
**Information technology — Coding of
audio-visual objects —**

**Part 10:
Advanced Video Coding**

**AMENDMENT 2: MVC extensions for
inclusion of depth maps**

Technologies de l'information — Codage des objets audiovisuels —

Partie 10: Codage visuel avancé

*AMENDEMENT 2: Extensions du codage vidéo multivues pour
l'inclusion de cartes de profondeur*



Reference number
ISO/IEC 14496-10:2012/Amd.2:2013(E)

AVA Documents
Document Sharing Hub
© ISO 2013



COPYRIGHT PROTECTED DOCUMENT

© ISO 2013

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
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

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

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 2 to ISO/IEC 14496-10:2012 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects —

Part 10: Advanced Video Coding

AMENDMENT 2: MVC extensions for inclusion of depth maps

In 0.6, add the following paragraph after the paragraph that starts with "Multiview video coding":

An extension of multiview video coding that additionally supports the inclusion of depth maps is specified in Annex I, allowing the construction of bitstreams that represent multiple views with corresponding depth views. In a similar manner as with the multiview video coding specified in Annex H, bitstreams encoded as specified in Annex I may also contain sub-bitstreams that conform to this Specification.

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

Annex I specifies MVC extensions for inclusion of depth maps, referred to as multiview video coding with depth (MVCD). The reader is referred to Annex I for the entire decoding process for MVCD, which is specified there with references being made to clauses 2-9 and Annexes A-E and Annex H. Subclause I.10 specifies one profile for MVCD (Multiview and Depth).

In Clause 2, add the following additional normative reference:

- ISO 12232:2006, *Photography – Digital still cameras – Determination of exposure index, ISO speed ratings, standard output sensitivity, and recommended exposure index.*

In Clause 4, add the following additional abbreviation:

MVCD Multiview Video Coding with Depth

In 7.3.1, replace the syntax table with:

nal_unit(NumBytesInNALunit) {	C	Descriptor
forbidden_zero_bit	All	f(1)
nal_ref_idc	All	u(2)
nal_unit_type	All	u(5)
NumBytesInRBSP = 0		
nalUnitHeaderBytes = 1		
if(nal_unit_type == 14 nal_unit_type == 20 nal_unit_type == 21) {		
svc_extension_flag	All	u(1)
if(svc_extension_flag)		
nal_unit_header_svc_extension() /* specified in Annex G */	All	
else		
nal_unit_header_mvc_extension() /* specified in Annex H */	All	
nalUnitHeaderBytes += 3		
}		
for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {		
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {		
rbsp_byte [NumBytesInRBSP++]	All	b(8)
rbsp_byte [NumBytesInRBSP++]	All	b(8)
i += 2		
emulation_prevention_three_byte /* equal to 0x03 */	All	f(8)
} else		
rbsp_byte [NumBytesInRBSP++]	All	b(8)
}		
}		

In 7.3.2.1.1, replace the syntax table with:

seq_parameter_set_data() {	C	Descriptor
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
constraint_set4_flag	0	u(1)
constraint_set5_flag	0	u(1)
reserved_zero_2bits /* equal to 0 */	0	u(2)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
if(profile_idc == 100 profile_idc == 110 profile_idc == 122 profile_idc == 244 profile_idc == 44 profile_idc == 83 profile_idc == 86 profile_idc == 118 profile_idc == 128 profile_idc == 138) {		
chroma_format_idc	0	ue(v)
if(chroma_format_idc == 3)		
separate_colour_plane_flag	0	u(1)
bit_depth_luma_minus8	0	ue(v)
bit_depth_chroma_minus8	0	ue(v)
qpprime_y_zero_transform_bypass_flag	0	u(1)
seq_scaling_matrix_present_flag	0	u(1)
if(seq_scaling_matrix_present_flag)		
for(i = 0; i < ((chroma_format_idc != 3) ? 8 : 12); i++) {		
seq_scaling_list_present_flag[i]	0	u(1)
if(seq_scaling_list_present_flag[i])		
if(i < 6)		
scaling_list(ScalingList4x4[i], 16, UseDefaultScalingMatrix4x4Flag[i])	0	
else		
scaling_list(ScalingList8x8[i - 6], 64, UseDefaultScalingMatrix8x8Flag[i - 6])	0	
}		
}		
log2_max_frame_num_minus4	0	ue(v)
pic_order_cnt_type	0	ue(v)
if(pic_order_cnt_type == 0)		
log2_max_pic_order_cnt_lsb_minus4	0	ue(v)
else if(pic_order_cnt_type == 1) {		
delta_pic_order_always_zero_flag	0	u(1)
offset_for_non_ref_pic	0	se(v)
offset_for_top_to_bottom_field	0	se(v)
num_ref_frames_in_pic_order_cnt_cycle	0	ue(v)
for(i = 0; i < num_ref_frames_in_pic_order_cnt_cycle; i++)		
offset_for_ref_frame[i]	0	se(v)
}		
max_num_ref_frames	0	ue(v)
gaps_in_frame_num_value_allowed_flag	0	u(1)

pic_width_in_mbs_minus1	0	ue(v)
pic_height_in_map_units_minus1	0	ue(v)
frame_mbs_only_flag	0	u(1)
if(!frame_mbs_only_flag)		
mb_adaptive_frame_field_flag	0	u(1)
direct_8x8_inference_flag	0	u(1)
frame_cropping_flag	0	u(1)
if(frame_cropping_flag) {		
frame_crop_left_offset	0	ue(v)
frame_crop_right_offset	0	ue(v)
frame_crop_top_offset	0	ue(v)
frame_crop_bottom_offset	0	ue(v)
}		
vui_parameters_present_flag	0	u(1)
if(vui_parameters_present_flag)		
vui_parameters()	0	
}		

In 7.3.2.1.3, replace the syntax table with:

subset_seq_parameter_set_rbsp() {	C	Descriptor
seq_parameter_set_data()	0	
if(profile_idc == 83 profile_idc == 86) {		
seq_parameter_set_svc_extension() /* specified in Annex G */	0	
svc_vui_parameters_present_flag	0	u(1)
if(svc_vui_parameters_present_flag == 1)		
svc_vui_parameters_extension() /* specified in Annex G */	0	
} else if(profile_idc == 118 profile_idc == 128) {		
bit_equal_to_one /* equal to 1 */	0	f(1)
seq_parameter_set_mvc_extension() /* specified in Annex H */	0	
mvc_vui_parameters_present_flag	0	u(1)
if(mvc_vui_parameters_present_flag == 1)		
mvc_vui_parameters_extension() /* specified in Annex H */	0	
} else if(profile_idc == 138) {		
bit_equal_to_one /* equal to 1 */	0	f(1)
seq_parameter_set_mvcd_extension() /* specified in Annex I */	0	
}		
additional_extension2_flag	0	u(1)
if(additional_extension2_flag == 1)		
while(more_rbsp_data())		
additional_extension2_data_flag	0	u(1)
rbsp_trailing_bits()	0	
}		

In 7.3.3, replace the syntax table with:

slice_header() {	C	Descriptor
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
if(separate_colour_plane_flag == 1)		
colour_plane_id	2	u(2)
frame_num	2	u(v)
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		
if(IdrPicFlag)		
idr_pic_id	2	ue(v)
if(pic_order_cnt_type == 0) {		
pic_order_cnt_lsb	2	u(v)
if(bottom_field_pic_order_in_frame_present_flag && !field_pic_flag)		
delta_pic_order_cnt_bottom	2	se(v)
}		
if(pic_order_cnt_type == 1 && !delta_pic_order_always_zero_flag) {		
delta_pic_order_cnt[0]	2	se(v)
if(bottom_field_pic_order_in_frame_present_flag && !field_pic_flag)		
delta_pic_order_cnt[1]	2	se(v)
}		
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	2	ue(v)
if(slice_type == B)		
direct_spatial_mv_pred_flag	2	u(1)
if(slice_type == P slice_type == SP slice_type == B) {		
num_ref_idx_active_override_flag	2	u(1)
if(num_ref_idx_active_override_flag) {		
num_ref_idx_l0_active_minus1	2	ue(v)
if(slice_type == B)		
num_ref_idx_l1_active_minus1	2	ue(v)
}		
}		
if(nal_unit_type == 20 nal_unit_type == 21)		
ref_pic_list_mvc_modification() /* specified in Annex H */	2	
else		
ref_pic_list_modification()	2	
if((weighted_pred_flag && (slice_type == P slice_type == SP)) (weighted_bipred_idc == 1 && slice_type == B))		
pred_weight_table()	2	
if(nal_ref_idc != 0)		
dec_ref_pic_marking()	2	
if(entropy_coding_mode_flag && slice_type != I && slice_type != SI)		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)

if(slice_type == SP slice_type == SI) {		
if(slice_type == SP)		
sp_for_switch_flag	2	u(1)
slice_qs_delta	2	se(v)
}		
if(deblocking_filter_control_present_flag) {		
disable_deblocking_filter_idc	2	ue(v)
if(disable_deblocking_filter_idc != 1) {		
slice_alpha_c0_offset_div2	2	se(v)
slice_beta_offset_div2	2	se(v)
}		
}		
if(num_slice_groups_minus1 > 0 && slice_group_map_type >= 3 && slice_group_map_type <= 5)		
slice_group_change_cycle	2	u(v)
}		

