|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  7th Meeting: Geneva, CH, 21-30 November, 2011 | Document: JCTVC-G314 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **AHG21: On DPB management** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Ye-Kui Wang Ying Chen  5775 Morehouse Dr San Diego, CA 92121 USA | Tel: Email: | 1-858-651-8345 [yekuiw@qualcomm.com](mailto:yekuiw@qualcomm.com)  1-858-845-6589 [cheny@qualcomm.com](mailto:cheny@qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

This document proposes to adopt the reference picture set based DPB management mechanism, including reference picture set signaling and derivation, reference picture list initialization and modification, as well as DPB operations, as specified in the AHG21 output text. Furthermore, a method for long-term reference picture support based on the reference picture set based DPB management mechanism is proposed.

# The reference picture set based DPB management mechanism

The reference picture set (RPS) based DPB management mechanism as specified in the AHG21 output text, as included in JCTVC-G021, was developed based on the basic idea in JCTVC-F493.

In this mechanism, the RPS for each coded picture is directly signaled. An RPS is the set of reference pictures associated with the coded picture, consisting of all reference pictures that are prior to the coded picture in decoding order, that may be used for inter prediction of the coded picture or any picture following the coded picture in decoding order.

Syntax elements for signaling of RPS are included in both the PPS and slice header. For a particular coded picture, the RPS may be one of those alternative sets included in the PPS as indicated by a flag in the slice header, or directly signaled in the slice header.

After parsing the slice header, subsets of the RPS, namely, RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, and RefPicSetStFoll1, can be derived. RefPicSetStCurr0 consists of all reference pictures that are prior to the current picture in both decoding order and output order, and that may be used in inter prediction of the current picture. RefPicSetStCurr1 consists of all reference pictures that are prior to the current picture in decoding order, that succeed the current picture in output order, and that may be used in inter prediction of the current picture. RefPicSetStFoll0 consists of all reference pictures that are prior to the current picture in both decoding order and output order, that may be used in inter prediction of one or more of the pictures following the current picture in decoding order, and that are not used in inter prediction of the current picture. RefPicSetStFoll1 consists of all reference pictures that are prior to the current picture in decoding order, that succeed the current picture in output order, that may be used in inter prediction of one or more of the pictures following the current picture in decoding order, and that are not used in inter prediction of the current picture.

The RPS subsets RefPicSetStCurr0, RefPicSetStCurr1 are then used in reference picture list initialization and reference picture list modification.

Besides the above, the AHG21 output text also includes DPB management operations based on RPS.

We propose to adopt the RPS based DPB management mechanism as specified in the AHG21 output text.

# Long-term reference picture support

In addition, we propose a long-term reference picture support, which is the same as specified in an earlier draft version of the AHG21 output text, and is described in this section. Changes are highlighted.

#### Picture parameter set RBSP syntax

|  |  |
| --- | --- |
| pic\_parameter\_set\_rbsp( ) { | Descriptor |
| **pic\_parameter\_set\_id** | ue(v) |
| **seq\_parameter\_set\_id** | ue(v) |
| **entropy\_coding\_mode\_flag** | u(1) |
| **num\_ref\_pic\_sets** | ue(v) |
| for(idx = 0; idx < num\_ref\_pic\_sets; idx++) |  |
| ref\_pic\_set( idx ) |  |
| **long\_term\_ref\_pics\_present\_flag** | u(1) |
| if( long\_term\_ref\_pics\_present\_flag ) |  |
| **delta\_poc\_lt\_len\_minus4** | ue(v) |
| **...** |  |
| } |  |

### Slice header syntax

|  |  |
| --- | --- |
| slice\_header( ) { | Descriptor |
| ... |  |
| else |  |
| **ref\_pic\_set\_idx** | u(v) |
| if( long\_term\_ref\_pics\_present\_flag ) { |  |
| **num\_long\_term\_pics** | ue(v) |
| for( i = 0; i < num\_long\_term\_pics; i++ ) { |  |
| **delta\_poc\_lt\_minus1[** i **]** | u(v) |
| **used\_by\_curr\_pic\_lt\_flag[** i **]** | u(1) |
| } |  |
| } |  |
| } |  |
| ... |  |
| } |  |
| } |  |

#### Picture parameter set RBSP semantics

**…**

**long\_term\_ref\_pics\_present\_flag** equal to 0 specifies that no long-term reference picture is used for inter prediction of any coded picture referring to the picture parameter set and the syntax element delta\_poc\_lt\_len\_minus4 is not present. long\_term\_ref\_pics\_present\_flag equal to 1 specifies that long-term reference pictures may be used for inter prediction of one or more coded pictures referring to the picture parameter set and the syntax element delta\_poc\_lt\_len\_minus4 is present.

