scale Adjustment

The scale adjustment module performs scale adjustment on the transform coefficient output by the base coding neural network, to enlarge or narrow a dynamic range of the transform coefficient, and adjust the number of bits required by range coding.

The parameter of the scale adjustment module is the feature scale adjustment factor (featureScale). The value range of featureScale is [0, 29.97]. The scale adjustment factor is multiplied by the transform coefficients output by the neural network one by one. If featureScale is less than 1, it is equivalent to performing amplitude attenuation on the transform coefficient. If featureScale is greater than 1, it is equivalent to performing amplitude amplification on the transform coefficient.

E.4.5 Context Coding Neural Network

The context coding neural network (ContextEncoder) extracts the context information (referred to as the base range coding table selection information) of range coding from the transform coefficient obtained through scale adjustment, to guide range coding of the transform coefficient output from the base coding neural network.

The input of ContextEncoder is the absolute value of the base coding neural network transform coefficient obtained through scale adjustment. The composition unit of ContextEncoder is a CNN. For the structure configuration information, refer to Table E.2.

Table E.2 Structure configuration of context coding neural network

Item	**Value**
Number of CNN layers	3
Convolution kernel size	3, 3, 3
Stride	1, 2, 2
Number of channels	16, 16, 16
Activation function	ReLu, ReLu, None
HasBias	1, 1, 0

For the parameters of CNNs at each layer of the context coding neural network, refer to Table E.16 to Table E.20. The input of the context coding neural network is the transform coefficient with 16 channels, and each channel has 64 dimensions. The transform coefficient of the context coding neural network obtained through the neural network transform still has 16 channels, and each channel has 16 dimensions.

E.4.6 Context Quantization and Range Coding

The scalar quantization manner is used for the transform coefficient of the context coding neural network.

Range coding of the transform coefficient quantization index is performed based on several groups of pre-trained range coding tables, and each channel of the transform coefficient of the context coding neural network corresponds to one range coding table. For the code tables used in range coding, refer to Table B.1.

The bitstream obtained through range coding is referred to as the context coded bitstream, and the number of bits occupied by the bitstream is referred to as the number contextNumBytes of context coding bits. A byte is used as the minimum unit for range coding.

E.4.7 Base Quantization and Range Coding

The scalar quantization manner is used for the transform coefficient of the base coding neural network obtained through scale adjustment.

The range coding is performed based on several preset groups of range coding tables. For the code tables used in base range coding, refer to Table B.9. The range coding tables are a series of cumulative probability density tables (CDF) of Gaussian distribution with different standard deviation information. The number of range coding tables is 64 (denoted as numContextScale), and a corresponding standard deviation is denoted as contextScale.

Base range coding table selection information obtained through decoding is used to determine which one of these groups of range coding tables is used to code each dimension of the transform coefficient of the base coding neural network, that is, to determine a CDF index.

The CDF index of the base range coding is denoted as baseCdfIndex, and its determining process is described as the following pseudocode:

```
for i = 0 to baseNumLatentEncode-1:
  for j = 0 to baseNumLatentChannels-1:
    for index = 0 to numContextScale – 1:
      if contextScale[index] >= ctxDecOutput[i][j]
       baseCdfIndex[i][j] = index
       break
      end
    end
    if index == numContextScale:
      baseCdfIndex[i][j] = numContextScale - 1
    end
  end
end
```

ctxDecOutput is base range coding table selection information obtained through decoding, baseNumLatentEncode is a dimension of the transform-domain coefficient of the base neural network, and baseNumLatentChannels is the number of channels of the transform-domain coefficient of the base neural network.

For each dimension of the transform coefficient of the base coding neural network, a minimum standard deviation that is greater than or equal to the base range coding table selection information is searched for in a variance table of the base range coding, and an index corresponding to the minimum standard deviation is an index of the code table used by the range coding.

The decoder also obtains the range coding table index from the output of the context decoding neural network in a same manner.

A bitstream obtained through range coding is referred to as a base coded bitstream, and the number of bits occupied by the bitstream is referred to as the number baseNumBytes of base coded bits.

E.4.8 Scale Factor Update and Quantization

E.4.8.1 Scale Factor Update Strategy

A final objective of iterative update of the scale factor is to adjust the transform coefficient of the base coding neural network, so that the number of context coding bits and the number of base coding bits are less than or equal to the target number of coding bits.

The target number of coding bits is denoted as *targetBytes*, and a target of scale factor update may be represented as:(*contextNumBytes* + *baseNumBytes* < *targetNumBytes*.

In the iterative update of the scale factor, the following needs to be recorded: the upper limit (upBoundBytes) of the number of coding bits and the corresponding upper limit (upBoundScale) of the scale factor, and the lower limit (lowBoundBytes) of the number of coding bits and the corresponding lower limit (lowBoundScale) of the scale factor. The dichotomy is used to update the scale factor.

It is assumed that the current number of coding bits is currBytes, and the current scale factor is currScale. In this case, the manner of updating the upper limit of the number of coding bits, the upper limit of the scale factor, the lower limit of the number of coding bits, and the lower limit of the scale factor is shown in the following pseudocode:

```
if currBytes < targetBytes:
 if lowBoundBytes < currBytes:
  lowBoundScale = currScale
  lowBoundBytes = currBytes
 else if lowBoundBytes == currBytes:
  lowBoundScale = max(lowBoundScale, currScale)
  end
else if currBytes > targetBytes:
  if upBoundBytes > currBytes:
   upBoundScale = currScale
   upBoundBytes = currBytes
  else if upBoundBytes == currBytes:
   upBoundScale = min(upBoundScale, currScale)
  end
end
```

The objective of updating is to update the upper limit and the lower limit of the scale factor, so that the upper limit and the lower limit of the number of bits are close to the target number of bits. The details are as follows:

a) If the current number of coding bits is less than the target number of bits: if the lower limit of the number of bits is less than the current number of bits, the lower limit of the scale factor is updated to the current scale factor, and the lower limit of the number of bits is updated to the current number of bits. If the lower limit of the number of bits is equal to the current number of bits, the lower limit of scale factor is updated to a larger value of the current scale factor and the lower limit of scale factor.

b) If the current number of coding bits is greater than the target number of bits: if the upper limit of the number of bits is greater than the current number of bits, the upper limit of the scale factor is updated to the current scale factor, and the upper limit of the number of bits is updated to the current number of bits. If the upper limit of the number of bits is equal to the current number of bits, the upper limit of the scale factor is updated to a smaller value of the current scale factor and the upper limit of the scale factor.

   The adjustment of the scale factor is divided into two phases. In phase 1, the current number (currBytes) of coding bits does not reach the target number (targetBytes) of coding bits. In phase 2, the current number of encoding bits has passed through the target number of coding bits. In phase 1, only one of the upper limit and the lower limit exists. In the phase 2, both the upper limit and the lower limit exist.

   If the target number of coding bits is not reached, it indicates that:

   - the number of coding bits is greater than the target number of coding bits (currBytes > targetBytes) at all times from the start of scale factor iteration to the current iteration;

   - the number of coding bits is less than the target number of coding bits (currBytes < targetBytes) at all times from the start of iteration to the current iteration. The purpose of updating the scale factor in phase 1 is to enable the current number of bits to pass through the target number of bits as soon as possible. For the method for updating the scale factor, refer to formula (E.20).

$$newScale = currScale * \left(\frac{targetBytes}{currBytes}\right)^2 \qquad (E.20)$$

newScale indicates the updated scale factor.

The purpose of update in phase 2 is to enable the current number of bits to converge between the upper limit and the lower limit of the number as soon as possible. For the method for updating the scale factor, refer to formula (E.21).

$$newScale = lowBoundScale + lowerFac\frac{targetBytes - lowBoundBytes}{upBoundBytes - lowBoundBytes}(upBoundScale - lowBoundScale) \qquad (E.21)$$

E.4.8.2 Quantization Method of Scale Factor

In the basic configuration, the segmented quantization method is used for the scale factor based on different values of the scale factors.

If the value of the scale factor is within the range of [0, 1], 7-bit uniform quantization is performed. For the quantization index expression, refer to formula (E.22).

$$scaleQIdx = floor(0.5 + 127.0 * featureScale) \qquad (E.22)$$

For the quantized scale factor expression, refer to formula (E.23).

$$featureScaleQ = scaleQIdx/127.0 \qquad (E.23)$$

If the value of the scale factor is greater than 1, 7-bit non-uniform quantization is performed. For the quantization index expression, refer to formula (E.24).

$$scaleQIdx = floor(0.5 + 86.0 * log10(featureScale)) \qquad (E.24)$$

The value of the quantization index is limited to [0, 127].

For the quantized scale factor expression, refer to formula (E.25).

$$featureScaleQ = 10.0^{scaleQIdx/86.0} \qquad (E.25)$$

The value range of the scale factor is specified by the isFeatAmplified parameter. The value 0 of the parameter indicates that featureScale is less than or equal to 1.0. Otherwise, featureScale is greater than 1.0.

In the low-complexity configuration, a logarithm domain uniform quantization method is used for quantization of the scale factor. For the quantization index expression, refer to formula (E.26).

$$scaleQIdxLc = \lfloor 31.875 * log10(featureScale) + 255 + 0.5 \rfloor \qquad (E.26)$$

The value of the quantization index is limited to [0, 255].

For the quantized scale factor expression, refer to formula (E.27).

$$featureScale = 10^{(scaleQIdxLc - 255.0)/31.875} \qquad (E.27)$$

The updated and quantized scale factor is used for scale adjustment in the next rate iteration.

The maximum number of rate iterations is set to 5. If the number of bits in the last iteration is greater than the target number of bits, a scale factor that is closest to and less than the target number of bits, a corresponding base coded bitstream, and a corresponding context coded bitstream are used.

E.4.9 Noise Filling Parameter Extraction

The noise filling parameter indicates an average error between the transform coefficients of the base coding neural network after scale adjustment and the quantized transform coefficients.

Before the noise filling parameter is extracted, the number of transform coefficients for noise filling needs to be determined, which is denoted as *numLatentNF*, and is determined based on the number *numLinesNonZero* of valid spectral lines input through quantization coding. The number of valid spectral lines is equal to the number of spectral lines of the core frequency band except the BWE frequency band.

In the main profile, the pseudocode for calculating *numLatentNF* is as follows:

```
numLatentNF = numLinesNonZero
for i = 0 to baseNumLayers-1:
  numLatentNF /= stride[i]
end
```

baseNumLayers indicates the number of layers of the base coding neural network (baseEncoder), and stride[i] indicates the stride parameter of the CNN at the i-th layer of the baseEncoder.

That is, the number *numLatentNF* of transform coefficients that require noise filling is equal to the number *numLinesNonZero* of spectral lines of the core band divided by the stride parameter of the CNN at each layer.

In the low-complexity configuration, for calculation of *numLatentNF*, refer to formula (E.28).

$$numLatentNF = numLinesNonZero \,/\, numLatentChannels \quad \text{(E.28)}$$

numLatentChannels indicates the number of channels of the neural network transform-domain coefficient. The range of the transform coefficient that requires noise filling is determined based on the spectrum grouping information.

If the spectrum group number parameter numGroups is 1, the start point startIdx of noise filling is 0, and the end point endIdx is numLatentNFC.

If the spectrum group number parameter numGroups is 2, noise filling is performed in two segments, one segment corresponds to the spectral coefficients in the transient group of the short frame, and the other segment corresponds to the spectral coefficients in the non-transient group of the short frame.

The number of subframes in the transient group is denoted as numTransientBlock, and the number of subframes in the non-transient group is denoted as numOtherBlcok. Therefore, the pseudocode of the transform coefficient range corresponding to the spectral coefficient of the transient group is:

```
startIdx[0] = 0
endIdx[0] = numTransientBlock * (numLatentNF / N_BLOCK_SHORT)
```

N_BLOCK_SHORT is the number of subframes in the short frame, and a value is 8. The transform coefficient range corresponding to the spectral coefficient of the non-transient group is as follows:

```
startIdx[1] = numTransientBlock * (numLatentDim / N_BLOCK_SHORT)
endIdx[1] = startIdx[1] + numOtherBlock * (numLatentNF / N_BLOCK_SHORT)
```

numLatentDim is the dimension of each channel of the transform coefficient output by the base coding neural network, and is 64 in the current configuration. The process of calculating the noise filling parameter is as follows:

a) For each channel of the transform coefficient of the base coding neural network, within the transform coefficient range corresponding to the transient group and the non-transient group, an average value of absolute values of quantization errors of transform coefficients quantized to 0 is calculated.

b) An average value of these quantization errors in all channels is calculated, and is used as the noise filling parameter *nfParam*.

Quantization of noise filling parameter:

a) 3-bit uniform scalar quantization is performed. For the quantization index expression, refer to formula (E.29):

$$nfParamQIdx = floor(0.5 + nfParam * 23.34) \quad \text{(E.29)}$$

The value of the quantization index is limited to [0, 7].

b) For the quantized noise filling parameter, refer to formula (E.30).

$$nfParamQ = nfParamQIdx/23.34 \quad \text{(E.30)}$$

E.5 HOA Spatial Coding

The HOA spatial coder may include sound field component analysis, sound field component synthesis, and other component calculation, and conversion from the HOA signal into a transmission channel signal and side information.

The sound field component analysis includes virtual speaker configuration, virtual speaker set generation, coding analysis, and virtual speaker selection.

The virtual speaker configuration is used to generate virtual speaker configuration parameters based on the coder configuration information, to obtain a plurality of virtual speakers. The coder configuration information includes but is not limited to an HOA order, a coding bit rate, and the like. The virtual speaker configuration parameters include but are not limited to: the number of virtual speakers, the orders of the virtual speakers, and the position coordinates of the virtual speakers. The position coordinates of the virtual speakers include the horizontal angles and the pitch angles. For the position coordinates in a non-uniform distribution of virtual speakers, refer to Table B.48.

For the virtual speaker set generation, refer to section 7.13.3.3.

The coding analysis is used to perform coding analysis on the HOA signal, and analyze a sound field distribution feature of the HOA signal, including features such as number of sound sources of the HOA signal, directivity of the sound sources, and diffuseness. Singular value decomposition method can be used for coding analysis to analyze the sound sources based on the singular values. For example, a ratio temp[i] between singular values is used, and the value of temp[i] is sequentially determined from i=0. When the m-th sound field classification parameter meets temp[m] ≥ TH1, the sound field is directional and there are (m+1) sound sources; if temp[m] ≥ TH1 does not exist, the sound field type is a diffuse sound field, and TH1 is a preset sound source determining threshold, which may be a constant, for example, a value of TH1 may be 30.

The virtual speaker selection is used to determine a target virtual speaker matching the HOA signal and attribute information of the target virtual speaker based on the HOA signal, the sound field distribution feature of the HOA signal, and the coefficients of the plurality of virtual speakers. A common method is to perform a projection operation based on the HOA signal and the HOA coefficient of the virtual speaker, and screen the main virtual speakers used to represent the sound field components. Refer to formula (E.31).

$$P_{jil} = B_{ji}(\theta, \varphi) \times B_l(\theta, \varphi) \qquad \text{(E.31)}$$

θ indicates the azimuth of the virtual speaker;

φ indicates the pitch angle of the virtual speaker;

$B_{ji}(\theta, \varphi)$ indicates the j-th HOA coefficient of the HOA signal in the i-th round of screening;

$B_l(\theta, \varphi)$ indicates the HOA coefficient of the *l*-th virtual speaker.

When the maximum value of P_{jil} is obtained, the *l* virtual speakers are the target virtual speakers of the HOA signal, and the horizontal angle information and the azimuth information of the virtual speakers are the attribute information of the target virtual speakers. The number of initial target virtual speakers may be set to 2.

The sound field component synthesis is mainly used to generate virtual speaker signals in the transmission channels. The virtual speaker signal is generated based on the HOA signal and the attribute information of the target virtual speakers. For details about the synthesis of virtual speaker signals, refer to formula (E.32).

$$w = A^{-1}X \qquad \text{(E.32)}$$

w indicates the virtual speaker signal;

A indicates the HOA coefficient of the target virtual speaker, and has a size of (M x C);

A^{-1} indicates the inverse matrix of matrix A;

X indicates the coefficient of the HOA signal, and has a size of (M x L);

C indicates the number of target virtual speakers;

M indicates the number of channels of the N-order HOA signal;

L indicates the number of coefficients of the HOA signal.

The other component calculation is used to obtain the residual signal in the transmission channel. The residual signal may be obtained based on a difference between the HOA signal and the reconstructed sound field signal. The reconstructed sound field signal is a sound field signal reconstructed based on the virtual speaker signal and the target virtual speaker. For implementation steps, refer to the HOA spatial decoding process in section 7.13. Finally, the virtual speaker signal and the residual signal form a transmission channel signal. The transmission channel signal and the side information represented by the virtual speaker are jointly coded by the core coder to obtain a coded bitstream.

E.6 Tables of Parameters of Coding Neural Network

For related information of parameters of each CNN layer of the coding neural network, refer to Table E.3 to Table E.20.