Replace Table 7-1 with:

nal_unit_type	Content of NAL unit and RBSP syntax structure	C	Annex A NAL unit type class	Annex G and Annex H NAL unit type class	Annex I NAL unit type class
0	Unspecified		non-VCL	non-VCL	non-VCL
1	Coded slice of a non-IDR picture slice_layer_without_partitioning_rbsp()	2, 3, 4	VCL	VCL	VCL
2	Coded slice data partition A slice_data_partition_a_layer_rbsp()	2	VCL	not applicable	not applicable
3	Coded slice data partition B slice_data_partition_b_layer_rbsp()	3	VCL	not applicable	not applicable
4	Coded slice data partition C slice_data_partition_c_layer_rbsp()	4	VCL	not applicable	not applicable
5	Coded slice of an IDR picture slice_layer_without_partitioning_rbsp()	2, 3	VCL	VCL	VCL
6	Supplemental enhancement information (SEI) sei_rbsp()	5	non-VCL	non-VCL	non-VCL
7	Sequence parameter set seq_parameter_set_rbsp()	0	non-VCL	non-VCL	non-VCL
8	Picture parameter set pic_parameter_set_rbsp()	1	non-VCL	non-VCL	non-VCL
9	Access unit delimiter access_unit_delimiter_rbsp()	6	non-VCL	non-VCL	non-VCL
10	End of sequence end_of_seq_rbsp()	7	non-VCL	non-VCL	non-VCL
11	End of stream end_of_stream_rbsp()	8	non-VCL	non-VCL	non-VCL
12	Filler data filler_data_rbsp()	9	non-VCL	non-VCL	non-VCL
13	Sequence parameter set extension seq_parameter_set_extension_rbsp()	10	non-VCL	non-VCL	non-VCL
14	Prefix NAL unit prefix_nal_unit_rbsp()	2	non-VCL	suffix dependent	suffix dependent
15	Subset sequence parameter set subset_seq_parameter_set_rbsp()	0	non-VCL	non-VCL	non-VCL
16..18	Reserved		non-VCL	non-VCL	non-VCL
19	Coded slice of an auxiliary coded picture without partitioning slice_layer_without_partitioning_rbsp()	2, 3, 4	non-VCL	non-VCL	non-VCL
20	Coded slice extension slice_layer_extension_rbsp()	2, 3, 4	non-VCL	VCL	VCL
21	Coded slice extension for depth view components /*specified in Annex I */ slice_layer_extension_rbsp() /*	2, 3, 4	non-VCL	non-VCL	VCL

	specified in Annex I */				
22..23	Reserved		non-VCL	non-VCL	VCL
24..31	Unspecified		non-VCL	non-VCL	non-VCL

In 7.4.1, make the following changes:

Replace the following:

svc_extension_flag indicates whether a `nal_unit_header_svc_extension()` or `nal_unit_header_mvc_extension()` will follow next in the syntax structure.

with:

svc_extension_flag indicates whether a `nal_unit_header_svc_extension()` or `nal_unit_header_mvc_extension()` will follow next in the syntax structure. When `nal_unit_type` is equal to 21, `svc_extension_flag` shall be equal to 0 and the semantics of `svc_extension_flag` equal to 1 are reserved for future specification by ITU-T | ISO/IEC.

Add the following paragraph after the semantics of `svc_extension_flag` just before the semantics of `rbps_byte[i]`.

The value of `svc_extension_flag` shall be equal to 0 for coded video sequences conforming to one or more profiles specified in Annex I. Decoders conforming to one or more profiles specified in Annex I shall ignore (remove from the bitstream and discard) NAL units for which `nal_unit_type` is equal to 14, 20, or 21 and `svc_extension_flag` is equal to 1.

In 7.4.2.1.1, replace the following:

chroma_format_idc specifies the chroma sampling relative to the luma sampling as specified in clause 6.2. The value of `chroma_format_idc` shall be in the range of 0 to 3, inclusive. When `chroma_format_idc` is not present, it shall be inferred to be equal to 1 (4:2:0 chroma format).

with

chroma_format_idc specifies the chroma sampling relative to the luma sampling as specified in clause 6.2. The value of `chroma_format_idc` shall be in the range of 0 to 3, inclusive. When `chroma_format_idc` is not present and `profile_idc` is not equal to 138, `chroma_format_idc` shall be inferred to be equal to 1 (4:2:0 chroma format). When `chroma_format_idc` is not present and `profile_idc` is equal to 138, `chroma_format_idc` shall be inferred to be equal to 0 (4:0:0 chroma format), otherwise, it shall be inferred to be equal to 1 (4:2:0 chroma format).

In 7.4.2.1.3, replace the following:

additional_extension2_data_flag may have any value. It shall not affect the conformance to profiles specified in Annex A, G, or H.

with

additional_extension2_data_flag may have any value. It shall not affect the conformance to profiles specified in Annex A, G, H, or I.

Replace Annex C with:

Annex C Hypothetical reference decoder

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

This annex specifies the hypothetical reference decoder (HRD) and its use to check bitstream and decoder conformance.

Two types of bitstreams are subject to HRD conformance checking for this Recommendation | International Standard. The first such type of bitstream, called Type I bitstream, is a NAL unit stream containing only the VCL NAL units and filler data NAL units for all access units in the bitstream. The second type of bitstream, called a Type II bitstream, contains, in addition to the VCL NAL units and filler data NAL units for all access units in the bitstream, at least one of the following:

- additional non-VCL NAL units other than filler data NAL units,
- all `leading_zero_8bits`, `zero_byte`, `start_code_prefix_one_3bytes`, and `trailing_zero_8bits` syntax elements that form a byte stream from the NAL unit stream (as specified in Annex B).

Figure C-1 shows the types of bitstream conformance points checked by the HRD.

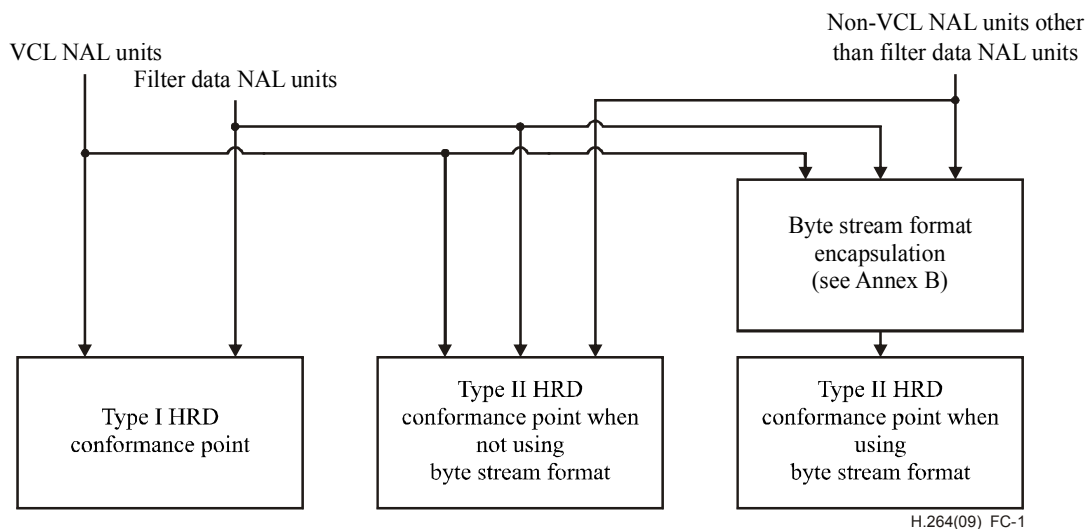


Figure C-1 – Structure of byte streams and NAL unit streams for HRD conformance checks

The syntax elements of non-VCL NAL units (or their default values for some of the syntax elements), required for the HRD, are specified in the semantics subclauses of clause 7, Annexes D and E, and subclauses G.7, G.13, G.14, H.7, H.13, H.14, I.7, I.13, and I.14.

