|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  9th Meeting: Geneva, CH, 27 April – 7 May 2012 | Document: JCTVC-I0335 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **On weighted prediction signaling** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Ying Chen Muhammed Coban Ye-Kui Wang Wei-Jung Chien  5775 Morehouse Dr  San Diego, CA 92121 USA | Tel: Email: | 1-858-845-6589 [cheny@qualcomm.com](mailto:cheny@qualcomm.com)  1-858-658-3937 [mcoban@qualcomm.com](mailto:mcoban@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:* | Qualcomm Incorporated | | |

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

# Abstract

In the current working draft of HEVC, for a B slice, prediction weights are either signalled for all the entries of the combined list or signalled for all the entries of reference picture list 0 and all the entries of reference picture list 1, depending on the ref\_pic\_list\_combination\_flag. More specifically, if list 0 and list 1 are identical and ref\_pic\_list\_combination\_flag is equal to 0 (if list 0 and list 1 are not identical, ref\_pic\_list\_combination\_flag must be equal to 1), separate weighted prediction parameters are signalled for each of list 0 and list 1, otherwise weighted prediction parameters for the combined list are signaled.

In this document, it is firstly proposed to make signaling of weighted prediction parameters independent of the value of ref\_pic\_list\_combination\_flag, as the combined list is constructed regardless of the value of ref\_pic\_list\_combination\_flag.

It is further proposed to adopt one of the four alternative signaling methods for weighted prediction parameters.

# Proposal

We first of all propose to make signaling of weighted prediction parameters independent of the value of ref\_pic\_list\_combination\_flag, as the combined list is constructed regardless of the value of ref\_pic\_list\_combination\_flag.

Then we propose to adopt one of the four different alternative signaling methods presented below.

## Method 1

Even when the weights for reference picture list 0 (List 0) and reference picture list 1 (List 1) are signalled, there are cases when weights for the combined list (List C) may still need to be signalled.

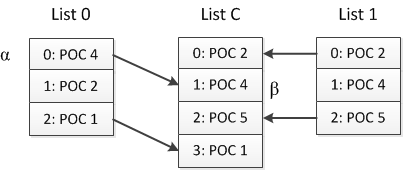


Figure 1: Different weights for an entry in List 0 and its mapped entry in List C.

Although an entry A in List 0 or List 1 is known to be mapped to an entry B in List C, it is possible that in certain scenarios, the weights used for entry A for bi-directional prediction differ from the weights used for entry B for uni-directional prediction. For example, as shown in Figure 1, α denotes the weights (including the weight for luma and the weight for chroma) for the first entry in List 0 (with POC 4), which is mapped to the second entry in List C. However, when this entry is used for uni-directional prediction from List C, the desirable weights β for this entry may be not equivalent to α.

To support this use case, while at the same time enabling efficient signaling of weights, including for other cases as well, it is proposed that three signaling mechanisms can be supported:

* To signal only the weights for List 0 and List 1 entries, and the weights of the entries in List C are derived to be the same as the respectively mapping entries in List 0 or List 1;
* To signal only the weights for List C entries, and the weights of the entries in List 0 and List 1 are derived to be the same as the respectively mapping entries List C, and it is required that the signaled combined list contains every reference picture in list 0 and list 1 when this signaling is in use;
* To signal the weights for List 0, List 1 and List C entries.

### Prediction weight table syntax

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

### Prediction weight table semantics

**wp\_weight\_signal\_idc** equal to 0 specifies that prediction weights for RefPicList0 and RefPicList1 are signalled, but prediction weights for the combined list are not signalled. wp\_weight\_signal\_idcequal to 1 specifies that prediction weights for the combined list are signalled, but prediction weights for RefPicList0 and RefPicList1 are not signalled. wp\_weight\_signal\_idc equal to 2 specifies that prediction weights for RefPicList0, RefPicList1, and the combined list are signalled. The value of wp\_weight\_signal\_idc shall be in the range of 0 to 2, inclusive. When not present, wp\_weight\_signal\_idc is inferred to be equal to 0.

## Method 2

In this method, two signaling mechanisms can be supported:

* To signal only the weights for List 0 and List 1 entries, and the weights of the entries in List C are derived to be the same as the respectively mapping entries in List 0 or List 1;
* To signal the weights for List C entries and entries in List 0 and List 1 which are not mapped to List C. For example, as shown in Figure 2, the weights (Wc0~Wc3) for all entries in List C are signalled, and the weights (W01, W11) for entries not mapped to List C are also signalled.

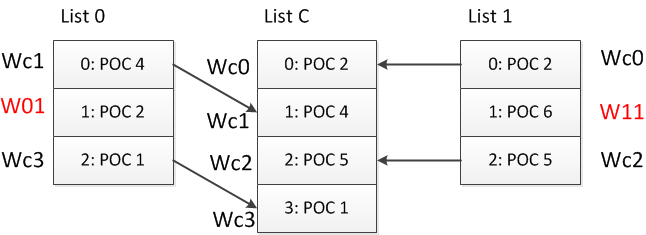


Figure 2: Weights for List C and the weights for unmapped entries in List 0/1.

