|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  9th Meeting: Geneva, CH, 27th April – 7th May, 2012 | Document: JCTVC-I0342 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **On reference picture set** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Ye-Kui Wang Adarsh K. Ramasubramonian Ying Chen  5775 Morehouse Drive, San Diego, CA 92121-1714 USA | Tel: Email: | 1-858-651-8345 [yekuiw@qualcomm.com](mailto:yekuiw@qualcomm.com)  1-858-658-5804 [aramasub@qualcomm.com](mailto:aramasub@qualcomm.com)  1-858-845-6589 [cheny@qualcomm.com](mailto:cheny@qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

This document proposes a changed method for derivation of reference picture set (RPS) and signaling of long-term reference pictures (LTRPs) to be included in the RPS of a coded picture in the slice header. It is reported that the proposed signaling of LTRPs in the slice header provides an average bit-count reduction of 28% compared to the method in the latest HEVC draft spec (WD 6) for the test case 2.7 in JCTVC-H0725 and a test cases wherein the first picture of the test sequences was the only LTRP signalled.

# Introduction

The current RPS derivation and signaling of LTRPs in the slice header have the following problems

1. As seen from HEVC WD6 subclause 8.3.2, Eqn. 8-6, the derivation of RPS subsets depends on the status of reference pictures in the DPB as a result of the previous picture (due to the wording of “there is a long-term reference picture picX in the DPB” or similar) – this is against the most fundamental principle of RPS: the status of reference pictures in the DPB is “intra coded”, i.e., not to depend on earlier status.
2. The second problem relates to the derivation of LTRPs in the RPS. From Eqn. 8-6 in subclause 8.3.2 of HEVC WD6, we can see that an LTRP is identified by its POC LSB (pic\_order\_cnt\_lsb) if delta\_poc\_msb\_present\_flag[ i ] is equal to 0, or its full POC value (PicOrderCntVal) otherwise. However, in the latter case, the part in Eqn. 8-5 for derivation of an LTRP’s full POC value, "( ( PicOrderCntVal − DeltaPocLt[ i ] + MaxPicOrderCntLsb ) % MaxPicOrderCntLsb ) – ( DeltaPocMSBCycleLt[ i ] )\* MaxPicOrderCntLsb", is simply incorrect. For example, if MaxPicOrderCntLsb = 256, and the current picture has full POC value equal to 256\*10 + 100 (LSB equal to 100), pictures with full POC values 256\*8 + 100 and 256\*6 + 100 are two LRTPs with the same POC LSB value 100. Then DeltaPocMSBCycleLt[ i ] for the two LRTPs would be equal to 2 and 4, respectively. However, according to Eqn. 8-5, the full POC values of the two LTRPs would be -256\*2 and -256\*4, respectively, which are obviously wrong.
3. Once a picture is signalled as a LTRP, it can never be signalled as a STRP.
4. The POC LSB of a LTRP to be included in the RPS of a picture is signalled in the slice header as a delta of the LSB of the delta POC values, coded using ue(v). This is inefficient, as the delta of the LSB of the delta POC values are typically big for LTRPs thus requiring many bits to represent. This is also not straightforward for encoders to figure out what value to signal in the slice header.
5. When there are two or more LTRPs having the same value of POC LSB, the full POC values of all these LTRPs must be signalled, wherein the POC MSB values are signalled as a delta of two continuous LTRPs having the same POC LSBs, using ue(v). For the first one of a set of continuously signalled LRTPs having the same POC LSBs, the POC MSB is directly signalled, using ue(v). There are two inefficiency aspects herein.
   1. The delta of POC MSB values and the directly signalled POC MSB value for the first LTRP in a set can be big thus requiring many bits to represent.
   2. In most cases, adding a few more LSB (or equivalently, making the length of the MSB a bit longer) would uniquely identify each of all the LTRPs, thus it is not necessary to signal all the MSB values for LTRPs having the same LSB values.

The proposal described below tries to keep the most fundamental principle of RPS, and also to solve other problems described above.

