AHG8: On History-Based Rice Parameter Derivations for Wavefront Parallel Processing

JVET-X0128

Yue Yu, Haoping Yu, Zhihuang Xie, Fan Wang, Luhang Xu, Dong Wang

History-based Rice parameter derivation

- A history-based Rice parameter derivation method JVET-V0106 was adopted for VVC v2;
- A HistValue is calculated to update locSumAbs to derive new Rice parameter;
- A counter/color, StatCoeff[3] is utilized and may be updated once per TU from the first abs_remainder[] or dec_abs_level[]

StatCoeff[i] = (StatCoeff[i] +Floor(Log2(abs_remainder[]) + 2) >> 1

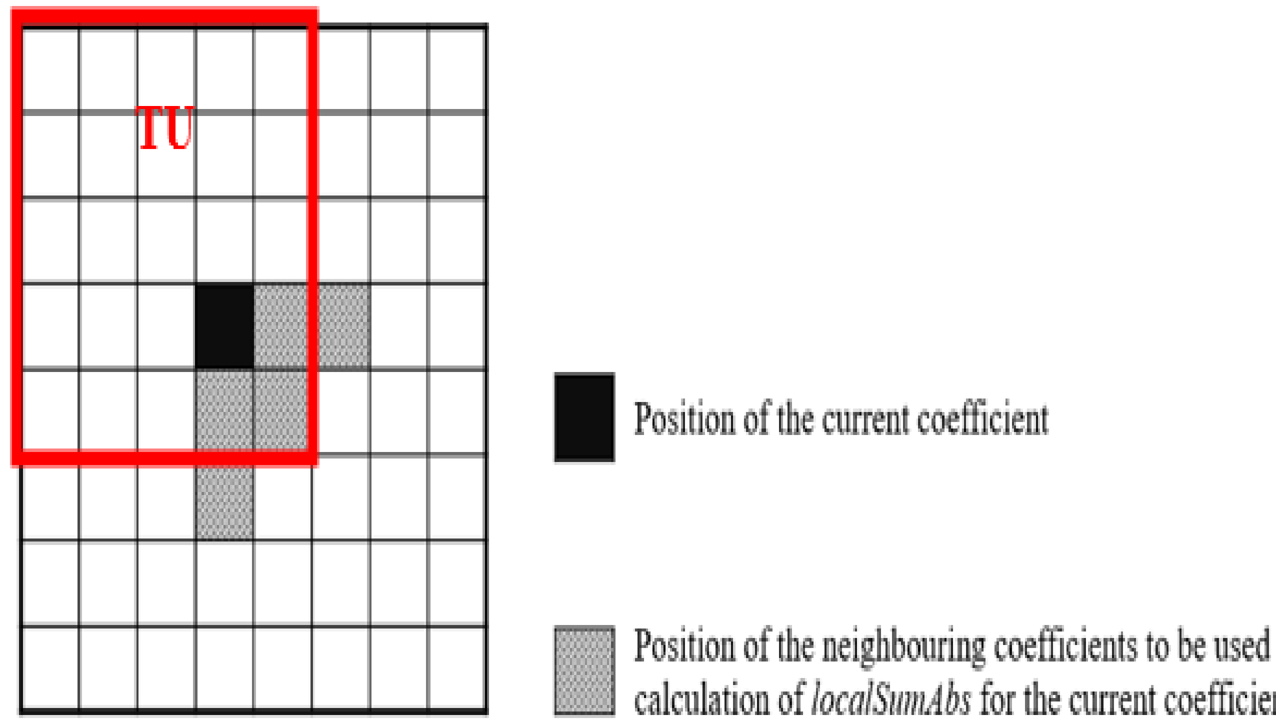
or

StatCoeff[i] = (StatCoeff[i] +Floor(Log2(abs_remainder[])) >> 1

Prior to TU coding, HistValue = 1 << StatCoeff[i]

and StatCoeff[i] is initialed as 2* Floor(Log2(BitDepth - 10))

