

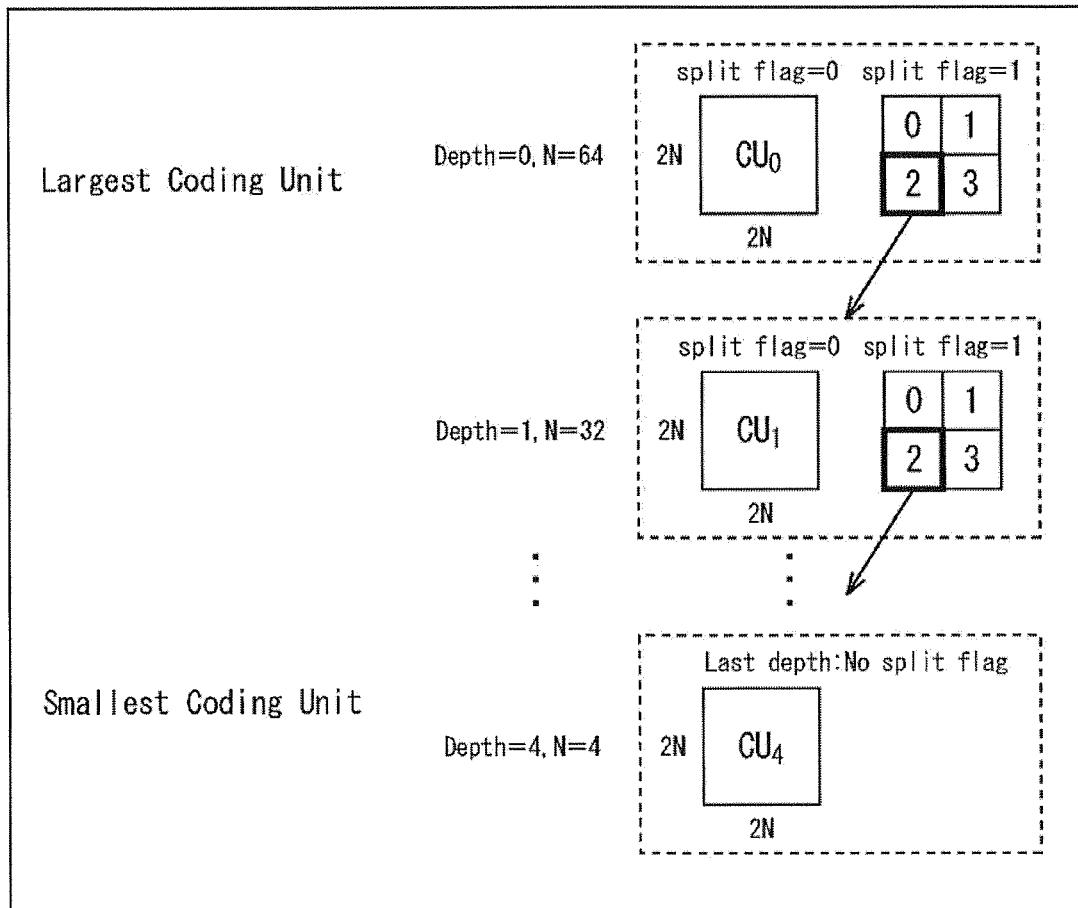
FIG. 1

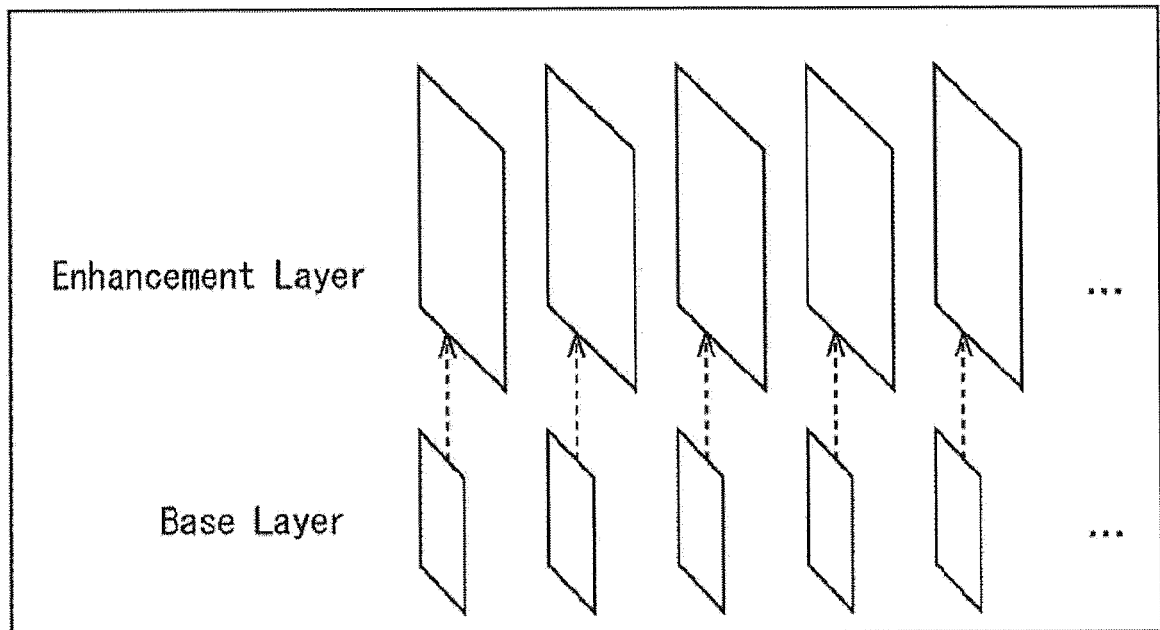
FIG. 2

FIG. 3

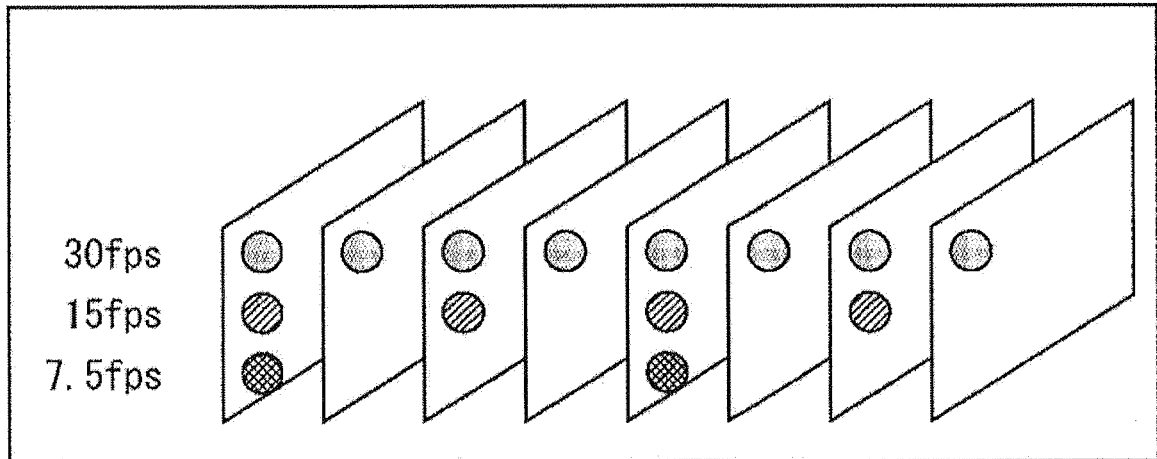


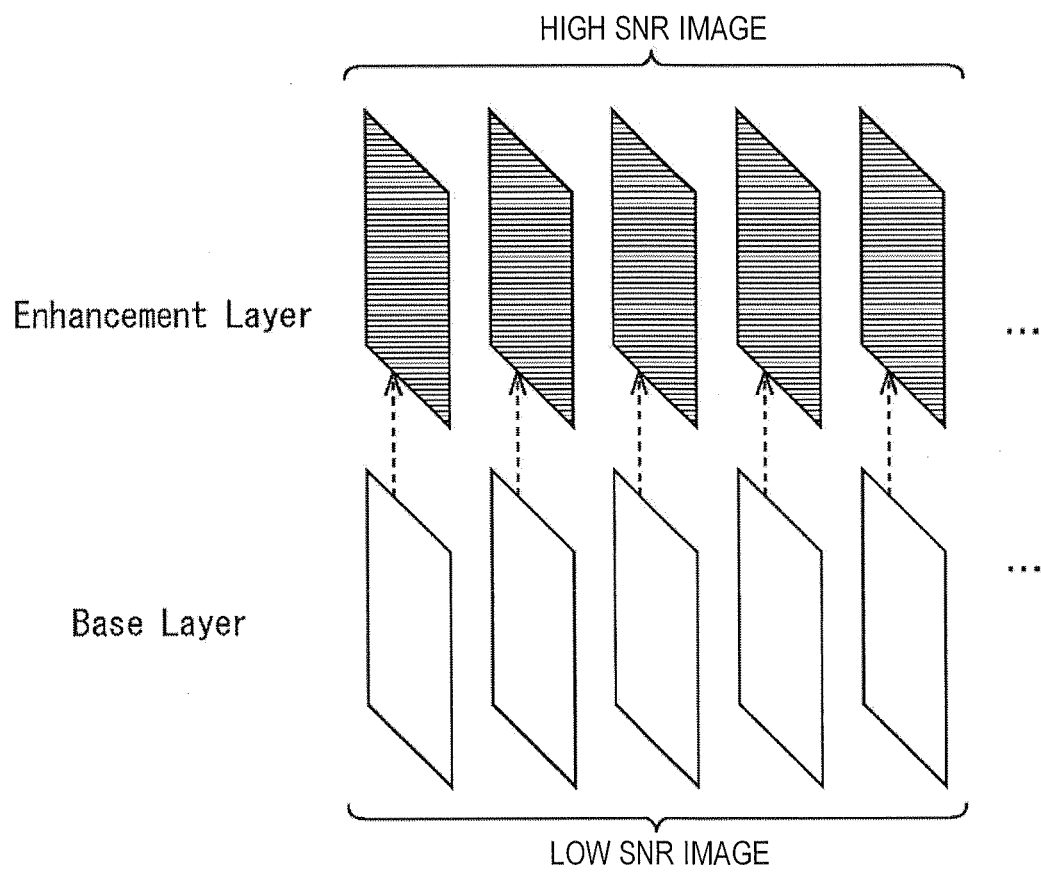
FIG. 4

FIG. 5

1	hrd_parameters(commonInfPresentFlag, MaxNumSubLayersMinus1) {	Descriptor
2	if(commonInfPresentFlag) {	
3	timing_info_present_flag	u(1)
4	if(timing_info_present_flag) {	
5	num_units_in_tick	u(32)
6	time_scale	u(32)
7	}	
8	nal_hrd_parameters_present_flag	u(1)
9	vcl_hrd_parameters_present_flag	u(1)
10	if(nal_hrd_parameters_present_flag vcl_hrd_parameters_present_flag) {	
11	sub_pic_cpb_params_present_flag	u(1)
12	if(sub_pic_cpb_params_present_flag) {	
13	tick_divisor_minus2	u(8)
14	du_cpb_removal_delay_length_minus1	u(5)
15	}	
16	bit_rate_scale	u(4)
17	cpb_size_scale	u(4)
18	initial_cpb_removal_delay_length_minus1	u(5)
19	cpb_removal_delay_length_minus1	u(5)
20	dpb_output_delay_length_minus1	u(5)
21	}	
22	}	
23	for(i=0; i<=MaxNumSubLayersMinus1; i++) {	
24	fixed_pic_rate_flag[i]	u(1)
25	if(fixed_pic_rate_flag[i])	
26	pic_duration_in_tc_minus1[i]	ue(v)
27	low_delay_hrd_flag[i]	u(1)
28	cpb_cnt_minus1[i]	ue(v)
29	if(nal_hrd_parameters_present_flag)	
30	sub_layer_hrd_parameters(i)	
31	if(vcl_hrd_parameters_present_flag)	
32	sub_layer_hrd_parameters(i)	
33	}	
34	}	

FIG. 6

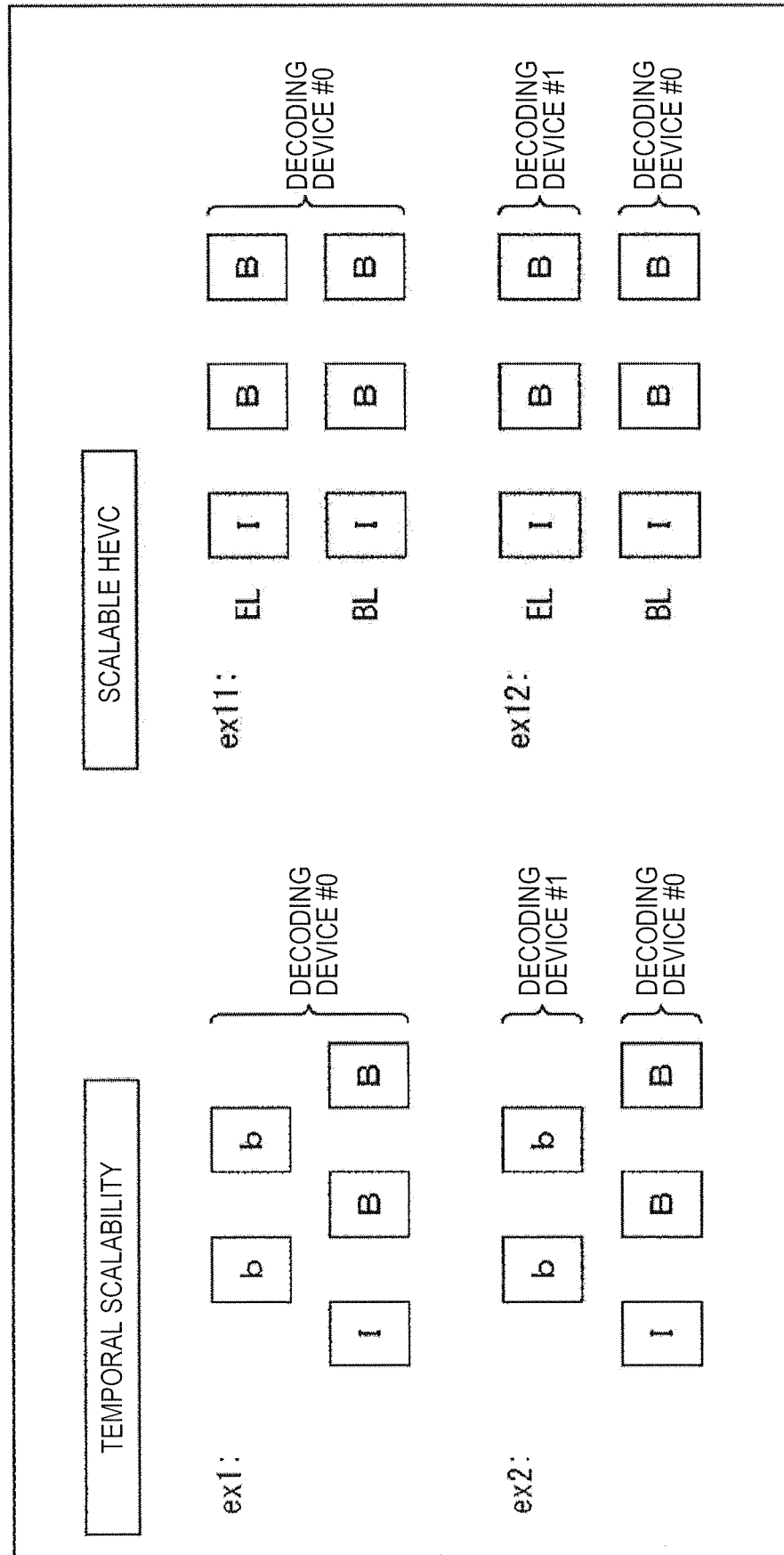


FIG. 7

1	hrd_parameters(commonInfPresentFlag, MaxNumSubLayersMinus1) {	Descriptor
2	if(commonInfPresentFlag) {	
3	timing_info_present_flag	u(1)
4	if(timing_info_present_flag) {	
5	num_units_in_tick	u(32)
6	time_scale	u(32)
7	}	
8	nal_hrd_parameters_present_flag	u(1)
9	vcl_hrd_parameters_present_flag	u(1)
10	hrd_parameters_type_flag	u(1)
11	if(nal_hrd_parameters_present_flag vcl_hrd_parameters_present_flag) {	
12	sub_pic_cpb_params_present_flag	u(1)
13	if(sub_pic_cpb_params_present_flag) {	
14	tick_divisor_minus2	u(8)
15	du_cpb_removal_delay_length_minus1	u(5)
16	}	
17	bit_rate_scale	u(4)
18	cpb_size_scale	u(4)
19	initial_cpb_removal_delay_length_minus1	u(5)
20	cpb_removal_delay_length_minus1	u(5)
21	dpb_output_delay_length_minus1	u(5)
22	}	
23	}	
24	for (i=0; i<=MaxNumSubLayersMinus1; i++) {	
25	sub_hrd_parameters_type_flag[i]	u(1)
26	fixed_pic_rate_flag[i]	u(1)
27	if(fixed_pic_rate_flag[i])	
28	pic_duration_in_tc_minus1[i]	ue(v)
29	low_delay_hrd_flag[i]	u(1)
30	cpb_ent_minus1[i]	ue(v)
31	if(nal_hrd_parameters_present_flag)	
32	sub_layer_hrd_parameters(i)	
33	if(vcl_hrd_parameters_present_flag)	
34	sub_layer_hrd_parameters(i)	
35	}	
36	}	

