|  |  |
| --- | --- |
| **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, 16 July – 20 July 2012 | Document: JCTVC-J0055 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **On Weighted Prediction Parameter Signalling** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Yong He, Yan Ye  9710 Scranton Rd, #250 San Diego, CA 92121 | Tel: Email: | +1-858-210-4803 [yong.he@interdigital.com](mailto:Yong.he@interdigital.com)  [yan.ye@interdigital.com](mailto:yan.ye@interdigital.com) |
| *Source:* | InterDigital Communications, LLC | | |

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

# Abstract

In HEVC working draft 7, the explicit Weighted Prediction parameters in B slices are always signalled for both reference picture list 0 and reference picture list 1 after list combination was removed at the 9th JCT-VC meeting. In this contribution, two revised WP parameter signalling methods are proposed. Up to two 1-bit flags are added to pred\_weight\_table(), with the first 1-bit flag indicating if any WP parameters are signalled for the entire list 1, and the second 1-bit flag indicating if any WP parameters are signalled for a particular entry on list 1. Simulation results show that, for fade-white sequences, when compared to HM7.0, the proposed signalling method 1 achieves BD rate of -0.1% and -1.6% for random access main and for low-delay B main, respectively. The proposed signalling method 2 achieves BD rate of -0.2% for random access main and -1.6% for low delay B main. Similar RD performance gains are observed for HE10 configurations and for fade-black sequences.

# Problem statement

At the 9th JCT-VC meeting, removal of reference list combination [4] was adopted. Table 1 shows the syntax table pred\_weight\_table() in the current HEVC working draft 7 [1].

As shown in Table 2, in the random access setting in the common test conditions [2], the reference pictures on list 0 and list 1 are often duplicated, whereas in the low delay B setting, list 0 and list 1 are completely identical. Since the WP parameters are always signalled for L0 and L1 in B slices according to Table 1, signalling redundancy can be significant.

1. WP parameter signaling in [1]

|  |  |
| --- | --- |
| 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 ) |  |
| 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) |
| } |  |
| } |  |
| } |  |
| } |  |

1. Reference picture list 0 and list 1 in random access

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **POC** | **RPS** | | **Ref pic lists** | |
| **StCurr0** | **StCurr1** | **L0** | **L1** |
| 16 | {8, 6, 4, 0} | {NULL} | {8, 6, 4, 0} | {8, 6, 4, 0} |
| 12 | {8, 6} | {16} | {8, 6} | {16, 8} |
| 10 | {8, 6} | {12, 16} | {8, 6} | {12, 16} |
| 9 | {8} | {10, 12, 16} | {8, 10} | {10, 12} |
| 11 | {10, 8} | {12, 16} | {10, 8} | {12, 16} |
| 14 | {12, 10, 8} | {16} | {12, 10} | {16, 12} |
| 13 | {12, 8} | {14, 16} | {12, 8} | {14, 16} |
| 15 | {14, 12, 8} | {16} | {14, 12} | {16, 14} |

# Proposed weighted prediction parameter signalling

## Method 1

To reduce signaling overhead, in the first method, we propose to add a 1-bit flag, **weights\_l1\_present\_flag,** to indicate whether any WP parameters for the L1 entries are signaled separately in the bitstream. As shown in Table 3, we condition this flag based on whether L0 and L1 have the same size, instead of on whether L0 and L1 are identical. This is because the latter decision requires L0 and L1 to be constructed first, causing interruption to the slice header parsing process. weights\_l1\_present\_flag equal to 1 specifies that syntax elements used to derive WP parameters of L1 are present. When weights\_l1\_present\_flag is equal to 0, syntax elements used to derive the WP parameters of L1 are not present; instead, the WP parameters of L1 are inferred. When weights\_l1\_present\_flag is not present, it is inferred to be 1.

