|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  14th Meeting: Vienna, AT, 25 July – 2 Aug. 2013 | Document: JCTVC-N0217r1 |

|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extensions**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  5th Meeting: Vienna, AT, 27 July – 2 Aug. 2013 | Document: JCT3V-E0097r1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **On SHVC High Level Syntax** | | |
| *Status:* | Input Document to JCT-VC and JCT-3V | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Yong He, Xiaoyu Xiu, Yan Ye, Yuwen He  9710 Scranton R-D, #250  San Diego, CA 92121  USA | Tel: Email: | 1-858-210-4807 Yong.He@InterDigital.com |
| *Source:* | InterDigital Communications, Inc. | | |

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

# Abstract

This contribution proposes several modifications to the current high level syntax in VPS and slice header to improve signaling flexibility and save bit overhead. These changes include (1) replace VPS syntax element max\_one\_active\_ref\_layer\_flag with max\_num\_active\_ref\_layers\_minus1 to improve signaling flexibility; (2) set condition on signaling inter\_layer\_pred\_layer\_idc to save slice header bits; (3) relocate sample and motion prediction syntax elements from slice header to PPS; (4) improved slice header signaling when inter\_layer\_sample\_pred\_only\_flag is set to 1.

# On max\_one\_active\_ref\_layer\_flag

In current SHVC working draft 2, ***max\_one\_active\_ref\_layer\_flag*** is signaled in VPS extension to specify if pictures from only one layer or more than 1 layer may be used for inter-layer prediction in the scalable system. The motivation is to use this flag to impose the restriction to allow inter-layer reference pictures from only one layer in the scalable system, which may be a desired restriction when scalable profiles/levels are defined. Depending on the value of *max\_one\_active\_ref\_layer\_flag*, the syntax element *num\_inter\_layer\_ref\_pics\_minus1* in the slice header may or may not be signaled in the slice header. When *max\_one\_active\_ref\_layer\_flag* is equal to 1, *num\_inter\_layer\_ref\_pic\_minus1* is inferred to be 0, and only one reference layer’s layer ID would be signaled in the slice header; otherwise, *num\_inter\_layer\_ref\_pic\_minus1* is signaled first, followed by the layer IDs of (num\_inter\_layer\_ref\_pics\_minus1 + 1) layers .

Here we propose to change *max\_one\_active\_ref\_layer\_flag* to ***max\_num\_active\_ref\_layers\_minus1*** and change the descriptor type to ue(v) as shown in Table 1. The new syntax element would indicate the maximum number of reference layers used in the decoding process to serve the purpose of capability exchange. The proposed change can also serve the purpose of defining appropriate profile/level constraints for the future, but is more flexible than the current 1-bit flag.

1. Proposed Video parameter set extension syntax element

|  |  |
| --- | --- |
| vps\_extension( ) { | Descriptor |
| while( !byte\_aligned( ) ) |  |
| **vps\_extension\_byte\_alignment\_reserved\_one\_bit** | u(1) |
| ... |  |
| **max\_one\_active\_ref\_layer\_flag** | u(1) |
| **max\_num\_active\_ref\_layers\_minus1** | ue(v) |
| ... |  |
| } |  |

**max\_num\_active\_ref\_layers\_minus1** plus 1 specifies the maximum number of pictures used for inter-layer prediction for each picture in the CVS. The value of max\_num\_active\_ref\_layers\_minus1 shall be in the range of 0 to NumDirectRefLayers[ nuh\_layer\_id ], exclusively. max\_num\_active\_ref\_layers\_minus1 equal to 0 specifies that only one picture may be used for inter-layer prediction for each picture in the CVS. When max\_num\_active\_ref\_layers\_minus1 is not present, it is inferred to be equal to 0.

When ***max\_num\_active\_ref\_layers\_minus1*** is equal to 0, the syntax element *num\_inter\_layer\_ref\_pics\_minus1* may not be signaled in the slice header.

In case at most one reference layer is allowed for inter-layer prediction, the bit cost of such syntax element in VPS extension is the same as the current *max\_one\_active\_ref\_layer\_flag* (1 bit).

The corresponding slice header syntax would be changed as shown in Table 2.

