**Working Draft Text Modifications for Method 1**

### 7.3.7 Prediction unit syntax

|  |  |
| --- | --- |
| prediction\_unit( x0, y0, log2CbSize ) { | Descriptor |
| if( skip\_flag[ x0 ][ y0 ] ) { |  |
| if( MaxNumMergeCand > 1 ) |  |
| **merge\_idx[** x0 **][** y0 **]** | ae(v) |
| } else if( PredMode = = MODE\_INTRA ) { |  |
| if( PartMode = = PART\_2Nx2N && pcm\_enabled\_flag &&  log2CbSize >= Log2MinIPCMCUSize &&  log2CbSize <= Log2MaxIPCMCUSize ) |  |
| **pcm\_flag** | ae(v) |
| if( pcm\_flag ) { |  |
| **num\_subsequent\_pcm** | tu(3) |
| NumPCMBlock = num\_subsequent\_pcm + 1 |  |
| while( !byte\_aligned( ) ) |  |
| **pcm\_alignment\_zero\_bit** | u(v) |
| pcm\_sample( x0, y0, log2CbSize ) |  |
| } else { |  |
| if(log2CbSize >=6) |  |
| **rem\_intra\_luma\_pred\_mode[** x0 **][**y0 **]** | ae(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 |  |
| **rem\_intra\_luma\_pred\_mode[** x0 **][**y0 **]** | ae(v) |
| } |  |
| **intra\_chroma\_pred\_mode**[ x0 ][ y0 ] | ae(v) |
| SignalledAsChromaDC =   ( chroma\_pred\_from\_luma\_enabled\_flag ?  intra\_chroma\_pred\_mode[ x0 ][ y0 ] = = 3 :  intra\_chroma\_pred\_mode[ x0 ][ y0 ] = = 2 ) |  |
| } |  |
| } else { /\* MODE\_INTER \*/ |  |
| **merge\_flag[** x0 **][** y0 **]** | ae(v) |
| ... |  |
| } |  |
| } |  |

### 8.4.1 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 8‑1 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\_Planar |
| 1 | Intra\_DC |
| Otherwise (2..34) | Intra\_Angular |
| 35 | Intra\_FromLuma (used only for chroma) |

IntraPredMode[ xB ][ yB ] labelled 0, 1, 2, .., 35 represents directions of predictions as illustrated in .



Figure ‑ – Intra prediction mode directions (informative)

When log2PUSize < 6, IntraPredMode[ xB ][ yB ] is derived as the following ordered steps.

1. The neighbouring locations ( xBA, yBA ) and ( xBB, yBB ) are set equal to ( xB−1, yB ) and ( xB, yB−1 ), respectively.
2. The minimum coding block addresses cbAddrA and cbAddrB of the coding treeblocks covering the locations ( xBA,  yBA ) and ( xBB, yBB ) respectively where are derived as follows.

cbAddrA =MinCbAddrZS[ xBA << Log2MinCbSize ][ yBA << Log2MinCbSize ] (8‑19)

cbAddrB =MinCbAddrZS[ xBB << Log2MinCbSize ][ yBB << Log2MinCbSize ] (8‑20)

1. The availability process for a minimum coding block address specified in subclause  is invoked once with the minmum coding block address cbAddrA as the input and the output assigned to availableA and once with the minmum coding block address cbAddrB as the input and the output assigned to availableB.
2. For N being either replaced A or B, the variables intraPredModeN are derived as follows.

* If availableN is equal to FALSE, intraPredModeN is set equal to Intra\_DC.
* Otherwise, if the coding unit covering ( xBN,  yBN ) is not coded as intra mode, intraPredModeN is set equal to Intra\_DC,
* Otherwise, if yB−1 is less than yCtb, intraPredModeA is set equal to IntraPredMode[ xBA ][ yBA ] and intraPredModeB is set equal to Intra\_DC.
* 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 ).

