INTERNATIONAL STANDARD

ISO/IEC 23002-4

Third edition
2018-03

Information technology — MPEG video technologies —

Part 4: Video tool library

Technologies de l'information — Technologies vidéo MPEG —

Partie 4: Bibliothèque d'outils vidéo

Reference number
ISO/IEC 23002-4:2018(E)

Documents
Document Sharing Hub
© ISO/IEC 2018

Full document details: ISO IEC 23002-4-2018

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2018

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 23002-4-2018

Contents

Page

Foreword ... viii
Introduction ... x

1 Scope ... 1

2 Normative references ... 1

3 Terms and definitions ... 2

4 FU description convention ... 2
4.1 FU interface ... 2
4.2 FU IDs ... 3
4.3 Token pool ... 7
4.4 Array data order ... 10
4.5 Input ports (reset_i, init_i, start_i) ... 10
4.6 FU block diagram notations ... 10
4.7 Conventions ... 11
4.8 Granular FU ... 11

5 General-purpose FUs ... 11
5.1 Syntax parsing ... 11
5.1.1 Generic syntax parser ... 11
5.1.2 Algo_Byte2bit ... 11
5.1.3 Mgnt_Select_MB_4 ... 12
5.1.4 Mgnt_Merger420 ... 12
5.1.5 Mgnt_Select_MB_8 ... 12
5.1.6 Mgnt_SynP_DEMUX ... 13
5.1.7 Mgnt_SynP_BytePreprocessor ... 13
5.2 General processing FUs ... 14
5.2.1 Algo_InverseQuantization1D ... 14
5.2.2 Algo_InverseQuantizationND ... 15
5.2.3 Algo_InversePrediction1D ... 18
5.2.4 Algo_InversePredictionND ... 20
5.2.5 Algo_ED_AD_StaticBit ... 24
5.2.6 Algo_ED_VLD ... 26
5.2.7 Alog_ED_AD_AdaptiveBit ... 27
5.2.8 Algo_ED_BitPrecision ... 28
5.2.9 Algo_ED_AD ... 31
5.2.10 Algo_ED_AD_EG ... 36
5.2.11 Algo_ED_4bitsD ... 42
5.2.12 Algo_ED_FixedLength ... 44
5.2.13 Algo_LookUpTable1D ... 45
5.2.14 Algo_ContextModeling ... 46
5.2.15 Algo_simpleMath_2op ... 48
5.2.16 Mgnt_Replicate_1_2 ... 50
5.2.17 Mgnt_Replicate_1_4 ... 50
5.2.18 Mgnt_Replicate_1_8 ... 51
5.2.19 Mgnt_MUX_2_1 ... 52
5.2.20 Mgnt_MUX_4_1 ... 53
5.2.21 Mgnt_MUX_8_1 ... 55
5.2.22 Mgnt_DEMUX_1_2 ... 56
5.2.23 Mgnt_DEMUX_1_4 ... 57
5.2.24 Mgnt_DEMUX_1_8 ... 59
5.2.25 Mgnt_ExtractSegment ... 60
5.2.26 Mgnt_ProviderValue ... 61
5.2.27 Mgnt_RepeatSegment ... 62
5.2.28 Mgnt_ExtractBytes ... 64
5.2.29 Mgnt_ExtractBits ... 65

Documents
Document Sharing Hub

© ISO/IEC 2018 — All rights reserved

Full document details: ISO IEC 23002-4-2018

5.2.30 Mgnt_Provider1D ... 66
5.2.31 Mgnt_Provider2D ... 67

6 FUs for MPEG-4 Simple Profile ... 69
6.1 General ... 69
6.2 Syntax parsing ... 69
6.2.1 Algo_SynP ... 69
6.2.2 Mgnt_BlockExpand ... 69
6.2.3 Mgnt_Splitter420B ... 70
6.2.4 Mgnt_Splitter420MV ... 70
6.2.5 Algo_MVR_MedianOfThreeLeftAndTopAndTopRight ... 71
6.2.6 Mgnt_Splitter_420_TYPE ... 71
6.2.7 Algo_VLDtableB6_MPEG4Part2 ... 72
6.2.8 Algo_VLDtableB7_MPEG4Part2 ... 72
6.2.9 Algo_VLDtableB8_MPEG4Part2 ... 73
6.2.10 Algo_VLDtableB12_MPEG4Part2 ... 73
6.2.11 Algo_VLDtableB13_MPEG4Part2 ... 73
6.2.12 Algo_VLDtableB14_MPEG4Part2 ... 74
6.2.13 Algo_VLDtableB15_MPEG4Part2 ... 74
6.2.14 Algo_VLDtableB16_MPEG4Part2 ... 75
6.2.15 Algo_VLDtableB17_MPEG4Part2 ... 75
6.3 Texture decoding ... 76
6.3.1 Algo_IQ_QSAndQmatrixMp4vOrH263Scaler ... 76
6.3.2 Algo_DCRAddr_ThreeLeftTop_8x8 ... 76
6.3.3 Algo_DCRAddr_ThreeLeftTop_16x16 ... 77
6.3.4 Algo_DCRInvPred_CHROMA_8x8 ... 77
6.3.5 Algo_DCRInvPred_LUMA_16x16 ... 78
6.3.6 Algo_IS_ZigzagOrAlternateHorizontalVertical_8x8 ... 79
6.3.7 Algo_IAP_AdaptiveHorizontalOrVerticalPred_8x8 ... 79
6.3.8 Algo_IAP_AdaptiveHorizontalOrVerticalPred_16x16 ... 80
6.3.9 Algo_IDCT2D_ISOIEC_23002_1 ... 80
6.3.10 Mgnt_DCSplit ... 81
6.4 Motion compensation ... 81
6.4.1 Mgnt_FB_w_Addess_8x8 ... 81
6.4.2 Mgnt_FB_w_Addess_16x16 ... 82
6.4.3 Algo_PictureReconstruction_Saturation ... 82
6.4.4 Algo_Interp_HalfpelBilinearRoundingControl ... 83

7 FUs for MPEG-4 AVC Constrained Baseline Profile ... 84
7.1 General ... 84
7.2 Syntax parsing ... 84
7.2.1 Algo_NALU ... 84
7.2.2 Algo_SynP ... 84
7.2.3 Algo_BlockExpand ... 85
7.2.4 Algo_BlockSplit ... 85
7.2.5 Algo_IntraPred_Split ... 86
7.2.6 Algo_Parser_I_PCM ... 86
7.2.7 Algo_DemuxParserInfoForBlocks_Chroma ... 87
7.2.8 Algo_DemuxParserInfoForBlocks_Luma ... 87
7.3 Texture decoding ... 88
7.3.1 Algo_IS_Zigzag_4x4 ... 88
7.3.2 Algo_DCR_Hadamard_LUMA_IHT1d ... 88
7.3.3 Algo_Transpose4x4 ... 89
7.3.4 Algo_DCR_Hadamard_LUMA_Reordering ... 89
7.3.5 Algo_DCR_Hadamard_LUMA_Scaling ... 90
7.3.6 Algo_DCR_Hadamard_CHROMA ... 90
7.3.7 Algo_IT4x4_1d ... 90
7.3.8 Algo_IT4x4_Addshift ... 91
7.3.9 Algo_IntraPred_LUMA_16x16 ... 91
7.3.10 Algo_IntraPred_LUMA_4x4 ... 92

Documents
Document Sharing Hub

© ISO/IEC 2018 — All rights reserved

Full document details: ISO IEC 23002-4-2018

7.3.11 Algo_Merge_4x4_to_16x16 ... 92
7.3.12 Algo_IQ_QSAndSLAndIDCTScaler_4x4 ... 92
7.3.13 Mgnt_IQ_INTRA16x16 ... 93
7.3.14 Algo_IntraPred_4x4_to_8x8 ... 93
7.3.15 Algo_IntraPred_CHROMA ... 94
7.3.16 Mgnt_Intra16x16 ... 94
7.3.17 Mgnt_Intra4x4 ... 95
7.3.18 Mgnt_IQ_Chroma ... 95
7.3.19 Mgnt_Buffer_Neighbour_FullMb ... 95
7.3.20 Mgnt_Buffer_Neighbour_YxY ... 96
7.3.21 Algo_Merge_4x4_to_16x16_norasterscan ... 96
7.3.22 Algo_Split_16x16_to_4x4_norasterscan ... 97
7.4 Motion compensation ... 97
7.4.1 Algo_Interp_EighthPelBilinear ... 97
7.4.2 Algo_Interp_SeparableSixTapQuarterPel ... 98
7.4.3 Algo_Interp_Reord ... 98
7.4.4 Algo_MvLXReconstr ... 99
7.4.5 Mgnt_DPB ... 99
7.4.6 Algo_MMCO ... 100
7.4.7 AlgoRefList ... 100
7.4.8 Mgnt_InterPred ... 101
7.4.9 Algo_RefIdxtoFrameNum ... 101
7.5 Filtering ... 102
7.5.1 Mgnt_DBF_AdaptiveFilter ... 102
7.5.2 Algo_DBF_AdaptiveFilter ... 102
7.5.3 Algo_MvComponentReorder ... 103
7.6 Renderer ... 103
7.6.1 Mgnt_POC ... 103
7.6.2 Mgnt_BufferRender ... 103
7.6.3 Mgnt_Merger420_AVC ... 104

8 FUs for MPEG-4 AVC Progressive High Profile ... 104
8.1 General ... 104
8.1.1 Overview ... 104
8.1.2 Algo_SynP ... 104
8.1.3 Algo_BlockExpand ... 106
8.1.4 Algo_DemuxParserInfoForBlocks_Luma ... 106
8.2 Texture decoding ... 107
8.2.1 Algo_IS_Zigzag_8x8 ... 107
8.2.2 Algo_IQ_QSAndSLAndIDCTScaler_8x8 ... 107
8.2.3 Algo_IIT_8x8 ... 107
8.2.4 Algo_IntraPred_LUMA_8x8 ... 108
8.2.5 Mgnt_Intra_8x8 ... 108
8.2.6 Algo_Merge_8x8_to_16x16 ... 109
8.2.7 Algo_DCR_Hadamard_CHROMA ... 109
8.2.8 Algo_DCR_Hadamard_LUMA_Scaling ... 110
8.2.9 Algo_IQ_QSAndSLAndIDCTScaler_4x4 ... 110
8.2.10 Algo_Merge_8x8_to_16x16_norasterscan ... 111
8.2.11 Algo_Split_16x16_to_8x8_norasterscan ... 111
8.2.12 Mgnt_I4x4_I8x8_demux ... 111
8.2.13 Mgnt_I4x4_I8x8_mux ... 112
8.3 Motion compensation ... 112
8.3.1 Algo_GeneratePredWeight ... 112
8.3.2 Mgnt_SelectMvpLX ... 113
8.3.3 Algo_MvLXReconstr ... 113
8.3.4 Algo_MvBuffer ... 114
8.3.5 Mgnt_SelectMvpLX ... 115
8.3.6 Algo_FrameNumToPocList ... 115
8.4 Filtering ... 115

Documents
Document Sharing Hub

© ISO/IEC 2018 — All rights reserved

Full document details: ISO IEC 23002-4-2018

8.4.1 Algo_DBF_AdaptiveFilter ... 115
8.4.2 Algo_MvComponentReorder ... 116

9 MPEG-4 Part 16 SC3DMC decoder specific FUs ... 117
9.1 General ... 117
9.2 Algo_ExtractMask_SC3DMC ... 117
9.3 MPEG-4 SC3DMC TFAN Specific FUs ... 118
9.3.1 Algo_DecodeConnectivity_TFAN ... 118
9.4 MPEG-4 SC3DMC SVA Specific FUs ... 123
9.4.1 Algo_ContextModeling_SVA_nType ... 123
9.4.2 Algo_ContextModeling_SVA_Indexes ... 125
9.4.3 Algo_ContextModeling_SVA_Vertex_Attribute ... 127
9.4.4 Algo_DecodeConnectivity_SVA ... 130
9.4.5 Algo_ExtractFaceDirection_SVA ... 134
9.4.6 Algo_Connectivity_InversePrediction_SVA ... 135

10 FUs for HEVC Main Profile ... 137
10.1 General ... 137
10.2 Syntax parsing ... 137
10.2.1 Algo_SynP ... 137
10.3 Texture decoding ... 138
10.3.1 Algo_IntraPrediction ... 138
10.3.2 selectCU ... 138
10.4 Motion compensation ... 139
10.4.1 Algo_InterPrediction ... 139
10.4.2 Mgnt_DecodedPictureBuffer ... 139
10.4.3 Algo_GenerateRefList ... 140
10.4.4 Algo_MvComponentPred ... 140
10.5 Filtering ... 141
10.5.1 Algo_SaoFilter ... 141
10.5.2 Algo_GenerateBs ... 141
10.5.3 Algo_DeblockingFilter ... 142
10.5.4 Algo_QpGen ... 142
10.6 MD5 check ... 142
10.6.1 Mgnt_MD5SplitInfo ... 142
10.6.2 Mgnt_padding ... 143
10.6.3 Mgnt_MD5Shifter ... 143
10.6.4 Mgnt_MD5Compute ... 143
10.7 Inverse transforms ... 144
10.7.1 Mgnt_IT_Splitter ... 144
10.7.2 Mgnt_IT_Merger ... 144
10.7.3 Mgnt_Block_Merger ... 145
10.7.4 Mgnt_Transpose_4x4 ... 145
10.7.5 Mgnt_Transpose_8x8 ... 145
10.7.6 Mgnt_Transpose_16x16 ... 146
10.7.7 Mgnt_Transpose_32x32 ... 146
10.7.8 Mgnt_Transpose_32x32 ... 146
10.7.9 Algo_IT4x4_1d ... 146
10.7.10 Algo_IT8x8_1d ... 147
10.7.11 Algo_IT16x16_1d ... 147
10.7.12 Algo_IT32x32_1d ... 147
10.7.13 Algo_invDST4x4_1st ... 148
10.7.14 Algo_invDST4x4_2nd ... 148

Annex A (normative) Naming convention of FU ... 149

Annex B (informative) FU network examples ... 151

Annex C (informative) FNL of MPEG-4 AVC Progressive High Profile decoder ... 186

Annex D (normative) Granular FUs concept ... 218

Documents
Document Sharing Hub

© ISO/IEC 2018 — All rights reserved

Full document details: ISO IEC 23002-4-2018

Annex E (informative) Granular FUs 219
Annex F (normative) Parallelogram prediction 236
Annex G (informative) Network specifications 240
Annex H (informative) Patent statement 241
Annex I (informative) Instantiation of generic syntax parser FU from BSD 242
Bibliography 261

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This third edition cancels and replaces the second edition (ISO/IEC 23002-4:2014), which has been technically revised.

It also incorporates ISO/IEC 23002-4:2014/Amd.1:2014, ISO/IEC 23002-4:2014/Amd.2:2015, and ISO/IEC 23002-4:2014/DAmd.3[1].

The changes compared to the previous edition are as follows.

— FU description convention described in Clause 4 has been technically updated.

— Functional unit (FU) and FU network (FN) descriptions for graphics tool library (GTL) have been added according to ISO/IEC 23002-4:2014/Amd.1:2014. Clause 5, Clause 9, Annexes D, E, F, and G have been added.

— FU and FN descriptions for high efficiency video coding (HEVC) have been added according to ISO/IEC 23002-4:2014/Amd.2:2015. Clause 10 has been added and Annex B has been updated.

[1] Draft amendment approved in DAM ballot but not published.

© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

— FU and FN descriptions for parser instantiation from bitstream syntax description (BSD) have added. Clause 5 has updated and Annex I has added.

— A patent statement has been added in Annex H.

A list of all parts in the ISO/IEC 23002 series can be found on the ISO website.

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

Introduction

This document defines the MPEG video tool library, which contains tools drawn from existing MPEG coding standards, such as ISO/IEC 14496-2 and ISO/IEC 14496-10, and ISO/IEC 23001-4 defines the methods capable of describing codec configurations in the reconfigurable video coding (RVC) framework.

This document primarily addresses reconfigurable video aspects and will only focus on the description of representation of video codec configurations under the RVC framework, but could be extended to a more generic reconfigurable media coding (RMC) framework.

