IEEE SA
STANDARDS ASSOCIATION

IEEE Standard for Advanced Audio Coding

STANDARDS

IEEE Computer Society

Developed by the
Data Compression Standards Committee



IEEE Std 1857.2™-2023
(Revision of IEEE Std 1857.2-2013)

IEEE

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1857.2-2023

IEEE Std 1857.2™-2023
(Revision of IEEE Std 1857.2-2013)

IEEE Standard for Advanced Audio Coding

Developed by the

Data Compression Standards Committee
of the
IEEE Computer Society

Approved 5 June 2023

IEEE SA Standards Board

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1857.2-2023

THIS IS A REDLINE DOCUMENT PRODUCED BY TECHSTREET, AN ASME COMPANY.

This document is intended to provide users with an indication of changes from one edition to the next. It includes a full-text version of the new document, plus an indication of changes from the previous version.

Redlines are designed to save time and improve efficiencies by using the latest software technology to find and highlight document changes. More professionals are using valuable new technologies like redlines, to help improve outcomes in a fast paced global business world.

Because it may not be technically possible to capture all changes accurately, it is recommended that users consult previous editions as appropriate. In all cases, only the current base version of this publication is to be considered the official document.

This Redline is produced solely by Techstreet. IEEE has not prepared this Redline and is not responsible for its accuracy.

445 Hoes Lane, Piscataway, NJ 08854 USA | http://standards.ieee.org | Tel. +1 732-981-0060 Fax +1 732-562-1571
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Redline Processing Notes:

1. ~~**Red Text**~~ - Red strikethrough text denotes deletions.

2. **Blue Text** - Blue underlined text denotes modifications and additions.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1857.2-2023

Abstract: A set of tools to support specific audio coding functions including general audio coding and lossless audio coding are defined in this standard. The tool set defined in combination provides regular high-quality and efficient coding tool sets for compression, decompression, processing, and representing of audio data to save bandwidth for transmission; to save space for storage; to speed up indexing and multimedia search; and to enhance performance when using mixed media for virtual reality and other applications that demand high bandwidth. The target applications and services include but are not limited to audio transmission, audio recording, Internet streaming, and other video/audio-enabled services and applications.

Keywords: audio coding, bitplane coding, entropy coding, IEEE 1857.2™, lossless coding, LPC, PQ-SPSC, scalable coding, vector quantization

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA
Copyright © 2023 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 25 September 2023. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-9801-2 STD26229
Print: ISBN 978-1-5044-9802-9 STDPD26229

IEEE prohibits discrimination, harassment and bullying. For more information, visit https://www.ieee.org/about/corporate/governance/p9-26.html.
No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1857.2-2023

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (https://standards.ieee.org/ipr/disclaimers.html), appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA, and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1857.2-2023

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents**.

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and Standards Coordinating Committees are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system.[1] An IEEE Account is needed to access the application.

Comments on standards should be submitted using the Contact Us form.[2]

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

[1] Available at: https://development.standards.ieee.org/myproject-web/public/view.html#landing.
[2] Available at: https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; https://www.copyright.com/. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore or contact IEEE.[3] For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website.[4] Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore. Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the IEEE SA Patent Policy.[5]

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at https://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing

[3] Available at: https://ieeexplore.ieee.org/browse/standards/collection/ieee.
[4] Available at: https://standards.ieee.org/standard/index.html.
[5] Available at: https://standards.ieee.org/about/sasb/patcom/materials.html.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1857.2-2023

Participants

At the time this draft standard was completed, the Audio Video Coding Working Group had the following membership:

Siwei Ma, *Chair*
Cliff Reader, *Vice Chair*

Weibei Dou
Haibin Huang
Yichao Huang
Jing Li
Min Lu
Xingde Pan
Tianshu Qu
Haiyan Shu
Shumin Xu
Xinhui Yang
Rongshan Yu
Tao Zhang

The following members of the individual Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Oscar Au
James Carroll
Eric Chai
Pin Chang
Xuesong Gao
Werner Hoelzl
Chang Huang
Piotr Karocki
Li Li
Fan Liang
Changyu Liu
Dong Liu
Jinghui Lu
Xiaoxu Luan
Siwei Ma
M. Sabarimalai Manikandan
Luntian Mou
Rajesh Murthy
Bansi Patel
Eugene Stoudenmire
Walter Struppler
Bo Sun
Yonghong Tian
John Vergis
Yaowei Wang
Feng Wu
Qianqian Xu
Yu Yuan
Oren Yuen

When the IEEE SA Standards Board approved this standard on 5 June 2023, it had the following membership:

David J. Law, *Chair*
Ted Burse, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

Sara R. Biyabani
Doug Edwards
Ramy Ahmed Fathy
Guido R. Hiertz
Yousef Kimiagar
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick
Joseph S. Levy
Howard Li
Gui Lin
Johnny Daozhuang Lin
Kevin W. Lu
Daleep C. Mohla
Andrew Myles
Paul Nikolich
Annette D. Reilly
Robby Robson
Lei Wang
F. Keith Waters
Karl Weber
Philip B. Winston
Don Wright

*Member Emeritus

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Introduction

This introduction is not part of IEEE Std 1857.2-2023, IEEE Standard for Advanced Audio Coding.

This standard provides regular high-quality and efficient coding tool sets for compression, decompression, and packaging of the multimedia data to save bandwidth for transmission on the Internet or to save space for storage.

