|  |  |
| --- | --- |
| depth\_mode\_parameters( x0 , y0 , log2CbSize ) { | **Descriptor** |
| if( ( log2CbSize = = 3 && PartMode[ xC ][ yC ] = = PART\_2Nx2N )  | | ( log2CbSize > 3 && log2CbSize < 6 ) ) |  |
| **depth\_intra\_mode\_set\_indication\_flag** | ae(v) |
| **depth\_intra\_mode**[ x0 ][ y0 ] | ae(v) |
| if ( DepthIntraMode[ x0 ][ y0 ] = = INTRA\_DEP\_DMM\_WFULL  | |    DepthIntraMode[ x0 ][ y0 ] = = INTRA\_DEP\_SDC\_DMM\_WFULL ) |  |
| **wedge\_full\_tab\_idx**[ x0  ][ y0 ] | ae(v) |
| else if( DepthIntraMode[ x0 ][ y0 ] = = INTRA\_DEP\_DMM\_WPREDTEX ) |  |
| **wedge\_predtex\_tab\_idx**[ x0 ][ y0 ] | ae(v) |
| } else if( DepthIntraMode[ x0 ][ y0 ] = = INTRA\_DEP\_CHAIN ) { |  |
| **edge\_start\_left\_flag**[ x0 ][ y0 ] | ae(v) |
| **edge\_start\_position**[ x0 ][ y0 ] | ae(v) |
| **num\_edge\_codes\_minus1**[ x0 ][ y0 ] | ae(v) |
| for( k = 0; k <= num\_edge\_codes\_minus1; k++ ) |  |
| **edge\_code**[ k ] | ae(v) |
| } |  |
| if( DmmFlag[ x0 ][ y0 ] ) { |  |
| **dmm\_dc\_flag**[ x0 ][ y0 ] | ae(v) |
| if ( dmm\_dc\_flag[ x0 ][ y0 ] ) |  |
| for( i = 0; i < 2; i ++ ) { |  |
| **dmm\_dc\_abs**[ x0 ][ y0 ][ i ] | ae(v) |
| if ( dmm\_dc\_abs[ x0 ][ y0 ][ i ]) |  |
| **dmm\_dc\_sign\_flag**[ x0 ][ y0 ][ i ] | ae(v) |
| } |  |
| } |  |
| else if( DepthIntraMode[ x0 ][ y0 ] = = INTRA\_DEP\_CHAIN ) { |  |
| **edge\_dc\_flag**[ x0 ][ y0 ] | ae(v) |
| if( edge\_dc\_flag[ x0 ][ y0 ] ) { |  |
| for( i = 0; i < 2; i++ ) { |  |
| **edge\_dc\_abs**[ x0 ][ y0 ][ i ] | ae(v) |
| if( edge\_dc\_abs[ x0 ][ y0 ][ i ] != 0 ) |  |
| **edge\_dc\_sign\_flag**[ x0 ][ y0 ][ i ] | ae(v) |
| } |  |
| } |  |
| } |  |
| ~~else if( SdcFlag[ x0 ][ y0 ] ) {~~ |  |
| ~~sdcNumSegments =   ( DepthIntraMode[ x0  ][ y0 ] = = INTRA\_DEP\_SDC\_DMM\_WFULL ) ? 2 : 1~~ |  |
| ~~for( i = 0; i < sdcNumSegments; i++ ) {~~ |  |
| **~~sdc\_residual\_flag~~**~~[ x0 ][ y0 ][ i ]~~ | ~~ae(v)~~ |
| ~~if( sdc\_residual\_flag[ x0 ][ y0 ][ i ])  {~~ |  |
| **~~sdc\_residual\_sign\_flag~~**~~[ x0 ][ y0 ][ i ]~~ | * + 1. ~~ae(v)~~ |
| **~~sdc\_residual\_abs\_minus1~~**~~[ x0 ][ y0 ][ i ]~~ | * + 1. ~~ae(v)~~ |
| ~~}~~ |  |
| ~~}~~ |  |
| ~~}~~ |  |
| } |  |