The objective of RVC is to offer a framework that is capable of configuring and specifying video codecs as a collection of "higher level" modules by using video coding tools. The video coding tools are defined in video tool libraries. This d defines the MPEG video tool library. The RVC framework principle could also support non-MPEG tool libraries, provided that their developers have taken care to obey the appropriate rules of operation.

For the purpose of framework deployment, an appropriate description is needed to describe configurations of decoders composed of or instantiated from a subset of video tools from either one or more libraries. As illustrated in Figure 1, the configuration information consists of

— bitstream syntax description, and

— network of functional units (FUs) description (also referred to as the decoder configuration)

that together constitute the entire decoder description.

Bitstreams of existing MPEG standards are specified by specific syntax structures and decoders are composed of various coding tools. Therefore, RVC includes support for bitstream syntax descriptions as well as video coding tools. As depicted in Figure 1, a typical RVC decoder requires two types of information, namely the decoder description and the encoded media (e.g. video bitstreams) data.

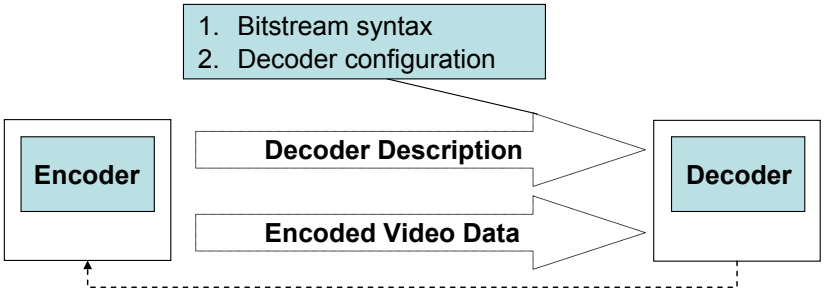


Figure 1 — Conceptual diagram of RVC

A more detailed description of the RVC decoder is illustrated in Figure 2. As shown in Figure 2, the decoder description is required for the configuration of a RVC decoder. The Bitstream Syntax Description (BSD) and FU Network Description (FND) (which compose the Decoder Description) are used to configure or compose an abstract decoder model (ADM) which is instantiated through the selection of FUs from tool libraries optionally with proper parameter assignment. Such ADM constitutes the behavioral reference model used in setting up a decoding solution under the RVC framework. The process of yielding a decoding solution may vary depending on the technologies used for the desired

© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

implementations. Examples of the instantiation of an ADM and generation of proprietary decoding solutions can be found in ISO/IEC 23001-4.

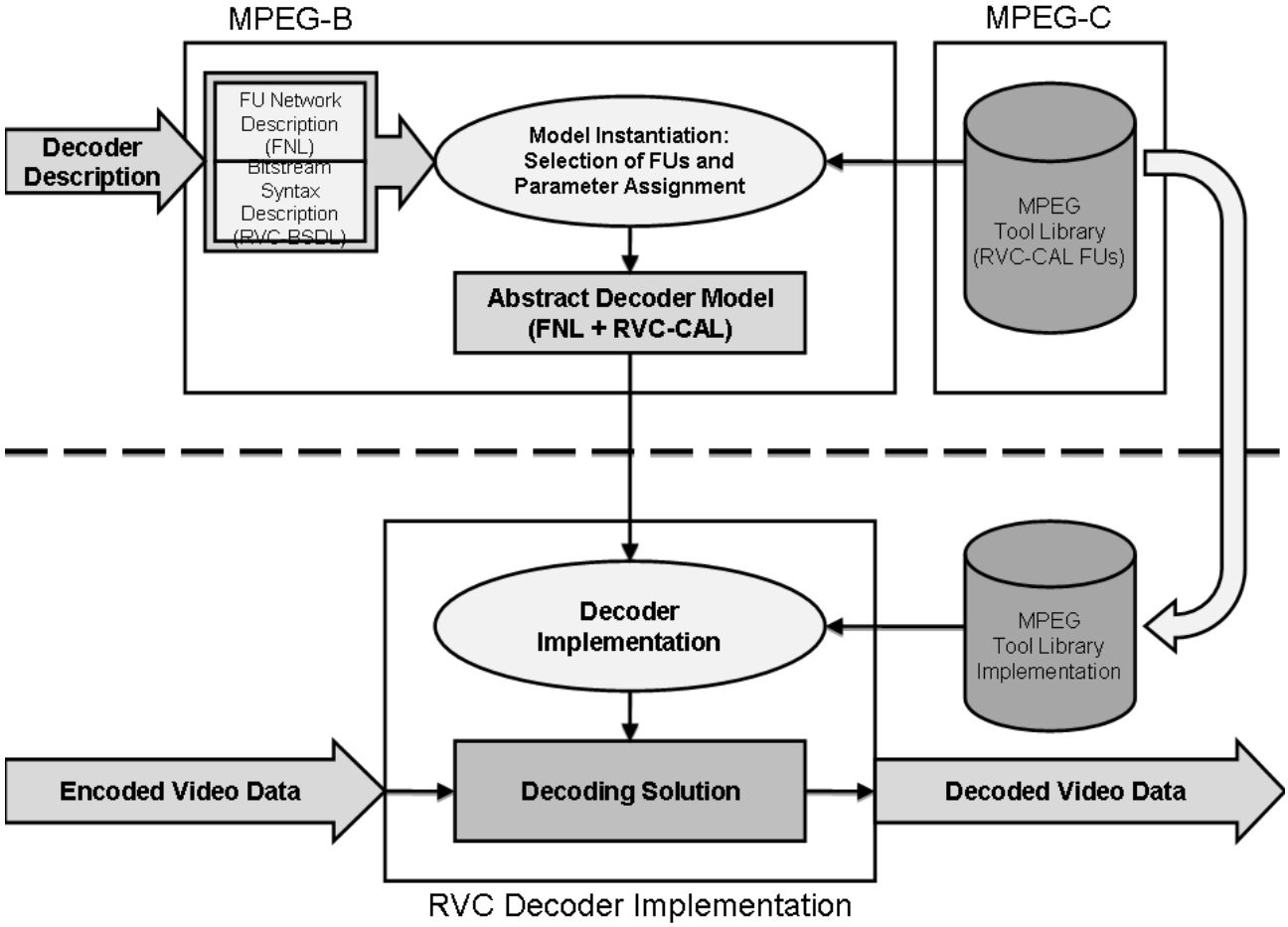


Figure 2 — Graphical representation of the process for setting up a decoding solution under the RVC framework

Within the RVC framework, the decoder description describes a particular decoder configuration and consists of the FND and the BSD. The FND describes the connectivity of the network of FUs used to form a decoder whereas the parsing process for the bitstream syntax is implicitly described by the BSD. These two descriptions are specified using two standard XML-based languages or dialects.

— Functional unit network language (FNL) is a language that describes the FND, known also as "network of FUs". The FNL specified normatively within the scope of the RVC framework is provided in ISO/IEC 23001-4.

— Bitstream syntax description language (BSDL), standardized in ISO/IEC 23001-5 (MPEG-B Part 5), describes the bitstream syntax and the parsing rules. A pertinent subset of this BSDL named RVC-BSDL is defined within the scope of the current RVC framework. This RVC-BSDL also includes possibilities for further extensions, which are necessary to provide complete description of video bitstreams. RVC-BSDL specified normatively within the scope of the RVC framework is provided in ISO/IEC 23001-4.

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

The decoder configuration specified using FNL, together with the specification of the bitstream syntax using RVC-BSDL fully specifies the ADM and provides an “executable” model of the RVC decoder description.

The instantiated ADM includes the information about the selected FUs and how they should be connected. As already mentioned, the FND with the network connection information is expressed by using FNL. Furthermore, the RVC framework specifies and uses a dataflow-oriented language called RVC-CAL for describing FUs' behavior. The normative specification of RVC-CAL is provided in ISO/IEC 23001-4. The ADM is the behavioural model that should be referred to in order to implement any RVC conformant decoder. Any RVC compliant decoding solution/implementation can be achieved by using proprietary non-normative tools and mechanisms that yield decoders that behave equivalent to the RVC ADM.

The decoder description, the MPEG tool library, and the associated instantiation of an ADM are normative. More precisely, the ADM is intended to be normative in terms of a behavioural model. In other words, what is normative is the input/output behaviour of the complete ADM, as well as the input/output behaviour of all the FUs that are included in the ADM.

A statement concerning patents is included in Annex H of this document.

© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

Information technology — MPEG video technologies —

Part 4:
Video tool library

1 Scope

This document defines the description of the MPEG video tool library (VTL) based on the decoder description specified in ISO/IEC 23001-4. This tool library defines the specification of FUs, which are sufficient to build complete decoding solutions according to the following coding standards:

— ISO/IEC 14496-2 (MPEG-4 Simple Profile),

— ISO/IEC 14496-10 (MPEG-4 AVC Constrained Baseline Profile and Progressive High Profile),

— ISO/IEC 14496-16 (MPEG-4 SC3DMC), and

— ISO/IEC 23008-2 (HEVC Main Profile).

The objective of ISO/IEC 23001-4 is to define the general framework principles, and this document defines the MPEG VTL that includes relevant tools (or FUs) from the existing MPEG coding standards. Each FU is defined in the form of a textual description, which can be found in 4.1. The input and output behaviour follows the conventions described in Clause 5, Clause 6, and Clause 7.

This document compliant implementations can be designed using any software or hardware language and components. The reference software for the textual specification of FUs is written in RVC-CAL language of which a formal syntax is provided in ISO/IEC 23001-4.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14496-2, *Information technology — Coding of audio-visual objects — Part 2: Visual*

ISO/IEC 14496-10, *Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding*

ISO/IEC 14496-16, *Information technology — Coding of audio-visual objects — Part 16: Animation Framework eXtension (AFX)*

ISO/IEC 23001-4, *Information technology — MPEG systems technologies — Part 4: Codec configuration representation*

ISO/IEC 23008-2, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding*

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 23001-4 apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

4 FU description convention

4.1 FU interfaces

As shown in Table 1, each FU is described with the following elements.

— **FU Name**: Name to represent the functional unit in this specification. The name of the FU is normative and follows the naming convention described in Annex A.

— **Description**: Textual explanation to describe the functionality of the FU. The description should be concise. The precise normative behaviour of the algorithm (input/output, timing etc.) is specified by the RVC-CAL reference code in ISO/IEC 23002-4:2014/Amd 1:2014.

— **Profiles@levels supported**: The profiles@level supported for this functional unit. It may append that a given range of values makes the FU behave for a given profile@level and another range of values makes the FU behave for another profile@level.

— **Input**: A token that is entering the FU through the designated input port. The token type refers to the token pool described in 4.3. The "name" field indicates the input port.

— **Output**: A token that is coming out of the FU through the designated output port. The "name" field indicates the output port.

— **Parameter (*optional*)**: Parameters are optionally described to adjust the behaviour of the FU. All the parameters shall be specified with name, description and range.

— In several FU diagrams, the ports are named with a trailing "_i" for the input port type and with a trailing "_o" for the output port type.

— Some FU diagrams contain as well the Finite State Machine diagram. The following conventions apply: INPUT - the action of reading a token or a set of tokens from the input port, OUTPUT - the action of writing the token or a set of tokens to an output port.

— "Parameter" is set at network configuration stage (cannot be changed during the process) and it is characteristic for each FU.

— Token RANGE: describes the mathematical interval for the token value

EXAMPLE

Token RANGE: { 0, 1} – binary value.

Token RANGE: [0 .. N], value ∈ [0, N] – real values, closed interval.

— All the FUs require the data to be in little-endian format.

© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

Table 1 — Template of description of an FU (example)

FU Name	e.g. Algo_IDCT2D_ISOIEC_23002_1	
Description	e.g. This module computes the 8×8 Inverse Discrete Cosine Transform (IDCT) defined as $f(x,y)=\frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)F(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$ with u, v, x, y = 0, 1, 2, …, N-1 where x, y are spatial coordinates in the sample domain u, v are coordinates in the transform domain $C(u),C(v)=\begin{cases}\frac{1}{\sqrt{2}} & \text{for } u,v=0\\ 1 & \text{otherwise}\end{cases}$ It inputs a list of 64 coefficients and outputs a list of 64 decoded coefficients.	
Profiles@levels supported	e.g. MPEG-4 SP	
Input		
Name	**Token**	
e.g. X	e.g. BLOCK token	
Output		
Name	**Token**	
e.g. Y	e.g. BLOCK token	
Parameter		
Name	**Description**	**Range**

4.2 FU IDs

FU of the specific functionality is identified by its unique identification number. Table 2 lists IDs and names of all FUs in VTL. IDs and names are used in FND to select FUs.

Table 2 — List of FUs and their IDs

ID	FU Name
1	org.sc29.wg11.common.Algo_SynP_Generic
2	org.sc29.wg11.mpeg4.part2.sp.parser.Algo_SynP
3	org.sc29.wg11.mpeg4.part2.sp.parser.Mgnt_BlockExpand
4	org.sc29.wg11.mpeg4.part2.sp.parser.Mgnt_Splitter420B
5	org.sc29.wg11.mpeg4.part2.sp.parser.Mgnt_Splitter420MV
6	org.sc29.wg11.mpeg4.part2.sp.parser.Algo_MVR_MedianOfThreeLeftAndTopAndTopRight
7	org.sc29.wg11.mpeg4.part2.sp.parser.Algo_MVSequence_LeftAndTopAndTopRight
9	org.sc29.wg11.mpeg4.part2.sp.parser.Mgnt_Splitter_420_TYPE
10	org.sc29.wg11.mpeg4.part2.sp.parser.vlc.Algo_VLDtableB6_MPEG4Part2
11	org.sc29.wg11.mpeg4.part2.sp.parser.vlc.Algo_VLDtableB7_MPEG4Part2
12	org.sc29.wg11.mpeg4.part2.sp.parser.vlc.Algo_VLDtableB8_MPEG4Part2
13	org.sc29.wg11.mpeg4.part2.sp.parser.vlc.Algo_VLDtableB12_MPEG4Part2
14	org.sc29.wg11.mpeg4.part2.sp.parser.vlc.Algo_VLDtableB13_MPEG4Part2
15	org.sc29.wg11.mpeg4.part2.sp.parser.vlc.Algo_VLDtableB14_MPEG4Part2

