|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  10th Meeting: Stockholm, Sweden, July 11-20, 2012 | Document: JCTVC-J0087  M25409 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **AHG6: Independent luma and chroma SAO on/off control at slice level** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Minhua Zhou Texas Instruments Inc., USA | Tel: Email:  : | +1-214-480-3816 [zhou@ti.com](mailto:zhou@ti.com) |
| *Source:* | Texas Instruments Inc. | | |

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

# Abstract

In the current HM7.0 design it is normatively restricted at slice level that chroma SAO has to be turned off when luma SAO is off. Such a normative encoder restriction is undesirable because it restricts encoder flexibility. This contribution advocates removing such a restriction and making SAO on/off control flag fully independent at slice level. The proposed change does not affect coding efficiency.

# Introduction

In the current HM7.0 design it is normatively restricted at slice level that chroma SAO has to be turned off if luma SAO is off. Such a normative encoder restriction is undesirable because it restricts encoder flexibility. This contribution advocates removing such a restriction and making SAO on/off control flag fully independent at slice level. With the proposed change, the SAO syntax in slice header is simplified (see Table 1 vs. table 2).

|  |  |
| --- | --- |
| if( sample\_adaptive\_offset\_enabled\_flag ) { |  |
| **slice\_sample\_adaptive\_offset\_flag[** 0 **]** | u(1) |
| if( slice\_sample\_adaptive\_offset\_flag[ 0 ] ) { |  |
| **slice\_sample\_adaptive\_offset\_flag[** 1 **]** | u(1) |
| **slice\_sample\_adaptive\_offset\_flag[** 2 **]** | u(1) |
| } |  |
| } |  |

**Table 1. HM7.0 SAO syntax in slice header**

|  |  |
| --- | --- |
| if( sample\_adaptive\_offset\_enabled\_flag ) { |  |
| **slice\_sample\_adaptive\_offset\_flag[** 0 **]** | u(1) |
| **slice\_sample\_adaptive\_offset\_flag[** 1 **]** | u(1) |
| **slice\_sample\_adaptive\_offset\_flag[** 2 **]** | u(1) |
| } |  |

**Table 2. Proposed SAO syntax in slice header**

# Test Settings and Conditions

The simulations of this document have used HM7.0 software, the simulation platform is LSF equipped with Intel(R) Xeon(R) CPU X5570 64 bits Linux machines of different frequencies, the common test conditions and reference configurations specified in [1] are followed.

# Experimental results

Table 3 summarizes the experimental results of proposed change, no BD-rate difference is observed. The bitsteams are not bi-exact in some cases, indicating that chroma SAO could be on when luma SAO is off.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | All Intra Main | | | All Intra HE10 | | |
|  | Y | U | V | Y | U | V |
| Class A | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class B | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class C | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class D | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class E | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| **Overall** | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
|  | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class F | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 96% | | | 100% | | |
| Dec Time[%] | 95% | | | 99% | | |
|  |  | | |  | | |
|  | **Random Access Main** | | | **Random Access HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class B | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class C | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class D | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class E |  |  |  |  |  |  |
| **Overall** | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
|  | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class F | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 102% | | | 101% | | |
| Dec Time[%] | 103% | | | 101% | | |
|  |  | | |  | | |
|  | **Low delay B Main** | | | **Low delay B HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A |  | | |  | | |
| Class B | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class C | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class D | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class E | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| **Overall** | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
|  | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class F | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 98% | | | 98% | | |
|  |  | | |  | | |
|  | **Low delay P Main** | | | **Low delay P HE10** | | |
|  | Y | U | V | Y | U | V |
| Class A |  | | |  | | |
| Class B | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class C | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class D | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class E | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| **Overall** | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
|  | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class F | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 97% | | | 96% | | |
| Dec Time[%] | 95% | | | 91% | | |

**Table 3. Experimental results of proposed algorithm**

# Conclusions

# The proposed change removes the normative SAO encoder restriction and makes the encoder more flexible. It is recommended to adopt this change.

