

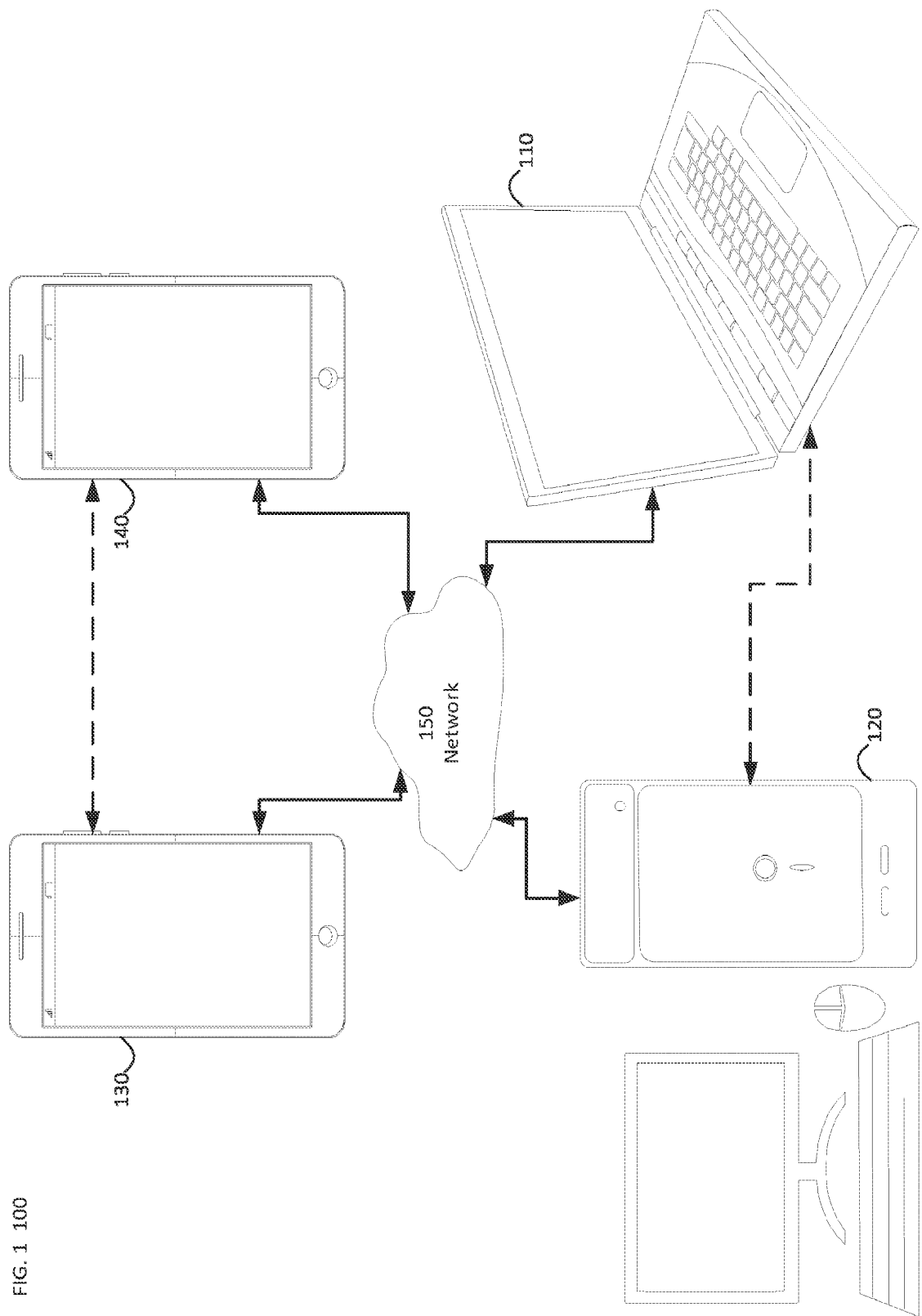


FIG. 1 100

FIG. 2

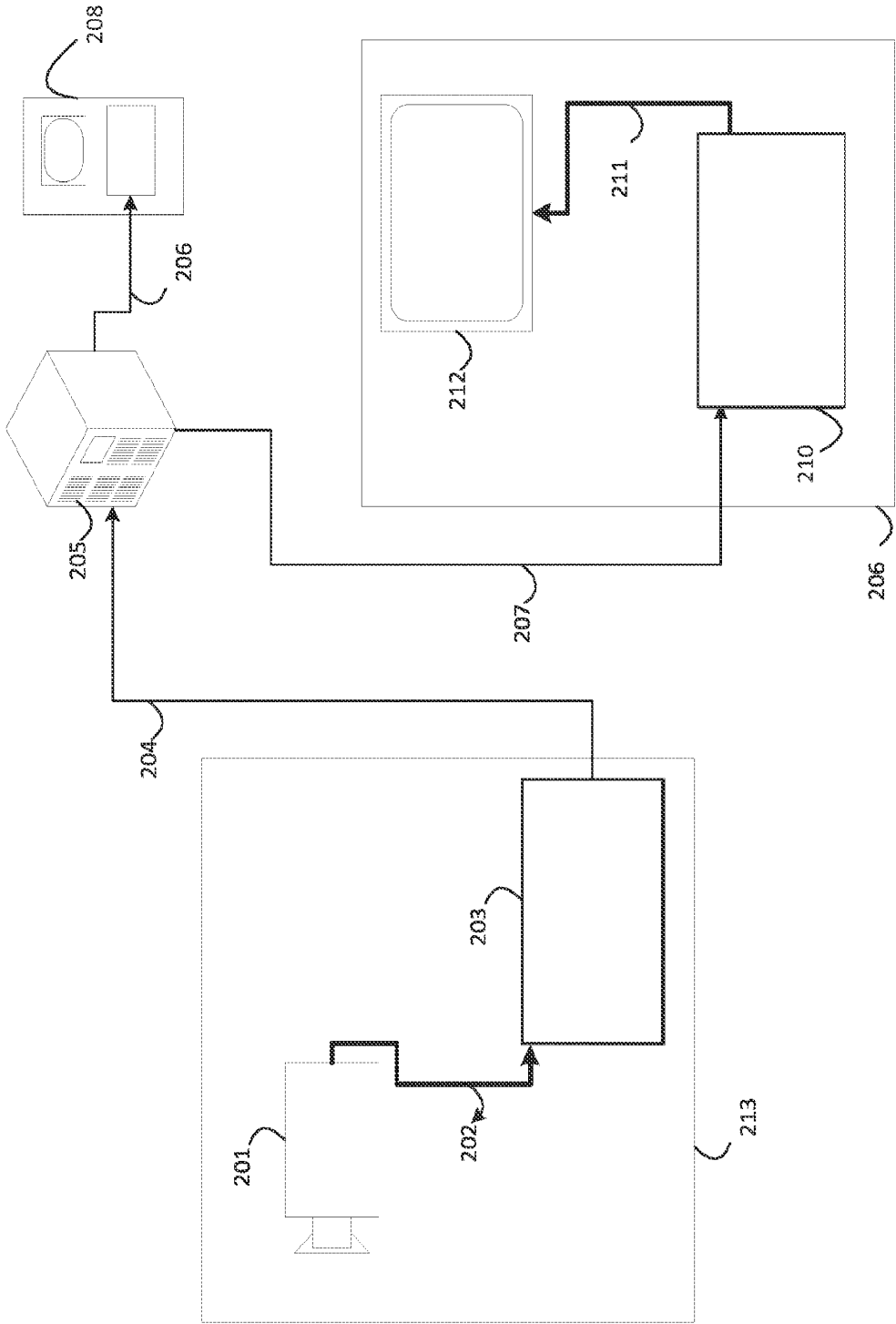


