|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  10th Meeting: Stockholm, SE, 11 – 20 July 2012 | Document: JCTVC-J0550 text |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **AHG10: NUH, VPS and SPS syntax designs agreed by the BoG on VPS and NUH** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* |  | | |
| *Author(s) or Contact(s):* | Ye-Kui Wang and Jill Boyce | Tel: Email: |  |
| *Source:* | BoG on VPS and NUH | | |

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

# Abstract

The v2 version of this document describes the syntax and semantics capturing the Track B adoptions based upon the BoG recommendations.The v3 version of this document removes the SEI message syntax, which is included in the r1 version of J0261.

# NAL unit header syntax and semantics

|  |  |
| --- | --- |
| nal\_unit( NumBytesInNALunit ) { | Descriptor |
| **forbidden\_zero\_bit** | f(1) |
| **~~nal\_ref\_flag~~** | ~~u(1)~~ |
| **nal\_unit\_type** | u(6) |
| **reserved\_zero\_6bits** | u(6) |
| **temporal\_id\_plus1** | u(3) |
| **~~reserved\_one\_5bits~~** | ~~u(5)~~ |
| NumBytesInRBSP = 0 |  |
| for( i = 2; i < NumBytesInNALunit; i++ ) { |  |
| if( i + 2 < NumBytesInNALunit && next\_bits( 24 ) = = 0x000003 ) { |  |
| **rbsp\_byte[** NumBytesInRBSP++ **]** | b(8) |
| **rbsp\_byte[** NumBytesInRBSP++ **]** | b(8) |
| i += 2 |  |
| **emulation\_prevention\_three\_byte** /\* equal to 0x03 \*/ | f(8) |
| } else |  |
| **rbsp\_byte[** NumBytesInRBSP++ **]** | b(8) |
| } |  |
| } |  |

NOTE 1 – The VCL is specified to efficiently represent the content of the video data. The NAL is specified to format that data and provide header information in a manner appropriate for conveyance on a variety of communication channels or storage media. All data are contained in NAL units, each of which contains an integer number of bytes. A NAL unit specifies a generic format for use in both packet-oriented and bitstream systems. The format of NAL units for both packet-oriented transport and byte stream is identical except that each NAL unit can be preceded by a start code prefix and extra padding bytes in the byte stream format.

NumBytesInNALunit specifies the size of the NAL unit in bytes. This value is required for decoding of the NAL unit. Some form of demarcation of NAL unit boundaries is necessary to enable inference of NumBytesInNALunit. One such demarcation method is specified in Annex B for the byte stream format. Other methods of demarcation may be specified outside of this Recommendation | International Standard.

**forbidden\_zero\_bit** shall be equal to 0.

**~~nal\_ref\_flag~~** ~~equal to 1 specifies that the content of the NAL unit contains a sequence parameter set, a picture parameter set, an adaptation parameter set or a slice of a reference picture.~~

~~nal\_ref\_flag equal to 0 for a NAL unit containing a slice indicates that the slice is part of a non-reference picture.~~

~~nal\_ref\_flag shall be equal to 1 for video parameter set, sequence parameter set, picture parameter set or adaptation parameter set NAL units. When nal\_ref\_flag is equal to 0 for one VCL NAL unit of a particular picture, it shall be equal to 0 for all VCL NAL units of the particular picture.~~

~~nal\_ref\_flag shall be equal to 1 for NAL units with nal\_unit\_type equal to 4, 5, 6, 7, or 8.~~

~~nal\_ref\_flag shall be equal to 0 for all NAL units having nal\_unit\_type equal to 29, 30, or 31.~~

**nal\_unit\_type** specifies the type of RBSP data structure contained in the NAL unit as specified in Table 7‑1.

NAL units that use nal\_unit\_type equal to 0 or in the range of 48..63, inclusive, shall not affect the decoding process specified in this Recommendation | International Standard.

NOTE 2 – NAL unit types 0 and 48..63 may be used as determined by the application. No decoding process for these values of nal\_unit\_type is specified in this Recommendation | International Standard. Since different applications might use NAL unit types 0 and 48..63 for different purposes, particular care must be exercised in the design of encoders that generate NAL units with nal\_unit\_type equal to 0 or in the range of 48 to 63, inclusive, and in the design of decoders that interpret the content of NAL units with nal\_unit\_type equal to 0 or in the range of 48 to 63, inclusive.

Decoders shall ignore (remove from the bitstream and discard) the contents of all NAL units that use reserved values of nal\_unit\_type.

