FINAL
DRAFT

AMENDMENT

ISO/IEC 23008-2:2013 FDAM 4

ISO/IEC JTC **1**/SC **29**

Secretariat: JISC

Voting begins on: **2015-05-06**

Voting terminates on: **2015-07-06**

Information technology — High efficiency coding and media delivery in heterogeneous environments —

Part 2: High efficiency video coding

AMENDMENT 4

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Technologies de l'information — Codage à haute efficacité et livraison des medias dans des environnements hétérogènes —

Partie 2: Codage vidéo à haute efficacité

AMENDEMENT 4

ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d36-5878-4565-9f89-2318538...-23008-2-2013-fdamd-4

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.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

Reference number
ISO/IEC 23008-2:2013/FDAM 4:2015(E)

© ISO/IEC 2015

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d36-5878-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2015, Published in Switzerland

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

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

CONTENTS

Page

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d36-5878-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d36-5878-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

LIST OF TABLES

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d36-5878-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d36-5878-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

Introduction

Replace the text of 0.2 with the following:

As the costs for both processing power and memory have reduced, network support for coded video data has diversified, and advances in video coding technology have progressed, the need has arisen for an industry standard for compressed video representation with substantially increased coding efficiency and enhanced robustness to network environments. Toward these ends the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) formed a Joint Collaborative Team on Video Coding (JCT-VC) in 2010 and a Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) in 2012 for development of a new Recommendation | International Standard. This Recommendation | International Standard was developed in the JCT-VC and the JCT-3V.

In 0.3 add the following sentence to the end of the clause:

Support for 3D enables joint representation of video content and depth information with multiple camera views.

In 0.5 add the following paragraph to the end of the clause:

Rec. ITU-T H.265 | ISO/IEC 23008-2 version 3 refers to the integrated text containing 3D extensions, additional supplement enhancement information, and corrections to various minor defects in the prior content of the specification.

In 0.8, replace "Annexes A through H" with "Annexes A through I".

In 0.8, add the following sentence after the sentence that starts with "Annex H":

Annex I contains support for 3D coding.

Sequence parameter set RBSP

In 7.3.2.2.1 and F.7.3.2.2.1 replace the row containing "if(sps_extension_present_flag)" and all following rows in the syntax table by the following:

iTeh STANDARD PREVIEW
(standards.iteh.ai)

if(sps_extension_present_flag) {	
sps_range_extension_flag	u(1)
sps_multilayer_extension_flag	u(1)
sps_3d_extension_flag	u(1)
sps_extension_5bits	u(5)
}	
if(sps_range_extension_flag)	
sps_range_extension( )	
if(sps_multilayer_extension_flag)	
sps_multilayer_extension( ) /* specified in Annex F */	
if(sps_3d_extension_flag)	
sps_3d_extension( ) /* specified in Annex I */	
if(sps_extension_5bits)	
while(more_rbsp_data( ))	
sps_extension_data_flag	u(1)
rbsp_trailing_bits( )	
}	

ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d36-5878-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

In 7.4.3.2.1 replace the semantics of sps_extension_present_flag with the following semantics:

sps_extension_present_flag equal to 1 specifies that the syntax elements sps_range_extension_flag, sps_multilayer_extension_flag, sps_3d_extension_flag, and sps_extension_5bits are present in the SPS RBSP syntax structure. sps_extension_present_flag equal to 0 specifies that these syntax elements are not present.

In 7.4.3.2.1 add the following semantics after semantics of sps_multilayer_extension_flag:

sps_3d_extension_flag equal to 1 specifies that the sps_3d_extension( ) syntax structure (specified in Annex I) is present in the SPS RBSP syntax structure. sps_3d_extension_flag equal to 0 specifies that the sps_3d_extension( ) syntax structure is not present. When not present, the value of sps_3d_extension_flag is inferred to be equal to 0.

In 7.4.3.2.1 replace all occurrences of "sps_extension_6bits" with "sps_extension_5bits".