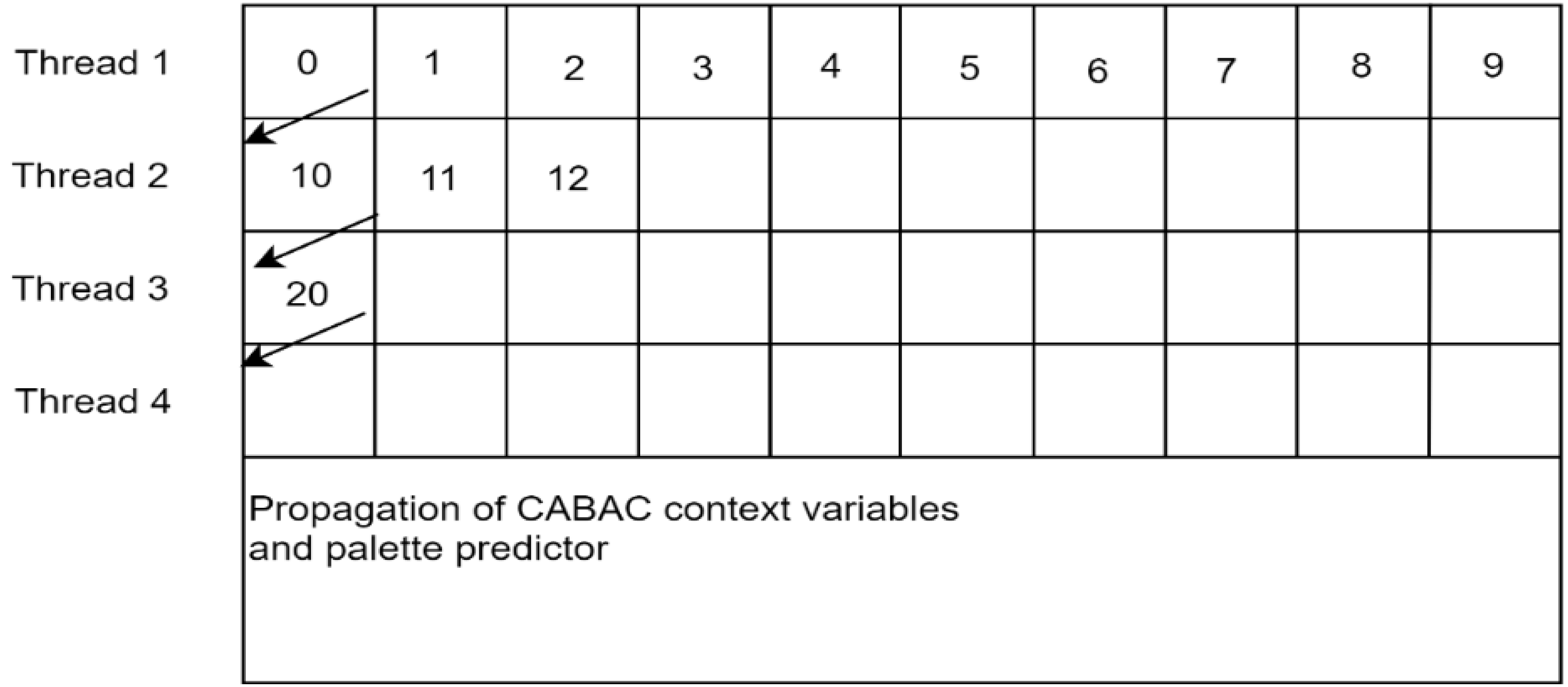
- There is TU dependency to update StatCoeff[]




```
locSumAbs = 0
if( xC < ( 1 << log2TbWidth ) − 1 ) {
    locSumAbs += AbsLevel[ xC + 1 ][ yC ]
    if( xC < ( 1 << log2TbWidth ) − 2 )
        locSumAbs += AbsLevel[ xC + 2 ][ yC ]
    else
        locSumAbs += HistValue
    if( yC < ( 1 << log2TbHeight ) − 1 )
        locSumAbs += AbsLevel[ xC + 1 ][ yC + 1 ]
    else
        locSumAbs += HistValue
} else
    locSumAbs += 2 * HistValue
if( yC < ( 1 << log2TbHeight ) − 1 ) {
    locSumAbs += AbsLevel[ xC ][ yC + 1 ]
    if( yC < ( 1 << log2TbHeight ) − 2 )
        locSumAbs += AbsLevel[ xC ][ yC + 2 ]
    else
        locSumAbs += HistValue
} else
    locSumAbs += HistValue
```

