**COMMITTEE DRAFT****© ISO/IEC 2020 – All rights reserved****Text of ISO/IEC CD 23090-15** **63****Part 15: Conformance testing for Versatile Video Coding****Information technology — Coded representation of immersive media****Élément introductif — Élément central — Partie 2: Titre de la partie****Information technology — Coded representation of immersive media — Part 15: Conformance testing for Versatile Video Coding****E****2020-10-16****(30) Committee****ISO/IEC****ISO/IEC J****202x****International Standard****202x****0009****ISO/IEC 23090‑****ISO/IEC 23090‑15****ISO/IEC 23090-15:202x****JISC****Coding of audio, picture, multimedia and hypermedia information****Information technology****5****29****1** **2****見出し 2****見出し 1****0****2****STD Version 2.1c2****30** **4** **ISO/IEC JTC 1/SC 29 /WG 5 N 0009**

Date: **2020-10-16**

**Text of ISO/IEC CD 23090-15**

ISO/IEC JTC 1/SC 29/WG 5

Secretariat:  JISC

**Information technology — Coded representation of immersive media — Part 15: Conformance testing for Versatile Video Coding**

*Élément introductif — Élément central — Partie 2: Titre de la partie*

|  |
| --- |
| **Warning**  This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.  Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation |

**Copyright notice**

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

[Indicate the full address, telephone number, fax number, telex number, and electronic mail address, as appropriate, of the Copyright Manger of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the working document has been prepared.]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

**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.

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](http://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](http://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 of 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 [www.iso.org/iso/foreword.html](http://www.iso.org/iso/foreword.html).

This document was prepared by Technical Committee ISO/IEC JTC1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*, in collaboration with ITU-T. The technically identical text is published as ITU-T H.266.1 (xx/202x).

A list of all parts in the ISO/IEC 23090 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](http://www.iso.org/members.html).

Information technology — Coded representation of immersive media — Part 15: Conformance testing for Versatile Video Coding

Editor’s note:

Due to the size of the electronic attachments, the associated conformance test set data files are provided separately, at the following location:

<http://wftp3.itu.int/av-arch/jvet-site/bitstream_exchange/draft_conformance/draft5>

These are also available by ftp at:

<ftp://ftp3.itu.int/jvet-site/bitstream_exchange/> draft\_conformance/draft5

(user id: avguest, password: Avguest201007)

For ftp access, it is suggested to use FileZilla, for which the site manager feature should be used and ftp encryption should be set to “Require implicit ftp over TLS”.

# Scope

This Recommendation | International Standard[[1]](#footnote-2) specifies a set of tests and procedures designed to indicate whether encoders or decoders meet the normative requirements specified in Rec. ITU‑T H.266 | ISO/IEC 23090-3.

# Normative references

## General

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

## Identical Recommendations | International Standards

– None.

## Paired Recommendations | International Standards equivalent in technical content

– Recommendation ITU-T H.266 (in force), *Versatile video coding*.

– ISO/IEC 23090-3: in force, *Information technology – Coded representation of immersive media– Part 3: Versatile video coding*.

– Recommendation ITU-T H.266.2 (in force), *Reference software for ITU-T H.266* *Versatile video coding*.

– ISO/IEC 23090-16: in force, *Information technology – Coded representation of immersive media – Part 16: Reference software for versatile video coding*.

# Definitions

For the purposes of this Recommendation | International Standard, the terms, definitions, abbreviations and symbols specified in Rec. ITU-T H.266 | ISO/IEC 23090-3 (particularly in clause 3) apply. The following terms are further clarified for purposes herein as follows.

**3.1 bitstream**: A Rec. ITU-T H.266 | ISO/IEC 23090-3 video bitstream.

**3.2 decoder**: A Rec. ITU-T H.266 | ISO/IEC 23090-3 video decoder, i.e., an embodiment of the decoding process specified by Rec. ITU-T H.266 | ISO/IEC 23090-3. The decoder does not include the display process, which is outside the scope of this Recommendation | International Standard.

**3.3 encoder**: An embodiment of a process, not specified in this Recommendation | International Standard (except in regard to identification of the reference software encoder), that produces a bitstream.

**3.4 reference software decoder**: The software decoder provided in Rec. ITU-T H.266.2 | ISO/IEC 23090-16.

**3.5 reference software encoder**: The software encoder provided in Rec. ITU-T H.266.2 | ISO/IEC 23090-16.

# Abbreviations and acronyms

For the purposes of this Recommendation | International Standard, relevant abbreviations and acronyms specified in clause 4 of Rec. ITU-T H.266 | ISO/IEC 23090-3 and the following abbreviations apply.

**AMVP**: Adaptive motion vector prediction

**CCLM**: Cross-component linear model

**CIIP**: Combined inter/intra prediction

**CST**: Chroma separate tree

**CTC**: Common test conditions

**DCT**: Discrete cosine transform

**DMVR**: Decoder-side motion vector refinement

**DQ**: Dependent quantization

**DST**: Discrete sine transform

**FTP**: File transfer protocol

**ISP**: Intra subblock partitioning

**JCCR**: Joint coding of chroma residuals

**MMVD**: Merge with MVD

**MPM**: Most probable mode

**MVD**: Motion vector difference.

**PDPC**: Position-dependent (intra) prediction combination.

**RPR**: Reference picture resampling

**SAD:** Sum of Absolute Differences

**SBT**: Subblock transform

**SbTMVP**: Subblock based temporal motion vector prediction.

**SMVD**: Symmetric MVD

**TMVP**: Temporal motion vector prediction.

# Conventions

For the purposes of this Recommendation | International Standard, relevant conventions are specified in clause 5 of Rec. ITU-T H.266 | ISO/IEC 23090-3.

# Conformance testing for ITU-T H.266 | ISO/IEC 23090-3

## Introduction

The following clauses specify normative tests for verifying conformance of video bitstreams as well as decoders. Those normative tests make use of test data (bitstream test suites) provided as an electronic annex to this Recommendation | International Standard and the reference software decoder specified in Rec. ITU-T H.266.2 | ISO/IEC 23090-16.

## Bitstream conformance

Bitstream conformance for Rec. ITU-T H.266 | ISO/IEC 23090-3 is specified by clause C.4 of Rec. ITU‑T H.266 | ISO/IEC 23090-3.

## Decoder conformance

Decoder conformance for Rec. ITU-T H.266 | ISO/IEC 23090-3 is specified by clause C.5 of Rec. ITU‑T H.266 | ISO/IEC 23090-3.

## Procedure to test bitstreams

A bitstream that claims conformance with Rec. ITU-T H.266 | ISO/IEC 23090-3 shall pass the following normative test.

The bitstream shall be decoded by processing it with the reference software decoder. When processed by the reference software decoder, the bitstream shall not cause any error or non-conformance messages to be reported by the reference software decoder. This test should not be applied to bitstreams that are known to contain errors introduced by transmission, as such errors are highly likely to result in bitstreams that lack conformance to Rec. ITU-T H.266 | ISO/IEC 23090-3. When a bitstream containing decoded picture hash SEI messages is processed by the reference software decoder, the reference software decoder’s hash value for each output picture shall match the hash value signalled in the corresponding decoded picture hash SEI message.

Successfully passing the reference software decoder test provides only a strong presumption that the bitstream under test is conforming to the video layer, i.e., that it does indeed meet all the requirements for the video layer (except Annexes C, D and E) specified in Rec. ITU-T H.266 | ISO/IEC 23090-3 that are tested by the reference software decoder.

Additional tests may be necessary to more thoroughly check that the bitstream properly meets all the requirements specified in Rec. ITU-T H.266 | ISO/IEC 23090-3 including the hypothetical reference decoder (HRD) conformance (based on Annexes C, D and E). These complementary tests may be performed using other video bitstream verifiers that perform more complete tests than those implemented by the reference software decoder.

Rec. ITU-T H.266 | ISO/IEC 23090-3 contains several informative recommendations that are not an integral part of that Recommendation | International Standard. When testing a bitstream for conformance, it may also be useful to test whether or not the bitstream follows those recommendations.

To check correctness of a bitstream, it is necessary to parse the entire bitstream and to extract all the syntax elements and other values derived from those syntactic elements and used by the decoding process specified in Rec. ITU‑T H.266 | ISO/IEC 23090-3.

A verifier may not necessarily perform all stages of the decoding process specified in Rec. ITU-T H.266 | ISO/IEC 23090-3 in order to verify bitstream correctness. Many tests can be performed on syntax elements in a state prior to their use in some processing stages.

## Procedure to test decoder conformance

### Conformance bitstreams

A bitstream has values of general\_profile\_idc, general\_tier\_flag, and general\_level\_idc corresponding to a set of specified constraints on a bitstream for which a decoder conforming to a specified profile, tier, and level is required in Annex A of Rec. ITU-T H.266 | ISO/IEC 23090-3 to properly perform the decoding process.

### Contents of the bitstream file

The conformance bitstreams are included in this Recommendation | International Standard as an electronic attachment. The following information is included in a single zipped file for each such bitstream.

* \*.bit – bitstream (mandatory)
* \*.txt – description (mandatory)
* \*.yuv.md5 – MD5 check sum of the complete decoded yuv file (mandatory)
* \*.md5 – MD5sum of the bitstream file (mandatory)
* \*.opl – output picture log (mandatory)
* \*.cfg – config file used to generate bitstream with VTM encoder software (optional, not applicable if VTM encoder release version not used)

### Requirements on output of the decoding process and timing

Two classes of decoder conformance are specified:

– output order conformance; and

– output timing conformance.

The output of the decoding process is specified in clause 8 and Annex C of Rec. ITU-T H.266 | ISO/IEC 23090-3.

For output order conformance, it is a requirement that all of the decoded pictures specified for output in Annex C of Rec. ITU-T H.266 | ISO/IEC 23090-3 shall be output by a conforming decoder in the specified order and that the values of the decoded samples in all of the pictures that are output shall be (exactly equal to) the values specified in clause 8 of Rec. ITU-T H.266 | ISO/IEC 23090-3.

For output timing conformance, it is a requirement that a conforming decoder shall also output the decoded samples at the rates and times specified in Annex C of Rec. ITU-T H.266 | ISO/IEC 23090-3.

The display process, which ordinarily follows the output of the decoding process, is outside the scope of this Recommendation | International Standard.

### Recommendations (informative)

This clause does not form an integral part of this Recommendation | International Standard.

In addition to the requirements, it is desirable that conforming decoders implement various informative recommendations specified in Rec. ITU-T H.266 | ISO/IEC 23090-3 that are not an integral part of that Recommendation | International Standard. This clause discusses some of these recommendations.

It is recommended that a conforming decoder be able to resume the decoding process as soon as possible after the loss or corruption of part of a bitstream.

### Static tests for output order conformance

Static tests of a video decoder require testing of the decoded samples, and can be accomplished when the decoded samples at the output of the decoding process are available. It may not be possible to perform this type of test with a production decoder (due to the lack of an appropriate accessible interface in the design at which to perform the test). In that case this test should be performed by the manufacturer during the design and development phase. Static tests are used for testing the decoding process. The test will check that the values of the samples decoded by the decoder under test shall be identical to the values of the samples decoded by the reference decoder. A hash operation performed on the values of the samples of the decoded pictures produced by the decoder under test shall produce the same results as the hash values provided in the .opl files associated with the bitstream.

### 6.5.6 Dynamic tests for output timing conformance

Dynamic tests are applied to check that all the decoded samples are output and that the timing of the output of the decoder's decoded samples conforms to the specification of clause 8 and Annex C of Rec. ITU‑T H.266 | ISO/IEC 23090-3, and to verify that the HRD models (as specified by the CPB and DPB specification in Annex C of Rec. ITU-T H.266 | ISO/IEC 23090-3) are not violated when the bits of the bitstream are delivered at the proper rate.

The dynamic test is often easier to perform on a complete decoding system, which may include a systems decoder, a video decoder and a display process. It may be possible to record the output of the display process and to check that display order and timing of decoded pictures are correct at the output of the display process. However, since the display process is not within the normative scope of Rec. ITU-T H.266 | ISO/IEC 23090-3, there may be cases where the output of the display process differs in timing or value even though the video decoder is conforming. In this case, the output of the video decoder itself (before the display process) would need to be captured in order to perform the dynamic tests on the video decoder. In particular the output order and timing of the decoded pictures shall be correct.

If buffering period and picture timing SEI messages are included in the test bitstream, HRD conformance shall be verified using the values of nal\_initial\_cpb\_removal\_delay, nal\_initial\_cpb\_removal\_offset, au\_cpb\_removal\_delay\_minus1 and pic\_dpb\_output\_delay that are included in the bitstream.

If buffering period and picture timing SEI messages are not included in the bitstream, the following inferences shall be made to generate the missing parameters:

– fixed\_pic\_rate\_general\_flag[ i ] shall be inferred to be equal to 1.

– low\_delay\_hrd\_flag[ i ] shall be inferred to be equal to 0.

– cbr\_flag[ subLayerId ][ j ] shall be inferred to be equal to 0.