Two types of HRD parameter sets (NAL HRD parameters and VCL HRD parameters) are used. The HRD parameter sets are signalled as follows:

- When the coded video sequence conforms to one or more of the profiles specified in Annex A and the decoding process specified in clauses 2-9 is applied, the HRD parameter sets are signalled through video usability information as specified in subclauses E.1 and E.2, which is part of the sequence parameter set syntax structure.
- When the coded video sequence conforms to one or more of the profiles specified in Annex G and the decoding process specified in Annex G is applied, the HRD parameter sets are signalled through the SVC video usability information extension as specified in subclauses G.14.1 and G.14.2, which is part of the subset sequence parameter set syntax structure.

NOTE 1 – For coded video sequences that conform to both, one or more of the profiles specified in Annex A and one or more of the profiles specified in Annex G, the signalling of the applicable HRD parameter sets is depending on whether the decoding process specified in clauses 2-9 or the decoding process specified in Annex G is applied.

- When the coded video sequence conforms to one or more of the profiles specified in Annex H and the decoding process specified in Annex H is applied, the HRD parameter sets are signalled through the MVC video usability information extension as specified in subclauses H.14.1 and H.14.2, which is part of the subset sequence parameter set syntax structure.

NOTE 2 – For coded video sequences that conform to both, one or more of the profiles specified in Annex A and one or more of the profiles specified in Annex H, the signalling of the applicable HRD parameter sets is depending on whether the decoding process specified in clauses 2-9 or the decoding process specified in Annex H is applied.

- When the coded video sequence conforms to one or more of the profiles specified in Annex I and the decoding process specified in Annex I is applied, the HRD parameter sets are signalled through the MVC video usability information extension as specified in subclause I.14, which is part of the subset sequence parameter set syntax structure.

NOTE 3 – For coded video sequences that conform to one or more of the profiles specified in Annex A, one or more of the profiles specified in Annex H and one or more of the profiles specified in Annex I, the signalling of the applicable HRD parameter sets is depending on whether the decoding process specified in clauses 2-9, the decoding process specified in Annex H or the decoding process specified in Annex I is applied.

All sequence parameter sets and picture parameter sets referred to in the VCL NAL units, and corresponding buffering period and picture timing SEI messages shall be conveyed to the HRD, in a timely manner, either in the bitstream (by non-VCL NAL units), or by other means not specified in this Recommendation | International Standard.

In Annexes C, D, and E and subclauses G.12, G.13, G.14, H.12, H.13, H.14, I.12, I.13, and I.14, the specification for "presence" of non-VCL NAL units is also satisfied when those NAL units (or just some of them) are conveyed to decoders (or to the HRD) by other means not specified by this Recommendation | International Standard. For the purpose of counting bits, only the appropriate bits that are actually present in the bitstream are counted.

NOTE 4 – As an example, synchronization of a non-VCL NAL unit, conveyed by means other than presence in the bitstream, with the NAL units that are present in the bitstream, can be achieved by indicating two points in the bitstream, between which the non-VCL NAL unit would have been present in the bitstream, had the encoder decided to convey it in the bitstream.

When the content of a non-VCL NAL unit is conveyed for the application by some means other than presence within the bitstream, the representation of the content of the non-VCL NAL unit is not required to use the same syntax specified in this annex.

NOTE 5 – When HRD information is contained within the bitstream, it is possible to verify the conformance of a bitstream to the requirements of this subclause based solely on information contained in the bitstream. When the HRD information is not present in the bitstream, as is the case for all "stand-alone" Type I bitstreams, conformance can only be verified when the HRD data is supplied by some other means not specified in this Recommendation | International Standard.

The HRD contains a coded picture buffer (CPB), an instantaneous decoding process, a decoded picture buffer (DPB), and output cropping as shown in Figure C-2.

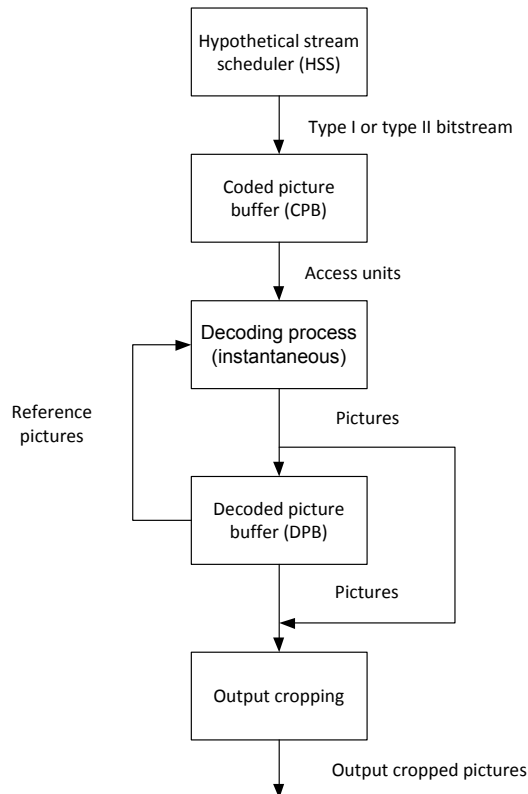


Figure C-2 – HRD buffer model

The CPB size (number of bits) is $CpbSize[SchedSelIdx]$. The DPB size (number of frame buffers) is $Max(1, max_dec_frame_buffering)$. When the coded video sequence conforms to one or more of the profiles specified in Annex H and the decoding process specified in Annex H is applied, the DPB size is specified in units of view components. When the coded video sequence conforms to one or more of the profiles specified in Annex I and the decoding process specified in Annex I is applied, the DPB is operated separately for texture view components and depth view components and the terms texture DPB and depth DPB are used, respectively. The texture DPB size is specified in units of texture view components and the depth DPB size is specified in units of depth view components.

The HRD operates as follows. Data associated with access units that flow into the CPB according to a specified arrival schedule are delivered by the HSS. The data associated with each access unit are removed and decoded instantaneously by the instantaneous decoding process at CPB removal times. Each decoded picture is placed in the DPB at its CPB removal time unless it is output at its CPB removal time and is a non-reference picture. When a picture is placed in the DPB it is removed from the DPB at the later of the DPB output time or the time that it is marked as "unused for reference".

For each picture in the bitstream, the variable `OutputFlag` for the decoded picture and, when applicable, the reference base picture is set as follows:

- If the coded video sequence containing the picture conforms to one or more of the profiles specified in Annex A and the decoding process specified in clauses 2-9 is applied, `OutputFlag` is set equal to 1.
- Otherwise, if the coded video sequence containing the picture conforms to one or more of the profiles specified in Annex G and the decoding process specified in Annex G is applied, the following applies:
 - For a reference base picture, `OutputFlag` is set equal to 0.

- For a decoded picture, OutputFlag is set equal to the value of the output_flag syntax element of the target layer representation.
- Otherwise, if the coded video sequence containing the picture conforms to one or more of the profiles specified in Annex H and the decoding process specified in Annex H is applied, the following applies:
- For the decoded view components of the target output views, OutputFlag is set equal to 1.
- For the decoded view components of other views, OutputFlag is set equal to 0.
- Otherwise (the coded video sequence containing the picture conforms to one or more of the profiles specified in Annex I and the decoding process specified in Annex I is applied), the following applies:
- For the decoded texture view components and corresponding depth view components with same VOIdx of the target output views, OutputFlag is set equal to 1.
- For the decoded texture view components and corresponding depth view components with same VOIdx of other views, OutputFlag is set equal to 0.

The operation of the CPB is specified in subclause C.1. The instantaneous decoder operation is specified in clauses 2-9 (for coded video sequences conforming to one or more of the profiles specified in Annex A) and in Annex G (for coded video sequences conforming to one or more of the profiles specified in Annex G) and in Annex H (for coded video sequences conforming to one or more of the profiles specified in Annex H) and in Annex I (for coded video sequences conforming to one or more of the profiles specified in Annex I). The operation of the DPB is specified in subclause C.2. The output cropping is specified in subclause C.2.2.