Table E.3 Layer 1 CNN (convolution kernel parameter kernel) of base coding neural network parameter

Parameter value
0x1.653c4a0000000p–7, –0x1.2df6900000000p–6,
–0x1.ea704a0000000p–7, 0x1.4e0c2c0000000p–6,
0x1.08852c0000000p–8, –0x1.d873b40000000p–11,
0x1.f20f720000000p–6, 0x1.34c98e0000000p–6,
0x1.7285d80000000p–9, 0x1.f5d3720000000p–10,

Table E.4 Layer 1 CNN (bias parameter bias) of base coding neural network parameter

Parameter value
0x1.1c2e720000000p–4, –0x1.9981880000000p–6,

Table E.5 Layer 1 CNN (GDN activation function beta parameter) of base coding neural network parameter

Parameter value
0x1.c855b20000000p+3, 0x1.2c9cc00000000p+4,

Table E.6 Layer 1 CNN (GDN activation function gamma parameter) of base coding neural network parameter

Parameter value
0x0.0p+0, 0x0.0p+0,
0x0.0p+0, 0x0.0p+0,

Table E.7 Layer 2 CNN (convolution kernel parameter kernel) of base coding neural network parameter

Parameter value
0x1.22a83c0000000p–3, 0x1.6042960000000p–3, 0x1.132b3a0000000p–3,

Parameter value
0x1.1510840000000p–2,
0x1.9c67040000000p–2, –0x1.ad7f9e0000000p–6, 0x1.f6abb60000000p–4, 0x1.3d6d2c0000000p–7,
0x1.5e1a420000000p–3, 0x1.78d15c0000000p–4, –0x1.7c56060000000p–3, 0x1.bd23b40000000p–3,
–0x1.74cd8c0000000p–2, –0x1.4a11720000000p–2, –0x1.fad6c00000000p–3, 0x1.6e17b60000000p–4,
0x1.5807d80000000p–4, –0x1.24e5620000000p–4, –0x1.691da40000000p–4, 0x1.0f94c20000000p–4,
0x1.7c700e0000000p–6, –0x1.ce30300000000p–11, 0x1.705f660000000p–3, –0x1.ee89be0000000p–3,
0x1.5406560000000p–4, 0x1.17d8f80000000p–7, –0x1.c443700000000p–6, 0x1.337cb80000000p–5,
–0x1.90a85e0000000p–5, 0x1.4a2aa60000000p–6, –0x1.43d3be0000000p–5, –0x1.8259300000000p–4,
–0x1.6e82060000000p–7, –0x1.41b5e60000000p–7, –0x1.8236620000000p–6, –0x1.7b40a40000000p–7,
–0x1.1a9c1e0000000p–4, –0x1.633cb40000000p–6, 0x1.0e4cf80000000p–4, 0x1.3e03da0000000p–5,

Table E.8 Layer 2 CNN (bias parameter bias) of base coding neural network parameter

Parameter value
–0x1.5894220000000p–8, –0x1.5b11d80000000p–9, –0x1.783dc80000000p–9, –0x1.f69d920000000p–10,

Table E.9 Layer 2 CNN (GDN activation function beta parameter) of base coding neural network parameter

Parameter value
0x1.cfa2680000000p+0, 0x1.425d460000000p+0, 0x1.a1cc0c0000000p–1, 0x1.84e8f40000000p+0,

Table E.10 Layer 2 CNN (GDN activation function gamma parameter) of base coding neural network parameter

Parameter value
0x1.bc167e0000000p–31, 0x0.0p+0, 0x1.1976f80000000p–32, 0x1.0926de0000000p–30,
0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0,
0x1.4e9dd00000000p–28, 0x0.0p+0, 0x1.9fac280000000p–38, 0x1.3399c20000000p–34,
0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0,

Table E.11 Layer 3 CNN (convolution kernel parameter kernel) of base coding neural network parameter

Parameter value
–0x1.3a932a0000000p–6, 0x1.ccae2e0000000p–10, 0x1.80cb680000000p–3, –0x1.714d440000000p–3, 0x1.8b8fd20000000p–10, –0x1.21c3800000000p–3, 0x1.308c720000000p–5, 0x1.7912dc0000000p–3,
–0x1.a6d08c0000000p–3, –0x1.23c6820000000p–8, –0x1.7ec64a0000000p–6, –0x1.8153da0000000p–3, –0x1.aae2920000000p–3, –0x1.5f67600000000p–4, –0x1.17b5520000000p–5, 0x1.8767a20000000p–5,
–0x1.2d0b740000000p–4, –0x1.760afe0000000p–4, 0x1.5cfdac0000000p–3, –0x1.84ef860000000p–5, –0x1.2a88fc0000000p–5, –0x1.49c2280000000p–3, 0x1.f163660000000p–6, 0x1.c563100000000p–4,
0x1.14d6d60000000p–3, 0x1.9056aa0000000p–7, 0x1.edead60000000p–5, 0x1.6c1f340000000p–5, 0x1.9b6a780000000p–4, –0x1.0dd4180000000p–5, 0x1.ddbee20000000p–6, 0x1.65926e0000000p–3,
–0x1.0ee9b60000000p–3, –0x1.8619380000000p–2, 0x1.c987ae0000000p–2, 0x1.aba5660000000p–3, 0x1.0212e40000000p–2, –0x1.98908e0000000p–2, 0x1.8b95f40000000p–3, 0x1.2cc4920000000p–4,
–0x1.c696c20000000p–2, –0x1.2a94460000000p–4, –0x1.28236c0000000p–4, –0x1.7fe2b00000000p–3, –0x1.2b63f40000000p–2, –0x1.27f4ec0000000p–3, 0x1.f764180000000p–4, –0x1.18befc0000000p–1,
0x1.2526900000000p–1, 0x1.d5aa580000000p–3, 0x1.0e85320000000p–4, –0x1.51ccac0000000p–5, 0x1.a18ec20000000p–4, 0x1.79064e0000000p–7, 0x1.1892980000000p–2, –0x1.8da1ce0000000p–3,
–0x1.cce77c0000000p–7, –0x1.b948520000000p–3, –0x1.5b65280000000p–3, 0x1.565d960000000p–1, 0x1.d700c00000000p–3, 0x1.130b120000000p–3, –0x1.04d9b20000000p–3, –0x1.20eb620000000p–4,
0x1.abac260000000p–4, –0x1.c8eb980000000p–5, 0x1.e836580000000p–2, 0x1.6b4d740000000p–4, –0x1.10469e0000000p–1, 0x1.2c9fd40000000p–3, –0x1.ec69e00000000p–3, –0x1.e710920000000p–3,

Parameter value
0x1.bca5460000000p–3, 0x1.af19540000000p–3, –0x1.7a2a400000000p–2, –0x1.8c92660000000p–4, –0x1.95c20e0000000p–2, –0x1.824db80000000p–2, –0x1.c3643c0000000p–5, 0x1.7982980000000p–3,

Table E.11 (continued)

Parameter value
–0x1.1b89220000000p–6, –0x1.4515580000000p–3, –0x1.bce3640000000p–4, –0x1.05f2dc0000000p–2, 0x1.57e9f60000000p–1, –0x1.53536c0000000p–3, –0x1.9cfed40000000p–3, –0x1.86c4e80000000p–4,
0x1.ea45a20000000p–3, –0x1.0d6ca40000000p–1, –0x1.1697ac0000000p–2, –0x1.0c642c0000000p–2, –0x1.29bf060000000p–3, 0x1.0901940000000p–3, 0x1.9251f80000000p–5, –0x1.fb8f980000000p–6,
0x1.d793880000000p–5, 0x1.ac848c0000000p–5, –0x1.5319f80000000p–3, 0x1.9100a40000000p–3, –0x1.fa51a80000000p–2, 0x1.c7a96a0000000p–5, 0x1.baa0b80000000p–2, 0x1.7bdbe00000000p–4,
0x1.ad0e440000000p–4, –0x1.3e29d20000000p–3, 0x1.ca83760000000p–4, 0x1.2454300000000p–4, –0x1.8ddfb60000000p–3, 0x1.395a2e0000000p–3, 0x1.9464fc0000000p–4, –0x1.2e13ce0000000p–6,
0x1.47e46e0000000p–3, –0x1.5c4f700000000p–4, –0x1.9f047c0000000p–4, 0x1.d5a4ee0000000p–4, –0x1.8b86ee0000000p–2, –0x1.7adf100000000p–3, –0x1.c4431e0000000p–5, –0x1.3bf2d40000000p–8,
0x1.b1b58e0000000p–3, –0x1.868ece0000000p–3, –0x1.3e04f80000000p–3, 0x1.8bbb9a0000000p–5, –0x1.161a460000000p–2, –0x1.08bfd00000000p–2, –0x1.d921240000000p–4, –0x1.467dda0000000p–6,
0x1.cf2d060000000p–7, –0x1.d6af500000000p–4, –0x1.eeeb480000000p–5, 0x1.04faa40000000p–3, –0x1.65ed500000000p–3, –0x1.37d4200000000p–5, –0x1.025d1c0000000p–7, 0x1.0f25b60000000p–5,
–0x1.8b21400000000p–3, 0x1.88fe340000000p–6, 0x1.45ddca0000000p–4, –0x1.da07560000000p–4, 0x1.a477480000000p–2, 0x1.e68c760000000p–4, –0x1.8524540000000p–4, –0x1.f7db780000000p–11,
–0x1.9377920000000p–5, –0x1.c9b9380000000p–4, 0x1.2947920000000p–7, 0x1.1c3d700000000p–3, –0x1.1d80e80000000p–3, 0x1.466b920000000p–7, –0x1.5890960000000p–4, 0x1.d99ca00000000p–6,
0x1.1f5e7a0000000p–6, –0x1.5ba5b80000000p–5, –0x1.4490840000000p–4, –0x1.cebfa60000000p–6, 0x1.71ce4a0000000p–4, –0x1.725bec0000000p–4, –0x1.6c43600000000p–5, 0x1.454cba0000000p–8,

Table E.12 Layer 3 CNN (bias parameter bias) of base coding neural network parameter

Parameter value
0x1.1020de0000000p–10, 0x1.0a7c0c0000000p–7, 0x1.e1953e0000000p–9, –0x1.2de4e20000000p–8, 0x1.9fa7720000000p–8, –0x1.a156500000000p–11, –0x1.b491540000000p–18, –0x1.27bb900000000p–10,

Table E.13 Layer 3 CNN (GDN activation function beta parameter) of base coding neural network parameter

Parameter value
0x1.d0d8fc0000000p–1, 0x1.becb2e0000000p–1, 0x1.3b81b00000000p–1, 0x1.dca0400000000p–1, 0x1.66fece0000000p+0, 0x1.b422b60000000p–2, 0x1.a465040000000p–1, 0x1.34fb960000000p–2,

Table E.14 Layer 3 CNN (GDN activation function gamma parameter) of base coding neural network parameter

Parameter value
0x1.94d3c00000000p–41, 0x0.0p+0, 0x0.0p+0, 0x1.34ee820000000p–24, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0,
0x0.0p+0, 0x1.040c120000000p–23, 0x0.0p+0, 0x1.3924620000000p–23, 0x1.43042c0000000p–21, 0x1.d3232a0000000p–28, 0x1.8898000000000p–30, 0x0.0p+0,
0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x1.8a89340000000p–30, 0x1.f394900000000p–39,
0x1.3ed34a0000000p–26, 0x1.9681e60000000p–29, 0x0.0p+0, 0x0.0p+0, 0x1.c2da620000000p–23, 0x0.0p+0, 0x1.75d6240000000p–31, 0x0.0p+0,
0x0.0p+0, 0x1.1e513a0000000p–32, 0x0.0p+0, 0x1.59a2b00000000p–25, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0,
0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0,
0x1.f72bfc0000000p–34, 0x1.40ffd60000000p–34, 0x1.ccc9100000000p–26, 0x1.2ca0c20000000p–22, 0x0.0p+0, 0x0.0p+0, 0x1.579bee0000000p–24, 0x1.a591000000000p–35,
0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x0.0p+0, 0x1.92c9240000000p–36, 0x1.2e93800000000p–34, 0x0.0p+0,

Table E.15 Layer 4 CNN (convolution kernel parameter kernel) of base coding neural network parameter

Parameter value
0x1.3397da0000000p–10, 0x1.8433d00000000p–7, 0x1.152e8a0000000p–5, –0x1.7766a20000000p–6, 0x1.93eb580000000p–7, 0x1.1427ac0000000p–10, 0x1.05abea0000000p–10, 0x1.7f9d400000000p–5, –0x1.0359920000000p–11, –0x1.947a260000000p–12, –0x1.11a51a0000000p–8, –0x1.e55b720000000p–7, –0x1.2146160000000p–3, 0x1.d1846e0000000p–12, –0x1.1ad2a20000000p–6, 0x1.435bce0000000p–5,
–0x1.8a68160000000p–11, 0x1.19dd6e0000000p–8, –0x1.3494720000000p–6, –0x1.4db3f00000000p–7, 0x1.4748b20000000p–8, 0x1.65ed780000000p–9, –0x1.9ddc5e0000000p–9, –0x1.df5c880000000p–6, 0x1.441fb60000000p–11, –0x1.b8a4880000000p–12, 0x1.7beae20000000p–8, 0x1.c24f640000000p–6, –0x1.faf3bc0000000p–5, –0x1.336ac00000000p–11, 0x1.12ce740000000p–5, –0x1.85e6180000000p–6,

Table E.15 (continued)

Parameter value
0x1.30544a0000000p–15, –0x1.e1fec60000000p–7, –0x1.32b0a40000000p–5, 0x1.accf460000000p–8, 0x1.ffbc080000000p–9, –0x1.0a3a0e0000000p–9, –0x1.ab52d00000000p–12, –0x1.94515e0000000p–5, 0x1.07e5600000000p–13, –0x1.42eaae0000000p–11, 0x1.5551ec0000000p–8, 0x1.0c170c0000000p–5, 0x1.3aea700000000p–6, –0x1.a182540000000p–13, –0x1.1f6a7a0000000p–9, –0x1.a9b0d20000000p–7,
–0x1.95b1f60000000p–13, 0x1.433bf40000000p–8, –0x1.1e17b00000000p–6, –0x1.ab7f420000000p–9, –0x1.a150400000000p–6, 0x1.e8e83c0000000p–10, –0x1.5ede280000000p–11, –0x1.1a5c220000000p–5, –0x1.2991fc0000000p–12, 0x1.f2f0080000000p–11, 0x1.da6c8e0000000p–9, 0x1.824a7a0000000p–7, 0x1.bef4ac0000000p–3, –0x1.3e62ee0000000p–11, –0x1.4feb900000000p–8, –0x1.a7842e0000000p–6,
–0x1.b8c0080000000p–10, –0x1.1eb5b60000000p–7, –0x1.74100e0000000p–9, 0x1.18e7740000000p–7, 0x1.d5fd860000000p–5, 0x1.d36b7c0000000p–14, 0x1.2297700000000p–9, 0x1.d4a7d40000000p–6, 0x1.496fe00000000p–14, –0x1.a955ac0000000p–12, –0x1.c510820000000p–10, –0x1.b2d8120000000p–6, –0x1.c5aafe0000000p–2, –0x1.62bfe40000000p–13, 0x1.1d88ac0000000p–5, 0x1.d4ccc20000000p–11,
0x1.894ee40000000p–12, –0x1.60a90c0000000p–6, –0x1.09ad1e0000000p–5, 0x1.2249460000000p–5, –0x1.54ec660000000p–6, –0x1.17b76a0000000p–8, –0x1.106ffe0000000p–12, –0x1.d640e00000000p–5, 0x1.2513b00000000p–10,

Parameter value
0x1.4354760000000p–14, 0x1.f1c07c0000000p–9, 0x1.74f8f60000000p–5, 0x1.2047ec0000000p–2, 0x1.9ff14c0000000p–12, 0x1.50efc00000000p–10, –0x1.7839a20000000p–6,
0x1.f0e0940000000p–12, –0x1.5e23640000000p–8, –0x1.9a4a040000000p–10, 0x1.50d2920000000p–5, –0x1.4651ae0000000p–4, –0x1.3da02a0000000p–9, –0x1.2dab180000000p–9, –0x1.c69d120000000p–5, 0x1.58fcac0000000p–9, 0x1.37bb140000000p–10, 0x1.179b060000000p–8, 0x1.dea4360000000p–5, 0x1.5bc8700000000p–1, 0x1.9a5c8c0000000p–11, 0x1.2d66920000000p–7, –0x1.d576020000000p–6,
–0x1.678a720000000p–17, 0x1.08afcc0000000p–9, –0x1.3369200000000p–8, 0x1.c1164a0000000p–8, –0x1.96a1a80000000p–6, 0x1.45053c0000000p–13, 0x1.8d1c620000000p–12, –0x1.72d4040000000p–7, 0x1.70bbb00000000p–12, 0x1.0b3f960000000p–11, 0x1.b0f29a0000000p–11, 0x1.e9c99e0000000p–12, 0x1.4299f40000000p–3, 0x1.21b6a80000000p–12, –0x1.c71b0e0000000p–10, –0x1.16d3d40000000p–6,
0x1.37f0d60000000p–8, –0x1.6f3c040000000p–5, 0x1.4b46820000000p–3, 0x1.687f480000000p–3, 0x1.03c9860000000p–1, –0x1.338c640000000p–5, –0x1.3e1ece0000000p–5, 0x1.8db7140000000p–2, 0x1.d79fc40000000p–6, 0x1.8233b40000000p–7, 0x1.3fe3460000000p–9, 0x1.2d59ca0000000p–2, 0x1.9206000000000p–2, –0x1.eebc3e0000000p–11, –0x1.395c9e0000000p–2, 0x1.33c3760000000p–4,

Table E.15 (continued)

Parameter value
–0x1.6884be0000000p–6, 0x1.3f046c0000000p–3, 0x1.8099680000000p–5, 0x1.011a8a0000000p–3, 0x1.84a9560000000p–4, 0x1.d6456a0000000p–6, 0x1.97cfdc0000000p–6, 0x1.12a13c0000000p–1, –0x1.26537e0000000p–7, –0x1.dc2e560000000p–10, –0x1.aa300a0000000p–6, 0x1.2da42c0000000p–5, 0x1.ac2e480000000p–3, –0x1.c3a01a0000000p–11, 0x1.21f5f20000000p–1, –0x1.35c6e60000000p–1,
0x1.0e068a0000000p–7, 0x1.a488da0000000p–7, –0x1.db32a60000000p–9, –0x1.1c951a0000000p–5, –0x1.12c25c0000000p–1, 0x1.5ffc2c0000000p–6, 0x1.9526040000000p–5, 0x1.2cc3b40000000p–1, –0x1.af202e0000000p–6, –0x1.ce0ac60000000p–8, –0x1.fc51b80000000p–8, –0x1.564fa00000000p–4, 0x1.f164140000000p–4, 0x1.09213c0000000p–8, –0x1.2a6c080000000p–3, 0x1.be0dd00000000p–8,
0x1.0a9de00000000p–11, 0x1.69a44c0000000p–6, –0x1.acf6700000000p–3, 0x1.8926020000000p–4, –0x1.3d15360000000p–2, 0x1.0a34f00000000p–7, 0x1.e752380000000p–7, –0x1.1f9a820000000p–2, –0x1.57c61c0000000p–7, –0x1.8e738e0000000p–9, 0x1.c1fd1c0000000p–7, 0x1.dece720000000p–3,

Parameter value
–0x1.c39f1a0000000p–5, –0x1.68dfee0000000p–12, 0x1.ec5ed20000000p–5, –0x1.e80e6c0000000p–2,
–0x1.b2428c0000000p–6, 0x1.cf2ad20000000p–4, 0x1.ae13000000000p–2, –0x1.32397c0000000p–3, –0x1.5f9d2e0000000p–4, 0x1.5bc4060000000p–7, –0x1.5da6d60000000p–7, –0x1.babeb60000000p–3, 0x1.afebf00000000p–8, 0x1.6cf7f40000000p–10, –0x1.c35d7c0000000p–5, –0x1.e1d99e0000000p–2, 0x1.7015360000000p–6, –0x1.0ba0720000000p–9, 0x1.0b3ca60000000p–1, 0x1.b1b4a40000000p–2,
0x1.2bbdf20000000p–6, –0x1.1e856c0000000p–4, –0x1.0696ea0000000p–2, –0x1.4243ec0000000p–2, –0x1.8805720000000p–2, 0x1.f1fc980000000p–6, 0x1.ba14fe0000000p–5, 0x1.82d4e80000000p–3, –0x1.899a800000000p–5, –0x1.45f58a0000000p–6, 0x1.2db58a0000000p–6, –0x1.7a29520000000p–2, 0x1.12b36e0000000p–3, 0x1.fee1a60000000p–9, –0x1.1ac2180000000p–2, 0x1.efab1e0000000p–3,
0x1.d876200000000p–7, –0x1.9b38aa0000000p–4, –0x1.6b4d960000000p–1, –0x1.a00dae0000000p–2, 0x1.1387b20000000p–1, 0x1.495d6e0000000p–5, 0x1.3292380000000p–5, –0x1.64d4680000000p–6, –0x1.c5a1b80000000p–5, –0x1.e02e1e0000000p–6, 0x1.c6cea80000000p–5, –0x1.a62fc20000000p–2, –0x1.4431920000000p–3, 0x1.bd31520000000p–11, 0x1.90772c0000000p–5, –0x1.8757b00000000p–3,
0x1.330b020000000p–9, –0x1.411fe80000000p–6, –0x1.51f92c0000000p–3, –0x1.8ef5040000000p–5, –0x1.1adcc60000000p–4, 0x1.d778280000000p–8, 0x1.f07dc40000000p–9, –0x1.55bfd80000000p–2, –0x1.d30fac0000000p–8, –0x1.216baa0000000p–8, 0x1.11243a0000000p–6, –0x1.10456a0000000p–8, 0x1.4d9eb00000000p–1, –0x1.5892900000000p–13, –0x1.1d39880000000p–8, –0x1.7cb6e80000000p–4,