– The frame rate of the bitstream shall be inferred to be equal to the frame rate value specified in the .txt file for the bitstream. If this is missing, then a frame rate of either 25 or 30000 ÷ 1001 can be inferred.

– The bit rate of the bitstream shall be inferred to be equal to the maximum value for the level specified in Table 136 in Rec. ITU-T H.266 | ISO/IEC 23090-3.

– CPB and DPB sizes shall be inferred to be equal to the maximum value for the level specified in Table 135 in Rec. ITU‑T H.266 | ISO/IEC 23090-3.

With the above inferences, the HRD shall be operated as follows.

– The CPB is filled starting at time t = 0, until it is full, before removal of the first access unit. This means that thebp\_nal\_initial\_cpb\_removal\_delay[ i ][ j ] shall be inferred to be equal to the total CPB buffer size divided by the bit rate divided by 90000 (rounded downwards) and bp\_vcl\_initial\_cpb\_removal\_offset[ i ][ j ] shall be inferred to be equal to zero.

– The first access unit is removed at time t = bp\_nal\_initial\_cpb\_removal\_delay[ i ][ j ] ÷ 90000 and subsequent access units are removed at intervals based on the picture distance

– Using these inferences, the CPB will not overflow or underflow and the DPB will not overflow.

### 6.5.7 Decoder conformance test of a particular profile, tier, and level

In order for a decoder of a particular profile, tier, and level to claim output order conformance to Rec. ITU‑T H.266 | ISO/IEC 23090-3 as specified by this Recommendation | International Standard, the decoder shall successfully pass the static test specified in clause 6.5.5 with all the bitstreams of the normative test suite specified for testing decoders of this particular profile, tier, and level combination.

In order for a decoder of a particular profile, tier, and level to claim output timing conformance to Rec. ITU‑T H.266 | ISO/IEC 23090-3 as specified by this Recommendation | International Standard, the decoder shall successfully pass both the static test specified in clause 6.5.5 and the dynamic test specified in clause 6.5.6 with all the bitstreams of the normative test suite specified for testing decoders of this particular profile, tier, and level. Tables 1 through 5 specify the normative test suites. The profile, tier, and level combination are described in the tables or in the .txt file associated with the bitstream.

## Specification of the test bitstreams

### General

Some characteristics of each bitstream listed in Table 1 are specified in this clause. In Table 1, the value "29.97" shall be interpreted as an approximation of an exact value of 30000 ÷ 1001 and the value "59.94" shall be interpreted as an approximation of an exact value of 60000 ÷ 1001.

### Test bitstreams – Coding tools for Main 10 profile with 4:2:0 chroma format and 10 bit depth

#### Chroma separate tree (CST)

##### Test bitstream CST\_A\_MediaTek

**Specification**: All pictures are coded in I slices with CST enabled. CST is tested with all possible luma and chroma block sizes, and luma-chroma block size combinations (e.g., luma block size is larger than, equal to, or smaller than the corresponding chroma block size).

**Functional stage**: Reconstruction process

**Purpose**: Check that the decoder can properly decode slices with CST enabled.

#### Dependent quantization (DQ)

##### Test bitstream DQ\_A\_HHI

**Specification**: The bitstream consists of three coded video sequences (CVSs), with the following properties:

* The first CVS uses dependent quantization for all pictures, all non-related features (inter tools, ALF, ...) are disabled, and MTS and LFNST are disabled.
* The second CVS uses dependent quantization for all pictures, all non-related features (inter tools, ALF, ...) are disabled, and MTS (for intra) and LFNST are enabled.
* The third CVS exercises a picture-level selection between dependent quantization, sign data hiding, and standard quantization, all non-related features (inter tools, ALF, ...) are disabled, and MTS (for intra) and LFNST are enabled.

**Functional stage**: Dependent quantization

**Purpose**: Check that the decoder can properly decode slices with DQ enabled.

##### Test bitstream DQ\_B\_HHI

**Specification**: The bitstream consists of three coded video sequences (CVSs) of resolution 1920 x 1080, with the following properties:

* The first CVS uses dependent quantization for all pictures, all non-related features (inter tools, ALF, ...) are disabled, and MTS and LFNST are disabled.
* The second CVS uses dependent quantization for all pictures, all non-related features (inter tools, ALF, ...) are disabled, and MTS (for intra) and LFNST are enabled.
* The third CVS exercises a picture-level selection between dependent quantization, sign data hiding, and standard quantization, all non-related features (inter tools, ALF, ...) are disabled, and MTS (for intra) and LFNST are enabled.

**Functional stage**: Dependent quantization

**Purpose**: Check that the decoder can properly decode slices with DQ enabled.

#### Cross-component linear model (CCLM)

##### Test bitstream CCLM\_A\_KDDI

**Specification**: The bitstream exercises corner cases for coding structures using CCLM with the following properties:

* POC0: Chroma CU size is 64x64.
* POC1: First split of CU is horizontal, i.e. CU size is 64x32.
* POC2: First split of CU is quad, i.e. CU size is 32x32.
* POC3: First and second split of CU are horizontal and vertical, respectively.
* POC4: First split of CU is vertical or ternary, i.e. none of condition is satisfied for CCLM.
* POC5: CU size is 64x64 and ISP is enabled.
* POC6: First luma split is something else than quad.

**Functional stage**: Intra prediction

**Purpose**: Check that the decoder can properly decode slices with CCLM enabled.

#### Multiple transform set (MTS)

##### Test bitstream MTS\_A\_LGE

**Specification**: The bitstream exercises the following transform features:

* 1st part
  + Explicit intra MTS on and explicit inter MTS off with low frequency non-separatble transform (LFNST) disabled
  + Include all test cases for ISP, MIP, Luma separate tree, and Chroma separate tree
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all possible block sizes and partitions where all MTS combinations can happen
* 2nd part
  + Implicit MTS on and explicit inter MTS off with LFNST disabled
  + Include all test cases for ISP, MIP, Luma separate tree, and Chroma separate tree
  + Include all possible block sizes and partitions (especially for ISP) for all allowable MTS combinations

**Functional stage**: Transform

**Purpose**: Check that the decoder can properly decode slices with MTS enabled.

##### Test bitstream MTS\_B\_LGE

**Specification**: The bitstream exercises the following transform features:

* 1st part
  + Explicit intra MTS on and explicit inter MTS off with LFNST disabled
  + Include all test cases for SBT, single tree and TU-tiling based on maximum transform size (64)
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all possible block sizes and partitions (especially for SBT) where all MTS combinations can happen
* 2nd part
  + Implicit intra MTS on and explicit inter MTS off with LFNST disabled
  + Include all test cases for SBT, single tree, and TU-tiling based on maximum transform size (64)
  + Include all possible block sizes and partitions (especially for SBT) where all MTS combinations can happen
* 3rd part
  + Implicit MTS on and explicit inter MTS off with LFNST disabled
  + Include all test cases for SBT and single tree
  + Include all possible block sizes and partitions (especially for SBT) where all MTS combinations can happen
* 4th part
  + Explicit intra MTS on and explicit inter MTS on with LFNST disabled
  + Include all test cases for SBT and single tree
  + Include all possible block sizes and partitions (especially for SBT) where all MTS combinations can happen

**Functional stage**: Transform

**Purpose**: Check that the decoder can properly decode slices with MTS enabled.

#### Adaptive loop filter (ALF)

##### Test bitstream ALF\_A\_Huawei

**Specification**: This bitstream uses both ALF and virtual boundary, as follows:

* Applies ALF virtual boundary (VB) at non-CTC CTU sizes (CTU size of 64 is used)
* Positions Luma VB at 4 lines (Pos : 60) and Chroma VB at 2 lines (Pos : 62) above the CTU height.

**Functional stage**: Adaptive loop filter

**Purpose**: Check that the decoder can properly decode slices with ALF enabled.

##### Test bitstream ALF\_B\_Huawei

**Specification**: This bitstream uses both ALF and virtual boundary, as follows:

* Applies ALF virtual boundary (VB) to sequences whose picture height is 1 CTU (CTU size of 128 is used as per CTC)
* Positions Luma VB at 4 lines (Pos : 124) and Chroma VB at 2 lines (Pos : 62) above the CTU height.

**Functional stage**: Adaptive loop filter

**Purpose**: Check that the decoder can properly decode slices with ALF enabled.

##### Test bitstream ALF\_C\_KDDI

**Specification**: Bitstream exercises clipping values of non-linear ALF.

**Functional stage**: Adaptive loop filter

**Purpose**: Check that the decoder can properly decode slices with ALF enabled.

##### Test bitstream ALF\_A\_Qualcomm

**Specification**: Bitstream uses multiple ALF APSs with LMCS enabled.

**Functional stage**: Adaptive loop filter

**Purpose**: Check that the decoder can properly decode slices with ALF enabled.

#### Affine motion model (AFF)

##### Test bitstream AFF\_A\_HUAWEI

**Specification**: The bitstream enables 6-parameter affine mode by SPS flag. All allowed blocks sizes of Affine merge mode are exercised multiple times. All allowed blocks sizes of Affine AMVP mode, including 4-parameter and 6-parameter Affine mode, are exercised multiple times. All allowed candidates for Affine merge mode, including two inherited candidates, four 6-parameter constructed candidates, two 4-parameter constructed candidates, and zero padding candidate are exercised multiple times. Inheritance of affine model from top CTU are exercised multiple times. Fallback mode for affine memory bandwidth restriction is triggered multiple times.

**Functional stage**: Affine mode inter prediction

**Purpose**: Check that the decoder can properly decode slices with affine mode enabled.

##### Test bitstream AFF\_B\_HUAWEI

**Specification**: The bitstream uses affine mode, with 6-parameter affine mode disabled by SPS flag. All allowed blocks sizes of Affine merge mode are exercised multiple times. All allowed blocks sizes of 4-parameter Affine AMVP mode are exercised multiple times. All allowed candidates for Affine merge mode, including two inherited candidates, two 4-parameter constructed candidates, and zero padding candidate are exercised multiple times. Inheritance of affine model from top CTU are exercised multiple times. Fallback mode for affine memory bandwidth restriction is triggered multiple times. All allowed blocks sizes of Affine merge mode are exercised multiple times. All allowed blocks sizes of 4-parameter Affine AMVP mode are exercised multiple times. All allowed candidates for Affine merge mode, including two inherited candidates, two 4-parameter constructed candidates, and zero padding candidate are exercised multiple times. Inheritance of affine model from top CTU are exercised multiple times. Fallback mode for affine memory bandwidth restriction is triggered multiple times.

**Functional stage**: Affine mode inter prediction

**Purpose**: Check that the decoder can properly decode slices with affine mode enabled.

#### Subblock-based temporal merging candidates (SbTMVP)

##### Test bitstream SbTMVP\_A\_Bytedance

**Specification**: The bitstream uses SbTMVP when affine is disabled.

**Functional stage**: Inter prediction process

**Purpose**: Check that the decoder can properly decode PUs with SbTMVP on and affine off.

##### Test bitstream SbTMVP\_B\_Bytedance

**Specification**: This bitstream disables SbTMVP.

**Functional stage**: Inter prediction process

**Purpose**: Check that the decoder can properly decode PUs with SbTMVP off.

#### Adaptive motion vector resolution (AMVR)

##### Test bitstream AMVR\_A\_HHI

**Specification**: The bitstream exercises translational and affine AMVR with different settings. It represents a concatenation of 5 coded video sequences (CVSs) with the following properties:

* The first CVS exercises translational AMVR with amvr\_precision\_idx equal to 1 (i.e., 1 luma sample motion vector resolution).
* The second CVS exercises translational AMVR with amvr\_precision\_idx equal to 2 (i.e., 4 luma samples motion vector resolution).
* The third CVS exercises translational AMVR with amvr\_precision\_idx equal to 0 (i.e., 1/2 luma sample motion vector resolution). This implies application of the Switchable Interpolation Filter (SIF).
* The fourth CVS exercises affine AMVR with amvr\_precision\_idx equal to 0 (i.e., 1/16 luma sample motion vector resolution).
* The fifth CVS exercises affine AMVR with amvr\_precision\_idx equal to 1 (i.e., 1 luma sample motion vector resolution).

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstreams with AMVR enabled.

##### Test bitstream AMVR\_B\_HHI

**Specification**: The bitstream exercises AMVR. It cycles frame-by-frame between the following variants:

* Translational AMVR with amvr\_precision\_idx equal to 1 (i.e., 1 luma sample motion vector resolution)
* Translational AMVR with amvr\_precision\_idx equal to 2 (i.e., 4 luma samples motion vector resolution)
* Translational AMVR with amvr\_precision\_idx equal to 0 (i.e., 1/2 luma sample motion vector resolution), this implies application of the Switchable Interpolation Filter (SIF)
* Affine AMVR with amvr\_precision\_idx equal to 0 (i.e., 1/16 luma sample motion vector resolution)
* Affine AMVR with amvr\_precision\_idx equal to 1 (i.e., 1 luma sample motion vector resolution)

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstreams with AMVR enabled.