Documents
Document Sharing Hub

© ISO/IEC 2018 — All rights reserved

Full document details: ISO IEC 23002-4-2018

ID	FU Name
16	org.sc29.wg11.mpeg4.part2.sp.parser.vlc.Algo_VLDtableB15_MPEG4Part2
17	org.sc29.wg11.mpeg4.part2.sp.parser.vlc.Algo_VLDtableB16_MPEG4Part2
18	org.sc29.wg11.mpeg4.part2.sp.parser.vlc.Algo_VLDtableB17_MPEG4Part2
19	org.sc29.wg11.mpeg4.part2.sp.texture.Algo_IQ_QSAndQmatrixMp4vOrH263Scaler
20	org.sc29.wg11.mpeg4.part2.sp.texture.dc_reconstruction.Algo_DCRAddr_ThreeLeftTop_8x8
21	org.sc29.wg11.mpeg4.part2.sp.texture.dc_reconstruction.Algo_DCRAddr_ThreeLeftTop_16x16
22	org.sc29.wg11.mpeg4.part2.sp.texture.dc_reconstruction.Algo_DCRInvPred_CHROMA_8x8
23	org.sc29.wg11.mpeg4.part2.sp.texture.dc_reconstruction.Algo_DCRInvPred_LUMA_16x16
24	org.sc29.wg11.mpeg4.part2.sp.texture.Algo_IS_ZigzagOrAlternateHorizontalVertical_8x8
25	org.sc29.wg11.mpeg4.part2.sp.texture.Algo_IAP_AdaptiveHorizontalOrVerticalPred_8x8
26	org.sc29.wg11.mpeg4.part2.sp.texture.Algo_IAP_AdaptiveHorizontalOrVerticalPred_16x16
27	org.sc29.wg11.mpeg4.part2.sp.texture.Algo_IDCT2D_ISOIEC_23002_1
28	org.sc29.wg11.mpeg4.part2.sp.texture.Mgnt_DCSplit
29	org.sc29.wg11.mpeg4.part2.sp.motion.Mgnt_FB_w_Addess_8x8
30	org.sc29.wg11.mpeg4.part2.sp.motion.Mgnt_FB_w_Addess_16x16
31	org.sc29.wg11.mpeg4.part2.sp.motion.Algo_PictureReconstruction_Saturation
32	org.sc29.wg11.mpeg4.part2.sp.motion.Algo_Interp_HalfpelBilinearRoundingControl
33	org.sc29.wg11.mpeg4.part10.cbp.parser.Algo_NALU
34	org.sc29.wg11.mpeg4.part10.cbp.parser.Algo_SynP
35	org.sc29.wg11.mpeg4.part10.cbp.parser.Algo_BlockExpand
36	org.sc29.wg11.mpeg4.part10.cbp.parser.Algo_BlockSplit
37	org.sc29.wg11.mpeg4.part10.cbp.parser.Algo_IntraPred_Split
38	org.sc29.wg11.mpeg4.part10.cbp.parser.Algo_Parser_I_PCM
39	org.sc29.wg11.mpeg4.part10.cbp.selectMacroblock.Algo_DemuxParserInfoForBlocks_Chroma
40	org.sc29.wg11.mpeg4.part10.cbp.selectMacroblock.Algo_DemuxParserInfoForBlocks_Luma
41	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_IS_Zigzag_4x4
42	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_DCR_Hadamard_LUMA_IHT1d
43	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_Transpose4x4
44	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_DCR_Hadamard_LUMA_Reordering
45	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_DCR_Hadamard_LUMA_Scaling
46	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_DCR_Hadamard_CHROMA
47	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_IT4x4_1d
48	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_IT4x4_Addshift
49	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_IntraPred_LUMA_16x16
50	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_IntraPred_LUMA_4x4
51	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_Merge_4x4_to_16x16

© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

ID	FU Name
52	org.sc29.wg11.mpeg4.part10.cbp.Residual.Algo_IQ_QSAndSLAndIDCTScaler_4x4
53	org.sc29.wg11.mpeg4.part10.cbp.Residual.Mgnt_IQ_INTRA16x16
54	org.sc29.wg11.mpeg4.part10.cbp.intraPred.Algo_Merge_4x4_to_8x8
55	org.sc29.wg11.mpeg4.part10.cbp.intraPred.Algo_IntraPred_CHROMA
56	org.sc29.wg11.mpeg4.part10.cbp.intraPred.Mgnt_Intra_16x16
57	org.sc29.wg11.mpeg4.part10.cbp.intraPred.Mgnt_Intra_4x4
58	org.sc29.wg11.mpeg4.part10.cbp.intraPred.Mgnt_IQ_Chroma
59	org.sc29.wg11.mpeg4.part10.cbp.intraPred.Mgnt_Buffer_Neighbour_FullMb
60	org.sc29.wg11.mpeg4.part10.cbp.intraPred.Mgnt_Buffer_Neighbour_YxY
61	org.sc29.wg11.mpeg4.part10.cbp. intraPred.Algo_Split_16x16_to_4x4_norasterscan
62	org.sc29.wg11.mpeg4.part10.cbp. intraPred.Algo_Merge_4x4_to_16x16_norasterscan
63	org.sc29.wg11.mpeg4.part10.cbp. interPred.Algo_Interp_Bilinear
64	org.sc29.wg11.mpeg4.part10.cbp. interPred.Algo_Interp_SeparableSixTapQuarterPel
65	org.sc29.wg11.mpeg4.part10.cbp.interPred.Algo_Interp_Reord
66	org.sc29.wg11.mpeg4.part10.cbp.interPred.Algo_MvLXReconstr
67	org.sc29.wg11.mpeg4.part10.cbp.interPred.Mgnt_DPB
68	org.sc29.wg11.mpeg4.part10.cbp.interPred.Algo_MMCO
69	org.sc29.wg11.mpeg4.part10.cbp.interPred.Algo_RefIdxtoFrameNum
70	org.sc29.wg11.mpeg4.part10.cbp.interPred.Algo_RefList
71	org.sc29.wg11.mpeg4.part10.cbp.interPred.Mgnt_InterPred
72	org.sc29.wg11.mpeg4.part10.cbp.deblockingFilter.Mgnt_DBF_AdaptiveFilter
73	org.sc29.wg11.mpeg4.part10.cbp.deblockingFilter.Algo_DBF_AdaptiveFilter
74	org.sc29.wg11.mpeg4.part10.cbp.display.Mgnt_POC
75	org.sc29.wg11.mpeg4.part10.cbp.display.Mgnt_BufferRender
76	org.sc29.wg11.common.Algo_Add_PixSat
77	org.sc29.wg11.common.Mgnt_Select_MB_4
78	org.sc29.wg11.common.Algo_Byte2bit
79	org.sc29.wg11.common.Algo_Mgnt_Merger420
80	org.sc29.wg11.common. Mgnt_Select_MB_8
81	org.sc29.wg11.mpeg4.part10.cbp.display.Mgnt_Merger420_AVC
82	org.sc29.wg11.mpeg4.part10.php.parser.Algo_SynP
83	org.sc29.wg11.mpeg4.part10.php.parser.Algo_BlockExpand
84	org.sc29.wg11.mpeg4.part10.php.selectMacroblock.Algo_DemuxParserInfoForBlocks_Luma
85	org.sc29.wg11.mpeg4.part10.php.Residual.Algo_IS_Zigzag_8x8
86	org.sc29.wg11.mpeg4.part10.php.Residual.Algo_IQ_QSAndSLAndIDCTScaler_8x8
87	org.sc29.wg11.mpeg4.part10.php.Residual.Algo_IIT_8x8
88	org.sc29.wg11.mpeg4.part10.php.Intrapred.Algo_IntraPred_LUMA_8x8
89	org.sc29.wg11.mpeg4.part10.php.Intrapred.Mgnt_Intra_8x8
90	org.sc29.wg11.mpeg4.part10.php.Residual.Algo_Merge_8x8_to_16x16
91	org.sc29.wg11.mpeg4.part10.php.Residual.Algo_DCR_Hadamard_CHROMA
92	org.sc29.wg11.mpeg4.part10.php.Residual.Algo_DCR_Hadamard_LUMA_Scaling
93	org.sc29.wg11.mpeg4.part10.php.Residual.Algo_IQ_QSAndSLAndIDCTScaler_4x4

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

ID	FU Name
94	org.sc29.wg11.mpeg4.part10.php.Intrapred.Algo_Merge_8x8_to_16x16_norasterscan
95	org.sc29.wg11.mpeg4.part10.php.Intrapred.Algo_Split_16x16_to_8x8_norasterscan
96	org.sc29.wg11.mpeg4.part10.php.Residual.Mgnt_I4x4_I8x8_demux
97	org.sc29.wg11.mpeg4.part10.php.Residual.Mgnt_I4x4_I8x8_mux
98	org.sc29.wg11.mpeg4.part10.php.interPred.Algo_GeneratePredWeight
99	org.sc29.wg11.mpeg4.part10.php.interPred.Mgnt_SelectMvpLX
100	org.sc29.wg11.mpeg4.part10.php.interPred.Algo_MvLXReconstr
101	org.sc29.wg11.mpeg4.part10.php.interPred.Algo_MvBuffer
102	org.sc29.wg11.mpeg4.part10.php.interPred.Mgnt_SelectMvpLX
103	org.sc29.wg11.mpeg4.part10.php.interPred.Mgnt_SelectRefIdx
104	org.sc29.wg11.mpeg4.part10.php.interPred.Algo_FrameNumToPocList
105	org.sc29.wg11.mpeg4.part10.php.deblockingFilter.Algo_DBF_AdaptiveFilter
106	org.sc29.wg11.mpeg4.part10.php.deblockingFilter.Algo_MvComponentReorder
107	org.sc29.wg11.mpeg4.part16.Algo_Parser_SC3DMC
108	org.sc29.wg11.mpeg4.part16.Algo_InverseQuantization1D
109	org.sc29.wg11.mpeg4.part16.Algo_InverseQuantizationND
110	org.sc29.wg11.mpeg4.part16.Algo_InversePrediction1D
111	org.sc29.wg11.mpeg4.part16.Algo_InversePredictionND
112	org.sc29.wg11.mpeg4.part16.Algo_ED_AD_StaticBit
113	org.sc29.wg11.mpeg4.part16.Algo_ED_AD_AdaptiveBit
114	org.sc29.wg11.mpeg4.part16.Algo_ED_VLD
115	org.sc29.wg11.mpeg4.part16.Algo_ED_BitPrecision
116	org.sc29.wg11.mpeg4.part16.Algo_ED_AD
117	org.sc29.wg11.mpeg4.part16.Algo_ED_AD_EG
118	org.sc29.wg11.mpeg4.part16.Algo_ContextModeling
119	org.sc29.wg11.mpeg4.part16.Algo_ContextModeling_SVA_nType
120	org.sc29.wg11.mpeg4.part16.Algo_ContextModeling_SVA_Indexes
121	org.sc29.wg11.mpeg4.part16.Algo_ContextModeling_SVA_Vertex_Attribute
122	org.sc29.wg11.mpeg4.part16.Algo_ED_4bitsD
123	org.sc29.wg11.mpeg4.part16.Algo_ED_FixedLength
124	org.sc29.wg11.mpeg4.part16.Algo_LookUpTable1D
125	org.sc29.wg11.mpeg4.part16.Algo_DecodeConnectivity_SVA
126	org.sc29.wg11.mpeg4.part16.Algo_DecodeConnectivity_TFAN
127	org.sc29.wg11.mpeg4.part16.Algo_ExtractMask_SC3DMC
128	org.sc29.wg11.mpeg4.part16.Algo_ExtractFaceDirection_SVA
129	org.sc29.wg11.mpeg4.part16.Algo_simpleMath_2op
130	org.sc29.wg11.mpeg4.part16.Algo_Connectivity_InversePrediction_SVA
131	org.sc29.wg11.mpeg4.part16.Mgnt_Replicate_1_2
132	org.sc29.wg11.mpeg4.part16.Mgnt_Replicate_1_4
133	org.sc29.wg11.mpeg4.part16.Mgnt_Replicate_1_8
134	org.sc29.wg11.mpeg4.part16.Mgnt_MUX_2_1
135	org.sc29.wg11.mpeg4.part16.Mgnt_MUX_4_1
136	org.sc29.wg11.mpeg4.part16.Mgnt_MUX_8_1
137	org.sc29.wg11.mpeg4.part16.Mgnt_DEMUX_1_2
138	org.sc29.wg11.mpeg4.part16.Mgnt_DEMUX_1_4
139	org.sc29.wg11.mpeg4.part16.Mgnt_DEMUX_1_8
140	org.sc29.wg11.mpeg4.part16.Mgnt_ExtractSegment
141	org.sc29.wg11.mpeg4.part16.Mgnt_ProviderValue
142	org.sc29.wg11.mpeg4.part16.Mgnt_RepeatSegment

Documents
Document Sharing Hub

© ISO/IEC 2018 — All rights reserved

Full document details: ISO IEC 23002-4-2018

ID	FU Name
143	org.sc29.wg11.mpeg4.part16.Mgnt_ExtractBytes
144	org.sc29.wg11.mpeg4.part16.Mgnt_ExtractBits
145	org.sc29.wg11.mpeg4.part16.Mgnt_Provider1D
146	org.sc29.wg11.mpeg4.part16.Mgnt_Provider2D
147	org.sc29.wg11.mpegh.part2.main.intra.selectCU
148	org.sc29.wg11.mpegh.part2.main.inter.Algo_InterPrediction
149	org.sc29.wg11.mpegh.part2.main.inter.Mgnt_DecodedPictureBuffer
150	org.sc29.wg11.mpegh.part2.main.inter.Algo_GenerateRefList
151	org.sc29.wg11.mpegh.part2.main.inter.Algo_MvComponentPred
152	org.sc29.wg11.mpegh.part2.main.Filters.Algo_SaoFilter
153	org.sc29.wg11.mpegh.part2.main.Filters.Algo_GenerateBs
154	org.sc29.wg11.mpegh.part2.main.Filters.Algo_DeblockingFilter
155	org.sc29.wg11.mpegh.part2.main.Filters.Algo_QpGen
156	org.sc29.wg11.mpegh.part2.common.Mgnt_MD5SplitInfo
157	org.sc29.wg11.mpegh.part2.common.Mgnt_padding
158	org.sc29.wg11.mpegh.part2.common.Mgnt_MD5Shifter
159	org.sc29.wg11.mpegh.part2.common.Mgnt_MD5Compute
160	org.sc29.wg11.mpegh.part2.main.IT.Mgnt_IT_Splitter
161	org.sc29.wg11.mpegh.part2.main.IT.Mgnt_IT_Merger
162	org.sc29.wg11.mpegh.part2.main.IT.Mgnt_Block_Merger
163	org.sc29.wg11.mpegh.part2.main.IT.Mgnt_Transpose_4x4
164	org.sc29.wg11.mpegh.part2.main.IT.Mgnt_Transpose_8x8
165	org.sc29.wg11.mpegh.part2.main.IT.Mgnt_Transpose_16x16
166	org.sc29.wg11.mpegh.part2.main.IT.Mgnt_Transpose_32x32
167	org.sc29.wg11.mpegh.part2.main.IT.Mgnt_Transpose_32x32
168	org.sc29.wg11.mpegh.part2.main.IT.Algo_IT4x4_1d
169	org.sc29.wg11.mpegh.part2.main.IT.Algo_IT8x8_1d
170	org.sc29.wg11.mpegh.part2.main.IT.Algo_IT16x16_1d
171	org.sc29.wg11.mpegh.part2.main.IT.Algo_IT32x32_1d
172	org.sc29.wg11.mpegh.part2.main.IT.Algo_invDST4x4_1st
173	org.sc29.wg11.mpegh.part2.main.IT.Algo_invDST4x4_2nd

4.3 Token pool

Every token is listed in the "token pool" that is the table of managing all tokens used in VTL. To facilitate the feasibility of connections among input and output ports of different FUs described in this specification, Table 3 lists all data elements (called "token", which is used throughout this document). The ID field here is informative and used for easy lookup.