Table E.15 (continued)

Parameter value
0x1.2878da0000000p–2, 0x1.2821560000000p–2, –0x1.75ff440000000p–2, 0x1.2b22a00000000p–1, –0x1.3c61660000000p–6, –0x1.57fc0a0000000p–4, –0x1.6b9b100000000p–2, 0x1.0645aa0000000p–3, –0x1.20d40a0000000p–2, –0x1.6ddafe0000000p–3, –0x1.0257a20000000p–1, –0x1.51bd780000000p–3, –0x1.c6879e0000000p–6, 0x1.a935e80000000p–5, –0x1.6e84d60000000p–3, 0x1.2865a80000000p–2,
–0x1.2066320000000p–1, 0x1.cfbdbe0000000p–3, –0x1.c4400e0000000p–3, –0x1.6690540000000p–3, 0x1.802e220000000p–3, 0x1.3ba2c80000000p–1, –0x1.1c9e100000000p–1, –0x1.8e6c160000000p–4, –0x1.489bdc0000000p–5, –0x1.0f74ba0000000p–3, –0x1.92ab800000000p–4, –0x1.31bdfa0000000p–9,

Parameter value
0x1.5a54f20000000p–10, –0x1.3ab7fa0000000p–3, –0x1.2324180000000p–3, –0x1.f8f89e0000000p–4,
0x1.3f70240000000p–3, –0x1.7fc8520000000p–3, –0x1.357b100000000p–3, 0x1.841eb80000000p–3, –0x1.8177080000000p–3, –0x1.115a160000000p–6, –0x1.2265fc0000000p–1, 0x1.2904680000000p–4, 0x1.982ea80000000p–4, 0x1.4d03100000000p–5, 0x1.0ed69e0000000p–1, 0x1.5943620000000p–3, –0x1.03bb300000000p–5, –0x1.59fa9c0000000p–7, 0x1.fb3b600000000p–2, 0x1.654bd20000000p–2,
–0x1.4193500000000p–4, –0x1.c9aa5c0000000p–4, 0x1.35537e0000000p–5, 0x1.0eb11a0000000p–1, –0x1.25d1d80000000p–3, 0x1.da35bc0000000p–2, 0x1.42572a0000000p–2, –0x1.07df960000000p–3, –0x1.ec34700000000p–3, –0x1.7e9d320000000p–4, 0x1.44ca300000000p–2, –0x1.99a3b80000000p–2, –0x1.23c90c0000000p–8, –0x1.351ed60000000p–6, 0x1.8d28620000000p–3, 0x1.25a1b60000000p–2,
–0x1.0989be0000000p–1, –0x1.76f8f80000000p–5, –0x1.48face0000000p–8, 0x1.812cdc0000000p–2, 0x1.135a220000000p–6, –0x1.b475dc0000000p–2, 0x1.f354040000000p–3, –0x1.b03c5a0000000p–5, 0x1.e5e2e00000000p–2, 0x1.0a565e0000000p–3, 0x1.ad21140000000p–4, –0x1.86160e0000000p–3, –0x1.5e14940000000p–6, –0x1.f3eeac0000000p–4, 0x1.47928c0000000p–3, 0x1.1336b60000000p–2,
0x1.f3e7d60000000p–3, 0x1.18edb20000000p–2, –0x1.cf80ea0000000p–4, 0x1.7167640000000p–3, –0x1.ad45220000000p–7, –0x1.074e300000000p–2, –0x1.79c60e0000000p–3, –0x1.778d6c0000000p–7, 0x1.86c7760000000p–2, 0x1.3de1280000000p–2, 0x1.8293ee0000000p–2, –0x1.1166d40000000p–3, 0x1.54cc3c0000000p–7, 0x1.1250cc0000000p–4, –0x1.f87c160000000p–2, –0x1.300bee0000000p–3,
–0x1.504e520000000p–4, –0x1.2e863a0000000p–2, 0x1.39b1ba0000000p–3, 0x1.24ef040000000p–3, –0x1.7787820000000p–2, 0x1.7604fe0000000p–3, 0x1.fdeee00000000p–5, –0x1.2e887c0000000p–3, 0x1.a680f40000000p–2, 0x1.a968900000000p–2, –0x1.1df04e0000000p–1, 0x1.2828a00000000p–7, –0x1.3948ac0000000p–6, 0x1.45dd540000000p–4, 0x1.622e880000000p–2, 0x1.0ee7320000000p–2,

Table E.15 (continued)

Parameter value
–0x1.18119c0000000p–6, 0x1.be7d060000000p–2, –0x1.537fb20000000p–1, –0x1.4fc14c0000000p–5, 0x1.70fc1c0000000p–9, 0x1.64217c0000000p–4, 0x1.69514a0000000p–2, –0x1.3813660000000p–6, –0x1.01da880000000p–10, 0x1.c7f2760000000p–5, 0x1.fb9be60000000p–5, 0x1.dcf3f40000000p–2,

Parameter value
–0x1.1ac8200000000p–4, 0x1.4c9ff60000000p–6, 0x1.3ffdd40000000p–5, 0x1.a525940000000p–3,
0x1.3b1f640000000p–3, –0x1.0d64900000000p–6, 0x1.9c622a0000000p–6, 0x1.0715d80000000p–9, –0x1.b36d540000000p–4, –0x1.0d7bd80000000p–2, –0x1.e518440000000p–4, –0x1.3e434c0000000p–7, 0x1.55027a0000000p–3, –0x1.144ca00000000p–1, 0x1.c09af20000000p–6, –0x1.15937c0000000p–6, –0x1.4ffbcc0000000p–11, –0x1.22db9a0000000p–2, –0x1.0f5cb80000000p–5, 0x1.a042740000000p–6,
0x1.24dd560000000p–3, –0x1.b8e4820000000p–9, 0x1.f9557c0000000p–12, 0x1.e310b60000000p–5, –0x1.6d8f300000000p–5, 0x1.20e5e20000000p–3, 0x1.b364b20000000p–4, –0x1.41d4900000000p–5, –0x1.2797da0000000p–9, 0x1.93833c0000000p–3, –0x1.8ba3cc0000000p–3, 0x1.f655f00000000p–7, –0x1.ca9fde0000000p–9, –0x1.f56a980000000p–3, –0x1.4ae2380000000p–6, 0x1.2e31520000000p–7,
–0x1.f04fb40000000p–2, –0x1.8ac6e80000000p–6, 0x1.fa11ea0000000p–7, –0x1.8a2f6a0000000p–4, 0x1.d5006a0000000p–5, –0x1.54b5bc0000000p–2, –0x1.3cb5340000000p–4, 0x1.98a5ac0000000p–4, –0x1.5d29340000000p–3, –0x1.4e841e0000000p–3, 0x1.77a26c0000000p–3, –0x1.2b3fc20000000p–5, 0x1.279e9c0000000p–7, 0x1.93cc0e0000000p–3, 0x1.9a10720000000p–5, –0x1.dde7aa0000000p–6,
–0x1.79f6d80000000p–3, 0x1.2274aa0000000p–9, 0x1.5044800000000p–10, –0x1.4360420000000p–4, –0x1.905db80000000p–8, –0x1.45ee160000000p–2, 0x1.08c0a60000000p–3, –0x1.817fb20000000p–5, 0x1.95fd1e0000000p–2, –0x1.1327840000000p–1, 0x1.0c0f5a0000000p–3, –0x1.1c646c0000000p–8, –0x1.5dde460000000p–8, 0x1.8e1ff80000000p–4, 0x1.1676b60000000p–6, 0x1.2fb4380000000p–7,
–0x1.434d3c0000000p–3, –0x1.07eafa0000000p–8, 0x1.75a4700000000p–6, –0x1.16a0160000000p–4, –0x1.459f920000000p–7, –0x1.162e820000000p–2, 0x1.99efc20000000p–5, –0x1.0d59a80000000p–7, 0x1.76a9140000000p–3, –0x1.a3c98a0000000p–2, 0x1.f42cc80000000p–9, –0x1.d1afea0000000p–7, –0x1.eeddc60000000p–10, 0x1.394b240000000p–5, 0x1.6b8a400000000p–7, 0x1.9ab77c0000000p–7,
0x1.0143900000000p–1, 0x1.fa47cc0000000p–8, –0x1.a168260000000p–8, 0x1.0189a00000000p–4, –0x1.51e7f80000000p–4, 0x1.271c540000000p–3, 0x1.492ce40000000p–6, –0x1.d0488a0000000p–5, 0x1.f464280000000p–4, –0x1.6e44ee0000000p–3, –0x1.7fcec60000000p–12, 0x1.9a63ca0000000p–9, –0x1.f7bf460000000p–9, –0x1.0861300000000p–2, –0x1.5369940000000p–5, 0x1.01a3c20000000p–6,

Table E.15 (continued)

Parameter value
–0x1.5948fe0000000p–2, –0x1.c24d160000000p–9, 0x1.4de2d20000000p–7, –0x1.70ddfc0000000p–4, 0x1.4cceb20000000p–4, –0x1.154d020000000p–2, 0x1.170b260000000p–3, 0x1.5097a80000000p–5, –0x1.93dc0a0000000p–7, –0x1.4410020000000p–3, 0x1.58f5f40000000p–2, –0x1.99e6aa0000000p–7, 0x1.bfd2f60000000p–9, 0x1.34d5ce0000000p–2, 0x1.6222e40000000p–5, –0x1.bdfeb40000000p–6,
–0x1.2be4c80000000p–4, –0x1.9b64100000000p–5, 0x1.c891c00000000p–5, 0x1.1d8f6c0000000p–5, –0x1.420cbe0000000p–4, –0x1.d74a620000000p–3, 0x1.1bc7fe0000000p–6, 0x1.89dca60000000p–4, –0x1.e4499c0000000p–2, 0x1.45b6de0000000p–5, –0x1.81cb0a0000000p–6, –0x1.2f70500000000p–5, 0x1.7162580000000p–7, –0x1.c4e78c0000000p–2, –0x1.69e43c0000000p–6, –0x1.2bf70e0000000p–6,
0x1.53d2ce0000000p–5, 0x1.bb17b20000000p–11, –0x1.0954a40000000p–14, 0x1.2f784a0000000p–9, 0x1.1db78e0000000p–9, –0x1.d435f60000000p–7, 0x1.6358400000000p–7, 0x1.fb23860000000p–13, 0x1.2f22120000000p–6, –0x1.1588d00000000p–6, 0x1.7b11aa0000000p–4, –0x1.2f661a0000000p–9, –0x1.a403d40000000p–15, 0x1.a979a00000000p–6, 0x1.6114040000000p–12, 0x1.c897920000000p–13,
0x1.464d940000000p–6, –0x1.78e9d60000000p–13, 0x1.3e25bc0000000p–9, –0x1.ada1700000000p–9, –0x1.07289c0000000p–10, –0x1.6d324e0000000p–6, 0x1.20aa300000000p–5, 0x1.6bdd380000000p–11, –0x1.0347660000000p–7, –0x1.9307060000000p–5, 0x1.4148160000000p–5, –0x1.15c1ee0000000p–9, 0x1.b93f760000000p–13, 0x1.0715080000000p–8, 0x1.57d9820000000p–12, –0x1.2dcf3a0000000p–9,
–0x1.e280b80000000p–6, 0x1.2f9ddc0000000p–10, –0x1.e863e60000000p–8, 0x1.516b5e0000000p–8, 0x1.1bc7ea0000000p–9, 0x1.80e7a00000000p–6, –0x1.3d1ec20000000p–4, –0x1.0618a20000000p–10, 0x1.2779d40000000p–6, 0x1.eb05980000000p–5, –0x1.21eada0000000p–4, 0x1.8cbf640000000p–9, –0x1.c48cde0000000p–12, 0x1.b88f640000000p–8, 0x1.b5f23e0000000p–12, 0x1.f5575e0000000p–8,
–0x1.382ec20000000p–7, 0x1.89194e0000000p–13, 0x1.c090620000000p–8, –0x1.595d840000000p–11, 0x1.b016c60000000p–12, –0x1.46502e0000000p–7, 0x1.7106820000000p–5, –0x1.06197e0000000p–10, –0x1.0a06f60000000p–7, 0x1.2dcfb80000000p–5, 0x1.c500d00000000p–11, –0x1.75ec400000000p–11, 0x1.6918d20000000p–13, –0x1.40a0d20000000p–7, –0x1.a934860000000p–11, –0x1.0f46ce0000000p–10,
–0x1.b29f020000000p–8, 0x1.1075040000000p–10, 0x1.4808a20000000p–9, 0x1.da1f480000000p–10, 0x1.9d197a0000000p–10, –0x1.dd81220000000p–10, 0x1.12819c0000000p–7, –0x1.0ae4b60000000p–10, 0x1.68f3dc0000000p–8, 0x1.360b740000000p–5, 0x1.ecb9aa0000000p–8, 0x1.8240760000000p–10, 0x1.fcb0320000000p–14, 0x1.1795b20000000p–8, –0x1.fcb58a0000000p–11,

Parameter value
0x1.81e9740000000p–10,

Table E.15 (continued)

Parameter value
0x1.1e0aea0000000p–5, 0x1.eb29dc0000000p–13, 0x1.c2784a0000000p–10, –0x1.4b65860000000p–9, –0x1.33f54a0000000p–9, –0x1.2304cc0000000p–6, 0x1.72ce620000000p–5, 0x1.aa09680000000p–10, –0x1.2ea3aa0000000p–9, –0x1.6cbd040000000p–5, 0x1.54c66e0000000p–4, –0x1.6ae3520000000p–9, 0x1.7b98bc0000000p–12, 0x1.9b9de20000000p–8, 0x1.d738ec0000000p–12, –0x1.9c14760000000p–8,
–0x1.1ff94e0000000p–5, 0x1.6765c60000000p–14, –0x1.78521c0000000p–10, 0x1.5412620000000p–9, 0x1.1221ca0000000p–12, 0x1.67aea40000000p–6, –0x1.23eb0a0000000p–5, –0x1.d0af120000000p–11, 0x1.2140c80000000p–9, 0x1.cb77860000000p–5, –0x1.3c430e0000000p–4, 0x1.484ab40000000p–10, –0x1.793a120000000p–14, –0x1.a82c960000000p–7, –0x1.90b2ac0000000p–12, 0x1.f26f880000000p–9,
0x1.3e4fc40000000p–5, –0x1.c629ba0000000p–13, –0x1.6b53920000000p–7, 0x1.012f2e0000000p–8, 0x1.6b9d200000000p–9, –0x1.4fb38e0000000p–9, –0x1.3f13960000000p–4, 0x1.9a47680000000p–11, 0x1.1e548c0000000p–5, –0x1.904d6c0000000p–5, 0x1.9c929e0000000p–5, –0x1.896a7e0000000p–9, 0x1.6ffe540000000p–15, 0x1.5ce2280000000p–5, 0x1.ddb7460000000p–11, 0x1.2786760000000p–8,

Table E.16 Layer 1 CNN (convolution kernel parameter kernel) of context coding neural network parameter

Parameter value
0x1.082fb40000000p–4, 0x1.80aa080000000p–3, –0x1.33c5d40000000p–4, –0x1.5ad6000000000p–5, 0x1.52699c0000000p–4, 0x1.1b68400000000p–3, –0x1.b676280000000p–6, –0x1.e9dc7a0000000p–5, –0x1.43a4880000000p–7, –0x1.91d32e0000000p–8, –0x1.2c6fe80000000p–4, –0x1.b4aa720000000p–4, –0x1.ab04400000000p–5, –0x1.9412920000000p–8, 0x1.4641cc0000000p–3, –0x1.06cd7a0000000p–5,
0x1.634b1c0000000p–4, –0x1.a6ba9a0000000p–3, –0x1.37ad280000000p–5, –0x1.2aacac0000000p–7, –0x1.36fc220000000p–6, 0x1.31406c0000000p–4, –0x1.7d49b80000000p–7, –0x1.513a6e0000000p–7, –0x1.c372ee0000000p–7, 0x1.baa7f00000000p–10, 0x1.d77d3c0000000p–10, –0x1.be4ab40000000p–4, –0x1.6708bc0000000p–6, 0x1.a224a80000000p–9, 0x1.2529640000000p–3, –0x1.9535ae0000000p–6,

Parameter value
0x1.8b756e0000000p–4, 0x1.a4c2040000000p–3, –0x1.044a140000000p–6, 0x1.bbee820000000p–10, –0x1.1de1140000000p–7, 0x1.3854880000000p–5, –0x1.aee8400000000p–7, –0x1.088cbc0000000p–4, –0x1.0990940000000p–7, 0x1.398f740000000p–9, –0x1.c38e5c0000000p–8, –0x1.d661e80000000p–4, –0x1.fbc7540000000p–7, –0x1.7e3fa00000000p–9, 0x1.10c3580000000p–3, –0x1.e3a0ec0000000p–7,

Table E.16 (continued)

