



Title: **MV-HEVC Draft Text 2**

Status: Output Document of JCT-3V

Purpose: Working Draft of MV-HEVC based on JCTVC-J1003_d7

Author(s) or Contact(s):

Gerhard Tech	Email: gerhard.tech@hhi.fraunhofer.de
Fraunhofer HHI	
Krzysztof Wegner	Email: kwegner@multimedia.edu.pl
Poznan University of Technology	
Ying Chen	Email: chen@qti.qualcomm.com
Qualcomm Incorporated	
Miska Hannuksela	Email: miska.hannuksela@nokia.com
Nokia Corporation	

Source: Editor

ABSTRACT

Working Draft 2 of MV-HEVC text specification based on **JCTVC-J1003_d7**.

In text blocks copied from JCTVC-J1003_d7 changes related to MV-HEVC are **highlighted**.

Ed. Notes (WD2) (based on JCT3V-A1004)

- (MVC-MH) Review, typo corrections, editorial improvement, and editor's notes
- (MVE-06) Incorporated introductory paragraph for view dependency change SEI message.
- (MVE-05) Incorporated invocation of sub-bitstream extraction process in general decoding process
- (MVE-04) Fixed construction of layerId list in general decoding process
- (MVC-KW) Review, typo corrections, editorial improvement.
- (MVE-03) Replacement of changes marks related to base spec by highlighting
- (MVE-01/JCT3V-B00046) Incorporated editorial note
- (MVC-GT) Review, typo corrections, editorial improvement.
- (MVC-CY) Review, typo corrections, editorial improvement.
- (MVE-02) Incorporated initial version of HRD text.
- (MVN-01/JCT3V-B0063) Incorporated view dependency change SEI.

Ed. Notes (WD1) (based on : JCT3V-A0012)

- (Rev3, KW) Review and small corrections
- (Rev2, GT) Review and text improvement
- Missing part in general decoding process
- (Replacement view_id by layer_id)
- Font issue fix.
- (Fix: picture marking)
- (Rev1, CY), Review and small corrections
- (MV07) Fix references
- (MV06) Improvement and update of interview prediction text.
- (MV02,MV03) Update of high level syntax and definitions
- (MV09) general HEVC decoding process
- (MV08) Additional sections/placeholders
- (MV04, MV05) Removal of low-level and depth tools
- (MV01) Removed HEVC spec
- Update of low level specification to match HEVC text specification 8(d7)

CONTENTS

	<i>Page</i>
Abstract	1
Contents	2
List of figures	3
List of tables	3
Annex F Multiview high efficient video coding	2
F.1 Scope	2
F.2 Normative references	2
F.3 Definitions	2
F.4 Abbreviations	3
F.5 Conventions	3
F.6 Source, coded, decoded and output data formats, scanning processes, and neighbouring relationships	3
F.7 Syntax and semantics	3
F.7.1 Method of specifying syntax in tabular form	3
F.7.2 Specification of syntax functions, categories, and descriptors	3
F.7.3 Syntax in tabular form	4
F.7.3.1 NAL unit syntax	4
F.7.3.2 Raw byte sequence payloads and RBSP trailing bits syntax	4
F.7.3.3 Profile, tier and level syntax	6
F.7.3.4 Operation point syntax	6
F.7.3.5 Slice header syntax	6
F.7.3.6 Slice data syntax	6
F.7.3.7 Coding tree unit syntax	6
F.7.3.8 Coding quadtree syntax	6
F.7.3.9 Coding unit syntax	6
F.7.3.10 Prediction unit syntax	6
F.7.3.11 Transform tree syntax	6
F.7.3.12 Transform coefficient syntax	6
F.7.3.13 Residual coding syntax	6
F.7.4 Semantics	7
F.7.4.1 NAL unit semantics	7
F.7.4.2 Raw byte sequence payloads and RBSP trailing bits semantics	7
F.7.4.3 Profile, tier and level semantics	8
F.7.4.4 Operation point semantics	8
F.7.4.5 Slice header semantics	8
F.7.4.6 Slice data semantics	8
F.7.4.7 Coding tree unit semantics	9
F.7.4.8 Coding quadtree semantics	9
F.7.4.9 Coding unit semantics	9
F.7.4.10 Prediction unit semantics	9
F.7.4.11 Transform tree semantics	9
F.7.4.12 Transform coefficient semantics	9
F.7.4.13 Residual coding semantics	9
F.8 MV-HEVC Decoding process	10
F.8.1 General decoding process	10
F.8.2 NAL unit decoding process	10
F.8.3 Slice decoding process	11
F.8.3.1 Decoding process for picture order count	11
F.8.3.2 Decoding process for reference picture set	11
F.8.3.3 Decoding process for generating unavailable reference pictures	15
F.8.3.4 Decoding process for reference picture lists construction	15
F.8.4 Decoding process for coding units coded in intra prediction mode	16
F.8.5 Decoding process for coding units coded in inter prediction mode	16
F.8.6 Scaling, transformation and array construction process prior to deblocking filter process	16
F.8.7 In-loop filter process	16

MV-HEVC

F.9	Parsing process	16
F.10	Specification of bitstream subsets.....	16
F.11	Profiles and levels	16
F.12	Byte stream format.....	16
F.13	MV-HEVC hypothetical reference decoder.....	16
F.13.1	General.....	16
F.13.2	Operation of the decoded picture buffer (DPB)	16
F.13.2.1	Removal of pictures from the DPB.....	17
F.13.2.2	Picture output.....	17
F.13.2.3	Current decoded picture marking and storage	17
F.13.3	Bitstream conformance	17
F.13.4	Decoder conformance	17
F.13.4.1	Operation of the output order DPB	17
F.13.4.2	Removal of pictures from the DPB.....	17
F.13.4.3	Picture decoding, marking and storage	17
F.14	MV-HEVC SEI messages.....	17
F.14.1	SEI message syntax.....	18
F.14.1.1	View dependency change SEI message syntax.....	18
F.14.2	SEI message semantics	18
F.14.2.1	View dependency change SEI message semantics	18
F.15	Video usability information	18

LIST OF FIGURES

No table of figures entries found.

LIST OF TABLES

No table of figures entries found.

Modifications of HEVC specification:

In Foreword replace paragraph that start with "In this Recommendation | International Standard Annexes":

In this Recommendation | International Standard Annexes **A** through F contain normative requirements and are an integral part of this Recommendation | International Standard.

In 0.7, add the following paragraph after the paragraph that starts with "Annex A":

Annex F specifies multiview high efficient video coding, referred to as MV-HEVC. The reader is referred to Annex F for the entire decoding process for MV-HEVC, which is specified there with references being made to clauses **2-9** and Annexes **A-E**.

