|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  7th Meeting: Geneva, CH, 21-30 November, 2011 | Document: JCTVC-G159 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Random Access Detection and Notification** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Hendry Seungwook Park  Yongjoon Jeon Byeongmoon Jeon  #221 Yangjae-dong, Seocho-gu,  Seoul 137-130,  Korea | Tel: Email: | +82-10-7738-9288 hendry.hendry@lge.com |
| *Source:* | LG Electronics | | |

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

# Abstract

Leading pictures that follow a Clean Random Access (CRA) picture in decoding order can still use reference pictures from previous GOP as their reference. Thus, when random access event occurs or when decoder is suddenly fed with an encoded stream started from a CRA NAL unit, the decoder should be able to know it that avoid decoding leading pictures that follow the CRA. However, current WD 4 of HEVC does not have any mechanism for decoder to detect or to be aware of random access event. This contribution proposes two mechanisms for random access detection and one mechanism to enable any application (e.g., video player application) to notify HEVC decoder when random access occurs.

# Introduction

Random access detection feature in decoder is an open issue. As illustrated in Figure 1, in normal condition, decoding of leading pictures shall have no problem since all their reference pictures are available in buffer. However, when random access occurs, in which decoding of some encoded pictures are skipped and decoder starts again at CRA pictures, then decoding of the leading pictures after that CRA picture shall have reference picture mismatches.

1. Buffer condition when normal play
2. Buffer condition when random access occurs

Figure 1 – Problem with decoding leading pictures when random access occurs

From discussions until the 6th JCTVC meeting, it seems like there is a common agreement that when random access event occurs, leading pictures that follow the CRA picture, which is used as the starting point for decoding after the random access event, shall not be decoded. However, it is not clear yet whether decoder should be able to actively detect the random access event or just passively rely on a system to indicate the event.

In this contribution, we offer three options than can cover both active random access detection and passive random access notification.

# Random Access Detection

Contribution JCTVC-E400 [2] and JCTVC-F604 [3] have proposed schemes to enable decoder to actively detect random access event. Both schemes proposed to signal POC of a key picture, so called “*brother picture*”, which is the closest key picture that precedes a CRA picture in decoding and display order, in the slice header of CRA picture. In addition to brother picture concept, [3] also add picture discontinuity count to tackle an issue that might occur due to the fact that frame\_num can wrap-around and POC can be reset.

The idea of signaling brother picture for random access detection seems very simple and can be effective. However, in our opinion, signaling brother picture only might have issue when the brother picture is lost. Decoder might misinterpret the missing of brother picture as a random event and skip decoding all the leading pictures while in fact it is only an event of picture loss which can be treated with error concealment.

## Option 1: RA Detection with Explicit Reference Picture Signaling in CRA Slice Header

Random access detection mechanism by signaling POC of reference picture that should be in the Decoded Picture Buffer (DPB) can be improved to deal with picture loss issue by simply sending not only a brother picture but all existing reference pictures in the DPB. In fact, this idea can be very well align with the current work in AHG21 which is working on the idea of signaling reference picture explicitly in slice header / PPS.

If the Explicit Reference Picture Signaling (ERPS) work in AHG21 is completed and adopted, we propose to extend its functionality to be used also as random access detection. This extended functionality can be simply described in Hypothetical Reference Decoder section.

Random access detection method by using ERPS in CRA slice is as the following: “*prior to decoding a CRA frame, decoder shall check if all reference frames whose POCs are signalled / listed in reference picture set, if all the reference frames are available, then the CRA frame is to be decoded in normal event, otherwise, random access event occurs and decoder shall discard frames that follow the CRA and whose POC is lower than that of the CRA*”

## Option 2: RA Detection with Reference Pictures Synchronization SEI

Reference picture synchronization SEI describes the list of reference pictures that should exists in DPB and marked as “used for reference” when a target frame, which is described in the SEI message, is received. The idea of reference picture synchronization SEI is the similar to ERPS concept except that since it is an SEI message, it does not enforce encoder to signal it. The SEI can be signaled at certain point / interval to ensure that the list of reference pictures in DPB is as expected to guarantee correct decoding process. This would provide error resilience as well as random access detection features.

If the ERPS concept, which is being developed in AHG21, is not adopted, we propose that the JCTVC consider the adoption of reference picture synchronization SEI. Table 1 tabulates the syntax of the proposed reference pictures synchronization SEI and the semantics

Table 1 – Syntax reference picture synchronization SEI message

|  |  |
| --- | --- |
| ref\_pic\_synch ( payloadSize ) { | **Descriptor** |
| **target\_frame\_poc** | ue(v) |
| **number\_of\_ref\_frame** | ue(v) |
| for( i = 0; i < number\_of\_ref\_pic\_minus1 + 1; i++ ) { |  |
| **sign\_bit\_of\_ref\_frame[i]** | f(1) |
| **abs\_ref\_frame[ i ]** | ue(v) |
| } |  |
| } |  |

**target\_frame\_poc** describes the POC of picture to which this SEI is applied.

**number\_of\_ref\_frame** indicates the number of reference frames explicitly signaled.

**sign\_bit\_of\_ ref\_frame[i]** describes the sign bit of the ith reference frame. 1 indicates positive sign while 0 indicates otherwise.

