|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  15th Meeting: Geneva, CH, 23 Oct. – 1 Nov. 2013 | Document: JCTVC-O0178\_r2 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Non-RCE3: Explicit signaling of intra RDPCM** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Jewon Kang, Rajan Joshi, Joel Sole, Marta Karczewicz  5775 Morehouse Drive San Diego, CA 92121 USA | Tel: Email: | 1-858-651-8457 [jewonk@qti.qualcomm.com](mailto:jewonk@qti.qualcomm.com) |
| *Source:* | Qualcomm Inc. | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This contribution proposes explicit signaling of intra RDPCM to align it with inter RDPCM. The existing inter RDPCM signaling scheme to signal the RDPCM mode ((DPCM\_OFF, DPCM\_VER, and DPCM\_HOR) is reused. Simulation results are provided that reportedly show that the proposed scheme achieves coding gains in all configurations for Class F, SC YUV, and SC RGB test sequences without incurring any loss for natural videos and RExt video sequences.

# Technical Description

In the current HEVC Range Extension (RExt) design [1], intra RDPCM [2] is applied in horizontal and vertical direction to horizontal and vertical intra prediction modes, respectively, for lossy and lossless coding. For the remaining intra prediction modes, RDPCM is not applied. On the other hand, in inter RDPCM [3] the DPCM mode (DPCM\_OFF, DPCM\_VER, and DPCM\_HOR) is explicitly signaled. We propose to signal the RDPCM mode explicitly for intra RDPCM to unify intra and inter RDPCM. The proposed method is used for lossy as well as lossless coding.

The existing inter RDPCM signaling scheme to signal the RDPCM mode ((DPCM\_OFF, DPCM\_VER, and DPCM\_HOR) is reused. The RDPCM mode is chosen based on the SAD of the residuals for each mode. No other R-D search is performed. The existing scheme for signaling inter RDPCM mode is reused. Separate contexts are used for the explicitly signaled intra RDPCM flags.

# Experimental results

The proposed method was tested for the test sequences and conditions specified in RCE3 [4], and the anchor is HM12.0+RExt-4.1 software. Table 1 shows the BD-rate results for lossy coding and Table 2 shows the BD-rate results for lossless coding.

Table 1: BD-rate results for the proposed method for lossy coding. Anchor is HM12.0+RExt-4.1.







Table 2: BD-rate results for the proposed method for lossless coding. Anchor is HM12+RExt4.1.

# Conclusion

In this contribution, explicit signaling of the intra RDPCM mode is proposed. The existing inter RDPCM scheme for signaling the RDPCM mode is reused. The RDPCM mode is chosen based on the SAD of the residuals for each mode. The proposed method brings BD-rate improvements for Class F, SC YUV, and SC RGB test sequences without incurring losses for any class of test sequences. The impact on run-time complexity is small.

# Working Draft text

|  |  |
| --- | --- |
| residual\_coding( x0, y0, log2TrafoSize, cIdx ) { | Descriptor |
| if( transform\_skip\_enabled\_flag && !cu\_transquant\_bypass\_flag &&   ( log2TrafoSize <= Log2MaxTransformSkipSize ) ) |  |
| **transform\_skip\_flag**[ x0 ][ y0 ][ cIdx ] | ae(v) |
| if( CuPredMode[ x0 ][ y0 ] = = MODE\_INTER &&   residual\_dpcm\_inter\_enabled\_flag && transform\_skip\_flag[ x0 ][ y0 ][ cIdx ] ) { |  |
| **inter\_rdpcm\_flag**[ x0 ][ y0 ] | ae(v) |
| if( inter\_rdpcm\_flag[ x0 ][ y0 ] ) |  |
| **inter\_rpdcm\_dir\_flag**[ x0 ][ y0 ] | ae(v) |
| } |  |
| if( CuPredMode[ x0 ][ y0 ] = = MODE\_INTRA &&   residual\_dpcm\_inter\_enabled\_flag && transform\_skip\_flag[ x0 ][ y0 ][ cIdx ] ) { |  |
| **intra\_rdpcm\_flag**[ x0 ][ y0 ] |  |
| if( intra\_rdpcm\_flag[ x0 ][ y0 ] ) |  |
| **intra\_rpdcm\_dir\_flag**[ x0 ][ y0 ] |  |
| } |  |
| **last\_sig\_coeff\_x\_prefix** | ae(v) |
| **last\_sig\_coeff\_y\_prefix** | ae(v) |
| … |  |
|  |  |
| … |  |
| firstSigScanPos = 16 |  |
| lastSigScanPos = −1 |  |
| numGreater1Flag = 0 |  |
| lastGreater1ScanPos = −1 |  |
| for( n = 15; n >= 0; n− − ) { |  |
| xC = ( xS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ] |  |
| yC = ( yS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ] |  |
| if( sig\_coeff\_flag[ xC ][ yC ] ) { |  |
| if( numGreater1Flag < 8 ) { |  |
| **coeff\_abs\_level\_greater1\_flag**[ n ] | ae(v) |
| numGreater1Flag++ |  |
| if( coeff\_abs\_level\_greater1\_flag[ n ] && lastGreater1ScanPos = = −1 ) |  |
| lastGreater1ScanPos = n |  |
| } |  |
| if( lastSigScanPos = = −1 ) |  |
| lastSigScanPos = n |  |
| firstSigScanPos = n |  |
| } |  |
| } |  |
| if( cu\_transquant\_bypass\_flag | |  | | ( CuPredMode[ xC ][ yC ] = = MODE\_INTRA &&   residual\_dpcm\_intra\_enabled\_flag && ~~transform\_skip\_flag[ x0 ][ y0 ][ cIdx ] &&~~   ~~( predModeIntra = = 10 | | predModeIntra = = 26 )~~ intra\_rdpcm\_flag[ x0 ][ y0 ][ cIdx ] )  | | ( CuPredMode[ xC ][ yC ] = = MODE\_INTER &&   residual\_dpcm\_inter\_enabled\_flag && inter\_rdpcm\_flag[ x0 ][ y0 ][ cIdx ] ) ) |  |
| signHidden = 0 |  |
| else |  |
| signHidden = ( lastSigScanPos − firstSigScanPos > 3 ) |  |
| if( lastGreater1ScanPos != −1 ) |  |
| **coeff\_abs\_level\_greater2\_flag**[ lastGreater1ScanPos ] | ae(v) |
| … |  |