NOTE 6 – Coded video sequences that conform to both, one or more of the profiles specified in Annex A and one or more of the profiles specified in Annex G, can be decoded either by the decoding process specified in clauses 2-9 or by the decoding process specified in Annex G. The decoding result and the HRD operation may be dependent on which of the decoding processes is applied.

NOTE 7 – Coded video sequences that conform both to one or more of the profiles specified in Annex A and one or more of the profiles specified in Annex H can be decoded either by the decoding process specified in clauses 2-9 or by the decoding process specified in Annex H. The decoding result and the HRD operation may be dependent on which of the decoding processes is applied.

NOTE 8 – Coded video sequences that conform to one or more of the profiles specified in Annex A, one or more of the profiles specified in Annex H and one or more of the profiles specified in Annex I, can be decoded either by the decoding process specified in clauses 2-9, by the decoding process specified in Annex H or by the decoding process specified in Annex I. The decoding result and the HRD operation may be dependent on which of the decoding processes is applied.

HSS and HRD information concerning the number of enumerated delivery schedules and their associated bit rates and buffer sizes is specified in subclauses E.1.1, E.1.2, E.2.1, E.2.2, G.14.1, G.14.2, H.14.1, H.14.2 and I.14. The HRD is initialised as specified by the buffering period SEI message as specified in subclauses D.1.1 and D.2.1. The removal timing of access units from the CPB and output timing from the DPB are specified in the picture timing SEI message as specified in subclauses D.1.2 and D.2.2. All timing information relating to a specific access unit shall arrive prior to the CPB removal time of the access unit.

When the coded video sequence conforms to one or more of the profiles specified in Annex G and the decoding process specified in Annex G is applied, the following is specified:

- (a) When an access unit contains one or more buffering period SEI messages that are included in scalable nesting SEI messages and are associated with values of DQId in the range of $((DQIdMax \gg 4) \ll 4)$ to $((DQIdMax \gg 4) \ll 4) + 15$, inclusive, the last of these buffering period SEI messages in decoding order is the buffering period SEI message that initialises the HRD. Let hrdDQId be the largest value of $16 * sei_dependency_id[i] + sei_quality_id[i]$ that is associated with the scalable nesting SEI message

containing the buffering period SEI message that initialises the HRD, let hrdDId and hrdQId be equal to $\text{hrdDQId} \gg 4$ and $\text{hrdDQId} \& 15$, respectively, and let hrdTId be the value of sei_temporal_id that is associated with the scalable nesting SEI message containing the buffering period SEI message that initialises the HRD.

- (b) The picture timing SEI messages that specify the removal timing of access units from the CPB and output timing from the DPB are the picture timing SEI messages that are included in scalable nesting SEI messages associated with values of $\text{sei_dependency_id}[i]$, $\text{sei_quality_id}[i]$, and sei_temporal_id equal to hrdDId , hrdQId , and hrdTId , respectively.
- (c) The HRD parameters that are used for conformance checking are the HRD parameters included in the SVC video usability information extension of the active SVC sequence parameter set that are associated with values of $\text{vui_ext_dependency_id}[i]$, $\text{vui_ext_quality_id}[i]$, and $\text{vui_ext_temporal_id}[i]$ equal to hrdDId , hrdQId , and hrdTId , respectively. For the specification in this annex, num_units_in_tick , time_scale , $\text{fixed_frame_rate_flag}$, $\text{nal_hrd_parameters_present_flag}$, $\text{vcl_hrd_parameters_present_flag}$, $\text{low_delay_hrd_flag}$, and $\text{pic_struct_present_flag}$ are substituted with the values of $\text{vui_ext_num_units_in_tick}[i]$, $\text{vui_ext_time_scale}[i]$, $\text{vui_ext_fixed_frame_rate_flag}[i]$, $\text{vui_ext_nal_hrd_parameters_present_flag}[i]$, $\text{vui_ext_vcl_hrd_parameters_present_flag}[i]$, $\text{vui_ext_low_delay_hrd_flag}[i]$, and $\text{vui_ext_pic_struct_present_flag}[i]$, respectively, with i being the value for which $\text{vui_ext_dependency_id}[i]$, $\text{vui_ext_quality_id}[i]$, and $\text{vui_ext_temporal_id}[i]$ are equal to hrdDId , hrdQId , and hrdTId , respectively.

When the coded video sequence conforms to one or more of the profiles specified in Annex H and the decoding process specified in Annex H is applied, the following is specified:

- (a) When an access unit contains one or more buffering period SEI messages that are included in MVC scalable nesting SEI messages, the buffering period SEI message that is associated with the operation point being decoded is the buffering period SEI message that initialises the HRD. Let $\text{hrdVId}[i]$ be equal to $\text{sei_op_view_id}[i]$ for all i in the range of 0 to $\text{num_view_components_op_minus1}$, inclusive, and let hrdTId be the value of $\text{sei_op_temporal_id}$, that are associated with the MVC scalable nesting SEI message containing the buffering period SEI message that initialises the HRD.
- (b) The picture timing SEI messages that specify the removal timing of access units from the CPB and output timing from the DPB are the picture timing SEI messages that are included in MVC scalable nesting SEI messages associated with values of $\text{sei_op_view_id}[i]$ equal to $\text{hrdVId}[i]$ for all i in the range of 0 to $\text{num_view_components_op_minus1}$, inclusive, and sei_temporal_id equal to hrdTId .
- (c) The HRD parameters that are used for conformance checking are the HRD parameters included in the MVC video usability information extension of the active MVC sequence parameter set that are associated with values of $\text{vui_mvc_view_id}[i][j]$ for all j in the range of 0 to $\text{vui_mvc_num_target_output_views_minus1}[i]$, inclusive, equal to $\text{hrdVId}[j]$, and the value of $\text{vui_mvc_temporal_id}[i]$ equal to hrdTId . For the specification in this annex, num_units_in_tick , time_scale , $\text{fixed_frame_rate_flag}$, $\text{nal_hrd_parameters_present_flag}$, $\text{vcl_hrd_parameters_present_flag}$, $\text{low_delay_hrd_flag}$, and $\text{pic_struct_present_flag}$ are substituted with the values of $\text{vui_mvc_num_units_in_tick}[i]$, $\text{vui_mvc_time_scale}[i]$, $\text{vui_mvc_fixed_frame_rate_flag}[i]$, $\text{vui_mvc_nal_hrd_parameters_present_flag}[i]$, $\text{vui_mvc_vcl_hrd_parameters_present_flag}[i]$, $\text{vui_mvc_low_delay_hrd_flag}[i]$, and $\text{vui_mvc_pic_struct_present_flag}[i]$, respectively, with i being the value for which $\text{vui_mvc_view_id}[i]$ is equal to $\text{hrdVId}[j]$ for all j in the range of 0 to $\text{vui_mvc_num_target_output_views_minus1}[i]$, inclusive, and $\text{vui_mvc_temporal_id}[i]$ equal to hrdTId .

When the coded video sequence conforms to one or more of the profiles specified in Annex I and the decoding process specified in Annex I is applied, the following is specified:

- (a) When an access unit contains one or more buffering period SEI messages that are included in MVCD scalable nesting SEI messages, the buffering period SEI message that is associated with the operation point being decoded is the buffering period SEI message that initialises the HRD. Let $\text{hrdVId}[i]$ be equal to $\text{sei_op_view_id}[i]$ for all i in the range of 0 to $\text{num_view_components_op_minus1}$, inclusive, and let hrdTId be the value of $\text{sei_op_temporal_id}$, that are associated with the MVCD scalable nesting SEI message containing the buffering period SEI message that initialises the HRD.

