### 8.3.1. Derivation process for luma intra prediction mode

Inputs to this process are:

– a luma location ( xB, yB ) specifying the top-left luma sample of the current block relative to the top‑left luma sample of the current picture,

– a variable log2PUSize specifying the size of the current prediction unit,

– variable arrays IntraPredMode (If available) that are previously (in decoding order) derived for adjacent coding units.

Output of this process is the variable IntraPredMode[ xB ][ yB ].

Table 8‑5 specifies the value for the intra prediction mode and the associated names.

Table 8‑1 – Specification of intra prediction mode and associated names

|  |  |
| --- | --- |
| **Intra prediction mode** | **Associated names** |
| 0 | Intra\_DC |
| 1-33 | Intra\_Angular (sorted based on direction) |
| 34 | Intra\_Planar |
| 35 | Intra\_FromLuma (used only for chroma) |

Table 8‑1 specifies the number of luma intra prediction modes intraPredModeNum depending on log2PUSize.

Table 8‑2 – Specification of intraPredModeNum

|  |  |
| --- | --- |
| **log2PUSize** | **intraPredModeNum** |
| 2 | 35 |
| 3 | 35 |
| 4 | 35 |
| 5 | 35 |
| 6 | 35 |

Table 8‑2 specifies the mapping table used for converting the number of intra prediction modes.

IntraPredMode[ xB ][ yB ] labelled 0, 1, 2, .., 34 represents directions of predictions as illustrated in Figure 8‑1.

Figure 8‑1 – Intra prediction mode directions (informative)

IntraPredMode[ xB ][ yB ] is derived as the following ordered steps. [Ed. (WJ): proponent suggests to move this part to the syntax since the other syntax elements utilize IntraPredMode. But it seems too complex to move all the following process to the syntax table. Maybe it’s better to move this part to the semantics section or simply avoid the use of IntraPredMode to parse the syntax item]

1. The derivation process for neighbouring treeblocks specified in subclause XXX with ( xB,  yB ) given as input and the output is assigned to tbAddrA and tbAddrB specifying the treeblock addresses of treeblocks covering ( xBA,  yBA ) and ( xBB, yBB ) respectively where ( xBA,  yBA ) is set equal to ( xB-1,  yB ) and ( xBB,  yBB ) is set equal to ( xB,  yB-1 ).
2. For N being either replaced A or B, the variables intraPredModeN are derived as follows.

* If the treeblock with address tbAddrN is not available, intraPredModeN is set equal to Intra\_Planar.
* Otherwise, if the coding unit covering ( xBN,  yBN ) is not coded as intra mode, intraPredModeN is set equal to Intra\_Planar,
* Otherwise, if yB-1 is smaller than YLCU, intraPredModeA is set equal to IntraPredMode[ xBA ][ yBA ] and intraPredModeB is set equal to Intra\_Planar.
* Otherwise, intraPredModeN is set equal to IntraPredMode[ xBN ][ yBN ], where IntraPredMode is the variable array assigned to the coding unit covering the luma location ( xBN, yBN ).

1. If candIntraPredModeA is equal to candIntraPredModeB, the candIntraPredModeA is modified as follows:

* If candIntraPredModeA is not equal to Intra\_Planar, candIntraPredModeA is set equal to Intra\_Planar
* Otherwise, candIntraPredModeA is set equal to Intra\_DC

1. The candModeList[x] is derived as follows:

candModeList[0] = Min( candIntraPredModeA, candIntraPredModeB ) (8‑13)  
candModeList[1] = Max( candIntraPredModeA, candIntraPredModeB ) (8‑13)

1. IntraPredMode[ xB ][ yB ] is derived by applying the following procedure:

* If prev\_intra\_pred\_flag[ xB ][ yB ] is true, the IntraPredMode[ xB ][ yB ] is set equal to candModeList[ mpm\_flag ][ xB ][ yB ]]
* Otherwise IntraPredMode[ xB ][ yB ] is derived by applying the following ordered steps:
  1. IntraPredMode[ xB ][ yB ] = rem\_intra\_luma\_pred\_mode
  2. When IntraPredMode[ xB ][ yB ] is equal or greater than candModeList[ 0 ], the value of IntraPredMode[ xB ][ yB ] is increased by one
  3. When IntraPredMode[ xB ][ yB ] is equal or greater than candModeList[ 1 ], the value of IntraPredMode[ xB ][ yB ] is increased by one

##### Specification of Intra\_Angular prediction mode

Inputs to this process are:

– neighbouring samples p[ x, y ], with x, y = -1..2\*nS-1,

– a variable nS specifying the prediction size.