8/56
FIG. 8

1	hrd_parameters(commonInfPresentFlag, MaxNumSubLayersMinus1) {	Descriptor
2	if(commonInfPresentFlag) {	
3	timing_info_present_flag	u(1)
4	if(timing_info_present_flag) {	
5	num_units_in_tick	u(32)
6	time_scale	u(32)
7	}	
8	nal_hrd_parameters_present_flag	u(1)
9	vcl_hrd_parameters_present_flag	u(1)
10	if(nal_hrd_parameters_present_flag vcl_hrd_parameters_present_flag) {	
11	hrd_parameters_type1_present_flag	u(1)
12	hrd_parameters_type2_present_flag	u(1)
13	if(hrd_parameters_type1_present_flag) {	
14	type1_sub_pic_cpb_params_present_flag	u(1)
15	if(type1_sub_pic_cpb_params_present_flag) {	
16	type1_tick_divisor_minus2	u(8)
17	type1_du_cpb_removal_delay_length_minus1	u(5)
18	}	
19	type1_bit_rate_scale	u(4)
20	type1_cpb_size_scale	u(4)
21	type1_initial_cpb_removal_delay_length_minus1	u(5)
22	type1_cpb_removal_delay_length_minus1	u(5)
23	type1_dpb_output_delay_length_minus1	u(5)
24	}	
25	if(hrd_parameters_type2_present_flag) {	
26	type2_sub_pic_cpb_params_present_flag	u(1)
27	if(type2_sub_pic_cpb_params_present_flag) {	
28	type2_tick_divisor_minus2	u(8)
29	type2_du_cpb_removal_delay_length_minus1	u(5)
30	}	
31	type2_bit_rate_scale	u(4)
32	type2_cpb_size_scale	u(4)
33	type2_initial_cpb_removal_delay_length_minus1	u(5)
34	type2_cpb_removal_delay_length_minus1	u(5)
35	type2_dpb_output_delay_length_minus1	u(5)
36	}	
37	}	
38	}	
39	for(i=0; i<=MaxNumSubLayersMinus1; i++) {	
40	sub_hrd_parameters_type1_present_flag[i]	u(1)
41	sub_hrd_parameters_type2_present_flag[i]	u(1)
42	fixed_pic_rate_flag[i]	u(1)
43	if(fixed_pic_rate_flag[i])	
44	pic_duration_in_to_minus1[i]	ue(v)
45	if(sub_hrd_parameters_type1_present_flag[i]) {	
46	type1_low_delay_hrd_flag[i]	u(1)
47	type1_cpb_cnt_minus1[i]	ue(v)

FIG. 9

48	if(nal_hrd_parameters_present_flag)	
49	sub_layer_hrd_parameters(i)	
50	if(vcl_hrd_parameters_present_flag)	
51	sub_layer_hrd_parameters(i)	
52	}	
53	if(sub_hrd_parameters_type2_present_flag[i]){	
54	type2_low_delay_hrd_flag[i]	u(1)
55	type2_cpb_cnt_minus1[i]	ue(v)
56	if(nal_hrd_parameters_present_flag)	
57	sub_layer_hrd_parameters(i)	
58	if(vcl_hrd_parameters_present_flag)	
59	sub_layer_hrd_parameters(i)	
60	}	
61	}	
62	}	