1. Proposed WP parameter signaling (method 1)

|  |  |
| --- | --- |
| 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 ) { |  |
| if( num\_ref\_idx\_l0\_active\_minus1 == num\_ref\_idx\_l1\_active\_minus1 ) |  |
| **weights\_l1\_present\_flag** | u(1) |
| if(weights\_l1\_present\_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) |
| } |  |
| } |  |
| } |  |
| } |  |
| } |  |
| } |  |

### Prediction weight table semantics (method 1)

**weights\_l1\_present\_flag** equal to 1 specifies that syntax elements used to derive the weighting factors LumaWeightL1[ i ] and ChromaWeightL1 [ i ][ j ] and the additive offsets LumaOffsetL1[ i ] and ChromaOffsetL1 [ i ][ j ] for the luma and chroma prediction values of list 1 prediction are present. When weights\_l1\_present\_flag is equal to 0, syntax elements used to derive the weighting factors and the additive offsets for luma and chroma prediction values of list 1 prediction are not present. When weights\_l1\_present\_flag is not present, it is inferred to be 1. When weights\_l1\_present\_flag is equal to 0, the weighting factors LumaWeightL1[ i ] and ChromaWeightL1 [ i ] and the additive offsets LumaOffsetl1[ i ] and ChromaOffsetl1[ i ] are derived by invoking the decoding process in 8.3.6.

## Method 2

As discussed above, even if L0 and L1 are not identical, they can have some repeated entries. An example is given in Figure 1, where the L0 and L1 entries for POC=12 in random access setting in common testing conditions [2][2] are shown. Since ref POC = 8 appears on both L0 and L1, signaling the WP parameters twice incurs additional bit overhead.



1. Redundant signaling of WP parameters when L0 and L1 have overlapping entries

In our second method, in addition to the flag weights\_l1\_present\_flag introduced in 2.1, we introduce another 1-bit flag, delta\_params\_present\_flag, for each entry in L1 to further reduce the overhead. Table 4 is the proposed syntax table for pred\_weight\_table() using method 2.

1. Proposed syntax table for WP parameters signaling (method 2)

|  |  |
| --- | --- |
| pred\_weight\_table( ) { | Descriptor |
| **luma\_log2\_weight\_denom** | ue(v) |
| if( chroma\_format\_idc != 0 ) |  |
| **delta\_chroma\_log2\_weight\_denom** | se(v) |
| for( i = 0; i <= num\_ref\_idx\_l0\_active\_minus1; i++ ) { |  |
| **luma\_weight\_l0\_flag** [ i ] | u(1) |
| if( luma\_weight\_l0\_flag [ i ]) { |  |
| **delta\_luma\_weight\_l0**[ i ] | se(v) |
| ~~luma\_offset\_l0[ i ]~~ **delta\_luma\_offset\_l0**[ i ] | se(v) |
| } |  |
| if( chroma\_format\_idc != 0 ) { |  |
| chroma\_weight\_l0\_flag [ i ] | u(1) |
| if( chroma\_weight\_l0\_flag [ i ]) |  |
| 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 ) |  |
| if( num\_ref\_idx\_l0\_active\_minus1 == num\_ref\_idx\_l1\_active\_minus1 ) |  |
| **weights\_l1\_present\_flag** | u(1) |
| if(weights\_l1\_present\_flag) { |  |
| for( i = 0; i <= num\_ref\_idx\_l1\_active\_minus1; i++ ) { |  |
| **delta\_params\_present\_flag** [ i ] |  |
| if(delta\_params\_present\_flag [ i ]) { |  |
| **luma\_weight\_l1\_flag** [ i ] | u(1) |
| if( luma\_weight\_l1\_flag [ i ] ) { |  |
| **delta\_luma\_weight\_l1**[ i ] | se(v) |
| ~~luma\_offset\_l1[ i ]~~ **delta\_luma\_offset\_l1**[ i ] | se(v) |
| } |  |
| if( chroma\_format\_idc != 0 ) { |  |
| **chroma\_weight\_l1\_flag** [ i ] | 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) |
| } |  |
| } |  |
| } |  |
| } |  |
| } |  |
| } |  |

### Proposed weighted prediction semantics (method 2)

**luma\_log2\_weight\_denom** is the base 2 logarithm of the denominator for all luma weighting factors. The value of luma\_log2\_weight\_denom shall be in the range of 0 to 7, inclusive.

**delta\_chroma\_log2\_weight\_denom** is the difference of the base 2 logarithm of the denominator for all chroma weighting factors.

The variable ChromaLog2WeightDenom is specified by luma\_log2\_weight\_denom + delta\_chroma\_log2\_weight\_denom and it shall be in the range of 0 to 7, inclusive.

