|  |  |
| --- | --- |
| **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-H0515r2 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **On Entropy Slices** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Muhammed Coban  Ye-Kui Wang  Jaehong Chon  Marta Karczewicz  5775 Morehouse Drive San Diego, CA 92121, USA | Tel: Email: | 1-858-658-3973 [mcoban@qualcomm.com](mailto:mcoban@qualcomm.com)  [yekuiw@qualcomm.com](mailto:yekuiw@qualcomm.com)  [jchon@qualcomm.com](mailto:jchon@qualcomm.com)  [martak@qualcomm.com](mailto:martak@qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

Entropy slices provide independently parsable units by disabling usage of context elements across slices and resetting of CABAC states at the beginning of each entropy slice, while allowing in-picture prediction across slices for decoding. Each entropy slice is encapsulated in its NAL unit with its light weight slice header. Tiles and wavefront parallel processing methods that offer alternative parallel processing mechanisms signal entry points to parallel decodable units in the main slice header. In a similar approach this document proposes that all entropy slices associated with the same “parent” slice are encapsulated into one coded slice NAL unit and the entry points of the entropy slices are signalled in the slice header. Simulation results show 0.1% to 0.2% BD rate coding gain for common test configurations.

# Introduction

Entropy slices provide independently parsable units that encapsulated in their own NAL units, by disabling usage of context elements across slices and resetting of CABAC states at the beginning of each entropy slice, while allowing in-picture prediction across slices for decoding. Entropy slices also differ from regular slices in their slice header structure. Entropy slices use a subset of the fields that are present in the regular slice header, while inheriting the missing fields from the main slice header that precedes the entropy slice in decoding order.

The current entropy slice design breaks a convention that has been followed for many years – no cross-dependency between NAL units of the same level/hierarchy, mostly for error resilience reasons. In practice, individual NAL units are often transported in their own packets, i.e. one NAL unit one packet, during transmission through networks. A problem with the current entropy slice design is that, for an entropy slice, if the associated slice (with a full slice header from which the entropy slice must take the missing slice header syntax elements) is lost, or another entropy slice (for which in-picture prediction across the two entropy slices is allowed) is lost, the entropy slice becomes useless.

This document proposes to keep the above-mentioned convention, by mandating that all entropy slices associated with the same “parent” slice are encapsulated into one coded slice NAL unit and the entry points of the entropy slices are signalled in the slice header.

# Proposal

Entropy slices are treated as sub-streams where each sub-stream is an individually parsable sub-stream that is byte aligned. Entry points to the sub-streams are signalled as byte offsets in the main slice’s slice header that precedes the first entropy slice (sub-stream). Entropy slices requires the signalling of the CU first CU address in the entropy slice. Signalling of the starting CU address is moved to the main slice header. Further more CABAC initialization of the entropy slices (ES) (sub-streams) are performed using the cabac\_init\_idc and slice\_qp\_delta parameters in the main slice header. The structure of the slice data is shown below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SliceHeader (other syntax elements) | SH:  ES Entry Point offsets | SH: ES Entry Point CU address | ES Data | ES Data | …… | ES Data | End\_of\_slice\_flag=1 |

* 1. ***Syntax changes for PPS and slice header***

The changed sequence parameter set syntax, picture parameter set syntax, and slice header syntax are as follows.

**Picture parameter set RBSP syntax**

|  |  |
| --- | --- |
| pic\_parameter\_set\_rbsp( ) { | **Descriptor** |
| **pic\_parameter\_set\_id** | ue(v) |
| **seq\_parameter\_set\_id** | ue(v) |
| **entropy\_coding\_synchro** | u(v) |
| **cabac\_istate\_reset\_flag** | u(1) |
| if( entropy\_coding\_synchro ) |  |
| **num\_substreams\_minus1** | ue(v) |
| **num\_short\_term\_ref\_pic\_sets** | ue(v) |
| for(idx = 0; idx < num\_short\_term\_ref\_pic\_sets; idx++) |  |
| short\_term\_ref\_pic\_set( idx ) |  |
| **long\_term\_ref\_pics\_present\_flag** | u(1) |
| **num\_temporal\_layer\_switching\_point\_flags** | ue(v) |
| for( i = 0; i < num\_temporal\_layer\_switching\_point\_flags; i++ ) |  |
| **temporal\_layer\_switching\_point\_flag**[ i ] | u(1) |
| **num\_ref\_idx\_l0\_default\_active\_minus1** | ue(v) |
| **num\_ref\_idx\_l1\_default\_active\_minus1** | ue(v) |
| [Ed. (BB): not present in HM software, depends on reference list construction decision ] |  |
| **pic\_init\_qp\_minus26** **/**\* relative to 26 \*/ | se(v) |
| [Ed. (BB): not present in HM software, signaled in slice header as absolute value slice\_qp, should be implemented to be used for slice\_qp\_delta] |  |
| **constrained\_intra\_pred\_flag** | u(1) |
| **slice\_granularity** | u(2) |
| **max\_cu\_qp\_delta\_depth** | ue(v) |
| [Ed. (BB): not present in HM software, should be implemented ] |  |
| **weighted\_pred\_flag** | u(1) |
| **weighted\_bipred\_idc** | u(2) |
| **tile\_info\_present\_flag** | u(1) |
| if( tile\_info\_present\_flag = = 1 **) {** |  |
| **num\_tile\_columns\_minus1** | ue(v) |
| **num\_tile\_rows\_minus1** | ue(v) |
| if( num\_tile\_columns\_minus1 != 0 | | num\_tile\_rows\_minus1 != 0 ) { |  |
| **tile\_boundary\_independence\_flag** | u(1) |
| **uniform\_spacing\_flag** | u(1) |
| if( !uniform\_spacing\_flag ) { |  |
| for( i = 0; i < num\_tile\_columns\_minus1; i++ ) |  |
| **column\_width[**i**]** | ue(v) |
| for( i = 0; i < num\_tile\_rows\_minus1; i++ ) |  |
| **row\_height[**i**]** | ue(v) |
| } |  |
| } |  |
| **}** |  |
| **entropy\_slice\_enabled \_flag** | u(1) |
| rbsp\_trailing\_bits( ) |  |
| } |  |

