|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  10th Meeting: Stockholm, SE, 11 – 20 July 2012 | Document: JCTVC-J0120 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **AHG9 & AHG13: Identical reference picture lists** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Ying Chen Muhammed Coban Ye-Kui Wang Marta Karczewicz  5775 Morehouse Dr San Diego, CA 92121 USA | Tel: Email: | 1-858-845-6589 [cheny@qualcomm.com](mailto:cheny@qualcomm.com)  1-858-658-3973 [mcoban@qualcomm.com](mailto:mcoban@qualcomm.com)  1-858-651-8345 [yekuiw@qualcomm.com](mailto:yekuiw@qualcomm.com)  1-858-658-5673 [martak@qualcomm.com](mailto:martak@qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

In HEVC, for B slices, there are chances when RefPicList0 and RefPicList1 are identical, and when explicit weighted prediction is in use, the prediction weights are also identical. It is proposed to indicate this and thus reduces the bits for signalling RefPicList1 and its prediction weights. It is reported that the proposed algorithm saves about 48% for the number of bits used to signal weighted prediction parameters for the LB case and about 10% for the RA case.

# Introduction

HEVC encoders may generate bitstreams in which each slice has a RefPicList1 that is equal to RefPicList0. RefPicList0 and RefPicList1 are considered identical when each entry in RefPicList1 is the same as the entry in RefPicList0 with the same reference index.

In low-delay configurations, typically the two reference picture lists of each slice are identical.

When explicit weighted prediction is used, the entry in RefPicList0 and the identical entry in RefPicList1 may typically use the same set of prediction weights. In scenarios where weighted prediction is used and the bits used for weighted prediction are considered as noticeable overhead, it make sense to signal only the prediction weights for entries in RefPicList0 and reuse the weights for RefPicList1 entries, with simple syntax change in the prediction weight table.

# Proposal

It is proposed that an identical\_lists\_flag is signalled in the slice header for a B slice. When this flag is true, the following applies:

* No reference picture list construction for RefPicList1
* No reference picture list modification syntax for RefPicList1
* No prediction weights are signalled for RefPicList1

## Syntax

### Slice header syntax

|  |  |
| --- | --- |
| slice\_header( ) { | Descriptor |
| **first\_slice\_in\_pic\_flag** | u(1) |
| **pic\_parameter\_set\_id** | ue(v) |
| if( !first\_slice\_in\_pic\_flag ) |  |
| **slice\_address** | u(v) |
| if( dependent\_slice\_enabled\_flag && !first\_slice\_in\_pic\_flag ) |  |
| **dependent\_slice\_flag** | u(1) |
| if( !dependent\_slice\_flag ) { |  |
| **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( RapPicFlag ) { |  |
| **rap\_pic\_id** | ue(v) |
| **no\_output\_of\_prior\_pics\_flag** | u(1) |
| } |  |
| if( !IdrPicFlag ) { |  |
| **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 |  |
| **short\_term\_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++ ) { |  |
| **poc\_lsb\_lt**[ i ] | u(v) |
| **delta\_poc\_msb\_present\_flag**[ i ] | u(1) |
| if( delta\_poc\_msb\_present\_flag[ i ] ) |  |
| **delta\_poc\_msb\_cycle\_lt**[ i ] | ue(v) |
| **used\_by\_curr\_pic\_lt\_flag**[ i ] | u(1) |
| } |  |
| } |  |
| } |  |
| if( sample\_adaptive\_offset\_enabled\_flag ) { |  |
| **slice\_sao\_interleaving\_flag** | u(1) |
| **slice\_sample\_adaptive\_offset\_flag** | u(1) |
| if( slice\_sao\_interleaving\_flag &&  slice\_sample\_adaptive\_offset\_flag ) { |  |
| **sao\_cb\_enable\_flag** | u(1) |
| **sao\_cr\_enable\_flag** | u(1) |
| } |  |
| } |  |
| if( ( sample\_adaptive\_offset\_enabled\_flag && !slice\_sao\_interleaving\_flag ) | |   adaptive\_loop\_filter\_enabled\_flag ) |  |
| **aps\_id** | ue(v) |
| if( slice\_type = = P | | slice\_type = = B ) { |  |
| if( sps\_temporal\_mvp\_enable\_flag ) |  |
| **pic\_temporal\_mvp\_enable\_flag** | u(1) |
| **identical\_lists\_flag** | u(1) |
| **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 ) |  |
| if( !identical\_lists\_flag ) |  |
| **num\_ref\_idx\_l1\_active\_minus1** | ue(v) |
| } |  |
| **…** |  |
| byte\_alignment( ) |  |
| } |  |

