

Byte Alignment of Slice Header (JCTVC-F131)

Vivienne Sze, Madhukar Budagavi, Akira Osamoto

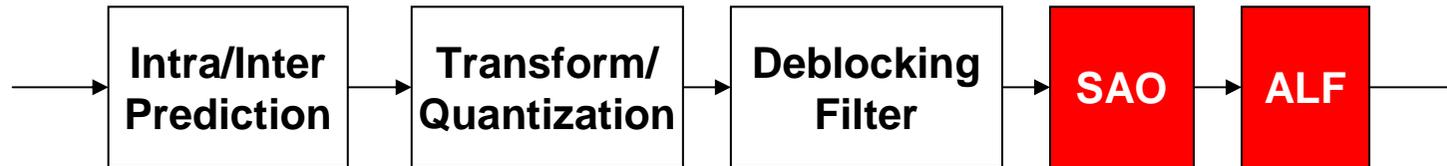
Texas Instruments

**Joint Collaborative Team on Video Coding (JCT-VC)
of ITU-T SG16 WP3 and ISO/IEC JTb1/SC29/WG11**

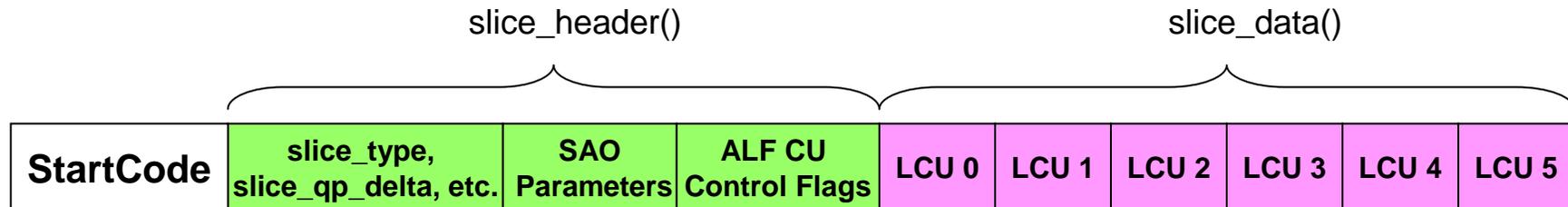
6th Meeting: Turin, IT, 13-22 July, 2011

Inconsistency in Order of Processing vs. Order of data in bitstream

- SAO and ALF are performed in the **last stage** of video coding



- However, SAO and ALF parameters are inserted in the slice header which is **before** slice data (generated from earlier stages of video coding)



- Inconsistency between order of processing vs. order of data transmission can result in issues for EPB insertion and CABAC encoding.

Proposed Changes

- Byte align slice header
 - To enable EPB insertion as bits are being produced
 - EPB insertion on slice data first, then on slice header
- For CAVLC, insert bit of 1 before byte alignment
 - To ensure that last byte in slice header is non-zero; this allows slice header and slice data to be easily stitched together without changing slice data
 - Similar to `rbsp_trailing_bits()` but for slice header
- Terminate CABAC after slice header
 - To enable CABAC encoding of bins as syntax elements are produced
- Note: Above only occurs if either SAO and/or ALF inserted syntax elements in slice header. Otherwise, no slice header termination required.

Experiment Results

- HM-3.0 under common conditions
- Simulation platform is LSF equipped with Intel(R) Xeon(R) CPU X5570@2.93GHz 64 bits Linux machines
- Results cross-checked by Samsung (F639)

Coding efficiency impact

	Intra	Random Access	Low Delay
HE (ALF on)	0.0	0.1	0.1
HE (ALF off)	0.0	0.1	0.1
LC (ALF on)	0.0	0.0	0.0
LC (ALF off)	0.0	0.0	0.0

Conclusions

- Byte align slice header to enable low delay EPB insertion
- Terminate CABAC in slice header such that syntax elements of slice_data does not have to wait for SAO or ALF
- Negligible impact on coding efficiency (up to 0.1%)
- Recommend for adoption into HEVC test model
 - Draft text available in contribution