Parameter value
0x1.4475c80000000p–4, –0x1.8dfabc0000000p–3, –0x1.90d80c0000000p–5, –0x1.77bb8e0000000p–8, –0x1.26b2ac0000000p–7, 0x1.f66ef40000000p–5, –0x1.3dfe6a0000000p–8, 0x1.a1ea8e0000000p–11, –0x1.14db860000000p–9, –0x1.6c95cc0000000p–8, 0x1.5033d40000000p–9, –0x1.1231200000000p–3, –0x1.8bc5a60000000p–7, –0x1.00ba3a0000000p–8, 0x1.f49d6c0000000p–4, –0x1.7484b80000000p–7,
0x1.45edbc0000000p–4, –0x1.8dd5940000000p–3, –0x1.00e3c80000000p–5, –0x1.82afde0000000p–7, –0x1.20cb3e0000000p–6, 0x1.2af7500000000p–5, –0x1.057ebe0000000p–10, –0x1.bfd61a0000000p–6, –0x1.8ba1b40000000p–8, –0x1.c078000000000p–8, 0x1.b0b1060000000p–9, –0x1.1b97920000000p–3, –0x1.c6e1260000000p–7, –0x1.e76ef20000000p–13, 0x1.df21bc0000000p–4, –0x1.5845b60000000p–8,
0x1.4f28900000000p–5, –0x1.3f9c0a0000000p–3, –0x1.0cf4a20000000p–4, 0x1.4428d60000000p–3, 0x1.7a85040000000p–4, 0x1.c4df120000000p–4, –0x1.ceea5e0000000p–6, –0x1.a2dec40000000p–6, –0x1.e993a20000000p–5, –0x1.0cc9f20000000p–6, –0x1.0e28760000000p–4, –0x1.bea5b80000000p–4, –0x1.e97f1a0000000p–8, –0x1.465c5c0000000p–7, 0x1.768d940000000p–4, –0x1.7077740000000p–5,
0x1.23c3fe0000000p–4, 0x1.a546d80000000p–3, –0x1.9c30ae0000000p–5, –0x1.e0ff9c0000000p–7, –0x1.673d440000000p–7, 0x1.8336c80000000p–4, –0x1.0f00d60000000p–7, –0x1.df60ce0000000p–8, –0x1.d2b4c80000000p–7, 0x1.bec3720000000p–9, –0x1.5447a60000000p–8, –0x1.79e6040000000p–4, –0x1.60c3b80000000p–6, –0x1.cdb5620000000p–7, 0x1.11d58c0000000p–3, –0x1.c4f9a80000000p–6,
0x1.92efb60000000p–4, –0x1.075b800000000p–3, –0x1.2183f40000000p–6, 0x1.e77bc00000000p–10, –0x1.4b942c0000000p–6, 0x1.a592ea0000000p–7, –0x1.a52ca00000000p–7, –0x1.c287540000000p–5, –0x1.3467e40000000p–9, 0x1.7a0b380000000p–8, 0x1.f1c2e00000000p–9, –0x1.38e8980000000p–3, –0x1.7dc61c0000000p–7, –0x1.0a4dee0000000p–8, 0x1.047b580000000p–3,

Parameter value
–0x1.25beb60000000p–7,
–0x1.6c68c80000000p–8, –0x1.dc6c240000000p–3, –0x1.8e72f00000000p–4, –0x1.c778e60000000p–6, –0x1.68d5dc0000000p–9, 0x1.3fda4e0000000p–3, –0x1.6dac040000000p–7, 0x1.08142e0000000p–3, –0x1.86d4ee0000000p–6, –0x1.7e4e9c0000000p–5, –0x1.aaf6b00000000p–6, –0x1.6dfa040000000p–4, –0x1.f6d9cc0000000p–9, –0x1.21e3340000000p–9, 0x1.c844200000000p–5, –0x1.4bbde60000000p–5,
0x1.c4b8220000000p–5, –0x1.017f360000000p–3, –0x1.a5b44c0000000p–4, –0x1.bef33c0000000p–4, –0x1.14750e0000000p–7, 0x1.cd268c0000000p–5, –0x1.9daf580000000p–6, 0x1.b20d840000000p–3, 0x1.cd3fb20000000p–4, –0x1.0b3f440000000p–4, –0x1.0276d20000000p–6, –0x1.2a5e600000000p–3, –0x1.857e720000000p–10, –0x1.bdbe4e0000000p–6, 0x1.3b7f200000000p–4, 0x1.bd11420000000p–4,

Table E.16 (continued)

Parameter value
0x1.f8b2b20000000p–9, –0x1.45b3d60000000p–3, –0x1.18aaa80000000p–4, –0x1.17b3720000000p–5, –0x1.25b0260000000p–5, 0x1.5a88c00000000p–3, –0x1.c81af40000000p–8, –0x1.e108080000000p–6, –0x1.8b571e0000000p–5, 0x1.d45bfe0000000p–9, –0x1.4c26420000000p–8, –0x1.a8836a0000000p–4, –0x1.f7ab420000000p–6, –0x1.51045e0000000p–6, 0x1.8324be0000000p–4, –0x1.edaef60000000p–5,
0x1.3a0eb40000000p–4, –0x1.5e3fdc0000000p–3, –0x1.c1d2ec0000000p–5, –0x1.8672100000000p–6, –0x1.0444700000000p–6, 0x1.7a56300000000p–5, 0x1.5373b60000000p–9, –0x1.dc1a5e0000000p–6, –0x1.c0eb580000000p–7, 0x1.07d1da0000000p–11, –0x1.61f14a0000000p–11, –0x1.e07dae0000000p–4, –0x1.3172fc0000000p–6, 0x1.2281e00000000p–12, 0x1.e566360000000p–4, –0x1.5bfbd20000000p–6,
0x1.c6cd660000000p–4, –0x1.95ab020000000p–3, –0x1.3217b40000000p–5, –0x1.280e540000000p–6, –0x1.27d4de0000000p–7, 0x1.72563c0000000p–6, –0x1.74690e0000000p–8, –0x1.167f980000000p–4, –0x1.219b500000000p–9, –0x1.fa63c00000000p–8, 0x1.add2160000000p–8, –0x1.b0f6fe0000000p–3, –0x1.26a0660000000p–7, 0x1.bccc680000000p–9, 0x1.1ce4a60000000p–3, –0x1.a432ba0000000p–6,
0x1.55d7280000000p–5, –0x1.acfdf60000000p–4, –0x1.2851de0000000p–3 0x1.ac136e0000000p–3, 0x1.bc0eea0000000p–4, 0x1.06379c0000000p–4, –0x1.89d3280000000p–5, –0x1.42ab3e0000000p–4, 0x1.4e3ff20000000p–4, –0x1.4a687e0000000p–5, –0x1.97cc1c0000000p–7, –0x1.41a9a20000000p–3,

Parameter value
0x1.93cea60000000p–8, –0x1.adacee0000000p–6, 0x1.c96c580000000p–5, 0x1.05de880000000p–10,
0x1.39abc00000000p–4, –0x1.b284420000000p–3, –0x1.3de8520000000p–5, –0x1.b577a40000000p–8, –0x1.319d680000000p–8, 0x1.e04c760000000p–6, –0x1.64e2700000000p–10, –0x1.bcf5b20000000p–5, 0x1.078b0c0000000p–8, –0x1.2efaca0000000p–9, 0x1.27fb460000000p–9, –0x1.f5eff00000000p–5, –0x1.d8f8c40000000p–8, –0x1.4cb75a0000000p–8, 0x1.ae8e640000000p–4, –0x1.03083e0000000p–8,
0x1.2fbaea0000000p–4, 0x1.50ee5e0000000p–3, –0x1.1b99100000000p–4, –0x1.a7e8840000000p–7, –0x1.944db40000000p–6, 0x1.9139420000000p–5, –0x1.27fc4a0000000p–8, –0x1.c7282a0000000p–5, –0x1.7093300000000p–9, 0x1.31318a0000000p–12, –0x1.a596d60000000p–9, –0x1.afb6880000000p–4, –0x1.4521180000000p–7, –0x1.dc89780000000p–9, 0x1.c63e2c0000000p–4, –0x1.5c1efc0000000p–6,
0x1.1ce2040000000p–5, –0x1.6d8f2c0000000p–5, –0x1.1a19000000000p–3, –0x1.3a03780000000p–4, –0x1.7d7aa00000000p–7, 0x1.22c25c0000000p–5, –0x1.d3693e0000000p–6, –0x1.2c51720000000p–2, –0x1.4ab0960000000p–6, 0x1.a82bf40000000p–9, –0x1.2354820000000p–4, –0x1.a5db080000000p–5, 0x1.d9c9ce0000000p–5, –0x1.06da060000000p–5, –0x1.c0bde80000000p–7, –0x1.f7f40c0000000p–6,

Table E.16 (continued)

Parameter value
0x1.62b1240000000p–5, –0x1.ec00060000000p–4, –0x1.03936c0000000p–3, –0x1.6a6a2c0000000p–4, –0x1.3834520000000p–5, 0x1.0dbee00000000p–8, 0x1.bf3f960000000p–4, –0x1.2c29b40000000p–3, –0x1.0ba6220000000p–4, 0x1.172d8c0000000p–4, –0x1.51357a0000000p–4, –0x1.04fcec0000000p–4, 0x1.e7078e0000000p–5, –0x1.ec8e940000000p–6, –0x1.3e372e0000000p–6, –0x1.00c1be0000000p–4,
0x1.71e79a0000000p–5, –0x1.7e65aa0000000p–4, –0x1.f580200000000p–4, –0x1.aadddc0000000p–4, 0x1.c075420000000p–4, –0x1.a987f40000000p–10, 0x1.cc86ce0000000p–4, –0x1.a530760000000p–3, –0x1.33fd500000000p–4, –0x1.2985b80000000p–7, –0x1.35119c0000000p–4, –0x1.33bb540000000p–3, 0x1.db97780000000p–5, –0x1.75b26a0000000p–5, –0x1.feb5440000000p–7, –0x1.1e5d180000000p–8,
0x1.5b73140000000p–5, –0x1.5725920000000p–5, 0x1.1ec2d40000000p–2, –0x1.7ef67e0000000p–4, –0x1.f87df80000000p–6, –0x1.fbbb040000000p–5, 0x1.a92e8a0000000p–4, –0x1.2903000000000p–3, 0x1.b253f00000000p–4, 0x1.1217aa0000000p–4, –0x1.5f47ee0000000p–4, –0x1.5a6ee00000000p–4,

Parameter value
0x1.68ffd60000000p–6, –0x1.afc9d40000000p–6, –0x1.209d7a0000000p–7, –0x1.99342c0000000p–6,
0x1.7463da0000000p–5, –0x1.5129460000000p–3, –0x1.38a8140000000p–3, –0x1.34f9e00000000p–4, –0x1.ee5e540000000p–7, 0x1.0f8a000000000p–7, –0x1.b114080000000p–5, –0x1.31d7560000000p–3, –0x1.b1ba220000000p–5, 0x1.117a840000000p–4, –0x1.12334e0000000p–4, –0x1.2b20120000000p–4, 0x1.f8cc8c0000000p–5, –0x1.243acc0000000p–5, –0x1.eb23660000000p–7, 0x1.ada34e0000000p–4,
0x1.6e922c0000000p–6, –0x1.be28fe0000000p–6, –0x1.11431e0000000p–3, 0x1.6d4b000000000p–3, –0x1.311e280000000p–7, 0x1.864c3c0000000p–5, –0x1.05542a0000000p–5, –0x1.b36c480000000p–4, 0x1.be2bc20000000p–8, –0x1.41d4c40000000p–6, –0x1.cdc33e0000000p–4, –0x1.de19940000000p–5, 0x1.9440040000000p–5, –0x1.fe42e40000000p–6, –0x1.8a8ee00000000p–7, –0x1.b1b4400000000p–6,
–0x1.66e9060000000p–7, –0x1.13902e0000000p–4, –0x1.43a2ec0000000p–3, –0x1.689a860000000p–4, –0x1.9ab1c00000000p–6, 0x1.d2c81e0000000p–5, 0x1.b9d0040000000p–4, –0x1.07448e0000000p–2, 0x1.353ebe0000000p–4, –0x1.92c95a0000000p–5, –0x1.faa7460000000p–4, –0x1.75000a0000000p–4, 0x1.9d94bc0000000p–7, –0x1.1813fe0000000p–5, –0x1.be11ce0000000p–7, –0x1.47cab40000000p–6,
0x1.19529e0000000p–4, –0x1.8fcc420000000p–4, 0x1.2979560000000p–2, –0x1.3c0c800000000p–4, –0x1.8278fe0000000p–8, –0x1.2a22060000000p–4, –0x1.be40920000000p–6, –0x1.191bac0000000p–3, 0x1.0d84c00000000p–3, 0x1.20d67e0000000p–4, –0x1.ddc8d60000000p–5, –0x1.dae8b60000000p–4, 0x1.8fe30a0000000p–6, –0x1.37693e0000000p–5, 0x1.8608900000000p–7, 0x1.7fa4a60000000p–4,

Table E.16 (continued)

Parameter value
0x1.6fef940000000p–4, –0x1.96f3ce0000000p–5, –0x1.00b56a0000000p–4, –0x1.c5bc2e0000000p–4, –0x1.27dd340000000p–6, –0x1.1d2a580000000p–5, –0x1.1a26d80000000p–6, 0x1.3c4aa80000000p–3, 0x1.08fb920000000p–4, –0x1.918f820000000p–5, –0x1.9b36b80000000p–4, –0x1.2a54960000000p–4, 0x1.97de940000000p–4, –0x1.33fedc0000000p–5, –0x1.1d1e300000000p–6, –0x1.c27ee40000000p–6,
0x1.059e540000000p–5, 0x1.2a6b440000000p–6, –0x1.0243060000000p–3, –0x1.6292ba0000000p–4, –0x1.5c43380000000p–6, 0x1.6e7d6a0000000p–7, –0x1.54eb240000000p–6, –0x1.5abb7a0000000p–3, –0x1.6f13b00000000p–8, –0x1.c5aad80000000p–8, –0x1.8f73560000000p–4, –0x1.88adc80000000p–5,

Parameter value
0x1.8bbe260000000p–5, –0x1.17e8720000000p–5, –0x1.b75f940000000p–7, –0x1.1c10280000000p–6,
0x1.8e6acc0000000p–5, –0x1.56ed7a0000000p–6, –0x1.db2f0a0000000p–4, –0x1.3387ea0000000p–4, –0x1.6b42f00000000p–5, 0x1.bad01e0000000p–6, –0x1.3af65e0000000p–7, –0x1.7b687e0000000p–3, 0x1.68f8260000000p–4, –0x1.9e6f2e0000000p–5, 0x1.dc9fc80000000p–3, –0x1.32d9e60000000p–4, 0x1.18646e0000000p–4, –0x1.ef8ef20000000p–6, –0x1.47fac60000000p–6, –0x1.a94b360000000p–5,
–0x1.29b3fe0000000p–11, –0x1.2c623e0000000p–3, –0x1.8507b40000000p–3, –0x1.2fa6060000000p–4, –0x1.4d10c20000000p–5, 0x1.4cc2820000000p–4, –0x1.8a4e420000000p–5, –0x1.70dde80000000p–3, –0x1.92b8b60000000p–4, 0x1.19f9e00000000p–5, –0x1.3480ba0000000p–4, –0x1.7e90aa0000000p–4, 0x1.3899f80000000p–5, –0x1.22d7ee0000000p–5, –0x1.5dd8c40000000p–6, 0x1.0d503e0000000p–3,
0x1.00c1ce0000000p–3, –0x1.5669780000000p–3, 0x1.0ba81e0000000p–2, –0x1.fa70fc0000000p–4, 0x1.584ac20000000p–4, –0x1.21b81e0000000p–4, –0x1.4543800000000p–5, –0x1.88d2220000000p–4, 0x1.378e520000000p–4, 0x1.5ecde80000000p–4, –0x1.7b51a60000000p–4, –0x1.a43c760000000p–4, 0x1.b95e240000000p–6, –0x1.b30d260000000p–6, 0x1.0cd4a40000000p–4, –0x1.268ede0000000p–6,
0x1.21187e0000000p–5, 0x1.094aae0000000p–5, –0x1.4e29100000000p–4, –0x1.cfb1580000000p–4, –0x1.cdf7ce0000000p–7, –0x1.19b14a0000000p–6, –0x1.82dd8a0000000p–6, –0x1.a06a9c0000000p–4, –0x1.dd95ac0000000p–7, –0x1.65edc60000000p–14, –0x1.0e03340000000p–3, –0x1.cc372a0000000p–7, 0x1.91decc0000000p–5, –0x1.ab2fb00000000p–5, –0x1.a084460000000p–6, –0x1.4025640000000p–8,
–0x1.556ed40000000p–7, –0x1.26d3b00000000p–4, –0x1.3c92100000000p–3, –0x1.5ec60c0000000p–4, –0x1.913b8c0000000p–8, 0x1.97091a0000000p–5, 0x1.0ba3340000000p–3, 0x1.d7cfb80000000p–3, –0x1.4af7f00000000p–5, –0x1.86a8c00000000p–5, –0x1.a60e200000000p–4, –0x1.b22df80000000p–4, 0x1.0d7a7e0000000p–6, –0x1.2d89600000000p–5, –0x1.038e120000000p–6, 0x1.93aea60000000p–4,

Table E.16 (continued)

Parameter value
–0x1.3c64540000000p–6, –0x1.7860fa0000000p–3, –0x1.9731640000000p–3, –0x1.9637de0000000p–5, –0x1.05cf860000000p–6, 0x1.d648100000000p–4, –0x1.4ead220000000p–8, –0x1.ae310e0000000p–3, –0x1.13f9400000000p–4, –0x1.1374c20000000p–4, –0x1.1f64620000000p–4, –0x1.56afb00000000p–4,

Parameter value
0x1.b917fc0000000p–6, –0x1.d6a0e20000000p–6, –0x1.f6b39a0000000p–8, 0x1.1ef8420000000p–3,
0x1.2030720000000p–8, 0x1.a067aa0000000p–9, –0x1.1f86f20000000p–4, –0x1.0788b40000000p–4, –0x1.09f2ec0000000p–11, 0x1.c102720000000p–6, –0x1.34cb8e0000000p–7, 0x1.a42d540000000p–7, –0x1.0034a80000000p–7, –0x1.34f5620000000p–11, –0x1.5cf3f20000000p–5, 0x1.c230d20000000p–6, 0x1.5636a40000000p–6, –0x1.0325e80000000p–5, –0x1.51c8ea0000000p–7, –0x1.d6a1360000000p–7,
0x1.6345540000000p–9, –0x1.8006800000000p–8, –0x1.44c6680000000p–4, –0x1.d60c280000000p–5, –0x1.2df8a80000000p–8, 0x1.568fcc0000000p–5, –0x1.3dda180000000p–6, –0x1.35e64c0000000p–4, –0x1.7575180000000p–8, –0x1.2e07820000000p–7, 0x1.fa101e0000000p–3, 0x1.8c251e0000000p–6, 0x1.bba0d40000000p–6, –0x1.1fbcfa0000000p–5, –0x1.1771ba0000000p–6, –0x1.8e2de00000000p–6,
0x1.690a500000000p–9, –0x1.3c17a80000000p–7, –0x1.15fe8c0000000p–4, –0x1.6655140000000p–5, –0x1.678bee0000000p–9, 0x1.21c8a20000000p–7, –0x1.1418200000000p–9, –0x1.3e2fe00000000p–4, –0x1.69db940000000p–7, 0x1.ea940a0000000p–10, –0x1.fa845e0000000p–4, 0x1.2c2a200000000p–5, 0x1.02a29c0000000p–6, –0x1.898c460000000p–5, –0x1.277a6e0000000p–6, –0x1.68fa380000000p–9,
0x1.2a78240000000p–9, –0x1.04caf00000000p–5, –0x1.062bea0000000p–4, –0x1.acc2920000000p–4, –0x1.10bf100000000p–7, 0x1.e11e560000000p–7, –0x1.c01fa20000000p–8, –0x1.1dd91e0000000p–5, 0x1.3360b80000000p–12, –0x1.d0f45a0000000p–9, –0x1.a30c080000000p–4, 0x1.97452e0000000p–6, 0x1.5e585e0000000p–7, –0x1.f4d4420000000p–6, –0x1.39df140000000p–6, –0x1.5b41d00000000p–6,
–0x1.6094880000000p–6, –0x1.b2a0700000000p–6, –0x1.dd05560000000p–4, 0x1.a0f8ac0000000p–3, –0x1.79bf020000000p–8, 0x1.c015e80000000p–5, –0x1.ca39200000000p–7, –0x1.c129180000000p–5, 0x1.6a68420000000p–9, –0x1.09d2e20000000p–8, –0x1.d55a600000000p–5, 0x1.c551900000000p–6, 0x1.8f712c0000000p–11, –0x1.6a69dc0000000p–5, –0x1.2b25ec0000000p–7, –0x1.21ace60000000p–7,
–0x1.459faa0000000p–6, –0x1.22e8ae0000000p–6, –0x1.6095340000000p–5, –0x1.ca95060000000p–5, –0x1.2e1a860000000p–8, 0x1.7f89720000000p–5, –0x1.9d620e0000000p–8, –0x1.0e15980000000p–8, 0x1.62f9740000000p–10, 0x1.0a58d40000000p–11, –0x1.be50d00000000p–6, 0x1.e6052e0000000p–6, –0x1.c33fd20000000p–11, –0x1.f2ca5e0000000p–6, –0x1.ecf48c0000000p–9, –0x1.561ad20000000p–8,

