



HIGH-LEVEL SYNTAX FOR BITSTREAM EXTRACTION

JCTVC-G607

RICKARD SJÖBERG, THOMAS RUSERT
ERICSSON AB

MOTIVATION

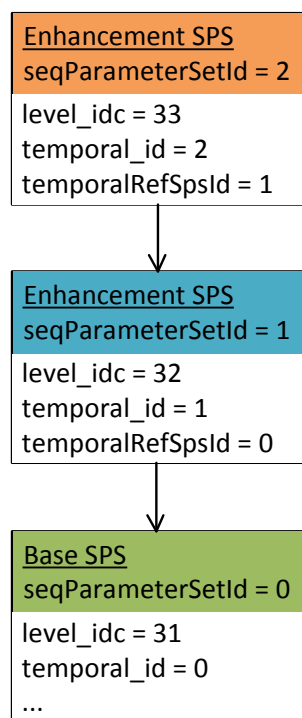
- › Bit stream scalability requires NAL units to convey information about the scalable layers they are associated with
- › SVC/MVC solution
 - Several different IDs carried in NAL unit headers (temporal_id, dependency_id, quality_id, priority_id, view_id)
 - Considering that several scalability extensions may be introduced in HEVC, this approach may lead to long NAL unit headers
 - Prefix NAL units and extended NAL unit headers and should be avoided for HEVC.
- › This contribution
 - Proposes a clean and extensible NAL unit header in the HEVC base specification for future extensions
 - No NAL unit header changes are needed when new scalability mechanisms are introduced
 - › Instead, new scalability mechanisms can be introduced in the SPS where they can be made dependent on profile_idc

PROPOSED BASIC DESIGN

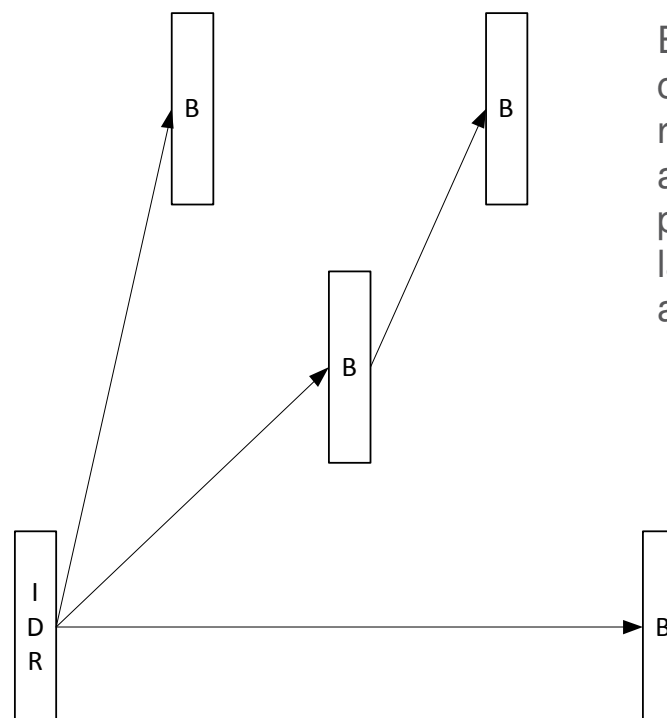
1. Signaling of SPS identifier seqParameterSetId in the NAL unit header
 - › Associate both VCL and non-VCL NAL units with scalable layers
 - › The seqParameterSetId serves as a priority identifier for simple bitstream thinning
 - › When scalability is not used, the NAL unit header can be reduced to 1 byte for most NAL units

2. One (extended) SPS per scalability layer, conveying
 - › Layer identifiers (currently only temporal_id)
 - › Decoder capabilities for each temporal layer can be extracted from the bitstream. Each layer have its own level_id
 - › Dependency information towards respective base layer (expressed by a link to a lower layer SPS)

PROPOSED BASIC DESIGN - EXAMPLE



(a)



(b)

Each enhancement SPS contains a reference to its respective reference SPS, as well as information that is particular for the respective layer, such as temporal_id and level_idc.