Annex F

Multiview high efficient video coding

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

This annex specifies multiview high efficient video coding, referred to as MV-HEVC.

F.1 Scope

Bitstreams and decoders conforming to the profile specified in this annex are completely specified in this annex with reference made to clauses 2-9 and Annexes A-E.

F.2 Normative references

The specifications in clause 2 apply.

F.3 Definitions

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

- F.3.1 access unit:** A set of *NAL units* that are consecutive in *decoding order* and contain exactly one *coded picture* consisting of one or more *view components*. In addition to the *coded slice NAL units* of the *coded picture*, the access unit may also contain other *NAL units* not containing *slices* of the *coded picture*. The decoding of an access unit always results in one *decoded picture* consisting of one or more *decoded view components*.
- F.3.2 base view:** A *view* that has *view order index* and *layer_id* equal to 0 in a *coded video sequence*. The base view can be decoded independently of other *views*, does not use *inter-view prediction*. The *bitstream subset* corresponding to the base view conforms to one or more of the *profiles* specified in Annex A. There is only one base view in a *coded video sequence*.
- F.3.3 bitstream subset:** A *bitstream* that is derived as a *subset* from a *bitstream* by discarding zero or more *NAL units*. A *bitstream subset* is also referred to as a *sub-bitstream*.
- F.3.4 broken link access (BLA) unit:** An *access unit* in which all *view components* are *BLA view components*.
- F.3.5 broken link access (BLA) view component:** A *view component* for which each *slice* has *nal_unit_type* equal to *BLA_W_TFD*, *BLA_W_DLP* or *BLA_N_LP*.
- F.3.6 clean random access (CRA) access unit:** An *access unit* in which all the *view components* are *CRA view components*.
- F.3.7 clean random access (CRA) view component:** A *view component* for which each *slice* has *nal_unit_type* equal to *CRA_NUT*.
- F.3.8 decoded view component:** A decoded view component is derived by decoding a *view component*.
- F.3.9 instantaneous decoding refresh (IDR) access unit:** An *access unit* in which all the *view components* are *IDR view components*.
- F.3.10 instantaneous decoding refresh (IDR) view component:** A *view component* for which each *slice* has *nal_unit_type* equal to *IDR_W_LP* or *IDR_N_LP*.
- F.3.11 inter-view coding:** Coding of a *block*, *treeblock*, *slice*, or *picture* that uses *inter-view prediction*.
- F.3.12 inter-view prediction:** A *prediction* derived from only data elements (e.g. sample value or motion vector) of *inter-view references* for decoding another *view component* in the same *access unit*.
- F.3.13 inter-view reference component:** A *view component* containing samples that may be used for *inter-view prediction* in the *decoding process* of subsequent *view components* in *decoding order*.
- F.3.14 inter-view reference picture:** A *decoded view component* of an *inter-view reference component*. An *inter-*

view reference picture contains samples that may be used for *inter-view prediction* in the *decoding process* of subsequent *pictures* in *decoding order*. Inter-view reference pictures are *reference pictures*.

- F.3.15 inter-view subset reference picture set:** A set of *reference pictures* associated with a *view component*, consisting of all *inter-view reference components* that may be used for *inter-view prediction* of the associated *picture* or any *picture* following the associated *picture* in *decoding order*.
- F.3.16 non-RAP access unit:** An *access unit* that is not a random access point *access unit*.
- F.3.17 non-RAP view component:** A *view component* that is not a *random access point view component*.
- F.3.18 non-base view:** A *view* that is not the *base view*. *VCL NAL units* of a non-base view have *view order index* larger than 0. Decoding of *view components* in a non-base view may require the use of *inter-view prediction*.
- F.3.19 random access point access unit:** An *access unit* in which the *coded picture* is a *RAP picture*.
- F.3.20 random access point (RAP) view component:** A *view component* containing only *slices* and for which each *slice* has *nal_unit_type* in the range of 7 to 12, inclusive. A random access point view component is not predicted from *inter prediction*.
- F.3.21 reference picture list:** A list of reference pictures and *inter-view reference components* that are used for inter prediction or inter-view prediction of a P or B slice. For the decoding process of a P slice, there is one reference picture list. For the decoding process of a B slice, there are two reference picture lists.
- F.3.22 sub-bitstream:** A *subset* of a *bitstream*. A *sub-bitstream* is also referred to as a *bitstream subset*.
- F.3.23 subset:** A subset contains only elements that are also contained in the set from which the subset is derived. The subset may be identical to the set from which it is derived.
- F.3.24 target output view:** A *view* that is to be output. The target output views are either indicated by external means or, when not indicated by external means, the target output view is the *base view*.
- NOTE – The output views may be requested by a receiver and may be negotiated between the receiver and the sender.
- F.3.25 view:** A sequence of *view components* associated with an identical value of *view order index*.
- F.3.26 view component:** A coded *representation* of a *view* in a single *access unit*.
- F.3.27 view order index:** An index that indicates the *decoding order* of *view components* in an *access unit*.

F.4 Abbreviations

The specification in clause 4 apply.

F.5 Conventions

The specification in clause 5 apply.

F.6 Source, coded, decoded and output data formats, scanning processes, and neighbouring relationships

The specification in clause 6 apply.

F.7 Syntax and semantics

This clause specifies syntax and semantics for coded video sequences that conform to one or more of the profiles specified in this annex.

F.7.1 Method of specifying syntax in tabular form

The specifications in subclause 7.1 apply.

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

The specifications in subclause 7.2 apply.

F.7.3 Syntax in tabular form

F.7.3.1 NAL unit syntax

The specifications in subclause 7.3.1 apply.

F.7.3.1.1 General NAL unit syntax

The specifications in subclause 7.3.1.1 apply.

