#### Transform tree syntax

|  |  |
| --- | --- |
| transform\_tree( x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx ) { | Descriptor |
| if( log2TrafoSize <= Log2MaxTrafoSize &&  log2TrafoSize > Log2MinTrafoSize &&  trafoDepth < MaxTrafoDepth && !( IntraSplitFlag && ( trafoDepth = = 0 ) ) ) |  |
| **split\_transform\_flag**[ x0 ][ y0 ][ trafoDepth ] | ae(v) |
| if(ChromaArrayType == 3 && trafoDepth==0 && split\_transform\_flag[x0][y0][trafoDepth] != 0 && log2TrafoSize>8) |  |
| **zero\_depth\_chroma\_tr\_flag** [x0][y0] |  |
| if( log2TrafoSize > 2 && ChromaArrayType > 0 && |  |
| ( zero\_depth\_chroma\_tr\_flag[x0][y0]==0 || zero\_depth\_chroma\_tr\_flag[x0][y0]==1 && trafoDepth==0)){ |  |
| if( trafoDepth = = 0 | | cbf\_cb[ xBase ][ yBase ][ trafoDepth − 1 ] ) { |  |
| **cbf\_cb**[ x0 ][ y0 ][ trafoDepth ] | ae(v) |
| if( ChromaArrayType = = 2 && ! split\_transform\_flag[ x0 ][ y0 ][ trafoDepth ] ) |  |
| **cbf\_cb**[ x0 ][ y0 + ( 1 << ( log2TrafoSize − 1 ) ) ][ trafoDepth ] |  |
| } |  |
| if( trafoDepth = = 0 | | cbf\_cr[ xBase ][ yBase ][ trafoDepth − 1 ] ) { |  |
| **cbf\_cr**[ x0 ][ y0 ][ trafoDepth ] | ae(v) |
| if( ChromaArrayType = = 2 && ! split\_transform\_flag[ x0 ][ y0 ][ trafoDepth ] ) |  |
| **cbf\_cr**[ x0 ][ y0 + ( 1 << ( log2TrafoSize − 1 ) ) ][ trafoDepth ] |  |
| } |  |
| } |  |
| if(zero\_depth\_chroma\_tr\_flag[x0][y0]==1 && trafoDepth==0) |  |
| { |  |
| for( tIdx = 0; tIdx < ( ChromaArrayType = = 2 ? 2 : 1 ); tIdx++) |  |
| if( cbf\_cb[ xBase ][ yBase ][ trafoDepth ] ) |  |
| residual\_coding( xBase, yBase + ( tIdx << log2TrafoSize ), log2TrafoSize, 1 ) |  |
| for( tIdx = 0; tIdx < ( ChromaArrayType = = 2 ? 2 : 1 ); tIdx++) |  |
| if( cbf\_cr[ xBase ][ yBase ][ trafoDepth ] ) |  |
| residual\_coding( xBase, yBase + ( tIdx << log2TrafoSize ), log2TrafoSize, 2 ) |  |
| } |  |
| if( split\_transform\_flag[ x0 ][ y0 ][ trafoDepth ] ) { |  |
| x1 = x0 + ( 1  <<  ( log2TrafoSize − 1 ) ) |  |
| y1 = y0 + ( 1  <<  ( log2TrafoSize − 1 ) ) |  |
| transform\_tree( x0, y0, x0, y0, log2TrafoSize − 1, trafoDepth + 1, 0 ) |  |
| transform\_tree( x1, y0, x0, y0, log2TrafoSize − 1, trafoDepth + 1, 1 ) |  |
| transform\_tree( x0, y1, x0, y0, log2TrafoSize − 1, trafoDepth + 1, 2 ) |  |
| transform\_tree( x1, y1, x0, y0, log2TrafoSize − 1, trafoDepth + 1, 3 ) |  |
| } else { |  |
| if( CuPredMode[ x0 ][ y0 ] = = MODE\_INTRA | | trafoDepth != 0 | |  cbf\_cb[ x0 ][ y0 ][ trafoDepth ] | | cbf\_cr[ x0 ][ y0 ][ trafoDepth ] | |   ( ChromaArrayType = = 2 &&   ( cbf\_cb[ x0 ][ y0 + ( 1 << ( log2TrafoSize − 1 ) ) ][ trafoDepth ] | |   cbf\_cr[ x0 ][ y0 + ( 1 << ( log2TrafoSize − 1 ) ) ][ trafoDepth ] ) ) ) |  |
| **cbf\_luma**[ x0 ][ y0 ][ trafoDepth ] | ae(v) |
|  |  |
| transform\_unit( x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx ) |  |
| } |  |
| } |  |