#### Bi-directional optical flow (BDOF)

##### Test bitstream BDOF\_A\_MediaTek

**Specification**: Bitstream exercises all possible implicit BDOF on/off conditions and subblock usages.

**Functional stage**: Inter prediction process

**Purpose**: Check the decoder can properly decode CUs with BDOF enabled.

#### Combined intra/inter prediction (CIIP)

##### Test bitstream CIIP\_A\_MediaTek

**Specification**: The bitstream exercises all possible inter direction, block sizes, and combining weights for CIIP.

**Functional stage**: Inter prediction process

**Purpose**: Check that the decoder can properly decode CUs with CIIP enabled.

#### Merge with MVD (MMVD)

##### Test bitstream MMVD\_A\_SAMSUNG

**Specification**: The bitstream uses MMVD with different numbers of MMVD distance entries

**Functional stage**: Inter prediction process

**Purpose**: Check that the decoder can properly decode bitstreams with merge with MMVD enabled.

#### Bi-predictive with CU weights (BCW)

##### Test bitstream BCW\_A\_MediaTek

**Specification**: The bitstream exercises all possible combining weights for BCW.

**Functional stage**: Inter prediction process

**Purpose**: Check the decoder can properly decode CUs with BCW enabled.

#### Multi-reference line prediction (MRLP)

##### Test bitstream MRLP\_A\_HHI

**Specification**: The bitstream contains all possible combinations of extended intra reference lines for luma indicated by intra\_luma\_ref\_idx={1,2} and most probable modes except the DC, indicated by intra\_luma\_mpm\_idx={0,1,2,3,4}. For the CUs at the top border of a CTU, extended references lines are not used in the MRL index is not present in the bitstream. All CTC tools are enabled.

**Functional stage**: Intra prediction and mode signalling processes

**Purpose**: Test all combinations of reference line indices and associated MPM signalling with all tools on.

##### Test bitstream MRLP\_B\_HHI

**Specification**: The bitstream contains all possible combinations of extended intra reference lines for luma indicated by intra\_luma\_ref\_idx={1,2} and most probable modes except the DC, indicated by intra\_luma\_mpm\_idx={0,1,2,3,4]. For the CUs at the top border of a CTU, extended references lines are not used in the MRL index is not present in the bitstream. The following prediction modes have been disabled. Intra: ISP, MIP, Inter: SBT, MMVD, Affine, SubPuMvp, IMV, BCW, BIO, CIIP, GPM, DisFracMMVD, AffineAmvr, DMVR, SMVD, PROF

**Functional stage**: Intra prediction and mode signalling processes

**Purpose**: Test all combinations of reference line indices and associated MPM signalling with specific intra and inter tools turned off.

#### Intra block copy mode (IBC)

##### Test bitstream IBC\_A\_Tencent

**Specification**: This bitstream exercises general IBC features, merge, skip and AMVP modes.

**Functional stage**: Intra prediction

**Purpose**: Check that the decoder can properly decode bitstreams with IBC enabled.

##### Test bitstream IBC\_B\_Tencent

**Specification**: This bitstream exercises general IBC features, merge, skip and AMVP modes, with BV predictor size equal to 1 (MaxNumIBCMergeCand=1)

**Functional stage**: Intra prediction

**Purpose**: Check that the decoder can properly decode bitstreams with IBC enabled.

##### Test bitstream IBC\_C\_Tencent

**Specification**: This bitstream exercises general IBC features, merge, skip and AMVP modes, with Dual Tree disabled (DualITree=0)

**Functional stage**: Intra prediction

**Purpose**: Check that the decoder can properly decode bitstreams with IBC enabled.

##### Test bitstream IBC\_D\_Tencent

**Specification**: This bitstream exercises general IBC features, merge, skip and AMVP modes, with AMVR disabled.

**Functional stage**: Intra prediction

**Purpose**: Check that the decoder can properly decode bitstreams with IBC enabled.

#### Intra sub-partitioning (ISP)

##### Test bitstream ISP\_A\_HHI

**Specification**: The bitstream contains various combinations of block sizes, ISP split types, intra modes and LFNST indices. It uses an AI configuration and 34 frames with QP 28.

**Functional stage**: Intra prediction

**Purpose**: Check that the decoder can properly decode bitstreams with ISP enabled.

##### Test bitstream ISP\_B\_HHI

**Specification**: The bitstream contains various combinations of block sizes, ISP split types, intra modes and LFNST indices. It uses a RA configuration and 161 frames with QP 34.

**Functional stage**: Intra prediction

**Purpose**: Check that the decoder can properly decode bitstreams with ISP enabled.

#### Decoder motion vector refinement (DMVR)

##### Test bitstream DMVR\_A\_Huawei

**Specification**: This bitstream exercises DMVR with the following features:

* All allowed blocks sizes of DMVR are exercised multiple times.
* Motion vector wraparound is enabled where DMVR uses wrapped around reference samples.
* BCW and explicit weighted biprediction is turned on for luma and chroma components, to test disabling of DMVR.
* All integer delta motion vector combinations are exercised multiple times.
* All fractional Delta MV are exercised multiple times.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstreams with DMVR enabled.

##### Test bitstream DMVR\_B\_KDDI

**Specification**: This bitstream exercises DMVR with corner cases of SAD variations.

**Functional stage**: Inter prediction process

**Purpose**: Check that the decoder can properly decode bitstreams with DMVR enabled.

#### Sub-block transform (SBT)

##### Test bitstream SBT\_A\_Huawei

**Specification**: The bitstream exercises SVT with all allowed blocks sizes exercised multiple times (at least once for each SBT mode for each allowed CU size).

**Functional stage**: Transform process

**Purpose**: Check that the decoder can properly decode bitstreams with DMVR enabled.

#### Luma mapping with chroma scaling (LMCS)

##### Test bitstream LMCS\_A\_Dolby

**Specification**: This bitstream tests control of LMCS at the slice level, with the picture split into 4 tiles and 4 rectangular slices.

**Functional stage**: In-loop filter process

**Purpose**: Check that the decoder can properly decode bitstreams with LMCS enabled.

##### Test bitstream LMCS\_B\_Dolby

**Specification**: This bitstream tests control of LMCS at the slice level, with the picture split into 8 rectangular slices, 12 tiles and 2 subpictures.

**Functional stage**: In-loop filter

**Purpose**: Check that the decoder can properly decode bitstreams with LMCS enabled.

#### Sign data hiding (SDH)

##### Test bitstream SDH\_A\_Dolby

**Specification**: This bitstream tests SDH on/off control at picture level.

**Functional stage**: In-loop filter

**Purpose**: Check that the decoder can properly decode bitstreams with LMCS enabled.

#### Symmetric motion vector difference (SMVD)

##### Test bitstream SMVD\_A\_HUAWEI

**Specification**: This bitstream exercises all allowed blocks sizes of SMVD mode multiple times.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstreams with SMVD enabled.

#### Block-based delta pulse code modulation (BDPCM)

##### Test bitstream BDPCM\_A\_Orange

**Specification**: This bitstream exercises BDPCM-coded block of each possible block size, in both luma and chroma.

**Functional stage**: Intra coding

**Purpose**: Check that the decoder can properly decode bitstreams with BDPCM enabled.

#### Matrix based intra prediction (MIP)

##### Test bitstream MIP\_A\_HHI

**Specification**: This bitstream exercises MIP in different combinations with other tools. The bitstream consists of three coded video sequences (CVSs) with the following properties:

* First CVS: For each M,N in {4,8,16,32,64}, the bitstream contains an MxN-luma-intra-block in which the intra-prediction signal is generated by a mip-mode and in which intra\_mip\_transposed\_flag is false and it contains an MxN-luma-intra-block in which the intra-prediction signal is generated by a mip-mode and in which intra\_mip\_transposed\_flag is true. For each mip-matrix occurring in the spec, the bitstream contains a luma-intra-block in which the intra-prediction signal is generated by a mip-mode that uses this mip-matrix.
* Second CVS: MIP is enabled, MTS and intra tools LFNST, ISP, MRL are disabled. Each slice is an intra slice.
* Third CVS: MIP is enabled, all other intra tools are enabled. Each slice is an intra slice.

**Functional stage**: Intra coding

**Purpose**: Check that the decoder can properly decode bitstreams with MIP enabled.

##### Test bitstream MIP\_B\_HHI

**Specification**: This bitstream exercises MIP with all other CTC tools enabled.

**Functional stage**: Intra coding

**Purpose**: Check that the decoder can properly decode bitstreams with MIP enabled.

#### Low frequency non-separable transform (LFNST)

##### Test bitstream LFNST\_A\_LGE

**Specification**: This bitstream exercises LFNST with MTS disabled, as follows:

* Include all test cases for ISP, MIP, Luma separate tree, and Chroma separate tree with LFNST
* Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
* Include all possible block sizes and partitions (especially for ISP) where LFNST can be applied

**Functional stage**: Transform

**Purpose**: Check that the decoder can properly decode bitstreams with LFNST enabled.

##### Test bitstream LFNST\_B\_LGE

**Specification**: This bitstream exercises LFNST with MTS disabled, as follows:

* Include all test cases for single tree with LFNST
* Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
* Include all possible block sizes where LFNST can be applied

**Functional stage**: Transform

**Purpose**: Check that the decoder can properly decode bitstreams with LFNST enabled.

##### Test bitstream LFNST\_C\_HHI

**Specification**: This bitstream exercises LFNST and its signaling in different combination with other tools. The bitstream consists of four coded video sequences with the following properties:

* The first CVS enables LFNST and explicit MTS; ISP and MIP are enabled.
* The second CVS enables LFNST and explicit MTS; ISP and MIP are disabled.
* The third CVS enables LFNST and explicit MTS; ISP is enabled and MIP is disabled.
* The fourth CVS enables LFNST and explicit MTS; ISP is disabled and MIP is enabled.

**Functional stage**: Transform

**Purpose**: Check that the decoder can properly decode bitstreams with LFNST enabled.

##### Test bitstream LFNST\_D\_HHI

**Specification**: This bitstream exercises LFNST and its signaling in different combination with other tools. The bitstream consists of four coded video sequences with the following properties:

* The first CVS enables LFNST and explicit MTS; ISP and MIP are enabled.
* The second CVS enables LFNST and explicit MTS; ISP and MIP are disabled.
* The third CVS enables LFNST and explicit MTS; ISP is enabled and MIP is disabled.
* The fourth CVS enables LFNST and explicit MTS; ISP is disabled and MIP is enabled.

**Functional stage**: Transform

**Purpose**: Check that the decoder can properly decode bitstreams with LFNST enabled.

#### Transform tool set (MTS\_LFNST )

##### Test bitstream MTS\_LFNST\_A\_LGE

**Specification**: This bitstream exercises various types of enabling of MTS and LFNST. The bitstream consists of five parts with the following properties:

* 1st part
  + Explicit intra MTS on and explicit inter MTS off (CTC)
  + Include all test cases for ISP, MIP, Luma separate tree, and Chroma separate tree
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
  + Include all possible block sizes and partitions (especially for ISP) where all combinations of MTS and LFNST can happen
* 2nd part
  + Explicit intra MTS on and explicit inter MTS off with maximum transform size set to 32
  + Include all test cases for ISP, MIP, Luma separate tree, Chroma separate tree, and TU-tiling based on maximum transform size (32)
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
  + Include all possible block sizes and partitions (especially for ISP) where all combinations of MTS and LFNST can happen
* 3rd part
  + Implicit MTS on and explicit inter MTS off
  + Include all test cases for ISP, MIP, Luma separate tree, and Chroma separate tree
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
  + Include all possible block sizes and partitions (especially for ISP) where all combinations of MTS and LFNST can happen
* 4th part
  + Implicit MTS on and explicit inter MTS on
  + Include all test cases for ISP, MIP, Luma separate tree, and Chroma separate tree
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
  + Include all possible block sizes and partitions (especially for ISP) where all combinations of MTS and LFNST can happen
* 5th part
  + Explicit MTS on and explicit inter MTS on
  + Include all test cases for ISP, MIP, Luma separate tree, and Chroma separate tree
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
  + Include all possible block sizes and partitions (especially for ISP) where all combinations of MTS and LFNST can happen

**Functional stage**: Transform

**Purpose**: Check that the decoder can properly decode bitstreams with MTS and LFNST enabled.

##### Test bitstream MTS\_LFNST\_B\_LGE

**Specification**: This bitstream exercises various types of enabling of MTS and LFNST. The bitstream consists of five parts with the following properties:

* 1st part
  + Explicit intra MTS on and explicit inter MTS off (CTC)
  + Include all test cases for SBT, single tree, and TU-tiling based on maximum transform size (64)
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
  + Include all possible block sizes and partitions (especially for SBT) where all combinations of MTS and LFNST can happen
* 2nd part
  + Explicit intra MTS on and explicit inter MTS off with maximum transform size set to 32
  + Include all test cases for SBT, single tree, and TU-tiling based on maximum transform size (32)
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
  + Include all possible block sizes and partitions (especially for SBT) where all combinations of MTS and LFNST can happen
* 3rd part
  + Implicit MTS on and explicit inter MTS off
  + Include all test cases for SBT, single tree and TU-tiling based on maximum transform size (64)
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
  + Include all possible block sizes and partitions (especially for SBT) where all combinations of MTS and LFNST can happen
* 4th part
  + Implicit MTS on and explicit inter MTS on
  + Include all test cases for SBT, single tree and TU-tiling based on maximum transform size (64)
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
  + Include all possible block sizes and partitions (especially for ISP) where all combinations of MTS and LFNST can happen
* 5th part
  + Explicit MTS on and explicit inter MTS on
  + Include all test cases for SBT, single tree and TU-tiling based on maximum transform size (64)
  + Include all candidates of explicit MTS, i.e. DCT2-DCT2, DST7-DST7, DCT8-DST7, DST7-DCT8, and DCT8-DCT8
  + Include all sets and candidates of LFNST, i.e. 4 sets and 2 candidates per set
  + Include all possible block sizes and partitions (especially for SBT) where all combinations of MTS and LFNST can happen

**Functional stage**: Transform

**Purpose**: Check that the decoder can properly decode bitstreams with MTS and LFNST enabled.

#### Joint coding of chroma residuals (JCCR)

##### Test bitstream JCCR\_A\_Nokia

**Specification**: Bitstream exercises all possible JCCR modes. In addition, different combinations for values of ph\_joint\_cbcr\_sign\_flag and sh\_joint\_cbcr\_qp\_offset syntax elements are included in the bitstream. Coded video contains three frames at resolution of 416x240.

