|  |  |
| --- | --- |
| **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-H0428 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Non-CE6: Postpone MPM sorting** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Vadim Seregin  Wei-Jung Chien  Marta Karczewicz  5775 Morehouse Drive San Diego, CA 92121, USA | Email: | [vseregin@qualcomm.com](mailto:vseregin@qualcomm.com)  [wchien@qualcomm.com](mailto:wchien@qualcomm.com)  [martak@qualcomm.com](mailto:martak@qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

To code luma intra prediction mode, two most probable modes are used. At first, most probable modes are sorted in the ascending order and then flag, indicating usage of the most probable mode, is signalled, following by another most probable mode flag or using sorted most probable modes in codeword reduction for the rest intra modes. However, if current luma intra mode is equal to one of the most probable modes, sorting is not necessary since the index is bypass-coded.. This contribution proposes removing MPM sorting for the cases when luma intra mode is one of the most probable modes. Experimental results revealed that this simplification has no impact on coding efficiency and has 0.00% loss in average.

# Introduction

In current test model, two different most probable intra modes are derived from the left,above neighbor CU or default intra mode (planar mode). The flowchart of current HM and proposed luma intra mode coding is shown on Fig. 1. If current intra mode is equal to one of the MPMs, the index of MPMs is signaled (MPM coding in Fig. 1); otherwise, a remaining mode is derived and signaled based on MPMs (rest of the modes coding in Fig. 1). Both the MPM coding and the rest of the modes coding are entropy coded in bypass mode.



Fig. 1. Luma intra mode coding for curent HM (a) and proposal (b)

It can be seen from Fig. 1, that sorting operation is not necessary in MPM coding since the bin is bypassed coded. In this contribution, it is proposed to remove MPM sorting from MPM coding, and sort MPMs only for luma intra modes different from most probable modes.

The removing of MPM sorting was implemented on top of HM5.0, and experiment results under common test conditions [1] for AI configurations are summarised in the following table.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | |
|  | Y | U | V | Y | U | V |
| Class A (8bit) | 0.00% | -0.01% | -0.02% | 0.00% | 0.01% | -0.01% |
| Class B | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.01% |
| Class C | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class D | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class E | 0.00% | 0.01% | 0.01% | 0.00% | -0.01% | -0.02% |
| **Overall** | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
|  | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class F | 0.01% | -0.01% | -0.01% | 0.00% | -0.01% | 0.01% |
| Enc Time[%] | 99% | | | 98% | | |
| Dec Time[%] | 99% | | | 99% | | |

# Conclusion

Removing of MPM sorting operation is proposed in this contribution. Since, MPM sorting operation is performed at both encoder and decoder, removing of MPM sorting for the intra mode, equal to one of the MPMs, reduces the complexity with no performance loss. We recommend to adopt removing of MPM sorting into HM.

# References

1. F. Bossen, “*Common HM test conditions and software reference configurations*,” JCTVC-G1200, 7th JCT-VC Meeting, Geneva, CH, November 2011.

# Patent rights declaration(s)

**Qualcomm 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).**

# Working Draft text

This text modification for working draft is based on JCTVC-G1103 version d8.

**8.3.1 Derivation process for luma intra prediction mode**

…

1. The derivation process for neighbouring treeblocks specified in subclause XXX with ( xB,  yB ) given as input and the output is assigned to tbAddrA and tbAddrB specifying the treeblock addresses of treeblocks covering ( xBA,  yBA ) and ( xBB, yBB ) respectively where ( xBA,  yBA ) is set equal to ( xB-1,  yB ) and ( xBB,  yBB ) is set equal to ( xB,  yB-1 ).
2. For N being either replaced A or B, the variables intraPredModeN are derived as follows.

* If the treeblock with address tbAddrN is not available, intraPredModeN is set equal to Intra\_Planar.
* Otherwise, if the coding unit covering ( xBN,  yBN ) is not coded as intra mode, intraPredModeN is set equal to Intra\_Planar,
* Otherwise, if yB-1 is smaller than YLCU, intraPredModeA is set equal to IntraPredMode[ xBA ][ yBA ] and intraPredModeB is set equal to Intra\_Planar.
* Otherwise, intraPredModeN is set equal to IntraPredMode[ xBN ][ yBN ], where IntraPredMode is the variable array assigned to the coding unit covering the luma location ( xBN, yBN ).

1. For N being either replaced A or B, the variables candIntraPredModeN are derived as follows.

* If intraPredModeN is greater than or equal to intraPredModeNum, candIntraPredModeN is set equal to Intra\_Planar.
* Otherwise, candIntraPredModeN is set equal to intraPredModeN

1. If candIntraPredModeA is equal to candIntraPredModeB, the candIntraPredModeA is modified as follows:

* If candIntraPredModeA is not equal to Intra\_Planar, candIntraPredModeA is set equal to Intra\_Planar
* Otherwise, candIntraPredModeA is set equal to Intra\_DC

1. ~~The candModeList[x] is derived as follows:~~

~~candModeList[0] = Min( candIntraPredModeA, candIntraPredModeB ) (8‑13)  
candModeList[1] = Max( candIntraPredModeA, candIntraPredModeB ) (8‑13)~~

1. IntraPredMode[ xB ][ yB ] is derived by applying the following procedure:

* If prev\_intra\_pred\_flag[ xB ][ yB ] is true, the IntraPredMode[ xB ][ yB ] is set equal to candModeList[ mpm\_flag ][ xB ][ yB ]]
* Otherwise IntraPredMode[ xB ][ yB ] is derived by applying the following ordered steps:
  1. The candModeList[x] is derived as follows:

candModeList[0] = Min( candIntraPredModeA, candIntraPredModeB ) (8‑13)  
candModeList[1] = Max( candIntraPredModeA, candIntraPredModeB ) (8‑13)

* 1. IntraPredMode[ xB ][ yB ] = rem\_intra\_luma\_pred\_mode
  2. When IntraPredMode[ xB ][ yB ] is equal or greater than candModeList[ 0 ], the value of IntraPredMode[ xB ][ yB ] is increased by one
  3. When IntraPredMode[ xB ][ yB ] is equal or greater than candModeList[ 1 ], the value of IntraPredMode[ xB ][ yB ] is increased by one