Wavefront Parallel Process (WPP)

- When WPP is enabled, each CTU row in a frame constitutes a separation partition, and the content of the adapted CABAC context variables and palette predictor are propagated from the first coded CTU of the preceding CTU row to the first CTU of the current CTU row.



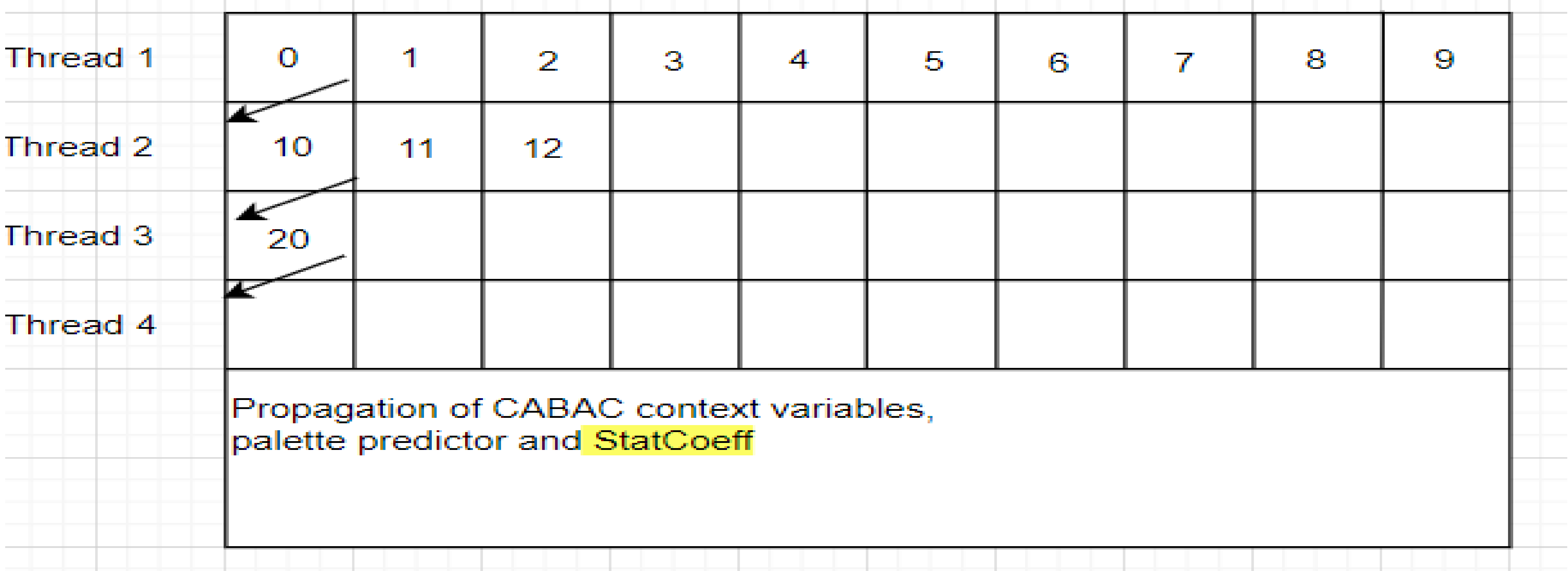
The Problem

- When history-based Rice parameter derivation is enabled, the StatCoeff in the preceding CTU row will be passed to the first TU in the current CTU row.

- When WPP is enabled, history-based Rice parameter derivation cannot be used.

