|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  8th Meeting: San José, CA, USA, 1–10 February, 2012 | Document: JCTVC-H0203 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **AHG4: Further latency reduction of CABAC initialization for dependent Tiles** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Hendry Sang Oh Jeong  Seung Wook Park Byeong Moon Jeon  #221 Yangjae-dong, Seocho-gu,  Seoul 137-130,  Korea | Tel: Email: | - [hendry.hendry@lge.com](mailto:hendry.hendry@lge.com)  [sangoh.jeong@lge.com](mailto:sangoh.jeong@lge.com)  [seungwook.park@lge.com](mailto:seungwook.park@lge.com)  [bm.jeon@lge.com](mailto:bm.jeon@lge.com) |
| *Source:* | LG Electronics | | |

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

# Abstract

When dependent tiles are processed, a low latency CABAC initialization scheme was proposed. In that proposal, the CABAC probabilities of the first LCU in each tile are inherited from those of the adjacent left LCU. This contribution builds on the proposal in that the CABAC probabilities of the first LCU in each tile of the first tile column, from the second tile row, are inherited from those of the adjacent upper LCU, on top of the probability inheritance from the left LCU in the proposal. It is claimed that the proposed method can further reduce processing latency in the proposal, by not having to wait for all the tiles in the upper LCU row of the row boundaries to be processed before inheriting the CABAC probabilities. Result from experiments report that the proposed method does not cause loss when 4 tiles (2 tile rows and 2 tile columns) and causes maximum 0.3% loss when 16 tiles (4 tile rows and 4 tile columns) is used.

# Introduction

In Geneva (7th) meeting, JCTVC-G197 [1] proposed to initialize CABAC probabilities of the first LCU in non-first tile, when dependent tile configuration is used, from adjacent left LCU as indicated by the arrows in Figure 1. It reportedly showed some reduction in the processing latency compared to the traditional raster scan order and saving memory to store column line buffer.

According to the proposal as in Figure 1, the CABAC probabilities of LCU #13 are inherited from those of LCU #4 instead of LCU #12, LCU #31 are inherited from LCU #18 instead of LCU #30. Using the method in the proposal, it is reported that processing the first LCU in different tiles can start as long as the left LCU is processed. For multi core applications, it allows parallel processing of each core to start as long as the left LCU has been processed.

Figure 1 – Low latency CABAC initialization

However, proposal [1] did not regulate CABAC probability initialization for the first LCU in the first-column-non-first-row tile. For example, CABAC probability table of the LCU #40 in Figure 1 still have to be initialized from that of LCU #39, which would impose extra latency for waiting for that LCU to be decoded.

In this document, we propose a simple change in which for the above case that was not covered yet by technique offered by [1]. The proposed change in this contribution may provide further reduce processing latency even more can be considered as extension to [1].

# Proposed changes

The proposal [1] reportedly reduced the latency in CABAC initialization for the first LCU of dependent tiles (i.e., tile\_boundary\_independence\_flag = 0, in SPS or PPS) right after tile column boundaries. Although it can reduce the latency caused by the tile column boundaries, the latency caused by the tile row boundaries still remain from the second tiles of the first tile column.

In this contribution, a method to achieve lower latency processing for dependent tiles is proposed by modifying the previous CABAC initialization [1] as follows:

For each dependent Tile, the following is applied:

* If a neighboring left tile is available, the CABAC probabilities of the first LCU are inherited from those of the adjacent left LCU.
* Else if a neighboring left tile is not available but a neighboring-top tile is available, the CABAC probabilities of the first LCU are inherited from those of the adjacent upper LCU.
* Else if both a neighboring left and a neighboring upper LCU are not available, the CABAC probabilities of the first LCU is reset.

Figure 2 illustrates the proposed lower latency CABAC initialization for dependent tiles.

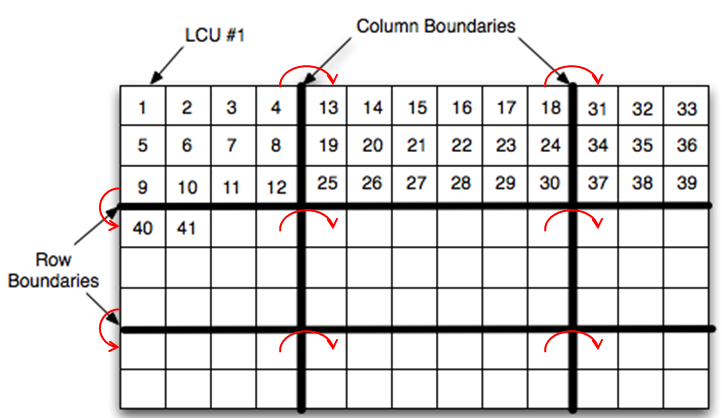


Figure 2 – Proposed lower latency CABAC initialization for the first LCU in dependent tiles

# Experiment Results

We have tested the effect of the proposed modification to CABAC probability initialization to the coding performance. The anchor of such test is the common test condition where a picture is uniformly divided into 4 (2 tile rows and 2 tile columns) and 16 (4 tile rows and 4 tile columns) dependent tiles.