Copyright © 2023 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Contents

1. Overview 9
1.1 Scope 9
1.2 Purpose 9
1.3 Word usage 9
1.4 Audio coding tools 9

2. Normative references 10

3. Definitions, acronyms, and abbreviations 10
3.1 Definitions 10
3.2 Symbols and abbreviations 13

4. General audio coding 17
4.1 Overview 17
4.2 Syntax 18
4.3 General information 25
4.4 Context-dependent Bitplane Coding, CBC 38
4.5 Post-Quatization Square Polar Stereo Coding, PQ-SPSC 48
4.6 Quantization 50
4.7 Multi-resolution synthesis 52
4.8 Linear prediction vector quantization decoding tool 53
4.9 Frequency-time domain mapping 56
4.10 Multi-channel extension 57
4.11 Lossless extension 59

5. Lossless audio coding 83
5.1 Overview 83
5.2 Semantics and syntax 83
5.3 Technical overview 87
5.4 Channel decorrelation 88
5.5 Integer lifting wavelet transform 90
5.6 Linear predictive coding 93
5.7 Residue pre-processing 96
5.8 Entropy coding 99

Annex A (normative) Method for keeping the syncword unique 113

Annex B (normative) AVS Audio Transport Format (AATF) and AVS Audio Storage Format (AASF) file formats 114
B.1 Bitstream syntax 114
B.2 Semantics 119

Annex C (normative) Tables for general audio coding 124

Annex D (informative) Encoder for general audio coding 202

Annex E (informative) Fixed-point reference decoder 215

Copyright © 2023 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

IEEE Standard for Advanced Audio Coding

~~*IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.*~~

~~*This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.*~~

1. Overview

1.1 Scope

This standard specifies the audio compression, decompression~~,~~ and packaging tools and mechanism to support the transmission and store ~~storage~~ of the multimedia data over the Internet in a highly efficient way. It can realize more efficient lossless compression than ALS m21 (BGMC1024) and the decoding speed is faster than EHAPE, and coding tool can be selected by taking into account compression efficiency and ~~under constraints that include limited~~ complexity ~~and bandwidth~~.

1.2 Purpose

This standard provides regular high-quality and efficient coding tool sets for compression, decompression~~,~~ and packaging of multimedia data to save bandwidth for transmission on the Internet or ~~to save~~ space for storage.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1857.2-2023

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).[1,2]

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

1.4 Audio coding tools

In this standard, a set of audio coding tools are defined for transmitting and decoding of recorded music and other audio signals. In this subclause, a short introduction for each coding tool is given. Detailed top level payload definitions, bitstream syntax, semantics, and the decoding process for each coding tool are specified in their corresponding subclauses.

1.4.1 General audio coding

Clause 4 ~~3~~ describes the coding method and decoding method of high-quality audio signals. It is applied to high-resolution digital broadcasting, high-density digital storage media, wireless broadband multimedia communication, network broadband, streaming transaction.

This clause provides a mixed framework of lossy and lossless audio coding. It includes a high-quality lossy coding sub-framework and a lossless extension coding sub-framework.

[1] The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

[2] The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

The lossy coding sub-framework supports audio coding of mono, stereo, and multichannel signals at a sampling frequency of 8 kHz ~ 96 kHz, and output bitstream of 16 kb/s ~ 96 kb/s per channel. At the same time, it also supports fine granule scalability, ; the granule step is 1 kb/s when the coding rate is below 16 kb/s ~ 79 kb/s per channel, and slightly larger than 1 kb/s when the coding rate is 80 kb/s ~ 96 kb/s per channel. In this sub-framework, time-frequency transform, frequency linear prediction and vector quantization, multiresolution analysis, post-quantization square polar stereo coding, and context-dependent bitplane coding are adopted.

The lossless extension coding sub-framework realizes lossless coding of the residual signal, and achieves a lossless framework that is compatible with the lossy coding bitstream. In this sub-framework, channel decorrelation, integer lifting wavelet, linear prediction, residuals handling, and arithmetic entropy coding are adopted.

1.4.2 Lossless audio coding

Clause 5 4 describes the lossless audio coding algorithm. It defines an efficient lossless compression technology that can be used for digital media storage, Internet wideband audio, and digital audio broadcasting areas.

Lossless audio coding supports multiple channels up to 32 channels, sampling frequency up to 16 MHz, and multi-bit resolutions: 8 bit, 16 bit, and 24 bit. It also supports random access.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

There are no normative references in this standard.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

~~1.4 Document structure~~

~~The structure of this document is as follows:~~

~~Clause 1: Overview~~

~~Clause 2: Definitions, symbols, and abbreviations~~

~~Clause 3: General audio coding~~

~~Clause 4: Lossless audio coding~~

~~Annex A (normative) Method for keeping the sync word unique~~

~~Annex B (normative) Audio Video Standard (AVS) Audio Transport Format (AATF) and AVS Audio Storage Format (AASF) file formats~~

~~Annex C (normative) Table for general audio coding~~

~~Annex D (informative) Encoder for general audio coding~~

~~Annex E (normative) Fixed-point reference decoder~~

3. Definitions, acronyms ~~symbols~~, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.[3] [4]

[3] *IEEE Standards Dictionary Online* is available at: http://dictionary.ieee.org. An IEEE account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

~~[4]IEEE Standards Dictionary Online subscription is available at:http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html~~

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

AASF: ~~Audio Video Standard (~~AVS~~)~~ Audio Storage Format.

AATF: ~~Audio Video Standard (~~AVS~~)~~ Audio Transport Format.

analysis filterbank: Filterbank in the encoder that transforms a broadband pulse-code modulation (PCM) audio signal into a set of spectral coefficients.

ancillary data: Part of the bitstream that might be used for transmission of ancillary data.

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

bitrate: The number of bits that are conveyed or processed to the decoder per unit of time.

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

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

byte: Sequence of 8 bits.

CBC band: A basic coding unit of Context-dependent Bitplane Coding (CBC) that contains 32 spectral coefficients.

CBC base layer: The base layer is the coding layer with the lowest quality in the Context-dependent Bitplane Coding (CBC). Each frame has one base layer that contains several sublayers.

CBC bitplane vector: Each bitplane vector contains 4 bits, obtained by Huffman coding of four adjacent spectral coefficients.

CBC enhancement layer: The enhancement layer is the coding layer used to enhance the quality of the base layer in the Context-dependent Bitplane Coding (CBC). Each frame has several sublayers according to ~~the~~ target coding bitrate.

CBC sublayer: Each sublayer contains one or more coding band of Context-dependent Bitplane Coding (CBC).

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

channel: A sequence of data representing an audio signal intended to be reproduced at one listening position.

channel_configuration_index: A 4-bit field that indicates the channel configuration of audio output. Each index specifies the number and mapping mode of channels in the current data block.

coded audio bitstream: A coded representation of an audio signal.

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

coding: The process to read the audio signal and generate bitstream.