NOTE 3 – This requirement allows future definition of compatible extensions to this Recommendation | International Standard.

Table 7‑1 – NAL unit type codes and NAL unit type classes

|  |  |  |
| --- | --- | --- |
| **nal\_unit\_type** | **Content of NAL unit and RBSP syntax structure** | **NAL unit type class** |
| 0 | Unspecified | non-VCL |
| 1 | Coded slice of a non-RAP, non-TFD and non-TLA picture slice\_layer\_rbsp( ) | VCL |
| 2 | Coded slice of a TFD picture  slice\_layer\_rbsp( ) | VCL |
| 3 | Coded slice of a non-TFD TLA picture  slice\_layer\_rbsp( ) | VCL |
| 4, 5 | Coded slice of a CRA picture  slice\_layer\_rbsp( ) | VCL |
| 6, 7 | Coded slice of a BLA picture  slice\_layer\_rbsp( ) | VCL |
| 8 | Coded slice of an IDR picture slice\_layer\_rbsp( ) | VCL |
| 9..24 | Reserved | n/a |
| 25 | Video parameter set video\_parameter\_set\_rbsp( ) | non-VCL |
| 26 | Sequence parameter set seq\_parameter\_set\_rbsp( ) | non-VCL |
| 27 | Picture parameter set pic\_parameter\_set\_rbsp( ) | non-VCL |
| 28 | Adaptation parameter set aps\_rbsp( ) | non-VCL |
| 29 | Access unit delimiter access\_unit\_delimiter\_rbsp( ) | non-VCL |
| 30 | Filler data filler\_data\_rbsp( ) | non-VCL |
| 31 | Supplemental enhancement information (SEI) sei\_rbsp( ) | non-VCL |
| 32..47 | Reserved | n/a |
| 48..63 | Unspecified | non-VCL |

NOTE 4 - A CRA picture having nal\_unit\_type equal to 4 may have associated TFD pictures present in the bitstream. A CRA picture having nal\_unit\_type equal to 5 does not have associated TFD pictures present in the bitstream. A BLA picture having nal\_unit\_type equal to 6 may have associated TFD pictures present in the bitstream. A BLA picture having nal\_unit\_type equal to 7 does not have associated TFD pictures present in the bitstream.

Coded slice NAL unit collectively refers to a VCL NAL unit, which has nal\_unit\_type in the range of 1 to 8, inclusive. The variable IdrPicFlag is specified as

IdrPicFlag = ( ( nal\_unit\_type = = 8 ) ? 1 : 0 ) (‑)

The variable RapPicFlag is specified as

RapPicFlag = ( ( nal\_unit\_type >= 4 && nal\_unit\_type <= 8 ) ? 1 : 0 ) (‑)

When the value of nal\_unit\_type is equal to any particular value in the range of 1 to 8, inclusive, for a NAL unit of a particular picture, all VCL NAL units of that particular picture shall have nal\_unit\_type equal to that particular value.

When the value of nal\_unit\_type is equal to 4 or 5 for all VCL NAL units of a particular picture, the particular picture is referred to as a CRA picture.

When the value of nal\_unit\_type is equal to 6 or 7 for all VCL NAL units of a particular picture, the particular picture is referred to as a BLA picture.

All coded pictures that follow a CRA or BLA picture both in decoding order and output order shall not use inter prediction from any picture that precedes the CRA or BLA picture either in decoding order or output order, and any picture that precedes the CRA or BLA picture in decoding order shall also precede the CRA or BLA picture in output order.

[Ed. Note (GJS/YK): The constraint on inter prediction should be expressed as a constraint on the reference picture set or the final reference picture list, whichever is easier to express.]

It is a requirement of bitstream conformance that no TFD pictures shall be present in the bitstream that are associated with a CRA picture having nal\_unit\_type equal to 5 or a BLA picture having nal\_unit\_type equal to 7.

When the value of nal\_unit\_type is equal to 8 for all VCL NAL units of a particular picture, that particular picture is referred to as an IDR picture. All coded pictures that follow an IDR picture in decoding order shall not use inter prediction from any picture that precedes the IDR picture in decoding order, and any picture that precedes the IDR picture in decoding order shall also precede the IDR picture in output order.

[Ed. Note (GJS/YK): The constraint on inter prediction should be expressed as a constraint on the reference picture set or the final reference picture list, whichever is easier to express.]

