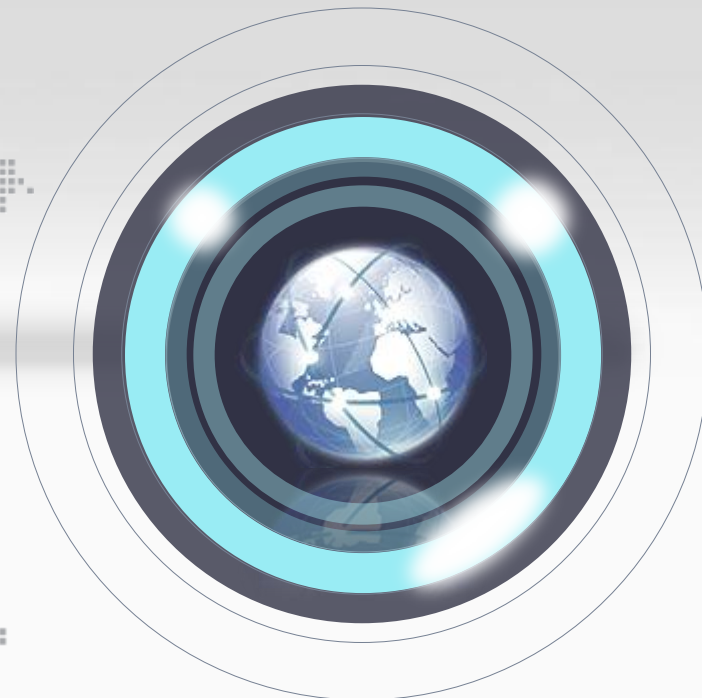


AHG5: Block size restriction of intra block copy

Sunil Lee, Chanyul Park, Elena Alshina, Chanyul Kim

Samsung Electronics



Contents

Inspire the World, Create the Future



I

Introduction

II

Algorithm Description

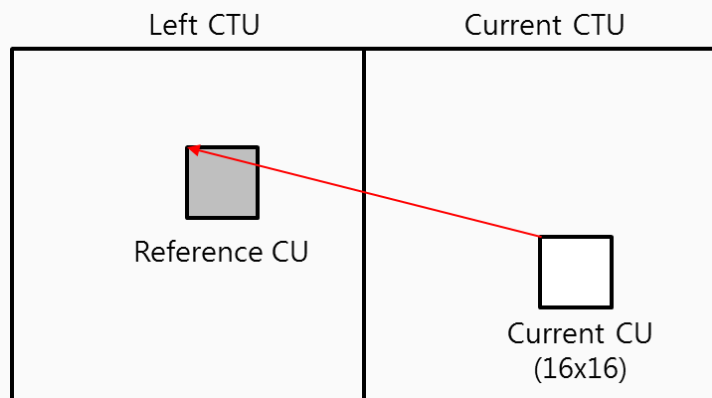
III

Test Results

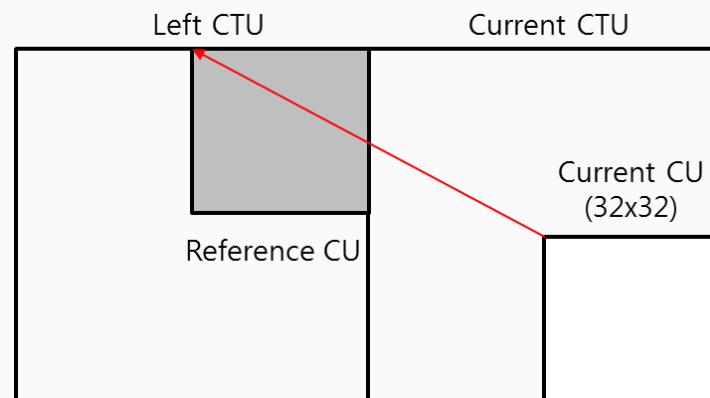
IV

Conclusion

- At the last meeting in Vienna, intra block copy (IBC) was adopted.
- Currently, IBC is allowed for all of the CU size. However, IBC is not effective for 32x32 and 64x64 CUs, according to the test results.
- Even in the latest HM12.0-RExt4.1 S/W, IBC for 32x32 and 64x64 CU is disabled at the encoder, when fast ME is enabled in IBC, as shown below.
- The main idea of this contribution is to normatively restrict the maximum block size in which IBC is enabled to avoid the redundant signaling of intra_bc_flag (common test condition: up to 16x16).



IBC is allowed for up to 16x16 CU



IBC is not allowed
for 32x32 and 64x64 CU



- It is proposed to add the syntax element, *log_max_intra_block_size_minus3*, which specifies the maximum CU size for which intra_bc_flag may be present.
- Its working principle is almost the same as the syntax element, *log2_max_transform_skip_size_minus2*, which specifies the maximum TU size for transform skip.
- The details of the changes in the text can be found in the proposal document.

Test Results (1/2)



- Anchor: HM12.0-RExt4.1 (AHG8, lossy coding anchor)
- Tested results: Normative restriction of IBC up to 16x16
- Some coding gain is observed, e.g. **-0.6% for SC YUV 444 in LB-MT**, while the encoding time is unchanged since IBC for 32x32 and 64x64 is already disabled by the encoder.

