

Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEGDocument **JVT-D145**

4th Meeting: Klagenfurt, Austria, 22-26 July, 2002

File: JVT-D145.doc

Generated: 2002-07-25

Title: High-Level Syntax, Semantics, and Decoder Process*Status:* Output of a Breakout Group*Contact:* Miska M. Hannuksela*Purpose:* Proposal

This document is the output of the NAL and high-level syntax breakout group. It presents the proposed high-level syntax elements, semantics, and decoder process relative to JVT-D017d2.

7.3 Syntax in tabular form**7.3.1 NAL unit syntax**

<code>nal_unit(NumBytesInEBSP) {</code>	Category	Descriptor
<code>forbidden_bit</code>		u(1)
<code>nal_priority</code>		u(2)
<code>nal_unit_type</code>		u(5)
<code>~~picture_header_flag~~</code>		u(1)
<code>~~non_stored_content_flag~~</code>		u(1)
<code>for(i=0; i<NumBytesInEBSP; i++)</code>		
<code> ebsp[i]</code>		b(8)
<code>}</code>		

Error! No text of specified style in document.

7.3.2 Raw and encapsulated byte sequence payloads

7.3.2.2 Independent GOP parameter set RBSP syntax

[Ed.Note: Should aspect_ratio_info follow video_signal_type ? Should sar_width, sar_height be e(v) ?]

	Category	Descriptor
igop_parameter_set_rbsp() {		
profile	0	u(8)
level	0	u(8)
igop_parameter_set_id	0	ue(v)
log2_max_frame_num_minus4	0	ue(v)
poc_coding_type	0	ue(v)
if (poc_coding_type == 1)		
log2_max_poc_minus4	0	ue(v)
else if (poc_coding_type == 2) {		
expected_delta_for_non_stored_picture		se(v)
num_frames_in_poc_cycle		ue(v)
for (i = 0; i < num_frames_in_poc_cycle; i++)		
expected_delta_poc		se(v)
}		
num_of_reference_pictures	0	ue(v)
required_frame_num_update_behaviour	0	u(1)
frame_width_in_MBs_minus1	0	ue(v)
frame_height_in_MBs_minus1	0	ue(v)
aspect_ratio_info	0	b(8)
if(aspect_ratio_info == "Extended SAR") {		
sar_width	0	u(8)
sar_height	0	u(8)
}		
video_signal_type	0	u(1)
if(video_signal_type) {		
video_format	0	u(3)
video_range	0	u(1)
colour_description	0	u(1)
if(colour_description) {		
colour_primaries	0	b(8)
transfer_characteristics	0	b(8)
matrix_coefficients	0	b(8)
}		
}		
constrained_intra_prediction_flag	0	u(1)
num_units_in_tick	0	u(32)
time_scale	0	u(32)
motion_vectors_over_picture_boundaries	0	u(1)
fixed_frame_rate	0	u(1)
rbsp_trailing_bits()	0	
}		

2 Error! No text of specified style in document.

7.3.2.x Picture parameter set RBSP syntax

picture_parameter_set_rbsp() {	Category	Descriptor
picture_parameter_set_id	0	ue(v)
igop_parameter_set_id	0	ue(v)
frame_cropping_flag	0	u(1)
if (frame_cropping_flag) {		
frame_cropping_rect_left_offset	0	ue(v)
frame_cropping_rect_right_offset	0	ue(v)
frame_cropping_rect_top_offset	0	ue(v)
frame_cropping_rect_bottom_offset	0	ue(v)
}		
entropy_coding_mode	0	ue(v)
motion_resolution	0	ue(v)
num_slice_groups_minus1	0	ue(v)
if(num_slice_groups_minus1 > 0) { /* use of Flexible MB Order */		
mb_allocation_map_type	0	ue(v)
if(mb_allocation_map_type == 0)		
for(i=0; i<=max_slice_group_id; i++)		
run_length	0	ue(v)
else if(mb_allocation_map_type == 2)		
for(i=0; i<num_mbs_in_picture; i++)		
slice_group_id	0	u(v)
}		
num_ref_pic_active_fwd_minus1	0	ue(v)
num_ref_pic_active_bwd_minus1	0	ue(v)
slice_qp_minus26 /* relative to 26 */	0	se(v)
slice_qp_s_minus26 /* relative to 26 */	0	se(v)
rbsp_trailing_bits()	0	
}		

7.3.2.x Picture delimiter RBSP syntax

picture_delimiter_rbsp() {	Category	Descriptor
picture_coding_type		u(3)
reserved		u(4)
non_stored_picture_flag		u(1)
}		

7.3.2.x Filler data RBSP syntax

filler_data_rbsp(NumBytesInRBSP) {	Category	Descriptor
for(i=0; i<NumBytesInRBSP; i++)		
rbsp[i]		b(8)
}		

Error! No text of specified style in document.

