|  |  |
| --- | --- |
| **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-K0254 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **AHG9: Parsing profile and level information of temporal sub-layers** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Takeshi Tsukuba  Tomoyuki Yamamoto  1-9-2 Nakase, Mihama-ku, Chiba-shi, Chiba 261-8520  JAPAN | Tel: Email: | +81-43-299-8526  tsukuba.takeshi@sharp.co.jp yamamoto.tomoyuki@sharp.co.jp |
| *Source:* | SHARP Corporation | | |

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

# Abstract

This contribution proposes a change to make the syntax of profile\_tier\_level() more parsing friendly. The proposal separates the part of sub-layer profile/level present flags and the part of the sub-layer profile/level information, which keeps profile\_tier\_level() syntax byte-aligned.

# Problem Statement

In HEVC DIS [1], there remains a drawback of parsing sub-layer profile\_tier\_level() information:

1. The profile and level information has been designed to fulfill byte-aligned thoughtfully. However at the sub-layer related syntax that design fails to function, which is highlighted in orange in Table.1. The design inconsistency may cause unnecessary hardware and software implementation complexity. Byte aligned information is readable by human and is beneficial for development and test.
2. Sub-layer profile and level present flags and those information are mixed / interleaved in the syntax structure. When a decoder extracts the specified sub-layer profile and level information or the syntax following profile\_tier\_level(), such as vps\_reserved\_zero\_12bits ( next\_essenstial\_info\_byte\_offset [2] ) on VPS and seq\_parameter\_set\_id on SPS, it needs to parse all of the syntax of the preceding sub-layers or entire profile\_tier\_level() since it cannot easily estimate the amount of bits related to sub-layer profile and level information.
3. When ProfilePresentFlag is equal to 0, signalling of sub-layer profile present flags is redundant. [Option 2]

Therefore, the current profile\_tier\_level() syntax structure should be fixed to be designed decoder-friendly and development-friendly.

Table : Profile, tier and level syntax in HM8

|  |  |
| --- | --- |
| profile\_tier\_level( ProfilePresentFlag, MaxNumSubLayersMinus1 ) { | **Descriptor** |
| if( ProfilePresentFlag ) { |  |
| **general\_profile\_space** | u(2) |
| **general\_tier\_flag** | u(1) |
| **general\_profile\_idc** | u(5) |
| for( i = 0; i < 32; i++ ) |  |
| **general\_profile\_compatibility\_flag**[ i ] | u(1) |
| **general\_reserved\_zero\_16bits** | u(16) |
| } |  |
| **general\_level\_idc** | u(8) |
| for( i = 0; i < MaxNumSubLayersMinus1; i++ ) { |  |
| **sub\_layer\_profile\_present\_flag**[ i ] | u(1) |
| **sub\_layer\_level\_present\_flag**[ i ] | u(1) |
| if( ProfilePresentFlag && sub\_layer\_profile\_present\_flag[ i ] ){ |  |
| **sub\_layer\_profile\_space**[ i ] | u(2) |
| **sub\_layer\_tier\_flag**[ i ] | u(1) |
| **sub\_layer\_profile\_idc**[ i ] | u(5) |
| for( j = 0; j < 32; j++ ) |  |
| **sub\_layer\_profile\_compatibility\_flag**[ i ][ j ] | u(1) |
| **sub\_layer\_reserved\_zero\_16bits**[ i ] | u(16) |
| } |  |
| if( sub\_layer\_level\_present\_flag[ i ] ) |  |
| **sub\_layer\_level\_idc**[ i ] | u(8) |
| } |  |
| } |  |

# Proposal

To extract or skip parsing sub-layer profile and level information more easily, we propose the following two options.

## Option 1:

The first option is to split the part of sub-layer profile/level present flags and the part of sub-layer profile/level information so that present flags are collectively signaled before the associated information. In addition, to make all the syntax of profile\_tier\_level() byte-aligned, bit\_equal\_to\_zero are inserted after the first part. The second part is naturally byte-aligned due to its design.

Table 2 shows the proposed profile\_tier\_level() syntax where the changes are highlighted in yellow. There is no semantics change.

Table : Proposed Profile, tier, level syntax on Option 1

|  |  |
| --- | --- |
| profile\_tier\_level( ProfilePresentFlag, MaxNumSubLayersMinus1 ) { | **Descriptor** |
| if( ProfilePresentFlag ) { |  |
| **general\_profile\_space** | u(2) |
| **general\_tier\_flag** | u(1) |
| **general\_profile\_idc** | u(5) |
| for( i = 0; i < 32; i++ ) |  |
| **general\_profile\_compatibility\_flag**[ i ] | u(1) |
| **general\_reserved\_zero\_16bits** | u(16) |
| } |  |
| **general\_level\_idc** | u(8) |
| for( i = 0; i < MaxNumSubLayersMinus1; i++ ) { |  |
| **sub\_layer\_profile\_present\_flag[ i ]** | u(1) |
| **sub\_layer\_level\_present\_flag[ i ]** | u(1) |
| } |  |
| While( !byte\_aligned( ) ) |  |
| **bit\_equal\_to\_zero** /\* equal to 0 \*/ | f(1) |
| for( i = 0; i < MaxNumSubLayersMinus1; i++ ) { |  |
| **~~sub\_layer\_profile\_present\_flag~~**~~[ i ]~~ | ~~u(1)~~ |
| **~~sub\_layer\_level\_present\_flag~~**~~[ i ]~~ | ~~u(1)~~ |
| if( ProfilePresentFlag && sub\_layer\_profile\_present\_flag[ i ] ) { |  |
| **sub\_layer\_profile\_space**[ i ] | u(2) |
| **sub\_layer\_tier\_flag**[ i ] | u(1) |
| **sub\_layer\_profile\_idc**[ i ] | u(5) |
| for( j = 0; j < 32; j++ ) |  |
| **sub\_layer\_profile\_compatibility\_flag**[ i ][ j ] | u(1) |
| **sub\_layer\_reserved\_zero\_16bits**[ i ] | u(16) |
| } |  |
| if( sub\_layer\_level\_present\_flag[ i ] ) |  |
| **sub\_layer\_level\_idc**[ i ] | u(8) |
| } |  |
| } |  |