FIG. 3 Decoder 210

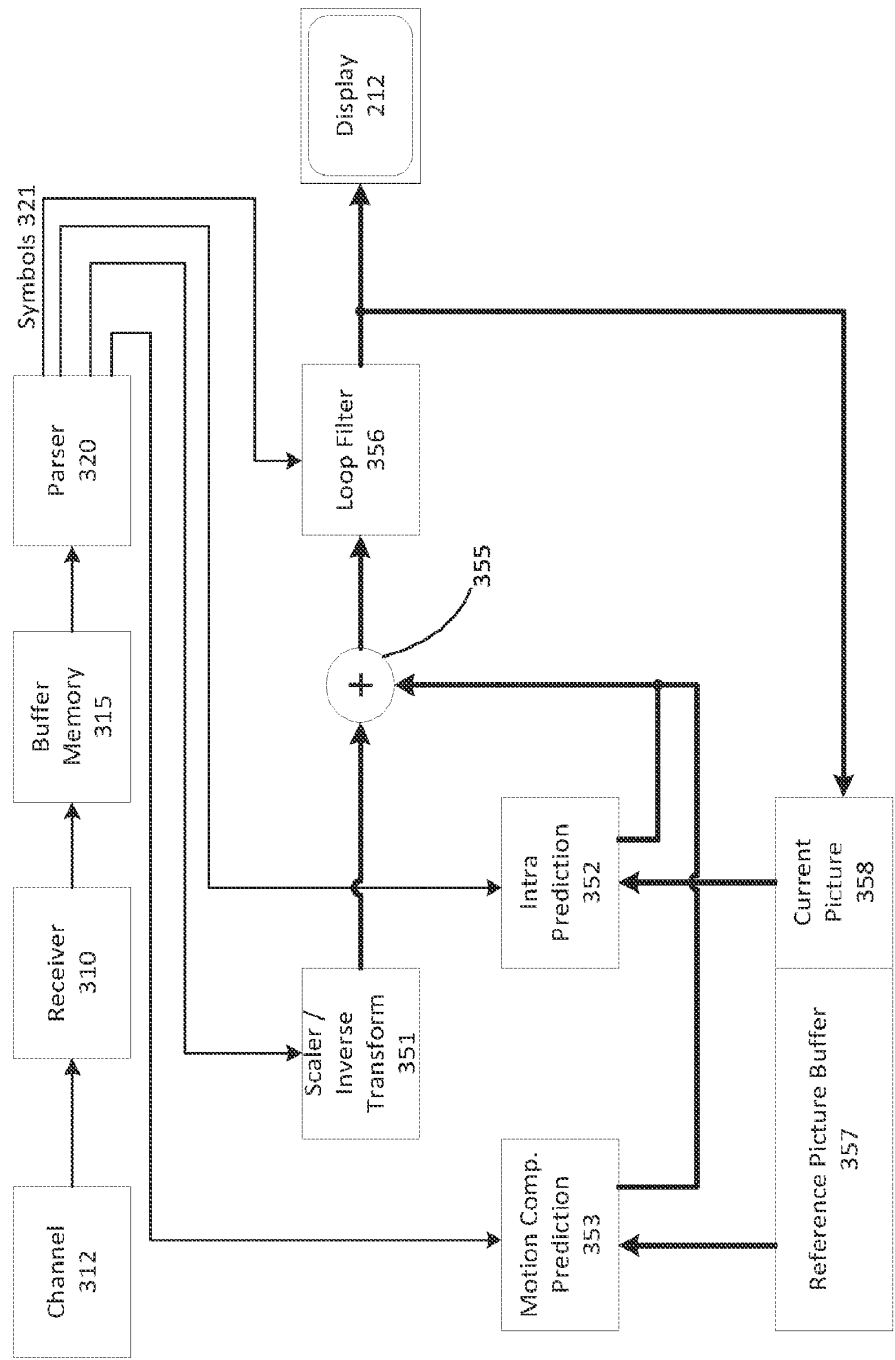
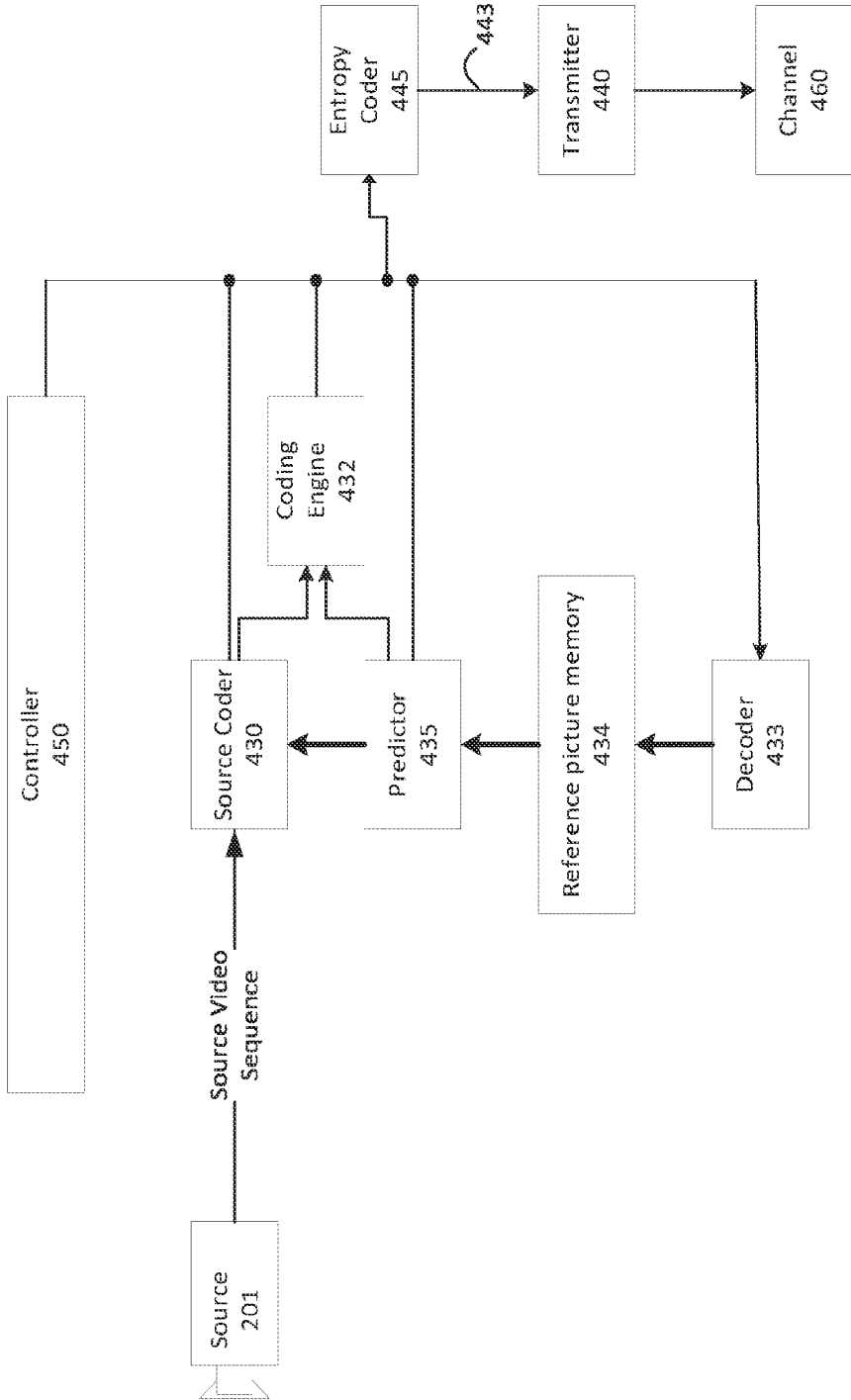


