|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  10th Meeting: Stockholm, Sweden, July 11-20, 2012 | Document: JCTVC-J0088  M25410 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | AHG4: Enable parallel decoding with tiles | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Minhua Zhou Texas Instruments Inc., USA | Tel: Email:  : | +1-214-480-3816 [zhou@ti.com](mailto:zhou@ti.com) |
| *Source:* | Texas Instruments Inc; | | |

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

# Abstract

Real-time UHD decoding can exceed the capability of a single core decoder. To enable parallel decoding on multi-core platforms, it is proposed to define companion “parallel” levels for level 5.2 and above, and mandate evenly divided sub-pictures (tiles) for those parallel levels to guarantee pixel-rate balancing among cores when sub-pictures are processed in parallel. The key points of proposal are: 1) A picture is divided into a number of sub-pictures of equal size (in units of LCUs); 2) Sub-pictures are independent, only in-loop filters are allowed to cross the sub-picture boundaries; 3) Tiles, slices, entropy slices and WPP are contained in sub-pictures and cannot cross sub-picture boundaries; 4) The sub-picture partitioning information is signaled with tile syntax. If sub-pictures are mandated, tiles have to be uniformly spaced in both directions (i.e. option 1 same degree of parallelism for encoder and decoder) or uniformly spaced in vertical direction only (i.e. option 2 to allow more parallelism on the encoder side than on the decoder side). 5) Sub-picture entries in bitstream are signaled in the APS. The proposed change guarantees pixel-rate balancing on multi-core platforms and allows building a multi-core decoder by simply replicating the single core decoder without need of increasing the line buffer size.

# Introduction

For years, clock-rate has stopped increasing even if semi-conduct process technology continues to advance; multi-core architecture is widely used to drive up the performance while keeping cost and power consumption in check. As video moving to UHD, a single core decoder won’t be capable of decoding UHD in real-time. Therefore, the HEVC should provide mechanism to enable parallel decoding on multi-core platforms.

Several parallel processing tools have been adopted into HEVC in addition to regular slices to support parallel processing for multi-core implementations. These tools include tiles, Wavefront Parallel Processing (WPP) and Entropy Slices (ES). So far the specification has focused on the parallel processing on the encoder side, aimed to enable the usage of parallel tools in encoders without imposing significant burden on single core decoders.

To make such a decoder multi-core architecture feasible, a picture would need to be divided into sub-pictures of equal size, so that pixel-rate balancing can be guaranteed when individual cores are processing sub-pictures in parallel. However, based on the current specification, such a pixel balancing is not guaranteed (i.e. equal amount of LCUs dispatched to the cores) by parallel processing tools like tiles, WPP, entropy slice and slices.

For parallel processing on multiple core platforms, the following design requirements should be considered.

* Line buffer size is very expensive. The selected parallel processing tool should have capability of dividing a picture into sub-pictures in horizontal direction to constrain line buffer size.
* Cross-core communication is a performance killer for multi-core solutions and consuming lots of power. The selected parallel processing tool should be able to minimize the cross-core data communication.
* Design validation is a significant part of design cost. The selected parallel processing tool should be able to minimize the design validation efforts for multi-core solutions once the single core is designed and validated. A desirable architecture is shown in Fig. 1, in which a 8Kx4K@30 decoder can be built by simply replicating the 4kx2k@30 single core decoder four times, and adding a sub-picture boundary processing core to perform e.g. de-blocking filter, Sample Adaptive Offset (SAO), Adaptive Loop Filter (ALF) along sub-picture boundaries.
* The selected parallel processing tool should minimize quality degradation by minimizing size of sub-picture boundaries.

**Fig. 1 – From single core to multiple core**

Among available parallel processing tool, tiles are a preferred choice to address the pixel rate balancing issue on multi-core platforms. Using the regular slices and WPP for the same purpose has the following drawbacks:

1. Regular slices: mandating regular slices of equal size (in terms of LCUs) could be a solution. However, there are use cases such as video conferencing in which a picture may create a large number of slices and each slice may contain a different number of LCUs. Therefore, it is hard to mandate slices of equal size in terms of LCUs. Also, slices are partitioned and coded in raster-scan order, which prevents from building a multi-core decoder by simply replicating single core decoders. For example, if a single core decoder is designed for e.g. 4Kx2K video, it would need to increase the single core line buffer size if cores are replicated to support e.g. 8Kx4K video.
2. WPP: for multi-core solutions, a shared MC cache is needed. Otherwise, it will incur huge memory bandwidth increase for motion compensation. Also, it needs frequent cross-core data communication for entropy decoding if sub-streams are processed on different cores in parallel. The line buffer of single core decoder needs to be increased too because WPP can be only partitioned vertically. A shared MC cache and increased line buffer would require re-design of single core decoder. Also, additional design validation is needed to ensure interconnection of cores. WPP is more appropriate for driving higher entropy decoding throughput by inserting parallel entropy decoding engines in a single core video decoder.

