H.7.4.9.5.1 Depth mode parameter semantics

The variable Log2MaxDmmCbSize is set equal to 5.

**depth\_intra\_mode\_set\_indication\_flag** indicates the set of possible depth intra modes.

The variables depthIntraModeSet is derived as specified in the following:

* If log2CbSize is equal to 6, depthIntraModeSet is set equal to 0.
* Otherwise, if log2CbSize is equal to 3 and PartMode[ xC ][ yC ] is equal to PART\_NxN, depthIntraModeSet is set equal to 1.
* Otherwise ( (log2CbSize  =  = 3 && PartMode[ xC ][ yC ]  = =  PART\_2Nx2N) || (log2CbSize > 3 && log2CbSize < 6 ), the following applies:
  + If depth\_intra\_mode\_set\_indication\_flag is equal to 1, depthIntraModeSet is set equal to 2.
  + Otherwise ( depth\_intra\_mode\_set\_indication\_flag is equal to 0 ), depthIntraModeSet is set equal to 3.

**depth\_intra\_mode**[ x0 ][ y0 ] specifies the depth intra mode of the current prediction unit. specifies the value of the variable depthIntraModeMaxLen depending on depthIntraModeSet and the value of the variable DepthIntraMode and the associated name depending on the on depth\_intra\_mode and depthIntraModeSet.

It is a requirement of bitstream conformance, that when TexturePredMode[ x0 ][ y0 ] is not equal to MODE\_INTRA or TextureIntraPredModeY[ x0 ][ y0 ] not in the range of 2 to 34, inclusive, DepthIntraMode is not equal to INTRA\_DEP\_DMM\_WPREDTEX.

The variable SdcFlag[ x0 ][ y0 ] is derived as specified in the following:

* 1. SdcFlag[ x0 ][ y0 ] = ( DepthIntraMode[ x0 ][ y0 ]  = =  INTRA\_DEP\_SDC\_MPM1 )  | |   (H‑16)  
      ( DepthIntraMode [ x0 ][ y0 ]  = =  INTRA\_DEP\_SDC\_MPM2 )  | |    
      ( DepthIntraMode [ x0 ][ y0 ]  = =  INTRA\_DEP\_SDC\_MPM3 )

The variable DmmFlag[ x0 ][ y0 ] is derived as specified in the following:

* 1. DmmFlag[ x0 ][ y0 ] = ( DepthIntraMode[ x0 ][ y0 ]  = =  INTRA\_DEP\_DMM\_WFULL )  | |   (H‑23)  
      ( DepthIntraMode[ x0 ][ y0 ]  = =  INTRA\_DEP\_DMM\_CPREDTEX )  | |    
      ( DepthIntraMode[ x0 ][ y0 ]  = =  INTRA\_DEP\_DMM\_WPREDTEX )

Table ‑2 – Specification of DepthIntraMode and associated name depending on depthIntraModeSet and depth\_intra\_mode and specification of and depthIntraModeMaxLen depending on depthIntraModeSet

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **depthIntraModeSet** | 0 | 1 | 2 | 3 |
|  | **depthIntraModeMaxLen** | 1 | 3 | 3 | 2 |
| **DepthIntraMode** | **Associated name** | **depth\_intra\_mode** | | | |
| 0 | INTRA\_DEP\_SDC\_MPM1 | 0 | - | - | 0 |
| 1 | INTRA\_DEP\_NONE | 1 | 0 | 0 | - |
| 2 | INTRA\_DEP\_SDC\_MPM2 | 2 | - | - | 1 |
| 3 | INTRA\_DEP\_DMM\_WFULL | - | 1 | 1 | - |
| 4 | INTRA\_DEP\_DMM\_CPREDTEX | - | - | - | 3 |
| 5 | INTRA\_DEP\_DMM\_WPREDTEX | - | 2 | 2 | - |
| 6 | INTRA\_DEP\_CHAIN | - | 3 | 3 | - |
| 7 | INTRA\_DEP\_SDC\_MPM3 | 3 | - |  | 2 |

H.8.4 Decoding process for coding units coded in intra prediction mode

H.8.4.2 Derivation process for luma intra prediction mode

Input to this process is a luma location ( xPb, yPb ) specifying the top-left sample of the current luma prediction block relative to the top left luma sample of the current picture.

In this process, the luma intra prediction mode IntraPredModeY[ xPb ][ yPb ] is derived.

specifies the value for the intra prediction mode and the associated names.

