|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  9th Meeting: Geneva, CH, 27 April – 7 May 2012 | Document: JCTVC-I0362\_r1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Virtual line buffer model and restriction on asymmetric tile configuration** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Sanjeev Kumar, Geert Van der Auwera, Ye-Kui Wang, Muhammed Coban, Marta Karczewicz 5775 Morehouse Drive San Diego, CA 92121, USA | Email: | [sanjeevk@qualcomm.com](mailto:sanjeevk@qualcomm.com)  [geertv@qualcomm.com](mailto:geertv@qualcomm.com)  [mcoban@qualcomm.com](mailto:mcoban@qualcomm.com)  [yekuiw@qualcomm.com](mailto:yekuiw@qualcomm.com)  [martak@qualcomm.com](mailto:martak@qualcomm.com) |
| *Source:* | Qualcomm Inc. | | |

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

# Abstract

It is proposed to restrict asymmetry of tile configurations in order to reduce loop filtering (Deblocking, Sample Adaptive Offset, Adaptive Loop Filter) line buffer requirement based on a proposed Virtual loop filter line buffer model.

# Tile configurations and Loop filter line buffer

In the HEVC committee draft (CD) [1] loop filtering operations (Deblocking, Sample Adaptive Offset and Adaptive Loop Filter) can cross tile boundaries. This increases line buffer requirements in implementations with Tile raster scan order processing. Without tiles, only horizontal line buffer is needed. With tiles, vertical line buffers are also needed. Total line buffer size depends on several factors including tile configuration and can also vary from one implementation to another. However, some general observations can be made which are applicable to a wide variety of implementations with Tile raster scan order processing. Symmetric tile configurations are in general better than asymmetric tile configurations from a worst case total line buffer perspective. Among symmetric tile configurations 2x2 tiling has the worst case line buffer requirement. This contribution proposes constraints on asymmetry of tile configurations so as to allow asymmetric tile configurations without paying the penalty in terms of extra loop filtering line buffer requirement. Section 2 proposes a virtual line buffer model which enables a systematic way to assess loop filter line buffer impact of different tile configurations and to enforce appropriate constraints based on this assessment.

# Virtual line buffer model

Exact line buffer requirements are implementation dependent. However, we can model how line buffer requirement for a wide range of implementations varies with picture size and tile configuration. Our model is applicable to resource constrained hardware implementations with tile raster scan processing. All line buffer numbers are accurate only up to an implementation dependent factor (constant of proportionality). E.g. although for deblocking we need 4 luma, 2 Cb, 2Cr lines and some extra information in the horizontal line buffer, our virtual line buffer model treats this as being proportional to picture width with implementation dependent constant of proportionality.

All tile configurations can be classified into four mutually exclusive categories. The following total line buffer size estimations are in integer units of the LCU size and “=” should be interpreted as “proportional to”.

Case -1. No tiles

Horizontal line buffer size = picture\_width / LCU\_size

Vertical line buffer size = 1

Total line buffer size = 1 + picture\_width / LCU\_size

Case-2. Only horizontal tiles

Num\_tile\_columns = 1

Num\_tile\_rows > 1

Horizontal line buffer size = picture\_width / LCU\_size

Vertical line buffer size = 1

Total line buffer size = 1 + picture\_width / LCU\_size

Case-3. Only vertical tiles

Num\_tile\_columns > 1

Num\_tile\_rows = 1

Horizontal line buffer size = max\_tile\_width / LCU\_size (same horizontal line buffer memory can be reused for different tiles)

Vertical line buffer size = 1 + picture\_height / LCU\_size

Total line buffer size = 1 + (max\_tile\_width + picture\_height) / LCU\_size (1)

Case-4. Both horizontal and vertical tiles

Num\_tile\_columns > 1

Num\_tile\_rows > 1

Horizontal line buffer size = picture\_width / LCU\_size (case-3 like re-use is not possible)

Vertical line buffer size = 1 + max\_tile\_height / LCU\_size (same vertical line buffer memory can be reused for different tiles)

Total line buffer size = 1 + (picture\_width + max\_tile\_height) / LCU\_size (2)

Next section proposes constraints on asymmetry of tile configurations based on virtual line buffer model to eliminate extra line buffer penalty of asymmetric tile configurations compared to symmetric tile configurations.

