

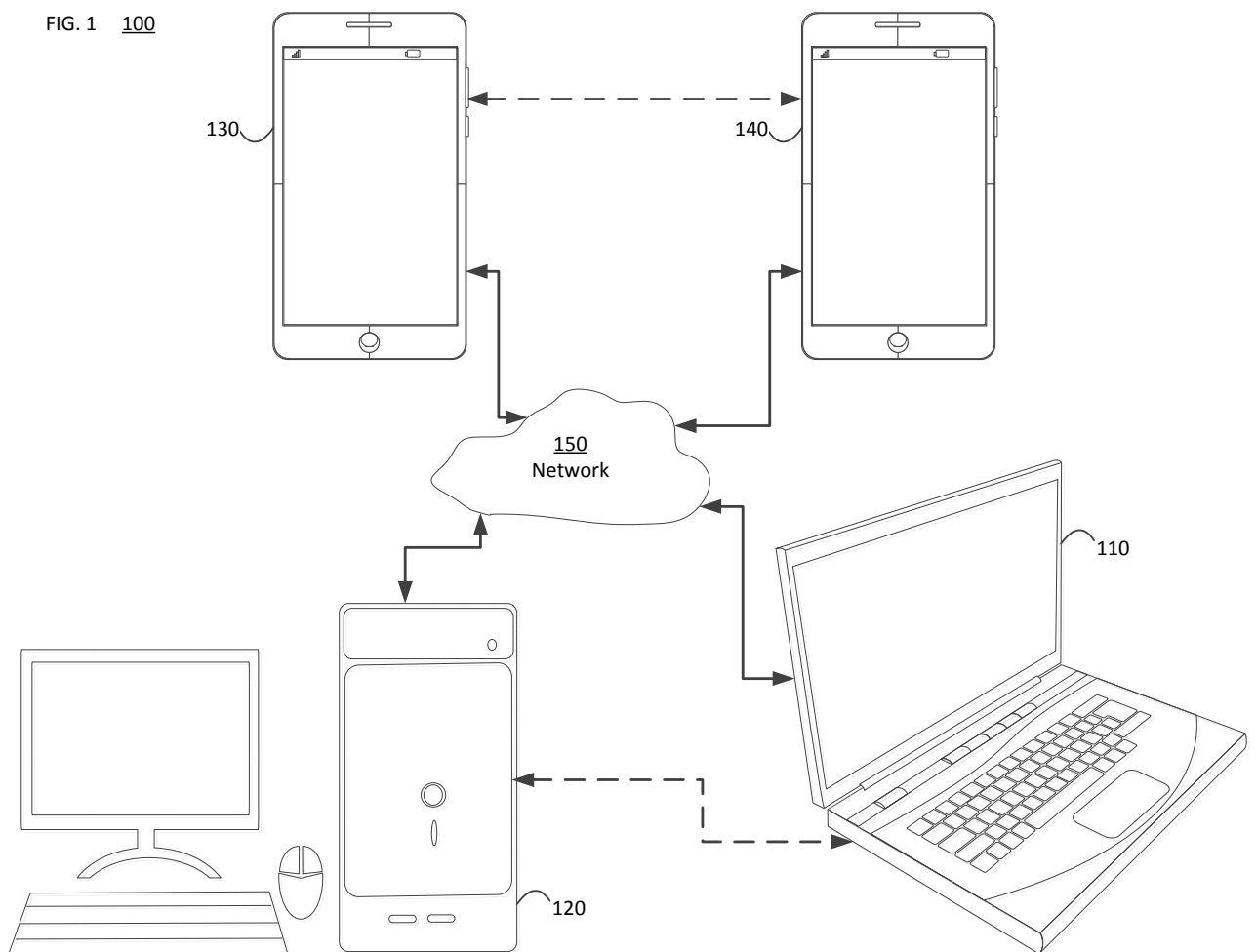
FIG. 1 100

FIG. 2 Streaming System 200

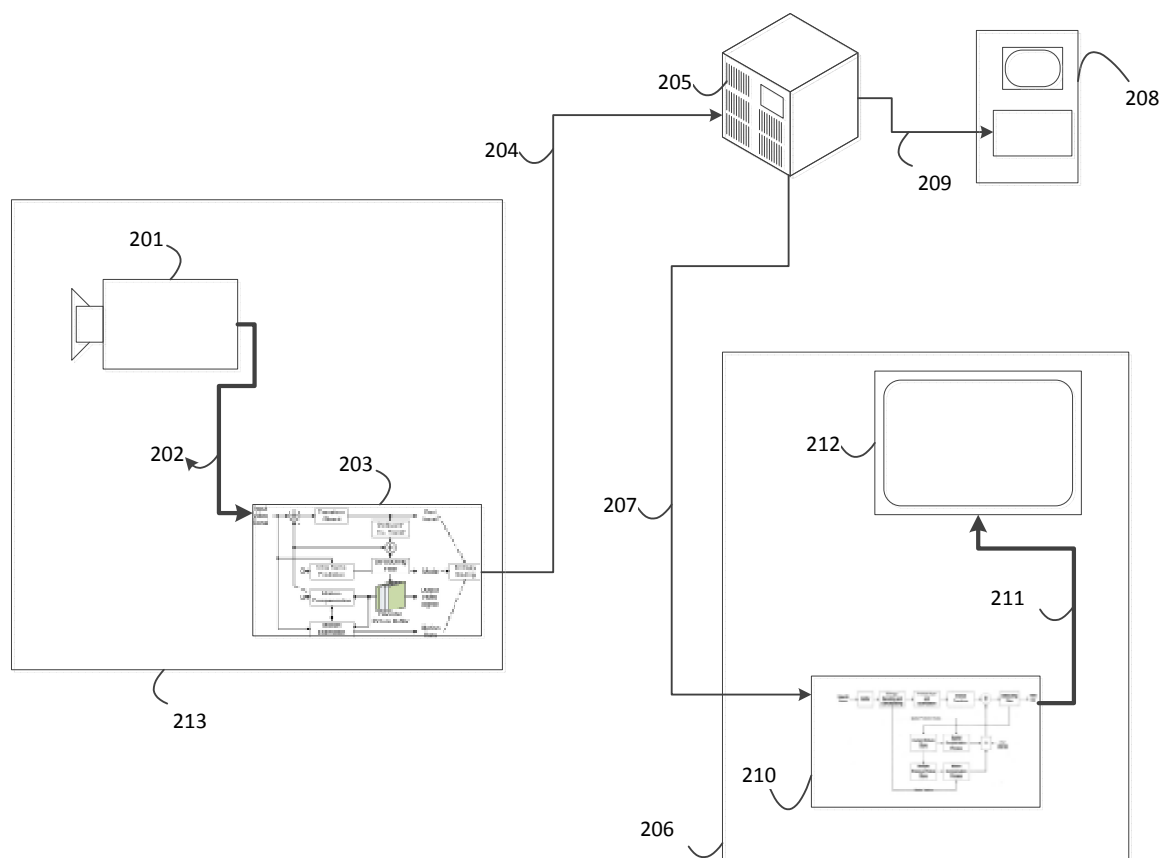


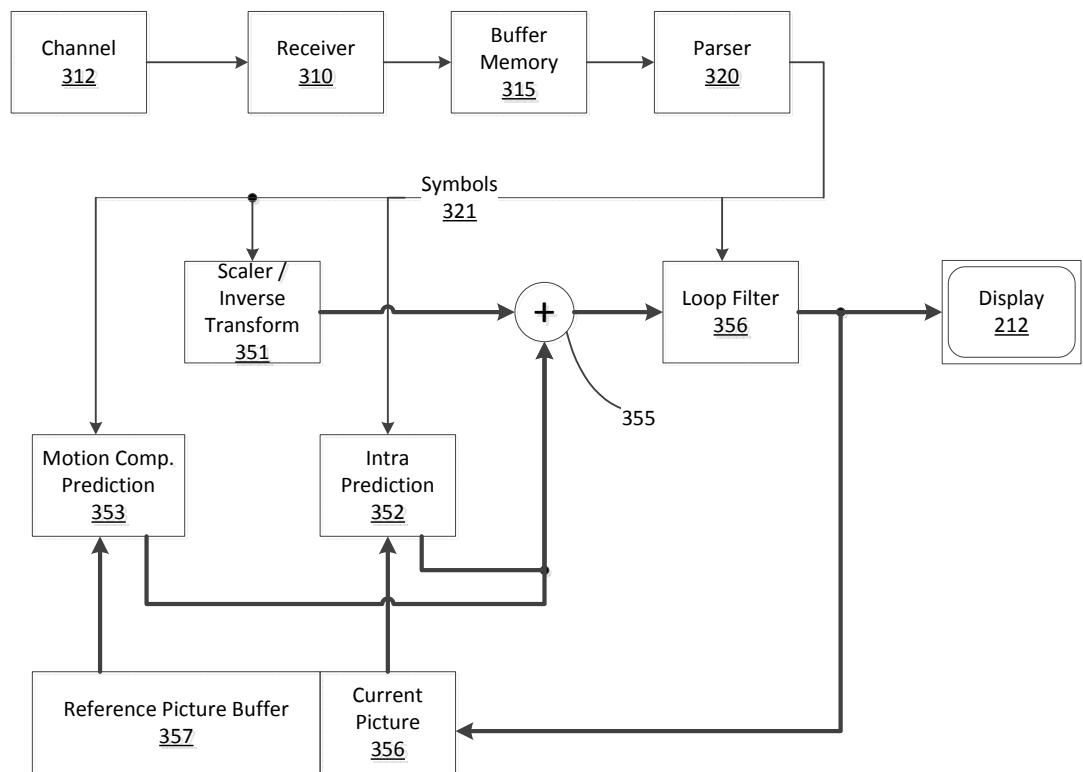
FIG. 3 Decoder 210

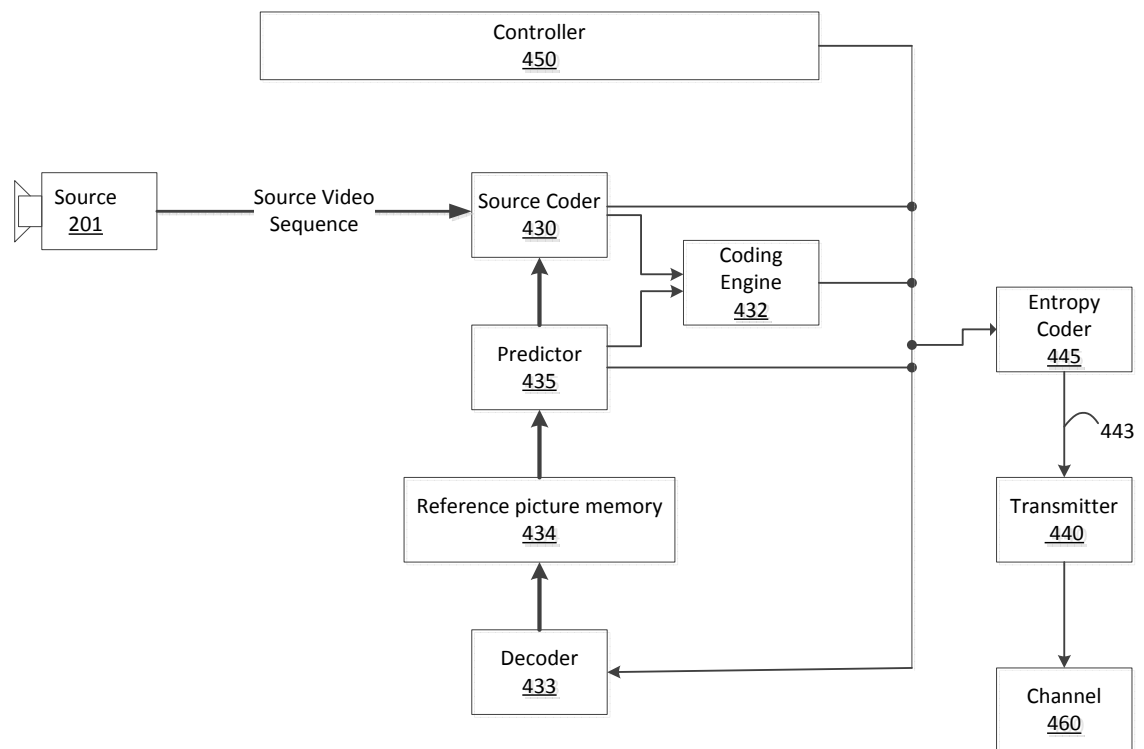
FIG. 4 Encoder 203

FIG. 5

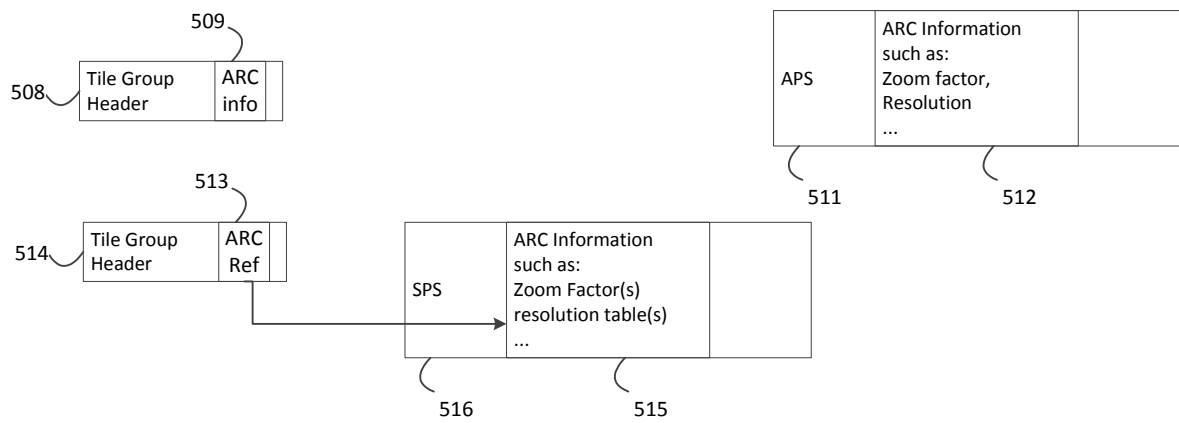
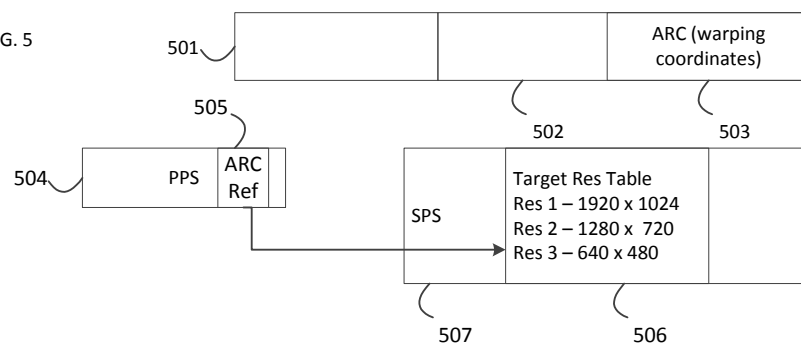


Fig. 6

601	tile_group_header() {	
	...	
603	if(adaptive_pic_resolution_change_flag) {	
602	dec_pic_size_idx	u(1)
	}	
	...	
	}	

610	seq_parameter_set_rbsp() {	Descriptor
	...	
611	adaptive_pic_resolution_change_flag	u(1)
612	if(adaptive_pic_resolution_change_flag) {	
613	output_pic_width_in_luma_samples	ue(v)
	output_pic_height_in_luma_samples	ue(v)
614	reference_pic_size_present_flag	u(1)
	if(reference_pic_size_present_flag)	
	{	
615	reference_pic_width_in_luma_samples	ue(v)
	reference_pic_height_in_luma_samples	ue(v)
	}	
616	num_dec_pic_size_in_luma_samples_minus1	ue(v)
	for(i = 0; i <= num_dec_pic_size_in_luma_samples_minus1; i++) {	
617	dec_pic_width_in_luma_samples[i]	ue(v)
	dec_pic_height_in_luma_samples[i]	ue(v)
	}	
	}	
	...	
	}	