Proposed Method 1

- Saving StatCoeff[cIdx] as StatCoeffWpp[cIdx], after the last TU of the first CTU in each CTU row is coded.
- Starting from the second CTU row, prior to the first TU coding, the synchronization process for Rice parameter derivation is applied to the first CTU for each CTU row. StatCoeff[cIdx] is synchronized with the saved StatCoeffWpp[cIdx] from the preceding CTU row.



Proposed Method 1

Specification Change

- **9.3.2.x1 Storage process for Rice parameter derivation**

```
for(cIdx = 0; cIdx < 3; cIdx++){
   StatCoeffWpp[cIdx] = StatCoeff[cIdx]
}
```

- **9.3.2.x2 Synchronization process for Rice parameter derivation**

```
for(cIdx = 0; cIdx < 3; cIdx++){
   StatCoeff[cIdx] = StatCoeffWpp[cIdx]
}
```

Proposed Method 2

- A fixed initial value for StatCoeff[cIdx] is used for coding the first abs_remainder[ ] or dec_abs_level[ ] in each CTU row. An initial value can be as follows,

$$\text{StatCoeff}[\ idx\ ] = 2 * \text{Floor}(\ \text{Log2}(\ \text{BitDepth} - 10\ )$$

where BitDepth specifies the bit depth, and Floor(x) represents the largest integer less than or equal to x.

Specification Change

- **9.3.2.x2 Synchronization process for Rice parameter derivation**

  for(cIdx = 0; cIdx < 3; cIdx++){

  StatCoeff[cIdx] = 2 * Floor(Log2(BitDepth − 10)

  }

Simulation Results

- Simulation conditions follow the CTC for high bit-depth coding;
- Method 1 over VTM-14.0 with WPP off, low QP, lossy.

HDR PQ	AI							
	Over VTM12.0_cebase							
	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV	EncT	DecT
PQ444	#VALUE!	#VALUE!	#VALUE!	0.03%	0.04%	0.02%	105%	106%
PQ422	#VALUE!	#VALUE!	#VALUE!	0.04%	0.04%	0.02%	107%	106%
Overall	#VALUE!	#VALUE!	#VALUE!	0.04%	0.04%	0.02%	106%	106%

	LDB							
	Over VTM12.0_cebase							
	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV	EncT	DecT
PQ444	#VALUE!	#VALUE!	#VALUE!	0.06%	0.06%	0.03%	91%	100%
PQ422	#VALUE!	#VALUE!	#VALUE!	0.12%	0.10%	0.04%	89%	93%
Overall	#VALUE!	#VALUE!	#VALUE!	0.09%	0.08%	0.03%	90%	96%

	RA							
	Over VTM12.0_cebase							
	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV	EncT	DecT
PQ444	0.08%	0.11%	0.01%	0.07%	0.11%	0.02%	104%	105%
PQ422	0.12%	0.06%	0.01%	0.11%	0.06%	0.01%	101%	102%
Overall	0.10%	0.09%	0.01%	0.09%	0.08%	0.01%	102%	103%

Overall PQ	#VALUE!	#VALUE!	#VALUE!	0.07%	0.07%	0.02%	99%	102%

HDR HLG	AI				
	Over VTM12.0_cebase				
	psnrY	psnrU	psnrV	EncT	DecT
HLG444	0.02%	0.01%	0.01%	108%	114%
HLG422	0.02%	0.01%	0.01%	103%	106%
Overall	0.02%	0.01%	0.01%	106%	110%

	LDB				
	Over VTM12.0_cebase				
	psnrY	psnrU	psnrV	EncT	DecT
HLG444	0.08%	0.04%	0.00%	89%	75%
HLG422	0.09%	0.04%	0.01%	91%	81%
Overall	0.08%	0.04%	0.00%	90%	78%

	RA				
	Over VTM12.0_cebase				
	psnrY	psnrU	psnrV	EncT	DecT
HLG444	0.09%	0.06%	0.01%	100%	93%
HLG422	0.10%	0.06%	0.02%	101%	91%
Overall	0.10%	0.06%	0.02%	100%	92%