Table ‑4 – Specification of intra prediction mode and associated names

|  |  |
| --- | --- |
| **Intra prediction mode** | **Associated name** |
| 0 | INTRA\_PLANAR |
| 1 | INTRA\_DC |
| 2..34 | INTRA\_ANGULAR2..INTRA\_ANGULAR34 |
| 35 | INTRA\_DMM\_WFULL |
| 36 | INTRA\_DMM\_WPREDTEX |
| 37 | INTRA\_DMM\_CPREDTEX |
| 38 | INTRA\_CHAIN |

IntraPredModeY[ xPb ][ yPb ] labelled 0..34 represents directions of predictions as illustrated in Figure 8 1.

* If DepthIntraMode[ xPb ][ yPb ] is equal to INTRA\_DEP\_DMM\_WFULL, IntraPredModeY[ xPb ][ yPb ] is set equal to INTRA\_DMM\_WFULL.
* Otherwise, if DepthIntraMode[ xPb ][ yPb ] is equal to INTRA\_DEP\_DMM\_WPREDTEX, IntraPredModeY[ xPb ][ yPb ] is set equal to INTRA\_DMM\_WPREDTEX.
* Otherwise if DepthIntraMode[ xPb ][ yPb ] is equal to INTRA\_DEP\_DMM\_CPREDTEX, IntraPredModeY[ xPb ][ yPb ] is set equal to INTRA\_DMM\_CPREDTEX.
* Otherwise if DepthIntraMode[ xPb ][ yPb ] is equal to INTRA\_DEP\_CHAIN, IntraPredModeY[ xPb ][ yPb ] is set equal to INTRA\_CHAIN.
* Otherwise ( DepthIntraMode[ xPb ][ yPb ] is equal to INTRA\_DEP\_NONE, INTRA\_DEP\_SDC\_MPM1, INTRA\_DEP\_SDC\_MPM2, or INTRA\_DEP\_SDC\_MPM3 ), IntraPredModeY[ xPb ][ yPb ] is derived as the following ordered steps:
  1. The neighbouring locations ( xNbA, yNbA ) and ( xNbB, yNbB ) are set equal to ( xPb − 1, yPb ) and ( xPb, yPb − 1 ), respectively.
  2. For X being replaced by either A or B, the variables candIntraPredModeX are derived as follows:
     + The availability derivation process for a block in z-scan order as specified in subclause 6.4.2 is invoked with the location ( xCurr, yCurr ) set equal to ( xPb, yPb ) and the neighbouring location ( xNbY, yNbY ) set equal to ( xNbX, yNbX ) as inputs, and the output is assigned to availableX.
     + The candidate intra prediction mode candIntraPredModeX is derived as follows:
       - If availableX is equal to FALSE, candIntraPredModeX is set equal to INTRA\_DC.
       - Otherwise, if CuPredMode[ xNbX ][ yNbX ] is not equal to MODE\_INTRA or pcm\_flag[ xNbX ][ yNbX ] is equal to 1, candIntraPredModeX is set equal to INTRA\_DC,
       - Otherwise, if X is equal to B and yPb − 1 is less than ( ( yPb  >>  CtbLog2SizeY )  <<  CtbLog2SizeY ), candIntraPredModeB is set equal to INTRA\_DC.
       - Otherwise, if candIntraPredModeX is larger than 34, candIntraPredModeX is set equal to INTRA\_PLANAR.
       - Otherwise, candIntraPredModeX is set equal to IntraPredModeY[ xNbX ][ yNbX ].
  3. The candModeList[ x ] with x = 0..2 is derived as follows:
     + If candIntraPredModeB is equal to candIntraPredModeA, the following applies:
       - If candIntraPredModeA is less than 2 (i.e. equal to INTRA\_PLANAR or INTRA\_DC), candModeList[ x ] with x = 0..2 is derived as follows:

candModeList[ 0 ] = INTRA\_PLANAR (‑27)

candModeList[ 1 ] = INTRA\_DC (‑28)

candModeList[ 2 ] = INTRA\_ANGULAR26 (‑29)

* + - * Otherwise, candModeList[ x ] with x = 0..2 is derived as follows:

candModeList[ 0 ] = candIntraPredModeA (‑30)

candModeList[ 1 ] = 2 + ( ( candIntraPredModeA + 29 ) % 32 ) (‑31)

candModeList[ 2 ] = 2 + ( ( candIntraPredModeA − 2 + 1 ) % 32 ) (‑32)