Table 3 — List of all token types that are used in the descriptions of FUs in this subclause

ID & Name	Description
1 BIT	Token which value is 0 or 1. The bits belongs to the non-decoded bitstream.
2 ACKNOWLEDGMENT	Boolean token (True or False) indicating a acknowledgment. True means it is ok, False, not ok.
3 MCBPC	Token representing the MCBPC element of syntax
4 CBPY	Token representing the CBPY element of syntax
5 DCT_DC_SIZE	Token representing the element of syntax DCT_DC_SIZE
6 DCT_DC_DIFF	Token representing the element of syntax DCT_DC_DIFF

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

ID & Name	Description
7 RUN	Token representing the RUN value in the decoding of the DCT coefficients
8 VALUE	Token representing the VALUE value in the decoding of the DCT coefficients
9 LAST	Token representing the LAST value in the decoding of the DCT coefficients
10 MEM_ADDRESS	Token representing an address in the memory of the frames
11 MEM_WIDTH	Token representing the width in pixel of the data to read or write in the memory of the frames starting from the MEM_ADDRESS information
12 MEM_HEIGHT	Token representing the height in pixel of the data to read or write in the memory of the frames starting from the MEM_ADDRESS information
13 MEM_DATA	Token representing a data stored in the memory of the frames
14 WIDTH	Token representing the width value of video frame in pixels
15 HEIGHT	Token representing the height value of video frame in pixels
16 SIZE	Token representing the size of the current frame in number of macroblock in width and number of macroblock in height
17 DC	Tokens representing the DC coefficients. Each token represent one coefficient
18 AC	Token representing AC coefficients without DC coefficients
19 BLOCK	Token representing BLOCK that consists of 8×8 pixels
20 MB	Token representing a macroblock that consists of BLOCKs
21 MVD	Tokens representing the motion vector differences decoded by the syntax parsing process
22 MV	Tokens representing the coordinates of the motion vectors
23 QUANT	Token representing the QUANT value of quantization
24 COORDINATE	Token representing coordinates of block or macroblocks
25 DISPLACEMENT	Token representing the displacement between pixels (e.g., half- or quarter-pixel)
26 SIGN	Token representing a sign.
27 ROUND	Boolean token (True or False) indicating whether rounding is to be made or not
28 BTYPE	Integer token (2 = INTRA), (1 = INTER), (15 = NEW VOP)
29 ACCODED	Boolean token (True or False) indicating whether AC is coded or not
30 ACPRED	Boolean token (True or False) indicating whether AC prediction is made or not
31 ACPRED_DIR	Token representing the order of prediction of the AC coefficients
32 MOTION	Boolean token (True or False) indicating whether motion predication is made or not
33 FOURMV	Boolean token (True or False) indicating whether FOURMV is to be used or not

© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

ID & Name	Description
34 F_CODE	Token representing a value of FCODE of VOP to specify the range of motion vectors
35 RBSP	Token representing the data in the Raw Byte Sequence Payload
36 NAL_SIZE	Token representing the size in byte of a Network Abstraction Layer unit
37 SUB_MB_TYPE	Token representing sub mb type value for sub partition
38 FRAME_NUM	Token representing the frame number of the current picture to decode
39 FRAME_TO_READ	Token representing the frame to read
40 PART_SIZE	Token representing the size in pixel for a partition of a macroblock, first the width of the partition, then the height
41 REF_ID	Token representing the identification of the decoded reference frame in memory
42 MB_ID	Token representing the number wich identified a macroblock in a frame in position x and y. The macroblocks are counted in a frame using raster scan order.
43 POC	Token representing the index of the frame to display
44 REF_ORDER	Token representing the index of frames to store into long frame reference and short terme reference
45 MMCO	Token representing the order of the index for frame to store in memory
46 PRED_MODE_INTRA	Token representing the prediction mode of an intra macroblock
47 MB_TYPE	Token representing the type of prediction used by a macroblock (Intra, Intra 4×4, Inter, new frame IDR, non IDR or new slice)
48 SLICE_DBF_PARAM	Tokens representing for each slice if deblocking filter is enabled and the offset used in accessing the α and tC0 deblocking filter tables for filtering operations and the offset used in accessing the β deblocking filter table for filtering operations for each slice
49 MB_DBF_PARAM	Tokens representing for each MB the offset used in accessing the β deblocking filter table for filtering operations
50 CBP_BLK	Token representing which of the sixteen 4×4 luma blocks of a macroblock may contain non-zero transform coefficient levels
51 SELECT	Token representing which blocks will selected (from 0-3)
52 ST_LIST	Token representing the short term reference list
53 ST_LIST_SZ	Token representing the size of the short term reference list
54 DEL_LIST	Token representing the list of reference frames that have to be deleted
55 BOOLEAN	Token which value is 0 or 1
56 SIGN	Token which value is 0 or 1
57 FLAG	Token which value is 0 or 1
58 UINT_2	Unsigned integer on 2 bits
59 UINT_4	Unsigned integer on 4 bits

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

ID & Name	Description
60 UINT_8	Unsigned integer on 8 bits
61 UINT_16	Unsigned integer on 16 bits
62 UINT_32	Unsigned integer on 32 bits

4.4 Array data order

— **Row based:** The data is processed or sent in a sequential order, row by row.

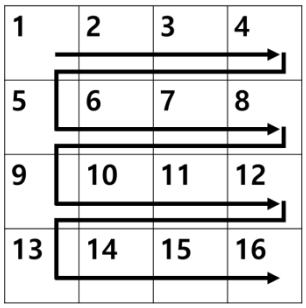


Figure 3 — Example of row-based array data order

— **Column based:** The data is processed or sent in a sequential order, column by column.

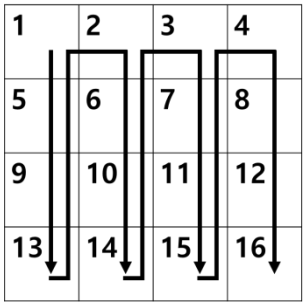


Figure 4 — Example of column-based array data order

4.5 Input ports (reset_i, init_i, start_i)

An FU does not have an outside synchronization signal or synchronization mechanism. These ports are used for the purpose of changing the values of the local variables to default or initialization values.

4.6 FU block diagram notations

The notation [EMBED] defines a part of the main FSM schematic that is described as a separate schematic (for complexity reasons). The [EMBED] schematic is an integrated part of the main FSM schematic.

The notation [MODULE] defines a part of the main FSM schematic that is defined as a separate FU. The module schematic is integrated in the main schematic with the entire FU logic, except the “START” FSM state. The INPUT/OUTPUT states do not read or write values from the ports, they refer to local variables relative to the FU that embeds the other schematic.

© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

4.7 Conventions

The significance of the "sign" port values is:

Value	Significance
0	negative
1	positive

The significance of the "flag" values is:

Value	Significance
0	false
1	true

4.8 Granular FU

When a decoder is implemented using the FUs, two or more instances of FUs can be grouped and can be dealt as a single instance of FU with bigger granularity. This is called Granular FU, and the use of Granular FU during implementation is normatively allowed. More detailed description about Granular FU concept is given in Annex D, and informative examples of Granular FU can be found from Annex E.

5 General-purpose FUs

5.1 Syntax parsing

5.1.1 Generic syntax parser

FU Name	Algo_SynP_Generic	
Description	This is a generic syntax parser that needs BSD as an input. Input and output port will be defined as the information in the BSD.	
Profiles@levels supported	All (Generic)	
Input		
Name	**Token**	
Output		
Name	**Token**	
Parameter		
Name	**Description**	**Range**

5.1.2 Algo_Byte2bit

FU Name	Algo_Byte2bit	
Description	This module will take 8 bits in once and will output them one by one.	
Profiles@levels supported	All (Generic)	
Input		
Name	**Token**	
BYTE	BIT token	
Output		
Name	**Token**	
BITS	BIT token	
Parameter		
Name	**Description**	**Range**
Package		
package org.sc29.wg11.common		

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

5.1.3 Mgnt_Select_MB_4

FU Name	Mgnt_Select_MB_4	
Description	This module selects data from one of its 4 inputs and sends it to its output (X) according to select information.	
Profiles@levels supported	MPEG-4 AVC Constrained BP	
Input		
Name	**Token**	
IN_0	MB token	
IN_1	MB token	
IN_2	MB token	
IN_3	MB token	
SELECT	SELECT token	
Output		
Name	**Token**	
OUT	MB token	
Parameter		
Name	**Description**	**Range**
MB_WIDTH	Size in pixel of a macroblock	[8;16]
Package		
package org.sc29.wg11.common		

5.1.4 Mgnt_Merger420

FU Name	Mgnt_Merger420	
Description	This module merges data from YUV to compose a MB 16×16.	
Profiles@levels supported	MPEG-4 AVC Constrained BP	
Input		
Name	**Token**	
Y	BLOCK token	
U	BLOCK token	
V	BLOCK token	
Select	SELECT token	
Output		
Name	**Token**	
YUV	MB token	
Parameter		
Name	**Description**	**Range**
MB_WIDTH	Size in pixel of a macroblock	[8;16]
Package		
package org.sc29.wg11.common		

5.1.5 Mgnt_Select_MB_8

FU Name	Mgnt_Select_MB_8
Description	This module selects data from one of its 8 inputs and sends it to its output according to select information.
Profiles@levels supported	MPEG-4 AVC Constrained BP

© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

Input		
Name	**Token**	
IN_0	MB token	
IN_1	MB token	
IN_2	MB token	
IN_3	MB token	
IN_4	MB token	
IN_5	MB token	
IN_6	MB token	
IN_7	MB token	
SELECT	SELECT token	
Output		
Name	**Token**	
OUT	MB token	
Parameter		
Name	**Description**	**Range**
MB_WIDTH	Size in pixel of a macroblock	[8;16]
Package		
package org.sc29.wg11.common		

5.1.6 Mgnt_SynP_DEMUX

FU Name	Mgnt_SynP_DEMUX	
Description	This DEMUX FU can be used to distribute serialized output through a number of different output ports. A potential use of this FU is distribution of output tokens derived from the generic syntax parser FU with serial output. The value of PORT_ID is used to designate the output port which the data from PORT_DATA will be sent. The output port will be defined as the information in the BSD or the FN. NOTE FU descriptions for the Informative examples of this FU can be found in Annex I.	
Profiles@levels supported	All (Generic)	
Input		
Name	**Token**	
PORT_ID	BIT token	
PORT_DATA	BIT token	
Output		
Name	**Token**	
Parameter		
Name	**Description**	**Range**

5.1.7 Mgnt_SynP_BytePreprocessor

FU Name	Mgnt_SynP_BytePreprocessor
Description	This FU pre-processes the input bitstream to remove the emulation prevention codes (i.e. decodes NAL RBSP structure). The output of this FU is a raw encoded video bistream.
Profiles@levels supported	HEVC Main Profile
Input	
Name	**Token**
BITS_IN	BIT token

© ISO/IEC 2018 — All rights reserved

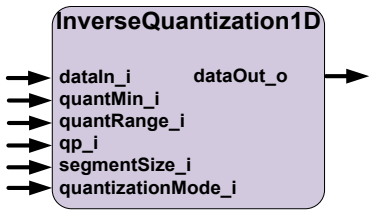
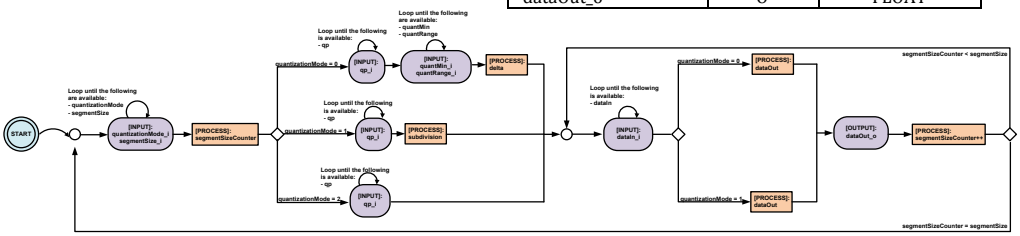
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

Output		
Name	**Token**	
BITS_OUT	BIT token	
Parameter		
Name	**Description**	**Range**
Package		
package org.sc29.wg11.mpegh.part2.main.synParser		

5.2 General processing FUs

5.2.1 Algo_InverseQuantization1D

FU Name	Algo_InverseQuantization1D

Description	
	
	Port Name / Direction (I/O) / Token TYPE
	dataIn_i / I / INT8, INT16, INT32, INT64
	qp_i / I / UINT_32
	quantMin_i / I / FLOAT
	quantRange_i / I / FLOAT
	segmentSize_i / I / UINT8, UINT16, UINT32, UINT64
	quantizationMode_i / I / UINT_2
	dataOut_o / O / FLOAT
	 
	
	Inverse Quantization Process:
	START:
	INPUT_SEGMENT_PARAM:
	INPUT:
	quantizationMode
	segmentSize
	SegmentSizeCounter = 0
	IF quantizatioMode = 0
	INPUT:
	qp
	quantMin
	quantRange
	IF (quantRange > 0.0)
	delta = ((1 << qp) - 1) / quantRange
	ELSE
	delta = 1.0
	IF quantizationMode = 1
	INPUT:
	qp
	INPUT_DATA_IN:
	INPUT:
	dataIn
	IF quantizatioMode = 0
	PROCESS:
	dataOut = quantMin + (dataIn / delta)
	IF quantizatioMode = 1
	PROCESS:
	dataOut = dataIn / qp
	OUTPUT
	dataOut
	SegmentSizeCounter ++

© ISO/IEC 2018 — All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

```
IF SegmentSizeCounter < segmentSize
    GOTO INPUT_DATA_IN
ELSE
    GOTO INPUT_SEGMENT_PARAM
```

The following table contains the quantization types index used in the Inverse Quantization 1D FU:

Name of quantization mode	Value
Uniform Quantization	**0**
Uniform Texture Quantization	**1**

The "Inverse Quantization" is an algorithm (step by step procedures) that allows a set of data to be represented with a limited set of values that are associated with its nearest representative.
For a number of "segmentSize" of input data (dataIn), it uses the same set of quantMin, quantRange and quantValue to produce a set of output data (dataOut) of size "segmentSize".

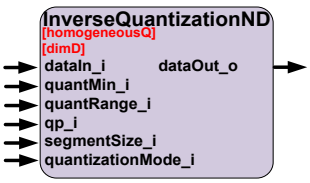
ISO Standards using the FU	ISO/IEC 14496-16
Profiles@levels supported	All (Generic)

Input	
Name	**Token**
dataIn_i	INT8, INT16, INT32, INT64
qp_i	UINT_32
quantMin_i	FLOAT
quantRange_i	FLOAT
segmentSize_i	UINT8, UINT16, UINT32, UINT64
quantizationMode_i	UINT_2
Output	
Name	**Token**
dataOut_o	FLOAT
Package	
package org.sc29.wg11.mpeg4.part16	

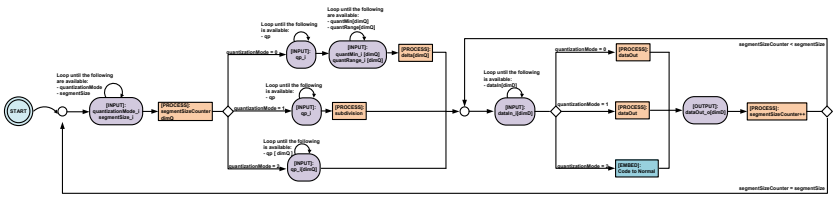
5.2.2 Algo_InverseQuantizationND

FU Name: Algo_InverseQuantizationND

Description



Port Name	Direction (I/O)	Token TYPE
dataIn_i	I	INT8, INT16, INT32, INT64
qp_i	I	UINT_32
quantMin_i	I	FLOAT
quantRange_i	I	FLOAT
segmentSize_i	I	UINT8, UINT16, UINT32, UINT64
quantizationMode_i	I	UINT_2
dataOut_o	O	FLOAT



© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

Inverse Quantization Process:

$$dimQ = \begin{cases} dimD, if\ homogeneousQ = 0 \\ 1, if\ homogeneousQ = 1 \end{cases}$$

```
START:
INPUT_SEGMENT_PARAM
INPUT:
        quantizationMode
        segmentSize
SegmentSizeCounter = 0
IF quantizatioMode = 0
    INPUT:
        qp
        quantMin [ dimQ ]
        quantRange [ dimQ ]
    WHILE dimQ_counter < dimQ
        IF ( quantRange [ dimQ_counter ] > 0.0 )
            delta [ dimQ_counter ] = ( ( 1 << qp ) - 1 ) / quantRange [ dimQ_counter ]
        ELSE
            delta [ dimQ_counter ] = 1.0;
        dimQ_counter++
IF quantizationMode = 1
    INPUT:
        Qp [ dimQ ]
IF quantizationMode = 2
    INPUT:
        qp
    PROCESS:
        Subdivision = (qp  - 3) / 2

INPUT_DATA_IN:
INPUT:
    dataIn [ dimQ ]
IF quantizatioMode = 0
    PROCESS:
        dataOut = quantMin [ segmentSizeCounter%dimD ] + (dataIn / delta [segmentSizeCounter%dimD])
IF quantizatioMode = 1
    PROCESS: EMBED Code 2 Normal
IF quantizatioMode = 2
    PROCESS:
        dataOut =  dataIn / qp [ segmentSizeCounter % dimD ]
OUTPUT
    dataOut
SegmentSizeCounter ++
IF SegmentSizeCounter < segmentSize
    GOTO INPUT_DATA_IN
ELSE
    GOTO INPUT_SEGMENT_PARAM


EMBED: Code 2 Normal
    Mask = ( 1 << ( 2 * subdivision) ) - 1
    tricode = data & mask;

    // Find y coordinate by solving 2nd degree equation
    factor = 1 << subdivision
    y = factor - sqrt ( (factor^2) - tricode )
    tricode = tricode + ( y * ( y - (2 * factor) ) )
    x = tricode / 2
    upsideDown = tricode % 2

    // Calculate coordinates for all vertices in triangle
    v1x = x + upsideDown
    v1y = y + upsideDown
    v2x = x + 1
    v2y = y
    v3x = x
    v3y = y + 1

    // Calculate coordinates of barycenter
    invMaxCoord = 1 / factor
    normal [ 0 ] = (v1x + v2x + v3x) * invMaxCoord
```


© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

```
normal [ 1 ] = (v1y + v2y + v3y) * invMaxCoord
normal [ 2 ] = 3 - normal [ 0 ] - normal [ 1 ]

// Flip component signs if necessary
octantCode = ( data >> 2 * subdivision ) & 0x7
if  (octantCode & 0x4)
        normal [ 0 ] = (-1) * normal [ 0 ]
if (octantCode & 0x2)
        normal [ 1 ] = (-1) * normal [ 1 ]
if (octantCode & 0x1)
        normal [ 2 ] = (-1) * normal [ 2 ]

invNorm:= 1 / sqrt ( (normal[ 0 ])^2 + (normal [ 1 ])^2 + (normal [ 2 ])^2 );

//Write the 3 output values
normal [ 0 ] = normal [ 0 ] * invNorm
normal [ 1 ] = normal [ 1 ] * invNorm
normal [ 2 ] = normal [ 2 ] * invNorm

dataOut = normal
```

The following table contains the quantization types index used in the Inverse Quantization ND FU:

Name of quantization mode	Value
Uniform Quantization	**0**
Normal Quantization	**1**
Uniform Texture Quantization	**2**

The "Inverse Quantization" is an algorithm (step by step procedures) that allows a set of data to be represented with a limited set of values that are associated with its nearest representative.
For a number of "segmentSize" x "dimD" of input data (dataIn), it uses the same set of quantMin, quantRange and quantValue of size "dimD" to produce a set of output data (dataOut) of size segmentSize" x "dimD".
For each set of size "dimD" of input data (dataIn) it uses the corresponding value of quantMin, quantRange and quantValue.