FIG. 7
Computer System 700

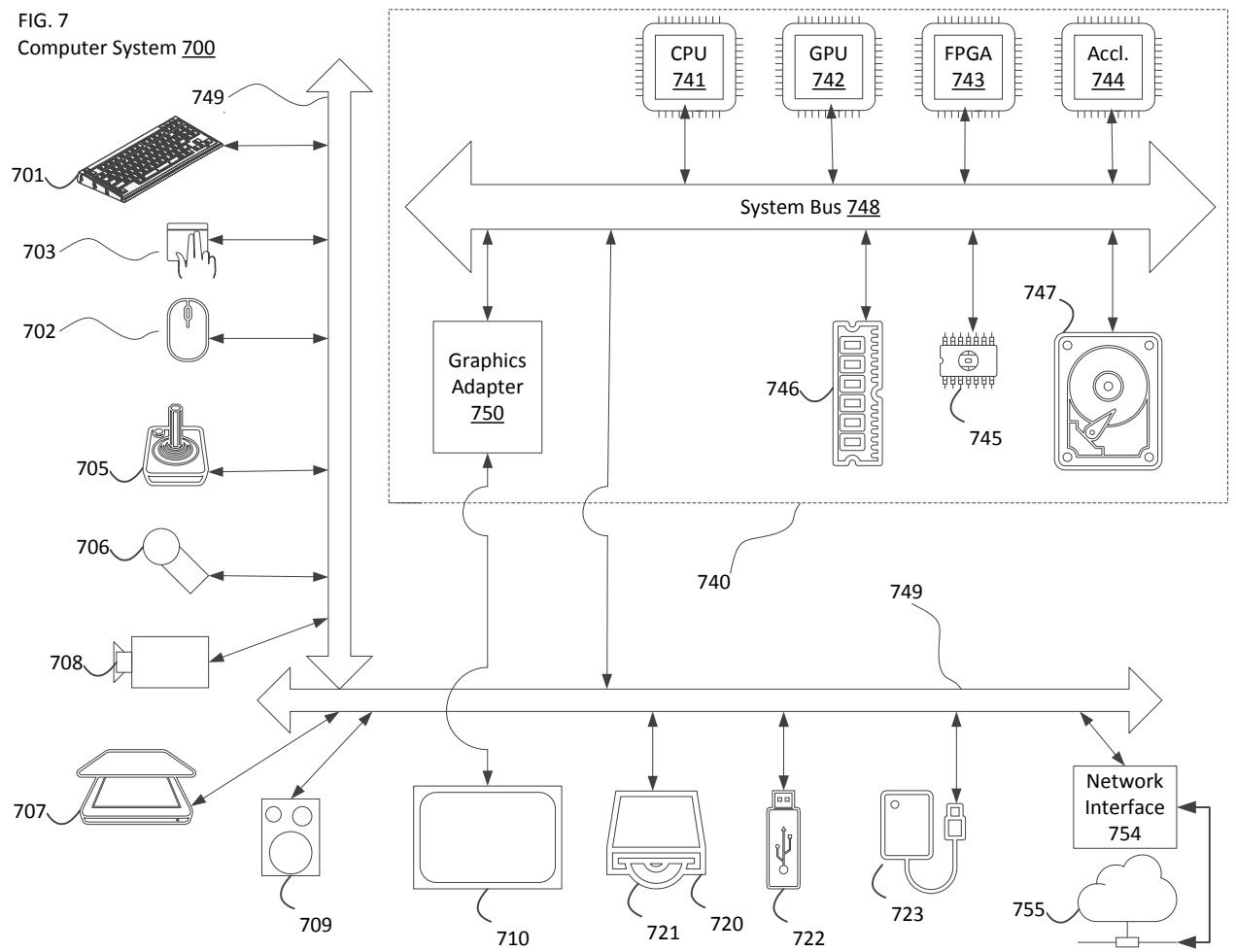


Fig. 8

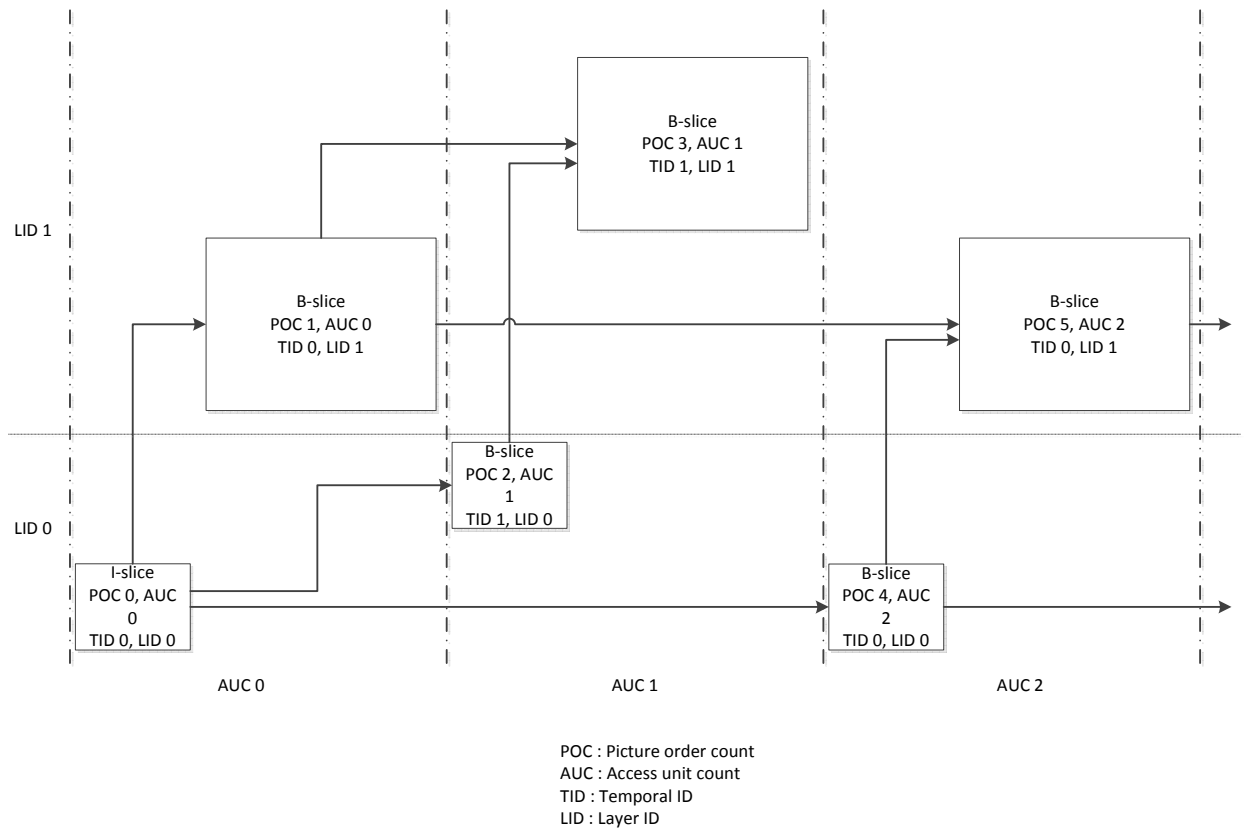


Fig. 9

<code>video_parameter_set_rbsp()</code> { ¹	Descriptor ¹
<code>vps_video_parameter_set_id</code> ¹	<code>u(4)</code> ¹
<code>vps_max_layers_minus1</code> ¹	<code>u(8)</code> ¹
<code>for(i = 0; i <= vps_max_layers_minus1; i++) {¹</code>	¹
<code> vps_included_layer_id[i]</code> ¹	<code>u(7)</code> ¹
<code> vps_reserved_zero_bit</code> ¹	<code>u(1)</code> ¹
<code>}</code> ¹	¹
<code>vps_constraint_info_present_flag</code> ¹	<code>u(1)</code> ¹
<code>vps_constant_poc_cycle_per_au</code> ¹	<code>u(1)</code> ¹
<code>if(vps_constant_poc_cycle_per_au)</code> ¹	¹
<code> vps_poc_cycle_au</code> ¹	<code>u(8)</code> ¹
<code>...¹</code>	¹
<code>}</code> ¹	¹
¹	
<code>slice_header()</code> { ¹	Descriptor ¹
<code>slice_pic_parameter_set_id</code> ¹	<code>ue(v)</code> ¹
<code>if(rect_slice_flag NumBricksInPic > 1)</code> ¹	¹
<code> slice_address</code> ¹	<code>u(v)</code> ¹
<code>if(!rect_slice_flag && !single_brick_per_slice_flag)</code> ¹	¹
<code> num_bricks_in_slice_minus1</code> ¹	<code>ue(v)</code> ¹
<code>slice_type</code> ¹	<code>ue(v)</code> ¹
<code>if(NalUnitType == GRA_NUT)</code> ¹	¹
<code> recovery_poc_cnt</code> ¹	<code>se(v)</code> ¹
<code>slice_pic_order_cnt_lsb</code> ¹	<code>u(v)</code> ¹
<code>...¹</code>	¹
<code>if(!vps_constant_poc_cycle_per_au)</code> ¹	¹
<code> slice_poc_cycle_au</code> ¹	<code>u(8)</code> ¹
<code> </code> ¹	¹
<code> </code> ¹	¹
<code>}</code> ¹	¹

