JCTVC-I0279

Proposed Modifications of explicit weighted prediction

Akiyuki Tanizawa
Takeshi Chujoh
Tomoo Yamakage

TOSHIBA
Leading Innovation >>>

Summary

- ## Weighted prediction of current WD6
  - Two issues on WP parameters signaling
  - One editorial issue on the mapping array of WP parameters

- ## Proposal
  - Modified and simplified syntax, semantics and decoding process are proposed

- ## Reference
  - Our proposal is basically similar as Method 3 in JCTVC-I0335 by Qualcomm and the proposal#2 of JCTVC-I0439 by LGE
  - JCTVC-I0279 by InterDigital propose an another solution to fix above issues

Weighted prediction (WP)

- Issue 1:
  - When ref_pic_list_combination_flag is 0, WP parameters for both L0 and L1 indices are signalled similar as H.264 WP. However, at that case list L0 and L1 are identical, we don't need to signal the WP parameter for list L1.

- Issue 2:
  - When ref_pic_list_combination_flag is 1, the ref_pic_list_combination() process allows list LC to be smaller than the combination of list L0 and list L1. At that case, since WP parameters signalling in current text of explicit weighted prediction is not fully defined, some WP parameters may be missing entries for L0 and L1 indices.

- Issue 3:
  - On the decoding process in 8.5.2.2.3.2 (weighted sample prediction process), mapping array RefIdxLCToRefIdxLx[] is incorrect. Mapping array RefIdxLxToRefIdxLC[] must be defined in this process.

Proposal

- Fix on Issue 1:
  - Delete WP parameter signaling for list L1 in B-slices

Current CD

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 && ref_pic_list_combination_flag == 0)) {	
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(slice_type == B) {	
if(ref_pic_list_combination_flag == 0) {	
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)
}	
...	
}	
} else {	
for(i = 0; i <= num_ref_idx_lc_active_minus1; i++) {	
luma_weight_lc_flag	u(1)
if(luma_weight_l1_flag) {	
delta_luma_weight_lc[i]	se(v)
luma_offset_lc[i]	se(v)
}	
if(chroma_format_idc != 0) {	
chroma_weight_lc_flag	u(1)
...	
}	
}	
}	
}	
}	

Proposal

| 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) {	
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(slice_type == B) {	
for(i = 0; i <= num_ref_idx_lc_active_minus1; i++) {	
luma_weight_lc_flag	u(1)
if(luma_weight_l1_flag) {	
delta_luma_weight_lc[i]	se(v)
luma_offset_lc[i]	se(v)
}	
...	
}	
}	
}	

If num_ref_idx_lc_active_minus1 is not present, then this value should be inferred to (num_ref_idx_l0_active_minus1) // L0 = L1 Where List l0 and l1 are identical at this case.

Pred_weight_table syntax is simplified considerably.

Proposal

- Fix on Issue 2:
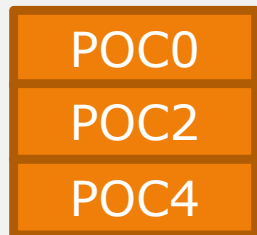 - Apply to default WP parameter for missing entries (L0 and L1)

Default WP parameter:

$$w_{0C} = (2 << logWDc)$$
$$o_{0C} = 0$$

Random Access
(M=4)

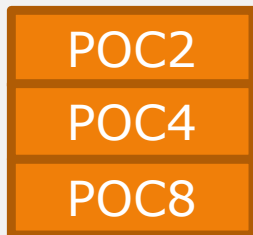| 0 | 1 | 2 | 3 | 4 |
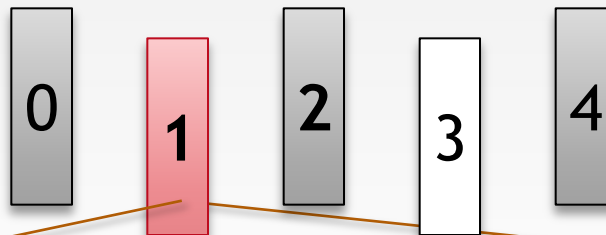
When Num_ref_idx for LC is same entries as ones for L0 and L1