H.7.4.9.3 Depth mode parameter semantics

**~~sdc\_residual\_flag~~**~~[ x0 ][ y0 ][ i ] equal to 0 specifies that the residual is zero for segment i. sdc\_residual\_flag equal to 1 specifies that the residual is non-zero and the sdc\_residual\_sign\_flag and sdc\_residual\_abs[ i ] syntax elements are present for segment i.~~

**~~sdc\_residual\_abs\_minus1~~**~~[ x0 ][ y0 ][ i ]~~~~and~~ **~~sdc\_residual\_sign\_flag~~**~~[ x0 ][ y0 ][ i ]~~~~are used to derive SdcResidual[ x0 ][ y0 ][ i ] for segment i as follows:~~

* 1. SdcResidual[ x0 ][ y0 ][i] = 0  
     ~~( 1 − 2 \*~~~~sdc\_residual\_sign\_flag[ x0 ][ y0 ][ i ] ) \* ( sdc\_residual\_mag\_minus1[ x0 ][ y0 ][ i ] + 1) (‑21)~~

Table H‑12 – Association of ctxIdx and syntax elements for each initializationType in the initialization process

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Syntax element** | **ctxIdxTable** | **initType** | | |
| **0** | **1** | **2** |
| coding\_unit( ) depth\_mode\_parameters( ) | depth\_intra\_mode |  | 0..7 | 8..15 | 16..23 |
| wedge\_full\_tab\_idx |  | 0 | 1 | 2 |
| wedge\_predtex\_tab\_idx |  | 0 | 1 | 2 |
| dmm\_delta\_end\_flag dmm\_delta\_end\_abs\_minus1 |  | 0 | 1 | 2 |
| dmm\_dc\_flag |  | 0 | 1 | 2 |
| dmm\_dc\_abs |  | 0 | 1 | 2 |
| edge\_code |  | 0 | 1 | 2 |
| iv\_res\_pred\_weight\_idx |  |  | 0..3 | 4..7 |
| ic\_flag |  |  | 0 | 1 |
| edge\_dc\_flag |  | 0 | 1 | 2 |
| edge\_dc\_abs |  | 0 | 1 | 2 |
| ~~sdc\_residual\_flag~~ |  | ~~0~~ | ~~1~~ | ~~2~~ |
| ~~sdc\_residual\_abs\_minus1~~ | ~~prefix:  suffix: na~~ | ~~prefix:0 suffix: na~~ | ~~prefix:1 suffix: na~~ | ~~prefix:0 suffix: na~~ |
| inter\_sdc\_flag |  | 0 | 1 | 2 |
| inter\_sdc\_resi\_abs\_minus1 |  | 0 | 1 | 2 |
| inter\_sdc\_resi\_sign\_flag |  | 0 | 1 | 2 |

~~Table ‑20 – Values of variable initValue for sdc\_residual\_abs\_minus1 ctxIdx~~

|  |  |  |  |
| --- | --- | --- | --- |
| **~~Initialization variable~~** | **~~sdc\_residual\_abs\_minus1~~** | | |
| **~~0~~** | **~~1~~** | **~~2~~** |
| **~~initValue~~** | ~~prefix: 155 suffix : na~~ | ~~prefix: 155 suffix : na~~ | ~~prefix: 155 suffix : na~~ |

~~Table ‑28 – Syntax elements and associated types of binarization, maxBinIdxCtx, ctxIdxTable, and ctxIdxOffset~~

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **~~Syntax element~~** | **~~initType~~** | **~~Type of binarization~~** | **~~maxBinIdxCtx~~** | **~~ctxIdxTable~~** | **~~ctxIdxOffset~~** |
| ~~sdc\_residual\_abs\_minus1~~ | ~~0~~ | ~~TU, cMax=5~~  ~~( specified in subclause )~~ | ~~prefix: 0,  suffix: na~~ | ~~prefix:  suffix: na~~ | ~~prefix: 0 suffix: na, (Bypass)~~ |
| ~~1~~ | ~~prefix: 0,  suffix: na~~ | ~~prefix:  suffix: na~~ | ~~prefix: 1 suffix: na, (Bypass)~~ |
| ~~2~~ | ~~prefix: 0,  suffix: na~~ | ~~prefix  suffix: na~~ | ~~prefix: 2 suffix: na, (Bypass)~~ |

