

**MEDIATEK**

# CE7-related: Unified Rice parameter derivation of `abs_remainder` and `dec_abs_level` syntax elements

Mohammed Golam Sarwer, Olena Chubach,  
Chih-Wei Hsu, Yu-Wen Huang, Shaw-Min Lei

Presented by Olena Chubach  
14th Meeting: Geneva, CH,  
19–27 March 2019

# Overall Summary

1. A template-based unified Rice parameter derivation is proposed for coding **dec\_abs\_level** and **abs\_remainder**
2. Additionally it is proposed to jointly code **abs\_remainder** and it's associated sign using a new syntax **abs\_remainder\_with\_sign**
3. Coding of **coeff\_sign\_flag** is moved to the second pass

|                 |    | Y      | U      | V      | Enc. T | Dec. T |
|-----------------|----|--------|--------|--------|--------|--------|
| Proposed Method | AI | -0.06% | -0.02% | -0.04% | 102%   | 101%   |
|                 | RA | -0.05% | -0.10% | -0.12% | 101%   | 100%   |
|                 | LB | -0.04% | 0.04%  | 0.24%  | 101%   | 100%   |

# A three-pass residual coding in VTM4.0

Pass 1: Context coding **sig\_coeff\_flag**, **abs\_level\_gt1\_flag**, **par\_level\_flag**, and **abs\_level\_gt3\_flag**

Pass 2: Bypass coding of **abs\_remainder** and **dec\_abs\_level**

Pass 3: Coding **coeff\_sign\_flag** of nonzero coefficients

- Rice parameter adaptation in VTM-4.0
  - for **abs\_remainder** (HEVC-like)
    - if( ricePar < 3 && rem > (3<<ricePar)-1 )
      - {
      - ricePar++;
      - }
  - for **dec\_abs\_level** (template-based)
    - g\_auiGoRiceParsCoeff[32] = { 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3 };
      - sumAll = clip2 (Sum of the neighbors of local template, 31);
      - rice = g\_auiGoRiceParsCoeff [sumAll];

# Proposed method

## 1. Template-based for both **abs\_remainder\_with\_sign** and **dec\_abs\_level**:

`uMinRicePar = transform_skip_flag ? 1 : 0;`

`g_auiGoRiceParsCoeff[32] = { 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3 };`

`sumAll = clip2 (Sum of the neighbors of local template, 31);`

`rice = g_auiGoRiceParsCoeff [sumAll];`

– coding **abs\_remainder\_with\_sign**

- if (rice < uMinRicePar ) rice = uMinRicePar

# Proposed method

## 2. Joint coding of **abs\_remainder** and it's associated sign using new syntax element **abs\_remainder\_with\_sign**

### Encoder Side:

absCoeff = absolute value of the Coeff

sign = Coeff < 0 ? 0: 1

**abs\_remainder\_with\_sign = (((absCoeff - 4) >> 1) << 1) + sign**

encode **abs\_remainder\_with\_sign**

### Decoder Side:

decode **abs\_remainder\_with\_sign**

**sign = abs\_remainder\_with\_sign & 1**

**absCoeff = (abs\_remainder\_with\_sign >> 1) << 1 + 4 + par\_level\_flag**

# Proposed method

- The pseudocode of **abs\_remainder\_with\_sign** (decoder side)

```

    uMinRicePar = transform_skip_flag ? 1 : 0;
    for(scanPos = firstSigPos; scanPos > firstPosMode2; scanPos-- )
    {
        TCoeff& tcoeff = coeff[ cctx.blockPos( scanPos ) ];
        sign = 0;
        if( tcoeff >= 4)
        {
            sumAll = cctx.templateAbsSum(scanPos, coeff);
            rice = g_auiGoRiceParsCoeff [sumAll];
            rice = (rice < uMinRicePar) ? uMinRicePar : rice;
            rem = decode abs_remainder_with_sign;
            sign = (rem & 1); // extract sign
            tcoeff += (rem >> 1) << 1; // extract absolute value
        }

        else if( tcoeff)
            sign = decode coeff_sign_flag;
        tcoeff = (sign == 0) ? -tcoeff : tcoeff;
    }

```

# Proposed method

- The pseudocode of **dec\_abs\_level** (decoder side)

```

for(scanPos = firstPosMode2; scanPos >= minSubPos; scanPos-- )
{
    TCoeff& tcoeff = coeff[ cctx.blockPos( scanPos ) ];
    sign = 0;
    sumAll = cctx.templateAbsSum(scanPos, coeff);
    rice = g_auiGoRiceParsCoeff [sumAll];
    pos0 = g_auiGoRicePosCoeff0[std::max(0, state - 1)][sumAll];
    rem = decode dec_abs_level;
    rem = ( rem == pos0 ? 0 : rem < pos0 ? rem+1 : rem );

    if( rem)
        sign = decode coeff_sign_flag;
    tcoeff = (sign == 0) ? -tcoeff : tcoeff;
}