7.3.3 Slice header syntax

	Category	Descriptor
<code>slice_header() {</code>		
<code> picture_parameter_set_id</code>	4	ue(v)
<code> frame_num</code>	4	u(v)
<code> picture_structure</code>	4	ue(v)
<code> first_mb_in_slice</code>	4	u(v)
<code> slice_coding_type</code>	4	ue(v)
<code> if (poc_coding_type == 1)</code>		
<code> coded_poc</code>	4	u(v)
<code> else if (poc_coding_type == 2)</code>		
<code> delta_poc</code>	4	se(v)
<code> num_ref_pic_active_override_flag</code>	4	u(1)
<code> if (num_ref_pic_active_override_flag) {</code>		
<code> if (slice_coding_type == Inter slice_coding_type == Bipred) {</code>		
<code> num_ref_pic_active_fwd_minus1</code>	4	ue(v)
<code> if (slice_coding_type == Bipred)</code>		
<code> num_ref_pic_active_bwd_minus1</code>	4	ue(v)
<code> }</code>		
<code> }</code>		
<code> rps_layer()</code>		
<code> slice_qp_delta</code>	4	se(v)
<code> if (slice_coding_type == SP slice_coding_type == SI) {</code>		
<code> if (slice_coding_type == SP)</code>		
<code> sp_for_switch_flag</code>	4	u(1)
<code> slice_qp_s_delta</code>	4	se(v)
<code> }</code>		
<code> if (slice_coding_type == Bipred) {</code>		
<code> /* Adaptive Bi-predictive weighting parameters? */</code>		
<code> }</code>		
<code>}</code>		

7.3.4 Reference picture selection layer syntax

4 Error! No text of specified style in document.

	Category	Descriptor
rps_layer() {		
if(coding_type() != Intra) {		
reference_reordering_indicator_fwd	3 4	u(1)
if(reference_reordering_indicator_fwd) {		
do {		
remapping_of_pic_nums_indicator	3 4	me(v)
if(remapping_of_pic_nums_indicator == 0 remapping_of_pic_nums_indicator == 1)		
abs_diff_pic_numbers	3 4	ue(v)
else if(remapping_of_pic_nums_indicator == 2)		
long_term_pic_index	3 4	ue(v)
} while(remapping_of_pic_nums_indicator != 3)		
}		
}		
if(coding_type() == Bipred) {		
reference_reordering_indicator_bwd	3 4	u(1)
if(reference_reordering_indicator_bwd) {		
do {		
remapping_of_pic_nums_indicator	3 4	me(v)
if(remapping_of_pic_nums_indicator == 0 remapping_of_pic_nums_indicator == 1)		
abs_diff_pic_numbers	3 4	ue(v)
else if(remapping_of_pic_nums_indicator == 2)		
long_term_pic_index	3 4	ue(v)
} while(remapping_of_pic_nums_indicator != 3)		
}		
}		
ref_pic_buffering_mode	3 4	u(1)
if(ref_pic_buffering_mode == 1)		
mmco_layer()		
}		

7.3.4 Memory management control operation layer syntax

Error! No text of specified style in document.

	Category	Descriptor
mmco_layer() {		
do {		
memory_management_control_operation	3 4 7	me(v)
if(memory_management_control_operation == 1 memory_management_control_operation == 3)		
difference_of_pic_nums	3 4 7	e(v)
else if(memory_management_control_operation == 2 memory_management_control_operation == 3)		
long_term_pic_index	3 4 7	ue(v)
else if(memory_management_control_operation == 4)		
max_long_term_pic_index_plus1	3 4 7	ue(v)
} while(memory_management_control_operation != 0 memory_management_control_operation != 5)		
}		
}		

8 Semantics

8.1 NAL unit semantics

The Video Coding Layer (VCL) is defined to efficiently represent the content of the video data, and the Network Abstraction Layer (NAL) is defined to format that data and provide header information in a manner appropriate for conveyance by the transport layers or storage media. The data is organized into NAL units, each of which contains an integer number of bytes. A NAL unit defines a generic format for use in both packet-oriented and bitstream systems. The format of NAL units for both packet-oriented transport and bitstream is identical except for the fact that each NAL unit can be preceded by a start code in a bitstream-oriented transport layer.

NumBytesInEBSP defines the size of the EBSP in bytes. This element of the NAL interface does not necessarily appear as a syntax element, but rather denotes the quantity of EBSP data to be carried by the system for the NAL unit.

forbidden_bit shall always be zero. The forbidden_bit may be used by external specifications to signal potentially corrupt NAL units.

nal_priority equal to 0 signals that the content of the EBSP belongs to a picture that is not stored in the multi-frame buffer. nal_priority shall not be 0 for parameter set information. If nal_priority is 0 for one slice or data partition EBSP of a particular picture, it shall be 0 for all slice and data partition EBSPs of the picture. nal_priority greater than 0 signals that the content of the EBSP belongs to a picture that is stored in the multi-frame buffer.

In addition to signalling non-stored content, nal_priority signals the relative transport priority of the NAL unit. Value 0 signals the lowest transport priority and the priority grows in ascending order of nal_priority values.

nal_unit_type indicates the type of element contained in the NAL unit according to the types specified in Table 8-1.

ebsp[i] an Encapsulated Byte Sequence Payload contains the NAL unit data in the encapsulated byte sequence payload (EBSP) format. Any sequence of bits can be formatted into a sequence of bytes in a manner defined as a Raw Byte Sequence Payload (RBSP), and any RBSP can be encapsulated in a manner that prevents emulation of byte stream start code prefixes in a manner defined as an encapsulated byte sequence payload (EBSP).