RAP picture collectively refers to a coded picture that is a CRA picture, a BLA picture or an IDR picture, and RAP access unit collectively refers to an access unit that is a CRA access unit, a BLA access unit or an IDR access unit.

NOTE 5 - Any parameter set (video parameter set, sequence parameter set, picture parameter set, or adaptation parameter set) must be available before the activation of the parameter set. To be able to perform random access from any particular RAP picture by discarding all access units before the particular RAP access unit (and to correctly decode the particular RAP access unit and all the subsequent access units in both decoding and output order), the following condition must be satisfied: each parameter set that is activated during the decoding of the particular RAP access unit or during the decoding of any subsequent access unit in decoding order is either present or provided through external means at or subsequent to that particular RAP access unit and prior to any NAL unit activating that parameter set.

When the value of nal\_unit\_type is equal to 3 for all VCL NAL units of a particular picture, that particular picture is referred to as a TLA picture. A TLA picture and all coded pictures with temporal\_id greater than or equal to the temporal\_id of the TLA picture that follow the TLA picture in decoding order shall not use inter prediction from any picture with temporal\_id greater than or equal to the temporal\_id of the TLA picture that precedes the TLA picture in decoding order. A TLA picture shall not be TFD picture; hence TLA picture is also referred to as non-TFD TLA picture.

[Ed. Note (GJS/YK): The constraint on inter prediction should be expressed as a constraint on the reference picture set or the final reference picture list, whichever is easier to express.]

When temporal\_id\_nesting\_flag is equal to 1 and temporal\_id is greater than 0, nal\_unit\_type shall be equal to 3

**reserved\_zero\_6bits** shall be equal to 0. Other values of reserved\_zero\_6bits may be specified in the future by ITU‑T | ISO/IEC. Decoders shall ignore (i.e. remove from the bitstream and discard) NAL units with values of reserved\_zero\_6bits not equal to 0.

NOTE 6 – It is anticipated that in future scalable and/or 3DV extensions of this specification, this field is used to identify layers that may be present in the coded video sequence, wherein a layer may e.g. be a spatial scalable layer, a quality scalable layer, a texture view or a depth view.

**temporal\_id\_plus1** minus 1 specifies a temporal identifier for the NAL unit. The value of temporal\_id\_plus1 shall be the same for all VCL NAL units of an access unit.

The variable TemporalId is specified as

TemporalId = temporal\_id\_plus1 − 1 (‑)

When an access unit is a RAP access unit, TemporalId for all VCL NAL units of the access unit shall be equal to 0. When nal\_unit\_type is equal to 3, TemporalId shall not be equal to 0.

The temporal\_id of an access unit is derived as equal to the TemporalId value of the VCL NAL units in the access unit.

For a non-VCL NAL unit, the value of TemporalId shall be equal to the minimum value of the TemporalId values of all access units the non-VCL NAL unit applies to. When nal\_unit\_type is equal to 26 (sequence parameter set), TemporalId shall be equal to 0. When nal\_unit\_type is equal to 29 (access unit delimiter) or 30 (filler data), TemporalId shall be equal to the TemporalId of the access unit containing the non-VCL NAL unit. When nal\_unit\_type is equal to 27 (picture parameter set) or 28 (adaptation parameter set), TemporalId may be less than, equal to, or greater than the TemporalId of the containing access unit. When nal\_unit\_type is equal to 31 (SEI), TemporalId shall not be less than the TemporalId of the containing access unit.

NOTE 7 – When nal\_unit\_type is equal to 26 (sequence parameter set), TemporalId must be equal to 0, as a sequence parameter set applies at least to a RAP access unit. When nal\_unit\_type is equal to 29 (access unit delimiter) or 30 (filler data), TemporalId must be equal to the TemporalId of the access unit containing the non-VCL NAL unit, as access unit delimiter or filler data only applies to the containing access unit. When nal\_unit\_type is equal to 27 (picture parameter set) or 28 (adaptation parameter set), TemporalId may be less than, equal to, or greater than the TemporalId of the containing access unit, as a picture parameter set or adaptation parameter set may be repeated in access units not referring to the picture parameter set or adaptation parameter set for improved error resilience, and all picture parameter sets or adaptation parameter sets may be included in the beginning of a bitstream wherein the first coded picture has TemporalId equal to 0. When nal\_unit\_type is equal to 31 (SEI), TemporalId may be equal to or greater than the TemporalId of the containing access unit, as an SEI NAL unit may contain a picture buffering SEI message or a picture timing SEI message that applies to a bitstream subset including access units for which the TemporalId values are greater than the TemporalId of the access unit containing the SEI NAL unit.