Table E.16 (continued)

Parameter value
0x1.621eb00000000p–6, –0x1.4af4200000000p–5, –0x1.abe6ec0000000p–6, –0x1.61de920000000p–4, –0x1.5c90480000000p–9, 0x1.56e8ba0000000p–8, –0x1.7ded500000000p–8, 0x1.9dfe3c0000000p–7, –0x1.1a5b520000000p–8, 0x1.8087f20000000p–9, –0x1.8f47160000000p–4, 0x1.0b31860000000p–5, 0x1.188fa40000000p–5, –0x1.bdc5080000000p–5, –0x1.12e67c0000000p–6, –0x1.1dad2a0000000p–6,
–0x1.b4a34c0000000p–6, –0x1.e554cc0000000p–8, –0x1.cdf63a0000000p–4, –0x1.fe23d80000000p–4, –0x1.7ca1dc0000000p–7, 0x1.8fd6660000000p–5, –0x1.537a140000000p–8, –0x1.8021b00000000p–4, 0x1.b5522e0000000p–9, –0x1.7a6d8a0000000p–7, –0x1.539b480000000p–4, 0x1.1905840000000p–6, –0x1.a8027c0000000p–7, –0x1.80b6060000000p–5, –0x1.ce2c720000000p–8, –0x1.a6fee20000000p–9,
–0x1.f5255c0000000p–6, –0x1.36b70c0000000p–7, –0x1.287a9e0000000p–5, –0x1.dc9e920000000p–5, –0x1.b76bc20000000p–8, 0x1.93fd8a0000000p–5, –0x1.70f7600000000p–8, –0x1.fcfa5e0000000p–6, 0x1.2ead9c0000000p–8, 0x1.e467120000000p–10, –0x1.e9c09e0000000p–5, 0x1.aa06620000000p–6, –0x1.2489000000000p–7, –0x1.31ffa00000000p–6, –0x1.1db7480000000p–8, –0x1.40b2fe0000000p–11,
–0x1.25db920000000p–6, –0x1.402a260000000p–5, –0x1.3d69d80000000p–5, –0x1.05d7d20000000p–6, –0x1.0a8fba0000000p–9, 0x1.6519600000000p–5, –0x1.13d6300000000p–7, –0x1.6c9ee00000000p–7, –0x1.f65d760000000p–9, 0x1.dfe9420000000p–11, –0x1.68db6a0000000p–6, 0x1.e870540000000p–7, 0x1.2dd32a0000000p–10, –0x1.33d8e60000000p–5, –0x1.2196500000000p–11, –0x1.0b791c0000000p–8,
0x1.6fc12e0000000p–10, –0x1.2173700000000p–6, –0x1.13c6ce0000000p–6, –0x1.1b88c20000000p–4, –0x1.1f5ff80000000p–8, 0x1.dbbd900000000p–6, –0x1.35f8ce0000000p–7, 0x1.0694be0000000p–5, –0x1.bcf91a0000000p–10, 0x1.1bc4b00000000p–11, –0x1.c030280000000p–5, 0x1.6dd8c20000000p–5, 0x1.769b360000000p–6, –0x1.89f0800000000p–5, –0x1.9817780000000p–7, –0x1.a1dff60000000p–7,
0x1.c040ce0000000p–6, –0x1.1c113c0000000p–6, –0x1.63879e0000000p–4, –0x1.b8a3dc0000000p–4, –0x1.762c2a0000000p–8, 0x1.1061f00000000p–6, –0x1.36b44e0000000p–6, –0x1.1fed9e0000000p–5, –0x1.5647020000000p–7, 0x1.e5067e0000000p–9, –0x1.94f9ba0000000p–4, 0x1.e1e6b20000000p–6, 0x1.6d6a940000000p–5, 0x1.1ea1080000000p–3, –0x1.bb4f4a0000000p–7, –0x1.2eff3a0000000p–6,
0x1.fa152c0000000p–5, –0x1.564c960000000p–7, –0x1.5dc22c0000000p–4, –0x1.3d4ebc0000000p–4, –0x1.906be20000000p–7, –0x1.25e1880000000p–5, –0x1.a0f35c0000000p–6, 0x1.1371de0000000p–4, –0x1.cf73240000000p–8, 0x1.b08fbc0000000p–8, –0x1.684aca0000000p–4, –0x1.13640c0000000p–6,

Parameter value
0x1.193a080000000p–4, –0x1.09591e0000000p–5, –0x1.7a8a320000000p–6, –0x1.15eb3a0000000p–6,

Table E.16 (continued)

Parameter value
–0x1.e6d41a0000000p–9, 0x1.3eba4c0000000p–10, –0x1.7012720000000p–7, –0x1.87d5500000000p–7, –0x1.3138b60000000p–8, 0x1.1ef4580000000p–6, –0x1.7f9c0a0000000p–7, –0x1.eea4c20000000p–10, –0x1.c3fbc40000000p–12, –0x1.49fe360000000p–9, –0x1.aed1500000000p–6, 0x1.8067960000000p–6, 0x1.96b6320000000p–8, –0x1.2009220000000p–5, –0x1.065e8a0000000p–7, –0x1.ecea9c0000000p–9,
–0x1.60def20000000p–7, –0x1.27d12a0000000p–6, –0x1.8fb8060000000p–5, –0x1.197d120000000p–4, –0x1.7d01420000000p–9, 0x1.bbbe540000000p–5, –0x1.a3a28c0000000p–8, –0x1.96f7000000000p–4, –0x1.32d5f80000000p–8, 0x1.a35dca0000000p–10, –0x1.4563ce0000000p–3, 0x1.f6c0ce0000000p–6, 0x1.9086dc0000000p–7, –0x1.1d31260000000p–4, –0x1.c276da0000000p–7, –0x1.9a875a0000000p–8,
0x1.971ed20000000p–8, –0x1.6cd2080000000p–5, –0x1.6590d80000000p–4, –0x1.8e7f480000000p–4, 0x1.1ecd780000000p–10, 0x1.217d8a0000000p–5, –0x1.b8141e0000000p–8, 0x1.6e57620000000p–4, –0x1.413bde0000000p–7, 0x1.71d7a00000000p–10, –0x1.8adda20000000p–3, 0x1.18eb2a0000000p–5, 0x1.1ba6a40000000p–5, –0x1.001ef00000000p–4, –0x1.6411980000000p–6, –0x1.c9c70c0000000p–8,

Table E.17 Layer 1 CNN (bias parameter bias) of context coding neural network parameter

Parameter value
0x1.f3f7b00000000p–2, 0x1.df46de0000000p–3, –0x1.012fc00000000p–2, 0x1.6736900000000p–7, –0x1.e9370a0000000p–6, 0x1.28ba000000000p–3, –0x1.5de3e20000000p–5, 0x1.0bbf240000000p–2, –0x1.3300f20000000p–5, 0x1.4df4960000000p–7, 0x1.0308060000000p–9, 0x1.f9de120000000p–2, –0x1.2999fa0000000p–3, –0x1.28c7c20000000p–3, –0x1.a76dc40000000p–7, –0x1.8955f20000000p–5,

Table E.18 Layer 2 CNN (convolution kernel parameter kernel) of context coding neural network parameter

Parameter value
–0x1.be1bc60000000p–5, 0x1.3cecd80000000p–4, –0x1.9f21ec0000000p–2, –0x1.c220f00000000p–5, –0x1.22cf7e0000000p–2, 0x1.58eb760000000p–4, –0x1.8336c40000000p–6, –0x1.059aba0000000p–2, –0x1.8743b20000000p–3, 0x1.aa892c0000000p–9, 0x1.1808de0000000p–5, –0x1.58c7b20000000p–3, 0x1.168df00000000p–3, 0x1.0285c40000000p–5, –0x1.aa26b40000000p–4, 0x1.96d7da0000000p–4,

Table E.18 (continued)

Parameter value
–0x1.df167c0000000p–5, –0x1.340a500000000p–6, 0x1.3407000000000p–6, –0x1.8c78160000000p–3, –0x1.a307bc0000000p–3, –0x1.8c2cba0000000p–6, –0x1.8d89960000000p–4, 0x1.925f800000000p–3, –0x1.22feae0000000p–3, –0x1.219f6a0000000p–7, –0x1.eb1b380000000p–4, 0x1.5555b40000000p–3, –0x1.34073e0000000p–5, –0x1.57e9080000000p–7, 0x1.0120920000000p–5, –0x1.38da300000000p–7,
–0x1.ed4a7e0000000p–3, –0x1.9b46180000000p–4, 0x1.21a4c20000000p–3, –0x1.fb89a80000000p–4, –0x1.dd40ee0000000p–6, –0x1.e367d20000000p–4, 0x1.78ab620000000p–3, –0x1.786bbc0000000p–4, 0x1.33b6860000000p–3, –0x1.419b380000000p–6, 0x1.1484ce0000000p–1, –0x1.d36e760000000p–3, –0x1.7be8300000000p–3, –0x1.6fef520000000p–5, –0x1.9ca19e0000000p–3, –0x1.a7e6ce0000000p–4,
–0x1.126cc60000000p–2, –0x1.e0c56a0000000p–4, 0x1.07b4120000000p–3, 0x1.a35e1a0000000p–5, 0x1.cb87280000000p–4, –0x1.1b48780000000p–3, 0x1.a71d7c0000000p–2, 0x1.69a3880000000p–4, 0x1.6cefec0000000p–4, –0x1.7dbca20000000p–6, –0x1.0fa1180000000p+1, –0x1.9707c00000000p–3, –0x1.bd459c0000000p–3, –0x1.ade2880000000p–5, 0x1.deebf60000000p–4, –0x1.eebd940000000p–4,
0x1.5cd8ca0000000p–3, 0x1.30288a0000000p–4, –0x1.37ab720000000p–4, –0x1.e38cfe0000000p–9, –0x1.9c21b20000000p–4, 0x1.65ff260000000p–4, –0x1.4a2ab80000000p–2, 0x1.d1330a0000000p–7, –0x1.e1b03c0000000p–3, 0x1.e105b20000000p–7, –0x1.153dba0000000p–1, –0x1.e2d13c0000000p–3, 0x1.1856620000000p–3, 0x1.0b30a80000000p–5, 0x1.fbf7140000000p–2, 0x1.36bc460000000p–4,
0x1.4675920000000p–4, 0x1.1c736e0000000p–4, –0x1.9b6e780000000p–3, 0x1.0bd8580000000p–11, –0x1.5affc20000000p–5, 0x1.47bb540000000p–4, –0x1.09c2f40000000p–3, –0x1.5cfe000000000p–8, –0x1.b98d740000000p–4, 0x1.6ffe380000000p–7, –0x1.224d9e0000000p–2, –0x1.1e8e760000000p–2, 0x1.040c340000000p–3, 0x1.f3617e0000000p–6, 0x1.b091020000000p–6,

Parameter value
0x1.34b07a0000000p–4,
0x1.3ff67c0000000p–3, 0x1.00d86e0000000p–4, –0x1.4c0aa60000000p–4, –0x1.3e27480000000p–5, –0x1.105eca0000000p–3, 0x1.23c2040000000p–4, –0x1.963bbc0000000p–1, –0x1.91059e0000000p–3, 0x1.45e5bc0000000p–4, 0x1.3c04580000000p–7, –0x1.eb39720000000p–2, 0x1.887fb20000000p–3, 0x1.d12ab20000000p–4, 0x1.bb0e880000000p–6, –0x1.128f3e0000000p–2, 0x1.19d9660000000p–4,
–0x1.2eff920000000p–3, –0x1.ab7bde0000000p–5, 0x1.4044660000000p–3, –0x1.74188e0000000p–7, 0x1.7178020000000p–3, –0x1.dea97a0000000p–5, 0x1.0f09f00000000p–4, 0x1.9796bc0000000p–4, –0x1.0e10580000000p–2, –0x1.a973e00000000p–8, –0x1.dcd1c40000000p–1, –0x1.91ab880000000p–3, –0x1.807d480000000p–4, –0x1.68e7180000000p–6, 0x1.f50a860000000p–4, –0x1.f06d440000000p–5,

Table E.18 (continued)

Parameter value
0x1.0b00940000000p–3, 0x1.ad8c8a0000000p–5, –0x1.04800a0000000p–4, 0x1.43b4620000000p–5, 0x1.e81d140000000p–6, 0x1.0e55800000000p–4, –0x1.5e9b040000000p–2, –0x1.b1a97e0000000p–4, 0x1.7771e00000000p–5, 0x1.2760ce0000000p–6, –0x1.88f0fe0000000p–2, –0x1.df76c60000000p–3, 0x1.9ec9000000000p–4, 0x1.9e6aa00000000p–6, –0x1.1fd3a00000000p–6, 0x1.5065e20000000p–5,
0x1.568aac0000000p–3, 0x1.14fdb40000000p–4, –0x1.fde6d20000000p–4, 0x1.2944440000000p–3, 0x1.8a00620000000p–3, 0x1.433fb20000000p–4, –0x1.ef94940000000p–2, 0x1.2b0aea0000000p–4, –0x1.02163c0000000p–3, 0x1.b455600000000p–7, –0x1.2a367e0000000p–1, –0x1.b90ea60000000p–3, 0x1.fda4b40000000p–4, 0x1.eba1a80000000p–6, –0x1.283d400000000p–9, 0x1.19f0420000000p–4,
–0x1.13954a0000000p–2, –0x1.d29e980000000p–4, 0x1.cb144c0000000p–4, 0x1.57351e0000000p–3, 0x1.53d7d80000000p–3, –0x1.0797220000000p–3, –0x1.e3d0a20000000p–3, 0x1.3baf300000000p–3, –0x1.adf81a0000000p–6, –0x1.b691ae0000000p–7, –0x1.f13a0e0000000p+0, –0x1.3e31de0000000p–3, –0x1.a315c00000000p–3, –0x1.913f460000000p–5, 0x1.9d7c560000000p–4, –0x1.0dfdd80000000p–3,
–0x1.01fcc60000000p–4, 0x1.5974e40000000p–5, 0x1.84d9e80000000p–2, 0x1.b7040e0000000p–3, –0x1.1e075c0000000p–5, –0x1.0298460000000p–5, 0x1.a74b2a0000000p–3, –0x1.9dc0a40000000p–6, 0x1.7faf580000000p–3, –0x1.2407de0000000p–3, –0x1.bf46f40000000p–2, 0x1.ad69720000000p–3, –0x1.0f83780000000p–8, –0x1.25ffd40000000p–6, 0x1.d155a60000000p–4,

Parameter value
0x1.33ffa40000000p–2,
0x1.97c8dc0000000p–3, 0x1.1217800000000p–7, 0x1.abebbe0000000p–3, –0x1.3786520000000p–3, 0x1.42cb280000000p–5, 0x1.86de420000000p–7, –0x1.81f9c80000000p–2, –0x1.7ea9da0000000p–4, –0x1.33b1440000000p–4, 0x1.6160140000000p–8, –0x1.19a66a0000000p–1, 0x1.10aa9e0000000p–4, 0x1.2174cc0000000p–6, 0x1.3757e40000000p–8, –0x1.34c6440000000p–2, 0x1.2c665e0000000p–9,
–0x1.cb0f220000000p–3, –0x1.bcf2980000000p–4, 0x1.0d6bdc0000000p–2, 0x1.0d08440000000p–3, 0x1.e67cae0000000p–5, –0x1.b03fc00000000p–4, –0x1.91d9ac0000000p–1, 0x1.37cf6e0000000p–6, –0x1.6e9a160000000p–3, 0x1.291f700000000p–6, 0x1.591ef20000000p+0, –0x1.88e0b40000000p–3, –0x1.6ca9840000000p–3, –0x1.30bd660000000p–5, 0x1.01af5e0000000p–3, –0x1.6f0e200000000p–3,
0x1.3329680000000p–4, –0x1.2af6220000000p–4, 0x1.6a2d520000000p–2, 0x1.70f01c0000000p–5, –0x1.7b469e0000000p–4, –0x1.3346e20000000p–4, 0x1.020f8c0000000p–3, –0x1.2b48ea0000000p–4, –0x1.1683f80000000p–5, 0x1.304ca80000000p–8, –0x1.0b54580000000p–5, –0x1.8427060000000p–3, –0x1.fcffda0000000p–4, –0x1.cb3c360000000p–6, –0x1.2c45820000000p–2, –0x1.b77f720000000p–4,

Table E.18 (continued)

Parameter value
0x1.21c60c0000000p–3, 0x1.b4540c0000000p–5, –0x1.69cf360000000p–4, 0x1.e129840000000p–5, –0x1.64843e0000000p–3, 0x1.ef9d1a0000000p–5, –0x1.52e2dc0000000p–1, 0x1.f573e20000000p–4, –0x1.5ba5b00000000p–5, 0x1.f1c46a0000000p–8, –0x1.b8d2d00000000p–2, –0x1.f647580000000p–4, 0x1.8a5f540000000p–4, 0x1.7853b00000000p–6, –0x1.53b8ce0000000p–3, 0x1.e737940000000p–5,
0x1.7b6ff40000000p–7, 0x1.212c0c0000000p–6, 0x1.c08f000000000p–4, –0x1.f000700000000p–5, –0x1.68b6540000000p–4, 0x1.14f9460000000p–6, –0x1.d4e8840000000p–4, –0x1.21782a0000000p–2, –0x1.9c94cc0000000p–3, –0x1.0385260000000p–8, 0x1.7ce1a20000000p–5, –0x1.97db220000000p–3, 0x1.cc862e0000000p–6, 0x1.7fbc740000000p–8, 0x1.92f2e20000000p–4, 0x1.ee6b060000000p–6,
–0x1.6b59520000000p–2, –0x1.2a58b00000000p–3, 0x1.b3fb6a0000000p–4, –0x1.ca12720000000p–3, –0x1.b72ef40000000p–4, –0x1.64b6f40000000p–3, –0x1.b9e7ae0000000p–7, –0x1.e2b3680000000p–4, –0x1.d29a680000000p–6, –0x1.1ae9cc0000000p–5, –0x1.f21b600000000p–1, 0x1.12b6760000000p–3,