- (b) The picture timing SEI messages that specify the removal timing of access units from the CPB and output timing from the DPB are the picture timing SEI messages that are included in MVCD scalable nesting SEI messages associated with values of `sei_op_view_id[i]` equal to `hrdVid[i]` for all `i` in the range of 0 to `num_view_components_op_minus1`, inclusive, and `sei_temporal_id` equal to `hrdTId`.
- (c) The HRD parameter sets that are used for conformance checking are the HRD parameter sets, included in the MVC video usability information extension of the active MVCD sequence parameter set, that are associated with values of `vui_mvc_view_id[i][j]` for all `j` in the range of 0 to `vui_mvc_num_target_output_views_minus1[i]`, inclusive, equal to `hrdVid[j]`, and the value of `vui_mvc_temporal_id[i]` equal to `hrdTId`. For the specification in this annex, `num_units_in_tick`, `time_scale`, `fixed_frame_rate_flag`, `nal_hrd_parameters_present_flag`, `vcl_hrd_parameters_present_flag`, `low_delay_hrd_flag`, and `pic_struct_present_flag` are substituted with the values of `vui_mvc_num_units_in_tick[i]`, `vui_mvc_time_scale[i]`, `vui_mvc_fixed_frame_rate_flag[i]`, `vui_mvc_nal_hrd_parameters_present_flag[i]`, `vui_mvc_vcl_hrd_parameters_present_flag[i]`, `vui_mvc_low_delay_hrd_flag[i]`, and `vui_mvc_pic_struct_present_flag[i]`, respectively, with `i` being the value for which `vui_mvc_view_id[i]` is equal to `hrdVid[j]` for all `j` in the range of 0 to `vui_mvc_num_target_output_views_minus1[i]`, inclusive, and `vui_mvc_temporal_id[i]` equal to `hrdTId`.

The HRD is used to check conformance of bitstreams and decoders as specified in subclauses C.3 and C.4, respectively.

NOTE 9 – While conformance is guaranteed under the assumption that all frame-rates and clocks used to generate the bitstream match exactly the values signalled in the bitstream, in a real system each of these may vary from the signalled or specified value.

All the arithmetic in this annex is done with real values, so that no rounding errors can propagate. For example, the number of bits in a CPB just prior to or after removal of an access unit is not necessarily an integer.

The variable t_c is derived as follows and is called a clock tick:

$$t_c = \text{num_units_in_tick} \div \text{time_scale} \quad (\text{C-1})$$

The following is specified for expressing the constraints in this annex:

- Let access unit `n` be the `n`-th access unit in decoding order with the first access unit being access unit 0.
- Let picture `n` be the primary coded picture or the decoded primary picture of access unit `n`.

C.1 Operation of coded picture buffer (CPB)

The specifications in this subclause apply independently to each set of CPB parameters that is present and to both the Type I and Type II conformance points shown in Figure C-1.

C.1.1 Timing of bitstream arrival

The HRD may be initialised at any one of the buffering period SEI messages. Prior to initialisation, the CPB is empty.

NOTE – After initialisation, the HRD is not initialised again by subsequent buffering period SEI messages.

Each access unit is referred to as access unit `n`, where the number `n` identifies the particular access unit. The access unit that is associated with the buffering period SEI message that initialises the CPB is referred to as access unit 0. The value of `n` is incremented by 1 for each subsequent access unit in decoding order.

The time at which the first bit of access unit `n` begins to enter the CPB is referred to as the initial arrival time $t_{ai}(n)$.

The initial arrival time of access units is derived as follows:

- If the access unit is access unit 0, $t_{ai}(0) = 0$,

- Otherwise (the access unit is access unit n with $n > 0$), the following applies:
 - If $\text{cbr_flag}[\text{SchedSelIdx}]$ is equal to 1, the initial arrival time for access unit n , is equal to the final arrival time (which is derived below) of access unit $n - 1$, i.e.,

$$t_{ai}(n) = t_{af}(n - 1) \quad (\text{C-2})$$

- Otherwise ($\text{cbr_flag}[\text{SchedSelIdx}]$ is equal to 0), the initial arrival time for access unit n is derived by

$$t_{ai}(n) = \text{Max}(t_{af}(n - 1), t_{ai,earliest}(n)) \quad (\text{C-3})$$

where $t_{ai,earliest}(n)$ is derived as follows:

- If access unit n is not the first access unit of a subsequent buffering period, $t_{ai,earliest}(n)$ is derived as

$$t_{ai,earliest}(n) = t_{r,n}(n) - (\text{initial_cpb_removal_delay}[\text{SchedSelIdx}] + \text{initial_cpb_removal_delay_offset}[\text{SchedSelIdx}]) \div 90000 \quad (\text{C-4})$$

with $t_{r,n}(n)$ being the nominal removal time of access unit n from the CPB as specified in subclause C.1.2 and $\text{initial_cpb_removal_delay}[\text{SchedSelIdx}]$ and $\text{initial_cpb_removal_delay_offset}[\text{SchedSelIdx}]$ being specified in the previous buffering period SEI message.

- Otherwise (access unit n is the first access unit of a subsequent buffering period), $t_{ai,earliest}(n)$ is derived as

$$t_{ai,earliest}(n) = t_{r,n}(n) - (\text{initial_cpb_removal_delay}[\text{SchedSelIdx}] \div 90000) \quad (\text{C-5})$$

with $\text{initial_cpb_removal_delay}[\text{SchedSelIdx}]$ being specified in the buffering period SEI message associated with access unit n .

The final arrival time for access unit n is derived by

$$t_{af}(n) = t_{ai}(n) + b(n) \div \text{BitRate}[\text{SchedSelIdx}] \quad (\text{C-6})$$

where $b(n)$ is the size in bits of access unit n , counting the bits of the VCL NAL units and the filler data NAL units for the Type I conformance point or all bits of the Type II bitstream for the Type II conformance point, where the Type I and Type II conformance points are as shown in Figure C-1.

The values of SchedSelIdx , $\text{BitRate}[\text{SchedSelIdx}]$, and $\text{CpbSize}[\text{SchedSelIdx}]$ are constrained as follows:

- If the content of the active sequence parameter sets for access unit n and access unit $n - 1$ differ, the HSS selects a value SchedSelIdx1 of SchedSelIdx from among the values of SchedSelIdx provided in the active sequence parameter set for access unit n that results in a $\text{BitRate}[\text{SchedSelIdx1}]$ or $\text{CpbSize}[\text{SchedSelIdx1}]$ for access unit n . The value of $\text{BitRate}[\text{SchedSelIdx1}]$ or $\text{CpbSize}[\text{SchedSelIdx1}]$ may differ from the value of $\text{BitRate}[\text{SchedSelIdx0}]$ or $\text{CpbSize}[\text{SchedSelIdx0}]$ for the value SchedSelIdx0 of SchedSelIdx that was in use for access unit $n - 1$.
- Otherwise, the HSS continues to operate with the previous values of SchedSelIdx , $\text{BitRate}[\text{SchedSelIdx}]$ and $\text{CpbSize}[\text{SchedSelIdx}]$.

When the HSS selects values of $\text{BitRate}[\text{SchedSelIdx}]$ or $\text{CpbSize}[\text{SchedSelIdx}]$ that differ from those of the previous access unit, the following applies:

- the variable $\text{BitRate}[\text{SchedSelIdx}]$ comes into effect at time $t_{ai}(n)$

- the variable $CpbSize[SchedSelIdx]$ comes into effect as follows:
 - If the new value of $CpbSize[SchedSelIdx]$ exceeds the old CPB size, it comes into effect at time $t_{ai}(n)$,
 - Otherwise, the new value of $CpbSize[SchedSelIdx]$ comes into effect at the time $t_r(n)$.

C.1.2 Timing of coded picture removal

When an access unit n is the access unit with n equal to 0 (the access unit that initialises the HRD), the nominal removal time of the access unit from the CPB is specified by

$$t_{r,n}(0) = \text{initial_cpb_removal_delay}[SchedSelIdx] \div 90000 \quad (C-7)$$

When an access unit n is the first access unit of a buffering period that does not initialise the HRD, the nominal removal time of the access unit from the CPB is specified by

$$t_{r,n}(n) = t_{r,n}(n_b) + t_c * \text{cpb_removal_delay}(n) \quad (C-8)$$

where $t_{r,n}(n_b)$ is the nominal removal time of the first access unit of the previous buffering period and $\text{cpb_removal_delay}(n)$ is the value of cpb_removal_delay specified in the picture timing SEI message associated with access unit n .

The nominal removal time $t_{r,n}(n)$ of an access unit n that is not the first access unit of a buffering period is given by

$$t_{r,n}(n) = t_{r,n}(n_b) + t_c * \text{cpb_removal_delay}(n) \quad (C-9)$$

where $t_{r,n}(n_b)$ is the nominal removal time of the first access unit of the current buffering period and $\text{cpb_removal_delay}(n)$ is the value of cpb_removal_delay specified in the picture timing SEI message associated with access unit n .