**Slice header syntax**

|  |  |
| --- | --- |
| slice\_header( ) { | **Descriptor** |
| **~~entropy\_slice\_flag~~** | ~~u(1)~~ |
| ~~if( !entropy\_slice\_flag ) {~~ |  |
| **slice\_type** | ue(v) |
| **pic\_parameter\_set\_id** | ue(v) |
| if( sample\_adaptive\_offset\_enabled\_flag || adaptive\_loop\_filter\_enabled\_flag ) |  |
| **aps\_id** | ue(v) |
| if( IdrPicFlag ) { |  |
| **idr\_pic\_id** | ue(v) |
| **no\_output\_of\_prior\_pics\_flag** | u(1) |
| } |  |
| else { |  |
| **pic\_order\_cnt\_lsb** | u(v) |
| **short\_term\_ref\_pic\_set\_pps\_flag** | u(1) |
| if( !short\_term\_ref\_pic\_set\_pps\_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++ ) { |  |
| **delta\_poc\_lsb\_lt\_minus1**[ i ] | ue(v) |
| **used\_by\_curr\_pic\_lt\_flag**[ i ] | u(1) |
| } |  |
| } |  |
| } |  |
| if( slice\_type = = P | | slice\_type = = B ) { |  |
| **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) |
| } |  |
| } |  |
| ref\_pic\_list\_modification( ) |  |
| ref\_pic\_list\_combination( ) |  |
| ~~}~~ |  |
| **first\_slice\_in\_pic\_flag** | u(1) |
| if( first\_slice\_in\_pic\_flag == 0 ) |  |
| **slice\_address** | u(v) |
| ~~if( !entropy\_slice\_flag ) {~~ |  |
| **slice\_qp\_delta** | se(v) |
| [Ed. (BB): coded as absolute value slice\_qp in HM software, should be implemented together with pic\_init\_qp\_minus26 ] |  |
| inherit\_dbl\_params\_from\_APS\_flag | u(1) |
| if ( !inherit\_dbl\_params\_from\_APS\_flag ) { |  |
| **disable\_deblocking\_filter\_flag** | u(1) |
| if ( !disable\_deblocking\_filter\_flag ) { |  |
| beta\_offset\_div2 | se(v) |
| tc\_offset\_div2 | se(v) |
| } |  |
| } |  |
| [Ed. (BB): Although it was agreed to use AVC-like deblocking control syntax, WD and HM should be changed to make use of the AVC-like syntax.] |  |
| if( slice\_type = = B ) |  |
| **collocated\_from\_l0\_flag** | u(1) |
| if( adaptive\_loop\_filter\_enabled\_flag && aps\_adaptive\_loop\_filter\_flag ) { |  |
| byte\_align( ) |  |
| alf\_cu\_control\_param( ) |  |
| byte\_align( ) |  |
| } |  |
| if( ( weighted\_pred\_flag && slice\_type = = P) | |  ( weighted\_bipred\_idc = = 1 && slice\_type = = B ) ) |  |
| pred\_weight\_table( ) |  |
| ~~}~~ |  |
| if( slice\_type = = P | | slice\_type = = B ) |  |
| 5\_minus\_max\_num\_merge\_cand | ue(v) |
| for( i = 0; i <num\_substreams\_minus1 + 1; i++ ){ |  |
| **substream\_length\_mode** | u(2) |
| **substream\_length[i]** | u(v) |
| } |  |
| if( entropy\_slice\_enabled\_flag ) { |  |
| **num\_entry\_offsets** | ue(v) |
| if( num\_entry\_offsets ) { |  |
| **offset\_len\_minus8** | ue(v) |
| for( i = 0; i < num\_entry\_offsets; i++ ) |  |
| **entry\_offset**[ i ] | u(v) |
| for( i = 0; i < num\_entry\_offsets; i++ ) |  |
| **entropy\_slice\_address**[ i ] | u(v) |
| } |  |
| } |  |
| } |  |