F.7.3.1.2 NAL unit header syntax

nal_unit_header() {	Descriptor
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
layer_id_nuh_reserved_zero_6bits	u(6)
nuh_temporal_id_plus1	u(3)
}	

F.7.3.2 Raw byte sequence payloads and Rbsp trailing bits syntax

F.7.3.2.1 Video parameter set Rbsp

[Ed. (MH): The video_parameter_set_rbsp() structure below is not aligned with the one in output document JCTVC-K1007. It is presumably desirable to achieve such alignment in future versions of the MV-HEVC specification.]

video_parameter_set_rbsp() {	Descriptor
video_parameter_set_id	u(4)
vps_temporal_id_nesting_flag	u(1)
vps_reserved_zero_2bits	u(2)
vps_max_layers_minus1_reserved_zero_6bits	u(6)
vps_max_sub_layers_minus1	u(3)
profile_and_tier_level(1, vps_max_sub_layers_minus1)	
// [Ed. (MH): to be added from HEVC draft text 9:] bit_rate_pic_rate_info(0, vps_max_sub_layers_minus1)	
vps_extension_offset_vps_reserved_zero_12bits	u(12)
for(i = 0; i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering[i]	ue(v)
vps_max_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
// [Ed. (MH): The adoption of JCTVC-K0180 should be reflected here as soon as there is HEVC draft text available.]	
vps_num_hrd_parameters	ue(v)
for(i = 0; i < vps_num_hrd_parameters; i++) {	
if(i > 0)	
op_point(i)	
hrd_parameters(i == 0, vps_max_sub_layers_minus1)	
}	
bit_equal_to_one	f(1)
vps_extension()	
vps_extension2_flag	u(1)
if(vps_extension2_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)

rbsp_trailing_bits()	
}	

F.7.3.2.1.1 Video parameter set extension syntax

[Ed. (MH): The vps_extension() structure below is not aligned with the one in output document JCTVC-K1007. It is presumably desirable to achieve such alignment in future versions of the MV-HEVC specification.]

vps_extension() {	Descriptor
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
num_additional_layer_operation_points	u(8)
num_additional_profile_level_sets	u(8)
for(i = 0; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
num_types_zero_4bits[i]	u(4)
type_zero_4bits[i]	u(4)
view_id[i]	u(8)
[Ed. (CY): In annex F, we probably want to ONLY support num_dimensions_minus1 equal to 0 and dimension_type is view id. We may just put reserved_8_zero_bits u(8) view_id, u(8) to make the syntax much cleaner].	
if (i > 0)	
num_direct_ref_layers[i]	u(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
for(i = 0; i <= num_additional_profile_level_sets; i++)	
profile_tier_level(1, vps_max_sub_layers_minus1)	
for(i = 0; i < num_additional_layer_operation_points; i++) {	
op_point(i)	
if (num_additional_profile_level_sets > 0)	
profile_level_idx[i]	u(8)
}	
}	

F.7.3.2.2 Sequence parameter set RBSP syntax

The specifications in subclause 7.3.2.2 apply.

F.7.3.2.3 Picture parameter set RBSP syntax

The specifications in subclause 7.3.2.3 apply.

F.7.3.2.4 Scaling list data syntax

The specifications in subclause 7.3.2.4 apply.

F.7.3.2.5 Scaling list syntax

The specifications in subclause 7.3.2.5 apply.

F.7.3.2.6 Supplemental enhancement information RBSP syntax

The specifications in subclause 7.3.2.6 apply.

F.7.3.2.7 Access unit delimiter RBSP syntax

The specifications in subclause 7.3.2.7 apply.

F.7.3.2.8 End of sequence RBSP syntax

The specifications in subclause 7.3.2.8 apply.

F.7.3.2.9 End of bitstream RBSP syntax

The specifications in subclause 7.3.2.9 apply.

F.7.3.2.10 Filler data RBSP syntax

The specifications in subclause 7.3.2.10 apply.

F.7.3.2.11 Slice layer RBSP syntax

The specifications in subclause 7.3.2.11 apply.

F.7.3.2.12 RBSP slice trailing bits syntax

The specifications in subclause 7.3.2.12 apply.

F.7.3.2.13 RBSP trailing bits syntax

The specifications in subclause 7.3.2.13 apply.

F.7.3.2.14 Byte alignment syntax

The specifications in subclause 7.3.2.14 apply.

F.7.3.3 Profile, tier and level syntax

The specifications in subclause 7.3.3 apply.

F.7.3.4 Operation point syntax

The specifications in subclause 7.3.4 apply.

F.7.3.5 Slice header syntax

The specifications in subclause 7.3.5 apply.

F.7.3.6 Slice data syntax

The specifications in subclause 7.3.6 apply.

F.7.3.7 Coding tree unit syntax

The specifications in subclause 7.3.7 apply.

F.7.3.8 Coding quadtree syntax

The specifications in subclause 7.3.8 apply.

F.7.3.9 Coding unit syntax

The specifications in subclause 7.3.9 apply.

F.7.3.10 Prediction unit syntax

The specifications in subclause 7.3.10 apply.

F.7.3.11 Transform tree syntax

The specifications in subclause 7.3.11 apply.

F.7.3.12 Transform coefficient syntax

The specifications in subclause 7.3.12 apply.

F.7.3.13 Residual coding syntax

The specifications in subclause 7.3.13 apply.

F.7.4 Semantics

F.7.4.1 NAL unit semantics

F.7.4.1.1 General NAL unit semantics

The specifications in subclause 7.4.1.1 apply.

F.7.4.1.2 NAL unit header semantics

The specifications in subclause 7.4.1.2 apply with following modifications and additions.

layer_id identifies the layer the NAL unit belongs to. **layer_id** shall be equal to 0 for any NAL unit of the base view. View order index of each NAL unit is derived to be equal to **layer_id**.

F.7.4.1.3 Encapsulation of an SODB within an RBSP (informative)

The specifications in subclause 7.4.1.3 apply.

F.7.4.1.4 Order of NAL units and association to coded pictures, access units, and video sequences

The specifications in subclause 7.4.1.4 apply.

F.7.4.2 Raw byte sequence payloads and RBSP trailing bits semantics

F.7.4.2.1 Video parameter set RBSP semantics

The specifications in subclause 7.4.2.1 apply with following modifications and additions.

vps_max_layers_minus1 plus 1 specifies the maximum number of layers that may be present in the bitstream. The value of **vps_max_layers_minus1** shall be in the range of 0 to 63, inclusive.

vps_extension_offset specifies the byte offset of the next set of fixed-length coded information in the video parameter set NAL unit, starting from the beginning of the NAL unit.

NOTE – Video parameter set information for non-base layer or view starts from a byte-aligned position of the video parameter set NAL unit, with fixed-length coded information that is essential for session negotiation and/or capability exchange. The byte offset helps to locate and access those essential information in the video parameter set NAL unit without the need of entropy decoding, which may not be equipped with some network entities that may desire to access only the information in the video parameter set that is essential for session negotiation and/or capability exchange.

bit_equal_to_one shall be equal to 1.

vps_extension2_flag equal to 0 specifies that no **vps_extension_data_flag** syntax elements are present in the video parameter set RBSP syntax structure. **vps_extension2_flag** shall be equal to 1 in bitstreams conforming to annex 0 of this Recommendation | International Standard. The value of 1 for **vps_extension2_flag** is reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore all data that follow the value 1 for **vps_extension2_flag** in a video parameter set NAL unit.