**delta\_poc\_lt\_len\_minus4** plus 4 specifies the length, in bits, of the delta\_poc\_lt\_minus1[ i ] syntax element. The value of delta\_poc\_lt\_len\_minus4 shall be in the range of 0 to 12, inclusive.

**…**

### Slice header semantics

**…**

**num\_long\_term\_pics** specifies the number of the long-term reference pictures that are to be included in the long-term reference picture set of the current picture. The value of num\_long\_term\_pics shall be in the range of 0 to max\_num\_ref\_frames – NumNegativePics[ StRpsIdx ] – NumPositivePics[ StRpsIdx ], inclusive. When not present, the value of num\_long\_term\_pics shall be inferred to be equal to 0.

**delta\_poc\_lt\_minus1[** i **]** plus 1 specifies the difference between the picture order count of the current picture and the picture order count of the i-th long-term reference picture that is included in the long-term reference picture set of the current picture. The number of bits used to represent delta\_poc\_lt\_minus1[ i ] shall be equal to delta\_poc\_lt\_len\_minus4 + 4.

**used\_by\_curr\_pic\_lt\_flag[** i **]** equal to 1 indicates that the i-th long-term reference picture may be used for reference by the current picture, used\_by\_curr\_pic\_lt\_flag[ i ] equal to 0 indicates that the i-th long-term reference picture is not used for reference by the current picture.

**…**

### Decoding process for reference picture set

This process is invoked once per picture, after decoding of a slice header but prior to the decoding of any coding unit and prior to the decoding process for reference picture list construction of the slice as specified in subclause 8.2.3. The process may result in marking one or more reference pictures as "unused for reference".

The reference picture set consists of six subsets, namely RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr, and RefPicSetLtFoll. Each subset consists of a list of picture order count values. The number of entries in RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr, and RefPicSetLtFoll are NumRpsStCurr0, NumRpsStCurr1, NumRpsStFoll0, NumRpsStFoll1, NumRpsLtCurr, and NumRpsLtFoll respectively.

If the current picture is an IDR picture, RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr, and RefPicSetLtFoll are all set to empty, and NumRpsStCurr0, NumRpsStCurr1, NumRpsStFoll0, NumRpsStFoll1, NumRpsLtCurr, and NumRpsLtFoll are all set to 0.

Otherwise, the following applies for derivation of the six subsets and the numbers of entries.

for( i = 0, j = 0, k = 0; i < NumNegativePics[ StRpsIdx ] ; i++ ) {  
 if( UsedByCurrPicS0[ StRpsIdx ][ i ] )  
 RefPicSetStCurr0[ j++ ] = PicOrderCnt + DeltaPocS0[ StRpsIdx ][ i ]  
 else  
 RefPicSetStFoll0[ k++ ] = PicOrderCnt + DeltaPocS0[ StRpsIdx ][ i ]  
}  
NumRpsStCurr0 = j  
NumRpsStFoll0 = k  
  
for( i = 0, j = 0, k = 0; i < NumPositivePics[ StRpsIdx ]; i++ ) {  
 if( UsedByCurrPicS1[ StRpsIdx ][ i ] )  
 RefPicSetStCurr1[ j++ ] = PicOrderCnt + DeltaPocS1[ StRpsIdx ][ i ]  
 else  
 RefPicSetStFoll1[ k++ ] = PicOrderCnt + DeltaPocS1[ StRpsIdx ][ i ]  
}  
NumRpsStCurr1 = j  
NumRpsStFoll1 = k (8-x)

for( i = 0, j = 0, k = 0; i < num\_long\_term\_pics; i++ ) {  
 if( used\_by\_curr\_pic\_lt\_flag[ i ] )  
 RefPicSetLtCurr[ j++ ] = PicOrderCnt – ( delta\_poc\_lt\_minus1[ i ] + 1 )  
 else  
 RefPicSetLtFoll[ k++ ] = PicOrderCnt – ( delta\_poc\_lt\_minus1[ i ] + 1 )  
}  
NumRpsLtCurr = j  
NumRpsLtFoll = k

Where PicOrderCnt is the picture order count of the current picture as specified in subclause 8.2.1.

A reference picture with a particular value of PicOrderCnt is referred to as included in the reference picture set of a coded picture if the reference picture is included in any of the six subsets of the reference picture set. A reference picture with a particular value of PicOrderCnt is referred to as included in a particular subset of the reference picture set if the particular value of PicOrderCnt is equal to one of the PicOrderCnt values included in that subset.