FIG. 4 Encoder 203



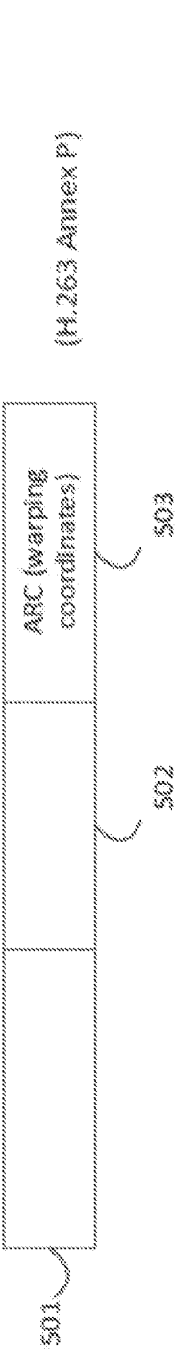


FIG. 5A

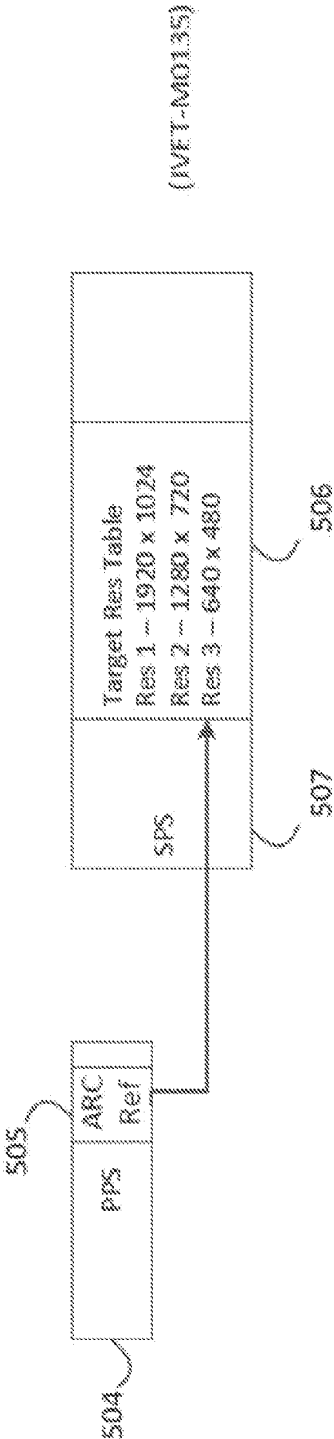


FIG. 5B

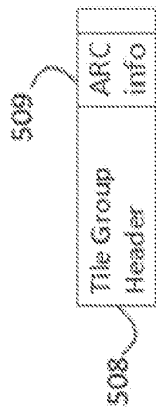


FIG. 5C



FIG. 5D

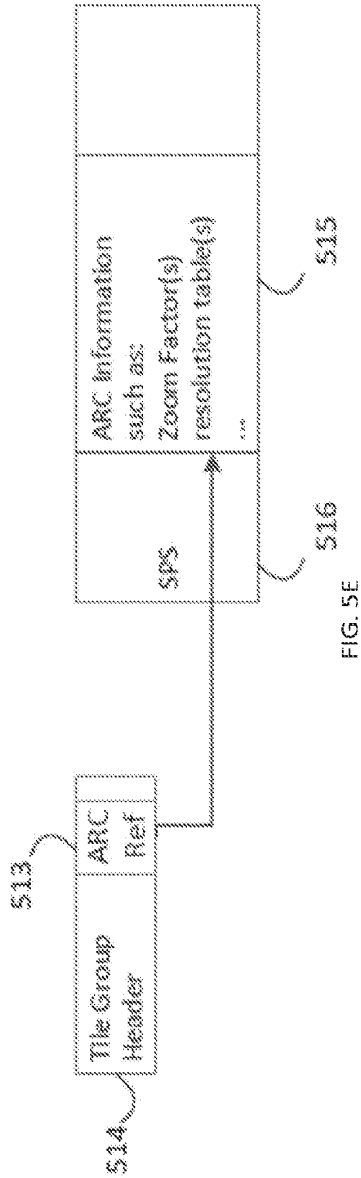


FIG. 5E

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

FIG. 6A

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. 6B

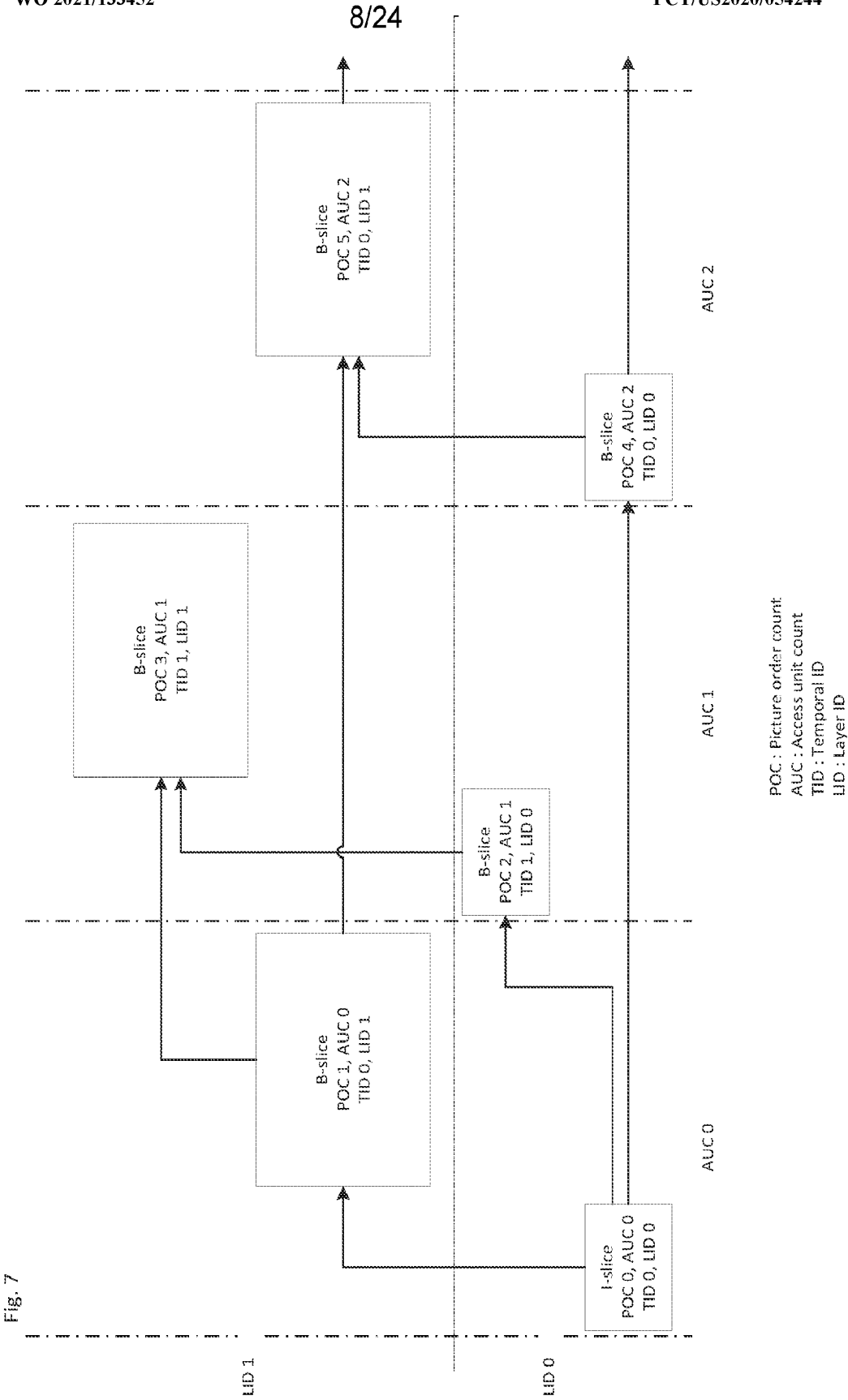


Fig. 8

<pre>video_parameters_get_hdr() { vps_video_parameter_set_id vps_max_layers_minus1 for (i=0; i <= vps_max_layers_minus1; i++) { vps_included_layer_id[i] vps_reserved_zero_bit } vps_constraint_info_present_flag