F.7.4.2.1.1 Video parameter set extension semantics

vps_extension_byte_alignment_reserved_one_bit shall be equal to 1.

num_additional_layer_operation_points specifies the maximum number of additional set of **layer_id** values for operation points present in the coded video sequences the video parameter set applies. [Ed. Note (CY): to update the syntax to be aligned with the VPS design in the HEVC base specification.]

num_additional_profile_level_sets specifies the number of additional sets of profile and level information present in the video parameter set.

num_types_zero_4bits[*i*] shall be equal to 0 and it indicates only one dimension type is signalled for the layer with **layer_id** equal to *i*.

type_zero_4bits[*i*] shall be equal to 0 and it indicates only the dimension type with index equal to 0 is signalled for the layer with **layer_id** equal to *i*. The dimension type with index equal to 0 is view identifier.

view_id[*i*] specifies the view identifier of the layer with **layer_id** equal to *i*.

num_direct_ref_layers[*i*] specifies the number of layers the *i*-th layer directly depends on. The inter-view subset reference picture set of the view components with **layer_id** equal to *i* contains **num_direct_ref_layers**[*i*] entries. When not present, the value of **num_direct_ref_layers**[*i*] is inferred to be equal to 0.

ref_layer_id[*i*][*j*] identifies the *j*-th layer the *i*-th layer directly depends on. The *j*-th entry of the inter-view subset

reference picture set of the view components with layer_id equal to i is identified by ref_layer_id[i][j].

profile_level_idx[i] specifies the index to the profile_tier_level syntax tables defined in this video parameter set. When profile_level_idx[i] is equal to 0, the profile_tier_level to be used for the current operation point is signalled in the video parameter set, when profile_level_idx[i] is not equal to 0, the profile_tier_level to be used for the current operation point is signalled in the video parameter set extension syntax.

F.7.4.2.2 Sequence parameter set RBSP semantics

The specifications in subclause 7.4.2.2 apply.

F.7.4.2.3 Picture parameter set RBSP semantics

The specifications in subclause 7.4.2.3 apply.

F.7.4.2.4 Scaling list data semantics

The specifications in subclause 7.4.2.4 apply.

F.7.4.2.5 Scaling list semantics

The specifications in subclause 7.4.2.5 apply.

F.7.4.2.6 Supplemental enhancement information RBSP semantics

The specifications in subclause 7.4.2.6 apply.

F.7.4.2.7 Access unit delimiter RBSP semantics

The specifications in subclause 7.4.2.7 apply.

F.7.4.2.8 End of sequence RBSP semantics

The specifications in subclause 7.4.2.8 apply.

F.7.4.2.9 End of bitstream RBSP semantics

The specifications in subclause 7.4.2.9 apply.

F.7.4.2.10 Filler data RBSP semantics

The specifications in subclause 7.4.2.10 apply.

F.7.4.2.11 Slice layer RBSP semantics

The specifications in subclause 7.4.2.11 apply.

F.7.4.2.12 RBSP slice trailing bits semantics

The specifications in subclause 7.4.2.12 apply.

F.7.4.2.13 RBSP trailing bits semantics

The specifications in subclause 7.4.2.13 apply.

F.7.4.2.14 Byte alignment semantics

The specifications in subclause 7.4.2.14 apply.

F.7.4.3 Profile, tier and level semantics

The specifications in subclause 7.4.3 apply.

F.7.4.4 Operation point semantics

The specifications in subclause 7.4.4 apply.

F.7.4.5 Slice header semantics

The specifications in subclause 7.4.5 apply.

F.7.4.6 Slice data semantics

The specifications in subclause 7.4.6 apply.

F.7.4.7 Coding tree unit semantics

The specifications in subclause 7.4.7 apply.

F.7.4.8 Coding quadtree semantics

The specifications in subclause 7.4.8 apply.

F.7.4.9 Coding unit semantics

The specifications in subclause 7.4.9 apply.

F.7.4.10 Prediction unit semantics

The specifications in subclause 7.4.10 apply.

F.7.4.11 Transform tree semantics

The specifications in subclause 7.4.11 apply.

F.7.4.12 Transform coefficient semantics

The specifications in subclause 7.4.12 apply.

F.7.4.13 Residual coding semantics

The specifications in subclause 7.4.13 apply.

F.8 MV-HEVC Decoding process

F.8.1 General decoding process

[Ed. (MH): Align this subclause with subclause 8.1 of the HEVC draft text as soon as the adopted contribution JVTVC-K0126 has been included in the HEVC draft text.]

This subclause specifies the decoding process for an access unit of a coded video sequence conforming to one or more of the profiles specified in Annex F. Specifically, this subclause specifies how the decoded picture with multiple view components is derived from syntax elements and global variables that are derived from NAL units in an access unit when the decoder is decoding the operation point identified by the target temporal level and the target output views.

The decoding process is specified such that all decoders shall produce numerically identical results for the target output views. Any decoding process that produces identical results for the target output views to the process described here conforms to the decoding process requirements of this Recommendation | International Standard.

Unless stated otherwise, the syntax elements and derived upper-case variables that are referred to by the decoding process specified in this subclause and all child processes invoked from the process specified in this subclause are the syntax elements and derived upper-case variables for the current access unit.

The target output views are either specified by external means not specified in this Specification, or, when not specified by external means, there shall be one target output view which is the base view, with layer_id equal to 0. Let OutputLayerIdList be the list of layer_id values, in increasing order of layer_id, of all the target output layers in one access unit. The list OutputLayerIdList shall not change within a coded video sequence.

The list layerIdList, which specifies the set of layer_id values of VCL NAL units to be decoded, is specified to contain the OutputLayerIdList and the values of ref_layer_id[i][j] for all non-zero values of i among OutputLayerIdList and for value j in the range of 0 to num_direct_ref_layers[i] – 1, inclusive.

The variable HighestTid, which identifies the highest temporal sub-layer, is specified as follows:

- If some external means not specified in this Specification is available to set the variable HighestTid to a value, HighestTid is set to the value provided by the external means.
- Otherwise, HighestTid is set to sps_max_sub_layers_minus1.

The sub-bitstream extraction process as specified in subclause 10.1 is applied with layerIdList and HighestTid as inputs and the output assigned to a bitstream referred to as BitstreamToDecode.

The multiview high efficient video decoding process specified in this subclause is repeatedly invoked for each view component with layer_id among layerIdList in increasing order of layer_id.

Outputs of the multiview high efficient video decoding process are decoded samples of the current picture including all decoded view components.