1. **slice segment header syntax**

|  |  |
| --- | --- |
| slice\_segment\_header( ) { | Descriptor |
| **...** |  |
| if( nuh\_layer\_id > 0 && NumDirectRefLayers[ nuh\_layer\_id ] > 0 ) { |  |
| **inter\_layer\_pred\_enabled\_flag** | u(1) |
| if( inter\_layer\_pred\_enabled\_flag && NumDirectRefLayers[ nuh\_layer\_id ] > 1) { |  |
| if(!max\_one\_active\_ref\_layer\_flag max\_num\_active\_ref\_layers\_minus1) |  |
| **num\_inter\_layer\_ref\_pics\_minus1** | u(v) |
| for( i = 0; i < NumActiveRefLayerPics; i++ ) |  |
| **inter\_layer\_pred\_layer\_idc[**i ] | u(v) |
| } |  |
| } |  |
| **...** |  |
| } |  |

The derivation of NumActiveRefLayerPics as specified in SHVC WD2 G.7.4.7.1 can be modified as follows:

The variable NumActiveRefLayerPics is derived as follows:

if( nuh\_layer\_id = = 0 | | NumDirectRefLayers[ nuh\_layer\_id ] = = 0 | | !inter\_layer\_pred\_enabled\_flag )  
 NumActiveRefLayerPics = 0  
else if(max\_one\_active\_ref\_layer\_flag max\_num\_active\_ref\_layers\_minus1 == 0 | | NumDirectRefLayers[ nuh\_layer\_id ] = = 1 )  
 NumActiveRefLayerPics = 1  
else  
 NumActiveRefLayerPics = num\_inter\_layer\_ref\_pics\_minus1 + 1

All slices of a coded picture shall have the same value of NumActiveRefLayerPics.

The benefits of using max\_num\_active\_ref\_layers\_minus1 are:

1. Indicating the maximum number of reference layers used in the decoding process can serve the purpose of capability exchange and profile/level definition;
2. Improving flexibility for profile/level constraints in the future;
3. Maintaining the same bit cost as the current *max\_one\_active\_ref\_layer\_flag* when only one active ref layer is used.

# Signaling condition on inter\_layer\_pred\_layer\_idc

In current SHVC working draft 2, the syntax element inter\_layer\_pred\_layer\_idc is always signaled in the slice header. However, it is redundant when the number of active reference layers is the same as number of direct reference layers (since every direct reference layer is considered active). The redundant signal bit can be removed by adding the signal condition as shown in Table 3.

1. Proposed General slice segment header syntax

|  |  |
| --- | --- |
| slice\_segment\_header( ) { | Descriptor |
| **....** |  |
| if( nuh\_layer\_id > 0 && NumDirectRefLayers[ nuh\_layer\_id ] > 0 ) { |  |
| **inter\_layer\_pred\_enabled\_flag** | u(1) |
| if( inter\_layer\_pred\_enabled\_flag && NumDirectRefLayers[ nuh\_layer\_id ] > 1) { |  |
| if( !max\_one\_active\_ref\_layer\_flag ) |  |
| **num\_inter\_layer\_ref\_pics\_minus1** | u(v) |
| if (NumActiveRefLayerPics != NumDirectRefLayers[ nuh\_layer\_id ] ) |  |
| for( i = 0; i < NumActiveRefLayerPics; i++ ) |  |
| **inter\_layer\_pred\_layer\_idc[**i ] | u(v) |
| } |  |
| } |  |
| .... |  |
| } |  |

Below is the pseudo code to derive inter\_layer\_pred\_layer\_idc when the number of active reference layers (NumActiveRefLayerPics) is the same as the number of direct reference layers (NumDirectReflayers). RefLayerId[i] is specified in SHVC WD 2 F.7.4.3.1.1.

|  |
| --- |
| if (NumActiveRefLayerPics = NumDirectRefLayers[nuh\_layer\_id]) {  for( i = 0; i < NumActiveRefLayerPics; i++ )  inter\_layer\_pred\_layer\_idc[i] = RefLayerId [nuh\_layer\_id][i]  } |

# Sample and motion prediction syntax elements

For error resilience considerations, slice header is sent for every slice in the picture (unless it is a dependent slice). Because a picture may consist of multiple slices, the bit cost of slice header is more of a concern than the bit cost of other parameter sets such as SPS (Sequence parameter set) or PPS (Picture parameter set), which are usually sent less frequently.