FIG. 10

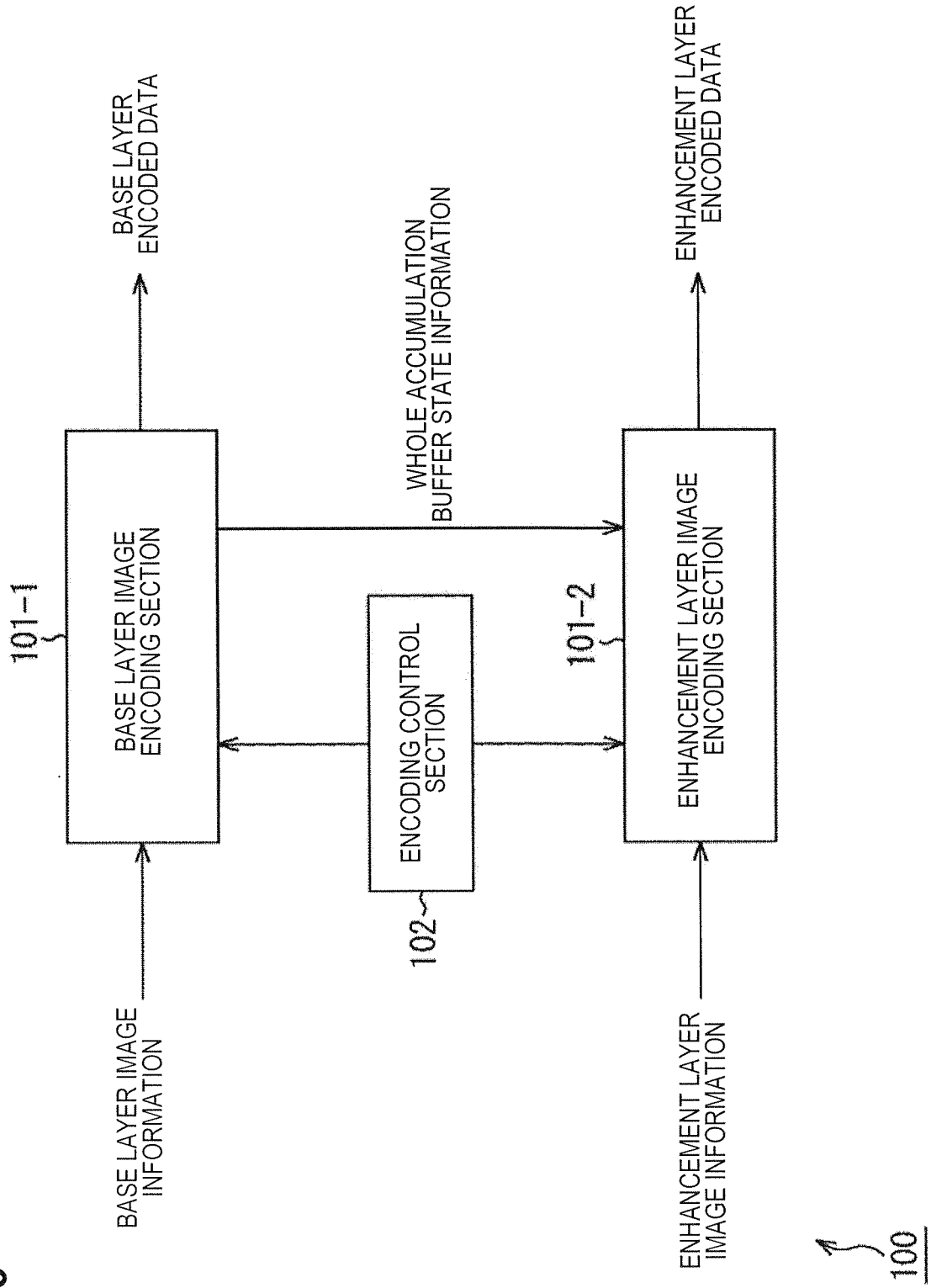


FIG. 12

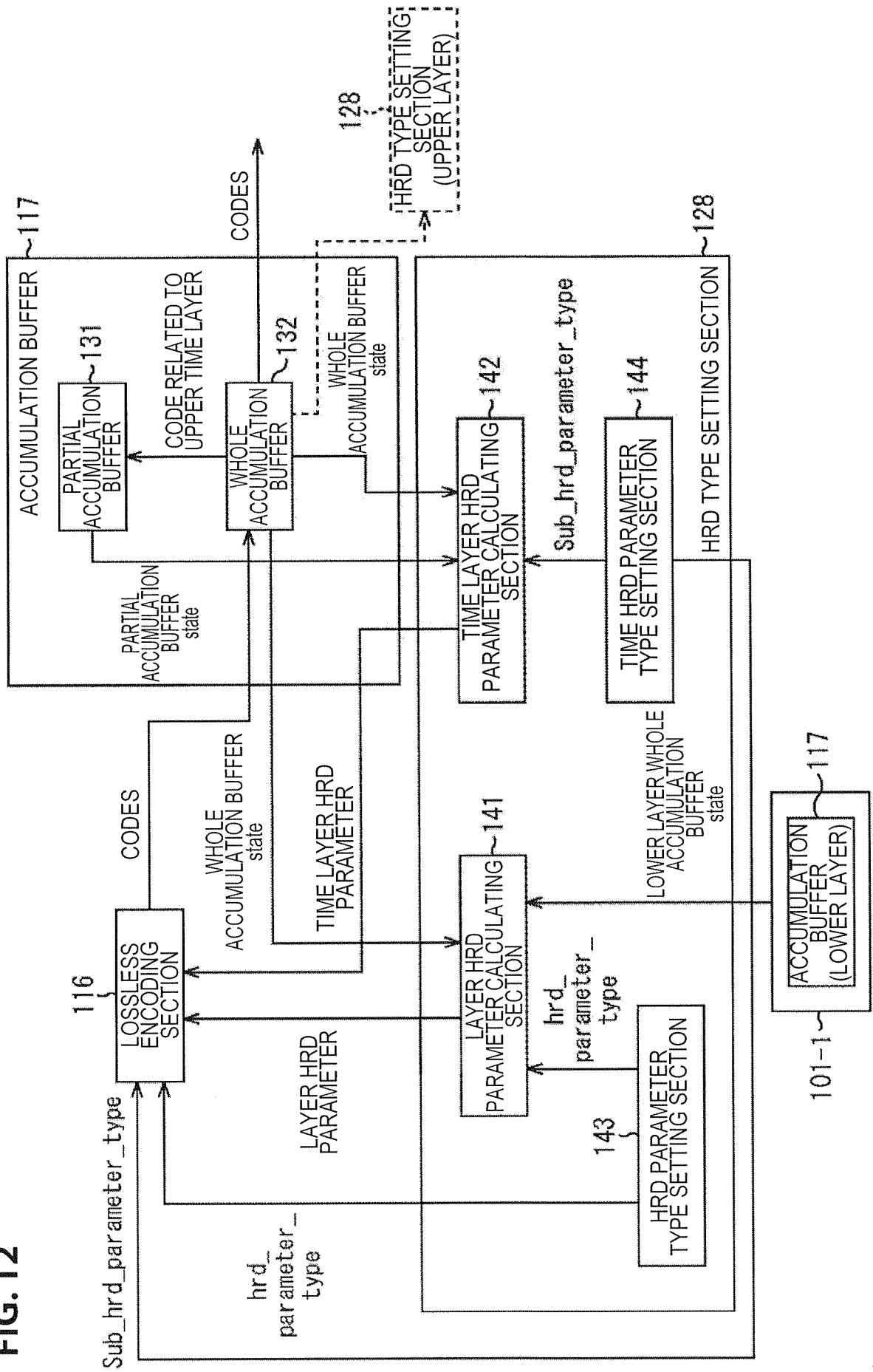


FIG. 13

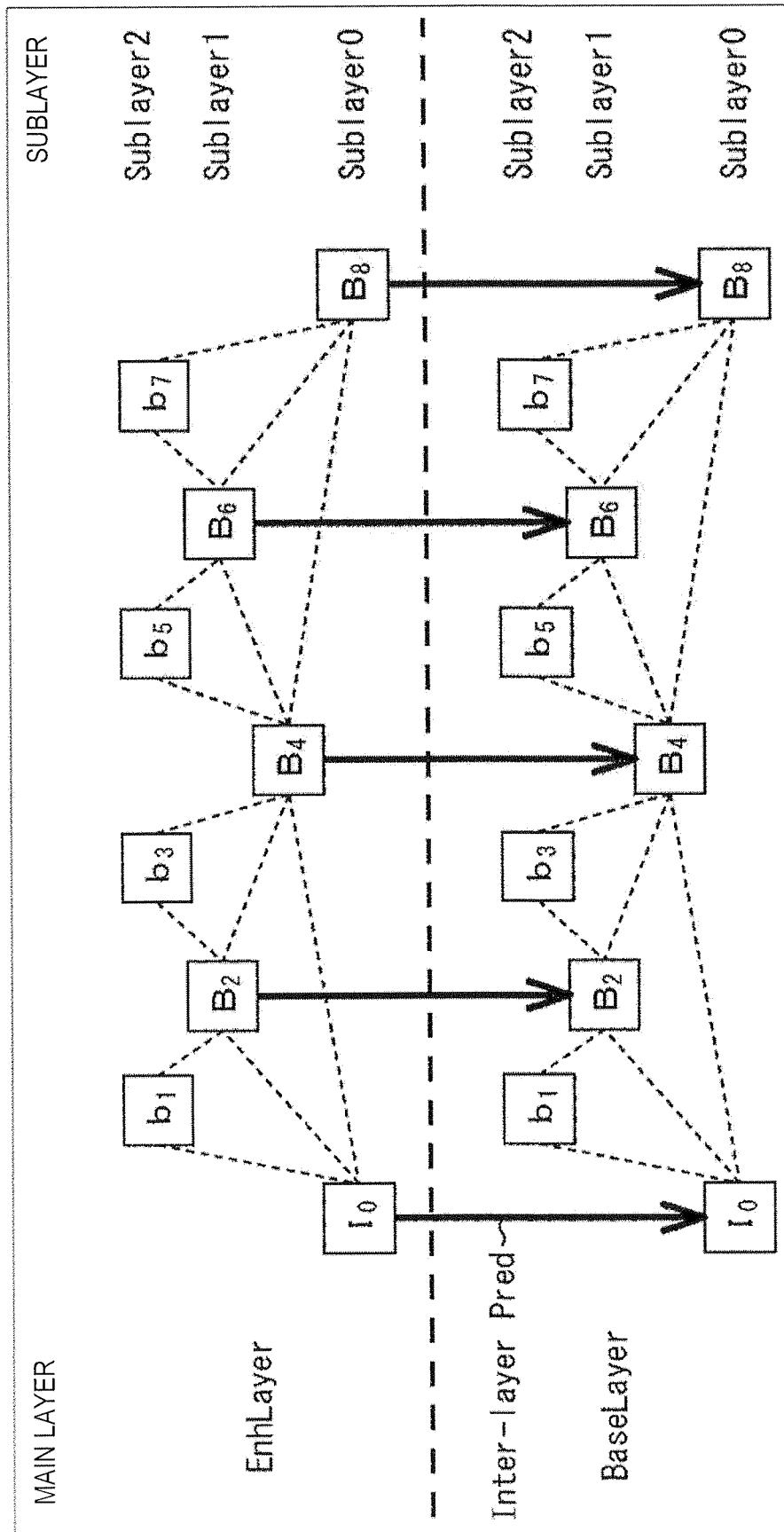


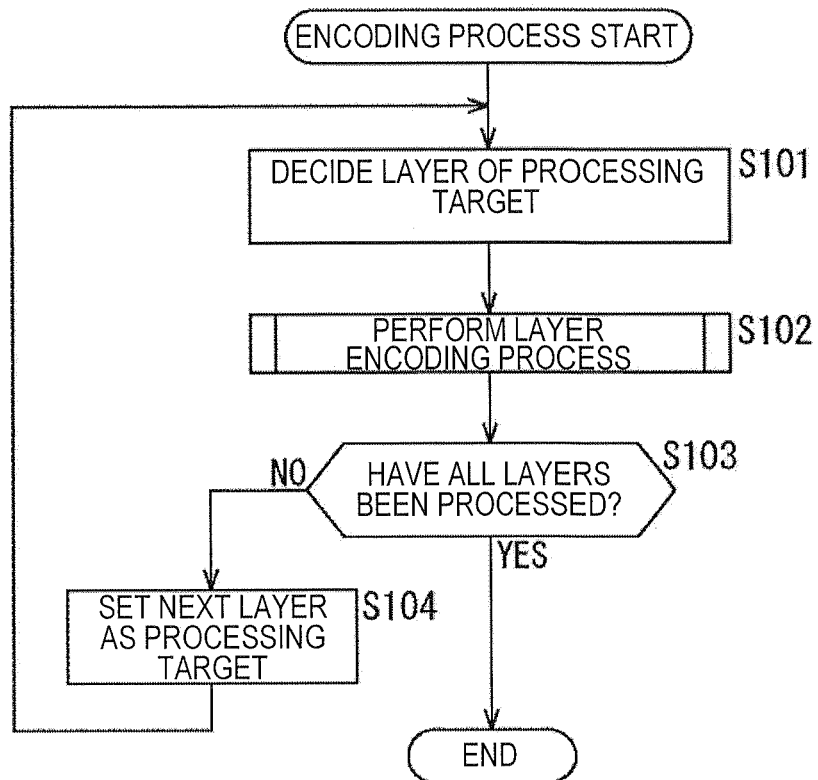
FIG. 14

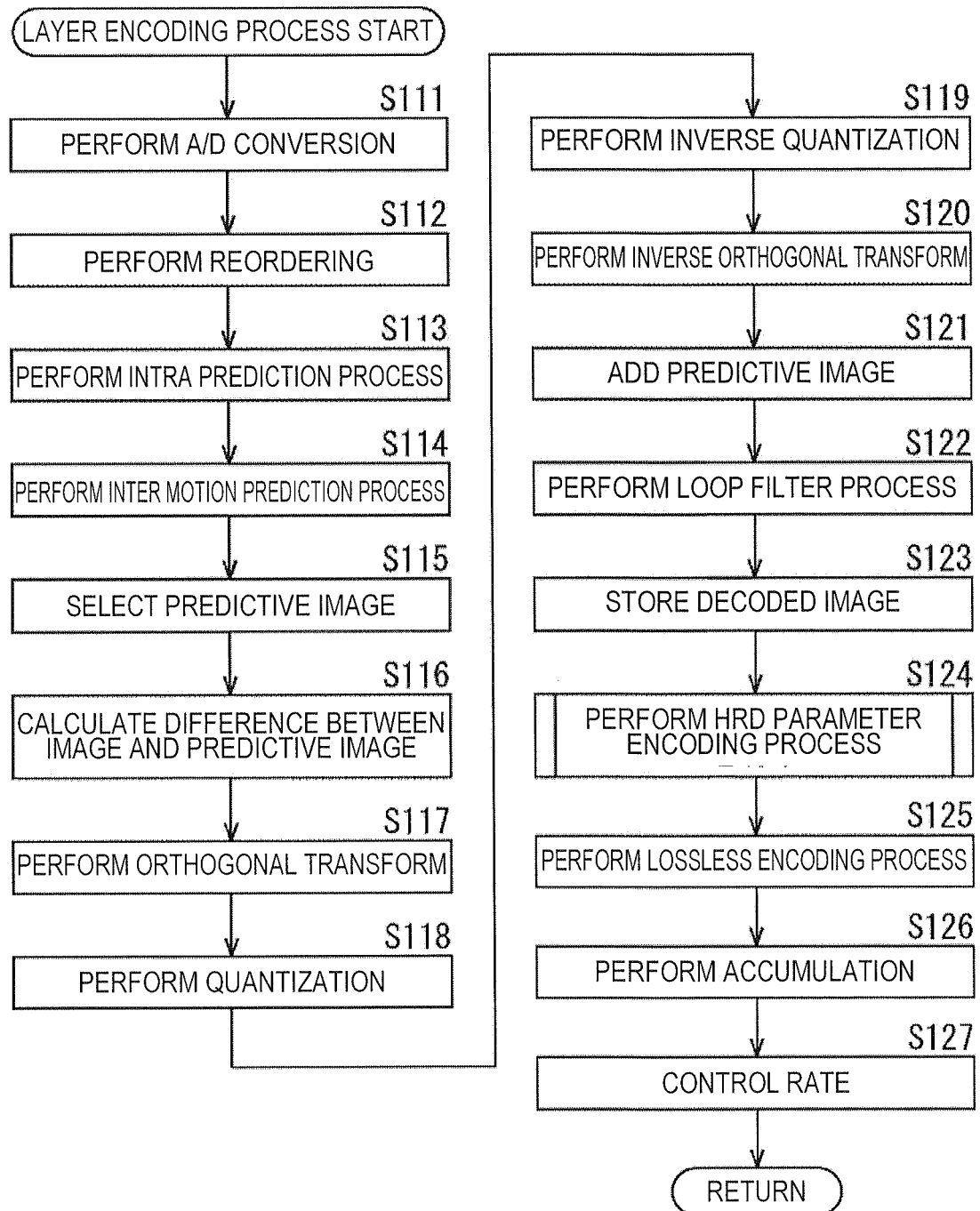
FIG. 15

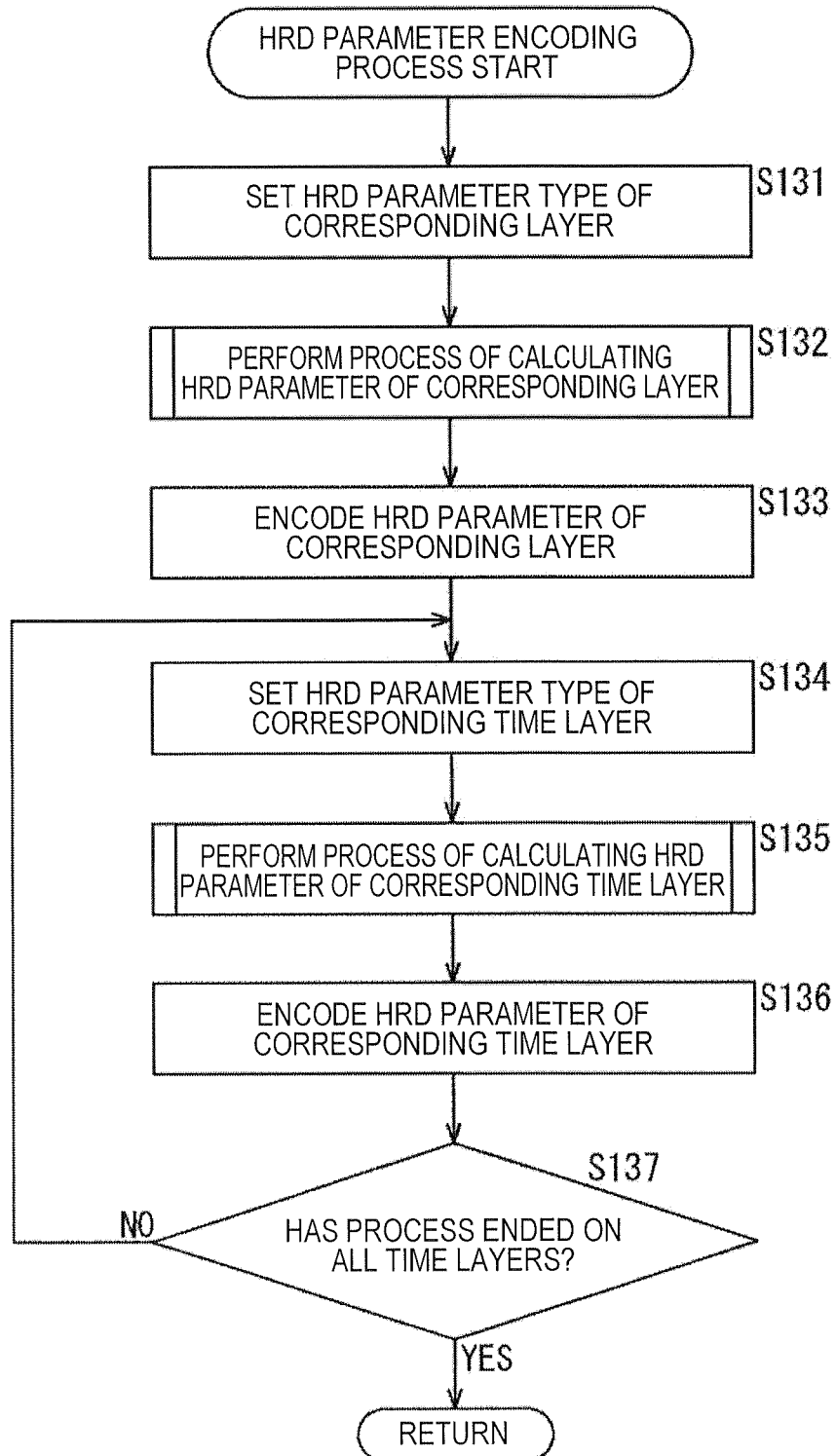
FIG. 16

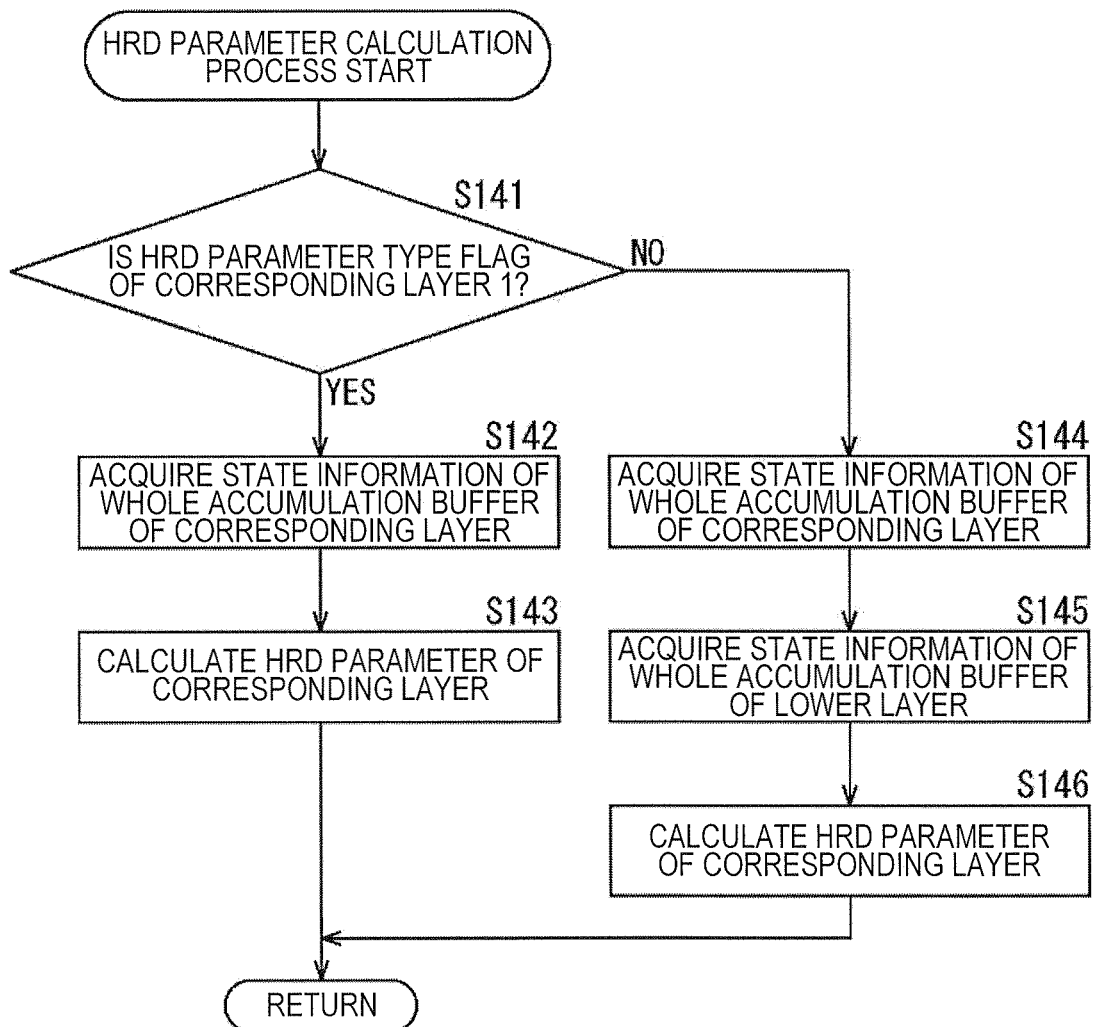
FIG. 17

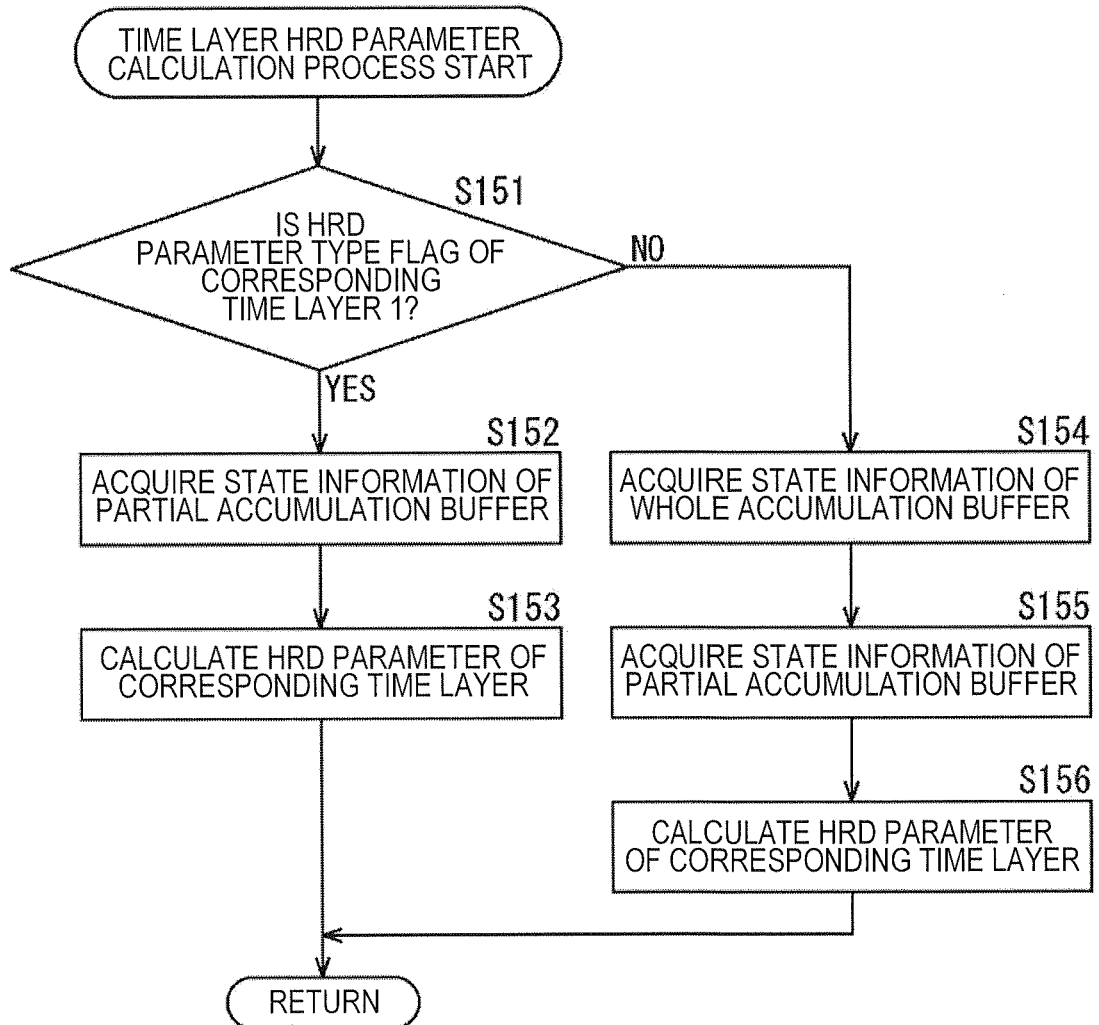
FIG. 18

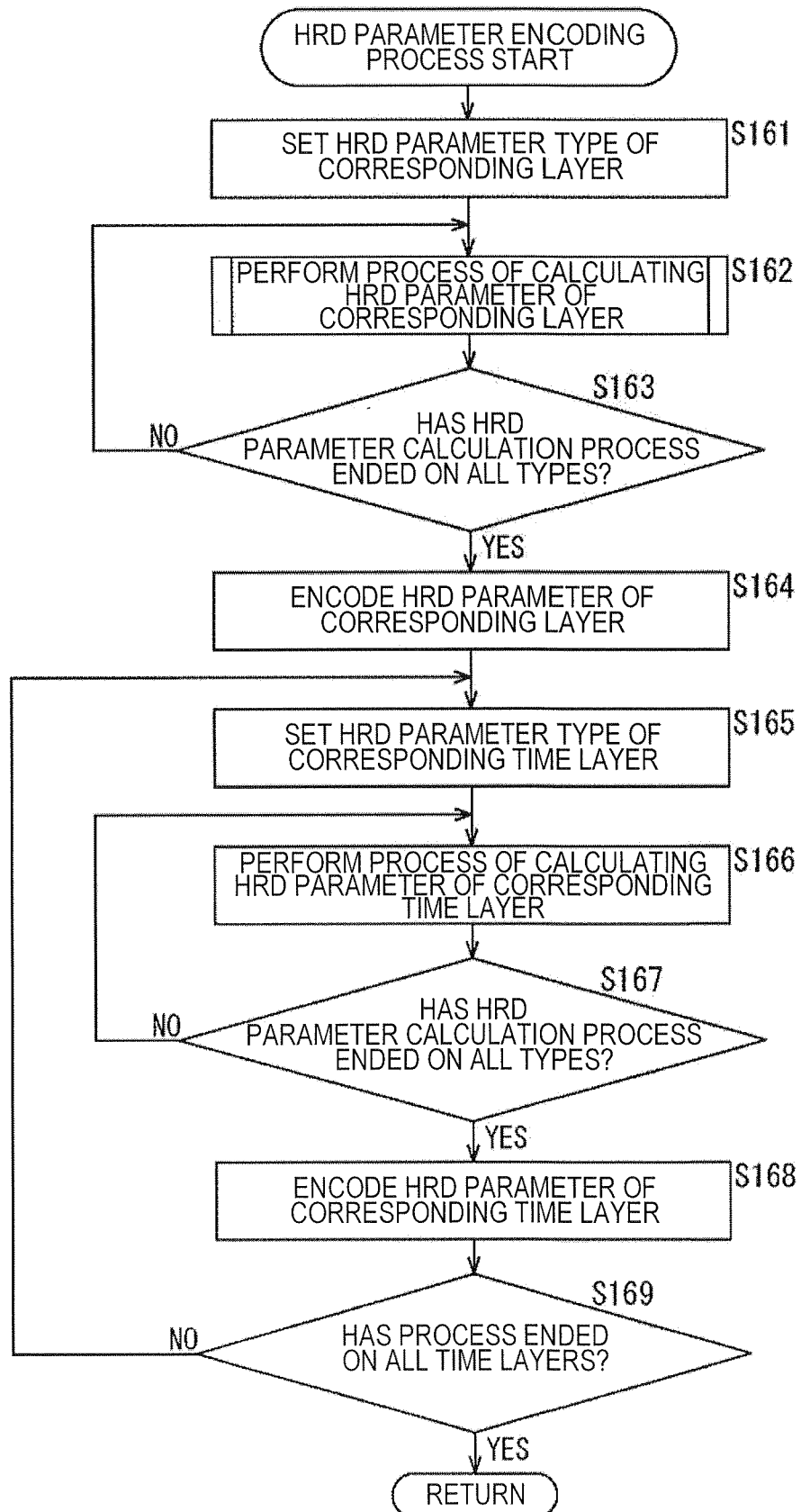
FIG. 19

FIG. 20

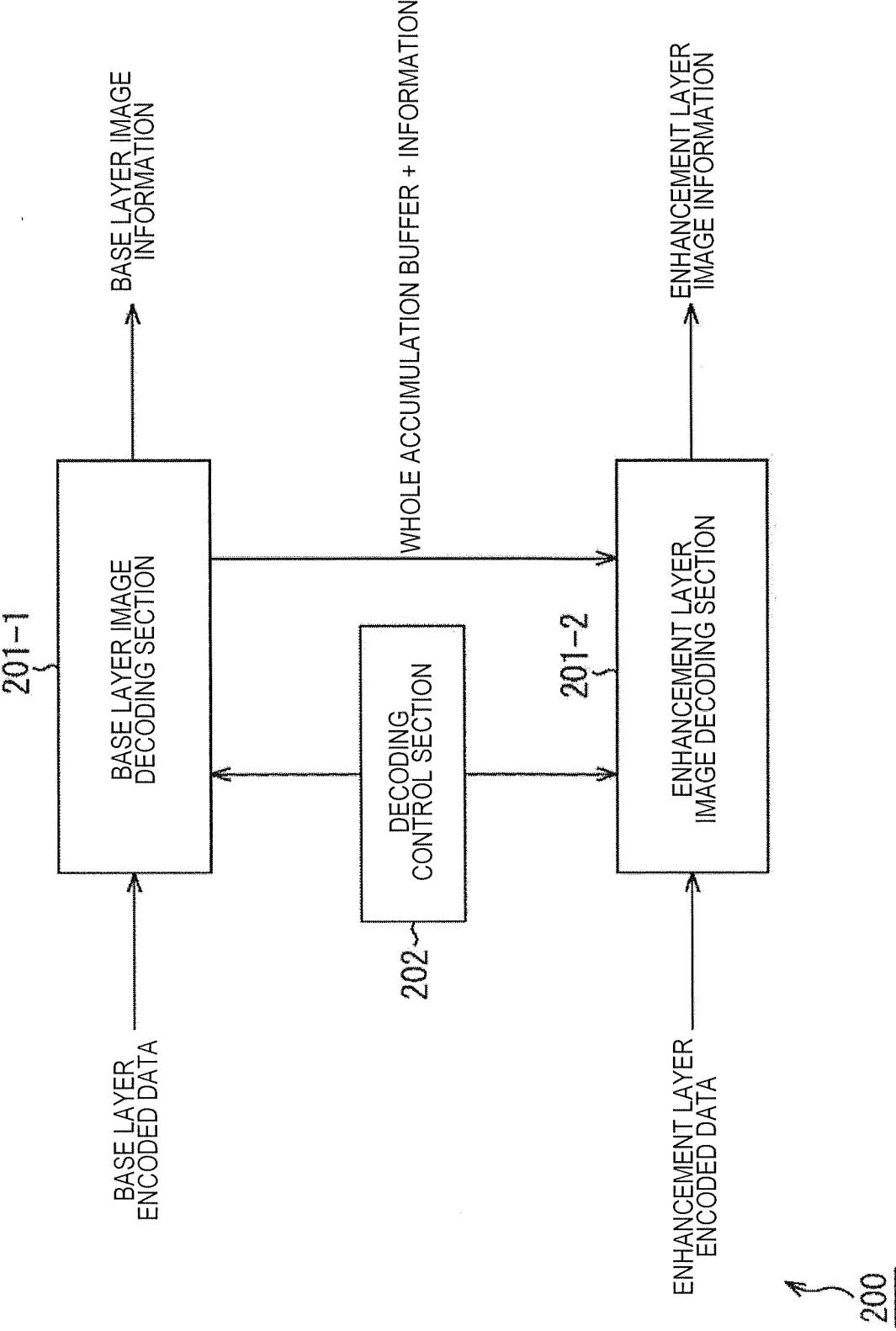


FIG. 21

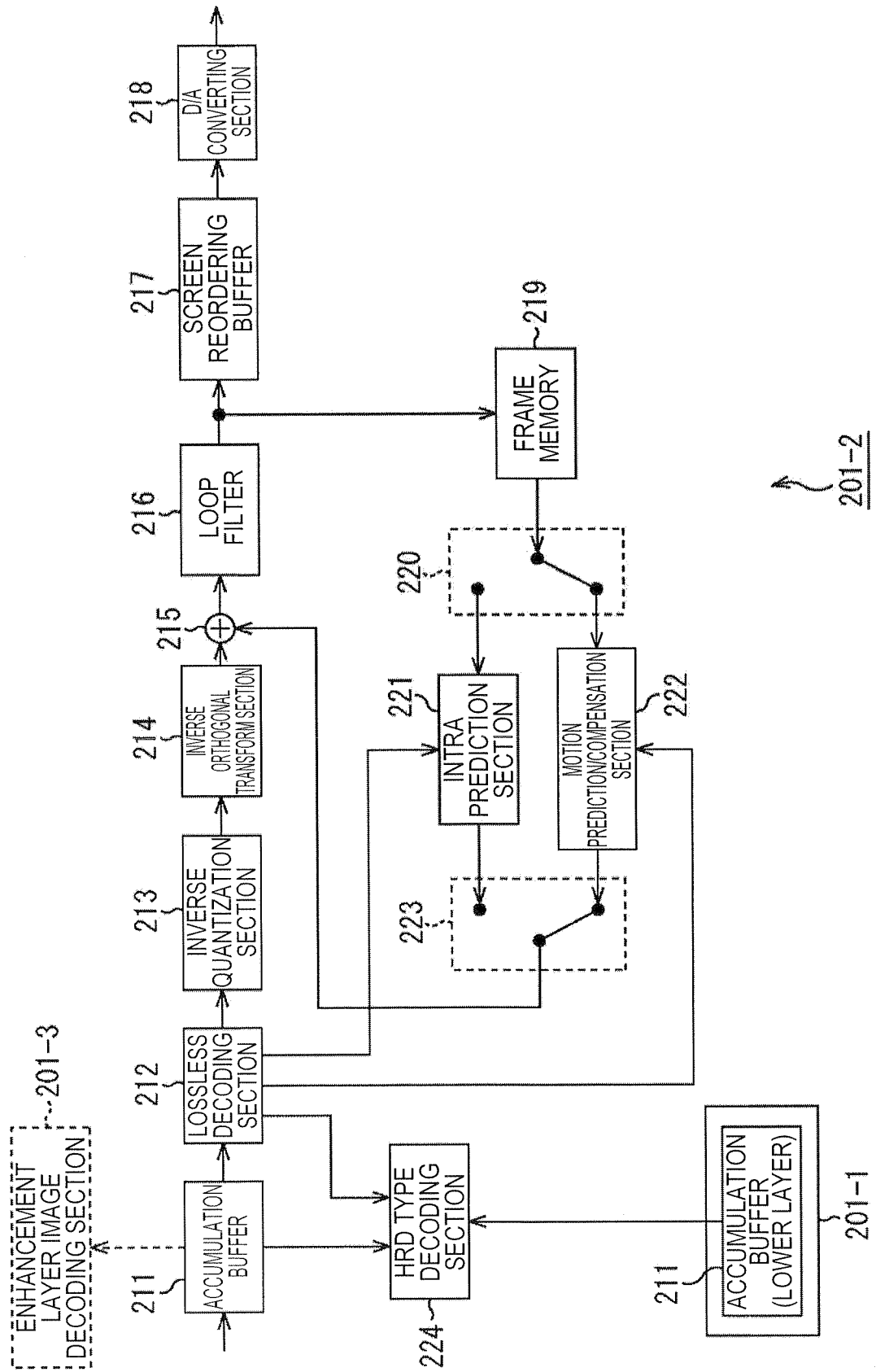


FIG. 22

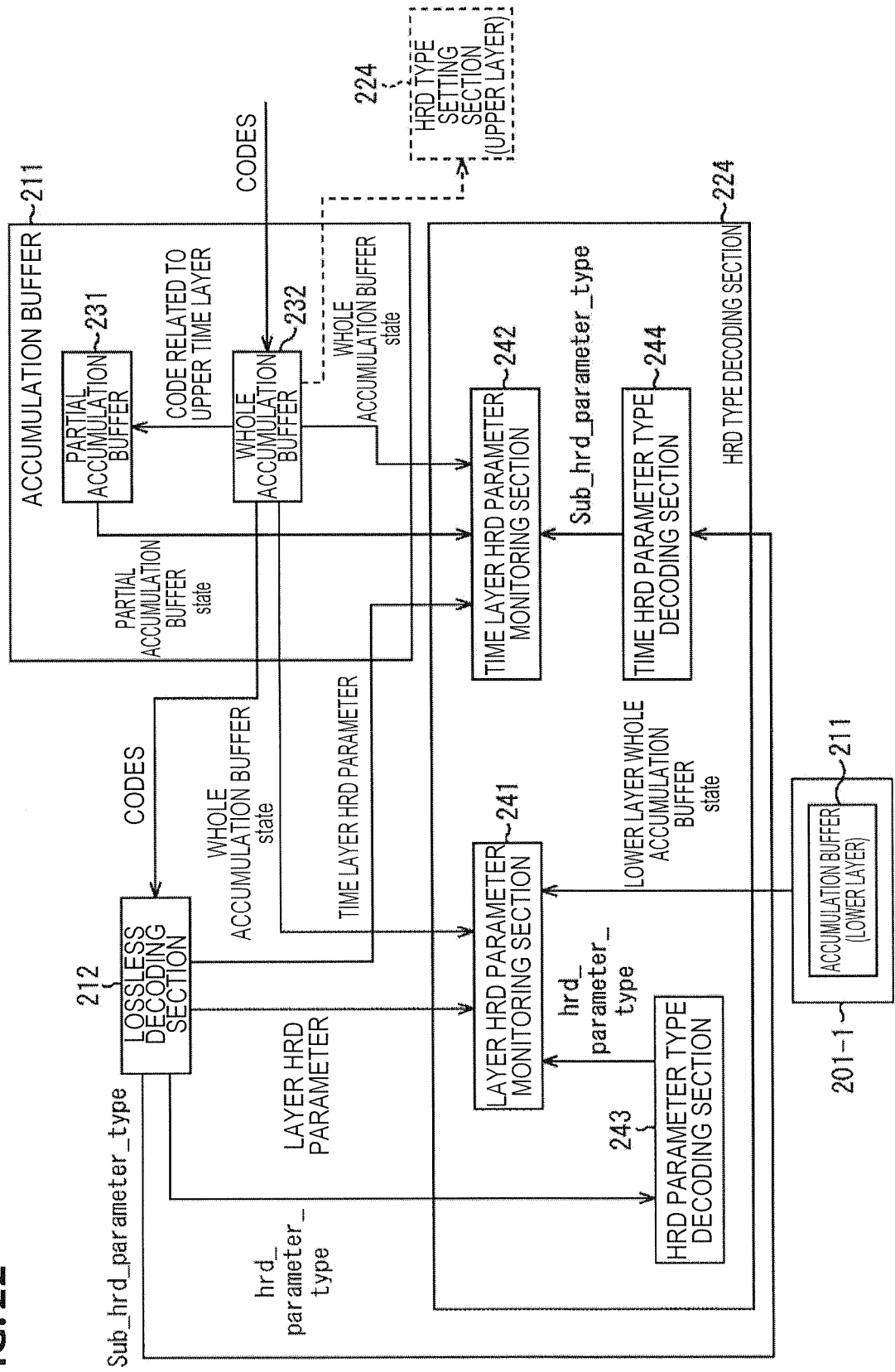


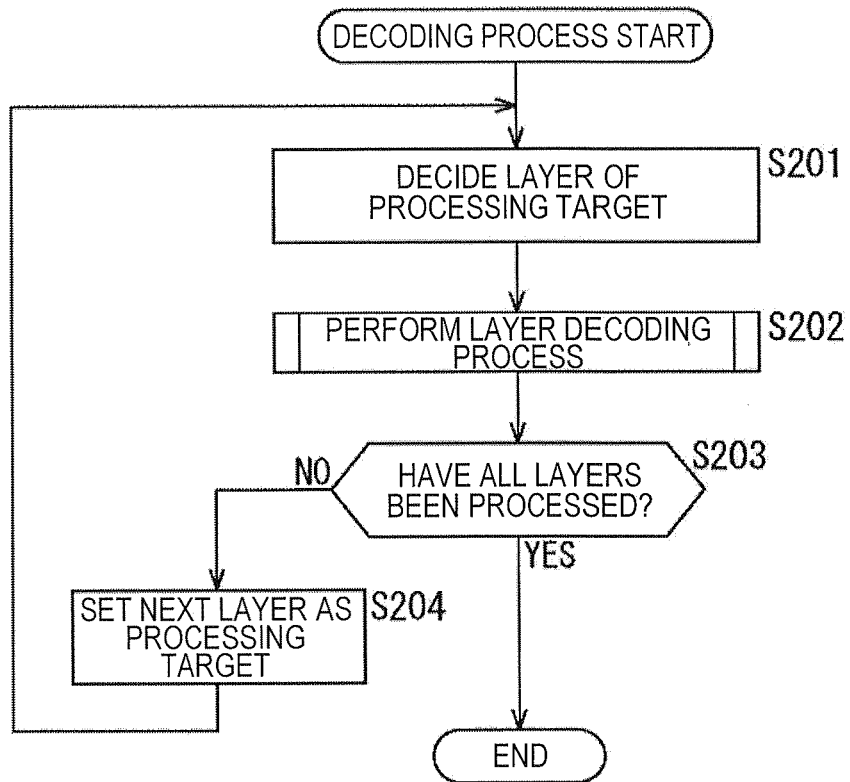
FIG. 23

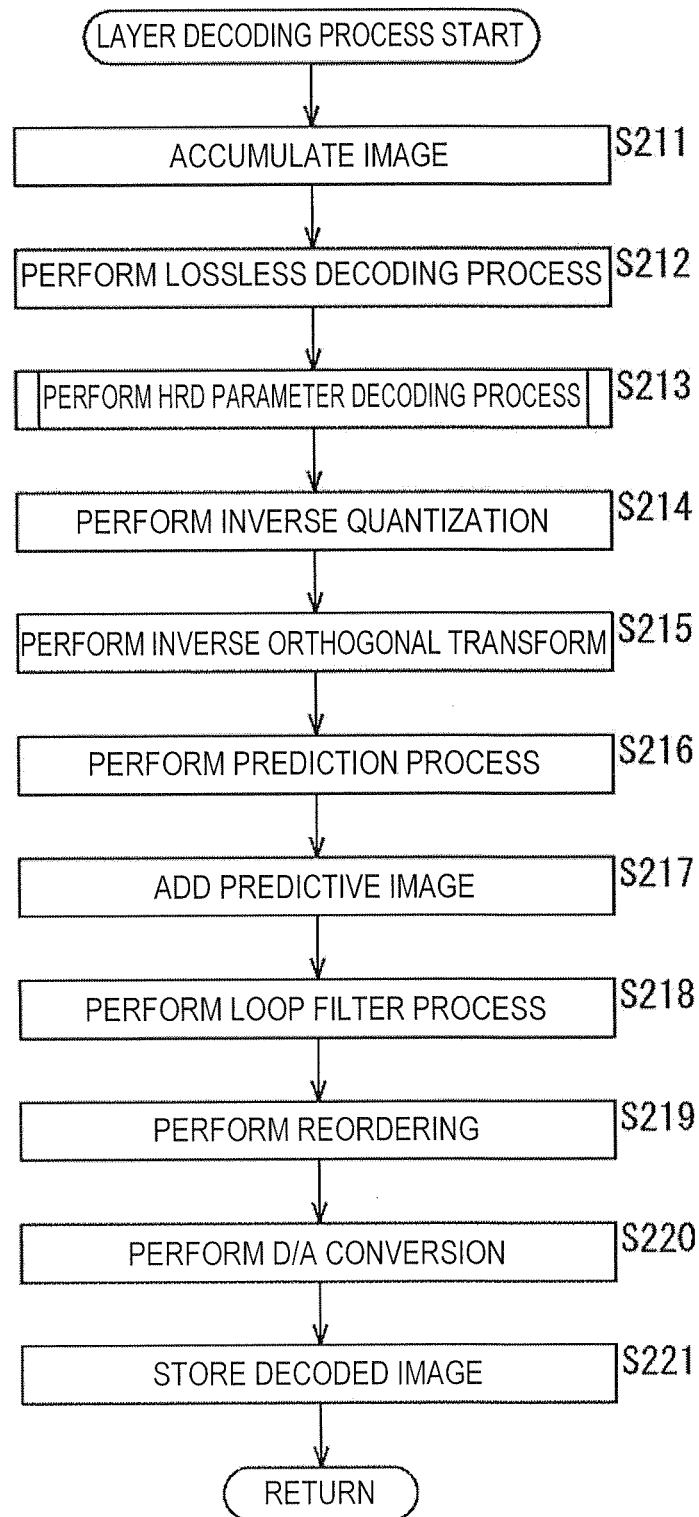
FIG. 24

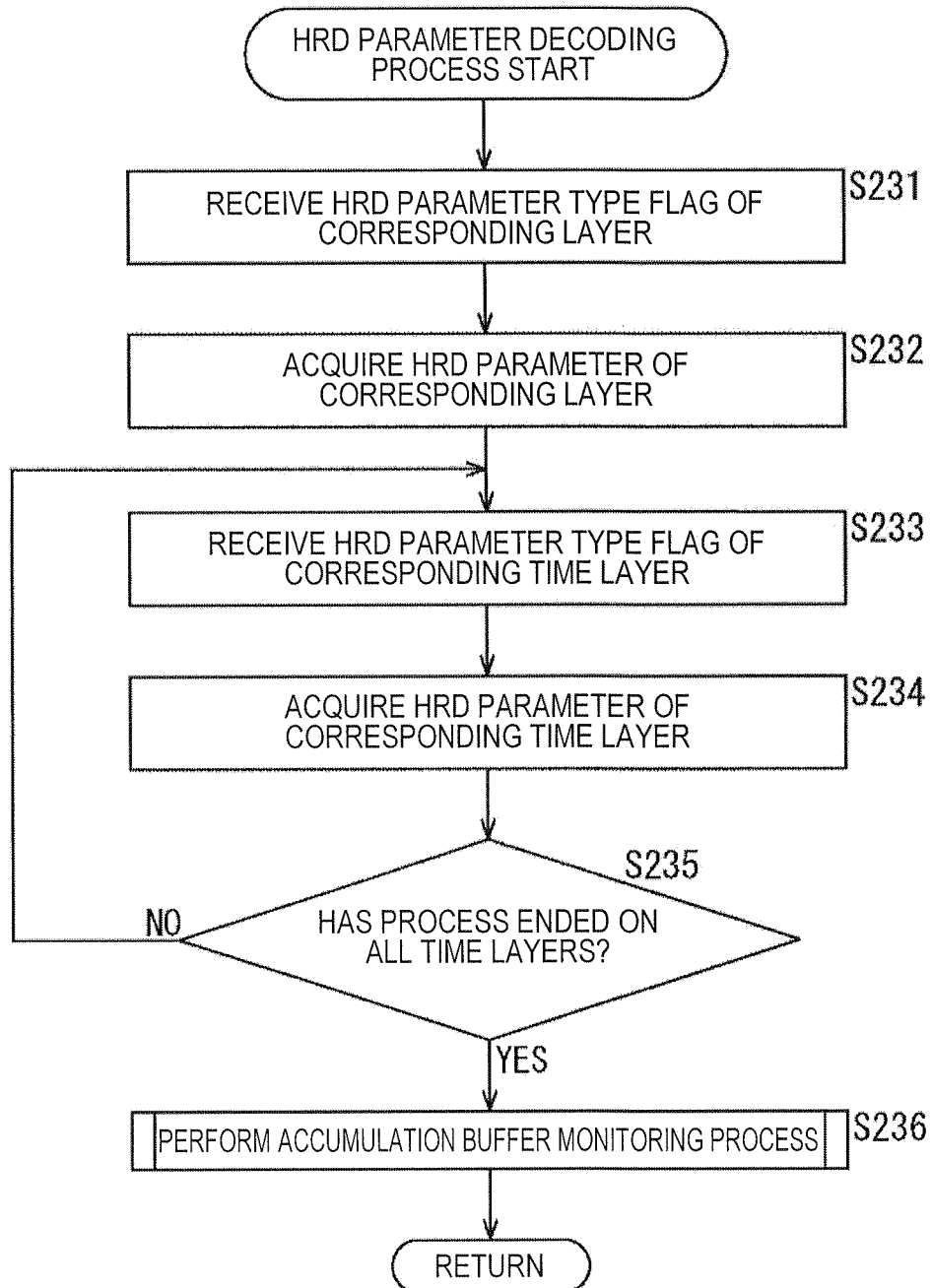
FIG. 25

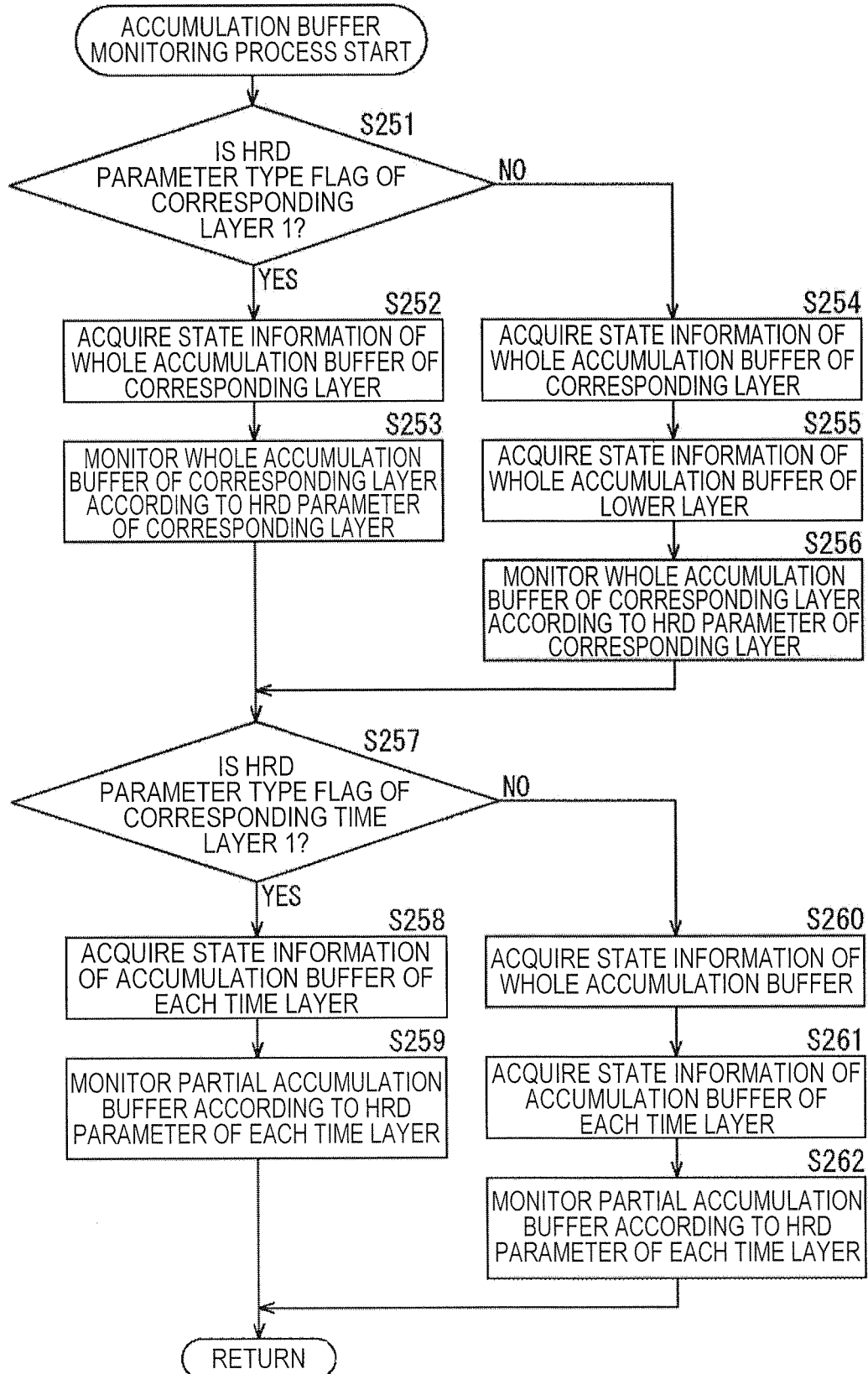
FIG. 26

FIG. 27

vps_extension() {	Descriptor
...	
for (i=0; i<=vps_num_layer_sets_minus1; i++)	
for (j=1; j<=vps_max_layer_id; j++)	
if (layer_id_included_flag[i][j]) {	
layer_set_hrd_layer_info_present_flag[i][j]	u(1)
if (layer_set_hrd_layer_info_present_flag[i][j]) {	
hrd_parameters(cprm_present_flag[i], vps_max_sub_layers_minus1)	
}	
...	
}	

FIG. 28

sps_extension() {	Descriptor
...	
layer_set_hrd_layer_info_present_flag	u(1)
if(layer_set_hrd_layer_info_present_flag) {	
hrd_parameters(1, sps_max_sub_layers_minus1)	
}	
...	
}	

29/56
FIG. 29

1	video_parameter_set_rbsp() {	Descriptor
2	vps_video_parameter_set_id	u(4)
3	vps_reserved_three_2bits	u(2)
4	vps_max_layers_minus1	u(6)
5	vps_max_sub_layers_minus1	u(3)
6	vps_temporal_id_nesting_flag	u(1)
7	vps_reserved_0xffff_16bits	u(16)
8	profile_tier_level(vps_max_sub_layers_minus1)	
9	vps_sub_layer_ordering_info_present_flag	u(1)
10	for (i=(vps_sub_layer_ordering_info_present_flag ? 0:vps_max_sub_layers_minus1); i <= vps_max_sub_layers_minus1; i++) {	
11	vps_max_dec_pic_buffering_minus1[i]	ue(v)
12	vps_max_num_reorder_pics[i]	ue(v)
13	vps_max_latency_increase_plus1[i]	ue(v)
14	}	
15	vps_max_layer_id	u(6)
16	vps_num_layer_sets_minus1	ue(v)
17	for (i=1; i <= vps_num_layer_sets_minus1; i++)	
18	for (j=0; j <= vps_max_layer_id; j++)	
19	layer_id_included_flag[i][j]	u(1)
20	vps_timing_info_present_flag	u(1)
21	if (vps_timing_info_present_flag) {	
22	vps_num_units_in_tick	u(32)
23	vps_time_scale	u(32)
24	vps_poc_proportional_to_timing_flag	u(1)
25	if (vps_poc_proportional_to_timing_flag)	
26	vps_num_ticks_poc_diff_one_minus1	ue(v)
27	vps_num_hrd_parameters	ue(v)
28	for (i=0; i<vps_num_hrd_parameters; i++) {	
29	hrd_layer_set_idx[i]	ue(v)
30	if (i>0)	
31	cprms_present_flag[i]	u(1)
32	hrd_parameters(cprms_present_flag[i], vps_max_sub_layers_minus1)	
33	}	
34	}	
35	vps_extension_flag	u(1)
36	if (vps_extension_flag)	
37	while (more_rbsp_data())	
38	vps_extension_data_flag	u(1)
39	rbsp_trailing_bits()	
40	}	

FIG. 30

