|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  17th Meeting: Valencia, ES, 27 March – 4 April 2014 | Document: JCTVC-Q0131-r1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **AHG18: Limiting the worst-case length for coeff\_abs\_level\_remaining syntax element to 32 bits** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal, Information | | |
| *Author(s) or Contact(s):* | Marta Karczewicz and Rajan Joshi 5775 Morehouse Drive San Diego, CA 92121-1714 USA | Tel: Email: | +1-858-658-5673 martak@qti.qualcomm.com  rajanj@qti.qualcomm.com |
| *Source:* | Qualcomm, Inc. | | |

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

# Abstract

This contribution proposes a modification of the binarization for the coeff\_abs\_level\_remaining syntax element. The number of prefix bits is restricted depending upon *MAX\_TR\_DYNAMIC\_RANGE* and a truncated unary representation is used for the prefix. The suffix bits are suitably modified for the highest suffix. It is asserted that with this modification, the length of the coeff\_abs\_level\_remaining syntax in the worst-case is limited to 32. The impact on BD-rate performance is reported to be negligible (less than 0.02%) for AHG18 common test conditions.

# Introduction

In HEVC main and main10 profiles, the length of the coeff\_abs\_level\_remaining syntax element is restricted to 32 bits in the worst-case. This is considered desirable for implementation in hardware as well as in software. In HEVC, irrespective of the bit-depth a video component, the transform coefficients are restricted to be in a 16-bit range (from -215 to (215-1), inclusive). Taking the significance map into account, the largest possible value for coeff\_abs\_level\_remaining syntax element is (215-1).

However in the current HEVC range extensions specification, the range of transform coefficients when using extended precision is -2(B+6) to 2(B+6)-1, inclusive, where B is the bit-depth a video component. For example, when using extended precision, for a 16-bit video component, the range of transform coefficients is -222 to 222-1, inclusive. In this case, the Golomb-Rice/Exponential-Golomb method used for coding coeff\_abs\_level\_remaining syntax element produces worst-case length of 46 bits, which occurs when cRiceParam is 0.

JCTVC-P0061 proposed a method to restrict the worst-case length for coeff\_abs\_level\_remaining syntax element to 32 bits. But an additional table was needed and the binarization was a bit different from the existing HEVC binarization.

# Proposed Method

In the current HEVC range extension specification, *MAX\_TR\_DYNAMIC\_RANGE* for a video component is defined as

*MAX\_TR\_DYNAMIC\_RANGE* = *Max(15, B+6)*, where *B* is the bit-depth of a video component.

Let

*max\_suffix\_length* = *MAX\_TR\_DYNAMIC\_RANGE*

*MAX\_TR\_DYNAMIC\_RANGE* also represents the maximum number of bits needed to represent the value of a coeff\_abs\_level\_remaining syntax element that may be present in a conforming bit-stream. This takes into account the fact that the significance value for a coefficient is sent separately. Also, let

*max\_prefix\_length = 32 – max\_suffix\_length*.

Thus, for 16-bit video component, *max\_suffix\_length* is 22 and *max\_prefix\_length* is 10. We propose the use of truncated unary representation for representing the prefix, where the maximum prefix length is *max\_prefix\_bits*. For the highest prefix (represented by 1111 …), the suffix length is set to *max\_suffix\_bits*. Table 1 and 2 show an examples of the binarization for video component bit-depth = 16.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input Value | Codeword prefix | Codeword suffix | Prefix code length | Suffix code length | Total codeword length |
| 0 | 0 |  | 1 | 0 | 1 |
| 1 | 10 |  | 2 | 0 | 2 |
| 2 | 110 |  | 3 | 0 | 3 |
| 3 | 1110 |  | 4 | 0 | 4 |
| [4, 5] | 11110 | x | 5 | 1 | 6 |
| [6, 9] | 111110 | xx | 6 | 2 | 8 |
| [10, 17] | 1111110 | xxx | 7 | 3 | 10 |
| [18, 33] | 11111110 | xxxx | 8 | 4 | 12 |
| [34, 65] | 111111110 | xxxxx | 9 | 5 | 14 |
| [66, 129] | 1111111110 | xxxxxx | 10 | 6 | 16 |
| [130, 4194433] | 1111111111 | xxx … 22 times | 10 | 22 | 32 |