vps_constant_poc_cycle_per_aub if (vps_constant_poc_cycle_per_aub) vps_poc_cycle_aub ... }</pre>	Descriptor: w(4) w(5) v w(7) w(1) v w(1) w(1) w(1) v w(3) v v
<pre>slice_header() { slice_parameter_set_id if (rect_slice_flag NumBricksInPic > 1) slice_address if (rect_slice_flag && (single_brick_per_slice_flag) num_bricks_in_slice_minus1 slice_type if (NalUnitType == GRA_NUT) recovery_poc_cnt slice_pic_order_cnt_lsb ... if (vps_constant_poc_cycle_per_aub slice_poc_cycle_aub } } }</pre>	Descriptor: w(v) v v w(v) v w(v) w(v) v w(v) w(v) v v w(3) v v v v

Fig. 9

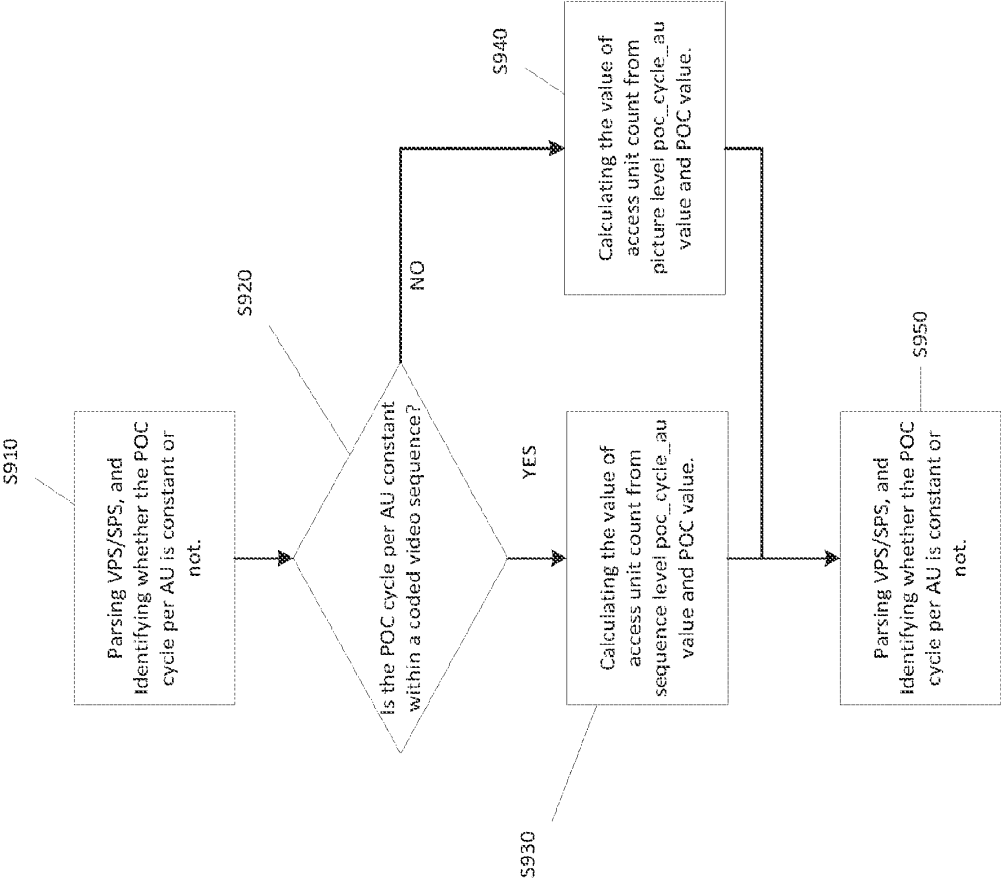
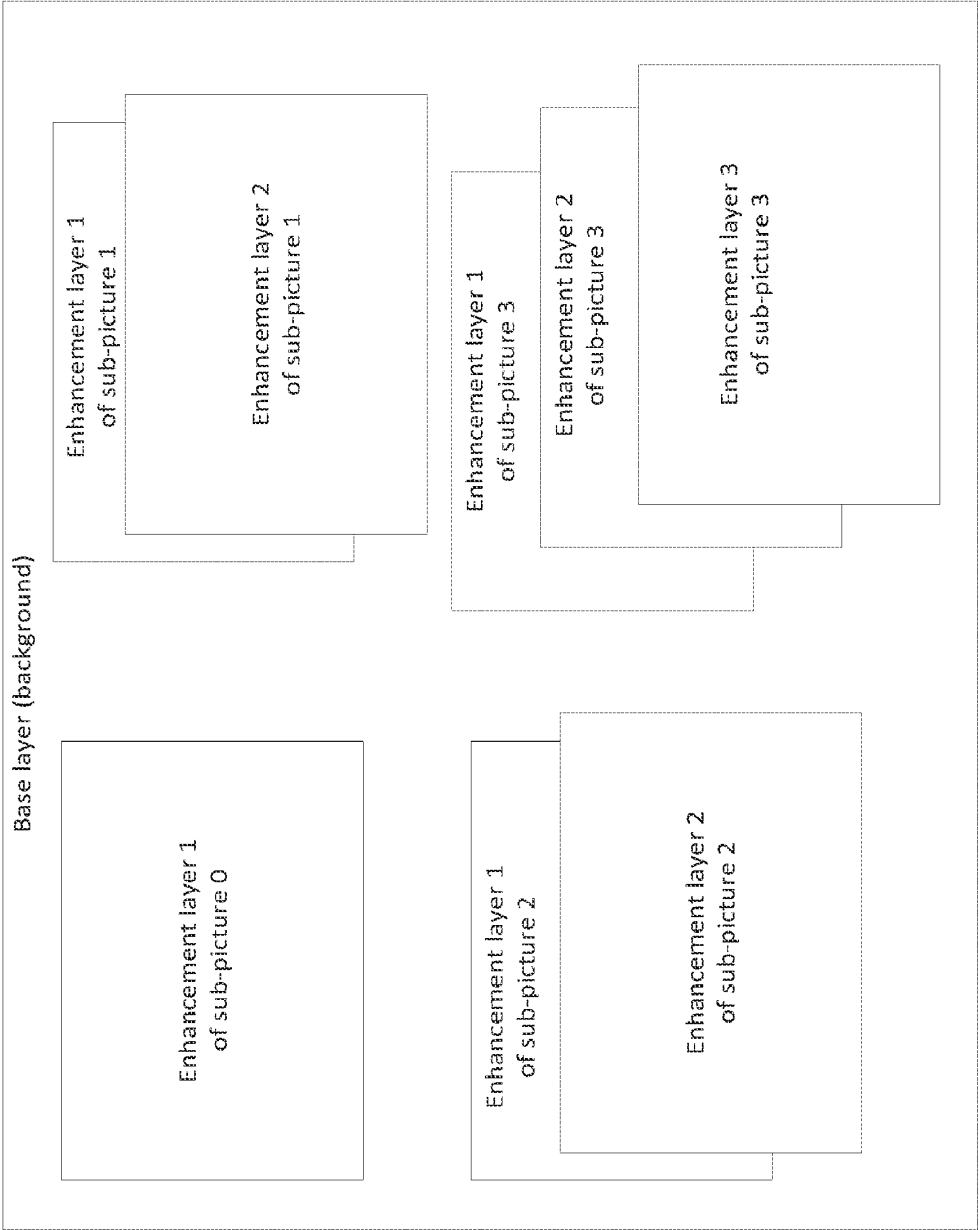


