



CABAC with Constrained Outstanding Bits

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



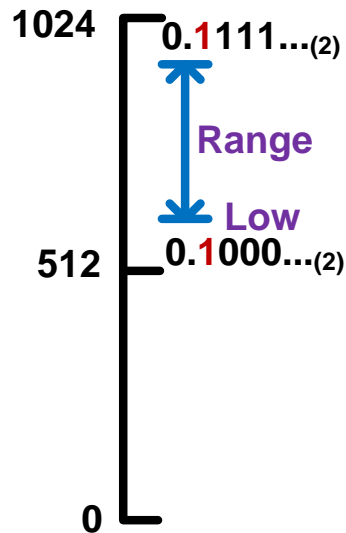
Overall Summary

- A coding interval adjustment procedure to constrain the bit count of continuous outstanding bits
 - There is no constraint on continuous outstanding bits in CABAC
 - There may be a sudden burst of many (theoretically, infinite) outputted bits to the bitstream buffer in one clock cycle
 - Range is divided into two parts, the larger one is selected as new coding interval to prevent infinite outstanding bits
- When setting maxBO equal to 10 or 12, no change in average coding efficiency and run time while at most 16 or 18 bits could be outputted at the same time

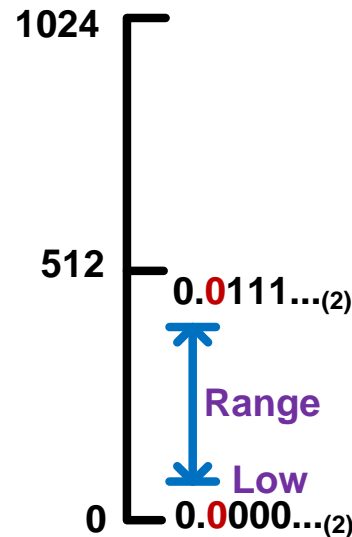
	AI-HE	RA-HE	LD-HE
BD-rate	0.0	0.0	0.0
Enc Time[%]	100%	101%	100%
Dec Time[%]	102%	100%	100%

Outstanding Bits in CABAC

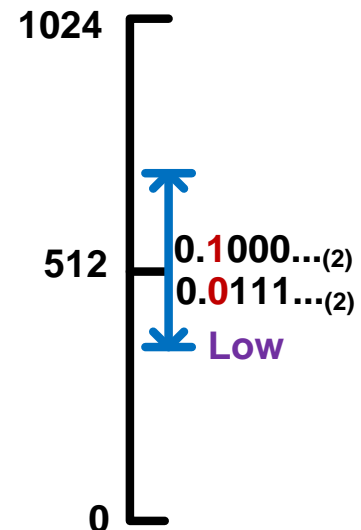
- When range < 256, renormalization is required
 - Intended to output 1 bit depending on the Low value
- If the coding interval is fully placed in upper or lower half, output one bit of 1 or 0
- Otherwise, bitOutstanding (BO) is increased by 1
 - Output bit cannot be determined as 1 or 0
 - BO means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



