JCTVC-N0276

Hardware oriented implementation on HEVC encoding

14th JCT-VC meeting

Renesas Electronics Corporation

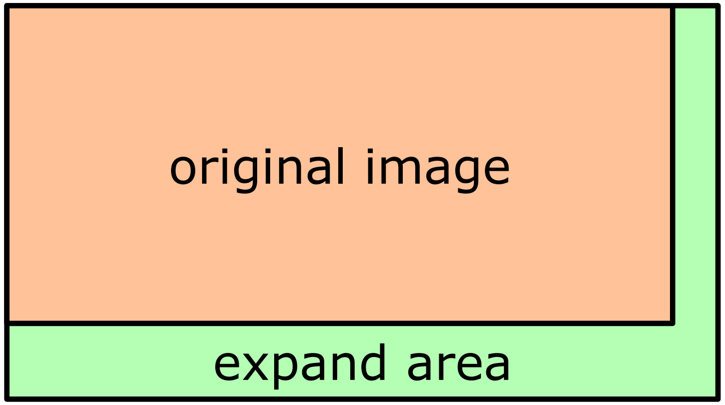
Ryoji HASHIMOTO, Seiji Mochizuki, Kenichi Iwata

Summary

- Hardware oriented implementation on HEVC encoding
  - Simple control logic
  - Little loss in coding efficiency

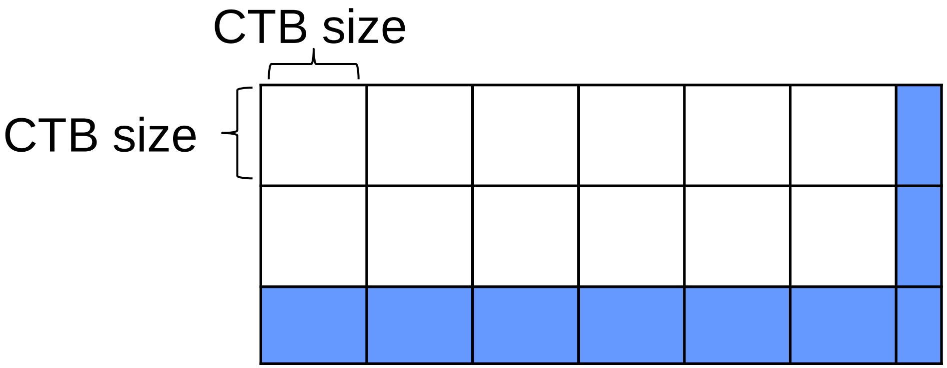
- Simplify control logic
  - Expand image to multiple of not CU size but CTB size
- Avoid loss in coding efficiency
  - Truncation of quantized coefficients in expanded area
  - Optimization in TU size partitioning

- BD rate Performance on HM11.0
  - AI-Main -0.9 %
  - RA-main -1.26 %



Motivation

- In developing a hardware of HEVC codec
  - Non-unique block size leads complex control logic in CTB, especially at right and bottom corner (blue area)



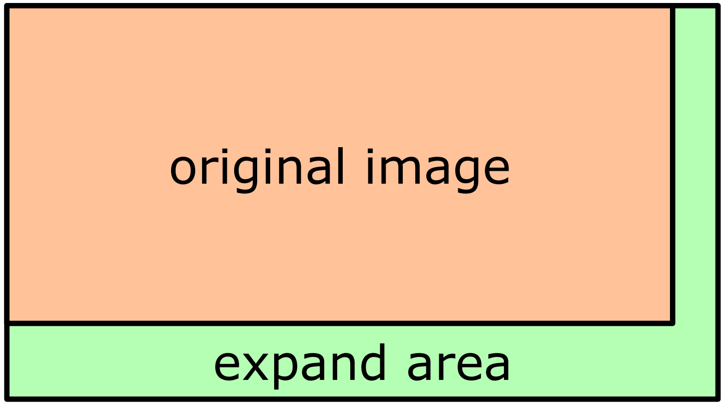
Specification of our hardware

Feature	Despription
Supported function	Encode/Decode
Profile/Level	Main/Level 5.1
Maximum resolution	4096x2176@30fps
Maximum bitrate	120 Mbps
Operation Frequency	260 MHz for 4096x2176@30fps

Expansion of input image

- Expand to multiple of CU size (same as HM)
  - Advantage
    - small overhead in coding efficiency
  - Disadvantage
    - Non-unique size at the corner in picture

- Expand to multiple of CTB size
  - Advantage
    - Unique size in picture makes control simpler
  - Disadvantage
    - large overhead,
      especially in large CTB size

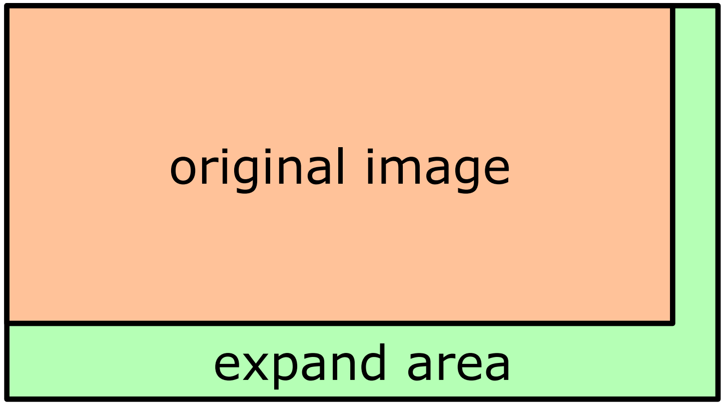


Is there any good way for simple control and small over head?

