### 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 ‑ – 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 candModeList[x] is modified as follows:

* If any of candIntraPredModeA or candIntraPredModeB is Planar or DC:

candModeList[0] = Planar

candModeList[1] = DC

candModeList[2] = Vertical

* Othersie:

candModeList[0] = candIntraPredModeA

candModeList[1] = candIntraPredModeA-1

candModeList[2] = candIntraPredModeA+1

1. If candIntraPredModeA is NOT equal to candIntraPredModeB, the candModeList[x] is derived as follows:

candModeList[0] = candIntraPredModeA

candModeList[1] = candIntraPredModeB

The third candidate candModeList[2] is selected among the candidates Planar, DC, and vertical (in this order) given that they do not appear in the two previous MPMs.

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 ]]
* The binarization process for mpm\_flag is specified as following table.

|  |  |
| --- | --- |
| mpm\_flag | bin\_string |
| 0 | 1 |
| 1 | 00 |
| 2 | 01 |

* 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
  4. When IntraPredMode[ xB ][ yB ] is equal or greater than candModeList[ 2 ], the value of IntraPredMode[ xB ][ yB ] is increased by one

#### Binarization process for rem\_intra\_luma\_pred\_mode

Input to this process is a request for the syntax element rem\_intra\_luma\_pred\_mode and log2PUSize.

Output of this process is the binarization of the syntax element.

The variable cNumBins is set to 5.

rem\_intra\_luma\_pred\_mode is binarized using 5 bit FL binarization

### Prediction unit syntax

|  |  |
| --- | --- |
| prediction\_unit( x0, y0, log2CUSize ) { | Descriptor |
| if( skip\_flag[ x0 ][ y0 ] ) { |  |
| if( MaxNumMergeCand > 1 ) |  |
| **merge\_idx[** x0 **][** y0 **]** | ae(v) |
| } else if( PredMode = = MODE\_INTRA ) { |  |
| if( PartMode == PART\_2Nx2N &&  log2CUSize >= Log2MinIPCMCUSize ) |  |
| **pcm\_flag** | ae(v) |
| if( pcm\_flag ) { |  |
| while ( !byte\_aligned( ) ) |  |
| **pcm\_alignment\_zero\_bit** | u(v) |
| for( i = 0; i < 1 << ( log2CUSize << 1 ); i++ ) |  |
| **pcm\_sample\_luma**[ i ] | u(v) |
| for( i = 0; i < ( 1 << ( log2CUSize << 1 ) ) >> 1; i++ ) |  |
| **pcm\_sample\_chroma**[ i ] | u(v) |
| } else { |  |
| **prev\_intra\_luma\_pred\_flag[** x0 **][** y0 **]** | ae(v) |
| if( prev\_intra\_luma\_pred\_flag[ x0 ][ y0 ] ) |  |
| **mpm\_idx[** x0 **][** y0 **]** | ae(v) |
| else { |  |
| **intra\_mode\_not\_planar[** x0 **][**y0 **]** | ae(v) |
| if( intra\_mode\_not\_planar**[** x0 **][**y0 **] )** |  |
| **rem\_intra\_luma\_pred\_mode[** x0 **][**y0 **]** | ae(v) |
| } |  |
| **intra\_chroma\_pred\_mode**[ x0 ][ y0 ] | ae(v) |
| SignaledAsChromaDC =   ( chroma\_pred\_from\_luma\_enabled\_flag ?  intra\_chroma\_pred\_mode[ x0 ][ y0 ] == 3 :  intra\_chroma\_pred\_mode[ x0 ][ y0 ] == 2 ) |  |
| } |  |
| … |  |