H.7.3.2.1.2 Video parameter set extension 2 syntax

|  |
| --- |
| vps\_extension2( ) { |
| while( !byte\_aligned( ) ) |
| **vps\_extension\_byte\_alignment\_reserved\_one\_bit** |
| for( i = 0; i <= vps\_max\_layers\_minus1; i++ ) { |
| layerId = layer\_id\_in\_nuh[ i ] |
| if ( layerId ! = 0 ) { |
| **cp\_present\_flag**[ layerId ] |
| if ( !VpsDepthFlag[ layerId ] ) { |
| **iv\_mv\_pred\_flag**[ layerId ] |
| **iv\_res\_pred\_flag**[ layerId ] |
| **depth\_refinement\_flag**[ layerId ] |
| **view\_synthesis\_pred\_flag**[ layerId ] |
| } else { |
| **mpi\_flag**[ layerId ] |
| **vps\_depth\_modes\_flag**[ layerId ] |
| **lim\_qt\_pred\_flag**[ layerId ] |
| if( vps\_depth\_modes\_flag[ layerId ] ) |
| **dlt\_flag**[ layerId ] |
| if( dlt\_flag[ layerId ] ) { |
| **num\_depth\_values\_in\_dlt**[ layerId ] |
| for ( j = 0; j < num\_depth\_values\_in\_dlt[ layerId ] ; j++) { |
| **dlt\_depth\_value**[ layerId ][ j ] |
| } |
| } |
| } |
| } |
| } |

**mpi\_flag**[ layerId ]indicates whether motion parameter inheritance is used in the decoding process of the layer with nuh\_layer\_id equal to layerId. mpi\_flag[ layerId ] equal to 0 specifies that motion parameter inheriatance is not used for the layer with nuh\_layer\_id equal to layerId. mpi\_flag[ layerId ]equal to 1 specifies that motion parameter inheritance may be used for the layer with nuh\_layer\_id equal to layerId. When not present, the value of mpi\_flag[ layerId ] shall be inferred to be equal to 0.

H.7.4.7 Slice segment header semantics

H.7.4.7.1 General slice segment header semantics

...

The variable DepthFlag is set equal to VpsDepthFlag[ nuh\_layer\_id ] and the variable ViewIdx is set equal to ViewOrderIndex[ nuh\_layer\_id ].

**five\_minus\_max\_num\_merge\_cand** specifies the maximum number of merging MVP candidates supported in the slice subtracted from 5. The maximum number of merging MVP candidates, MaxNumMergeCand is computed as

MaxNumMergeCand = 5 − five\_minus\_max\_num\_merge\_cand + iv\_mv\_pred\_flag[ nuh\_layer\_id ]~~+mpi\_flag[ nuh\_layer\_id ]~~ (H‑6)

The value of five\_minus\_max\_num\_merge\_cand shall be limited such that MaxNumMergeCand is in the range of 0 to (5 + iv\_mv\_pred\_flag[ nuh\_layer\_id ] ~~+mpi\_flag[ nuh\_layer\_id ]~~), inclusive.

...

H.8.5.3.2.1 Derivation process for luma motion vectors for merge mode

...

