|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extension Development**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  5th Meeting: Vienna, AT, 27 July – 2 Aug. 2013 | Document: JCT3V- E0156 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **CE 6.h: Results on Simplified Inter Mode Coding of Depth** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Hongbin Liu ([hongbin.liu@lge.com](mailto:hongbin.liu@lge.com))  Jie Jia ([jie.jia@lge.com](mailto:jie.jia@lge.com))  Junghak Nam ([junghak.nam@lge.com](mailto:junghak.nam@lge.com)) |  |  |
| *Source:* | LG Electronics | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Spec Change:

H.7.3.9.1 General Coding unit syntax

|  |  |
| --- | --- |
| if ( icEnableFlag ) |  |
| **ic\_flag** | ae(v) |
| if( DepthFlag && PredMode[ x0 ][ y0 ] ! = MODE\_INTRA && !skip\_flag[ x0 ][ y0 ] ) |  |
| **inter\_sdc\_flag** | ae(v) |
| if ( inter\_sdc\_flag ) |  |
| { |  |
| puNum = ( PartMode = = PART\_2Nx2N ) ? 1 : ( PartMode = = PART\_NxN ? 4 : 2 ) |  |
| for( i = 0; i < puNum; i++ ) |  |
| { |  |
| **inter\_sdc\_resi\_abs\_minus1** [ x0 ][ y0 ][ i ] | ae(v) |
| **inter\_sdc\_resi\_sign\_flag**[ x0 ][ y0 ][ i ] | ae(v) |
| } |  |
| } |  |
| if( !pcm\_flag ) { |  |
| if( PredMode[ x0 ][ y0 ] != MODE\_INTRA &&   !(PartMode = = PART\_2Nx2N && merge\_flag[x0][y0]) ) |  |
| **no\_residual\_syntax\_flag** | ae(v) |
| if( !no\_residual\_syntax\_flag && !sdc\_flag && !inter\_sdc\_flag ) { |  |
| MaxTrafoDepth = ( PredMode[ x0 ][ y0 ] = = MODE\_INTRA ?   max\_transform\_hierarchy\_depth\_intra + IntraSplitFlag :   max\_transform\_hierarchy\_depth\_inter ) |  |
| transform\_tree( x0, y0 x0, y0, log2CbSize, 0, 0 ) |  |
| } |  |
| if( sdc\_flag ) { |  |
| sdcNumSegments = ( sdc\_pred\_mode = = 1 ) ? 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 ] | ae(v) |
| **sdc\_residual\_abs\_minus1**[ x0 ][ y0 ][ i ] | ae(v) |
| } |  |
| } |  |
| } |  |
| } |  |

H.7.4.9.1 General coding unit semantics

**inter\_sdc\_flag** equal to 1 specifies simplified depth coding of inter modes is used for the current coding unit. inter\_sdc\_flagequal to 0 specifies simplified depth coding of inter modes is not used for the current coding unit. When not present, inter\_sdc\_flag is inferred to be equal to 0.

**inter\_sdc\_resi\_abs\_minus1**[ x0 ][ y0 ][i], **inter\_sdc\_resi\_sign\_flag**[ x0 ][ y0 ][i]are used to derive InterSdcResi[ x0 ][ y0 ][ i ] as follows:

InterSdcResi[x0][y0][ i ] = ( 1 − 2 \* inter\_sdc\_resi\_sign\_flag [x0][y0][ i ] ) \*

( inter\_sdc\_resi\_abs\_minus1 [x0][y0][ i ] + 1 ) (‑10)

H.8.5.3 Decoding process for the residual signal of coding units coded in inter prediction mode

* if( !inter\_sdc\_flag )

