|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  36th Meeting: Gothenburg, SE, 6–12 July 2019 | Document: JCTVC-AJ0026 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **On HRD for splicing** | | |
| *Status:* | Input document to JVET | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Yago Sanchez  Robert Skupin  Karsten Sühring Thomas Schierl | Email: | {firstname.lastname}@hhi.fraunhofer.de |
| *Source:* | Fraunhofer HHI | | |

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

# Abstract

This contribution provides details of an asserted bug of the Hypothetical Reference Decoder (HRD) model of VVC and HEVC, specifically when splicing. The fist aspect of this contribution proposes a bug-fix for the reported issue. In the second aspect it is proposed to add a flag to Picture Timing SEI message that allows a splicer to easily identify pictures within the bitstream, at which seamless splicing can be achieved.

# Introduction

Note: Document JVET-O0496 reports the same issue for VVC and proposes the same bug fix as the first aspect of this contribution.

The current VVC draft, as well as the HEVC specification, determines, when concatenationFlag is set to 1, that the AuNominalRemovalTime for the splicing point under some circumstances (non-seamless splicing) may be higher than   
AuNominalRemovalTime[ prevNonDiscardablePic ]+ ( auCpbRemovalDelayDeltaMinus1 + 1 ).

The text in HEVC specifies:

if( !concatenationFlag ) {  
 baseTime = AuNominalRemovalTime[ firstPicInPrevBuffPeriod ]  
 tmpCpbRemovalDelay = AuCpbRemovalDelayVal  
} else {  
 baseTime = AuNominalRemovalTime[ prevNonDiscardablePic ]  
 tmpCpbRemovalDelay =  
 Max( ( auCpbRemovalDelayDeltaMinus1 + 1 ), (C‑10)  
 Ceil( ( InitCpbRemovalDelay[ SchedSelIdx ] ÷ 90000 +  
 AuFinalArrivalTime[ n − 1 ] − AuNominalRemovalTime[ n − 1 ] ) ÷ ClockTick ) )  
}  
AuNominalRemovalTime[ n ] = baseTime + ClockTick \* ( tmpCpbRemovalDelay − CpbDelayOffset )

In the following figure, it is shown a splicing operation that does not allow for seamless splicing due to the fact that the InitCpbRemovalDelay[ SchedSelIdx ] imposes a later removal time after splicing than equidistant CPB removal times as present in seamless playout.



If instead of deriving the removal time of AU 6 from the AU n-1 (i.e. AU 5) as indicated above in the figure, we use the last non-discardable picture (AU 4) it would result in the following:

taf6= trm4 + >= trm6 (CPB underflow resulting in non-seamless splicing)

trm6= taf5+InitCpbRemovalDelay

=taf6 – trm4= Size/Bitrate – (trm4 – taf5)

trm4+Size/Bitrate – (trm4 – taf5)>= taf5+InitCpbRemovalDelay

Size/Bitrate – (trm4 – taf5)>= (taf5 – trm4)+InitCpbRemovalDelay

taf5+ Size/Bitrate = **taf6 >= trm4 + (taf5 – trm4)+InitCpbRemovalDelay**

As seen in the derived formula, the part marked in yellow in C-10 above is not correct. Therefore, a fix is proposed in section 2.1.

Note: The contribution in JVET-O0496 also proposes additional signaling for identification of splicing points that lead to a seamless splicing operation as explained below.

In addition, in order for a splicer to identify access units in the original bitstream that qualify as a seamless splicing point, the splicer needs to keep track of taf5, trm5, trmX, where X is the prevNonDiscardablePic. It is asserted that it would be desirable for a splicer to be able to make sure that a splicing point is a seamless splicing point in a simpler manner. A solution is proposed in section 2.2 to ease the determination of a seamless splicing points.

# Proposal

## Bug-fix

The proposed fix is as follows:

if( !concatenationFlag ) {  
 baseTime = AuNominalRemovalTime[ firstPicInPrevBuffPeriod ]  
 tmpCpbRemovalDelay = AuCpbRemovalDelayVal  
} else {  
 baseTime1 = AuNominalRemovalTime[ prevNonDiscardablePic ]  
 tmpCpbRemovalDelay1 = ( auCpbRemovalDelayDeltaMinus1 + 1 )  
 baseTime2 = AuNominalRemovalTime[ n − 1 ]  
 tmpCpbRemovalDelay2 =   
 Ceil( ( InitCpbRemovalDelay[ SchedSelIdx ] ÷ 90000 +  
 AuFinalArrivalTime[ n − 1 ] − AuNominalRemovalTime[ n − 1 ] ) ÷ ClockTick )   
 (C‑10)  
 if( baseTime1 + ClockTick \* tmpCpbRemovalDelay1 <   
 baseTime2 + ClockTick \* tmpCpbRemovalDelay2 ) {  
 baseTime = baseTime2  
 tmpCpbRemovalDelay = tmpCpbRemovalDelay2  
 } else {  
 baseTime = baseTime1  
 tmpCpbRemovalDelay = tmpCpbRemovalDelay1  
 }  
}  
AuNominalRemovalTime[ n ] = baseTime + ClockTick \* ( tmpCpbRemovalDelay − CpbDelayOffset )