For each view component, the specifications in clause 8.1 apply, with the following modifications. [Ed. (MH): The entire clause 8.1 of HEVC draft text 8 (JCTVC-J1003_d7) or draft text 9 (JCTVC-K1003_v7) is not as such applicable to MV-HEVC decoding. It is suggested that clause 8.1 is split into two clauses, a first one dealing with bitstream-level decoding operations and a second one dealing with view component level decoding operations. Then, the bitstream-level decoding operations can be specified in F.8.1, while the view component level decoding operations can be repetitively performed for each view component among layerIdList.]

- All invocations of the process specified in subclause 8.3.2 are replaced with invocations of the process specified in subclause F.8.3.2.
- All invocations of the process specified in subclause 8.3.3.1 are replaced with invocations of the process specified in subclause F.8.3.3.1.
- All invocations of the process specified in subclause 8.3.4.1 are replaced with invocations of the process specified in subclause F.8.3.4.1.

[Ed. (GT): Subclauses 8.3.3.1 and 8.3.4.1 are incorrectly referenced as 8.3.3 and 8.3.4 in subclause 8.1 of HEVC spec 8, d7]

F.8.2 NAL unit decoding process

The specifications in subclause 8.2 apply.

F.8.3 Slice decoding process

F.8.3.1 Decoding process for picture order count

The specifications in subclause 8.3.1 apply.

F.8.3.2 Decoding process for reference picture set

This process is invoked once per picture, after decoding of a slice header but prior to the decoding of any coding unit and prior to the decoding process for reference picture list construction of the slice as specified in subclause F.8.3.3.1. The process may result in marking one or more reference pictures as "unused for reference" or "used for long-term reference".

NOTE 1 – The reference picture set is an absolute description of the reference pictures used in the decoding process of the current and future coded pictures. The reference picture set signalling is explicit in the sense that all reference pictures included in the reference picture set are listed explicitly and there is no default reference picture set construction process in the decoder that depends on the status of the decoded picture buffer.

Short-term reference pictures are identified within a view by their PicOrderCntVal values. Long-term reference pictures are identified within a view either by their PicOrderCntVal values or their pic_order_cnt_lsb values. Inter-view reference pictures are identified further by their layer_id values.

Five lists of picture order count values are constructed to derive the reference picture set; PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, and PocLtFoll with NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, and NumPocLtFoll number of elements, respectively.

If the current picture is an IDR or BLA picture, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, and PocLtFoll are all set to empty, and NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, and NumPocLtFoll are all set to 0.

- Otherwise, the following applies for derivation of the five lists of picture order count values and the numbers of entries.

```

for( i = 0, j = 0, k = 0; i < NumNegativePics[ StRpsIdx ]; i++ )
    if( UsedByCurrPicS0[ StRpsIdx ][ i ] )
        PocStCurrBefore[ j++ ] = PicOrderCntVal + DeltaPocS0[ StRpsIdx ][ i ]
    else
        PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS0[ StRpsIdx ][ i ]
NumPocStCurrBefore = j

for( i = 0, j = 0; i < NumPositivePics[ StRpsIdx ]; i++ )
    if( UsedByCurrPicS1[ StRpsIdx ][ i ] )
        PocStCurrAfter[ j++ ] = PicOrderCntVal + DeltaPocS1[ StRpsIdx ][ i ]
    else
        PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS1[ StRpsIdx ][ i ]
NumPocStCurrAfter = j
NumPocStFoll = k
for( i = 0, j = 0, k = 0; i < num_long_term_sps + num_long_term_pics; i++ ) {
    pocLt = PocLsbLt[ i ]
    if( delta_poc_msb_present_flag[ i ] )
        pocLt += PicOrderCntVal - DeltaPocMSBCycleLt[ i ] * MaxPicOrderCntLsb - pic_order_cnt_lsb
    if( UsedByCurrPicLt[ i ] ) {
        PocLtCurr[ j ] = pocLt
        CurrDeltaPocMsbPresentFlag[ j++ ] = delta_poc_msb_present_flag[ i ]
    } else {
        PocLtFoll[ k ] = pocLt
        FollDeltaPocMsbPresentFlag[ k++ ] = delta_poc_msb_present_flag[ i ]
    }
}
NumPocLtCurr = j
NumPocLtFoll = k
    
```

(F-1)

where PicOrderCntVal is the picture order count of the current picture as specified in subclause 8.3.1.

NOTE 2 – A value of StRpsIdx in the range from 0 to num_short_term_ref_pic_sets – 1, inclusive, indicates that a short-term reference picture set from the active sequence parameter set is being used, where StRpsIdx is the index of the short-term reference picture set to the list of short-term reference picture sets in the order in which they are signalled in the

sequenceparameter set. $StRpsIdx$ equal to $num_short_term_ref_pic_sets$ indicates that a short-term reference picture set explicitly signalled in the slice header is being used.

For each i in the range of 0 to $NumPocLtCurr - 1$, inclusive, when $CurrDeltaPocMsbPresentFlag[i]$ is equal to 1, it is a requirement of bitstream conformance that the following conditions apply:

- There shall be no j in the range of 0 to $NumPocStCurrBefore - 1$, inclusive, for which $PocLtCurr[i]$ is equal to $PocStCurrBefore[j]$.
- There shall be no j in the range of 0 to $NumPocStCurrAfter - 1$, inclusive, for which $PocLtCurr[i]$ is equal to $PocStCurrAfter[j]$.
- There shall be no j in the range of 0 to $NumPocStFoll - 1$, inclusive, for which $PocLtCurr[i]$ is equal to $PocStFoll[j]$.

For each i in the range of 0 to $NumPocLtFoll - 1$, inclusive, when $FollDeltaPocMsbPresentFlag[i]$ is equal to 1, it is a requirement of bitstream conformance that the following conditions apply:

- There shall be no j in the range of 0 to $NumPocStCurrBefore - 1$, inclusive, for which $PocLtFoll[i]$ is equal to $PocStCurrBefore[j]$.
- There shall be no j in the range of 0 to $NumPocStCurrAfter - 1$, inclusive, for which $PocLtFoll[i]$ is equal to $PocStCurrAfter[j]$.
- There shall be no j in the range of 0 to $NumPocStFoll - 1$, inclusive, for which $PocLtFoll[i]$ is equal to $PocStFoll[j]$.