FIG. 10



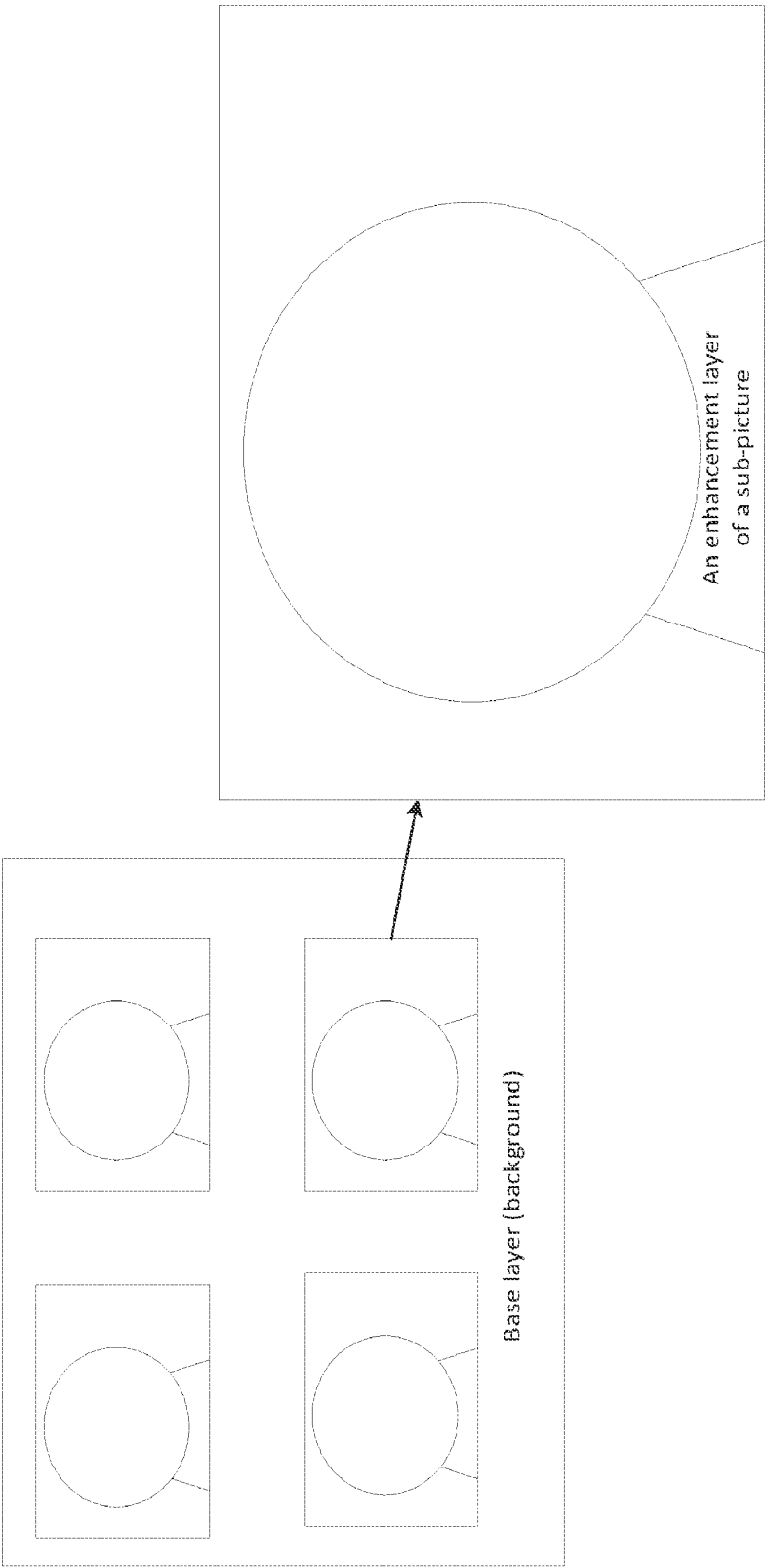
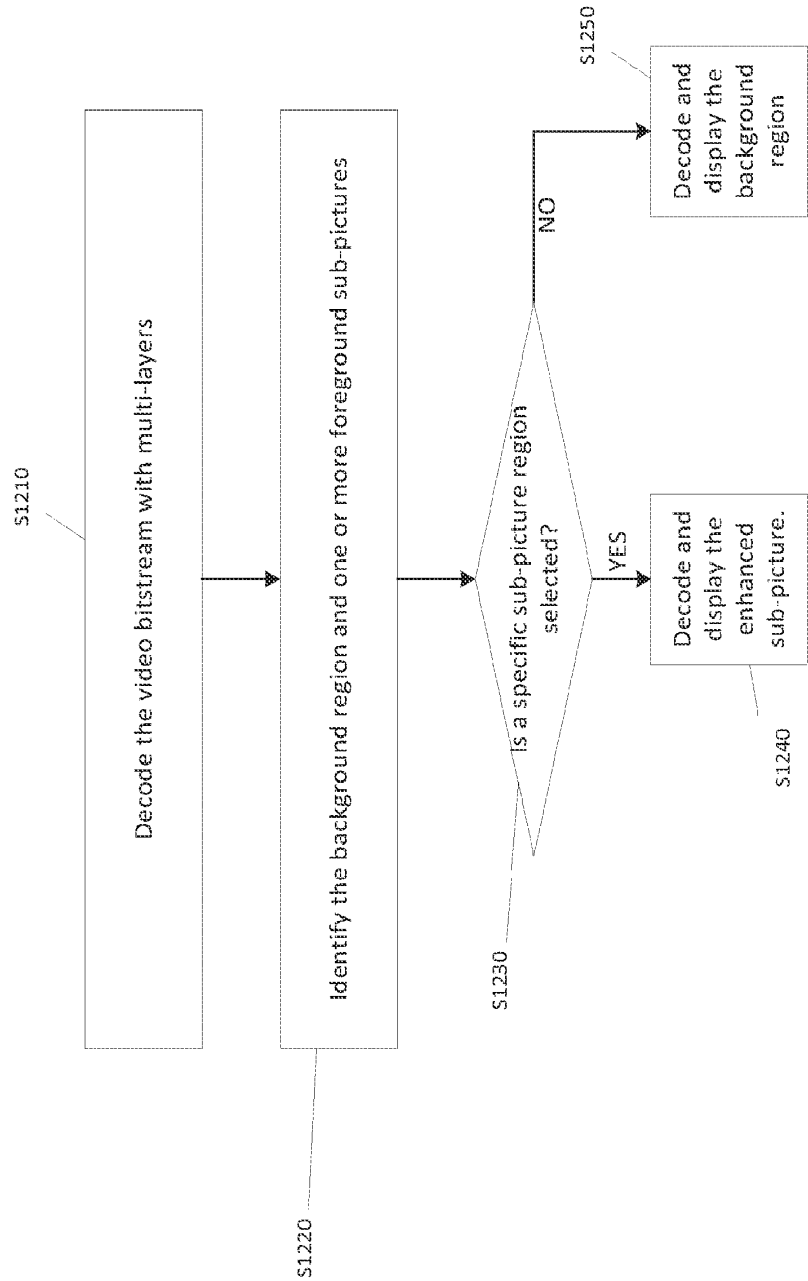