## Identification of seamless splicing points

In addition, this contribution proposes adding some signalling to enable an splicer to identify whether an AU fullfils the following condition and splicing right after the current AU leads to a seamless splicing operation. In order to determine whether the AU can be used for splicing the time difference between the removal time and arrival time needs to be higher than a given threshold.

AuNominalRemovalTime[ n − 1 ] − AuFinalArrivalTime[ n − 1 ] > Threshold

Where the threshold is:

Threshold = InitCpbRemovalDelay[ SchedSelIdx ] ÷ 90000 – 1/framerate

**Option 1:**

A flag in Picture Timing SEI message is proposed to ease the identification of seamless splicing points. In addition, the seamless splicing signaling is done for a Buffering period SEI message of the spliced bitstream that has a value of InitCpbRemovalDelay equal or smaller than that of the current Buffering period SEI message.

|  |  |
| --- | --- |
| pic\_timing( payloadSize ) { | Descriptor |
| … |  |
| if( CpbDpbDelaysPresentFlag ) { |  |
| **…** |  |
| **seamless\_spliceable\_flag** | u(1) |
| } |  |
| } |  |

**seamless\_spliceable\_flag** equal to 1 specifies that the difference of the final arrival time and CPB removal time access unit associated with the picture timing SEI message is such that seamless splicing is achieved when followed by an access unit with a Buffering Period SEI message with concatenation\_flag equal to 1 and InitCpbRemovalDelay equal or smaller than the value of InitCpbRemovalDelay of the Buffering period associated with the access unit associated with the picture timing SEI message. When seamless\_spliceable\_flag is equal to 0, the difference of the final arrival time and CPB removal time access unit associated with the picture timing SEI message can be any.

**Option 2:**

It is asserted that the solution proposed in Option 1 might be limited since InitCpbRemovalDelay might change within the bitstream between two Buffering Period SEI messages. Only InitCpbRemovalDelay + InitCpbRemovalDelayOffset  is constant.

In order to allow for a more flexible solution, in addition to the seamless splicing signalling of Option 1, a value is indicated in the Buffering Period SEI message, which might be different than the InitCpbRemovalDelay. This new value is used to compute the described threshold above. Thus, a bistream that has a higher InitCpbRemovalDelay than that of the current bitstream but smaller than max\_val\_initial\_removal\_delay\_for\_seamless\_splicing (see below) can be spliced together and seamless splicing can still be achieved.

|  |  |
| --- | --- |
| buffering\_period( payloadSize ) { | **Descriptor** |
| **…** |  |
| **concatenation\_flag** | u(1) |
| **cpb\_removal\_delay\_delta\_minus1** | u(v) |
| **cpb\_removal\_delay\_deltas\_present\_flag** | u(1) |
| if( cpb\_removal\_delay\_deltas\_present\_flag ) { |  |
| **num\_cpb\_removal\_delay\_deltas\_minus1** | ue(v) |
| for( i = 0; i <= num\_cpb\_removal\_delay\_deltas\_minus1; i++ ) |  |
| **cpb\_removal\_delay\_delta**[ i ] | u(v) |
| } |  |
| … |  |
| **max\_val\_initial\_removal\_delay\_for\_seamless\_splicing** | u (v) |
| … |  |
| } |  |

**max\_val\_initial\_removal\_delay\_for\_seamless\_splicing** specifies the maximum value of InitCpbRemovalDelays for which seamless\_spliceable\_flag in Picture Timing SEI message indicates that a seamless splicing is achieved.

The semantics of seamless\_spliceable\_flag in the picture timing SEI message as proposed in Option 1 are modified as follows:

**seamless\_spliceable\_flag** equal to 1 specifies that the difference of the final arrival time and CPB removal time access unit associated with the picture timing SEI message is such that seamless splicing is achieved when followed by an access unit with a Buffering Period SEI message with concatenation\_flag equal to 1 and InitCpbRemovalDelay equal or smaller than the value of max\_val\_initial\_removal\_delay\_for\_seamless\_splicing of the Buffering period associated with the access unit associated with the picture timing SEI message. When seamless\_spliceable\_flag I equal to 0, the difference of the final arrival time and CPB removal time access unit associated with the picture timing SEI message can be any.

# Patent rights declaration(s)

**Fraunhofer HHI 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).**