Parameter value
–0x1.1709100000000p–2, –0x1.0da0a80000000p–4, –0x1.593a520000000p–3, –0x1.1d7b1c0000000p–3,
–0x1.0e49980000000p–2, –0x1.db5e9e0000000p–4, 0x1.46c4060000000p–4, –0x1.9b3f800000000p–4, 0x1.936a2e0000000p–4, –0x1.07d32a0000000p–3, –0x1.6422880000000p+0, 0x1.4499400000000p–4, 0x1.45a9980000000p–5, –0x1.6588b60000000p–7, –0x1.3329e00000000p+1, 0x1.0491ee0000000p–3, –0x1.a6e5100000000p–3, –0x1.9393740000000p–5, –0x1.ef36ae0000000p–5, –0x1.1a3f8e0000000p–3,
–0x1.000eec0000000p–4, –0x1.16a2a80000000p–5, 0x1.0345d80000000p–6, –0x1.b7c69c0000000p–5, 0x1.5e1f000000000p–4, –0x1.2c69fe0000000p–5, –0x1.243ef00000000p–1, 0x1.63b35c0000000p–3, 0x1.430ecc0000000p–3, 0x1.4c37020000000p–11, –0x1.9330820000000p–3, –0x1.e6c98a0000000p–4, –0x1.dce9aa0000000p–5, –0x1.d3a0c80000000p–7, 0x1.715b760000000p–2, –0x1.7fd7320000000p–5,
0x1.6a1d7e0000000p–3, 0x1.37156e0000000p–4, –0x1.1123820000000p–4, –0x1.9472860000000p–5, 0x1.ec36e80000000p–4, 0x1.5966c20000000p–4, –0x1.cce5c40000000p–1, –0x1.63de540000000p–7, 0x1.4ac0920000000p–4, 0x1.ecb01a0000000p–8, –0x1.23de2c0000000p–1, –0x1.338d960000000p–3, 0x1.142aee0000000p–3, 0x1.02978a0000000p–5, –0x1.4fc06c0000000p–3, 0x1.7252da0000000p–4,
0x1.ada25c0000000p–4, 0x1.92f7b80000000p–5, 0x1.7d53300000000p–6, –0x1.3c28700000000p–5, 0x1.8dc9c80000000p–5, 0x1.b51cbc0000000p–5, –0x1.4aa39a0000000p–2, 0x1.f453b20000000p–6, –0x1.3a2d340000000p–2, 0x1.13c70a0000000p–9, –0x1.4226dc0000000p–3, –0x1.20792a0000000p–4, 0x1.6002cc0000000p–4, 0x1.45672a0000000p–6, 0x1.1b0d860000000p–4, 0x1.01a8960000000p–4,

Table E.18 (continued)

Parameter value
0x1.aba4260000000p–3, 0x1.55d17a0000000p–4, –0x1.a969200000000p–5, 0x1.5b02200000000p–3, –0x1.bb941a0000000p–4, 0x1.7fc7aa0000000p–4, –0x1.9e6bdc0000000p–1, –0x1.a7facc0000000p–4, –0x1.02da2a0000000p–5, 0x1.2e28d00000000p–7, –0x1.f1cca00000000p–2, 0x1.0217f00000000p–5, 0x1.340b0c0000000p–3, 0x1.22138e0000000p–5, –0x1.3c0ac60000000p–2, 0x1.9316f80000000p–4,
–0x1.8115160000000p–5, –0x1.4071b60000000p–5, 0x1.72eca80000000p–4, –0x1.40d6800000000p–5, 0x1.b0e5500000000p–4, –0x1.7fd9180000000p–5, –0x1.a33da20000000p–4, –0x1.139b280000000p–3, 0x1.1054460000000p–4,

Parameter value
–0x1.50616e0000000p–7, –0x1.a2d9100000000p–2, –0x1.617ed20000000p–3, –0x1.280eaa0000000p–4, –0x1.2294280000000p–6, 0x1.9e1d660000000p–8, –0x1.21e9320000000p–5,
0x1.198a320000000p–3, 0x1.dc54800000000p–5, –0x1.31b3660000000p–5, 0x1.92e9600000000p–3, –0x1.0f19aa0000000p–2, 0x1.04da2c0000000p–4, –0x1.42d7640000000p–1, 0x1.0be8c60000000p–4, 0x1.d790ba0000000p–7, 0x1.a1fc220000000p–9, –0x1.98bbe40000000p–2, 0x1.73d4d60000000p–5, 0x1.a739940000000p–4, 0x1.883e5c0000000p–6, 0x1.9597c20000000p–6, 0x1.2e04980000000p–4,
0x1.be469c0000000p–3, 0x1.762ae20000000p–4, –0x1.1a6ea80000000p–4, 0x1.f0c6300000000p–4, –0x1.123f1a0000000p–3, 0x1.8fca0c0000000p–4, –0x1.864ee40000000p–1, –0x1.3fefa60000000p–3, 0x1.9bd24e0000000p–3, 0x1.b546380000000p–10, –0x1.dfdf400000000p–2, –0x1.22a9ac0000000p–4, 0x1.465a620000000p–3, 0x1.2d34660000000p–5, –0x1.e995340000000p–3, 0x1.f1a8500000000p–4,
–0x1.5831c20000000p–4, –0x1.69c1de0000000p–5, 0x1.28b2a00000000p–6, –0x1.2f6ed80000000p–6, –0x1.9a14a80000000p–7, –0x1.6d79920000000p–6, –0x1.945c700000000p–1, –0x1.40b58e0000000p–3, –0x1.81fab00000000p–3, 0x1.6f01520000000p–5, –0x1.f774e60000000p–5, –0x1.4e923e0000000p–3, –0x1.add6d60000000p–5, –0x1.c5e2500000000p–8, 0x1.f654ca0000000p–2, –0x1.156c7e0000000p–3,
0x1.5f1d680000000p–3, 0x1.a920940000000p–5, 0x1.76a7540000000p–3, –0x1.85a6e20000000p–3, –0x1.9ccb760000000p–4, 0x1.48e5ca0000000p–5, 0x1.12d67a0000000p–7, 0x1.7fa6640000000p–4, –0x1.eda2500000000p–6, –0x1.cc404a0000000p–6, 0x1.2269f60000000p–1, 0x1.2dd8820000000p–4, 0x1.2f8c6e0000000p–4, 0x1.d8c8380000000p–7, 0x1.8939560000000p–6, 0x1.e423300000000p–4,
0x1.ed3a3a0000000p–4, 0x1.4952ae0000000p–5, –0x1.4ccc8c0000000p–3, 0x1.0e7dfc0000000p–2, 0x1.40bb2c0000000p–4, 0x1.fb3dca0000000p–5, –0x1.3463520000000p+0, –0x1.0519f40000000p–3, –0x1.34dcc40000000p–2, 0x1.25f21e0000000p–5, –0x1.b0bf5e0000000p–2, 0x1.ceb69c0000000p–4, 0x1.6efa3e0000000p–4, 0x1.97d7980000000p–6, 0x1.8f11980000000p–1, –0x1.970f9c0000000p–8,

Table E.18 (continued)

Parameter value
0x1.be5d980000000p–10, –0x1.2acc1a0000000p–6, –0x1.bcfebe0000000p–7, –0x1.557fb60000000p–3, –0x1.2a6a280000000p–3, 0x1.e31e1a0000000p–5, –0x1.adb2c40000000p–3, 0x1.6f90f40000000p–5, 0x1.0f6d620000000p–4,

Parameter value
0x1.2d925a0000000p–3, 0x1.63acf20000000p–3, 0x1.5778ce0000000p–3, 0x1.891f220000000p–5, 0x1.bfd8860000000p–6, 0x1.2fc5b80000000p–5, –0x1.1a6e560000000p–2,
0x1.0c54960000000p–5, –0x1.0502180000000p–8, –0x1.79aa560000000p–4, 0x1.1a29040000000p–5, –0x1.2f98560000000p–5, 0x1.417ede0000000p–8, 0x1.7ef1040000000p–2, –0x1.39fc800000000p–3, 0x1.191d3c0000000p–3, 0x1.1c68980000000p–6, 0x1.4ac3680000000p–6, –0x1.8a89a20000000p–4, 0x1.75dbf20000000p–9, 0x1.8af1c40000000p–9, 0x1.108e300000000p–3, –0x1.1b6f280000000p–5,
0x1.5f68a20000000p–3, 0x1.26c0600000000p–4, –0x1.f42e960000000p–5, –0x1.e7f6ee0000000p–6, 0x1.c6ef460000000p–4, 0x1.4f5a6e0000000p–4, –0x1.7c83680000000p–1, 0x1.2b340c0000000p–4, –0x1.0a27dc0000000p–2, 0x1.4085bc0000000p–7, –0x1.b693ec0000000p–2, –0x1.ed6a120000000p–3, 0x1.0b71680000000p–3, 0x1.fd5cd60000000p–6, –0x1.f076620000000p–3, 0x1.4c2bec0000000p–4,
0x1.127dee0000000p–5, –0x1.c30c340000000p–6, 0x1.d228060000000p–3, –0x1.f742860000000p–3, –0x1.b124b80000000p–3, –0x1.248ac40000000p–6, –0x1.1eff5e0000000p–2, –0x1.eb49f20000000p–3, –0x1.a88bf20000000p–6, 0x1.348bcc0000000p–6, 0x1.f464520000000p–3, –0x1.f136120000000p–3, –0x1.2ec4740000000p–5, –0x1.0732540000000p–7, –0x1.7dc5140000000p–1, –0x1.1cf6380000000p–4,
–0x1.8e075a0000000p–3, –0x1.7a28420000000p–4, 0x1.73dbda0000000p–4, 0x1.e76aa80000000p–4, –0x1.2567ec0000000p–3, –0x1.8dd6060000000p–4, –0x1.1e17840000000p–3, 0x1.517f260000000p–4, 0x1.f3e0700000000p–5, 0x1.0ed76a0000000p–9, –0x1.b5e4000000000p–2, 0x1.de30f20000000p–7, –0x1.457ad00000000p–3, –0x1.2abe620000000p–5, –0x1.f2e8da0000000p–2, –0x1.093dc00000000p–3,
–0x1.8e89e60000000p–6, –0x1.7304b40000000p–7, –0x1.3d6e320000000p–7, 0x1.5beae40000000p–4, –0x1.bf515a0000000p–5, –0x1.0d18420000000p–6, –0x1.935c280000000p–1, –0x1.ddca960000000p–4, –0x1.cce85a0000000p–4, –0x1.954e720000000p–8, 0x1.97d78a0000000p–6, –0x1.1e9a7a0000000p–5, –0x1.7d61fc0000000p–6, –0x1.8943060000000p–8, 0x1.1d73820000000p–3, –0x1.210b1c0000000p–8,
–0x1.47da340000000p–5, –0x1.083ec60000000p–6, –0x1.0172600000000p–8, –0x1.c71c980000000p–4, –0x1.57a88c0000000p–3, –0x1.381ff40000000p–6, 0x1.8f009c0000000p–3, 0x1.afd7240000000p–3, 0x1.6f2fde0000000p–4, –0x1.b31be60000000p–9, –0x1.7090720000000p–4, –0x1.59da560000000p–3, –0x1.e4863a0000000p–6, –0x1.f8f0960000000p–8, –0x1.23d4f80000000p–9, –0x1.1a852c0000000p–6,

Table E.18 (continued)

Parameter value
0x1.2d7d000000000p–6, 0x1.0ef25a0000000p–7, –0x1.149dfe0000000p–6, 0x1.7bb0ae0000000p–6, –0x1.4938be0000000p–4, 0x1.5dff440000000p–6, –0x1.e205000000000p–1, –0x1.4c7e1a0000000p–3, –0x1.7ea91a0000000p–4, 0x1.9234c40000000p–6, –0x1.f94cfe0000000p–3, 0x1.6dcaec0000000p–4, 0x1.b472680000000p–6, 0x1.250a820000000p–7, –0x1.3fb4a60000000p–3, –0x1.0796b80000000p–5,
–0x1.33ae4e0000000p–5, –0x1.b9781e0000000p–6, 0x1.6c3fe60000000p–4, –0x1.54a6ae0000000p–3, –0x1.1966a40000000p–3, –0x1.edb9940000000p–6, –0x1.9102220000000p–2, –0x1.191eb60000000p–2, 0x1.aed68a0000000p–6, –0x1.ccef9c0000000p–9, 0x1.556c840000000p–3, –0x1.037ff20000000p–3, –0x1.8e627c0000000p–5, –0x1.8cda080000000p–7, –0x1.b376740000000p–1, –0x1.f7d1500000000p–6,
–0x1.3144620000000p–6, –0x1.b42fd20000000p–8, 0x1.73f4380000000p–9, –0x1.ac832e0000000p–3, 0x1.65b6980000000p–3, –0x1.4c95340000000p–7, –0x1.2fa9000000000p–2, 0x1.c6d3000000000p–5, –0x1.61c9820000000p–4, –0x1.72ddb00000000p–8, 0x1.f03a640000000p–4, –0x1.a2a4380000000p–6, –0x1.dbd2480000000p–7, –0x1.b9cc720000000p–9, 0x1.864d3c0000000p–3, 0x1.9eee740000000p–11,
–0x1.2ef2c00000000p–4, –0x1.24f5e80000000p–5, –0x1.b49dac0000000p–7, –0x1.e964380000000p–3, –0x1.ee7ba80000000p–3, –0x1.2bcc380000000p–7, 0x1.96cca00000000p–4, 0x1.2851820000000p–3, 0x1.7d7e960000000p–3, 0x1.b497e60000000p–5, –0x1.2474aa0000000p–3, –0x1.431cb20000000p–3, –0x1.11e7640000000p–5, –0x1.881d880000000p–11, –0x1.824e1c0000000p–2, –0x1.1d943a0000000p–3,
–0x1.8195c20000000p–7, –0x1.1e0db80000000p–11, –0x1.2e26160000000p–7, 0x1.1faba80000000p–8, –0x1.e4e1740000000p–5, –0x1.7007e80000000p–10, –0x1.2e25dc0000000p–1, 0x1.94c9340000000p–7, –0x1.3e95860000000p–2, –0x1.bd377e0000000p–10, –0x1.d64f260000000p–5, –0x1.dd50240000000p–7, –0x1.e6d9c80000000p–10, –0x1.be536a0000000p–15, –0x1.9732340000000p–2, 0x1.9481940000000p–10,
–0x1.3b94560000000p–8, 0x1.964b1c0000000p–9, –0x1.b7ed800000000p–7, 0x1.9c5f640000000p–4, –0x1.6687220000000p–3, –0x1.c544900000000p–11, –0x1.e4c7da0000000p–3, –0x1.041d980000000p–5, –0x1.43a4a20000000p–3, –0x1.70d74a0000000p–8, –0x1.c83f6e0000000p–6, –0x1.c1cb1c0000000p–4, 0x1.f1ecf60000000p–10, –0x1.4aff3c0000000p–11, –0x1.aa2cae0000000p–3, 0x1.b2ca200000000p–7,
0x1.6094080000000p–6, 0x1.43e11a0000000p–8, –0x1.208d080000000p–6, –0x1.1ae7a80000000p–3, 0x1.96359a0000000p–5, 0x1.c54bde0000000p–6, 0x1.bea1c00000000p–3, –0x1.a7d3ca0000000p–3, –0x1.3de3280000000p–5,

Parameter value
0x1.4bad560000000p–5, –0x1.af75fa0000000p–7, –0x1.3cd84e0000000p–4, 0x1.04bb300000000p–5, 0x1.77b5b80000000p–7, 0x1.8150740000000p–4, –0x1.f5e5480000000p–5,

Table E.18 (continued)

Parameter value
0x1.0f2b9c0000000p–2, 0x1.0d95f20000000p–5, –0x1.3bc8000000000p–4, –0x1.48a3d60000000p–3, –0x1.be65020000000p–3, 0x1.fefafa0000000p–6, 0x1.8610860000000p–3, –0x1.0a93ea0000000p–4, –0x1.4ade8c0000000p–4, –0x1.eff9f60000000p–8, 0x1.638a760000000p–3, 0x1.9900b40000000p–4, 0x1.b1e2400000000p–5, 0x1.6d48620000000p–7, –0x1.123b160000000p–2, 0x1.d660560000000p–5,
–0x1.b48b320000000p–4, –0x1.13cbc20000000p–7, –0x1.787cde0000000p–3, –0x1.04b6780000000p–2, –0x1.20e36e0000000p–4, –0x1.06acea0000000p–10, 0x1.190f860000000p–2, 0x1.0e42100000000p–5, –0x1.a9c9b80000000p–5, 0x1.0ce2760000000p–6, –0x1.408e800000000p–4, 0x1.0c2cbe0000000p–4, –0x1.86e8e00000000p–8, 0x1.4dc4d60000000p–9, –0x1.3acc240000000p–2, –0x1.4044280000000p–5,
0x1.848fce0000000p–6, 0x1.e48f140000000p–8, –0x1.6ee7120000000p–6, 0x1.8abe280000000p–6, –0x1.f0c4c60000000p–4, 0x1.6e83bc0000000p–5, 0x1.73edb40000000p–2, –0x1.9709240000000p–3, 0x1.d274960000000p–4, 0x1.2c14180000000p–4, –0x1.1380c00000000p–3, –0x1.ff261e0000000p–3, 0x1.a344360000000p–5, 0x1.527b420000000p–6, –0x1.7f902a0000000p–3, –0x1.da980e0000000p–4,
–0x1.91dcc20000000p–6, 0x1.d193500000000p–6, –0x1.748a520000000p–3, –0x1.d5338c0000000p–4, –0x1.95d5880000000p–3, 0x1.6ee9160000000p–8, 0x1.e366fc0000000p–2, –0x1.43ae980000000p–5, 0x1.06ba340000000p–3, –0x1.6647a00000000p–5, –0x1.04eb8e0000000p–3, –0x1.7d341e0000000p–3, 0x1.98b6740000000p–6, 0x1.0fa1140000000p–9, –0x1.8752f60000000p–1, 0x1.cf6efa0000000p–4,
–0x1.1ac8860000000p–8, –0x1.21737a0000000p–10, 0x1.a174960000000p–10, –0x1.a6b3fa0000000p–3, 0x1.9d8d720000000p–6, –0x1.5779380000000p–9, –0x1.43b12e0000000p–2, –0x1.89e7520000000p–4, 0x1.5719ce0000000p–5, –0x1.63dec00000000p–9, 0x1.9177a80000000p–5, 0x1.54c06e0000000p–5, –0x1.a7adda0000000p–9, –0x1.1893720000000p–10, –0x1.a9421c0000000p–9, 0x1.73f4140000000p–9,

Table E.19 Layer 2 CNN (bias parameter bias) of context coding neural network parameter

Parameter value
–0x1.e93a5a0000000p–1, 0x1.20d5680000000p–1, 0x1.4360d60000000p+0, 0x1.3302100000000p–6, –0x1.19b6760000000p–5, 0x1.6ef2000000000p–6, 0x1.0e89020000000p–1, –0x1.dae8fa0000000p–5, –0x1.2e19580000000p–3, 0x1.828eea0000000p–1, 0x1.1e9b100000000p–1, –0x1.a1aba40000000p–6, 0x1.e97bb40000000p–3, 0x1.96cef80000000p–1, –0x1.15e39a0000000p–6, 0x1.88d3740000000p–1,