ISO Standards using the FU	ISO/IEC 14496-16
Profiles@levels supported	All (Generic)

Input

Name	Token
dataIn_i	INT8, INT16, INT32, INT64
qp_i	UINT_32
quantMin_i	FLOAT
quantRange_i	FLOAT
segmentSize_i	UINT8, UINT16, UINT32, UINT64
quantizationMode_i	UINT_2

Output

Name	Token
dataOut_o	FLOAT

Parameter

Name	Description	Type/range
dimD	Describes the number of tokens of type dataIn_i that are consumed at each firing. This parameter is set at the network configuration level.	Type: Integer Range: [1 .. 2^5]
homogeneousQ	Describes the number of tokens of type quantRange_i, quantMin_i and quantValue_i that are necessary for the inverse quantization process. This parameter is set at the network configuration level. The number of tokens is equal to dimD if this parameter is 0 and the number of tokens is equal to 1 if this parameter is 1	Type: Boolean Range: {0,1}

Package

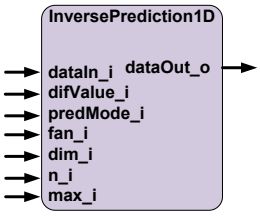
package org.sc29.wg11.mpeg4.part16

© ISO/IEC 2018 — All rights reserved

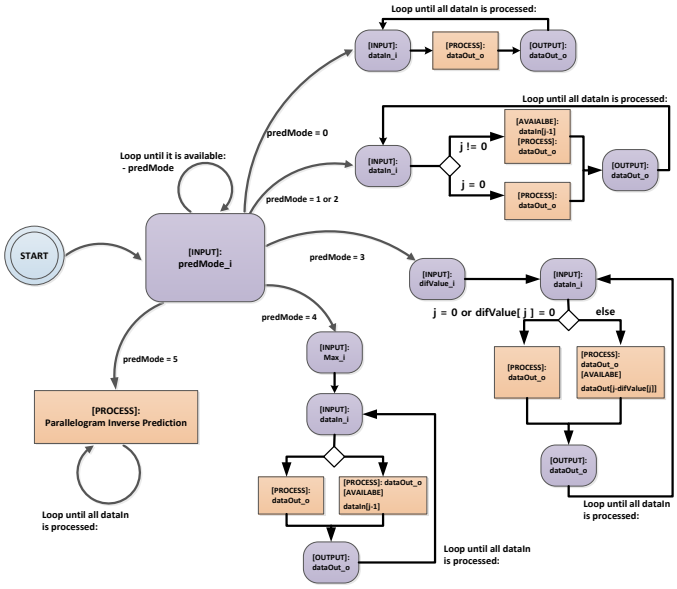
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

5.2.3 Algo_InversePrediction1D

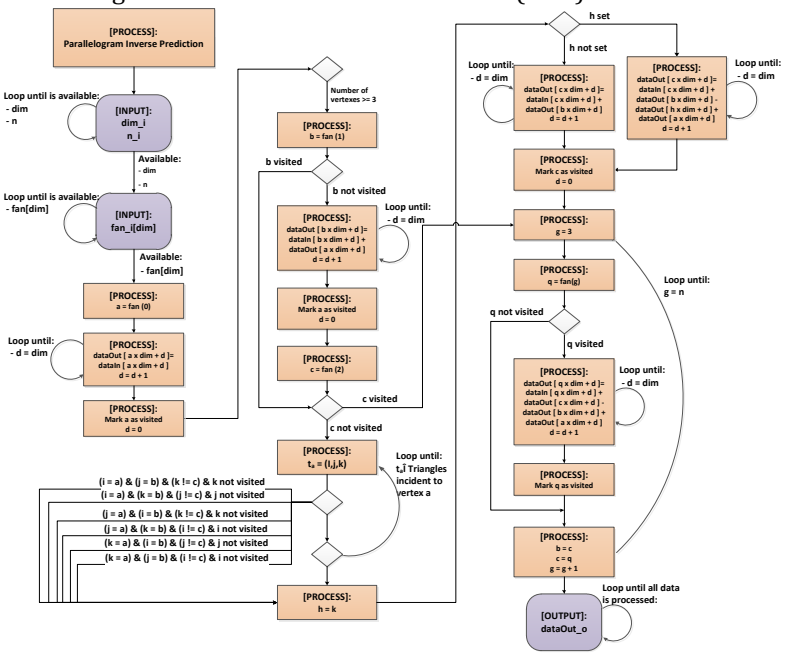
FU Name	Algo_InversePrediction1D
Description	This FU allows five modes of inverse prediction. Not all the ports are used for each prediction mode. This FU can handle one input token at a time.



Port Name	Direction (I/O)	Token TYPE
dataIn_i	I	INT8, INT16, INT32, INT64
difValue_i	I	INT8, INT16, INT32, INT64
predMode_i	I	UINT_4
fan_i	I	INT8, INT16, INT32, INT64
dim_i	I	UINT8, UINT16, UINT32, UINT64
n_i	I	INT8, INT16, INT32, INT64
max_i	I	UINT8, UINT16, UINT32, UINT64
dataOut_o	O	INT8, INT16, INT32, INT64



Process Parallelogram Inverse Prediction Schematic (FSM):



© ISO/IEC 2018 — All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

Inverse Prediction Process:

Switch (predMode)
{

Case 0: NP – No Prediction

$$dataOut[j] = dataIn[j], \qquad \forall j \in \{0 \,..\, N-1\}$$

Case 1: Diff – Differential Prediction

$$dataOut[j] = \begin{cases} dataIn[j], \forall j \in 0 \\ dataIn[j] + dataOut[j-1], \forall\, j \in \{1 \,..\, N-1\} \end{cases}$$

Case 2: XOR – based prediction

$$dataOut[j] = \begin{cases} dataIn[j], \forall j \in 0 \\ dataIn[j] \otimes dataOut[j-1], \forall\, j \in \{1 \,..\, N-1\} \end{cases}$$

Case 3: Adaptive Prediction

$$dataOut[j] = \begin{cases} dataIn[j], if\ difValue[j] = 0\ or\ j = 0 \\ dataIn[j] + dataOut[j - difValue[j]], otherwise \end{cases}$$

Case 4: Circular Differential Prediction

$$dataOut[j] = \begin{cases} dataIn[j], \forall j \in 0 \\ d, if\ dataIn[j] < dataIn[j-1], where\ d = (\ldots) \\ -d, otherwise \end{cases}$$

$$d = \begin{cases} dataIn[j-1] + M_d - dataIn[j], if\ dataIn[j] > dataIn[j-1] \\ dataIn[j] + M_d - dataIn[j-1], otherwise \end{cases}$$

Case 5: Parallelogram Inverse Prediction

```
a = fan ( 0 )
if a not visited
    d = 0
    WHILE d < dim
        dataOut [ a x dim + d ] = dataIn [ a x dim + d ]
        d = d + 1
    Mark a as visited
If number of vertexes > 3
    b = fan ( 1 )
    if b not visited
        d = 0
        WHILE d < dim
            dataOut [ b x dim + d ] = dataIn [ b x dim + d ] + dataOut  [ a x dim + d ]
            d = d + 1
        Mark b as visited
    c = fan ( 2 )
    if c not visited
        init h, ta
        WHILE ta = (I,j,k) ∈Triangles incident to vertex a
            If ( i=a & j=b & k≠c and k not visited ) h = k, break
            If ( i=a & k=b & j≠c and j not visited ) h = k, break
            If ( j=a & i=b & k≠c and k not visited ) h = k, break
            If ( j=a & k=b & i≠c and i not visited ) h = k, break
            If ( k=a & i=b & j≠c and j not visited ) h = k, break
            If ( k=a & b=b & i≠c and i not visited ) h = k, break
        If h not set
            d = 0
            WHILE d < dim
                dataOut [ c x dim + d ]=dataIn [ c x dim + d ]+ dataOut [ b x dim + d ]
                d = d + 1
        else
            d = 0
            WHILE d < dim
            dataOut [ c x dim + d ] =
                                dataIn [ c x dim + d ] + dataOut [ b x dim + d ]
                                – dataOut [ h x dim + d ] + dataOut [ a x dim + d ]
            d = d + 1
```

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

```
            Mark c as visited
        g = 3, d = 0
        WHILE g < fanSize
            q = fan ( g )
            if q not visited
                WHILE d < dim
                    dataOut [ q x dim + d ] =
                                dataIn [ q x dim + d ] + dataOut [ c x dim + d ]
                                - dataOut [ b x dim + d ] + dataOut [ a x dim + d ]
                    d = d + 1
                b = c
                c = q
```

The following table contains the prediction types index used in the Inverse Prediction 1D FU:

Name of prediction mode	Value
No Prediction	**0**
Differential Prediction	**1**
XOR based prediction	**2**
Adaptive Differential Prediction	**3**
Circular Differential Prediction	**4**
Parallelogram Inverse Prediction	**5**

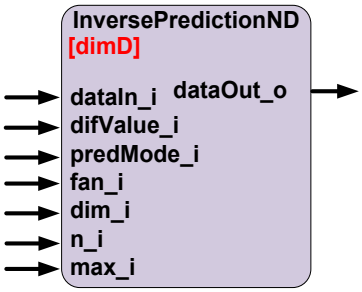
The detailed description of the inverse parallelogram prediction is described in Annex F.

ISO Standards using the FU	ISO/IEC 14496-16	
Profiles@levels supported	All (Generic)	
Input		
Name	**Token**	
dataIn_i	INT8, INT16, INT32, INT64	
difValue_i	INT8, INT16, INT32, INT64	
predMode_i	UINT_4	
fan_i	INT8, INT16, INT32, INT64	
dim_i	UINT8, UINT16, UINT32, UINT64	
n_i	INT8, INT16, INT32, INT64	
max_i	UINT8, UINT16, UINT32, UINT64	
Output		
Name	**Token**	
dataOut_o	INT8, INT16, INT32, INT64	
Parameter		
Name	**Description**	**Range**
Package		
package org.sc29.wg11.mpeg4.part16		

5.2.4 Algo_InversePredictionND

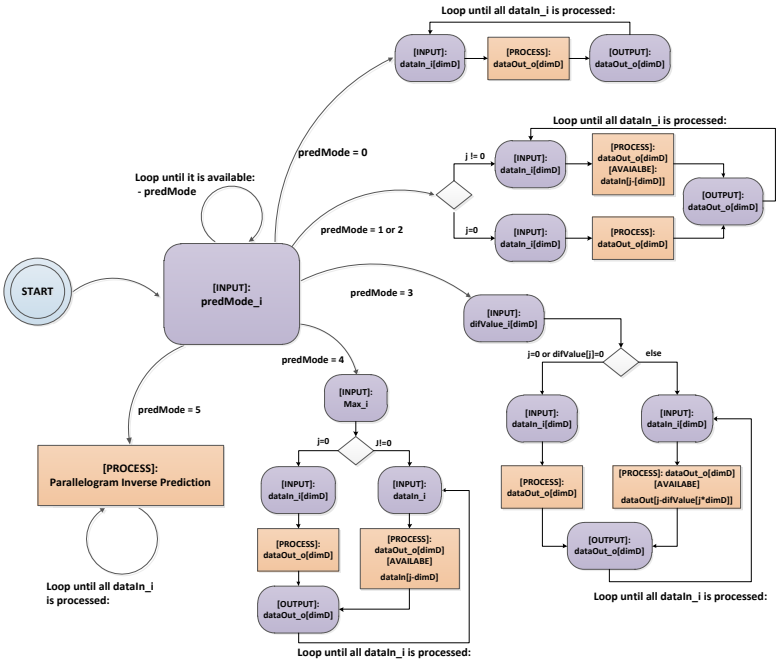
FU Name	Algo_InversePredictionND
Description	This FU allows five modes of inverse prediction. Not all the ports are used for each prediction mode. This FU can handle a number of dimD input tokens at a time.

© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018



Port Name	Direction (I/O)	Token RANGE
dataIn_i	I	INT8, INT16, INT32, INT64
difValue_i	I	INT8, INT16, INT32, INT64
predMode_i	I	UINT_4
fan_i	I	INT8, INT16, INT32, INT64
dim_i	I	UINT8, UINT16, UINT32, UINT64
n_i	I	INT8, INT16, INT32, INT64
max_i	I	UINT8, UINT16, UINT32, UINT64
dataOut_o	O	INT8, INT16, INT32, INT64

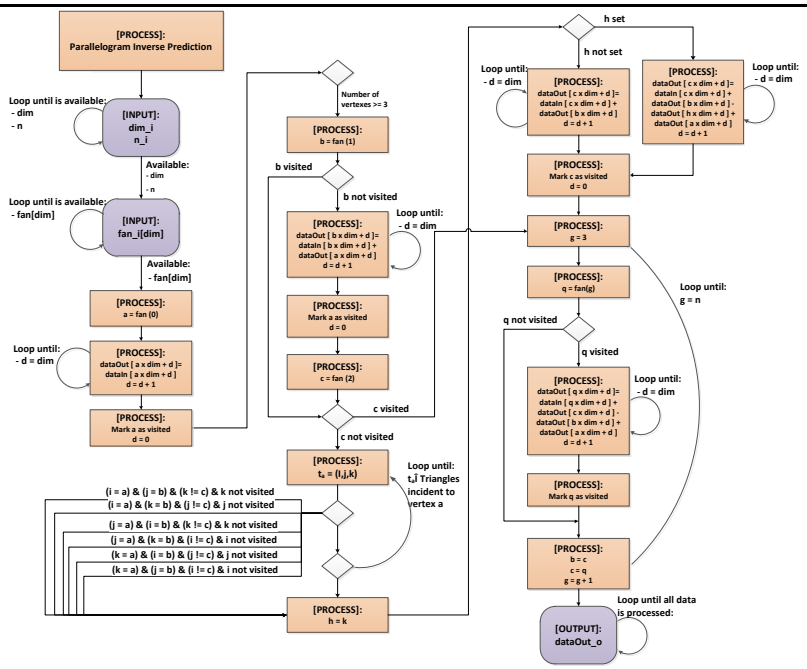
Process Schematic (FSM):



Process Parallelogram Inverse Prediction Schematic (FSM):

