|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  10th Meeting: Stockholm, SE, 11 – 20 July 2012 | Document: JCTVC-J0108 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **AHG9: Splicing-friendly coding of some parameters** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Ye-Kui Wang Ying Chen  5775 Morehouse Drive San Diego, CA 92121 USA | Tel: Email: | 1-858-651-8345 [yekuiw@qualcomm.com](mailto:yekuiw@qualcomm.com)  1-858-845-6589 [cheny@qualcomm.com](mailto:cheny@qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

During splicing, two bitstreams may refer to few parameter sets with the same ID for each type of parameter sets but with different content. This document proposes that all parameter set IDs are fixed-length coded, and placed before any entropy-coded syntax elements in each parameter set or coded slice NAL unit. Furthermore, it is proposed that the syntax element no\_output\_of\_prior\_pics\_flag and the syntax element rap\_pic\_id are placed before any entropy-coded syntax elements in the slice header, and the syntax element rap\_pic\_id is fixed-length coded. It is asserted that the changes enable lightweight splicing of bitstreams.

# Introduction

In HEVC, the video, sequence, picture and adaptation parameter set mechanism in HEVC decouples the transmission of infrequently changing information from the transmission of coded block data. Video, sequence, picture and adaptation parameter sets may, in some applications, be conveyed "out-of-band", i.e., not transported together with the units containing coded video data. Out-of-band transmission is typically reliable.

In latest HEVC draft specification, an identifier of a video sequence parameter set (VPS), sequence parameter set (SPS), picture parameter set (PPS) or adaptation parameter set (APS) is coded using 'ue(v)'. Each SPS includes an SPS ID and a VPS ID, each PPS includes a PPS ID and an SPS ID, and each slice header includes a PPS ID and possibly an APS ID.

Bitstream splicing refers to the concatenation of two or more bitstreams or parts thereof. For example, a first bitstream may be appended by a second bitstream, possibly with some modifications to either one or both of the bitstreams to generate a spliced bitstream. The first coded picture in the second bitstream is also referred to as the splicing point. Therefore, pictures since the splicing point in the spliced bitstream were originated from the second bitstream while pictures preceding the splicing point in the spliced bitstream were originated from the first bitstream.

When the splicing point is a CRA picture, in the splice bitstream, the splicing point CRA picture should be changed to be a BLA picture.

Splicing of bitstreams is performed by bitstream splicers. Bitstream splicers are often lightweight and much less intelligent than encoders. For example, they may not be equipped with entropy decoding and encoding capabilities.

# Problems

## Changing a CRA picture to a BLA picture

The values of the following syntax elements need to be changed in changing of a CRA picture to a BLA picture:

* nal\_unit\_type in the NAL unit header (from 4 to 6, or from 5 to 7)
* no\_output\_of\_prior\_pics\_flag in the slice header (to the desirable value)

1. If the previous picture in decoding order is a RAP picture and the rap\_pic\_id of the current slice is equal to the rap\_pic\_id of the previous picture, the following applies:
   * If the next picture in decoding order is not a RAP picture, the value of rap\_pic\_id of the current slice is changed to be different than the rap\_pic\_id of the previous picture in decoding order.
   * Otherwise (the next picture in decoding order is a RAP picture), the value of rap\_pic\_id of the current picture is changed to be a value that is different than the rap\_pic\_id of both the previous picture and the next picture in decoding order.

Among the above, changing the value of nal\_unit\_type in the NAL unit header is simple, without the need of entropy decoding and encoding. However, changing the value of no\_output\_of\_prior\_pics\_flag and rap\_pic\_id requires entropy decoding and encoding, as there are entropy-coded syntax elements earlier in the slice header. For rap\_pic\_id, as it is currently ue(v) coding, changing of the value may need to shift the following bits in the coded slice NAL unit. This can be cumbersome for lightweight bitstream splicers.

## Changing the value of a parameter set ID

Bitstreams often use few SPSs and few PPSs, or even just one SPS and one PPS and use the smallest SPS ID value 0 and the smallest PPS ID value 0. In this case, if two bitstreams or parts thereof are spliced, it is likely that the same SPS or PPS ID is referred to by the splicing point and by the picture immediately preceding the spicing point picture, while actually two different SPSs or PPSs are used. In this case, in the spliced bitstream, the SPS with that particular SPS ID referred to by the picture immediately preceding the spicing point is effectively updated by the SPS referred to by the splicing point picture. Consequently, SPSs of the spliced bitstream cannot be put in the beginning of the spliced bitstream, and out-of-band transmission of SPSs is impossible.

However, since only a few different SPSs are used anyway, it would be beneficial to enable out-of-band transmission of SPSs for the spliced bitstream.

One way to enable out-of-band transmission of SPSs for the spliced bitstream is to change the SPS IDs in the second bitstream that are the same as some SPS IDs in the first bitstream to be different, such that each SPS is uniquely identified by a particular SPS ID throughout the spliced bitstream. The same applies to other types of parameter sets, i.e., VPS, PPS and APS.

In changing the value of a parameter set ID, when the ID is entropy coded (e.g. using 'ue(v)'), then entropy decoding and entropy encoding are always required, and shifting of all bits after the ID in the coded slice and the calling of the byte alignment processes for slice header and entries of tiles or wavefronts (when present) are typically required. This can be very cumbersome for lightweight bitstream splicers, which are typically much less intelligent than encoders. Furthermore, this operation is needed for all coded slices referring to that particular parameter set ID.

# Proposal

To solve the above problems and to enable lightweight bitstream splicing operations without entropy decoding and encoding and other entropy coding related operations, the following syntax changes are proposed. Basically, all parameter set IDs are proposed to be fixed-length coded and placed before any entropy-coded syntax elements in each parameter set or coded slice NAL unit, and the syntax elements no\_output\_of\_prior\_pics\_flag and rap\_pic\_id are moved ahead to be before any entropy-coded syntax elements in the slice header, and rap\_pic\_id is fixed-length coded.