For each i in the range of 0 to $NumPocLtCurr - 1$, inclusive, when $CurrDeltaPocMsbPresentFlag[i]$ is equal to 0, it is a requirement of bitstream conformance that the following conditions apply:

- There shall be no j in the range of 0 to $NumPocStCurrBefore - 1$, inclusive, for which $PocLtCurr[i]$ is equal to $(PocStCurrBefore[j] \& (MaxPicOrderCntLsb - 1))$.
- There shall be no j in the range of 0 to $NumPocStCurrAfter - 1$, inclusive, for which $PocLtCurr[i]$ is equal to $(PocStCurrAfter[j] \& (MaxPicOrderCntLsb - 1))$.
- There shall be no j in the range of 0 to $NumPocStFoll - 1$, inclusive, for which $PocLtCurr[i]$ is equal to $(PocStFoll[j] \& (MaxPicOrderCntLsb - 1))$.

For each i in the range of 0 to $NumPocLtFoll - 1$, inclusive, when $FollDeltaPocMsbPresentFlag[i]$ is equal to 0, it is a requirement of bitstream conformance that the following conditions apply:

- There shall be no j in the range of 0 to $NumPocStCurrBefore - 1$, inclusive, for which $PocLtFoll[i]$ is equal to $(PocStCurrBefore[j] \& (MaxPicOrderCntLsb - 1))$.
- There shall be no j in the range of 0 to $NumPocStCurrAfter - 1$, inclusive, for which $PocLtFoll[i]$ is equal to $(PocStCurrAfter[j] \& (MaxPicOrderCntLsb - 1))$.
- There shall be no j in the range of 0 to $NumPocStFoll - 1$, inclusive, for which $PocLtFoll[i]$ is equal to $(PocStFoll[j] \& (MaxPicOrderCntLsb - 1))$.

The variable $layerId$ is set equal to the $layer_id$ of the current view component. The list $LayerIdInterView$ specifying the layer identifiers of the inter-view subset reference picture set and the variable $NumIvCurr$ specifying the number of entries of $LayerIdInterView$, are derived as follows.

```
for( i = 0; i < num_direct_ref_layers[ layerId ]; i++ )
    LayerIdInterView[ i ] = ref_layer_id[ layerId ][ i ] (F-2)
```

```
NumIvCurr = num_direct_ref_layers[ layerId ] (F-3)
```

The reference picture set consists of ~~five~~ **six** lists of reference pictures; $RefPicSetStCurrBefore$, $RefPicSetStCurrAfter$, $RefPicSetStFoll$, $RefPicSetLtCurr$, ~~and~~ $RefPicSetLtFoll$, ~~and~~ $RefPicSetIvCurr$. The variable $NumPocTotalCurr$ is set equal to $NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr + NumIvCurr$. When decoding a P or B slice, it is a requirement of bitstream conformance that the value of $NumPocTotalCurr$ shall not be equal to 0.

NOTE 3 – $RefPicSetStCurrBefore$, $RefPicSetStCurrAfter$ and $RefPicSetLtCurr$ contains all reference pictures that may be used in inter prediction of the current picture and that may be used in inter prediction of one or more of the pictures following the current picture in decoding order. $RefPicSetStFoll$ and $RefPicSetLtFoll$ consists of all reference pictures that are *not* used in inter prediction of the current picture but may be used in inter prediction of one or more of the pictures following the current picture in decoding order.

A reference picture can be marked as "unused for reference", "used for short-term reference", or "used for long-term reference", but only one among these three. When a reference picture is referred to as being marked as "used for reference", this collectively refers to the picture being marked as "used for short-term reference" or "used for long-term reference" (but not both).

The derivation process for the reference picture set and picture marking are performed according to the following ordered steps, where DPB refers to the decoded picture buffer as described in Annex C and the variable `layerId` is set equal to the `layer_id` of the current view component.

1. The following applies:

```

for( i = 0; i < NumPocLtCurr; i++ )
    if( !CurrDeltaPocMsbPresentFlag[ i ] )
        if( there is a long-term reference picture picX in the DPB
            with layer_id equal to layerId and pic_order_cnt_lsb equal to PocLtCurr[ i ] )
            RefPicSetLtCurr[ i ] = picX
        else if( there is a short-term reference picture picY in the DPB
            with layer_id equal to layerId and pic_order_cnt_lsb equal to PocLtCurr[ i ] )
            RefPicSetLtCurr[ i ] = picY
        else
            RefPicSetLtCurr[ i ] = "no reference picture"
    else
        if( there is a long-term reference picture picX in the DPB
            with layer_id equal to layerId and PicOrderCntVal equal to PocLtCurr[ i ] )
            RefPicSetLtCurr[ i ] = picX
        else if( there is a short-term reference picture picY in the DPB
            with layer_id equal to layerId and PicOrderCntVal equal to PocLtCurr[ i ] )
            RefPicSetLtCurr[ i ] = picY
        else
            RefPicSetLtCurr[ i ] = "no reference picture"

```

(F-4)

```

for( i = 0; i < NumPocLtFoll; i++ )
    if( !FollDeltaPocMsbPresentFlag[ i ] )
        if( there is a long-term reference picture picX in the DPB
            with layer_id equal to layerId and pic_order_cnt_lsb equal to PocLtFoll[ i ] )
            RefPicSetLtFoll[ i ] = picX
        else if( there is a short-term reference picture picY in the DPB
            with layer_id equal to layerId and pic_order_cnt_lsb equal to PocLtFoll[ i ] )
            RefPicSetLtFoll[ i ] = picY
        else
            RefPicSetLtFoll[ i ] = "no reference picture"
    else
        if( there is a long-term reference picture picX in the DPB
            with layer_id equal to layerId and PicOrderCntVal to PocLtFoll[ i ] )
            RefPicSetLtFoll[ i ] = picX
        else if( there is a short-term reference picture picY in the DPB
            with layer_id equal to layerId and PicOrderCntVal equal to PocLtFoll[ i ] )
            RefPicSetLtFoll[ i ] = picY
        else
            RefPicSetLtFoll[ i ] = "no reference picture"

```

2. All reference pictures included in `RefPicSetLtCurr` and `RefPicSetLtFoll` are marked as "used for long-term reference".

3. The following applies:

```

for( i = 0; i < NumIvCurr; i++ )
    RefPicSetIvCurr[ i ] = "inter-view reference picture with layer_id equal to LayerIdInterView[ i ]"

```

4. All reference pictures included in `RefPicSetIvCurr` are marked as "used for long-term reference".

5. The following applies:

```

for( i = 0; i < NumPocStCurrBefore; i++ )
    if( there is a short-term reference picture picX in the DPB

```

```

with layer_id equal to layerId and
with PicOrderCntVal equal to PocStCurrBefore[ i ]
RefPicSetStCurrBefore[ i ] = picX
else
  RefPicSetStCurrBefore[ i ] = "no reference picture"

for( i = 0; i < NumPocStCurrAfter; i++ )
  if( there is a short-term reference picture picX in the DPB
with layer_id equal to layerId and
with PicOrderCntVal equal to PocStCurrAfter[ i ] )
  RefPicSetStCurrAfter[ i ] = picX
  else
    RefPicSetStCurrAfter[ i ] = "no reference picture"

for( i = 0; i < NumPocStFoll; i++ )
  if( there is a short-term reference picture picX in the DPB
with layer_id equal to layerId and
with PicOrderCntVal equal to PocStFoll[ i ] )
  RefPicSetStFoll[ i ] = picX
  else
    RefPicSetStFoll[ i ] = "no reference picture"

```

(F-5)

6. All reference pictures with layer_id values equal to layerId in the decoded picture buffer that are not included in RefPicSetLtCurr, RefPicSetLtFoll, RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetStFoll are marked as "unused for reference".