### Reference picture list modification syntax

|  |  |
| --- | --- |
| ref\_pic\_list\_modification( ) { | Descriptor |
| if( slice\_type = = P | | slice\_type = = B ) { |  |
| **ref\_pic\_list\_modification\_flag\_l0** | u(1) |
| if( ref\_pic\_list\_modification\_flag\_l0 && NumPocTotalCurr > 1 ) |  |
| for( i = 0; i <= num\_ref\_idx\_l0\_active\_minus1; i++ ) |  |
| **list\_entry\_l0**[ i ] | u(v) |
| } |  |
| if( slice\_type = = B && !identical\_lists\_flag ) { |  |
| **ref\_pic\_list\_modification\_flag\_l1** | u(1) |
| if( ref\_pic\_list\_modification\_flag\_l1 && NumPocTotalCurr > 1 ) |  |
| for( i = 0; i <= num\_ref\_idx\_l1\_active\_minus1; i++ ) |  |
| **list\_entry\_l1**[ i ] | u(v) |
| } |  |
| } |  |

### Prediction weight table syntax

|  |  |
| --- | --- |
| pred\_weight\_table( ) { | Descriptor |
| **luma\_log2\_weight\_denom** | ue(v) |
| if( chroma\_format\_idc != 0 ) |  |
| **delta\_chroma\_log2\_weight\_denom** | se(v) |
| if( slice\_type = = P | | slice\_type = = B ) |  |
| for( i = 0; i <= num\_ref\_idx\_l0\_active\_minus1; i++ ) { |  |
| **luma\_weight\_l0\_flag** | u(1) |
| if( luma\_weight\_l0\_flag ) { |  |
| **delta\_luma\_weight\_l0[** i **]** | se(v) |
| **luma\_offset\_l0[** i **]** | se(v) |
| } |  |
| if( chroma\_format\_idc != 0 ) { |  |
| **chroma\_weight\_l0\_flag** | u(1) |
| if( chroma\_weight\_l0\_flag ) |  |
| for( j =0; j < 2; j++ ) { |  |
| **delta\_chroma\_weight\_l0[** i **][** j **]** | se(v) |
| **delta\_chroma\_offset\_l0[** i **][** j **]** | se(v) |
| } |  |
| } |  |
| } |  |
| if( slice\_type = = B && !identical\_lists\_flag ) |  |
| for( i = 0; i <= num\_ref\_idx\_l1\_active\_minus1; i++ ) { |  |
| **luma\_weight\_l1\_flag** | u(1) |
| if( luma\_weight\_l1\_flag ) { |  |
| **delta\_luma\_weight\_l1[** i **]** | se(v) |
| **luma\_offset\_l1[** i **]** | se(v) |
| } |  |
| if( chroma\_format\_idc != 0 ) { |  |
| **chroma\_weight\_l1\_flag** | u(1) |
| if( chroma\_weight\_l1\_flag ) |  |
| for( j = 0; j < 2; j++ ) { |  |
| **delta\_chroma\_weight\_l1[** i **][** j **]** | se(v) |
| **delta\_chroma\_offset\_l1[** i **][** j **]** | se(v) |
| } |  |
| } |  |
| } |  |
| } |  |

## Semantics

### Slice header semantics

**identical\_lists\_flag** equal to 1 indicates that RefPicList0 and RefPicList1 are identical for the current slice, which is a B slice. identical\_lists\_flagequal to 0 indicates that RefPicList0 and RefPicList1 are not identical for the current slice, which is a B slice.