FIG. 11

Fig. 12



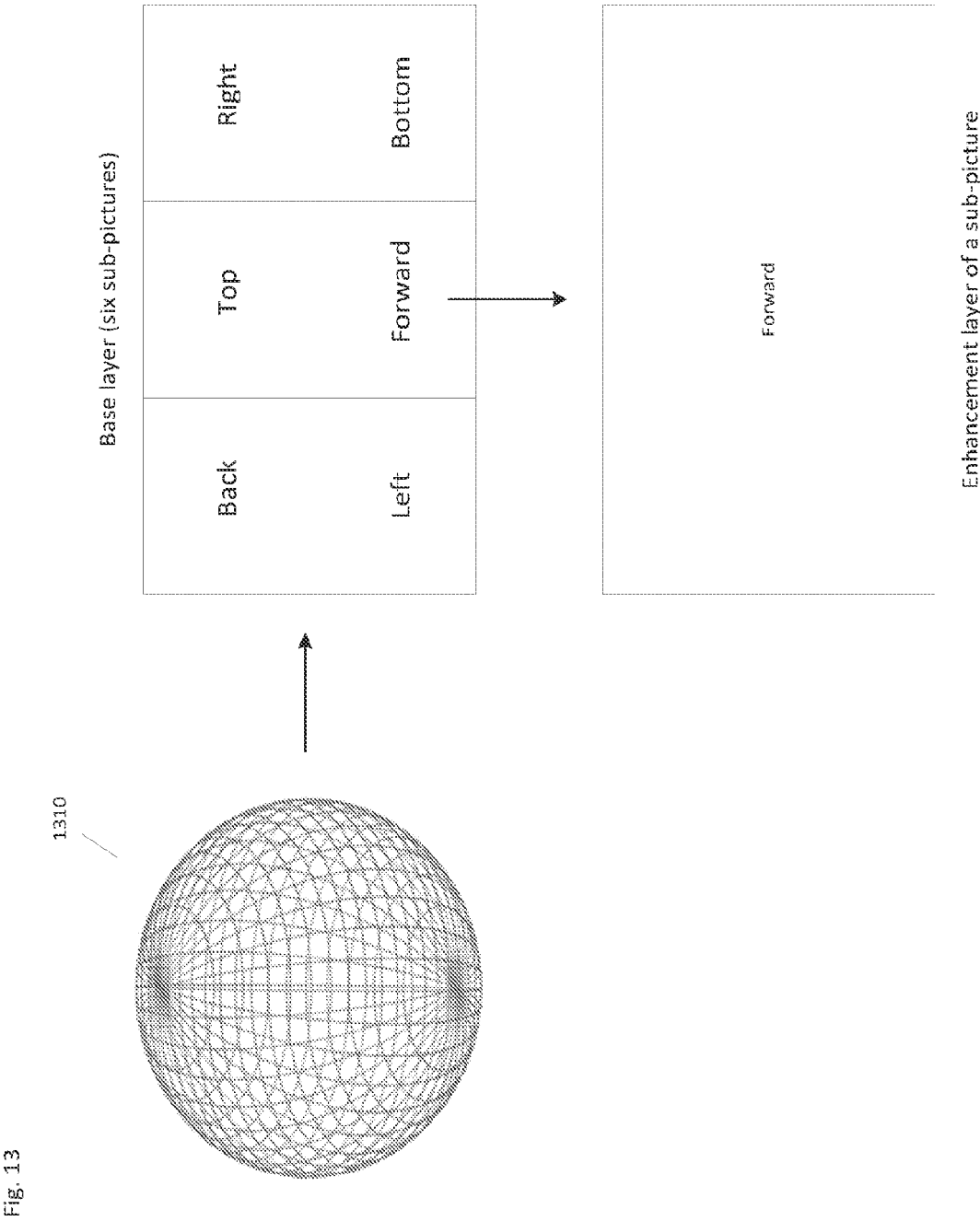
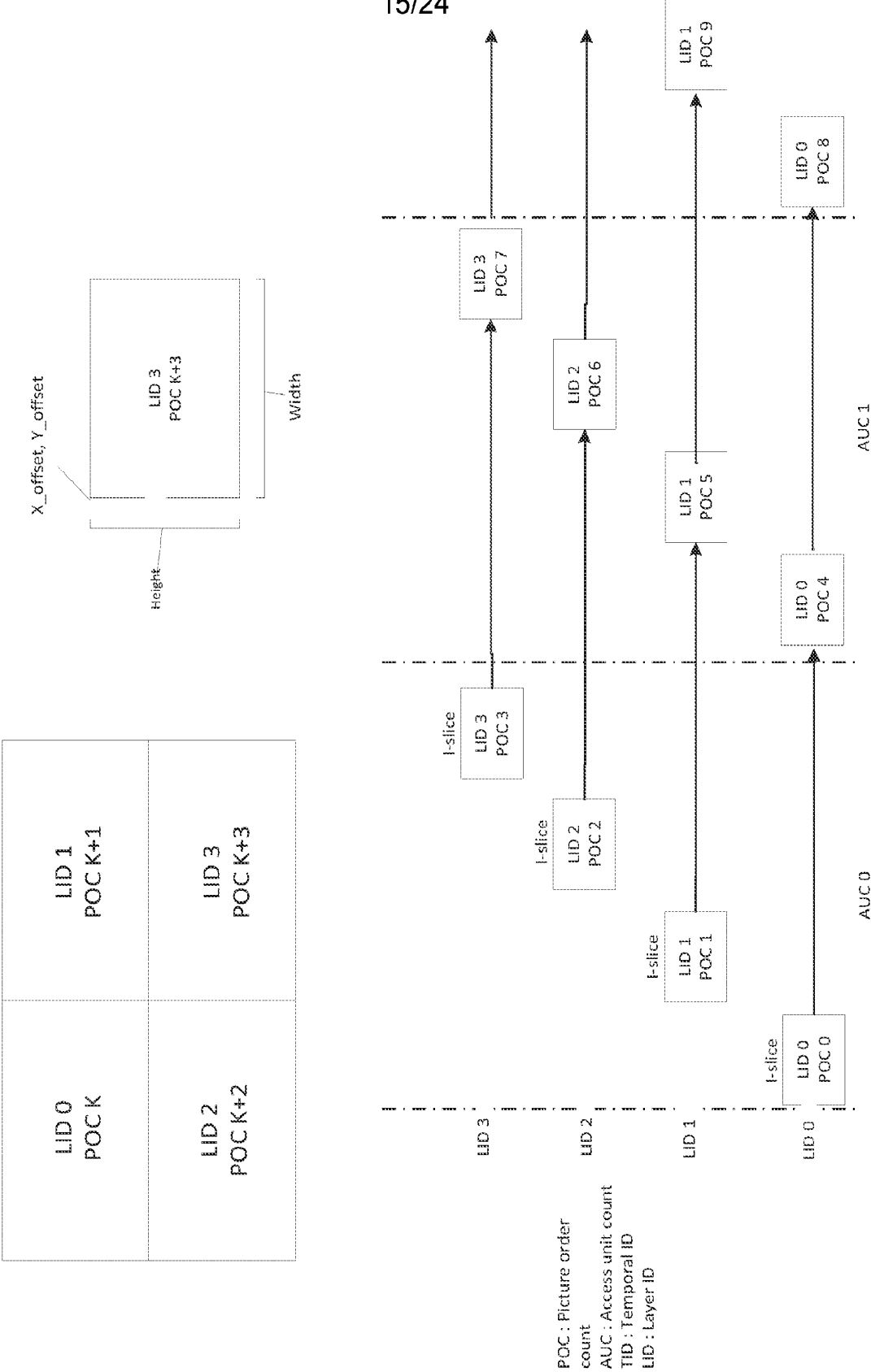


