

JCTVC-K0123: AHG9: Reference picture set clean-ups

Adarsh K. Ramasubramonian and
Ye-Kui Wang(Qualcomm)

Summary

1. Addition of restrictions such that each LTRP signalled (explicitly or indexed) in the slice header shall be a distinct reference picture
2. Removal of the restriction that POC LSBs for LTRPs are signalled in a non-increasing order

1. LTRP should not be repeated in slice header

- Same LTRP may be signalled (either indexed or explicitly) in the slice header
- The `used_by_curr_pic_lt_flag` could be signalled both zero and one for different instances
- **Solution:** Ensure that the LTRP is not signalled in as both used and not used by the current picture. (following slide)

Changes to decoding process - 1

For each i in the range of 0 to $\text{NumPocLtCurr} - 1$, inclusive, when $\text{CurrDeltaPocMsbPresentFlag}[i]$ is equal to 1, it is a requirement of bitstream conformance that the following conditions apply:

- There shall be no j in the range of 0 to $\text{NumPocStCurrBefore} - 1$, inclusive, for which $\text{PocLtCurr}[i]$ is equal to $\text{PocStCurrBefore}[j]$.
- There shall be no j in the range of 0 to $\text{NumPocStCurrAfter} - 1$, inclusive, for which $\text{PocLtCurr}[i]$ is equal to $\text{PocStCurrAfter}[j]$.
- There shall be no j in the range of 0 to $\text{NumPocStFoll} - 1$, inclusive, for which $\text{PocLtCurr}[i]$ is equal to $\text{PocStFoll}[j]$.
- There shall be no j in the range of 0 to $\text{NumPocLtCurr} - 1$, inclusive, where j is not equal to i , for which $\text{PocLtCurr}[i]$ is equal to $\text{PocLtCurr}[j]$.

For each i in the range of 0 to $\text{NumPocLtFoll} - 1$, inclusive, when $\text{FollDeltaPocMsbPresentFlag}[i]$ is equal to 1, it is a requirement of bitstream conformance that the following conditions apply:

- There shall be no j in the range of 0 to $\text{NumPocStCurrBefore} - 1$, inclusive, for which $\text{PocLtFoll}[i]$ is equal to $\text{PocStCurrBefore}[j]$.
- There shall be no j in the range of 0 to $\text{NumPocStCurrAfter} - 1$, inclusive, for which $\text{PocLtFoll}[i]$ is equal to $\text{PocStCurrAfter}[j]$.
- There shall be no j in the range of 0 to $\text{NumPocStFoll} - 1$, inclusive, for which $\text{PocLtFoll}[i]$ is equal to $\text{PocStFoll}[j]$.
- There shall be no j in the range of 0 to $\text{NumPocLtFoll} - 1$, inclusive, where j is not equal to i , for which $\text{PocLtFoll}[i]$ is equal to $\text{PocLtFoll}[j]$.
- There shall be no j in the range of 0 to $\text{NumPocLtCurr} - 1$, inclusive, for which $\text{PocLtFoll}[i]$ is equal to $\text{PocLtCurr}[j]$.

Changes to decoding process - 2

For each i in the range of 0 to $\text{NumPocLtCurr} - 1$, inclusive, when $\text{CurrDeltaPocMsbPresentFlag}[i]$ is equal to 0, it is a requirement of bitstream conformance that the following conditions apply:

- There shall be no j in the range of 0 to $\text{NumPocStCurrBefore} - 1$, inclusive, for which $\text{PocLtCurr}[i]$ is equal to $(\text{PocStCurrBefore}[j] \& (\text{MaxPicOrderCntLsb} - 1))$.
- There shall be no j in the range of 0 to $\text{NumPocStCurrAfter} - 1$, inclusive, for which $\text{PocLtCurr}[i]$ is equal to $(\text{PocStCurrAfter}[j] \& (\text{MaxPicOrderCntLsb} - 1))$.
- There shall be no j in the range of 0 to $\text{NumPocStFoll} - 1$, inclusive, for which $\text{PocLtCurr}[i]$ is equal to $(\text{PocStFoll}[j] \& (\text{MaxPicOrderCntLsb} - 1))$.
- There shall be no j in the range of 0 to $\text{NumPocLtCurr} - 1$, inclusive, where j is not equal to i , for which $\text{PocLtCurr}[i]$ is equal to $(\text{PocLtCurr}[j] \& (\text{MaxPicOrderCntLsb} - 1))$.