The removal time of access unit n is specified as follows:

- If $\text{low_delay_hrd_flag}$ is equal to 0 or $t_{r,n}(n) \geq t_{af}(n)$, the removal time of access unit n is specified by

$$t_r(n) = t_{r,n}(n) \quad (C-10)$$

- Otherwise ($\text{low_delay_hrd_flag}$ is equal to 1 and $t_{r,n}(n) < t_{af}(n)$), the removal time of access unit n is specified by

$$t_r(n) = t_{r,n}(n) + t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) \div t_c) \quad (C-11)$$

NOTE – The latter case indicates that the size of access unit n , $b(n)$, is so large that it prevents removal at the nominal removal time.

When an access unit n is the first access unit of a buffering period, n_b is set equal to n at the removal time $t_r(n)$ of the access unit n .

C.2 Operation of the decoded picture buffer (DPB)

The decoded picture buffer contains frame buffers. When a coded video sequence conforming to one or more of the profiles specified in Annex A is decoded by applying the decoding process specified in clauses 2-9, each of the frame buffers may contain a decoded frame, a decoded complementary field pair or a single (non-paired) decoded field that is marked as "used for reference" (reference pictures) or is held for future output (reordered or delayed pictures). When a coded video sequence conforming to one or more of the profiles specified in Annex G is decoded by applying the decoding process specified in Annex G, each frame buffer may contain a decoded frame, a decoded complementary field

pair, a single (non-paired) decoded field, a decoded reference base frame, a decoded reference base complementary field pair or a single (non-paired) decoded reference base field that is marked as "used for reference" (reference pictures) or is held for future output (reordered or delayed pictures). When a coded video sequence conforming to one or more of the profiles specified in Annex H is decoded by applying the decoding process specified in Annex H, each of the frame buffers may contain a decoded frame view component, a decoded complementary field view component pair, or a single (non-paired) decoded field view component that is marked as "used for reference" (reference pictures) or is held for future output (reordered or delayed pictures) or is held as reference for inter-view prediction (inter-view only reference components). When a coded video sequence conforming to one or more of the profiles specified in Annex I is decoded by applying the decoding process specified in Annex I, each of the frame buffers of the texture DPB may contain a, a decoded depth frame view component, a decoded complementary texture field view component pair, or a single (non-paired) decoded texture field view component that is marked as "used for reference" (reference pictures) or is held for future output (reordered or delayed pictures) or is held as reference for inter-view prediction (inter-view only reference components). When a coded video sequence conforming to one or more of the profiles specified in Annex I is decoded by applying the decoding process specified in Annex I, each of the frame buffers of the depth DPB may contain a decoded depth frame view component, a decoded complementary depth field view component pair, or a single (non-paired) decoded depth field view component that is marked as "used for reference" (reference pictures) or is held for future output (reordered or delayed pictures) or is held as reference for inter-view prediction (inter-view only reference components).

Prior to initialisation, the DPB is empty (the DPB fullness is set to zero). The following steps specified in this subclause all happen instantaneously at $t_r(n)$ and in the order listed. When the decoding process specified in Annex H or Annex I is applied, the view components of the current primary coded picture are processed by applying the ordered steps to each view component in increasing order of the associated view order index VOIdx. During the invocation of the process for a particular texture view, only the texture view components of the particular view are considered and during the invocation of the process for a particular depth view, only the depth view components of the particular view are considered. For each view component of the current primary coded picture, the corresponding depth view component with the same view order index VOIdx, if present, is processed after the texture view component.

1. The process of decoding gaps in frame_num and storing "non-existing" frames as specified in subclause C.2.1 is invoked.
2. The picture decoding and output process as specified in subclause C.2.2 is invoked.
3. The process of removing pictures from the DPB before possible insertion of the current picture as specified in subclause C.2.3 is invoked.
4. The process of marking and storing the current decoded picture as specified in subclause C.2.4 is invoked.

NOTE – When the decoding process specified in Annex G is applied, the DPB is only operated for decoded pictures and reference base pictures associated with decoded pictures. The DPB is not operated for layer pictures with dependency_id less than DependencyIdMax (and associated reference base pictures). All decoded pictures and associated reference base pictures are decoded pictures and associated reference base pictures for dependency_id equal to DependencyIdMax, which represent the results of the decoding process specified in subclause G.8.

C.2.1 Decoding of gaps in frame_num and storage of "non-existing" frames

When the decoding process specified in Annex H is applied, the process specified in this subclause is invoked for a particular view with view order index VOIdx, with "picture" being replaced by "view component", "frame" being replaced by "frame view component", and "field" being replaced by "field view component". During the invocation of the process for a particular view, only view components of the particular view are considered and view components of other views are not marked as "unused for reference" or removed from the DPB. When the decoding process specified in Annex I is applied, the process specified in this subclause for Annex H is invoked for particular texture view or depth view with view order index VOIdx, with each "view component" being replaced by "texture view component" or "depth view component", "frame view component" being replaced by "texture frame view component" or "depth frame view component", and "field view component" being replaced by "texture field view component". During the invocation of the process for a particular texture view, only the texture view components of the particular view are considered and during the invocation of the process for a particular depth view, only the depth view components of the particular view are considered and view components of other views are not marked as "unused for reference" or removed from the DPB.

The DPB fullness represents the total number of non-empty frame buffers. When the decoding process specified in Annex H is applied, this includes frame buffers that contain view components of other views. When the decoding process specified in Annex I is applied, this includes frame buffers that contain texture or depth view components of other views.

When applicable, gaps in frame_num are detected by the decoding process and the generated frames are marked and inserted into the DPB as specified below.

Gaps in frame_num are detected by the decoding process and the generated frames are marked as specified in subclauses 8.2.5.2 and G.8.2.5.

After the marking of each generated frame, each picture m marked by the "sliding window" process as "unused for reference" is removed from the DPB when it is also marked as "non-existing" or its DPB output time is less than or equal to the CPB removal time of the current picture n ; i.e., $t_{o,dpb}(m) \leq t_r(n)$, or it has OutputFlag equal to 0. When a frame or the last field in a frame buffer is removed from the DPB, the DPB fullness is decremented by one. The "non-existing" generated frame is inserted into the DPB and the DPB fullness is incremented by one.

C.2.2 Picture decoding and output

When the decoding process specified in Annex H is applied, the process specified in this subclause is invoked for a particular view with view order index VOIdx.

When the decoding process specified in Annex I is applied, the process specified in this subclause is invoked for a particular texture view or depth view with view order index VOIdx.

The decoding of the current picture or view component (when applying the decoding process specified in Annex H or Annex I) and the derivation of the DPB output time (if applicable) is specified as follows:

- If the decoding process specified in clause 8 or Annex G is applied, the following applies:
 - The current primary coded picture n is decoded.
 - When picture n has OutputFlag equal to 1, its DPB output time $t_{o,dpb}(n)$ is derived by

$$t_{o,dpb}(n) = t_r(n) + t_c * dpb_output_delay(n) \quad (C-12)$$

where $dpb_output_delay(n)$ is the value of dpb_output_delay specified in the picture timing SEI message associated with access unit n .

- Otherwise (the decoding process specified in Annex H or Annex I is applied), the following applies:
 - The view component with view order index VOIdx of the current primary coded picture n is decoded.
 - When VOIdx is equal to VOIdxMin and any of the view components of picture n has OutputFlag equal to 1, the DPB output time $t_{o,dpb}(n)$ for picture n is derived by Equation C-12, where $dpb_output_delay(n)$ is the value of dpb_output_delay specified in the picture timing SEI message associated with access unit n .

The output of the current picture or view component (when applying the decoding process specified in Annex H or Annex I) is specified as follows:

- If OutputFlag is equal to 1 and $t_{o,dpb}(n) = t_r(n)$, the current picture or view component is output.

NOTE 1 – When the current picture or view component has nal_ref_idc greater than 0 (when using the decoding process specified in Annex G, nal_ref_idc is the syntax element of the target layer representation), it will be stored in the DPB.

- Otherwise, if OutputFlag is equal to 0, the current picture or view component is not output, but it may be stored in the DPB as specified in subclause C.2.4.

- Otherwise (OutputFlag is equal to 1 and $t_{o,dpb}(n) > t_r(n)$), the current picture or view component is output later and will be stored in the DPB (as specified in subclause C.2.4) and is output at time $t_{o,dpb}(n)$ unless indicated not to be output by the decoding or inference of no_output_of_prior_pics_flag equal to 1 at a time that precedes $t_{o,dpb}(n)$.

