|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  8th Meeting: San José, CA, USA, 1–10 February, 2012 | Document: JCTVC-H0504 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Comments on Reference Picture Lists Combination Syntax** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Hendry Yong Joon Jeon Seung Wook Park Sang Oh Jeong Byeong Moon Jeon  #221 Yangjae-dong, Seocho-gu,  Seoul 137-130,  Korea  Ying Chen Ye-Kui Wang Wei-Jung Chien  5775 Morehouse Dr San Diego, CA 92121 USA | Tel: Email: | [hendry.hendry@lge.com](mailto:hendry.hendry@lge.com)  [yongjoon.jeon@lge.com](mailto:yongjoon.jeon@lge.com)  [seungwook.park@lge.com](mailto:seungwook.park@lge.com)  [sangoh.jeong@lge.com](mailto:sangoh.jeong@lge.com)  [bm.jeon@lge.com](mailto:bm.jeon@lge.com)  1-858-845-6589 [cheny@qualcomm.com](mailto:cheny@qualcomm.com)  1-858-651-8345 [yekuiw@qualcomm.com](mailto:yekuiw@qualcomm.com)  1-858-845-1795 [wchien@qualcomm.com](mailto:wchien@qualcomm.com) |
| *Source:* | LG and Qualcomm Incorporated | | |

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

# Abstract

It was reported that there was some redundancy on the reference picture list combination syntax in the current HEVC WD. More specifically when the ref\_pic\_list\_comibination\_flag is equal to 0, it indicates that two reference picture lists are identical, but both lists are still signaled separately. In this document, it is proposed that the location of this flag be moved to an earlier place of the slice header, such that the redundancy can be removed by not signaling the reference picture list 1 when ref\_pic\_list\_comibination\_flag is equal to 0. Furthermore, the name of this flag is changed to avoid confusion.

# Introduction

In the current HEVC WD, ref\_pic\_list\_comibination\_flag is signalled for a B slice and when this flag equals to 1, either implicit reference picture lists combination or explicit reference picture list combination is used to generate a combined list; when this flag equal to 0, reference picture list 0 (RefPicList0) and reference picture list 1 (RefPicList1) are identical thus no list combination is needed.

Based on the current design, there are two ways to decide whether RefPicList0 and RefPicList1 are identical thus no reference picture list combination is needed. One way is to check the ref\_pic\_list\_combination\_flag, and the other way is to compare these two lists.

However, when ref\_pic\_list\_combination\_flag is equal to 0, there is no need to signal the RefPicList1. Such a redundancy is reported in the last JCTVC meeting in Geneva.

# Proposal

It is proposed that the ref\_pic\_list\_combination\_flag be moved to an earlier place of the slice header to avoid the redundancy and the semantics of this flag can be kept unchanged.

Note that the modifications are highlighted in green.

## Slice header syntax

|  |  |
| --- | --- |
| slice\_header( ) { | Descriptor |
| **entropy\_slice\_flag** | u(1) |
| if( !entropy\_slice\_flag ) { |  |
| **slice\_type** | ue(v) |
| **pic\_parameter\_set\_id** | ue(v) |
| **…** |  |
| if( slice\_type = = P | | slice\_type = = B ) { |  |
| **num\_ref\_idx\_active\_override\_flag** | u(1) |
| if( slice\_type = = B ) |  |
| **identical\_lists\_flag** | u(1) |
| if( num\_ref\_idx\_active\_override\_flag ) { |  |
| **num\_ref\_idx\_l0\_active\_minus1** | ue(v) |
| if( !identcial\_lists\_flag ) |  |
| **num\_ref\_idx\_l1\_active\_minus1** | ue(v) |
| } |  |
| } |  |
| ref\_pic\_list\_modification( ) |  |
| ref\_pic\_list\_combination( ) |  |
| **…** |  |
| } |  |

## Reference picture list modification syntax

|  |  |
| --- | --- |
| ref\_pic\_list\_modification( ) { | Descriptor |
| if( slice\_type % 5 != 2 && slice\_type % 5 != 4 ) { |  |
| **ref\_pic\_list\_modification\_flag\_l0** | u(1) |
| if( ref\_pic\_list\_modification\_flag\_l0 ) |  |
| do { |  |
| **modification\_of\_pic\_nums\_idc** | ue(v) |
| if( modification\_of\_pic\_nums\_idc = = 0 | |  modification\_of\_pic\_nums\_idc = = 1 ) |  |
| **abs\_diff\_pic\_num\_minus1** | ue(v) |
| else if( modification\_of\_pic\_nums\_idc = = 2 ) |  |
| **long\_term\_pic\_num** | ue(v) |
| } while( modification\_of\_pic\_nums\_idc != 3 ) |  |
| } |  |
| if( slice\_type % 5 = = 1 && !identical\_lists\_flag ) { |  |
| **ref\_pic\_list\_modification\_flag\_l1** | u(1) |
| if( ref\_pic\_list\_modification\_flag\_l1 ) |  |
| do { |  |
| **modification\_of\_pic\_nums\_idc** | ue(v) |
| if( modification\_of\_pic\_nums\_idc = = 0 | |  modification\_of\_pic\_nums\_idc = = 1 ) |  |
| **abs\_diff\_pic\_num\_minus1** | ue(v) |
| else if( modification\_of\_pic\_nums\_idc = = 2 ) |  |
| **long\_term\_pic\_num** | ue(v) |
| } while( modification\_of\_pic\_nums\_idc != 3 ) |  |
| } |  |
| } |  |