**~~reserved\_one\_5bits~~** ~~shall be equal to '00001'. Other values of reserved\_one\_5bits may be specified in the future by ITU‑T | ISO/IEC. Decoders shall ignore (i.e. remove from the bitstream and discard) NAL units with values of reserved\_one\_5bits not equal to '00001'.~~

# Video parameter set RBSP syntax and semantics

|  |  |
| --- | --- |
| video\_parameter\_set\_rbsp( ) { | Descriptor |
| **video\_parameter\_set\_id** | u(4) |
| **temporal\_id\_nesting\_flag** | u(1) |
| **reserved\_zero\_2bits** | u(2) |
| **reserved\_zero\_6bits** | u(6) |
| **vps\_max\_sub\_layers\_minus1** | u(3) |
| profile\_level( 1, vps\_max\_sub\_layers\_minus1 ) |  |
| **next\_essential\_info\_byte\_offset** | u(12) |
| for( i = 0; i <= vps\_max\_sub\_layers\_minus1; i++ ) { |  |
| **max\_dec\_pic\_buffering[** i **]** | ue(v) |
| **max\_num\_reorder\_pics[** i **]** | ue(v) |
| **max\_latency\_increase[** i **]** | ue(v) |
| } |  |
| **vps\_extension\_flag** | u(1) |
| if( vps\_extension\_flag ) |  |
| while( more\_rbsp\_data( ) ) |  |
| **vps\_extension\_data\_flag** | u(1) |
| rbsp\_trailing\_bits( ) |  |
| } |  |

**video\_parameter\_set\_id** identifies a video parameter set. ~~The value of video\_parameter\_set\_id shall be in the range of 0 to 15, inclusive.~~

**vps\_max\_sub\_layers\_minus1** plus 1 specifies the maximum number of temporal sub-layers that may be present in the bitstream. The value of vps\_max\_sub\_layers\_minus1 shall be in the range of 0 to 6, inclusive.

[Ed.(YK): Add a definition of (temporal) sub-layer in Clause 3.]

**vps\_temporal\_id\_nesting\_flag** specifies whether inter prediction is additionally restricted for the coded video sequence.

Dependent on vps\_temporal\_id\_nesting\_flag, the following applies.

– If vps\_temporal\_id\_nesting\_flag is equal to 0, additional constraints may not be obeyed.

– Otherwise (vps\_temporal\_id\_nesting\_flag is equal to 1), the following applies.

– For each access unit auA with TemporalId equal to tIdA, an access unit auB with TemporalId equal to tIdB that is less than or equal to tIdA shall not be referenced by inter prediction when there exists an access unit auC with TemporalId equal to tIdC that is less than tIdB, which follows the access unit auB and precedes the access unit auA in decoding order.

NOTE 1 – The syntax element vps\_temporal\_id\_nesting\_flag is used to indicate that temporal up-switching, i.e., switching from decoding of up to a specific TemporalId tIdN to decoding up to a TemporalId tIdM that is greater than tIdN, is always possible.

**reserved\_zero\_6bits** shall be equal to 0 in bitstreams conforming to this Recommendation | International Standard. Other values for reserved\_zero\_6bits are reserved for future use by ITU-T | ISO/IEC. Decoders should ignore values of reserved\_zero\_6bits not equal to 0 in a video parameter set NAL unit.

NOTE 2 – It is anticipated that in future scalable and/or 3DV extensions of this specification, this field specifies the maximum number of layers that may be present in the coded video sequence, wherein a layer may e.g. be a spatial scalable layer, a quality scalable layer, a texture view or a depth view.

**reserved\_zero\_2bits** shall be equal to 0 in bitstreams conforming to this Recommendation | International Standard. Other values for reserved\_zero\_2bits are reserved for future use by ITU-T | ISO/IEC. Decoders should ignore values of reserved\_zero\_2bits not equal to 0 in a video parameter set NAL unit.

**next\_essential\_info\_byte\_offset** specifies the byte offset of the next set of profile and/or level information and other fixed-length coded information in the VPS NAL unit, starting from the beginning of the NAL unit. Decoders conforming to this Recommendation | International Standard shall ignore the value of next\_essential\_info\_byte\_offset.