**Functional stage**: TU reconstruction.

**Purpose**: Check that the decoder can properly decode TUs with different JCCR modes, different JCCR QP offsets and different JCCR signs.

##### Test bitstream JCCR\_B\_Nokia

**Specification**: Bitstream exercises all possible JCCR modes. In addition, different combinations for values of ph\_joint\_cbcr\_sign\_flag and sh\_joint\_cbcr\_qp\_offset syntax elements are included in the bitstream. Coded video contains three frames at resolution of 1920x1080.

**Functional stage**: TU reconstruction.

**Purpose**: Check that the decoder can properly decode TUs with different JCCR modes, different JCCR QP offsets and different JCCR signs.

##### Test bitstream JCCR\_C\_HHI

**Specification**: This bitstream exercises joint chroma residual coding in combination with other tools. The bitstream consists of two coded video sequences with the following properties:

* The first CVS uses a random selection of the jointCbCr sign flag, forces the usage of all possible jointCbCr modes, non-related features (inter tools, ALF, ...) are disabled, MTS, LFNST, LMCS, and DQ are disabled.
* The first CVS uses a random selection of the jointCbCr sign flag, forces the usage of all possible jointCbCr modes, non-related features (inter tools, ALF, ...) are disabled, MTS (for intra), LFNST, LMCS, and DQ are enabled.

**Functional stage**: TU reconstruction

**Purpose**: Check that the decoder can properly decode TUs with different JCCR modes

##### Test bitstream JCCR\_D\_HHI

**Specification**: This bitstream exercises joint chroma residual coding in combination with other tools. The bitstream consists of two coded video sequences with the following properties:

* The first CVS uses a random selection of the jointCbCr sign flag, forces the usage of all possible jointCbCr modes, non-related features (inter tools, ALF, ...) are disabled, MTS, LFNST, LMCS, and DQ are disabled.
* The first CVS uses a random selection of the jointCbCr sign flag, forces the usage of all possible jointCbCr modes, non-related features (inter tools, ALF, ...) are disabled, MTS (for intra), LFNST, LMCS, and DQ are enabled.

**Functional stage**: TU reconstruction

**Purpose**: Check that the decoder can properly decode TUs with different JCCR modes

##### Test bitstream JCCR\_E\_Nokia

**Specification**: Bitstream exercises all possible JCCR modes. In addition, different combinations for values of ph\_joint\_cbcr\_sign\_flag and sh\_joint\_cbcr\_qp\_offset syntax elements are included in the bitstream. Coded video contains three frames at resolution of 416x240.

**Functional stage**: TU reconstruction.

**Purpose**: Check that the decoder can properly decode TUs with different JCCR modes, different JCCR QP offsets and different JCCR signs.

##### Test bitstream JCCR\_F\_Nokia

**Specification**: Bitstream exercises all possible JCCR modes. In addition, different combinations for values of ph\_joint\_cbcr\_sign\_flag and sh\_joint\_cbcr\_qp\_offset syntax elements are included in the bitstream. Coded video contains three frames at resolution of 1920x1080.

**Functional stage**: TU reconstruction.

**Purpose**: Check that the decoder can properly decode TUs with different JCCR modes, different JCCR QP offsets and different JCCR signs.

#### Temporal motion vector predictor (TMVP)

##### Test bitstream TMVP\_A\_Chipsnmedia

**Specification**: TBP

**Functional stage**:

**Purpose**:

##### Test bitstream TMVP\_B\_Chipsnmedia

**Specification**: TBP

**Functional stage**:

**Purpose**:

##### Test bitstream TMVP\_C\_Chipsnmedia

**Specification**: TBP

**Functional stage**:

**Purpose**:

##### Test bitstream TMVP\_D\_Chipsnmedia

**Specification**: TBP

**Functional stage**:

**Purpose**:

#### Motion vector compression (MVCOMP)

##### Test bitstream MVCOMP\_A\_Sharp

**Specification**: This bitstream includes large motion vectors that are stored in the temporal motion vector buffer and that are later retrieved for motion vector prediction.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstreams with large motion vectors.

#### Sampled adaptive offset (SAO)

##### Test bitstream SAO\_A\_SAMSUNG

**Specification**: This bitstream uses SAO with ALF and CCALF disabled.

**Functional stage**: In-loop filter

**Purpose**: Check that the decoder can properly decode bitstreams with various in-loop filter combinations.

##### Test bitstream SAO\_B\_SAMSUNG

**Specification**: This bitstream uses SAO with LMCS disabled.

**Functional stage**: In-loop filter

**Purpose**: Check that the decoder can properly decode bitstreams with various in-loop filter combinations.

##### Test bitstream SAO \_C\_SAMSUNG

**Specification**: This bitstream uses SAO with ALF, CCALF, and LMCS disabled.

**Functional stage**: In-loop filter

**Purpose**: Check that the decoder can properly decode bitstreams with various in-loop filter combinations.

#### Prediction refinement using optical flow (PROF)

##### Test bitstream PROF\_A\_Interdigital

**Specification**: This bitstream contains pictures with high rotation motion.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstreams with PROF enabled.

##### Test bitstream PROF\_B \_Interdigital

**Specification**: The bitstream contains pictures with high zoom and rotation motion.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstreams with PROF enabled.

#### Deblocking (DEBLOCKING)

##### Test bitstream DEBLOCKING\_A\_Sharp

**Specification**: This bitstream has luma deblocking filters of lengths (7,7), (7,5), (5,7), (7,3), (3,7), (5,5), (5,3) and (3,5).

**Functional stage**: In-loop filter

**Purpose**: Check that the decoder can properly decode bitstreams with deblocking filter enabled.

##### Test bitstream DEBLOCKING\_B\_Sharp

**Specification**: This bitstream has luma deblocking filters of lengths (3,3) and (1,3).

**Functional stage**: In-loop filter

**Purpose**: Check that the decoder can properly decode bitstreams with deblocking filter enabled.

##### Test bitstream DEBLOCKING\_C\_Huawei

**Specification**: This bitstream tests that luma deblocking is performed on a 4 x 4 deblocking grid and ensures that constrained weak filter ( 1 + 1 ) is applied when one of the blocks sharing the edge has size <= 4 samples in the direction of deblocking.

**Functional stage**: In-loop filter

**Purpose**: Check that the decoder can properly decode bitstreams with deblocking filter enabled.

##### Test bitstream DEBLOCKING\_E\_Ericsson

**Specification**: Bitstream exercises all luma and chroma deblocking lengths for deblocking of transform and prediction block and sub-block boundaries.

**Functional stage**: In-loop filter

**Purpose**: Check that the decoder can properly decode bitstreams with deblocking filter enabled.

##### Test bitstream DEBLOCKING\_F\_Ericsson

**Specification**: Bitstream exercises deblocking control features luma adaptive deblocking and control of beta and tc for both luma and chroma.

**Functional stage**: In-loop filter

**Purpose**: Check that the decoder can properly decode bitstreams with deblocking filter enabled.

#### Weighted prediction (WP)

##### Test bitstream WP\_A\_InterDigital

**Specification**: The bitstream was encoded in random access configuration. The content has fading to black. WP have been disabled for pictures with Tid equal to 2.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstreams with weighted prediction enabled.

##### Test bitstream WP\_B\_InterDigital

**Specification**: The bitstream was encoded in low-delay configuration. The content has flashing and fading.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstreams with weighted prediction enabled.

#### Intra prediction (IP)

##### Test bitstream IP\_A\_Huawei

**Specification**: TBP

**Functional stage**:

**Purpose**:

##### Test bitstream IP\_B\_Nokia

**Specification**: Bitstream exercises all the intra prediction modes.

**Functional stage**: Intra sample prediction.

**Purpose**: Test intra sample reconstruction process, especially wide angle modes in non-square blocks.

#### Luma intra prediction mode (MPM)

##### Test bitstream MPM\_A\_LGE

**Specification**: This bitstream contains MPM candidate for all the sizes of the blocks, i.e. all WxH-sized TUs, where both W and H are equal to one of the following values: {4, 8, 16 or 32}. All intra coding tools except MIP are enabled.

**Functional stage**: Intra prediction

**Purpose**: Check that the decoder can properly decode all MPM modes.

#### CTU sizes (CTU, CU)

##### Test bitstream CTU\_A\_MediaTek

**Specification**: Bitstream exercises all possible CU sizes when maximum CTU size is set to 128x128.

**Functional stage**: Partitioning

**Purpose**: Check that a decoder can parse and reconstruct correctly when maximum CTU size is set to 128x128.

##### Test bitstream CTU\_B\_MediaTek

**Specification**: Bitstream exercises all possible CU sizes when maximum CTU size is set to 64x64.

**Functional stage**: Partitioning

**Purpose**: Check that a decoder can parse and reconstruct correctly when maximum CTU size is set to 64x64.

##### Test bitstream CTU\_C\_MediaTek

**Specification**: Bitstream exercises all possible CU sizes when maximum CTU size is set to 32x32.

**Functional stage**: Test the parsing and reconstruction of slices.

**Purpose**: Check that a decoder can parse and reconstruct correctly when maximum CTU size is set to 32x32.

#### Trees and partitioning (TREE, QTBTT)

##### Test bitstream TREE\_A\_HHI

**Specification**: This bitstream exercises a range of tree size and depths for CTUSize=32.

**Functional stage**: Partitioning

**Purpose**: Check that the decoder can properly decode all partitioning modes.

##### Test bitstream TREE\_B\_HHI

**Specification**: This bitstream exercises a range of tree size and depths for CTUSize=64.

**Functional stage**: Partitioning

**Purpose**: Check that the decoder can properly decode all partitioning modes.

##### Test bitstream TREE\_C\_HHI

**Specification**: This bitstream exercises a range of tree size and depths for CTUSize=128.

**Functional stage**: Partitioning

**Purpose**: Check that the decoder can properly decode all partitioning modes.

##### Test bitstream QTBTT\_A\_MediaTek

**Specification**: Bitstream exercises all possible range of CU sizes and depths for QTBTT partitions.

**Functional stage**: Test the parsing and reconstruction of slices.

**Purpose**: Check that a decoder can parse and reconstruct correctly for all exercise range of CU sizes and depth for QTBTT partitions.

#### Boundary partition (BOUNDARY)

##### Test bitstream BOUNDARY\_A\_Huawei

**Specification**: This bitstream tests boundary handling on specific resolution with WidthxHeight, where Width = 256+8\*n, Heigth = 256 + 8\*m, m and n belong to {0...15}. QT depths for boundary blocks are selected as {1,2,3,4} with POC={1,2,3,4}, respectively

**Functional stage**: Partitioning

**Purpose**: Check that the decoder can properly decode all partitioning modes.

#### Transform (TRANS)

##### Test bitstream TRANS\_A\_Chipsnmedia

**Specification**: TBP

**Functional stage**:

**Purpose**:

##### Test bitstream TRANS\_B\_Chipsnmedia

**Specification**: TBP

**Functional stage**:

**Purpose**:

#### Quantization (QUANT)

##### Test bitstream QUANT\_A\_Huawei

**Specification**: This bitstream tests CU level QP adaptation.

**Functional stage**: Quantization

**Purpose**: Check that the decoder can properly decode all quantization modes.

##### Test bitstream QUANT\_B\_Huawei

**Specification**: This bitstream tests CU level QP adaptation.

**Functional stage**: Quantization

**Purpose**: Check that the decoder can properly decode all quantization modes.

##### Test bitstream QUANT\_C\_Huawei

**Specification**: This bitstream uses low QP with transform skip, so that deblocking filtering not used.

**Functional stage**: Quantization, In-loop filtering

**Purpose**: Check that the decoder can properly decode all quantization modes.

##### Test bitstream QUANT\_D\_Huawei

**Specification :** Deblocking filtering is forced to be used by setting LoopFilterTcOffset\_div2 and LoopFilterTcOffset\_div2 to 12.