Within the ebsp data, an ebsp[i] byte having the value zero (0x00) shall not be immediately followed by a byte ebsp[i+1] having any of the following three values:

- zero (0x00)
- one (0x01)
- two (0x02)

If an ebsp[i] byte having the value zero (0x00) is immediately followed by an ebsp[i+1] byte having the value 0x03, these bytes shall not be immediately followed within the EBSP by a byte ebsp[i+2] having any value other than the following four values:

- zero (0x00)
- one (0x01)

- two (0x02)
- three (0x03)

When an `ebsp[ i ]` byte having the value zero (0x00) is immediately followed within the EBS by the special `ebsp[ i+1 ]` byte value of three (0x03), the `ebsp[ i+1 ]` value 0x03 shall be removed from the bitstream by the decoder and discarded. This special value is used to allow an RBS payload to contain any sequence of byte values without allowing the EBS to contain the three special two-byte patterns described above.

Code	NAL Unit Type (nal_unit_type)
0x0	Reserved for external use
0x1	Slice
0x2	DPA
0x3	DPB
0x4	DPC
0x5	IDR
0x6	Supplemental Enhancement Information (SEI)
0x7	Independent GOP Parameter Set (GPS)
0x8	Picture Parameter Set (PPS)
0x9	Picture Delimiter (PD)
0xA	Filler Data (FD)
0xB – 0x17	Reserved
0x18-0x1F	Reserved for external use

Table 8-1 – NAL Unit Type (NUT) Codes

An instantaneous decoder refresh picture (IDR picture) implies that all pictures in the multi-picture buffer are marked as “unused” except the current picture. Moreover, the maximum long-term index is reset to zero. An IDR picture contains only Intra or SI slices, and IDR slice type shall be used for all slices of an IDR picture. The picture starts an independent group of pictures (independent GOP) that lasts until the next IDR picture.

8.2.3 Independent GOP parameter set RBS semantics

The parameter set mechanism decouples the transmission of infrequently changing information from the transmission of coded macroblock data. An independent GOP parameter set includes the parameters that remain the same within an independent GOP. A picture parameter set includes the parameters that remain the same within a picture. Each slice of a picture refers to one picture parameter set, and a picture parameter set refers to one independent GOP parameter set. The independent GOP parameter set, the picture parameter set, and the slice header contain all the parameter values needed to correctly decode the VCL data.

It is recommended to convey parameter sets out-of-band using a reliable transport mechanism. However, in applications where the bitstream shall be self-contained, in-band parameter set information units may be used. In error-prone transmission environments, in-band parameter set information units should be protected in a way that assures their successful reception. Synchronization between in-band and out-of-band transmission of the parameter set information is outside of the scope of this Recommendation | International Standard.

Every picture parameter set referred from slices within an independent GOP shall refer to the same independent GOP parameter set.

profile and **level** indicate the profile and level that the coded data of the independent GOP conforms to.

igop_parameter_set_id identifies the parameter set to be addressed.

log2_max_frame_num_minus4 specifies the MAX_PN constant used in frame number related arithmetic. $MAX\_PN = 2^{(\log2\_max\_frame\_num\_minus4 + 4)}$.

Error! No text of specified style in document.

poc_coding_type equal to 0 indicates that no picture order count is conveyed in the coded data. poc_coding_type equal to 1 or 2 indicate the method to code picture order count. See subclause QQ.

log2_max_poc_minus4 specifies the max_poc constant used in picture order count related arithmetic. $\text{max\_poc} = 2^{(\text{log2\_max\_tr\_minus4} + 4)}$. The size of the coded_poc parameter in the slice header is $\text{log2\_max\_tr\_minus4} + 4$ bits.

expected_delta_for_non_stored_picture indicates an expected picture order count difference of a non-stored picture compared to a stored picture having the same frame_num as the non-stored picture.

num_frames_in_poc_cycle signals the number of frame numbers in a picture order count cycle. A picture order count cycle is a repetitive pattern of picture order count differences, each of which corresponds to a frame_num increment of one.

expected_delta_poc indicates an expected difference of picture order count corresponding to a frame_num increment of one. Notation $\text{expected\_delta\_poc}_n$ is used to indicate the expected_delta_poc corresponding to index n in the picture order count cycle. $\text{expected\_delta\_poc\_in\_poc\_cycle}$ is a sum of all values of expected_delta_poc calculated according to equation 7-QQ.

$$\begin{aligned} \text{expected\_delta\_poc\_in\_poc\_cycle} &= 0; \\ \text{for } (i = 0; i \leq \text{frame\_num\_on\_poc\_cycle}; i++) \\ \text{expected\_delta\_poc\_in\_poc\_cycle} &+= \text{expected\_delta\_poc}_i \end{aligned} \quad (7\text{-}QQ)$$

num_of_reference_pictures defines the total number of short- and long-term picture buffers in the multi-frame buffer.

If **required_frame_num_update_behaviour** is 1, a specific decoder behaviour in case of missing frame numbers is mandated (see subclause 9.1.1.2).

frame_width_in_MBs_minus1 and **frame_height_in_MBs_minus1** define the size of the luma frame array internal to the decoder in units of macroblocks. The frame width and height in units of macroblocks is computed by adding 1 to the decoded values of each of these parameters.

video_signal_type: A flag which if set to '1' indicates the presence of video_signal_type information. If set to '0', the video signal type is undefined or specified externally.

video_format: This is a three bit integer indicating the representation of the pictures before being coded in accordance with this Recommendation | International Standard. Its meaning is defined in Table 8-2. If the video_signal_type() is not present in the bitstream then the video format may be assumed to be "Unspecified video format".