`layer_id_included_flag[i][j]` equal to 1 specifies that the value of `nuh_layer_id` equal to `j` is included in the layer identifier list `layerSetLayerIdList[i]`. `layer_id_included_flag[i][j]` equal to 0 specifies that the value of `nuh_layer_id` equal to `j` is not included in the layer identifier list `layerSetLayerIdList[i]`.

The value of `numLayersInIdList[0]` is set equal to 1 and the value of `layerSetLayerIdList[0][0]` is set equal to 0.

For each value of `i` in the range of 1 to `vps_num_layer_sets_minus1`, inclusive, the variable `numLayersInIdList[i]` and the layer identifier list `layerSetLayerIdList[i]` are derived as follows:

```

n=0
for (m=0; m <= vps_max_layer_id; m++)
    if (layer_id_included_flag[i][m])
        layerSetLayerIdList[i][n++] = m
numLayersInIdList[i] = n

```

For each value of `i` in the range of 1 to `vps_num_layer_sets_minus1`, inclusive, `numLayersInIdList[i]` shall be in the range of 1 to `vps_max_layers_minus1 + 1`, inclusive.

When `numLayersInIdList[iA]` is equal to `numLayersInIdList[iB]` for any `iA` and `iB` in the range of 0 to `vps_num_layer_sets_minus1`, inclusive, with `iA` not equal to `iB`, the value of `layerSetLayerIdList[iA][n]` shall not be equal to `layerSetLayerIdList[iB][n]` for at least one value of `n` in the range of 0 to `numLayersInIdList[iA]`, inclusive.

A layer set is identified by the associated layer identifier list. The `j`-th layer set specified by the VPS is associated with the layer identifier list `layerSetLayerIdList[i]`, for `i` in the range of 0 to `vps_num_layer_sets_minus1`, inclusive.

A **layer set** consists of all **operation points** that are associated with the same layer identifier list.

Each operation point is identified by the associated layer identifier list, denoted as `OpLayerIdList`, which consists of the list of `nuh_layer_id` values of all NAL units included in the operation point, in increasing order of `nuh_layer_id` values, and a variable `OpTid`, which is equal to the highest `TemporalId` of all NAL units included in the operation point. The **bitstream subset** associated with the operation point identified by `OpLayerIdList` and `OpTid` is the output of the sub-bitstream extraction process as specified in clause 10 with the bitstream, the target highest `TemporalId` equal to `OpTid`, and the target layer identifier list equal to `OpLayerIdList` as inputs. The `OpLayerIdList` and `OpTid` that identify an operation point are also referred to as the `OpLayerIdList` and `OpTid` associated with the operation point, respectively.

FIG. 31

LayerSet[1] = {0, 1, 2}

LayerSet[2] = {0, 2}

↓

layer_id_included_flag[1][0] = 1 // 0 IS INCLUDED IN LayerSet[1]

layer_id_included_flag[1][1] = 1 // 1 IS INCLUDED IN LayerSet[1]

layer_id_included_flag[1][2] = 1 // 2 IS INCLUDED IN LayerSet[1]

layer_id_included_flag[2][0] = 1 // 0 IS INCLUDED IN LayerSet[2]

layer_id_included_flag[2][1] = 0 // 1 IS NOT INCLUDED IN LayerSet[1]

layer_id_included_flag[2][2] = 1 // 2 IS INCLUDED IN LayerSet[2]

