|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  12th Meeting: Geneva, CH, 14–23 Jan. 2013 | Document: JCTVC-L0179r1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Output flag location** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Jill Boyce, Wonkap Jang, Danny Hong, Stephan Wenger 433 Hackensack Ave  Hackensack, NJ 07601, USA  Miska Hannuksela | Tel: Email: | +1 201 478 6145 {jill, wonkap, danny, stephan}@vidyo.com  miska.hannuksela@nokia.com |
| *Source:* | Vidyo, Nokia | | |

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

# Abstract

In SVC, output\_flag was included in the NAL unit header extension, to indicate that a coded picture not be output. In the current HEVC draft specification, pic\_output\_flag is located in the slice segment header, and made optional according to output\_flag\_present\_flag in the PPS. The location of the pic\_output\_flag in the slice header following variable length coded syntax elements is burdensome to a middle box that sometimes changes the value of that flag. Two options for alternate solutions to indicate that a coded picture not be displayed are proposed which simplify middle box operation and improve robustness. In the first proposed option, pic\_output\_flag is placed early in the slice segment header, before any variable length coded elements, and is always present. In the second proposed option, a “no display” SEI message is introduced, and the pic\_output\_flag syntax element is removed from the slice segment header, and the output\_flag\_present\_flag is removed from the PPS.

In the r1 version, based on a suggestion from Miska Hannuksela during the initial presentation discussion, a third option is proposed, as a variant on the first option, where pic\_output\_flag is present only for the first slice segment in the picture. He is added as an author for the r1 version.

# Problem Statement

In SVC, output\_flag is included in the NAL unit header SVC extension. When its value is 0, the decoder will decode the coded picture, but not output it. In JVT-W047r1 [1], several use cases for output\_flag were described, including the thinning of a scalable bitstream by a middle box.

In the current HEVC draft specification, pic\_output\_flag is conditionally located in the slice segment header, dependent on a presence flag in the PPS. The pic\_output\_flag location follows variable length coded syntax elements in the slice segment header. This location places a higher burden on the middle box to change the value of the output\_flag than was the case in SVC, and reduces loss robustness. The middle box must parse all of the preceding syntax elements in the slice segment header, including VLC-coded syntax elements, parse the stored PPS to determine if pic\_output\_flag is present, and then change the value of the pic\_output\_flag. If any packet errors occurred in the path from the encoder to the middle box, and the middle box misses a PPS, the middle box could accidentally inject an error into a coded packet when attempting to change the value of a pic\_output\_flag syntax element that wasn’t present in the packet.

The current HEVC draft specification uses a PicOutputFlag variable, which is either assigned the value of pic\_output\_flag or its value is inferred for some picture coding orders associated with RAPs. The PicOutputFlag variable is used in the HRD operation.

From the HM9 draft specification:

PicOutputFlag is set as follows:

– If the current picture is a RASL picture and the previous RAP picture in decoding order is a BLA picture or is a CRA picture that is the first coded picture in the bitstream, PicOutputFlag is set equal to 0.

– Otherwise, PicOutputFlag is set equal to pic\_output\_flag.

# Proposed solutions

It is desirable for a middle box to be able to easily change the value of the pic\_output\_flag or otherwise indicate that a coded picture not be displayed, without having to parse variable length coded syntax elements and parameter sets. Two options are proposed which make it easier for a middle box to signal that a coded picture not be displayed.

## Option 1: Slice segment header

In the first option, the pic\_output\_flag syntax element is moved to a location early in the slice header. It is always present, and not dependent on an output\_flag\_present\_flag, which is proposed to be removed from the PPS. No other changes to the specification are required.

