#### 7.3.8.5 Coding unit syntax

|  |  |
| --- | --- |
| coding\_unit( x0, y0, log2CbSize ) { | Descriptor |
| …… |  |
| if( !pcm\_flag[ x0 ][ y0 ] ) { |  |
| if( CuPredMode[ x0 ][ y0 ] != MODE\_INTRA &&  !( PartMode = = PART\_2Nx2N && merge\_flag[ x0 ][ y0 ] ) ) |  |
| **rqt\_root\_cbf** | ae(v) |
| if( rqt\_root\_cbf ) { |  |
| ~~if( residual\_adaptive\_colour\_transform\_enabled\_flag &&  ( CuPredMode[ x0 ][ y0 ] = = MODE\_INTER | |  ( PartMode = = PART\_2Nx2N &&  intra\_chroma\_pred\_mode[ x0 ][ y0 ] = = 4 ) | |  ( intra\_chroma\_pred\_mode[ x0 ][ y0 ] = = 4 &&  intra\_chroma\_pred\_mode[ x0 + nCbS/2 ][ y0 ] = = 4 &&  intra\_chroma\_pred\_mode[ x0 ][ y0 + nCbS/2 ] = = 4 &&  intra\_chroma\_pred\_mode[ x0 + nCbS/2 ][ y0 + nCbS/2 ] = = 4 ) ) )~~ |  |
| **~~cu\_residual\_act\_flag~~** | ~~ae(v)~~ |
| MaxTrafoDepth = ( CuPredMode[ x0 ][ y0 ] = = MODE\_INTRA ?  ( max\_transform\_hierarchy\_depth\_intra + IntraSplitFlag ) :  max\_transform\_hierarchy\_depth\_inter ) |  |
| transform\_tree( x0, y0, x0, y0, log2CbSize, 0, 0 ) |  |
| } |  |
| } |  |
| …… |  |
| } |  |

#### 7.3.8.11 Transform unit syntax

|  |  |
| --- | --- |
| transform\_unit( x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx ) { | Descriptor |
| …… |  |
| if( cbfLuma | | cbfChroma ) { |  |
| xP = ( x0 >> MinCbLog2SizeY ) << MinCbLog2SizeY |  |
| yP = ( y0 >> MinCbLog2SizeY ) << MinCbLog2SizeY |  |
| nCbS = 1 << MinCbLog2SizeY |  |
| if( residual\_adaptive\_colour\_transform\_enabled\_flag &&  ( CuPredMode[ x0 ][ y0 ] = = MODE\_INTER | |  ( PartMode[ x0 ][ y0 ] = = PART\_2Nx2N &&  intra\_chroma\_pred\_mode[ x0 ][ y0 ] = = 4 ) | |  ( intra\_chroma\_pred\_mode[ xP ][ yP ] = = 4 &&  intra\_chroma\_pred\_mode[ xP + nCbS/2 ][ yP ] = = 4 &&  intra\_chroma\_pred\_mode[ xP ][ yP + nCbS/2 ] = = 4 &&  intra\_chroma\_pred\_mode[ xP + nCbS/2 ][ y0 + nCbS/2 ] = = 4 ) ) ) |  |
| **tu\_ycgco\_residual\_flag** | ae(v) |
| if( cu\_qp\_delta\_enabled\_flag && !IsCuQpDeltaCoded ) { |  |
| **cu\_qp\_delta\_abs** | ae(v) |
| if( cu\_qp\_delta\_abs ) |  |
| **cu\_qp\_delta\_sign\_flag** | ae(v) |
| } |  |
| …… |  |
| } |  |
| } |  |

#### 7.4.9.5 Coding unit semantics

**~~cu\_residual\_act\_flag~~** ~~equal to 1 specifies that adaptive colour transform is applied to the residual samples of the current coding unit. cu\_residual\_act\_flag equal to 0 specifies that adaptive colour transform is not applied to the residual samples of the current coding unit. When this flag is equal to 1 and the coding unit is intra coded, chroma modes of all prediction units within the current coding unit (i.e., intra\_chroma\_pred\_mode) shall be equal to 4.~~