NOTE 3 – In a future extension of this specification, e.g. a scalable coding extension or a 3DV extension, VPS information for non-base layer or view may be included in the VPS NAL unit, after the VPS information for the base layer or view. It is anticipated that the VPS information for non-base layer or view also starts from fixed-length coded profile, level and other information that are essential for session negotiation and/or capability exchange. The byte offset specified by next\_essential\_info\_byte\_offset would then help to locate and access that essential information in the VPS NAL unit without the need of entropy decoding, which may not be equipped with some network entities that may desire to access only all the information in the VPS that is essential for session negotiation and/or capability exchange.

**max\_dec\_pic\_buffering[** i **]** specifies the required size of the decoded picture buffer in units of picture storage buffers for the bitstream subset as specified in subclause 10.1 with i as input. The coded video sequence shall not require a decoded picture buffer with size of more than Max( 1, max\_dec\_pic\_buffering[ i ] ) picture storage buffers to enable the output of decoded pictures at the output times specified by dpb\_output\_delay of the picture timing SEI messages. The value of max\_dec\_pic\_buffering[ i ] shall be in the range of 0 to MaxDpbSize (as specified in subclause A.4), inclusive.

**max\_num\_reorder\_pics[** i **]** indicates the maximum allowed number of pictures preceding any picture in decoding order and succeeding that picture in output order for the bitstream subset as specified in subclause 10.1 with i as input. The value of max\_num\_reorder\_pics[ i ] shall be in the range of 0 to max\_dec\_pic\_buffering[ i ], inclusive.

**max\_latency\_increase[** i **]** not equal to 0 is used to compute the value of MaxLatencyPictures[ i ] as specified by setting MaxLatencyPictures[ i ] to max\_num\_reorder\_pics[ i ] + max\_latency\_increase[ i ] for the bitstream subset as specified in subclause 10.1 with i as input. When max\_latency\_increase[ i ] is not equal to 0, the value of MaxLatencyPictures[ i ] specifies the maximum number of pictures that can precede any picture in the coded video sequence in output order and follow that picture in decoding order for the bitstream subset as specified in subclause 10.1 with i as input. When max\_latency\_increase[ i ] is equal to 0, no corresponding limit is expressed. The value of max\_latency\_increase[ i ] shall be in the range of 0 to 232 − 1, inclusive.

**vps\_extension\_flag** equal to 0 specifies that no vps\_extension\_data\_flag syntax elements are present in the video parameter set RBSP syntax structure. vps\_extension\_flag shall be equal to 0 in bitstreams conforming to this Recommendation | International Standard. The value of 1 for vps\_extension\_flag is reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore all data that follow the value 1 for vps\_extension\_flag in a video parameter set NAL unit.

**vps\_extension\_data\_flag** may have any value. It shall not affect the conformance to profiles specified in this Recommendation | International Standard.

## Profile and level syntax and semantics

|  |  |
| --- | --- |
| profile\_level( ProfilePresentFlag, MaxNumSubLayersMinus1 ) { |  |
| if( ProfilePresentFlag ) { |  |
| **profile\_space** | u(3) |
| **profile\_idc** | u(5) |
| for( i = 0; i < 32; i++ ) |  |
| **profile\_compatability\_flag[** i **]** | u(1) |
| **constraint\_flags** | u(16) |
| } |  |
| **level\_idc** | u(8) |
| for( i = 0; i < MaxNumSubLayersMinus1; i++ ) { |  |
| **sub\_layer\_profile\_present\_flag[** i **]** | u(1) |
| **sub\_layer\_level\_present\_flag[** i **]** | u(1) |
| if( ProfilePresentFlag && sub\_layer\_profile\_present\_flag[ i ] ) { |  |
| **sub\_layer\_profile\_space[** i **]** | u(3) |
| **sub\_layer\_profile\_idc[** i **]** | u(5) |
| for( j = 0; j < 32; j++ ) |  |
| **sub\_layer\_profile\_compatability\_flag[** i **][** j **]** | u(1) |
| **sub\_layer\_constraint\_flags[** i **]** | u(16) |
| } |  |
| if( sub\_layer\_level\_present\_flag[ i ] ) |  |
| **sub\_layer\_level\_idc**[ i ] | u(8) |
| } |  |
| } |  |

**profile\_space** specifies the context for the interpretation of profile\_idc and profile\_combatibility\_flag[ i ] for all possible values of i. The value of profile\_space shall be equal to 0 in bitstreams conforming to this Recommendation | International Standard. Other values for profile\_space are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the coded video sequence if profile\_space is not equal to 0.

When profile\_space is equal to 0, **profile\_idc** indicates the profile to which the coded video sequence conforms.