## Option 2

The second option is further to separate sub-layer profile present flags and level present flags besides that separation of those present flags and its information. In this option, sub-layer profile present flag is only signaled when ProfilePresentFlag is equal to 1. Whole part is byte aligned with inserting padding bits (bit\_equal\_to\_zero).

Table 3 shows proposed profile\_tier\_level() syntax where the changes are highlighted in yellow. Semantics of sub\_layer\_profile\_present\_flag[ i ] is changed as follows.

**sub\_layer\_profile\_present\_flag**[ i ] equal to 1, when ProfilePresentFlag is equal to 1, specifies that profile information is present in the profile\_tier\_level( ) syntax structure for the representation of the sub-layer with TemporalId equal to i. sub\_layer\_profile\_present\_flag[ i ] equal to 0 specifies that profile information is not present in the profile\_tier\_level( ) syntax structure for the representations of the sub-layer with TemporalId equal to i. When ProfilePresentFlag is equal to 0, sub\_layer\_profile\_present\_flag[ i ] is inferred to be equal to 0.

Table : Proposed Profile, tier, level syntax on Option 2

|  |  |
| --- | --- |
| profile\_tier\_level( ProfilePresentFlag, MaxNumSubLayersMinus1 ) { | **Descriptor** |
| if( ProfilePresentFlag ) { |  |
| **general\_profile\_space** | u(2) |
| **general\_tier\_flag** | u(1) |
| **general\_profile\_idc** | u(5) |
| for( i = 0; i < 32; i++ ) |  |
| **general\_profile\_compatibility\_flag**[ i ] | u(1) |
| **general\_reserved\_zero\_16bits** | u(16) |
| } |  |
| **general\_level\_idc** | u(8) |
| if( ProfilePresentFlag ){ |  |
| for( i = 0; i < MaxNumSubLayersMinus1; i++ ) { |  |
| **sub\_layer\_profile\_present\_flag[ i ]** | u(1) |
| } |  |
| While( !byte\_aligned( ) ) |  |
| **bit\_equal\_to\_zero** /\* equal to 0 \*/ | f(1) |
| } |  |
| for( i = 0; i < MaxNumSubLayersMinus1; i++ ) { |  |
| **sub\_layer\_level\_present\_flag[ i ]** | u(1) |
| } |  |
| While( !byte\_aligned( ) ) |  |
| **bit\_equal\_to\_zero** /\* equal to 0 \*/ | f(1) |
| for( i = 0; i < MaxNumSubLayersMinus1; i++ ) { |  |
| **~~sub\_layer\_profile\_present\_flag~~**~~[ i ]~~ | ~~u(1)~~ |
| **~~sub\_layer\_level\_present\_flag~~**~~[ i ]~~ | ~~u(1)~~ |
| if( ProfilePresentFlag && sub\_layer\_profile\_present\_flag[ i ] ) { |  |
| **sub\_layer\_profile\_space**[ i ] | u(2) |
| **sub\_layer\_tier\_flag**[ i ] | u(1) |
| **sub\_layer\_profile\_idc**[ i ] | u(5) |
| for( j = 0; j < 32; j++ ) |  |
| **sub\_layer\_profile\_compatibility\_flag**[ i ][ j ] | u(1) |
| **sub\_layer\_reserved\_zero\_16bits**[ i ] | u(16) |
| } |  |
| if( sub\_layer\_level\_present\_flag[ i ] ) |  |
| **sub\_layer\_level\_idc**[ i ] | u(8) |
| } |  |
| } |  |
|  |  |

# Conclusion

This contribution presents the improved profile\_tier\_level() syntax structure, which is more decoder-friendly and development. We recommend one of the proposed options to be adopted as a part of the next HM.

# References

[1] B. Bross, W-J Han, J-R Ohm, G. J. Sullivan, and T. Wiegand, “High efficiency video coding (HEVC) text specification draft 8,” JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 10th Meeting: Stockholm, SE, July, 2012

[2] Y-K Wang, “Solutions considered for NAL unit header and video parameter set for HEVC extensions,” JCTVC-J1007, JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 10th Meeting: Stockholm, SE, July, 2012

# Patent rights declaration(s)

**Sharp Corporation 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).**