|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  7th Meeting: Geneva, 21-30 Nov, 2011 | Document: JCTVC-G152  WG11 Number: |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Title:* | **Method and syntax for quantization matrices representation** | | | |
| *Status:* | Input Document to JCT-VC | | | |
| *Purpose:* | Proposal | | | |
| *Author(s) or Contact(s):* | Ximin Zhang  Shan Liu Shawmin Lei  2860 Junction Ave.  San Jose, CA95134, USA | *Emails:* | ximin.zhang@mediatek.com shan.liu@mediatek.com shawmin.lei@mediatek.com |
| *Source:* | MediaTek USA Inc. | | | |

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

# Abstract

This contribution proposes method and syntax for encoding quantization matrices. Specifically, an implicit method is proposed to generate 16x4, 4x16, 16x16, 32x8, 8x32 and 32x32 quantization matrices. By doing this, no entropy coding is needed to compress the large quantization matrices and non-square quantization matrices. The proposed method is implemented in HM4.0 CE4 software. Experimental results report average 14.7% BD rate gain for All Intra, 44.1% BD rate gain for Random Access, 56.4% BD rate gain for Low Delay B and 55.6% BD rate gain for Low Delay P compared to AVC quantization matrices encoding method. The average decoder complexity is reduced by around 2%.

1. Introduction

Quantization matrix has been successfully utilized in previous video coding standard MEPG2 and H.264/AVC to improve the subjective video quality. Due to its effectiveness, the quantization matrix has been widely used in numerous commercial products. It is natural to adopt it in HEVC.

In MEPG2 and H.264/AVC, default quantization matrix is defined. The encoder can decide if the current picture will use flat quantization matrix, default quantization matrix or user defined quantization matrix. If user defined quantization matrix is selected, it will be coded and send to the decoder in the bitstream. In MEPG2 and H.264/AVC, the maximum size of quantization matrix is 8x8. The memory requirement for default quantization matrix and the number of bits to send the user defined quantization matrix is not a big problem. In HEVC, 4x16, 16x4, 16x16, 8x32, 32x8, 32x32 transform have been adopted. The data size of the quantization matrix is increased considerably by using the traditional quantization matrix representation method as in H.264/AVC. For instance, 32 bytes are used to store two (Intra and Inter) 4x4 default quantization matrix and 128 bytes are used to store two 8x8 default quantization matrix. Using the same way to store two 32x32 default quantization matrix only will use 2048 bytes. A more efficient quantization matrix representation method is required in HEVC.

# Proposed Method

In this contribution, we propose to implicitly generate 16x4, 4x16, 16x16, 32x8, 8x32 and 32x32 quantization matrices for both default and user defined quantization matrix.

The proposed method can be conducted as the following two steps for 16x16 and 32x32 matrices.

1. Mapped the coefficients of the smaller available quantization matrix to the coefficients of the bigger quantization matrix with similar frequency.
2. Generate the other coefficients with interpolation from the mapped coefficients from step 1.

In one example embodiment, the 16x16 quantization matrix is generated from 8x8 quantization matrix. In the first step, the coefficients of the 8x8 quantization matrix is mapped to the coefficients of the 16x16 quantization matrix as equation (1), (2), (3) and (4). In the following equations, “*t*” is a constant and its value can be 1 to 7. In the illustrated example, the 16x16 quantization matrix is generated from default AVC 8x8 quantization matrix with *t*=1.

(1)

(2)

(3)

(4)

After the mapping, the other coefficients are generated by equation (5) and (6) if their immediate left and right neighbors, or immediate upper and lower neighbors are mapped coefficients.

(5)

(6)

For the coefficients whose immediate left or right neighbors, and immediate upper or lower neighbors are not mapped coefficients, the process as in equations (7), (8), (9) and (10) is applied.