(c)

PROPOSED NAL UNIT HEADER SYNTAX

- › seqParameterSetId
 - introduced as a generic layer identifier and priority indicator in the NAL unit header.
 - The presence of it is optional, indicated by a u(1) flag.
 - If present, seqParameterSetId is coded using a byte-aligned variable length code, to cover future scalability extensions
 - If not present, there are no layers in the bitstream
 - › Note that seqParameterSetId must be present for SPS and PPS NAL units
 - ›
- › temporal_id
 - The temporal_id has been moved to the sequence parameter set
- › output_flag
 - The output_flag has been moved to the slice header

PROPOSED NAL UNIT HEADER SYNTAX

nal_unit(NumBytesInNALunit) {	Descriptor
forbidden_zero_bit	f(1)
<u>nal_ref_flag</u>	<u>u(1)</u>
nal_unit_type	u(5)
<u>seq_parameter_set_id_present_flag</u>	<u>u(1)</u>
nalUnitHeaderBytes = 1	
<u>if(seq_parameter_set_id_present_flag == 1) {</u>	
<u>seqParameterSetId = 0</u>	
<u>do {</u>	
<u>sps_id_7bits</u>	<u>u(7)</u>
<u>seq_parameter_set_id_extension_flag</u>	<u>u(1)</u>
<u>seqParameterSetId = seqParameterSetId * 128 + sps_id_7bits</u>	
<u>nalUnitHeaderBytes += 1</u>	
<u>} while(seq_parameter_set_id_extension_flag == 1)</u>	
<u>}</u>	
if(nal_unit_type == 1 nal_unit_type == 4 nal_unit_type == 5) {	
temporal_id	u(3)
output_flag	u(1)
reserved_one_4bits	u(4)
nalUnitHeaderBytes += 1	
}	

- › temporal_id moved to SPS
- › output_flag moved to slice header
- › 1 byte NAL unit header possible for non-scalable streams
 - SPS and PPS NAL units must have seq_parameter_set_id_present_flag equal to 1
 - Coded slice NAL units reference the PPS in the slice header

PROPOSED SPS SYNTAX

- › Mode one – “base SPS”
 - The base SPS contains essentially the same syntax elements as the current SPS
 - No seq_parameter_set_id is needed, the seqParameterSetId from the NAL unit header is used instead
 - The base SPS is associated with the base layer of the coded sequence

- › Mode two – “Enhancement SPS”
 - A flag temporal_dependency_flag indicates the “enhancement SPS” mode
 - Future flags for other types of scalability may be defined in the future
 - Each enhancement SPS has a reference SPS
 - The values level_idc and temporal_id in the reference SPS are overridden by the values present in the enhancement SPS
 - The enhancement SPS is associated with enhancement layers of the coded sequence

- › profile_idc
 - The profile_idc is currently kept the same for all layers in a sequence
 - This means that no changes to the SPS activation is needed, it is always the base SPS that is active
 - In the future, different profile_id values should be supported for e.g. view scalability

PROPOSED SPS SYNTAX

seq_parameter_set_rbsp() {	Descriptor
profile_idc	u(8)
reserved_zero_8bits /* equal to 0 */	u(8)
level_idc	u(8)
<u>temporal_dependency_flag</u>	<u>u(1)</u>
if(<u>temporal_dependency_flag == 1</u>) {	
<u>temporalRefSpsId = 0</u>	
do {	
<u>ref_sps_id_7bits</u>	<u>u(7)</u>
<u>ref_sps_id_extension_flag</u>	<u>u(1)</u>
<u>temporalRefSpsId = temporalRefSpsId * 128 + ref_sps_id_7bits</u>	
} while(<u>ref_sps_id_extension_flag == 1</u>)	
<u>temporal_id</u>	<u>u(3)</u>
} else {	
<u>base_seq_parameter_set_rbsp()</u>	
}	
rbsp_trailing_bits()	
}	

> Enhancement SPS contains

- Profile
- Level
- temporalRefSpsId
- temporal_id

base_seq_parameter_set_rbsp() {	Descriptor
profile_idc	u(8)
reserved_zero_8bits /* equal to 0 */	u(8)
level_idc	u(8)
seq_parameter_set_id	ue(v)
max_temporal_layers_minus1	u(3)
pic_width_in_luma_samples	u(16)
pic_height_in_luma_samples	u(16)
...	
rbsp_trailing_bits()	
}	