# Proposal

## Slice header syntax and semantics

The slice header syntax is changed as follows, with syntax elements poc\_lsb\_len\_delta[ i ] and poc\_lsb\_lt[ i ] added, syntax elements delta\_poc\_lsb\_lt[ i ], delta\_poc\_msb\_present\_flag[ i ] and delta\_poc\_msb\_cycle\_lt\_minus1[ i ] removed, and other syntax elements unchanged.

|  |  |
| --- | --- |
| slice\_header( ) { | Descriptor |
| **...** |  |
| if( long\_term\_ref\_pics\_present\_flag ) { |  |
| **num\_long\_term\_pics** | ue(v) |
| for( i = 0; i < num\_long\_term\_pics; i++ ) { |  |
| **poc\_lsb\_len\_delta**[ i ] | ue(v) |
| **poc\_lsb\_lt**[ i ] | u(v) |
| **used\_by\_curr\_pic\_lt\_flag**[ i ] | u(1) |
| } |  |
| } |  |
| **...** |  |
| } |  |

The slice header semantics is changed as follows, with semantics of syntax elements poc\_lsb\_len\_delta[ i ] and poc\_lsb\_lt[ i ] added, semantics of syntax elements delta\_poc\_lsb\_lt[ i ], delta\_poc\_msb\_present\_flag[ i ] and delta\_poc\_msb\_cycle\_lt\_minus1[ i ] removed, and semantics for other syntax elements unchanged.

**poc\_lsb\_len\_delta**[ i ] is used to specify the number of bits to represent poc\_lsb\_lt[ i ].

The variable PocLsbLtLen[ i ] is derived as follows.

if(i = = 0)   
 PocLsbLtLen[ i ] = log2\_max\_pic\_order\_cnt\_lsb\_minus4 + 4 + poc\_lsb\_len\_delta[ i ]  
else  
 PocLsbLtLen[ i ] = PocLsbLtLen[ i − 1 ]+ poc\_lsb\_len\_delta[ i ]

**poc\_lsb\_lt**[ i ] specifies the least significant bits of the picture order count value of the i-th long-term reference picture that is included in the long-term reference picture set of the current picture. The length of poc\_lsb\_lt[ i ] is PocLsbLtLen[ i ].

### 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 8.3.3. The process may result in marking one or more reference pictures as "unused for reference".

NOTE 1 – The reference picture set is an absolute description of the reference pictures used in the decoding process of the current and future coded pictures. The reference picture set signalling is explicit in the sense that all reference pictures included in the reference picture set are listed explicitly and there is no default reference picture set construction process in the decoder that depends on the status of the decoded picture buffer.

Short-term reference pictures are identified by their PicOrderCntVal values. Long-term reference pictures are identified by the least significant bits of their PicOrderCntVal values.

Five lists of picture order count values or least significant bits of picture order count are constructed to derive the reference picture set; PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, and PocLtFoll with NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, and NumPocLtFoll number of elements, respectively.

* If the current picture is an IDR picture, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, and PocLtFoll are all set to empty, and NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, and NumPocLtFoll are all set to 0.
* Otherwise, the following applies for derivation of the five lists of picture order count values and the numbers of entries.

for( i = 0, j = 0, k = 0; i < NumNegativePics[ StRpsIdx ] ; i++ )  
 if( UsedByCurrPicS0[ StRpsIdx ][ i ] )  
 PocStCurrBefore[ j++ ] = PicOrderCntVal + DeltaPocS0[ StRpsIdx ][ i ]  
 else  
 PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS0[ StRpsIdx ][ i ]  
NumPocStCurrBefore = j  
  
for( i = 0, j = 0; i < NumPositivePics[ StRpsIdx ]; i++ )  
 if( UsedByCurrPicS1[ StRpsIdx ][ i ] )  
 PocStCurrAfter[ j++ ] = PicOrderCntVal + DeltaPocS1[ StRpsIdx ][ i ]  
 else  
 PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS1[ StRpsIdx ][ i ]  
NumPocStCurrAfter = j  
NumPocStFoll = k (8‑5)

for( i = 0, j = 0, k = 0; i < num\_long\_term\_pics; i++ )  
 if( used\_by\_curr\_pic\_lt\_flag[ i ] ) {  
 PocLtCurr[ j ] = poc\_lsb\_lt[ i ]  
 PocLsbLenCurr[ j ] = PocLsbLtLen[ i ]  
 j++  
 }  
 else {  
 PocLtFoll[ k ] = poc\_lsb\_lt[ i ]  
 ExtPocLsbLenFoll[ k ] = PocLsbLtLen[ i ]  
 k++  
 }  
NumPocLtCurr = j  
NumPocLtFoll = k

where PicOrderCntVal is the picture order count of the current picture as specified in subclause 8.2.1.