Table 8-2 – Meaning of video_format

video_format	Meaning
000	Component
001	PAL
010	NTSC
011	SECAM
100	MAC
101	Unspecified video format
110	Reserved
111	Reserved

video_range: This one-bit flag indicates the black level and range of the luminance and chrominance signals.

colour_description: A flag which if set to '1' indicates the presence of colour_primaries, transfer_characteristics and matrix_coefficients in the bitstream.

colour_primaries: This 8-bit integer describes the chromaticity coordinates of the source primaries, and is defined in Table 8-3.

Table 8-3 – Colour Primaries

Value	Primaries
0	Reserved

8 Error! No text of specified style in document.

1	ITU-R Recommendation BT.709 primary x y green 0,300 0,600 blue 0,150 0,060 red 0,640 0,330 white D65 0,3127 0,3290
2	Unspecified video Image characteristics are unknown.
3	Reserved
4	ITU-R Recommendation BT.470-2 System M primary x y green 0,21 0,71 blue 0,14 0,08 red 0,67 0,33 white C 0,310 0,316
5	ITU-R Recommendation BT.470-2 System B, G primary x y green 0,29 0,60 blue 0,15 0,06 red 0,64 0,33 white D65 0,3127 0,3290
6	SMPTE 170M primary x y green 0,310 0,595 blue 0,155 0,070 red 0,630 0,340 white D65 0,3127 0,3290
7	SMPTE 240M (1987) primary x y green 0,310 0,595 blue 0,155 0,070 red 0,630 0,340 white D65 0,3127 0,3290
8	Generic film (colour filters using Illuminant C) primary x y green 0,243 0,692 (Wratten 58) blue 0,145 0,049 (Wratten 47) red 0,681 0,319 (Wratten 25)
9-255	Reserved

In the case that video_signal_type() is not present in the bitstream or colour_description is zero the chromaticity is unspecified or specified externally.

transfer_characteristics: This 8-bit integer describes the opto-electronic transfer characteristic of the source picture, and is defined in Table 8-4.

Table 8-4 – Transfer Characteristics

Value	Transfer Characteristic
0	Reserved

Error! No text of specified style in document.

1	ITU-R Recommendation BT.709 $V = 1,099 L_c^{0,45} - 0,099$ for $1 \geq L_c \geq 0,018$ $V = 4,500 L_c$ for $0,018 > L_c \geq 0$
2	Unspecified video Image characteristics are unknown.
3	Reserved
4	ITU-R Recommendation BT.470-2 System M Assumed display gamma 2,2
5	ITU-R Recommendation BT.470-2 System B, G Assumed display gamma 2,8
6	SMPTE 170M $V = 1,099 L_c^{0,45} - 0,099$ for $1 \geq L_c \geq 0,018$ $V = 4,500 L_c$ for $0,018 > L_c \geq 0$
7	SMPTE 240M (1987) $V = 1,1115 L_c^{0,45} - 0,1115$ for $L_c \geq 0,0228$ $V = 4,0 L_c$ for $0,0228 > L_c$
8	Linear transfer characteristics i.e. $V = L_c$
9	Logarithmic transfer characteristic (100:1 range) $V = 1,0 - \log_{10}(L_c)/2$ for $1 = L_c = 0,01$ $V = 0,0$ for $0,01 > L_c$
10	Logarithmic transfer characteristic (316.22777:1 range) $V = 1,0 - \log_{10}(L_c)/2,5$ for $1 = L_c = 0,0031622777$ $V = 0,0$ for $0,0031622777 > L_c$
11-255	Reserved

In the case that video_signal_type() is not present in the bitstream or colour_description is zero the transfer characteristics are unspecified or are specified externally.

matrix_coefficients: This 8-bit integer describes the matrix coefficients used in deriving luminance and chrominance signals from the green, blue, and red primaries, as specified in Table 8-5.

Using this table:

E^*Y is analogue with values between 0 and 1
 E^*R , E^*G , and E^*B are analogue with values between 0 and 1
 E^*PB and E^*PR are analogue between the values -0,5 and 0,5
White is defined as $E^*R = E^*G = E^*B = 1$
White equivalently given by $E^*Y = 1$, $E^*PB = 0$, $E^*PR = 0$
 $E^*Y = K_R * E^*R + (1 - K_R - K_B) * E^*G + K_B * E^*B$
 $E^*PB = \frac{0,5}{1 - K_R} (E^*R - E^*Y)$

$$E'_{PB} = \frac{0,5}{1 - K_B} (E'B - E'Y)$$

Y, Cb, and Cr are related to E'Y, E'PB, and E'PR by the following formulae:

if video_range=0:

$$\begin{aligned} Y &= \text{round}(219 * E'Y + 16) \\ Cb &= \text{round}(224 * E'PB + 128) \\ Cr &= \text{round}(224 * E'PR + 128) \end{aligned}$$

if video_range=1:

$$\begin{aligned} Y &= \text{round}(255 * E'Y) \\ Cb &= \text{round}(255 * E'PB + 128) \\ Cr &= \text{round}(255 * E'PR + 128) \end{aligned}$$

Table 8-5 – Matrix Coefficients

Value	Matrix
0	Reserved
1	ITU-R Recommendation BT.709 $K_G = 0,7152; K_R = 0,2126$
2	Unspecified video Image characteristics are unknown.
3	Reserved
4	FCC $K_G = 0,59; K_R = 0,30$
5	ITU-R Recommendation BT.470-2 System B, G: $K_G = 0,587; K_R = 0,299$
6	SMPTE 170M $K_G = 0,587; K_R = 0,299$
7	SMPTE 240M (1987) $K_G = 0,701; K_R = 0,212$
8-255	Reserved

In the case that video_signal_type() is not present in the bitstream or colour_description is zero the matrix coefficients are assumed to be undefined or specified externally.