Picture parameter set RBSP

In 7.3.2.3.1 replace the row containing "if(pps_extension_present_flag)" and all following rows in the syntax table with the following:

if(pps_extension_present_flag) {	
pps_range_extension_flag	u(1)
pps_multilayer_extension_flag	u(1)
pps_3d_extension_flag	u(1)
pps_extension_5bits	u(5)
}	
if(pps_range_extension_flag)	
pps_range_extension( )	
if(pps_multilayer_extension_flag)	
pps_multilayer_extension( ) /* specified in Annex F */	
if(pps_3d_extension_flag)	
pps_3d_extension( ) /* specified in Annex I */	
if(pps_extension_5bits)	
while(more_rbsp_data( ))	
pps_extension_data_flag	u(1)
rbsp_trailing_bits( )	
}	

iTeh STANDARD PREVIEW
(standards.iteh.ai)

In 7.4.3.3.1 replace the semantics of pps_extension_present_flag with the following semantics:

ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d56-5878-4565-9f89-2318538f7e92a/iso-iec-23008-2-2013-fdamd-4

pps_extension_present_flag equal to 1 specifies that the syntax elements pps_range_extension_flag, pps_multilayer_extension_flag, pps_3d_extension_flag, and pps_extension_5bits are present in the picture parameter set RBSP syntax structure. pps_extension_present_flag equal to 0 specifies that these syntax elements are not present.

In 7.4.3.3.1 add the following semantics after semantics of pps_multilayer_extension_flag:

pps_3d_extension_flag equal to 1 specifies that the pps_3d_extension( ) syntax structure (specified in Annex I) is present in the PPS RBSP syntax structure. pps_3d_extension_flag equal to 0 specifies that the pps_3d_extension( ) syntax structure is not present. When not present, the value of pps_3d_extension_flag is inferred to be equal to 0.

In 7.4.3.3.1 replace all occurrences of "pps_extension_6bits" with "pps_extension_5bits".

General SEI message syntax

In D.2.1 add the following rows to the syntax table after the row containing "multiview_view_position(payloadSize) / specified in Annex G */":*

else if(payloadType = = 181)	
alternative_depth_info(payloadSize) /* specified in Annex I */	

Common decoding process for a coded picture

In F.8.1.3 add the following paragraph before the paragraph starting with "After all slices of the current picture have been decoded,":

– Otherwise, general_profile_idc in the profile_tier_level( ) syntax structure VpsProfileTierLevel[profile_tier_level_idx[TargetOlsIdx][lIdx]] is equal to 8, the decoding process for the current picture takes as inputs the syntax elements and upper-case variables from clause I.7 and the decoding process of clause I.8.1.2 is invoked.

Video parameter set RBSP semantics

In F.7.4.3.1 replace the semantics of vps_extension2_flag with the following semantics:

vps_extension2_flag equal to 0 specifies that no vps_extension_data_flag syntax elements are present in the VPS RBSP syntax structure. vps_extension2_flag equal to 1 specifies vps_extension_data_flag syntax elements are present in the VPS RBSP syntax structure. Decoders conforming to a profile specified in Annexes A, G, or H shall ignore all data that follow the value 1 for vps_extension2_flag in a VPS NAL unit.

In F.7.4.3.1 add the following semantics:

vps_extension_data_flag may have any value. Its presence and value do not affect decoder conformance to profiles specified in Annexes A, G, or H. Decoders conforming to a profile specified in Annexes A, G, or H shall ignore all vps_extension_data_flag syntax elements.

In F.7.4.3.1.1 replace Table F.1 with the following table:

Table F.1 – Mapping of ScalabiltyId to scalability dimensions

scalability mask index	Scalability dimension	ScalabilityId mapping
0	Texture or depth	DepthLayerFlag
1	Multiview	ViewOrderIdx
2	Spatial/quality scalability	DependencyId
3	Auxiliary	AuxId
4-15	Reserved	

iTeh STANDARD PREVIEW
(standards.iteh.ai)

In F.7.4.3.1.1 remove Note 2.

In F.7.4.3.1.1 replace the paragraph starting with "The variable ScalabilityId[i][smIdx] specifying the identifier" and the following equation (F-2), with the following:

ISO/IEC 23008-2:2013/FDAmd 4

The variable ScalabilityId[i][smIdx] specifying the identifier of the smIdx-th scalability dimension type of the i-th layer, and the variables DepthLayerFlag[lId], ViewOrderIdx[lId], DependencyId[lId], and AuxId[lId] specifying the depth flag, view order index, the spatial/quality scalability identifier, and the auxiliary identifier, respectively, of the layer with nuh_layer_id equal to lId are derived as follows:

```
NumViews = 1
for( i = 0; i  <=  MaxLayersMinus1; i++ ) {
    lId = layer_id_in_nuh[ i ]
    for( smIdx= 0, j = 0; smIdx < 16; smIdx++ ) {
        if( scalability_mask_flag[ smIdx ] )
            ScalabilityId[ i ][ smIdx ] = dimension_id[ i ][ j++ ]
        else
            ScalabilityId[ i ][ smIdx ] = 0
    }
    DepthLayerFlag[ lId ] = ScalabilityId[ i ][ 0 ]
    ViewOrderIdx[ lId ] = ScalabilityId[ i ][ 1 ]
    DependencyId[ lId ] = ScalabilityId[ i ][ 2 ]                    (F-2)
    AuxId[ lId ] = ScalabilityId[ i ][ 3 ]
    if( i > 0 ) {
        newViewFlag = 1
        for( j = 0; j < i; j++ )
            if( ViewOrderIdx[ lId ]  = =  ViewOrderIdx[ layer_id_in_nuh[ j ] ] )
                newViewFlag = 0
        NumViews  +=  newViewFlag
    }
}
```