* + - Otherwise (candIntraPredModeB is not equal to candIntraPredModeA), the following applies:
      * candModeList[ 0 ] and candModeList[ 1 ] are derived as follows:

candModeList[ 0 ] = candIntraPredModeA (‑33)

candModeList[ 1 ] = candIntraPredModeB (‑34)

* + - * If neither of candModeList[ 0 ] and candModeList[ 1 ] is equal to INTRA\_PLANAR, candModeList[ 2 ] is set equal to INTRA\_PLANAR,
      * Otherwise, if neither of candModeList[ 0 ] and candModeList[ 1 ] is equal to INTRA\_DC, candModeList[ 2 ] is set equal to INTRA\_DC,
      * Otherwise, candModeList[ 2 ] is set equal to INTRA\_ANGULAR26.
  1. IntraPredModeY[ xPb ][ yPb ] is derived by applying the following procedure:
     + If DepthIntraMode[ xPb ][ yPb ] is equal to INTRA\_DEP\_SDC\_MPM1, IntraPredModeY[ xPb ][ yPb ] is set equal to candModeList[ 0 ].
     + Otherwise if DepthIntraMode[ xPb ][ yPb ] is equal to INTRA\_DEP\_SDC\_MPM2, IntraPredMode[ xB ][ yB ] is set equal to candModeList[ 1 ].
     + Otherwise if DepthIntraMode[ xPb ][ yPb ] is equal to INTRA\_DEP\_SDC\_MPM3 and log2CbSize is equal to 6, IntraPredMode[ xB ][ yB ] is set equal to candModeList[ 2 ].
     + Otherwise if DepthIntraMode[ xPb ][ yPb ] is equal to INTRA\_DEP\_SDC\_MPM3, IntraPredMode[ xB ][ yB ] is set equal to INTRA\_DMM\_WFULL.
     + Otherwise if DepthIntraMode[ xPb ][ yPb ] is equal to INTRA\_DEP\_NONE and prev\_intra\_luma\_pred\_flag[ xPb ][ yPb ] is equal to 1, the IntraPredModeY[ xPb ][ yPb ] is set equal to candModeList[ mpm\_idx ].
     + Otherwise if DepthIntraMode[ xPb ][ yPb ] is equal to INTRA\_DEP\_NONE, IntraPredModeY[ xPb ][ yPb ] is derived by applying the following ordered steps:

1. The array candModeList[ x ], x = 0..2 is modified as the following ordered steps:
   1. When candModeList[ 0 ] is greater than candModeList[ 1 ], both values are swapped as follows:

( candModeList[ 0 ], candModeList[ 1 ] ) = Swap( candModeList[ 0 ], candModeList[ 1 ] ) (‑35)

* 1. When candModeList[ 0 ] is greater than candModeList[ 2 ], both values are swapped as follows:

( candModeList[ 0 ], candModeList[ 2 ] ) = Swap( candModeList[ 0 ], candModeList[ 2 ] ) (‑36)

* 1. When candModeList[ 1 ] is greater than candModeList[ 2 ], both values are swapped as follows:

( candModeList[ 1 ], candModeList[ 2 ] ) = Swap( candModeList[ 1 ], candModeList[ 2 ] ) (‑37)

1. IntraPredModeY[ xPb ][ yPb ] is derived by the following ordered steps:
   1. IntraPredModeY[ xPb ][ yPb ] is set equal to rem\_intra\_luma\_pred\_mode[ xPb ][ yPb ].
   2. For i equal to 0 to 2, inclusive, when IntraPredModeY[ xPb ][ yPb ] is greater than or equal to candModeList[ i ], the value of IntraPredModeY[ xPb ][ yPb ] is incremented by one.

8.4.4.3 Depth value reconstruction process

Inputs to this process are:

* a luma location ( xTb, yTb ) specifying the top-left luma sample of the current block relative to the top-left luma sample of the current picture,
* a variable nTbS specifying the transform block size,
* predicted samples predSamples[ x ][ y ], with x, y =0..nTbS − 1
* the intra prediction mode predModeIntra,

Output of this process is:

* reconstructed depth value samples resSamples[ x ][ y ], with x, y = 0.. nTbS − 1.