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018



Inverse Prediction ND Process:
Switch(predMode)
{

Case 0: NP – No Prediction

$$dataOut[j] = dataIn[j], \qquad \forall j \in \{0\,..\,(N \times dimD) - 1\}$$

Case 1: Diff – Differential Prediction

$$dataOut[j] = \begin{cases} dataIn[j], \forall j \in 0 \\ dataIn[j] + dataOut[j - dimD], \forall\, j \in \{1\,..\,(N \times dimD) - 1\} \end{cases}$$

Case 2: XOR – based prediction

$$dataOut[j] = \begin{cases} dataIn[j], \forall j \in 0 \\ dataIn[j] \otimes dataOut[j - dimD], \forall\, j \in \{1\,..\,(N \times dimD) - 1\} \end{cases}$$

Case 3: Adaptive Prediction

$$dataOut[j] = \begin{cases} dataIn[j], if\ difValue[j] = 0\ or\ j = 0 \\ dataIn[j] + dataOut[j - difValue[j] \times dimD], otherwise \end{cases}$$

Case 4: Circular Differential Prediction

$$dataOut[j] = \begin{cases} dataIn[j], if\ difValue[j] = 0\ or\ j = 0 \\ dataIn[j] + dataOut[j - difValue[j] \times dimD], otherwise \end{cases}$$

$$d = \begin{cases} dataIn[j - dimD] + M_d - dataIn[j], if\ dataIn[j] > dataIn[j - dimD] \\ dataIn[j] + M_d - dataIn[j - dimD], otherwise \end{cases}$$

}

Case 5: Parallelogram Inverse Prediction

```
a = fan ( 0 )
if a not visited
    d = 0
    WHILE d < dim
        dataOut [ a x dim + d ] = dataIn [ a x dim + d ]
        d = d + 1
    Mark a as visited
If number of vertexes > 3
    b = fan ( 1 )
    if b not visited
        d = 0
        WHILE d < dim
```


© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

```
                dataOut [ b x dim + d ] = dataIn [ b x dim + d ] + dataOut  [ a x dim + d ]
                d = d + 1
            Mark b as visited
        c = fan ( 2 )
        if c not visited
            init h, ta
            WHILE ta = (I,j,k) ∈Triangles incident to vertex a
                If ( i=a & j=b & k≠c and k not visited ) h = k, break
                If ( i=a & k=b & j≠c and j not visited ) h = k, break
                If ( j=a & i=b & k≠c and k not visited ) h = k, break
                If ( j=a & k=b & i≠c and i not visited ) h = k, break
                If ( k=a & i=b & j≠c and j not visited ) h = k, break
                If ( k=a & b=b & i≠c and i not visited ) h = k, break
            If h not set
                d = 0
                WHILE d < dim
                    dataOut [ c x dim + d ]=dataIn [ c x dim + d ]+ dataOut [ b x dim + d ]
                    d = d + 1
            else
                d = 0
                WHILE d < dim
                dataOut [ c x dim + d ] =
                                dataIn [ c x dim + d ] + dataOut [ b x dim + d ]
                                – dataOut [ h x dim + d ] + dataOut [ a x dim + d ]
                d = d + 1
            Mark c as visited
        g = 3, d = 0
        WHILE g < fanSize
            q = fan ( g )
            if q not visited
                WHILE d < dim
                    dataOut [ q x dim + d ] =
                            dataIn [ q x dim + d ] + dataOut [ c x dim + d ]
                            - dataOut [ b x dim + d ] + dataOut [ a x dim + d ]
                    d = d + 1
                b = c
                c = q
```

The following table contains the prediction types index used in the Inverse Prediction ND FU:

Name of prediction mode	Value
No Prediction	**0**
Differential Prediction	**1**
XOR based prediction	**2**
Adaptive Differential Prediction	**3**
Circular Differential Prediction	**4**
Parallelogram Inverse Prediction	**5**

The detailed description of the inverse parallelogram prediction is described in Annex F.

ISO Standards using the FU	ISO/IEC 14496-16
Profiles@levels supported	All (Generic)
Input	
Name	**Token**
dataIn_i	INT8, INT16, INT32, INT64
difValue_i	INT8, INT16, INT32, INT64
predMode_i	UINT_4
fan_i	INT8, INT16, INT32, INT64
dim_i	UINT8, UINT16, UINT32, UINT64
n_i	INT8, INT16, INT32, INT64
max_i	UINT8, UINT16, UINT32, UINT64
Output	
Name	**Token**
dataOut_o	INT8, INT16, INT32, INT64

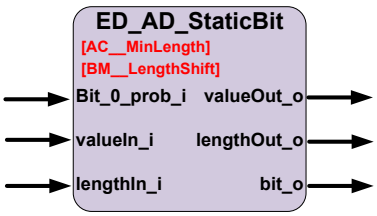
© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

Parameter		
Name	**Description**	**Range**
dimD	Describes the number of tokens of type dataIn_i that are consumed at each firing. This parameter is set at the network configuration level.	Type: Integer Range: [1 .. 2^5]
Package		
package org.sc29.wg11.mpeg4.part16		

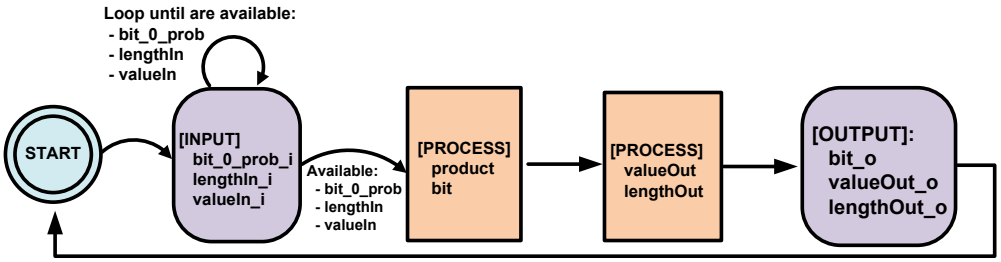
5.2.5 Algo_ED_AD_StaticBit

FU Name	Algo_ED_AD_StaticBit
Description	This FU describes the arithmetic decoding process based on a static bit model.



Port Name	Direction (I/O)	Token RANGE
Bit_0_prob_i	I	UINT8, UINT16, UINT32, UINT64
valueIn_i	I	UINT8, UINT16, UINT32, UINT64
length_i	I	UINT8, UINT16, UINT32, UINT64
valueOut_o	O	UINT8, UINT16, UINT32, UINT64
lengthOut_o	O	UINT8, UINT16, UINT32, UINT64
bit_o	O	UINT8, UINT16, UINT32, UINT64

Process Schematic (FSM):



ED_AD_StaticBit Process:

```
START
INPUT:
        Bit_0_prob_i
        lengthIn_i
        valueIn_i
product = bit_0_prob * ( lengthIn >> BM_LengthShift )
bit = ( valueIn >= product )
if ( bit == 0 )
        valueOut = valueIn
        lengthOut = product
else
        valueOut = valueIn – product
        lengthOut = lengthIn – product
OUTPUT:
valueOut_o
lengthOut_o
GOTO START
```

ISO Standards using the FU	ISO/IEC 14496-16
Profiles@levels supported	All (Generic)
Input	
Name	**Token**
Bit_0_prob_i	UINT8, UINT16, UINT32, UINT64
valueIn_i	UINT8, UINT16, UINT32, UINT64

© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

length_i	UINT8, UINT16, UINT32, UINT64	
Output		
Name	**Token**	
valueOut_o	UINT8, UINT16, UINT32, UINT64	
lengthOut_o	UINT8, UINT16, UINT32, UINT64	
bit_o	UINT8, UINT16, UINT32, UINT64	
Parameter		
Name	**Description**	**Range**
AC__MinLength	Describes the threshold for renormalization. This parameter is set at the network configuration level.	Type:Unsigned Integer Range: [1 .. 2^{32}]
BM__LengthShift	Describes the length bits discarded before multiplication. This parameter is set at the network configuration level.	Type:Unsigned Integer Range: [1 .. 2^{5}]
Package		
package org.sc29.wg11.mpeg4.part16		

© ISO/IEC 2018 — All rights reserved

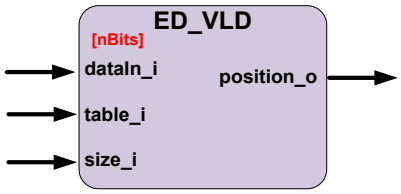
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

5.2.6 Algo_ED_VLD

FU Name: Algo_ED_VLD

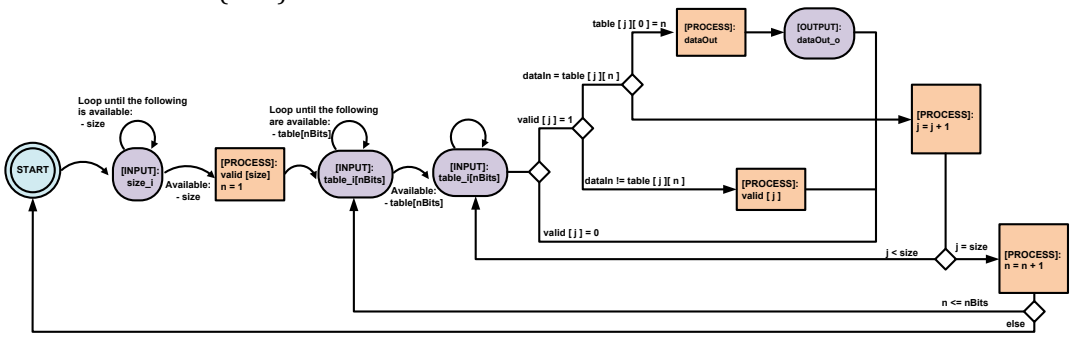
Description

This FU describes the Variable Length Decoding process.



Port Name	Direction (I/O)	Token RANGE
dataIn_i	I	UINT8, UINT16, UINT32, UINT64
table_i	I	UINT8, UINT16, UINT32, UINT64
size_i	I	UINT8, UINT16, UINT32, UINT64
position_o	O	UINT8, UINT16, UINT32, UINT64

Process Schematic (FSM):



ED_VLD Process:

```
START
SET valid[size] = 1
SET n = 1
DO
    DO
        If valid[j] = 1
            If dataIn=table[j][n]
                If table[j][0] = n
                    dataOut = j
            else
                valid[j] = 0
        j = j + 1
    WHILE j < size
    n = n + 1
WHILE n < nBits
GOTO START
```

The matrix tokens have to be send column based.

ISO Standards using the FU: ISO/IEC 14496-16

Profiles@levels supported: All (Generic)

Input

Name	Token
dataIn_i	UINT8, UINT16, UINT32, UINT64
table_i	UINT8, UINT16, UINT32, UINT64
size_i	UINT8, UINT16, UINT32, UINT64

Output

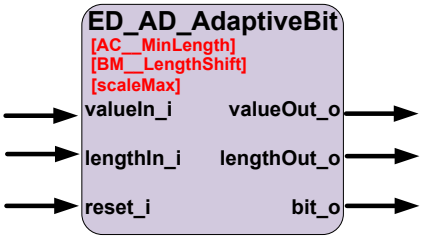
Name	Token
position_o	UINT8, UINT16, UINT32, UINT64

© ISO/IEC 2018 — All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

Parameter		
Name	**Description**	**Range**
nBits	Describes the length of the bits used for the search algorithm. This parameter is set at the network configuration level.	Type: Integer Range: [1 .. 2^5]
Package		
package org.sc29.wg11.mpeg4.part16		

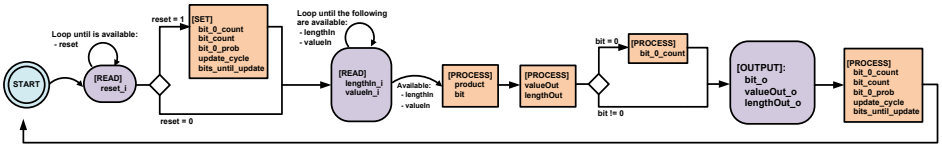
5.2.7 Alog_ED_AD_AdaptiveBit

FU Name	Algo_ED_AD_AdaptiveBit
Description	This FU describes the arithmetic decoding process based on a adaptive bit model as presented in [3,4].



Port Name	Direction (I/O)	Token RANGE
valueIn_i	I	UINT8, UINT16, UINT32, UINT64
length_i	I	UINT8, UINT16, UINT32, UINT64
reset_i	I	BOOLEAN
valueOut_o	O	UINT8, UINT16, UINT32, UINT64
lengthOut_o	O	UINT8, UINT16, UINT32, UINT64
bit_o	O	UINT8, UINT16, UINT32, UINT64

Process Schematic (FSM):



ED_AD_AdaptiveBit Process:

```
START
If reset = 1
    bit_0_count = 1;
    bit_count   = 2;
    bit_0_prob  = 1U << (BM__LengthShift - 1);
    update_cycle = bits_until_update = 4;

product = bit_0_prob * ( lengthIn >> BM_LengthShift )
bit = ( valueIn >= product )

if ( bit == 0 )
    valueOut = valueIn
    lengthOut = product
    ++bit_0_count
else
    valueOut = valueIn – product
    lengthOut = lengthIn – product

if (bit_count += update_cycle) > BM__MaxCount
    bit_count = (bit_count + 1) >> 1
    bit_0_count = (bit_0_count + 1) >> 1
    if bit_0_count = bit_count
            ++bit_count
scale = scaleMax / bit_count;
bit_0_prob = (bit_0_count * scale) >> (31 - BM__LengthShift);
update_cycle = (5 * update_cycle) >> 2;
if update_cycle > 64
    update_cycle = 64;
bits_until_update = update_cycle;
GOTO START
```

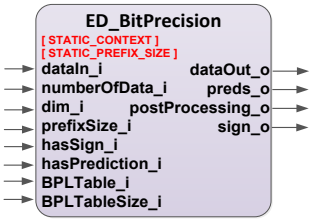
© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

ISO Standards using the FU	ISO/IEC 14496-16	
Profiles@levels supported	All (Generic)	
Input		
Name	**Token**	
valueIn_i	UINT8, UINT16, UINT32, UINT64	
length_i	UINT8, UINT16, UINT32, UINT64	
reset_i	BOOLEAN	
Output		
Name	**Token**	
valueOut_o	UINT8, UINT16, UINT32, UINT64	
lengthOut_o	UINT8, UINT16, UINT32, UINT64	
bit_o	UINT8, UINT16, UINT32, UINT64	
Parameter		
Name	**Description**	**Range**
AC_MinLength	Describes the threshold for renormalization. This parameter is set at the network configuration level.	Type:Unsigned Integer Range: [1 .. 2^{32}]
BM_LengthShift	Describes the length bits discarded before multiplication. This parameter is set at the network configuration level.	Type:Unsigned Integer Range: [1 .. 2^{5}]
scaleMax	Describes the max value to compute the scaled bit 0 probability. This parameter is set at the network configuration level.	Type:Unsigned Integer Range: [1 .. 2^{32}]
Package		
package org.sc29.wg11.mpeg4.part16		

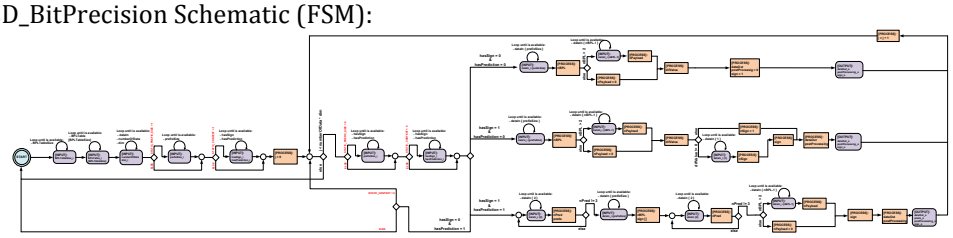
5.2.8 Algo_ED_BitPrecision

FU Name	Algo_ED_BitPrecision
Description	This FU describes the Entropy Decoding Bit Precision processes.