FIG. 32

1	buffering_period(payloadSize) {	Descriptor
2	bp_seq_parameter_set_id	ue(v)
3	if(!sub_pic_hrd_params_present_flag)	
4	irap_cpb_params_present_flag	u(1)
5	if(irap_cpb_params_present_flag) {	
6	cpb_delay_offset	u(v)
7	dpb_delay_offset	u(v)
8	}	
9	concatenation_flag	u(1)
10	au_cpb_removal_delay_delta_minus1	u(v)
11	if(NalHrdBpPresentFlag) {	
12	for(i= 0; i <= CpbCnt; i++) {	
13	nal_initial_cpb_removal_delay[i]	u(v)
14	nal_initial_cpb_removal_offset[i]	u(v)
15	if(sub_pic_hrd_params_present_flag irap_cpb_params_present_flag) {	
16	nal_initial_alt_cpb_removal_delay[i]	u(v)
17	nal_initial_alt_cpb_removal_offset[i]	u(v)
18	}	
19	}	
20	}	
21	if(VclHrdBpPresentFlag) {	
22	for(i=0; i <= CpbCnt; i++) {	
23	vcl_initial_cpb_removal_delay[i]	u(v)
24	vcl_initial_cpb_removal_offset[i]	u(v)
25	if(sub_pic_hrd_params_present_flag irap_cpb_params_present_flag) {	
26	vcl_initial_alt_cpb_removal_delay[i]	u(v)
27	vcl_initial_alt_cpb_removal_offset[i]	u(v)
28	}	
29	}	
30	}	
31	}	

FIG. 33

1	buffering_period(payloadSize) {	Descriptor
2	associated_parameter_set_flag	u(1)
3	if(associated_parameter_set_flag == 0) {	
4	bp_video_parameter_set_id	ue(v)
5	vps_num_bp_parameters	ue(v)
6	for(i=0; i < vps_num_bp_parameters; i++) {	
7	bp_layer_set_idx[i]	ue(v)
8	if(i > 0)	
9	oprms_present_flag[i]	u(1)
10	layer_buffering_period(payloadSize)	
11	}	
12	} else {	
13	bp_seq_parameter_set_id	ue(v)
14	layer_buffering_period(payloadSize)	
15	}	
16	}	

FIG. 34

1	layer_buffering_period(payloadSize) {	Descriptor
2	if(!sub_pic_hrd_params_present_flag)	
3	irap_cpb_params_present_flag	u(1)
4	if(irap_cpb_params_present_flag) {	
5	cpb_delay_offset	u(v)
6	dpb_delay_offset	u(v)
7	}	
8	concatenation_flag	u(1)
9	au_cpb_removal_delay_delta_minus1	u(v)
10	if(NalHrdBpPresentFlag) {	
11	for(i=0; i <= CpbCnt; i++) {	
12	nal_initial_cpb_removal_delay[i]	u(v)
13	nal_initial_cpb_removal_offset[i]	u(v)
14	if(sub_pic_hrd_params_present_flag irap_cpb_params_present_flag) {	
15	nal_initial_alt_cpb_removal_delay[i]	u(v)
16	nal_initial_alt_cpb_removal_offset[i]	u(v)
17	}	
18	layer_specific_parameters_present_flag	u(1)
19	if(layer_specific_parameters_present_flag) {	
20	for(i=0; i <= CpbCnt; i++) {	
21	layer_nal_initial_cpb_removal_delay[i]	u(v)
22	layer_nal_initial_cpb_removal_offset[i]	u(v)
23	if(sub_pic_hrd_params_present_flag irap_cpb_params_present_flag) {	
24	layer_nal_initial_alt_cpb_removal_delay[i]	u(v)
25	layer_nal_initial_alt_cpb_removal_offset[i]	u(v)
26	}	
27	}	
28	}	
29	}	
30	if(VolHrdBpPresentFlag) {	
31	for(i=0; i <= CpbCnt; i++) {	
32	vol_initial_cpb_removal_delay[i]	u(v)
33	vol_initial_cpb_removal_offset[i]	u(v)
34	if(sub_pic_hrd_params_present_flag irap_cpb_params_present_flag) {	
35	vol_initial_alt_cpb_removal_delay[i]	u(v)
36	vol_initial_alt_cpb_removal_offset[i]	u(v)
37	}	
38	layer_specific_parameters_present_flag	u(1)
39	if(layer_specific_parameters_present_flag) {	
40	for(i=0; i <= CpbCnt; i++) {	