Depending on predModeIntra the array wedgePattern[ x ][ y ] with x, y =0..nTbS − 1 specifying the binary segmentation pattern is derived as follows.

* If predModeIntra is equal to INTRA\_DMM\_WFULL, the following applies.
  + 1. wedgePattern = WedgePatternTable[ Log2( nTbS ) ][ wedge\_full\_tab\_idx[ xTb ][ yTb ] ]
* Otherwise ( predModeIntra is not equal to INTRA\_DMM\_WFULL ), the following applies.
  + For x, y = 0..nTbS − 1 wedgePattern[ x ][ y ] is set equal to 0.

Depending on dlt\_flag[ nuh\_layer\_id ] the reconstructed depth value samples resSamples[ x ][ y ] are derived as specified in the following:

* If dlt\_flag[ nuh\_layer\_id ] is equal to 0, the following applies:
  + For x, y = 0..nTbS − 1, the reconstructed depth value samples resSamples[ x ][ y ] are derived as specified in the following:
    - 1. resSamples[ x ][ y ] = predSamples[ x ][ y ] + SdcResidual[ xTb ][ yTb ][wedgePattern[ x ][ y ] ] (H‑59)
* Otherwise ( dlt\_flag[ nuh\_layer\_id ] is equal to 1 ), the following applies:
  + The variables dcPred[ 0 ] and dcPred[ 1 ] are derived as specified in the following:
    - if predModeIntra is equal to INTRA\_DMM\_WFULL , the following applies.
      * 1. dcPred[ wedgePattern[ 0 ][ 0 ] ] = predSamples[ 0 ][ 0 ] (H‑62)
        2. dcPred[ wedgePattern[ nTbS − 1 ][ 0 ] ] = predSamples[ nTbS − 1 ][ 0 ] (H‑63)
        3. dcPred[ wedgePattern[ 0 ][ nTbS − 1 ] ] = predSamples[ 0 ][ nTbS − 1 ] (H‑64)
        4. dcPred[ wedgePattern[ nTbS − 1 ][ nTbS − 1 ] ] = predSamples[ nTbS − 1 ][ nTbS − 1 ] (H‑65)
  + Otherwise, the following applies.
  + sumPred = 0  
     numPred = 0.   
     for( x = 0; x < ( nT >> 2 ); x++ ) {   
     xS  = x << 2    
     for ( y = 0; y < ( nT >> 2 ) ;y++ ) {   
     yS = y << 2   
     sumPred += predSamples[ xS ][ yS ]  
     numPred += 1  
     }  
     }  
     dcPred[ 0 ] = ( numPred > 0 ) ? ( sumPred / numPred ) : 0
  + For x, y = 0..nTbS − 1, the reconstructed depth value samples resSamples[ x ][ y ] are derived as specified in the following:
    - 1. dltIdxPred = DepthValue2Idx[ dcPred[ wedgePattern[ x ][ y ] ] ] (H‑66)
      2. dltIdxResi = SdcResidual[ xTb ][ yTb ][wedgePattern[ x ][ y ] ] (H‑67)
      3. resSamples[ x ][ y ] = predSamples[ x ][ y ] + Idx2DepthValue[ dltIdxPred + dltIdxResi ]  −   
          dcPred[ wedgePattern[ x ][ y ] ] (H‑68)



H.9.3.2.2 Initialization process for context variables

The specifications in subclause 9.3.2.2 apply with the following modifications.

* All references to the process specified in subclauses7.3.8.1 through 7.3.8.11 are replaced with references to the process specified in subclauses H.7.3.8.1 to H.7.3.8.11.
* Table H‑12 is appended to the end of Table 9-4.
* Table H‑13 to Table H‑24 are appended to the end of the subclause.

1. Table H‑ – Values of initValue for sdc\_residual\_flag ctxIdx

|  |  |  |  |
| --- | --- | --- | --- |
| **Initialization variable** | **ctxIdx of sdc\_residual\_flag** | | |
| **0** | **1** | **2** |
| **initValue** | 154 | 154 | 154 |

2. Table H‑ – Values of initValue for sdc\_residual\_abs\_minus1 ctxIdx

|  |  |  |  |
| --- | --- | --- | --- |
| **Initialization variable** | **ctxIdx of sdc\_residual\_abs\_minus1** | | |
| **0** | **1** | **2** |
| **initValue** | prefix: 154 suffix : na | prefix: 154 suffix : na | prefix: 154 suffix : na |