

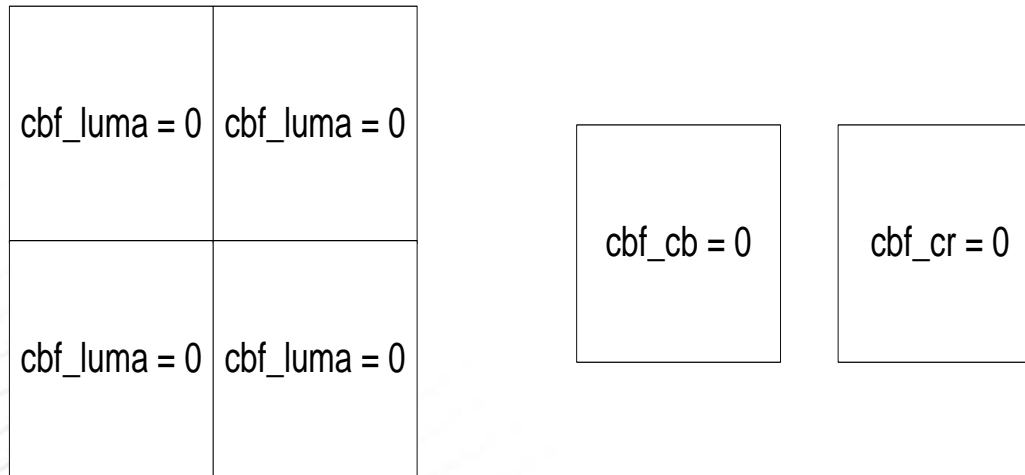
CLARIFICATION OF THE SEMANTICS OF NO_RESIDUAL_DATA_FLAG



Zhijie Yang, Peisong Chen, Wade Wan

- **no_residual_data_flag equal to 1 specifies that no residual data are present for the current coding unit. no_residual_data_flag equal to 0 specifies that residual data are present for the current coding unit.**
- **Two interpretations of “residual data are present”**
 - Interpretation #1: it could mean there exists residual syntax element(s) in the current coding unit such as cbf_cb, cbf_cr, etc.
 - Interpretation #2: it could mean there are one or more non-zero transform coefficients in the current coding unit.

- a 64x64 CTB with partition type equal to 2Nx2N, merge_flag equal to 1
 - The value of no_residual_data_flag is inferred to be equal to 0
 - the first layer split transform flag is inferred to be equal to 1. Furthermore, all four cbf_luma flags, one cbf_cb flag and one cbf_cr flag are equal to 0



According to the Interpretation #1, the stream is compliant because there exists residual syntax element(s) in the current CTB. However, the current HM software cannot handle the example described above correctly and will instead incorrectly derive the context index of the skip flag

According to the Interpretation #2, the stream is NOT compliant because all the transform coefficients are zero.

Recommended semantics change

`no_residual_data_flag` equal to 1 specifies that ~~no residual data are~~
`transform_tree()` **syntax is not** present for the current coding unit.
`no_residual_data_flag` equal to 0 specifies that ~~residual data are~~
`transform_tree()` **syntax is** present for the current coding unit.

Recommended software change

Match `QtRootCbf` to the definition of `no_residual_data_flag`