In F.7.4.3.1.1 replace the semantics of direct_dep_type_len_minus2 with the following:

direct_dep_type_len_minus2 plus 2 specifies the number of bits of the direct_dependency_type[i][j] and the direct_dependency_all_layers_type syntax elements. In bitstreams conforming to this version of this Specification the

value of direct_dep_type_len_minus2 shall be equal 0 or 1. Although the value of direct_dep_type_len_minus2 shall be equal to 0 or 1 in this version of this Specification, decoders shall allow other values of direct_dep_type_len_minus2 in the range of 0 to 30, inclusive, to appear in the syntax.

In F.7.4.3.1.1 replace the semantics of direct_dependency_all_layers_type with the following:

direct_dependency_all_layers_type, when present, specifies the inferred value of direct_dependency_type[i][j] for all combinations of i-th and j-th layers. The length of the direct_dependency_all_layers_type syntax element is direct_dep_type_len_minus2 + 2 bits. Although the value of direct_dependency_all_layers_type is required to be in the range of 0 to 6, inclusive, in this version of this Specification, decoders shall allow values of direct_dependency_all_layers_type in the range of 0 to $2^{32} - 2$, inclusive, to appear in the syntax.

In F.7.4.3.1.1 replace the semantics of direct_dependency_type with the following:

direct_dependency_type[i][j] indicates the type of dependency between the layer with nuh_layer_id equal layer_id_in_nuh[i] and the layer with nuh_layer_id equal to layer_id_in_nuh[j]. direct_dependency_type[i][j] equal to 0 specifies that the layer with nuh_layer_id equal to layer_id_in_nuh[j] may be used for inter-layer sample prediction but is not used for inter-layer motion prediction of the layer with nuh_layer_id equal layer_id_in_nuh[i]. direct_dependency_type[i][j] equal to 1 specifies that the layer with nuh_layer_id equal to layer_id_in_nuh[j] may be used for inter-layer motion prediction but is not used for inter-layer sample prediction of the layer with nuh_layer_id equal layer_id_in_nuh[i]. direct_dependency_type[i][j] equal to 2 specifies that the layer with nuh_layer_id equal to layer_id_in_nuh[j] may be used for both inter-layer motion prediction and inter-layer sample prediction of the layer with nuh_layer_id equal layer_id_in_nuh[i]. The length of the direct_dependency_type[i][j] syntax element is direct_dep_type_len_minus2 + 2 bits. Although the value of direct_dependency_type[i][j] shall be in the range of 0 to 2, inclusive, when the layer with nuh_layer_id equal to layer_id_in_nuh[i] conforms to a profile specified in Annexes A, G, or H, and in the range of 0 to 6, inclusive, when the layer with nuh_layer_id equal to layer_id_in_nuh[i] conforms to a profile specified in Annex I, decoders shall allow values of direct_dependency_type[i][j] in the range of 0 to $2^{32} - 2$, inclusive, to appear in the syntax.

Profiles, tiers, and levels

iTeh STANDARD PREVIEW
(standards.iteh.ai)

In F.11.2 add the following row to the end of Table F.3:

3D Main	3D Main, Multiview Main, Main, Main Still Picture

ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d36-5878-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

In G.11.1.1 and H.11.1.1 remove " and sps_extension_6bits equal to 0 only".

In G.11.1.1 and H.11.1.1 remove " and pps_extension_6bits equal to 0 only".

Add a new Annex I as follows:

Annex I

3D high efficiency video coding

(This annex forms an integral part of this Recommendation | International Standard.)

I.1 Scope

This annex specifies syntax, semantics, and decoding processes for 3D high efficiency video coding that use the syntax, semantics, and decoding processes specified in clauses 2-10 and Annexes A-G. This annex also specifies profiles, tiers, and levels for 3D high efficiency video coding.