**Functional stage**: Quantization, In-loop filtering

**Purpose**: Check that the decoder can properly decode all quantization modes.

#### Scaling list (SCALING)

##### Test bitstream SCALING\_A\_InterDigital

**Specification**: TBP

**Functional stage**:

**Purpose**:

#### Entropy coding (ENTROPY)

##### Test bitstream ENTROPY\_A\_Qualcomm

**Specification**: Bitstream tests CABAC initialization, sweeping QP from 0 to 63.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENTROPY\_B\_Sharp

**Specification**: The bitstream includes all combinations of cabac\_init\_flag in slice header (0, 1, absent)

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

#### Entropy coding (ENTMAINTIER)

##### Test bitstream ENTMAINTIER\_A\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 4. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENTMAINTIER\_B\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 4.1. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENTMAINTIER\_C\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 5. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENTMAINTIER\_D\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 5.1. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

#### Entropy coding (ENTHIGHTIER)

##### Test bitstream ENTHIGHTIER\_A\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 4 for High Tier. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENTHIGHTIER\_B\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 4.1 for High Tier. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENTHIGHTIER\_C\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 5 for High Tier. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENTHIGHTIER\_D\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 5.1 for High Tier. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

#### All merge modes (MERGE)

##### Test bitstream MERGE\_A\_Qualcomm

**Specification**: This bitstream exercises the maximum number of merge candidate (MaxNumMergeCand) = 1 and Maximum number of GMO merge candidate (MaxNumGpmMergeCand ) = 0.

**Functional stage**: Inter prediction

**Purpose**: Check the decoder properly decodes all merge modes.

##### Test bitstream MERGE\_B\_Qualcomm

**Specification**: This bitstream exercises the maximum number of merge candidate (MaxNumMergeCand) = 2 and Maximum number of GMO merge candidate (MaxNumGpmMergeCand ) = 2.

**Functional stage**: Inter prediction

**Purpose**: Check the decoder properly decodes all merge modes.

##### Test bitstream MERGE\_C\_Qualcomm

**Specification**: This bitstream exercises the maximum number of merge candidate (MaxNumMergeCand) = 3 and Maximum number of GMO merge candidate (MaxNumGpmMergeCand ) = 3.

**Functional stage**: Inter prediction

**Purpose**: Check the decoder properly decodes all merge modes.

##### Test bitstream MERGE\_D\_Qualcomm

**Specification**: This bitstream exercises the maximum number of merge candidate (MaxNumMergeCand) = 4 and Maximum number of GMO merge candidate (MaxNumGpmMergeCand ) = 4.

**Functional stage**: Inter prediction

**Purpose**: Check the decoder properly decodes all merge modes.

##### Test bitstream MERGE\_E\_Qualcomm

**Specification**: This bitstream exercises the maximum number of merge candidate (MaxNumMergeCand) = 5 and Maximum number of GMO merge candidate (MaxNumGpmMergeCand ) = 5.

**Functional stage**: Inter prediction

**Purpose**: Check the decoder properly decodes all merge modes.

##### Test bitstream MERGE\_F\_Qualcomm

**Specification**: This bitstream exercises the maximum number of merge candidate (MaxNumMergeCand) = 1 and Maximum number of GMO merge candidate (MaxNumGpmMergeCand ) = 0.

**Functional stage**: Inter prediction

**Purpose**: Check the decoder properly decodes all merge modes.

##### Test bitstream MERGE\_G\_Qualcomm

**Specification**: This bitstream exercises the maximum number of merge candidate (MaxNumMergeCand) = 2 and Maximum number of GMO merge candidate (MaxNumGpmMergeCand ) = 2

**Functional stage**: Inter prediction

**Purpose**: Check the decoder properly decodes all merge modes.

##### Test bitstream MERGE\_H\_Qualcomm

**Specification**: This bitstream exercises the maximum number of merge candidate (MaxNumMergeCand) = 3 and Maximum number of GMO merge candidate (MaxNumGpmMergeCand ) = 3.

**Functional stage**: Inter prediction

**Purpose**: Check the decoder properly decodes all merge modes.

##### Test bitstream MERGE\_I\_Qualcomm

**Specification**: This bitstream exercises the maximum number of merge candidate (MaxNumMergeCand) = 4 and Maximum number of GMO merge candidate (MaxNumGpmMergeCand ) = 4.

**Functional stage**: Inter prediction

**Purpose**: Check the decoder properly decodes all merge modes.

##### Test bitstream MERGE\_J\_Qualcomm

**Specification**: This bitstream exercises the

**Functional stage**: Inter prediction

**Purpose**: Check the decoder properly decodes all merge modes.

#### Position dependent prediction combination (PDPC)

##### Test bitstream PDPC\_A\_Qualcomm

**Specification**: This bitstream tests the clipping function in PDPC for horizontal or vertical intra prediction modes.

**Functional stage**: Intra coding

**Purpose**: Check that the decoder properly decodes the bitstream when PDPC is enabled.

##### Test bitstream PDPC\_B\_Qualcomm

**Specification**: This bitstream uses DPC with various block sizes.

**Functional stage**: Intra coding

**Purpose**: Check that the decoder properly decodes the bitstream when PDPC is enabled.

##### Test bitstream PDPC\_C\_Qualcomm

**Specification**: This bitstream tests the clipping function in PDPC and the PDPC conditional check on the intra prediction mode. In this test each picture is a single I-slice where all the luma blocks are encoded using an identical intra prediction mode.

**Functional stage**: Intra coding

**Purpose**: Check that the decoder properly decodes the bitstream when PDPC is enabled.

#### Wavefronts (WPP)

##### Test bitstream WPP\_A\_Sharp

**Specification**: The bitstream is encoded with sps\_entropy\_coding\_sync\_enabled\_flag equal to 1.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstreams with wavefront enabled.

##### Test bitstream WPP\_B\_Sharp

**Specification**: The bitstream is encoded with rectangular tile and sps\_entropy\_coding\_sync\_enabled\_flag equal to 1 and pictures contain 4 tiles.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstreams with wavefront enabled and tiles.

#### Lossless and near-lossless, include transform skip (LOSSLESS)

##### Test bitstream LOSSLESS\_A\_HHI

**Specification**: The coded slices are either I or B, and all blocks employ the transform skip mode and the regular residual coding stage for entropy coding.

**Functional stage**: Test the parsing and reconstruction of slices.

**Purpose**: Check that a decoder can parse and reconstruct correctly when the bitstream consists of transform skip mode and the regular residual coding stage operating at the lossless operation point.

##### Test bitstream LOSSLESS\_B\_HHI

**Specification**: The coded slices are either I or B, and most of the blocks employ the transform skip mode and the corresponding residual coding stage for entropy coding.

**Functional stage**: Test the parsing and reconstruction of slices.

**Purpose**: Check that a decoder can parse and reconstruct correctly when the bitstream consists of a high amount of transform skip mode but using the transform skip residual coding for entropy coding.

#### Reference picture resizing (RPR)

##### Test bitstream RPR\_A\_Alibaba

**Specification**: This bitstream has 4 pictures. The bitstream contains CUs encoded with inter-prediction mode using reference pictures with a higher resolution than the current picture. The luma resolution is 832x480 for pictures 0 and 1 and 1664x960 for pictures 2 and 3.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode a bitstream with RPR enabled.

##### Test bitstream RPR\_B\_Alibaba

**Specification**: The bitstream contains CUs encoded with inter-prediction mode using reference pictures with a higher resolution than the current picture. The luma resolution is 832x480 for pictures 0 and 1 and of 416x240 for pictures 2 and 3.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode a bitstream with RPR enabled.

##### Test bitstream RPR\_C\_Alibaba

**Specification**: The bitstream contains CUs encoded with inter-prediction mode using reference pictures with a higher resolution than the current picture. The luma resolution is 832x480 for pictures 0 and 1 and of 560x320 for pictures 2 and 3.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode a bitstream with RPR enabled.

#### Cross-Component ALF (CCALF)

##### Test bitstream CCALF\_A\_Sharp

**Specification**: This bitstream enables CCALF with filters that would exceed the output dynamic range and require a clip.

**Functional stage**: In-loop filtering

**Purpose**: Check that the decoder can properly decode a bitstream with CCALF enabled.

##### Test bitstream CCALF\_B\_Sharp

**Specification**: This bitstream enables CCALF for all CTUs in the bit-stream.

**Functional stage**: In-loop filtering

**Purpose**: Check that the decoder can properly decode a bitstream with CCALF enabled.

##### Test bitstream CCALF\_C\_Sharp

**Specification**: This bitstream changes CCALF filters on a picture-by-picture basis.

**Functional stage**: In-loop filtering

**Purpose**: Check that the decoder can properly decode a bitstream with CCALF enabled.

##### Test bitstream CCALF\_D\_Sharp

**Specification**: This bitstream enables CCALF for random CTUs in the bitstream (both channels).

**Functional stage**: In-loop filtering

**Purpose**: Check that the decoder can properly decode a bitstream with CCALF enabled.

#### Geometric Partitioning Mode (GPM)

##### Test bitstream GPM\_A\_Alibaba

**Specification**: This bitstream contains CUs with all the combinations of geometric partition modes, i.e. all the WxH sized CUs with 0 – 63 geometric partition modes. The value of WxH is equal to one of the following values: {8x8, 8x16, 8x32, 16x8, 16x16, 16x32, 16x64, 32x8, 32x16, 32x32, 32x64, 64x16, 64x32 and 64x64}.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode a bitstream with GPM enabled.

##### Test bitstream CodingToolsSets\_A\_Tencent

**Specification**: This bitstream disables many coding tools.

**Functional stage**: General decoding

**Purpose**: Check that a decoder can decode bitstreams using various combinations of coding tools.

##### Test bitstream CodingToolsSets\_B\_Tencent

**Specification**:

**Functional stage**: General decoding

**Purpose**: Check that a decoder can decode bitstreams using various combinations of coding tools.

##### Test bitstream CodingToolsSets\_C\_Tencent

**Specification**:

**Functional stage**: General decoding

**Purpose**: Check that a decoder can decode bitstreams using various combinations of coding tools.

##### Test bitstream CodingToolsSets\_D\_Tencent

**Specification**:

**Functional stage**: General decoding

**Purpose**: Check that a decoder can decode bitstreams using various combinations of coding tools.

##### Test bitstream CodingToolsSets\_E\_Tencent

**Specification**:

**Functional stage**: General decoding

**Purpose**: Check that a decoder can decode bitstreams using various combinations of coding tools.

### Test bitstreams – High-level syntax features for 4:2:0 chroma format and 10 bit depth

#### Access unit delimiter (AUD)

##### Test bitstream: AUD\_A\_Broadcom

**Specification:** Pictures may or may not include associated Access Unit Delimiter (AUD) NAL units. The first 10 pictures of this bitstream do not include AUD, the next 10 pictures included AUD, and finally the last 10 pictures do not include AUD.

**Functional stage:** High level syntax processing / picture boundary processing

**Purpose:** Check that the decoder can handle and transition between pictures with and without associated Access Unit Delimiter NAL units.

#### Filler (FILLER)

##### Test bitstream FILLER\_A\_Bytedance

**Specification**: Each picture includes associated Filler data NAL units.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode when filler data NAL units are present in the bitstream.

#### Decoding Capability Indication (DCI)

##### Test bitstream DCI\_A\_Tencent

**Specification**: The bitstream includes a DCI NAL unit.

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly decode when DCI NAL unit is present in the bitstream.

##### Test bitstream DCI\_B\_Tencent

**Specification**: The bitstream does not include a DCI NAL unit.

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly decode when DCI NAL unit is not present in the bitstream.

#### Sequence parameter set (SPS)

##### Test bitstream SPS\_A\_Bytedance

**Specification**: Multiple SPSs are signalled in the same CVS. SPS with SPS ID equal to 0 is used and the other SPSs are never referenced.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode when multiple SPSs, including unreferenced ones, are contained in the same CVS.

##### Test bitstream SPS\_B\_Bytedance

**Specification**: Multiple SPSs are signalled in the bitstream. Different SPS IDs are used in the different CVSs.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode when different CVSs use different SPSs with different SPS IDs.

##### Test bitstream SPS\_C\_Bytedance

**Specification**: Multiple SPSs are signalled in the bitstream and SPS with SPS ID equal to 0 is used in the bitstream. ALF and BCW are enabled in SPS for the first two CVSs and are disabled in SPS for the third CVS.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode when the content of an SPS is replaced while using the same SPS ID in different CVSs.

#### Video usability information (PQ, HLG)

##### Test bitstream PQ\_A\_Dolby

**Specification**: This bitstream uses VUI transfer characteristics for PQ content.

**Functional stage**: High level syntax

**Purpose**: Check that the decoder can properly parse the VUI.

##### Test bitstream HLG\_B\_NHK

**Specification**: This bitstream uses VUI to indicate "backward-compatible HLG" which is encoded using transfer characteristics with encoded as 1 and also include the alternative transfer characteristics SEI message with preferred transfer characteristics set to 18.

