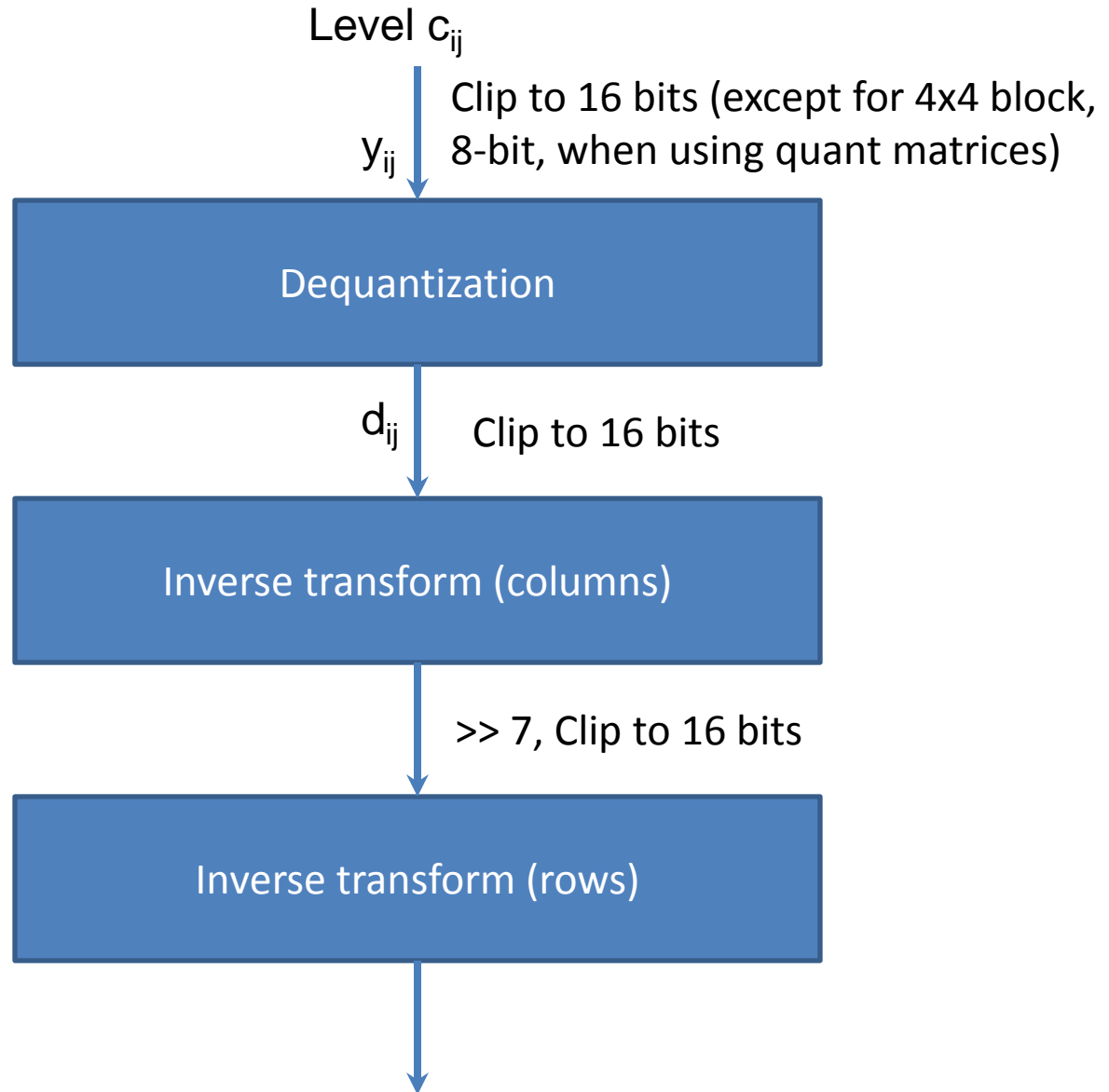


Elimination of clipping before dequantization

R. Joshi, J. Sole, X. Li, and M. Karczewicz

Qualcomm

Dequantization and inverse transform



Notation

- B: internal bit depth
 - InternalBitDepth in the common config files
- N: transform size
- M: $\log_2(N)$
- levelScale[k]: { 40, 45, 51, 57, 64, 72 } with $k = 0, 1, \dots, 5$
- M[i][j]: 8-bit unsigned quantizer matrix entries
- c_{ij} : decoded coefficient level values
- d_{ij} : Dequantized level values