Fig. 10

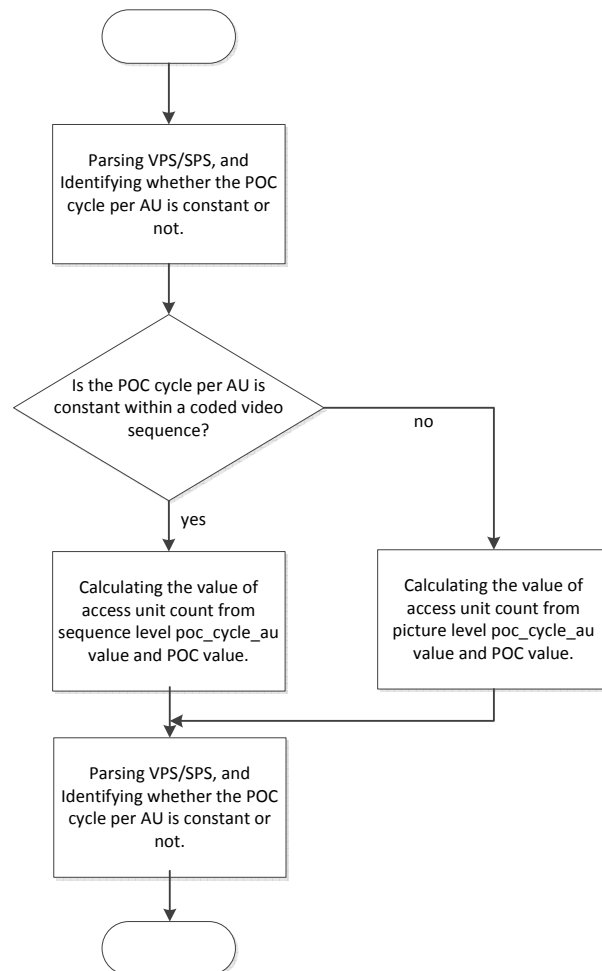


FIG. 11

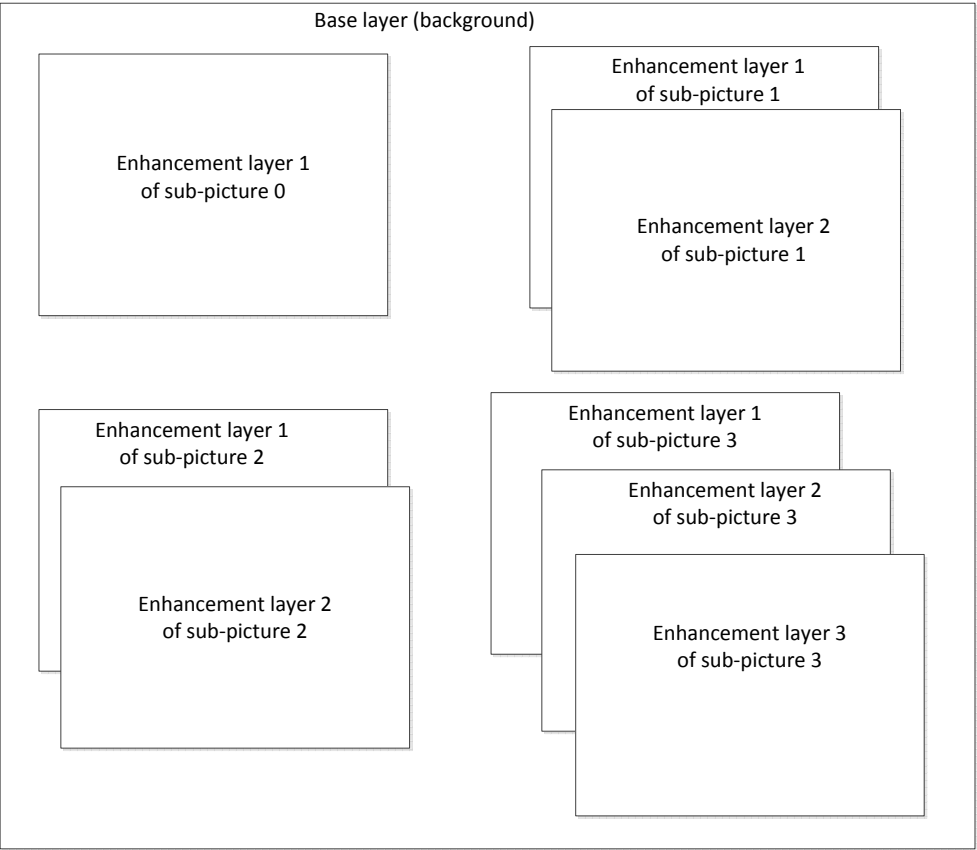


FIG. 12

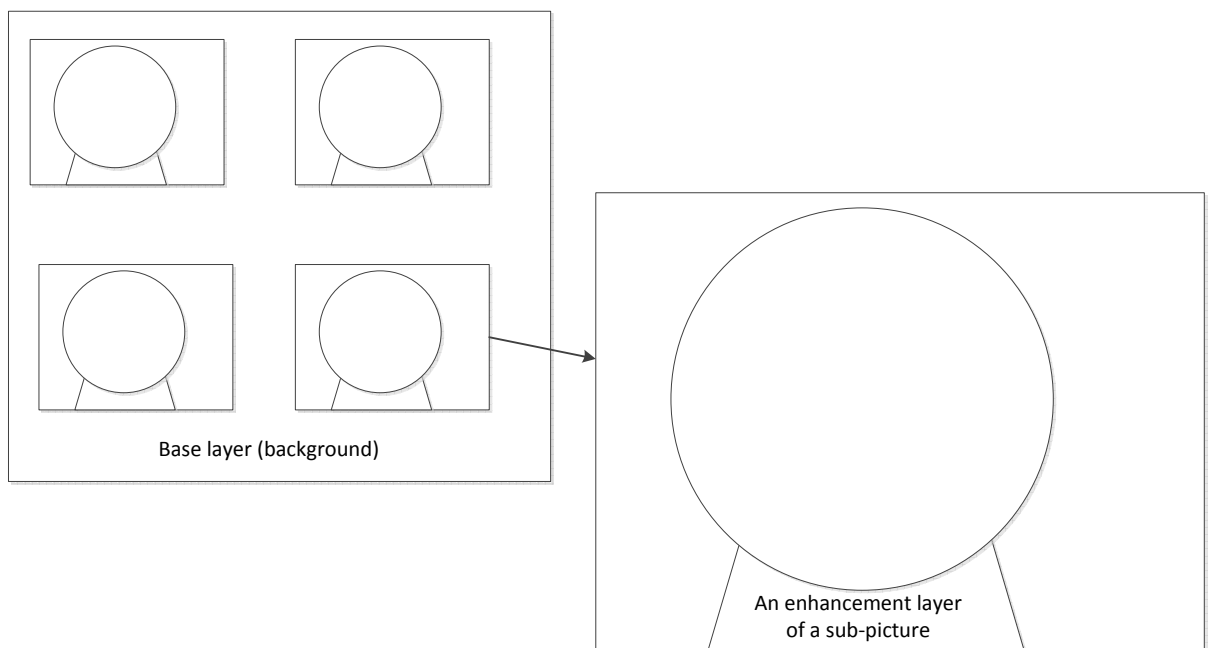


Fig. 13

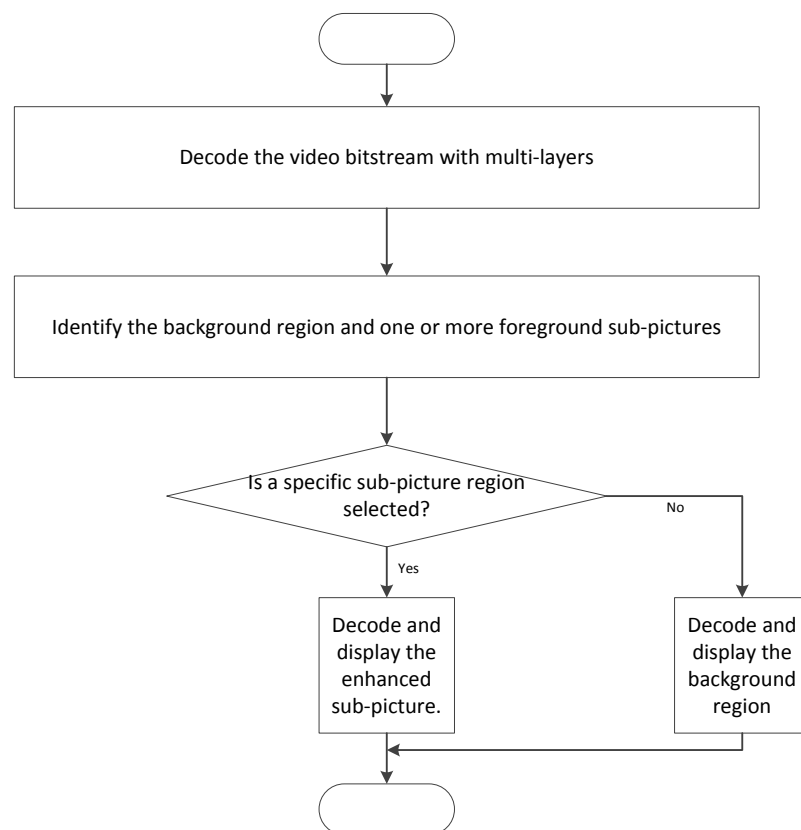


Fig. 14

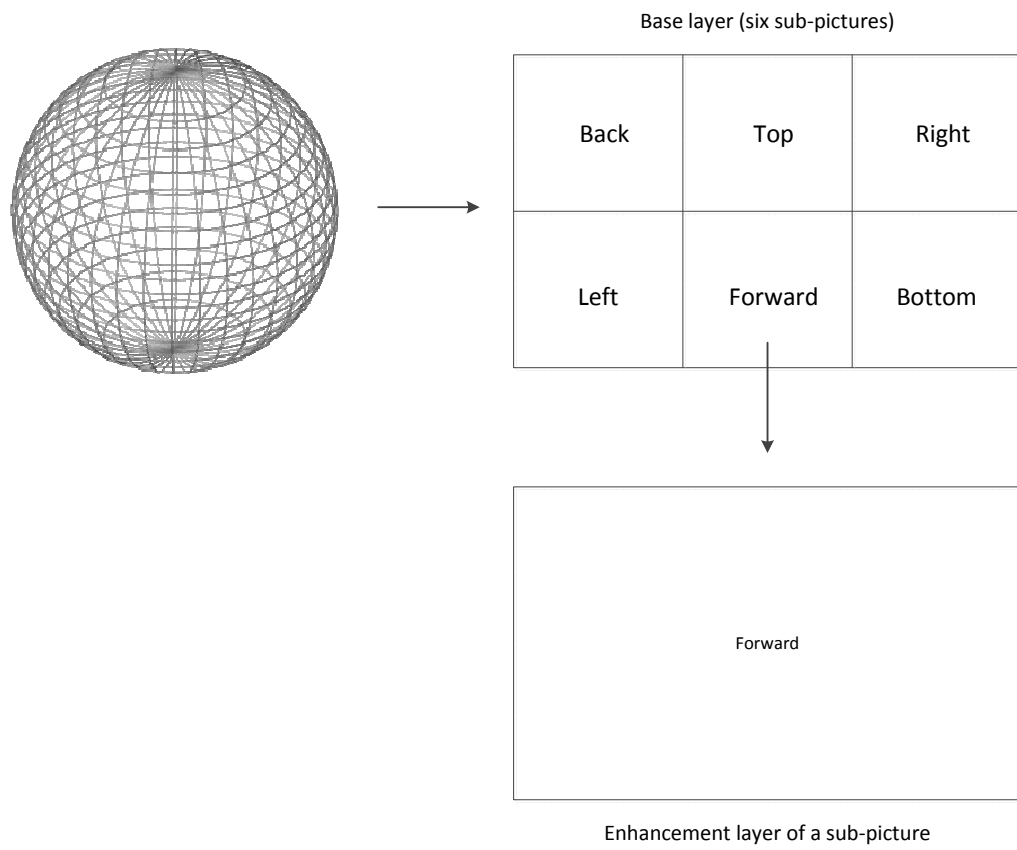


Fig. 15

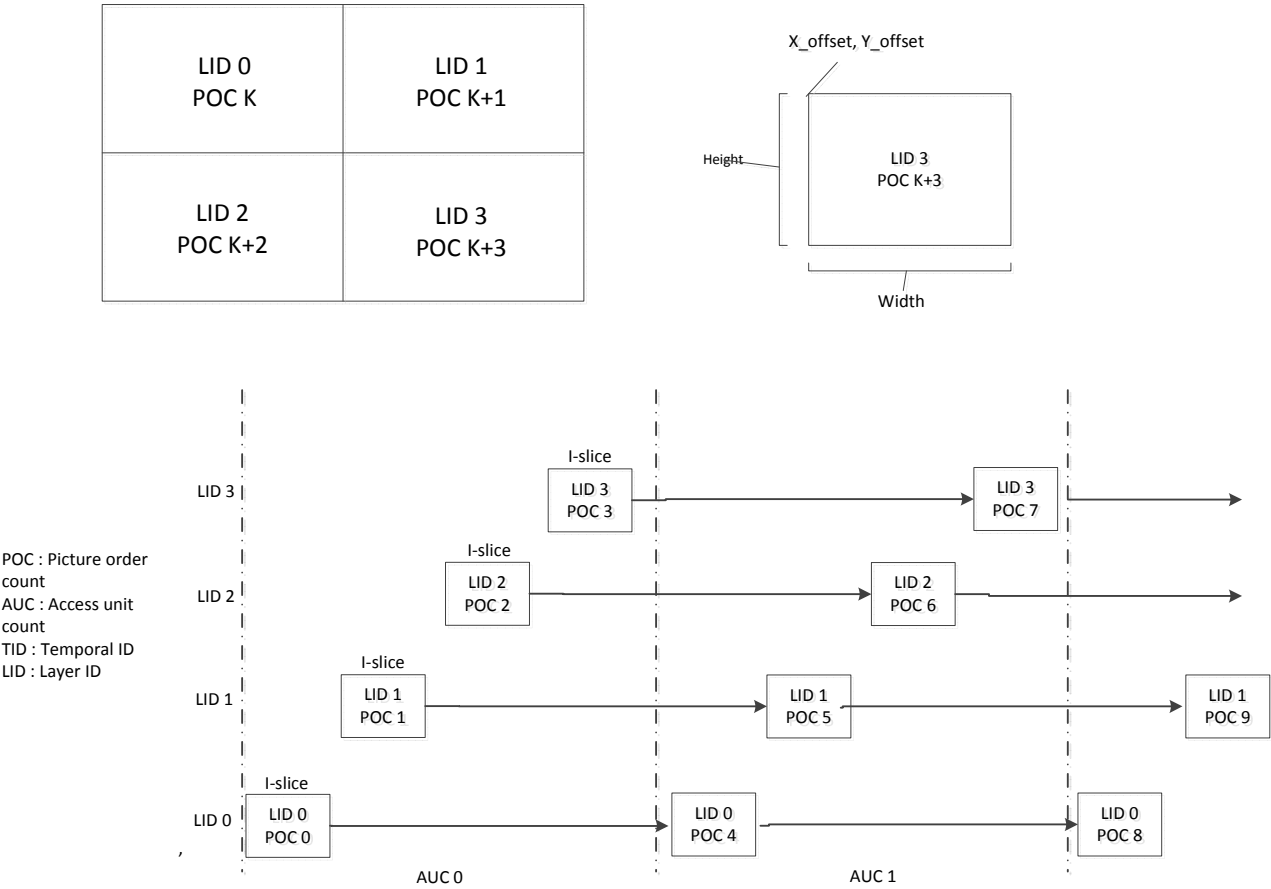


Fig. 16

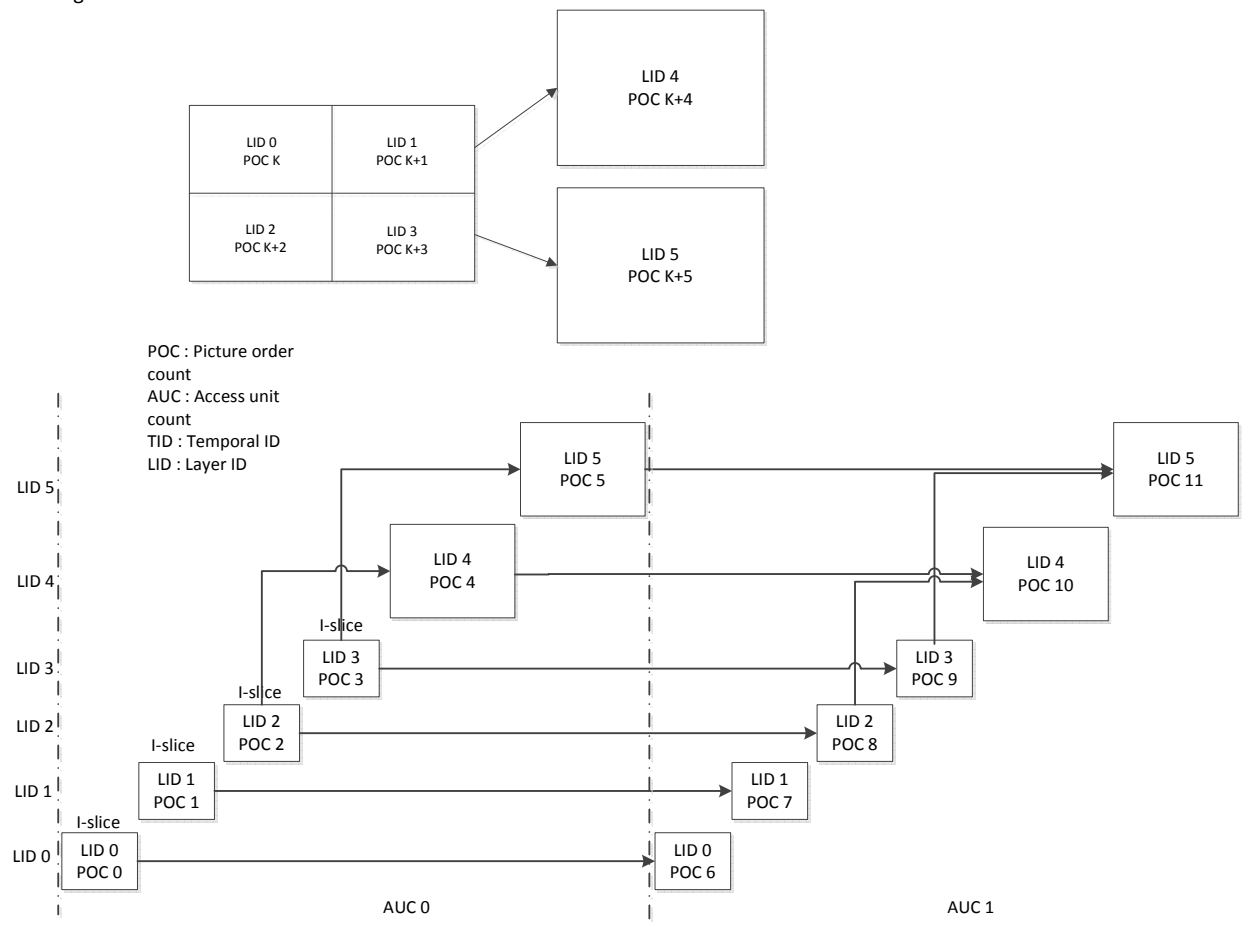


Fig. 17

video_parameter_set_rbsp() { ^o	Descriptor ^o
