JCTVC-H0271

AHG9: CABAC with Constrained Outstanding Bits

Tzu-Der (Peter) Chuang, Ching-Yeh Chen, Yu-Wen Huang, Shawmin Lei

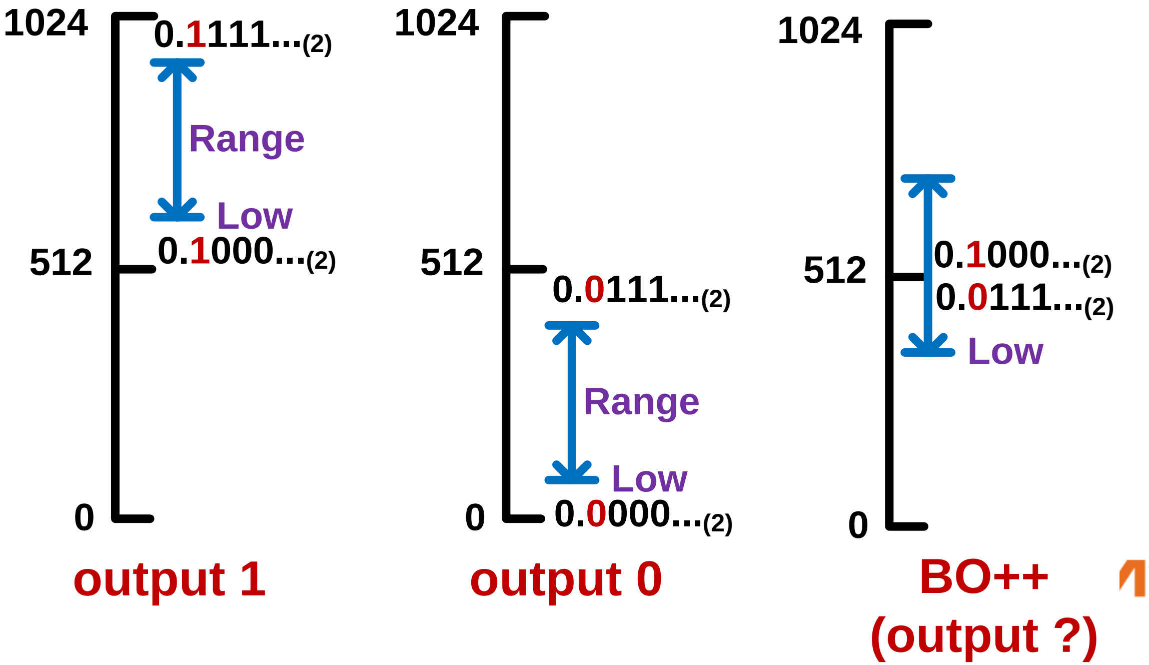
Presented by Tzu-Der (Peter) Chuang,
8th JCT-VC Meeting in San José
1–10 February, 2012

Overall Summary

- Two coding interval adjustment procedures to constrain the bit count of continuous outstanding bits
  - No constraint on continuous outstanding bits in CABAC now and can cause a sudden burst of hundreds of (theoretically, infinite) outputted bits to the bitstream buffer in one clock cycle
- Method 1 – selecting the larger range interval
  - Coding interval is divided into two parts, the larger one is selected as new coding interval to flush outstanding bits
- Method 2 – sending one bypass bin
  - Simply code a bypass bin to flush outstanding bits
- No change in coding efficiency and run time while at most 18 bits could be outputted at the same time