The proposed slice segment header and PPS syntax are below.

|  |  |
| --- | --- |
| slice\_segment\_header( ) { | Descriptor |
| **first\_slice\_segment\_in\_pic\_flag** | u(1) |
| **pic\_output\_flag** | u(1) |
| if( RapPicFlag ) |  |
| **no\_output\_of\_prior\_pics\_flag** | u(1) |
| **pic\_parameter\_set\_id** [Ed. (GJS): Violates syntax element naming convention by havin the same name as a syntax element in the picture parameter set.] | ue(v) |
| if( !first\_slice\_segment\_in\_pic\_flag ) { |  |
| if(dependent\_slice\_segments\_enabled\_flag ) |  |
| **dependent\_slice\_segment\_flag** | u(1) |
| **slice\_segment\_address** | u(v) |
| } |  |
| if( !dependent\_slice\_segment\_flag ) { |  |
| for ( i = 0; i < num\_extra\_slice\_header\_bits; i++ ) |  |
| **slice\_reserved\_undetermined\_flag**[ i ] | u(1) |
| **slice\_type** | ue(v) |
|  |  |
|  |  |
| if( separate\_colour\_plane\_flag = = 1 ) |  |
| **colour\_plane\_id** | u(2) |
| … |  |

|  |  |
| --- | --- |
| pic\_parameter\_set\_rbsp( ) { | Descriptor |
| **pps\_pic\_parameter\_set\_id** | ue(v) |
| **pps\_seq\_parameter\_set\_id** | ue(v) |
| **dependent\_slice\_segments\_enabled\_flag** | u(1) |
| **sign\_data\_hiding\_flag** | u(1) |
| **cabac\_init\_present\_flag** | u(1) |
| **num\_ref\_idx\_l0\_default\_active\_minus1** | ue(v) |
| **num\_ref\_idx\_l1\_default\_active\_minus1** | ue(v) |
| **init\_qp\_minus26** | se(v) |
| **constrained\_intra\_pred\_flag** | u(1) |
| **transform\_skip\_enabled\_flag** | u(1) |
| **cu\_qp\_delta\_enabled\_flag** | u(1) |
| if ( cu\_qp\_delta\_enabled\_flag ) |  |
| **diff\_cu\_qp\_delta\_depth** | ue(v) |
| **pps\_cb\_qp\_offset** | se(v) |
| **pps\_cr\_qp\_offset** | se(v) |
| **pps\_slice\_chroma\_qp\_offsets\_present\_flag** | u(1) |
| **weighted\_pred\_flag** | u(1) |
| **weighted\_bipred\_flag** | u(1) |
| **transquant\_bypass\_enable\_flag** | u(1) |
| … |  |

## Option 2: SEI message

In the second option, a no display SEI message is proposed. The new SEI message may be repeated in the access unit, and is a prefix SEI message. When the no display SEI message is present, the current coded picture is not displayed.

In this option, the pic\_output\_flag syntax element is removed from the slice segment header, and the output\_flag\_present\_flag is removed from the PPS. This option is based upon the assumption that the middle box would be unable to easily change the value of the pic\_output\_flag in the current location in the slice segment header. To avoid confusion of possibly having different signals in the SEI message and the slice segment header, the pic\_output\_flag would be removed, as well as its presence flag in the PPS, output\_flag\_present\_flag.

In the current draft specification, where the PicOutputFlag variable assignment refers to the pic\_output\_flag syntax element, it is replaced by 1, e.g. as if pic\_output\_flag was always set for output. Using an SEI message means that the HRD and any normative operations are unaffected by the “no display” designation of an access unit.

The attachment provides a full marked specification for the option 2 approach. The proposed SEI message has no payload, and applies to the current access unit only, with no persistence. The syntax and semantics of the proposed SEI message are below.

### No display SEI message syntax and semantics

|  |  |
| --- | --- |
| no\_display( payloadSize ) { | Descriptor |
| } |  |

D.2.25 No display SEI message semantics

The no display SEI message indicates that the current picture should not affect the content of the display.

