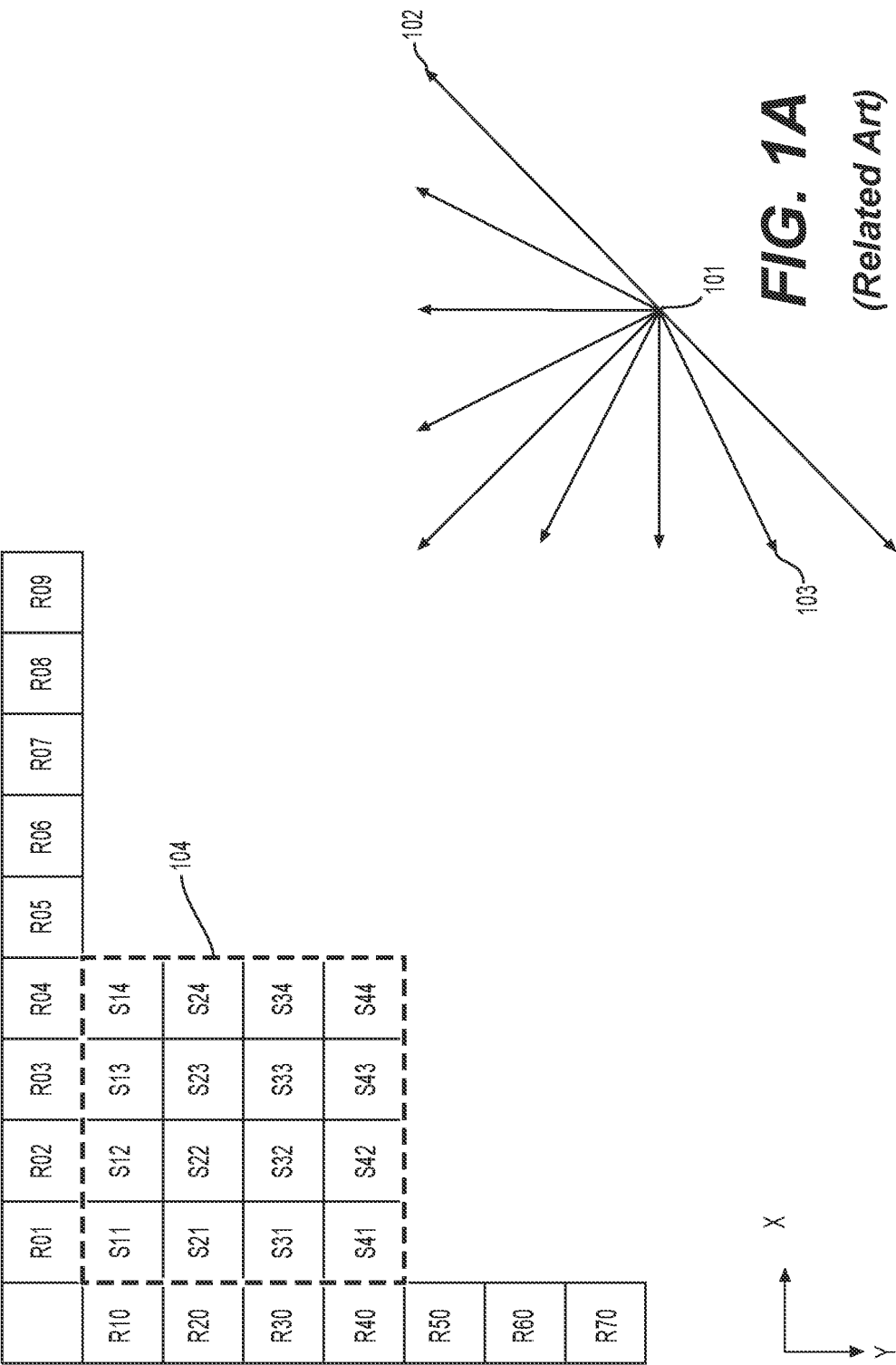
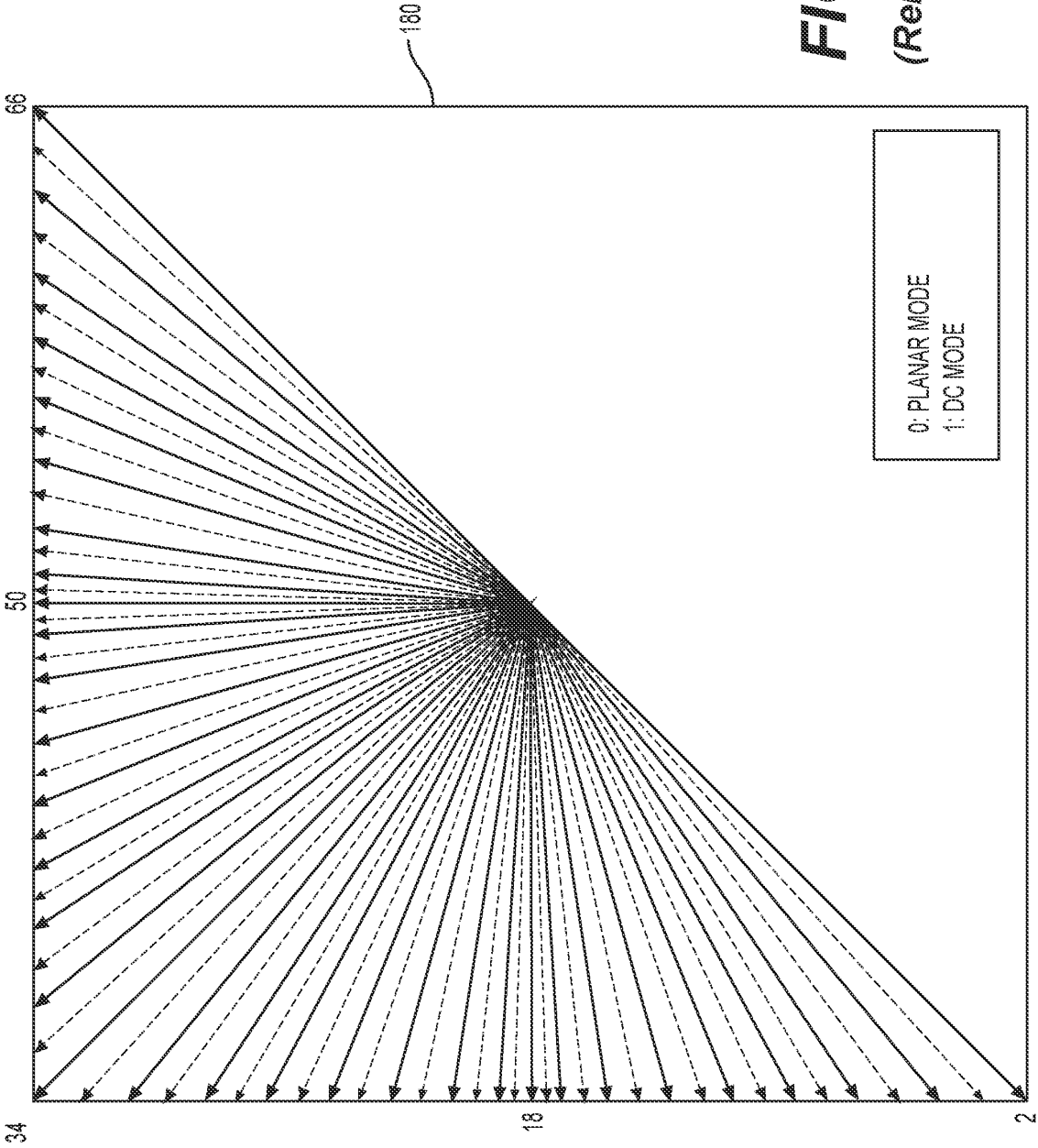






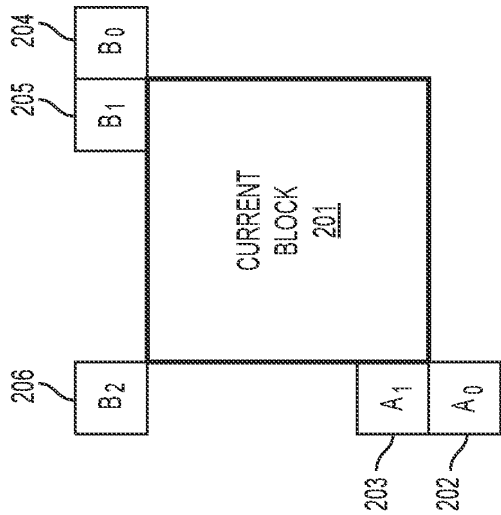
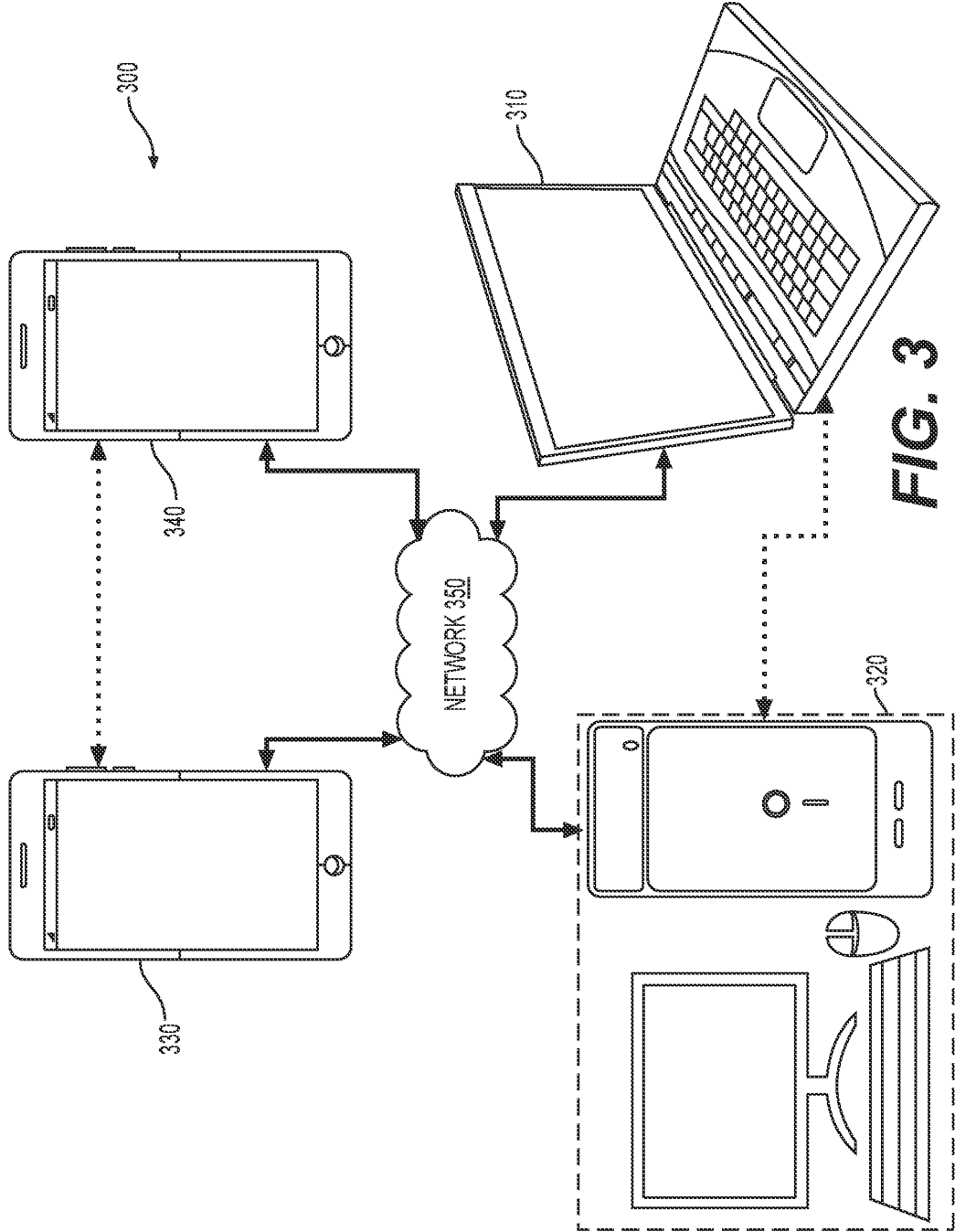