NOTE 4 – There may be one or more reference pictures that are included in the reference picture set but not present in the decoded picture buffer. Entries in RefPicSetStFoll or RefPicSetLtFoll that are equal to "no reference picture" should be ignored. Unless either of the following two conditions is true, an unintentional picture loss should be inferred for each entry in RefPicSetStCurrBefore, RefPicSetStCurrAfter and RefPicSetLtCurr that is equal to "no reference picture": a) the previous RAP picture in decoding order is the first coded picture in the bitstream and the current coded picture is a TFD picture; b) the previous RAP picture in decoding order is a BLA picture and the current coded picture is a TFD picture.

It is a requirement of bitstream conformance that the reference picture set is restricted as follows:

[Ed. (GT): Restrictions might be revised to also consider RefPicSetIvCurr.]

- There shall be no picture included in RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetLtCurr for which one or more of the following is true.
 - The picture has nal_unit_type equal to TRAIL_N, TSA_N or STSA_N and TemporalId equal to that of the current picture.
 - The picture has TemporalId greater than that of the current picture.
- When the current picture is a TSA picture, there shall be no picture included in the reference picture set with TemporalId equal to the TemporalId of the current picture.
- When the current picture is an STSA picture, there shall be no picture included in RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetLtCurr that has TemporalId equal to that of the current picture.
- When the current picture is a picture that follows, in decoding order, an STSA picture that has TemporalId equal to that of the current picture, there shall be no picture that has TemporalId equal to that of the current picture included in RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetLtCurr that preceded the STSA picture in decoding order.
- When the current picture is a IDR picture, there shall be no picture included in the reference picture set that is not an inter-view reference picture.
- When the current picture is a RAP picture, there shall be no picture included in the reference picture set that precedes, in decoding order, any preceding RAP picture in decoding order (when present).
- When the current picture is a trailing picture, there shall be no picture in the reference picture set that precedes the associated RAP picture in output order or decoding order.
- When the current picture is a TFD picture and the associated RAP picture is not the first coded picture in the bitstream and is not a BLA picture, there shall be no entry in RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetLtCurr that is equal to "no reference picture".
- When the current picture is not a TFD picture, the following three restrictions apply.

- There shall be no picture included in RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetLtCurr that is a TFD picture.
- There shall be no picture included in RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetLtCurr that precedes, in decoding order, the previous RAP picture in decoding order.
- There shall be no entry in RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetLtCurr that is equal to "no reference picture".
- When the `sps_temporal_id_nesting_flag` is equal to 1, the following applies. Let `tIdA` be the value of `TemporalId` of the current picture `picA`. Any picture `picB` with `TemporalId` equal to `tIdB` that is less than or equal to `tIdA` shall not be included in `RefPicSetStCurrBefore`, `RefPicSetStCurrAfter` and `RefPicSetLtCurr` of `picA` when there exists a picture `picC` with `TemporalId` equal to `tIdC` that is less than `tIdB`, which follows the picture `picB` in decoding order and precedes the picture `picA` in decoding order.

NOTE 5 – A picture cannot be included in more than one of the **five-six** reference picture set lists.

F.8.3.3 Decoding process for generating unavailable reference pictures

F.8.3.3.1 General decoding process for generating unavailable reference pictures

The specifications in subclause **8.3.3.1** apply with the following modification.

- All references to the process specified in subclause **8.3.2** are replaced with references to the process specified in subclause F.8.3.2.

F.8.3.3.2 Generation of one unavailable picture

The specifications in subclause **8.3.3.2** apply.

F.8.3.4 Decoding process for reference picture lists construction

F.8.3.4.1 General decoding process for reference picture lists construction

The specifications in subclause **8.3.4.1** apply with the following modification:

- All references to the process specified in subclause **8.3.2** are replaced with references to the process specified in subclause F.8.3.2
- All invocations of the process specified in subclause **8.3.4.2** are replaced with invocations of the process specified in subclause F.8.3.4.2.

F.8.3.4.2 Initialization process for reference picture lists

This process is invoked when decoding a P or B slice header.

The variable `NumRpsCurrTempList0` is set equal to `Max(num_ref_idx_l0_active_minus1 + 1, NumPocTotalCurr)` and the list `RefPicListTemp0` is constructed as follows:

```

rIdx = 0
while( rIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetLtCurr[ i ]
    for( i = 0; i < NumIvCurr && rIdx < NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetIvCurr[ i ]
}

```

The list `RefPicList0` is constructed as follows:

```

for( rIdx = 0; rIdx <= num_ref_idx_l0_active_minus1; rIdx++ )
    RefPicList0[ rIdx ] = ref_pic_list_modification_flag_l0 ? RefPicListTemp0[ list_entry_l0[ rIdx ] ] :
        RefPicListTemp0[ rIdx ]

```

When the slice is a B slice, the variable `NumRpsCurrTempList1` is set equal to `Max(num_ref_idx_l1_active_minus1 + 1, NumPocTotalCurr)` and the list `RefPicListTemp1` is constructed as follows:

```

rIdx = 0
while( rIdx < NumRpsCurrTempList1 ) {
  for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
    RefPicListTemp1[ rIdx ] = RefPicSetStCurrAfter[ i ]
  for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
    RefPicListTemp1[ rIdx ] = RefPicSetStCurrBefore[ i ]
  for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
    RefPicListTemp1[ rIdx ] = RefPicSetLtCurr[ i ]
  for( i = 0; i < NumIvCurr && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
    RefPicListTemp1[ rIdx ] = RefPicSetIvCurr[ i ]
}

```

When the slice is a B slice, the list RefPicList1 is constructed as follows:

```

for( rIdx = 0; rIdx <= num_ref_idx_l1_active_minus1; rIdx++ )
  RefPicList1[ rIdx ] = ref_pic_list_modification_flag_l1 ? RefPicListTemp1[ list_entry_l1[ rIdx ] ] :
    RefPicListTemp1[ rIdx ]

```

[Ed. (KW): Currently the inter-view reference picture is always appended at the end of the list. JCT3V-B0046 proposes to signal the inter-view reference picture position directly into the bitstream. TBD.]