# References

[1] F. Bossen, “Common test conditions and software reference configurations,” JCT-VC Document, JCTVC-I1100, 9th Meeting: Geneva, Switzerland, 27 April – 07 May, 2012

[2] [B. Bross](mailto:benjamin.bross@hhi.fraunhofer.de), [W.-J. Han](mailto:wjhan.han@samsung.com), [J.-R. Ohm](mailto:ohm@ient.rwth-aachen.de), [G. J. Sullivan](mailto:garysull@microsoft.com), [T. Wiegand](mailto:thomas.wiegand@hhi.fraunhofer.de) “High Efficiency Video Coding (HEVC) text specification draft 7,” JCT-VC Document, JCTVC-I1003, 9th Meeting: Geneva, Switzerland, 27 April – 07 May, 2012.

# Patent rights declaration(s)

**Texas Instruments, Inc. does not 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).**

# CD text

in section 7.3.3 Slice header syntax, modify the syntax as highlighted in yellow.

|  |  |
| --- | --- |
| slice\_header( ) { | Descriptor |
| **first\_slice\_in\_pic\_flag** | u(1) |
| **pic\_parameter\_set\_id** | ue(v) |
| if( !first\_slice\_in\_pic\_flag ) |  |
| **slice\_address** | u(v) |
| if( dependent\_slice\_enabled\_flag && !first\_slice\_in\_pic\_flag ) |  |
| **dependent\_slice\_flag** | u(1) |
| if( !dependent\_slice\_flag ) { |  |
| **slice\_type** | ue(v) |
| if( output\_flag\_present\_flag ) |  |
| **pic\_output\_flag** | u(1) |
| if( **separate\_colour\_plane\_flag** = = 1 ) |  |
| **colour\_plane\_id** | u(2) |
| if( RapPicFlag ) { |  |
| **rap\_pic\_id** | ue(v) |
| **no\_output\_of\_prior\_pics\_flag** | u(1) |
| } |  |
| if( !IdrPicFlag ) { |  |
| **pic\_order\_cnt\_lsb** | u(v) |
| **short\_term\_ref\_pic\_set\_sps\_flag** | u(1) |
| if( !short\_term\_ref\_pic\_set\_sps\_flag ) |  |
| short\_term\_ref\_pic\_set( num\_short\_term\_ref\_pic\_sets ) |  |
| else |  |
| **short\_term\_ref\_pic\_set\_idx** | u(v) |
| if( long\_term\_ref\_pics\_present\_flag ) { |  |
| **num\_long\_term\_pics** | ue(v) |
| for( i = 0; i < num\_long\_term\_pics; i++ ) { |  |
| **poc\_lsb\_lt**[ i ] | u(v) |
| **delta\_poc\_msb\_present\_flag**[ i ] | u(1) |
| if( delta\_poc\_msb\_present\_flag[ i ] ) |  |
| **delta\_poc\_msb\_cycle\_lt**[ i ] | ue(v) |
| **used\_by\_curr\_pic\_lt\_flag**[ i ] | u(1) |
| } |  |
| } |  |
| } |  |
| if( sample\_adaptive\_offset\_enabled\_flag ) { |  |
| **slice\_sample\_adaptive\_offset\_flag[** 0 **]** | u(1) |
| ~~if( slice\_sample\_adaptive\_offset\_flag[ 0 ] ) {~~ |  |
| **slice\_sample\_adaptive\_offset\_flag[** 1 **]** | u(1) |
| **slice\_sample\_adaptive\_offset\_flag[** 2 **]** | u(1) |
| ~~}~~ |  |
| } |  |
| if(adaptive\_loop\_filter\_enabled\_flag ) |  |
| **aps\_id** | ue(v) |
| if( slice\_type = = P | | slice\_type = = B ) { |  |
| if( sps\_temporal\_mvp\_enable\_flag ) |  |
| **pic\_temporal\_mvp\_enable\_flag** | u(1) |