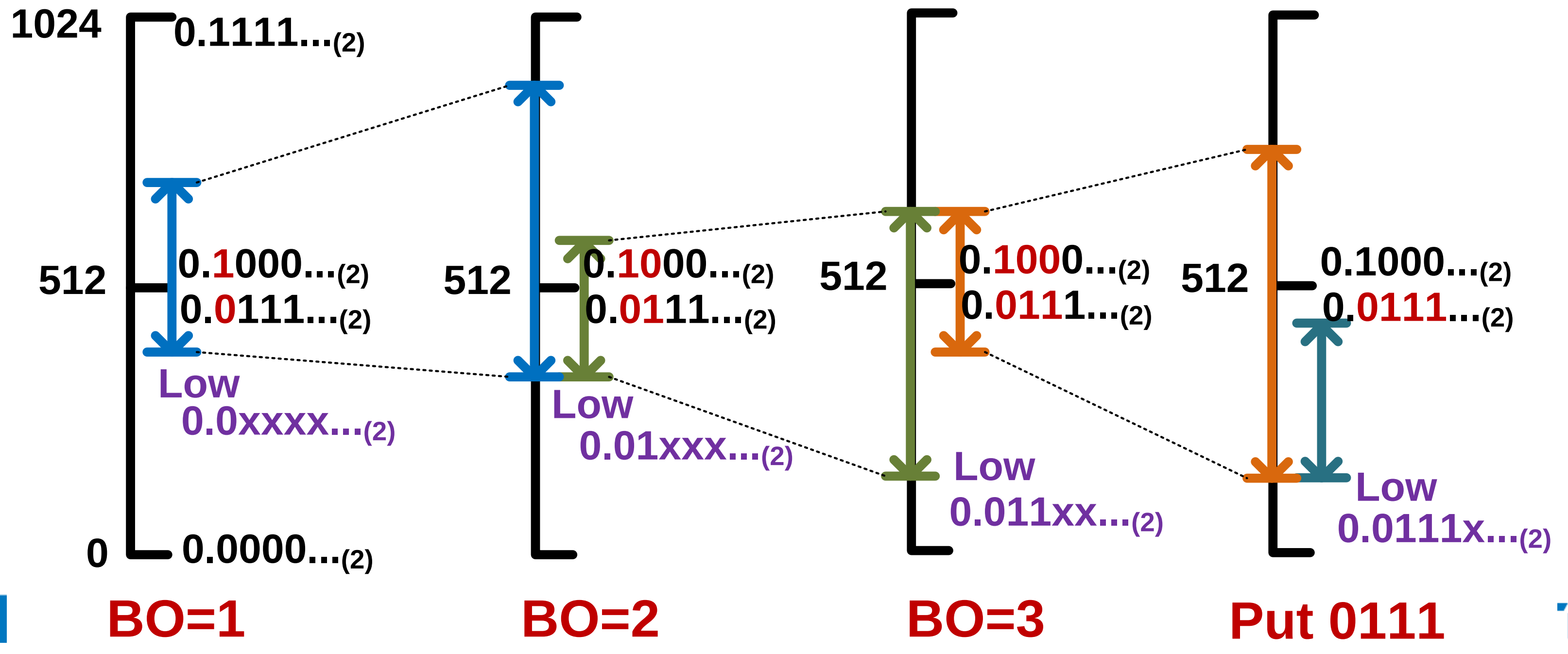
Outstanding Bits in CABAC

- When range < 256, renormalization is required.
  - Intended to output 1 bit depending on the Low value
- If the coding interval is fully placed in upper or lower half, output one bit of 1 or 0.
- Otherwise, bitOutstanding (BO) is increased by 1
  - Output bit cannot be determined as 1 or 0
  - BO means the bit count of the outstanding bits that should be outputted



Outstanding Bits in CABAC

- If the coding interval crosses 512, BO is increased by 1.
  - Cannot output BO bits since we do not know the output bits are 1000000… or 0111111…
- Unless the coding interval is fully placed in the upper half or the lower half, 1000000… or 0111111… can be outputted.

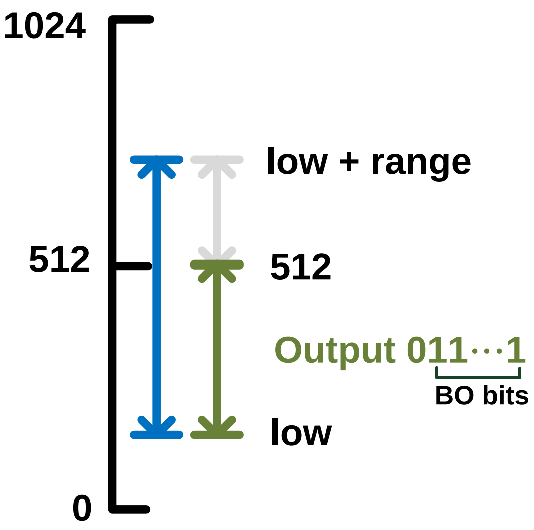


Unlimited Outstanding Bits

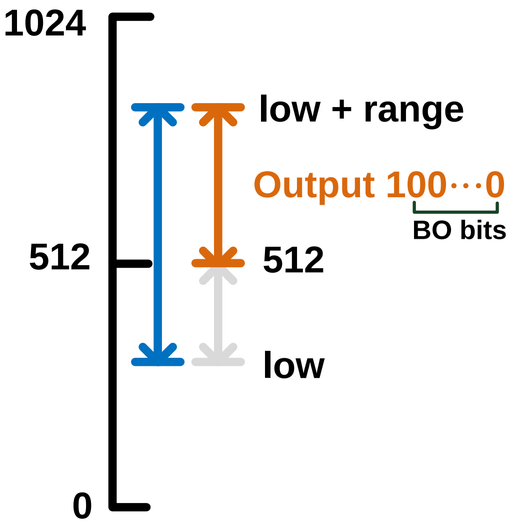
- No constraint on BO in CABAC
  - The coding interval can always cross 512, and BO would increase indefinitely.
  - The BO number up to 189 is observed in the common test condition of HM-3.0.
- Unlimited outstanding bits will increase the difficulty of hardware implementation
  - There may be a sudden burst of many outputted bits to the bitstream buffer in one clock cycle.