**Functional stage**: High level syntax

**Purpose**: Check that the decoder can properly parse the VUI.

#### Picture parameter set (PPS)

##### Test bitstream PPS\_A\_Bytedance

**Specification**: Multiple PPSs are signalled in the bitstream. PPS with PPS ID equal to 0 is used for each picture and the other PPSs are never referenced.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode when multiple PPSs, including unreferenced ones, are contained in the same CVS.

##### Test bitstream PPS\_B\_Bytedance

**Specification**: Each picture uses an individual PPS with a different PPS ID.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode when PPS IDs are switched for individual pictures in the bitstream.

##### Test bitstream PPS\_C\_Bytedance

**Specification**: Multiple PPSs are signalled in the bitstream. PPS with PPS ID equal to 0 is used for the first two CVSs. The content of the PPS is updated and used for all pictures in the second CVS.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode when the content of a PPS is replaced while using the same PPS ID in different pictures.

#### Mixed NUT (NUT)

##### Test bitstream MNUT\_A\_Nokia

**Specification**: TBP

**Functional stage**: High level syntax processing.

**Purpose**:

#### Extension of parameter set (PSEXT)

##### Test bitstream: PSEXT\_A\_Nokia

**Specification:** The extension flag of the following parameter sets is set to one. DCI, VPS, SPS, PPS, APS

**Functional stage**: Test the handling when the extension\_flag is set to one and the related extension\_data is absent for the following parameter sets. DCI, VPS, SPS, PPS, APS

**Purpose:** Check that the decoder can parse the extension\_flag set to one and handle when the related extension\_data is absent for the following parameter sets. DCI, VPS, SPS, PPS, APS

##### Test bitstream: PSEXT\_B\_Nokia

**Specification:** The extension flag of the following parameter sets is set to one. DCI, VPS, SPS, PPS, APS

**Functional stage:** Test the handling when the extension\_flag is set to one and the related extension\_data has one or more bits for the following parameter sets. DCI, VPS, SPS, PPS, APS

**Purpose:** Check that the decoder can parse the extension\_flag set to one and handle when the related extension\_data has one or more bits for the following parameter sets. DCI, VPS, SPS, PPS, APS

#### Hypothetical reference decoder (HRD)

##### Test bitstream HRD\_A\_Fujitsu

**Specification**: This bitstream tests AU-based HRD operation, using the Buffering Period SEI and Picture Timing SEI messages.

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly process HRD data.

##### Test bitstream HRD\_B\_Fujitsu

**Specification**: This bitstream tests AU-based HRD operation, using the Buffering Period SEI and Picture Timing SEI messages, with 2 DUs in each AU.

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly process HRD data.

#### Adaptation parameter set (APSALF, APSLMCS, APSMULT, SUFAPS)

##### Test bitstream APSALF\_A\_Qualcomm

**Specification**: This bitstream uses multiple ALF APS, with only ALF APS is present in the bistream (LMCS and scaling list are disabled)

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly decode bitstreams using APSs.

##### Test bitstream APSLMCS\_A\_Dolby

**Specification**: The bitstream uses multiple (3) LMCS APS (APS id = 0, 1 and 2), only LMCS APS is present in the bitstream (ALF is disabled).

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly decode bitstreams using APSs.

##### Test bitstream APSLMCS\_B\_Dolby

**Specification**: The bitstream uses multiple (3) LMCS APS (APS id = 0, 1 and 2), both LMCS APS and ALF APS are present in the bitstream.

**Functional stage**: High level syntax procssing

**Purpose**: Check that the decoder can properly decode bitstreams using APSs.

##### Test bitstream APSLMCS\_C\_Dolby

**Specification**: The bitstream tests the use of LMCS APS with a large variation of lmcs CW values ([15 ~ 320]) in each of the 16 bins.

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly decode bitstreams using APSs.

##### Test bitstream APSMULT\_A\_MediaTek

**Specification**: Multiple scaling list APSs (with scaling list APS ID equal to 0 and 1) are signalled in the bitstream. For each picture, the referenced scaling list APS ID is decided according to the picture's POC number.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode when multiple scaling list APSs are contained in the same CVS.

##### Test bitstream APSMULT\_B\_MediaTek

**Specification**: Multiple scaling list APSs are signalled in the bitstream with the same scaling list APS ID. When a scaling list ASP is signalled, it will overwrite the exitsed scaling list ASP.

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly decode when multiple scaling list APSs are contained in the same CVS, and they can be overwritten by each other.

##### Test bitstream SUFAPS \_A\_HHI

**Specification**: TBP

**Functional stage**:

**Purpose**:

#### Random Access Point (RAP)

##### Test bitstream RAP\_A\_HHI

**Specification**: TBP

**Functional stage**:

**Purpose**:

#### Picture output (POUT)

##### Test bitstream POUT\_A\_Sharplabs

**Specification**: This bitstream exercises picture output related syntax, with both values of ph\_pic\_output\_flag.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly output pictures.

#### Gradual decoder refresh (GDR )

##### Test bitstream GDR\_A\_Ericsson

**Specification**: Bitstreams starting with a GDR picture, both ph\_recovery\_poc\_cnt equal to 0 and larger than 0

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can decode and handle GDR signaling.

#### Picture order count (POC)

##### Test bitstream POC\_A\_Nokia

**Specification**: This bitstream exercises POC derivation, including POC reset, using the sps\_poc\_msb\_cycle\_flag, ph\_poc\_msb\_cycle\_present, ph\_poc\_msb\_cycle\_val and sps\_poc\_msb\_cycle\_len\_minus1 syntax elements.

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly derive POC values.

#### Tiles (TILE)

##### Test bitstream TILE\_A\_Nokia

**Specification:** Each picture contains a single tile and single slice.

**Functional stage:** High level syntax processing

**Purpose:** Check that the decoder can properly decode single tile and single slice case.

##### Test bitstream TILE\_B\_Nokia

**Specification:** Each picture contains uniform tile partitioning along both horizontal and vertical directions with each tile containing single slice.

**Functional stage:** High level syntax processing

**Purpose:** Check that the decoder can properly decode uniformly partitioned tiles.

##### Test bitstream TILE\_C\_Nokia

**Specification:** Each picture contains tile partitioning with one row and N columns with each tile containing single slice.

**Functional stage:** High level syntax processing

**Purpose:** Check that the decoder can properly decode when tile partitioning has one row and N columns.

##### Test bitstream TILE\_D\_Nokia

**Specification:** Each picture contains tile partitioning with N rows and one column with each tile containing single slice.

**Functional stage:** High level syntax processing

**Purpose:** Check that the decoder can properly decode when tile partitioning has N rows and one column.

##### Test bitstream TILE\_E\_Nokia

**Specification:** Each picture contains tile partitioning with M rows and N columns with each tile containing single slice.

**Functional stage:** High level syntax processing

**Purpose:** Check that the decoder can properly decode when tile partitioning has M rows and N columns.

##### Test bitstream TILE\_F\_Nokia

**Specification:** Each picture contains tile partitioning with M rows and N columns with each tile containing multiple slices.

**Functional stage:** High level syntax processing

**Purpose:** Check that the decoder can properly decode when tile partitioning has M rows and N columns with each tile containing multiple slices.

##### Test bitstream TILE\_G\_Nokia

**Specification:** Each picture contains tile partitioning with M rows and N columns and the whole picture containing a single slice.

**Functional stage:** High level syntax processing

**Purpose:** Check that the decoder can properly decode when tile partitioning has M rows and N columns and the whole picture containing a single slice.

#### Slices (SLICES)

##### Test bitstream SLICES\_A\_HUAWEI

**Specification**: This bitstream exercises several different slice/tile layouts.

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly decode when a variety of slice and tile layouts are used.

#### Subpictures (SUBPIC)

##### Test bitstream SUBPIC\_A\_HUAWEI

**Specification**: This bitstream exercises several different subpicture layouts.

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly decode when a variety of subpicture layouts are used.

##### Test bitstream SUBPIC\_B\_HUAWEI

**Specification**: This bitstream exercises several different subpicture and slice layouts.

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly decode when a variety of subpicture layouts with slices are used.

#### Picture header and slice header (PHSH)

##### Test bitstream PHSH\_A\_LGE

**Specification**: TBP

**Functional stage**:

**Purpose**:

##### Test bitstream PHSH\_B\_Sharp

**Specification**: This bitstreams includes two CVS. In the first CVS, the picture header is included in the slice header. In the second CVS, the picture header is in its own NAL unit.

**Functional stage**: High level syntax processing

**Purpose**: Check that the decoder can properly decode a picture header whether included in a slice header or not.

#### Temporal scalability (TEMPSCAL)

##### Test bitstream TEMPSCAL\_A\_Panasonic

**Specification**: This bitstream has 6 temporal sublayers with a GOP size of 32.

**Functional stage**: High level syntax processing / GOP processing

**Purpose**: Check that the decoder can handle a GOP of 32.

##### Test bitstream TEMPSCAL\_B\_Panasonic

**Specification**: This bitstream has 5 temporal sublayers with a GOP size of 16 and HRD signalling (including buffering period and picture timing SEI message with timing) for all temporal sublayers.

**Functional stage**: High level syntax processing / Hypothetical Reference Decoder

**Purpose**: Check the hypothetical reference decoder for temporal sublayers.

##### Test bitstream TEMPSCAL\_C\_Panasonic

**Specification**: This bitstream has 3 temporal sublayers with a GOP size of 6.

**Functional stage**: High level syntax processing / GOP processing

**Purpose**: Check that the decoder can properly decode bitstreams with various hierarchy structures.

#### Inter Layer Reference Picture Lists (ILRPL)

##### Test bitstream ILRPL\_A\_Huawei

**Specification**: TBP. This bitstream contains two layers, with layer 1 referencing layer 0.

**Functional stage**: High level syntax processing/scalability

**Purpose**: Check that the decoder can properly decode bitstreams using inter layer reference prediction.

#### Spatial Scalability (SPATSCAL)

##### Test bitstream SPATSCAL\_A\_Qualcomm

**Specification**: Spatial scalability with noncontinuous 3 layers 0, 30, 50. Scaling ratios are (1.05x, 0.75x), (0.51x, 0.69x), (0.54x, 0.51x)

**Functional stage**: Spatial scalability

**Purpose**: Check that the decoder can properly decode bitstreams with noncontinuous layers and atypical scaling ratios.

#### Reference picture lists (RPL)

##### Test bitstream RPL\_A\_ERICSSON

**Specification**: RPL in SPS, PH and SH. Active and inactive entries. Maximum RPL length

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode and handle reference picture lists with short term references.

#### Long term ref picture (LTRP)

##### Test bitstream LTRP\_A\_Ericsson

**Specification**: LTRP handling and picture marking

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode and handle reference picture lists with short and long term references.

#### Number of active ref pics (ACTPIC)

##### Test bitstream ACTPIC\_A\_Huawei

**Specification**: This bitstream contains 1 active picture in list 0, and 1 active picture in list1.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode bitstreams containing various numbers of active reference pictures.

##### Test bitstream ACTPIC\_B\_Huawei

**Specification**: This bitstream contains 1 active picture in list 0, and up to 2 active frames in list 1.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode bitstreams containing various numbers of active reference pictures.

##### Test bitstream ACTPIC\_C\_Huawei

**Specification**: This bitstream contains 1 active picture in list 0, and up to 2 active frames in list 1.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can properly decode bitstreams containing various numbers of active reference pictures.

#### Virtual boundaries (VIRTUAL)

##### Test bitstream VIRTUAL\_A\_MediaTek

**Specification**: Virtual boundaries are enabled and signalled in SPS with 3 vertical and 2 horizontal virtual boundaries.

**Functional stage**: High level syntax processing and in-loop filter process.

**Purpose**: Check that the decoder can properly decode and handle the virtual boundaries signalled in the SPS.

##### Test bitstream VIRTUAL\_B\_MediaTek

**Specification**: Virtual boundaries are enabled and signalled in PH only for pictures with an odd POC value. When virtual boundaries are enabled in a picture, 2 vertical and 1 horizontal virtual boundaries are applied.

**Functional stage**: High level syntax processing and in-loop filter process.

**Purpose**: Check that the decoder can properly decode and handle the virtual boundaries signalled in the PH.

#### Reference wraparound (WRAP)

##### Test bitstream WRAP\_A\_InterDigital

**Specification**: This bitstream uses reference wraparound mode with content using the PERP format for 360ᵒ video.

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstream using reference wraparound.

##### Test bitstream WRAP\_B\_InterDigital

**Specification**:

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstream using reference wraparound.

##### Test bitstream WRAP\_C\_InterDigital

**Specification**:

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstream using reference wraparound.

##### Test bitstream WRAP\_D\_InterDigital

**Specification**:

**Functional stage**: Inter prediction

**Purpose**: Check that the decoder can properly decode bitstream using reference wraparound.

#### 360 Video (CUBEMAP, ERP)

##### Test bitstream CUBEMAP\_A\_MediaTek

**Specification**: A generalized cubemap projection SEI message is signalled in the bitstream with packing type equal to 2, mapping function equal to 2, and guard band flag equal to 0 to indicate that the coded pictures are 3x2-packed non-uniform cubemap projected pictures without guard bands.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can parse the generalized cubemap projection SEI message.