```

# Simulation results (CTC test conditions)

|                 |    | Y      | U      | V      | Enc. T | Dec. T |
|-----------------|----|--------|--------|--------|--------|--------|
| Proposed Method | AI | -0.06% | -0.02% | -0.04% | 102%   | 101%   |
|                 | RA | -0.05% | -0.10% | -0.12% | 101%   | 100%   |
|                 | LB | -0.04% | 0.04%  | 0.24%  | 101%   | 100%   |

|                | All Intra Main10 |        |        |      |      |
|----------------|------------------|--------|--------|------|------|
|                | Over VTM-4.0     |        |        |      |      |
|                | Y                | U      | V      | EncT | DecT |
| Class A1       | -0.06%           | 0.07%  | -0.06% | 102% | 96%  |
| Class A2       | -0.05%           | 0.08%  | -0.02% | 103% | 98%  |
| Class B        | -0.07%           | -0.08% | 0.02%  | 102% | 101% |
| Class C        | -0.04%           | -0.09% | -0.04% | 102% | 102% |
| Class E        | -0.09%           | -0.02% | -0.13% | 102% | 103% |
| <b>Overall</b> | -0.06%           | -0.02% | -0.04% | 102% | 100% |
| Class D        | -0.07%           | -0.01% | -0.10% | 101% | 102% |
| Class F        | 0.02%            | -0.05% | 0.03%  | 102% | 101% |

|                | Random access Main10 |        |        |      |      |
|----------------|----------------------|--------|--------|------|------|
|                | Over VTM-4.0         |        |        |      |      |
|                | Y                    | U      | V      | EncT | DecT |
| Class A1       | -0.06%               | -0.18% | -0.06% | 101% | 101% |
| Class A2       | -0.04%               | -0.04% | 0.09%  | 101% | 100% |
| Class B        | -0.05%               | -0.03% | -0.22% | 101% | 100% |
| Class C        | -0.05%               | -0.19% | -0.19% | 100% | 100% |
| Class E        |                      |        |        |      |      |
| <b>Overall</b> | -0.05%               | -0.10% | -0.12% | 101% | 100% |
| Class D        | -0.04%               | -0.20% | -0.31% | 101% | 100% |
| Class F        | 0.07%                | -0.20% | -0.08% | 101% | 100% |

|                | Low delay B Main10 |        |        |      |      |
|----------------|--------------------|--------|--------|------|------|
|                | Over VTM-4.0       |        |        |      |      |
|                | Y                  | U      | V      | EncT | DecT |
| Class A1       |                    |        |        |      |      |
| Class A2       |                    |        |        |      |      |
| Class B        | -0.02%             | -0.14% | -0.39% | 101% | 102% |
| Class C        | -0.05%             | 0.15%  | 0.00%  | 101% | 101% |
| Class E        | -0.05%             | 0.18%  | 1.60%  | 100% | 95%  |
| <b>Overall</b> | -0.04%             | 0.04%  | 0.24%  | 101% | 100% |
| Class D        | -0.05%             | 0.54%  | 0.29%  | 102% | 101% |
| Class F        | -0.08%             | -0.08% | 0.10%  | 101% | 101% |

Thank Qualcomm  
for cross-checking

Anchor: VTM4.0

# Simulation results (low QP conditions)

|                 |    | Y      | U      | V      | Enc. T | Dec. T |
|-----------------|----|--------|--------|--------|--------|--------|
| Proposed Method | AI | -0.07% | -0.07% | -0.05% | 104%   | 102%   |
|                 | RA | -0.03% | -0.04% | -0.04% | 102%   | 102%   |
|                 | LB | -0.02% | -0.02% | 0.00%  | 102%   | 101%   |

|                | All Intra Main10 |        |        |      |      |
|----------------|------------------|--------|--------|------|------|
|                | Over VTM-4.0     |        |        |      |      |
|                | Y                | U      | V      | EncT | DecT |
| Class A1       | -0.04%           | -0.02% | 0.00%  | 104% | 102% |
| Class A2       | -0.10%           | -0.08% | -0.07% | 104% | 104% |
| Class B        | -0.09%           | -0.07% | -0.07% | 104% | 102% |
| Class C        | -0.02%           | -0.10% | -0.08% | 105% | 101% |
| Class E        | -0.11%           | -0.04% | -0.03% | 105% | 101% |
| <b>Overall</b> | -0.07%           | -0.07% | -0.05% | 104% | 102% |
| Class D        | 0.03%            | -0.01% | 0.04%  | 103% | 101% |
| Class F        | -0.05%           | 0.01%  | -0.01% | 103% | 100% |