Fig.2 illustrates the advantages of using tiles for parallel decoding purpose. In this example, a picture is divided into 4 sub-pictures of equal size. With tiles, the picture can be divided into 2x2 sub-pictures. However, with slices or WPPs, the picture would need to be divided into 1x4 sub-pictures. Sub-picture partitioning using tiles are apparently advantageous in the follows aspects: 1) sub-picture width is only half of picture width, which cuts the line buffer size by half when compared to slice or WPPs; 2) less number of sub-picture boundaries ( 2 vs. 3 boundaries see Fig. 2) leads to better quality; 3) tiles are independent which minimizes cross-core communications, only reference picture data is shared among cores; 4) cores can be simply replicated to build multi-core solutions without need of additional design validation efforts to verify cross-core interconnections.

**Fig. 2 - Sub-picture partitioning**

# Algorithm description

To enable the multi-core architecture illustrated in Fig.1, constraints have to be imposed on the HEVC bitstreams. We propose to define companion “parallel” levels for level 5.2 and above, and mandate evenly divided sub-pictures for high levels to support parallel decoding of UHD video, and use tile syntax to signal sub-picture partitioning.

## Sub-picture partitioning

If sub-pictures are mandated for a parallel level, the sub-pictures share the following characteristics:

1. An incoming picture should be evenly divided into **num\_subPic\_columns x num\_subPic\_rows** sub-pictures so that pixel-rate balancing is guaranteed for multiple core platforms. A sub-picture contains an integer number of LCUs.
2. Sub-pictures should be independent, only de-blocking filter, Sample Adaptive Offset (SAO), Adaptive Loop Filter (ALF) can cross sub-picture boundary to minimize the data communication among the cores
3. Tiles, slices, entropy slices and WPP should be contained in sub-pictures and cannot cross sub-picture
4. Sub-picture entries in bitstream are signaled in the APS
5. The sub-picture partitioning information is signaled with tile syntax, two options are provided.

## Signalling of the sub-picture partitioning with tile syntax

There are two options to signal sub-pictures.

### Option 1 signaling method

As shown in Fig. 3, in this option uniformly spaced tiles are mandated for high levels and a sub-picture is simply a tile. Encoder and decoder have the same degree of parallelism. num\_subPic\_columns and num\_subPic\_rows are set equal to num\_tile\_columns\_minus1 + 1 and num\_tile\_rows\_minus1+1, respectively. num\_tile\_columns\_minus1, num\_tile\_rows\_minus1 and uniform\_spacing\_flag are fixed as shown in Table 1. No syntax change is required.

**Fig. 3 – Example of signaling the sub-picture boundaries for 2x2 sub-picture partitioning of option 1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Level*** | **Max luma pixel rate MaxLumaPR**  **(samples/sec)** | **Max luma picture size MaxLumaFS (samples)** | **Max bit rate MaxBR**  **(1000 bits/s)** | **Min Compression Ratio MinCR** | **MaxDpbSize (picture storage buffers)** | **Max CPB size**  **(1000 bits)** | **num\_tile\_columns\_minus1 + 1** | **num\_tile\_rows\_minus1 + 1** | **uniform\_spacing\_flag** |
| ***1*** | 552,960 | 36,864 | 128 | 2 | 6 | 350 | - | - | - |
| ***2*** | 3,686,400 | 122,880 | 1,000 | 2 | 6 | 1,000 | - | - | - |
| ***3*** | 13,762,560 | 458,752 | 5,000 | 2 | 6 | 5,000 | - | - | - |
| ***3.1*** | 33,177,600 | 983,040 | 9,000 | 2 | 6 | 9,000 | - | - | - |
| ***4*** | 62,668,800 | 2,088,960 | 15,000 | 4 | 6 | 15,000 | - | - | - |
| ***4.1*** | 62,668,800 | 2,088,960 | 30,000 | 4 | 6 | 30,000 | - | - | - |
| ***4.2*** | 133,693,440 | 2,228,224 | 30,000 | 4 | 6 | 30,000 | - | - | - |
| ***4.3*** | 133,693,440 | 2,228,224 | 50,000 | 4 | 6 | 50,000 | - | - | - |
| ***5*** | 267,386,880 | 8,912,896 | 50,000 | 6 | 6 | 50,000 | - | - | - |
| ***5.1*** | 267,386,880 | 8,912,896 | 100,000 | 8 | 6 | 100,000 | - | - | - |
| ***5.2p*** | 534,773,760 | 8,912,896 | 150,000 | 8 | 6 | 150,000 | 1 | 2 | 1 |
| ***6p*** | 1,002,700,800 | 33,423,360 | 300,000 | 8 | 6 | 300,000 | 2 | 2 | 1 |
| ***6.1p*** | 2,005,401,600 | 33,423,360 | 500,000 | 8 | 6 | 500,000 | 2 | 4 | 1 |
| ***6.2p*** | 4,010,803,200 | 33,423,360 | 800,000 | 6 | 6 | 800,000 | 2 | 8 | 1 |

