|  |  |
| --- | --- |
| seq\_parameter\_set\_rbsp( ) { | Descriptor |
| … |  |
| if(nuh\_layer\_id>0) { |  |
| **ilsao\_enable\_flag** | u(1) |
| } |  |
| **sps\_extension\_flag** | u(1) |
| … |  |
| } |  |

|  |  |
| --- | --- |
| slice\_segment\_header( ) { | Descriptor |
| … |  |
| if( !dependent\_slice\_segment\_flag ) { |  |
| … |  |
| if( nuh\_layer\_id > 0 && NumDirectRefLayers[ nuh\_layer\_id ] > 0 ) { |  |
| … |  |
| if( inter\_layer\_pred\_enabled\_flag) { |  |
| if ( ilsao\_enable\_flag && first\_slice\_segment\_in\_pic\_flag) |  |
| ilsao\_param( 0 ) |  |
| ilsao\_param( 1 ) |  |
| } |  |
| } |  |
| } |  |
| } |  |
| … |  |
| byte\_alignment( ) |  |
| } |  |

**ilsao\_enable\_flag** equal to 1 specifies that the interlayer sample adaptive offset process is applied to the reconstructed picture for inter-layer prediction. ilsao\_enable\_flag equal to 0 specifies that the sample adaptive offset process is not applied to the reconstructed picture for inter-layer prediction.

|  |  |
| --- | --- |
| ilsao\_param( saoIdx ){ | Descriptor |
| for( cIdx = 0; cIdx < 3; cIdx++ ) |  |
| if( cIdx = = 0 ) |  |
| **ilsao\_type\_idx\_luma** | ue(v) |
| else if( cIdx = = 1 ) |  |
| **ilsao\_type\_idx\_chroma** | ue(v) |
| if( SaoTypeIdx[ cIdx ][ saoIdx ] != 0 ) { |  |
| for( i = 0; i < 4; i++ ) |  |
| **ilsao\_offset\_abs**[ cIdx ][ saoIdx ] [ i ] | ue(v) |
| if( SaoTypeIdx[ cIdx ][ rx ] = = 1 ) { |  |
| for( i = 0; i < 4; i++ ) |  |
| if( sao\_offset\_abs[ cIdx ][ saoIdx ] [ i ] != 0 ) |  |
| **ilsao\_offset\_sign**[ cIdx ][ saoIdx ] [ i ] | u(1) |
| **ilsao\_band\_position**[ cIdx ][ saoIdx ] | ue(v) |
| } else { |  |
| if( cIdx = = 0 ) |  |
| **ilsao\_eo\_class\_luma** | u(2) |
| if( cIdx = = 1 ) |  |
| **ilsao\_eo\_class\_chroma** | u(2) |
| } |  |
| } |  |
| } |  |
| } |  |

**ilsao\_type\_idx\_luma** specifies the offset type for the luma component. The array IlSaoTypeIdx[ cIdx ][ saoIdx ] specifies the offset type as specified in Table 7‑8 for the coding tree block at the location ( rx, ry ) for the colour component cIdx. The value of IlSaoTypeIdx[ 0 ][ cIdx ][ saoIdx ]is derived as follows:

* If ilsao\_type\_idx\_luma is present, IlSaoTypeIdx [0][ saoIdx ] is set equal to ilsao\_type\_idx\_luma.

**ilsao\_type\_idx\_chroma** specifies the offset type for the chroma components. The values of IlSaoTypeIdx[ cIdx ][ saoIdx ] are derived as follows for cIdx equal to 1..2:

* If ilsao\_type\_idx\_chroma is present, IlSaoTypeIdx[ cIdx ][ saoIdx ] is set equal to ilsao\_type\_idx\_chroma.

Table A.1 – Specification of the ILSAO type

|  |  |
| --- | --- |
| **IlSaoTypeIdx[ cIdx ][ rx ][ ry ]** | **ILSAO type (informative)** |
| 0 | Not applied |
| 1 | Band offset |
| 2 | Edge offset |

**ilsao\_offset\_abs**[ cIdx ][ saoIdx ][ i ] specifies the offset value of i-th category for the ILSAO process saoIdx for the colour component cIdx.

**ilsao\_offset\_sign**[ cIdx ][ saoIdx ][ i ] specifies the sign of the offset value of i-th category for the ILSAO process saoIdx for the colour component cIdx when IlSaoTypeIdx[ cIdx ][ saoIdx ] is equal to 1.

The variable offsetSign is derived as follows:

* If IlSaoTypeIdx[ cIdx ][ saoIdx ] is equal to 2 and i is equal to 2 or 3, offsetSign is set equal to −1.
* Otherwise, if IlSaoTypeIdx[ cIdx ][ saoIdx ] is equal to 2 and i is equal to 0 or 1, offsetSign is set equal to 1.
* Otherwise (IlSaoTypeIdx[ cIdx ][ saoIdx ] is equal to 1), the following applies:
* If ilsao\_offset\_sign[ cIdx ][ saoIdx ][ i ] is equal to 0, offsetSign is set equal to equal to 1.
* Otherwise, offsetSign is set equal to equal to −1.

The variable bitDepth is derived as follows:

* If cIdx is equal to 0, bitDepth is set equal to BitDepthY.
* Otherwise (cIdx is equal to 1 or 2), bitDepth is set equal to BitDepthC.

The list IlSaoOffsetVal[ cIdx ][ rx ][ ry ][ i ] for i ranging from 0 to 4, inclusive, is derived as follows:

IlSaoOffsetVal[ cIdx ][ rx ][ ry ][ 0 ] = 0  
for( i = 0; i < 4; i++ )  
IlSaoOffsetVal[ cIdx ][ rx ][ ry ][ i + 1 ] = (7‑58)  
 offsetSign \* ilsao\_offset\_abs[ cIdx ][ rx ][ ry ][ i ]  <<  ( bitDepth − Min( bitDepth, 10 ) )

**ilsao\_band\_position**[ cIdx ][ saoIdx ] specifies the displacement of the band offset of the sample range when IlSaoTypeIdx[ cIdx ][ saoIdx ] is equal to 1.

**ilsao\_eo\_class**\_**luma** specifies the edge offset class for the luma component. The array IlSaoEoClass[ cIdx ][ saoIdx ] specifies the offset type as specified in Table 7‑9 for the ILSAO process saoIdx for the colour component cIdx. The value of IlSaoEoClass[ 0 ][ saoIdx ] is derived as follows:

* If ilsao\_eo\_class\_luma is present, IlSaoEoClass[ 0 ][ saoIdx ] is set equal to ilsao\_eo\_class\_luma.

**ilsao\_eo\_class\_chroma** specifies the edge offset class for the chroma components. The values of IlSaoEoClass[ cIdx ][ saoIdx ] are derived as follows for cIdx equal to 1..2:

Table A.2 – Specification of the ILSAO edge offset class

|  |  |
| --- | --- |
| **IlSaoEoClass[ cIdx ][ saoIdx ]** | **ILSAO edge offset class (informative)** |
| 0 | 1D 0-degree edge offset |
| 1 | 1D 90-degree edge offset |
| 2 | 1D 135-degree edge offset |
| 3 | 1D 45-degree edge offset |