JVET-M0172

CE5.1.9: CABAC engine with simplified range sub-interval derivation

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

Presented by Tzu-Der (Peter) Chuang
13th Meeting: Marrakech, MA
9–18 Jan

Overall Summary

- A double-window probability estimation model with linear state representation and simplified range sub-interval derivation is proposed
  - P0 += 1023*Y – (P0 >> 4)
  - P1 += 16383*Y – (P1>> 7)
  - P = (P0 <<4) + P1
- Proposed to use a 32x8x8-bits LUT or a 6-bit by 5/4-bit multiplier for rangeLPS derivation
  - Results of using multiplier method are identical to those of using LUT
  - Table size is 1/144 of that in JEM
- -0.76%/-0.55%/-0.49% BD-rate in AI/RA/LB

RangeLPS Derivation in JEM

- In JEM, a two-parameter counter-based CABAC is adopted
  - $P'_0 = P_0 + ((32768*y - P_0)>>K)$
  - $P'_1 = P_1 + ((32768*y - P_1)>>8)$
  - $P = (P_0 + P_1)/2$
- A large look-up table (LUT) is used to derive the RangeLPS
  - $512\times64\times9$-bit LUT is used
  - The table size is 144x of the size of the rangeLPS LUT used in HEVC

A Double-Window Probability Estimation Model with Linear State Representation

- Similar to the JEM fixed double-window CABAC
  - Two probability states for a context. The size of buffers are 10 and 14 bits buffers.

```
P0 -= (P0 >> 4)
P1 -= (P1 >> 7)
if(bin == 1){
   P0 += 1023
   P1 += 16383
}
P = (P0 << 4) + P1
```

Proposed Method-A

- Proposed to use a 32x8x8-bits LUT or a 6-bit by 5-bit multiplier for rangeOne derivation
  - The LUT size is the same as the LUT used in HEVC
  - Results of using multiplier method are identical to those of using LUT
  - Users can choose to use LUT or multiplier in their implementations

probIdx = (P >= 16384) ? 63 – (P>>9) : (P>>9)

rangeLPS = ((2 × probIdx +1) × (2 × (range >> 5) + 1)) >> 3

or

rangeLPS = LUT_32x8[probIdx][(range >> 5) & 7]

Proposed Method-B

- Proposed to use a 32x8x8-bits LUT or a 6-bit by 4-bit multiplier for rangeOne derivation
  - In method-B, it guarantees that during renormalization after processing an MPS, the range register is left shifted by at most one bit position

```
probIdx   = ( P >= 16384 ) ? 63 – (P>>9) : (P>>9)
rangeLPS  = ((2 × probIdx +1) × (range >> 5 ) >> 3
 or
rangeLPS  = LUT_32x8[ probIdx ][ ( range >> 5 ) & 7 ]
```

Supplementary Data

- Method-A supplementary:
  - Use a 5-bits by 5-bits multiplier

probIdx = (P >= 16384) ? 31 – (P>>10) : (P>>10)
rangeLPS = ((2 × probIdx +1) × (2 × (range >> 5) + 1)) >> 2

- Method-B supplementary :
  - Use a 5-bits by 4-bits multiplier

probIdx = (P >= 16384) ? 31 – (P>>10) : (P>>10)
rangeLPS = ((2 × probIdx +1) × (range >> 5)) >> 2

Results

- The proposed methods are implemented on VTM3.0

Method		VTM		
		Y	U	V
Method-A, 32x8x8 LUT or 6-bit x 5-bit multiplier	AI	-0.79	-1.05	-1.18
	RA	-0.57	-1.05	-1.00
	LB	-0.50	-0.82	-0.66
Method-B, 32x8x8 LUT or 6-bit x 4-bit multiplier	AI	-0.76	-1.02	-1.15
	RA	-0.55	-1.03	-0.98
	LB	-0.49	-0.81	-0.65
Method-A, 16x8x8 LUT or 5-bit x 5-bit multiplier	AI	-0.72	-0.98	-1.11
	RA	-0.51	-0.99	-0.94
	LB	-0.45	-0.77	-0.61
Method-B, 16x8x8 LUT or 5-bit x 4-bit multiplier	AI	-0.70	-0.95	-1.08
	RA	-0.49	-0.97	-0.92
	LB	-0.43	-0.75	-0.59

Thank Samsung for cross-checking

Conclusions

- Proposed a double-window probability estimation model with linear state representation and simplified range sub-interval derivation
  - 10-bits and 14-bits buffers are used for a context
- Proposed to use a 32x8x8-bits LUT or a 6-bit by 5/4-bit multiplier for rangeLPS derivation
  - Results of using multiplier method are identical to those of using LUT
  - Table size is 1/144 of that in JEM
- -0.76%/-0.55%/-0.49% BD-rate in AI/RA/LB