BD-rate Y	AI-MT	AI-HT	AI-SHT	RA-MT	RA-HT	LB-MT	LB-HT
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class B	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
SC RGB 444	0.0%	0.0%	0.0%	-0.1%	-0.1%	0.3%	0.2%
Animation RGB	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%
SC YUV 444	0.0%	0.0%	0.0%	-0.1%	-0.1%	-0.6%	-0.6%
Animation YUV	0.0%	0.0%	0.0%	0.0%	0.0%	-0.1%	0.0%
RangeExt	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
SC(444) GBR	0.1%	0.1%	0.0%	0.0%	0.1%	0.5%	0.8%
SC(444) YUV	-0.1%	0.0%	0.0%	-0.1%	-0.1%	-0.8%	0.3%
Enc Time[%]	101%	101%	101%	101%	101%	100%	100%
Dec Time[%]	102%	102%	101%	102%	102%	102%	101%



- Anchor: HM12.0-RExt4.1 (AHG8, lossless coding anchor)
- Tested results: Normative restriction of IBC up to 16x16
- The proposed normative restriction has almost no impact in lossless coding, since the bit-rate for texture coding is so dominant in lossless coding that the bit-rate saving by avoiding the redundant signaling of *intra_bc_flag* is not significant.

Bit-rate saving (Average)	AI-Main	RA-Main	LB-Main
Class F	0.0%	0.0%	0.0%
Class B	0.0%	0.0%	0.0%
SC RGB 444	0.0%	0.0%	0.0%
Animation RGB	0.0%	0.0%	0.0%
SC YUV 444	0.0%	0.1%	0.0%
Animation YUV	0.0%	0.0%	0.0%
RangeExt	0.0%	0.0%	0.0%
SC(444) GBR	0.0%	-0.1%	0.7%
SC(444) YUV	0.0%	0.2%	0.0%
Enc Time[%]	100%	100%	100%
Dec Time[%]	101%	100%	100%



- In this contribution, it is proposed to specify the maximum CU size in which intra block copy is enabled in HEVC range extension text to make the design clean and to avoid the redundant signaling of flags for intra block copy.
- It is also proposed to enable intra block copy up to 16x16 CU in the common test condition, considering the trade-off between the coding gain and complexity.
- It is suggested that the proposed change is adopted into the next version of HEVC range extension text and reference software.

Inspire the World, Create the Future

THANK YOU





- Anchor: HM12.0-RExt4.1 (AHG8, lossy coding anchor, fast ME in IBC disabled)
- Tested results: Normative restriction of IBC up to **16x16 (fast ME disabled)**
- The performance loss is marginal, while the encoding time is reduced by 7 ~ 11%.

BD-rate Y	AI-MT	AI-HT	AI-SHT	RA-MT	RA-HT	LB-MT	LB-HT
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	0.1%
Class B	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
SC RGB 444	0.2%	0.2%	0.2%	0.0%	0.0%	0.1%	0.2%
Animation RGB	0.0%	0.0%	0.0%	0.0%	0.0%	-0.1%	0.0%
SC YUV 444	0.3%	0.2%	0.2%	-0.2%	-0.1%	0.0%	0.0%
Animation YUV	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
RangeExt	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
SC(444) GBR	1.0%	1.0%	1.0%	0.9%	0.8%	0.1%	0.7%
SC(444) YUV	0.4%	0.3%	0.3%	0.4%	0.1%	-1.1%	-1.2%
Enc Time[%]	89%	89%	89%	93%	93%	93%	93%
Dec Time[%]	99%	99%	99%	99%	99%	98%	98%



- Anchor: HM12.0-RExt4.1 (AHG8, lossy coding anchor, fast ME in IBC disabled)
- Tested results: Normative restriction of IBC up to **8x8 (fast ME disabled)**
- Even though the encoding time reduction is significant (14~26%), the performance loss is not trivial any more.

BD-rate Y	AI-MT	AI-HT	AI-SHT	RA-MT	RA-HT	LB-MT	LB-HT
Class F	0.6%	0.4%	0.3%	0.4%	0.3%	0.5%	0.3%
Class B	0.1%	0.1%	0.0%	0.1%	0.0%	0.0%	0.0%
SC RGB 444	1.3%	1.3%	1.1%	1.0%	0.8%	0.2%	0.3%
Animation RGB	0.3%	0.1%	0.1%	0.1%	0.1%	0.1%	0.0%
SC YUV 444	2.0%	1.7%	1.5%	1.3%	1.1%	1.8%	1.3%
Animation YUV	0.4%	0.2%	0.1%	0.1%	0.1%	0.1%	0.0%
RangeExt	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
SC(444) GBR	7.7%	7.6%	7.3%	6.6%	6.4%	2.6%	2.0%
SC(444) YUV	5.4%	4.4%	3.6%	4.5%	3.6%	1.6%	0.8%
Enc Time[%]	74%	74%	74%	84%	84%	86%	86%
Dec Time[%]	104%	103%	103%	100%	99%	99%	99%