#### Coding unit syntax

|  |  |
| --- | --- |
| coding\_unit( x0, y0, log2CbSize ) { | Descriptor |
| … |  |
| } else if( intra\_bc\_flag[ x0 ][ y0 ] ) { |  |
| mvd\_coding( x0, y0, 2) |  |
| if( PartMode = = PART\_2NxN ) |  |
| mvd\_coding( x0, y0 + ( nCbS / 2 ), 2) |  |
| else if( PartMode = = PART\_Nx2N ) |  |
| mvd\_coding( x0 + ( nCbS / 2 ), y0, 2) |  |
| else if( PartMode = = PART\_NxN ) { |  |
| mvd\_coding( x0 + ( nCbS / 2 ), y0, 2) |  |
| mvd\_coding( x0, y0 + ( nCbS / 2 ), 2) |  |
| mvd\_coding( x0 + ( nCbS / 2 ), y0 + ( nCbS / 2 ), 2) |  |
| } |  |
| else if( bvIntra[ x0 ][ y0 ][ 0 ] == 0 || bvIntra[ x0 ][ y0 ][ 1 ] == 0) |  |
| **symmetric\_ibc\_flag**[x0][y0] | ae(v) |
| } else { |  |
| … |  |

#### Coding quadtree semantics

…

**symmeric\_ibc\_flag**[ x0 ][ y0 ] equal to 1 specifies that the current coding unit is coded in symmectic intra block copying mode. symmeric\_ibc\_flag [ x0 ][ y0 ] equal to 0 specifies that the current coding unit is not coded in symmectic intra block copying mode. When not present, the value of symmeric\_ibc\_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.

…

**8.4.5.2.7 Specification of intra block copying prediction mode**

…

The (nTbS)x(nTbS) array of predicted samples samples, with x, y = 0..nTbS − 1, is derived as follows:

– The reference sample location (xRefCmp, yRefCmp ) is specified by:

( xRefCmp, yRefCmp ) = ( xTbCmp + x + bv[ 0 ], yTbCmp + y + bv[ 1 ] ) (8‑65)

Each sample at the location ( xRefCmp, yRefCmp ) is assigned to predSamples[ x ][ y ] if symmeric\_ibc\_flag[xTbY][ yTbY] is equal to 0.

Otherwise (symmeric\_ibc\_flag[xTbY][ yTbY] is equal to 1), each sample at the location ( xRefCmp, yRefCmp ) is assigned to predSamples[ nTbS-1-x ][ y ] if bv[1] is equal to 0; Otherwise (bv[1] is not equal to 0), each sample at the location ( xRefCmp, yRefCmp ) is assigned to predSamples[ x ][ nTbS-1-y ] if bv[0] is equal to 0.

…

Table 9‑4 – Association of ctxIdx and syntax elements for each initializationType in the initialization process

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Syntax structure** | **Syntax element** | **ctxTable** | **initType** | | |
| **0** | **1** | **2** |
| coding\_unit( ) | … |  |  |  |  |
| intra\_bc\_flag[ ][ ] |  | 0 | 1 | 2 |
| symmetric\_ibc\_flag[ ][ ] | Table 9-34’ | 0…2 | 0…2 | 0…2 |
| pred\_mode\_flag |  |  | 0 | 1 |
| … |  |  |  |  |

Table 9‑34’ – Values of initValue for ctxIdx of symmetric\_ibc\_flag

|  |  |  |  |
| --- | --- | --- | --- |
| **Initialization variable** | **ctxIdx of sao\_merge\_left\_flag and sao\_merge\_up\_flag** | | |
| **0** | **1** | **2** |
| **initValue** | 146 | 154 | 157 |

**Table 9‑34** – Syntax elements and associated binarizations

| **Syntax structure** | **Syntax element** | **Binarization** | |
| --- | --- | --- | --- |
| **Process** | **Input parameters** |
| … | … | … | … |
| coding\_unit( ) | … | … | … |
| cu\_skip\_flag | FL | cMax = 1 |
| intra\_bc\_flag | FL | cMax = 1 |
| symmetric\_ibc\_flag | FL | cMax = 1 |
| pred\_mode\_flag | FL | cMax = 1 |

| Table 9‑39 – Assignment of ctxInc to syntax elements with context coded bins | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Syntax element** | **binIdx** | | | | | |
| **0** | **1** | **2** | **3** | **4** | **>= 5** |
| … |  |  |  |  |  |  |
| intra\_bc\_flag | 0 | na | na | na | na | na |
| symmetric\_ibc\_flag | 0 | na | na | na | na | na |
| pred\_mode\_flag | 0 | na | na | na | na | na |
| … |  |  |  |  |  |  |

Table 9‑40 – Specification of ctxInc using left and above syntax elements

|  |  |  |  |
| --- | --- | --- | --- |
| **Syntax element** | **condL** | **condA** | **ctxInc** |
| split\_cu\_flag[ x0 ][ y0 ] | CtDepth[ xNbL ][ yNbL ] > cqtDepth | CtDepth[ xNbA ][ yNbA ] > cqtDepth | ( condL  &&  availableL ) + ( condA  &&  availableA ) |
| cu\_skip\_flag[ x0 ][ y0 ] | cu\_skip\_flag[ xNbL ][ yNbL ] | cu\_skip\_flag[ xNbA ][ yNbA ] | ( condL  &&  availableL ) + ( condA  &&  availableA ) |
| symmetric\_ibc\_flag[ x0 ][ y0 ] | symmetric\_ibc\_flag[ xNbL ][ yNbL ] | symmetric\_ibc\_flag [ xNbA ][ yNbA ] | ( condL  &&  availableL ) + ( condA  &&  availableA ) |