output 1



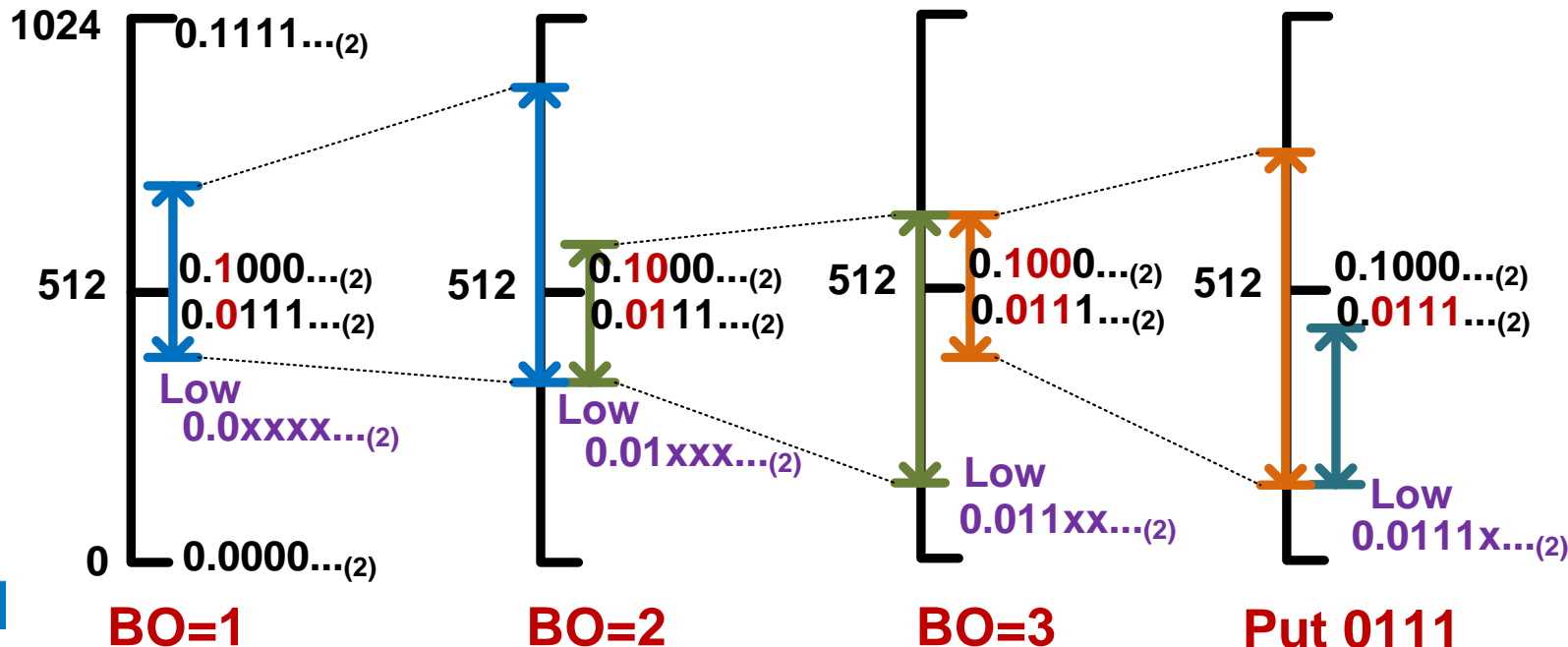
output 0



BO++
(output ?)

Outstanding Bits in CABAC

- If the coding interval crosses 512, BO is increased by 1
 - Cannot output BO bits since we don't know the output bits are 1000... or 0111...
- After the coding interval is fully placed in the upper half or the lower half, 1000... or 0111... be outputted.



No Constraint on Outstanding Bits in CABAC

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

Constraint on Continuous 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

Simulation Results

- No bit rate increase when MaxBO is equal to 10 or 12
 - At most 16 or 18 bits could be outputted at the same time

	MaxBO	4	6	8	10	12	14	16	24
AI	Class A	0.9	0.2	0.1	0.0	0.0	0.0	0.0	0.0
	Class B	0.9	0.2	0.1	0.0	0.0	0.0	0.0	0.0
	Class C	0.9	0.2	0.1	0.0	0.0	0.0	0.0	0.0
	Class D	0.9	0.2	0.1	0.0	0.0	0.0	0.0	0.0
	Class E	0.9	0.2	0.0	0.0	0.0	0.0	0.0	0.0
	All	0.9	0.2	0.1	0.0	0.0	0.0	0.0	0.0
	Enc Time[%]	100%	100%	100%	100%	100%	99%	100%	100%
	Dec Time[%]	101%	102%	101%	101%	102%	102%	102%	102%
RA	Class A	0.8	0.2	0.0	-0.1	-0.1	0.0	0.0	0.0
	Class B	0.9	0.2	0.0	0.0	0.0	0.0	0.0	0.0
	Class C	0.9	0.2	0.0	0.0	0.0	0.0	0.0	0.0
	Class D	0.9	0.2	0.0	0.0	0.0	0.0	0.0	0.0
	Class E								
	All	0.9	0.2	0.0	0.0	0.0	0.0	0.0	0.0
	Enc Time[%]	100%	100%	100%	100%	101%	100%	100%	100%
	Dec Time[%]	100%	99%	100%	100%	100%	100%	100%	100%
LD	Class A								
	Class B	0.9	0.2	0.0	0.0	0.0	0.0	0.0	0.0
	Class C	0.9	0.2	0.1	0.0	0.0	0.0	0.0	0.0
	Class D	0.9	0.2	0.0	0.0	0.0	0.0	0.0	0.0
	Class E	0.8	0.2	0.0	0.1	0.0	0.0	-0.1	0.0
	All	0.9	0.2	0.0	0.0	0.0	0.0	0.0	0.0
	Enc Time[%]	100%	100%	100%	100%	100%	100%	100%	100%
	Dec Time[%]	100%	100%	100%	100%	100%	100%	100%	100%

Cross Verification

- We thank HHI for crosschecking our proposal
 - JCTVC-F452
- BD-rates and run times are confirmed

Conclusions

- Proposed a coding interval adjustment procedure
 - Set a constraint on the bit count of continuous outstanding bits
 - Range is divided into two parts, the larger one is selected as new coding interval to prevent infinite outstanding bits
 - Makes hardware implementation much easier
- We suggest to set maxBO equal to 10 or 12
 - No change in coding efficiency and run time
 - At most 16 or 18 bits could be outputted at the same time