Fig. 14



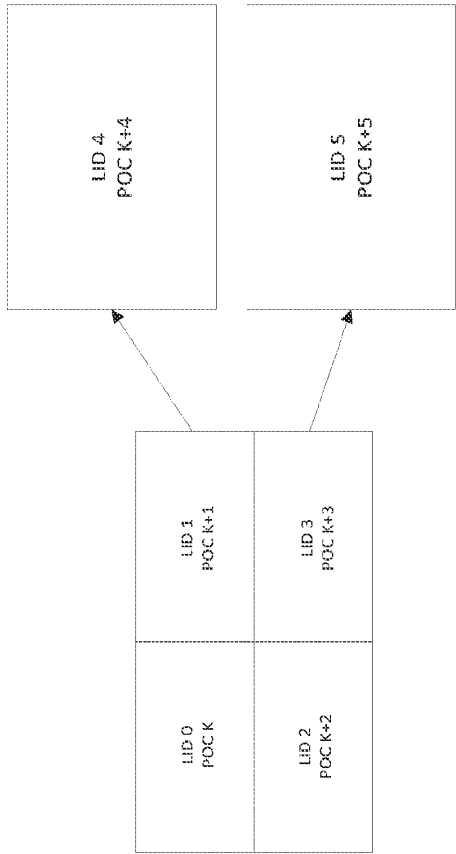
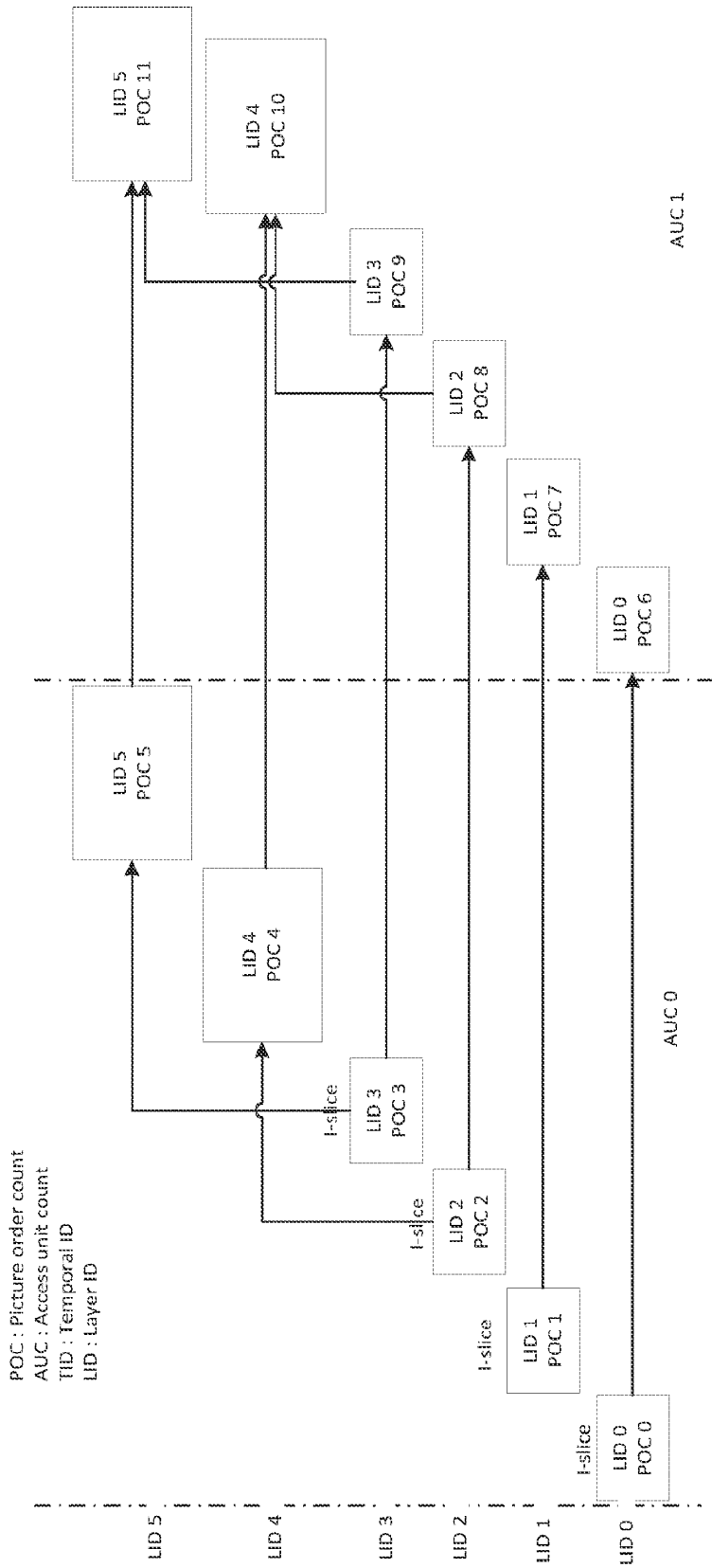