In SHVC WD2 G.7.4.7.1, it is specified that the variables such as *NumActiveRefLayerPics, inter\_layer\_pred\_layer\_idc* and *collocated\_ref\_layer\_idx* shall be the same for all slices of a coded picture. Instead of the slice header, it may be beneficial to signal these syntax elements, such as *inter\_layer\_pred\_enable\_flag, num\_inter\_layer\_ref\_pics\_minus1, inter\_layer\_pred\_layer\_idc, inter\_layer\_sample\_pred\_only\_flag, alt\_collocated\_indicate\_flag* and *collocated\_ref\_layer\_idx*, in PPS so that the same syntax won’t be duplicated for each slice within a picture.

Table 4 and Table 5 are the proposed PPS extension syntax table and corresponding slice header syntax.

1. Proposed picture parameter set extension syntax

|  |  |
| --- | --- |
| pps\_extension( ) { | **Descriptor** |
| if( nuh\_layer\_id > 0 ) { |  |
| NumActiveRefLayerPics = 0 |  |
| **inter\_layer\_pred\_enabled\_flag** | u(1) |
| if( inter\_layer\_pred\_enabled\_flag ) { |  |
| **num\_inter\_layer\_ref\_pics\_minus1** | u(v) |
| NumActiveRefLayerPics = num\_inter\_layer\_ref\_pics\_minus1 + 1 |  |
| for( i = 0; i < NumActiveRefLayerPics; i++ ) |  |
| **inter\_layer\_pred\_layer\_idc[**i ] | u(v) |
| } |  |
| if(NumActiveRefLayerPics > 0 ) |  |
| **inter\_layer\_sample\_pred\_only\_flag** | u(1) |
| if( nuh\_layer\_id > 0 ) |  |
| **alt\_collocated\_indication\_flag** | u(1) |
| if( alt\_collocated\_indication\_flag ) |  |
| **collocated\_ref\_layer\_idx** | ue(v) |
| } |  |
| } |  |

1. Proposed slice segment header syntax

|  |  |
| --- | --- |
| slice\_segment\_header( ) { | Descriptor |
| **...** |  |
| if( nuh\_layer\_id > 0 && NumDirectRefLayers[ nuh\_layer\_id ] > 0 ) { |  |
| **inter\_layer\_pred\_enabled\_flag** | u(1) |
| if( inter\_layer\_pred\_enabled\_flag && NumDirectRefLayers[ nuh\_layer\_id ] > 1) { |  |
| if( !max\_one\_active\_ref\_layer\_flag ) |  |
| **num\_inter\_layer\_ref\_pics\_minus1** | u(v) |
| for( i = 0; i < NumActiveRefLayerPics; i++ ) |  |
| **inter\_layer\_pred\_layer\_idc[**i ] | u(v) |
| } |  |
| } |  |
| if( NumSamplePredRefLayers[ nuh\_layer\_id ] > 0 && NumActiveRefLayerPics > 0 ) |  |
| **inter\_layer\_sample\_pred\_only\_flag** | u(1) |
| **...** |  |
| if( slice\_temporal\_mvp\_enabled\_flag ) { |  |
| if( nuh\_layer\_id > 0 && NumActiveMotionPredRefLayers > 0 ) |  |
| **alt\_collocated\_indication\_flag** | u(1) |
| if( alt\_collocated\_indication\_flag ) |  |
| if( NumActiveMotionPredRefLayers > 1 ) |  |
| **collocated\_ref\_layer\_idx** | ue(v) |
| else { |  |
| if( slice\_type = = B ) |  |
| **collocated\_from\_l0\_flag** | u(1) |
| if( ( collocated\_from\_l0\_flag && num\_ref\_idx\_l0\_active\_minus1 > 0 ) | |  ( !collocated\_from\_l0\_flag && num\_ref\_idx\_l1\_active\_minus1 > 0 ) ) |  |
| **collocated\_ref\_idx** | ue(v) |
| } |  |
| } |  |
| **...** |  |
| } |  |

# On inter\_layer\_sample\_pred\_only\_flag

### Signaling condition for IDR NAL units

As specified in SHVC WD2, ***inter\_layer\_sample\_pred\_only\_flag*** equal to 1 indicates that inter prediction using temporal reference pictures in the EL is not allowed when decoding the current picture. In other words, the reference picture lists L0 and L1 would not include any temporal reference pictures.