vps_video_parameter_set_id ^o	<u>u(4)</u> ^o
vps_max_layers_minus1 ^o	<u>u(8)</u> ^o
for(i = 0; i <= vps_max_layers_minus1; i++) { ^o	^o
vps_included_layer_id[i] ^o	<u>u(7)</u> ^o
vps_reserved_zero_bit ^o	<u>u(1)</u> ^o
} ^o	^o
... ^o	^o
vps_sub_picture_dividing_flag ^o	<u>u(1)</u> ^o
if(vps_sub_picture_dividing_flag) { ^o	^o
vps_full_pic_width_in_luma_samples ^o	<u>ue(v)</u> ^o
vps_full_pic_height_in_luma_samples ^o	<u>ue(v)</u> ^o
} ^o	^o
... ^o	^o
} ^o	^o
^o	
▪ seq_parameter_set_rbsp() { ^o	Descriptor ^o
sps_decoding_parameter_set_id ^o	<u>u(4)</u> ^o
sps_video_parameter_set_id ^o	<u>u(4)</u> ^o
sps_max_sub_layers_minus1 ^o	<u>u(3)</u> ^o
... ^o	^o
pic_width_in_luma_samples ^o	<u>ue(v)</u> ^o
pic_height_in_luma_samples ^o	<u>ue(v)</u> ^o
if(vps_sub_picture_dividing_flag) { ^o	^o
pic_offset_x ^o	<u>ue(v)</u> ^o
pic_offset_y ^o	<u>ue(v)</u> ^o
} ^o	^o
... ^o	^o
} ^o	^o