(7)

(8)

(9)

(10)

**AVC Default INTRA 8x8 matrix**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 10 | 13 | 16 | 18 | 23 | 25 | 27 |
| 10 | 11 | 16 | 18 | 23 | 25 | 27 | 29 |
| 13 | 16 | 18 | 23 | 25 | 27 | 29 | 31 |
| 16 | 18 | 23 | 25 | 27 | 29 | 31 | 33 |
| 18 | 23 | 25 | 27 | 29 | 31 | 33 | 36 |
| 23 | 25 | 27 | 29 | 31 | 33 | 36 | 38 |
| 25 | 27 | 29 | 31 | 33 | 36 | 38 | 40 |
| 27 | 29 | 31 | 33 | 36 | 38 | 40 | 42 |

**Generated INTRA 16X16 matrix from INTRA 8X8 matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 8 | 10 | 11 | 12 | 13 | 15 | 16 | 17 | 18 | 21 | 23 | 24 | 25 | 26 | 27 |
| 8 | 10 | 11 | 12 | 13 | 15 | 16 | 17 | 19 | 21 | 23 | 24 | 25 | 26 | 27 | 28 |
| 10 | 11 | 11 | 12 | 14 | 16 | 17 | 18 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 11 | 12 | 12 | 13 | 14 | 16 | 18 | 19 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 12 | 13 | 14 | 15 | 16 | 17 | 19 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 13 | 15 | 16 | 16 | 17 | 18 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 15 | 16 | 17 | 18 | 19 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 16 | 17 | 18 | 19 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
| 17 | 19 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 34 | 35 |
| 18 | 21 | 23 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 35 | 36 |
| 21 | 23 | 24 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 34 | 35 | 36 | 37 |
| 23 | 24 | 25 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 35 | 36 | 37 | 38 |
| 24 | 25 | 26 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 34 | 35 | 36 | 37 | 38 | 39 |
| 25 | 26 | 27 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 35 | 36 | 37 | 38 | 39 | 40 |
| 26 | 27 | 28 | 28 | 29 | 30 | 31 | 32 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 |
| 27 | 28 | 29 | 29 | 30 | 31 | 32 | 33 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 |

In this proposal, the rectangular quantization matrix is proposed to implicitly generated by coefficients mapping from the square quantization matrix. For instance, 4x16, 16x4 quantization matrix can be obtained by coefficients mapping from the 16x16 quantization matrix; 8x32, 32x8 quantization matrix can be obtained by coefficients mapping from the 32x32 quantization matrix.

The following example shows how to generate the 4x16 and 16x4 quantization matrix from the 16x16 quantization matrix. The four red highlighted columns are used as the 4x16 quantization matrix and the four yellow highlighted rows are used as the 16x4 quantization matrix.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 8 | 10 | 11 | 12 | 13 | 15 | 16 | 17 | 18 | 21 | 23 | 24 | 25 | 26 | 27 |
| 8 | 10 | 11 | 12 | 13 | 15 | 16 | 17 | 19 | 21 | 23 | 24 | 25 | 26 | 27 | 28 |
| 10 | 11 | 11 | 12 | 14 | 16 | 17 | 18 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 11 | 12 | 12 | 13 | 14 | 16 | 18 | 19 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 12 | 13 | 14 | 15 | 16 | 17 | 19 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 13 | 15 | 16 | 16 | 17 | 18 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 15 | 16 | 17 | 18 | 19 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 16 | 17 | 18 | 19 | 21 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
| 17 | 19 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 34 | 35 |
| 18 | 21 | 23 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 35 | 36 |
| 21 | 23 | 24 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 34 | 35 | 36 | 37 |
| 23 | 24 | 25 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 35 | 36 | 37 | 38 |
| 24 | 25 | 26 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 34 | 35 | 36 | 37 | 38 | 39 |
| 25 | 26 | 27 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 35 | 36 | 37 | 38 | 39 | 40 |
| 26 | 27 | 28 | 28 | 29 | 30 | 31 | 32 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 |
| 27 | 28 | 29 | 29 | 30 | 31 | 32 | 33 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 |

In general, the 4x16 quantization matrix is mapped from the first, 5th,10th and 15th column and the 16x4 quantization matrix is mapped from the first, 5th,10th and 15th row of the corresponding 16x16 matrix. The 8x32 quantization matrix is mapped from the first, 4th, 8th , 12th, 16th, 21 th, 26th and 31th column and the 32x8 quantization matrix is mapped from the first, 4th, 8th , 12th, 16th, 21 th, 26th and 31th row of the corresponding 32x32 matrix.

# Proposed Syntax

In this proposal, a set of syntax is proposed to added in SPS (sequence parameter set RBSP syntax), PPS (picture parameter set RBSP syntax) and APS (adaptation parameter set RBSP syntax) to signal the quantization matrix to the decoder.