NOTE 2 – A value of StRpsIdx in the range from 0 to num\_short\_term\_ref\_pic\_sets − 1, inclusive, indicates that a short-term reference picture set from the active sequence parameter set is being used, where StRpsIdx is the index of the short-term reference picture set to the list of short-term reference picture sets in the order in which they are signalled in the sequenceparameter set. StRpsIdx equal to num\_short\_term\_ref\_pic\_sets indicates that a short-term reference picture set explicitly signalled in the slice header is being used.

The reference picture set consists of five lists of reference pictures; RefPicSetStCurrBefore, RefPicSetStCurrAfter, RefPicSetStFoll, RefPicSetLtCurr and RefPicSetLtFoll. The variable NumPocTotalCurr is set equal to NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr. When decoding a P or B slice, it is a requirement of bitstream conformance that the value of NumPocTotalCurr shall not be equal to 0.

NOTE 3 – RefPicSetStCurrBefore, RefPicSetStCurrAfter and RefPicSetLtCurr contains all reference pictures that may be used in inter prediction of the current picture and that may be used in inter prediction of one or more of the pictures following the current picture in decoding order. RefPicSetStFoll and RefPicSetLtFoll consists of all reference pictures that are *not* used in inter prediction of the current picture but may be used in inter prediction of one or more of the pictures following the current picture in decoding order.

The marking of a reference picture can be "unused for reference", "used for short-term reference", or "used for long-term reference", but only one among these three. When a reference picture is referred to as being marked as "used for reference", this collectively refers to the picture being marked as "used for short-term reference" or "used for long-term reference" (but not both). A reference picture that is marked as "used for short-term reference" is referred to as a short‑term reference picture. A reference picture that is marked as "used for long-term reference" is referred to as a long‑term reference picture*.*

The derivation process for the reference picture set and picture marking are performed according to the following ordered steps, where DPB refers to the decoded picture buffer as described in Annex C:

1. The following applies:

for( i = 0; i < NumPocStCurrBefore; i++ )  
 if( there is a reference picture picX in the DPB with PicOrderCntVal equal to PocStCurrBefore[ i ] )  
 RefPicSetStCurrBefore[ i ] = picX  
 else  
 RefPicSetStCurrBefore[ i ] = "no reference picture"

for( i = 0; i < NumPocStCurrAfter; i++ )  
 if( there is a reference picture picX in the DPB with PicOrderCntVal equal to PocStCurrAfter[ i ] )  
 RefPicSetStCurrAfter[ i ] = picX  
 else  
 RefPicSetStCurrAfter[ i ] = "no reference picture" (8‑6)

for( i = 0; i < NumPocStFoll; i++ )  
 if( there is a reference picture picX in the DPB with PicOrderCntVal equal to PocStFoll[ i ] )  
 RefPicSetStFoll[ i ] = picX  
 else  
 RefPicSetStFoll[ i ] = "no reference picture"

1. All reference pictures included in RefPicSetStCurrBefore, RefPicSetStCurrAfter and RefPicSetStFoll are marked as "used for short-term reference".
2. The following applies:

for( i = 0; i < NumPocLtCurr; i++ ) {  
 if( there is a reference picture picX in the DPB, not included in RefPicSetStCurrBefore,  
 RefPicSetStCurrAfter or RefPicSetStFoll, with PicOrderCntVal for which  
 Abs( PicOrderCntVal) % ( 2PocLsbLenCurr[ i ] ) is equal to PocLtCurr[ i ] )  
 RefPicSetLtCurr[ i ] = picX  
 else   
 RefPicSetLtCurr[ i ] = "no reference picture"  
} (8‑7)

for( i = 0; i < NumPocLtFoll; i++ ) {  
 if( there is a reference picture picX in the DPB, not included in RefPicSetStCurrBefore,  
 RefPicSetStCurrAfter or RefPicSetStFoll, with PicOrderCntVal for which  
 Abs( PicOrderCntVal) % ( 2PocLsbLenFoll[ i ] ) is equal to PocLtFoll[ i ] )  
 RefPicSetLtFoll[ i ] = picX  
 else   
 RefPicSetLtFoll[ i ] = "no reference picture"  
}

1. All reference pictures included in RefPicSetLtCurr and RefPicSetLtFoll are marked as "used for long-term reference"
2. All reference pictures in the decoded picture buffer that are not included in RefPicSetLtCurr, RefPicSetLtFoll, RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetStFoll are marked as "unused for reference".