Output of this process is:

– predicted samples predSamples[ x, y ], with x, y =0..nS-1.

This intra prediction mode is invoked when intraPredMode is not Planar, DC, Vertical, or Horizontal.



Figure 8‑2 illustrates the total 34 intra angles and Table 8‑7 specifies the mapping table between intraPredModer and the angle parameter intraPredAngle.



Figure 8‑2 – Intra prediction angle definition (informative)

Table 8‑7 – Specification of intraPredAngle

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **intraPredOrder** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| **intraPredAngle** | - | 32 | 26 | 21 | 17 | 13 | 9 | 5 | 2 | - | -2 | -5 | -9 | -13 | -17 | -21 | -26 |
| **intraPredOrder** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** | **32** | **33** |
| **intraPredAngle** | -32 | -26 | -21 | -17 | -13 | -9 | -5 | -2 | - | 2 | 5 | 9 | 13 | 17 | 21 | 26 | 32 |

Table 8‑8 further specifies the mapping table between intraPredMode and the inverse angle parameter invAngle.

Table 8‑8 – Specification of invAngle

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **intraPredOrder** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **invAngle** | -256 | -315 | -390 | -482 | -630 | -910 | -1638 | -4096 |
| **intraPredOrder** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** |
| **invAngle** | -315 | -390 | -482 | -630 | -910 | -1638 | -4096 | - |

The reference pixel array refMain[ x ], with x=-nS..2\*nS is specified as follows.

– If intraPredMode is less than 17,

refMain[ x ] = p[ -1+x, -1 ], with x=0..nS (8‑36)

* If intraPredAngle is less than 0,

refMain[ x ] = p[ -1, -1+( ( x\*invAngle+128 )>>8 ) ], with x=( nS\*intraPredAngle ) >>5..-1 (8‑37)

* Otherwise,

refMain[ x ] = p[ -1+x, -1 ], with x=nS+1..2\*nS (8‑38)

Otherwise,

refMain[ x ] = p[ -1, -1+x ], with x=0..nS (8‑39)

* If intraPredAngle is less than 0,

refMain[ x ] = p[ -1+( ( x\*invAngle+128 )>>8 ), -1 ], with x=( nS\*intraPredAngle ) >>5..-1 (8‑40)

* Otherwise,

refMain[ x ] = p[ -1, -1+x ], with x=nS+1..2\*nS (8‑41)

The values of the prediction samples predSamples[ x, y ], with x, y = 0..nS-1 are derived by the following procedures.

– The index variable iIdx and the multiplication factor iFact are derived by

iIdx = ( ( y + 1 )\*intraPredAngle ) >> 5 (8‑42)

iFact = ( ( y + 1 )\*intraPredAngle ) && 31 (8‑43)

– Depending on the value of iFact, the following applies.

* If iFact is not equal to 0, the value of the prediction samples predSamples[ x, y ] is derived by

predSamples[ x, y ] = ( ( 32 – iFact )\*refMain[ x+iIdx+1 ] + iFact\*refMain[ x+iIdx+2] + 16 ) >> 5 (8‑44)

* Otherwise, the value of the prediction samples predSamples[ x, y ] is derived by

predSamples[ x, y ] = refMain[ x+iIdx+1 ] (8‑45)

### Transformation process for scaled transform coefficients

Inputs of this process are:

– a variable nW specifying the width of the current transform unit,

– a variable nH specifying the height of the current transform unit,

– a (nW)x(nH) array d of scaled transform coefficients with elements dij.

– a variable cIdx specifying the chroma component of the current block,

Output of this process is residual samples as a (nW)x(nH) array r with elements rij.