Table 1: Proposed binarization for bit-depth = 16, cRiceParam = 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input Value | Codeword prefix | Codeword suffix | Prefix code length | Suffix code length | Total codeword length |
| [0, 3] | 0 | xx | 1 | 2 | 3 |
| [4, 7] | 10 | xx | 2 | 2 | 4 |
| [8, 11] | 110 | xx | 3 | 2 | 5 |
| [12, 15] | 1110 | xx | 4 | 2 | 6 |
| [16, 23] | 11110 | xxx | 5 | 3 | 8 |
| [24, 39] | 111110 | xxxx | 6 | 4 | 10 |
| [40, 71] | 1111110 | xxxxx | 7 | 5 | 12 |
| [72, 135] | 11111110 | xxxxxx | 8 | 6 | 14 |
| [136, 263] | 111111110 | xxxxxxx | 9 | 7 | 16 |
| [264, 519] | 1111111110 | xxxxxxxx | 10 | 8 | 18 |
| [520, 4194823] | 1111111111 | xxx … 22 times | 10 | 22 | 32 |

Table 2: Proposed binarization for bit-depth = 16, cRiceParam = 2

# Simulation Results

The proposal was implemented on top of HM13.0\_RExt6.0. Simulations are performed under AHG18, test conditions. The performance is compared to the anchor in terms of BD-rate savings. The simulations were performed on a LINUX cluster (SUSE 10.4) with gcc v4.1.2 (64 bit). The CPUs were Intel Xeon E5 2680 16-core CPUs (2.7GHz) with 64 GB (shared) memory for a cluster node. Table 3 shows the BD-rate performance under AHG18 lossless test conditions. Table 4 shows the BD-rate performance under AHG18 lossy test conditions.



Table 3: Bit rate changes for the proposed method (anchor: HM13.0\_RExt6.0) under AHG18 lossless test conditions





Table 4: BD-rate for the proposed method (anchor: HM13.0\_RExt6.0) under AHG18 lossy test conditions

# Conclusions

A modification to the binarization for the coeff\_abs\_level\_remaining syntax element is proposed. The number of prefix bits is restricted depending upon *MAX\_TR\_DYNAMIC\_RANGE* and a truncated unary representation is used for the prefix. The suffix bits are suitably modified for the highest suffix to accommodate all conforming values of the coeff\_abs\_level\_remaining syntax element. It is asserted that with this modification, the length of the coeff\_abs\_level\_remaining syntax in the worst-case is limited to 32. The impact on BD-rate performance is reported to be negligible (less than 0.02%) for AHG18 common test conditions.

# Draft Text Specification

The draft text is the same as in Q0073, which proposes exactly the same idea.

(Based on JCTVC-O1005)

**9.3.3.X Limited exponential-Golomb-order-k binarization process**

Input to this process is a request for a limited exponential-Golomb binarization for a syntax element with unsigned value synVal, the Rice parameter riceParam and a flag isChroma indicating whether or not the current syntax element relates to a chroma coefficient.

Output of this process is the limited exponential-Golomb-order-k binarization of the syntax element synVal.

The variable maxTrDynamicRange is derived as follows:

– If isChroma is equal to 0,

maxTrDynamicRange = extended\_precision\_processing\_flag ? Max( 15, BitDepthY + 6 ) : 15

– Otherwise,

maxTrDynamicRange = extended\_precision\_processing\_flag ? Max( 15, BitDepthC + 6 ) : 15

The variable maximumPrefixLength is set equal to (29 – maxTrDynamicRange).