* 1. Depending on iv\_mv\_pred\_flag[ nuh\_layer\_id ], the following applies.
     + If iv\_mv\_pred\_flag[ nuh\_layer\_id ] is equal to 0, the flags availableFlagIvMC, availableIvMCShift and availableFlagIvDC are set equal to 0.
     + Otherwise (iv\_mv\_pred\_flag[ nuh\_layer\_id ] is equal to 1), the derivation process for the inter-view merge candidates as specified in subclause H.8.5.3.2.10 is invoked with the luma location ( xPb, yPb ), the variables nPbW and nPbH, as the inputs and the output is assigned to the availability flags availableFlagIvMC, availableIvMCShift and availableFlagIvDC, the reference indices refIdxLXIvMC, refIdxLXIvMCShift and refIdxLXIvDC, the prediction list utilization flags predFlagLXIvMC, predFlagLXivMCShift and predFlagLXIvDC, and the motion vectors mvLXIvMC, mvLXIvMCShift and mvLXIvDC (with X being 0 or 1, respectively)..
  2. Depending on view\_synthesis\_pred\_flag[ nuh\_layer\_id ], the following applies.
     + If view\_synthesis\_pred\_flag[ nuh\_layer\_id ] is equal to 0, the flag availableFlagVSP is set equal to 0.
     + Otherwise (view\_synthesis\_pred\_flag[ nuh\_layer\_id ] is equal to 1), the derivation process for a view synthesis prediction merge candidate as specified in subclause H.8.5.3.2.13 is invoked with the luma locations ( xCb, yCb ) as input and the outputs are the availability flag availableFlagVSP, the reference indices refIdxL0VSP and refIdxL1VSP, the prediction list utilization flags predFlagL0VSP and predFlagL1VSP, and the motion vectors mvL0VSP and mvL1VSP.
  3. Depending on mpi\_flag[ nuh\_layer\_id ], the following applies.
     + If mpi\_flag[ nuh\_layer\_id ] is equal to 0, the variable availableFlagT is set equal to 0.
     + Otherwise (mpi\_flag[ nuh\_layer\_id ] is equal to 1), the derivation process for the texture merging candidate as specified in subclause H.8.5.3.2.14 is invoked with the luma location ( xPb, yPb ), the variables nPbW and nPbH as the inputs and the outputs are the flag availableFlagT, the prediction utilization flags predFlagL0T and predFlagL1T, the reference indices refIdxL0T and refIdxL1T, and the motion vectors mvL0T and mvL1T.
  4. The merge candidate lists mergeCandList and mergeCandIsVspFlag are constructed as specified by the following ordered steps:
  5. The variable numMergeCand is set equal to 0.
  6. When availableFlagT is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to T, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to 0 and the variable numMergeCand is increased by 1.
  7. When availableFlagIvMC is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to IvMC, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to 0 and the variable numMergeCand is increased by 1.
  8. When availableFlagA1 is equal to 1, the following applies:
     + - When the following condition is true,
         * availableFlagT = = 0 && availableFlagIvMC  = = 0,
         1. or one or more of the following conditions are true, with N being replaced by T and IvMC:
         * availableFlagN = = 1 && predFlagLXN  !=  predFlagLXA1, (with X being replaced by 0 and 1),
         * availableFlagN = = 1 && mvLXN  !=  mvLXA1 (with X being replaced by 0 and 1),
         * availableFlagN = = 1 && refIdxLXN  !=  refIdxLXA1 (with X being replaced by 0 and 1),

the entry mergeCandList[ numMergeCand ] is set equal to A1, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to VspModeFlag[ xPb − 1 ][ yPb + nPbH − 1 ] and the variable numMergeCand is increased by 1.

* 1. When availableFlagB1 is equal to 1, the following applies:
     + - When the following condition is true ,
         * availableFlagT = = 0 && availableFlagIvMC = = 0,

or one or more of the following conditions is true, with N being replaced by T and IvMC:

* + - * + availableFlagN = = 1 && predFlagLXN  !=  predFlagLXB1, (with X being replaced by 0 and 1),
        + availableFlagN = = 1 && mvLXN  !=  mvLXB1 (with X being replaced by 0 and 1),
        + availableFlagN = = 1 && refIdxLXN  !=  refIdxLXB1 (with X being replaced by 0 and 1),

the entry mergeCandList[ numMergeCand ] is set equal to B1, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to VspModeFlag[ xPb + nPbW − 1 ][ yPb − 1 ] and the variable numMergeCand is increased by 1.