NOTE 2 – When the coded video sequence conforms to a profile specified in Annex H and the decoding process specified in Annex H is used, the view components of all the target output views of a picture are output at the same time instant and in increasing order of the view order index VOIdx.

NOTE 3 – When the coded video sequence conforms to a profile specified in Annex I and the decoding process specified in Annex I is used, the view components of all the target output views of a picture are output at the same time instant and in increasing order of the view order index VOIdx. A depth view component, if present, follows the texture view component within the same view component.

When output, the picture or view component shall be cropped, using the cropping rectangle specified in the active sequence parameter set for the picture or view component.

When the decoding process specified in clause 8 or Annex G is applied, the current picture n is a picture that is output and is not the last picture of the bitstream that is output, the value of $\Delta t_{o,dpb}(n)$ is derived by

$$\Delta t_{o,dpb}(n) = t_{o,dpb}(n_n) - t_{o,dpb}(n) \quad (C-13)$$

where n_n indicates the picture that follows after picture n in output order and has OutputFlag equal to 1.

When the decoding process specified in Annex H or Annex I is applied, the current picture n is a picture that contains at least one view component that is output and the current picture is not the last picture of the bitstream that contains at least one view component that is output and VOIdx is equal to VOIdxMin, the value of $\Delta t_{o,dpb}(n)$ is derived by Equation C-13, where n_n indicates the picture that follows after picture n in output order and contains at least one any view component with OutputFlag equal to 1.

The decoded picture or view component is temporarily stored (not in the DPB).

C.2.3 Removal of pictures from the DPB before possible insertion of the current picture

When the decoding process specified in Annex H is applied, the process specified in this subclause is invoked for a particular view with view order index VOIdx, with "picture" being replaced by "view component", "frame" being replaced by "frame view component", and "field" being replaced by "field view component".

When the decoding process specified in Annex I is applied, the process specified in this subclause for Annex I is invoked for particular texture view and depth view with view order index VOIdx, with each "view component" being replaced by "texture view component" or "depth view component", "frame view component" being replaced by "texture frame view component" or "depth frame view component", and "field view component" being replaced by "texture field view component". During the invocation of the process for a particular texture view, only the texture view components of the particular view are considered and during the invocation of the process for a particular depth view, only the depth view components of the particular view are considered.

When the decoding process specified in Annex H or Annex I is applied, the following process is specified for removing inter-view only reference components of the current access unit from the DPB. By this process, view components of the current view with view order index VOIdx are not removed from the DPB, but inter-view only reference components of other views may be removed. The removal of inter-view only reference components is specified as follows:

- If the view order index VOIdx of the current view is equal to VOIdxMax, all inter-view only reference components m for which any of the following conditions are true are removed from the DPB:
 - OutputFlag is equal to 0,
 - The DPB output time $t_{o,dpb}(m)$ of the picture containing the view component m is less than or equal to the CPB removal time $t_r(n)$ of the current picture.

- Otherwise (the view order index VOIdx of the current view is less than VOIdxMax), all inter-view only reference components m for which both of the following conditions are true are removed from the DPB:
 - OutputFlag is equal to 0 or the DPB output time $t_{o,dpb}(m)$ of the picture containing the view component m is less than or equal to the CPB removal time $t_r(n)$ of the current picture,
 - One of the following conditions is true:
 - The current view component is a view component of an anchor picture and the view_id of the inter-view only reference component m is not equal to any value of anchor_ref_IX[k][j], with X being equal to 0 or 1, k being any integer value greater than the view order index VOIdx of the current view, and j being any integer value in the range of 0 to Max(0, num_anchor_refs_IX[k] – 1), inclusive,
 - The current view component is not a view component of an anchor picture and the view_id of the inter-view only reference component m is not equal to any value of non_anchor_ref_IX[k][j], with X being equal to 0 or 1, k being any integer value greater than the view order index VOIdx of the current view, and j being any integer value in the range of 0 to Max(0, num_non_anchor_refs_IX[k] – 1), inclusive.

When the decoding process specified in Annex H is applied, for the following processes specified in this subclause, only view components of the particular view for which this subclause is invoked are considered, and view components of other views are not marked as "unused for reference" or removed from the DPB. When the decoding process specified in Annex I is applied, for the following processes specified for Annex I in this subclause, during the invocation of the process for a particular texture view, only texture view components of the particular texture view are considered and during the invocation of the process for a particular depth view, only depth view components of the particular depth view are considered, and view components of other views are not marked as "unused for reference" or removed from the DPB. The DPB fullness represents the total number of non-empty frame buffers. When the decoding process specified in Annex H is applied, this includes frame buffers that contain texture view components of other views. When the decoding process specified in Annex I is applied, this includes frame buffers that contain texture or depth view components of other views.

The removal of pictures from the DPB before possible insertion of the current picture proceeds as follows:

- If the decoded picture is an IDR picture the following applies:
 1. All reference pictures in the DPB are marked as "unused for reference" as specified in subclause 8.2.5.1 when a coded video sequence conforming to one or more of the profiles specified in Annex A is decoded by applying the decoding process specified in clauses 2-9, or as specified in subclause G.8.2.4 when a coded video sequence conforming to one or more of the profiles specified in Annex G is decoded by applying the decoding process specified in Annex G, or as specified in subclause H.8.3 when a coded video sequence conforming to one or more of the profiles specified in Annex H is decoded by applying the decoding process specified in Annex H, or as specified in subclause I.8.3 when a coded video sequence conforming to one or more of the profiles specified in Annex I is decoded by applying the decoding process specified in Annex I.
 2. When the IDR picture is not the first IDR picture decoded and the value of PicWidthInMbs or FrameHeightInMbs or max_dec_frame_buffering derived from the active sequence parameter set is different from the value of PicWidthInMbs or FrameHeightInMbs or max_dec_frame_buffering derived from the sequence parameter set that was active for the preceding picture, respectively, no_output_of_prior_pics_flag is inferred to be equal to 1 by the HRD, regardless of the actual value of no_output_of_prior_pics_flag.

NOTE – Decoder implementations should try to handle frame or DPB size changes more gracefully than the HRD in regard to changes in PicWidthInMbs or FrameHeightInMbs.
 3. When no_output_of_prior_pics_flag is equal to 1 or is inferred to be equal to 1, all frame buffers in the DPB are emptied without output of the pictures they contain, and DPB fullness is set to 0.
- Otherwise (the decoded picture is not an IDR picture), the following applies:

- If the slice header of the current picture includes `memory_management_control_operation` equal to 5, all reference pictures in the DPB are marked as "unused for reference".
- Otherwise (the slice header of the current picture does not include `memory_management_control_operation` equal to 5), the decoded reference picture marking process specified in subclause 8.2.5 is invoked when a coded video sequence conforming to one or more of the profiles specified in Annex A is decoded by applying the decoding process specified in clauses 2-9, or the decoded reference picture marking process specified in subclause G.8.2.4 is invoked when a coded video sequence conforming to one or more of the profiles specified in Annex G is decoded by applying the decoding process specified in Annex G, or the decoded reference picture marking process specified in subclause H.8.3 is invoked when a coded video sequence conforming to one or more of the profiles specified in Annex H is decoded by applying the decoding process specified in Annex H, or the decoded reference picture marking process specified in subclause I.8.3 is invoked when a coded video sequence conforming to one or more of the profiles specified in Annex I is decoded by applying the decoding process specified in Annex I.

All pictures m in the DPB, for which all of the following conditions are true, are removed from the DPB:

- picture m is marked as "unused for reference" or picture m is a non-reference picture. When a picture is a reference frame, it is considered to be marked as "unused for reference" only when both of its fields have been marked as "unused for reference",
- picture m is marked as "non-existing" or it has `OutputFlag` equal to 0 or its DPB output time $t_{o,dpb}(m)$ is less than or equal to the CPB removal time $t_r(n)$ of the current picture n .

When a frame or the last field in a frame buffer is removed from the DPB, the DPB fullness is decremented by one.

C.2.4 Current decoded picture marking and storage

When the decoding process specified in Annex H is applied, the process specified in this subclause is invoked for a particular view with view order index $VOIdx$, with "picture" being replaced by "view component", "frame" being replaced by "frame view component", and "field" being replaced by "field view component". When the decoding process specified in Annex I is applied, the process specified in this subclause for Annex I is invoked for particular texture view and depth view with view order index $VOIdx$, with each "view component" being replaced by "texture view component" and "depth view component", "frame view component" being replaced by "texture frame view component" and "depth frame view component", and "field view component" being replaced by "texture field view component". In subclause C.2.4.2, the DPB output time $t_{o,dpb}(n)$ and the CPB removal time $t_r(n)$ of a view component are the DPB output time and the CPB removal time of the picture n containing the view component, depth view component or texture view component.