NOTE 4 – 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. When the first coded picture in the bitstream is an IDR picture or the current coded picture is not a leading picture of the first coded picture in the bitstream, entries in RefPicSetStFoll or RefPicSetLtFoll that are equal to "no reference picture" should be ignored. When the first coded picture in the bitstream is not a CRA picture or the current coded picture is not a leading picture of the first coded picture in the bitstream, an unintentional picture loss should be inferred for each entry in RefPicSetStCurrBefore, RefPicSetStCurrAfter and RefPicSetLtCurr that is equal to "no reference picture".

It is a requirement of bitstream conformance that the 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 RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetLtCurr.
* For each value of i in the range of 0 to NumPocLtCurr − 1, inclusive, there shall be no more than one reference picture in the DPB that is not included in RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetStFoll, and that has PicOrderCntVal for which Abs( PicOrderCntVal) % ( 2PocLsbLenCurr[ i ] is equal to PocLtCurr[ i ]. For each value of i in the range of 0 to NumPocLtFoll− 1, inclusive, there shall not be more than one reference picture in the DPB that is not included in RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetStFoll, and that has PicOrderCntVal for which Abs( PicOrderCntVal) % ( 2PocLsbLenFoll[ i ] is equal to PocLtFoll[ i ].
* There shall be no reference picture included in the reference picture set that precedes, in output order, any CRA picture that precedes the current picture both in decoding order and output order.
* When the first coded picture in the bitstream is an IDR picture or the current coded picture is not a leading picture of the first coded picture in the bitstream, there shall be no entry in RefPicSetStCurrBefore, RefPicSetStCurrAfter or RefPicSetLtCurr that is equal to "no reference picture".

NOTE 5 – A reference picture cannot be included in more than one of the five reference picture set lists.

# Signaling efficiency

We present the results obtained for signalling efficiency for the proposed method for signalling LTRPs as against the performance of the latest working draft WD6 [2]. We used the test case 2.7 specified in the common conditions for reference picture marking and list construction [1], where two scenes alternate after certain intervals of time and a picture from the two scenes before is used as a long term picture for the first picture of the current scene. We present the bit-counts of the related syntax elements. The syntax elements used for the bit-counts are as follows:

* For WD6: num\_long\_term\_pics, delta\_poc\_lsb\_lt, delta\_poc\_msb\_present\_flag, delta\_poc\_msb\_cycle\_lt\_minus1, and used\_by\_curr\_pic\_lt\_flag.
* For proposed method: num\_long\_term\_pics, poc\_lsb\_len\_delta, poc\_lsb\_lt, and used\_by\_curr\_pic\_lt\_flag.

The comparison of the bit-counts is presented in Table 1. We observe that the proposed method provides an average bit-count reduction of 28% compared to the WD6.

In tests 3 and 4, the first picture (IDR) of the test sequences was the only LTRP signalled.

Table : Comaprison of bit-count of LTRP-related syntax elements at the slice header.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Test 1 (10s- 30s – 10s – 30s) | | Test 2 (20s- 60s – 20s – 60s) | | Test 3 (IDR picture as LTRP, 80s) | | Test 4 (IDR picture as LTRP, 160s) | |
|  | CD | Proposed method | CD | Proposed method | CD | Proposed method | CD | Proposed method |
| Total bit-count | 63771 | 45927 | 127479 | 91975 | 72035 | 51879 | 144289 | 103879 |
| %reduction |  | 28.0% |  | 27.9% |  | 28.0% |  | 28.0% |

# References

[1] Y.-K. Wang, M. M. Hannuksela, T. K. Tan, R. Sjöberg, and Yan Ye, “Common conditions for reference picture marking and list construction proposals,” JCT-VC documents JCTVC-H0725, San Jose, February 2012.

[2] B. Bross, W-J. Han, J-R. Ohm, G. J. Sullivan, and T. Wiegand, “High efficiency video coding (HEVC) text specification draft 6,” JCT-VC document JCTVC-H1003\_dK, San Jose, February, 2012.

# 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).**