**intra\_rdpcm\_flag**[ x0 ][ y0 ][ cIdx ] specifies whether the residual modification process for blocks using a transform bypass is applied to the associated transform block or not. The array indices x0, y0 specify the location ( x0, y0 ) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture. The array index cIdx specifies an indicator for the colour component; it is equal to 0 for luma, equal to 1 for Cb, and equal to 2 for Cr. intra\_rdpcm\_flag[ x0 ][ y0 ][ cIdx ] equal to 1 specifies that the residual modification process is applied to the current transform block. intra\_rdpcm\_flag[ x0 ][ y0 ][ cIdx ] equal to 0 specifies that no residual modification process is applied to the current transform block.

**intra\_rdpcm\_dir\_flag**[ x0 ][ y0 ][ cIdx ] specifies the direction to be used by the residual modification process for the associated transform block. The array indices x0, y0 specify the location ( x0, y0 ) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture. The array index cIdx specifies an indicator for the colour component; it is equal to 0 for luma, equal to 1 for Cb, and equal to 2 for Cr.

**8.4.4.1** **General decoding process for intra blocks**

Inputs to this process are:

– a sample location ( xTb0, yTb0 ) specifying the top-left sample of the current transform block relative to the top‑left sample of the current picture,

– a variable log2TrafoSize specifying the size of the current transform block,

– a variable trafoDepth specifying the hierarchy depth of the current block relative to the coding unit,

– a variable predModeIntra specifying the intra prediction mode,

– a variable predModeIntraBc specifying the intra block copying mode,

– a variable bvIntra specifying the intra block copying vector,

– a variable cIdx specifying the colour component of the current block.

Output of this process is a modified reconstructed picture before deblocking filtering.

The luma sample location ( xTbY, yTbY ) specifying the top-left sample of the current luma transform block relative to the top-left luma sample of the current picture is derived as follows:

( xTbY, yTbY ) = ( cIdx  = =  0 ) ? ( xTb0, yTb0 ) : ( xTb0 \* SubWidthC, yTb0 \* SubHeightC ) (8‑26)

The variable splitFlag is derived as follows:

– If cIdx is equal to 0, splitFlag is set equal to split\_transform\_flag[ xTbY ][ yTbY ][ trafoDepth ].

– Otherwise, if all of the following conditions are true, splitFlag is set equal to 1.

* cIdx is greater than 0
* split\_transform\_flag[ xTbY ][ yTbY ][ trafoDepth ] is equal to 1
* log2TrafoSize is greater than 2

– Otherwise, splitFlag is set equal to 0.

