|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  11th Meeting: Shanghai, CN, 10–19 Oct. 2012 | Document: JCTVC-K0143 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **On handling of minimum POC range at decoder** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Kimihiko Kazui, Junpei Koyama, Satoshi Shimada and Akira Nakagawa  4-1-1 Kamikodanaka, Nakahara-ku 211-8588 Kawasaki JAPAN | Tel: Email: | +81-44-754-2639  kazui.kimihiko@jp.fujitsu.com koyama.junpei@jp.fujitsu.com  sshimada@jp.fujitsu.com anaka@jp.fujitsu.com |
| *Source:* | FUJITSU LABORATORIES LTD. | | |

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

# Abstract

This contribution provides the text which describes how a decoder can handle a POC value without having 32-bit range limit, according to the resolution at the 10th JCT-VC meeting. It is also proposed to extend the range of the POC value to 64-bit as once decided at the 9th JCT-VC meeting.

# Introduction

In JCTVC-I0045 "Cyclic POC"[4] at the 9th JCT-VC meeting it was proposed to modify the definition of the 32-bit range POC value so that it becomes cyclic. A low delay video coding apparatus using the intra-slice scheme normally encodes all pictures except the first picture in inter mode. If the cyclic property is not introduced to the POC value, the apparatus is required to insert an IDR picture when the POC value reaches to the maximum value (i.e. 231-1). This situation happens when the apparatus operates continuously more than 1 year. The insertion of an IDR picture in the middle of the bitstream becomes a burden to a manufacturer in terms of product designing and validation because such insertion is quite rare. The intention of JCTVC-I0045 was to remove such burden.

The resolution at the 9th JCT-VC meeting was to extend the range of the POC value from 32-bit to 64-bit instead of introducing the cyclic property to the POC value. This extension practically makes the range of POC value infinite. As the result, the burden to the apparatus can be removed. The rationale of the 64-bit range extension is that the current restriction on the difference between two POC values[[1]](#footnote-1) allows a decoder to handle the POC value with smaller range limit than what is specified in the standard.

However, it was pointed out that the current specification does not contain the concrete description on how a decoder can handle the POC value with smaller range than what is specified at the tenth JCT-VC meeting. As the result, it was concluded to revert the range of the POC value from 64-bit to 32-bit. It was also concluded to revert again to 64-bit if such concrete description is provided [3].

The following sentence is excerpted from JCTVC-J1000 [4].

|  |
| --- |
| If adequate text is provided to describe how a decoder can handle POC without having such a range limit, we can review the description of that scheme and consider including it in the standard and removing (or increasing) the 32 bit range limit. That aspect is for further study |

This contribution provides the text which describes how a decoder can handle a POC value without having 32-bit range limit, according to the resolution at the 10th JCT-VC meeting. It is also proposed to extend the range of the POC value to 64-bit as once decided at the 9th JCT-VC meeting.

# Proposed text

*Replace subclause 8.3.1 in [3] by:*

Output of this process is PicOrderCntVal, the picture order count of the current picture.

Picture order counts are used to identify pictures, for deriving motion parameters in merge mode and motion vector prediction, to represent picture order differences between pictures for motion vector derivation, and for decoder conformance checking (see subclause C.5).

Each coded picture is associated with one picture order count, denoted as PicOrderCntVal.

When none of the following conditions is true: [Ed. (GJS): Logical structure of sentence seems strange.]

* The current picture is an IDR
* The current picture is a BLA picture
* The current picture is a CRA picture and is the first coded picture in the bitstream

the variables prevPicOrderCntLsb and prevPicOrderCntMsb are derived as follows. Let prevTid0Pic be the previous reference picture in decoding order that has TemporalId equal to 0. The variable prevPicOrderCntLsb is set equal to pic\_order\_cnt\_lsb of prevTid0Pic, and the variable prevPicOrderCntMsb is set equal to PicOrderCntMsb of prevTid0Pic.

The variable PicOrderCntMsb of the current picture is derived as follows.

* If the current picture is an IDR or a BLA picture, or if the first coded picture in the bitstream is a CRA picture and the current picture is the first coded picture in the bitstream, PicOrderCntMsb is set equal to 0.
* Otherwise, PicOrderCntMsb is derived as specified by the following pseudo-code:

if( ( pic\_order\_cnt\_lsb < prevPicOrderCntLsb ) &&  
 ( ( prevPicOrderCntLsb − pic\_order\_cnt\_lsb ) >= ( MaxPicOrderCntLsb / 2 ) ) )  
 PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb (8‑1)  
else if( (pic\_order\_cnt\_lsb > prevPicOrderCntLsb ) &&  
 ( (pic\_order\_cnt\_lsb − prevPicOrderCntLsb ) > ( MaxPicOrderCntLsb / 2 ) ) )  
 PicOrderCntMsb = prevPicOrderCntMsb − MaxPicOrderCntLsb  
else  
 PicOrderCntMsb = prevPicOrderCntMsb

PicOrderCntVal is derived as

PicOrderCntVal = PicOrderCntMsb + pic\_order\_cnt\_lsb (8‑2)

NOTE  – All IDR pictures will have PicOrderCntVal equal to 0 since pic\_order\_lsb is inferred to be 0 for IDR pictures and prevPicOrderCntLsb and prevPicOrderCntMsb are both set equal to 0.

The value of PicOrderCntVal shall be in the range of −263 to 263 − 1, inclusive. In one coded video sequence, the PicOrderCntVal values for any two coded pictures shall not be the same.

