|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  8th Meeting: San José, CA, USA, 1–10 February, 2012 | Document: JCTVC-H0496r2 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **On bitstreams starting with CRA pictures** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Ye-Kui Wang Ying Chen Marta Karczewicz Jianle Chen  5775 Morehouse Dr San Diego, CA 92121 USA | Tel: Email: | 1-858-651-8345 [yekuiw@qualcomm.com](mailto:yekuiw@qualcomm.com)  1-858-845-6589 [cheny@qualcomm.com](mailto:cheny@qualcomm.com)  1-858-658-5673 [martak@qualcomm.com](mailto:martak@qualcomm.com)  1-858-651-8028 [cjianle@qualcomm.com](mailto:cjianle@qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

This document reproposes the proposal in JCTVC-G319.

In the current HEVC design, CRA (Clean Random Access) pictures are identified by a new NAL unit type. It is very common that a device with a conforming decoder performs random access at a CRA picture. However, a bitstream starting at a CRA picture is considered non-conforming, thus a conforming decoder may not be able to properly handle such bitstreams.

In this proposal, it is proposed that a bitstream staring from a CRA picture can be conforming. Such a conforming bitstream may or may not contain leading pictures associated with the CRA picture. A leading picture associated with a CRA picture is a coded picture that follows the CRA picture in decoding order but precedes the CRA picture in output order. The proposed normative changes include: 1) skipping the decoding and output of the leading pictures associated with the starting CRA picture, when present; 2) HRD modifications to guarantee that all bitstream conforming conditions are fulfilled by a conforming bitstream starting with a CRA picture, regardless of whether the leading pictures associated with the CRA picture are present.

Revision 1 of this document provides the required spec text changes need to enable bistreams starting with a CRA picture with leading pictures being present to be conforming. The text is included in an attachment to this document.

Revision 2 of this document provides an improved specification text changes needed to enable bistreams starting with a CRA picture with leading pictures being present to be conforming. The text is included in an attachment to this document. The improvement is about specification of the varaiables for the previous POC MSB values and the previous POC LSB values for the CRA picture that starts the bitstream, for derviation of the POC value. It was commented that SPS activation process would also need to be changed for bitstreams starting at a CRA picture, but it seemed that the SPS activation process in the latest WD is sufficient without changes.

# Introduction

The same proposal was proposed in JCTVC-G319. The following comments were noted in the meeting report of the previous JCT-VC meeting regarding the proposal:

“It was remarked that it may be difficult to specify conformance of a bitstream in such a case, as there would be implied dangling references to non-existing information. A bitstream needs to be testable for conformance (without access to unavailable data).

“The concept seems useful, if it is possible to specify the bitstream conformance clearly, and the decoder behaviour as well. Further study is encouraged.”

As can be seen, the main concern was how to clearly specify bitstream conformance, particular due to the implied dangling references to non-existing information.

We think that bitstream conformance can be clearly specified, as documented in the proposal below. In particular, the issue of dangling references to non-existing information can be avoid by ignoring the leading pictures.

# Background

A CRA picture is a coded picture containing only I slices and for which each slice has nal\_unit\_type equal to 4; all coded pictures that follow the CRA picture both in decoding order and output order shall not use inter prediction from any picture that precedes the CRA picture either in decoding order or output order; and any picture that precedes the CRA picture in decoding order also precedes the CRA picture in output order. All pictures following a CRA in output order can be correctly decoded without inter prediction from any picture that precedes the CRA picture either in decoding order or output order.

As shown in Figure 1, pictures with POC value equal to 8 and 16 can be CRA pictures. The pictures following a CRA picture in decoding order but preceding the CAR picture in output order are called the leading pictures of the CRA picture.