* 1. When availableFlagB0 is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to B0, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to VspModeFlag[ xPb + nPbW ][ yPb − 1 ] and the variable numMergeCand is increased by 1.
  2. When availableFlagIvDC is equal to 1, and one or more of the following conditions is true,
     + - availableFlagA1  = =  0,
       - predFlagLXA1  !=  predFlagLXIvDC, (with X being replaced by 0 and 1),
       - mvLXA1  !=  mvLXIvDC(with X being replaced by 0 and 1),
       - refIdxLXA1  !=  refIdxLXIvDC(with X being replaced by 0 and 1),

and one or more of the following conditions is true,

* + - * availableFlagB1  = =  0,
      * predFlagLXB1  !=  predFlagLXIvDC, (with X being replaced by 0 and 1),
      * mvLXB1  !=  mvLXIvDC(with X being replaced by 0 and 1),
      * refIdxLXB1  !=  refIdxLXIvDC(with X being replaced by 0 and 1),

the entry mergeCandList[ numMergeCand ] is set equal to IvDC, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to 0 and the variable numMergeCand is increased by 1.

* 1. When availableFlagVSP is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to VSP, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal 1 and the variable numMergeCand is increased by 1.
  2. When availableFlagA0 is equal to 1 and numMergeCand is less than 5 + iv\_mv\_pred\_flag[ nuh\_layer\_id ]~~+ mpi\_flag[ nuh\_layer\_id ]~~, the entry mergeCandList[ numMergeCand ] is set equal to A0, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to VspModeFlag[ xPb − 1 ][ yPb + nPbH ] and the variable numMergeCand is increased by 1.
  3. When availableFlagB2 is equal to 1 and numMergeCand is less than 4 + iv\_mv\_pred\_flag[ nuh\_layer\_id ] +~~mpi\_flag[ nuh\_layer\_id ]~~, the entry mergeCandList[ numMergeCand ] is set equal to B2, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to VspModeFlag[ xPb − 1 ][ yPb − 1 ] and the variable numMergeCand is increased by 1.
  4. When availableFlagIvMCShift is equal to 1 and numMergeCand is less than 5 + iv\_mv\_pred\_flag[ nuh\_layer\_id ]~~+ mpi\_flag[ nuh\_layer\_id ]~~, and one or more of the following conditions are true,
     + - availableFlagIvMC  = =  0,
       - predFlagLXMC  !=  predFlagLXMCShift(with X being replaced by 0 and 1),
       - mvLXMC  !=  mvLXIvMCShift(with X being replaced by 0 and 1),
       - refIdxLXMC  !=  refIdxLXMCShift(with X being replaced by 0 and 1),

the entry mergeCandList[ numMergeCand ] is set equal to IvMCShift, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to 0 and the variable numMergeCand is increased by 1.

* 1. A variable availableFlagIvDCShift is set to 0 and when all of the following conditions are true
     + - DepthFlag is equal to 0,
       - availableFlagIvMCShift is equal to 0,
       - numMergeCand is less than 5 + iv\_mv\_pred\_flag[ nuh\_layer\_id ]~~+ mpi\_flag[ nuh\_layer\_id ]~~,

the derivation process for the shifted disparity merging candidate as specified in subclause H.8.5.3.2.15 is invoked with the availability flags availableFlagN, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N, of every candidate N being in mergeCandList, mergeCandList, mergeCandIsVspFlag, and numMergeCand as the inputs and the outputs are the flag availableFlagIvDCShift, the prediction utilization flags predFlagL0IvDCShift and predFlagL1IvDCShift, the reference indices refIdxL0IvDCShift and refIdxL1IvDCShift, and the motion vectors mvL0IvDCShift and mvL1IvDCShift. When availableFlagIvDCShift is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to IvDCShift, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to 0 and the variable numMergeCand is increased by 1.

When availableFlagCol is equal to 1 and numMergeCand is less than 5 + iv\_mv\_pred\_flag[ nuh\_layer\_id ] ~~+ mpi\_flag[ nuh\_layer\_id ]~~, the entry mergeCandList[ numMergeCand ] is set equal to Col, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to 0 and the variable numMergeCand is increased by 1.

…

H.8.5.3.2.14 Derivation process for the texture merging candidate

This process is not invoked when mpi\_flag[ nuh\_layer\_id ]is equal to 0.

…