##### Test bitstream CUBEMAP\_B\_MediaTek

**Specification**: A generalized cubemap projection SEI message is signalled in the bitstream with packing type equal to 3, mapping function equal to 1, and guard band flag equal to 1 to indicate that the coded pictures are 6x1-packed equal-angular cubemap projected pictures with guard bands.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can parse the generalized cubemap projection SEI message.

##### Test bitstream CUBEMAP\_C\_MediaTek

**Specification**: A generalized cubemap projection SEI message is signalled in the bitstream with packing type equal to 4, mapping function equal to 0, and guard band flag equal to 1 to indicate that the coded pictures are 5x1-packed hemicubemap projected pictures with guard bands.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can parse the generalized cubemap projection SEI message.

##### Test bitstream ERP\_A\_MediaTek

**Specification**: An equirectangular projection SEI message is signalled in the bitstream with guard band flag equal to 1 to indicate that the coded pictures are equirectangular projected pictures with guard bands on the left and right sides.

**Functional stage**: High level syntax processing.

**Purpose**: Check that the decoder can parse the equirectangular projection SEI message.

#### Conformance cropping window (CROP)

##### Test bitstream CROP\_A\_Panasonic

**Specification**: This bitstream uses large offsets for the conformance cropping window that are not aligned with CTU boundaries.

**Functional stage**: High level syntax processing / conformance cropping

**Purpose**: Check that the decoder outputs the correct conformance cropped region.

##### Test bitstream CROP\_B\_Panasonic

**Specification**:

**Functional stage**: High level syntax processing / conformance cropping

**Purpose**: Check that the decoder outputs the correct conformance cropped region.

#### Bumping (BUMP)

##### Test bitstream BUMP\_A\_LGE

**Specification**: TBP

**Functional stage**:

**Purpose**:

#### Decoded picture buffer (DPB)

##### Test bitstream DPB\_A\_Sharplabs

**Specification**: This bitstream signals sublayer decoded picture buffer (DPB) sizes for multiple sublayers.

**Functional stage**: High-level syntax processing

**Purpose**: Check that decoder can properly decode bitstreams exercising various DPB parameters.

##### Test bitstream DPB\_B\_Sharplabs

**Specification**: This bitstream signals sublayer decoded picture buffer (DPB) size for a single sublayer.

**Functional stage**: High-level syntax processing

**Purpose**: Check that the decoder can properly decode bitstreams exercising various DPB parameters.

#### Field pictures (FIELD)

##### Test bitstream FIELD\_A\_Panasonic

**Specification**: Each picture includes a frame-field information SEI message indicating either it is top or bottom field coded.

**Functional stage**: Test field coding when sps\_field\_seq\_flag is equal to 1.

**Purpose**: Check that the decoder can properly decode pictures coded in field coding.

### Test bitstreams – Additional chroma formats and bit depths for Main 10 profile

#### 8 bit 4:0:0 (8b400)

##### Test bitstream 8b400\_A\_Bytedance

**Specification**: 8-bit 4:0:0 bitstream at a low resolution

**Functional stage**: Additional chroma format and bit depth setting

**Purpose**: Check that the decoder can properly handle the chroma format of 4:0:0 and InternalBitDepth of 8.

##### Test bitstream 8b400\_B\_Bytedance

**Specification**: 8-bit 4:0:0 bitstream at a higher resolution

**Functional stage**: Additional chroma format and bit depth setting

**Purpose**: Check that the decoder can properly handle the chroma format of 4:0:0 and InternalBitDepth of 8.

#### 8 bit 4:2:0 (8b420)

##### Test bitstream 8420\_A\_Bytedance

**Specification**: 8-bit 4:2:0 bitstream at a low resolution

**Functional stage**: Additional bit depth setting

**Purpose**: Check that the decoder can properly handle the InternalBitDepth of 8.

##### Test bitstream 8420\_B\_Bytedance

**Specification**: 8-bit 4:2:0 bitstream at a higher resolution

**Functional stage**: Additional bit depth setting

**Purpose**: Check that the decoder can properly handle the InternalBitDepth of 8.

### Test bitstreams – Coding tools for Main 4:4:4 10 profile for 4:4:4 chroma format and 10 bit depth

#### 10-bit 4:4:4 (10b444)

##### Test bitstream 10b444\_A\_Kwai

**Specification**: TBP

**Functional stage**:

**Purpose**:

##### Test bitstream 10b444\_B\_Kwai

**Specification**: TBP

**Functional stage**:

**Purpose**:

#### Adaptive Color Transform (ACT)

##### Test bitstream ACT\_A\_Kwai

**Specification**: This bitstream tests ACT with CU level adaptation of the color spaces between RGB and YCoCg. The bitstream is Main 444 10 profile, main tier, level 6.

**Functional stage**: Adaptive color transform

**Purpose**: Check that the decoder can properly decode a bitstream with ACT enabled.

#### Palette mode (PALETTE)

##### Test bitstream PALETTE\_A\_Alibaba

**Specification**: This bitstream forces the majority of the CUs to be coded using palette mode in a Random Access configuration. The bitstream is conformant with the Main 444 profile, Main tier.

**Functional stage**: Palette

**Purpose**: Check that the decoder can properly decode a bitstream with palette mode enabled.

##### Test bitstream PALETTE\_B\_Alibaba

**Specification**: This bitstream forces the majority of the CUs to be coded using palette mode in an All Intra configuration. The bitstream is conformant with the Main 444 profile, Main tier.

**Functional stage**: Palette

**Purpose**: Check that the decoder can properly decode a bitstream with palette mode enabled.

##### Test bitstream PALETTE\_C\_Alibaba

**Specification**: This bitstream forces the majority of the CUs to be coded using palette mode in a Low Delay B configuration. The bitstream is conformant with the Main 444 profile, Main tier.

**Functional stage**: Palette

**Purpose**: Check that the decoder can properly decode a bitstream with palette mode enabled.

##### Test bitstream PALETTE\_D\_Alibaba

**Specification**: This bitstream forces the majority of the CUs to be coded using palette mode in an All Intra configuration. The bitstream is conformant with the Main 444 profile, Main tier.

**Functional stage**: Palette

**Purpose**: Check that the decoder can properly decode a bitstream with palette mode enabled.

##### Test bitstream PALETTE\_E\_Alibaba

**Specification**: This bitstream forces the majority of the CUs to be coded using palette mode in an All Intra configuration. The bitstream use 4:2:0 chroma format but is conformant with the Main 444 profile, Main tier, Level 3, which supports use of palette mode for 4:2:0.

**Functional stage**: Palette

**Purpose**: Check that the decoder can properly decode a bitstream with palette mode enabled.

#### Entropy coding (ENT444MAINTIER)

##### Test bitstream ENT444MAINTIER\_A\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 4. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENT444MAINTIER\_B\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 4.1. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENT444MAINTIER\_C\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 5. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENT444MAINTIER\_D\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 5.1. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

#### Entropy coding (ENT444HIGHTIER)

##### Test bitstream ENT444HIGHTIER\_A\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 4 for High Tier. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENT444HIGHTIER\_B\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 4.1 for High Tier. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENT444HIGHTIER\_C\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 5 for High Tier. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

##### Test bitstream ENT444HIGHTIER\_D\_Sony

**Specification**: The bitstream contains 3 independent sequences containing one picture, formed from one slice. Each sequence contains the maximum number of bits, given the profile, level and tier, and assuming equal distribution of bits between pictures coded at the maximum pixel rate for the level, at Level 5.1 for High Tier. The 3 concatenated sequences are as follows:

* The first picture does not require any CABAC zero words,
* The second picture requires one CABAC zero word,
* The third picture requires a substantial quantity of CABAC zero words (75% of the bit-stream is padding).

All 3 pictures have a very low subjective quality level due to the artificial nature of these bit-streams. All VCL NAL units contain almost (within 3 bytes worth) of the maximum number of allowed bins, for their size.

**Functional stage**: Entropy coding

**Purpose**: Check the decoder properly decodes all entropy coding modes.

### Test bitstreams – Additional chroma formats and bit depths for Main 4:4:4 10 profile

#### 8 bit 4:2:2 (8b422)

##### Test bitstream 8b422\_A\_Sony

**Specification**: 8-bit 4:2:2 bitstream for Main 4:4:4 10 profile, in All Intra mode

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

##### Test bitstream 8b422\_B\_Sony

**Specification**: 8-bit 4:2:2 bitstream for Main 4:4:4 10 profile, in All Intra mode

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

##### Test bitstream 8b422\_C\_Sony

**Specification**: 8-bit 4:2:2 bitstream for Main 4:4:4 10 profile, in Low Delay P mode

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

##### Test bitstream 8b422\_D\_Sony

**Specification**: 8-bit 4:2:2 bitstream for Main 4:4:4 10 profile, in Low Delay P mode

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

##### Test bitstream 8b422\_E\_Sony

**Specification**: 8-bit 4:2:2 bitstream for Main 4:4:4 10 profile, in Random Access mode

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

##### Test bitstream 8b422\_F\_Sony

**Specification**: 8-bit 4:2:2 bitstream for Main 4:4:4 10 profile, in Random Access mode

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

##### Test bitstream 8b422\_G\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

##### Test bitstream 8b422\_H\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

##### Test bitstream 8b422\_I\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

##### Test bitstream 8b422\_J\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

##### Test bitstream 8b422\_K\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

##### Test bitstream 8b422\_L\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:2:2 chroma format

#### 8 bit 4:4:4 (8b444)

##### Test bitstream 8b444\_A\_KWAI

**Specification**: TBP

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 8-bit depth and 4:4:4 chroma format

#### 10 bit 4:0:0 (10b400)

##### Test bitstream 10b400\_A\_Bytedance

**Specification**: 10-bit 4:0:0 bitstream at a low resolution

**Functional stage**: Additional chroma format setting

**Purpose**: Check that the decoder can properly handle the chroma format of 4:0:0.

##### Test bitstream 10b400\_B\_Bytedance

**Specification**:10-bit 4:0:0 bitstream at a higher resolution

**Functional stage**: Additional chroma format setting

**Purpose**: Check that the decoder can properly handle the chroma format of 4:0:0.

#### 10 bit 4:2:2 (10b422)

##### Test bitstream 10b422\_A\_Sony

**Specification**: 10-bit 4:2:2 bitstream for Main 4:4:4 10 profile, in All Intra mode

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

##### Test bitstream 10b422\_B\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

##### Test bitstream 10b422\_C\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

##### Test bitstream 10b422\_D\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

##### Test bitstream 10b422\_E\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

##### Test bitstream 10b422\_F\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

##### Test bitstream 10b422\_G\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

##### Test bitstream 10b422\_H\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

##### Test bitstream 10b422\_I\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

##### Test bitstream 10b422\_J\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

##### Test bitstream 10b422\_K\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

##### Test bitstream 10b422\_L\_Sony

**Specification**:

**Functional stage**: Reconstruction

**Purpose**: Check that the decoder can properly decode content coded with 10-bit depth and 4:2:2 chroma format

### Test bitstreams – Multilayer Main 10 profile

#### Video parameter set (VPS)

##### Test bitstream VPS\_A\_INTEL

**Specification**: The bitstream contains two dependent layers. Layer 0 is encoded with 208x120 input video and layer 1 is encoded with 832x480 input video. Inter-layer prediction is enabled for layer 1 (spatial scalability).

**Functional stage**: High-level syntax

**Purpose**: Check that the decoder can properly decode when the VPS is present in the bitstream.

#### Layered coding with OLS (OLS)

##### Test bitstream OLS\_A\_Tencent

**Specification**: The bitstream is encoded with two layers, disallowing inter-layer prediction.

**Functional stage**: High level syntax processing with multi-layers

**Purpose**: Check that the decoder can properly decode when two layers are independently coded and extract a sub-bitstream.

##### Test bitstream OLS\_B\_Tencent

**Specification**: The bitstream is encoded with two layers, allowing inter-layer prediction.

**Functional stage**: High level syntax processing with multi-layers

**Purpose**: Check that the decoder can properly decode when two layers are coded with layer dependencies, and extract a sub-bitstream.

##### Test bitstream OLS\_C\_Tencent

**Specification**: The bitstream is encoded with three layers, allowing inter-layer prediction.

**Functional stage**: High level syntax processing with multi-layers

**Purpose**: Check that the decoder can properly decode when three layers are coded with layer dependencies, and extract a sub-bitstream.