[Ed. (DS): We might prefer not to use the profile\_idc value zero, or reserve it to mean "no profile signalled, bitstream is unconstrained"; this gives us one spare bit in the profile compatibility flags array.]

When profile\_space is equal to 0, **profile\_compatability\_flag[** i **]** equal to 1 indicates that the coded video sequence conforms to the profile indicated by profile\_idc equal to i. When profile\_space is equal to 0, profile\_compatability\_flag[ profile\_idc ] shall be equal to 1.

**constraint\_flags** shall be equal to 0 in bitstreams conforming to this Recommendation | International Standard. Other values for for constraint\_flags are reserved for future use by ITU-T | ISO/IEC. Decoders should ignore the value of constraint\_flags.

**level\_idc** indicates the maximum level to which the coded video sequence conforms when the profile\_level( ) syntax structure is included in a video parameter set. level\_idc indicates the level to which the coded video sequence conforms when the profile\_level( ) syntax structure is included in a sequence parameter set.

NOTE 1 – A greater value of level\_idc indicates a higher level. The maximum level signalled in the video parameter set for a coded video sequence may be higher than the level signalled in the sequence parameter set for the same coded video sequence.

NOTE 2 – The level\_idc definitions need to be independent of profile\_idc, as a decoder may not recognize the the profile\_idc value but instead recognize, and be able to decode, one or more of the profiles indicated by the profile compatibility flags.

NOTE 3 – For bitstreams compatible with more than one profile, the value of profile\_idc should be set to the "best viewed as" profile.

NOTE 4 – The constraint\_flags can be used to indicate the respect of further constraints on the bitstream (e.g. that a selected tool, permitted by the profiles signalled, is nonetheless not used). These flags are ideally profile independent, but unlike level\_idc, it is allowed to be contextual on profile\_idc.

**sub\_layer\_profile\_present\_flag[** i **]** equal to 1 specifies that profile information may be present in the profile\_level( ) syntax structure for the representation of the sub-layer with TemporalId equal to i. sub\_layer\_profile\_present\_flag**[** i **]** equal to 0 specifies that profile information is not present in the profile\_level( ) syntax structure for the representations of the sub-layer with TemporalId equal to i.

[Ed.(YK): Add a definition of (temporal) sub-layer representation or representation of (temporal) sub-layer in Clause 3 to consist of all NAL units with TemporalId not greater than the TemporalId of the sub-layer.]

**sub\_layer\_level\_present\_flag[** i **]** equal to 1 specifies that level information is present in the profile\_level( ) syntax structure for the representation of the sub-layer with TemporalId equal to i. sub\_layer\_level\_present\_flag**[** i **]** equal to 0 specifies that level information is not present in the profile\_level( ) syntax structure for the representation of the sub-layer with TemporalId equal to i.

**sub\_layer\_profile\_space[** i **]**, **sub\_layer\_profile\_idc[** i **]** , **sub\_layer\_profile\_compatability\_flag[** i **][** j **]** , **sub\_layer\_constraint\_flags[** i **]** , and **sub\_layer\_level\_idc**[ i ] have the semantics of profile\_space, profile\_idc, profile\_compatability\_flag[ j ], constraint\_flags, and level\_idc, respectively, but apply to the representation of the sub-layer with TemporalId equal to i.