### Prediction weight table syntax

|  |  |
| --- | --- |
| pred\_weight\_table( ) { | Descriptor |
| if ( slice\_type = = B ) |  |
| **weights\_from\_list0\_list1\_only\_flag** | u(1) |
| **luma\_log2\_weight\_denom** | ue(v) |
| if( chroma\_format\_idc != 0 ) |  |
| **delta\_chroma\_log2\_weight\_denom** | se(v) |
| if( slice\_type = = P | | slice\_type = = B ) { |  |
| for( i = 0; i <= num\_ref\_idx\_l0\_active\_minus1; i++ ) { |  |
| if ( weights\_from\_list0\_list1\_only\_flag | | !L0Mapped[ i ] ) { |  |
| **luma\_weight\_l0\_flag** | u(1) |
| if( luma\_weight\_l0\_flag ) { |  |
| **delta\_luma\_weight\_l0[** i **]** | se(v) |
| **luma\_offset\_l0[** i **]** | se(v) |
| } |  |
| if( chroma\_format\_idc != 0 ) { |  |
| **chroma\_weight\_l0\_flag** | u(1) |
| if( chroma\_weight\_l0\_flag ) |  |
| for( j =0; j < 2; j++ ) { |  |
| **delta\_chroma\_weight\_l0[** i **][** j **]** | se(v) |
| **delta\_chroma\_offset\_l0[** i **][** j **]** | se(v) |
| } |  |
| } |  |
| } |  |
| } |  |
| } |  |
| if( slice\_type = = B ) { |  |
| for( i = 0; i <= num\_ref\_idx\_l1\_active\_minus1; i++ ) { |  |
| if( weights\_from\_list0\_list1\_only\_flag | | !L1Mapped[ i ] ) { |  |
| **luma\_weight\_l1\_flag** | u(1) |
| if( luma\_weight\_l1\_flag ) { |  |
| **delta\_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++ ) { |  |
| **delta\_chroma\_weight\_l1[** i **][** j **]** | se(v) |
| **delta\_chroma\_offset\_l1[** i **][** j **]** | se(v) |
| } |  |
| } |  |
| } |  |
| } |  |
| if( !weights\_from\_list0\_list1\_only\_flag ) |  |
| for( i = 0; i <= num\_ref\_idx\_lc\_active\_minus1; i++ ) { |  |
| **luma\_weight\_lc\_flag** | u(1) |
| if( luma\_weight\_l1\_flag ) { |  |
| **delta\_luma\_weight\_lc[** i **]** | se(v) |
| **luma\_offset\_lc[** i **]** | se(v) |
| } |  |
| if( chroma\_format\_idc != 0 ) { |  |
| **chroma\_weight\_lc\_flag** | u(1) |
| if( chroma\_weight\_lc\_flag ) |  |
| for( j = 0; j < 2; j++ ) { |  |
| **delta\_chroma\_weight\_lc[** i **][** j **]** | se(v) |
| **delta\_chroma\_offset\_lc[** i **][** j **]** | se(v) |
| } |  |
| } |  |
| } |  |
| } |  |
| } |  |

### Prediction weight table semantics

**weights\_from\_list0\_list1\_only\_flag** equal to 1 specifies that prediction weights for all entries in RefPicList0 and RefPicList1 are signalled, but prediction weights for the combined list are not signalled. weights\_from\_list0\_list1\_only\_flag equal to 0 specifies that prediction weights for all entries in the combined list are signalled and prediction weights for entries in RefPicList0 or RefPicList1 are conditionally signalled.

If the entry RefPicListX[ i ] (X being 0 or 1) is mapped to one or more entry in the combined list, LXMapped[ i ] (X being 0 or 1) is derived to be 1, and the prediction weights for RefPicListX[ i ] is set to be the same as the prediction weights of the first mapping entry in the combined list. otherwise, LXMapped[ i ] is derived to be 0.

## Method 3

For a B slice, only the weights for the combined list are signalled regardless of whether or not the two lists are identical. It is required that the signaled combined list contains every reference picture in list 0 and list 1. The weights of the entries in List0 and List1 are derived to be the same as the respectively mapping entries List C.

### Prediction weight table syntax

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

### Prediction weight table semantics

No change except removal of semantics for removed syntax elements for signaling of prediction weights for list 1 entries.

## Method 4

For a B slice, two signaling mechanisms can be supported:

* To signal only the weights for List 0 and List 1 entries, and the weights of the entries in List C are derived to be the same as the respectively mapping entries in List 0 or List 1;
* To signal only the weights for List C entries, and the weights of the entries in List0 and List1 are derived to be the same as the respectively mapping entries List C, and it is required that the signaled combined list contains every reference picture in list 0 and list 1 when this signaling is in use.

### Prediction weight table syntax

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

### Prediction weight table semantics

**wp\_weight\_lc\_only\_flag** equal to 1 specifies that prediction weights for the combined list are signalled, but prediction weights for RefPicList0 or RefPicList1 are not signalled. wp\_weight\_lc\_only\_flagequal to 0 specifies that prediction weights for RefPicList0 and prediction weights for RefPicList1 are signalled, but prediction weights for the combined list are not signalled.

# 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).**