|  |  |
| --- | --- |
| **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-N0157r1 |

|  |  |
| --- | --- |
| **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-E0076r1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **MV-HEVC/SHVC HLS: On signalling of sps\_max\_sub\_layers\_minus1** | | |
| *Status:* | Input Document to JCT-VC and JCT-3V | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Jung Won Kang  Hahyun Lee Jinho Lee Jin Soo Choi | Tel: Email: | +82-42-860-5137 [jungwon@etri.re.kr](mailto:jungwon@etri.re.kr) [hanilee@etri.re.kr](mailto:hanilee@etri.re.kr) [jinosoul@etri.re.kr](mailto:jinosoul@etri.re.kr) [jschoi@etri.re.kr](mailto:jschoi@etri.re.kr) |
| *Source:* | ETRI (Electronics and Telecommunications Research Institute) | | |

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

# Abstract

In this document, it is asserted that there are parsing dependency between VPS and SPS and unnecessary signalling of sps\_max\_dec\_pic\_buffering\_minus1, sps\_max\_num\_reorder\_pics, and sps\_max\_latency\_increase\_plus1 parameters since sps\_max\_sub\_layers\_minus1 is signalled only when nuh\_layer\_id is equal to 0 and is inferred to be equal to vps\_max\_sub\_layers\_minus1 in VPS extension. The document proposes signalling of maximum number of temporal sub-layer for each layer, and sps\_max\_dec\_pic\_buffering\_minus1, sps\_max\_num\_reorder\_pics, and sps\_max\_latency\_increase\_plus1 parameters for sub-layers of each layer in VPS extension.

In revision 2, max. DPB size derivation process for each layer in a output layer set is described.

# Problem Statement

In the current SHVC as well as MV-HEVC draft text, sps\_max\_sub\_layers\_minus1 is signalled only when nuh\_layer\_id is equal to 0.When sps\_max\_sub\_layers\_minus1 is not present in SPS which is the case of nuh\_layer\_id > 0, sps\_max\_sub\_layers\_minus1 is inferred to be equal to vps\_max\_sub\_layers\_minus1 signalled in VPS. This introduces parsing dependency between VPS and SPS and may cause unnecessary signalling of sps\_max\_dec\_pic\_buffering\_minus1[i], sps\_max\_num\_reorder\_pics[i], and sps\_max\_latency\_increase\_plus1[i].

First, on parsing dependency, sps\_max\_dec\_pic\_buffering\_minus1[i], sps\_max\_num\_reorder\_pics[i], and sps\_max\_latency\_increase\_plus1[i] are signalled for i in the range of 0 to sps\_max\_sub\_layers\_minus1 which is inferred to be equal to vps\_max\_sub\_layers\_minus1 when nuh\_layer\_id of the SPS is greater than 0. This introduces parsing dependency between VPS and SPS.

Second, since vps\_max\_sub\_layers\_minus1 + 1 is the maximum number of temporal sub-layers in the bitstream and frame rate for each layers could be different in SHVC as decided at the 13th JCT-VC meeting, actual maximum number of temporal sub-layers for each layer could be less than vps\_max\_sub\_layers\_minus1+1. Therefore, sps\_max\_dec\_pic\_buffering\_minus1[i], sps\_max\_num\_reorder\_pics[i], and sps\_max\_latency\_increase\_plus1[i] should be signalled for i in the range of 0 to the maximum number of temporal sub-layer of the layer having nuh\_layer\_id of the SPS not vps\_max\_sub\_layers\_minus1.

* **Syntax table in SHVC draft text 2 and MV-HEVC draft text 4:**

|  |  |
| --- | --- |
| seq\_parameter\_set\_rbsp( ) { | Descriptor |
| **sps\_video\_parameter\_set\_id** | u(4) |
| if( nuh\_layer\_id = = 0 ) { |  |
| **sps\_max\_sub\_layers\_minus1** | u(3) |
| **sps\_temporal\_id\_nesting\_flag** | u(1) |
| profile\_tier\_level( 1, sps\_max\_sub\_layers\_minus1 ) |  |
| } |  |
| **…** |  |
| **sps\_sub\_layer\_ordering\_info\_present\_flag** | u(1) |
| for( i = ( sps\_sub\_layer\_ordering\_info\_present\_flag ? 0 : sps\_max\_sub\_layers\_minus1 );  i <= sps\_max\_sub\_layers\_minus1; i++ ) { |  |
| **sps\_max\_dec\_pic\_buffering\_minus1**[ i ] | ue(v) |
| **sps\_max\_num\_reorder\_pics**[ i ] | ue(v) |
| **sps\_max\_latency\_increase\_plus1**[ i ] | ue(v) |
| } |  |
|  |  |
| } |  |