Proposed Method 1

- Define a outstanding bits constraint, MaxBO
- After renormal., if BO >= MaxBO, do interval adjustment
- Divide coding interval into two parts, the cut-off point is 512
- Select the larger part as new coding interval
  - Can output 0111… or 1000… right away



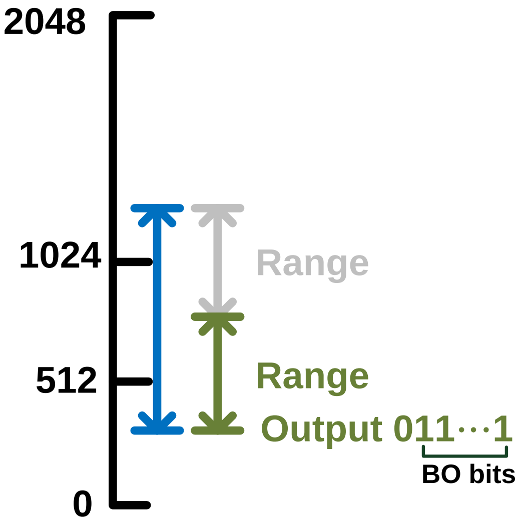
BO >= maxBO,
Lower part >= upper part,
Select lower part,
Output 01⋯1,
BO reset to 0



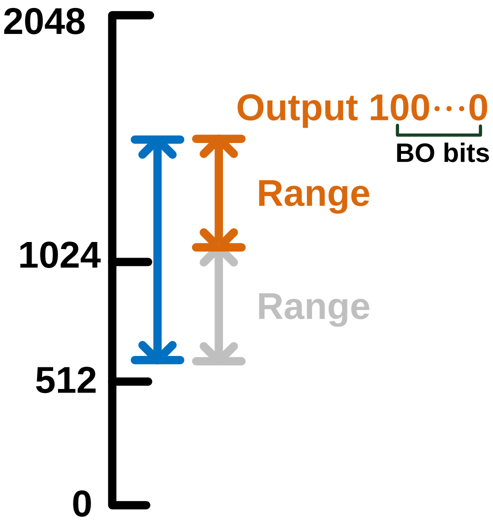
BO >= maxBO,
Lower part < upper part,
Select upper part,
Output 10⋯0,
BO reset to 0

Proposed Method 2

- Divide coding interval into two equal parts
  - Double the low and coding interval as the bypass bin coding
- Select the part that does not cross 1024
  - Can output 0111… or 1000… right away
- The range is not changed after coding interval adjustment
  - It will not affect the rangeLPS derivation of the next bin



BO >= maxBO,
Lower part >= upper part,
Select lower part,
Output 01···1,
BO reset to 0



BO >= maxBO,
Lower part < upper part,
Select upper part,
Output 10···0,
BO reset to 0

Limited Outstanding Bits

- If the maxBO is set equal to N, at most (N+6) bits could be outputted at the same time
  - BO can be increased by at most 7 in the renormalization
  - (N+6) continuous outstanding bits outputted when
    - After renormalization, the BO equals to (N-1)
    - In the renormalization of next bin, the BO is increased by 7
- When maxBO is set equal to 12, at most 18 bits could be outputted at the same time

Results of Method 1

- JCTVC-G1200 anchor
- MaxBO is equal 12
- No coding efficiency loss
- Run time is roughly the same as the anchor

	All Intra HE			All Intra LC		
	Y	U	V	Y	U	V
Class A (8bit)	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%
Class C	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class D	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class E	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Overall	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	101%			101%		

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A (8bit)	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%
Class C	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class D	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class E						
Overall	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	100%			100%		

	Low Delay B HE			Low Delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class C	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class D	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class E	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Overall	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	100%			100%		

Results of Method 2

- JCTVC-G1200 anchor
- MaxBO is equal 12
- No coding efficiency loss
- Run time is roughly the same as the anchor

	All Intra HE			All Intra LC		
	Y	U	V	Y	U	V
Class A (8bit)	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%
Class C	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class D	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class E	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Overall	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	101%			102%		

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A (8bit)	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%
Class C	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class D	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class E						
Overall	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	100%			100%		

	Low Delay B HE			Low Delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class C	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class D	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class E	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Overall	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	100%			100%		

Cross Verification

- We thank Fujitsu for crosschecking our proposal
  - JCTVC-H0648
- BD-rates and run times are confirmed

Conclusions

- Proposed two coding interval adjustment procedures
  - Set a constraint on the bit count of continuous outstanding bits
  - Method 1 – divide the coding interval into two parts and select the one is fully placed in upper or lower half
  - Method 2 – send one bypass bin
  - Makes hardware implementation much easier
  - Method 1 has better coding efficiency while method 2 is simpler
- We suggest to set maxBO equal to 12
  - No change in coding efficiency and run time
  - At most 18 bits could be outputted at the same time