> Base SPS contains

- All usual SPS elements
- No seq_parameter_set_id, it is moved to the NAL unit header

PROPOSED PPS AND SLICE HEADER SYNTAX

pic_parameter_set_rbsp() {	Descriptor
pic_parameter_set_id	ue(v)
seq_parameter_set_id	ue(v)
...	
}	

slice_header() {	Descriptor
lightweight_slice_flag	u(1)
if(!lightweight_slice_flag) {	
<u>output_flag</u>	<u>u(1)</u>
slice_type	ue(v)
pic_parameter_set_id	ue(v)
...	

BITSTREAM EXTRACTION

- › There is an SPS for each scalable layer of the bitstream
 - “base SPS” for the base layer
 - “extended SPS” for the enhancement layers

- › To extract/forward a sub-stream
 - By parsing available SPSs a decoder or network node can understand layer properties and dependencies
 - Then an SPS that suits the requirements is selected
 - The respective reference SPSs are followed until the base SPS is reached, creating a list of seqParameterSetIds
 - The NAL units with seqParameterSetId present in the list are extracted/forwarded

- › Note that seqParameterSetId serves as a priority identifier for simple bitstream thinning

SUMMARY

› Proposal

- Use seqParameterSetId as a generic layer indicator in the NAL unit header
- Have one SPS per scalable layer convey layer-specific information
- Use “links” included in enhancement layer SPSs to indicate dependencies between scalability layers

› Advantages

- Extensibility for different types of scalability
- No future changes to NAL unit header is needed when scalability types are added
- Enables simple association of non-VCL NAL units to layers for extraction
- Only one byte NAL unit header needed for non-scalable streams
- Enables different level_idc per temporal layer



ERICSSON

SPS ACTIVATION

- › No changes are proposed for PPS activation.
 - The sequence parameter set identifier is conveyed in the NAL unit header (seqParameterSetId) instead of the PPS syntax structure
- › SPS activation is triggered when an SPS is referred to by activation of a PPS (or referred to by a buffering period SEI message)
 - If the referred SPS is a base SPS, then it is activated as usual.
 - If the referred SPS is an enhancement SPS, then the associated base SPS is activated instead.
- › At this time we only activate base SPSes
 - Note that we currently only carry profile_id, level_idc and temporal_id in the enhancement SPS
 - We currently require profile_id to be the same in all temporal layers
 - Note that in future scalability extensions to HEVC, it may be beneficial to allow that multiple SPSs are active at the same time, such as one SPS for each scalable layer

TEXT CHANGES FOR SPS ACTIVATION

A sequence parameter set RBSP includes parameters that can be referred to by one or more picture parameter set RBSPs or one or more SEI NAL units containing a buffering period SEI message. Each sequence parameter set RBSP is initially considered not active at the start of the operation of the decoding process. At most one sequence parameter set RBSP is considered active at any given moment during the operation of the decoding process, and the activation of any particular sequence parameter set RBSP results in the deactivation of the previously-active sequence parameter set RBSP (if any).

When a base sequence parameter set RBSP (with a particular value of [seqParameterSetId](#)) is not already active and it is referred to by activation of a picture parameter set RBSP (using that value of [seqParameterSetId](#)) or is referred to by activation of a picture parameter set RBSP using a value of [seqParameterSetId](#) that refers to an enhancement sequence parameter set RBSP that has an associated base sequence parameter set RBSP with that value of [seqParameterSetId](#) or is referred to by an SEI NAL unit containing a buffering period SEI message (using that value of [seqParameterSetId](#)), it is activated. This base sequence parameter set RBSP is called the active sequence parameter set RBSP until it is deactivated by the activation of another base sequence parameter set RBSP. A base sequence parameter set RBSP, with that particular value of [seqParameterSetId](#), shall be available to the decoding process prior to its activation. An activated base sequence parameter set RBSP shall remain active for the entire coded video sequence.