F.8.4 Decoding process for coding units coded in intra prediction mode

The specifications in subclause 8.4 apply.

F.8.5 Decoding process for coding units coded in inter prediction mode

The specifications in subclause 8.5 apply.

F.8.6 Scaling, transformation and array construction process prior to deblocking filter process

The specifications in subclause 8.6 apply.

F.8.7 In-loop filter process

The specifications in subclause 8.7 apply.

F.9 Parsing process

The specifications in clause 9 apply.

F.10 Specification of bitstream subsets

The specifications in clause 10 apply.

F.11 Profiles and levels

The specifications in Annex A apply, with the following modification:

[Ed. (GT:) To be done.]

F.12 Byte stream format

The specifications in Annex B apply

F.13 MV-HEVC hypothetical reference decoder

F.13.1 General

The specification in subclause C.1 applies by considering that each access unit contains all the view components to be decoded, when the CPB is to be operated in the access unit level. [Ed. (CY): the HEVC base specification already considers CPB that involves coded pictures of multiple layers]

F.13.2 Operation of the decoded picture buffer (DPB)

The decoded picture buffer contains picture storage buffers. Each of the picture storage buffers may contain a decoded view component that is marked as "used for reference" or is held for future output (reordered or delayed pictures). Prior

to initialization, the DPB is empty (the DPB fullness is set to zero). The following steps of the subclauses of this subclause happen in the sequence as listed below.

F.13.2.1 Removal of pictures from the DPB

When decoding a coded video sequence conforming to a profile specified in Annex F using the decoding process specified in Annex F, the process specified in subclause C.2.1 is repeatedly invoked for each view in the ascending order of layer_id, with "picture" being replaced by "view component". During the invocation of the process for a particular view, only view components of the particular view may be removed from the DPB.

F.13.2.2 Picture output

When the coded video sequence containing the picture conforms to a profile specified in Annex F and the decoding process specified in Annex F is used, and picture n is output, the view components of all the target output views as specified in OutputLayerIdList in picture n are output at the same time instance and in ascending order of layer_id. In addition, the process specified in subclause C.2.2 is repeatedly invoked for each view in ascending order of layer_id, with "picture" being replaced by "view component". [Ed. (CY): the target output views are not properly identified.]

F.13.2.3 Current decoded picture marking and storage

When decoding a coded video sequence conforming to a profile specified in Annex F using the decoding process specified in Annex F, the process as specified in subclause C.2.3 is repeatedly invoked for each view in ascending order of layer_id, with "picture" being replaced by "view component",

F.13.3 Bitstream conformance

The specification in subclause C.3 applies..

F.13.4 Decoder conformance

The specification in subclause C.4 general part (the specification text between C.4 and C.4.1) applies. [Ed. (CY): it was suggested that text between C.4 and C.4.1 should be re-structured to be a new C.4.1 general, if that is done, a new subclause is to be created to refer to the new C.4.1].

F.13.4.1 Operation of the output order DPB

Each of the picture storage buffers may contain a decoded view component that is marked as "used for reference" or is held for future output. At HRD initialisation, the DPB fullness, measured in frames, is set to 0. The following steps all happen instantaneously when an access unit is removed from the CPB, and in the order listed.

F.13.4.2 Removal of pictures from the DPB

When decoding a coded video sequence conforming to a profile specified in Annex F using the decoding process specified in Annex F, the process as specified in subclause C.4.2 is repeatedly invoked for each view in ascending order of layer_id, with "picture" being replaced by "view component". During the invocation of the subclause C.4.2 for a particular view, only view components of the particular view may be removed from the DPB.

F.13.4.2.1 "Bumping" process

[Ed. (MH): This subclause seems unnecessary as F.13.4.2.1 is never referred to by any other subclause.] The specification in subclause C.4.2.1 applies.

F.13.4.3 Picture decoding, marking and storage

When decoding a coded video sequence conforming to a profile specified in Annex F using the decoding process specified in Annex F, the decoding process as specified in subclause C.4.3 is repeatedly invoked for each view in ascending order of layer_id, with "picture" being replaced by "view component". [Ed. (CY): implicit marking of inter-view reference pictures is currently not supported.]

F.14 MV-HEVC SEI messages

The specifications in Annex D together with the extensions and modifications specified in this subclause apply.

F.14.1 SEI message syntax

F.14.1.1 View dependency change SEI message syntax

	Descriptor
view_dependency_change(payloadSize) {	
active_vps_id	u(4)
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_disable_flag[i][j]	u(1)
}	

F.14.2 SEI message semantics

F.14.2.1 View dependency change SEI message semantics

This SEI message indicates that the view dependency information changes starting with the current access unit containing the SEI message and is always interpreted with respect to the active video parameter set. When present, the view dependency change SEI message applies to the target access unit set that consists of the current access unit and all the subsequent access units, in decoding order, until the next view dependency change SEI message or the end of the coded video sequence, whichever is earlier in decoding order.

NOTE 1 – The dependent views for any view are always a subset of those indicated by the active video parameter set.

NOTE 2 – View dependency change SEI messages do not have a cumulative effect.

Some of the views indicated by the following syntax elements may not be present in the target access unit set.

active_vps_id identifies an active video parameter set that contains the inter-view dependency relationship information. The value of **active_vps_id** shall be equal to the value of **video_parameter_set_id** in the sequence parameter set referenced by a view component of the coded picture of the access unit containing the view dependency change SEI message. The value of **active_vps_id** shall be in the range of 0 to 15, inclusive.

ref_layer_disable_flag[i][j] equal to 1 indicates that the *j*-th entry of the inter-view reference picture of the view component with **layer_id** equal to *i* will not be present in any of the reference picture lists after reference picture list modification. **ref_layer_disable_flag[i][j]** equal to 0 indicates that the *j*-th entry of the inter-view reference picture of the view component with **layer_id** equal to *i* may be present in at least one of the reference picture lists after reference picture list modification. [Ed. (MH): Consider a constraint or a note that **ref_layer_disable_flag[i][j]** shall be equal to 1 if the **ref_layer_disable_flag[i][j]** in the previous view dependency change SEI message for the same coded video sequence was equal to 1. This constraint would disallow re-enabling a view dependency which had already been disabled in the previous view dependency change SEI message for the same coded video sequence and hence allow safe discarding of those view components that are not intended for output and are no longer needed for inter-view prediction according to the previous view dependency change SEI message.]

F.15 Video usability information

The specifications in Annex E apply.