Context-dependent Bitplane Coding (CBC): An entropy coding method that exploits the context information in the bit layers.

data element: Unit of data that has precise meaning or precise semantics.

decoded stream: The decoded reconstruction of a compressed bitstream.

decoder: An embodiment of a decoding process.

decoding: The process to read the bitstream and generate the audio samples.

encoder: An embodiment of a coding process.

entropy coding: Variable-length lossless coding of the digital representation of a signal that reduces statistical redundancy.

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

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

FLPVQ: Frequency-domain Linear Prediction and Vector Quantization.

frequency-time transform: Transformation of the signal from frequency domain to time domain.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

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

IFLPVQ: Inverse Frequency-domain Linear Prediction and Vector Quantization.

IMDCT: Inverse Modified Discrete Cosine Transform.

inter-channel decorrelation: Decorrelation between channels to remove the redundancy.

LPC: Linear Predictive Coding.

LSB: Least Significant Bit.

MDCT: Modified Discrete Cosine Transform.

modulation window: A data structure of spectral coefficients that is reorganized by the modulation bands after the processing of spectral coefficients. Modulation windows can be sorted as a windows sequence ordered by modulation.

MSB: Most Significant Bit.

multi-resolution filtering: A multi-resolution time-frequency domain mapping technique based on multirate sampling filter.

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

Parcor coefficients: Parcor representation of the predictor coefficients.

reserved: Some special syntactic elements that may be used in the future for extensions of this part. All the reserved values should be set to "1" unless specially ~~specifically~~ stated.

sampling frequency (fs): The number of samples per unit of time taken from a continuous audio signal to make a discrete audio signal.

scalability: An ability of a decoder that decodes ordered set ~~sets~~ of bitstream to reconstruct audio streams.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

scalefactor band: A set of spectral coefficients that are scaled by one scalefactor.

scalefactor index: A numerical code for a scalefactor.

scalefactor: Factor by which a set of values are scaled before quantization.

side information: Information in the bitstream necessary for controlling the decoding.

spectral coefficient: Discrete frequency domain data output from the analysis filterbank.

spectral envelope: The frequency-amplitude curve of the signal after a time-frequency transform.

square polar stereo code: A stereo coding method that eliminates correlation of the spectral signal of left and right channels by matrix transformation.

square polar Stereo mode: An audio coding method using the square polar stereo code algorithm.

steady signal: A type of signal whose energy and other characteristics change slowly over time. It is analyzed by long window sequences.

stuffing: Code words that may be inserted into the coded bitstream and are to be discarded in the decoding process. Their purpose is to increase the bitrate of the stream that would otherwise be lower than the desired bitrate. Beyond that, the additional coding data can use these bytes or bits as well.

syncword: Data of 12 bits embedded in an audio bitstream that indicates the start of a frame.

synthesis filterbank: Filterbank in the decoder that reconstructs a pulse-code modulation (PCM) audio signal from subband samples.

temporal stability two-level decision: A two-level decision method used to describe the temporal stability of the signal time domain and to decide whether frequency-domain linear prediction and vector quantization (FLPQV) is applied.

time-frequency transform: Transformation of the signal from time domain to frequency domain.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

transient signal: A type of siganl whose energy and other characteristics change rapidly over time. It is analyzed by short window sequences.

variable length code (VLC): A code word assigned by a variable length encoder *See:* **variable length coding**.

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

variable length decoder: A procedure to obtain the symbols encoded with a variable length coding technique.

variable length encoder: A procedure to assign variable length code (VLC) words to symbols.

3.2 Symbols and abbreviations

3.2.1 Arithmetic operators

+	Addition
–	Subtraction (as a binary operator) or negation (as a unary operator)
++	Increment
– –	Decrement
*	Multiplication
^	Power
/	Integer division with truncation of the result toward zero. For example, 6/4 and-6/-4 are truncated to 1, and -6/4 and 6/-4 are truncated to -1.
//	Integer division with rounding to the nearest integer. Half-integer values are rounded away from zero unless otherwise specified. For example, 3//2 is rounded to 2, and –3//2 is rounded to –2.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

DIV Integer division with truncation of the result towards ~~toward~~ $-\infty$.

| | Absolute value. $|x| = x$ when $x > 0$

$|x| = 0$ when $x = 0$

$|x| = -x$ when $x < 0$

% Modulus operator. Defined only for positive numbers.

sign( ) sign. $\text{sign}(x) = 1$ when $x > 0$

$\text{sign}(x) = 0$ when $x = 0$

$\text{sign}(x) = -1$ when $x < 0$

INT ( ) Truncation to integer operator. Returns the integer part of the real-valued argument.

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

sin Sine

cos Cosine

exp Exponential

$\sqrt{\ }$ Square root

lg Common (Briggsian) logarithm ($\log_{10}$)

ln Natural (Napierian) logarithm ($\log_e$)

lb Binary logarithm ($\log_2$)

ceil() Ceiling operator. Returns the smallest integer that is greater than or equal to the real-valued argument.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

3.2.2 Logical operators

||	Logical OR
&&	Logical AND
!	Logical NOT

3.2.3 Relational operators

>	Greater than
≥	Greater than or equal to
<	Less than
≤	Less than or equal to
==	Equal to
≠	Not equal to
max [,...,]	Maximum value in the argument list.
min [,...,]	Minimum value in the argument list.

3.2.4 Bitwise operators

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

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

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

3.2.5 Assignment

= Assignment operator.

3.2.6 Mnemonics

The following mnemonics are defined to describe the different data types used in the coded bitstream payload.

bslbf Bit string, left bit first, where "left" is the order in which bit strings are written in ISO/IEC 14496. Bit strings are written as a string of 1s and 0s within single quote marks, e.g., '1000 0001'. Blanks within a bit string are for ease of reading and have no significance.

L, C, R, LS, RS

Left, center, right, left surround, and right surround audio signals.

rpchof Remainder polynomial coefficients, highest order first. (Audio)

uimsbf Unsigned integer, most significant bit first.

vlclbf Variable length code, left bit first, where "left" refers to the order in which the VLC codes are written.

window Number of the actual time slot in case of block_type = 2, $0 \leq$ window ≤ 2. (Audio)

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

3.2.7 Constants

π 3.14159265358...

e 2.71828182845...