In the case that video_signal_type() is not present in the bitstream, video_range is assumed to have the value 0 (a nominal range of Y from 16 to 235).

aspect_ratio_info is an eight-bit integer which defines the value of sample aspect ratio. Table 8-6 shows the meaning of the code. If aspect_ratio_info indicates Extended SAR, sample_aspect_ratio is represented by sar_width and sar_height. The sar_width and sar_height shall be relatively prime. If aspect_ratio_info is zero or if either sar_width or sar_height are zero, the sample aspect ratio shall be considered unspecified or specified externally.

Table 8-6 – Meaning of sample aspect ratio

aspect_ratio_info	Sample aspect ratio
0000 0000	Undefined or specified externally
0000 0001	1:1 ("Square")
0000 0010	12:11 (625-type for 4:3 picture)
0000 0011	10:11 (525-type for 4:3 picture)
0000 0100	16:11 (625-type stretched for 16:9 picture)
0000 0101	40:33 (525-type stretched for 16:9 picture)
0000 0110	24:11 (Half-wide 4:3 for 625)
0000 0111	20:11 (Half-wide 4:3 for 525)
0000 1000	32:11 (Half-wide 16:9 for 625)

Error! No text of specified style in document.

0000 1001	80:33 (Half-wide 16:9 for 525)
0000 1010	18:11 (2/3-wide 4:3 for 625)
0000 1011	15:11 (2/3-wide 4:3 for 525)
0000 1100	24:11 (2/3-wide 16:9 for 625)
0000 1101	20:11 (2/3-wide 16:9 for 525)
0000 1110	16:11 (3/4-wide 4:3 for 625)
0000 1111	40:33 (3/4-wide 4:3 for 525)
0001 0000	64:33 (3/4-wide 16:9 for 625)
0001 0001	160:99 (3/4-wide 16:9 for 525)
0001 0010 to 1111 1110	Reserved
1111 1111	Extended SAR

sar_width is an 8-bit unsigned integer which indicates the horizontal size of sample aspect ratio. A zero value is forbidden.

sar_height is an 8-bit unsigned integer which indicates the vertical size of sample aspect ratio. A zero value is forbidden.

constrained_intra_prediction_flag equal to zero stands for normal intra prediction, whereas one stands for the constrained intra prediction. In the constrained intra prediction mode, no intra prediction is done from inter blocks.

num_units_in_tick is the number of time units of a clock operating at the frequency **time_scale** Hz that corresponds to one increment of a clock tick counter defined externally or in Annex C. A clock tick is the minimum interval of time that can be represented in the coded data. For example, if the clock frequency of a video signal is $(30\ 000) \div 1001$ Hz, **time_scale** may be 30 000 and **num_units_in_tick** may be 1001.

time_scale is the number of time units which pass in one second. For example, a time coordinate system that measures time using a 27 MHz clock has a **time_scale** of 27 000 000.

motion_vectors_over_picture_boundaries equal to 0 indicates that no motion vector points outside the picture boundaries.

8.2.x Picture parameter set RBSP semantics

picture_parameter_set_id identifies the picture parameter set to be addressed.

igop_parameter_set_id refers to the independent GOP parameter set that is associated with this picture parameter set.

frame_cropping_flag equal to zero signals that defines that the entire reconstructed picture area shall be the output of the decoding process.

frame_cropping_rect_left, **frame_cropping_rect_right**, **frame_cropping_rect_top**, **frame_cropping_rect_bottom** define the area of the luma picture internal array which shall be the output of the decoding process. The decoded values of these offsets consist of non-negative integer values, and the output of the decoding process is defined as the area within the rectangle containing luma samples with horizontal coordinates from **cropping_rect_left** to $16 \cdot (\text{picture\_width\_in\_MBs\_minus1} + 1) - (\text{cropping\_rect\_right} + 1)$ and with vertical coordinates from **cropping_rect_top** to $16 \cdot (\text{picture\_height\_in\_MBs\_minus1} + 1) - (\text{cropping\_rect\_bottom} + 1)$, inclusive.

entropy_coding_mode equal to zero stands for the non-arithmetic variable-length coding, whereas value one stands for the arithmetic variable-length coding (see clause 10).

motion_resolution equal to zero stands for 1/4-sample motion resolution, and equal to one stands for 1/8-sample motion resolution.

num_slice_groups_minus1: The number of slice groups is equal to **num_slice_groups_minus1** + 1. If **num_slice_groups_minus1** is zero, all slices of the picture belong to the same slice group, and no flexible macroblock ordering is applied. If **num_slice_groups_minus1** is greater than zero, flexible macroblock ordering is in use. Values greater than 7 are Reserved for future use.

mb_allocation_map_type: The macroblock allocation map type is present only if **num_slice_groups_minus1** is greater than 0. This parameter indicates how the macroblock allocation map is coded. Values ranging from 0 to 2 are valid.

run_length codeword follows for each slice group. It signals the number of consecutive macroblocks that are assigned to the slice group in raster scan order