FIG. 2
(Related Art)



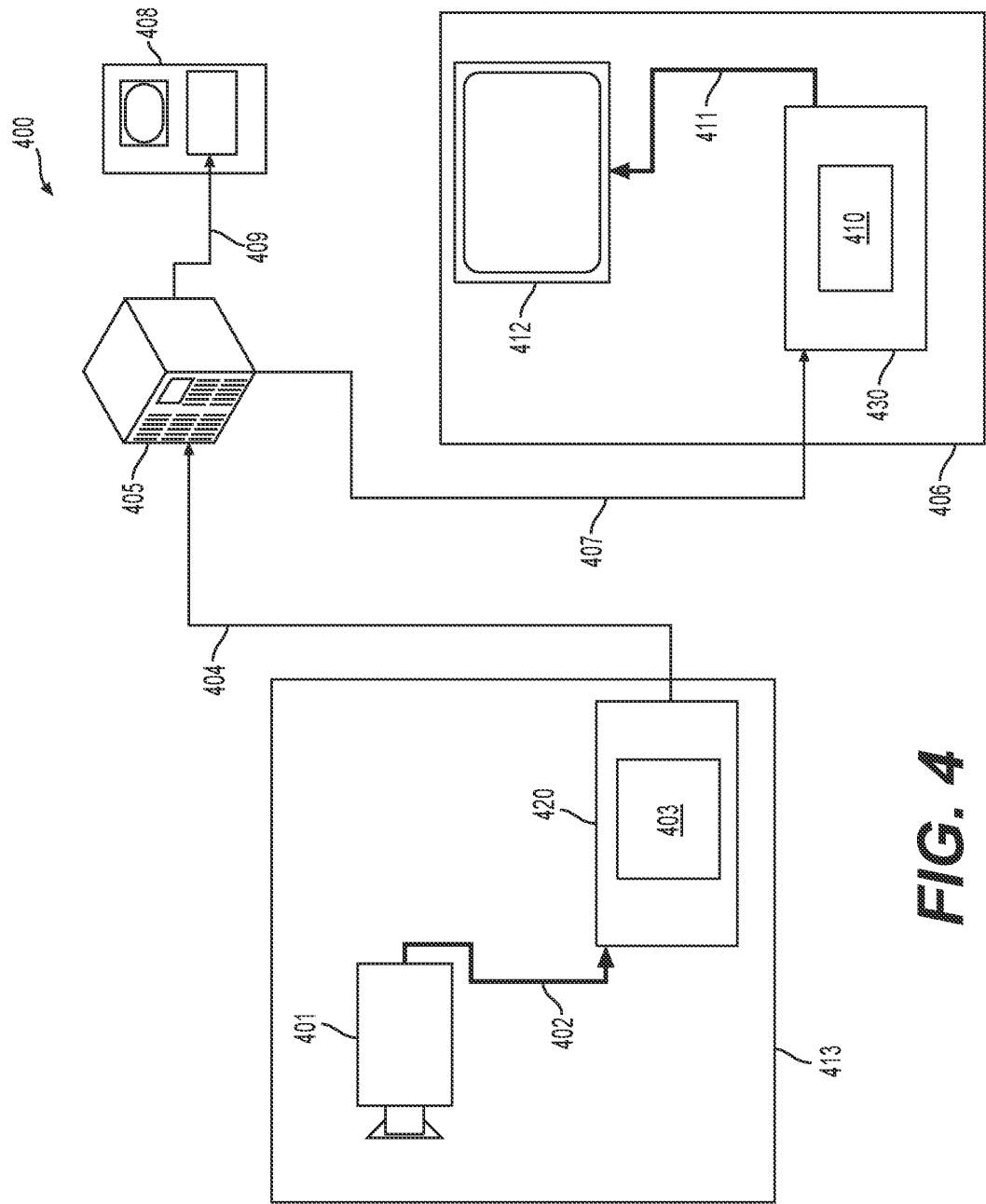


FIG. 4

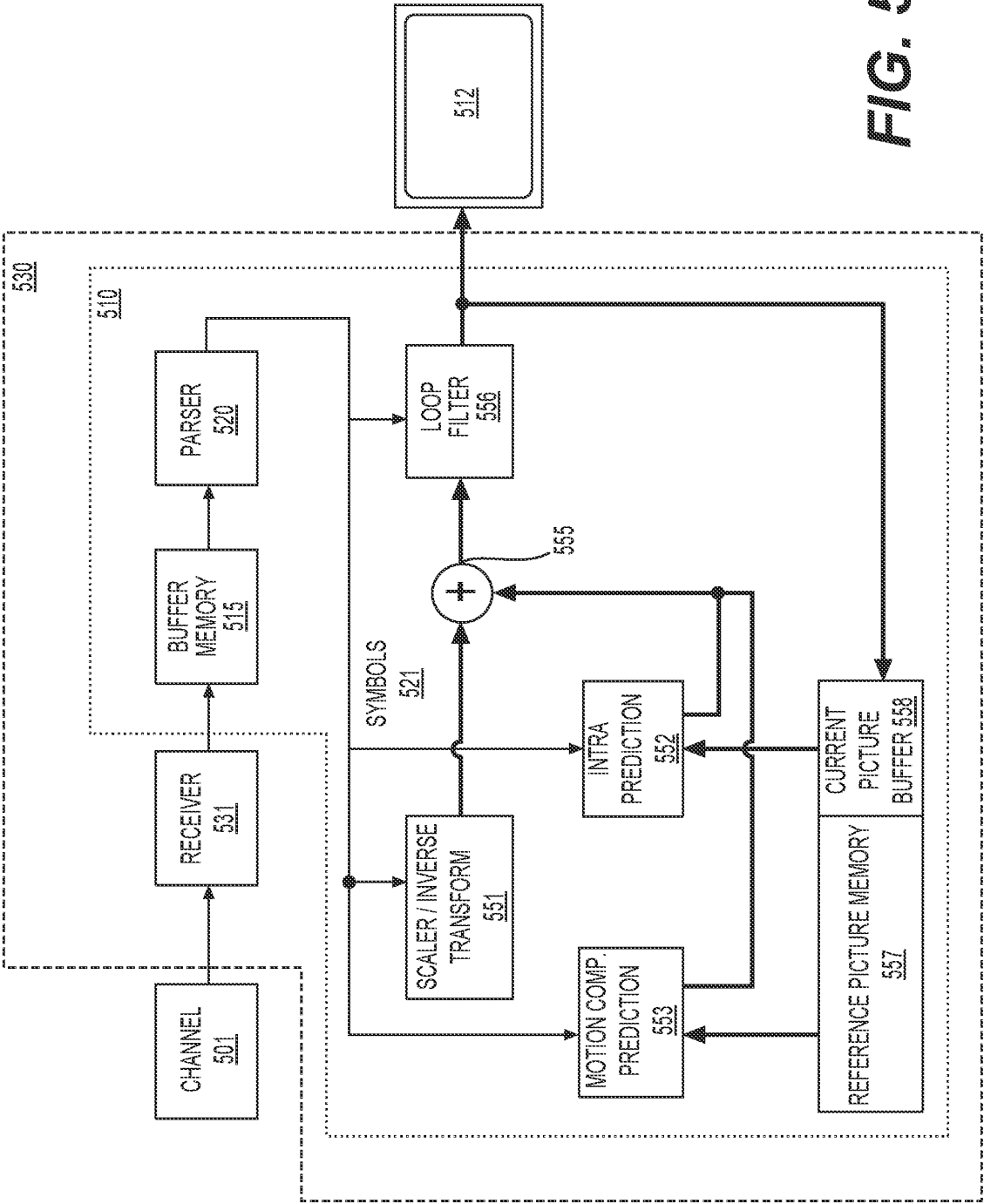


FIG. 5

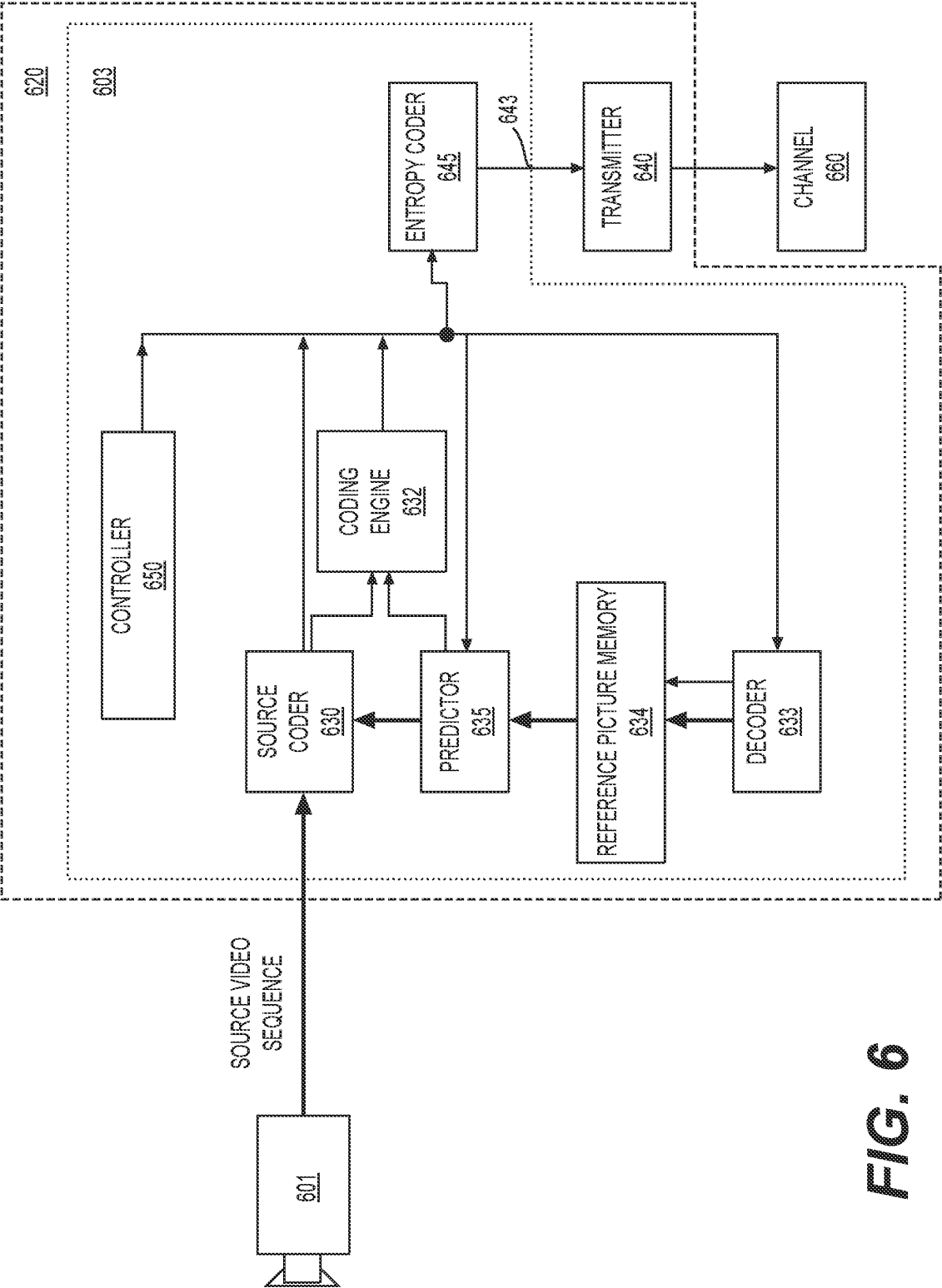


FIG. 6

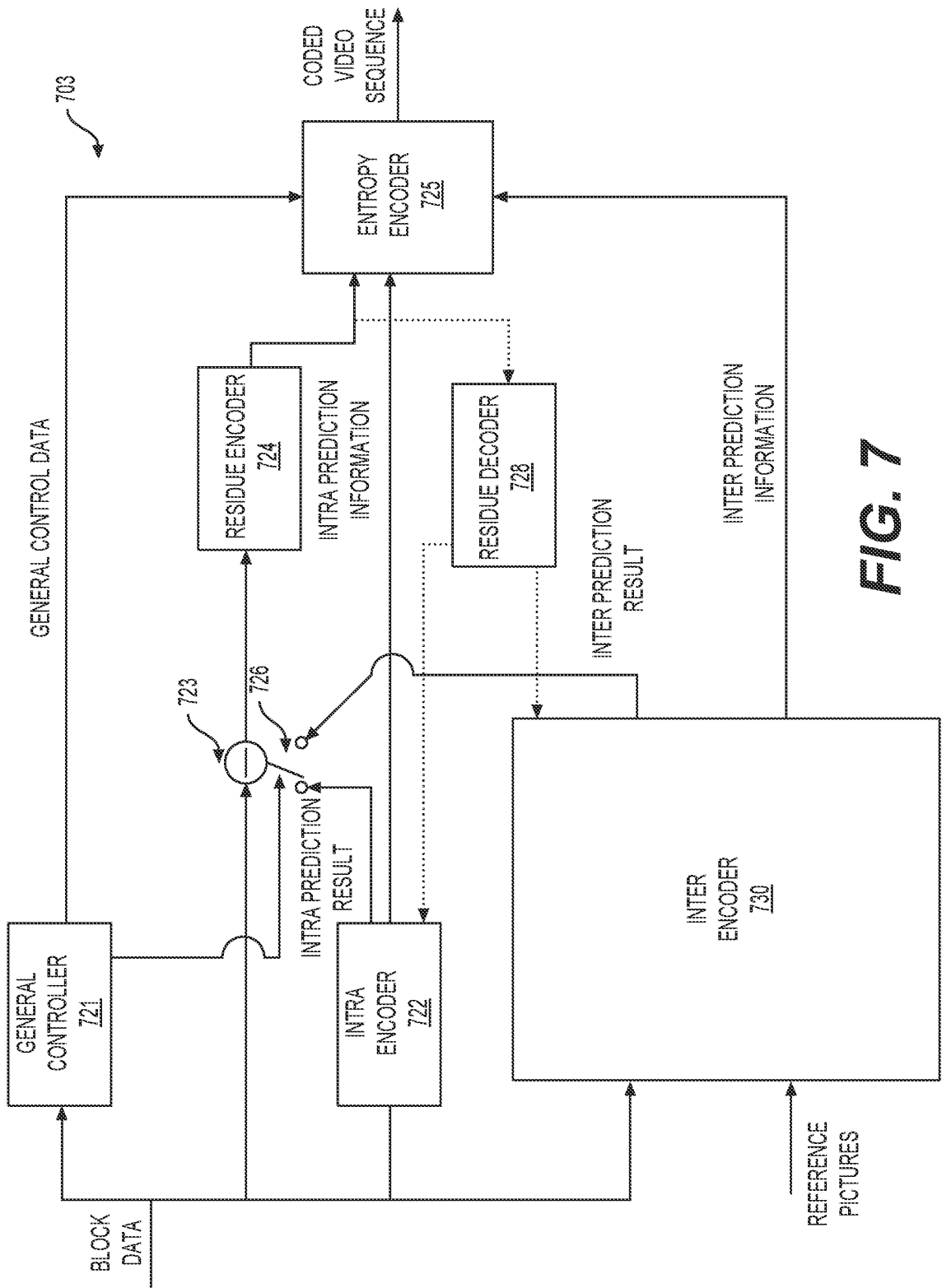


FIG. 7

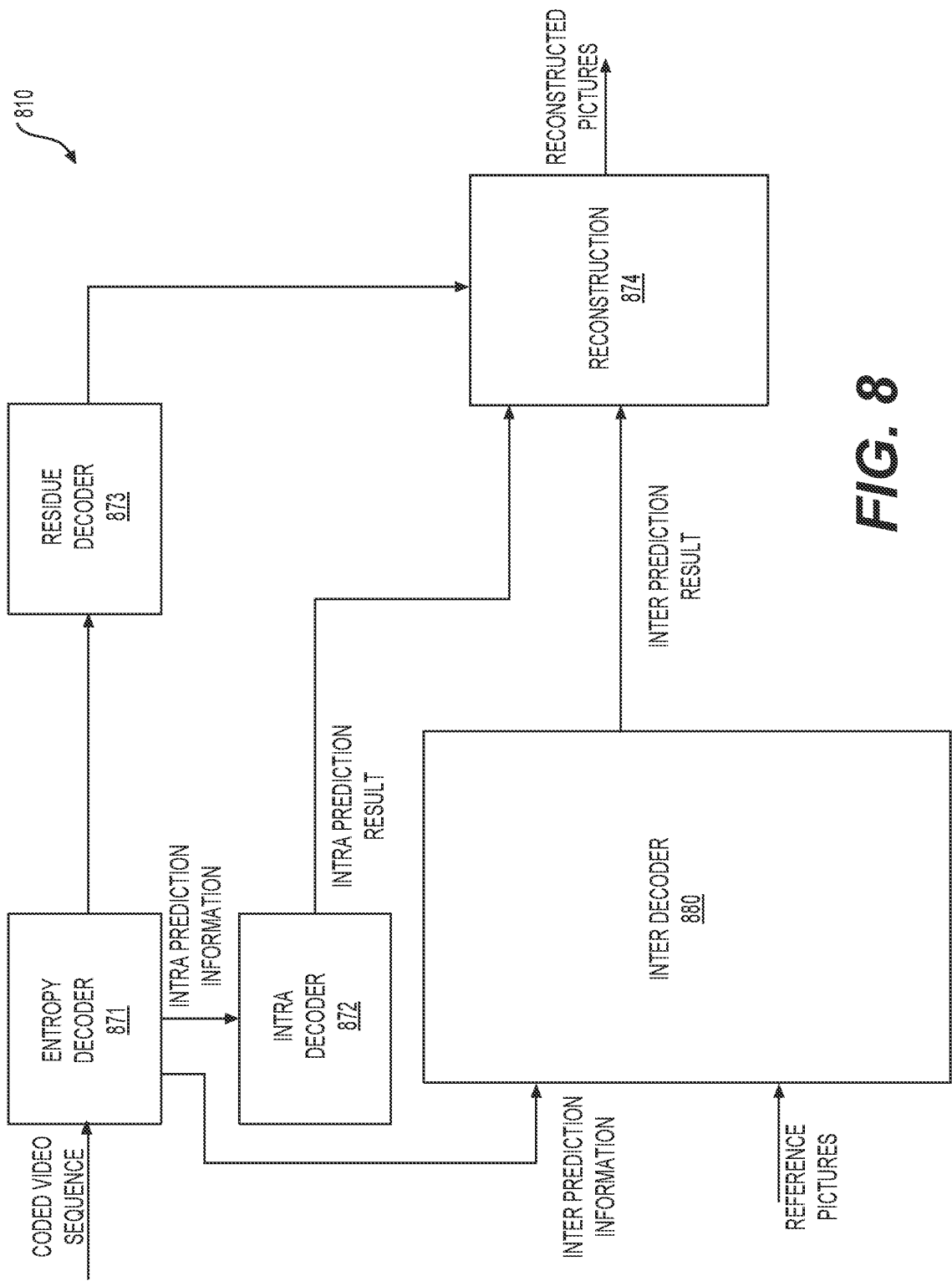


FIG. 8

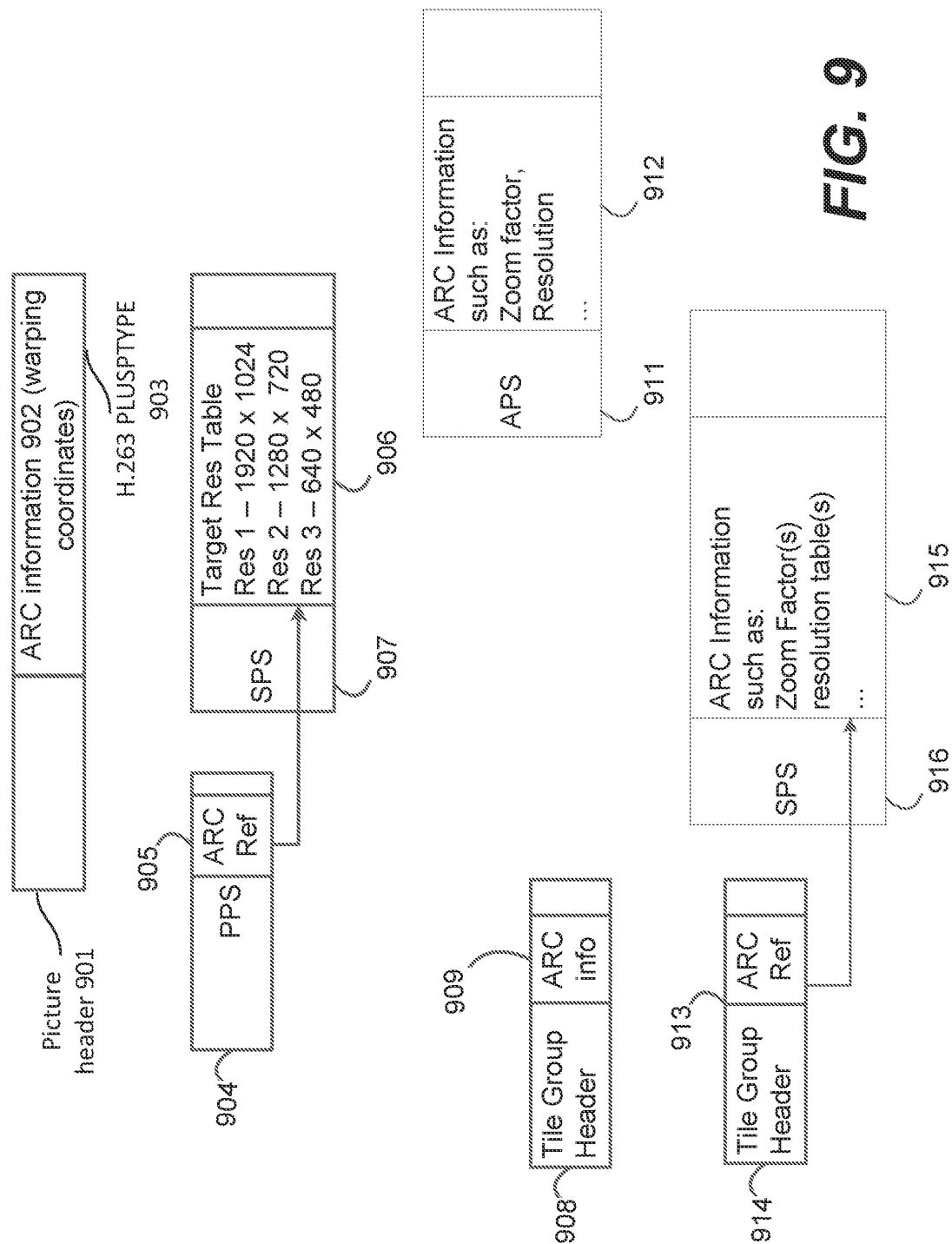


FIG. 9

1001	tile_group_header() {	
	...	
1003	if(adaptive_pic_resolution_change_flag) {	
1002	dec_pic_size_idx	u(1)
	}	
	...	
	}	

1010	seq_parameter_set_rbsp() {	Descriptor
	...	
1011	adaptive_pic_resolution_change_flag	u(1)
1012	if(adaptive_pic_resolution_change_flag) {	
1013 [	output_pic_width_in_luma_samples	ue(v)
	output_pic_height_in_luma_samples	ue(v)
1014	reference_pic_size_present_flag	u(1)
	if(reference_pic_size_present_flag)	
	{	
1015 [	reference_pic_width_in_luma_samples	ue(v)
	reference_pic_height_in_luma_samples	ue(v)
	}	
1016	num_dec_pic_size_in_luma_samples_minus1	ue(v)
	for(i = 0; i <=	
	num_dec_pic_size_in_luma_samples_minus1;	
	i++) {	
1017 [	dec_pic_width_in_luma_samples[i]	ue(v)
	dec_pic_height_in_luma_samples[i]	ue(v)
	}	
	}	
	...	
	}	

FIG. 10

	Descriptor
profile_tier_level(profileTierPresentFlag, maxNumSubLayersMinus1) {	
if(profileTierPresentFlag) {	
general_profile_idc	u(7)
general_tier_flag	u(1)
general_constraint_info()	
}	
general_level_idc	u(8)
if(profileTierPresentFlag) {	
num_sub_profiles	u(8)
for(i = 0; i < num_sub_profiles; i++)	
general_sub_profile_idc[i]	u(32)
}	
for(i = 0; i < maxNumSubLayersMinus1; i++)	
sublayer_level_present_flag[i]	u(1)
while(!byte_aligned())	
ptl_alignment_zero_bit	f(1)
for(i = 0; i < maxNumSubLayersMinus1; i++)	
if(sublayer_level_present_flag[i])	
sublayer_level_idc[i]	u(8)
}	

FIG. 11

	general_constraint_info() {	Descriptor
1201	general_non_packed_constraint_flag	u(1)
1202	general_frame_only_constraint_flag	u(1)
1205	general_non_projected_constraint_flag	u(1)
	general_one_picture_only_constraint_flag	u(1)
	no_lossless_coding_tool_constraint_flag	u(1)
	no_lossy_coding_tool_constraint_flag	u(1)
	intra_only_constraint_flag	u(1)
	max_bitdepth_constraint_idc	u(4)
	max_chroma_format_constraint_idc	u(2)
	single_layer_constraint_flag	u(1)
	all_layers_independent_constraint_flag	u(1)
	no_ref_pic_resampling_constraint_flag	u(1)
	no_res_change_in_clvs_constraint_flag	u(1)
	one_tile_per_pic_constraint_flag	u(1)
	pic_header_in_slice_header_constraint_flag	u(1)
	one_slice_per_pic_constraint_flag	u(1)
	one_subpic_per_pic_constraint_flag	u(1)
	...	
	no_partition_constraints_override_constraint_flag	u(1)
1206	no_sao_constraint_flag	u(1)