## Reference picture lists combination syntax

|  |  |
| --- | --- |
| ref\_pic\_list\_combination( ) { | Descriptor |
| if( slice\_type % 5 = = 1 ) {// b slice |  |
| **~~ref\_pic\_list\_combination\_flag~~** | ~~u(1)~~ |
| if( !identical\_lists\_flag  ~~ref\_pic\_list\_combination\_flag~~ ) { |  |
| **num\_ref\_idx\_lc\_active\_minus1** | ue(v) |
| **ref\_pic\_list\_modification\_flag\_lc** | u(1) |
| if( ref\_pic\_list\_modification\_flag\_lc) |  |
| for ( i =0; i <= num\_ref\_idx\_lc\_active\_minus1; i++ ) { |  |
| **pic\_from\_list\_0\_flag** | u(1) |
| **ref\_idx\_list\_curr** | ue(v) |
| } |  |
| } |  |
| ~~}~~ |  |
| } |  |

## Slice header semantics

**identcial\_lists\_flag** **~~ref\_pic\_list\_combination\_flag~~** equal to ~~1~~ 0 indicates that the reference picture list 0 and the reference picture list 1 are combined to be an additional reference picture lists combination used for the prediction units being uni-directional predicted. This flag equal to ~~0~~ 1 indicates that the reference picture list 0 and reference picture list 1 are identical thus reference picture list 0 is used as the reference picture lists combination. The reference picture lists combination is set to be empty at the start of the loop defined in ~~this~~ the reference picture lists combination syntax table.

Note that no matter the idential\_lists\_and\_combine\_flag (the previous ref\_pic\_list\_combination\_flag) is 0 or 1, there is a combined list. The difference is that when idential\_lists\_and\_combine\_flag is equal to 1, the combined list is set to reference picture list 0 and when idential\_lists\_and\_combine\_flag is equal to 0, the combined list is actually combined by the explicit mechanism or implicit mechanism based on the ref\_pic\_list\_modification\_flag\_lc.

# Discussion

Similar changes may be made for the prediction weights table syntax and the collocated\_from\_l0\_flag syntax in the slice header based on the same spirit as the proposal.

We don’t include in the proposal part of this document but we recommend the following syntax modifications to be examined and considered by the group.

## Prediction weight table syntax

|  |  |
| --- | --- |
| pred\_weight\_table( ) { | Descriptor |
| **luma\_log2\_weight\_denom** | ue(v) |
| if( chroma\_format\_idc != 0 ) |  |
| **chroma\_log2\_weight\_denom** | ue(v) |
| for( i = 0; i <= num\_ref\_idx\_l0\_active\_minus1; i++ ) { |  |
| **…** |  |
| } |  |
| if( slice\_type = = B && !identical\_lists\_flag ) |  |
| for( i = 0; i <= num\_ref\_idx\_l1\_active\_minus1; i++ ) { |  |
| **luma\_weight\_l1\_flag** | u(1) |
| if( luma\_weight\_l1\_flag ) { |  |
| **luma\_weight\_l1[** i **]** | se(v) |
| **luma\_offset\_l1[** i **]** | se(v) |
| } |  |
| if( chroma\_format\_idc != 0 ) { |  |
| **chroma\_weight\_l1\_flag** | u(1) |
| if( chroma\_weight\_l1\_flag ) |  |
| for( j = 0; j < 2; j++ ) { |  |
| **chroma\_weight\_l1[** i **][** j **]** | se(v) |
| **chroma\_offset\_l1[** i **][** j **]** | se(v) |
| } |  |
| **}** |  |
| } |  |
| } |  |

## Slice header syntax

|  |  |
| --- | --- |
| slice\_header( ) { | **Descriptor** |
| **…** |  |
| if( !entropy\_slice\_flag ) { |  |
| **…** |  |
| if( slice\_type = = B && !identical\_lists\_flag ) |  |
| **collocated\_from\_l0\_flag** | u(1) |
| **…** |  |
| } |  |
| **…** |  |
| } |  |

# Patent rights declaration(s)

**LG Electronics 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).**

**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).**