|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  21st Meeting: Warsaw, PL, 19–26 June 2015 | Document: JCTVC-U0099 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Modified derivation of NumPaletteIndices** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Rajan Joshi, Wei Pu, Marta Karczewicz, Fend Zou, Vadim Seregin  5775 Morehouse Drive San Diego, CA 92121, USA | Tel: Email: | 1-858-658-4511 [rajanj@qti.qualcomm.com](mailto:rajanj@qti.qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

It is asserted that there is a mismatch between HEVC SCC text specification version 3 and SCM 4.0 software in the derivation of NumPaletteIndices. Furthermore a modification of the derivation of NumPaletteIndices is proposed. This modification removes one condition check and associated operations. Simulation results are provided to show that there is virtually no BD-rate change due to this modification.

# Introduction

In the HEVC SCC draft text specification version 3 (JCTVC-T1005-v2), the number of palette indices in the block (NumPaletteIndices) is coded using the syntax element **num\_palette\_indices\_idc**. The number of palette indices in the block is derived as:

if( num\_palette\_indices\_idc >= ( MaxPaletteIndex – 1) \* 32 )  
 NumPaletteIndices = num\_palette\_indices\_idc + 1  
else if(num\_palette\_indices\_idc % 32 = = 31)  
 NumPaletteIndices = MaxPaletteIndex – (num\_palette\_indices\_idc + 1 ) / 32 (7‑79)  
else  
 NumPaletteIndices = ( num\_palette\_indices\_idc / 32 ) \* 31 ) + (num\_palette\_indices\_idc % 32) + MaxPaletteIndex

There is a mismatch between SCM 4.0 software and specification text. This is because, in SCM 4.0, the variable uiIndexMaxSize corresponds to (MaxPaletteIndex+1). Presently, the text specification and software are identical if uiIndexMaxSize corresponds to MaxPaletteIndex. Thus, the current specification should be fixed to match with the software. Furthermore, the upper branch of the if statement can be removed without impacting the BD-rate performance.

# Proposal

## Draft text specification and software mismatch

The mismatch between draft text specification and software can be removed by replacing MaxPaletteIndex with (MaxPaletteIndex+1).

if( num\_palette\_indices\_idc >= MaxPaletteIndex \* 32   
 NumPaletteIndices = num\_palette\_indices\_idc + 1  
else if(num\_palette\_indices\_idc % 32 = = 31)  
 NumPaletteIndices = MaxPaletteIndex + 1– (num\_palette\_indices\_idc + 1 ) / 32 (7‑79)  
else  
 NumPaletteIndices = ( num\_palette\_indices\_idc / 32 ) \* 31 ) + (num\_palette\_indices\_idc % 32) + MaxPaletteIndex + 1

and

cRiceParam = 2 + (MaxPaletteIndex + 1) / 6 (9‑30)

## Simplification of calculation of NumPaletteIndices

On the encoder side, num\_palette\_indices\_idc is derived as follows. First, (MaxPaletteIndex + 1) is subtracted from NumPaletteIndices to get mappedNumPaletteIndices. If it is negative (−n, n >= 0), num\_palette\_indices\_idc is set to (32\*n – 1). If mappedNumPaletteIndices is positive, num\_palette\_indices\_idc is derived by shifting mappedNumPaletteIndices appropriately to account for the negative values mapped to positive integers. Since NumPaletteIndices can not be less than 1, the offset version can not be less than (−MaxPaletteIndex). Thus, the maximum upward shift is restricted to MaxPaletteIndex. The first branch in the if statement relates to this. This condition occurs very rarely and may not change the number of bits needed for binarization of num\_palette\_indices\_idc.

Instead, we propose to get rid of the first branch to simplify the derivation of NumPaletteIndices to

~~if( num\_palette\_indices\_idc >= MaxPaletteIndex \* 32   
 NumPaletteIndices = num\_palette\_indices\_idc + 1  
else~~ if(num\_palette\_indices\_idc % 32 = = 31)  
 NumPaletteIndices = MaxPaletteIndex + 1– (num\_palette\_indices\_idc + 1 ) / 32 (7‑79)  
else  
 NumPaletteIndices = ( num\_palette\_indices\_idc / 32 ) \* 31 ) + (num\_palette\_indices\_idc % 32) + MaxPaletteIndex + 1

# Simulation results

Simulations are carried out under the SCC common test conditions (JCTVC-T1015) on a LINUX cluster. The encoding and decoding times are not reliable. Only All-Intra lossy results are presented in this section. The complete test results can be found in the proposal package. Tables 1 and 2 show the BD-rate results for the proposed simplification for the All-Intra configuration for full-frame IBC and 1×4 CTU IBC, respectively. The anchor is SCM 4.0.

Table 1: BD-rate results for the proposed simplification, full-frame IBC, All-Intra configuration, anchor SCM 4.0



Table 2: BD-rate results for the proposed simplification, 1×4 CTU IBC, All-Intra configuration, anchor SCM 4.0



# Conclusion

There is a mismatch between HEVC SCC text specification version 3 and SCM 4.0 software in the derivation of NumPaletteIndices. It is recommended that the text be amended to match the software. A simplification of the derivation to calculate NumPaletteIndices from **num\_palette\_indices\_idc** is proposed. The simplification removes one branch from the if-else if-else group in the derivation process. Simulation results show that there is virtually no BD-rate change due to this modification.

# Patent rights declaration(s)

**Qualcomm Incorporated 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).**