**abs\_ref\_frame[i]** indicates the relative POC of the ith reference frames.

Random access detection method by using reference picture synchronization SEI message is as the following: “after *decoding a CRA frame which is associated with a reference picture synchronization SEI, if all reference frames whose POCs are signalled in the SEI message exist in the DPB, then the CRA frame is to be decoded in normal event, otherwise, random access event occurs and decoder shall discard frames that follow the CRA and whose POC is lower than that of the CRA*”

# Option 3: Random Access Notification

When random access occurs, a system that runs on top of decoder knows and might notify the decoder. However, there is no standardized mechanism of how a system can notify decoder about the occurrence of random access.

One might argue that when random access occurs, the system can just simply remove all leading pictures and not feed them to decoder. However, while in certain case, it might be possible for the system to filter out leading pictures in random access event, in general, it is not easy for the system to recognize a leading picture because there is no information about it in NAL unit header. Thus, we cannot rely on the system to filter out leading pictures when random access occurs. Instead, we think that it would be more elegant to provide a standardized mechanism for the system to notify decoder when random access occurs and let the decoder does the work of discarding the unnecessary leading pictures.

In the 6th JCTVC meeting, the following changes to NAL unit header were agreed:

* Reduce bits assigned for signaling nal\_ref\_idc from 2 bits to 1 bit
* Change the name of nal\_ref\_idc to nal\_ref\_flag
* Assign the 1 bit taken from nal\_ref\_idc to nal\_unit\_type.

We propose that the 1 bit taken from nal\_ref\_idc is not given to nal\_unit\_type, but instead, it should be given to a new syntax element called NAL dependent flag (i.e., nal\_dpd\_flag).

The following table tabulates NAL unit header syntax after addition of the proposed new nal\_dpd\_flag.

|  |  |
| --- | --- |
| nal\_unit( NumBytesInNALunit ) { | Descriptor |
| **forbidden\_zero\_bit** | f(1) |
| **nal\_ref\_flag** | f(1) |
| **nal\_dpd\_flag** | f(1) |
| **nal\_unit\_type** | u(5) |
| NumBytesInRBSP = 0 |  |
| nalUnitHeaderBytes = 1 |  |
| if( nal\_unit\_type = = 1 | | nal\_unit\_type = = 4 | | nal\_unit\_type = = 5 ) { |  |
| **temporal\_id** | u(3) |
| **output\_flag** | u(1) |
| **reserved\_one\_4bits** | u(4) |
| nalUnitHeaderBytes += 1 |  |
| } |  |
| for( i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++ ) { |  |
| if( i + 2 < NumBytesInNALunit && next\_bits( 24 ) = = 0x000003 ) { |  |
| **rbsp\_byte[** NumBytesInRBSP++ **]** | b(8) |
| **rbsp\_byte[** NumBytesInRBSP++ **]** | b(8) |
| i += 2 |  |
| **emulation\_prevention\_three\_byte** /\* equal to 0x03 \*/ | f(8) |
| } else |  |
| **rbsp\_byte[** NumBytesInRBSP++ **]** | b(8) |
| } |  |
| } |  |

The semantic of nal\_dpd\_flag is as follows:

**nal\_dpd\_flag** describes specific event that are associated with the nal\_unit\_type. The value of nal\_dpd\_flag is always set to zero when the bitstream is created. When the value of nal\_dpd\_flag is one, depending on the nal\_unit\_type, the following event has occurs:

* If nal\_unit\_type == 4, then random access to the current NAL unit has occurred.

It is worth to note that the proposed new flag currently has defined event that is associated with nal\_type\_type 4. In the future, more events that are associated with other types of NAL units can be defined / added to the proposed semantic of nal\_dpd\_flag.

The proposed new element changes NAL unit header structure is as the following:

Figure 1 – Proposed NAL unit header structure

Random Access Notification Mechanism (Informative)

At encoding time, nal\_dpd\_flag of all NAL units is set to zero. When random access occurs (e.g., user do fast forwarding during consuming the video), video player takes the following steps:

* Find the next NAL unit whose nal\_unit\_type == 4. Any NAL units in the bitstreams before that NAL unit shall be omitted.
* Set the value of nal\_dpd\_flag of the NAL unit to one.
* Feed the NAL unit to decoder

# Conclusion

Being able to know that random access has occurred is important for decoder so that it can skip decoding and outputting leading pictures. However, in current HEVC specification, there is no specified mechanism that allows decoder to do so. We propose two schemes that allow decoder to actively detect whether random access occurs and one scheme that allow other entities (e.g., video player) to notify decoder when random access occurs.

We would like to recommend that the JCTVC considers the three schemes and adopts at least one of them to complete decoder capability to deal with random access event.

# Patent rights declaration(s)

**LG Electronics 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).**

# Reference

1. JCTVC-F803\_d5, “WD4: Working Draft 4 of High-Efficiency Video Coding,” 6th JCT-VC Meeting, 6th Meeting: Torino, IT, 14-22 July, 2011.
2. JCTVC-E400, “Comments on Clean Decoding Refresh Pictures,” 5th JCT-VC Meeting, 5th Meeting: Geneva, CH, 16-23 March, 2011.
3. JCTVC-F604, “Detection of a CDR for random access,” 6th JCT-VC Meeting, 6th Meeting: Torino, IT, 14-22 July, 2011.