In sequence level, “**seq\_quant\_matrix\_present\_flag**” is used in SPS to signal if the whole sequence will use quantization matrix or not. If it is true, “**seq\_quant\_matrix\_present\_idx**” is checked for each transform size and each channel (Y,U,V) to signal if the specified transform size and each channel will use quantization matrix or not.If it is true and the transform size is 4x4 or 8x8, entropy coding is used for user defined quantization matrix. Otherwise, “**implicit\_quant\_matrix\_present\_idx**” is checked to signal if the rectangle quantization matrix and the bigger quantization matrix will be implicitly generated from the 4x4 and 8x8 quantization matrix. If “**implicit\_quant\_matrix\_present\_idx**” is not true, entropy coding is used for user defined quantization matrix.

In picture level, “**pic\_quant\_matrix\_present\_flag**” is used in PPS to signal if the current picture will use quantization matrix or not. If “**pic\_quant\_matrix\_present\_flag**” is true, “**pic\_quant\_matrix\_present\_idx**” is checked for each transform size and each channel (Y,U,V) to signal if the specified transform size and each channel will use quantization matrix or not.If it is true and the transform size is 4x4 or 8x8, entropy coding is used for user defined quantization matrix. Otherwise, “**implicit\_quant\_matrix\_present\_idx**” is checked to signal if the rectangle quantization matrix and the bigger quantization matrix will be implicitly generated from the 4x4 and 8x8 quantization matrix. If “**implicit\_quant\_matrix\_present\_idx**” is not true, entropy coding is used for user defined quantization matrix.

# Experimental Results

Simulations were conducted following common test conditions defined in JCTVC-F900 [1] and CE4 suggestion. Anchor data was generated using AVC method provide by CE4 HM4.0 software and the symmetrical quantization matrix. Table 1 reports the results under common test condition with the symmetrical quantization matrices provided by CE4. As the results show, xx% BD rate improvement is obtained compared to the anchor with no subjective quality degradation.

Table 1 Results for the proposed implicit quantization matrices generation method, compared with HM4.0 CE4 AVC method under common test condition.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  | **All Intra HE** | | | **All Intra LC** | | |
|  |  | Y | U | V | Y | U | V |
|  | Class A | -5.6% | -2.1% | -1.0% | -6.2% | -0.1% | 1.0% |
|  | Class B | -6.7% | -6.0% | -6.0% | -6.7% | -5.4% | -5.4% |
|  | Class C | -15.0% | -13.9% | -14.0% | -14.0% | -13.0% | -13.1% |
|  | Class D | -32.4% | -30.4% | -30.7% | -31.0% | -29.2% | -29.4% |
|  | Class E | -17.8% | -16.4% | -16.1% | -16.4% | -14.8% | -14.7% |
|  | Class F |  |  |  |  |  |  |
|  | **Overall** | -14.9% | -13.3% | -13.1% | -14.4% | -12.0% | -11.9% |
|  |  | -14.9% | -13.3% | -13.1% | -14.4% | -12.1% | -12.0% |
|  | Enc Time[%] | 99% | | | 99% | | |
|  | Dec Time[%] | 98% | | | 98% | | |
|  |  |  |  |  |  |  |  |
|  |  | **Random Access HE** | | | **Random Access LC** | | |
|  |  | Y | U | V | Y | U | V |
|  | Class A | -21.0% | -17.3% | -14.7% | -19.9% | -15.0% | -12.2% |
|  | Class B | -35.0% | -32.1% | -30.8% | -33.4% | -30.3% | -29.1% |
|  | Class C | -52.3% | -50.2% | -50.2% | -51.2% | -49.1% | -49.1% |
|  | Class D | -72.6% | -71.1% | -71.2% | -71.9% | -70.6% | -70.8% |
|  | Class E |  |  |  |  |  |  |
|  | Class F |  |  |  |  |  |  |
|  | **Overall** | -44.6% | -42.0% | -41.1% | -43.5% | -40.6% | -39.6% |
|  |  | -44.6% | -42.1% | -41.2% | -43.5% | -40.7% | -39.7% |
|  | Enc Time[%] | 100% | | | 100% | | |
|  | Dec Time[%] | 98% | | | 99% | | |
|  |  |  |  |  |  |  |  |
|  |  | **Low delay B HE** | | | **Low delay B LC** | | |
|  |  | Y | U | V | Y | U | V |
|  | Class A |  |  |  |  |  |  |
|  | Class B | -37.1% | -33.0% | -31.9% | -35.7% | -31.5% | -30.0% |
|  | Class C | -52.5% | -50.0% | -49.8% | -51.4% | -48.9% | -49.0% |
|  | Class D | -72.2% | -70.4% | -70.6% | -71.7% | -69.9% | -70.2% |
|  | Class E | -75.6% | -73.6% | -73.4% | -74.3% | -72.0% | -72.4% |
|  | Class F |  |  |  |  |  |  |
|  | **Overall** | -57.0% | -54.2% | -53.9% | -55.9% | -53.1% | -52.8% |
|  |  | -56.9% | -54.2% | -54.0% | -55.9% | -53.1% | -52.9% |
|  | Enc Time[%] | 100% | | | 100% | | |
|  | Dec Time[%] | 98% | | | 99% | | |
|  |  |  |  |  |  |  |  |
|  |  | **Low delay P HE** | | | **Low delay P LC** | | |
|  |  | Y | U | V | Y | U | V |
|  | Class A |  |  |  |  |  |  |
|  | Class B | -36.2% | -32.3% | -31.1% | -34.5% | -30.4% | -29.0% |
|  | Class C | -51.8% | -49.1% | -48.9% | -50.7% | -48.1% | -48.2% |
|  | Class D | -71.7% | -69.7% | -69.9% | -71.2% | -69.2% | -69.7% |
|  | Class E | -75.0% | -72.9% | -72.8% | -73.6% | -71.3% | -71.8% |
|  | Class F |  |  |  |  |  |  |
|  | **Overall** | -56.2% | -53.5% | -53.1% | -55.0% | -52.2% | -52.0% |
|  |  | -56.2% | -53.5% | -53.2% | -55.0% | -52.2% | -52.1% |
|  | Enc Time[%] | 100% | | | 100% | | |
|  | Dec Time[%] | 98% | | | 99% | | |
|  |  |  |  |  |  |  |  |