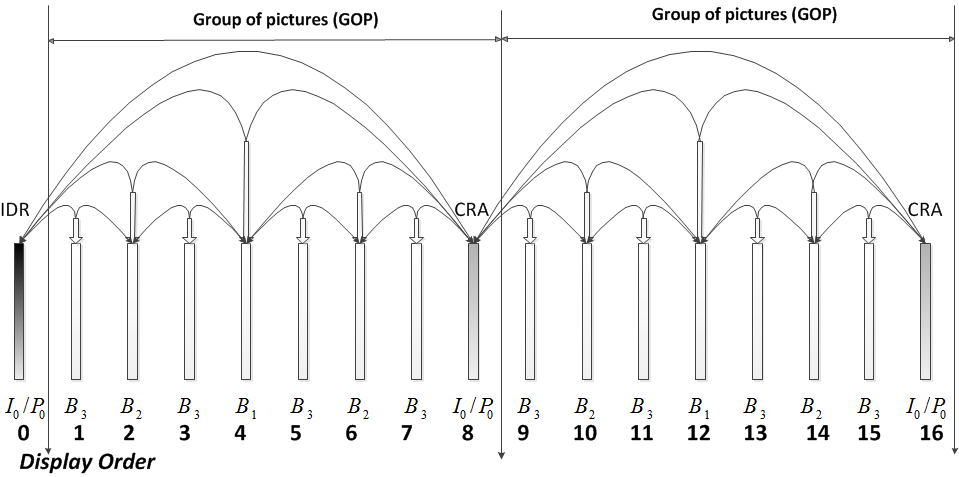


Figure 1: CRA pictures and leading pictures.

In broadcasting and streaming applications, random access is a very important feature and CRA pictures can be used to tradeoff the frequency of random access points and coding efficiency. For example, the recently finalized MPEG Dynamic Adaptive Streaming over HTTP (DASH) explicitly allows signaling of open-GOP random access points, which are effectively CRA pictures. To make a bitstream starting at a CRA picture conforming would be convenient for implementers of decoders, since the normative behaviors are clearly specified for random access from CRA pictures.

In video streaming, for random access or steaming adaptation (by bitstream switching) at a CRA picture, the client may choose not to request the leading pictures associated with the CRA picture. If not imitated by the client, the server may proactively choose not to transmit the leading pictures.

Though it is possible to rely on non-conforming decoders and systems to take advantage of the current CRA design as part of the optimization of a video system, supporting certain level of CRA related codec optimization in a normative way is desirable. The shortcoming that AVC does not support normative open-GOP random accessibility has been discussed during the development of the DASH standard. One potential problem associated with handling of bitstreams starting with an open-GOP random access point by conforming decoders, when the leading pictures are not present, is that the CPB may overflow, because the coded bits of pictures after the leading pictures in decoding order flow into the CPB earlier than when the leading pictures are not present, while the decoding times of these pictures are unchanged.

# Proposal

It is proposed that normative decoding processes are specified in HEVC, such that conforming decoders are guaranteed to be able to properly decode bitstreams starting with a CRA picture and such a bitstream is conforming.

## Definitions

***leading picture***: a *coded* *picture* associated with a CRA picture, that follows the CRA picture in decoding order but precedes the CRA picture in output order. A leading picture associated with a CRA picture has picture order count less than the picture order count of the CRA picture.

## Decoding processes changes

### Handling of leading pictures

If the bitstream starts with a CRA picture, the leading pictures associated with the CRA picture, i.e., the coded pictures with PicOrderCnt less than the PicOrderCnt value of the CRA picture, if present in the bitsteram, are ignored (removed from the bitstream and discarded).

## HRD changes

The HRD changes are summarized below:

* When the leading pictures associated with the CRA picture that starts the bitstream are not present, the CPB removal times of the coded pictures following the CRA in decoding order may need to be shifted earlier to guarantee no CPB overflow. For this purpose, a CPB remove time delay offset is added into the buffering period SEI message.
* The bitstream conformance text is changed, to allow a conforming bitstream to start with a CRA picture, and to require that all bitstream conforming conditions are fulfilled for such a conforming bitstream regardless of whether the leading pictures associated with the starting CRA picture are not present.

The changes are provided below, wherein the modified or added parts of the syntax table are highlighted.

### Buffering period SEI message syntax

|  |  |  |
| --- | --- | --- |
| buffering\_period( payloadSize ) { | **C** | **Descriptor** |
| **seq\_parameter\_set\_id** | 5 | ue(v) |
| **cra\_para\_present\_flag** | 5 | u(1) |
| if( NalHrdBpPresentFlag ) { |  |  |
| for( SchedSelIdx = 0; SchedSelIdx <= cpb\_cnt\_minus1; SchedSelIdx++ ) { |  |  |
| **initial\_cpb\_removal\_delay[** SchedSelIdx **]** | 5 | u(v) |
| **initial\_cpb\_removal\_delay\_offset[** SchedSelIdx **]** | 5 | u(v) |
| if ( cra\_para\_present\_flag ) |  |  |
| **random\_access\_removal\_delay\_offset[** SchedSelIdx **]** | 5 | u(v) |
| } |  |  |
| } |  |  |
| if( VclHrdBpPresentFlag ) { |  |  |
| for( SchedSelIdx = 0; SchedSelIdx <= cpb\_cnt\_minus1; SchedSelIdx++ ) { |  |  |
| **initial\_cpb\_removal\_delay[** SchedSelIdx **]** | 5 | u(v) |
| **initial\_cpb\_removal\_delay\_offset[** SchedSelIdx **]** | 5 | u(v) |
| if ( cra\_para\_present\_flag ) |  |  |
| **random\_access\_removal\_delay\_offset[** SchedSelIdx **]** | 5 | u(v) |
| } |  |  |
| } |  |  |
| } |  |  |