I.2 Normative references

The list of normative references in clause G.2 apply.

I.3 Definitions

For the purpose of this annex, the following definitions apply in addition to the definitions in clause G.3. These definitions are either not present in clause G.3 or replace definitions in clause G.3.

I.3.1 **depth intra contour prediction**: A *prediction* of a *partition pattern* for a *prediction block* in a *picture* of a *depth layer* derived from samples of a *picture* included in the same *access unit* and in the *texture layer* of the same *view*.

I.3.2 **depth layer**: A *layer* with a nuh_layer_id value equal to i, such that DepthLayerFlag[i] is equal to 1 and DependencyId[i] and AuxId[i] are equal to 0.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

I.3.3 **depth look-up table**: A list containing *depth values*.

I.3.4 **depth value**: A sample value of a *decoded picture* of a *depth layer*.

ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/4e2ed136-5878-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

I.3.5 **disparity vector**: A *motion vector* used for inter-view prediction.

I.3.6 **inter-component prediction**: An *inter-layer prediction* where the *reference pictures* are associated with a DepthFlag value different from the DepthFlag value of the current *picture*.

I.3.7 **inter-view prediction**: An *inter-layer prediction* where the *reference pictures* are associated with *reference view order index* values different from the ViewIdx value of the current *picture*.

I.3.8 **intra prediction**: A *prediction* derived from only data elements (e.g., sample values) of the same decoded *slice* and additionally *may* be using *depth intra contour prediction*.

I.3.9 **partition pattern**: An MxM (M-column by M-row) array of *flags* defining two *sub-block partitions* of an MxM *prediction block*.

I.3.10 **prediction block**: A rectangular MxN *block* of samples on which either the same *prediction* or *partitioning* in *sub-block partitions* is applied.

I.3.11 **reference view order index**: A ViewIdx value associated with a *reference picture* used for *inter-view prediction*.

I.3.12 **sub-block partition**: A subset of samples of a *prediction block* on which the same *prediction* is applied.

I.3.13 **texture layer**: A *layer* with a nuh_layer_id value equal to i, such that DepthLayerFlag[i], DependencyId[i], and AuxId[i] are equal to 0.

I.4 Abbreviations

The specifications in clause G.4 apply.

I.5 Conventions

The specifications in clause G.5 apply.

I.6 Bitstream and picture formats, partitionings, scanning processes, and neighbouring relationships

I.6.1 Bitstream formats

The specifications in clause 6.1 apply.

I.6.2 Source, decoded, and output picture formats

The specifications in clause 6.2 apply.

I.6.3 Partitioning of pictures, slices, slice segments, tiles, coding tree units, and coding tree blocks

The specifications in clause 6.3 and its subclauses apply.

I.6.4 Availability processes

The specifications in clause 6.4 apply.

I.6.5 Scanning processes

The specifications in clause 6.5 and its subclauses apply.

I.6.6 Derivation process for a wedgelet partition pattern table

NOTE – Tables and values resulting from this process are independent of any information contained in the bitstream.

The list WedgePatternTable[log2BlkSize] of partition patterns of size (1 << log2BlkSize)x(1 << log2BlkSize) and the variable NumWedgePattern[log2BlkSize] specifying the number of partition patterns in list WedgePatternTable[log2BlkSize] are derived as follows:

– For log2BlkSize in the range of 2 to 4, inclusive, the following applies:
  – NumWedgePattern[log2BlkSize] is set equal to 0.
  – The variable resShift is set equal to (log2BlkSize == 4) ? 0 : 1.
  – The variable wBlkSize is set equal to (1 << (log2BlkSize + resShift)).
  – For wedgeOri in the range of 0 to 5, inclusive, the following applies:
    – The variable posEnd is set equal to NumWedgePattern[log2BlkSize].
    – If wedgeOri is equal to 0 or 4, the following applies:
      – The variables sizeScaleS and sizeScaleE are derived as follows:

$$sizeScaleS = (log2BlkSize > 3) ? 2 : 1 \quad (I-1)$$

$$sizeScaleE = (wedgeOri < 4 \ \&\& \ log2BlkSize > 3) ? : 2 : 1 \quad (I-2)$$

      – For m in the range of 0 to (wBlkSize / sizeScaleS − 1), inclusive, the following applies:
        – For n in the range of 0 to (wBlkSize / sizeScaleE − 1), inclusive, the following applies:
          – The wedgelet partition pattern generation process as specified in clause I.6.6.1 is invoked with patternSize equal to (1 << log2BlkSize), the variable resShift, the variable wedgeOri, the variable (xS, yS) equal to (m * sizeScaleS , 0), the variable (xE, yE) equal to (wedgeOri == 0) ? (0, n * sizeScaleE) : (n * sizeScaleE, wBlkSize − 1) as inputs, and the output is the partition pattern curWedgePattern.
          – The wedgelet partition pattern table insertion process as specified in clause I.6.6.2 is invoked with the variable log2BlkSize and the partition pattern curWedgePattern as inputs.
    – Otherwise (wedgeOri is equal to 1, 2, 3, or 5), the following applies:
      – For curPos in the range of posStart to posEnd − 1, inclusive, the following applies:
        – The partition pattern curWedgePattern[x][y] is derived as follows:

```
for( y = 0; y < ( 1 << log2BlkSize ); y++ )
    for( x = 0; x < ( 1 << log2BlkSize ); x++ )                                  (I-3)
        curWedgePattern[ x ][ y ] = 1 −
            WedgePatternTable[ log2BlkSize ][ curPos ][ y ][ ( 1 << log2BlkSize ) − 1 − x ]
```

    – The variable posStart is set equal to posEnd.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/7452bd56-3678-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

– NumWedgePattern[5] is set equal to NumWedgePattern[4].

– For k = 0..NumWedgePattern[5] − 1, the following applies:

  – For x, y = 0..(1 << 5) − 1, the following applies:

$$WedgePatternTable[5][k][x][y] = WedgePatternTable[4][k][x >> 1][y >> 1] \quad (I-4)$$

I.6.6.1 Wedgelet partition pattern generation process

Inputs to this process are:

– a variable patternSize specifying the partition pattern size,

– a variable resShift specifying the precision of the partition pattern start and end locations relative to patternSize,

– a variable wedgeOri specifying the orientation of the partition pattern,

– a location (xS, yS) specifying the boundary start of a sub-block partition,

– a location (xE, yE) specifying the boundary end of a sub-block partition.

Output of this process is the partition pattern wedgePattern[x][y] of size (patternSize)x(patternSize).

The values of the partition pattern wedgePattern[x][y] are derived as specified by the following ordered steps:

1. For x, y = 0..patternSize − 1, wedgePattern[x][y] is set equal to 0.

2. The samples of the partition pattern wedgePattern that form a line between (xS, yS) and (xE, yE) are set equal to 1 as follows:

```
( x0, y0 ) = ( xS, yS )
( x1, y1 ) = ( xE, yE )
if( abs( yE − yS ) > abs( xE − xS ) ) {
    ( x0, y0 ) = Swap( x0, y0 )
    ( x1, y1 ) = Swap( x1, y1 )
}
if( x0 > x1 ) {
    ( x0, x1 ) = Swap( x0, x1 )
    ( y0, y1 ) = Swap( y0, y1 )
}
sumErr = 0                                                    (I-5)
posY = y0
for( posX = x0; posX <= x1; posX++ ) {
    if( abs( yE − yS ) > abs( xE − xS ) )
        wedgePattern[ posY >> resShift ][ posX >> resShift ] = 1
    else
        wedgePattern[ posX >> resShift ][ posY >> resShift ] = 1
    sumErr += ( abs( y1 − y0 ) << 1 )
    if( sumErr >= ( x1 − x0 ) ) {
        posY += ( y0 < y1 ) ? 1 : −1
        sumErr −= ( x1 − x0 ) << 1
    }
}
```

3. The samples of wedgePattern are modified as follows:

```
for( y = 0; y <= ( yE >> resShift ); y++ )
    for( x = 0; ( x <= patternSize − 1 ) && ( wedgePattern[ x ][ y ] == 0 ); x++ )   (I-6)
        wedgePattern[ x ][ y ] = 1
```

I.6.6.2 Wedgelet partition pattern table insertion process

Inputs to this process are:

– a variable log2BlkSize specifying the partition pattern size,

– a partition pattern wedgePattern[x][y], with x, y = 0..(1 << log2BlkSize) − 1.

The variable validPatternFlag is set equal to 0 and the following applies:

1. For x, y = 0..(1 << log2BlkSize) − 1, the following applies:

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d36-5878-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

– When wedgePattern[x][y] is not equal to wedgePattern[0][0], validPatternFlag is set equal to 1.