**entropy\_slice\_enabled\_flag** equal to 1 specifies that a coded slice referring to the picture parameter set may consist of entropy slices. entropy\_slice\_enabled\_flagequal to 0 specifies that a coded slice referring to the picture parameter set shall not contain entropy slices. When not present, the value of entropy\_slice\_enabled\_flagshall be inferred to be 0.

**num\_entry\_offsets** specifies the number of entry\_offset[ i ] syntax elements in the slice header. When not present, the value of num\_entry\_offsets shall be inferred to be 0.

**offset\_len\_minus8** plus 8 specifies the length, in bits, of the entry\_offset[ i ] syntax elements.

**entry\_offset**[ i ] specifies the i-th entry offset, in bytes. The coded slice NAL unit consists of num\_entry\_offsets + 1 sub-streams, with sub-stream index values in the range of 0 to num\_entry+offsets, inclusive. Sub-stream 0 consists of bytes 0 to entry\_offset[ 0 ] – 1, inclusive, of the coded slice NAL unit, and sub-stream k, with k in the range of 1 to num\_entry\_offsets - 1, inclusive, consists of bytes entry\_offset[ k-1 ] to entry\_offset[ k ] – 1, inclusive, of the coded slice NAL unit, and the last sub-stream (with sub-stream index equal to num\_entry\_offsets) consists of rest bytes of the coded slice NAL unit.

NOTE – The NAL unit header and the slice header of a coded slice NAL unit are always included in sub-stream 0.

When entropy\_slice\_enabled\_flag is equal to 1, each sub-stream shall contain all coded bits of one entropy slice and shall not contain any coded bits of another entropy slice.

# Results

The proposed entropy slice encapsulation method and the entry point signaling scheme were implemented in HM5.1. The simulations were conducted based on the JCT-VC common test condition defined in JCTVC-G1000 using max 18000 bin/slice limitation for the anchor and the proposed scheme. Table 1 shows summary of simulations. As can be seen from the results on average there is 0.1% - 0.2% BD coding gain, though BD coding gain is not the reason for the proposal.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | | **All Intra HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A (8bit) | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% |  |  |  |
| Class B | -0.1% | -0.1% | -0.1% | -0.2% | -0.2% | -0.2% |  |  |  |
| Class C | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% |  |  |  |
| Class D | -0.1% | -0.2% | -0.2% | -0.1% | -0.2% | -0.2% |  |  |  |
| Class E | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% |  |  |  |
| **Overall** | -0.1% | -0.1% | -0.1% | -0.1% | -0.2% | -0.2% |  |  |  |
|  | -0.1% | -0.1% | -0.1% | -0.1% | -0.2% | -0.2% |  |  |  |
| Class F |  |  |  |  |  |  |  |  |  |
| Enc Time[%] | 98% | | | 100% | | |  | | |
| Dec Time[%] | 77% | | | 72% | | |  | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **Random Access HE** | | | **Random Access LC** | | | **Random Access HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A (8bit) | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% |
| Class B | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% |
| Class C | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% |  |  |  |
| Class D | 0.0% | -0.1% | -0.1% | 0.0% | -0.1% | -0.1% |  |  |  |
| Class E |  |  |  |  |  |  |  |  |  |
| **Overall** | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.2% | -0.2% | -0.2% |
|  | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.2% | -0.2% | -0.2% |
| Class F |  |  |  |  |  |  |  |  |  |
| Enc Time[%] | 100% | | | 98% | | | 96% | | |
| Dec Time[%] | 90% | | | 89% | | | 92% | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **Low delay B HE** | | | **Low delay B LC** | | | **Low delay B HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |  |  |  |
| Class B | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% | -0.2% |  |  |  |
| Class C | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% |  |  |  |
| Class D | 0.0% | 0.0% | 0.0% | 0.0% | -0.1% | -0.1% |  |  |  |
| Class E | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |  |  |  |
| **Overall** | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% |  |  |  |
|  | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% | -0.1% |  |  |  |
| Class F |  |  |  |  |  |  |  |  |  |
| Enc Time[%] | 98% | | | 100% | | |  | | |
| Dec Time[%] | 89% | | | 88% | | |  | | |

# Patent rights declaration(s)

**Qualcomm Inc. 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).**