|  |  |
| --- | --- |
| video\_parameter\_set\_rbsp( ) { | Descriptor |
| **vps\_video\_parameter\_set\_id** | u(4) |
| **vps\_reserved\_three\_2bits** | u(2) |
| **vps\_max\_layers\_minus1** | u(6) |
| **vps\_max\_sub\_layers\_minus1** | u(3) |
| **vps\_temporal\_id\_nesting\_flag** | u(1) |
| **vps\_extension\_offset** **//**vps\_reserved\_0xffff\_16bits | u(16) |
| profile\_tier\_level( 1, vps\_max\_sub\_layers\_minus1 ) |  |
| **vps\_sub\_layer\_ordering\_info\_present\_flag** | u(1) |
| for( i = ( vps\_sub\_layer\_ordering\_info\_present\_flag ? 0 : vps\_max\_sub\_layers\_minus1 );  i <= vps\_max\_sub\_layers\_minus1; i++ ) { |  |
| **vps\_max\_dec\_pic\_buffering\_minus1**[ i ] | ue(v) |
| **vps\_max\_num\_reorder\_pics**[ i ] | ue(v) |
| **vps\_max\_latency\_increase\_plus1**[ i ] | ue(v) |
| } |  |
| **…** |  |
| } |  |

# Proposal

To avoid parsing dependency and unnecessary signalling of sps\_max\_dec\_pic\_buffering\_minus1, sps\_max\_num\_reorder\_pics, and sps\_max\_latency\_increase\_plus1 parameters, it is proposed to signal maximum number of temporal sub-layer for each layer, and sps\_max\_dec\_pic\_buffering\_minus1, sps\_max\_num\_reorder\_pics, and sps\_max\_latency\_increase\_plus1 parameters for sub-layers of each layer in VPS extension. Signalling these syntax elements which are DPB related parameters in VPS extension can help in session negotiation.

* **Proposed syntax and semantics**

|  |  |
| --- | --- |
| vps\_extension( ) { | Descriptor |
| … |  |
| for( i = 1; i <= vps\_max\_layers\_minus1; i ++ ) { |  |
| **max\_sub\_layers\_vps\_prediction\_flag**[i] | u(1) |
| if(!max\_sub\_layers\_vps\_prediction\_flag[i]) |  |
| **max\_sub\_layers\_vps\_minus1**[i] | u(3) |
| } |  |
| for( i = 1; i <= vps\_max\_layers\_minus1; i ++ ) { |  |
| if(max\_sub\_layers\_vps\_prediction\_flag[i]) |  |
| **sub\_layer\_ordering\_info\_vps\_prediction\_flag**[i] | u(1) |
| if(!sub\_layer\_ordering\_info\_vps\_prediction\_flag[i]) |  |
| **sub\_layer\_ordering\_info\_vps\_present\_flag**[i] | u(1) |
| for( j = ( sub\_layer\_ordering\_info\_vps\_present\_flag[i] ? 0 : max\_sub\_layers\_vps\_minus1[i] ); j <= max\_sub\_layers\_vps\_minus1[i]; i++ ) { |  |
| **max\_dec\_pic\_buffering\_minus1**[ i ][ j ] | ue(v) |
| **max\_num\_reorder\_pics**[ i ][ j ] | ue(v) |
| **max\_latency\_increase\_plus1**[ i ][ j] | ue(v) |
| } |  |
| } |  |
| } |  |
| … |  |
| } |  |

**max\_sub\_layers\_vps\_prediction\_flag**[i]equal to 1 specifies that max\_sub\_layers\_vps\_minus1[i] is inferred to be equal to vps\_max\_sub\_layers\_minus1 and sub\_layer\_ordering\_info\_vps\_prediction\_flag[i] is present. max\_sub\_layers\_vps\_prediction\_flag[i] equal to 0 specifies that max\_sub\_layers\_vps\_minus1[i] is explicitly signalled.