12 Error! No text of specified style in document.

mb_allocation_map_type 0 is used to interleave slices. After the macroblocks of the last slice group have been assigned, the process begins again from the first slice group. The process ends when all the macroblocks of a picture have been assigned. For example, to signal interleaved slices in a QCIF picture, the number of slice groups is two and run_length of 11 for both slice groups.

mb_allocation_map_type 1 is used to define a dispersed macroblock allocation. The macroblock allocation map is formed using the following formula, where n is the number of columns in the picture (in macroblocks) and p is the number of slice groups to be coded. Specifically, macroblock position x is assigned to slice group S according to the equation

$$S = \begin{cases} [(x\%n) + 1]\%p & \text{for } (x/n) \text{ even} \\ [(x\%n) + 1 + p/2]\%p & \text{for } (x/n) \text{ odd} \end{cases} \quad (8-1)$$

where “%” and “/” represent the modulo operation and division with truncation respectively.

mb_allocation_map_type 2 is used to explicitly assign a slice group to each macroblock location in raster scan order.

slice_group_id identifies a slice group of a macroblock ranging from 0 to 7.

num_ref_pic_active_fwd_minus1 specifies the number of reference pictures minus 1 in the forward reference set that are used to decode the picture.

num_ref_pic_active_bwd_minus1 specifies the number of reference pictures minus 1 in the backward reference set that are used to decode the picture.

slice_qp_minus26 specifies the value of the QP_Y for the MBs in a I, P and B slice unless modified by the value of slice_qp_delta in the slice header and/or delta_qp in the macroblock layer. The decoded value of this parameter shall be in the range of -26 to +25, inclusive. From this value, the initial QP_Y parameter for the slice is computed as:

$$QP_Y = 26 + \text{slice\_qp\_minus26} \quad (8-1)$$

The value of QP_Y is initialized to the above result and this value is used for the decoding of each macroblock in the slice unless updated by a slice_qp_delta in the slice header and/or a delta_qp sent in the macroblock layer.

slice_qp_s_minus26 specifies the value of the QPS_Y for the MBs in a SP and SI slice unless modified by the value of slice_qp_delta in the slice header. The decoded value of this parameter shall be in the range of -26 to +25, inclusive. The QPS_Y parameter for the slice is computed from slice_qp_s_minus26 as:

$$QPS_Y = 26 + \text{slice\_qp\_s\_minus26} \quad (8-2)$$

The value of QPS_Y is initialized to the above result and this value is used for the decoding of each macroblock in the slice unless updated by a slice_qp_delta in the slice header.

8.2.x Picture delimiter RBSP semantics

The picture delimiter may be used to signal the boundary between pictures, i.e., if present, it shall be inserted before the first NAL unit of a picture in decoding order. There is no normative decoder process associated with the picture delimiter.

picture_coding_type signals which slice coding types are used in the following picture in decoding order. The picture_coding_type is coded according to Table QQ. The table also shows the slice coding types that can occur in a picture with a given type.

Table 8-x – Meaning of picture_coding_type

picture_coding_type	Meaning	Allowed slice_coding_types
000	Intra	Intra
001	Inter	Intra, Inter
010	Bipred	Intra, Inter, Bipred
011	SI	SI

Error! No text of specified style in document.

100	SP	SI, SP
101	Mixed-I	Intra, SI
110	Mixed-P	Intra, SI, Inter, SP
111	Mixed-B	Intra, SI, Inter, SP, Bipred

non_stored_content_flag equal to 1 indicates that the picture is not stored in the multi-frame buffer.

8.2.x Filler data RBSP semantics

The filler data RBSP contains bytes whose value shall be equal to 0xFF.

8.2.4 Supplemental enhancement information RBSP semantics

Supplemental Enhancement Information (SEI) contains information that is not necessary to decode VCL data correctly but is helpful for decoding or presentation purposes.

An SEI element contains one or more SEI messages. Each SEI message consists of a SEI header and SEI payload. The type and size of the SEI payload are coded using an extensible syntax. The SEI payload size is indicated in bytes. Valid SEI payload types are listed in Annex C.

The SEI payload may have a SEI payload header. For example, a payload header may indicate to which picture the particular data belongs. The payload header shall be defined for each payload type separately. Definitions of SEI payloads are specified in Annex C.

An SEI message is associated with the next slice or data partition RBSP in transmission order.

byte_ff is a byte equal to 0xFF identifying a need for a longer representation of the syntax structure it is used within.

last_payload_type_byte identifies the payload type of the last entry in an SEI message.

last_payload_size_byte identifies the size of the last entry in an SEI message.

8.2.6 Slice layer RBSP semantics

The slice layer RBSP consists of a slice header defined in subclause 8.3 and the following slice data.

8.2.7 Data partition RBSP semantics

When data partitioning is in use, the coded data for a single slice is divided into three separate partitions. Partition A contains the header symbols of all coded MBs, partition B the intra Coded Block Patterns (coded_block_patterns) and coefficients, and partition C the inter coded block patterns and coefficients.

slice_id Each slice of a picture is associated a unique slice identifier within the picture. The first coded slice of the picture shall have identifier 0 and the identifier shall be incremented by one per each coded slice. Note that the coding order of slices may not be identical to the normal scan order.