#### Operating point information NAL unit (OPI)

##### Test bitstream OPI\_A\_Nokia

**Specification**: TBP

**Functional stage**:

**Purpose**:

## Conformance test suites for Rec. ITU-T H.266 | ISO/IEC 23090-3

### Bitstreams for Main 10 profile

**Table 1: Coding tool bitstreams for Main 10 profile**

|  |  |  |  |
| --- | --- | --- | --- |
| **Categories** | **Tool description** | **Feature**  **Name** | **Bitstream features** |
| CTU partition | Chroma separate tree | CST | Test CST on and off |
| Transform and quantization | Dependent quantization | DQ | Dependent quantization enabled for all pictures (with and without enabling MTS and LFNST); picture/slice-level switching between dependent quantization, sign data hiding, and conventional quantization |
| Intra coding | Cross-component linear model | CCLM | Exercise CCLM with different CU sizes |
| Transform and quantization | Multiple transform set | MTS | Iinclude MTS combinations of (intra, inter): (implicit, none), (explicit, none), (implicit, explicit), (explicit, explicit) |
| In-loop filter | Adaptive loop filter | ALF | ALF virtual boundary processing for line buffer reduction; Exercise clipping values of non-linear ALF |
| Inter coding | Affine motion model | AFF | Include Affine AMVP and  Affine Merge; Control flags |
| Inter coding | Subblock-based temporal merging candidates | SbTMVP | Test SbTMVP off and test SbTMVP on when affine is off |
| Inter coding | Adaptive motion vector resolution | AMVR | Include SIF |
| Inter coding | Bi-directional optical flow | BDOF | Test BDOF with different implicit on/off decision and subblock usages |
| Inter coding | Combined intra/inter prediction | CIIP | Test CIIP for different sizes and different combining weights |
| Inter coding | Merge with MVD | MMVD |  |
| Inter coding | Bi-predictive with CU weights | BCW | Test BCW with different BCW weights |
| Intra coding | Multi-reference line prediction | MRLP | Test all MRL and MPM indices, CUs at the top border of a CTU do not use extended references lines and the MRL index is not present in the bitstream. |
| SCC coding | Intra block copy mode | IBC | Test IBC feature with different options and and combination with CST and AMVR |
| Intra coding | Intra sub-partitioning | ISP |  |
| Inter coding | Decoder motion vector refinement | DMVR | Exercise enabling conditions, MV wraparound and MV clip at picture boundary, Exercise corner cases of SAD variations |
| Transform and quantization | Sub-block transform | SBT | Different block sizes |
| In-loop filter | Luma mapping with chroma scaling | LMCS | Exercise multiple APSes, slice level LMCS on/off |
| Entropy coding | Sign data hiding | SDH |  |
| Inter coding | Symmetric motion vector difference | SMVD | Long-term reference handling, mvd\_l1\_zero\_flag handling |
| Intra coding | Block-based delta pulse code modulation | BDPCM |  |
| Intra coding | Matrix based intra prediction | MIP |  |
| Transform and quantization | Low frequency non-separable transform | LFNST | Various block sizes /shapes |
| Transform and quantization | Transform tool set | MTS\_LFNST | Tool on or off of MTS and LFNST together with implicit or explicit MTS |
| Transform and quantization | Joint coding of chroma residuals | JCCR | Exercise all modes of joint coding of chroma residuals (JCCR) |
| Inter coding | Temporal motion vector predictor | TMVP |  |
| Inter coding | Motion vector compression | MVCOMP | MV compression for temporal storage (including corner cases) |
| In-loop filter | Sampled adaptive offset | SAO |  |
| Inter coding | Prediction refinement using optical flow | PROF | Various non-translational motion parameters for PROF |
| In-loop filter | Deblocking | DEBLOCKING | Exercise luma adaptive deblock filter and long tap filter |
| Inter coding | Weighted prediction | WP | Various combinations with other inter tools |
| Intra coding | Intra prediction | IP | Enable all modes, especially the wide-angle modes. |
| Intra coding | Luma intra prediction mode | MPM | Enable all conditions to generate MPM candidate |
| CTU partition | CTU, CU sizes | CTU | Exercise range of CTU, CU sizes |
| CTU partition | Trees and partitioning | TREE | Exercise range of sizes and depths of TREE, TT, QT |
| CTU partition | Trees and partitioning | QTBTT | Exercise range of sizes and depths of QT, TT |
| CTU partition | Boundary partition | BOUNDARY | Boundary are sizes 8…120 samples, all combinations of QT and TREE |
| Transform and quantization | Transform | TRANS | min and max transform, min number of entropy coded coeff., max number of coeff. |
| Transform and quantization | Quantization | QUANT | CU level delta QP, CU level Chroma delta QP, transform-quant bypass with DB |
| Transform and quantization | Scaling list | SCALING | Exercise multiple APSs, quantization matrices |
| Entropy coding | Entropy coding | ENTROPY | CABAC initialization: QP sweep to ensure proper initialization for every QP value; disabling signaling of cabac\_init\_idc |
| Entropy coding | Entropy coding | ENTMAINTIER | Max bins and bits, min bits for Main Tier |
| Entropy coding | Entropy coding | ENTHIGHTIER | Max bins and bits for High tier |
| Inter coding | All merge modes | MERGE | max number of merge candidates |
| Intra coding | Position dependent prediction combination (PDPC) | PDPC | Force clipping. Different PU sizes and shapes. |
| Entropy coding | Wavefronts | WPP |  |
| Transform and quantization | Lossless and near-lossless, include transform skip | LOSSLESS |  |
| Inter coding | Reference picture resizing | RPR | Multiple RPR scale factors, RPR cropping window and scaling window offsets |
| In-loop filter | CC-ALF | CCALF | Enable CC-ALF for all CTUs in the bit-stream, Slice-by-slice adaptation, CTU-by-CTU adaptation, Dynamic range exercise (including clip) |
| Inter coding | Geometric Partitioning Mode | GPM |  |

**Table** 3**: Coding tools sets bitstreams for Main 10 profile**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Name** | CodingToolsSets | | | | |
| **Syntax element** | **A** | **B** | **C** | **D** | **E** |
| slice\_type | I | I, P | I | I, P | I, P, B |
| bit\_depth\_minus8 | 0 | 0 | 0 | 0 | 0 |
| separate\_colour\_plane\_flag | 0 | 0 | 0 | 0 | 0 |
| ref\_pic\_resampling\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_log2\_ctu\_size\_minus5 | 0 | 0 | 1 | 1 | 1 |
| subpics\_present\_flag | 0 | 0 | 0 | 0 | 1 |
| bit\_depth\_minus8 | 0 | 0 | 2 | 2 | 2 |
| sps\_weighted\_pred\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_weighted\_bipred\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_max\_luma\_transform\_size\_64\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_sao\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_alf\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_transform\_skip\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_bdpcm\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_temporal\_mvp\_enabled\_flag | 0 | 0 | 0 | 1 | 1 |
| sps\_sbtmvp\_enabled\_flag | 0 | 0 | 0 | 1 | 1 |
| sps\_amvr\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_bdof\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_bdof\_pic\_present\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_smvd\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_dmvr\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_dmvr\_pic\_present\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_mmvd\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_isp\_enabled\_flag | 0 | 0 | 1 | 1 | 1 |
| sps\_mrl\_enabled\_flag | 0 | 0 | 0 | 1 | 1 |
| sps\_mip\_enabled\_flag | 0 | 0 | 0 | 1 | 1 |
| sps\_mts\_enabled\_flag | 0 | 0 | 1 | 1 | 1 |
| sps\_sbt\_enabled\_flag | 0 | 0 | 0 | 1 | 1 |
| sps\_affine\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_bcw\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_ibc\_enabled\_flag | 0 | 0 | 0 | 1 | 1 |
| sps\_ciip\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_fpel\_mmvd\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_geo\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_lmcs\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_lfnst\_enabled\_flag | 0 | 0 | 0 | 1 | 1 |
| sps\_ladf\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |
| sps\_scaling\_list\_enabled\_flag | 0 | 0 | 0 | 0 | 1 |

**Table 3: High-level syntax feature bitstreams for Main 10 profile**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Feature**  **Name** | **Bitstream features** |
| Access unit delimiter | AUD | AUD NAL units present |
| Filler data | FILLER | Filler data NAL units present |
| Decoding Capability Information | DCI | DPS present |
| Sequence parameter set | SPS | Multiple SPSs |
| Video usability information | PQ | PQ EOTF |
| Video usability information | HLG | HLG EOTF |
| Picture parameter set | PPS | Multiple PPSs |
| Mixed NUT | MNUT | Mixed NAL unit type within a picture |
| Extension of parameter set | PSEXT | Extension data in DPS, VPS, SPS, PPS, PH, SH |
| Hypothetical reference decoder | HRD | HRD signalling, including HSS, DU, rept., Field(?) |
| Adaptation parameter set | APSALF | Multiple APSs of each type (ALF) |
| Adaptation parameter set | APSLMCS | Multiple APSs of each type (LMCS) |
| Adaptation parameter set | APSMULT | Multiple APSs of each type (Scaling list) |
| Adaptation parameter set | SUFAPS | Suffix APS NAL units |
| Random access point | RAP | IRAP functionality |
| Picture output | POUT | Pic output flag, no output pic |
| Gradual decoder refresh | GDR | GDR |
| Picture order count | POC | POC derivation, POC reset, POC MSB present |
| Tiles | TILE | Pictures partitions in tiles with same and different tile size, slices in tiles, tiles in slices |
| Slices | SLICES | Raster-scan and rectangular slices |
| Subpictures | SUBPIC | Various number of sub-pictures, sub-pictures of varying sizes, multiple tiles in some sub-pictures, mix of independent and non-independent subpictures, use of subpicture ID in PPS and in PH |
| Picture header and slice header | PHSH | Signalling control of syntax elements in PH and SH, Bitstream with both picture header in slice header as well as picture header in its own NALU, Collocated picture for TMVP signalled in picture header |
| Temporal scalability | TEMPSCAL | Maximum number of temporal layers, varying number of temporal sub-layers |
| Inter-layer reference picture list | ILRPL | Inter-layer ref pic list |
| Reference picture lists | RPL | Reference list modification |
| Long term ref picture | LTRP | LTRP handling |
| Number of active ref pics | ACTPIC | Using default, using different num of active ref pic in slices |
| Virtual boundaries | VIRTUAL | Loop filters on/off at non-tile/slice boundaries |
| Reference wraparound | WRAP |  |
| 360 Video | CUBEMAP | Cube map layout and SEI |
| 360 Video | ERP | Equirectangular projection layout and SEI |
| Conformance cropping window | CROP | Use of conformance cropping parameters, varying parameters, large offsets, odd offset values for 4:4:4, Cropping window signaling in both SPS and PPS |
| Bumping | BUMP | Bumping process |
| Decoded picture buffer | DPB | Sublayer DPB size signalling |
| Field pictures | FIELD | frame-field information (SPS) |

**Table 2: Additional chroma formats and bit depths Main 10 profile**

|  |  |  |
| --- | --- | --- |
| **Tool description** | **Feature**  **Name** | **Bitstream features** |
| 8b 4:0:0 | 8b400 | 8-bit 4:0:0 in Main 10 profile |
| 8b 4:2:0 | 8b420 | 8-bit 4:2:0 in Main 10 profile |
| 10b 4:0:0 | 10b400 | 10-bit 4:0:0 in Main 10 profile |

### Bitstreams for Main 4:4:4 10 profile

**Table 5: Coding tools for Main 4:4:4 10 profile with 4:4:4 10 bit**

|  |  |  |  |
| --- | --- | --- | --- |
| **Categories** | **Tool description** | **Feature**  **Name** | **Bitstream features** |
| Basic 4:4:4 | 10-bit 4:4:4 | 10b444 | No 4:4:4-specific coding tools enabled |
| SCC coding | Adaptive Color Transform | ACT |  |
| SCC coding | Palette mode | PALETTE |  |
| Entropy | Entropy coding for 4:4:4 main tier | ENT444MAINTIER | Max bins and bits for Main 444 Main Tier |
| Entropy | Entropy coding for 4:4:4 high tier | ENT444HIGHTIER | Max bins and bits for Main 444 High Tier |

**Table 6: Additional chroma formats and bit depths Main 4:4:4 10 profile**

|  |  |  |
| --- | --- | --- |
| **Tool description** | **Feature**  **Name** | **Bitstream features** |
| 8b 4:2:2 | 8b422 | 8-bit 4:2:2 in Main 4:4:4 10 profile |
| 8b 4:4:4 | 8b444 | 8-bit 4:4:4 in Main 4:4:4 10 profile |
| 10b 4:2:2 | 10b422 | 10-bit 4:2:2 in Main 4:4:4 10 profile |

### Bitstreams for Multilayer Main 10 profile

**Table 7: Bitstreams for Multilayer Main 10 profile**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Feature**  **Name** | **Bitstream features** |
| Video parameter set | VPS | Single and multiple VPSs |
| Spatial scalability | SPATSCAL | Different resolution layers, including unusual scaling ratios |
| Layered coding with OLS | OLS | Layered coding w/ and w/o inter-layer prediction, with 2 and 3 layers |
| Operating point information NAL unit | OPI | Use OPI NAL unit in bitstream to determine output |

### Bitstreams for Multilayer Main 4:4:4 10 profile

TBP

1. This Recommendation | International Standard includes an electronic attachment containing the conformance bitstreams identified within the text. The conformance bitstreams needed for this Recommendation are available at the following link: <http://wftp3.itu.int/av-arch/jvet-site/bitstream_exchange/draft_conformance/draft5>. The bitstreams can also be downloaded from the ITU-T Test Signal Database. [↑](#footnote-ref-2)