Table E.18 (continued)

Parameter value
0x1.776cdc0000000p–13, 0x1.2d54bc0000000p–14, 0x1.ee43440000000p–15, 0x1.d0f0f20000000p–14, 0x1.029d440000000p+0, –0x1.828bbe0000000p–14, –0x1.3fb09e0000000p–14, 0x1.203b6a0000000p–14, 0x1.a43e940000000p–16, –0x1.1d505c0000000p–14, –0x1.596b880000000p–17, 0x1.72c4de0000000p–15, 0x1.862c3e0000000p–1, 0x1.5399ae0000000p–16, 0x1.10935a0000000p–14, –0x1.6bc7fe0000000p–15,
0x1.a49b9a0000000p–5, –0x1.51880a0000000p–3, 0x1.99b6120000000p–5, 0x1.e9ae800000000p–5, –0x1.09e4560000000p–2, –0x1.6e3a160000000p–4, 0x1.21c6320000000p–4, 0x1.ccee6e0000000p–6, 0x1.ae2d720000000p–4, –0x1.7d4eb20000000p–3, 0x1.faf2880000000p–5, –0x1.c392580000000p–5, –0x1.43d7700000000p–3, –0x1.d0e4e60000000p–6, –0x1.1867f60000000p–5, –0x1.fcb3c40000000p–4,
–0x1.e8158a0000000p–13, –0x1.24c26c0000000p–15, –0x1.f38de80000000p–16, –0x1.5a531a0000000p–14, 0x1.03659e0000000p+0, 0x1.5619d40000000p–14, 0x1.284dd80000000p–13, –0x1.54dcce0000000p–14, 0x1.46049e0000000p–13, 0x1.5b05c00000000p–15, 0x1.1b3bfe0000000p–14, –0x1.7340460000000p–14, 0x1.f859620000000p–1, –0x1.42e9160000000p–14, –0x1.337c220000000p–13, 0x1.2db4760000000p–16,
0x1.5db3a80000000p–3, 0x1.99dcd00000000p–5, –0x1.a0a4560000000p–5, 0x1.10d6a40000000p–3, 0x1.ef17540000000p–3, –0x1.77060e0000000p–5, –0x1.50b30e0000000p–4, 0x1.7e6e600000000p–5, –0x1.2375d60000000p–3, 0x1.caf4f60000000p–4, 0x1.b31f4c0000000p–3, 0x1.f058420000000p–6, –0x1.2f4ec60000000p–3, 0x1.7027740000000p–3, –0x1.b5fa6c0000000p–7, 0x1.c3b51e0000000p–4,
0x1.13c71e0000000p–3, –0x1.42e6ee0000000p–3, 0x1.820c420000000p–3, 0x1.46da540000000p–3, –0x1.25af2c0000000p–1, –0x1.24f1ac0000000p–11, –0x1.f7579e0000000p–4, –0x1.a124dc0000000p–7, 0x1.69b8d80000000p–3, 0x1.d2814c0000000p–7, –0x1.bfb3760000000p–5, 0x1.9ea47e0000000p–5, –0x1.2007640000000p–3, 0x1.6718160000000p–3, –0x1.a25cb40000000p–4, –0x1.4f07a60000000p–4,

Parameter value
0x1.bd90700000000p–6, 0x1.1980580000000p–4, –0x1.d4cdc00000000p–7, –0x1.2862380000000p–3, –0x1.733a780000000p–2, –0x1.fde1080000000p–5, –0x1.4618ea0000000p–5, 0x1.b723100000000p–4, 0x1.185f980000000p–11, 0x1.dec2920000000p–6, –0x1.9e9da40000000p–6, 0x1.7351da0000000p–5, –0x1.e5d3ca0000000p–2, 0x1.a0aab20000000p–5, 0x1.7af3a00000000p–5, –0x1.1c7dbe0000000p–7,
–0x1.4f7ec00000000p–15, –0x1.9e0c540000000p–13, 0x1.1a3d3c0000000p–17, 0x1.cfea120000000p–12, 0x1.ce459a0000000p+0, 0x1.06ef3c0000000p–14, –0x1.0b775e0000000p–12, 0x1.b5fa9e0000000p–13, –0x1.435a460000000p–14, 0x1.823c9a0000000p–17, –0x1.cd4eac0000000p–14, 0x1.e28a4e0000000p–13, 0x1.bbb7a40000000p–2, –0x1.c2a8b60000000p–14, –0x1.f692700000000p–20, –0x1.3ab5740000000p–15,

Table E.18 (continued)

Parameter value
0x1.44f4e00000000p–4, –0x1.4d32a00000000p–3, 0x1.a3e4ee0000000p–4, –0x1.c5591c0000000p–5, –0x1.34bfbc0000000p–10, –0x1.ca7ff00000000p–5, –0x1.80fdfe0000000p–4, 0x1.0456840000000p–5, –0x1.035dfa0000000p–3, 0x1.e869da0000000p–3, 0x1.60a9da0000000p–5, –0x1.0a3bfc0000000p–5, –0x1.d1dff20000000p–5, –0x1.f343320000000p–5, –0x1.4f46d40000000p–4, –0x1.be75380000000p–3,
0x1.3d87d80000000p–3, –0x1.05d9100000000p–3, –0x1.78eaf00000000p–3, –0x1.3dd8540000000p–3, –0x1.0da3660000000p–1, –0x1.e7ed240000000p–5, 0x1.687ca40000000p–3, –0x1.441d200000000p–3, –0x1.c5df420000000p–5, 0x1.ad379e0000000p–4, –0x1.0e6fe60000000p–5, –0x1.4843e20000000p–4, –0x1.402d1a0000000p–1, –0x1.44df160000000p–5, 0x1.01734a0000000p–4, 0x1.4c4c560000000p–4,
–0x1.3314c20000000p–2, 0x1.23f38e0000000p–7, –0x1.72a6ea0000000p–4, –0x1.ad644a0000000p–3, 0x1.79e03c0000000p–8, 0x1.cf43b40000000p–3, 0x1.95f07c0000000p–3, –0x1.a9966a0000000p–6, 0x1.bd50ec0000000p–4, 0x1.2c4b280000000p–4, 0x1.23b6480000000p–6, –0x1.49c0d20000000p–4, –0x1.0d285a0000000p–1, 0x1.49fb9e0000000p–3, –0x1.2097920000000p–6, –0x1.18e8e00000000p–6,
–0x1.0bac2a0000000p–12, 0x1.e682060000000p–13, –0x1.7e6a340000000p–13, –0x1.134f280000000p–13, 0x1.15e42c0000000p+0, –0x1.287d860000000p–13, 0x1.17d6ce0000000p–13, –0x1.1446540000000p–13, 0x1.6f54440000000p–17, –0x1.2998880000000p–14, –0x1.918a760000000p–16, –0x1.c6dbf80000000p–14, 0x1.ac971c0000000p–1, 0x1.5650d80000000p–15, 0x1.c5d4ce0000000p–14, 0x1.cf77220000000p–16,

Parameter value
0x1.020fc20000000p–3, 0x1.514a9a0000000p–3, –0x1.0eb98a0000000p–4, –0x1.7b3da20000000p–5, –0x1.77790e0000000p–3, –0x1.e5cbf40000000p–3, –0x1.eb18600000000p–4, –0x1.5431d20000000p–3, –0x1.7551260000000p–4, 0x1.ae58960000000p–3, –0x1.d97cbe0000000p–3, –0x1.fb0c280000000p–7, 0x1.e3d4fa0000000p–3, –0x1.cb830a0000000p–3, 0x1.a873000000000p–5, 0x1.d2afca0000000p–6,
0x1.989a060000000p–4, 0x1.06c3140000000p–5, 0x1.604fc40000000p–6, 0x1.86390a0000000p–3, –0x1.06904c0000000p–1, 0x1.d278a80000000p–9, –0x1.fd7b600000000p–4, –0x1.3eb03e0000000p–4, –0x1.73e3b60000000p–4, 0x1.e0e46e0000000p–6, –0x1.f5318e0000000p–6, 0x1.fc42120000000p–6, –0x1.32195a0000000p–1, –0x1.1811080000000p–4, –0x1.e27b560000000p–6, 0x1.eee9900000000p–5,
–0x1.dec0120000000p–4, –0x1.9a232a0000000p–6, –0x1.8bb8540000000p–8, –0x1.349a980000000p–3, –0x1.2c6e440000000p–1, –0x1.3455a40000000p–7, 0x1.5c8f240000000p–3, 0x1.390aae0000000p–9, 0x1.ada89e0000000p–8, 0x1.17073a0000000p–5, 0x1.83651e0000000p–5, –0x1.b5eb040000000p–6, –0x1.4836a40000000p–2, –0x1.815aae0000000p–5, 0x1.2738060000000p–4, 0x1.0d5aea0000000p–5,

Table E.18 (continued)

Parameter value
–0x1.edf9340000000p–5, –0x1.0240400000000p–6, –0x1.f664860000000p–6, –0x1.489e4a0000000p–5, 0x1.ef5b0a0000000p–3, 0x1.75b8220000000p–3, –0x1.fbccf40000000p–5, –0x1.b799840000000p–7, –0x1.0c2b8a0000000p–4, –0x1.3e45780000000p–6, 0x1.4388420000000p–9, –0x1.e264820000000p–8, –0x1.1783480000000p–1, 0x1.7f94200000000p–5, 0x1.148da00000000p–4, –0x1.84d3680000000p–7,
–0x1.38099c0000000p–3, 0x1.1470da0000000p–5, –0x1.a4ef4e0000000p–5, –0x1.09f9080000000p–3, –0x1.01b5b80000000p+0, 0x1.cd66880000000p–4, 0x1.6a8bbc0000000p–4, –0x1.7803040000000p–8, 0x1.26ef820000000p–5, 0x1.1167140000000p–4, –0x1.4fd1980000000p–11, –0x1.b6251a0000000p–6, –0x1.2d48860000000p–4, 0x1.500db60000000p–4, 0x1.5e80920000000p–8, 0x1.6f82480000000p–7,
0x1.c28c020000000p–14, 0x1.0c130c0000000p–14, –0x1.5f46aa0000000p–16, 0x1.b49a440000000p–14, 0x1.2a07c20000000p+0, 0x1.5b18aa0000000p–17, –0x1.127c300000000p–12, 0x1.f621ae0000000p–14, 0x1.9df7020000000p–13, 0x1.2d23000000000p–14, –0x1.dd205e0000000p–14, –0x1.ba04200000000p–16, –0x1.9bc80a0000000p–1, –0x1.f744520000000p–14, –0x1.0474d40000000p–19, –0x1.d37cde0000000p–15,

Parameter value
–0x1.e415520000000p–3, 0x1.fd75cc0000000p–4, –0x1.0a47c60000000p–3, –0x1.acd9bc0000000p–4, –0x1.80a6ea0000000p–1, 0x1.0b7b8a0000000p–7, –0x1.e1351c0000000p–6, –0x1.7493980000000p–7, 0x1.54398c0000000p–3, 0x1.f83f160000000p–4, 0x1.286e840000000p–6, 0x1.27e8660000000p–3, 0x1.1384680000000p–2, 0x1.a5fa3c0000000p–5, –0x1.b0650a0000000p–8, –0x1.3ef8c40000000p–4,
–0x1.3c42ec0000000p–12, 0x1.f2c54a0000000p–16, –0x1.4304040000000p–15, –0x1.da34580000000p–13, 0x1.1d2d940000000p+0, 0x1.8f11c60000000p–14, 0x1.49b9da0000000p–13, –0x1.9f56500000000p–15, 0x1.08f82a0000000p–12, –0x1.c230500000000p–17, 0x1.8277da0000000p–14, –0x1.ef67900000000p–15, –0x1.c089bc0000000p–1, 0x1.b2c78a0000000p–15, –0x1.a9076c0000000p–19, –0x1.31c98a0000000p–17,
0x1.9936a20000000p–9, 0x1.b5ab8c0000000p–11, –0x1.bb04d40000000p–12, 0x1.49419e0000000p–9, –0x1.3d1ce60000000p+0, –0x1.ea420a0000000p–13, –0x1.f6256e0000000p–10, –0x1.3253040000000p–10, –0x1.4fa75c0000000p–10, –0x1.e4d5ba0000000p–11, 0x1.0cc3ea0000000p–10, 0x1.10273c0000000p–9, 0x1.b428480000000p–1, 0x1.2d4c560000000p–12, –0x1.4e92460000000p–11, 0x1.7288d80000000p–9,
0x1.9187ee0000000p–11, –0x1.0f908e0000000p–3, 0x1.3f89020000000p–4, 0x1.c533940000000p–3, 0x1.ea923c0000000p–6, 0x1.dc91440000000p–4, 0x1.72e4960000000p–5, –0x1.70c28e0000000p–3, –0x1.2c67c60000000p–3, –0x1.79dfae0000000p–4, 0x1.4ae9820000000p–3, 0x1.9ec0da0000000p–3, –0x1.3982f60000000p–3, –0x1.23c1800000000p–3, 0x1.c890a80000000p–5, 0x1.81ce460000000p–3,

Table E.18 (continued)

Parameter value
0x1.a458940000000p–4, 0x1.4f44c20000000p–4, 0x1.26ee020000000p–3, 0x1.0baa380000000p–3, –0x1.8602f60000000p–2, –0x1.21f83c0000000p–7, –0x1.e442580000000p–7, –0x1.b4bfd40000000p–4, –0x1.6f73780000000p–4, –0x1.4b3bd00000000p–4, –0x1.c1dca80000000p–6, –0x1.0336640000000p–3, 0x1.1810a00000000p–1, 0x1.5f7fc80000000p–5, –0x1.2b34400000000p–5, 0x1.c9a6ac0000000p–4,
–0x1.53c7c00000000p–12, –0x1.f766e40000000p–13, –0x1.1e3c1e0000000p–12, –0x1.80f1ae0000000p–13, 0x1.f03b540000000p–1, 0x1.352c1c0000000p–18, –0x1.28ddca0000000p–13, 0x1.089a300000000p–15, 0x1.87802c0000000p–14, 0x1.b4b7720000000p–16, –0x1.1116120000000p–12, –0x1.417dc80000000p–18, –0x1.94fc060000000p–1, 0x1.103f4c0000000p–16, 0x1.27d0600000000p–14, –0x1.c8bce80000000p–13,

Parameter value
0x1.007f2e0000000p–3, 0x1.8d798c0000000p–4, 0x1.934ea60000000p–3, 0x1.edbe520000000p–4, –0x1.3c1e100000000p–3, 0x1.5a13c00000000p–4, 0x1.16f4fc0000000p–3, 0x1.abd4a60000000p–3, 0x1.394ba40000000p–7, –0x1.1f74bc0000000p–4, 0x1.e26f0c0000000p–4, –0x1.ccd18e0000000p–5, –0x1.87ffac0000000p–3, –0x1.173be80000000p–4, –0x1.0474340000000p–3, –0x1.e4bc3a0000000p–3,
0x1.8cc47a0000000p–4, –0x1.b2bd580000000p–3, –0x1.7d15640000000p–3, 0x1.93e30a0000000p–3, 0x1.04c6800000000p–3, –0x1.75409a0000000p–7, 0x1.3ee17e0000000p–3, –0x1.ab17ee0000000p–4, 0x1.c304660000000p–6, –0x1.0ecba20000000p–5, –0x1.9c9f020000000p–5, 0x1.3b39f80000000p–3, –0x1.07113e0000000p–3, 0x1.014dbe0000000p–3, 0x1.38e1180000000p–3, –0x1.298ebc0000000p–6,
–0x1.d7ec500000000p–5, –0x1.71324c0000000p–4, –0x1.99310c0000000p–3, –0x1.9b746c0000000p–4, –0x1.8432aa0000000p–2, 0x1.2a466c0000000p–6, 0x1.0b087c0000000p–4, –0x1.3ad6660000000p–4, –0x1.9040f40000000p–6, –0x1.b284620000000p–5, –0x1.fa54480000000p–5, 0x1.d6446a0000000p–7, 0x1.0047060000000p–2, –0x1.f804aa0000000p–6, 0x1.5c756a0000000p–4, 0x1.cb99ac0000000p–5,
–0x1.0a26800000000p–12, 0x1.00bc600000000p–15, –0x1.c614ce0000000p–14, –0x1.e22b380000000p–19, 0x1.1653780000000p+0, 0x1.2d3d340000000p–14, 0x1.7edc4a0000000p–13, 0x1.0a2e8c0000000p–16, –0x1.bec9fc0000000p–13, –0x1.6dc5fa0000000p–15, 0x1.0afae60000000p–15, –0x1.2601bc0000000p–13, –0x1.cabb220000000p–1, 0x1.73c6280000000p–14, 0x1.128e6c0000000p–14, 0x1.8e4af00000000p–15,
–0x1.fed4500000000p–3, 0x1.51c8040000000p–5, –0x1.b8b9f00000000p–4, 0x1.c4cd9a0000000p–5, 0x1.a0f5c40000000p–5, 0x1.dc8b500000000p–4, –0x1.67df6a0000000p–4, 0x1.5f8abe0000000p–7, 0x1.4f8b8a0000000p–5, –0x1.c4fd180000000p–4, 0x1.ea2ce80000000p–5, –0x1.0544da0000000p–6, –0x1.2018400000000p–4, 0x1.cd7c8a0000000p–4, 0x1.a19ac00000000p–4, –0x1.b1a2080000000p–3,

Table E.18 (continued)

Parameter value
0x1.8a2d9a0000000p–5, –0x1.97e10a0000000p–5, 0x1.0c95d00000000p–4, –0x1.e4d72c0000000p–8, –0x1.2e96640000000p–1, 0x1.1be06e0000000p–7, 0x1.1246460000000p–7, 0x1.8759e20000000p–4, –0x1.c328640000000p–7, 0x1.9eb0b60000000p–6, –0x1.78b3140000000p–11, 0x1.e37b820000000p–6, 0x1.2d7fc60000000p–1, –0x1.9657f20000000p–5, 0x1.6245740000000p–9, –0x1.07cdae0000000p–4,