## Syntax changes

|  |  |
| --- | --- |
| video\_parameter\_set\_rbsp( ) { | Descriptor |
| **vps\_max\_temporal\_layers\_minus1** | u(3) |
| **vps\_max\_layers\_minus1** | u(5) |
| **video\_parameter\_set\_id** | u(4) |
| **vps\_temporal\_id\_nesting\_flag** | u(1) |
| **...** |  |
| } |  |

|  |  |
| --- | --- |
| seq\_parameter\_set\_rbsp( ) { | Descriptor |
| **profile\_space** | u(3) |
| **profile\_idc** | u(5) |
| **constraint\_flags** | u(16) |
| **level\_idc** | u(8) |
| for( i = 0; i < 32; i++ ) |  |
| **profile\_compatability\_flag[** i **]** | u(1) |
| **seq\_parameter\_set\_id** | u(5) |
| **video\_parameter\_set\_id** | u(4) |
| **...** |  |
| } |  |

|  |  |
| --- | --- |
| pic\_parameter\_set\_rbsp( ) { | Descriptor |
| **pic\_parameter\_set\_id** | u(8) |
| **seq\_parameter\_set\_id** | u(5) |
| **sign\_data\_hiding\_flag** | u(1) |
| **...** |  |
| } |  |

|  |  |
| --- | --- |
| aps\_rbsp( ) { | Descriptor |
| **aps\_id** | u(5) |
| **aps\_adaptive\_loop\_filter\_flag** | u(1) |
| ... |  |
| } |  |

|  |  |
| --- | --- |
| slice\_header( ) { | Descriptor |
| **first\_slice\_in\_pic\_flag** | u(1) |
| **pic\_parameter\_set\_id** | u(8) |
| **aps\_id** | u(5) |
| if( RapPicFlag ) { |  |
| **no\_output\_of\_prior\_pics\_flag** | u(1) |
| **rap\_pic\_id** | u(5) |
| } |  |
| if( !first\_slice\_in\_pic\_flag ) |  |
| **slice\_address** | u(v) |
| if( dependent\_slice\_enabled\_flag && !first\_slice\_in\_pic\_flag ) |  |
| **dependent\_slice\_flag** | u(1) |
| if( !dependent\_slice\_flag ) { |  |
| **slice\_type** | ue(v) |
| if( output\_flag\_present\_flag ) |  |
| **pic\_output\_flag** | u(1) |
| if( **separate\_colour\_plane\_flag** = = 1 ) |  |
| **colour\_plane\_id** | u(2) |
| ~~if( RapPicFlag ) {~~ |  |
| **~~no\_output\_of\_prior\_pics\_flag~~** | ~~u(1)~~ |
| **~~rap\_pic\_id~~** | ~~ue(v)~~ |
| ~~}~~ |  |
| ... |  |
| if( slice\_type = = P | | slice\_type = = B ) { |  |
| ... |  |
| } |  |

## Semantics changes

### Video parameter set semantics

**video\_parameter\_set\_id** identifies a video parameter set.

### Sequence parameter set semantics

**seq\_parameter\_set\_id** identifies a sequence parameter set.

**video\_parameter\_set\_id** refers to the active video parameter set.

### Picture parameter set semantics

**pic\_parameter\_set\_id** identifies a picture parameter set.

**seq\_parameter\_set\_id** refers to the active sequence parameter set.

### Adaptation parameter set semantics

**aps\_id** identifies an adaptation parameter set. The value of aps\_id shall be greater than 0 for any adaptation parameter set.

### Slice header semantics

**pic\_parameter\_set\_id** refers to the active picture parameter set.

**aps\_id** greater than 0 refers to the active adaptation parameter set. aps\_id equal to 0 indicates that the current slice does not refer to an adaptation parameter set. The value of aps\_id shall be equal to 0 when adapative\_loop\_filter\_enabled\_flag is equal to 0.

**no\_output\_of\_prior\_pics\_flag** specifies how the previously-decoded pictures in the decoded picture buffer are treated after decoding of an IDR or a BLA picture. See Annex C. When the IDR or BLA picture is the first IDR or BLA picture in the bitstream, the value of no\_output\_of\_prior\_pics\_flag has no effect on the decoding process. When the IDR or BLA picture is not the first IDR or BLA picture in the bitstream and the value of pic\_width\_in\_luma\_samples or pic\_height\_in\_luma\_samples or sps\_max\_dec\_pic\_buffering[ sps\_max\_temporal\_layers\_minus1 ] derived from the active sequence parameter set is different from the value of pic\_width\_in\_luma\_samples or pic\_height\_in\_luma\_samples or sps\_max\_dec\_pic\_buffering[ sps\_max\_temporal\_layers\_minus1 ] derived from the sequence parameter set active for the preceding picture, no\_output\_of\_prior\_pics\_flag equal to 1 may (but should not) be inferred by the decoder, regardless of the actual value of no\_output\_of\_prior\_pics\_flag. When nal\_unit\_type is equal to 4 or 5, the value of no\_output\_of\_prior\_pics\_flag shall be equal to 1.

**rap\_pic\_id** identifies a RAP picture. The values of rap\_pic\_id in all the slices of a RAP picture shall remain unchanged. When two consecutive access units in decoding order are both RAP access units, the value of rap\_pic\_id in the slices of the first such RAP access unit shall differ from the rap\_pic\_id in the second such RAP access unit. ~~The value of rap\_pic\_id shall be in the range of 0 to 65535, inclusive.~~

# 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).**