~~When cu\_residual\_act\_flag is not present, it is inferred to be equal to 0.~~

~~When cu\_transquant\_bypass\_flag is equal to 1 and bit\_depth\_luma\_minus8 is not equal to bit\_depth\_chroma\_minus8, the value of cu\_residual\_act\_flag, when present, shall be equal to 0.~~

#### 7.4.9.11 Transform unit semantics

**tu\_residual\_act\_flag** equal to 1 specifies that adaptive colour transform is applied to the residual samples of the current transform unit. tu\_residual\_act\_flag equal to 0 specifies that adaptive colour transform is not applied to the residual samples of the current transform unit. When cu\_residual\_act\_flag is not present, it is inferred to be equal to 0.

When cu\_transquant\_bypass\_flag is equal to 1 and bit\_depth\_luma\_minus8 is not equal to bit\_depth\_chroma\_minus8, the value of tu\_residual\_act\_flag, when present, shall be equal to 0.

#### 8.4.1 General decoding process for coding units coded in intra prediction mode

……

– Otherwise (pcm\_flag[ xCb ][ yCb ] is equal to 0, palette\_mode\_flag[ xCb ][ yCb ] is equal to 0 ), if IntraSplitFlag is equal to 0, the following ordered steps apply:

……

1. The general decoding process for intra blocks as specified in clause 8.4.4.1 is invoked with the luma location ( xCb, yCb ), the variable log2TrafoSize set equal to log2CbSize, the variable trafoDepth set equal to 0, the variable predModeIntra set equal to IntraPredModeY[ xCb ][ yCb ] and the variable cIdx set equal to 0, and variable controlParaAct set to (~~cu\_residual\_act\_flag[ xCb ][ yCb ]~~ residual\_adaptive\_colour\_transform\_enabled\_flag ? 2 : 3 ) as inputs, and the output is a modified reconstructed picture before deblocking filtering.

– Otherwise (pcm\_flag[ xCb ][ yCb ] is equal to 0, palette\_mode\_flag[ xCb ][ yCb ] is equal to 0 and IntraSplitFlag is equal to 1), for the variable blkIdx proceeding over the values 0..3, the following ordered steps apply:

……

1. The general decoding process for intra blocks as specified in clause 8.4.4.1 is invoked with the luma location ( xPb, yPb ), the variable log2TrafoSize set equal to log2CbSize − 1, the variable trafoDepth set equal to 1, the variable predModeIntra set equal to IntraPredModeY[ xPb ][ yPb ] and the variable cIdx set equal to 0, and variable controlParaACT set to ( ~~cu\_residual\_act\_flag[ xCb ][ yCb ]~~ residual\_adaptive\_colour\_transform\_enabled\_flag ? 2 : 3 ) as inputs, and the output is a modified reconstructed picture before deblocking filtering.

When ChromaArrayType is not equal to 0, the following applies.

……

– Otherwise (pcm\_flag[ xCb ][ yCb ] is equal to 0, palette\_mode\_flag[ xCb ][ yCb ] is equal to 0 ), if IntraSplitFlag is equal to 0 or ChromaArrayType is not equal to 3, the following ordered steps apply:

1. The derivation process for the chroma intra prediction mode as specified in clause 8.4.3 is invoked with the luma location ( xCb, yCb ) as input, and the output is the variable IntraPredModeC.
2. The general decoding process for intra blocks as specified in clause 8.4.4.1 is invoked with the chroma location ( xCb / SubWidthC, yCb / SubHeightC ), the variable log2TrafoSize set equal to log2CbSizeC, the variable trafoDepth set equal to 0, the variable predModeIntra set equal to IntraPredModeC and the variable cIdx set equal to 1, and variable controlParaAct set to ( ~~cu\_residual\_act\_flag[ xCb ][ yCb ]~~ residual\_adaptive\_colour\_transform\_enabled\_flag ? 2 : 3 ) as inputs, and the output is a modified reconstructed picture before deblocking filtering.
3. The general decoding process for intra blocks as specified in clause 8.4.4.1 is invoked with the chroma location ( xCb / SubWidthC, yCb / SubHeightC ), the variable log2TrafoSize set equal to log2CbSizeC, the variable trafoDepth set equal to 0, the variable predModeIntra set equal to IntraPredModeC and the variable cIdx set equal to 2, and variable controlParaAct set to ( ~~cu\_residual\_act\_flag[ xCb ][ yCb ]~~ residual\_adaptive\_colour\_transform\_enabled\_flag ? 2 : 3 ) as inputs, and the output is a modified reconstructed picture before deblocking filtering.

