|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  7th Meeting: Geneva, CH, 21-30 November, 2011 | Document: JCTVC-G330 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Syntax elements in adaptation parameter set** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Ying Chen Ye-Kui Wang Marta Karczewicz  5775 Morehouse Dr San Diego, CA 92121 USA | Tel: Email: | 1-858-845-6589 [cheny@qualcomm.com](mailto:cheny@qualcomm.com)  1-858-651-8345 [yekuiw@qualcomm.com](mailto:yekuiw@qualcomm.com)  1-858-658-5673 [martak@qualcomm.com](mailto:martak@qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This document discusses what syntax elements to be included into the Adaptation Parameter Set (APS). Currently, the APS may include ALF and SAO parameters. In this document, inclusion of other syntax elements, namely 1) reference picture list construction related syntax elements; 2) weighted prediction related syntax elements; 3) decoded picture buffer management related syntax elements; and 4) quantization matrices table, in APS is discussed.

# Reference picture list construction related syntax elements

The reference picture list construction (RPLC) related syntax elements include the following syntax elements, currently present in the slice header:

* num\_ref\_idx\_lX\_active\_minus1 (with X equal to 0 or 1)
* ref\_pic\_list\_modification( )
* ref\_pic\_list\_combination( )

In many cases, the reference picture lists are constructed in a same way for slice in one picture. It is more efficient in these cases to put the syntax elements into the APS.

The following options are possible, depending on whether different slices in one picture can refer to different APS’s:

1. If all slices in one picture must refer to the same APS, then there are two options:
   1. RPLC syntax elements are only present in the slice header, not in the APS.
   2. RPLC syntax elements are always present in the APS, and optionally present in the slice header. For a slice wherein the slice header includes RPLC syntax elements, their values override those in the referred APS.
2. If different slices in one picture may refer to different APS’s, then RPLC syntax elements are only present in the APS, and no need to be present in the slice header.

# Weighted prediction related syntax elements

The weighted prediction (WP) related syntax elements, when available, fall into the same category as RPLC related syntax elements, and thus the options in Section 1 apply.

# Decoded picture buffer management related syntax elements

The decoded picture buffer (DPB) management in HEVC Working Draft (WD) 4 is the same as in AVC, wherein decoded reference picture marking syntax table is signaled as part of the slice header.

The reference picture set (RPS) based DPB management mechanism as specified in the AHG21 output text, as included in JCTVC-G021, was developed based on the basic idea in JCTVC-F493. For RPS related syntax elements, they must be the same for all slices in one picture. Thus, the options here, regardless of whether different slices in one picture can refer to different APS’s, are:

1. The picture-level RPS related syntax elements are always present in the slice header, not in the APS.
2. The picture-level RPS related syntax elements are always present in the APS, not in the slice header. In this case, there are two options.
   1. Keep the RPS’s signaled in the PPS thus a RPS in APS can refer to a RPS signaled in the PPS.
   2. Do not include any RPS related syntax elements in the PPS. The advantage is that the RPS related syntax would only be present in the APS and hence the overall syntax looks simpler. The disadvantage is that the required number of APS’s is increased, potentially to multiple times larger. The APS overhead, if APS also contains other types of information, is increased, potentially to multiple times higher.

# Quantization matrices table

In AVC, quantization matrices are signaled both in SPS and PPS. To our knowledge, in the HEVC context, all slices of a picture have the same quantization matrices table, which may have up to 24 matrices. Thus, the options here, regardless of whether different slices in one picture can refer to different APS’s, are:

1. The quantization matrices table is always present in the slice header, not in the APS.
2. The quantization matrices table is always present in the APS, not in the slice header.

# Syntax design for specific options mentioned above

The syntax and semantics provided below are for the following of the above discussed options:

* 1.b in Section 1 for RPLC
* 2.a in Section 3 for RPS
* 2 in Section 4.