Overall HLG	0.07%	0.04%	0.01%	99%	93%

SVT RGB	AI				
	Over VTM12.0_cebase				
	psnrG	psnrB	psnrR	EncT	DecT
SVT16	0.01%	0.02%	0.01%	104%	99%
SVT12	0.01%	0.01%	0.01%	107%	101%
Overall	0.01%	0.01%	0.01%	105%	100%

	LDB				
	Over VTM12.0_cebase				
	psnrG	psnrB	psnrR	EncT	DecT
SVT16	0.03%	0.02%	0.02%	94%	94%
SVT12	0.02%	0.01%	0.01%	93%	94%
Overall	0.02%	0.02%	0.01%	94%	94%

	RA				
	Over VTM12.0_cebase				
	psnrG	psnrB	psnrR	EncT	DecT
SVT16	0.03%	0.02%	0.02%	105%	98%
SVT12	0.02%	0.01%	0.01%	105%	103%
Overall	0.03%	0.02%	0.01%	105%	100%

Overall RGB	0.02%	0.01%	0.01%	101%	98%

- Method 1 over VTM-14.0 with WPP off, lossless

PQ	All Intra			Low delay B			Random Access		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM-14.0	proposed		VTM-14.0	proposed		VTM-14.0	proposed	
PQ444	2.6	2.6	0.01%	3.1	3.1	0.01%	3.1	3.1	0.01%
PQ422	2.4	2.4	0.01%	2.9	2.9	0.01%	2.9	2.9	0.01%
Overall	**2.5**	**2.5**	**0.01%**	**3.0**	**3.0**	**0.01%**	**3.0**	**3.0**	**0.01%**
Enc Time[%]	95%			106%			99%		
Dec Time[%]	98%			106%			108%		

HLG	All Intra			Low delay B			Random Access		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM-14.0	proposed		VTM-14.0	proposed		VTM-14.0	proposed	
HLG444	1.8	1.8	0.02%	2.0	2.0	0.02%	2.0	2.0	0.02%
HLG422	1.7	1.7	0.02%	1.9	1.9	0.04%	1.9	1.9	0.04%
Overall	**1.7**	**1.7**	**0.02%**	**2.0**	**2.0**	**0.03%**	**2.0**	**2.0**	**0.03%**
Enc Time[%]	91%			102%			95%		
Dec Time[%]	89%			101%			103%		

SVT	All Intra			Low delay B			Random Access		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM-14.0	proposed		VTM-14.0	proposed		VTM-14.0	proposed	
SVT16	1.2	1.2	0.01%	1.2	1.2	0.02%	1.2	1.2	0.02%
SVT12	1.3	1.3	0.01%	1.3	1.3	0.02%	1.3	1.3	0.02%
Overall	**1.2**	**1.2**	**0.01%**	**1.3**	**1.3**	**0.02%**	**1.3**	**1.3**	**0.02%**
Enc Time[%]	99%			94%			99%		
Dec Time[%]	97%			83%			95%		

- Method 1 over VTM-14.0 with WPP off, normal QP

	Random Access									
	Over VTM-14.0									
			wPSNR			PSNR				
	DE100	PSNR-L100	Y	U	V	Y	U	V	EncT	DecT
Class H1	1.01%	1.07%	1.09%	1.19%	1.03%	1.09%	1.14%	1.17%	99%	99%
Class H2						1.24%	1.54%	0.76%	99%	98%
Overall	1.01%	1.07%	1.09%	1.19%	1.03%	1.16%	1.34%	0.96%	99%	98%

	All Intra									
	Over VTM-14.0									
				wPSNR			PSNR			
	DE100	PSNR-L100	Y	U	V	Y	U	V	EncT	DecT
Class H1	-0.19%	0.14%	#VALUE!	#VALUE!	#VALUE!	0.14%	-0.11%	-0.40%	99%	98%
Class H2						0.13%	-0.23%	-0.37%	106%	101%
Overall	-0.19%	0.14%	#VALUE!	#VALUE!	#VALUE!	0.14%	-0.17%	-0.39%	103%	100%