The function PicOrderCnt( picX ) is specified as follows:

PicOrderCnt( picX ) = PicOrderCntVal of the picture picX (8‑3)

The function DiffPicOrderCnt( picA, picB ) is specified as follows:

DiffPicOrderCnt( picA, picB ) = PicOrderCnt( picA ) − PicOrderCnt( picB ) (8‑4)

The bitstream shall not contain data that result in values of DiffPicOrderCnt( picA, picB ) used in the decoding process that are not in the range of −215 to 215 − 1, inclusive. [Ed. (GJS/YKW): Include the limit in all places where such a POC difference is used in the decoding process (i.e. TMVP).]

NOTE  – Let X be the current picture and Y and Z be two other pictures in the same sequence, Y and Z are considered to be in the same output order direction from X when both DiffPicOrderCnt( X, Y ) and DiffPicOrderCnt( X, Z ) are positive or both are negative.

NOTE  – Many encoders assign PicOrderCntVal proportional to the sampling time of the corresponding picture relative to the sampling time of the previous IDR or BLA picture.

NOTE 4 – The range limitation on the difference of two POC values referred in the decoding process of any access unit enables a decoder to handle PicOrderCntVal with the range limit smaller than [-263, 263 - 1], since only the relative value of POC is used in the decoding process. The minimum range limit of the difference of two stored POC values needed for yielding the same result as in [-263, 263 - 1] range limit is [-224, 216 - 1] (i.e. 25-bit range). In order to implement such smaller range limit, a signed x-bit (x = [25, 64]) register, in which a negative value is represented in 2's complement, is typically used to store LSB x-bit of each variable relating to a POC value (such as PicOrderCntVal and PicOrderCntMsb). Addition, subtraction and comparison of two POC-related variables stored in signed x-bit registers are also performed in x-bit range, without using a carry bit or a borrow bit. As the result, PicOrderCntVal is seen as the variable whose range limit is [-2x-1, 2x-1 - 1] and whose value is cyclic.

# Examples

The following examples show that the operations of PicOrderCntVal stored in signed 32-bit register can still yield the correct result even when PicOrderCntVal is at the upper limit and lower limit of 32-bit representation.

A variable with suffix '\_64' is stored in signed 64-bit register. A variable with suffix '\_32' is stored in signed 32-bit register.

Case 1: A = 231-1, B = 1

C = A + B = 231 (64-bit) or -231 (32-bit)

D = C - A = 1

E = A - C = -1

A\_64 = 0x0000 0000 7FFF FFFF

A\_32 = 0x7FFF FFFF ( = A\_64 & 0xFFFF FFFF)

B\_64 = 0x0000 0000 0000 0001

B\_32 = 0x0000 0001

C\_64 = 0x0000 0000 8000 0000

C\_32 = 0x8000 0000 ( = -231)

D\_64 = 0x0000 0000 0000 0001

D\_32 = 0x0000 0001 ( =D\_64 & 0xFFFF FFFF)

E\_64 = 0xFFFF FFFF FFFF FFFF

E\_32 = 0xFFFF FFFF ( =E\_64 & 0xFFFF FFFF)

Case 2: A = 232+231+1, B = -2

C = A + B = 232+231-1 (64-bit) or 231-1 (32-bit)

D = C - A = -2

E = A - C = 2

A\_64 = 0x0000 0001 8000 0001

A\_32 = 0x8000 0001 ( = -231+1) ( = A\_64 & 0xFFFF FFFF)

B\_64 = 0xFFFF FFFF FFFF FFFE

B\_32 = 0xFFFF FFFE

C\_64 = 0x0000 0001 7FFF FFFF

C\_32 = 0x7FFF FFFF (=231-1)

D\_64 = 0xFFFF FFFF FFFF FFFE

D\_32 = 0xFFFF FFFE = (D\_64 | 0xFFFF FFFF)

E\_64 = 0x0000 0000 0000 0002

E\_32 = 0x0000 0002 = (E\_64 | 0xFFFF FFFF)

Case 3: A = 232-1, B = 1

C = A + B = 232 (64-bit) or 0 (32-bit)

D = C - A = 1

E = A - C = -1

A\_64 = 0x0000 0000 FFFF FFFF

A\_32 = 0xFFFF FFFF ( = -1) ( = A\_64 & 0xFFFF FFFF)

B\_64 = 0x0000 0000 0000 0001

B\_32 = 0x0000 0001

C\_64 = 0x0000 0001 0000 000

C\_32 = 0x0000 0000 (=0)

D\_64 = 0x0000 0000 0000 0001

D\_32 = 0x0000 0001 = (D\_64 | 0xFFFF FFFF)

E\_64 = 0xFFFF FFFF FFFF FFFF

E\_32 = 0xFFFF FFFF = (E\_64 | 0xFFFF FFFF)

In the above examples, variable A contains the absolute POC value and variable B contains the delta value.

# Conclusion

It is proposed to adopt the proposed text into the latest draft specification text.

# References

[1] "High Efficiency Video Coding (HEVC) text specification draft 8", JCTVC-J1003, July 2012, Stockholm Sweden

[2] K.Kazui et al., "Cyclic POC", JCTVC-I0045, July 2012, Stockholm Sweden

[3] “Proposed Editorial Improvements for High efficiency video coding (HEVC) Text Specification Draft 8”, JCTVC-K0030, October 2012, Shanghai China

[4] "Meeting Report of 10th JCT-VC Meeting", JCTVC-J1000, July 2012, Stockholm Sweden

# Patent rights declaration(s)

**FUJITSU LIMITED 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).**

1. The draft specification text (version 8) defines the range limit of a POC value as 32-bit. On the order hand, the decoding process only uses the difference of two POC values instead of using the absolute value. Since the difference between any two POC values used at the decoding process of a coded picture is limited to 24-bit, a decoder only handling the LSB 25-bit of the POC value can still decode a picture correctly. [↑](#footnote-ref-1)