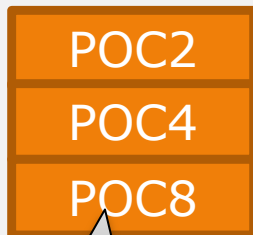| *List 0*	*List 1*	*List LC*
POC0	POC2	POC2
POC2	POC4	POC4
POC4	POC8	POC8
num_ref_idx_l0=3	num_ref_idx_l1=3	num_ref_idx_lc=3

No problem

When Num_ref_idx for LC is less entries than ones for L0 and L1
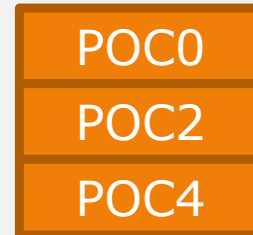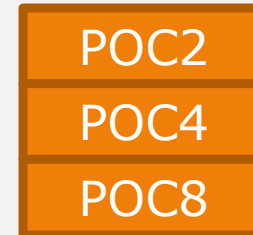
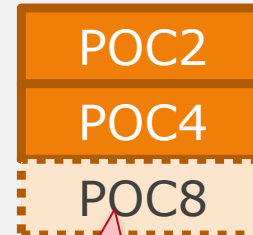| *List 0*	*List 1*	*List LC*
POC0	POC2	POC2
POC2	POC4	POC4
POC4	POC8	POC8
num_ref_idx_l0=3	num_ref_idx_l1=3	num_ref_idx_lc=2

Missing entry of WP param.

By applying to default WP parameter for missing entries, the above issue is simply fixed

Proposal

- Fix on Issue 3:
  - Define mapping array RefIdxLxToRefIdxLC in "Mapping process for reference picture lists combination in B slices" and modify weighted prediction sample process as follows:

Current CD

– If C is equal to L for luma samples,

$$logWDc = luma\_log2\_weight\_denom+ shift1 \qquad (8\text{-}232)$$

$$w_{0C} = LumaWeightLC[\ RefIdx\mathbf{LC}ToRefIdx\mathbf{Lx}[\ refIdxL0\ ]\ ] \qquad (8\text{-}233)$$

$$w_{1C} = LumaWeightLC[\ RefIdx\mathbf{LC}ToRefIdx\mathbf{Lx}[\ refIdxL1\ ]\ ] \qquad (8\text{-}234)$$

$$o_{0C} = luma\_offset\_lc[\ RefIdx\mathbf{LC}ToRefIdx\mathbf{Lx}[\ refIdxL0\ ]\ ] * (\ 1 << (\ BitDepth_Y - 8\ )\ ) \qquad (8\text{-}235)$$

$$o_{1C} = luma\_offset\_lc[\ RefIdx\mathbf{LC}ToRefIdx\mathbf{Lx}[\ refIdxL1\ ]\ ] * (\ 1 << (\ BitDepth_Y - 8\ )\ ) \qquad (8\text{-}236)$$

Proposal

– If C is equal to L for luma samples,

$$logWDc = luma\_log2\_weight\_denom+ shift1 \qquad (8\text{-}232)$$

$$w_{0C} = LumaWeightLC[\ RefIdx\mathbf{L0}ToRefIdx\mathbf{LC}[\ refIdxL0\ ]\ ] \qquad (8\text{-}233)$$

$$w_{1C} = LumaWeightLC[\ RefIdx\mathbf{L1}ToRefIdx\mathbf{LC}[\ refIdxL1\ ]\ ] \qquad (8\text{-}234)$$

$$o_{0C} = luma\_offset\_lc[\ RefIdx\mathbf{L0}ToRefIdx\mathbf{LC}[\ refIdxL0\ ]\ ] * (\ 1 << (\ BitDepth_Y - 8\ )\ ) \qquad (8\text{-}235)$$

$$o_{1C} = luma\_offset\_lc[\ RefIdx\mathbf{L1}ToRefIdx\mathbf{LC}[\ refIdxL1\ ]\ ] * (\ 1 << (\ BitDepth_Y - 8\ )\ ) \qquad (8\text{-}236)$$

Conclusion

- Proposal
  - To remove redundant representation of WP signalling when ref_pic_list_combination_flag is 0, simplified syntax is presented
  - To cope with missing entries of WP parameter when num_ref_idx for LC is less than those for L0 and L1, the default WP parameter is defined
  - To fix the editorial issue of mapping array, decoding processes are modified

- Suggestion;
  - To solve these issues on WP, the best way should be selected.