Currently, the syntax element *inter\_layer\_sample\_pred\_only\_flag* is signaled in every slice regardless of its NAL unit type. However, IDR pictures in the enhancement layer are naturally pictures without inter prediction using temporal reference pictures. Therefore, *inter\_layer\_sample\_pred\_only\_flag* is redundant for IDR NAL unit in the EL. To remove such redundancy, the condition highlighted in yellow in Table 7 may be applied.

1. Proposed General slice segment header syntax

|  |  |
| --- | --- |
| slice\_segment\_header( ) { | Descriptor |
| **....** |  |
| if( NumSamplePredRefLayers[ nuh\_layer\_id ] > 0 && NumActiveRefLayerPics > 0 && (nal\_unit\_type != IDR\_W\_RADL && nal\_unit\_type != IDR\_N\_LP)) |  |
| **inter\_layer\_sample\_pred\_only\_flag** | u(1) |
| .... |  |
| } |  |

### Motion prediction syntax elements

When ***inter\_layer\_sample\_pred\_only\_flag*** is equal to 1, the only reference pictures available are the inter-layer reference pictures. Since SHVC requires all motion vectors from inter layer reference pictures to be equal to zero, the temporal motion vector prediction (TMVP) process can be bypassed; and all syntax elements in the slice header related to TMVP can be skipped as well.

#### *Option 1*

Here we propose to signal *slice\_temporal\_mvp\_enabled\_flag* based not only on *sps\_temporal\_mvp\_enabled\_flag* (as specified in SHVC WD G.7.3.6.1), but also on the *inter\_layer\_sample\_pred\_only\_flag* for the enhancement layer.

Table 9 shows such signaling example. The variable *InterRefEnabledInRPLFlag* derivation is specified in SHVC WD G.7.4.7.1.

In order to condition *slice\_temporal\_mvp\_enabled\_flag* upon *inter\_layer\_sample\_pred\_only\_flag*, the signaling of *inter\_layer\_sample\_pred\_only\_flag* and sample prediction syntax structure shall be prior to *slice\_temporal\_mvp\_enabled\_flag*. When *slice\_temporal\_mvp\_enabled\_flag* is not signaled (for example because *inter\_layer\_sample\_pred\_only\_flag* is set equal to 1), *slice\_temporal\_mvp\_enabled\_flag* is inferred to be 0.

When *slice\_temporal\_mvp\_enabled\_flag* is 0, the syntax elements such as *alt\_collocated\_indication\_flag*, *collocated\_ref\_layer\_idx*, *collocated\_from\_l0\_flag* and *collocated\_ref\_idx* will be skipped and thus signaling overhead will be reduced.