Depending on PredMode and IntraPredMode, the following applies:

– If PredMode is equal to MODE\_INTRA, log2(nW\*nH) is equal to 4, and cIdx is equal to 0, the variables horizTrType and vertTrType are specified as Table 8‑11 with IntraPredMode as input. [Ed. (WJ): DST is applied only for luma 4x4 block]

– Otherwise, the variables horizTrType and vertTrType are set equal to 0.

Table 8‑13 – Specification of horizTrType and vertTrType

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IntraPredMode** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| **horizTrType** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **vertTrType** | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IntraPredMode** | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 |
| **horizTrType** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **vertTrType** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

The constructed residual samples are derived as specified in the following ordered steps.

1. Each (horizontal) row of scaled transform coefficients dij (i=0..nW−1, j=0..nH−1) is transformed to eij (i=0..nW−1, j=0..nH−1) by invoking the one-dimensional transformation process as specified in subclause 8.5.4.1 according to the width of the transform unit nW, with the (nW)x(nH) array d and the transform type variable horizTrType as the inputs and the output is the (nW)x(nH) array e.
2. The intermediate sample values gij (i=0..nW-1, j=0..nH-1) are derived by

gij = Clip3( -32768, 32767, ( eij + 64 ) >> 7 ) (8‑159)

1. Each (vertical) column of the resulting matrix gij (i=0..nW−1, j=0..nH−1) is transformed to fij (i=0..nW−1, j=0..nH−1) by invoking the one-dimensional transformation process as specified in subclause 8.5.4.1 according to the height of the transform unit nH, with the (nW)x(nH) array e and the transform type variable vertTrType as the inputs and the output is the (nW)x(nH) array f.
2. The variable shift is derived as follows:

– If cIdx is equal to 0,

shift = 20 – BitDepthY (8‑150)

– Otherwise,

shift = 20 – BitDepthC (8‑150)

1. The residual sample value rij with i=0..(nW)−1, j=0..(nH)−1 is derived as follows.

rij = ( fij + (1 << ( shift – 1) ) ) >> shift (8‑160)

##### 8.3.3.1.2 Filtering process of neighbouring samples

Inputs to this process are:

– neighbouring samples p[ x, y ], with x, y = -1..2\*nS-1,

– a variable nS specifying the prediction size.

Output of this process is:

– filtered samples pF[ x, y ],. with x, y = -1..2\*nS-1.

Table 8‑5 – Specification of intraDiagDistance [ nS ] for various prediction unit sizes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **nS = 4** | **nS = 8** | **nS = 16** | **nS = 32** | **nS = 64** |
| **Value** | 10 | 7 | 1 | 0 | 10 |

The value of intraFilterType[ nS ][ IntraPredMode ] is derived as follows:

– The minimum distance (absolute value of the differences between mode numbers) of the intraPredMode from Vertical and Horizontal modes is computed

– If the minimum distance for each prediction unit nS is larger than of intraDiagDistance[ nS ], intraFilterType for that intraPredMode is set to 1, otherwise it is set to zero. intraFilterType is always set to 1 for the case that intraPredMode is equal to Planar.

Filtered sample array pF[ x, y ] with x = -1..nS\*2-1 and y = -1..nS\*2-1  are derived as follows:

– When intraFilterType[ nS ][ IntraPredMode ] is equal to 1, the following applies:

pF[ -1, nS\*2-1 ] = p[ -1, nS\*2-1 ] (8‑17)

pF[ nS\*2-1, -1 ] = p[ nS\*2-1, -1 ] (8‑18)

pF[ -1, y ] = ( p[ -1, y+1 ] + 2\*p[ -1, y ] + p[ -1, y-1 ] + 2 ) >> 2 for y = nS\*2-2..0 (8‑19)

pF[ -1, -1] = ( p[ -1, 0 ] + 2\*p[ -1, -1] + p[ 0, -1 ] + 2) >> 2 (8‑20)

pF[ x, -1 ] = ( p[ x-1, -1 ] + 2\*p[ x, -1 ] + p[ x+1, -1 ] + 2 ) >> 2 for x = 0..nS\*2-2 (8‑21)