– Otherwise (pcm\_flag[ xCb ][ yCb ] is equal to 0, palette\_mode\_flag[ xCb ][ yCb ] is equal to 0, IntraSplitFlag is equal to 1 and ChromaArrayType is equal to 3), for the variable blkIdx proceeding over the values 0..3, the following ordered steps apply:

1. The variable xPb is set equal to xCb + ( nCbS  >>  1 ) \* ( blkIdx % 2 ).
2. The variable yPb is set equal to yCb + ( nCbS  >>  1 ) \* ( blkIdx / 2 ).
3. The derivation process for the chroma intra prediction mode as specified in clause 8.4.3 is invoked with the luma location ( xPb, yPb ) as input, and the output is the variable IntraPredModeC.
4. The general decoding process for intra blocks as specified in clause 8.4.4.1 is invoked with the chroma location ( xPb, yPb ), the variable log2TrafoSize set equal to log2CbSizeC − 1, the variable trafoDepth set equal to 1, the variable predModeIntra set equal to IntraPredModeC and the variable cIdx set equal to 1, and variable controlParaAct set to ( ~~cu\_residual\_act\_flag[ xCb ][ yCb ]~~ residual\_adaptive\_colour\_transform\_enabled\_flag ? 2 : 3 ) as inputs, and the output is a modified reconstructed picture before deblocking filtering.
5. The general decoding process for intra blocks as specified in clause 8.4.4.1 is invoked with the chroma location ( xPb, yPb ), the variable log2TrafoSize set equal to log2CbSizeC − 1, the variable trafoDepth set equal to 1, the variable predModeIntra set equal to IntraPredModeC and the variable cIdx set equal to 2, and variable controlParaAct set to ( ~~cu\_residual\_act\_flag[ xCb ][ yCb ]~~ residual\_adaptive\_colour\_transform\_enabled\_flag ? 2 : 3 ) as inputs, and the output is a modified reconstructed picture before deblocking filtering.

### 8.4.4.1 General decoding process for intra blocks

……

– 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. When controlParaAct is equal to 1 and cIdx is equal to 2 and tu\_ycgco\_residual\_flag[ xTbY ][ yTbY ] is equal to 1, the residual modification process for residual blocks using adaptive colour transform as specified in clause 8.6.8 is invoked with the variable blkSize set equal to nTbS, the (nTbS)x(nTbS) array rY set equal to the corresponding luma residual sample array resSamples of the current transform block, the (nTbS)x(nTbS) array rCb set equal to the corresponding chroma residual sample array resSamples when cIdx is equal to 1 of the current transform block, and the (nTbS)x(nTbS) array rCr set equal to the array resSamples as inputs, and the output are modified versions of the three residual sample arrays.

......

#### 8.5.4.3 Decoding process for chroma residual blocks

……

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

……

1. When ~~cu\_residual\_act\_flag[ xCb ][ yCb ]~~tu\_ycgco\_residual\_flag[ xCb + xB0 ][ yCb + xB0 ] is equal to 1 and cIdx is equal to 2, the residual modification process for residual blocks using adaptive colour transform as specified in clause 8.6.8 is invoked with the variable blkSize set equal to nTbS, the (nTbS)x(nTbS) array rY set equal to the corresponding luma residual sample array resSamples of the current transform block, the (nTbS)x(nTbS) array rCb set equal to the corresponding chroma residual sample array resSamples when cIdx is equal to 1 of the current transform block, and the (nTbS)x(nTbS) array rCr set equal to the array resSamples as inputs, and the output are modified versions of the three residual array.