1. Proposed slice segment header syntax

|  |  |
| --- | --- |
| slice\_segment\_header( ) { | Descriptor |
| **first\_slice\_segment\_in\_pic\_flag** | u(1) |
| if( nal\_unit\_type >= BLA\_W\_LP && nal\_unit\_type <= RSV\_IRAP\_VCL23 ) |  |
| **no\_output\_of\_prior\_pics\_flag** | u(1) |
| **slice\_pic\_parameter\_set\_id** | ue(v) |
| if( !first\_slice\_segment\_in\_pic\_flag ) { |  |
| if( dependent\_slice\_segments\_enabled\_flag ) |  |
| **dependent\_slice\_segment\_flag** | u(1) |
| **slice\_segment\_address** | u(v) |
| } |  |
| if( !dependent\_slice\_segment\_flag ) { |  |
| if( num\_extra\_slice\_header\_bits > 0 ) |  |
| **discardable\_flag** | u(1) |
| for( i = 1; i < num\_extra\_slice\_header\_bits ; i++ ) |  |
| **slice\_reserved\_flag**[ i ] | u(1) |
| **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( nal\_unit\_type != IDR\_W\_RADL && nal\_unit\_type != IDR\_N\_LP ) { |  |
| **slice\_pic\_order\_cnt\_lsb** | u(v) |
| **short\_term\_ref\_pic\_set\_sps\_flag** | u(1) |
| if( !short\_term\_ref\_pic\_set\_sps\_flag ) |  |
| short\_term\_ref\_pic\_set( num\_short\_term\_ref\_pic\_sets ) |  |
| else if( num\_short\_term\_ref\_pic\_sets > 1 ) |  |
| **short\_term\_ref\_pic\_set\_idx** | u(v) |
| if( long\_term\_ref\_pics\_present\_flag ) { |  |
| if( num\_long\_term\_ref\_pics\_sps > 0 ) |  |
| **num\_long\_term\_sps** | ue(v) |
| **num\_long\_term\_pics** | ue(v) |
| for( i = 0; i < num\_long\_term\_sps + num\_long\_term\_pics; i++ ) { |  |
| if( i < num\_long\_term\_sps ) { |  |
| if( num\_long\_term\_ref\_pics\_sps > 1 ) |  |
| **lt\_idx\_sps**[ i ] | u(v) |
| } else { |  |
| **poc\_lsb\_lt**[ i ] | u(v) |
| **used\_by\_curr\_pic\_lt\_flag**[ i ] | u(1) |
| } |  |
| **delta\_poc\_msb\_present\_flag**[ i ] | u(1) |
| if( delta\_poc\_msb\_present\_flag[ i ] ) |  |
| **delta\_poc\_msb\_cycle\_lt**[ i ] | ue(v) |
| } |  |
| } |  |
| } |  |
| if( nuh\_layer\_id > 0 && NumDirectRefLayers[ nuh\_layer\_id ] > 0 ) { |  |
| **inter\_layer\_pred\_enabled\_flag** | u(1) |
| if( inter\_layer\_pred\_enabled\_flag && NumDirectRefLayers[ nuh\_layer\_id ] > 1) { |  |
| if( !max\_one\_active\_ref\_layer\_flag ) |  |
| **num\_inter\_layer\_ref\_pics\_minus1** | u(v) |
| for( i = 0; i < NumActiveRefLayerPics; i++ ) |  |
| **inter\_layer\_pred\_layer\_idc[**i ] | u(v) |
| } |  |
| } |  |
| if( NumSamplePredRefLayers[ nuh\_layer\_id ] > 0 && NumActiveRefLayerPics > 0 ) |  |
| **inter\_layer\_sample\_pred\_only\_flag** | u(1) |
| if( nal\_unit\_type != IDR\_W\_RADL && nal\_unit\_type != IDR\_N\_LP ) { |  |
| if( (sps\_temporal\_mvp\_enabled\_flag ) || (nuh\_layer\_id > 0 && InterRefEnabledInRPLFlag ) ) |  |
| **slice\_temporal\_mvp\_enabled\_flag** | u(1) |
| } |  |
| if( nuh\_layer\_id > 0 && NumDirectRefLayers[ nuh\_layer\_id ] > 0 ) { |  |
| **inter\_layer\_pred\_enabled\_flag** | u(1) |
| if( inter\_layer\_pred\_enabled\_flag && NumDirectRefLayers[ nuh\_layer\_id ] > 1) { |  |
| if( !max\_one\_active\_ref\_layer\_flag ) |  |
| **num\_inter\_layer\_ref\_pics\_minus1** | u(v) |
| for( i = 0; i < NumActiveRefLayerPics; i++ ) |  |
| **inter\_layer\_pred\_layer\_idc[**i ] | u(v) |
| } |  |
| } |  |
| if( NumSamplePredRefLayers[ nuh\_layer\_id ] > 0 && NumActiveRefLayerPics > 0 ) |  |
| **inter\_layer\_sample\_pred\_only\_flag** | u(1) |
| if( sample\_adaptive\_offset\_enabled\_flag ) { |  |
| **slice\_sao\_luma\_flag** | u(1) |
| **slice\_sao\_chroma\_flag** | u(1) |
| } |  |
| 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) |
| } |  |
| if( lists\_modification\_present\_flag && NumPocTotalCurr > 1 ) |  |
| ref\_pic\_lists\_modification( ) |  |
| if( slice\_type = = B ) |  |
| **mvd\_l1\_zero\_flag** | u(1) |
| if( cabac\_init\_present\_flag ) |  |
| **cabac\_init\_flag** | u(1) |
| if( slice\_temporal\_mvp\_enabled\_flag ) { |  |
| if( nuh\_layer\_id > 0 && NumActiveMotionPredRefLayers > 0 ) |  |
| **alt\_collocated\_indication\_flag** | u(1) |
| if( alt\_collocated\_indication\_flag ) |  |
| if( NumActiveMotionPredRefLayers > 1 ) |  |
| **collocated\_ref\_layer\_idx** | ue(v) |
| else { |  |
| if( slice\_type = = B ) |  |
| **collocated\_from\_l0\_flag** | u(1) |
| if( ( collocated\_from\_l0\_flag && num\_ref\_idx\_l0\_active\_minus1 > 0 ) | |  ( !collocated\_from\_l0\_flag && num\_ref\_idx\_l1\_active\_minus1 > 0 ) ) |  |
| **collocated\_ref\_idx** | ue(v) |
| } |  |
| } |  |
| if( ( weighted\_pred\_flag && slice\_type = = P ) | |  ( weighted\_bipred\_flag && slice\_type = = B ) ) |  |
| pred\_weight\_table( ) |  |
| **five\_minus\_max\_num\_merge\_cand** | ue(v) |
| } |  |
| **slice\_qp\_delta** | se(v) |
| if( pps\_slice\_chroma\_qp\_offsets\_present\_flag ) { |  |
| **slice\_cb\_qp\_offset** | se(v) |
| **slice\_cr\_qp\_offset** | se(v) |
| } |  |
| if( deblocking\_filter\_override\_enabled\_flag ) |  |
| **deblocking\_filter\_override\_flag** | u(1) |
| if( deblocking\_filter\_override\_flag ) { |  |
| **slice\_deblocking\_filter\_disabled\_flag** | u(1) |
| if( !slice\_deblocking\_filter\_disabled\_flag ) { |  |
| **slice\_beta\_offset\_div2** | se(v) |
| **slice\_tc\_offset\_div2** | se(v) |
| } |  |
| } |  |
| if( pps\_loop\_filter\_across\_slices\_enabled\_flag &&  ( slice\_sao\_luma\_flag | | slice\_sao\_chroma\_flag | |  !slice\_deblocking\_filter\_disabled\_flag ) ) |  |
| **slice\_loop\_filter\_across\_slices\_enabled\_flag** | u(1) |
| } |  |
| if( tiles\_enabled\_flag | | entropy\_coding\_sync\_enabled\_flag ) { |  |
| **num\_entry\_point\_offsets** | ue(v) |
| if( num\_entry\_point\_offsets > 0 ) { |  |
| **offset\_len\_minus1** | ue(v) |
| for( i = 0; i < num\_entry\_point\_offsets; i++ ) |  |
| **entry\_point\_offset\_minus1**[ i ] | u(v) |
| } |  |
| } |  |
| if( slice\_segment\_header\_extension\_present\_flag ) { |  |
| **slice\_segment\_header\_extension\_length** | ue(v) |
| for( i = 0; i < slice\_segment\_header\_extension\_length; i++) |  |
| **slice\_segment\_header\_extension\_data\_byte**[ i ] | u(8) |
| } |  |
| byte\_alignment( ) |  |
| } |  |

