

# JVET-P0152

## CE4-related: On maximum number of subblock-based merging candidates

**Authors:**

Olena Chubach, Chun-Chia Chen, Chih-Wei Hsu, Ching-Yeh Chen, Tzu-Der Chuang,  
Yu-Wen Huang, Shaw-Min Lei

**Presenter:** Olena Chubach

# Overall Summary

- Adopted JVET-00238 introduced a mechanism to enable signaling some of the slice parameters either in the PPS or in the slice header
- However, there is an issue in the VVC specification draft for `pps_five_minus_max_num_subblock_merge_cand_plus1` when affine mode is disabled in SPS.
- Moreover, after adopting JVET-00238 the issue of subblock-based merge candidate list containing only one zero motion vector (MV) candidate was reintroduced
  - This issue was considered illegal and solved by adopting JVET-00220
- In this contribution it is proposed to fix these two issues.

# Introduction

- Depending on value of **pps\_or\_slice\_flag** in the PPS the parameter values of the below listed syntax elements may be conditionally signalled in the PPS or slice headers, or always signaled in slice headers.
  - dep\_quant\_enabled\_flag,
  - ref\_pic\_list\_sps\_flag,
  - slice\_temporal\_mvp\_enabled\_flag,
  - mvd\_l1\_zero\_flag,
  - collocated\_from\_l0\_flag,
  - six\_minus\_max\_num\_merge\_cand
  - five\_minus\_max\_num\_subblock\_merge\_cand
  - max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand
- If **pps\_or\_slice\_flag** is equal to 1, then for each of the above syntax elements:
  - Decode a corresponding syntax element in the PPS.
  - If the value of the corresponding syntax element in the PPS is equal to 0, then the parameter value is signalled by a syntax element in the slice header. Otherwise, the parameter value for the slice header syntax element is derived as the value of the corresponding syntax element in the PPS minus 1.

# Problem Statement

- When `five_minus_max_num_subblock_merge_cand` is not present in the slice header, following applies:
  - If `sps_affine_enabled_flag` is equal to 0, the value of `five_minus_max_num_subblock_merge_cand` is inferred to be equal to  $5 - (\text{sps\_sbtmvp\_enabled\_flag} \ \&\& \ \text{slice\_temporal\_mvp\_enabled\_flag})$
  - Otherwise (`sps_affine_enabled_flag` is equal to 1), the value of `five_minus_max_num_subblock_merge_cand` is inferred to be equal to `pps_five_minus_max_num_subblock_merge_cand_plus1 - 1`.
- Two issues are asserted:
  - When `sps_affine_enabled_flag = 0`, the value of `five_minus_max_num_subblock_merge_cand` is inferred but not considering the `pps_five_minus_max_num_subblock_merge_cand_plus1`, which results in inconsistent inference between slice header and PPS.
  - When the number of subblock-based merging candidates is 1 (`pps_five_minus_max_num_subblock_merge_cand_plus1 = 5`), SbTMVP is allowed on sequence level (`sps_sbtmvp_enabled_flag = 1`), but slice disallows any SbTMVP (`slice_temporal_mvp_enabled_flag = 0`), then subblock-based merging candidate list only contains one zero MV candidate. This case is considered illegal, and was fixed by adopting JVET-00220.

# Proposed Method

`five_minus_max_num_subblock_merge_cand` specifies the maximum number of subblock-based merging motion vector prediction (MVP) candidates supported in the slice subtracted from 5.

When `five_minus_max_num_subblock_merge_cand` is not present, the following applies:

- If `sps_affine_enabled_flag` is equal to 0 and `pps_five_minus_max_num_subblock_merge_cand_plus1` is equal to 0, the value of `five_minus_max_num_subblock_merge_cand` is inferred to be equal to  $5 - (\text{sps\_sbtmvp\_enabled\_flag} \ \&\& \ \text{slice\_temporal\_mvp\_enabled\_flag})$
- Otherwise (~~`sps_affine_enabled_flag` is equal to 1~~), the value of `five_minus_max_num_subblock_merge_cand` is inferred to be equal to `pps_five_minus_max_num_subblock_merge_cand_plus1 - 1`.

# Proposed Method

pps\_five\_minus\_max\_num\_subblock\_merge\_cand\_plus1 equal to 0 specifies that five\_minus\_max\_num\_subblock\_merge\_cand is present in slice header of slices referring to the PPS if sps\_affine\_enabled\_flag is equal to 1. pps\_five\_minus\_max\_num\_subblock\_merge\_cand\_plus1 larger than 0 specifies that five\_minus\_max\_num\_subblock\_merge\_cand is not present in slice header of slices referring to the PPS.

If sps\_affine\_enabled\_flag is equal to 1, the value of pps\_five\_minus\_max\_num\_subblock\_merge\_cand\_plus1 shall be in the range of 0 to 6, inclusive.

Otherwise, if all of the following conditions are true, the value of pps\_five\_minus\_max\_num\_subblock\_merge\_cand\_plus1 shall be 0.

- sps\_temporal\_mvp\_enabled\_flag is equal to 1
- sps\_sbtmvp\_enabled\_flag is equal to 1
- pps\_temporal\_mvp\_enabled\_idc is equal to 0

TMVP is not controlled by PPS -> slice level control is forced

Otherwise, if all of the following conditions are true, the value of pps\_five\_minus\_max\_num\_subblock\_merge\_cand\_plus1 shall be in the range of 5 to 6, inclusive

- sps\_temporal\_mvp\_enabled\_flag is equal to 1
- sps\_sbtmvp\_enabled\_flag is equal to 1
- pps\_temporal\_mvp\_enabled\_idc is equal to 2

PPS allows TMVP -> at most one candidate is allowed in PPS

Otherwise, the value of pps\_five\_minus\_max\_num\_subblock\_merge\_cand\_plus1 shall be 6.

# Conclusions

- In this contribution it is proposed to fix two issues in the VVC specification draft related to `pps_five_minus_max_num_subblock_merge_cand_plus1` for the case when affine mode is disabled in SPS.
- It is proposed to adopt proposed fixes and include them into the next VVC specification draft.