**luma\_weight\_l0\_flag[** i **]** equal to 1 specifies that weighting factors for the luma component of list 0 prediction using RefPicList0[ i ] are present. luma\_weight\_l0\_flag[ i ] equal to 0 specifies that these weighting factors are not present.

**delta\_luma\_weight\_l0[** i **]** is the difference of the weighting factor applied to the luma prediction value for list 0 prediction using RefPicList0[ i ].

The variable LumaWeightL0[ i ] is specified ~~by (1 << luma\_log2\_weight\_denom ) + delta\_luma\_weight\_l0[ i ]~~ by invoking the process in 8.3.6. ~~When luma\_weight\_l0\_flag is equal to 1,~~ The value of LumaWeightL0[ i ] shall be in the range of −128 to 127, inclusive. ~~When luma\_weight\_l0\_flag~~~~is equal to 0, LumaWeightL0[ i ] is inferred to be equal to 2~~~~luma\_log2\_weight\_denom~~ ~~for RefPicList0[ i ].~~

**~~luma\_offset\_l0[~~**~~i~~**~~]~~** **delta\_luma\_offset\_l0**[ i ] is the difference of the additive offset applied to the luma prediction value for list 0 prediction using RefPicList0[ i ]. The variable LumaOffsetL0[ i ] is specified by invoking the process in 8.3.7. The value of ~~luma\_offset\_l0~~LumaOffsetL0[ i ] shall be in the range of −128 to 127, inclusive. ~~When luma\_weight\_l0\_flag~~~~is equal to 0, luma\_offset\_l0LumaOffsetL0[ i ] is inferred as equal to be 0 for RefPicList0[ i ].~~

**chroma\_weight\_l0\_flag**[ i ] equal to 1 specifies that weighting factors for the chroma prediction values of list 0 prediction using RefPicList0[ i ] are present. chroma\_weight\_l0\_flag[ i ] equal to 0 specifies that these weighting factors are not present.

**delta\_chroma\_weight\_l0[** i **][** j **]** is the difference of the weighting factor applied to the chroma prediction values for list 0 prediction using RefPicList0[ i ] with j equal to 0 for Cb and j equal to 1 for Cr.

The variable ChromaWeightL0[ i ][ j ] is specified ~~by ( 1 << ChromaLog2WeightDenom ) + delta\_chroma\_weight\_l0[ i ][ j ]~~ by invoking the process in 8.3.6. ~~When chroma\_weight\_l0\_flag is equal to 1, the~~ The value of ChromaWeightL0[ i ][ j ] shall be in the range of −128 to 127, inclusive. ~~When chroma\_weight\_l0\_flag is equal to 0~~**~~,~~** ~~ChromaWeightL0[ i ][ j ] is inferred to be equal to 2~~~~ChromaLog2WeightDenom~~ ~~for RefPicList0[ i ].~~

**delta\_chroma\_offset\_l0[** i **][** j **]** is the difference of the additive offset applied to the chroma prediction values for list 0 prediction using RefPicList0[ i ] with j equal to 0 for Cb and j equal to 1 for Cr.

The variable ChromaOffsetL0[ i ][ j ] is specified by invoking the process in 8.3.6. ~~as follows:~~

~~shift = 1 << ( BitDepth~~~~C~~ ~~− 1 )~~

~~ChromaOffsetL0[ i ][ j ] = (delta\_chroma\_offset\_l0[i][j] –   
 ( (shift\*ChromaWeightL0[ i ][ j ]) >> ChromaLog2WeightDenom ) − shift ) (7‑70)~~

The variable ChromaOffsetL0[ i ][ j ] shall be in the range of −127 to 128, inclusive. ~~When chroma\_weight\_l0\_flag is equal to 0~~**~~,~~** ~~ChromaOffsetL0[ i ][ j ] is inferred to be equal to 0 for RefPicList0[ i ].~~