FIG. 35

41	layer_vcl_initial_cpb_removal_delay[i]	u(v)
42	layer_vcl_initial_cpb_removal_offset[i]	u(v)
43	if(sub_pic_hrd_params_present_flag irap_cpb_params_present_flag) {	
44	layer_vcl_initial_alt_cpb_removal_delay[i]	u(v)
45	layer_vcl_initial_alt_cpb_removal_offset[i]	u(v)
46	}	
47	}	
48	}	
49	}	
50	}	

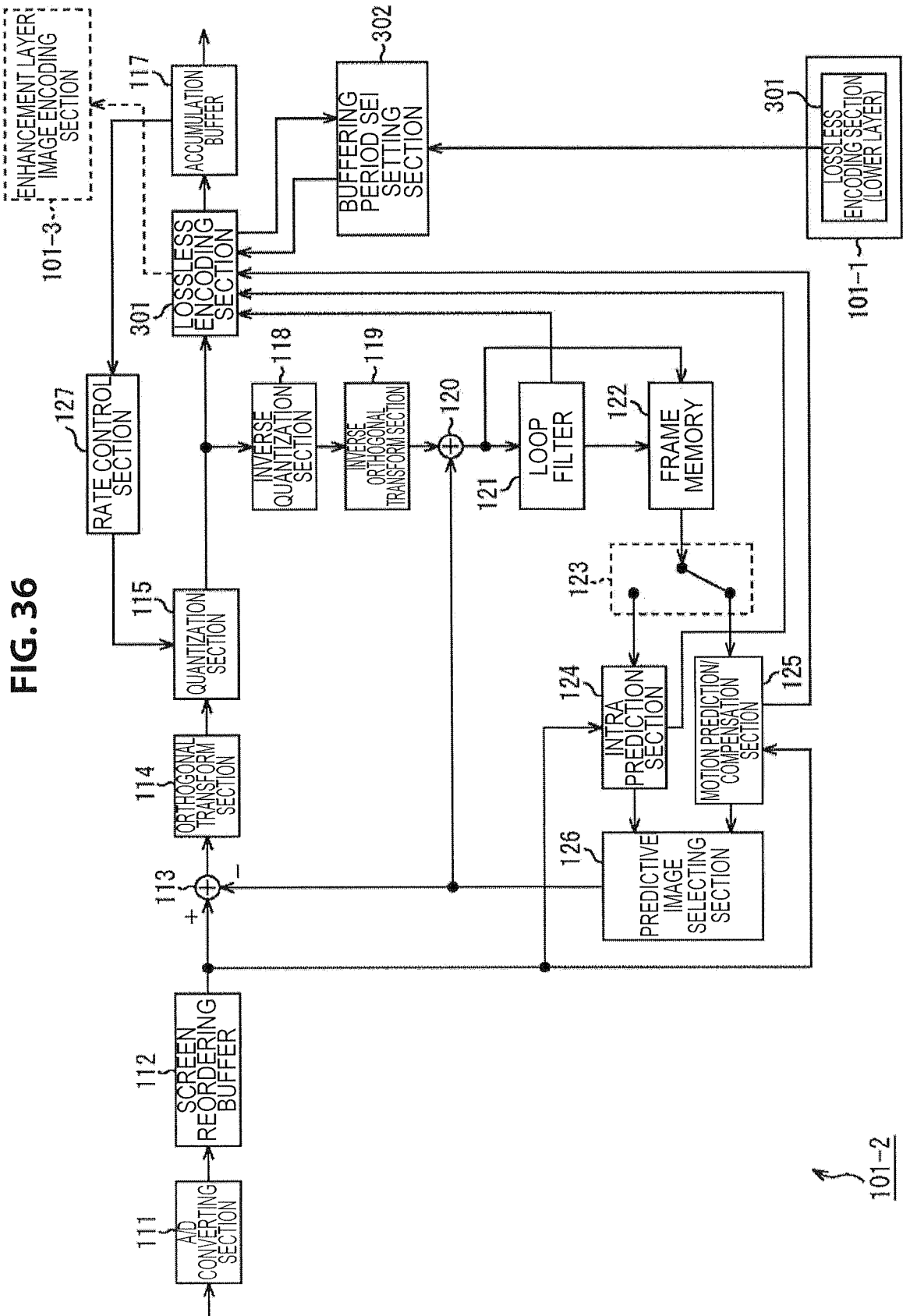


