**Draft text changes**

In 8.4.4

….

– ~~When the derivation process for z-scan order block availability as specified in subclause  is invoked with ( xCurr, yCurr ) set equal to ( xCb, yCb ) and the neighbouringluma location ( xNbY, yNbY ) set equal to ( xCb + bvIntra[ 0 ], yCb + bvIntra[ 1 ] ) as inputs, the output shall be equal to TRUE.~~

* When the derivation process for z-scan order block availability as specified in subclause 6.4.1 is invoked with ( xCurr, yCurr ) set equal to ( xCb + nCbS - 1, yCb + nCbS - 1 ) and the neighbouringluma location ( xNbY, yNbY ) set equal to ( xCb + bvIntra[ 0 ] + nCbS - 1 , yCb + bvIntra[ 1 ] + nCbS - 1) as inputs, the output shall be equal to TRUE.

~~– One or both of the following conditions shall be true:~~

~~– bvIntra[ 0 ] + nCbS<= 0~~

~~– bvIntra[ 1 ] + nCbS<= 0~~

In 8.4.5.2.7

….

The variable bv representing the block vector for prediction in full-sample units is derived as follows:

bv[ 0 ] = bvIntra[ 0 ] >> ( ( ( cIdx = = 0 ) ? 1 : SubWidthC ) − 1 ) (8‑63)

bv[ 1 ] = bvIntra[ 1 ] >> ( ( ( cIdx = = 0 ) ? 1 : SubHeightC ) − 1 ) (8‑64)

The (nTbS)x(nTbS) array of predicted samples samples, with x, y = 0..nTbS − 1, are derived as follows:

– The reference sample location (xRefCmp, yRefCmp ) is specified by:

( xRefCmp, yRefCmp ) = ( xTbCmp + x + bv[ 0 ], yTbCmp + y + bv[ 1 ] ) (8‑65)

If xRefCmp >= xTbCmp and yRefCmp >= yTbCmp, then (xRefCmp, yRefCmp) is modified as follows:

* If xTbCmp = 0 and bv[0] = 0

The variable bitDepth is derived as follows:

* + - If cIdx is equal to 0, bitDepth is specified as BitDepthY.
    - Otherwise, bitDepth is specified as BitDepthC.

Each sample at the location ( xRefCmp, yRefCmp ) is specified by (1<<(bitDepth-1)).

– Otherwise

(xRefCmp, yRefCmp) = (xTbCmp + bv[0] - 1, yRefCmp + bv[1]).

* Each sample at the location ( xRefCmp, yRefCmp )is assigned to predSamples[ x ][ y ].