……

When log2PUSize >= 6, IntraPredMode[ xB ][ yB ] is derived as specified in Table 8-1.1

Table 8‑1.1 – Specification of intra luma prediction mode and binarization for log2PUSize>=6

|  |  |  |
| --- | --- | --- |
| **Intra prediction mode** | **Associated names** | **Code word** |
| 0 | Intra\_Planar | 00 |
| 1 | Intra\_DC | 01 |
| 10 | Intra\_Vertical | 10 |
| 26 | Intra\_Horizontal | 11 |

**Table 9-32 Syntax elements and associated types of binarization, maxBinIdxCtx, ctxIdxTable, and ctxIdxOffset**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ... | | | | | |
| rem\_intra\_luma\_pred\_mode | all | FL, cMax = 31 if logsPUSize<6 | na | na | na, (uses Decode Bypass) |
| FL, cMax = 2 if logsPUSize>=6 |

**Working Draft Text Modifications for Method 2**

### 7.3.7 Prediction unit syntax

|  |  |
| --- | --- |
| prediction\_unit( x0, y0, log2CbSize ) { | Descriptor |
| if( skip\_flag[ x0 ][ y0 ] ) { |  |
| if( MaxNumMergeCand > 1 ) |  |
| **merge\_idx[** x0 **][** y0 **]** | ae(v) |
| } else if( PredMode = = MODE\_INTRA ) { |  |
| if( PartMode = = PART\_2Nx2N && pcm\_enabled\_flag &&  log2CbSize >= Log2MinIPCMCUSize &&  log2CbSize <= Log2MaxIPCMCUSize ) |  |
| **pcm\_flag** | ae(v) |
| if( pcm\_flag ) { |  |
| **num\_subsequent\_pcm** | tu(3) |
| NumPCMBlock = num\_subsequent\_pcm + 1 |  |
| while( !byte\_aligned( ) ) |  |
| **pcm\_alignment\_zero\_bit** | u(v) |
| pcm\_sample( x0, y0, log2CbSize ) |  |
| } else { |  |
| if(log2CbSize <6){ |  |
| **prev\_intra\_luma\_pred\_flag[** x0 **][** y0 **]** | ae(v) |
| if( prev\_intra\_luma\_pred\_flag[ x0 ][ y0 ] ) |  |
| **mpm\_idx[** x0 **][** y0 **]** | ae(v) |
| else |  |
| **rem\_intra\_luma\_pred\_mode[** x0 **][**y0 **]** | ae(v) |
| } |  |
| if(log2CbSize <6 | | chroma\_pred\_from\_luma\_enabled\_flag) |  |
| **intra\_chroma\_pred\_mode**[ x0 ][ y0 ] | ae(v) |
| SignalledAsChromaDC =   ( chroma\_pred\_from\_luma\_enabled\_flag ?  intra\_chroma\_pred\_mode[ x0 ][ y0 ] = = 3 :  intra\_chroma\_pred\_mode[ x0 ][ y0 ] = = 2 ) |  |
| } |  |
| } else { /\* MODE\_INTER \*/ |  |
| **merge\_flag[** x0 **][** y0 **]** | ae(v) |
| ... |  |
| } |  |
| } |  |

### 8.4.1 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 ].

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\_Planar |
| 1 | Intra\_DC |
| Otherwise (2..34) | Intra\_Angular |
| 35 | Intra\_FromLuma (used only for chroma) |

IntraPredMode[ xB ][ yB ] labelled 0, 1, 2, .., 35 represents directions of predictions as illustrated in .



Figure ‑ – Intra prediction mode directions (informative)

When log2PUSize < 6, IntraPredMode[ xB ][ yB ] is derived as the following ordered steps.