When RefPicList0 and RefPicList1 are identical, each entry of RefPicList1 points to the same picture as the entry in RefPicList0 with the same value of reference index. In addition, when explicit weighted prediction is used, each entry in RefPicList1 has the same set of prediction weights as the entry in RefPicList0 with same value of reference index.

# Simulation results

The proposed identical list signalling has been implemented in HM7.0. The simulations were conducted based on the JCT-VC common test condition defined in [1] using fade-to-black and fade-to-white sequences. The proposed changes affect RA\_Main, RA\_HE10, LB\_Main, and LB\_HE10 configurations. Table 1 shows a summary of the simulations for fade-to-black sequences and Table 2 shows a summary of the simulations for fade-to-white sequences, both with explicit weighted prediction.

Note that the implementation was only done for weighted prediction signaling.

Table 1: Summary of simulation results for proposed algorithm on fade-to-black sequences with weighted prediction.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Access Main** | | | **Random Access HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class B | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class C | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% |
| Class D | -0.2% | -0.2% | -0.2% | -0.4% | -0.4% | -0.4% |
| Class E |  |  |  |  |  |  |
| **Overall** | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% |
| Enc Time[%] | 98% | | | 99% | | |
| Dec Time[%] | 100% | | | 100% | | |
|  |  |  |  |  |  |  |
|  | **Low delay B Main** | | | **Low delay B HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | -0.3% | -0.3% | -0.3% | -0.5% | -0.4% | -0.4% |
| Class C | -0.7% | -0.6% | -0.6% | -1.1% | -1.0% | -1.0% |
| Class D | -2.2% | -2.0% | -2.1% | -3.4% | -3.1% | -3.1% |
| Class E | -2.3% | -2.0% | -2.0% | -3.9% | -3.5% | -3.5% |
| **Overall** | -1.3% | -1.1% | -1.1% | -2.0% | -1.8% | -1.8% |
| Enc Time[%] | 101% | | | 102% | | |
| Dec Time[%] | 102% | | | 100% | | |

Table 2: Summary of simulation results for proposed algorithm on fade-to-white sequences with weighted prediction.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Access Main** | | | **Random Access HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class B | 0.0% | 0.0% | 0.0% | -0.1% | 0.0% | 0.0% |
| Class C | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% |
| Class D | -0.3% | -0.3% | -0.3% | -0.5% | -0.4% | -0.4% |
| Class E |  |  |  |  |  |  |
| **Overall** | -0.1% | -0.1% | -0.1% | -0.2% | -0.1% | -0.2% |
| Enc Time[%] | 96% | | | 96% | | |
| Dec Time[%] | 100% | | | 101% | | |
|  |  |  |  |  |  |  |
|  | **Low delay B Main** | | | **Low delay B HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | -0.2% | -0.4% | 0.0% | -0.6% | -0.5% | -0.5% |
| Class C | -0.9% | -0.8% | -0.8% | -1.3% | -1.1% | -1.1% |
| Class D | -2.8% | -2.5% | -2.6% | -3.9% | -3.5% | -3.6% |
| Class E | -3.0% | -2.5% | -2.6% | -4.5% | -4.0% | -4.1% |
| **Overall** | -1.5% | -1.4% | -1.3% | -2.3% | -2.1% | -2.1% |
| Enc Time[%] | 99% | | | 100% | | |
| Dec Time[%] | 101% | | | 101% | | |

The average bit savings for signalling of weighted prediction parameters in the slice header have been measured for RA and LB conditions respectively. The results are tabulated below in Tables 3 and 4 for fade-to-black and fade-to-white sequences. As can be seen from the results, the proposed algorithm saves ~48% for the number of bits used to signal weighted prediction parameters for the LB case and ~10% for the RA case.

Table 3: Summary of simulation results for measuring weighted prediction parameter bits in the slice header for fade-to-black sequences.



Table 4: Summary of simulation results for measuring weighted prediction parameter bits in the slice header for fade-to-white sequences.



# References

1. F. Bossen, “Common test conditions and software reference configurations,” JCT-VC document JCTVC-I1100, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP3 and ISO/IEC JTC1/SC29/WG11 9th meeting, Geneva, April 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).**