# SPS syntax and semantics

|  |  |
| --- | --- |
| seq\_parameter\_set\_rbsp( ) { | Descriptor |
| **video\_parameter\_set\_id** | u(4) |
| **sps\_max\_sub\_layers\_minus1** | u(3) |
| **reserved\_zero\_bit** | u(1) |
| profile\_level( 1, sps\_max\_sub\_layers\_minus1 ) |  |
| **profile\_space** | u(3) |
| **profile\_idc** | u(5) |
| **constraint\_flags** | u(16) |
| **level\_idc** | u(8) |
| for( i = 0; i < 32; i++ ) |  |
| **profile\_compatability\_flag[** i **]** | u(1) |
| **seq\_parameter\_set\_id** | ue(v) |
| **video\_parameter\_set\_id** | ue(v) |
| **chroma\_format\_idc** | ue(v) |
| if( chroma\_format\_idc = = 3 ) |  |
| **separate\_colour\_plane\_flag** | u(1) |
| **sps\_max\_temporal\_layers\_minus1** | u(3) |
| **pic\_width\_in\_luma\_samples** | ue(v) |
| **pic\_height\_in\_luma\_samples** | ue(v) |
| **pic\_cropping\_flag** | u(1) |
| if( pic\_cropping\_flag ) { |  |
| **pic\_crop\_left\_offset** | ue(v) |
| **pic\_crop\_right\_offset** | ue(v) |
| **pic\_crop\_top\_offset** | ue(v) |
| **pic\_crop\_bottom\_offset** | ue(v) |
| } |  |
| **bit\_depth\_luma\_minus8** | ue(v) |
| **bit\_depth\_chroma\_minus8** | ue(v) |
| [Ed. (BB): chroma bit depth present in HM software but not used further ] |  |
| **pcm\_enabled\_flag** | u(1) |
| if( pcm\_enabled\_flag ) { |  |
| **pcm\_sample\_bit\_depth\_luma\_minus1** | u(4) |
| **pcm\_sample\_bit\_depth\_chroma\_minus1** | u(4) |
| } |  |
| **log2\_max\_pic\_order\_cnt\_lsb\_minus4** | ue(v) |
| for( i = 0; i <= sps\_max\_temporal\_layers\_minus1; i++ ) { |  |
| **sps\_max\_dec\_pic\_buffering[** i **]** | ue(v) |
| **sps\_num\_reorder\_pics[** i **]** | ue(v) |
| **sps\_max\_latency\_increase[** i **]** | ue(v) |
| } |  |
| **restricted\_ref\_pic\_lists\_flag** | u(1) |
| if( restricted\_ref\_pic\_lists\_flag ) |  |
| **lists\_modification\_present\_flag** | u(1) |
| **log2\_min\_coding\_block\_size\_minus3** | ue(v) |
| **log2\_diff\_max\_min\_coding\_block\_size** | ue(v) |
| **log2\_min\_transform\_block\_size\_minus2** | ue(v) |
| **log2\_diff\_max\_min\_transform\_block\_size** | ue(v) |
| if( pcm\_enabled\_flag ) { |  |
| **log2\_min\_pcm\_coding\_block\_size\_minus3** | ue(v) |
| **log2\_diff\_max\_min\_pcm\_coding\_block\_size** | ue(v) |
| } |  |
| **max\_transform\_hierarchy\_depth\_inter** | ue(v) |
| **max\_transform\_hierarchy\_depth\_intra** | ue(v) |
| **scaling\_list\_enable\_flag** | u(1) |
| if( scaling\_list\_enable\_flag ) { |  |
| **sps\_scaling\_list\_data\_present\_flag** | u(1) |
| if( sps\_scaling\_list\_data\_present\_flag ) |  |
| scaling\_list\_param( ) |  |
| } |  |
| **chroma\_pred\_from\_luma\_enabled\_flag** | u(1) |
| **transform\_skip\_enabled\_flag** | u(1) |
| **seq\_loop\_filter\_across\_slices\_enabled\_flag** | u(1) |
| **asymmetric\_motion\_partitions\_enabled\_flag** | u(1) |
| **nsrqt\_enabled\_flag** | u(1) |
| **sample\_adaptive\_offset\_enabled\_flag** | u(1) |
| **adaptive\_loop\_filter\_enabled\_flag** | u(1) |
| if( pcm\_enabled\_flag ) |  |
| **pcm\_loop\_filter\_disable\_flag** | u(1) |
| **sps\_temporal\_id\_nesting\_flag** | u(1) |
| [Ed. (BB): x y padding syntax missing here, present in HM software ] |  |
| **num\_short\_term\_ref\_pic\_sets** | ue(v) |
| for( i = 0; i < num\_short\_term\_ref\_pic\_sets; i++) |  |
| short\_term\_ref\_pic\_set( i ) |  |
| **long\_term\_ref\_pics\_present\_flag** | u(1) |
| **sps\_temporal\_mvp\_enable\_flag** | u(1) |
| **vui\_parameters\_present\_flag** | u(1) |
| if( vui\_parameters\_present\_flag ) |  |
| vui\_parameters( ) |  |
| **sps\_extension\_flag** | u(1) |
| if( sps\_extension\_flag ) |  |
| while( more\_rbsp\_data( ) ) |  |
| **sps\_extension\_data\_flag** | u(1) |
| rbsp\_trailing\_bits( ) |  |
| } |  |

**video\_parameter\_set\_id** identifies a video parameter set. ~~The value of video\_parameter\_set\_id shall be in the range of 0 to 15, inclusive.~~

**sps\_max\_sub\_layers\_minus1** plus 1 specifies the maximum number of temporal sub-layers that may be present in the bitstream. The value of sps\_max\_sub\_layers\_minus1 shall be in the range of 0 to 6, inclusive.