# Proposed restriction on asymmetry of tile configuration

Based on 4 cases in the virtual line buffer model, it can be seen that for symmetric tile configurations, if picture\_width >= picture\_height, worst case happens for the 2x2 tile configuration, when Total virtual line buffer size = picture\_width / LCU\_size + 1 + picture\_height / (2 \* LCU\_size). If picture\_width < picture\_height, worst case happens for the 2x1 tile configuration (2 vertical tiles), when Total virtual line buffer size = picture\_width / (2 \* LCU\_size ) + 1 + picture\_height / LCU\_size. Expression for total virtual line buffer size for these two cases (picture\_width >= picture\_heigth and picture\_width < picture\_height) can be unified as, Total virtual line buffer size = max(picture\_width,picture\_heigth)/LCU\_size + min(picture\_width,picture\_height)/(2\*LCU\_size) + 1.

However, for asymmetric tile configurations, total line buffer size can be as large as (picture\_width + picture\_height) / LCU\_size. This proposal intends to improve upon this worst case to match that for symmetric tile configurations.

Worst case virtual loop filter line buffer requirement for asymmetric tile configuration can exceed that for symmetric tile configuration only for Case-3 and Case-4. These cases need to be handled separately using equations (1) and (2). In these equations, if we substitute total\_line\_buffer\_size <= max(picture\_width, picture\_height)/LCU\_size + min(picture\_width,picture\_height)/(2\*LCU\_size) + 1, we can find the following equivalent constraints on the max\_tile\_width and max\_tile\_height, respectively:

Case-3:

max\_tile\_width / LCU\_size

<=

max(picture\_width, picture\_heigth)/LCU\_size + min(picture\_width,picture\_height)/(2\*LCU\_size) – picture\_heigth/LCU\_size

For picture sizes where right hand side of above constraint becomes less than 384/LCU\_size, then there is a conflict with a main profile constraint of minimum\_tile\_width >= 384 luma samples. To resolve this conflict, we modify above constraint to the following form.

384/LCU\_size

<=

max\_tile\_width / LCU\_size

<=

max( 384/LCU\_size, max(picture\_width, picture\_heigth)/LCU\_size + min(picture\_width,picture\_height)/(2\*LCU\_size) – picture\_heigth/LCU\_size )

Case-4:

max\_tile\_height / LCU\_size

<=

max(picture\_width, picture\_heigth)/LCU\_size + min(picture\_width,picture\_height)/(2\*LCU\_size) – picture\_width/LCU\_size

# Line buffer savings

Considering only Deblocking and only pixel line buffers (4 luma, 2Cb, 2Cr) for 8 bits/pixels for 4k x 2k picture, this proposal reduces worst case total loop filter line buffer by 6KB (worst case virtual line buffer size is reduced from 4k+2k to 4k+1k). Taking into account other information that needs to be maintained in line buffer for Deblocking, Sample Adaptive Offset and Adaptive Loop Filter, total realized savings can be even bigger.

# Text change

In Section 7.4.2.1 of CD Text:

It is a requirement of bit stream conformance that if tiles\_or\_entropy\_coding\_sync\_idc = 1 and uniform\_spacing\_flag = 0, then values of row\_width[i] and column\_width[i] shall satisfy following conditions for all valid i:

If num\_tile\_columns\_minus1 > 0 and num\_tile\_rows\_minus1 = 0

384>>Log2CtbSize

<=

column\_width[i]

<=

max( 384>>Log2CtbSize, max(PicWidthInCtbs, PicHeightInCtbs) +

min(PicWidthInCtbs, PicHeightInCtbs)>>1 – PicHeightInCtbs )

If num\_tile\_columns\_minus1 > 0 and num\_tile\_rows\_minus1 > 0

row\_height[i]

<=

max(PicWidthInCtbs, PicHeightInCtbs)

+ min(PicWidthInCtbs, PicHeightInCtbs)>>1 – PicWidthInCtbs

# Conclusion

It is proposed to restrict the asymmetry of tile configuration to reduce loop filter line buffer requirement for Deblocking, Sample Adaptive Offset and Adaptive Loop Filter. For 4k x 2k picture, line buffer savings are more than 6KB.

# References

[1] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, T. Wiegand, “High efficiency video coding (HEVC) text specification draft 6,” 8th JCT-VC Meeting, San Jose, CA, USA, Feb. 2012

# Patent rights declaration

**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).**