2. For k = 0..NumWedgePattern[log2BlkSize] − 1, the following applies:
   – The variable patIdenticalFlag is set equal to 1.
   – For x, y = 0..(1 << log2BlkSize) − 1, the following applies:
     – When wedgePattern[x][y] is not equal to WedgePatternTable[log2BlkSize][k][x][y], patIdenticalFlag is set equal to 0.
   – When patIdenticalFlag is equal to 1, validPatternFlag is set equal to 0.
3. For k = 0..NumWedgePattern[log2BlkSize] − 1, the following applies:
   – The variable patInvIdenticalFlag is set equal to 1.
   – For x, y = 0..(1 << log2BlkSize) − 1, the following applies:
     – When wedgePattern[x][y] is equal to WedgePatternTable[log2BlkSize][k][x][y], patInvIdenticalFlag is set equal to 0.
   – When patInvIdenticalFlag is equal to 1, validPatternFlag is set equal to 0.

When validPatternFlag is equal to 1, the following applies:

– The pattern WedgePatternTable[log2BlkSize][NumWedgePattern[log2BlkSize]] is set equal to wedgePattern.
– The value of NumWedgePattern[log2BlkSize] is incremented by one.

I.7 Syntax and semantics

I.7.1 Method of specifying syntax in tabular form

The specifications in clause F.7.1 apply.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

I.7.2 Specification of syntax functions, categories, and descriptors

The specifications in clause F.7.2 apply.

ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d36-5878-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

I.7.3 Syntax in tabular form

I.7.3.1 NAL unit syntax

The specifications in clause F.7.3.1 and all its subclauses apply.

I.7.3.2 Raw byte sequence payloads and RBSP trailing bits syntax

I.7.3.2.1 Video parameter set RBSP

video_parameter_set_rbsp( ) {	**Descriptor**
vps_video_parameter_set_id	u(4)
vps_base_layer_internal_flag	u(1)
vps_base_layer_available_flag	u(1)
vps_max_layers_minus1	u(6)
vps_max_sub_layers_minus1	u(3)
vps_temporal_id_nesting_flag	u(1)
vps_reserved_0xffff_16bits	u(16)
profile_tier_level(1, vps_max_sub_layers_minus1)	
vps_sub_layer_ordering_info_present_flag	u(1)
for(i = (vps_sub_layer_ordering_info_present_flag ? 0 : vps_max_sub_layers_minus1); i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering_minus1[i]	ue(v)
vps_max_num_reorder_pics[i]	ue(v)
vps_max_latency_increase_plus1[i]	ue(v)
}	

vps_max_layer_id	u(6)
vps_num_layer_sets_minus1	ue(v)
for(i = 1; i <= vps_num_layer_sets_minus1; i++)	
for(j = 0; j <= vps_max_layer_id; j++)	
layer_id_included_flag[i][j]	u(1)
vps_timing_info_present_flag	u(1)
if(vps_timing_info_present_flag) {	
vps_num_units_in_tick	u(32)
vps_time_scale	u(32)
vps_poc_proportional_to_timing_flag	u(1)
if(vps_poc_proportional_to_timing_flag)	
vps_num_ticks_poc_diff_one_minus1	ue(v)
vps_num_hrd_parameters	ue(v)
for(i = 0; i < vps_num_hrd_parameters; i++) {	
hrd_layer_set_idx[i]	ue(v)
if(i > 0)	
cprms_present_flag[i]	u(1)
hrd_parameters(cprms_present_flag[i], vps_max_sub_layers_minus1)	
}	
}	
vps_extension_flag	u(1)
if(vps_extension_flag) {	
while(!byte_aligned( ))	
vps_extension_alignment_bit_equal_to_one	u(1)
vps_extension( )	
vps_extension2_flag	u(1)
if(vps_extension2_flag) {	
vps_3d_extension_flag	u(1)
if(vps_3d_extension_flag) {	
while(!byte_aligned( ))	
vps_3d_extension_alignment_bit_equal_to_one	u(1)
vps_3d_extension( )	
}	
vps_extension3_flag	u(1)
if(vps_extension3_flag)	
while(more_rbsp_data( ))	
vps_extension_data_flag	u(1)
}	
}	
rbsp_trailing_bits( )	
}	

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 23008-2:2013/FDAmd 4
https://standards.iteh.ai/catalog/standards/sist/74523d36-5878-4565-9f89-23185387e92a/iso-iec-23008-2-2013-fdamd-4

I.7.3.2.1.1 Video parameter set extension syntax

The specifications in clause F.7.3.2.1.1 apply.

I.7.3.2.1.2 Representation format syntax

The specifications in clause F.7.3.2.1.2 apply.