Depending on the value of splitFlag, the following applies:

– If splitFlag is equal to 1, the following ordered steps apply:

1. The variables xTb1 and yTb1 are derived as follows:
   * The variable xTb1 is set equal to xTb0 + ( 1  <<  ( log2TrafoSize − 1 ) ).
   * The variable yTb1 is set equal to yTb0 + ( 1  <<  ( log2TrafoSize − 1 ) ).
2. The general decoding process for intra blocks as specified in this subclause is invoked with the location ( xTb0, yTb0 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode predModeIntra, and the variable cIdx as inputs, and the output is a modified reconstructed picture before deblocking filtering.
3. The general decoding process for intra blocks as specified in this subclause is invoked with the location ( xTb1, yTb0 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode predModeIntra, and the variable cIdx as inputs, and the output is a modified reconstructed picture before deblocking filtering.
4. The general decoding process for intra blocks as specified in this subclause is invoked with the location ( xTb0, yTb1 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode predModeIntra, and the variable cIdx as inputs, and the output is a modified reconstructed picture before deblocking filtering.
5. The general decoding process for intra blocks as specified in this subclause is invoked with the location ( xTb1, yTb1 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode predModeIntra, and the variable cIdx as inputs, and the output is a modified reconstructed picture before deblocking filtering.

– Otherwise (splitFlag is equal to 0), for the variable blkIdx proceeding over the values 0..( cIdx > 0  &&  ChromaArrayType  = =  2 ? 1 : 0 ), the following ordered steps apply:

1. The variable nTbS is set equal to 1  <<  log2TrafoSize.
2. The variable yTbOffset is set equal to blkIdx \* nTbS.
3. The variable residualDpcm is derived as follows:
   * If residual\_dpcm\_intra\_enabled\_flag is equal to 1 and one or more of the following conditions are true, residualDpcm is set equal to 1.
     + cu\_transquant\_bypass\_flag is equal to 1
     + transform\_skip\_flag[ xTb0 ][ yTb0 + yTbOffset ][ cIdx ] is equal to 1
   * Otherwise, residualDpcm is set equal to 0.
4. Depending upon the value of predModeIntraBc, the following applies

– When predModeIntraBc is equal to 0, the general intra sample prediction process as specified in subclause 8.4.4.2.1 is invoked with the transform block location ( xTb0, yTb0 + yTbOffset ), the intra prediction mode predModeIntra, the transform block size nTbS, and the variable cIdx as inputs, and the output is an (nTbS)x(nTbS) array predSamples.

– Otherwise (predModeIntraBc is equal to 1), the intra block copying process as specified in subclause 8.4.4.2.7 is invoked with the transform block location ( xTb0, yTb0 + yTbOffset ), the transform block size nTbS, the variable bvIntra, and the variable cIdx as inputs, and the output is an (nTbS)x(nTbS) array predSamples.

1. The scaling and transformation process as specified in subclause 8.6.2 is invoked with the luma location ( xTbY, yTbY + yTbOffset \* SubHeightC ), the variable trafoDepth, the variable cIdx, and the transform size trafoSize set equal to nTbS as inputs, and the output is an (nTbS)x(nTbS) array resSamples.
2. When residualDpcm is equal to 1 and either predModeIntra is equal to 10, or predModeIntra is equal to 26, the directional residual modification process for blocks using a transform bypass as specified in subclause 8.6.5 is invoked with the variable mDir set equal to intra\_rdpcm\_dir\_flag[ xCb + xB0 ][ yCb + yB0 ][ 0 ], the variable nTbS, and the (nTbS)x(nTbS) array r set equal to the array transformBlock as inputs, and the output is a modified (nTbS)x(nTbS) array resSamples as inputs, and the output is a modified (nTbS)x(nTbS) array resSamples.
3. The picture reconstruction process prior to in-loop filtering for a colour component as specified in subclause 8.6.6 is invoked with the transform block location ( xTb0, yTb0 + yTbOffset ), the variables nCurrSw and nCurrSh both set equal to nTbS, the variable cIdx, the (nTbS)x(nTbS) array predSamples, and the (nTbS)x(nTbS) array resSamples as inputs.

# References

1. D. Flynn, J. Sole, T. Suzuki, “Range Extensions Draft 4,” JCTVC-N1005, Vienna, Austria, Aug. 2013.
2. Rajan Joshi, “RCE2 subtest C.2: Extension of residual DPCM to lossy coding,” JCTVC-N0052, Vienna, Austria, Aug. 2013.
3. Matteo Naccari and Marta Mark,” RCE2: Experimental results for Test C.1,” JCTVC-N0075, Vienna, Austria, Aug. 2013.
4. A. Saxena, D. Kwon, M. Naccari, C. Pang, “HEVC Range Extensions Core Experiment 3 (RCE3): Intra Prediction techniques,” JCTVC-N1123, Vienna, Austria, Aug. 2013.

# Patent rights declaration(s)

**Qualcomm Inc. may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**