8.3 Slice header semantics

picture_parameter_set_id indicates the picture parameter set in use.

frame_num labels the frame. frame_num shall be incremented by 1 for each coded picture in coding order, in modulo MAX_frame_num operation, relative to the frame_num of the previous stored frame in coding order. For non-stored frames, frame_num shall be incremented from the value in the most temporally recent stored frame which precedes the non-stored frame in coding order. The frame_num serves as a unique ID for each frame stored in the multi-frame buffer. Therefore, a frame cannot be kept in the buffer after its frame_num has been used by another frame unless it has been assigned a long-term frame index as specified below. No frame_num of a frame to be added to the multi-frame buffer shall equal to any other among the short-term frames in the multi-frame buffer. A decoder which encounters a frame number on a current frame having a value equal to the frame number of some other short-term stored frame in the multi-frame buffer should treat this condition as an error.

picture_structure identifies the picture structure.

Code_number=0: Progressive frame picture.

Code_number=1: Top field picture.

Code_number=2: Bottom field picture.

Code_number=3: Interlaced frame picture, whose top field precedes its bottom field in time.

Code_number=4: Interlaced frame picture, whose bottom field precedes its top field in time.

Note that when top field and bottom field pictures are coded for a frame, the one that is decoded first is the one that occurs first in time.

first_mb_in_slice specifies the macroblock position of the first macroblock contained in this slice.

slice_coding_type indicates the coding type of the slice.

Table 8-x – Meaning of slice_coding_type

slice_coding_type	Meaning
0	Inter
1	Bipred
2	Intra
3	SP
4	SI

coded_poc carries the picture order count coded in modulo max_poc arithmetic. The first picture of an independent GOP in decoding order shall have coded_poc equal to 0. The size of the coded_poc parameter is $\log_2 \text{max\_poc\_minus4} + 4$ bits.

delta_poc signals the picture order count difference compared to the expected picture order count as described in section QQ.

num_ref_pic_active_override_flag equal to zero indicates that the num_ref_pic_active_fwd_minus1 and num_ref_pic_active_bwd_minus1 specified in the referred picture parameter set are in effect. num_ref_pic_active_override_flag equal to one indicates that the num_ref_pic_active_fwd_minus1 and num_ref_pic_active_bwd_minus1 specified in the referred picture parameter set are overridden by the corresponding values given in the parameter set.

num_ref_pic_active_fwd_minus1 specifies the number of reference pictures minus 1 in the forward reference set that are used to decode the picture.

num_ref_pic_active_bwd_minus1 specifies the number of reference pictures minus 1 in the backward reference set that are used to decode the picture.

slice_qp_delta specifies the value of the QP_Y for the MBs in the slice unless modified by the value of delta_qp in the macroblock layer. From this value, the initial QP_Y parameter for the slice is computed as:

$$QP_Y = 26 + \text{slice\_qp\_minus26} + \text{slice\_qp\_delta} \quad (8-x)$$

The initial decoded QP_Y parameter shall be in the range of 0 to 51, inclusive. The value of QP_Y is initialized to the above result and this value is used for the decoding of each macroblock in the slice unless updated by a delta_qp sent in the macroblock layer.

sp_for_switch_flag controls the value of scaled transform coefficients in a manner that enables switching between different video bitstreams using an SP picture.

slice_qp_s_delta is signalled for SP and SI slices. The QPS_Y parameter for the slice is computed as:

$$QPS_Y = 26 + \text{slice\_qp\_s\_minus26} + \text{slice\_qp\_s\_delta} \quad (8-2)$$

The value of QPS_Y shall be in the range of 0 to 51, inclusive. This value of QPS_Y is used for the decoding of all macroblocks in the slice.

8.4 Reference picture selection layer (rps_layer) semantics

The contents of the rps_layer shall be repeated in each slice header of a picture.

Error! No text of specified style in document.

9 Decoding process

9.x NAL unit delivery order

This section presents the requirements for the NAL unit delivery order to the decoder.

An independent GOP parameter set shall be delivered before it is referred.

A picture parameter set shall be delivered before it is referred.

No slices or data partitions of one picture shall be delivered in between slices or data partitions of another picture. Pictures shall be delivered in decoding order.

Depending on the profile in use, arbitrary slice ordering may or may not be allowed. If arbitrary slice ordering is allowed, slices and data partitions of one picture may be delivered in any order. If arbitrary slice ordering is not allowed, slices shall be delivered in raster scan order, and data partition A of a slice shall be delivered before data partition B of the slice which shall be delivered before data partition C of the slice.

A picture delimiter, if present, shall be delivered as the next NAL unit after the last slice or data partition of a picture. "QQ: "shall be delivered immediately before the SEIs and slices of a given picture"

9.x NAL unit decoding

A received independent GOP parameter set information unit shall come into effect just before the next IDR picture in decoding order is decoded. If independent GOP parameter sets containing the same identifier is received more than once between one IDR point and the immediately following IDR point in decoding order, then the last such independent GOP parameter set in decoding order shall become the effective value for that parameter set identifier.

A received picture parameter set with a given picture parameter set identifier shall come into effect just prior to the first slice of the next picture in decoding order. If a picture parameter set containing the same identifier is received more than once between the first slice of a picture and prior to the last slice of that picture, then the last such picture parameter set received before the first slice of the picture for which it comes into effect shall become the effective value for that parameter set.