Table 1 – Parallel Level limits (assuming 4kx2k@30 single core decoder capability) for option 1

### Option 2 signaling method

This option allows more tiles than sub-pictures, i.e. a sub-picture may contain multiple tiles. An encoder may have a higher degree of parallelism than a decoder. To signal sub-picture with tile syntax, the following rules are applied:

1. Tiles have to be evenly divided (i.e. uniformly spaced) in vertical direction. If a picture is evenly divided into **num\_subPic\_columns** x **num\_subPic\_rows** sub-pictures, then num\_tile\_rows\_minus1 is constrained to be equal to num\_subPic\_rows -1. This constraint is important for guaranteeing pixel-rate balancing because tiles are processed in rater-scan order within a picture. Without this constraint, sub-picture data could be interleaved in the bitstream.
2. In horizontal direction tiles can either be evenly divided or non-uniformed spaced, **num\_tile\_columns\_minus1** can be larger or equal to **num\_subPic\_columns -1,** but it is constrained that there must be **num\_subPic\_columns -1** right-hand tile column boundaries are coincident with the vertical sub-picture boundaries. The horizontal location (in units of LCUs) of the i-th vertical sub-picture boundary is computed by
   * ( ( i + 1 )\*PicWidthInCtbs / num\_subPic\_columns, i = 0, 1, …, num\_subPic\_columns -1.
3. Number of sub-picture **num\_subPic\_columns x num\_subPic\_rows** should be specified and mandated for levels based on the sample rates and picture resultions. If number of sub-picture **num\_subPic\_columns x num\_subPic\_rows** is set equal to 1 for a level, constraint 1 and 2escribed above do not apply to that level.
4. Number of sub-picture **num\_subPic\_columns** x **num\_subPic\_rows** should be specified in a way in which line buffer size increase of single core decoder is not required when single cores are replicated to support higher resolution video.

Fig. 4 shows an example how to achieve an even sub-picture partitions of 2x2 sub-pictures by using the tile syntax. In this case, a picture is evenly divided into 2 tile rows in vertical direction, while in the horizontal direction it is unevenly divided into 4 columns, but there is one right-hand tile column boundary coincident with the sub-picture boundary, which is right-hand column boundary of Tile02 and Tile21. i.e. column\_width[0] + column\_width[1] has to be equal to PicWidthInCtbs/2 in this case.

In the bitstream, it should be signaled that Tile00 is start location for sub-picture 0, Tile10 is start location for sub-picture 1, Tile20 is start location for sub-picture 2 and Tile30 is start location for sub-picture 3. With this constrained tile partitions and signaling, sub-pictures can be dispatched and processed by multi-cores in parallel with equal pixel rate loading.