Truncation of quantized coefficient

- In decoder side, user does not realize the image quality of expanded area at all
  - We can reduce bits in expanded area without loss of image quality in original image

- Method for bit reduction
  - Truncate all quantized coefficient to 0 in a TU which locates at expanded area
    - No effect to the image quality in original image
  - In TU size decision, select TU size which does not exceed picture boundary
    - Increase the # of TU to be truncated



Experimental results with HM11(1)

- condition 1 : expand to CU size (HM default)
- condition 2 : expand to CTB size without coefficient truncation
- condition 3 : expand to CTB size with coefficient truncation

cond 2 against cond1

	AI-Main	RA-Main	LP-main
Class A	0.00	0.00	-
Class B	0.22	0.29	0.48
Class C	1.19	2.85	3.39
Class D	2.68	6.37	7.29
All	1.03	2.25	3.47

cond 3 against cond1

	AI-Main	RA-Main	LP-Main
Class A	0.00	0.00	-
Class B	0.11	0.43	0.26
Class C	0.10	1.04	0.62
Class D	0.32	2.34	2.07
All	0.15	0.92	0.93

Experimental results with HM11(2)

- condition 1 : expand to CU size (HM default)
- condition 2 : expand to CTB size without coefficient truncation
- condition 3 : expand to CTB size with coefficient truncation

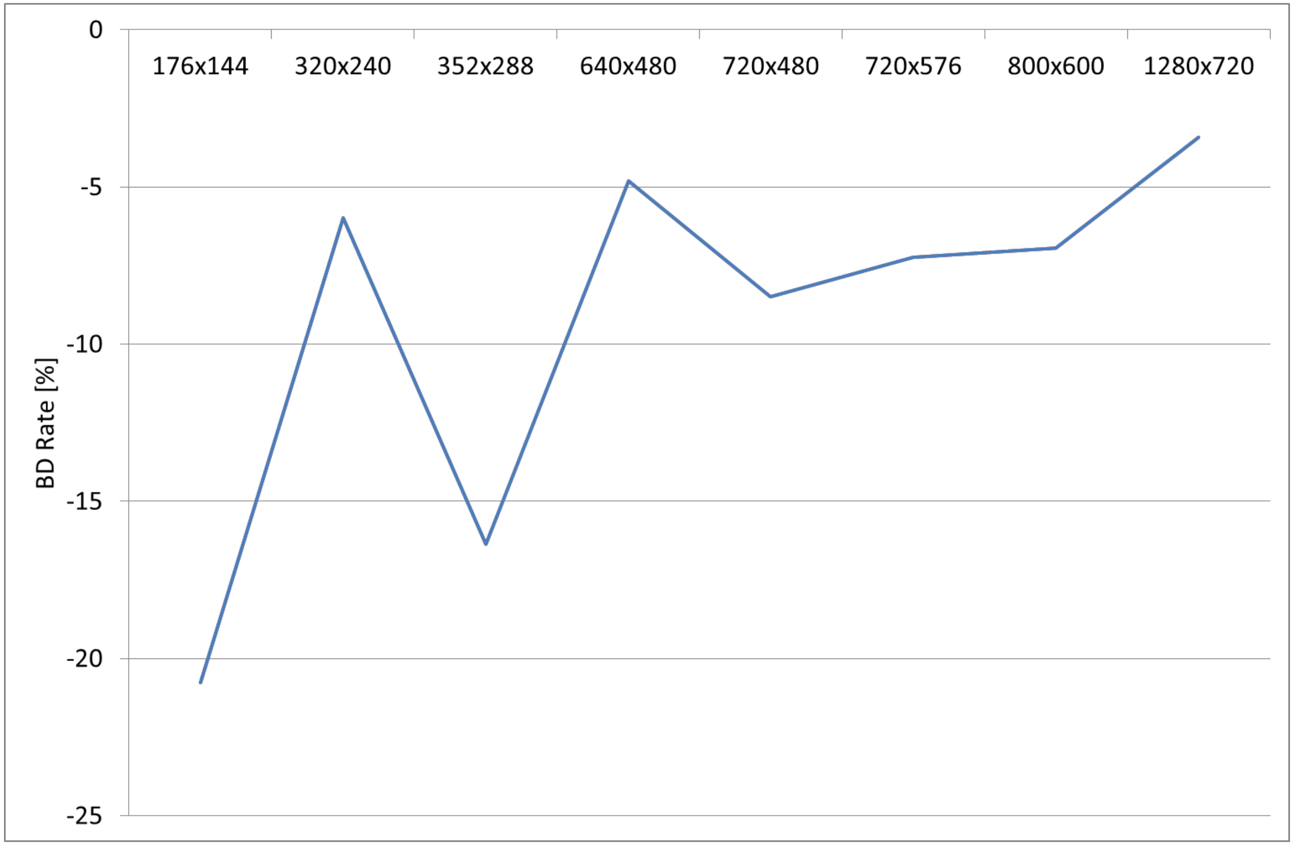
cond 3 against cond2

	AI-Main	RA-Main	LP-main
Class A	-0.86	0.00	-
Class B	-0.11	0.14	-0.22
Class C	-1.18	-1.76	-2.68
Class D	-2.30	-3.78	-4.83
All	-0.86	-1.26	-2.40

Our hardware can save many bits
by adopting this method

Experimental results with HM11(3)

- Test on various major image size in LP-Main condition
  - Tested by cut downed class B sequences (top left samples)



Proposed method is effective in various major image size
Especially, the number of pixels in non-displayed area is large.

Conclusion

- Our hardware adopts simple control and less overhead encoding methods
  - Coefficient truncation in expand area
  - Optimization of TU size partitioning

- Performance evaluation of proposed method in HM11
  - 0.15/0.92/0.93 overhead against expansion based on CU size
  - 0.86/1.26/2.40 reduction against the method without truncation