



# **JCTVC-L0181/JCT3V-C0041: Proposed VPS extension semantics and editorial cleanups to syntax**

Jill Boyce, Ye-Kui Wang, Sachin Deshpande  
**Vidyo, Qualcomm, Sharp**



# Introduction

- **Current working design for VPS extension syntax document in JCTVC-K1007 (and JCT3V-B1007)**
  - Many semantics missing
- **Minor editorial syntax change proposed, does not change bitstream**
- **Bug fix proposed**
- **Semantics proposed**

# Scalability mask

- Change from single u(16) syntax element to loop of 16 u(1) flags
  - Makes it easier to address individual flag values

vps_extension( ) {	Descriptor
while( !byte_aligned( ) )	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
<del>scalability_mask</del>	<del>u(16)</del>
for( i = 0, numScalabilityTypes = 0 ; i < 16; i++ ) {	
scalability_mask_flag[ i ]	u(1)
numScalabilityTypes += scalability_mask_flag[ i ]	
}	

scalability\_mask\_flag [ i ] equal to 1 indicates that dimension\_id syntax elements corresponding to the i-th scalability dimension in Table XX are present.  
scalability\_mask\_flag[ i ] equal to 0 indicates that dimension\_id syntax elements corresponding to the i-th scalability dimension are not present.

# Fix syntax error

- JCTVC-K1007/JCT3V-B1007 mistakenly refers to a syntax element that had been removed from the working design, num\_dimensions\_minus1
- Proposed syntax modification is the intent of adopted aspect of contribution JCTVC-K0223
  - loop over numScalabilityTypes variable

for( i = 1; i <= vps_max_layers_minus1; i++ ) {	
if( vps_nuh_layer_id_present_flag )	
layer_id_in_nuh[ i ]	u(6)
for( j = 0; j < numScalabilityTypesnum_dimensions_minus1; j++ )	
dimension_id[ i ][ j ]	u(v)
}	

# Semantics

- Clarify relationship between scalability mask dimensions and DependencyId, QualityId, DepthFlag, and ViewId

scalability_mask_flag index	Scalability dimension	ScalabilityId mapping
0	None (base HEVC)	
10	spatial	DependencyId
21	quality	QualityId
32	depth	DepthFlag
43	multiview	ViewId
54 - 15	Reserved/unspecified	Reserved

scalability\_mask\_flag [ i ] equal to 1 indicates that dimension\_id syntax elements corresponding to the i-th scalability dimension in Table XX are present. scalability\_mask\_flag[ i ] equal to 0 indicates that dimension\_id syntax elements corresponding to the i-th scalability dimension are not present.

~~signals a pattern of 0 and 1 bits with each bit corresponding to one scalability dimension as indicated by the table below. A value of 1 for a particular scalability dimension indicates that this scalability dimension is present. A value of 0 for a particular scalability dimension indicates that this scalability dimension is not present. The values of NumScalabilityTypes is equal to the sum of number of bits in the scalability\_mask having value of 1. Thus~~

~~$$\text{NumScalabilityTypes} = \sum_{k=0}^{15} \text{scalability\_mask}(k)$$~~

# Semantics (continued)

**dimension\_id\_len\_minus1**[ j ] plus 1 specifies the length, in bits, of the **dimension\_id**[ i ][ j ] syntax element.

**vps\_nuh\_layer\_id\_present\_flag** specifies whether the **layer\_id\_in\_nuh**[ i ] syntax is present.

**layer\_id\_in\_nuh**[ i ] specifies the value of the **nuh\_layer\_id** syntax element in VCL NAL units of the i-th layer. When not present, the value of **layer\_id\_in\_nuh**[ i ] is inferred to be equal to i.

**dimension\_id**[ i ][ j ] specifies the identifier of the j-th scalability dimension type of the i-th layer. When not present, the value of **dimension\_id**[ i ][ j ] is inferred to be equal to 0. The number of bits used for the representation of **dimension\_id**[ i ][ j ] is **dimension\_id\_len\_minus1**[ j ] + 1 bits.

The following applies:

```
for( i = 0; i <= vps_max_layers_minus1; i++ )
    for( smIdx= 0, j=0; smIdx< 16; smIdx ++ )
        if( ( i != 0 ) && scalability_mask_flag[ smIdx ] )
            ScalabilityId[ i ][ smIdx ] = dimension_id[ i ][ j++ ]
        else
            ScalabilityId[ i ][ smIdx ] = 0
```

The following applies:

- For each layer indexed by  $i = 0..vps\_max\_layers\_minus1 + 1$ 
  - DependencyId[ **layer\_id\_in\_nuh**[ i ] ] = ScalabilityId[ i ][ 0 ]
  - QualityId[ **layer\_id\_in\_nuh**[ i ] ] = ScalabilityId[ i ][ 1 ]
  - DepthFlag[ **layer\_id\_in\_nuh**[ i ] ] = ScalabilityId[ i ][ 2 ]
  - ViewId[ **layer\_id\_in\_nuh**[ i ] ] = ScalabilityId[ i ][ 3 ]