### Scaling process for transform coefficients

Inputs of this process are:

– a variable nW specifying the width of the current transform unit,

– a variable nH specifying the height of the current transform unit,

– a (nW)x(nH) array c of transform coefficients with elements cij,

– a variable cIdx specifying the chroma component of the current block,

– a variable qP specifying the quantization parameter.

Output of this process is scaled transform coefficients as a (nW)x(nH) array of d with elements dij.

The variable log2TrSize is derived as follows:

log2TrSize = ( log2( NW ) + log2( NH ) ) >> 1 (8‑158)

The variable shift is derived as follows:

– If cIdx is equal to 0,

shift = BitDepthY + log2TrSize – 9 (8‑150)

levelLimit = 1<<min(15,12 + BitDepthy + log2TrSize - QP/6) (8‑150)

– Otherwise,

shift = BitDepthC + log2TrSize – 9 (8‑150)

levelLimit = 1<<min(15,12 + BitDepthC + log2TrSize - QP/6) (8‑150)

The scaling array levelScale[·] is specified as levelScale[k] = { 40, 45, 51, 57, 64, 72 } with k=0..5.

The variable shiftScale is set equal to shift + 4 – QP/6.

The elements of array M[ i ][ j ] with i = 0..nW – 1, j = 0..nH – 1 are set equal to ScalingList[ SizeID ][ RefMatrixID ][ trafoType ][ i\*nW+j ], where SizeID and RefMatrixID are specified in **Error! Reference source not found.** and **Error! Reference source not found.**, respectively, and trafoType is derived by

trafoType = ( ( nW = = nH ) ? 0 : ( ( nW > nH ) ? 1 : 2 ) ) (8‑158)

The scaled transform coefficient dij with i = 0..nW − 1, j = 0..nH − 1 is derived as follows.

– If scaling\_list\_present\_flag is equal to 1, [Ed. (WJ): should be replaced by better conditioning]

yij = Clip3(-32768, 32767,cij)

dij = Clip3( -32768, 32767,( ( yij \* levelScale[ qP%6 ] << ( qP/6 ) ) + ( 1 << ( shift – 1 ) ) ) >> shift ) (8‑158)

– Otherwise, if shiftScale is greater than 0,

yij = Clip3(-32768, 32767,cij)

dij = Clip3( -32768, 32767, ( yij \* M[ i ][ j ]\*levelScale[ qP%6 ] +

( 1 << ( shiftScale – 1 ) ) ) >> shiftScale ) (8‑158)

– Otherwise,

yij = Clip3(-levelLimit, levelLimit - 1),cij)

dij = Clip3( -32768, 32767, ( yij \* M[ i ][ j ]\*levelScale[ qP%6 ] ) << ( -shiftScale ) ) (8‑158)

[Ed. (WJ): do we need to clip cij to 16b before computing dij? – related to G719 – maybe not]