**max\_sub\_layers\_vps\_minus1**[i]plus1 specifies the maximum number of temporal sub-layers that may be present in a ith-layer. max\_sub\_layers\_vps\_minus1[0] is inferred to be equal to sps\_max\_sub\_layers\_minus1. The value of sps\_max\_sub\_layers\_minus1 shall be in the range of 0 to 6, inclusive.

**sub\_layer\_ordering\_info\_vps\_prediction\_flag**[i]equal to 1 specifies that max\_dec\_pic\_buffering\_minus[i][j], max\_num\_reorder\_pics[i][j], and max\_latency\_increase\_plus1[i][j] are inferred to be equal to vps\_max\_dec\_pic\_buffering\_minus1[j], vps\_max\_num\_reorder\_pics[j], and vps\_max\_latency\_increase\_plus1[j] respectively. sub\_layer\_ordering\_info\_vps\_prediction\_flag[i] equal to 0 specifies that max\_dec\_pic\_buffering\_minus[i][j], max\_num\_reorder\_pics[i][j], and max\_latency\_increase\_plus1[i][j] are explicitly signalled. When not present, the value of sub\_layer\_ordering\_info\_vps\_prediction\_flag[i] is inferred to be equal to 0.

**sub\_layer\_ordering\_info\_vps\_present\_flag**[i]equal to 1 specifies that max\_dec\_pic\_buffering\_minus[i][j], max\_num\_reorder\_pics[i][j], and max\_latency\_increase\_plus1[i][j] are present for max\_sub\_layers\_vps\_minus1[i]+1 sub layers. sub\_layer\_ordering\_info\_vps\_present\_flag[i] equal to 0 specifies that max\_dec\_pic\_buffering\_minus[i][max\_sub\_layer\_vps\_minus1[i]], max\_num\_reorder\_pics[i][max\_sub\_layer\_vps\_minus1[i]], and max\_latency\_increase\_plus1[i][max\_sub\_layer\_vps\_minus1[i]] apply to all sub-layers.

**max\_dec\_pic\_buffering\_minus1**[i][j]plus1specifies the maximum required size of the decoded picture buffer for the ith-layer in units of picture storage buffers when HighestTid is equal to j.

**max\_num\_reorder\_pics**[i][j]indicates the maximum allowed number of pictures that can precede any picture in the ith-layer in decoding order and follow that picture in output order when HighestTid is equal to i. The value of max\_num\_reorder\_pics[i][j] shall be in the range of 0 to max\_dec\_pic\_buffering\_minus1[i][j], inclusive.

**max\_latency\_increase\_plus1**[i][j]not equal to 0 is used to compute the value of MaxLatencyPictures[i][j], which specifies the maximum number of pictures that can precede any picture in the ith-layer in output order and follow that picture in decoding order when HighestTid is equal to j.

When max\_latency\_increase\_plus1[i][j] is not equal to 0, the value of MaxLatencyPictures[i][j] is specified as follows:

MaxLatencyPictures[i][j] = sps\_max\_num\_reorder\_pics[i][j] + sps\_max\_latency\_increase\_plus1[i][j] – 1

* **Max. DPB size derivation for each layer in a output layer set**

Based on the proposed syntax elements, max DPB size for each layer can be derived as explained below.

Max. DPB size of whose layer is an output layer should be different from max. DPB size whose layer is not an output layer.

* Case 1: when a layer is an output layer

In this case, all pictures in the output layer is decoded. Therefore, max\_dec\_reorder\_pics[i][j] signalled in VPS extension is used as max. DPB size.

* Case 2: when a layer is not an output layer

In this case, a picture that is not used for inter-layer prediction in the output layer is not needed to be decoded. If max\_tid\_il\_ref\_pics\_plus1[i] = K for i-th layer, and i-th layer is not a output layer, a picture having temporalId is greater than K in this layer doesn’t need to be decoded or can be marked as unused for reference layer after decoding. Therefore, max. DPB size for this layer can be derived as follows.

For 0<=j<=K-1, max. DPB[i][j] = max\_dec\_pic\_buffering[i][j]

For K<=j<=HighestTid, max. DPB[i][j] = max\_dec\_pic\_buffering[i][K-1]

# Patent rights declaration(s)

**ETRI 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).**