	no_alf_constraint_flag	u(1)
	no_radl_constraint_flag	u(1)
	no_idr_constraint_flag	u(1)
	no_cra_constraint_flag	u(1)
	no_gdr_constraint_flag	u(1)
	no_aps_constraint_flag	u(1)
	whilelet_byte_aligned()	
1203	gci_alignment_zero_bit	f(1)
	gci_num_reserved_bytes	u(8)
	for(i = 0; i < gci_num_reserved_bytes; i++)	
1204	gci_reserved_byte[i]	u(8)
	}	

FIG. 12

profile tier level(profileTierPresentFlag, maxNumSubLayersMinus1) {		
if(profileTierPresentFlag) {		
general profile idc		u(7)
general tier flag		u(1)
num available constraint flags		u(8)
general constraint info(num available constraint flags)		
}		
general level idc		u(8)
if(profileTierPresentFlag) {		
num sub profiles		u(8)
for(i = 0; i < num sub profiles; i++)		
general sub profile idc[i]		u(32)
}		
for(i = 0; i < maxNumSubLayersMinus1; i++)		
sublayer level present flag[i]		u(1)
while(!byte aligned())		
pdl alignment zero bit		f(1)
for(i = 0; i < maxNumSubLayersMinus1; i++)		
if(sublayer level present flag[i])		
sublayer level idc[i]		u(8)
}		

1301

FIG. 13A

Continued from FIG. 13A

general constraint info(num available constraint flags) {		
count = 0		
if(count < num available constraint flags)		
general non packed constraint flag		u(1)
count++		
if(count < num available constraint flags)		
general frame only constraint flag		u(1)
count++		
if(count < num available constraint flags)		
general non protected constraint flag		u(1)
count++		
...		
while(!byte aligned())		
gci alignment zero bit		f(1)
gci num reserved bytes		u(8)
for(i = 0; i < gci num reserved bytes; i++)		
gci reserved byte[i]		u(8)
}		

1302

FIG. 13B

1400

	general_constraint_info() {		Descriptor
1401	general_frame_structure_constraint_group_flag		u(1)
1402	high_level_functionality_constraint_group_flag		u(1)
1403	scalability_constraint_group_flag		u(1)
1404	partitioning_constraint_group_flag		u(1)
1405	intra_coding_tool_constraint_group_flag		u(1)
1406	inter_coding_tool_constraint_group_flag		u(1)
1407	transform_constraint_group_flag		u(1)
1408	inloop_filtering_constraint_group_flag		u(1)
	...		
	if(general_frame_structure_constraint_group_flag) {		
	general_non_packed_constraint_flag		u(1)
1411	general_frame_only_constraint_flag		u(1)
1412	general_non_projected_constraint_flag		u(1)
1413	general_one_picture_only_constraint_flag		u(1)
1414	}		
	}		

1410

FIG. 14A

1400

Continued from FIG. 14A

1420

if(high_level_functionality_constraint_group_flag) {	
{	
pic_header_in_slice_header_constraint_flag	u(1)
{	
no_rdl_constraint_flag	u(1)
no_idr_constraint_flag	u(1)
no_cra_constraint_flag	u(1)
