#### Coding unit syntax

|  |  |
| --- | --- |
| coding\_unit( x0, y0, log2CbSize ) { | Descriptor |
| **…** | ae(v) |
| if( rqt\_root\_cbf ) { |  |
| 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, 1 ) |  |
| transform\_tree( x0, y0, x0, y0, log2CbSize, 0, 0, 0 ) |  |
| } |  |
| } |  |
| } |  |
| } |  |

#### Transform tree syntax

|  |  |
| --- | --- |
| transform\_tree( x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx, cIdx ) { | Descriptor |
| if( log2TrafoSize <= Log2MaxTrafoSize &&  log2TrafoSize > Log2MinTrafoSize &&  trafoDepth < MaxTrafoDepth && !( IntraSplitFlag && ( trafoDepth = = 0 ) ) ) |  |
| if(cIdx==0 && chroma\_transform\_depth[x0][y0] <= luma\_transform\_depth[x0][y0]) **split\_transform\_flag**[ x0 ][ y0 ][ trafoDepth ] | ae(v) |
| else **split\_chroma\_transform\_flag**[ x0 ][ y0 ][ trafoDepth ] | ae(v) |
| if( log2TrafoSize > 2 && ChromaArrayType > 0 && cIdx) { |  |
| 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( 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, cIdx ) |  |
| transform\_tree( x1, y0, x0, y0, log2TrafoSize − 1, trafoDepth + 1, 1, cIdx ) |  |
| transform\_tree( x0, y1, x0, y0, log2TrafoSize − 1, trafoDepth + 1, 2, cIdx ) |  |
| transform\_tree( x1, y1, x0, y0, log2TrafoSize − 1, trafoDepth + 1, 3, cIdx ) |  |
| } else if(cIdx==0){ |  |
| 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 , cIdx) |  |
| } |  |
| } |  |

#### Transform unit syntax

|  |  |
| --- | --- |
| transform\_unit( x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx, cIdx  ) { | 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 ] && cIdx ==0 ) |  |
| 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 && && cIdx ==1) { |  |
| 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 ) |  |
| } |  |
| } |  |
| } |  |

**split\_transform\_flag**[ x0 ][ y0 ][ trafoDepth ] specifies whether a block is split into four blocks with half horizontal and half vertical size for the purpose of 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. The array index trafoDepth specifies the current subdivision level of a coding block into blocks for the purpose of transform coding. trafoDepth is equal to 0 for blocks that correspond to coding blocks.

The variable interSplitFlag is derived as follows:

* If max\_transform\_hierarchy\_depth\_inter is equal to 0 and CuPredMode[ x0 ][ y0 ] is equal to MODE\_INTER and PartMode is not equal to PART\_2Nx2N and trafoDepth is equal to 0, interSplitFlag is set equal to 1.
* Otherwise, interSplitFlag is set equal to 0.

The variable luma\_transform\_depth[x0][y0] is derived as follows:

* If trafoDepth is equal to 0, luma\_transform\_depth[x0][y0] is set as 0.
* If split\_transform\_flag[x0][y0][trafoDepth] is equal to 1, luma\_transform\_depth[x0][y0] is increased by 1.

When split\_transform\_flag[ x0 ][ y0 ][ trafoDepth ] is not present, it is inferred as follows:

* If one or more of the following conditions are true, the value of split\_transform\_flag[ x0 ][ y0 ][ trafoDepth ] is inferred to be equal to 1:
* log2TrafoSize is greater than Log2MaxTrafoSize
* IntraSplitFlag is equal to 1 and trafoDepth is equal to 0
* interSplitFlag is equal to 1
* Otherwise, if luma\_transform\_depth[x0][y0] is smaller than chroma\_transform\_depth[x0][y0], the value of split\_transform\_flag[ x0 ][ y0 ][ trafoDepth ] is inferred to be equal to 1.
* Otherwise, the value of split\_transform\_flag[ x0 ][ y0 ][ trafoDepth ] is inferred to be equal to 0.

**split\_chroma transform\_flag**[ x0 ][ y0 ][ trafoDepth ] specifies whether a chroma block is split into four blocks with half horizontal and half vertical size for the purpose of 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. The array index trafoDepth specifies the current subdivision level of a coding block into blocks for the purpose of transform coding. trafoDepth is equal to 0 for blocks that correspond to coding blocks.

The variable chroma\_transform\_depth[x0][y0] is derived as follows:

* If trafoDepth is equal to 0, chroma\_transform\_depth[x0][y0] is set as 0.
* If split\_chroma\_transform\_flag[x0][y0][trafoDepth] is equal to 1, chroma\_transform\_depth[x0][y0] is increased by 1.