The specifications in subclause 8.5.3 apply

* else
  + derive horizontal and vertical offset xOff and yOff as follows:
  + if ( PartMode = = PART\_2Nx2N )

xOff = nCSL, yOff = nCSL,

* + else if ( PartMode = = PART\_2NxN )

xOff = nCSL, yOff = nCSL >> 1,

* + else if ( PartMode = = PART\_2NxnU )

xOff = nCSL, yOff = nCSL >> 2,

* + else if ( PartMode = = PART\_2NxnD )

xOff = nCSL, yOff = ( nCSL >> 1 ) + ( nCSL >> 2 ),

* + else if ( PartMode = = PART\_Nx2N )

xOff = nCSL >> 1, yOff = nCSL,

* + else if ( PartMode = = PART\_nLx2N )

xOff = nCSL >> 2, yOff = nCSL,

* + else if ( PartMode = = PART\_nRx2N )

xOff = ( nCSL >> 1 ) + ( nCSL >> 2 ), yOff = nCSL,

* + else if ( PartMode = = PART\_NxN )

xOff = nCSL >> 1, yOff = nCSL >> 1,

* + derive residual array tmpInterSdcResi[4] as follows:
  + if ( PartMode = = PART\_2Nx2N )

for ( i = 0; i < 4; i++ )

tmpInterSdcResi[ i ] = InterSdcResi[x0][y0][ 0 ];

* + else if( PartMode = = PART\_2NxN || PartMode = = PART\_2NxnU || PartMode = = PART\_2NxnD )
    - * + set tmpInterSdcResi[ 0 ] and tmpInterSdcResi[ 1 ] to InterSdcResi[ x0 ][ y0 ][ 0 ]
        + set tmpInterSdcResi[ 2 ] and tmpInterSdcResi[ 3 ] to InterSdcResi[ x0 ][ y0 ][ 1 ]
  + else if( PartMode = = PART\_Nx2N || PartMode = = PART\_nLx2N || PartMode = = PART\_nRx2N )
    - * + set tmpInterSdcResi[ 0 ] and tmpInterSdcResi[ 2 ] to InterSdcResi[ x0 ][ y0 ][ 0 ]
        + set tmpInterSdcResi[ 1 ] and tmpInterSdcResi[ 3 ] to InterSdcResi[ x0 ][ y0 ][ 1 ]
  + else if ( PartMode = = PART\_NxN )

for ( i = 0; i < 4; i++ )

tmpInterSdcResi[ i ] = InterSdcResi[x0][y0][ i ];

* + derive residual array resSamplesL as follows:

for ( i = 0; i < yOff; i++ )

{

for ( j = 0; j < xOff; j++ )

resSamplesL[ i ][ j ] = tmpInterSdcResi[ 0 ];

for ( j = xOff; j < nCSL; j++ )

resSamplesL[ i ][ j ] = tmpInterSdcResi[ 1 ];

}

for ( i = yOff; i < nCSL; i++ )

{

for ( j = 0; j < xOff; j++ )

resSamplesL[ i ][ j ] = tmpInterSdcResi[ 2 ];

for ( j = xOff; j < nCSL; j++ )

resSamplesL[ i ][ j ] = tmpInterSdcResi[ 3 ];

}

the following addition at the end of the subclause:

For x in the range of 0 to nCSL − 1 and y in the range of 0 to nCSL − 1, the following applies:

* + ResSamplesL[ xC + x ][ yC + y ] is set equal to resSamplesL[ x ][ y ].

For x in the range of 0 to nCSC − 1 and y in the range of 0 to nCSC − 1, the following applies:

* + ResSamplesCb[ xC /2 + x ][ yC /2 + x] is set equal to resSamplesCb[ x ][ y ].
  + ResSamplesCr[ xC /2 + x ][ yC /2 + x ] is set equal to resSamplesCr[ x ][ y ].

H.9.3.1.1 Initialization process for context variables

Table H‑13 – Association of ctxIdx and syntax elements for each initializationType in the initialization process

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Syntax element** | **ctxIdxTable** | **initType** | | |
| **0** | **1** | **2** |
| coding\_unit() | dmm\_flag |  | 0 | 1 | 2 |
| dmm\_mode |  | 0 | 1 | 2 |
| wedge\_full\_tab\_idx |  | 0 | 1 | 2 |
| wedge\_predtex\_tab\_idx | Table H‑17 | 0 | 1 | 2 |
| dmm\_delta\_end\_flag dmm\_delta\_end\_abs\_minus1 |  | 0 | 1 | 2 |
| dmm\_dc\_1\_abs dmm\_dc\_2\_abs |  | 0 | 1 | 2 |
| ic\_flag |  |  | 0 | 1 |
| mvp\_l0\_idx mpv\_l1\_idx |  |  | 0..1 | 2..3 |
| sdc\_flag |  | 0..2 | 3..5 | 6..8 |
| sdc\_residual\_flag |  | 0..1 | 2..3 | 4..5 |
| sdc\_residual\_sign\_flag |  | 0 | 1 | 2 |
| sdc\_residual\_abs\_minus1 |  | 0..19 | 20..39 | 40..59 |
| sdc\_pred\_mode |  | 0..2 | 3..5 | 6..8 |
| inter\_sdc\_flag | xx | 0 | 1 | 2 |
| inter\_sdc\_resi\_abs\_minus1 |  | 0 | 1 | 2 |
| inter\_sdc\_resi\_sign\_flag |  | 0 | 1 | 2 |