**weights\_l1\_present\_flag** equal to 1 specifies that syntax elements used to derive the weighting factors LumaWeightL1[ i ] and ChromaWeightL1 [ i ][ j ] and the additive offsets LumaOffsetL1[ i ] and ChromaOffsetL1 [ i ][ j ] for the luma and chroma prediction values of list 1 prediction are present. When weights\_l1\_present\_flag is equal to 0, syntax elements used to derive the weighting factors and the additive offsets for luma and chroma prediction values of list 1 prediction are not present. When weights\_l1\_present\_flag is not present, it is inferred to be 1. When weights\_l1\_present\_flag is equal to 0, the weighting factors LumaWeightL1[ i ] and ChromaWeightL1 [ i ] and the additive offsets LumaOffsetl1[ i ] and ChromaOffsetl1[ i ] are derived by invoking the decoding process in 8.3.6

**delta\_params\_present\_flag**[ i ] equal to 1 specifies that the syntax elements luma\_weight\_l1\_flag, delta\_luma\_weight\_l1, delta\_luma\_offset\_l1, chroma\_weight\_l1\_flag, delta\_chroma\_weight\_l1, delta\_chroma\_offset\_l1 are present. When delta\_params\_present\_flag is equal to 0, the syntax elements luma\_weight\_l1\_flag, delta\_luma\_weight\_l1, delta\_luma\_offset\_l1, chroma\_weight\_l1\_flag, delta\_chroma\_weight\_l1, delta\_chroma\_offset\_l1 are not present.

**luma\_weight\_l1\_flag, delta\_luma\_weight\_l1**, **delta\_luma\_offset\_l1**, **chroma\_weight\_l1\_flag**, **delta\_chroma\_weight\_l1**, **delta\_chroma\_offset\_l1** have the same semantics as luma\_weight\_l0\_flag, delta\_luma\_weight\_l0, delta\_luma\_offset\_l0, chroma\_weight\_l0\_flag, delta\_chroma\_weight\_l0, delta\_chroma\_offset\_l0, respectively, with l0, list 0, and List0 replaced by l1, list 1, and List1, respectively.

The values of LumaWeightL1[ i ], LumaOffsetL1[ i ] , ChromaWeightL1[ i ][ j ], ChromaOffsetL1[ i ][ j ] are derived by invoking the process in 8.3.6. The value of LumaWeightL1[ i ], LumaOffsetL1[ i ], ChromaWeightL1[ i ][ j ], ChromaOffsetL1[ i ][ j ] shall be in the range of −128 to 127, inclusive.

WD changes, including syntax, semantics and decoding process changes, related to the proposed schemes can be found in the separate file attached with this contribution.

# Simulation results

The proposed two methods are implemented on HM7.0 to evaluate the performance. The fading sequences used in the simulation are generated using the fading tool in [3].

## Method 1

Table 5 and Table 6 show the BD performance of method 1 for fade-black and fade-white sequences, respectively. For main profile, the overall gain is up to 1.6% for LDB mode, and up to 0.1% for RA mode. The overall gain is slightly higher for HE10 setting. The encoding and decoding times are not reliable since our computing grid consists of cores with various speeds.

1. BD performance of proposed method 1 (Fade to Black)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Access Main** | | | **Random Access HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A (8bit) | 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% | -0.2% | -0.1% | -0.1% |
|  | -0.1% | -0.1% | -0.1% | -0.2% | -0.1% | -0.1% |
| Class F | 0.0% | 0.0% | 0.0% | -0.1% | -0.1% | -0.1% |
| Enc Time[%] | 135% | | | 153% | | |
| Dec Time[%] | 148% | | | 176% | | |
|  |  |  |  |  |  |  |
|  | **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% |
|  | -1.3% | -1.1% | -1.2% | -2.0% | -1.8% | -1.8% |
| Class F | -0.8% | -0.7% | -0.8% | -1.4% | -1.3% | -1.3% |
| Enc Time[%] | 89% | | | 151% | | |
| Dec Time[%] | 88% | | | 175% | | |

1. BD performance of proposed method 1 (Fade to White)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Access Main** | | | **Random Access HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A (8bit) | 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% | -0.2% | -0.2% |
|  | -0.1% | -0.1% | -0.1% | -0.2% | -0.2% | -0.2% |
| Class F | -0.1% | 0.0% | 0.0% | -0.1% | -0.1% | -0.1% |
| Enc Time[%] | 137% | | | 99% | | |
| Dec Time[%] | 150% | | | 100% | | |
|  |  |  |  |  |  |  |
|  | **Low delay B Main** | | | **Low delay B HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | -0.4% | -0.3% | -0.3% | -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** | -1.6% | -1.4% | -1.4% | -2.3% | -2.1% | -2.1% |
|  | -1.6% | -1.4% | -1.5% | -2.3% | -2.1% | -2.1% |
| Class F | -1.0% | -0.9% | -0.9% | -1.6% | -1.4% | -1.4% |
| Enc Time[%] | 91% | | | 99% | | |
| Dec Time[%] | 89% | | | 98% | | |