# Conclusions

This contribution reports the method and results for encoding 16x4, 4x16, 16x16, 32x8, 8x32 and 32x32 quantization matrices for both default and user defined scenario. By doing this, the entropy coding for the above matrices are saved. Experimental results report report average 14.7% BD rate gain for All Intra, 44.1% BD rate gain for Random Access, 56.4% BD rate gain for Low Delay B and 55.6% BD rate gain for Low Delay P compared to AVC quantization matrices encoding method. The average decoder complexity is reduced by around 2%. It is recommended to include the quantization matrix support and adopt the proposed implicit quantization matrices generation in HM.

# References

1. Frank Bossen, “Common test conditions and software reference configurations”, JCTVC-F900, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Torino, Italy, July 2011.
2. HM 4.0 Software, <http://hevc.kw.bbc.co.uk/trac/browser/tags/HM-4.0>.

# Patent rights declaration(s)

**MediaTek Inc. may 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).**

# Appendix

## Sequence parameter set RBSP syntax

|  |  |
| --- | --- |
| seq\_parameter\_set\_rbsp( ) { | Descriptor |
| **profile\_idc** | u(8) |
| **reserved\_zero\_8bits** /\* equal to 0 \***/** | u(8) |
| **level\_idc** | u(8) |
| **seq\_parameter\_set\_id** | ue(v) |
| **max\_temporal\_layers\_minus1** | u(3) |
| **pic\_width\_in\_luma\_samples** | u(16) |
| **pic\_height\_in\_luma\_samples** | u(16) |
| **bit\_depth\_luma\_minus8** | ue(v) |
| **bit\_depth\_chroma\_minus8** | ue(v) |
| **pcm\_bit\_depth\_luma\_minus1** | u(4) |
| **pcm\_bit\_depth\_chroma\_minus1** | u(4) |
| **seq\_quant\_matrix\_present\_flag** | u(1) |
| if( seq\_quant\_matrix\_present\_flag ) |  |
| for( quant\_matrix\_id = 0; quant\_matrix\_id < 36; quant\_matrix\_id++ ) { |  |
| **seq\_quant\_matrix\_present\_idx[**quant\_matrix\_id**]** | u(1) |
| if(seq\_quant\_matrix\_present\_idx**[**quant\_matrix\_id**]**) { |  |
| if(quant\_matrix\_id<12){ |  |
| quant\_matrix\_coding(quant\_matrix\_id) |  |
| }else{ |  |
| **implicit\_quant\_matrix\_present\_idx[**quant\_matrix\_id**]** | u(1) |
| if(implicit\_quant\_matrix\_present\_idx**[**quant\_matrix\_id**]**) { |  |
| quant\_matrix\_derivation(quant\_matrix\_id) |  |
| }else{ |  |
| quant\_matrix\_coding(quant\_matrix\_id) |  |
| } |  |
| } |  |
| } |  |
| } |  |
| **log2\_max\_frame\_num\_minus4** | ue(v) |
| **pic\_order\_cnt\_type** | ue(v) |
| if( pic\_order\_cnt\_type = = 0 ) |  |
| **log2\_max\_pic\_order\_cnt\_lsb\_minus4** | ue(v) |
| else if( pic\_order\_cnt\_type = = 1 ) { |  |