Port Name	Direction (I/O)	Token RANGE
dataIn_i	I	UINT_8
numerOfData_i	I	UINT8, UINT16, UINT32, UINT64
dim_i	I	UINT8, UINT16, UINT32, UINT64
prefixSize_i	I	UINT_8
hasSign_i	I	BOOLEAN
hasPrediction_i	I	BOOLEAN
BPLTable_i	I	UINT8, UINT16, UINT32, UINT64
BPLTableSize_i	I	UINT8, UINT16, UINT32, UINT64
dataOut_o	O	INT8, INT16, INT32, INT64
preds_o	O	INT8, INT16, INT32, INT64
postProcessing_o	O	BOOLEAN
sign_o	O	BOOLEAN

ED_BitPrecision Schematic (FSM):



ED_BitPrecision process:

START

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

```
INPUT: BPLTableSize_i
INPUT: BPLTable_i [ BPLTableSize ]
INPUT: numberOfData_i
INPUT: dim_i
IF STATIC_CONTEXT = 1
    INPUT: hasSign_i
    INPUT: hasPrediction_i
IF STATIC_PREFIX_SIZE = 1
    INPUT: prefixSize_i
j = 0
PROCESS_CHUNK:
WHILE j < numberOfData * dim
    IF STATIC_PREFIX_SIZE = 0
        INPUT: prefixSize_i
    IF STATIC_CONTEXT = 0
        INPUT: hasSign_i
        INPUT: hasPrediction_i
    IF hasSign = 0 & hasPrediction = 0
        INPUT: dataIn { prefixSize }
        nBPL = dataIn
        IF nBPL > 2
            INPUT: dataIn { nBPL – 1 }
            nPayLoad = dataIn
        ELSE
            nPayLoad = 0
        difValue = BPLTable [ nBPL ] + nPayLoad
        dataOut = difValue
        postProcessing = 0
        sign = 0
        OUTPUT: dataOut
        OUTPUT: postProcessing_o
        OUTPUT: sign_o
            ELSE IF hasSign = 1 & hasPrediction = 0
                INPUT: dataIn { prefixSize }
                nBPL = dataIn
                if nBPL > 2
                    INPUT: dataIn { nBPL – 1 }
                    nPayLoad = dataIn
                ELSE
                    nPayLoad = 0
                difValue = BPLTable [ nBPL ] + nPayLoad
                IF difValue != 0
                    INPUT: dataIn { 1 }
                    nSign = dataIn
                ELSE
                    nSign = 1
                sign = nSign
                postProcessing = 1
                dataOut = difValue
                OUTPUT: dataOut
                OUTPUT: postProcessing_o
                OUTPUT: sign_o
            ELSE IF hasSign = 1 & hasPrediction = 1
                signDef[2] = { 1,-1 }
                predsOut [ j ] = 0
                DO
                    INPUT: dataIn { 2 }
                    nPred = dataIn
                    preds += nPred
                    WHILE nPred != 3
                INPUT: dataIn { prefixSize }
                nBPL = dataIn
                IF ( nBPL > 2 )
                    INPUT: dataIn { nBPL – 1 }
                    nPayLoad = dataIn
                ELSE
                    nPayLoad = 0
                difValue = BPLTable [ nBPL ] + nPayLoad
                dataOut = difValue
                IF preds != 0
                    postProcessing = 1
                ELSE
                    postProcessing = 0
```

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

```
                    IF postProcessing != 0
                        sign = signDef [ difValue % 2 ]
                    ELSE
                        sign = 1
                    OUTPUT: dataOut_o
                    OUTPUT: preds_o
                    OUTPUT: postProcessing_o
                    OUTPUT: sign_o
                ELSE IF hasSign = 1 & hasPrediction = 0
                    IF STATIC_CONTEXT = 0
                        PROCESS_CHUNK
                    ELSE
                        GOTO START
    GOTO START
```

An input of size {N} has the meaning of a input of a N-size bit value.

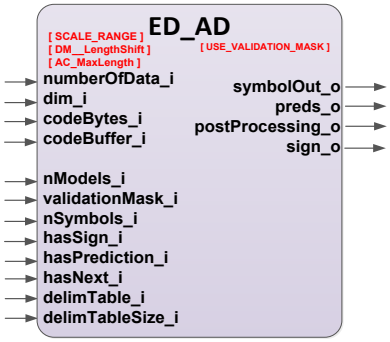
NOTE The case when both hasSign=1 and hasPrediction=0 is not considered. The behaviour in this case is to return to the START stage (if the parameter STATIC_CONTEXT=1) or to process the next data (if the parameter STATIC_CONTEXT=0).

ISO Standards using the FU	ISO/IEC 14496-16	
Profiles@levels supported	All (Generic)	
Input		
Name	**Token**	
dataIn_i	UINT_8	
numerOfData_i	UINT8, UINT16, UINT32, UINT64	
dim_i	UINT8, UINT16, UINT32, UINT64	
prefixSize_i	UINT_8	
hasSign_i	BOOLEAN	
hasPrediction_i	BOOLEAN	
BPLTable_i	UINT8, UINT16, UINT32, UINT64	
BPLTableSize_i	UINT8, UINT16, UINT32, UINT64	
Output		
Name	**Token**	
dataOut_o	INT8, INT16, INT32, INT64	
preds_o	INT8, INT16, INT32, INT64	
postProcessing_o	BOOLEAN	
sign_o	BOOLEAN	
Parameter		
Name	**Description**	**Range**
STATIC_CONTEXT	Describes whether the context data (hasSign and hasPrediction) is read for every processing iteration. If set to 1, the context data is read only once and reused during the process, if set to 0, the context data is read for every processing interation. This parameter is set at the network configuration level.	Type:Boolean Range: {0,1}
STATIC_PREFIX_ SIZE	Describes whether the prefix size value is read for every processing iteration. If set to 1, the prefix value is read only once and reused during the process, if set to 0, the prefix value is read for every processing interation. This parameter is set at the network configuration level.	Type:Boolean Range: {0,1}
Package		
package org.sc29.wg11.mpeg4.part16		

© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

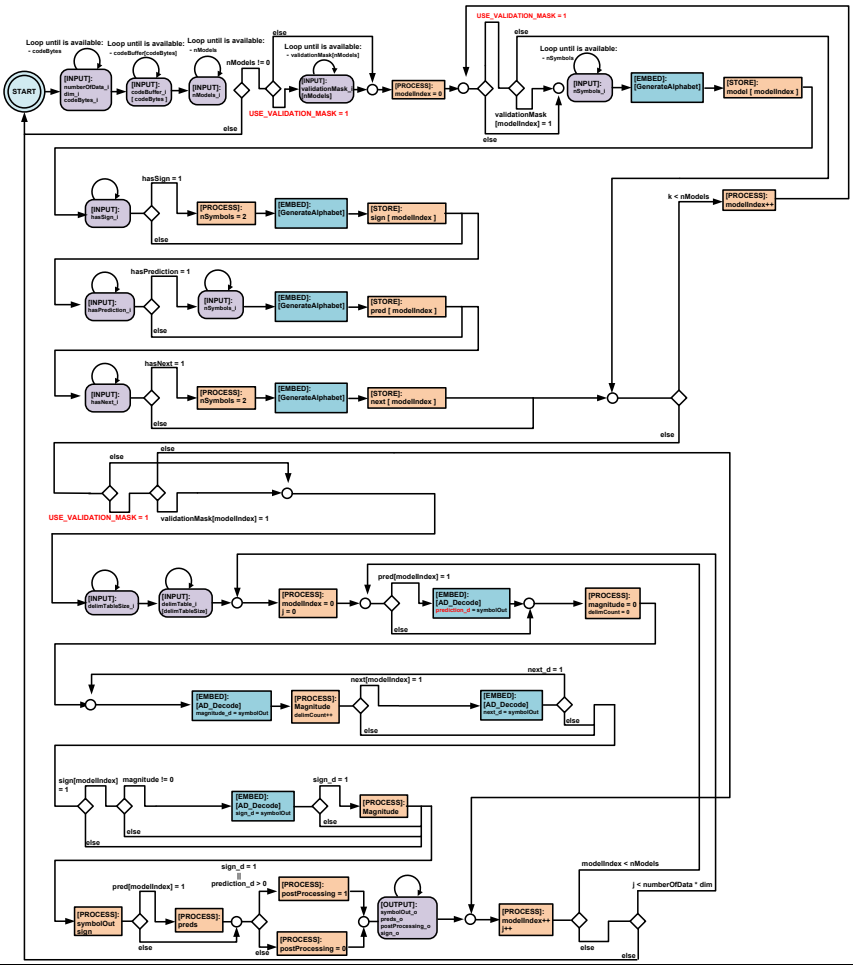
5.2.9 Algo_ED_AD

FU Name	Algo_ED_AD
Description	This FU describes the Entropy Decoding Arithmetic Decoding processes.



Port Name	Direction (I/O)	Token RANGE
numberOfData_i	I	UINT8, UINT16, UINT32, UINT64
dim_i	I	UINT8, UINT16, UINT32, UINT64
codeBytes_i	I	UINT8, UINT16, UINT32, UINT64
codeBuffer_i	I	UINT_8
nModels_i	I	UINT_8
validationMask_i	I	BOOLEAN
nSymbols_i	I	UINT8, UINT16, UINT32, UINT64
hasSign_i	I	BOOLEAN
hasPrediction_i	I	BOOLEAN
hasNext_i	I	BOOLEAN
delimTableSize_i	I	UINT_8
delimTable_i	I	UINT8, UINT16, UINT32, UINT64
symbolOut_o	O	INT8, INT16, INT32, INT64
preds_o	O	INT8, INT16, INT32, INT64
postProcessing_o	O	BOOLEAN
sign_o	O	BOOLEAN

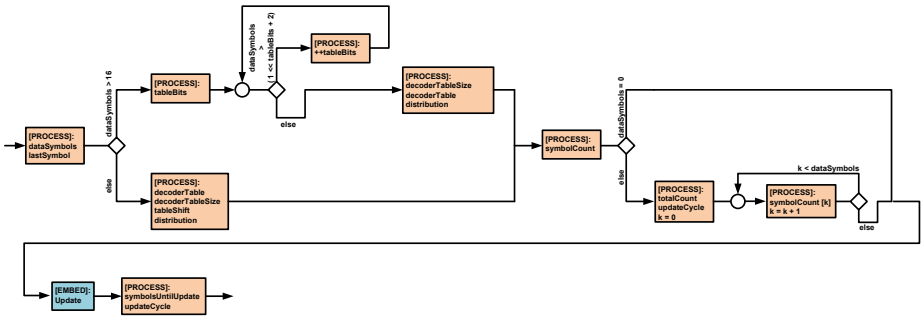
ED_AD Schematic (FSM):



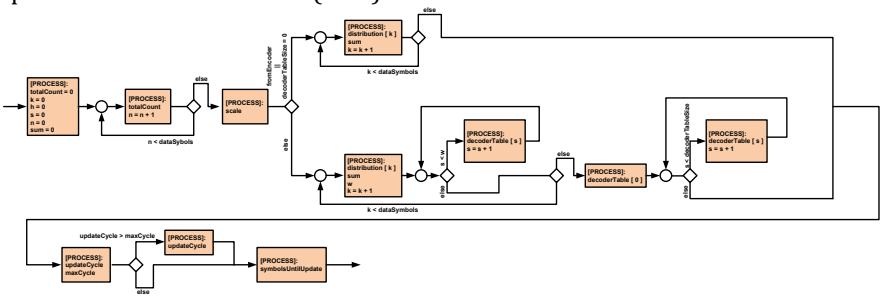
© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

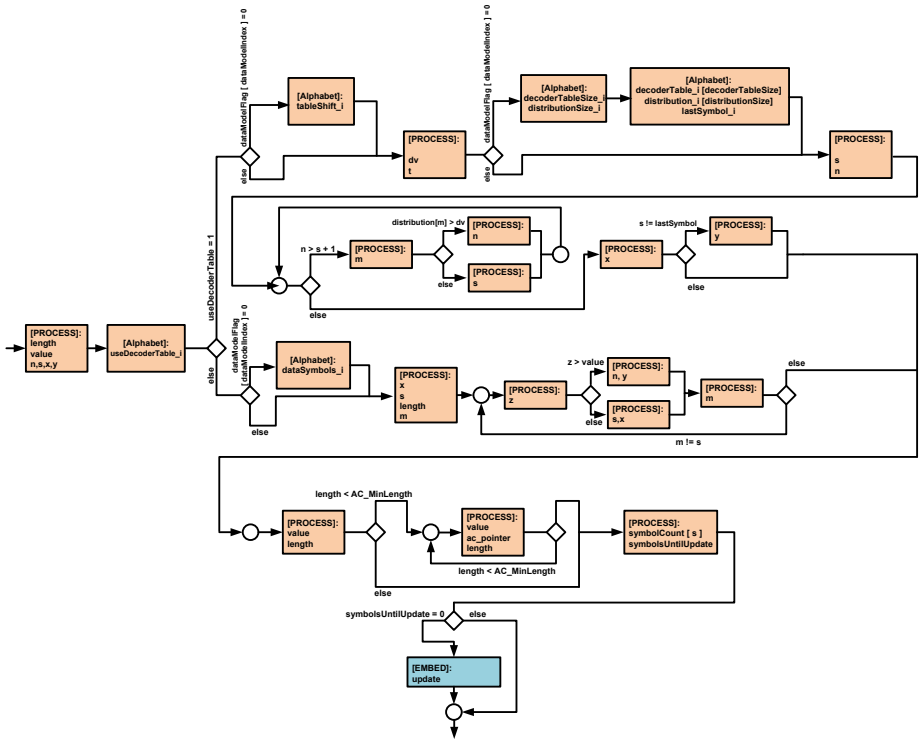
Generate Alphabet Process Schematic (FSM):



Update - Process Schematic (FSM):



Generate Alphabet Process:
Arithmetic Decode Adaptive Data - Process Schematic (FSM):



ED_AD process:

START
INPUT: numberOfData_i
INPUT: dim_i
INPUT: codeBytes_i
INPUT: codeBuffer_i[codeBytes]
INPUT: nModels_i
if nModels != 0

© ISO/IEC 2018 — All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

```
if USE_VALIDATION_MASK = 1
    INPUT: validationMask_i [ nModels ]
modelIndex = 0
k = 0
DO
  if USE_VALIDATION_MASK = 1
    if validationMask [ modelIndex ] = 1
        GOTO MODEL_CONFIG
    else
        GOTO NEXT_MODEL_CONFIG
  else
    GOTO MODEL_CONFIG

  MODEL_CONFIG:
     INPUT: nSymbols_i
     EMBED: Generate Alphabet
     STORE: model [ modelIndex ]
     INPUT: hasSign_i
     if hasSign = 1
          nSymbols = 2
          EMBED: Generate Alphabet
          STORE: sign [ modelIndex ]
     INPUT: hasPrediction_i
     if hasPrediction = 1
          INPUT: nSymbols_i
          EMBED: Generate Alphabet
          STORE: pred [ modelIndex ]
     INPUT: hasNext_i
          nSymbols = 2
          EMBED: Generate Alphabet
          STORE: next [ modelIndex ]
     k = k + 1
     WHILE k < nModels
INPUT: delimTableSize_i
INPUT: delimTable_i [ delimTableSize ]

INIT:
modelIndex = 0
j = 0
DECODE:
    if USE_VALIDATION_MASK = 1
        if validationMask [ modelIndex ] = 1
            GOTO MODEL_PROCESS
        else
            GOTO NEXT_MODEL
    else
        GOTO MODEL_PROCESS

MODEL_PROCESS:
    if pred[modelIndex] = 1
        EMBED: AD_Decode (prediction_d = symbolOut)
    magnitude = 0
    delimCount = 0
    HAS_NEXT:
        EMBED: AD_DECODE (magnitude_d = symbolOut)
        Magnitude
        delimCount++
        if next [ modelIndex ] = 1
            EMBED: AD_DECODE (next_d = symbolOut)
        if next_d = 1
            GOTO HAS_NEXT
        if sign[modelIndex] = 1
            if magnitude != 0
                EMBED: AD_DECODE (sign_d = symbolCount)
        symbolOut = magnitude
        sign = sign_d
        if pred[modelIndex] = 1
            preds = prediction_d
        if sign_d = 1 || prediction_d > 0
            postProcessing = 1
        else
            postProcessing = 0
        OUTPUT: symbolOut_o
```