FIG. 37

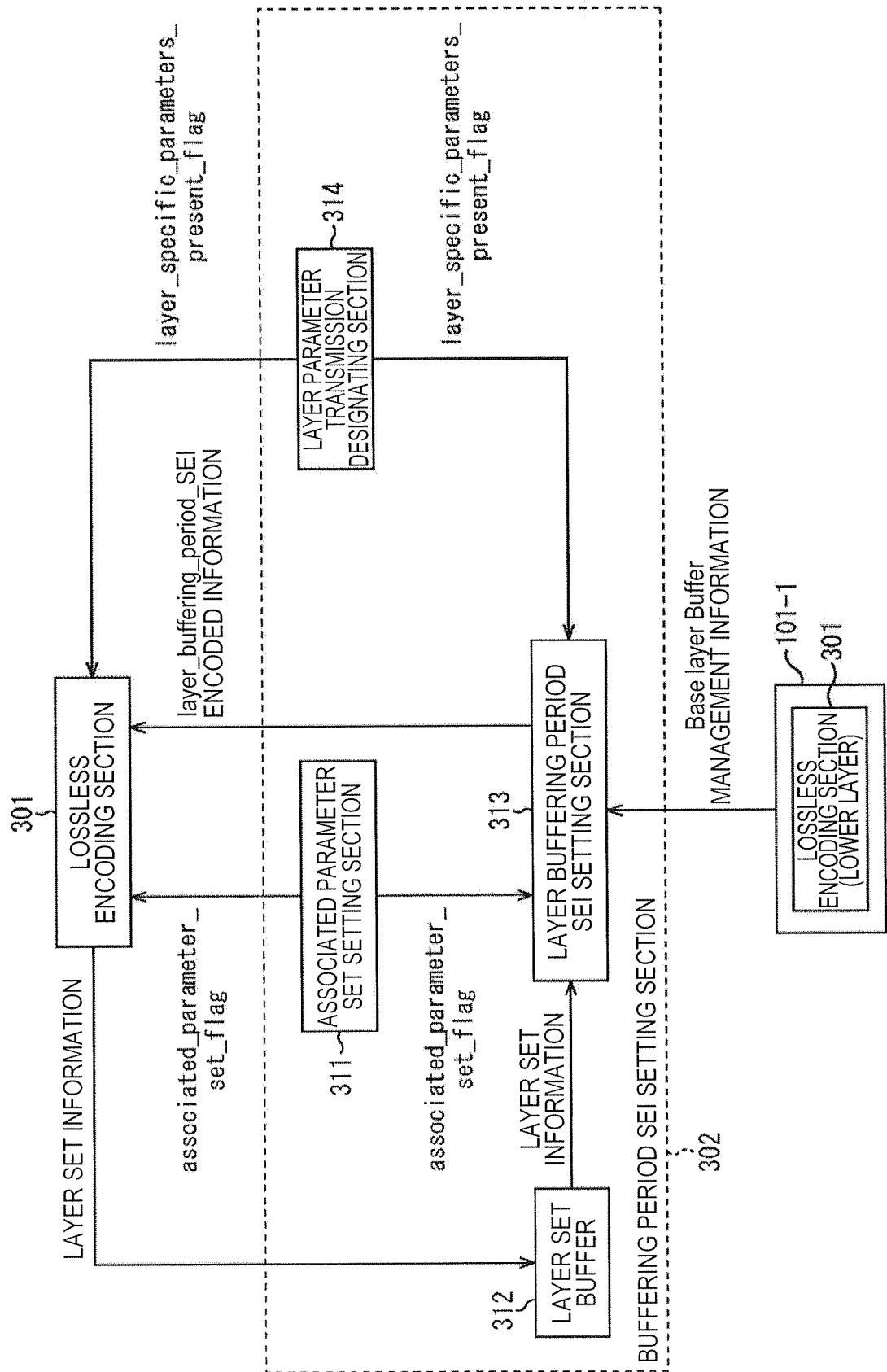


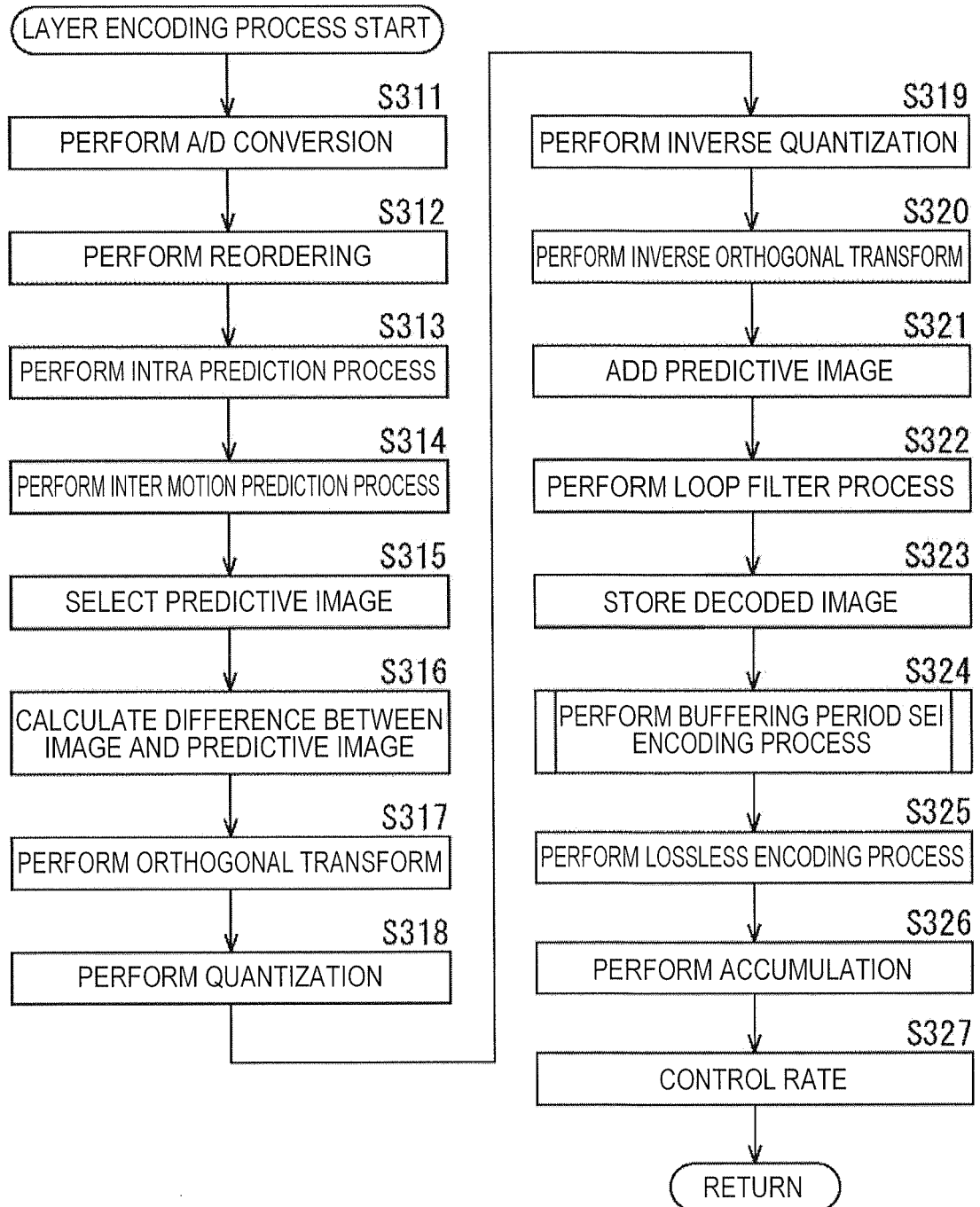
FIG. 38

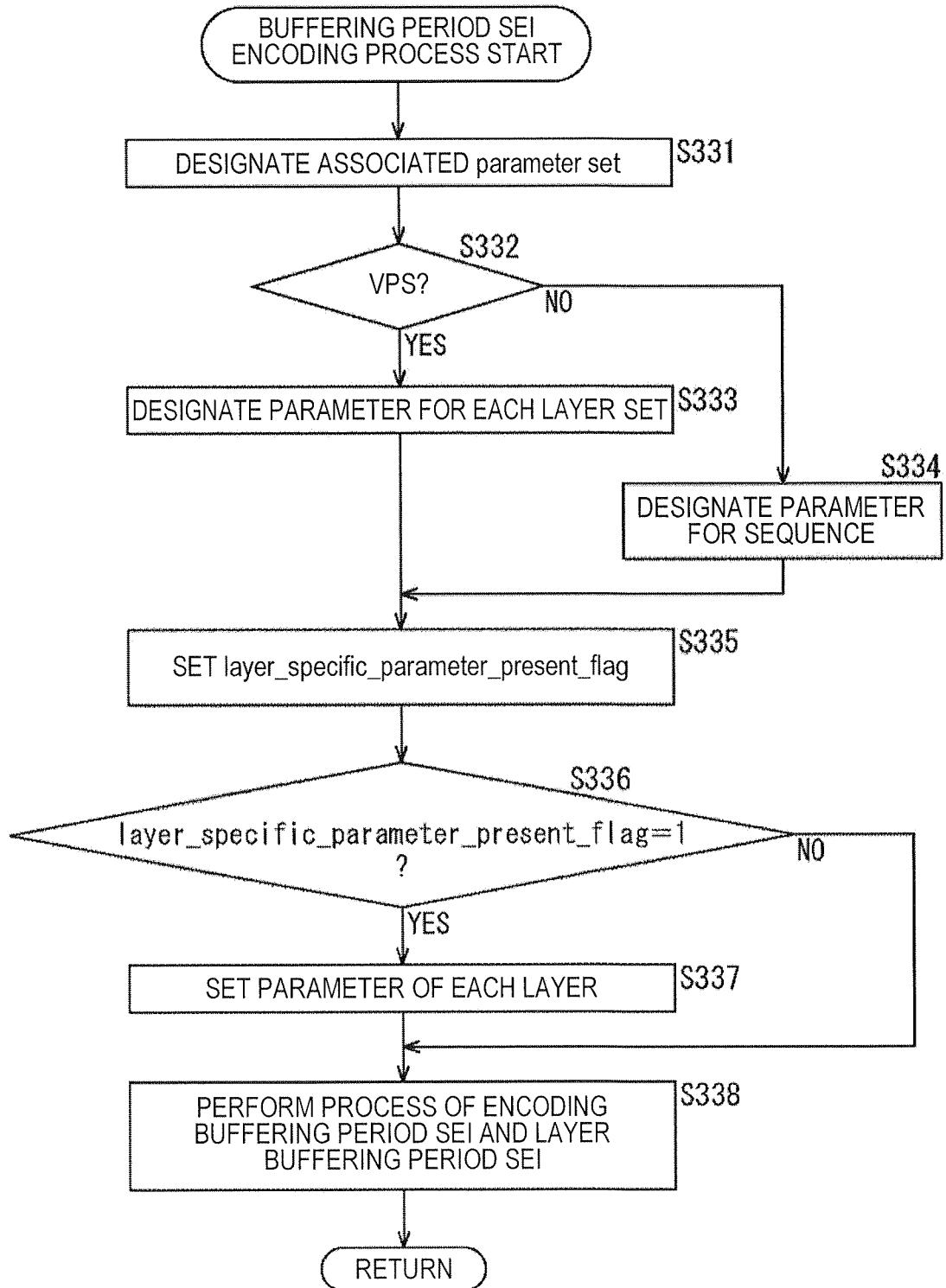
FIG. 39

FIG. 40

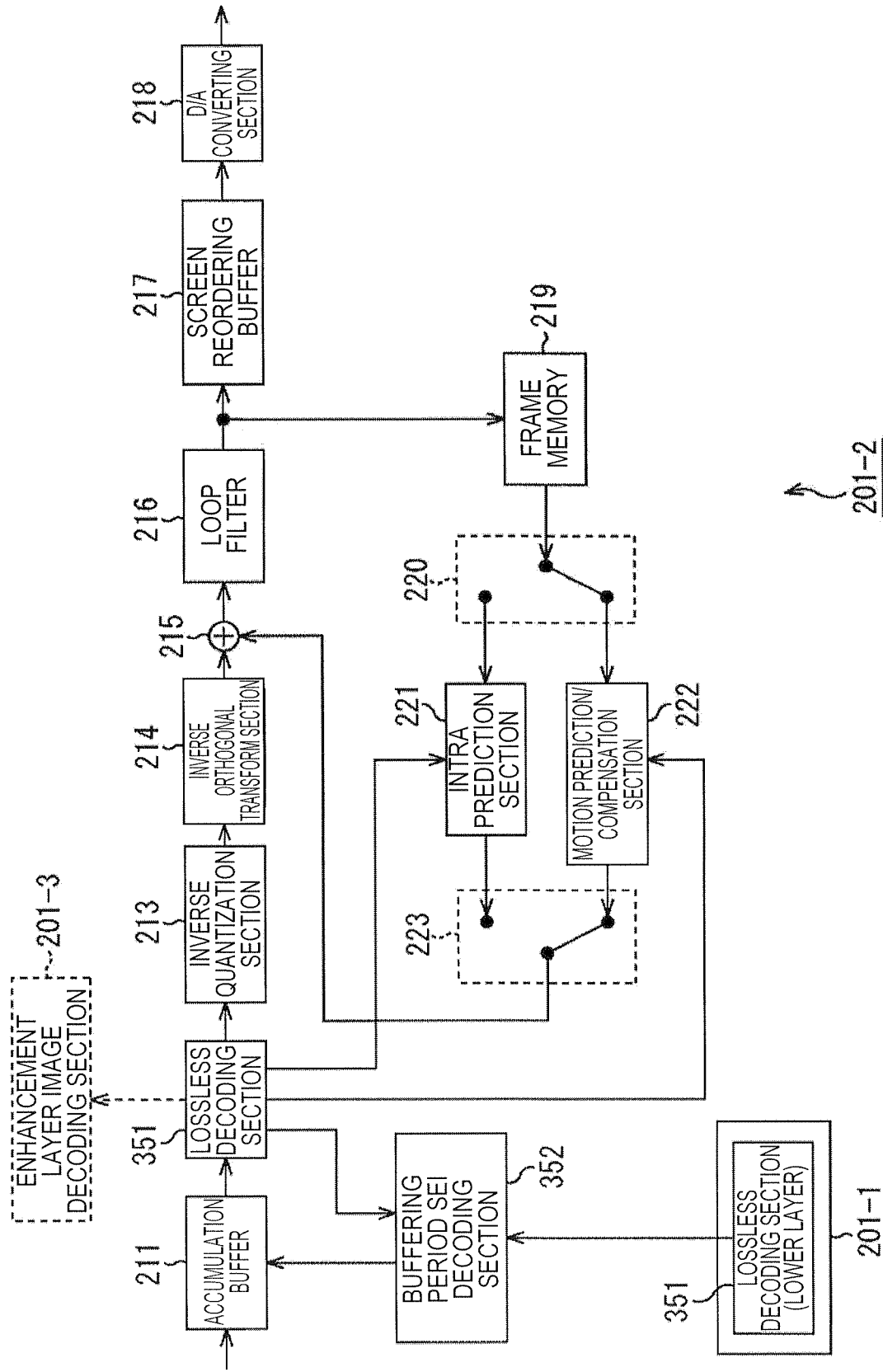


FIG. 41

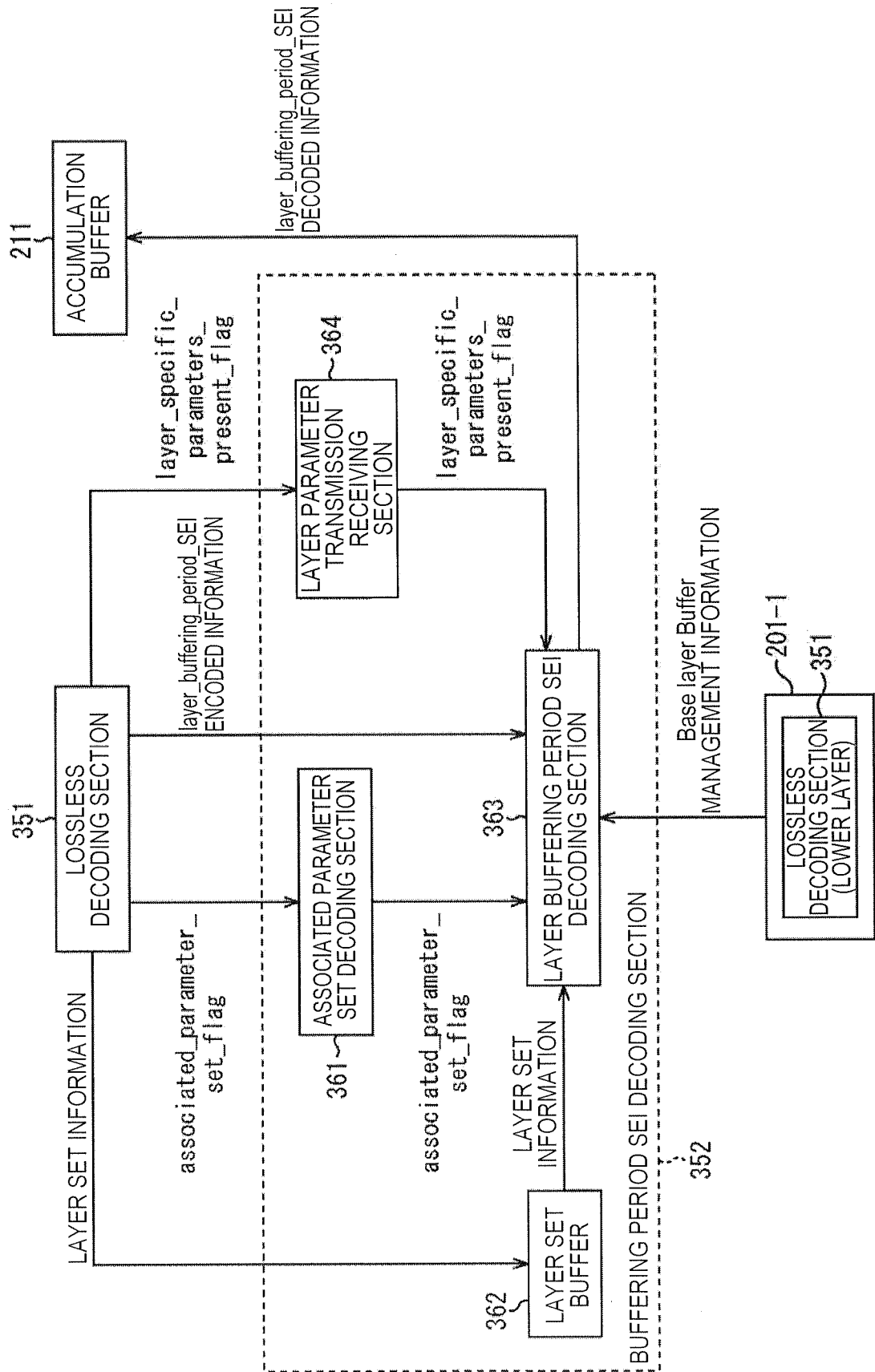


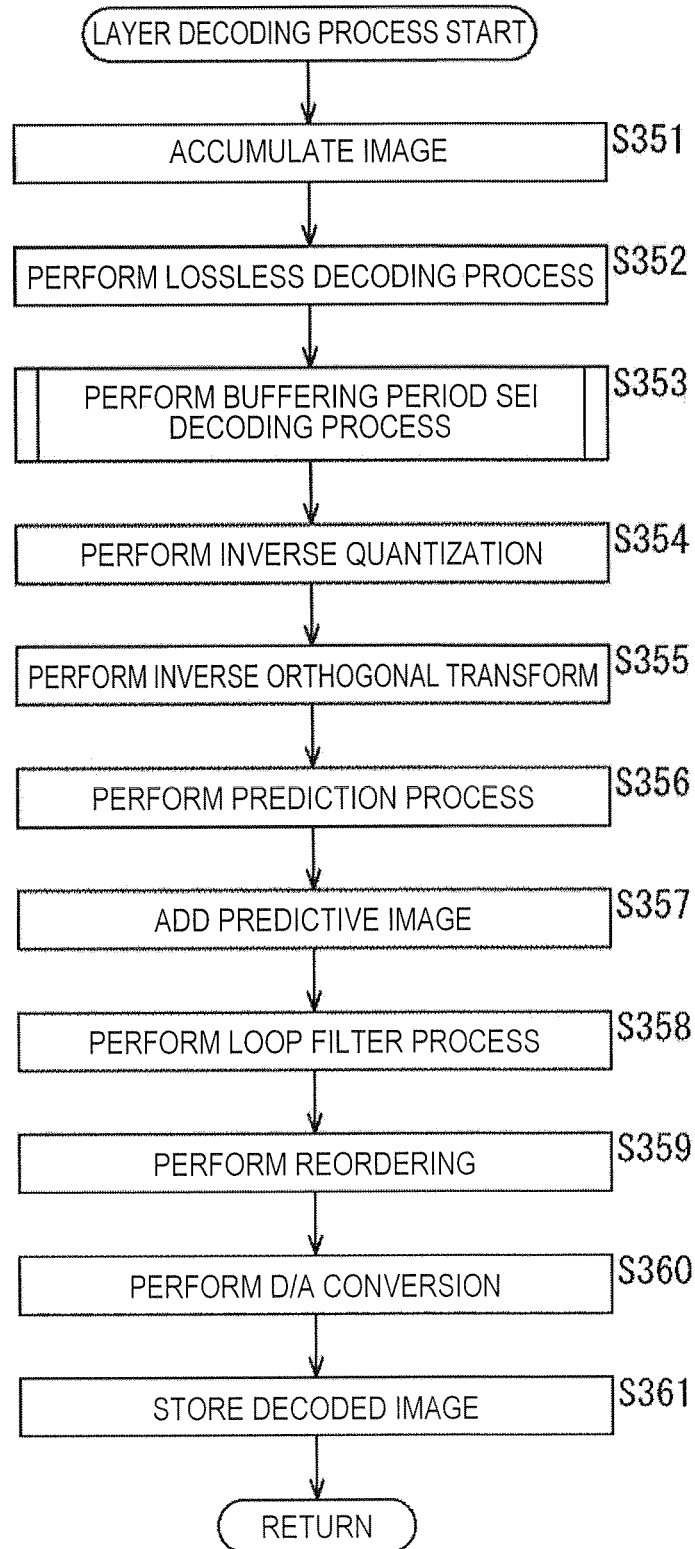
FIG. 42

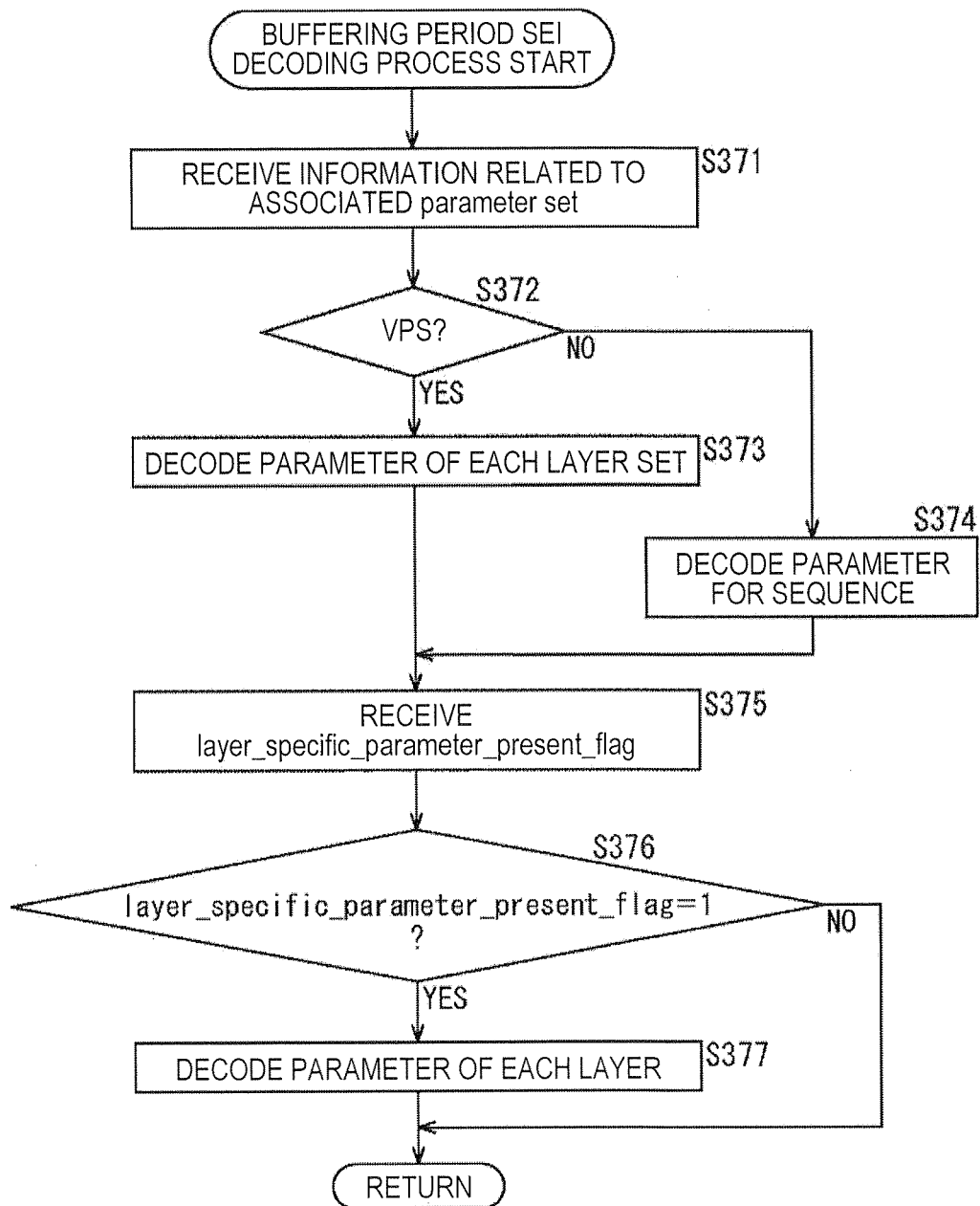