#### Transform unit syntax

|  |  |
| --- | --- |
| transform\_unit( x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx ) { | Descriptor |
| log2TrafoSizeC = log2TrafoSize − ( ChromaArrayType = = 3 ? 0 : 1 ) |  |
| if( cbf\_luma[ x0 ][ y0 ][ trafoDepth ] | |   cbf\_cb[ x0 ][ y0 ][ trafoDepth ] | |   cbf\_cr[ x0 ][ y0 ][ trafoDepth ] | |   ( ChromaArrayType = = 2 &&   ( cbf\_cb[ x0 ][ y0 + ( 1 << log2TrafoSizeC ) ][ trafoDepth ] | |   cbf\_cr[ x0 ][ y0 + ( 1 << log2TrafoSizeC ) ][ trafoDepth ] ) ) ) { |  |
| 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) |
| } |  |
| if( cbf\_luma[ x0 ][ y0 ][ trafoDepth ] ) |  |
| residual\_coding( x0, y0, log2TrafoSize, 0 ) |  |
| if( log2TrafoSize > 2 | | ChromaArrayType = = 3 ) { |  |
| for( tIdx = 0; tIdx < ( ChromaArrayType = = 2 ? 2 : 1 ); tIdx++) |  |
| if( cbf\_cb[ x0 ][ y0 + ( tIdx << log2TrafoSizeC ) ][ trafoDepth ] ) |  |
| residual\_coding( x0, y0 + ( tIdx << log2TrafoSizeC ), log2TrafoSizeC, 1 ) |  |
| for( tIdx = 0; tIdx < ( ChromaArrayType = = 2 ? 2 : 1 ); tIdx++) |  |
| if( cbf\_cr[ x0 ][ y0 + ( tIdx << log2TrafoSizeC ) ][ trafoDepth ] ) |  |
| residual\_coding( x0, y0 + ( tIdx << log2TrafoSizeC ), log2TrafoSizeC, 2 ) |  |
| } else if( blkIdx = = 3 && zero\_depth\_chroma\_tr\_flag[x0][y0]== 0 ) { |  |
| for( tIdx = 0; tIdx < ( ChromaArrayType = = 2 ? 2 : 1 ); tIdx++) |  |
| if( cbf\_cb[ xBase ][ yBase ][ trafoDepth ] ) |  |
| residual\_coding( xBase, yBase + ( tIdx << log2TrafoSize ), log2TrafoSize, 1 ) |  |
| for( tIdx = 0; tIdx < ( ChromaArrayType = = 2 ? 2 : 1 ); tIdx++) |  |
| if( cbf\_cr[ xBase ][ yBase ][ trafoDepth ] ) |  |
| residual\_coding( xBase, yBase + ( tIdx << log2TrafoSize ), log2TrafoSize, 2 ) |  |
| } |  |
| } |  |
| } |  |

**zero\_depth\_chroma\_tr\_flag** [ x0 ][ y0 ] specifies that a block is not split at all for the the purpose of chroma transform coding. The array indices x0, y0 specify the location ( x0, y0 ) of the top-left luma sample of the considered block relative to the top-left luma sample of the picture.

When zero\_depth\_chroma\_tr\_flag [ x0 ][ y0 ] is not present, it is inferred to be equal to 0