Fig. 18

<code>sub_region_partitioning_info(payloadSize) {</code>	Descriptor
<code>num_sub_region</code>	<code>ug(v)</code>
<code>num_layers</code>	<code>ug(v)</code>
<code>for(i = 0; i <= num_layers; i++)</code>	
<code>layer_id [i]</code>	<code>ug(v)</code>
<code>for(i = 1; i < num_layers; i++)</code>	
<code>for(i = 0; i < i; i++)</code>	
<code>direct_dependency_flag[i][j]</code>	<code>u(1)</code>
<code>for(i = 0; i < num_sub_region; i++) {</code>	
<code>num_layers_for_region[i]</code>	<code>ug(v)</code>
<code>for(i = 0; i < num_layers_for_region[i]; i++)</code>	
<code>sub_region_layer_id[i][j]</code>	<code>ug(v)</code>
<code>sub_region_offset_x[i]</code>	<code>ug(v)</code>
<code>sub_region_offset_y[i]</code>	<code>ug(v)</code>
<code>sub_region_width[i]</code>	<code>ug(v)</code>
<code>sub_region_height [i]</code>	<code>ug(v)</code>
<code>}</code>	
<code>...</code>	
<code>}</code>	

Fig. 19

video parameter set rbsp() {	Descriptor
...	
vps_max_layers_minus1	u(6)
num_output_layer_sets	ue(v)
num_profile_tier_level	ue(v)
for(i = 0; i < num_profile_tier_level; i++)	
profile_tier_level(vps_max_sub_layers_minus1)	
for(i = 0; i < num_output_layer_sets; i++)	
for(j = 0; j < NumLayersInIdList[i]; j++) {	
output_layer_flag[i][j]	u(1)
profile_tier_level_idx[i][j]	u(v)
}	
...	
}	

Fig. 20

video parameter set rbsp() {	Descriptor
...	
vps_max_layers_minus1	u(6)
if(vps_max_layers_minus1 > 0) {	
num_output_layer_sets	ue(v)
num_profile_tier_level	ue(v)
}	
for(i = 0; i < num_profile_tier_level; i++)	
profile_tier_level(vps_max_sub_layers_minus1)	
for(i = 0; i < num_output_layer_sets; i++) {	
vps_output_layers_mode[i]	u(2)
vps_ptl_signal_flag[i]	u(1)
for(j = 0; j < NumLayersInIdList[I]; j++) {	
if(vps_output_layers_mode[i] == 2)	
output_layer_flag[I][j]	u(1)
if(vps_ptl_signal_flag[i])	
profile_tier_level_idx[I][j]	u(v)
}	
}	
...	
}	

Fig. 21

video parameter set rbsp() {	Descriptor
...	
vps_max_layers_minus1	u(6)
if(vps_max_layers_minus1 > 0) {	
num_output_layer_sets	ue(v)
num_profile_tier_level	ue(v)
}	
max_subpics_minus1	u(8)
for(i = 0; i < max_subpics_minus1; i++) {	
sub_pic_id[i]	u(8)
}	
for(i = 0; i < num_profile_tier_level; i++)	
profile_tier_level(vps_max_sub_layers_minus1)	
for(i = 0; i < num_output_layer_sets; i++) {	
vps_output_layers_mode[i]	u(2)
vps_ptl_signal_flag[i]	u(1)
for(j = 0; j < NumLayersInIdList[i]; j++) {	
num_output_subpic_layer[i][j]	ue(v)
for(k = 0; k < num_output_subpic_layer[i][j]; k++)	
sub_pic_id_layer[i][j][k]	u(8)
if(vps_output_layers_mode[i] == 2)	
output_layer_flag[i][j]	u(1)
if(vps_ptl_signal_flag[i])	
profile_tier_level_idx[i][j]	u(v)
}	
}	
...	
}	

Fig. 22

pic parameter set rbsp() {	Descriptor
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	u(4)
mixed_nalu_types_in_pic_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pps_conformance_window_flag	u(1)
if pps_conformance_window_flag {	
pps_conf_win_left_offset	ue(v)
pps_conf_win_right_offset	ue(v)
pps_conf_win_top_offset	ue(v)
pps_conf_win_bottom_offset	ue(v)
}	
scaling_window_explicit_signalling_flag	u(1)
if scaling_window_explicit_signalling_flag {	
scaling_win_left_offset	ue(v)
scaling_win_right_offset	ue(v)
scaling_win_top_offset	ue(v)
scaling_win_bottom_offset	ue(v)
}	
output_flag_present_flag	u(1)
subpic_id_mapping_in_pps_flag	u(1)
if subpic_id_mapping_in_pps_flag {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpic_minus1; i++)	
pps_subpic_id[i]	u(v)
}	
...	
}	

Fig. 23

seq_parameter_set_rbsp() {	Descriptor
...	
subpic_info_present_flag	u(1)
if(subpic_info_present_flag) {	
sps_num_subpics_minus2	ue(v)
sps_independent_subpics_flag	u(1)
for(i = 0; i <= sps_num_subpics_minus2 + 1; i++) {	
if(i > 0 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_ctu_top_left_x[i]	u(v)
if(i > 0 && pic_height_max_in_luma_samples > CtbSizeY) {	
subpic_ctu_top_left_y[i]	u(v)
if(i < sps_num_subpics_minus2 + 1 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_width_minus1[i]	u(v)
if(i < sps_num_subpics_minus2 + 1 && pic_height_max_in_luma_samples > CtbSizeY)	
subpic_height_minus1[i]	u(v)
if(!sps_independent_subpics_flag) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
sps_subpic_id_len_minus1	ue(v)
subpic_id_mapping_explicitly_signalled_flag	u(1)
if(subpic_id_mapping_explicitly_signalled_flag) {	
subpic_id_mapping_in_sps_flag	u(1)
if(subpic_id_mapping_in_sps_flag)	
for(i = 0; i <= sps_num_subpics_minus2 + 1; i++)	
sps_subpic_id[i]	u(v)
}	
}	
...	
}	

Fig. 24

pic_parameter_set_rbsp() {	Descriptor
...	
mixed_nalu_types_in_pic_flag	u(1)
...	
subpic_id_mapping_in_pps_flag	u(1)
if(subpic_id_mapping_in_pps_flag) {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpics_minus1; i++)	
pps_subpic_id[i]	u(v)
}	
pps_independent_subpics_flag	u(1)
if(!pps_independent_subpics_flag) {	
if(!subpic_id_mapping_in_pps_flag)	
pps_num_subpics_minus1	ue(v)
for(i = 0; i <= pps_num_subpics_minus1; i++) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
...	
}	