The marking and storage of the current decoded picture is specified as follows:

- If the current picture is a reference picture, the marking and storage process for reference pictures as specified in subclause C.2.4.1 is invoked.
- Otherwise (the current picture is a non-reference picture), the storage process for non-reference pictures as specified in subclause C.2.4.2 is invoked.

C.2.4.1 Marking and storage of a reference picture into the DPB

The current picture is stored in the DPB as follows:

- If the current decoded picture is a second field (in decoding order) of a complementary reference field pair, and the first field of the pair is still in the DPB, the current decoded picture is stored in the same frame buffer as the first field of the pair.
- Otherwise, the current decoded picture is stored in an empty frame buffer, and the DPB fullness is incremented by one.

When the coded video sequence conforms to one or more of the profiles specified in Annex G and the decoding process specified in Annex G is applied and the current picture has `store_ref_base_pic_flag` equal to 1 (i.e., the current picture is associated with a reference base picture), the associated reference base picture is stored in the DPB as follows:

- If the reference base picture is a second field (in decoding order) of a complementary reference base field pair, and the first field of the pair is still in the DPB, the reference base picture is stored in the same frame buffer as the first field of the pair.
- Otherwise, the reference base picture is stored in an empty frame buffer, and the DPB fullness is incremented by one.

C.2.4.2 Storage of a non-reference picture into the DPB

The variable `storePicFlag` is derived as follows:

- If any of the following conditions are true, `storePicFlag` is set equal to 1:
 - the current picture n has `OutputFlag` equal to 1 and $t_{o,dpb}(n) > t_r(n)$,
 - the decoding process specified in Annex H or Annex I is used and the current view component has a view order index `VOIDx` less than `VOIDxMax` and `inter_view_flag` equal to 1.
- Otherwise, `storePicFlag` is set equal to 0.

When `storePicFlag` is equal to 1, the current picture is stored in the DPB as follows:

- If the current decoded picture is a second field (in decoding order) of a complementary non-reference field pair, and the first field of the pair is still in the DPB, the current decoded picture is stored in the same frame buffer as the first field of the pair.
- Otherwise, the current decoded picture is stored in an empty frame buffer, and the DPB fullness is incremented by one.

C.3 Bitstream conformance

A bitstream of coded data conforming to this Recommendation | International Standard fulfils the following requirements.

The bitstream is constructed according to the syntax, semantics, and constraints specified in this Recommendation | International Standard outside of this annex.

The bitstream is tested by the HRD as specified below:

For Type I bitstreams, the number of tests carried out is equal to `cpb_cnt_minus1 + 1` where `cpb_cnt_minus1` is either the syntax element of `hrd_parameters()` following the `vcl_hrd_parameters_present_flag` or is determined by the application by other means not specified in this Recommendation | International Standard. One test is carried out for each bit rate and CPB size combination specified by `hrd_parameters()` following the `vcl_hrd_parameters_present_flag`. Each of these tests is conducted at the Type I conformance point shown in Figure C-1.

For Type II bitstreams there are two sets of tests. The number of tests of the first set is equal to `cpb_cnt_minus1 + 1` where `cpb_cnt_minus1` is either the syntax element of `hrd_parameters()` following the `vcl_hrd_parameters_present_flag` or is determined by the application by other means not specified in this Recommendation | International Standard. One test is carried out for each bit rate and CPB size combination. Each of these tests is conducted at the Type I conformance point shown in Figure C-1. For these tests, only VCL and filler data NAL units are counted for the input bit rate and CPB storage.

The number of tests of the second set, for Type II bitstreams, is equal to `cpb_cnt_minus1 + 1` where `cpb_cnt_minus1` is either the syntax element of `hrd_parameters()` following the `nal_hrd_parameters_present_flag` or is determined by the application by other means not specified in this Recommendation | International Standard. One test is carried out for each

bit rate and CPB size combination specified by `hrd_parameters()` following the `nal_hrd_parameters_present_flag`. Each of these tests is conducted at the Type II conformance point shown in Figure C-1. For these tests, all NAL units (of a Type II NAL unit stream) or all bytes (of a byte stream) are counted for the input bit rate and CPB storage.

NOTE 1 – NAL HRD parameters established by a value of `SchedSelIdx` for the Type II conformance point shown in Figure C-1 are sufficient to also establish VCL HRD conformance for the Type I conformance point shown in Figure C-1 for the same values of `initial_cpb_removal_delay[SchedSelIdx]`, `BitRate[SchedSelIdx]`, and `CpbSize[SchedSelIdx]` for the VBR case (`cbr_flag[SchedSelIdx]` equal to 0). This is because the data flow into the Type I conformance point is a subset of the data flow into the Type II conformance point and because, for the VBR case, the CPB is allowed to become empty and stay empty until the time a next picture is scheduled to begin to arrive. For example, when a coded video sequence conforming to one or more of the profiles specified in Annex A is decoded by applying the decoding process specified in clauses 2-9, when NAL HRD parameters are provided for the Type II conformance point that not only fall within the bounds set for NAL HRD parameters for profile conformance in item j) of subclause A.3.1 or item h) of subclause A.3.3 (depending on the profile in use) but also fall within the bounds set for VCL HRD parameters for profile conformance in item i) of subclause A.3.1 or item g) of subclause A.3.3 (depending on the profile in use), conformance of the VCL HRD for the Type I conformance point is also assured to fall within the bounds of item i) of subclause A.3.1.

For conforming bitstreams, all of the following conditions shall be fulfilled for each of the tests:

1. For each access unit n , with $n > 0$, associated with a buffering period SEI message, with $\Delta t_{g,90}(n)$ specified by

$$\Delta t_{g,90}(n) = 90000 * (t_{r,n}(n) - t_{af}(n-1)) \quad (C-14)$$

the value of `initial_cpb_removal_delay[SchedSelIdx]` shall be constrained as follows:

- If `cbr_flag[SchedSelIdx]` is equal to 0,

$$\text{initial_cpb_removal_delay}[SchedSelIdx] \leq \text{Ceil}(\Delta t_{g,90}(n)) \quad (C-15)$$

- Otherwise (`cbr_flag[SchedSelIdx]` is equal to 1),

$$\text{Floor}(\Delta t_{g,90}(n)) \leq \text{initial_cpb_removal_delay}[SchedSelIdx] \leq \text{Ceil}(\Delta t_{g,90}(n)) \quad (C-16)$$

NOTE 2 – The exact number of bits in the CPB at the removal time of each picture may depend on which buffering period SEI message is selected to initialise the HRD. Encoders must take this into account to ensure that all specified constraints must be obeyed regardless of which buffering period SEI message is selected to initialise the HRD, as the HRD may be initialised at any one of the buffering period SEI messages.

2. A CPB overflow is specified as the condition in which the total number of bits in the CPB is larger than the CPB size. The CPB shall never overflow.
3. A CPB underflow is specified as the condition in which $t_{r,n}(n)$ is less than $t_{af}(n)$. When `low_delay_hrd_flag` is equal to 0, the CPB shall never underflow.
4. The nominal removal times of pictures from the CPB (starting from the second picture in decoding order), shall satisfy the constraints on $t_{r,n}(n)$ and $t_r(n)$ expressed in subclauses A.3.1 through A.3.3 for the profile and level specified in the bitstream when a coded video sequence conforming to one or more of the profiles specified in Annex A is decoded by applying the decoding process specified in clauses 2-9, and they shall satisfy the constraints on $t_{r,n}(n)$ and $t_r(n)$ expressed in subclauses G.10.2.1 and G.10.2.2 for profile and level specified in the bitstream when a coded video sequence conforming to one or more of the profiles specified in Annex G is decoded by applying the decoding process specified in Annex G, and they shall satisfy the constraints on $t_{r,n}(n)$ and $t_r(n)$ expressed in subclause H.10.2 for the profile and level specified in the bitstream when a coded video sequence conforming to one or more of the profiles specified in Annex H is decoded by applying the decoding process specified in Annex H, and they shall satisfy the constraints on $t_{r,n}(n)$ and $t_r(n)$ expressed in subclause I.10.2 for the profile and level specified in the bitstream when a coded video sequence conforming to