### Buffering period SEI message semantics

The semantics of the new syntax elements are as follows.

**cra\_para\_present\_flag** equal to 1 indicates the presence of the random\_access\_removal\_delay\_offset[ SchedSelIdx ] syntax element. cra\_para\_present\_flag equal to 0 indicates the absence of the random\_access\_removal\_delay\_offset[ SchedSelIdx ] syntax element. This flag shall not be 0 when the associated picture is not a CRA picture.

**random\_access\_removal\_delay\_offset**[ SchedSelIdx ] specifies a CPB removal time offset for the SchedSelIdx-th CPB. It is in units of a 90 kHz clock. random\_access\_removal\_delay\_offset[ SchedSelIdx ] shall not exceed initial\_cpb\_removal\_delay[ SchedSelIdx ]+ initial\_cpb\_removal\_delay\_offset[ SchedSelIdx ]. When not present, the value shall be inferred to be equal to 0.

### Operation of coded picture buffer (CPB)

The specifications in this subclause apply independently to each set of CPB parameters that is present and to both Type I and Type II conformance.

#### Timing of bitstream arrival

The HRD may be initialised at any one of the buffering period SEI messages. Prior to initialisation, the CPB is empty.

NOTE - After initialisation, the HRD is not initialised again by subsequent buffering period SEI messages.

The access unit that is associated with the buffering period SEI message that initializes the CPB is referred to as access unit 0. All other access units are referred to as access unit n with n being incremented by 1 for the next access unit in decoding order.

If the first access unit is a CRA access unit, and the leading pictures are not present and the cra\_para\_present\_flag is equal to 1, useUpdatePara is set to 1, otherwise, useUpdatePara is set to 0.

If useUpdatePara is equal to 1, DelayOffset[SchedSelIdx] is set to random\_access\_removal\_delay\_offset[ SchedSelIdx ], otherwise, DelayOffset[SchedSelIdx] is set to 0.

The time at which the first bit of access unit n begins to enter the CPB is referred to as the initial arrival time tai( n ).

The initial arrival time of access units is derived as follows.

* If the access unit is access unit 0, tai( 0 ) = 0,
* Otherwise (the access unit is access unit n with n > 0), the following applies.
* If cbr\_flag[ SchedSelIdx ] is equal to 1, the initial arrival time for access unit n, is equal to the final arrival time (which is derived below) of access unit n ‑ 1, i.e.

tai( n ) = taf( n – 1 ) (C-2)

* Otherwise, if cbr\_flag[ SchedSelIdx ] is equal to 0 and access unit n is not the first access unit of a subsequent buffering period, the initial arrival time for access unit n is derived by

tai( n ) = Max( taf( n – 1 ), tai,earliest( n ) ) (C-3)

where tai,earliest( n ) is given as follows

tai,earliest( n ) = tr,n( n ) –   
( initial\_cpb\_removal\_delay[ SchedSelIdx ] + initial\_cpb\_removal\_delay\_offset[ SchedSelIdx ] )  90000 (C-4)

with tr,n( n ) being the nominal removal time of access unit n from the CPB as specified in subclause C.1.2 and initial\_cpb\_removal\_delay[ SchedSelIdx ] and initial\_cpb\_removal\_delay\_offset[ SchedSelIdx ] being specified in the previous buffering period SEI message.

* Otherwise (cbr\_flag[ SchedSelIdx ] is equal to 0 and the subsequent access unit n is the first access unit of a subsequent buffering period), the initial arrival time for the access unit n is derived by

tai( n ) = tr,n( n ) – ( initial\_cpb\_removal\_delay[ SchedSelIdx ]  90000 ) (C-5)

with initial\_cpb\_removal\_delay[ SchedSelIdx ] being specified in the buffering period SEI message associated with access unit n.