**Fig. 4 – Example of signaling the sub-picture boundaries for 2x2 sub-picture partitioning of option 2**

For option 2 the tile syntax in PPS (picture parameter set) is modified as shown in Table 2. In the vertical direction the non-uniformed spaced tiles are allowed only if sub-pictures are not mandated. num\_subpic\_columns and num\_subpic\_rows are specified in Table 3 and can be derived by level\_id.

|  |  |
| --- | --- |
| … |  |
| if( tiles\_or\_entropy\_coding\_sync\_idc = = 1 ) { |  |
| **num\_tile\_columns\_minus1** | ue(v) |
| **num\_tile\_rows\_minus1** | ue(v) |
| **uniform\_spacing\_flag** | u(1) |
| if( !uniform\_spacing\_flag ) { |  |
| for( i = 0; i < num\_tile\_columns\_minus1; i++ ) |  |
| **column\_width[**i **]** | ue(v) |
| If (num\_subpic\_columns == 1 && num\_subpic\_rows ==1) |  |
| for( i = 0; i < num\_tile\_rows\_minus1; i++ ) |  |
| **row\_height[**i **]** | ue(v) |
| } |  |
| **loop\_filter\_across\_tiles\_enabled\_flag** | u(1) |
| } |  |
| … |  |

**Table 2 Modified Tile syntax for the SPS and PPS**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Level*** | **Max luma pixel rate MaxLumaPR**  **(samples/sec)** | **Max luma picture size MaxLumaFS (samples)** | **Max bit rate MaxBR**  **(1000 bits/s)** | **Min Compression Ratio MinCR** | **MaxDpbSize (picture storage buffers)** | **Max CPB size**  **(1000 bits)** | **num\_subpic\_columns** | **num\_subpic\_rows** |
| ***1*** | 552,960 | 36,864 | 128 | 2 | 6 | 350 | 1 | 1 |
| ***2*** | 3,686,400 | 122,880 | 1,000 | 2 | 6 | 1,000 | 1 | 1 |
| ***3*** | 13,762,560 | 458,752 | 5,000 | 2 | 6 | 5,000 | 1 | 1 |
| ***3.1*** | 33,177,600 | 983,040 | 9,000 | 2 | 6 | 9,000 | 1 | 1 |
| ***4*** | 62,668,800 | 2,088,960 | 15,000 | 4 | 6 | 15,000 | 1 | 1 |
| ***4.1*** | 62,668,800 | 2,088,960 | 30,000 | 4 | 6 | 30,000 | 1 | 1 |
| ***4.2*** | 133,693,440 | 2,228,224 | 30,000 | 4 | 6 | 30,000 | 1 | 1 |
| ***4.3*** | 133,693,440 | 2,228,224 | 50,000 | 4 | 6 | 50,000 | 1 | 1 |
| ***5*** | 267,386,880 | 8,912,896 | 50,000 | 6 | 6 | 50,000 | 1 | 1 |
| ***5.1*** | 267,386,880 | 8,912,896 | 100,000 | 8 | 6 | 100,000 | 1 | 1 |
| ***5.2p*** | 534,773,760 | 8,912,896 | 150,000 | 8 | 6 | 150,000 | 1 | 2 |
| ***6p*** | 1,002,700,800 | 33,423,360 | 300,000 | 8 | 6 | 300,000 | 2 | 2 |
| ***6.1p*** | 2,005,401,600 | 33,423,360 | 500,000 | 8 | 6 | 500,000 | 2 | 4 |
| ***6.2p*** | 4,010,803,200 | 33,423,360 | 800,000 | 6 | 6 | 800,000 | 2 | 8 |

Table 3 – Parallel level limits (assuming 4kx2k@30 single core decoder capability) for option 2

# Conclusion and recommendation

It is recommended to define companion “parallel” levels for level 5.2 and above, and mandate tiles (either option 1 or 2) for those levels to enable parallel decoding on multi-core platforms.

# References

[1] F. Bossen, “Common test conditions and software reference configurations,” JCT-VC Document, JCTVC-I1100, 9th Meeting: Geneva, Switzerland, 27 April – 07 May, 2012

[2] [B. Bross](mailto:benjamin.bross@hhi.fraunhofer.de), [W.-J. Han](mailto:wjhan.han@samsung.com), [J.-R. Ohm](mailto:ohm@ient.rwth-aachen.de), [G. J. Sullivan](mailto:garysull@microsoft.com), [T. Wiegand](mailto:thomas.wiegand@hhi.fraunhofer.de) “High Efficiency Video Coding (HEVC) text specification draft 7,” JCT-VC Document, JCTVC-I1003, 9th Meeting: Geneva, Switzerland, 27 April – 07 May, 2012.

[3] M. Zhou, “AHG4: Enable parallel decoding with tiles”, JCT-VC Document, JCTVC-I1003, 9th Meeting: Geneva, Switzerland, 27 April – 07 May, 2012.

.

# Patent rights declaration(s)

**Texas Instruments, 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).**