

Syntax on Parallel Processing Information Signalling (JCTVC-H0293)

**Yungo Park
Seungsoo Jeong
JaeHyun Kim
JeongHoon Park
(Samsung)**

Background

- ❖ HEVC has **many tools** to support **parallel processing** for multi-core environments → Tiles, WPP, Slice, Entropy Slice
- ❖ Multi-core system is currently being widely used in many CE devices
 - Parallel processing for video, especially in high resolution, is getting more important
- ❖ **Early knowing** the **parallel processing information** in SPS level would be helpful for **the decoder** to **prepare its multi-core decoding strategy**
 - After SPS parsing, generally the decoder starts initialization process
 - During the initialization, the decoder allocates CPU for decoding processing
- ❖ In HEVC, the hints for parallel processing is **dispersed** in SPS, PPS and Slice Header

	Slice	Entropy Slice	Tile	Wavefront
SPS	×	×	○	×
PPS	×	×	○	○
Slice Header	○	○	—	—

Proposed

- ❖ Signal one bit flag to notify whether the bitstream has a possibility for parallel processing or not
 - Under the condition, all existing syntax (Tile, WPP) are put together
- ❖ Signal the syntax whether the bitstream has pictures which are divided to slices or not
 - The strategy for parallel processing could be different in case of slices and entropy slices
 - Even though tiles and WPP can support parallel processing, it's curious that all devices will use these.
 - Slices are still good tools for parallel processing
- ❖ All parallel processing information (or picture partition) could come together and could be signaled at one time

	Slice	Entropy Slice	Tile	Wavefront
SPS	○	○	○	○
PPS	○	○	○	○
Slice Header	○	○	—	—

Proposed Syntax

Syntax for SPS

seq_parameter_set_rbsp() {	Descriptor
...	
parallel_processing_param_enabled_sps_flag	u(1)
if(parallel_processing_param_enabled_sps_flag)	
parallel_processing_param()	
...	
}	

Syntax for PPS

pic_parameter_set_rbsp() {	Descriptor
...	
if(parallel_processing_param_enabled_sps_flag) {	
parallel_processing_param_enabled_pps_flag	u(1)
if(parallel_processing_param_enabled_pps_flag)	
parallel_processing_param()	
}	
...	
}	

Syntax for parallel_processing_param()

parallel_processing_param() {	Descriptor
tile_enabled_flag	u(1)
if(tile_enabled_flag) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
tile_boundary_independence_flag	u(1)
uniform_spacing_flag	u(1)
if(!uniform_spacing_flag) {	
for(i = 0; i < num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
}	
entropy_coding_synchro	u(v)
if(entropy_coding_synchro)	
num_substreams_minus1	ue(v)
slice_enabled_flag	u(1)
if(slice_enabled_flag)	
entropy_slice_enabled_flag	u(1)
}	

Newly Defined Syntax

Existing but Moved Syntax

Conclusion

- ❖ It is proposed that **one flag signalling** which notifies whether the bitstream has parallel processing (or picture partition) unit **in SPS(PPS)** or not
 - Early notification for parallel processing information
- ❖ It is proposed that the **pictures that have slices or entropy slices** can be early **notified in SPS and PPS**
 - **In AVC**, before parsing each slice header, it was **impossible** for the decoder to know whether this picture consists of slice or not
 - Early SPS or PPS level picture slicing information could be helpful for multi-core decoder
- ❖ It is proposed that **parallel processing related syntax come together in SPS and PPS**
 - More readable and syntax clean up

Supplement

- ❖ The possible use cases of slice
 - Packetization
 - GDR (Gradual Decoder Refresh)
 - Parallel processing