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

3.2.8 Method of describing bitstream payload syntax

The bitstream payload retrieved by the decoder is described in the syntax section ~~subclause~~ of each subpart ~~clause~~. Each data item in the bitstream payload is in bold type.

It is described by the following:-

- Its name.
- Its length in bits, where "X..Y" indicates that the number of bits is one of the values between X and Y including X and Y. "{X;Y}" means the number of bits is X or Y, depending on the value of other data elements in the bitstream payload).
- a mnemonic for its type and order of transmission.

Data elements forming a logic unit are encapsulated in data functions.

```
data_function ();          Data function call.

data_function () {         Data function entity.
    . . .
}
```

The action caused by a decoded data element in a bitstream payload depends on the value of that data element and on data elements previously decoded. The decoding of the data elements and the definition of the state variables used in their decoding are described in the subclauses following ~~in~~ the syntax subclause of each clause. The following constructs are used to express the conditions when data elements are present, and are in normal type:

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

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

while (condition) { **data_element** . . . }	If the condition is true, then the group of data elements occurs next in the data stream. This repeats until the condition is not true.
do { **data_element** . . . } while (condition)	The data element always occurs at least once. The data element is repeated until the condition is not true.
if (condition) { **data_element** . . . } else { **data_element** . . . }	If the condition is true, then the first group of data elements occurs next in the data stream If the condition is not true, then the second group of data elements occurs next in the data stream.
switch (expression) { case const-expr: **data_element**; break; case const-expr: **data_element**; }	If the condition formed by the comparison of expression and const-expr is true, then the data stream continues with the subsequent data elements. An optionally break statement can be used to immediately leave the switch, data elements beyond a break do not occur in the data stream.
for (expr1; expr2; expr3) { **data_element** . . . }	Expr1 is an expression specifying the initialization of the loop. Normally it specifies the initial state of the counter. Expr2 is a condition specifying a test made before each iteration of the loop. The loop terminates when the condition is not true. Expr3 is an expression that is performed at the end of each iteration of the loop, normally it increments a counter.

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

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

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

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

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

While the syntax is expressed in procedural terms, it should not be assumed that this implements a satisfactory decoding procedure. In particular, it defines a correct and error-free input bitstream payload. Actual decoders shall include a means to deal with incorrect bitstream payloads and to find the start of the described elements.

3.2.9 Arithmetic data types

INT32	32-bit signed integer using two's complement
INT64	64-bit signed integer using two's complement

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

4. General audio coding

4.1 Overview

Clause 4 ~~3~~ describes the coding method and decoding method of a high-quality audio signal. It is applied to high-resolution digital broadcasting, high-density digital storage media, wireless broadband multimedia communication, and network broadband streaming transactions.

This clause provides a mixed framework of lossy and lossless audio coding. It includes a high-quality lossy coding sub-framework and a lossless extension coding sub-framework.

The lossy coding sub-framework supports audio coding of mono, stereo, and multichannel signals at a sampling frequency of 8 kHz ~ 96 kHz and an output bitstream of 16 kb/s ~ 96kb/s per channel. At the same time, it also supports fine granule scalability; the granule step is 1 kb/s when the coding rate is below 16 kb/s ~ 79 kb/s per channel, and slightly larger than 1 kb/s when the coding rate is 80 kb/s ~ 96 kb/s per channel. In this sub-framework, time-frequency transform, frequency linear prediction and vector quantization, multiresolution analysis, post-quantization square polar stereo coding, and context-dependent bitplane coding are adopted.

The lossless extension coding sub-framework realizes lossless coding of the residual signal and achieves a lossless framework that is compatible with the lossy coding bitstream. In this sub-framework, channel decorrelation, integer lifting wavelet, linear prediction, residuals handling~~,~~ and arithmetic entropy coding are adopted.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

4.2 Syntax

4.2.1 Profile definition

Element	Description / Values
coding_profile	2 bits, format of the bitstream: 00 – original lossy bistream ; 01 – lossy to lossless extension; 10 – lossless only bitstream; 11 – Reserved.
sampling_frequency_index	Support 0x0 – 0xb

In this subclause, the following coding profiles are defined:

a) Lossy profile (coding_profile = 00)

b) Lossy to lossless profile (coding_profile = 01)

c) Lossless only profile (coding_profile = 10)

d) Reserved (coding_profile = 11)

Note that the sampling_frequency_index from ‘ “0xc-0xe’” are supported in lossless only profile.

4.2.2 Raw data

A raw data block can be decoded directly. In original data blocks, each frame contains a basic layer and can contain a number of enhanced layers according to objective encoding rate. The syntax of ga_raw_data_block() is shown in Table 4 3.1.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.1—Syntax of ga_raw_data_block( )

Syntex	No. of bits	Mnemonic
ga_raw_data_block()		
{		
if (2 !=coding_profile) {		
cbc_base_element();		
layer = slayer_size;		
while (data_available () && layer<(top_layer+slayer_size)) {		
cbc_layer_element(layer);		
layer++;		
}		
byte_alignment();		
if (data_available()) {		
while(data_available())		
extended_cbc_raw_data_block();		
}		
if (fill_enable) {		
if (1==coding_profile) {		
lossless_data_block();		
}		
else {		
for(i=0; i<fill_ele_number+1; i++)		
fill_element(i);		
}		
}		
}		
else {		
lossless_data_length;	**16**	**uimsbf**
lossless_data_block();		
}		
}		

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Syntax of cbc_base_element() is shown in Table 4 ~~3~~.2.