no_gdr_constraint_flag	u(1)
no_aps_constraint_flag	u(1)
{	
{	
if(intra_coding_tool_constraint_group_flag) {	
{	
{	
if(inter_coding_tool_constraint_group_flag) {	
{	
{	
{	
while(!byte_aligned())	
gci_alignment_zero_bit	l(1)
gci_num_reserved_bytes	u(8)
for(i = 0; i < gci_num_reserved_bytes; i++)	
gci_reserved_byte[i]	u(8)
{	

1430

FIG. 14B

1500

	Descriptor
general_constraints_info() {	
gci_present_flag	u(1)
if(gci_present_flag) {	
/* general */	
gci_intra_only_constraint_flag	u(1)
gci_all_layers_independent_constraint_flag	u(1)
gci_one_au_only_constraint_flag	u(1)
/* picture format */	
gci_sixteen_minus_max_bitdepth_constraint_idc	u(4)
gci_three_minus_max_chroma_format_constraint_idc	u(2)
/* NAL unit type related */	
gci_no_mixed_nalu_types_in_pic_constraint_flag	u(1)
gci_no_trail_constraint_flag	u(1)
gci_no_stsa_constraint_flag	u(1)
gci_no_rasl_constraint_flag	u(1)
gci_no_rasl_constraint_flag	u(1)
gci_no_idr_constraint_flag	u(1)
gci_no_cra_constraint_flag	u(1)
gci_no_gdr_constraint_flag	u(1)
gci_no_aps_constraint_flag	u(1)
gci_no_idr_rpl_constraint_flag	u(1)

FIG. 15A

1500

Continued from FIG. 15A

/* tile, slice, subpicture partitioning */	
gci one tile per pic constraint flag	u(1)
gci_pic_header_in_slice_header_constraint_flag	u(1)
gci one slice per pic constraint flag	u(1)
gci no rectangular slice constraint flag	u(1)
gci_one_slice_per_subpic_constraint_flag	u(1)
gci no subpic info constraint flag	u(1)
/* CTU and block partitioning */	
gci_three_minus_max_log2_ctu_size_constraint_idc	u(2)
gci_no_partition_constraints_override_constraint_flag	u(1)
gci no mtt constraint flag	
gci_no_qtbt_dual_tree_intra_constraint_flag	u(1)
/* intra */	
gci no palette constraint flag	u(1)
gci no ibc constraint flag	u(1)
gci no isp constraint flag	u(1)
gci no mrl constraint flag	u(1)
gci no mip constraint flag	u(1)
gci no cclm constraint flag	u(1)

FIG. 15B

Continued from FIG. 15B

1500

/* inter */		
gci_no_ref_pic_resampling_constraint_flag		u(1)
gci_no_res_change_in_clvs_constraint_flag		u(1)
gci_no_weighted_prediction_constraint_flag		u(1)
gci_no_ref_wraparound_constraint_flag		u(1)
gci_no_temporal_mvp_constraint_flag		u(1)
gci_no_sbtmvp_constraint_flag		u(1)
gci_no_amvr_constraint_flag		u(1)
gci_no_bdof_constraint_flag		u(1)
gci_no_smvd_constraint_flag		u(1)
gci_no_dmvr_constraint_flag		u(1)
gci_no_mmvd_constraint_flag		u(1)
gci_no_affine_motion_constraint_flag		u(1)
gci_no_prof_constraint_flag		u(1)
gci_no_bew_constraint_flag		u(1)
gci_no_ciip_constraint_flag		u(1)
gci_no_gpm_constraint_flag		u(1)