9.2 Slice decoding

Decoding of a new picture is started, if one of the following three conditions is true:

1. The frame number of the slice to be decoded is different from the frame number of the previously decoded slice.
2. The frame number of the slice to be decoded is the same as the previously decoded slice, poc_coding_type is 1, and coded_poc is different from the coded_poc in the previously decoded slice.
3. The frame number of the slice to be decoded is the same as the previously decoded slice, poc_coding_type is 2, and if delta_poc is different from the delta_poc in the previously decoded slice.

Each coded picture is associated with a picture order count, called poc, which is a 32-bit signed integer. The first decoded picture of an independent GOP shall have poc equal to 0. The poc of each stored picture shall be stored as long as the picture stays in the multi-frame buffer.

If poc_coding_type is 1, the decoder shall maintain a picture order count offset, called poc_offset, which is a 32-bit signed integer. The poc_offset shall be zero for the first decoded picture of an independent GOP.

If poc_coding_type is 1 and if the decoding of a new picture is started, the poc_offset is updated according to equation 8-QQ. The poc of the picture is calculated according to equation 8-QQ. The coded_poc of the previous picture in decoding order is herein called previous_poc in equation 8-QQ.

$$\begin{aligned} &\text{if (coded\_poc < previous\_poc \&\& previous\_poc - coded\_poc \>= max\_poc / 2)} \\ &\quad \text{poc\_offset = poc\_offset + max\_poc} \\ &\text{else if (coded\_poc > previous\_poc \&\& coded\_poc - previous\_poc \>= max\_poc / 2)} \\ &\quad \text{poc\_offset = poc\_offset - max\_poc} \end{aligned} \quad (8\text{-}QQ)$$

$$\text{poc = poc\_offset + coded\_poc} \quad (8\text{-}QQ)$$

If poc_coding_type is 2, the decoder shall maintain a frame_num_offset that is a 32-bit unsigned integer. The frame_num_offset of the first picture of an independent GOP shall be zero.

16 Error! No text of specified style in document.

If poc_coding_type is 2 and if the decoding of a new picture is started, the poc of the picture is calculated according to equation 8-QQ. abs_frame_num and poc_cycle_count in equation 8-QQ and 8-QQ are 32-bit unsigned integers. The frame_num of the previous picture in decoding order is herein called previous_frame_num.

$$\begin{aligned} &\text{if (frame\_num < previous\_frame\_num)} \\ &\quad \text{frame\_num\_offset} = \text{frame\_num\_offset} + \text{max\_frame\_num} \end{aligned} \quad (8\text{-QQ})$$

$$\text{abs\_frame\_num} = \text{frame\_num\_offset} + \text{frame\_num} \quad (8\text{-QQ})$$

$$\text{poc\_cycle\_count} = \text{abs\_frame\_num} / \text{num\_frames\_in\_poc\_cycle} \quad (8\text{-QQ})$$

$$\text{frame\_num\_in\_poc\_cycle} = \text{abs\_frame\_num} \% \text{num\_frames\_in\_poc\_cycle} \quad (8\text{-QQ})$$

$$\begin{aligned} &\text{if (nal\_priority} == 0) \\ &\quad \text{poc} = \text{poc\_cycle\_count} \times \text{expected\_delta\_per\_poc\_cycle} + \\ &\quad \quad \text{expected\_delta\_poc\_in\_cycle} + \\ &\quad \quad \text{delta\_poc} + \\ &\quad \quad \text{expected\_delta\_for\_non\_stored\_picture} \\ &\text{else} \\ &\quad \text{poc} = \text{poc\_cycle\_count} \times \text{expected\_delta\_per\_poc\_cycle} + \\ &\quad \quad \text{expected\_delta\_poc\_in\_cycle} + \\ &\quad \quad \text{delta\_poc} \end{aligned} \quad (8\text{-QQ})$$

poc, poc_offset, frame_num_offset and abs_frame_num shall not under- or overflow.

The macroblock address of the first macroblock of the slice is used as an index to the macroblock allocation map to determine which slice group the slice belongs to. Using the calculated macroblock address, the macroblock can be reconstructed directly. Since no other macroblock belonging to this slice has been decoded (as it is the first macroblock after the slice header), all in-picture prediction mechanisms are reset before decoding and reconstruction.

When mb_allocation_map_type equals to zero, the macroblock address of the next macroblock is implicitly the current macroblock address plus 1. When mb_allocation_map_type is not zero, the next macroblock address is determined by searching the macroblock allocation map for the next macroblock in raster-scan order belonging to the slice group being decoded.

A picture reference index parameter (ref_idx_fwd or ref_idx_bwd) is a relative index into an ordered set of reference pictures. For the coding of a P picture, there is one such set of reference pictures, called the forward reference set. For the coding of a B picture, there are two such sets of reference pictures, called the forward and backward reference sets.

The remapping_of_pic_nums_indicator, abs_diff_pic_numbers, and long_term_pict_index fields allow the order of that relative indexing into the multi-frame buffer to be temporarily altered from the default index order for the decoding of a particular slice. An remapping_of_pic_nums_indicator "end loop" indication indicates the end of a list of re-ordering commands for the forward or backward reference set.

C.2.x MMCO repetition syntax

mmco_repetition (PayloadType, PayloadSize) {	Category	Descriptor
original_frame_num	7	u(v)
mmco_layer()		
}		

C.3.x MMCO repetition semantics

original_frame_num identifies the picture were the repeated memory management control operation originally occurred.

Error! No text of specified style in document.

mmco_layer() shall contain a copy of the memory management control operation of the picture whose frame_num was original_frame_num.