1. The neighbouring locations ( xBA, yBA ) and ( xBB, yBB ) are set equal to ( xB−1, yB ) and ( xB, yB−1 ), respectively.
2. The minimum coding block addresses cbAddrA and cbAddrB of the coding treeblocks covering the locations ( xBA,  yBA ) and ( xBB, yBB ) respectively where are derived as follows.
   1. cbAddrA =MinCbAddrZS[ xBA << Log2MinCbSize ][ yBA << Log2MinCbSize ] (8‑19)
   2. cbAddrB =MinCbAddrZS[ xBB << Log2MinCbSize ][ yBB << Log2MinCbSize ] (8‑20)
3. The availability process for a minimum coding block address specified in subclause  is invoked once with the minmum coding block address cbAddrA as the input and the output assigned to availableA and once with the minmum coding block address cbAddrB as the input and the output assigned to availableB.
4. For N being either replaced A or B, the variables intraPredModeN are derived as follows.

* If availableN is equal to FALSE, intraPredModeN is set equal to Intra\_DC.
* Otherwise, if the coding unit covering ( xBN,  yBN ) is not coded as intra mode, intraPredModeN is set equal to Intra\_DC,
* Otherwise, if yB−1 is less than yCtb, intraPredModeA is set equal to IntraPredMode[ xBA ][ yBA ] and intraPredModeB is set equal to Intra\_DC.
* 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 ).

……

When log2PUSize >= 6, one prediction mode is used and inferred as Intra\_Planar.

### 8.4.2 Derivation process for chroma intra prediction mode

Input to this process is 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.

Output of this process is the variable IntraPredModeC.

When log2PUSize >= 6 and chroma\_pred\_from\_luma\_enabled\_flag is equal to 0, IntraPredModeC is set equal to IntraPredMode[ xB ][ yB ]; When log2PUSize >= 6 and chroma\_pred\_from\_luma\_enabled\_flag is equal to 1, IntraPredModeC is equal to IntraPredMode[ xB ][ yB ] if intra\_chroma\_pred\_mode is 0 or equal to LM if intra\_chroma\_pred\_mode is 1. When log2PUSize < 6, The chroma intra prediction mode IntraPredModeC is derived as specifed in or with intra\_chroma\_pred\_mode, IntraPredMode[ xB ][ yB ] and chroma\_pred\_from\_luma\_enabled\_flag as inputs.

Table 8‑2 – Specification of IntraPredModeC according to the values of intra\_chroma\_pred\_mode and IntraPredMode[ xB ][ yB ] when chroma\_pred\_from\_luma\_enabled\_flag is equal to 1

Table 8‑3 – Specification of IntraPredModeC according to the values of intra\_chroma\_pred\_mode and IntraPredMode[ xB ][ yB ] when chroma\_pred\_from\_luma\_enabled\_flag is equal to 0

#### 9.2.2.9 Binarization process for intra\_chroma\_pred\_mode

Input to this process is a request for a binarization for the syntax element intra\_chroma\_pred\_mode and the chroma\_pred\_from\_luma\_enabled\_flag.

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

Table 9‑36.1 – Specifcation of intra\_chroma\_pred\_mode depending on chroma\_pred\_from\_luma\_enabled\_flag if log2PUSize<6

|  |  |  |
| --- | --- | --- |
| **Value of intra\_chroma\_pred\_mode** | **chroma\_pred\_from\_luma\_enabled\_flag = 1** | **chroma\_pred\_from\_luma\_enabled\_flag = 0** |
| 5 | 0 | n/a |
| 4 | 10 | 0 |
| 0 | 1100 | 100 |
| 1 | 1101 | 101 |
| 2 | 1110 | 110 |
| 3 | 1111 | 111 |

Table 9‑36.2 – Specifcation of intra\_chroma\_pred\_mode depending on chroma\_pred\_from\_luma\_enabled\_flag if log2PUSize>=6

|  |  |  |
| --- | --- | --- |
| **Value of intra\_chroma\_pred\_mode** | **chroma\_pred\_from\_luma\_enabled\_flag = 1** | **chroma\_pred\_from\_luma\_enabled\_flag = 0** |
| 0 | 0 | n/a |
| 1 | 1 | n/a |