Table 4.2—Syntax of cbc_base_element( )

Syntax	No. of bits	Mnemonic
cbc_base_element()		
{		
frame_length_data();		
mc_present;	**1**	**bslbf**
if (mc_present)		
element_length_data_FLFR();		
cbc_scalable_header();		
cbc_general_header(0);		
flpvq_data();		
byte_alignment();		
for (slayer = 0; slayer < slayer_size; slayer++)		
cbc_layer_element(slayer);		
}		

Syntax of extended_cbc_raw_data_block() is shown in Table 4.3. ~~3.~~

Table 4.3—Syntax of extended_cbc_raw_data_block( )

Syntax	No. of bits	Mnemonic
extended_cbc_raw_data_block()		
{		
extended_cbc_base_element();		
layer=slayer_size;		
while(data_available() && layer<(top_layer+slayer_size)) {		
cbc_layer_element(layer);		
layer++;		
}		
byte_alignment();		
}		

Syntax of extended_cbc_base_element() is shown in Table ~~3.~~4.4.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.4—Syntax of extended_cbc_base_element()

Syntax	No. of bits	Mnemonic
extended_cbc_base_element()		
{		
element_length_data()		
channel_configuration_index	**4**	**bslbf**
scalable_header_reuse	**1**	**bslbf**
reserved_bit	**1**	**bslbf**
if (!scalable_header_reuse)		
cbc_scalable_header();		
cbc_general_header(1);		
flpvq_data();		
byte_alignment();		
for (slayer = 0; slayer < slayer_size; slayer++)		
cbc_layer_element(slayer);		
}		

Syntax of element_length_data() is shown in Table 4 ~~3~~.5.

Table 4.5—Syntax of element_length_data()

Syntax	No. of bits	Mnemonic
element_length_data()		
{		
base_element_length	**7**	**uimbf**
lengthend_flag	**1**	**bslbf**
while (lengthend_flag) {		
ehanc_element_length	**3**	**uimbf**
lengthend_flag	**1**	**bslbf**
}		
}		

Syntax of cbc_scalable_header() is shown in Table 4 ~~3~~.6.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.6—Syntax of cbc_scalable_header( )

Syntax	No. of bits	Mnemonic
cbc_scalable_header()		
{		
top_layer;	**6**	**uimbf**
base_snf_thr;	**2**	**uimbf**
base_band;	**5**	**uimbf**
}		

Syntax of cbc_general_header() is shown in Table 4 ~~3~~.7.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.7—Syntax of cbc_general_header( )

Syntax	No. of bits	Mnemonic
cbc_general_header(ext_flag)		
{		
reserved_bit;	**1**	**bslbf**
for (ch = 0;ch < nch; ch++)		
scalefactor[ch][0];	**8**	**uimbf**
signal_type;	**1**	**uimsbf**
window_shape;	**1**	**uimsbf**
if (signal_type == TRANSIENT_TYPE) {		
max_sfb;	**4**	**uimsbf**
scale_factor_grouping;	**7**	**uimsbf**
coding_band_mode;	**1**	**uimsbf**
} else {		
max_sfb;	**6**	**uimsbf**
}		
if (nch == 2)		
sc_present;	**2**	**bslbf**
if (!ext_flag) {		
fill_enable	**1**	**uimsbf**
if (fill_enable) {		
fill_ele_number	**7**	**uimsbf**
for(i=0;i<fill_ele_number+1;i++){		
fill_cnt[i] = **fill_length[i];**	**4**	**uimsbf**
if (fill_length[i]==15)		
fill_cnt[i] += **fill_esc_length[i];**	**8**	**uimsbf**
}		
}		
}		
}		

Syntax of flpvaq_data() is shown in Table 4 ~~3~~.8.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.8—Syntax of flpvq_data()

Syntax	No. of bits	Mnemonic
flpvq_data()		
{		
for (ch = 0; ch < nch; ch++) {		
bs_flp_present[ch];	**1**	**uimsbf**
if (bs_flp_present[ch]){		
bs_pred_direction[ch];	**1**	**uimsbf**
bs_vqcode1indx[ch];	**10**	**uimsbf**
bs_vqcode2indx[ch];	**10**	**uimsbf**
bs_vqcode3indx[ch];	**9**	**uimsbf**
bs_vqcode4indx[ch];	**8**	**uimsbf**
}		
}		
}		

Syntax of cbc_layer_element() is shown in Table 4 ~~3~~.9.

Table 4.9—Syntax of cbc_layer_element( )

Syntax	No. of bits	Mnemonic
cbc_layer_element(layer)		
{		
layer_cband_si(layer);		
layer_sfb_si(layer);		
cbc_layer_spectra (layer);		
cbc_lower_spectra(layer);		
}		

Syntax of layer_cband_si() is shown in Table 4 ~~3~~.10.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.10—Syntax of layer_cband_si( )

Syntax	No. of bits	Mnemonic
layer_cband_si(layer)		
{		
g = layer_group[layer];		
for (ch = 0; ch < nch; ch++) {		
for (cband = layer_start_cband[layer];		
cband < layer_end_cband [layer]; cband++) {		
hcode_cband_si[ch][g][cband];	**1..18**	**bslbf**
if (hcode_cband_si[ch][g][cband] == 0x329C8)		
cband_si_escape[ch][g][cband];	**4**	**bslbf**
}		
}		
}		

Syntax of layer_sfb_si() is shown in Table 4 ~~3~~.11.

Table 4.11—Syntax of layer_sfb_si( )

Syntax	No. of bits	Mnemonic
layer_sfb_si (layer)		
{		
g = layer_group[layer];		
for (ch = 0; ch < nch; ch++)		
for (sfb = layer_start_sfb[layer];sfb < layer_end_sfb[layer]; sfb++) {		
if (stereo_side_info_coded[g][sfb] == 0) {		
if (sc_present !=2) {		
if (sc_present == 1) {		
hcode_sc_used[g][sfb];	**1**	**bslbf**
} else if (sc_present == 3) {		
}		
}		
stereo_side_info_coded[g][sfb] = 1;		
}		
hcode_scf_index[ch][g][sfb];	**1..19**	**bslbf**
}		
}		

Syntax of cbc_layer_spectra() is shown in Table 4 ~~3~~.12.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.12—Syntax of cbc_layer_spectra( )

Syntax	No. of bits	Mnemonic
cbc_layer_spectra(layer)		
{		
g = layer_group[layer];		
start_index[g] = layer_start_ index[layer];		
end_index[g] = layer_end_ index[layer];		
if (layer < slayer_size)		
thr_snf = base_snf_thr;		
else		
thr_snf = 0;		
cbc_spectral_data (g, g+1, thr_snf, cur_snf);		
}		

Syntax of cbc_lower_spectra() is shown in Table 4 ~~3~~.13.