The derived reference picture set is restricted as follows:

* There shall be no reference picture with temporal\_id greater than that of the current picture included in RefPicSetStCurr0, RefPicSetStCurr1 and RefPicSetLtCurr.
* A particular reference picture with a particular value of PicOrderCnt shall not be included in more than one of the six subsets of the reference picture set.
* A reference picture that is included in RefPicSetStCurr0, RefPicSetStCurr1 or RefPicSetLtCurr shall be present in the decoded picture buffer marked as "used for reference".

After the reference picture set is derived, the following applies:

* If the current picture is an IDR picture, all reference pictures are marked as "unused for reference".
* Otherwise (the current picture is not an IDR picture), all reference pictures marked as "used for reference" that are not included in the reference picture set are marked as "unused for reference". There may be one or more reference pictures that are included in the reference picture set but that are not present in the decoded picture buffer. If such a reference picture is included in RefPicSetStFoll0, RefPicSetStFoll1 or RefPicSetLtFoll the absence shall be ignored.

#### Reference picture list modification semantics

The syntax elements list\_modification\_idc and ref\_pic\_set\_idx specify the change from the initial reference picture lists to the reference picture lists to be used for decoding the slice.

**ref\_pic\_list\_modification\_flag\_l0** equal to 1 specifies that the syntax element list\_modification\_idc is present for specifying reference picture list 0. ref\_pic\_list\_modification\_flag\_l0 equal to 0 specifies that this syntax element is not present.

When ref\_pic\_list\_modification\_flag\_l0 is equal to 1, the number of times that list\_modification\_idc is not equal to 3 following ref\_pic\_list\_modification\_flag\_l0 shall not exceed num\_ref\_idx\_l0\_active\_minus1 + 1.

**ref\_pic\_list\_modification\_flag\_l1** equal to 1 specifies that the syntax element list\_modification\_idc is present for specifying reference picture list 1. ref\_pic\_list\_modification\_flag\_l1 equal to 0 specifies that this syntax element is not present.

When ref\_pic\_list\_modification\_flag\_l1 is equal to 1, the number of times that list\_modification\_idc is not equal to 3 following ref\_pic\_list\_modification\_flag\_l1 shall not exceed num\_ref\_idx\_l1\_active\_minus1 + 1.

**list\_modification\_idc** together with ref\_pic\_set\_idx specifies which of the reference pictures are re-mapped. The values of list\_modification\_idc are specified in Table 7‑4. The value of the first list\_modification\_idc that follows immediately after ref\_pic\_list\_modification\_flag\_l0 or ref\_pic\_list\_modification\_flag\_l1 shall not be equal to 3.

Table 7‑4 – list\_modification\_idc operations for modification of reference picture lists

|  |  |
| --- | --- |
| **list\_modification\_idc** | **modification specified** |
| 0 | For list 0: ref\_pic\_set\_idx is present and corresponds to an index to RefPicSetStCurr0; For list 1: ref\_pic\_set\_idx is present and corresponds to an index to RefPicSetStCurr1 |
| 1 | For list 0: ref\_pic\_set\_idx is present and corresponds to an index to RefPicSetStCurr1; For list 1: ref\_pic\_set\_idx is present and corresponds to an index to RefPicSetStCurr0 |
| 2 | ref\_pic\_set\_idx is present and corresponds to an index to RefPicSetLtCurr |
| 3 | End loop for modification of the initial reference picture list |

**ref\_pic\_set\_idx** specifies the index, to RefPicSetStCurr0, RefPicSetStCurr1 or RefPicSetLtCurr, of the reference picture being moved to the current index in the reference picture list. The value of ref\_pic\_set\_idx shall be in the range of 0 to max\_num\_ref\_frames, inclusive.

#### Initialisation process for reference picture lists

This initialisation process is invoked when decoding a P or B slice header.

When decoding a P or B slice, there shall be at least one reference picture in RefPicSetStCurr0, RefPicSetStCurr1 or RefPicSetLtCurr.