From the test result, the following can be observed. For experiments with 4 tiles, the proposed modification, in average, does not cause any loss for any cases. For experiments with 16 tiles, the proposed modification, in average, only cause at maximum 0.3% luma loss. Considering the advantage of low processing latency that is offered, such loss perhaps can be acceptable.

The result for 4 tiles experiments was cross-checked by MediaTek while the result for 16 tiles experiments was cross-checked by eBrisk. We would like to thank both companies for the help.

Table 1 – 4 tiles -- Intra only

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | |
|  | Y | U | V | Y | U | V |
| Class A (8bit) | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class B | 0.0% | -0.1% | -0.1% | 0.0% | 0.0% | 0.0% |
| Class C | 0.0% | -0.1% | -0.1% | 0.0% | 0.0% | 0.0% |
| Class D | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class E | 0.0% | 0.1% | -0.1% | 0.0% | -0.1% | 0.0% |
| **Overall** | 0.0% | 0.0% | -0.1% | 0.0% | 0.0% | 0.0% |
|  | 0.0% | 0.0% | -0.1% | 0.0% | 0.0% | 0.0% |
| Class F | 0.0% | 0.0% | -0.1% | 0.0% | 0.0% | -0.1% |
| Enc Time[%] | 98% | | | 98% | | |
| Dec Time[%] | 99% | | | 99% | | |

Table 2 – 4 tiles – random access

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Access HE** | | | **Random Access LC** | | |
|  | Y | U | V | Y | U | V |
| Class A (8bit) | 0.1% | 0.0% | 0.0% | 0.0% | 0.0% | -0.2% |
| Class B | 0.0% | 0.0% | 0.1% | 0.0% | 0.1% | 0.0% |
| Class C | 0.0% | 0.3% | 0.1% | 0.0% | 0.1% | 0.0% |
| Class D | 0.0% | 0.4% | 0.0% | 0.1% | 0.2% | 0.0% |
| Class E |  |  |  |  |  |  |
| **Overall** | 0.0% | 0.2% | 0.1% | 0.0% | 0.1% | 0.0% |
|  | 0.0% | 0.2% | 0.0% | 0.0% | 0.1% | 0.0% |
| Class F | 0.1% | 0.1% | 0.1% | 0.1% | 0.1% | 0.1% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 101% | | | 102% | | |

Table 3 – 4 tiles – low delay B

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Low delay B HE** | | | **Low delay B LC** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | 0.0% | 0.0% | 0.3% | 0.0% | -0.1% | -0.3% |
| Class C | 0.1% | 0.0% | 0.0% | 0.0% | 0.1% | 0.2% |
| Class D | 0.1% | 0.6% | -0.2% | 0.1% | 0.1% | -0.4% |
| Class E | -0.1% | 0.0% | -0.8% | 0.0% | 0.4% | 0.0% |
| **Overall** | 0.0% | 0.1% | -0.1% | 0.0% | 0.1% | -0.2% |
|  | 0.0% | 0.1% | -0.1% | 0.0% | 0.1% | -0.2% |
| Class F | 0.2% | -0.1% | 0.2% | -0.1% | 0.1% | 0.7% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 99% | | | 98% | | |

Table 4 – 4 tiles – low delay P

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Low delay P HE** | | | **Low delay P LC** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | 0.0% | 0.0% | 0.3% | 0.0% | 0.0% | -0.2% |
| Class C | 0.1% | 0.0% | 0.0% | 0.0% | 0.2% | 0.1% |
| Class D | 0.0% | 0.6% | -0.2% | 0.0% | -0.1% | 0.7% |
| Class E | 0.0% | 0.0% | -0.8% | 0.1% | -0.3% | -0.2% |
| **Overall** | 0.0% | 0.1% | -0.1% | 0.0% | 0.0% | 0.1% |
|  | 0.0% | 0.1% | -0.1% | 0.0% | 0.0% | 0.0% |
| Class F | 0.3% | -0.1% | 0.2% | -0.1% | -0.2% | -0.1% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 99% | | | 99% | | |