~~H.9.3.2.1.10 Binarization process for sdc\_residual\_abs\_minus1~~

1. ~~Input to this process is a request for the a syntax element sdc\_residual\_abs\_minus1,~~
2. ~~Output of this process is the binarization of the syntax element.~~
3. ~~The bin string is a concatenation of a prefix bin string and, when present, a suffix bin string.~~
4. ~~The variable numDepthValues is derived as follows:~~ 
   1. ~~numDepthValues = dlt\_flag[ nuh\_layer\_id ] ? (‑242)  
       num\_depth\_values\_in\_dlt[ nuh\_layer\_id ] : ( 1  <<  BitDepth~~~~Y~~~~) − 1~~
5. ~~The variable cMaxPrefix is derived as follows:~~ 
   1. ~~cMaxPrefix = ( numDepthValues \* 3 )  >>  2 )~~
6. ~~For the derivation of the prefix bin string, the following applies:~~

* ~~If sdc\_residual\_abs\_minus1 is less than cMaxPrefix, the prefix bin string is a bit string of length sdc\_residual\_abs\_minus1+ 1 indexed by binIdx. The bins for binIdx less than sdc\_residual\_abs\_minus1 are equal to 1. The bin with binIdx equal to sdc\_residual\_abs\_minus1 is equal to 0.~~
* ~~Otherwise, the prefix bin string is a bit string of length cMaxPrefix with all bins being equal to 1.~~

~~When sdc\_residual\_abs\_minus1 is greater than cMaxPrefix, the suffix of the bin string is present and it is derived as follows:~~

* + ~~The suffix value suffixVal, is derived as follows:~~
    1. ~~suffixVal = sdc\_residual\_abs\_minus1 − cMaxPrefix  (‑243)~~
  + ~~The suffix of the bin string is specified by Fixed-length (FL) binarization process as specified in subclause with suffixVal and cMax equal to ( numDepthValues − cMaxPrefix ) as the inputs.~~

Table H‑31 –Assignment of ctxIdxInc to syntax elements with context coded bins

| **Syntax element** | **binIdx** | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **>=5** |
| wedge\_full\_tab\_idx | 0 | 0 | 0 | 0 | 0 | 0 |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dmm\_dc\_flag | 0 | na | na | na | na | na |
| dmm\_dc\_abs | 0 | 0 | 0 | 0 | 0 | 0 |
| dmm\_dc\_sign\_flag | bypass | 0 | 0 | 0 | 0 | 0 |
| edge\_dc\_flag | 0 | na | na | na | na | na |
| edge\_dc\_abs | 0 | 0 | 0 | 0 | 0 | 0 |
| edge\_dc\_sign\_flag | bypass | 0 | 0 | 0 | 0 | 0 |
| edge\_code | 0 | 0 | 0 | 0 | 0 | 0 |
| res\_pred\_flag | 0 | na | na | na | na | na |
| ic\_flag | 0 | na | na | na | na | na |
| ~~sdc\_residual\_flag~~ | ~~bypass~~ | ~~na~~ | ~~na~~ | ~~na~~ | ~~na~~ | ~~na~~ |
| ~~sdc\_residual\_abs\_minus1~~ | ~~prefix: 0 suffix: bypass~~ | ~~prefix: 0 suffix: bypass~~ | ~~prefix: 0 suffix: bypass~~ | ~~prefix: 0 suffix: bypass~~ | ~~prefix: 0 suffix: bypass~~ | ~~prefix: 0 suffix: bypass~~ |
| inter\_sdc\_flag | 0 | 0 | 0 | 0 | 0 | 0 |
| inter\_sdc\_resi\_abs\_minus1 | 0 | 0 | 0 | 0 | 0 | 0 |
| inter\_sdc\_resi\_sign\_flag | 0 | 0 | 0 | 0 | 0 | 0 |