FIG. 15C

/* transform_quantization_residual */	
gci_no_luma_transform_size_64_constraint_flag	u(1)
gci_no_transform_skip_constraint_flag	u(1)
gci_no_bdpcm_constraint_flag	u(1)
gci_no_mis_constraint_flag	u(1)
gci_no_lfst_constraint_flag	u(1)
gci_no_joint_cber_constraint_flag	u(1)
gci_no_sbt_constraint_flag	u(1)
gci_no_act_constraint_flag	u(1)
gci_no_explicit_scaling_list_constraint_flag	u(1)
gci_no_dep_quant_constraint_flag	u(1)
gci_no_sign_data_hiding_constraint_flag	u(1)
gci_no_cu_qp_delta_constraint_flag	u(1)
gci_no_chroma_qp_offset_constraint_flag	u(1)
/* loop filter */	
gci_no_sao_constraint_flag	u(1)
gci_no_alf_constraint_flag	u(1)
gci_no_cclaf_constraint_flag	u(1)
gci_no_lmcs_constraint_flag	u(1)
gci_no_ladf_constraint_flag	u(1)
gci_no_virtual_boundaries_constraint_flag	u(1)
gci_num_reserved_bits	u(8)
for(i = 0; i < gci_num_reserved_bits; i++)	
gci_reserved_zero_bit[i]	u(1)
{	
while(!byte_aligned())	
gci_alignment_zero_bit	
}	
f(1)	

1500

1517

Continued
from FIG. 15C

1518

FIG. 15D

	Descriptor
profile_tier_level(profileTierPresentFlag, MaxNumSubLayersMinus1) {	
if(profileTierPresentFlag) {	
general_profile_idc	u(7)
general_tier_flag	u(1)
}	
general_level_idc	u(8)
ptl_frame_only_constraint_flag	u(1)
ptl_multilayer_enabled_flag	u(1)
if(profileTierPresentFlag)	
general_constraints_info()	
for(i = MaxNumSubLayersMinus1 - 1; i >= 0; i--)	
ptl_sublayer_level_present_flag[i]	u(1)
while(!byte_aligned())	
ptl_reserved_zero_bit	u(1)
for(i = MaxNumSubLayersMinus1 - 1; i >= 0; i--)	
if(ptl_sublayer_level_present_flag[i])	
sublayer_level_idc[i]	u(8)
if(profileTierPresentFlag) {	
ptl_num_sub_profiles	u(8)
for(i = 0; i < ptl_num_sub_profiles; i++)	
general_sub_profile_idc[i]	u(32)
}	
}	

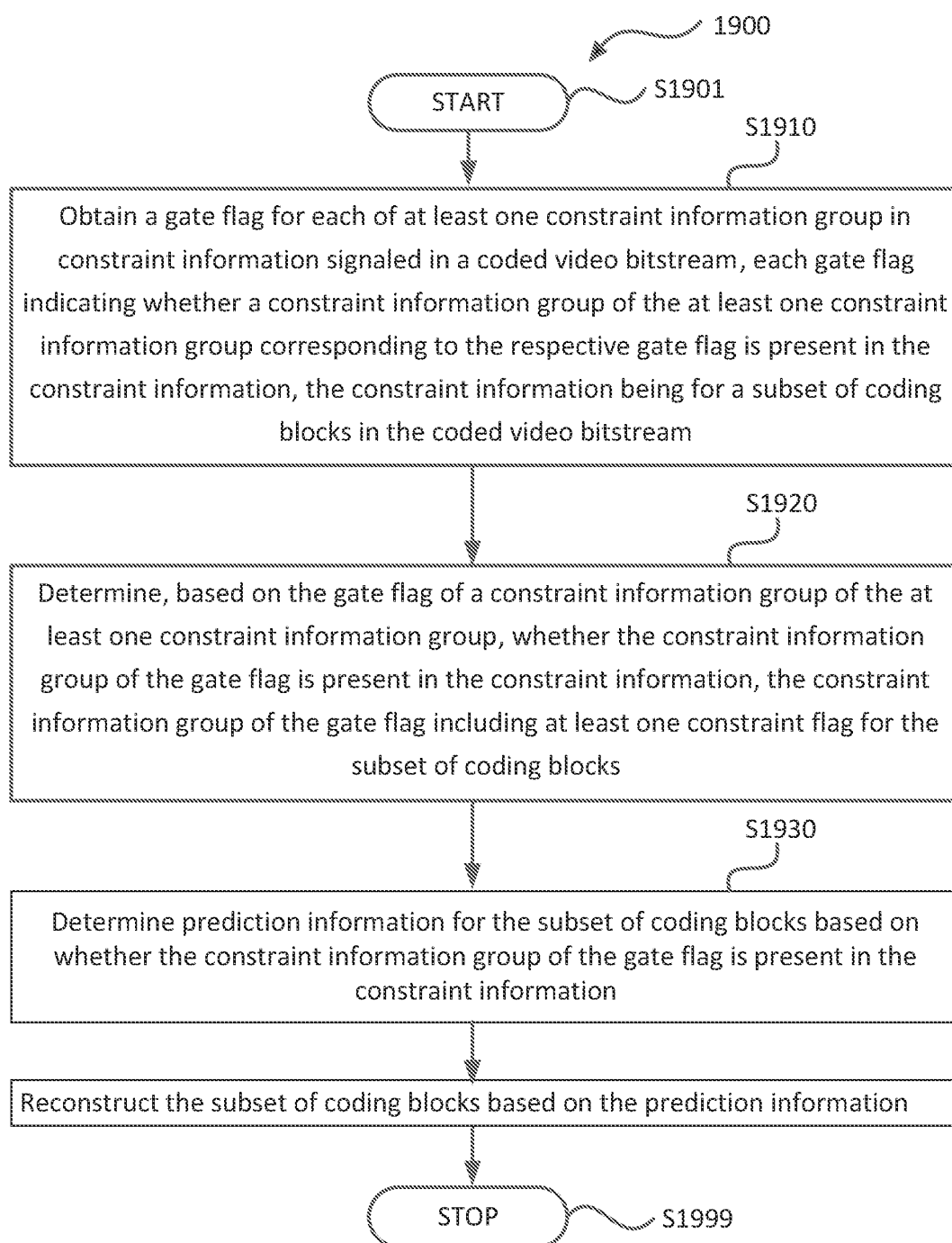
FIG. 16

FIG. 17

seq_parameter_set_rbsp() {	Descriptor
sps_seq_parameter_set_id	u(4)
...	
sps_field_seq_flag	u(1)
...	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