## Adaptation parameter set RBSP for APS

|  |  |
| --- | --- |
| aps\_rbsp( ) { | Descriptor |
| **aps\_id** | ue(v) |
| **aps\_sample\_adaptive\_offset\_flag** | u(1) |
| **aps\_adaptive\_loop\_filter\_flag** | u(1) |
| if( aps\_sample\_adaptive\_offset\_flag || aps\_adaptive\_loop\_filter\_flag ) { |  |
| **aps\_cabac\_use\_flag** | u(1) |
| if( aps\_cabac\_use\_flag ) { |  |
| **aps\_cabac\_init\_idc** | ue(v) |
| **aps\_cabac\_init\_qp\_minus26** | se(v) |
| } |  |
| } |  |
| if( slice\_type = = P | | slice\_type = = B ) { |  |
| **num\_ref\_idx\_active\_override\_flag** | u(1) |
| if( num\_ref\_idx\_active\_override\_flag ) { |  |
| **num\_ref\_idx\_l0\_active\_minus1** | ue(v) |
| if( slice\_type = = B ) |  |
| **num\_ref\_idx\_l1\_active\_minus1** | ue(v) |
| } |  |
| } |  |
| ref\_pic\_list\_modification( ) |  |
| ref\_pic\_list\_combination( ) |  |
| ref\_picture\_set( ) |  |
| if (quantizaton\_matrix\_table\_present\_flag ) |  |
| quantization\_matrix\_table( ) |  |
| /\* Insert non-CABAC stuff above this line \*/ |  |
| if( aps\_adaptive\_loop\_filter\_flag ) { |  |
| **alf\_data\_byte\_count** /\* to enable skipping past data without parsing it \*/ | u(8) |
| /\* byte\_align() this byte align to happen between the non-CABAC and CABAC parts of the alf\_param() Once there is an all CABAC alf\_param(), enable this byte\_align() \*/ |  |
| alf\_param( ) |  |
| byte\_align() |  |
| } |  |
| /\* insert CABAC stuff below this line; make sure its byte-aligned \*/ |  |
| if( aps\_sample\_adaptive\_offset\_flag ) { |  |
| **sao\_data\_byte\_count** /\* to enable skipping past data without parsing it \*/ | u(8) |
| byte\_align () |  |
| sao\_param( ) |  |
| /\* byte\_align() this final byte align unnecessary as being taken care of by rbsp\_trailing\_bits() \*/ |  |
| rbsp\_trailing\_bits( ) |  |
| } |  |

## Slice header syntax

|  |  |
| --- | --- |
| slice\_header( ) { | Descriptor |
| **lightweight\_slice\_flag** | u(1) |
| if( !lightweight\_slice\_flag ) { |  |
| **slice\_type** | ue(v) |
| **pic\_parameter\_set\_id** | ue(v) |
| **aps\_id** | ue(v) |
| **rplc\_override\_flag** | u(1) |
| **frame\_num** | u(v) |
| if( IdrPicFlag ) |  |
| **idr\_pic\_id** | ue(v) |
| if( pic\_order\_cnt\_type = = 0 ) |  |
| **pic\_order\_cnt\_lsb /\*** | u(v) |
| if( rplc\_override\_flag ) { |  |
| if( slice\_type = = P | | slice\_type = = B ) { |  |
| **num\_ref\_idx\_active\_override\_flag** | u(1) |
| if( num\_ref\_idx\_active\_override\_flag ) { |  |
| **num\_ref\_idx\_l0\_active\_minus1** | ue(v) |
| if( slice\_type = = B ) |  |
| **num\_ref\_idx\_l1\_active\_minus1** | ue(v) |
| } |  |
| } |  |
| ref\_pic\_list\_modification( ) |  |
| ref\_pic\_list\_combination( ) |  |
| } |  |
| } |  |
| if( entropy\_coding\_mode\_flag && slice\_type != I) |  |
| **cabac\_init\_idc** | ue(v) |
| **first\_slice\_in\_pic\_flag** | u(1) |
| if( first\_slice\_in\_pic\_flag == 0 ) |  |
| **slice\_address** | u(v) |
| if( !lightweight\_slice\_flag ) { |  |
| **slice\_qp\_delta** | se(v) |
| if( deblocking\_filter\_control\_present\_flag ) { |  |
| **disable\_deblocking\_filter\_idc** |  |
| if( disable\_deblocking\_filter\_idc != 1 ) { |  |
| **slice\_alpha\_c0\_offset\_div2** |  |
| **slice\_beta\_offset\_div2** |  |
| } |  |
| } |  |
| if( slice\_type = = B ) |  |
| **collocated\_from\_l0\_flag** | u(1) |
| if( adaptive\_loop\_filter\_enabled\_flag && aps\_adaptive\_loop\_filter\_flag ) { |  |
| byte\_align() |  |
| alf\_cu\_control\_param( ) |  |
| byte\_align() |  |
| } |  |
| } |  |

## Semantics

The reference picture list construction related syntax elements have the same semantics as those in the slice header, when an APS is referred by a slice header.

quantizaton\_matrix\_table\_present\_flag is signaled in the SPS or PPS to indicate whether quantization matrix table is present or not.

**rplc\_override\_flag** equal to 1 specifies that the reference picture list construction related syntax elements and syntax structures, including the num\_ref\_idx\_active\_override\_flag, the num\_ref\_idx\_lx\_active\_minus (with x being 0 or 1), ref\_pic\_list\_modification( ) and ref\_pic\_list\_combination( ), are signaled in the slice header and overwrite those in the APS. rplc\_override\_flag equal to 0 specifies that the above syntax elements and syntax structures are not signaled in the slice header and the syntax elements signaled in the APS are used for the current slice.

# Patent rights declaration(s)

**Qualcomm Incorporated may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**