## Method 2

Table 7 and Table 8 are the BD performance for the proposed method 2 for the fade-black and fade-white sequences, respectively. Compared to method 1, the additional flag delta\_params\_present\_flag for L1 entries offers extra 0.1% gain for the random access setting.

1. BD performance of proposed method 2 (Fade to Black)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Access Main** | | | **Random Access HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A (8bit) | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class B | -0.1% | 0.0% | 0.0% | -0.1% | -0.1% | -0.1% |
| Class C | -0.1% | -0.1% | -0.1% | -0.2% | -0.2% | -0.2% |
| Class D | -0.4% | -0.4% | -0.4% | -0.7% | -0.7% | -0.7% |
| Class E |  |  |  |  |  |  |
| **Overall** | -0.2% | -0.2% | -0.2% | -0.3% | -0.3% | -0.3% |
|  | -0.2% | -0.2% | -0.2% | -0.3% | -0.3% | -0.3% |
| Class F | -0.1% | -0.1% | -0.1% | -0.2% | -0.2% | -0.2% |
| Enc Time[%] | 134% | | | 100% | | |
| Dec Time[%] | 147% | | | 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% |
|  | -1.3% | -1.1% | -1.2% | -2.0% | -1.8% | -1.8% |
| Class F | -0.8% | -0.7% | -0.8% | -1.4% | -1.3% | -1.3% |
| Enc Time[%] | 88% | | | 97% | | |
| Dec Time[%] | 86% | | | 97% | | |

1. BD performance of proposed method 2 (Fade to White)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Access Main** | | | **Random Access HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A (8bit) | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class B | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% |
| Class C | -0.2% | -0.2% | -0.2% | -0.3% | -0.2% | -0.2% |
| Class D | -0.6% | -0.5% | -0.5% | -0.9% | -0.8% | -0.8% |
| Class E |  |  |  |  |  |  |
| **Overall** | -0.2% | -0.2% | -0.2% | -0.3% | -0.3% | -0.3% |
|  | -0.2% | -0.2% | -0.2% | -0.3% | -0.3% | -0.3% |
| Class F | -0.1% | -0.1% | -0.1% | -0.2% | -0.2% | -0.2% |
| Enc Time[%] | 160% | | | 150% | | |
| Dec Time[%] | 178% | | | 173% | | |
|  |  |  |  |  |  |  |
|  | **Low delay B Main** | | | **Low delay B HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | -0.4% | -0.3% | -0.3% | -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** | -1.6% | -1.4% | -1.4% | -2.3% | -2.1% | -2.1% |
|  | -1.6% | -1.4% | -1.5% | -2.3% | -2.1% | -2.1% |
| Class F | -1.0% | -0.9% | -0.9% | -1.6% | -1.4% | -1.4% |
| Enc Time[%] | 101% | | | 155% | | |
| Dec Time[%] | 101% | | | 178% | | |

# References

1. [B. Bross](mailto:benjamin.bross@hhi.fraunhofer.de), [W.-J. Han](mailto:wjhan.han@samsung.com), [J.-R. Ohm](mailto:ohm@ient.rwth-aachen.de), [G. J. Sullivan](mailto:garysull@microsoft.com), [T. Wiegand](mailto:thomas.wiegand@hhi.fraunhofer.de). High efficiency video coding (HEVC) text specification draft 7. Document no JCTVC-I1003\_d5. May 2012.
2. F. Bossen. Common HM test conditions and software reference configurations. Document no JCTVC-I1100. May 2012.
3. P. Bordes, T.K. Tan, “JCT-VC AHG report: Weighted prediction (AHG 18)”, Document no JCTVC-F018, July 2011.
4. Tammy Lee, Jeonghoon Park, “On Reference List Combination”, Document no JCTVC-I0125, May 2012

# Patent rights declaration(s)

**InterDigital Communications, LLC 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).**