The final arrival time for access unit n is derived by

taf( n ) = tai( n ) + b( n )  BitRate[ SchedSelIdx ] (C-6)

where b( n ) is the size in bits of access unit n, counting the bits of the Type I bitstream for Type I conformance or the bits of the Type II bitstream for Type II conformance.

The values of SchedSelIdx, BitRate[ SchedSelIdx ], and CpbSize[ SchedSelIdx ] are constrained as follows.

* If access unit n and access unit n - 1 are part of different coded video sequences and the content of the active sequence parameter sets of the two coded video sequences differ, the HSS may select a value SchedSelIdx1 of SchedSelIdx from among the values of SchedSelIdx provided for the coded video sequence containing access unit n that results in a BitRate[ SchedSelIdx1 ] or CpbSize[ SchedSelIdx1 ] for the second of the two coded video sequences (which contains access unit n – 1) that differs from the value of BitRate[ SchedSelIdx0 ] or CpbSize[ SchedSelIdx0 ] for the value SchedSelIdx0 of SchedSelIdx  that was in use for the coded video sequence containing access unit n - 1.
* Otherwise, the HSS continues to operate with the previous values of SchedSelIdx, BitRate[ SchedSelIdx ] and CpbSize[ SchedSelIdx ].

When the HSS selects values of BitRate[ SchedSelIdx ] or CpbSize[ SchedSelIdx ] that differ from those of the previous access unit, the following applies.

* the variable BitRate[ SchedSelIdx ] comes into effect at time tai( n )
* the variable CpbSize[ SchedSelIdx ] comes into effect as follows.
* If the new value of CpbSize[ SchedSelIdx ] exceeds the old CPB size, it comes into effect at time tai( n ),
* Otherwise, the new value of CpbSize[ SchedSelIdx ] comes into effect at the time tr( n ).

#### Timing of coded picture removal

It is assumed that the nominal CPB removal time and the CPB removal time of a coded picture are calculated immediately after the previous coded picture is removed from the CPB, or, for access unit 0, when the HRD is initialised.

For access unit 0, the nominal removal time of the access unit from the CPB is specified by

tr,n( 0 ) = initial\_cpb\_removal\_delay[ SchedSelIdx ] 90000 (C-7)

At the removal time of access unit 0, the variable nb is set equal to 0.

Immediately after the removal of access unit 0 from the CPB, tr,n ( 0 ) is set to equal to tr,n( 0 ) – **(**DelayOffset[SchedSelIdx] 90000 ).

NOTE – The effective CPB removal time of access unit 0 is not shifted, but for all pictures after access unit 0 in decoding order, the effective CPB removal time is shifted earlier by **(**DelayOffset[SchedSelIdx] 90000 ).

For the first access unit of a buffering period that does not initialise the HRD, the nominal removal time of the access unit from the CPB is specified by

tr,n( n ) = tr,n ( nb ) + tc \* cpb\_removal\_delay( n ) (C-8)

where tr,n( nb ) is the nominal removal time of the first picture of the previous buffering period and cpb\_removal\_delay( n ) is specified in the picture timing SEI message associated with access unit n.

When an access unit n is the first access unit of a buffering period that does not initialise the HRD, nb is set equal to n at the removal time of access unit n.

The nominal removal time tr,n(n) of an access unit n that is not the first access unit of a buffering period is given by

tr,n( n ) = tr,n( nb ) + tc \* cpb\_removal\_delay( n ) (C-9)

The removal time of access unit n is specified as follows.

* If low\_delay\_hrd\_flag is equal to 0 or tr,n( n ) >= taf( n ), the removal time of access unit n is specified by

tr( n ) = tr,n( n ) (C-10)

* Otherwise (low\_delay\_hrd\_flag is equal to 1 and tr,n( n ) < taf( n )), the removal time of access unit n is specified by

tr( n ) = tr,n( n ) + tc \* Ceil( ( taf( n ) - tr,n( n ) ) tc ) (C-11)

NOTE – The latter case indicates that the size access unit n, b(n), is so large that it prevents removal at the nominal removal time.

### Bitstream conformance

The first coded picture, in decoding order, of a conforming bitstream may be an IDR picture or a CRA picture. For a conforming bitstream starting with a CRA picture, the bitstream subset generated by discarding the access units that contain any leading picture associated with the starting CRA picture, when present, shall still be a conforming bitstream.

# 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).**