### SEI payload syntax

|  |  |
| --- | --- |
| sei\_payload( payloadType, payloadSize ) { | Descriptor |
| if( nal\_unit\_type = = PREFIX\_SEI\_NUT ) |  |
| if( payloadType = = 0 ) |  |
| buffering\_period( payloadSize ) |  |
| else if( payloadType = = 1 ) |  |
| pic\_timing( payloadSize ) |  |
| else if( payloadType = = 2 ) |  |
| pan\_scan\_rect( payloadSize ) |  |
| else if( payloadType = = 3 ) |  |
| filler\_payload( payloadSize ) |  |
| else if( payloadType = = 4 ) |  |
| user\_data\_registered\_itu\_t\_t35( payloadSize ) |  |
| else if( payloadType = = 5 ) |  |
| user\_data\_unregistered( payloadSize ) |  |
| else if( payloadType = = 6 ) |  |
| recovery\_point( payloadSize ) |  |
| else if( payloadType = = 9 ) |  |
| scene\_info( payloadSize ) |  |
| else if( payloadType = = 15 ) |  |
| full\_frame\_snapshot( payloadSize ) |  |
| else if( payloadType = = 16 ) |  |
| progressive\_refinement\_segment\_start( payloadSize ) |  |
| else if( payloadType = = 17 ) |  |
| progressive\_refinement\_segment\_end( payloadSize ) |  |
| else if( payloadType = = 19 ) |  |
| film\_grain\_characteristics( payloadSize ) |  |
| else if( payloadType = = 22 ) |  |
| post\_filter\_hint( payloadSize ) |  |
| else if( payloadType = = 23 ) |  |
| tone\_mapping\_info( payloadSize ) |  |
| else if( payloadType = = 45 ) |  |
| frame\_packing\_arrangement( payloadSize ) |  |
| else if( payloadType = = 47 ) [Ed. (GJS): Check numbering w.r.t. AVC.] |  |
| display\_orientation( payloadSize ) |  |
| else if( payloadType = = 128 ) |  |
| sop\_description( payloadSize ) |  |
| else if( payloadType = = 129 ) |  |
| active\_parameter\_sets( payloadSize ) |  |
| else if( payloadType = = 130 ) |  |
| decoding\_unit\_info( payloadSize ) |  |
| else if( payloadType = = 131 ) |  |
| tl0\_index( payloadSize ) |  |
| else if( payloadType = = 133 ) |  |
| scalable\_nesting( payloadSize ) |  |
| else if( payloadType = = 134 ) |  |
| region\_refresh\_info( payloadSize ) |  |
| else if( payloadType = = 135 ) |  |
| no\_display( payloadSize ) |  |
| else |  |
| reserved\_sei\_message( payloadSize ) |  |

### Semantics change to 8.1 General decoding process

PicOutputFlag is set as follows:

– If the current picture is a RASL picture and the previous RAP picture in decoding order is a BLA picture or is a CRA picture that is the first coded picture in the bitstream, PicOutputFlag is set equal to 0.

– Otherwise, PicOutputFlag is set equal to1.

## Option 3: Slice segment header variant

Option 3 is a variant on the first option. As in the first option, the pic\_output\_flag syntax element is moved to a location early in the slice header. It is present only for the first slice segment in the picture, and not dependent on an output\_flag\_present\_flag, which is proposed to be removed from the PPS. No other changes to the specification are required.

|  |  |
| --- | --- |
| slice\_segment\_header( ) { | Descriptor |
| **first\_slice\_segment\_in\_pic\_flag** | u(1) |
| if ( first\_slice\_segment\_in\_pic\_flag ) |  |
| **pic\_output\_flag** | u(1) |
| if( RapPicFlag ) |  |
| **no\_output\_of\_prior\_pics\_flag** | u(1) |
| **pic\_parameter\_set\_id** [Ed. (GJS): Violates syntax element naming convention by havin the same name as a syntax element in the picture parameter set.] | ue(v) |
| if( !first\_slice\_segment\_in\_pic\_flag ) { |  |
| if(dependent\_slice\_segments\_enabled\_flag ) |  |
| **dependent\_slice\_segment\_flag** | u(1) |
| **slice\_segment\_address** | u(v) |
| } |  |
| if( !dependent\_slice\_segment\_flag ) { |  |
| for ( i = 0; i < num\_extra\_slice\_header\_bits; i++ ) |  |
| **slice\_reserved\_undetermined\_flag**[ i ] | u(1) |
| **slice\_type** | ue(v) |
|  |  |
|  |  |
| if( separate\_colour\_plane\_flag = = 1 ) |  |
| **colour\_plane\_id** | u(2) |
| … |  |

# References

[1] Miska M. Hannuksela, Ye-Kui Wang, JVT-W047r1, “Pictures not for output in SVC”, San Jose, California, USA, 21–27 April, 2007.

# Patent rights declaration(s)

**Vidyo may have IPR 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).**

**Nokia may have IPR 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).**