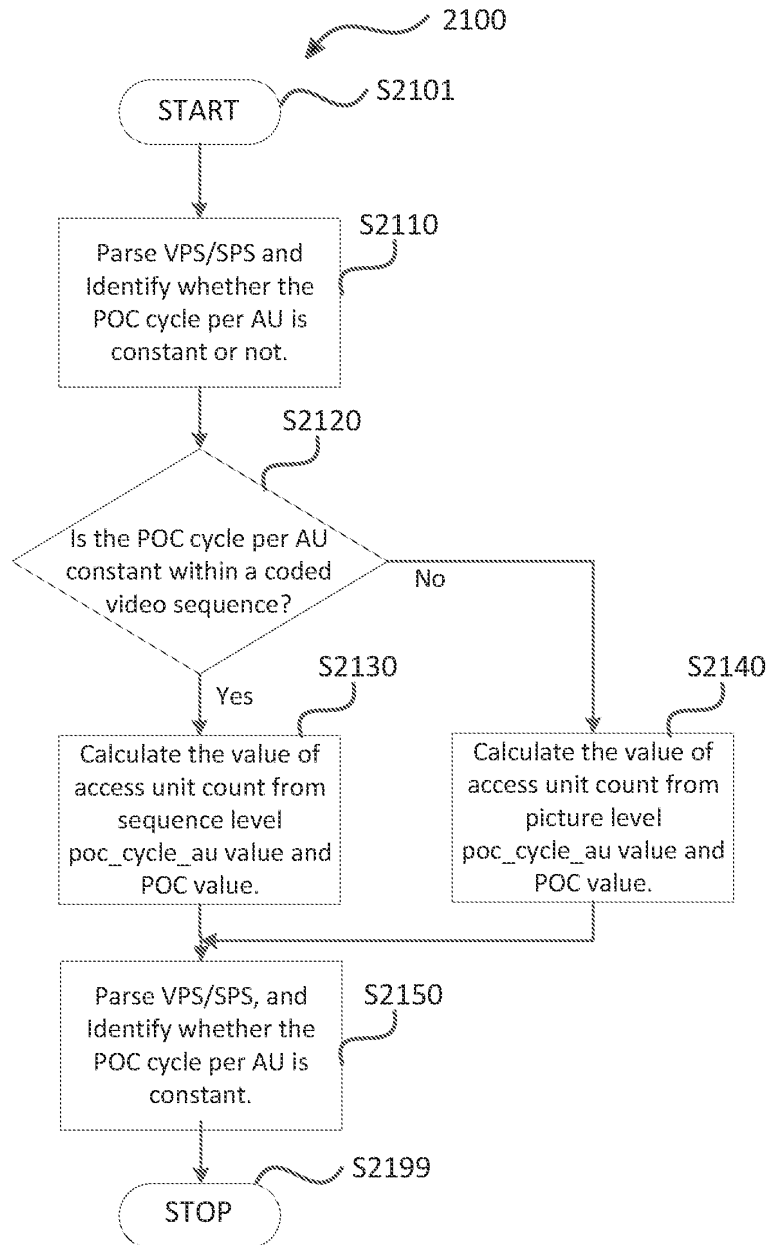
FIG. 18

video_parameter_set_rbsp() {	Descriptor
...	
vps_field_seq_flag	u(1)
...	
}	

**FIG. 19**

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

FIG. 20

**FIG. 21**

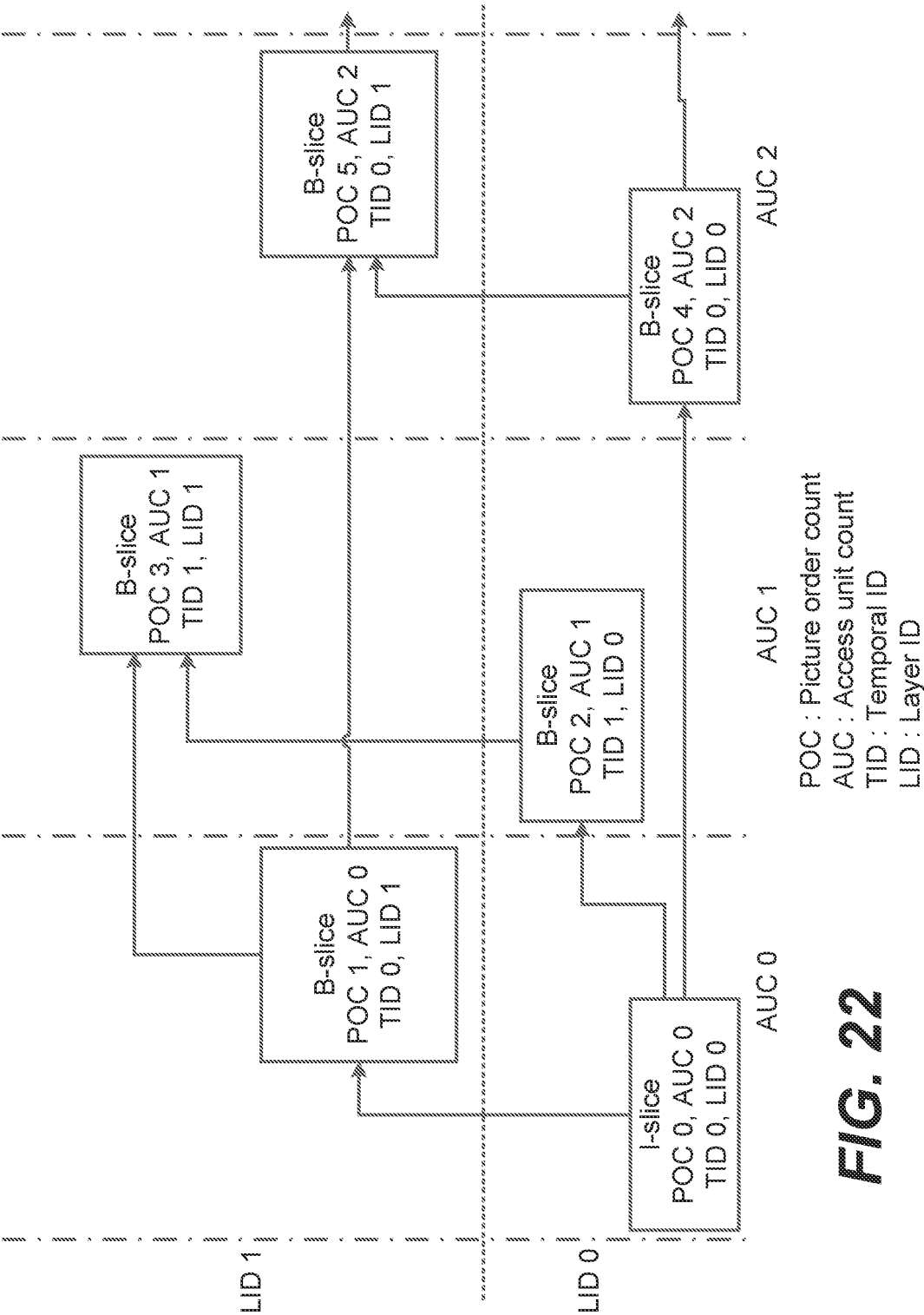


FIG. 22

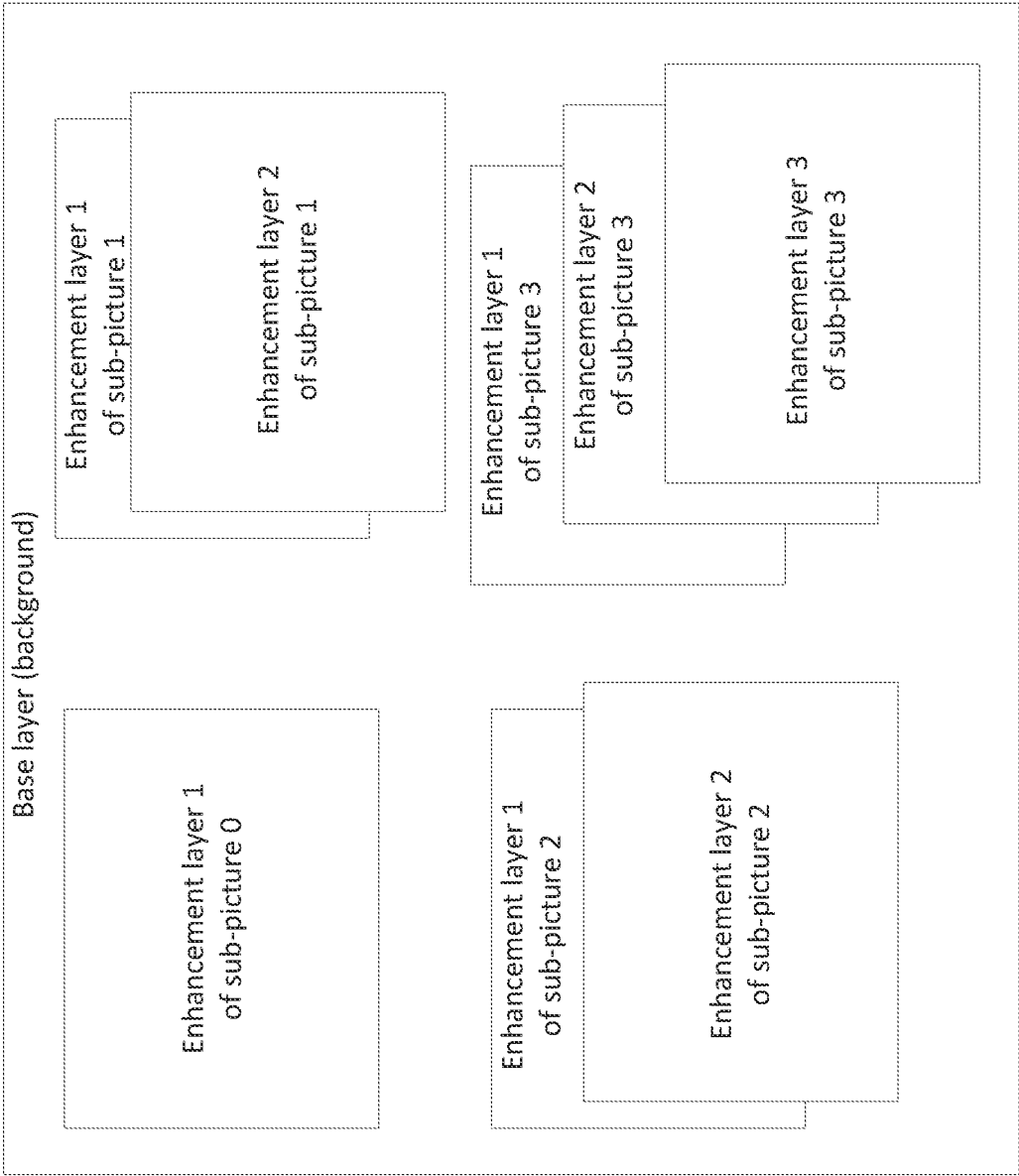
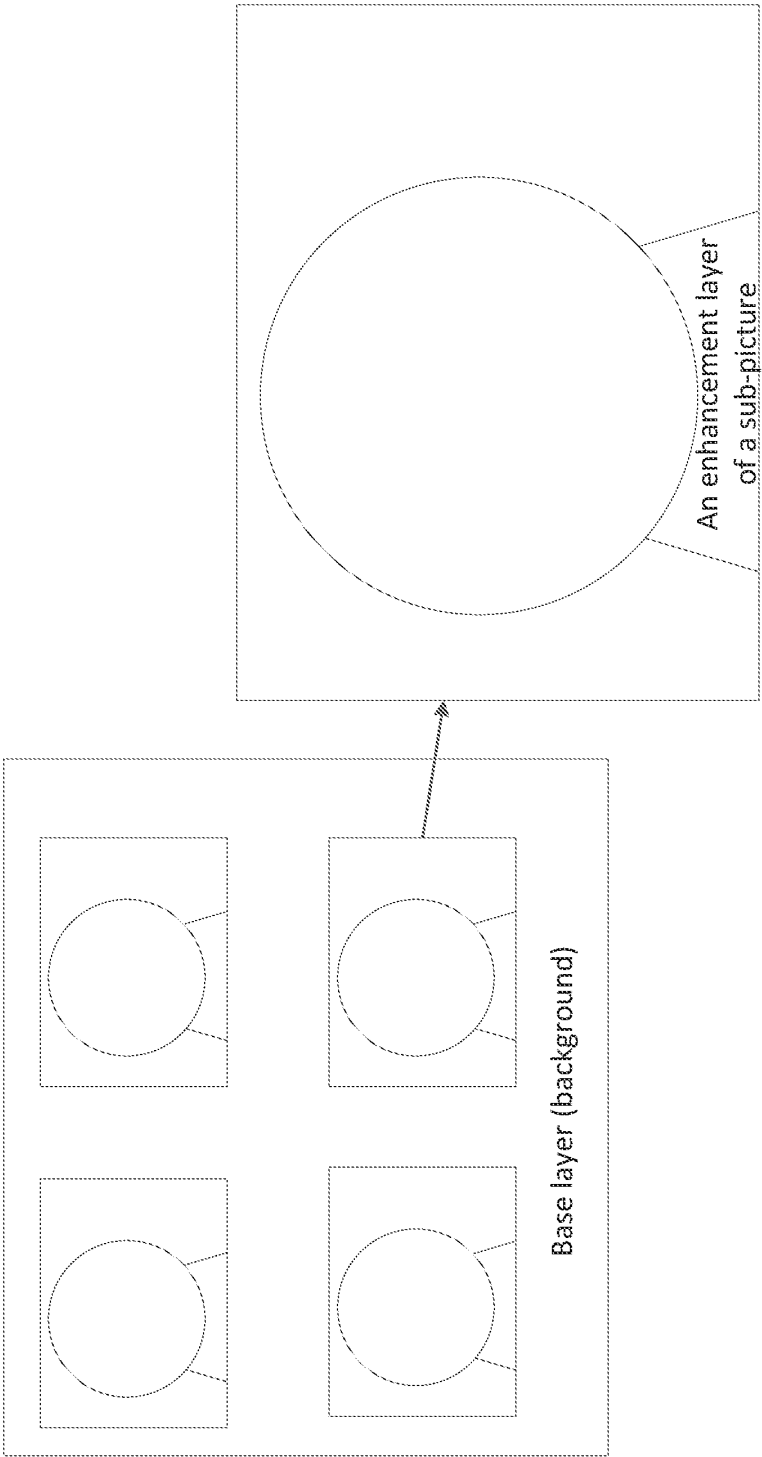