Table 4.13—Syntax of cbc_lower_spectra()

Syntax	No. of bits	Mnemonic
cbc_lower_spectra(layer)		
{		
for (g = 0; g < num_window_groups; g++) {		
start_index[g] = 0;		
end_index[g] = 0;		
}		
for (play = 0; play < layer; play++) {		
end_index[layer_group[play]] = layer_end_index[play];		
}		
cbc_spectral_data (0, num_window_groups, 0, unc_snf);		
}		

Syntax of cbc_spectral_data() is shown in Table 4 ~~3~~.14.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.14—Syntax of cbc_spectral_data ( )

Syntax	No. of bits	Mnemonic
cbc_spectral_data(start_g, end_g, thr_snf, cur_snf)		
{		
if (!layer_data_available()) return;		
for (snf = maxsnf; snf > thr_snf; snf--)		
for (g = start_g; g < end_g; g++)		
for (i = start_index[g]; i < end_index[g]; i++)		
for (ch = 0; ch < nch; ch++) {		
if (cur_snf[ch][g][i] < snf) continue;		
if (!sample[ch][g][i] || sign_is_coded[ch] [g][i]) {		
model_select();		
upper_vector_mapping();		
hcod_sliced_bit[ch][g][i][snf];	**1..13**	**bslbf**
}		
if (sample[ch][g][i] && !sign_is_coded[ch] [g][i]) {		
if (layer_data_available()) return;		
hcod_sign[ch][g][i];	**1**	**bslbf**
sign_is_coded[ch][g][i] = 1;		
}		
cur_snf[ch][g][i]--;		
if (!layer_data_available()) return;		
}		
}		

Syntax of fill_element() is shown in Table 4 ~~3~~.15.

Table 4.15—Syntax of fill_element( )

Syntax	No. of bits	Mnemonic
fill_element(fill_idx)		
{		
fill_type	**4**	**uimsbf**
switch(fill_type) {		
case FILL_DFT:		
for(i=0;i<8*(fill_cnt[fill_idx]-1)+4;i++)		
fill_dft_bits[i]	**1**	**uimsbf**
}		
}		

Syntax of frame_length_data() is shown in Table 4 ~~3~~.16.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.16—Syntax of frame_length_data( )

Syntax	No. of bits	Mnemonic
frame_length_data()		
{		
base_frame_length;	**7**	**uimbf**
length_end_flag;	**1**	**bslbf**
enhance_length_cnt = 0;		
while (length_end_flag) {		
enhance_frame_length[enhance_length_cnt++];	**3**	**uimbf**
length_end_flag;	**1**	**bslbf**
}		
}		

Syntax of element_length_data_FLFR() is shown in Table 4.~~3.~~17.

Table 4.17—Syntax of element_length_data_FLFR()

Syntax	No. of bits	Mnemonic
element_length_data_FLFR()		
{		
base_element_length_FLFR	**7**	**uimbf**
lengthend_flag	**1**	**bslbf**
while (lengthend_flag) {		
ehanc_element_length_FLFR	**3**	**uimbf**
lengthend_flag	**1**	**bslbf**
}		
}		

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

4.3 General information

4.3.1 Introduction

The AVS audio coder and decoder make use of a number of required and optional modules. Table 4 ~~3~~.18 lists the modules and their status such as *required* or *optional*. Required modules are mandatory in any possible profile. Optional modules may not be required in some profiles.

Table 4.18—Modules

Module Name	Required / Optional
Bitstream Deformatter	Required
Context-dependent Bitplane Decoding	Required
Inverse Quantization	Required
Post-Quantization Domain Square Polar Stereo Decoding	Optional
Multi-resolution Synthesis	Optional
Inverse Frequency-domain Linear Prediction and Vector Quantization	Optional
Inverse Time-Frequency Transform	Required

The basic structure of the AVS audio system is shown in Figure 4. ~~3~~1 and Figure 4. ~~3~~2. As is shown in Table 4 ~~3~~.18, there are both required and optional modules in the decoder~~;~~, refer to Figure 4. ~~3~~2. The data flow in this diagram is from left to right, top to bottom. The functions of the decoder are to find the description of the quantized audio spectra in the bitstream, decode the quantized values and other reconstruction information, reconstruct the quantized spectra, process the reconstructed spectra through whatever modules are active in the bitstream in order to arrive at the actual signal spectra as described by the input bitstream, and finally convert the reconstructed spectral coefficients from frequency domain to the time domain.

The input to the Bitstream Deformatter Module is the AVS audio data stream. The module separates each part of the AVS audio data stream into the parts for each module and provides each of the modules with the decoding information related to that module.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

The input to the Bitstream Deformatter is: AVS audio coded stream.

The outputs from the Bitstream Deformatter Module are as follows:

- The noiselessly coded spectra
- Context-dependent bitplane decoding lossless decoding control information
- Quantization domain square polar stereo decoding control information
- Signal type information
- Frequency domain linear predictive and vector quantization control information

The Context-dependent Bitplane Decoding Module takes information from the bitstream deformatter, decodes the Huffman coded data, and reconstructs the quantized spectra, mode information of coding band and scalefactor.

The inputs to the Context-dependent Bitplane Decoding Module are as follows:

- The noiselessly coded mode information of coding bands
- The noiselessly coded scalefactors
- The noiselessly coded spectra

The output(s) from the Context-dependent Bitplane Decoding Module is (are) as follows:

- The mode of coding band
- The quantized values for the spectra

The Post-Quantization Square Polar Stereo Decoding Module implements stereo decoding on the square polar of spectra.

The inputs to the Post-Quantization Square Polar Stereo Decoding Module are as follows:

- The quantized values for the spectra

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

- The square polar stereo and the coupling control information
- The scalefactors after context-dependent bitplane decoding

The output(s) from the Post-Quantization Square Polar Stereo Decoding Module is (are) as follows:

- The quantized spectra after post-quantization square polar stereo decoding
- The scalefactors after post-quantization square polar stereo decoding

The Inverse Quantization Module takes the quantized values for the spectra, and converts the integer values to the scaled, reconstructed spectra. This quantizer is a non-uniform quantizer.

The inputs to the Inverse Quantization Module are as follows:

- The quantized values for the spectra
- The decoded integer representation of the scalefactors

The output(s) of the Inverse Quantization tool is (are) as follows:

- The scaled, inversely quantized spectra

The Multi-Resolution Synthesis Module implements multi-resolution processing on audio spectral coefficient of transient signal.

