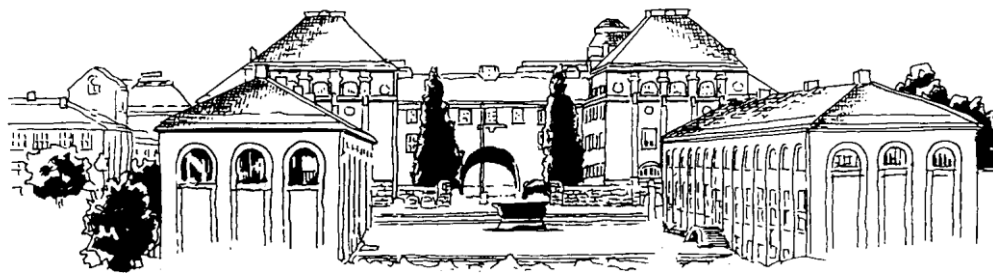


**JVET-P0270**

# **Non-CE7: Modified signaling method of cu\_cbf and tu\_cbf\_luma**



**Information & Telecommunication Research Institute (ITRI) Lab**  
**Sejong University, Seoul, Republic of Korea**  
**Dong-Jae Won, Jae-Min Ha, Joo-Hee Moon**

# Contents

---

1. Introduction
2. Proposed method
3. Test results
4. Conclusions

# 1. Introduction

## - Current VVC draft

### Coding unit syntax:

- `cu_cbf` in dual tree is signaled only when current CU is coded by `MODE_IBC` and `general_merge_flag` is equal to zero. And then `tu_cbf_luma` is inferred by `cu_cbf`

Table 1. Coding unit syntax in current VVC draft 6

<code>coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {</code>	Descriptor
<code>...</code>	
<code>if( CuPredMode[ chType ][ x0 ][ y0 ] != MODE_INTRA &amp;&amp; !pred_mode_plt_flag &amp;&amp; general_merge_flag[ x0 ][ y0 ] == 0 )</code>	
<code>cu_cbf</code>	<code>ae(v)</code>

- But, it is inconsistent in terms of its semantic in the condition  
→ Because the `cbf` information for luma is signaled by `tu_cbf_luma` in the other coding conditions

## 2. Proposed method

### - Proposed syntax changes

#### Proposed method 1:

- `cu_cbf` is signaled only when current tree type is equal to single tree
- The `cbf` information for luma are informed by `tu_cbf_luma` in any case

Table 2. Proposed Coding unit syntax

coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {	Descriptor
...	
if( CuPredMode[ chType ][ x0 ][ y0 ] != MODE_INTRA && !pred_mode_plt_flag && general_merge_flag[ x0 ][ y0 ] == 0 && treeType == SINGLE_TREE )	
<b>cu_cbf</b>	ae(v)

## 2. Proposed method

### - Proposed syntax changes

#### Proposed method 1:

- When current CU is coded by MODE\_IBC in dual tree, one signaling condition is added for tu\_cbf\_luma

Table 3. Proposed Transform unit syntax

transform_unit( x0, y0, tbWidth, tbHeight, treeType, subTuIndex, chType ) {	Descriptor
...	
if( treeType == SINGLE_TREE    treeType == DUAL_TREE_LUMA ) {	
if( ( IntraSubPartitionsSplitType == ISP_NO_SPLIT && !( cu_sbt_flag && ( ( subTuIndex == 0 && cu_sbt_pos_flag )    ( subTuIndex == 1 && !cu_sbt_pos_flag ) ) ) && ( CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA    ( CuPredMode[ chType ][ x0 ][ y0 ] == MODE_IBC && treeType == DUAL_TREE_LUMA )    tu_cbf_cb[ x0 ][ y0 ]    tu_cbf_cr[ x0 ][ y0 ]    CbWidth[ chType ][ x0 ][ y0 ] > MaxTbSizeY    CbHeight[ chType ][ x0 ][ y0 ] > MaxTbSizeY ) )    ( IntraSubPartitionsSplitType != ISP_NO_SPLIT && ( subTuIndex < NumIntraSubPartitions - 1    !InferTuCbfLuma ) ) )	
<b>tu_cbf_luma</b> [ x0 ][ y0 ]	ae(v)
if( IntraSubPartitionsSplitType != ISP_NO_SPLIT )	
InferTuCbfLuma = InferTuCbfLuma && !tu_cbf_luma[ x0 ][ y0 ]	
}	

## 2. Proposed method

- Proposed additional context for tu\_cbf\_luma

### Proposed method 2:

Table 4. CtxInc derivation process for tu\_cbf\_luma

Conditions	ctxInc
If intra_bdpcm_flag is equal to 1, ctxInc is set equal to	1
Otherwise, if current coding unit is coded by MODE_IBC, ctxInc is set equal to	2
Otherwise, if IntraSubpartitionsSplitType is equal to ISP_NO_SPLIT , ctxInc is set equal to	0
Otherwise, if the current transform unit is the first one to be parsed in a coding unit or the value of tu_cbf_luma of the previous luma transform unit in the current coding unit is equal to 0, ctxInc is set equal to	23
Otherwise, ctxInc is set equal to	34

### 3. Test results

#### - Test results in AI

Test result of method 1 with IBC being on:

	All Intra Main 10				
	Over VTM-6. 0				
	Y	U	V	EncT	DecT
Class A1	#VALUE!	#VALUE!	#VALUE!	#NUM!	#NUM!
Class A2	#VALUE!	#VALUE!	#VALUE!	#NUM!	#NUM!
Class B	0.00%	-0.05%	-0.01%	100%	99%
Class C	-0.01%	0.05%	0.09%	101%	99%
Class E	0.00%	0.07%	0.13%	101%	99%
<b>Overall</b>	#VALUE!	#VALUE!	#VALUE!	#NUM!	#NUM!
Class D	0.00%	0.01%	0.12%	100%	101%
Class F	0.01%	-0.14%	0.04%	101%	100%

Test result of method 1+2 with IBC being on:

	All Intra Main 10				
	Over VTM-6. 0				
	Y	U	V	EncT	DecT
Class A1	#VALUE!	#VALUE!	#VALUE!	#DIV/0!	#DIV/0!
Class A2	#VALUE!	#VALUE!	#VALUE!	#NUM!	#NUM!
Class B	0.00%	-0.03%	0.02%	101%	101%
Class C	0.00%	0.01%	0.05%	101%	99%
Class E	-0.01%	-0.02%	0.02%	100%	99%
<b>Overall</b>	#VALUE!	#VALUE!	#VALUE!	#DIV/0!	#DIV/0!
Class D	0.00%	-0.03%	-0.01%	100%	100%
Class F	#VALUE!	#VALUE!	#VALUE!	#NUM!	#NUM!

# 4. conclusions

---

## Summary:

- Current signaling structure of cu\_cbf and tu\_cbf\_luma is not consistent in terms of its semantic
- Change the signaling condition of cu\_cbf and tu\_cbf\_luma
- It is suggested to adopt the proposed method into VVC next version



End

---

Thank ETRI  
for cross-checking