Fig. 15



POC : Picture order count
AUC : Access unit count
TID : Temporal ID
LID : Layer ID

Descriptor ^a	
video_parameter_set_id ^b	video_parameter_set_id ^b
video_video_parameter_set_id ^b	video_video_parameter_set_id ^b
vps_max_layers_minus1 ^b	vps_max_layers_minus1 ^b
for(i = 0; i <= vps_max_layers_minus1; i++) { ^c	
vps_included_layer_id[i] ^b	vps_included_layer_id[i] ^b
vps_reserved_zero_bits ^b	vps_reserved_zero_bits ^b
} ^b	} ^b
... ^b	... ^b
vps_sub_picture_dividing_flag ^b	vps_sub_picture_dividing_flag ^b
if(vps_sub_picture_dividing_flag) { ^c	if(vps_sub_picture_dividing_flag) { ^c
vps_full_pic_width_in_luma_samples ^b	vps_full_pic_width_in_luma_samples ^b
vps_full_pic_height_in_luma_samples ^b	vps_full_pic_height_in_luma_samples ^b
} ^b	} ^b
... ^b	... ^b
} ^b	} ^b

FIG. 16A

•var parameter_set_flag() {	Description
aps_decoding_parameter_set_id;	u(4)
aps_video_parameter_set_id;	u(4)
aps_max_sub_layers_minus1;	u(5)
...;	;
pic_width_in_luma_samples;	ue(v)
pic_height_in_luma_samples;	ue(v)
if(aps_sub_picture_dividing_flag) {	;
pic_offset_x;	ue(v)
pic_offset_y;	ue(v)
};	;
...;	;
};	;

FIG. 16B

Fig. 17

sub_region_partitioning_info(payloadSize) { ⁴	Descriptor ⁴
num_sub_region ²	ue(v) ²
num_layers ²	ue(v) ²
for(i = 0; i <= num_layers; i++) ²	²
layer_id [i] ²	ue(v) ²
for(i = 1; i < num_layers; i++) ²	²
for(j = 0; j < i; i++) ²	²
direct_dependency_flag [i] [j] ²	ue(1) ²
for(i = 0; i < num_sub_region; i++) { ²	²
num_layers_for_region [i] ²	ue(v) ²
for(j = 0; j < num_layers_for_region [i]; i++) ²	²
sub_region_layer_id [i] [j] ²	ue(v) ²
sub_region_offset_x [i] ²	ue(v) ²
sub_region_offset_y [i] ²	ue(v) ²
sub_region_width [i] ²	ue(v) ²
sub_region_height [i] ²	ue(v) ²
} ²	²
... ²	²
} ²	²

Fig. 18

video_parameter_set_rsp() {	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 < NumLayersInProfileTierLevel[i][i++]; {	
output_layer_flag[i][j]	u(1)
profile_tier_level_idx[i][j]	u(v)
}	
}	
}	
}	

Fig. 19

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_minus1[i]	u(2)
vps_pul_signal_flag[i]	u(1)
for(j = 0; j < NumLayersInLsg[i][i++])	
if(vps_output_layers_minus1[i] == 2)	
output_layer_flag[i][j]	u(1)
if vps_pul_signal_flag[i]	
profile_tier_level_id[i][j]	ue(v)
}	
}	
...	
}	

Fig. 20

	Descriptor
video_parameter_set_rbsp() {	
...	
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[i].vps_max_sub_layers_minus1	
for (i = 0; i < num_output_layer_sets; i++) {	
vps_output_layers_mode[i]	u(2)
vps_pdt_signal_flag[i]	u(1)
for (i = 0; i < NumLayersInIdSet[i]; i++) {	
num_output_subpic_layer_fill[i]	ue(v)
for (k = 0; k < num_output_subpic_layer_fill[k]; k++) {	
sub_pic_id_layer_fill[i][k]	u(8)
if (vps_output_layers_mode[i] == 2) {	
output_layer_flag[i][k]	u(1)
if (vps_pdt_signal_flag[i]) {	
profile_tier_level_id[i][k]	u(v)
}	
}	
}	
}	
}	

Fig. 21

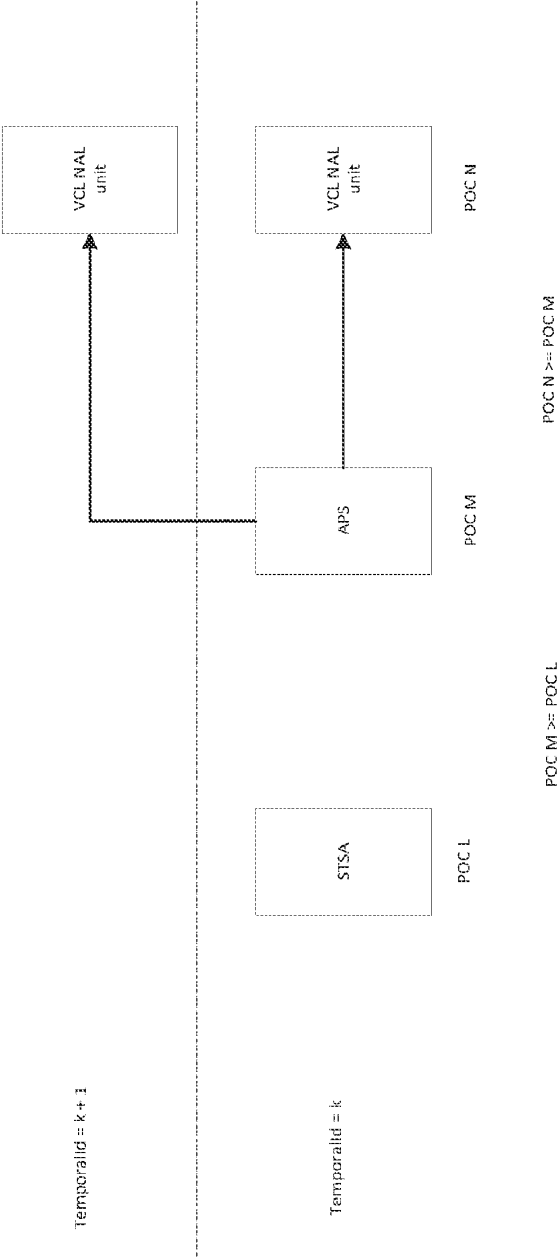


FIG. 22

2200