Dequantization (no quant matrices)

- No quantization matrices
 - $\text{Shift} = B + M - 9$
 - $y_{ij} = \text{Clip3}(-32768, 32767, c_{ij})$
 - $\text{coeffQ} = ((y_{ij} * \text{IQ}[\text{QP}\%6] \ll (\text{QP}/6)) + \text{offset}) \gg \text{shift},$
where $\text{offset} = 1 \ll (\text{shift}-1)$
- Intermediate bit calculations: 31 bits signed

Dequantization (quant matrices)

- $\text{shiftScale} = B + M - 9 + 4 - (QP/6)$
- if ($\text{shiftScale} > 0$)
 - $y_{ij} = \text{Clip3}(-32768, 32767, c_{ij})$
 - $d_{ij} = (y_{ij} * W[i][j] * IQ[QP\%6] + \text{offset}) \gg (\text{shiftScale}),$
where $\text{offset} = 1 \ll (\text{shiftScale} - 1)$
- Else
 - y_{ij} obtained by clipping c_{ij} to 16 or 15 bits (dependent on M)
 - $d_{ij} = (y_{ij} * M[i][j] * IQ[QP\%6]) \ll (-\text{shiftScale})$
- Intermediate calculations 32 bits.

Clipping before dequantization

- Introduced at the last meeting to avoid 32-bit overflow for intermediate calculations
 - H0312 and H0541
 - Only occurs for 4×4 transforms when using quant matrices
 - Meeting notes: “Adopt (text in H0312 [2]) as a placeholder”
- Three normative clipping operations in the dequantization and inverse transform stages
 - Before dequantization, before 1st inverse transform stage and before 2nd inverse transform stage.
- Propose to remove clipping before dequantization stage.

Method 1

- Change the interpretation of quant matrix entries
- Interpret entries as scale factors normalized by 32 instead of 16.
- The only change in the specification is the definition of shiftScale
 - $\text{shiftScale} = B + M - 9 + 5 - (QP/6)$

Method 1 – *cont.*

- Existing quantization matrices can be used by increasing the base QP by 6.
 - Possible up to base QP of 45.
- Otherwise existing quantization matrix entries can be doubled
 - Entries over 127 in the existing design would be clipped to 255.

Method 1 – *cont.*

- Unified processing for dequantization
 - If scaling_list_present_flag is equal to 0,
 - $\text{shift} = B + M - 9 - (QP/6)$.
 - Otherwise
 - $\text{shift} = B + M - 9 + 5 - (QP/6)$
 - If ($\text{shift} > 0$)
$$d_{ij} = \text{Clip3}(-32768, 32767, ((c_{ij} * M[i][j] * \text{levelScale}[QP\%6]) + (1 \ll (\text{shift} - 1))) \gg \text{shift})$$
 - Otherwise
$$d_{ij} = \text{Clip3}(-32768, 32767, (c_{ij} * M[i][j] * \text{levelScale}[QP\%6]) \ll (-\text{shift}))$$
- If scaling_list_present_flag is equal to 0, $M[i][j]=1$ for all i, j .

Method 2

- If quantization matrix is being used and
 - If shiftscale ≤ 0
 - $d_{ij} = (y_{ij} * M[i][j] * IQ[QP\%6]) \ll (\min(2, -\text{shiftScale}))$
- In that case the downshift at the end of first transform stage is adjusted as follows:
 - If shiftscale ≥ -2
 - downshift = 7
 - Otherwise
 - downshift = $(7 + \text{shiftScale} + 2)$
- No need for clip before dequantization
- Only affects (QP ≥ 48 and M=2) when quantization matrix is being used.

Method 2 – *cont.*

- Left shifts have (almost) no impact on transform operation as long as the downshift after the transform is adjusted appropriately.
- This is because the transform can be represented as a matrix multiplication (no right-shifts)
- The only difference comes in because of the clip to 16-bit before 1st transform stage.
 - Less left-shift would lead to less clipping, which is in fact more desirable.

Conclusions

- Proposed 2 different methods to remove clipping before dequantization.
 - Method 1: Interpreting quantization matrix entries as scale factors normalized by 32 instead of 16.
 - Method 2: Restricting the left shifts to be less than or equal to 2 when quantization matrices are being used.
- Recommend adoption of Method 2.
 - Minimal change.
 - No interaction with quantization matrices.