The following procedure is conducted to construct the initial RefPicList0:

cIdx = 0  
while( cIdx <= num\_ref\_idx\_l0\_active\_minus1 )  
{  
 for( i=0; i < NumRpsStCurr0 && cIdx <= num\_ref\_idx\_l0\_active\_minus1; cIdx++, i++ )  
 RefPicList0[ cIdx ] = RefPicSetStCurr0[ i ]   
 for( i=0; i < NumRpsStCurr1 && cIdx <= num\_ref\_idx\_l0\_active\_minus1; cIdx++, i++ )  
 RefPicList0[ cIdx ] = RefPicSetStCurr1[ i ]   
 for( i=0; i < NumRpsLtCurr && cIdx <= num\_ref\_idx\_l0\_active\_minus1; cIdx++, i++ )  
 RefPicList0[ cIdx ] = RefPicSetLtCurr[ i ]  
} (8-x)

The following procedure is conducted to construct the initial RefPicList1:

cIdx = 0  
while( cIdx <= num\_ref\_idx\_l1\_active\_minus1 )  
{  
 for( i=0; i < NumRpsStCurr1 && cIdx <= num\_ref\_idx\_l1\_active\_minus1; cIdx++, i++ )  
 RefPicList1[ cIdx ] = RefPicSetStCurr1[ i ]   
 for( i=0; i < NumRpsStCurr0 && cIdx <= num\_ref\_idx\_l1\_active\_minus1; cIdx++, i++ )  
 RefPicList1[ cIdx ] = RefPicSetStCurr0[ i ]   
 for( i=0; i < NumRpsLtCurr && cIdx <= num\_ref\_idx\_l1\_active\_minus1; cIdx++, i++ )  
 RefPicList1[ cIdx ] = RefPicSetLtCurr[ i ]  
} (8-x)

#### Modification process for reference picture lists

When ref\_pic\_list\_modification\_flag\_l0 is equal to 1, the following applies:

1. Let refIdxL0 be an index into the reference picture list RefPicList0. It is initially set equal to 0.
2. The corresponding syntax elements list\_modification\_idc are processed in the order they occur in the bitstream. For each of these syntax elements, the following applies.

– If list\_modification\_idc is equal to 0, 1, or 2, the process specified in subclause 8.2.3.2.1 is invoked with refIdxL0 as input, and the output is assigned to refIdxL0.

– Otherwise (list\_modification\_idc is equal to 3), the modification process for reference picture list RefPicList0 is finished.

When the current slice is a B slice and ref\_pic\_list\_modification\_flag\_l1 is equal to 1, the following applies:

1. Let refIdxL1 be an index into the reference picture list RefPicList1. It is initially set equal to 0.
2. The corresponding syntax elements list\_modification\_idc are processed in the order they occur in the bitstream. For each of these syntax elements, the following applies.

– If list\_modification\_idc is equal to 0, 1, or 2, the process specified in subclause 8.2.3.2.1 is invoked with refIdxL1 as input, and the output is assigned to refIdxL1.

– Otherwise (list\_modification\_idc is equal to 3), the modification process for reference picture list RefPicList1 is finished.

#### Moving of one reference picture to a reference picture list index

Input to this process is an index refIdxLX (with X being 0 or 1).

Output of this process is an incremented index refIdxLX.

– If list\_modification\_idc is equal to 0, the following applies.

– If the current reference picture list is RefPicList0, curRefPicSet is set to RefPicSetStCurr0.  
– Otherwise (the current reference picture list is RefPicList1), curRefPicSet is set to RefPicSetStCurr1.

– Otherwise, if list\_modification\_idc is equal to 1, the following applies.

– If the current reference picture list is RefPicList0, curRefPicSet is set to RefPicSetStCurr1.  
– Otherwise (the current reference picture list is RefPicList1), curRefPicSet is set to RefPicSetStCurr0.

– Otherwise, if list\_modification\_idc is equal to 2, curRefPicSet is set to RefPicSetLtCurr.

The variable pocLX is derived as follows.

pocLX = curRefPicSet[ ref\_pic\_set\_idx ] (8‑x)

The following procedure is conducted to place the picture picR with PicOrderCnt( picR ) equal to pocLX into the index position refIdxLX, shift the position of any other remaining pictures to later in the list, and increment the value of refIdxLX.

for( cIdx = num\_ref\_idx\_lX\_active\_minus1 + 1; cIdx > refIdxLX; cIdx-- )  
 RefPicListX[ cIdx ] = RefPicListX[ cIdx − 1]  
RefPicListX[ refIdxLX++ ] = pocLX  
nIdx = refIdxLX  
for( cIdx = refIdxLX; cIdx <= num\_ref\_idx\_lX\_active\_minus1 + 1; cIdx++ ) (8‑1)  
 if( PicOrderCnt( RefPicListX[ cIdx ] ) != pocLX )  
 RefPicListX[ nIdx++ ] = RefPicListX[ cIdx ]

NOTE 1 – Within this pseudo-code procedure, the length of the list RefPicListX is temporarily made one element longer than the length needed for the final list. After the execution of this procedure, only elements 0 through num\_ref\_idx\_lX\_active\_minus1 of the list need to be retained.

# 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).**