# Draft Text Specification

The proposed text changes are based on the document JCTVC-L1005-v4-JCTVC-L1003\_v34.doc. The changes are marked in yellow.

### Method 1 (average of vertical and horizontal neighbors)

**8.4.4.2.1 General intra sample prediction**

Inputs to this process are:

– a sample location ( xTbCmp, yTbCmp ) specifying the top-left sample of the current transform block relative to the top‑left sample of the current picture,

– a variable predModeIntra specifying the intra prediction mode,

– a variable nTbS specifying the transform block size,

– a variable cIdx specifying the colour component of the current block.

– a variable cIdx specifying the colour component of the current block~~.~~,

* the residual sample array values r[ x ][ y ], with x, y = 0..nTbS−1.

**8.4.4.2.5 Specification of intra prediction mode INTRA\_DC**

Inputs to this process are:

– the neighbouring samples p[ x ][ y ], with x = −1, y = −1..nTbS \* 2 − 1 and x = 0..nTbS \* 2 − 1, y = −1,

– a variable nTbS specifying the transform block size,

– a variable cIdx specifying the colour component of the current block.

Outputs of this process are the predicted samples predSamples[ x ][ y ], with x, y = 0..nTbS − 1.

If cu\_transquant\_bypass\_flag is equal to 0, the ~~The~~ values of the prediction samples predSamples[ x ][ y ], with x, y = 0..nTbS − 1, are derived by the following ordered steps:

1. A variable dcVal is derived as follows:

dcVal =  (8‑41)

where k = Log2( nTbS ).

1. Depending on the value of the colour component index cIdx, the following applies:

* If cIdx is equal to 0 and nTbS is less than 32, the following applies:

predSamples[ 0 ][ 0 ] = ( p[ −1 ][ 0 ] + 2 \* dcVal + p[ 0 ][ −1 ] + 2 )  >>  2 (8‑42)

predSamples[ x ][ 0 ] = ( p[ x ][ −1 ] + 3 \* dcVal + 2 )  >>  2, with x = 1..nTbS − 1 (8‑43)

predSamples[ 0 ][ y ] = ( p[ −1 ][ y ] + 3 \* dcVal + 2 )  >>  2, with y = 1..nTbS − 1 (8‑44)

predSamples[ x ][ y ] = dcVal, with x, y = 1..nTbS − 1 (8‑45)

* Otherwise, the prediction samples predSamples[ x ][ y ] are derived as follows:

predSamples[ x ][ y ] = dcVal, with x, y = 0..nTbS − 1 (8‑46)

Otherwise (if cu\_transquant\_bypass\_flag is equal to 1) the values of the prediction samples predSamples[ x ][ y ], with x, y = 0..nTbS − 1, are derived as follows:

– predSamples[x][y] = .

The neighboring samples p[ x ][ y ], with x, y = 0..nTbS − 1, are reconstructed by

– p[x][y] = r[x][y] + predSamples[x][y].

### Method 2 (vertical plus horizontal minus diagonal up-left)

**8.4.4.2.1 General intra sample prediction**

Inputs to this process are:

– a sample location ( xTbCmp, yTbCmp ) specifying the top-left sample of the current transform block relative to the top‑left sample of the current picture,

– a variable predModeIntra specifying the intra prediction mode,

– a variable nTbS specifying the transform block size,

– a variable cIdx specifying the colour component of the current block.

– a variable cIdx specifying the colour component of the current block~~.~~,

* the residual sample array values r[ x ][ y ], with x, y = 0..nTbS−1.

**8.4.4.2.5 Specification of intra prediction mode INTRA\_DC**

Inputs to this process are:

– the neighbouring samples p[ x ][ y ], with x = −1, y = −1..nTbS \* 2 − 1 and x = 0..nTbS \* 2 − 1, y = −1,

– a variable nTbS specifying the transform block size,

– a variable cIdx specifying the colour component of the current block.

Outputs of this process are the predicted samples predSamples[ x ][ y ], with x, y = 0..nTbS − 1.

If cu\_transquant\_bypass\_flag is equal to 0, the ~~The~~ values of the prediction samples predSamples[ x ][ y ], with x, y = 0..nTbS − 1, are derived by the following ordered steps:

1. A variable dcVal is derived as follows:

dcVal =  (8‑41)

where k = Log2( nTbS ).

1. Depending on the value of the colour component index cIdx, the following applies:

* If cIdx is equal to 0 and nTbS is less than 32, the following applies:

predSamples[ 0 ][ 0 ] = ( p[ −1 ][ 0 ] + 2 \* dcVal + p[ 0 ][ −1 ] + 2 )  >>  2 (8‑42)

predSamples[ x ][ 0 ] = ( p[ x ][ −1 ] + 3 \* dcVal + 2 )  >>  2, with x = 1..nTbS − 1 (8‑43)

predSamples[ 0 ][ y ] = ( p[ −1 ][ y ] + 3 \* dcVal + 2 )  >>  2, with y = 1..nTbS − 1 (8‑44)

predSamples[ x ][ y ] = dcVal, with x, y = 1..nTbS − 1 (8‑45)

* Otherwise, the prediction samples predSamples[ x ][ y ] are derived as follows:

predSamples[ x ][ y ] = dcVal, with x, y = 0..nTbS − 1 (8‑46)

Otherwise (if cu\_transquant\_bypass\_flag is equal to 1) the values of the prediction samples predSamples[ x ][ y ], with x, y = 0..nTbS − 1, are derived as follows:

– predSamples[x][y] = .

The neighboring samples p[ x ][ y ], with x, y = 0..nTbS − 1, are reconstructed by

– p[x][y] = r[x][y] + predSamples[x][y].