Table H‑xx – Values of variable initValue for inter\_sdc\_flag ctxIdx

|  |  |  |  |
| --- | --- | --- | --- |
| **Initialization variable** | **inter\_sdc\_flag** | | |
| **0** | **1** | **2** |
| **initValue** | 154 | 154 | 154 |

Table H‑xx – Values of variable initValue for inter\_sdc\_resi\_abs\_minus1 ctxIdx

|  |  |  |  |
| --- | --- | --- | --- |
| **Initialization variable** | **inter\_sdc\_resi\_abs\_minus1** | | |
| **0** | **1** | **2** |
| **initValue** | 154 | 154 | 154 |

Table H‑xx – Values of variable initValue for inter\_sdc\_resi\_sign\_flag ctxIdx

|  |  |  |  |
| --- | --- | --- | --- |
| **Initialization variable** | **inter\_sdc\_resi\_sign\_flag** | | |
| **0** | **1** | **2** |
| **initValue** | 154 | 154 | 154 |

H.9.3.2 Binarization process

Table H‑27 – Syntax elements and associated types of binarization, maxBinIdxCtx, ctxIdxTable, and ctxIdxOffset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| sdc\_flag | 0 | FL, cMax = 1 | 0 |  | 0 |
| 1 | 0 |  | 3 |
| 2 | 0 |  | 6 |
| inter\_sdc\_flag | 0 | FL, cMax = 1 | 0 | xx | 0 |
| 1 | 0 | xx | 1 |
| 2 | 0 | xx | 2 |
| inter\_sdc\_resi\_abs\_minus1 | 0 | UEG0 | 0 | xx | 0 |
| 1 | 0 | xx | 1 |
| 2 | 0 | xx | 2 |
| inter\_sdc\_resi\_sign\_flag | 0 | FL, cMax = 1 | 0 | xx | 0 |
| 1 | 0 | xx | 1 |
| 2 | 0 | xx | 2 |

H.9.3.3.1 Derivation process for ctxIdx

Table H‑30 – Assignment of ctxIdxInc to syntax elements with context coded bins

| **Syntax element** | **binIdx** | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **>=5** |
| dmm\_flag | 0 | na | na | na | na | na |
| dmm\_mode | 0 | 0 | 0 | 0 | 0 | 0 |
| wedge\_full\_tab\_idx | 0 | 0 | 0 | 0 | 0 | 0 |
| dmm\_delta\_end\_flag | 0 | na | na | na | na | na |
| dmm\_delta\_end\_abs\_minus1 | 0 | 0 | 0 | 0 | 0 | 0 |
| dmm\_dc\_1\_abs dmm\_dc\_2\_abs | 0 | 0 | 0 | 0 | 0 | 0 |
| res\_pred\_flag | 0 | na | na | na | na | na |
| ic\_flag | 0 | na | na | na | na | na |
| mvp\_l0\_idx, mvp\_l1\_idx | 0 | 1 | na | na | na | na |
| 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 |
| sdc\_flag | 0,1,2 (subclause G.9.3.3.1.1) | na | na | na | na | na |
| sdc\_residual\_flag | 0,1 (subclause G.9.3.3.1.7) | na | na | na | na | na |
| sdc\_residual\_sign\_flag | 0,1 (subclause G.9.3.3.1.7) | na | na | na | na | na |
| sdc\_residual\_abs\_minus1 | 0,10 (subclause H.9.3.3.1.7) | | | | | |
| sdc\_pred\_mode | 0 | 1 | 2 | na | na | na |
|  |  |  |  |  |  |  |