FIG. 23

FIG. 24



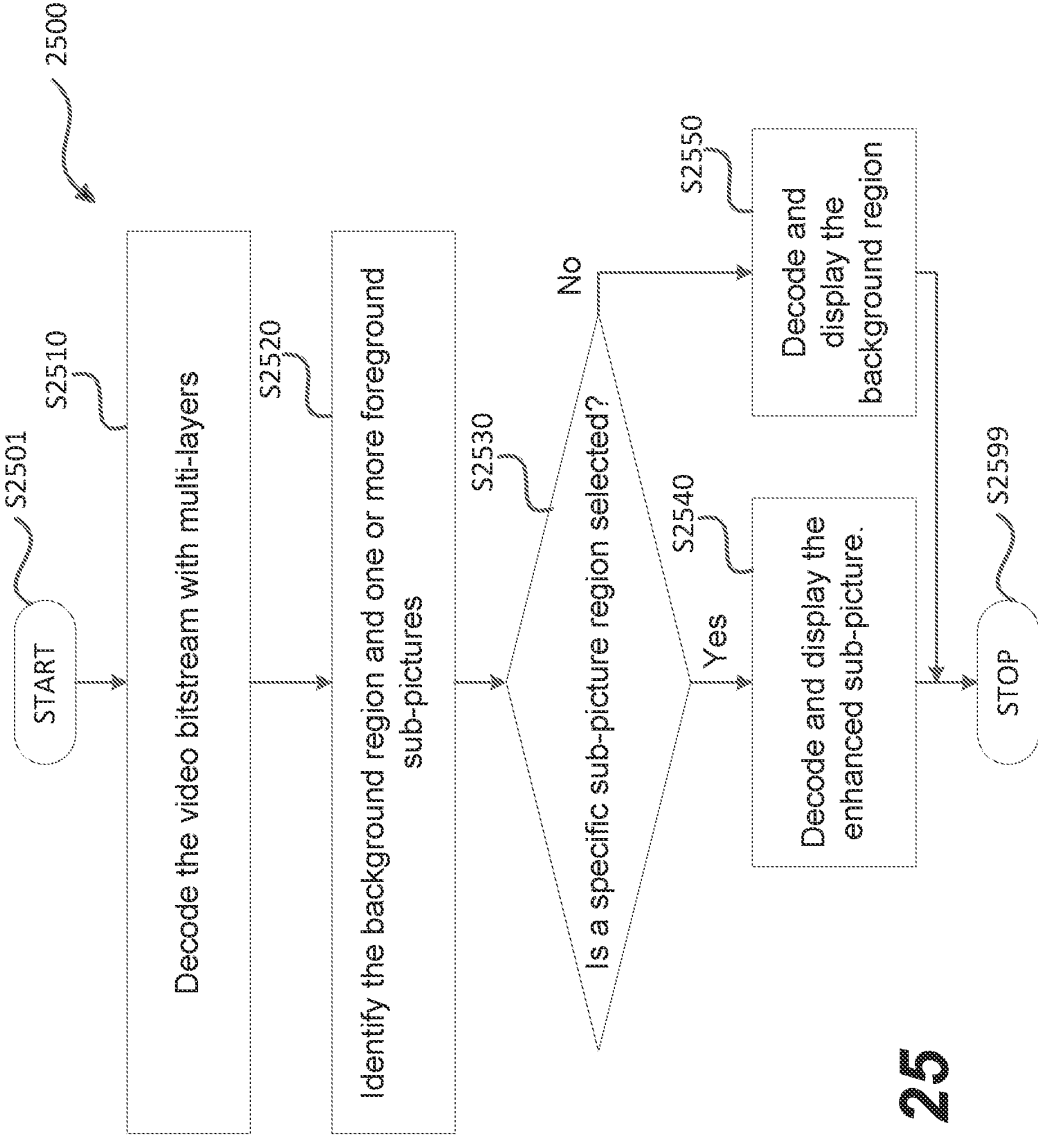


FIG. 25

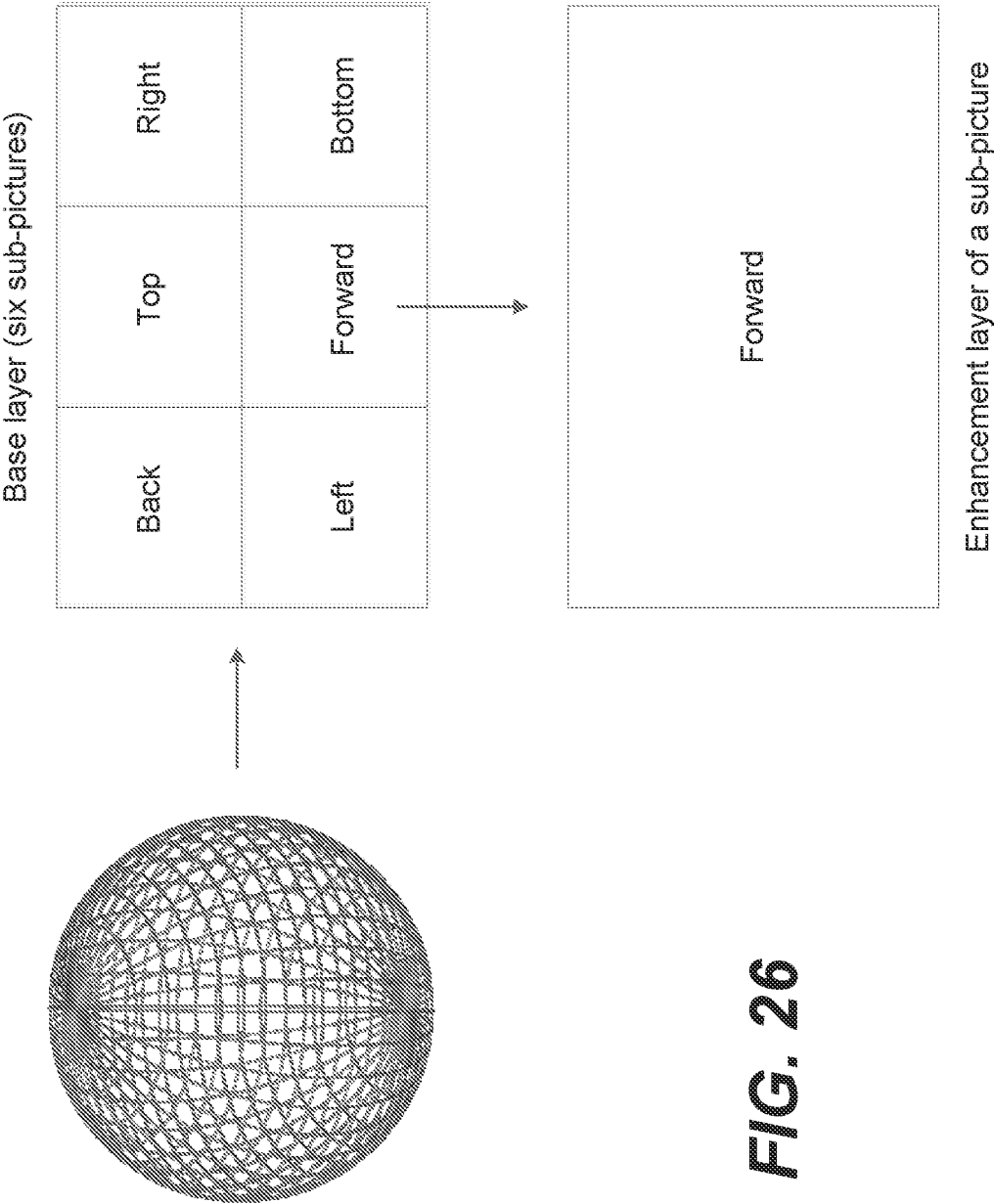
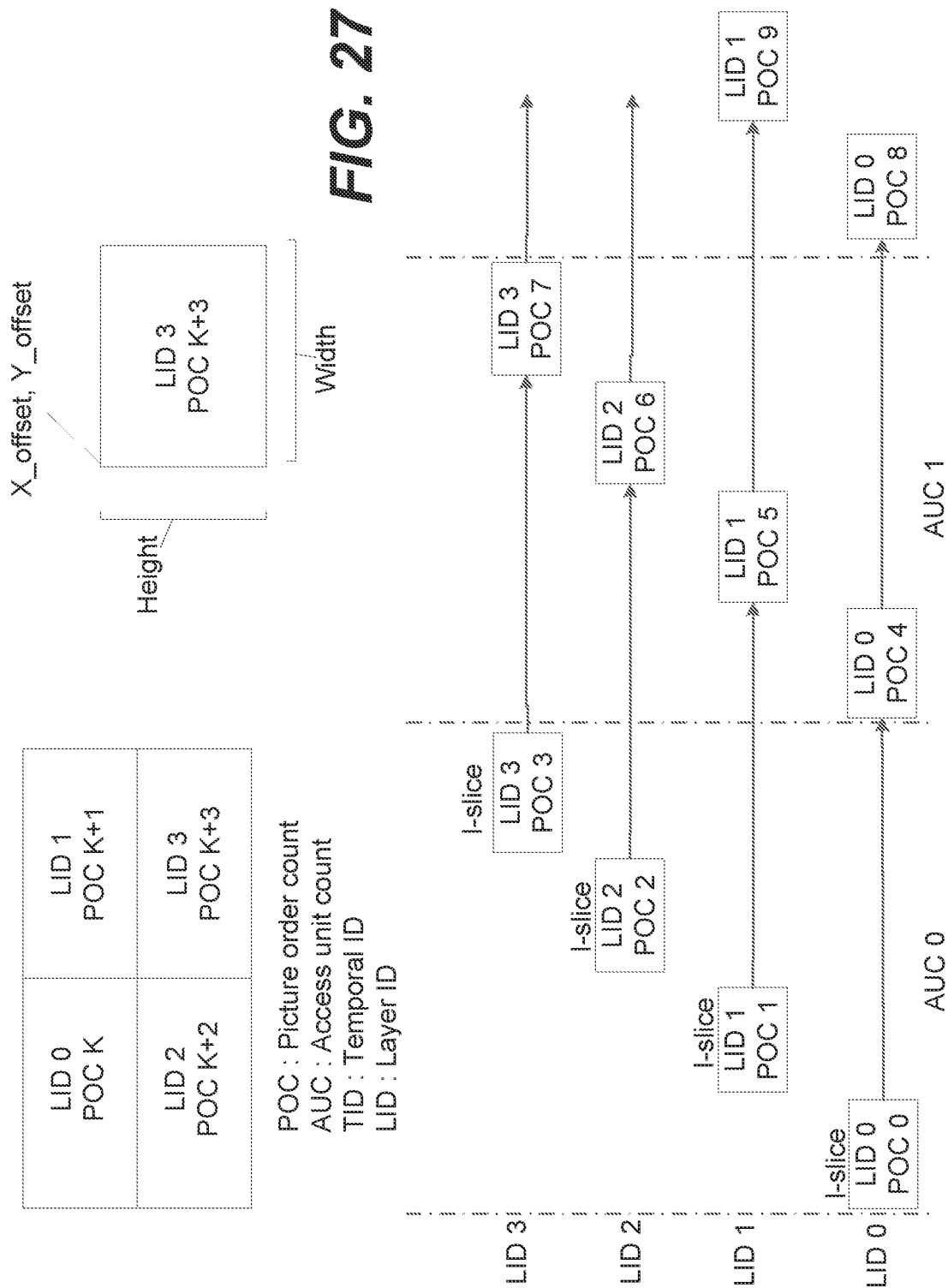
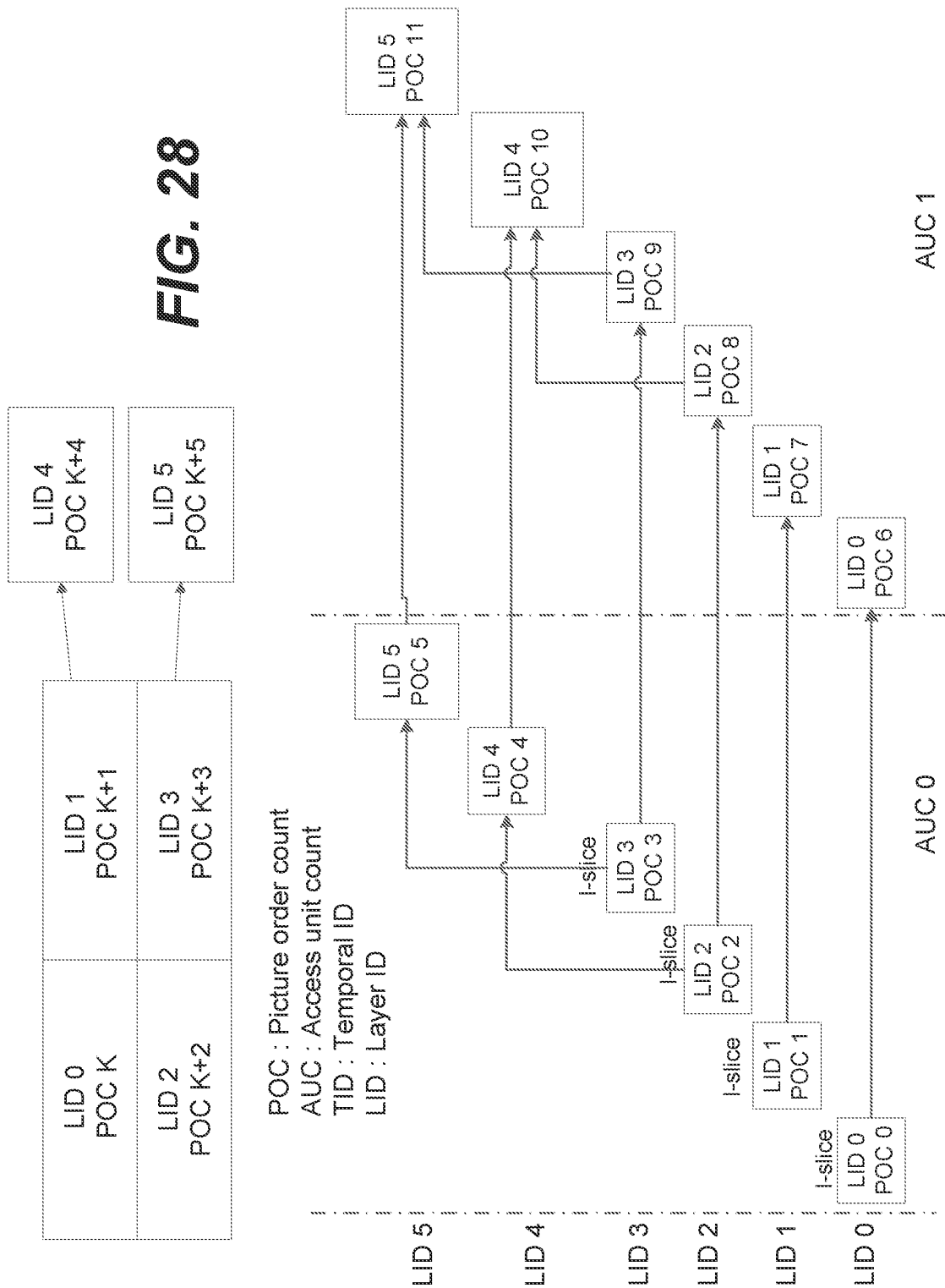