| **delta\_pic\_order\_always\_zero\_flag** | u(1) |
| **offset\_for\_non\_ref\_pic** | se(v) |
| **num\_ref\_frames\_in\_pic\_order\_cnt\_cycle** | ue(v) |
| for( i = 0; i < num\_ref\_frames\_in\_pic\_order\_cnt\_cycle; i++ ) |  |
| **offset\_for\_ref\_frame[ i ]** | se(v) |
| } |  |
| **max\_num\_ref\_frames** | ue(v) |
| **gaps\_in\_frame\_num\_value\_allowed\_flag** | u(1) |
| **log2\_min\_coding\_block\_size\_minus3** | ue(v) |
| **log2\_diff\_max\_min\_coding\_block\_size** | ue(v) |
| **log2\_min\_transform\_block\_size\_minus2** | ue(v) |
| **log2\_diff\_max\_min\_transform\_block\_size** | ue(v) |
| **log2\_min\_pcm\_coding\_block\_size\_minus3** | ue(v) |
| **max\_transform\_hierarchy\_depth\_inter** | ue(v) |
| **max\_transform\_hierarchy\_depth\_intra** | ue(v) |
| **chroma\_pred\_from\_luma\_enabled\_flag** | u(1) |
| **loop\_filter\_across\_slice\_flag** | u(1) |
| **sample\_adaptive\_offset\_enabled\_flag** | u(1) |
| **adaptive\_loop\_filter\_enabled\_flag** | u(1) |
| **pcm\_loop\_filter\_disable\_flag** | u(1) |
| **cu\_qp\_delta\_enabled\_flag** | u(1) |
| **temporal\_id\_nesting\_flag** | u(1) |
| rbsp\_trailing\_bits( ) |  |
| } |  |

## Picture parameter set RBSP syntax

|  |  |
| --- | --- |
| pic\_parameter\_set\_rbsp( ) { | Descriptor |
| **pic\_parameter\_set\_id** | ue(v) |
| **seq\_parameter\_set\_id** | ue(v) |
| **entropy\_coding\_mode\_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) |
| **pic\_init\_qp\_minus26** **/**\* relative to 26 \*/ | se(v) |
| **constrained\_intra\_pred\_flag** | u(1) |
| **slice\_granularity** | u(2) |
| **pic\_quant\_matrix\_present\_flag** | u(1) |
| if( pic\_quant\_matrix\_present\_flag ) |  |
| for( quant\_matrix\_id = 0; quant\_matrix\_id < 36; quant\_matrix\_id++ ) { |  |
| **pic\_quant\_matrix\_present\_idx[**quant\_matrix\_id**]** | u(1) |
| if(pic\_quant\_matrix\_present\_idx**[**quant\_matrix\_id**]**) { |  |
| if(quant\_matrix\_id<12){ |  |
| quant\_matrix\_coding(quant\_matrix\_id) |  |
| }else{ |  |
| **implicit\_quant\_matrix\_present\_idx[**quant\_matrix\_id**]** | u(1) |
| if(implicit\_quant\_matrix\_present\_idx**[**quant\_matrix\_id**]**) { |  |
| quant\_matrix\_derivation(quant\_matrix\_id) |  |
| }else{ |  |
| quant\_matrix\_coding(quant\_matrix\_id) |  |
| } |  |
| } |  |
| } |  |
| } |  |
| **shared\_pps\_info\_enabled\_flag** | u(1) |
| if( shared\_pps\_info\_enabled\_flag ) |  |
| if( adaptive\_loop\_filter\_enabled\_flag ) |  |
| alf\_param( ) |  |
| if( cu\_qp\_delta\_enabled\_flag ) |  |
| **max\_cu\_qp\_delta\_depth** | u(4) |
| rbsp\_trailing\_bits( ) |  |
| } |  |