For each i in the range of 0 to $\text{NumPocLtFoll} - 1$, inclusive, when $\text{FollDeltaPocMsbPresentFlag}[i]$ is equal to 0, it is a requirement of bitstream conformance that the following conditions apply:

- There shall be no j in the range of 0 to $\text{NumPocStCurrBefore} - 1$, inclusive, for which $\text{PocLtFoll}[i]$ is equal to $(\text{PocStCurrBefore}[j] \& (\text{MaxPicOrderCntLsb} - 1))$.
- There shall be no j in the range of 0 to $\text{NumPocStCurrAfter} - 1$, inclusive, for which $\text{PocLtFoll}[i]$ is equal to $(\text{PocStCurrAfter}[j] \& (\text{MaxPicOrderCntLsb} - 1))$.
- There shall be no j in the range of 0 to $\text{NumPocStFoll} - 1$, inclusive, for which $\text{PocLtFoll}[i]$ is equal to $(\text{PocStFoll}[j] \& (\text{MaxPicOrderCntLsb} - 1))$.
- There shall be no j in the range of 0 to $\text{NumPocLtFoll} - 1$, inclusive, where j is not equal to i , for which $\text{PocLtFoll}[i]$ is equal to $(\text{PocLtFoll}[j] \& (\text{MaxPicOrderCntLsb} - 1))$.
- There shall be no j in the range of 0 to $\text{NumPocLtCurr} - 1$, inclusive, for which $\text{PocLtFoll}[i]$ is equal to $(\text{PocLtCurr}[j] \& (\text{MaxPicOrderCntLsb} - 1))$.

Remove non-increasing order of POC LSBs

- POC LSBs have non-increasing restriction
 - Useful when LSBs were signalled $ue(v)$
 - Disallows optimal order that may avoid RPLM signalling
 - Less efficient
- Removal of non-increasing constraint
 - Remove the constraint in semantics of `poc_lsb_lt[i]` and `lt_idx_sps[i]`
 - Remove one condition in derivation of `DeltaPocMsbCycleLt[i]`

Example – current signalling

POC = 308
 LSB = 20
 MSB cycle = 9

LTRP[0]

POC = 84
 LSB = 20
 MSB cycle = 2

LTRP[1]

POC = 170
 LSB = 10
 MSB cycle = 5

LTRP[2]

POC = 311

current picture

DeltaPocMSBCycleLt[i]	1
poc_lsb_lt[i]	20
delta_poc_msb_cycle_lt[i]	1
# of bits for MSB cycle	3

DeltaPocMSBCycleLt[i]	8
poc_lsb_lt[i]	20
delta_poc_msb_cycle_lt[i]	7
# of bits for MSB cycle	7

DeltaPocMSBCycleLt[i]	5
poc_lsb_lt[i]	10
delta_poc_msb_cycle_lt[i]	5
# of bits for MSB cycle	5

Total bits for
 MSB cycle = 15

poc_lsb_lt[i] specifies the value of 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 the poc_lsb_lt[i] syntax element is $\log_2 \text{max_pic_order_cnt_lsb_minus4} + 4$ bits. For any values of j and k in the range of num_long_term_sps to num_long_term_sps + num_long_term_pics - 1, inclusive, if j is less than k, poc_lsb_lt[j] shall not be less than poc_lsb_lt[k].

```

if( i == 0 || i == num_long_term_sps || PocLsbLt[ i - 1 ] != PocLsbLt[ i ] )
  DeltaPocMSBCycleLt[ i ] = delta_poc_msb_cycle_lt[ i ]          (7-37)
else
  DeltaPocMSBCycleLt[ i ] = delta_poc_msb_cycle_lt[ i ] + DeltaPocMSBCycleLt[ i - 1 ]
  
```

Example – proposed signalling

POC = 308
 LSB = 20
 MSB cycle = 9

LTRP[0]

POC = 170
 LSB = 10
 MSB cycle = 5

LTRP[1]

POC = 84
 LSB = 20
 MSB cycle = 2

LTRP[2]

POC = 311

current picture

DeltaPocMSBCycleLt[i]	1
poc_lsb_lt[i]	20
delta_poc_msb_cycle_lt[i]	1
# of bits for MSB cycle	3

DeltaPocMSBCycleLt[i]	5
poc_lsb_lt[i]	10
delta_poc_msb_cycle_lt[i]	4
# of bits for MSB cycle	5

DeltaPocMSBCycleLt[i]	8
poc_lsb_lt[i]	20
delta_poc_msb_cycle_lt[i]	3
# of bits for MSB cycle	5

Total bits for
 MSB cycle = 13

poc_lsb_lt[i] specifies the value of 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 the poc_lsb_lt[i] syntax element is $\log_2 \text{max_pic_order_cnt_lsb_minus4} + 4$ bits. ~~For any values of j and k in the range of num_long_term_sps to num_long_term_sps + num_long_term_pics - 1, inclusive, if j is less than k, poc_lsb_lt[j] shall not be less than poc_lsb_lt[k].~~

```

if( i == 0 || i == num_long_term_sps || PocLsbLt[ i - 1 ] != PocLsbLt[ i ] )
    DeltaPocMSBCycleLt[ i ] = delta_poc_msb_cycle_lt[ i ]           (7-37)
else
    DeltaPocMSBCycleLt[ i ] = delta_poc_msb_cycle_lt[ i ] + DeltaPocMSBCycleLt[ i - 1 ]
    
```