- Method 2 over VTM-14.0 with WPP off, low QP, lossy

HDR PQ	AI							
	Over VTM12.0_cebase							
	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV	EncT	DecT
PQ444	#VALUE!	#VALUE!	#VALUE!	0.03%	0.04%	0.02%	102%	107%
PQ422	#VALUE!	#VALUE!	#VALUE!	0.04%	0.04%	0.02%	106%	107%
Overall	#VALUE!	#VALUE!	#VALUE!	0.04%	0.04%	0.02%	104%	107%
	LDB							
	Over VTM12.0_cebase							
	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV	EncT	DecT
PQ444	#VALUE!	#VALUE!	#VALUE!	0.06%	0.06%	0.03%	92%	98%
PQ422	#VALUE!	#VALUE!	#VALUE!	0.12%	0.10%	0.04%	90%	91%
Overall	#VALUE!	#VALUE!	#VALUE!	0.09%	0.08%	0.03%	91%	95%
	RA							
	Over VTM12.0_cebase							
	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV	EncT	DecT
PQ444	0.08%	0.11%	0.01%	0.07%	0.11%	0.02%	97%	95%
PQ422	0.12%	0.06%	0.02%	0.11%	0.06%	0.01%	94%	92%
Overall	0.10%	0.09%	0.01%	0.09%	0.08%	0.01%	96%	94%
Overall PQ	#VALUE!	#VALUE!	#VALUE!	0.07%	0.07%	0.02%	97%	98%

HDR HLG	AI				
	Over VTM12.0_cebase				
	psnrY	psnrU	psnrV	EncT	DecT
HLG444	0.02%	0.01%	0.01%	106%	109%
HLG422	0.02%	0.01%	0.01%	102%	112%
Overall	0.02%	0.01%	0.01%	104%	110%
	LDB				
	Over VTM12.0_cebase				
	psnrY	psnrU	psnrV	EncT	DecT
HLG444	0.08%	0.04%	0.00%	93%	81%
HLG422	0.09%	0.04%	0.01%	92%	84%
Overall	0.08%	0.04%	0.00%	92%	82%
	RA				
	Over VTM12.0_cebase				
	psnrY	psnrU	psnrV	EncT	DecT
HLG444	0.09%	0.06%	0.01%	92%	84%
HLG422	0.10%	0.06%	0.02%	94%	86%
Overall	0.10%	0.06%	0.02%	93%	85%
Overall HLG	0.07%	0.04%	0.01%	97%	92%

SVT RGB	AI				
	Over VTM12.0_cebase				
	psnrG	psnrB	psnrR	EncT	DecT
SVT16	0.01%	0.02%	0.01%	101%	95%
SVT12	0.01%	0.01%	0.01%	105%	98%
Overall	0.01%	0.01%	0.01%	103%	97%
	LDB				
	Over VTM12.0_cebase				
	psnrG	psnrB	psnrR	EncT	DecT
SVT16	0.03%	0.02%	0.02%	98%	93%
SVT12	0.02%	0.01%	0.01%	97%	96%
Overall	0.02%	0.02%	0.01%	98%	95%
	RA				
	Over VTM12.0_cebase				
	psnrG	psnrB	psnrR	EncT	DecT
SVT16	0.03%	0.02%	0.02%	94%	96%
SVT12	0.02%	0.01%	0.01%	92%	98%
Overall	0.03%	0.02%	0.01%	93%	97%
Overall RGB	0.02%	0.02%	0.01%	98%	96%

- Method 2 over VTM-14.0 with WPP off, lossless

PQ	All Intra			Low delay B			Random Access		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM-14.0	proposed		VTM-14.0	proposed		VTM-14.0	proposed	
PQ444	2.6	2.6	0.01%	3.1	3.1	0.01%	3.1	3.1	0.01%
PQ422	2.4	2.4	0.01%	2.9	2.9	0.01%	2.9	2.9	0.01%
Overall	**2.5**	**2.5**	**0.01%**	**3.0**	**3.0**	**0.01%**	**3.0**	**3.0**	**0.01%**
Enc Time[%]	104%			100%			99%		
Dec Time[%]	100%			94%			98%		