FIG. 26





	Descriptor
video_parameter_set_rbsp() {	
vps_video_parameter_set_id	u(4)
vps_max_layers_minus1	u(8)
for(i = 0; i <= vps_max_layers_minus1; i++) {	4 ²
vps_included_layer_id[i]	u(7)
vps_reserved_zero_bit	u(1)
} 4 ²	4 ²
... 4 ²	4 ²
vps_sub_picture_dividing_flag	u(1)
if(vps_sub_picture_dividing_flag) {	4 ²
vps_full_pic_width_in_luma_samples	ue(v)
vps_full_pic_height_in_luma_samples	ue(v)
}	4 ²
... 4 ²	4 ²
} 4 ²	4 ²

FIG. 29A

Continued from FIG. 29A

seq_parameter_set_rbsp() {	Descriptor
sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
...	⋮
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
if(vps_sub_picture_dividing_flag) {	⋮
pic_offset_x	ue(v)
pic_offset_y	ue(v)
}	⋮
...	⋮
}	⋮

FIG. 29B

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

FIG. 30

	Descriptor
video parameter set rbsp() {	
...	
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. 31

	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)
}	
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(i = 0; i < NumLayersInList[i]; i++) {	
if(vps_output_layers_mode[i] == 2)	
output_layer_flag[i][i]	u(1)
if(vps_ptl_signal_flag[i])	
profile_tier_level_idx[i][i]	u(v)
}	
}	
...	
}	

FIG. 32

FIG. 33

	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[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 < NumLayersInList[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. 34

video_parameter_set_rbsp() {	Descriptor
vps_video_parameter_set_id	u(4)
vps_max_layers_minus1	u(6)
vps_max_sublayers_minus1	u(3)
if(vps_max_layers_minus1 > 0 && vps_max_sublayers_minus1 > 0)	
vps_all_layers_same_num_sublayers_flag	u(1)
if(vps_max_layers_minus1 > 0)	
vps_all_independent_layers_flag	u(1)
for(i = 0; i <= vps_max_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
if(i > 0 && !vps_all_independent_layers_flag) {	
vps_independent_layer_flag[i]	u(1)
if(!vps_independent_layer_flag[i]) {	
for(j = 0; j < i; j++)	
vps_direct_ref_layer_flag[i][j]	u(1)
max_tid_ref_present_flag[i]	u(1)
if(max_tid_ref_present_flag[i])	
max_tid_ref_pics_plus1[i]	u(3)
}	
}	
}	
if(vps_max_layers_minus1 > 0) {	
if(vps_all_independent_layers_flag)	
each_layer_is_an_ols_flag	u(1)
if(!each_layer_is_an_ols_flag) {	
if(!vps_all_independent_layers_flag)	
ols_mode_idc	u(2)
if(ols_mode_idc == 2) {	
num_output_layer_sets_minus1	u(8)
for(i = 1; i <= num_output_layer_sets_minus1; i++)	
for(j = 0; j <= vps_max_layers_minus1; j++)	
ols_output_layer_flag[i][j]	u(1)
}	
}	
}	
}	
}	
}	

FIG. 35

video parameter set rbsp() {	Descriptor
vps video parameter set id	u(4)
vps max layers minus1	u(6)
vps max sublayers minus1	u(3)
if(vps max layers minus1 > 0 && vps max sublayers minus1 > 0)	
vps all layers same num sublayers flag	u(1)
if(vps max layers minus1 > 0)	
vps all independent layers flag	u(1)
for(i = 0; i <= vps max layers minus1; i++) {	
vps layer id[i]	u(6)
if(i > 0 && !vps all independent layers flag) {	
vps independent layer flag[i]	u(1)
if(!vps independent layer flag[i]) {	
for(j = 0; j < i; j++)	
vps direct ref layer flag[i][j]	u(1)
max tid ref present flag[i]	u(1)
if(max tid ref present flag[i])	
max tid il ref pics plus1[i]	u(3)
}	
}	
}	
if(vps max layers minus1 > 0) {	
if(vps all independent layers flag)	
each layer is an ols flag	u(1)
if(!each layer is an ols flag) {	
if(!vps all independent layers flag)	
ols mode idc	u(2)
if(ols mode idc == 2 && vps max layers minus1 > 1) {	
num output layer sets minus1	u(8)
for(i = 1; i <= num output layer sets minus1; i++)	
for(j = 0; j <= vps max layers minus1; j++)	
ols output layer flag[i][j]	u(1)
}	
}	
}	
}	
...	
}	

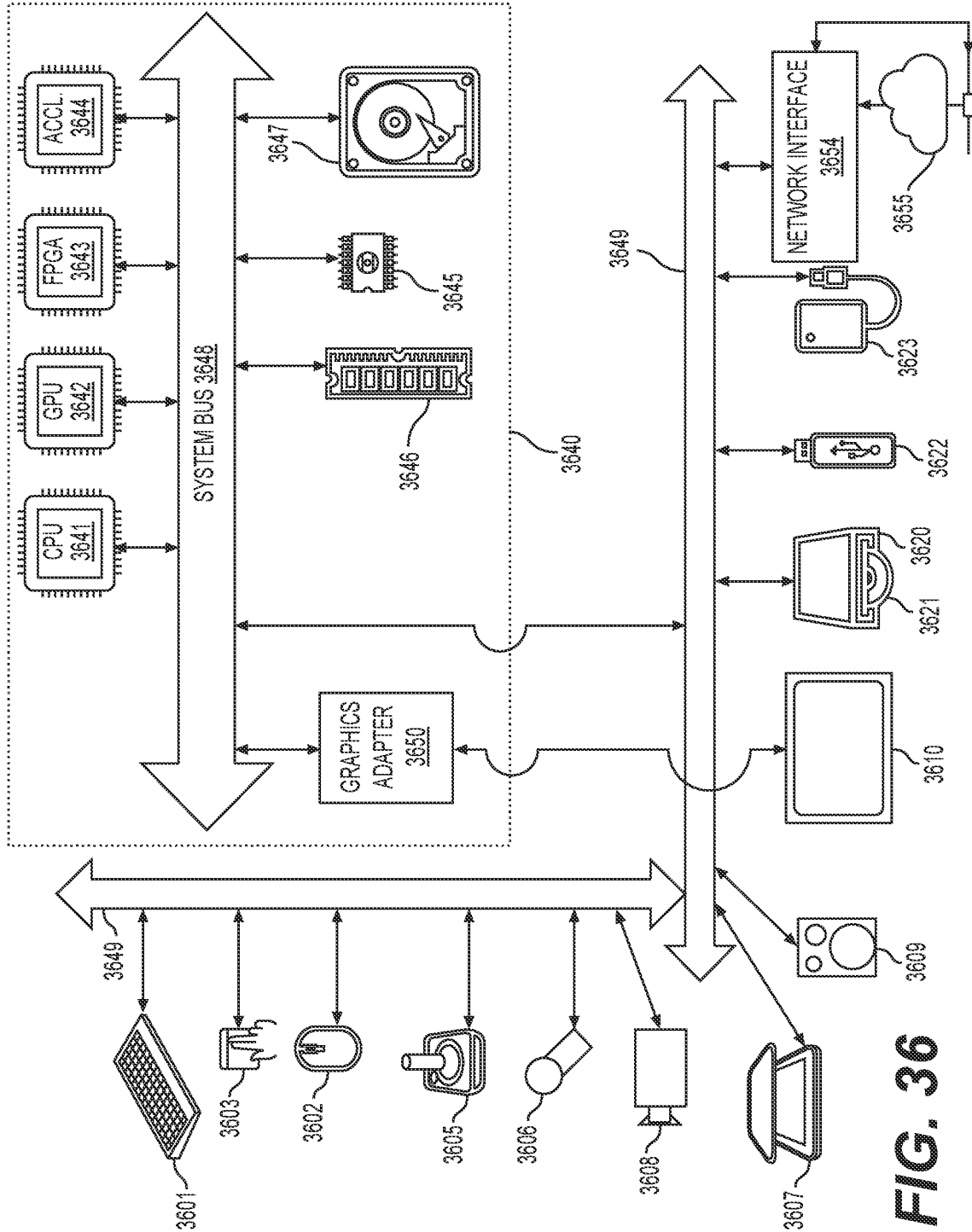


FIG. 36