| **num\_ref\_idx\_active\_override\_flag** | u(1) |
| if( num\_ref\_idx\_active\_override\_flag ) { |  |
| **num\_ref\_idx\_l0\_active\_minus1** | ue(v) |
| if( slice\_type = = B ) |  |
| **num\_ref\_idx\_l1\_active\_minus1** | ue(v) |
| } |  |
| } |  |
| if( lists\_modification\_present\_flag ) |  |
| ref\_pic\_list\_modification( ) |  |
| if( slice\_type = = B ) |  |
| **mvd\_l1\_zero\_flag** | u(1) |
| if( cabac\_init\_present\_flag && slice\_type != I ) |  |
| **cabac\_init\_flag** | u(1) |
| **slice\_qp\_delta** | se(v) |
| if( deblocking\_filter\_control\_present\_flag ) { |  |
| if( deblocking\_filter\_override\_enabled\_flag ) |  |
| **deblocking\_filter\_override\_flag** | u(1) |
| if( deblocking\_filter\_override\_flag ) { |  |
| **slice\_header\_disable\_deblocking\_filter\_flag** | u(1) |
| if( !slice\_header\_disable\_deblocking\_filter\_flag ) { |  |
| **beta\_offset\_div2** | se(v) |
| **tc\_offset\_div2** | se(v) |
| } |  |
| } |  |
| } |  |
| if( pic\_temporal\_mvp\_enable\_flag ) { |  |
| if( slice\_type = = B ) |  |
| **collocated\_from\_l0\_flag** | u(1) |
| if( slice\_type != I &&   ((collocated\_from\_l0\_flag && num\_ref\_idx\_l0\_active\_minus1 > 0) | |  (!collocated\_from\_l0\_flag && num\_ref\_idx\_l1\_active\_minus1 > 0) ) |  |
| **collocated\_ref\_idx** | ue(v) |
| } |  |
| if( ( weighted\_pred\_flag && slice\_type = = P) | |  ( weighted\_bipred\_idc = = 1 && slice\_type = = B ) ) |  |
| pred\_weight\_table( ) |  |
| if( slice\_type = = P | | slice\_type = = B ) |  |
| **five\_minus\_max\_num\_merge\_cand** | ue(v) |
| if( adaptive\_loop\_filter\_enabled\_flag ) { |  |
| **slice\_adaptive\_loop\_filter\_flag** | u(1) |
| if( slice\_adaptive\_loop\_filter\_flag && alf\_coef\_in\_slice\_flag ) |  |
| alf\_param( ) |  |
| if( slice\_adaptive\_loop\_filter\_flag && !alf\_coef\_in\_slice\_flag ) |  |
| alf\_cu\_control\_param( ) |  |
| } |  |
| if( seq\_loop\_filter\_across\_slices\_enabled\_flag &&  ( slice\_adaptive\_loop\_filter\_flag | | slice\_sample\_adaptive\_offset\_flag | |  !disable\_deblocking\_filter\_flag ) ) |  |
| **slice\_loop\_filter\_across\_slices\_enabled\_flag** | u(1) |
| } |  |
| if( tiles\_or\_entropy\_coding\_sync\_idc = = 1 | |  tiles\_or\_entropy\_coding\_sync\_idc = = 2 ) { |  |
| **num\_entry\_point\_offsets** | ue(v) |
| if( num\_entry\_point\_offsets > 0 ) { |  |
| **offset\_len\_minus1** | ue(v) |
| for( i = 0; i < num\_entry\_point\_offsets; i++ ) |  |
| **entry\_point\_offset**[ i ] | u(v) |
| } |  |
| } |  |
| if( slice\_header\_extension\_present\_flag ) { |  |
| **slice\_header\_extension\_length** | ue(v) |
| for( i = 0; i < slice\_header\_extension\_length; i++) |  |
| **slice\_header\_extension\_data\_byte** | u(8) |
| } |  |
| byte\_alignment( ) |  |
| } |  |