**reserved\_zero\_bit** shall be equal to 0 in bitstreams conforming to this Recommendation | International Standard. The value 1 for reserved\_zero\_bit is reserved for future use by ITU-T | ISO/IEC. Decoders should ignore values of reserved\_zero\_bit not equal to 0 in a video parameter set NAL unit.

Semantics of the removed syntax elements are removed, and semantics of remaining syntax elements remain unchanged.

# Decoding process

## General [Ed. (GJS): Create similar subclauses for other places where there is "orphaned" text that is at the parent level of a section that contains subclauses.]

Outputs of this process are decoded samples of the current picture (sometimes referred to by the variable CurrPic).

Depending on the value of chroma\_format\_idc, the number of sample arrays of the current picture is as follows.

– If chroma\_format\_idc is equal to 0, the current picture consists of 1 sample array SL.

– Otherwise (chroma\_format\_idc is not equal to 0), the current picture consists of 3 sample arrays SL, SCb, SCr.

This clause describes the decoding process, given syntax elements and upper-case variables from clause 7.

The decoding process is specified such that all decoders shall produce numerically identical results. Any decoding process that produces identical results to the process described here conforms to the decoding process requirements of this Recommendation | International Standard.

The variable HighestTemporalId that identifies the highest sub-layer is specified by external means not specified in this Specification, or, when not specified by external means, HighestTemporalId is set to vps\_max\_sub\_layers\_minus1.

Each picture referred to in this clause is a complete coded picture.

Depending on the value of separate\_colour\_plane\_flag, the decoding process is structured as follows.

– If separate\_colour\_plane\_flag is equal to 0, the decoding process is invoked a single time with the current picture being the output.

– Otherwise (separate\_colour\_plane\_flag is equal to 1), the decoding process is invoked three times. Inputs to the decoding process are all NAL units of the coded picture with identical value of colour\_plane\_id. The decoding process of NAL units with a particular value of colour\_plane\_id is specified as if only a coded video sequence with monochrome colour format with that particular value of colour\_plane\_id would be present in the bitstream. The output of each of the three decoding processes is assigned to the 3 sample arrays of the current picture with the NAL units with colour\_plane\_id equal to 0 being assigned to SL, the NAL units with colour\_plane\_id equal to 1 being assigned to SCb, and the NAL units with colour\_plane\_id equal to 2 being assigned to SCr.

NOTE – The variable ChromaArrayType is derived as 0 when separate\_colour\_plane\_flag is equal to 1 and chroma\_format\_idc is equal to 3. In the decoding process, the value of this variable is evaluated resulting in operations identical to that of monochrome pictures with chroma\_format\_idc being equal to 0.

The decoding process operates as follows:

1. The decoding of NAL units is specified in subclause 8.2.
2. The processes in subclause 8.3 specify decoding processes using syntax elements in the slice layer and above:

– Variables and functions relating to picture order count are derived in subclause 8.3.1 (only needed to be invoked for one slice of a picture).

– The decoding process for reference picture set in subclause 8.3.2 is invoked, wherein reference pictures may be marked as "unused for reference" (only needed to be invoked for one slice of a picture).

– PicOutputFlag is set as follows:

– If the first coded picture in the bitstream is a CRA picture, and the current picture is a TFD picture associated with the CRA picture, or if the previous RAP picture preceding the current picture in decoding order is a BLA picture and the current picture is a TFD picture associated with the BLA picture, PicOutputFlag is set equal to 0 and the decoding process for generating unavailable reference pictures specified in subclause 8.3.3 is invoked (only needed to be invoked for one slice of a picture).

– Otherwise, PicOutputFlag is set equal to pic\_output\_flag.

– At the beginning of the decoding process for each P or B slice, the decoding process for reference picture lists construction specified in subclause 8.3.4 is invoked for derivation of reference picture list 0 (RefPicList0), and when decoding a B slice, reference picture list 1 (RefPicList1).

– After all slices of the current picture have been decoded, if it is a reference picture the decoded reference picture is marked as "used for short-term reference", otherwise (it is a non-reference picture) the decoded picture is marked as "unused for reference", as specified in subclause 8.3.5.

1. The processes in subclauses 8.4, 8.5, 8.6, and 8.7 specify decoding processes using syntax elements in the coding tree unit layer and above.