#### Option 2

To keep the current signaling order without moving the sample and motion prediction syntax elements prior to slice\_temporal\_mvp\_enabled\_flag, an alternative solution is to apply extra condition (InterRefEnabledInRPLFlag) for signaling the TMVP parameters as shown in Table 8.

The derivation process for temporal luma motion vector prediction (SHVC WD2 G.8.5.3.2.7) is modified as follows.

The variables mvLXCol and availableFlagLXCol are derived as follows:

* If slice\_temporal\_mvp\_enabled\_flag is equal to 0 or InterRefEnabledInRPLFlag is equal to 0, both components of mvLXCol are set equal to 0 and availableFlagLXCol is set equal to 0.

1. Proposed General slice segment header syntax

|  |  |
| --- | --- |
| slice\_segment\_header( ) { | Descriptor |
| **...** |  |
| if( slice\_temporal\_mvp\_enabled\_flag && InterRefEnabledInRPLFlag) { |  |
| if( nuh\_layer\_id > 0 && NumActiveMotionPredRefLayers > 0 ) |  |
| **alt\_collocated\_indication\_flag** | u(1) |
| if( alt\_collocated\_indication\_flag ) |  |
| if( NumActiveMotionPredRefLayers > 1 ) |  |
| **collocated\_ref\_layer\_idx** | ue(v) |
| else { |  |
| if( slice\_type = = B ) |  |
| **collocated\_from\_l0\_flag** | u(1) |
| if( ( collocated\_from\_l0\_flag && num\_ref\_idx\_l0\_active\_minus1 > 0 ) | |  ( !collocated\_from\_l0\_flag && num\_ref\_idx\_l1\_active\_minus1 > 0 ) ) |  |
| **collocated\_ref\_idx** | ue(v) |
| } |  |
| } |  |
| ... |  |
| } |  |

# Patent rights declaration(s)

**InterDigital Communications, Inc. 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).**