Parameter value
0x1.5032300000000p–5, –0x1.ce7e300000000p–4, –0x1.d923d20000000p–4, 0x1.2906fc0000000p–7, –0x1.8500420000000p–1, –0x1.83c1660000000p–5, –0x1.0207ce0000000p–5, 0x1.594d860000000p–5, 0x1.f5afc60000000p–6, –0x1.9a87460000000p–6, 0x1.fc75e20000000p–4, –0x1.a710600000000p–5, –0x1.6df2be0000000p–9, 0x1.b47d760000000p–6, –0x1.4a76c20000000p–6, 0x1.1da58a0000000p–7,
–0x1.abda9e0000000p–12, 0x1.5aa4140000000p–16, –0x1.2450c00000000p–10, –0x1.34c4cc0000000p–16, –0x1.b0bae20000000p+0, 0x1.4ffcea0000000p–10, –0x1.84679c0000000p–11, 0x1.59f2d40000000p–11, –0x1.422d860000000p–10, –0x1.b96ca60000000p–14, –0x1.3648d60000000p–12, 0x1.6c7e880000000p–11, 0x1.85a63c0000000p–3, 0x1.38500a0000000p–11, 0x1.475f400000000p–11, –0x1.3347780000000p–11,
0x1.4a21d60000000p–6, –0x1.e35e720000000p–5, –0x1.12e4740000000p–4, –0x1.3325aa0000000p–6, –0x1.79cda00000000p–1, 0x1.de5a080000000p–9, 0x1.049ef00000000p–5, –0x1.5a90740000000p–5, –0x1.5ee4fc0000000p–5, –0x1.ab85b20000000p–5, –0x1.ba515c0000000p–6, –0x1.eb079c0000000p–6, 0x1.2ce6360000000p–2, –0x1.09ae860000000p–6, 0x1.28d9fe0000000p–5, 0x1.916fa60000000p–5,
–0x1.adab320000000p–13, 0x1.01ab520000000p–14, –0x1.2a942a0000000p–13, –0x1.dc1f8a0000000p–14, –0x1.737a400000000p–6, 0x1.0486700000000p–13, 0x1.0a0dbc0000000p–15, 0x1.52a1640000000p–14, 0x1.c904f80000000p–15, 0x1.3db2b20000000p–13, 0x1.0eec840000000p–14, –0x1.6a1d980000000p–13, –0x1.34644e0000000p–9, –0x1.9c9d9e0000000p–14, –0x1.680fb40000000p–13, –0x1.bc87080000000p–13,
–0x1.6629440000000p–3, –0x1.8d81e20000000p–4, –0x1.fb5efe0000000p–4, –0x1.8fc7e20000000p–8, –0x1.38387a0000000p–5, –0x1.a0903a0000000p–4, 0x1.3569ec0000000p–4, 0x1.ba10ce0000000p–5, 0x1.219e840000000p–4, 0x1.7d39d40000000p–6, 0x1.e2ca740000000p–5, 0x1.b38c7e0000000p–3, –0x1.7d84860000000p–6, –0x1.6483d60000000p–4, –0x1.7581640000000p–4, 0x1.807f5c0000000p–3,
0x1.12c8200000000p–12, 0x1.0a01a20000000p–14, 0x1.d272c60000000p–13, 0x1.b238ea0000000p–15, 0x1.3b54860000000p–4, 0x1.67e85c0000000p–15, –0x1.9fb3740000000p–15, –0x1.942ad40000000p–14, 0x1.ea9b9e0000000p–14, –0x1.3e88140000000p–13, 0x1.d0887e0000000p–15, –0x1.492f3a0000000p–14, –0x1.1fb5ea0000000p–6, –0x1.57bfd00000000p–16, 0x1.abd8880000000p–14, –0x1.8317b80000000p–16,

Table E.18 (continued)

Parameter value
0x1.046b780000000p–5, 0x1.35f2ea0000000p–7, 0x1.716ce80000000p–4, 0x1.cba37c0000000p–3, –0x1.4ea90c0000000p–3, 0x1.a5bce80000000p–3, 0x1.f62ab00000000p–5, –0x1.5f17780000000p–5, 0x1.2dc8d80000000p–4, 0x1.d973200000000p–3, –0x1.38b9e40000000p–3, 0x1.fab0de0000000p–3, –0x1.833a780000000p–3, 0x1.d38ee80000000p–6, 0x1.61edda0000000p–4, 0x1.3640020000000p–5,
0x1.a14d320000000p–4, 0x1.d115960000000p–3, –0x1.8f5d000000000p–3, 0x1.944e1c0000000p–4, 0x1.052fb20000000p–16, –0x1.c0d73c0000000p–4, 0x1.ae2a980000000p–3, 0x1.93e4c00000000p–8, 0x1.97d5040000000p–3, 0x1.3210700000000p–3, –0x1.ac534a0000000p–4, –0x1.8829a20000000p–9, 0x1.5afac00000000p–3, 0x1.887a740000000p–4, 0x1.284cb00000000p–5, 0x1.aae2ec0000000p–3,
–0x1.56dbac0000000p–4, 0x1.b6c3e40000000p–4, –0x1.081bf40000000p–3, –0x1.f52fac0000000p–4, 0x1.f7a9e00000000p–4, 0x1.5296200000000p–6, 0x1.11c1ea0000000p–3, –0x1.44c1760000000p–5, –0x1.4ca4dc0000000p–3, 0x1.283e3e0000000p–4, 0x1.2f24a00000000p–5, –0x1.06edfc0000000p–4, –0x1.39de4e0000000p–3, 0x1.f9610a0000000p–5, 0x1.76c06e0000000p–5, –0x1.5ca87a0000000p–4,
–0x1.f6e0cc0000000p–18, –0x1.9a24ea0000000p–13, 0x1.e45cce0000000p–13, 0x1.4d39740000000p–12, –0x1.3627620000000p–5, –0x1.9552ce0000000p–12, –0x1.fbfa420000000p–13, –0x1.9b47200000000p–15, 0x1.791dfa0000000p–12, 0x1.7a82760000000p–16, –0x1.7674860000000p–15, 0x1.57340c0000000p–13, –0x1.2815240000000p–7, 0x1.32605a0000000p–13, –0x1.1f2b4e0000000p–13, –0x1.ac63860000000p–13,
–0x1.be21240000000p–3, 0x1.c0c0400000000p–4, –0x1.0093780000000p–8, –0x1.0cd1900000000p–3, 0x1.d507e20000000p–4, 0x1.3c03960000000p–3, –0x1.6f38a20000000p–5, 0x1.cc12900000000p–3, –0x1.516a860000000p–3, –0x1.4b4dc40000000p–3, 0x1.4c2fc00000000p–5, 0x1.d1097e0000000p–3, 0x1.fcd2bc0000000p–3, –0x1.e3e4f00000000p–4, 0x1.0e0ccc0000000p–4, –0x1.7d9a0a0000000p–7,
0x1.94e0ba0000000p–3, 0x1.cbac700000000p–4, 0x1.2008380000000p–4, 0x1.bfbfa80000000p–3, 0x1.0053a40000000p–3, –0x1.f4a76e0000000p–4, 0x1.01297a0000000p–3, 0x1.62bb500000000p–3, 0x1.b60ac00000000p–4, –0x1.0e43540000000p–5, –0x1.75ad9e0000000p–3, 0x1.0a5f6a0000000p–3, –0x1.48d9fa0000000p–3, –0x1.5280300000000p–4, –0x1.68eb180000000p–5, –0x1.6891820000000p–6,
–0x1.ae0ea80000000p–7, –0x1.42e21a0000000p–5, 0x1.fe90520000000p–5, –0x1.9eea3c0000000p–5, –0x1.0ad2b20000000p–2, 0x1.d55a400000000p–7, –0x1.a32f3c0000000p–4, –0x1.e587be0000000p–8, 0x1.573f4a0000000p–5, –0x1.e69eb00000000p–7, 0x1.fd85800000000p–5, –0x1.967dd60000000p–6,

Parameter value
0x1.23c11c0000000p–4, 0x1.8f86940000000p–5, 0x1.e656c80000000p–6, –0x1.c1efd00000000p–4,

Table E.18 (continued)

Parameter value
–0x1.d327d20000000p–14, 0x1.e6b2480000000p–17, 0x1.0336ce0000000p–13, 0x1.36571e0000000p–14, –0x1.4caa1c0000000p–5, 0x1.53ba9c0000000p–16, –0x1.1369360000000p–13, 0x1.e3b7f80000000p–16, –0x1.40c00e0000000p–16, –0x1.a4d1ac0000000p–15, 0x1.750cc60000000p–17, –0x1.1e82880000000p–13, 0x1.0312140000000p–5, 0x1.05a3c20000000p–15, 0x1.11c31c0000000p–14, 0x1.3a6a740000000p–16,
0x1.f8a0640000000p–4, 0x1.70a21e0000000p–4, –0x1.13e8c40000000p–4, 0x1.984e760000000p–3, 0x1.ab042a0000000p–3, 0x1.702faa0000000p–6, –0x1.f2dcea0000000p–3, 0x1.5172e20000000p–4, –0x1.edc9180000000p–3, –0x1.44fc820000000p–3, 0x1.b868b40000000p–4, –0x1.0d201c0000000p–4, 0x1.c96d4c0000000p–4, –0x1.8f41620000000p–6, –0x1.8e12940000000p–4, –0x1.44f6660000000p–4,
0x1.e97fbe0000000p–4, –0x1.62608e0000000p–5, 0x1.129df60000000p–3, 0x1.7003d00000000p–4, 0x1.e639260000000p–6, 0x1.44df2a0000000p–6, –0x1.eee2160000000p–4, 0x1.4733780000000p–7, 0x1.75ff860000000p–4, –0x1.e82ef00000000p–5, –0x1.7fb0680000000p–5, –0x1.b2b6800000000p–6, 0x1.cb094a0000000p–5, –0x1.dc3e1e0000000p–7, –0x1.e3d82e0000000p–10, –0x1.869b540000000p–9,
0x1.65aa2a0000000p–4, 0x1.a9ff120000000p–4, –0x1.4742e00000000p–5, 0x1.e3d9c60000000p–5, –0x1.9777c80000000p–3, 0x1.bee9900000000p–6, 0x1.d4845c0000000p–4, –0x1.18d1dc0000000p–6, –0x1.9df9e20000000p–4, 0x1.056ef40000000p–6, –0x1.afbffa0000000p–4, –0x1.0f363c0000000p–4, 0x1.ace6640000000p–4, –0x1.58e6ee0000000p–6, 0x1.3006580000000p–8, 0x1.c944da0000000p–5,
0x1.53f07c0000000p–5, 0x1.4aaaba0000000p–5, –0x1.7e1e560000000p–7, 0x1.d959b40000000p–4, 0x1.013dd00000000p–4, –0x1.22d3000000000p–3, –0x1.c76bd20000000p–8, 0x1.9bdc6e0000000p–3, –0x1.0b51a80000000p–7, –0x1.09d9180000000p–3, 0x1.7550940000000p–4, 0x1.7ce65a0000000p–6, 0x1.701d280000000p–6, 0x1.70df240000000p–7, –0x1.c81be20000000p–5, –0x1.a3c1920000000p–5,
0x1.0ba3820000000p–6, 0x1.9259980000000p–7, 0x1.0ba80c0000000p–5, –0x1.ffe6440000000p–6, –0x1.d515840000000p–4, 0x1.a2d04c0000000p–6, –0x1.44b43e0000000p–5, –0x1.087e640000000p–6, –0x1.9056040000000p–7, –0x1.29f58a0000000p–9, 0x1.2acef40000000p–6, –0x1.b341060000000p–5,

Parameter value
0x1.b276180000000p–5, 0x1.4adf040000000p–5, 0x1.06bc0c0000000p–5, –0x1.58b5ee0000000p–4,

Annex F
(Informative)
Metadata Coding

A simple quantization coding method is used for metadata coding. A metadata coding method may be derived based on a value range of each metadata described in each semantic in sections 9.1 to 9.16 and a dequantization method for each metadata described in section 9.17.

Annex G
(Informative)
Correspondence between Coding Metadata in this Document and Metadata in ITU-R BS.2076-2

Table G.1 shows the correspondence between the coding metadata in chapter 9 and the metadata in the ITU-R BS.2076-2.

Table G.1 Correspondence between Coding Metadata and Metadata in ITU-R BS.2076-2

Metadata element in ITU-R BS.2076-2	**Sub-element and attribute in ITU-R BS.2076-2**	**Coding metadata in chapter 9 in this document**	**Correspondence to metadata in ITU-R BS.2076-2**
audioTrackFormat	All sub-elements and attributes	None	—
audioStreamFormat	All sub-elements and attributes	None	—
audioChannelFormat	audioChannelFormatID	None	In this document, channelFormatIdx has the same meaning as audioChannelFormatID.
	typeLabel	None	In this document, the typeLabel in the audioChannelFormat metadata is consistent with the typeLabel in the audioPackFormat metadata at its upper layer.
	typeDefinition	None	—
	frequency	None	—
audioBlockFormat	audioBlockFormatID	None	—
	rtime	None	—
	duration	None	—
	gain	channelGainUnit, channelGain_QFlag, and channelGain	Consistent

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
	importance	importance	Consistent
	headLocked	b_headLocked	Consistent
	headphoneVirtualise: bypass	None	—

Table G.1 (continued)

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
audioBlockFormat	headphoneVirtualise: DRR	None	—
	speakerLabel (typelabel=directSpeakers)	None	When the value of packFormatID is 0 to 31, the packFormatID metadata is used in the ITU-R BS.2094-1 to obtain the speakerLabel described in GY/T 316-2018. When the value of packFormatID is 32 to 63, position metadata (the azimuth and the elevation) is used in GY/T 316-2018 to obtain the corresponding speakerLabel.
	position (typelabel=directSpeakers)	DirectSpeakersPosition	Consistent
	outputChannel FormatIDRef	None	—
	jumpPosition (typelabel=matrix)	None	—
	interpolationLength (typelabel=matrix)	None	—
	coefficient (typelabel=matrix)	None	In this document, matrixCoef metadata is used to adjust the linear gain in coefficient

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
			metadata.
	position (typelabel=object polar)	obj_position_azimuth, obj_position_elevation, obj_position_distance, obj_width_horizontal, obj_hight_vertical, and obj_depth_distance	Consistent
	position (typelabel=object Cartesian)	obj_position_x, obj_position_y, obj_position_z, obj_width_x, obj_width_y, and obj_width_z	Consistent
	cartesian	cartesianDm	Consistent
	diffuse	diffuse	Consistent
	channelLock	channelLock	Consistent
	maxDistance	channelLock_maxDistance	Consistent
	objectDivergence	objectDivergence	Consistent
	azimuthRange	objectDivergence_azimuthRange	Consistent

Table G.1 (continued)

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
audioBlockFormat	positionRange	None	—
	jumpPosition (typelabel=object)	jumpPosition	Consistent

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
	interpolationLength (typelabel=object)	None	In this document, the dynamic metadata in frames contains the result obtained through interpolationLength interpolation.
	zoneExclusion	None	—
	screenRef (typelabel=object)	obj_screenRef	Consistent
	equation	None	—
	order	hoaOrder	Consistent
	degree	None	This document uses the ACN SN3D manner.
	normalization	normalization	Consistent
	nfcRefDist	nfcRefDist	Consistent
	screenRef (typelabel=HOA)	screenRef	Consistent
audioPackFormat	audioPackFormatID	packFormatIdx	In this document, packFormatIdx has the same meaning as audioPackFormatID.
	audioPackFormatName	None	—
	typeLabel	typeLabel	Consistent
	typeDefinition	None	—
	importance	audioPackFormatImportance	Consistent
	audioChannelFormatIDRef	None	In this document, refChannelIdx has the same meaning as audioChannelFormatIDRef.

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
	audioPackFormatIDRef	None	In this document, audioPackFormat metadata is converted by referencing another audioPackFormat format into two independent audioPackFormat formats.

Table G.1 (continued)

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
audioPackFormat	absoluteDistance	absoluteDistance	Consistent
	encodePackFormatIDRef (typelabel=Matrix)	None	—
	decodePackFormatIDRef (typelabel=Matrix)	None	—
	inputPackFormatIDRef (typelabel=Matrix)	None	—
	outputPackFormatIDRef （typelabel=Matrix)	None	—
	normalization	normalization	Consistent
	nfcRefDist	nfcRefDist	Consistent
	screenRef (typelabel=HOA)	screenRef	Consistent
audioObject	audioObjectID	None	In this document, objectIdx has the same meaning as audioObjectID.

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
	audioObjectName	ObjectName	The character coding format is not limited in this document, and can be automatically identified based on coded binary data. However, the GB2312 or GBK character coding format is recommended for Chinese and English characters.
	language	audioObjectLanguage	Consistent
	start	None	—
	duration	None	—
	dialogue	Dialogue	Consistent
	importance	audioObjectImportance	Consistent
	interact	b_interact	Consistent
	disableDucking	b_disableDucking	Consistent
	audioPackFormatIDRef	None	In this document, refPackFormatIdx has the same meaning as audioPackFormatIDRef.

Table G.1 (continued)

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
audioObject	audioObjectIDRef	None	In this document, audioObject metadata is converted by referencing another audioObject format into two independent

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
			audioObject formats.
	audioComplementaryObjectGroupLabel	ComplementaryObjectIdx	Consistent
	audioComplementary ObjectIDRef		
	audioTrackUIDRef	None	—
	audioObjectInteraction	audioObjectInteraction	Consistent
	gain	objectGainQFlag and objectGain	Consistent
	gainUnit	objectGainUnit	Consistent
	headLocked	b_headLocked	Consistent
	positionOffset	None	—
	mute	b_mute	Consistent
	alternativeValueSet	None	—
audioContent	audioContentID	None	In this document, contentIdx has the same meaning as audioContentID.
	audioContentName	None	—
	language	audioContentLanguage	Consistent
	audioObjectIDRef	None	In this document, refObjectIdx has the same meaning as audioObjectIDRef.
	loudnessMetadata	loudnessMetadata	Consistent
	dialogue	Dialogue	Consistent

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
	alternativeValueSetIDRef	None	—

Table G.1 (continued)

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
audioProgramme	audioProgrammeID	None	—
	audioProgrammeName	None	—
	language	audioProgrammeLanguage	Consistent
	start	None	—
	end	None	—
	maxDuckingDepth	maxDuckingDepth	Consistent
	audioContentIDRef	refContentIdx	In this document, refContentIdx has the same meaning as audioContentIDRef.
	loudnessMetadata	loudnessMetadata	Consistent
	audioProgrammeReferenceScreen	b_audioProgrammeReferenceScreen	Consistent
	authoringInformation	None	—
	alternativeValueSetIDRef	None	—
audioTrackUID	All Sub-element and Attribute	None	—
loudnessMetadata	loudnessMethod	None	—

Metadata element in ITU-R BS.2076-2	Sub-element and attribute in ITU-R BS.2076-2	Coding metadata in chapter 9 in this document	Correspondence to metadata in ITU-R BS.2076-2
	loudnessRecType	None	—
	loudnessCorrectionType	None	—
	integratedLoudness	integratedLoudness	Consistent
	loudnessRange	loudnessRange	Consistent
	maxTruePeak	maxTruePeak	Consistent
	maxMomentary	maxMomentary	Consistent
	maxShortTerm	maxShortTerm	Consistent
	dialogueLoudness	dialogueLoudness	Consistent

References

[1] IEEE Std 1857.2TM-2013 IEEE Standard for Advanced Audio Coding

[2] ITU-R BS.2088 Long-form file format for the international exchange of audio programme materials with metadata

[3] J. Ballé, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," in Proc. Int. Conf. Learn. Represent., 2016, pp. 1–14.

[4] Wen Gao, Tiejun Huang, and Cliff Reader, Weibei Dou, Xilin Chen, "IEEE Standards for Advanced Audio and Video Coding in Emeraging Applications", Published by the IEEE Computer Society, MAY 2014, 81–83.