|                | Random access Main10 |        |        |      |      |
|----------------|----------------------|--------|--------|------|------|
|                | Over VTM-4.0         |        |        |      |      |
|                | Y                    | U      | V      | EncT | DecT |
| Class A1       | -0.03%               | -0.05% | -0.03% | 102% | 102% |
| Class A2       | -0.04%               | -0.02% | -0.02% | 102% | 103% |
| Class B        | -0.04%               | -0.03% | -0.03% | 102% | 103% |
| Class C        | -0.02%               | -0.05% | -0.05% | 103% | 101% |
| Class E        |                      |        |        |      |      |
| <b>Overall</b> | -0.03%               | -0.04% | -0.04% | 102% | 102% |
| Class D        | 0.00%                | 0.00%  | -0.14% | 102% | 102% |
| Class F        | -0.03%               | -0.08% | -0.01% | 102% | 103% |

|                | Low delay B Main10 |        |        |      |      |
|----------------|--------------------|--------|--------|------|------|
|                | Over VTM-4.0       |        |        |      |      |
|                | Y                  | U      | V      | EncT | DecT |
| Class A1       |                    |        |        |      |      |
| Class A2       |                    |        |        |      |      |
| Class B        | -0.04%             | -0.01% | -0.01% | 102% | 102% |
| Class C        | -0.01%             | 0.00%  | 0.01%  | 102% | 101% |
| Class E        | -0.01%             | -0.05% | 0.01%  | 102% | 101% |
| <b>Overall</b> | -0.02%             | -0.02% | 0.00%  | 102% | 101% |
| Class D        | -0.03%             | -0.06% | -0.02% | 102% | 101% |
| Class F        | 0.07%              | 0.13%  | 0.18%  | 101% | 100% |

Thank Qualcomm  
for cross-checking

Anchor: VTM4.0

# Conclusions

- It is proposed to apply a unified template-based Rice parameter derivation process for coding **abs\_remainder** and **dec\_abs\_level**
- **abs\_remainder** and it's associated sign are proposed to be jointly coded using a new syntax **abs\_remainder\_with\_sign**
- The number of coding passes is reduced from three to two

|                 |    | Y      | U      | V      | Enc. T | Dec. T |
|-----------------|----|--------|--------|--------|--------|--------|
| Proposed Method | AI | -0.06% | -0.02% | -0.04% | 102%   | 101%   |
|                 | RA | -0.05% | -0.10% | -0.12% | 101%   | 100%   |
|                 | LB | -0.04% | 0.04%  | 0.24%  | 101%   | 100%   |



*everyday genius*

# Obtaining abs\_remainder\_with\_sign

| Sign | absLevel | par_level_flag | abs_remainder in VTM4.0 | abs_remainder_with_sign | Reconstructed<br>abs level | Sign |
|------|----------|----------------|-------------------------|-------------------------|----------------------------|------|
| -1   | 4        | 0              | 0                       | 0                       | 4                          | 0    |
| 1    | 4        | 0              | 0                       | 1                       | 4                          | 1    |
| -1   | 5        | 1              | 0                       | 0                       | 5                          | 0    |
| 1    | 5        | 1              | 0                       | 1                       | 5                          | 1    |
| -1   | 6        | 0              | 1                       | 2                       | 6                          | 0    |
| 1    | 6        | 0              | 1                       | 3                       | 6                          | 1    |
| -1   | 7        | 1              | 1                       | 2                       | 7                          | 0    |
| 1    | 7        | 1              | 1                       | 3                       | 7                          | 1    |
| -1   | 8        | 0              | 2                       | 4                       | 8                          | 0    |
| 1    | 8        | 0              | 2                       | 5                       | 8                          | 1    |
| -1   | 9        | 1              | 2                       | 4                       | 9                          | 0    |
| 1    | 9        | 1              | 2                       | 5                       | 9                          | 1    |
| -1   | 10       | 0              | 3                       | 6                       | 10                         | 0    |
| 1    | 10       | 0              | 3                       | 7                       | 10                         | 1    |

# When Sign data hiding is enabled

Some of the coefficients do not have sign value even those which are non-zero.

In the encoder side:

$\text{absCoeff} = \text{absolute value of the Coeff}$

$\text{validSign} = 0$  for the coefficient which does not have associated sign due to sign hiding

$\text{sign} = \text{Coeff} < 0 \mid \mid \text{!validSign} ? 0 : 1$

$\text{abs\_remainder\_with\_sign} = ((\text{absCoeff} - 4) \gg 1) \ll \text{validSign} + \text{sign}$

encode  $\text{abs\_remainder\_with\_sign}$

In the decoder side:

$\text{sign} = \text{validSign} ? (\text{abs\_remainder\_with\_sign} \& 1) : 0$

$\text{absCoeff} = (\text{abs\_remainder\_with\_sign} \gg \text{validSign}) \ll 1 + 4 + \text{par\_level\_flag}$