**Draft Text Specification**

**7.3.8.5 Coding unit syntax**

|  |  |
| --- | --- |
| coding\_unit( x0, y0, log2CbSize ) { | Descriptor |
| … |  |
| } else if( intra\_bc\_flag[ x0 ][ y0 ] ) { |  |
| if( PartMode != PART\_NxN ) |  |
| **line\_bc\_flag**[ x0 ][ y0 ] | ae(v) |
| if( line\_bc\_flag[ x0 ][ y0 ] && PartMode = = PART\_2Nx2N) |  |
| **row\_splitting\_flag**[ x0 ][ y0 ] | ae(v) |
| ~~m~~bvd\_coding( x0, y0, ~~2~~nCbS) |  |
| if( PartMode = = PART\_2NxN ) |  |
| ~~m~~bvd\_coding( x0, y0 + ( nCbS / 2 ), ~~2~~nCbS) |  |
| else if( PartMode = = PART\_Nx2N ) |  |
| ~~m~~bvd\_coding( x0 + ( nCbS / 2 ), y0, ~~2~~nCbS) |  |
| else if( PartMode = = PART\_NxN ) { |  |
| ~~m~~bvd\_coding( x0 + ( nCbS / 2 ), y0, ~~2~~nCbS) |  |
| ~~m~~bvd\_coding( x0, y0 + ( nCbS / 2 ), ~~2~~nCbS) |  |
| ~~m~~bvd\_coding( x0 + ( nCbS / 2 ), y0 + ( nCbS / 2 ), ~~2~~nCbS) |  |
| } |  |
| } else { |  |
| … |  |

**line\_bc\_flag**[ x0 ][ y0 ] equal to 1 specifies that the current coding unit is coded in line-based intra copy mode. line\_bc\_flag[ x0 ][ y0 ] equal to 0 specifies that the current coding unit is coded in intra block copying mode. When not present, the value of line\_bc\_flag is inferred to be equal to 0. The array indices x0, y0 specify the location ( x0, y0 ) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture.

**row\_splitting\_flag**[ x0 ][ y0 ] equal to 1 specifies that the current prediction unit is row-wise splitting. row\_splitting\_flag[ x0 ][ y0 ] equal to 0 specifies that the current prediction unit is column-wise splitting. When not present, the value of row\_splitting\_flag is derived as follows:

* If PartMode is equal to PART\_2NxN, row\_splitting\_flag[ x0 ][ y0 ] is set equal to 0.
* Otherwise, if PartMode is equal to PART\_Nx2N, row\_splitting\_flag[ x0 ][ y0 ] is set equal to 1.
* Otherwise, row\_splitting\_flag[ x0 ][ y0 ] is set equal to 0.

The array indices x0, y0 specify the location ( x0, y0 ) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture.

**7.3.8.9 Motion vector difference syntax**

|  |  |
| --- | --- |
| bvd\_coding( x0, y0, nCbS ) { | **Descriptor** |
| for( i = 0; i < ( line\_bc\_flag[ x0 ][ y0 ] ? nCbS : 1 ); i++ ) |  |
| mvd\_coding( x0, y0, 2 ) |  |
| } |  |

#### **Motion vector difference semantics**

The variable BvdIntra[ x0 ][ y0 ][ compIdx ] specifies the difference between a vector component to be used for the intra block copying prediction mode and its prediction. The value of BvdIntra[ x0 ][ y0 ][ compIdx ] shall be in the range of −128 to 128, inclusive. The array indices x0, y0 specify the location ( x0, y0 ) of the top-left luma sample of the considered prediction block relative to the top-left luma sample of the picture. The horizontal block vector component is assigned compIdx = 0 and the vertical block vector component is assigned compIdx = 1. When the considered prediction block is coded in line-based intra copy mode, the array indices i, j specify the location (i, j) of the top-left luma sample of the considered block relative to the top-left luma sample of the prediction block. Otherwise, i and j are inferred to be equal to 0.

* If refList is equal to 0, MvdL0[ x0 ][ y0 ][ compIdx ] is set equal to lMvd[ compIdx ] for compIdx = 0..1.
* Otherwise (refList is equal to 1), MvdL1[ x0 ][ y0 ][ compIdx ] is set equal to lMvd[ compIdx ] for compIdx = 0..1.
* Otherwise (refList is equal to 2), BvdIntra[ x0 + i ][ y0 + j ][ compIdx ] is set equal to lMvd[ compIdx ] for compIdx = 0..1.