The inputs to the Multi-Resolution Synthesis Module are as follows:

- The inversely quantized spectra / the residual spectra

The output(s) from the Multi-Resolution Synthesis Module is (are) as follows:

- The reconstructed spectra / the residual spectra

Frequency-domain Linear Prediction and Vector Quantization (FLPVQ) module used Frequency-domain Linear Prediction technique and Vector Quantization technique to improve the coding efficiency effectively.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

The inputs to the Inverse Frequency-domain Linear Prediction and Vector Quantization (IFLPVQ) Module are as follows:

- The residual spectra-
- The vector indices
- The control information

The output(s) from the IFLPVQ Module is (are) as follows:

- Modified Discrete Cosine Transform (MDCT) spectral coefficients

The Inverse Time-Frequency Transform Module applies the inverse of the time-frequency mapping that was carried out in the encoder. An inverse modified discrete cosine transform (IMDCT) is used for this module, ; the spectral structure of the MDCT coefficients with an input of 1024 samples depends on the window sequence that is in use.

The inputs to the Inverse Time-Frequency Transform Module are as follows:

- The inversely quantized spectra coefficients

The output(s) from the Inverse Time-Frequency Transform Module is (are) as follows:

- The time domain reconstructed audio data

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

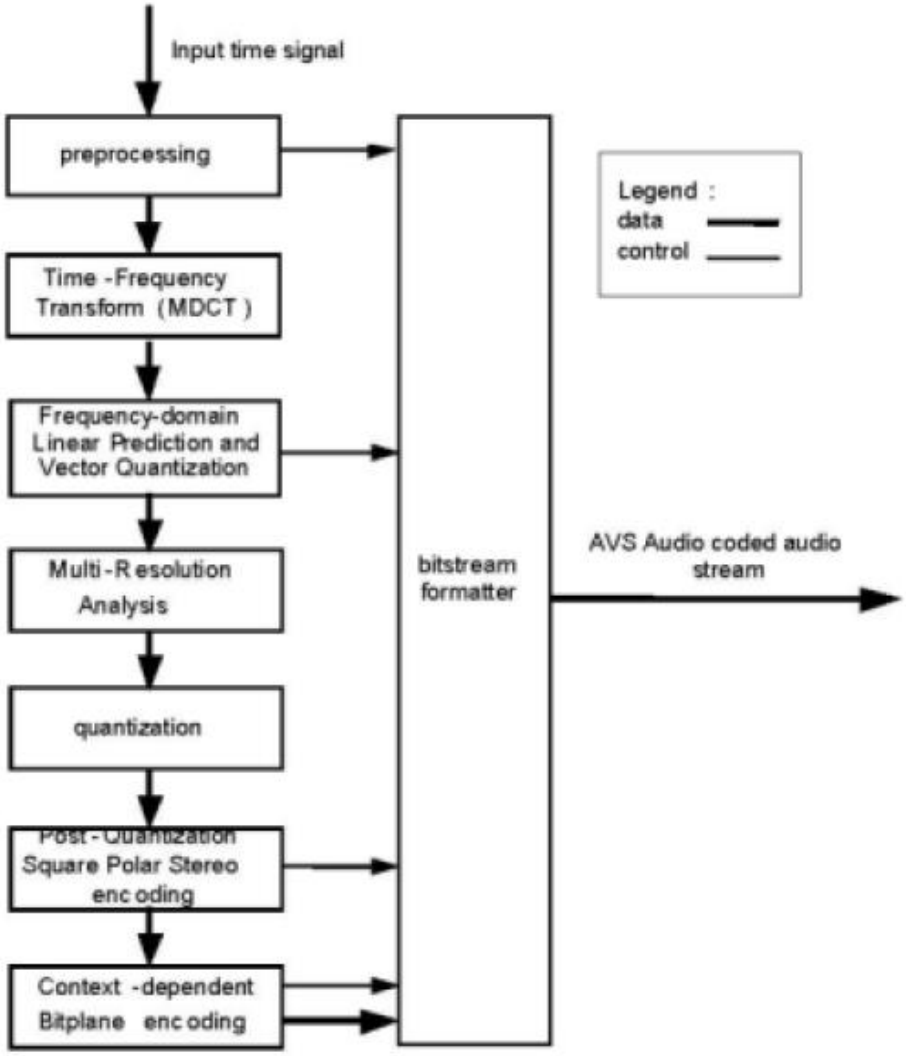


Figure 4.1—AVS audio encoder block diagram

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

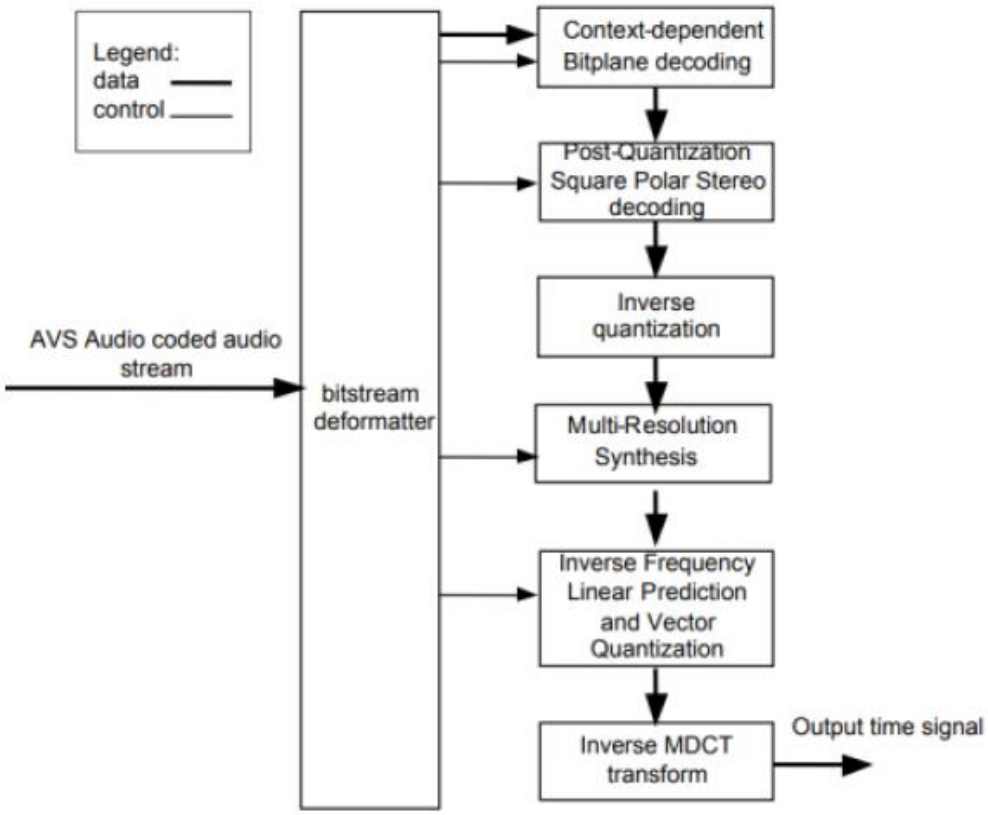


Figure 4.2—AVS audio decoder block diagram

4.3.2 Decoding of raw data

4.3.2.1 Definition ~~Definitions~~

ga_raw_data_block () Block of raw data that contains 1024 samples of audio data, related information~~,~~ and other data.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

4.3.2.2 Decoding process

Assuming that the start of a raw_data_block is known, it can be decoded without any additional transport information to produce 1024 audio samples per output channel. The sampling rate of the audio signal may be specified by an index of sampling frequency (sampling_frequency_index).

4.3.2.3 Grouping and interleaving

The spectral coefficients are constructed to form the modulation window sequence after multi-resolution analysis in frequency domain. Each modulation window sequence is divided into different scalefactor bands in the coding process; thus optimal grouping and interleaving can be used to achieve greater coding efficiency.

The coefficients can be grouped such that they share scalefactors amongst ~~among~~ all scalefactor bands within the group. In addition, the coefficients within a group are interleaved by interchanging the order of scalefactor bands and windows, thus providing for greater coding efficiency. To be specific, assume that before interleaving the set of 1024 coefficients c are indexed as:

$$c[g][w][k/4][k\%4]$$

where

g is the index on groups

w is the index on windows within a group

k is the index on coefficients within a window

After interleaving, the coefficients are indexed as $c[g]\ [k/4]\ [w]\ [k\%4]$~~.~~

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

4.3.2.4 Table of scalefactor bands

Table 4.19—Scalefactor bands of STATIONARY_TYPE at 44.1 ~~kHz~~ and 48 kHz

fs [kHz]	44.1, 48
num_swb_station_type	49
swb	swb_offset_station_type
0	0
1	4
2	8
3	12
4	16
5	20
6	24
7	28
8	32
9	36
10	40
11	48
12	56
13	64
14	72
15	80
16	88
17	96
18	108
19	120
20	132
21	144
22	160
23	176
24	196

swb	swb_offset_station_type
25	216
26	240
27	264
28	292
29	320
30	352
31	384
32	416
33	448
34	480
35	512
36	544
37	576
38	608
39	640
40	672
41	704
42	736
43	768
44	800
45	832
46	864
47	896
48	928
	1024

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.20—Scalefactor bands of TRANSIENT_TYPE at 32 ~~kHz~~, 44.1 ~~kHz,~~ and 48 kHz

fs [kHz]	32, 44.1, 48
num_swb_transient_type	14
swb	swb_offset_transient_type
0	0
1	4
2	8
3	12
4	16
5	20
6	28
7	36

swb	swb_offset_transient_type
8	44
9	56
10	68
11	80
12	96
13	112
	128

Table 4.21—Scalefactor bands of STATIONARY_TYPE at 32 kHz

fs [kHz]	32
num_swb_station_type	51
swb	swb_offset_station_type
0	0
1	4
2	8
3	12
4	16
5	20
6	24
7	28
8	32
9	36
10	40
11	48
12	56
13	64
14	72
15	80
16	88
17	96
18	108
19	120
20	132
21	144
22	160
23	176
24	196
25	216

swb	swb_offset_station_type
26	240
27	264
28	292
29	320
30	352
31	384
32	416
33	448
34	480
35	512
36	544
37	576
38	608
39	640
40	672
41	704
42	736
43	768
44	800
45	832
46	864
47	896
48	928
49	960
50	992
	1024

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.22—Scalefactor bands of STATIONARY_TYPE at 8 kHz

fs [kHz]	8
num_swb_station_type	40
swb	swb_offset_station_type
0	0
1	12
2	24
3	36
4	48
5	60
6	72
7	84
8	96
9	108
10	120
11	132
12	144
13	156
14	172
15	188
16	204
17	220
18	236
19	252
20	268

swb	swb_offset_station_type
21	288
22	308
23	328
24	348
25	372
26	396
27	420
28	448
29	476
30	508
31	544
32	580
33	620
34	664
35	712
36	764
37	820
38	880
39	944
	1024

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023

Table 4.23—Scalefactor bands of TRANSIENT_TYPE at 8 kHz

fs [kHz]	8
num_swb_transient_type	15
swb	swb_offset_transient_type
0	0
1	4
2	8
3	12
4	16
5	20
6	24
7	28

swb	swb_offset_transient_type
8	36
9	44
10	52
11	60
12	72
13	88
14	108
	128

Table 4.24—Scalefactor bands of STATIONARY_TYPE at 11.025 ~~kHz~~, 12& ~~kHz, and~~ 16 kHz

fs [kHz]	11.025, 12, 16
num_swb_station_type	43
swb	swb_offset_station_type
0	0
1	8
2	16
3	24
4	32
5	40
6	48
7	56
8	64
9	72
10	80
11	88
12	100
13	112
14	124
15	136
16	148
17	160
18	172
19	184
20	196
21	212

swb	swb_offset_station_type
22	228
23	244
24	260
25	280
26	300
27	320
28	344
29	368
30	396
31	424
32	456
33	492
34	532
35	572
36	616
37	664
38	716
39	772
40	832
41	896
42	960
	1024

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 01:43:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1857.2-2023