The bin string of the limited exponential-Golomb-order-k binarization process of a syntax element synVal is specified as follows, where each call of the function put( X ), with X being equal to 0 or 1, adds the binary value X at the end of the bin string:

codeValue = synVal >> riceParam  
prefixLength = 0

while( (prefixLength < maximumPrefixLength) && (codeValue > ( (2 << prefixLength) – 2) ) )  
{  
 prefixLength++  
 put(1)  
}

if (prefixLength = = maximumPrefixLength)  
{  
 totalSuffixLength = maxTrDynamicRange  
 put(1)  
}  
else  
{  
 totalSuffixLength = prefixLength + riceParam  
 put(0)  
}

synVal = synVal – ( ( (1 << prefixLength) – 1) << riceParam)

while ( (totalSuffixLength – –) > 0)  
{  
 put( (synVal >> totalSuffixLength) & 1)  
}

#### 9.3.3.9 Binarization process for coeff\_abs\_level\_remaining

**…**

The variable cMax is derived from cRiceParam as:

cMax = 43  <<  cRiceParam

The binarization of the syntax element coeff\_abs\_level\_remaining[ n ] is a concatenation of a prefix bin string and (when present) a suffix bin string.

For the derivation of the prefix bin string, the following applies:

* The prefix value of cu\_qp\_delta\_abs, prefixVal, is derived as follows:

prefixVal = Min( cMax, coeff\_abs\_level\_remaining[ n ] )

* The prefix bin string is specified by invoking the TR binarization process as specified in subclause 9.3.3.2 for prefixVal with the variables cMax and cRiceParam as inputs.

When the prefix bin string is equal to the bit string of length 43 with all bits equal to 1, the suffix bin string is present and it is derived as follows:

* The suffix value of cu\_qp\_delta\_abs, suffixVal, is derived as follows:

suffixVal = coeff\_abs\_level\_remaining[ n ] – cMax

* When extended\_precision\_processing\_flag is equal to 1, the suffix bin string is specified by invoking the limited exponential-Golomb-order-k binarization process as specified in subclause 9.3.3.X for suffixVal with the variable isChroma set equal to 1 if cIdx is greater than 0 and 0 otherwise.
* Otherwise (extended\_precision\_processing\_flag is equal to 0), the suffix bin string is specified by invoking the EGk binarization process as specified in subclause 9.3.3.3 for suffixVal with the Exp-Golomb order k set equal to cRiceParam~~+ 1~~.

#### 9.3.3.9 Binarization process for coeff\_abs\_level\_remaining

**…**

The variable cMax is derived from cRiceParam as:

cMax = 43  <<  cRiceParam

The binarization of the syntax element coeff\_abs\_level\_remaining[ n ] is a concatenation of a prefix bin string and (when present) a suffix bin string.

For the derivation of the prefix bin string, the following applies:

* The prefix value of cu\_qp\_delta\_abs, prefixVal, is derived as follows:

prefixVal = Min( cMax, coeff\_abs\_level\_remaining[ n ] )

* The prefix bin string is specified by invoking the TR binarization process as specified in subclause 9.3.3.2 for prefixVal with the variables cMax and cRiceParam as inputs.

When the prefix bin string is equal to the bit string of length 43 with all bits equal to 1, the suffix bin string is present and it is derived as follows:

* The suffix value of cu\_qp\_delta\_abs, suffixVal, is derived as follows:

suffixVal = coeff\_abs\_level\_remaining[ n ] – cMax

* When extended\_precision\_processing\_flag is equal to 1, the suffix bin string is specified by invoking the limited exponential-Golomb-order-k binarization process as specified in subclause 9.3.3.X for suffixVal with the variable isChroma set equal to 1 if cIdx is greater than 0 and 0 otherwise.
* Otherwise (extended\_precision\_processing\_flag is equal to 0), the suffix bin string is specified by invoking the EGk binarization process as specified in subclause 9.3.3.3 for suffixVal with the Exp-Golomb order k set equal to cRiceParam~~+ 1~~.

# 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).**