HLG	All Intra			Low delay B			Random Access		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM-14.0	proposed		VTM-14.0	proposed		VTM-14.0	proposed	
HLG444	1.8	1.8	0.02%	2.0	2.0	0.02%	2.0	2.0	0.02%
HLG422	1.7	1.7	0.02%	1.9	1.9	0.04%	1.9	1.9	0.04%
Overall	**1.7**	**1.7**	**0.02%**	**2.0**	**2.0**	**0.03%**	**2.0**	**2.0**	**0.03%**
Enc Time[%]	104%			99%			97%		
Dec Time[%]	104%			89%			93%		

SVT	All Intra			Low delay B			Random Access		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM-14.0	proposed		VTM-14.0	proposed		VTM-14.0	proposed	
SVT16	1.2	1.2	0.01%	1.2	1.2	0.02%	1.2	1.2	0.02%
SVT12	1.3	1.3	0.02%	1.3	1.3	0.03%	1.3	1.3	0.03%
Overall	**1.2**	**1.2**	**0.01%**	**1.3**	**1.3**	**0.02%**	**1.3**	**1.3**	**0.02%**
Enc Time[%]	105%			99%			100%		
Dec Time[%]	111%			91%			104%		

- Method 2 over VTM-14.0 with WPP off, normal QP

	Random Access									
	Over VTM-14.0									
			wPSNR			PSNR				
	DE100	PSNR-L100	Y	U	V	Y	U	V	EncT	DecT
Class H1	1.01%	1.07%	1.09%	1.19%	1.03%	1.09%	1.14%	1.17%	101%	108%
Class H2						1.24%	1.54%	0.76%	101%	106%
Overall	1.01%	1.07%	1.09%	1.19%	1.03%	1.16%	1.34%	0.96%	101%	107%

	All Intra									
	Over VTM-14.0									
			wPSNR			PSNR				
	DE100	PSNR-L100	Y	U	V	Y	U	V	EncT	DecT
Class H1	-0.19%	0.14%	#VALUE!	#VALUE!	#VALUE!	0.14%	-0.11%	-0.40%	100%	100%
Class H2						0.13%	-0.23%	-0.37%	103%	104%
Overall	-0.19%	0.14%	#VALUE!	#VALUE!	#VALUE!	0.14%	-0.17%	-0.39%	102%	102%

Comparison btw method 1 and method 2

- The same coding performance;
- Method 2 does not require extra memory to store StatCoeff, hardware implementation friendly;
- Method 2 involve less specification change.

Method 1 Specification Change

9.3.2.x1 Storage process for Rice parameter derivation

```
for(cIdx = 0; cIdx < 3; cIdx++){
  StatCoeffWpp[cIdx] = StatCoeff[cIdx]
}
```

9.3.2.x2 Synchronization process for Rice parameter derivation

```
for(cIdx = 0; cIdx < 3; cIdx++){
  StatCoeff[cIdx] = StatCoeffWpp[cIdx]
}
```

Method 2 Specification Change

9.3.2.x2 Synchronization process for Rice parameter derivation

```
for(cIdx = 0; cIdx < 3; cIdx++){
  StatCoeff[cIdx] = 2 * Floor( Log2( BitDepth − 10 )
}
```

- Recommendation: method 2

Conclusion

- Method 1 proposes that StatCoeff is saved after the last TU of the first CTU in each CTU row is coded. Prior to the first TU coding, the synchronization process is applied to the first CTU for each CTU row where the StatCoeff is synchronized with saved StatCoeff from the preceding CTU row.

- Method 2 propose that a fixed initial value for the StatCoeff is used prior to the coding of first TU in each CTU row.

- Suggest adopting method 2 because of the same coding performance while method 2 does not require extra memory and less specification changes.

Thank Qualcomm for crosschecking!

Thank you