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

```
                    OUTPUT: preds_o
                    OUTPUT: postProcessing_o
                    OUTPUT: sign_o

                    NEXT_MODEL:
                        modelIndex++
                        j++

                    if modelIndex < nModels
                        GOTO DECODE
                    if j < numberOfData * dim
                        GOTO INIT
GOTO START
```

Generate Alphabet Process:

```
dataSymbols = nSymbols
lastSymbol = nSymbols – 1
if dataSymbols > 16
    tableBits = 3
    WHILE dataSymbols > ( 1 << (tableBits + 2) )
        ++tableBits
    tableSize = 1 << tableBits
    tableShift = DM__LengthShift – tableBits
    distribution [ 2 * dataSymbols + tableSize + 2]
    decoderTable = distribution + 2 * dataSymbols
else
    decoderTable = 0
    tableSize = 0
    tableShift = 0
    distribution [ 2 * dataSymbols ]
symbolCount = distribution + dataSymbols
if dataSymbols != 0
totalCount = 0
updateCycle = dataSymbols
k = 0
WHILE k < dataSymbols
    symbolCount [ k ] = 1
    k = k + 1
EMBED: Update
symbolsUntilUpdate = ( dataSymbols + 6 ) >> 1
updateCycle = ( dataSymbols + 6 ) >> 1
```

Update – Process:

```
totalCount = 0
k = 0
h = 0
s = 0
n = 0
sum = 0
WHILE n < dataSymbols
    totalCount += ( symbolCount [ n ] = (symbolCount [ n ] + 1 ) >> 1 )
scale = SCALE_RANGE / totalCount
if fromEncofer = true | tableSize = 0
    WHILE k < dataSymbols
        distribution [ k ] = ( scale * sum ) >> ( 31 – DM_LengthShift )
        sum += symbolCount [ k ]
        k = k + 1
else
    WHILE k < dataSymbols
        distribution [ k ] = ( scale * sum ) >> ( 31 – DM_LengthShift )
        sum += symbolCount [ k ]
        w = distribution [ k ] >> tableShift
        WHILE  s < w
            decoderTable [++s] = k – 1
    decoderTable [ 0 ] = 0
    WHILE s <= tableSize
        decoderTable [++s] = dataSymbols – 1
updateCycle = ( 5 * updateCycle ) >> 2
maxCycle = ( dataSymbols + 6 ) << 3
if ( updateCycle > maxCycle )
    updateCycle = maxCycle
symbolsUntilUpdate = updateCycle
```


© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

Arithmetic Decode Adaptive Data Process:

```
codeBytes = 0
WHILE codeBytes <= 0
        INPUT: codeBytes
length = 0
value = 0
n = 0
s = 0
x = 0
y = 0
if useDecoderTable = 1
        if dataModelFlag [ dataModelIndex ] = 0
                INPUT: tableShift
        dv  = value / ( length >>= DM__LengthShift )
        t = dv >> tableShift
        s = decoderTable [ t ]
        n = decoderTable [ t+1 ] + 1
        WHILE n > s + 1
                m = ( s + n ) >> 1
                if distribution [ m ] > dv
                        n = m
                else
                        s = m
        x = distribution [ s ] * length
        if s != lastSymbol
                y = distribution [ s + 1] * length
else
        x = 0
        s = 0
        length >>= DM_LengthShift
        m = ( n = dataSymbols ) >> 1
        DO
                z = length * distribution [ m ]
                if z > value
                        n = m
                        y = z
                else
                        s = m
                        x = z
                WHILE m = ((s+n) >> 1 ) != s
value -= x
length = y – x
if length < AC__MinLength
DO
        value = (value << 8) | ++ac_pointer
        WHILE (length <<= 8) < AC__MinLength
++symbolCount [ s ]
if  -- symbolsUntilUpdate = 0
        EMBED: Update
```

ISO Standards using the FU	ISO/IEC 14496-16
Profiles@levels supported	All (Generic)
Input	
Name	**Token**
numberOfData_i	UINT8, UINT16, UINT32, UINT64
dim_i	UINT8, UINT16, UINT32, UINT64
codeBytes_i	UINT8, UINT16, UINT32, UINT64
codeBuffer_i	UINT_8
nModels_i	UINT_8
validationMask_i	BOOLEAN
nSymbols_i	UINT8, UINT16, UINT32, UINT64
hasSign_i	BOOLEAN
hasPrediction_i	BOOLEAN
hasNext_i	BOOLEAN
delimTableSize_i	UINT_8

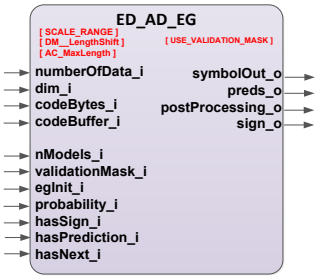
© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018

delimTable_i	UINT8, UINT16, UINT32, UINT64	
Output		
Name	**Token**	
symbolOut_o	INT8, INT16, INT32, INT64	
preds_o	INT8, INT16, INT32, INT64	
postProcessing_o	BOOLEAN	
sign_o	BOOLEAN	
Parameter		
Name	**Description**	**Type / Range**
AC__MaxLength	Describes the maximum AC interval. This parameter is set at the network configuration level.	Type:Unsigned Integer Range: [1 .. 2^{32}]
DM__LengthShift	Describes the number of bits discarded before multiplication. This parameter is set at the network configuration level.	Type:Unsigned Integer Range: [1 .. 2^{32}]
SCALE_RANGE	Describes the range for the scale value. This parameter is set at the network configuration level.	Type:Unsigned Integer Range: [1 .. 2^{32}]
USE_VALIDATION_MASK	Indicates whether to use the validation mask input port or not.(0 – NO, 1 – YES)	Type:Unsigned Integer Range: {0,1}
Package		
package org.sc29.wg11.mpeg4.part16		

5.2.10 Algo_ED_AD_EG

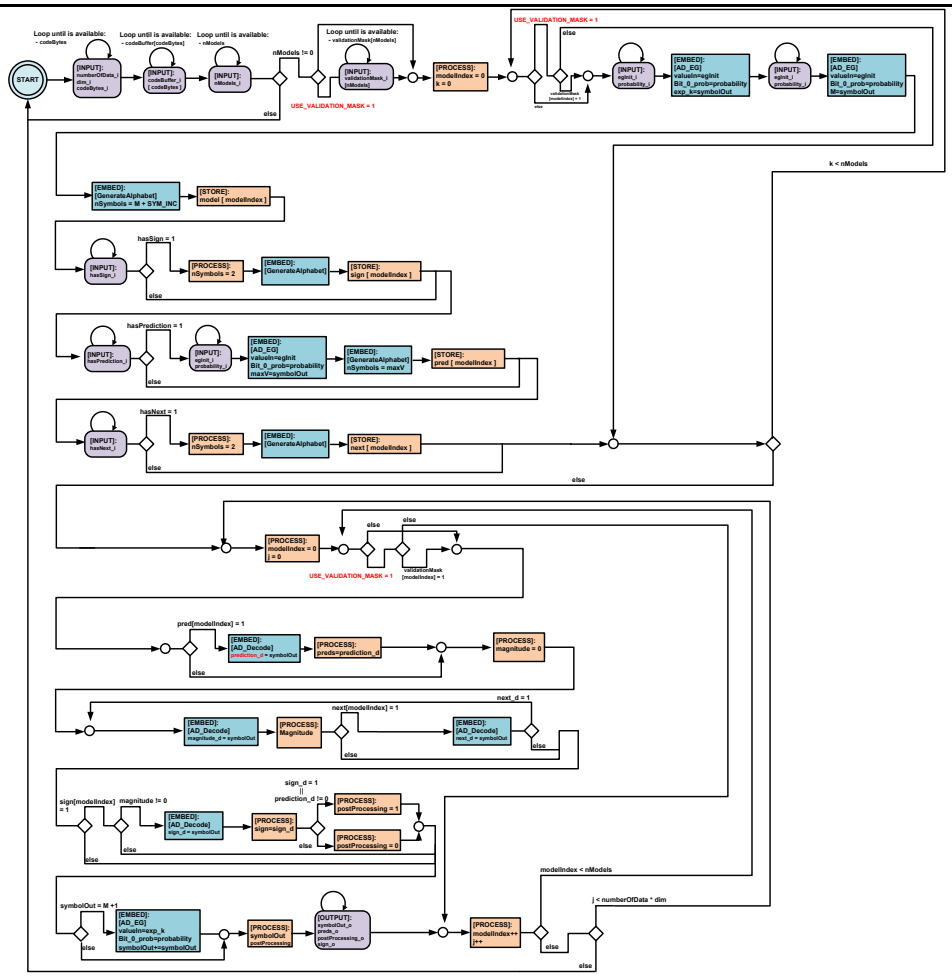
FU Name	Algo_ED_AD_EG
Description	This FU describes the Entropy Decoding Arithmetic Decoding Exponential Golomb processes.



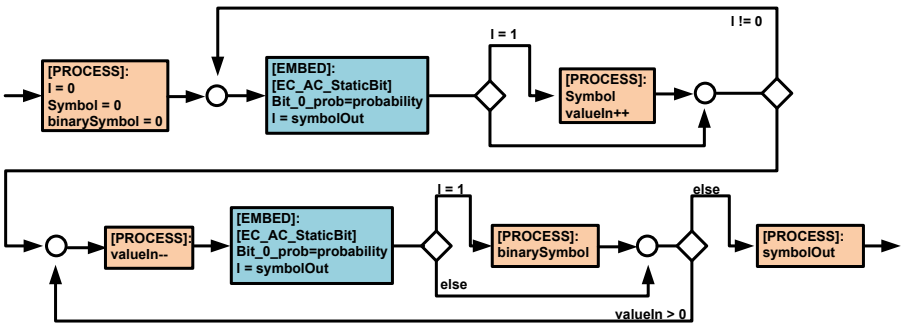
Port Name	Direction (I/O)	Token RANGE
numberOfData_i	I	UINT8, UINT16, UINT32, UINT64
dim_i	I	UINT8, UINT16, UINT32, UINT64
codeBytes_i	I	UINT8, UINT16, UINT32, UINT64
codeBuffer_i	I	UINT_8
nModels_i	I	UINT_8
validationMask_i	I	BOOLEAN
egInit_i	I	UINT8, UINT16, UINT32, UINT64
probability_i	I	UINT8, UINT16, UINT32, UINT64
hasSign_i	I	BOOLEAN
hasPrediction_i	I	BOOLEAN
hasNext_i	I	BOOLEAN
symbolOut_o	O	INT8, INT16, INT32, INT64
preds_o	O	INT8, INT16, INT32, INT64
postProcessing_o	O	BOOLEAN
sign_o	O	BOOLEAN

ED_AD_EG Schematic (FSM):

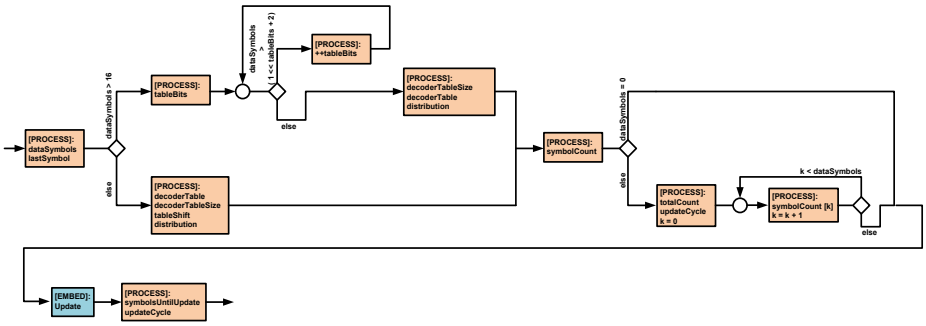
© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018



ED_AD_EG Process Schematic (FSM):



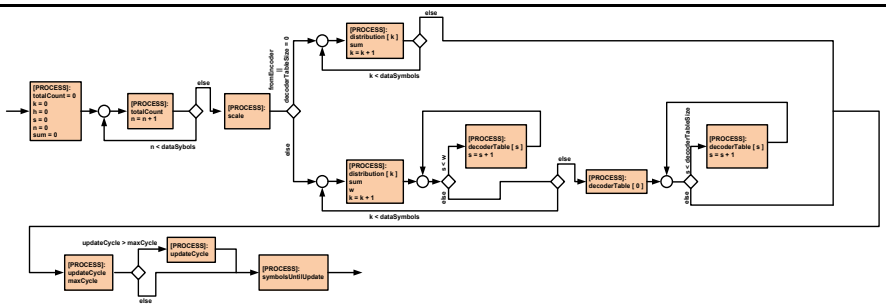
Generate Alphabet Process Schematic (FSM):



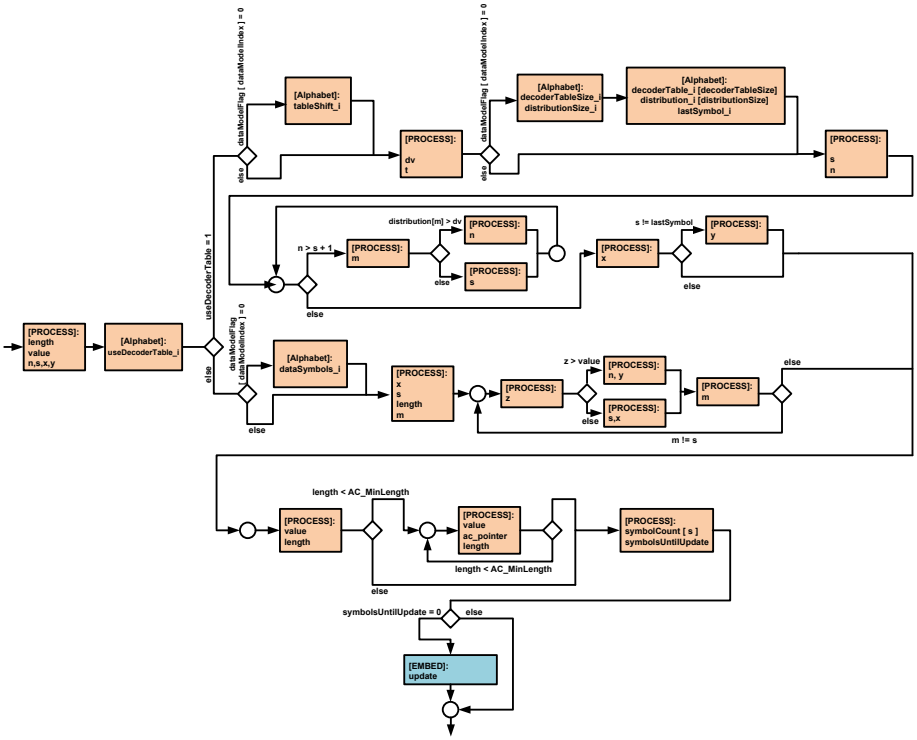
Update - Process Schematic (FSM):

© ISO/IEC 2018 — All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018



Arithmetic Decode Adaptive Data - Process Schematic (FSM):



ED_AD_EG process:

```
START
INPUT: numberOfData_i
INPUT: dim_i
INPUT: codeBytes_i
INPUT: delimTableSize_i
INPUT: codeBuffer_i[codeBytes]
INPUT: delimTable_i[delimTableSize]
INPUT: nModels_i
if USE_VALIDATION_MASK = 1
    INPUT: validationMask[nModels]
if nModels != 0
    modelIndex = 0
    k = 0
     INPUT: egInit_i
     INPUT: probability_i
     EMBED: AD_EG (valueIn=egInit, Bit_0_prob=probability,exp_k=symbolOut)
     INPUT: egInit_i
     INPUT: probability_i
     EMBED: AD_EG (valueIn=egInit, Bit_0_prob=probability,exp_k=symbolOut)
    DO
        if USE_VALIDATION_MASK = 1
            if validationMask [nModels] = 1
                EMBED: Generate Alphabet (nSymbols=M+SYM_INC)
                STORE: model [ modelIndex ]
                INPUT: hasSign_i
                if hasSign = 1
                    nSymbols = 2
                    EMBED: Generate Alphabet
                    STORE: sign [ modelIndex ]
```


© ISO/IEC 2018 — All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23002-4-2018