Table 5 – 16 tiles -- Intra only

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | |
|  | Y | U | V | Y | U | V |
| Class A (8bit) | 0.0% | 0.1% | 0.1% | 0.0% | 0.0% | 0.0% |
| Class B | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Class C | 0.0% | -0.1% | -0.1% | 0.0% | 0.0% | 0.0% |
| Class D | 0.0% | 0.0% | 0.0% | 0.0% | -0.1% | -0.1% |
| Class E | 0.0% | 0.2% | -0.1% | 0.0% | -0.2% | -0.2% |
| **Overall** | 0.0% | 0.0% | 0.0% | 0.0% | -0.1% | -0.1% |
|  | 0.0% | 0.0% | 0.0% | 0.0% | -0.1% | -0.1% |
| Class F | 0.0% | -0.2% | -0.2% | 0.0% | -0.1% | -0.1% |
| Enc Time[%] | 99% | | | 98% | | |
| Dec Time[%] | 99% | | | 99% | | |

Table 6 – 16 tiles – random access

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Access HE** | | | **Random Access LC** | | |
|  | Y | U | V | Y | U | V |
| Class A (8bit) | 0.1% | 0.4% | 0.1% | 0.0% | -0.1% | 0.1% |
| Class B | 0.1% | 0.1% | 0.2% | 0.0% | 0.1% | 0.1% |
| Class C | 0.1% | 0.5% | 0.4% | 0.1% | -0.1% | 0.1% |
| Class D | 0.3% | 0.5% | 0.6% | 0.3% | 0.1% | -0.3% |
| Class E |  |  |  |  |  |  |
| **Overall** | 0.2% | 0.3% | 0.3% | 0.1% | 0.0% | 0.0% |
|  | 0.2% | 0.3% | 0.3% | 0.1% | 0.0% | 0.0% |
| Class F | 0.3% | 0.2% | 0.4% | 0.1% | -0.1% | 0.0% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 101% | | | 102% | | |

Table 7 – 16 tiles – low delay B

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Low delay B HE** | | | **Low delay B LC** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | 0.1% | 0.0% | 0.3% | 0.0% | 0.3% | 0.4% |
| Class C | 0.2% | 0.5% | 0.4% | 0.1% | 0.6% | 0.5% |
| Class D | 0.6% | 0.8% | 0.8% | 0.7% | 0.9% | 0.3% |
| Class E | 0.4% | 0.5% | -0.5% | 0.4% | -1.1% | -0.4% |
| **Overall** | 0.3% | 0.4% | 0.3% | 0.3% | 0.3% | 0.3% |
|  | 0.3% | 0.5% | 0.3% | 0.3% | 0.3% | 0.3% |
| Class F | 0.6% | 0.5% | 0.1% | 0.1% | 0.3% | 0.2% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 99% | | | 98% | | |

Table 8 – 16 tiles – low delay P

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Low delay P HE** | | | **Low delay P LC** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | 0.1% | 0.0% | 0.3% | 0.0% | 0.2% | 0.5% |
| Class C | 0.2% | 0.5% | 0.4% | 0.1% | 0.3% | 0.5% |
| Class D | 0.6% | 0.8% | 0.8% | 0.4% | 0.0% | 0.6% |
| Class E | 0.6% | 0.5% | -0.5% | 0.5% | -1.4% | -0.1% |
| **Overall** | 0.3% | 0.4% | 0.3% | 0.2% | -0.1% | 0.4% |
|  | 0.3% | 0.5% | 0.3% | 0.2% | -0.1% | 0.4% |
| Class F | 0.5% | 0.5% | 0.1% | 0.4% | 0.8% | 0.2% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 99% | | | 99% | | |

# Conclusion

This contribution extends method offered by [1] for CABAC probability table initialization for the first LCU in non-first dependent tile. The proposed extension can further reduce processing latency while having very small impact to the coding efficiency. We request to the JCTVC group to consider adoption of the proposed method to complete what have been adopted from [1] in the last meeting.

# Patent rights declaration(s)

**LG Electronics 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).**

# Reference

[1] JCTVC-G197, “AHG4: Low latency CABAC initialization for dependent Tiles,” 7th JCT-VC Meeting, 7th Meeting: Geneva, CH, 21-30 November, 2011.