FIG. 43

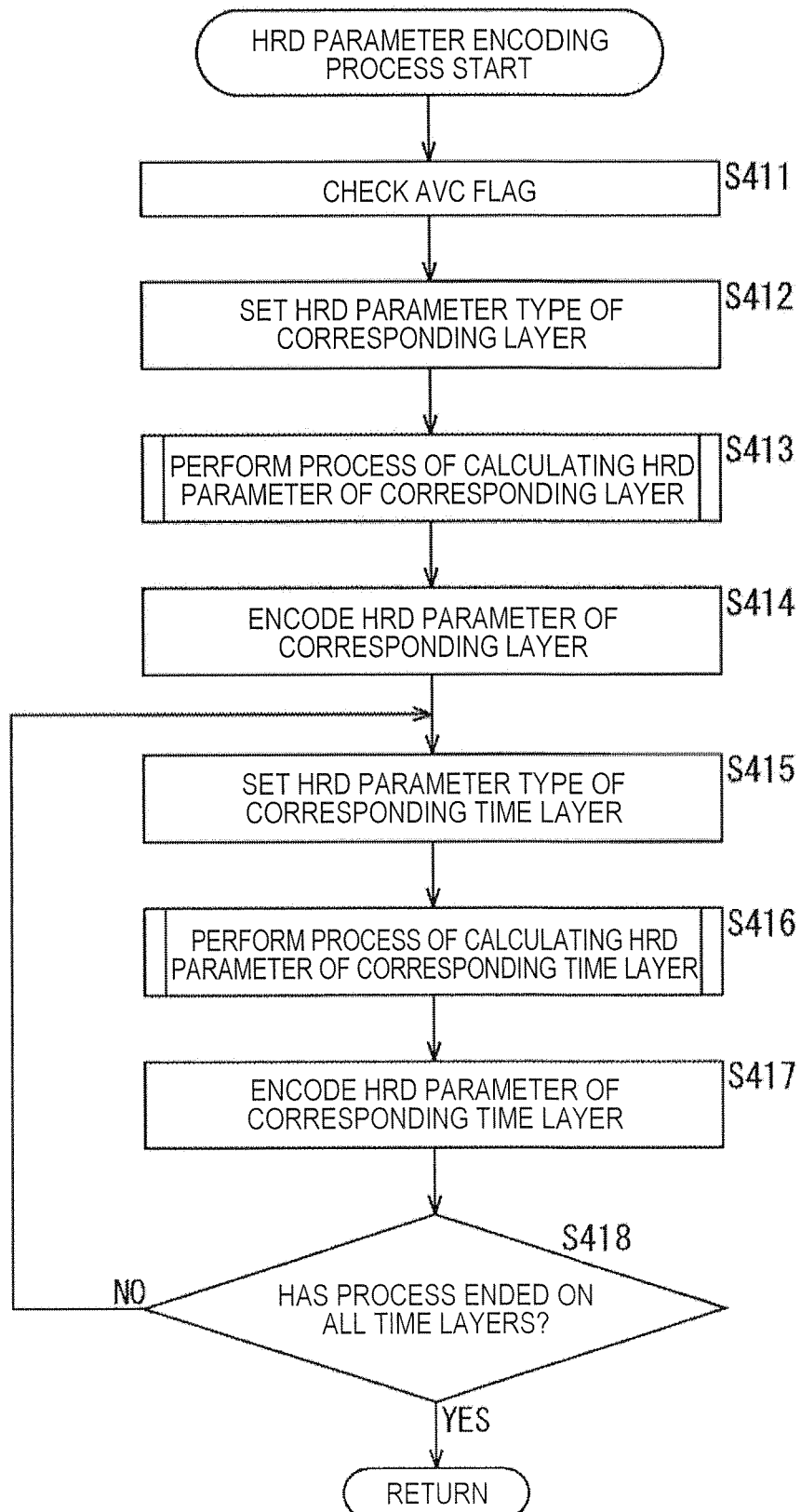
FIG. 44

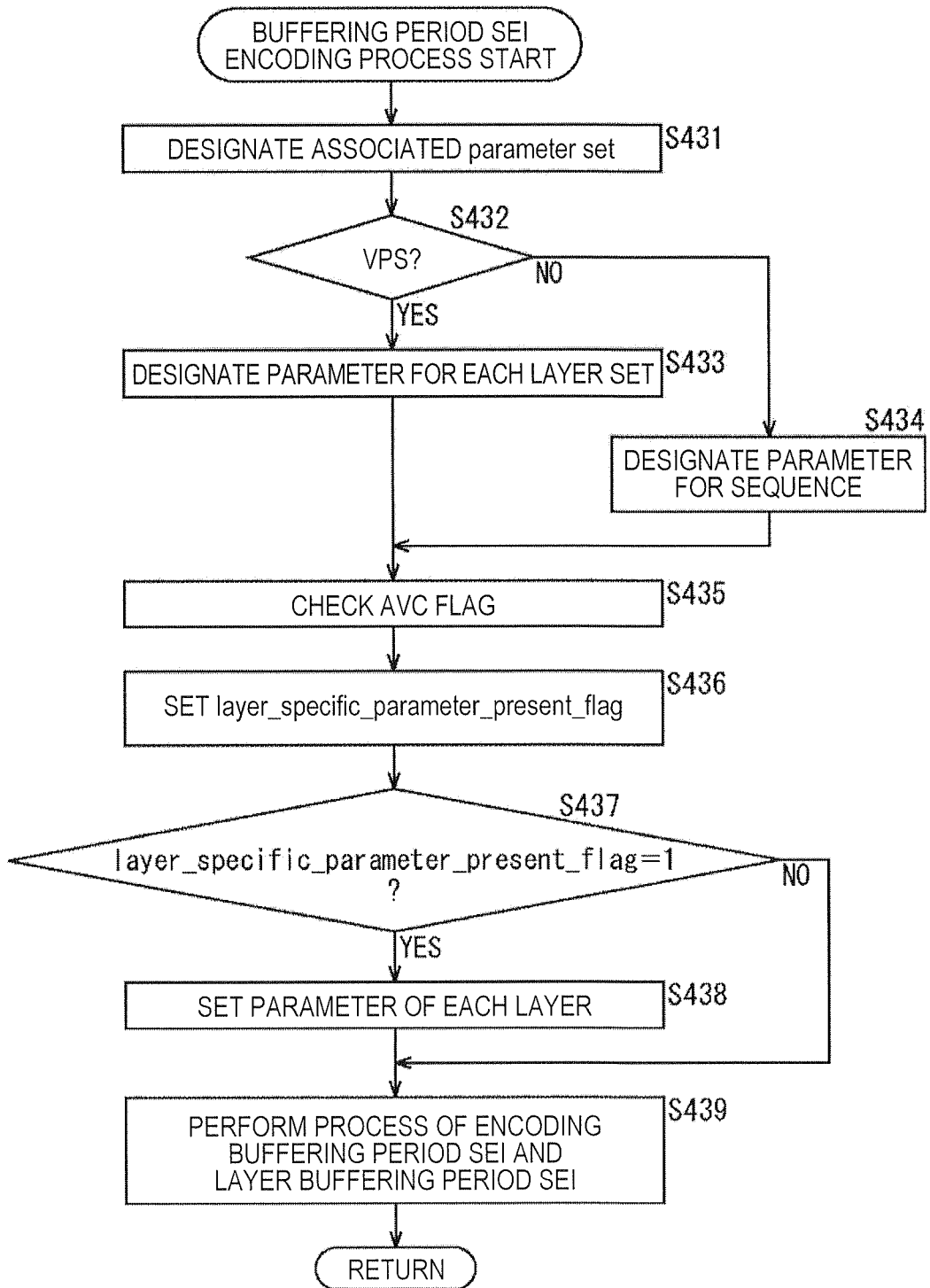
FIG. 45

FIG. 46

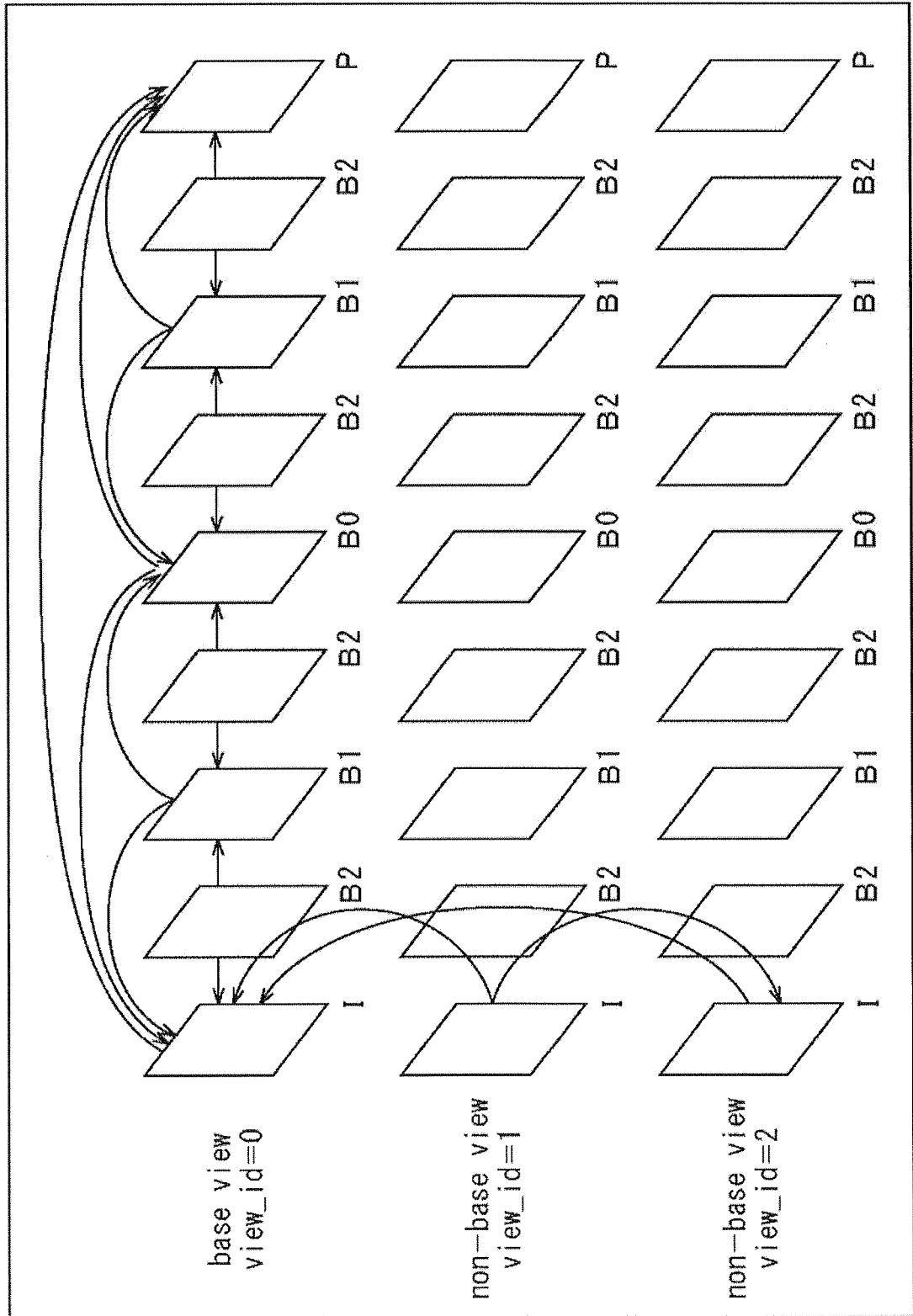


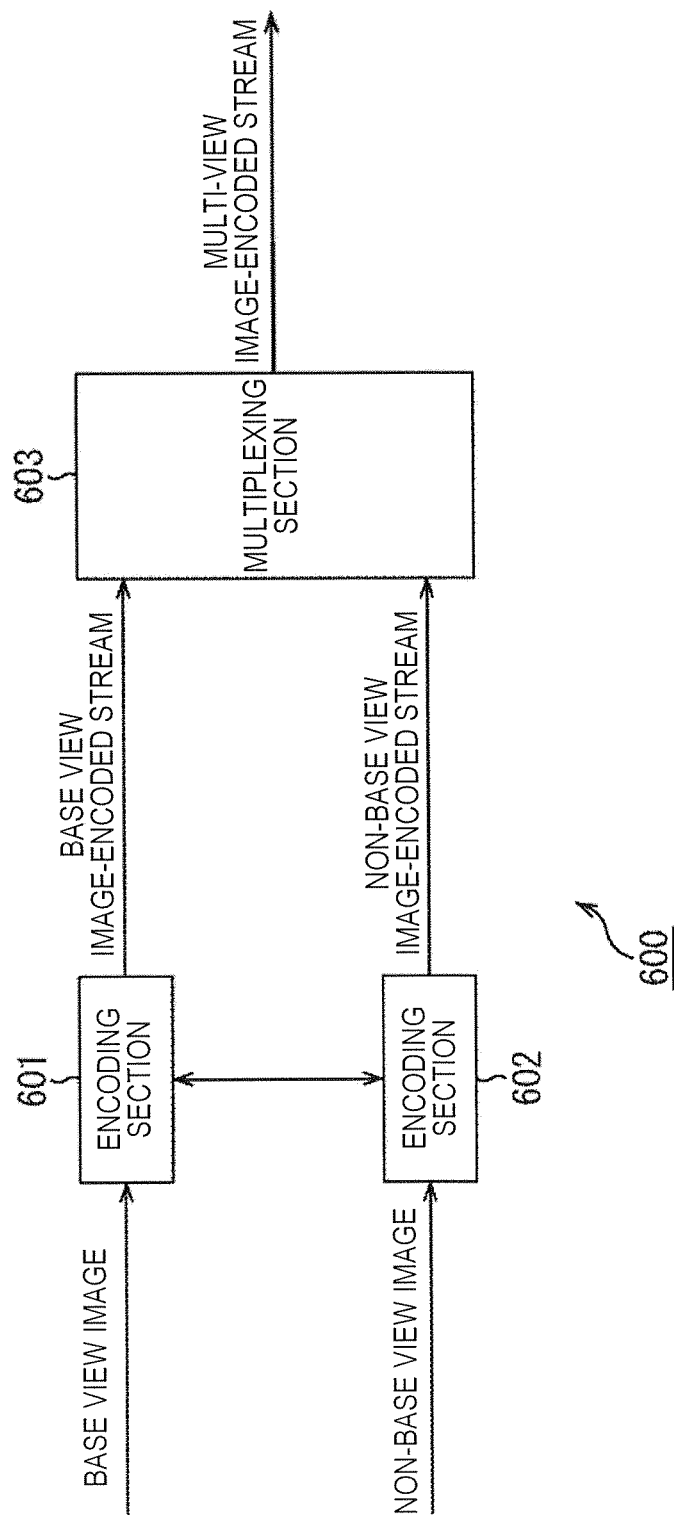
FIG. 47

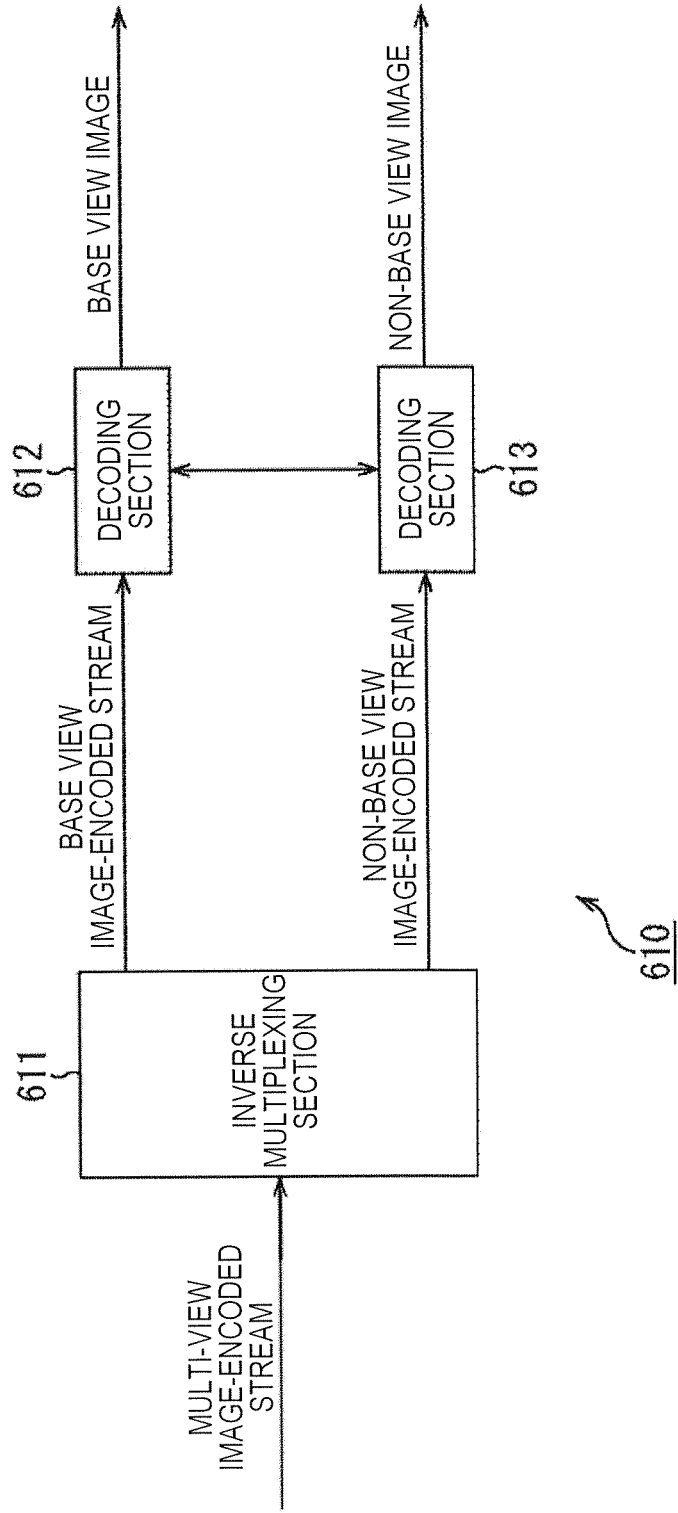
FIG. 48

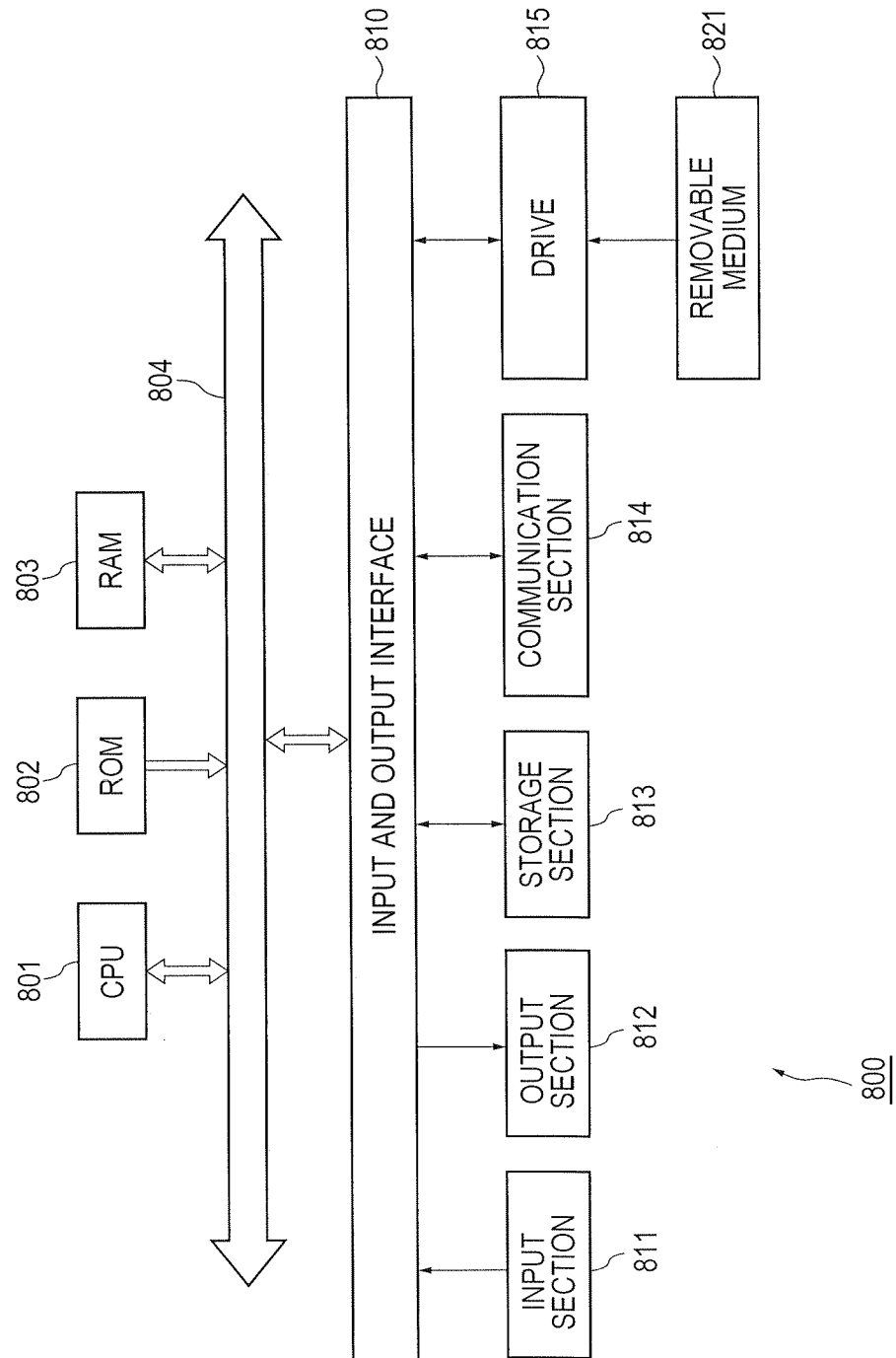
FIG. 49

FIG. 50

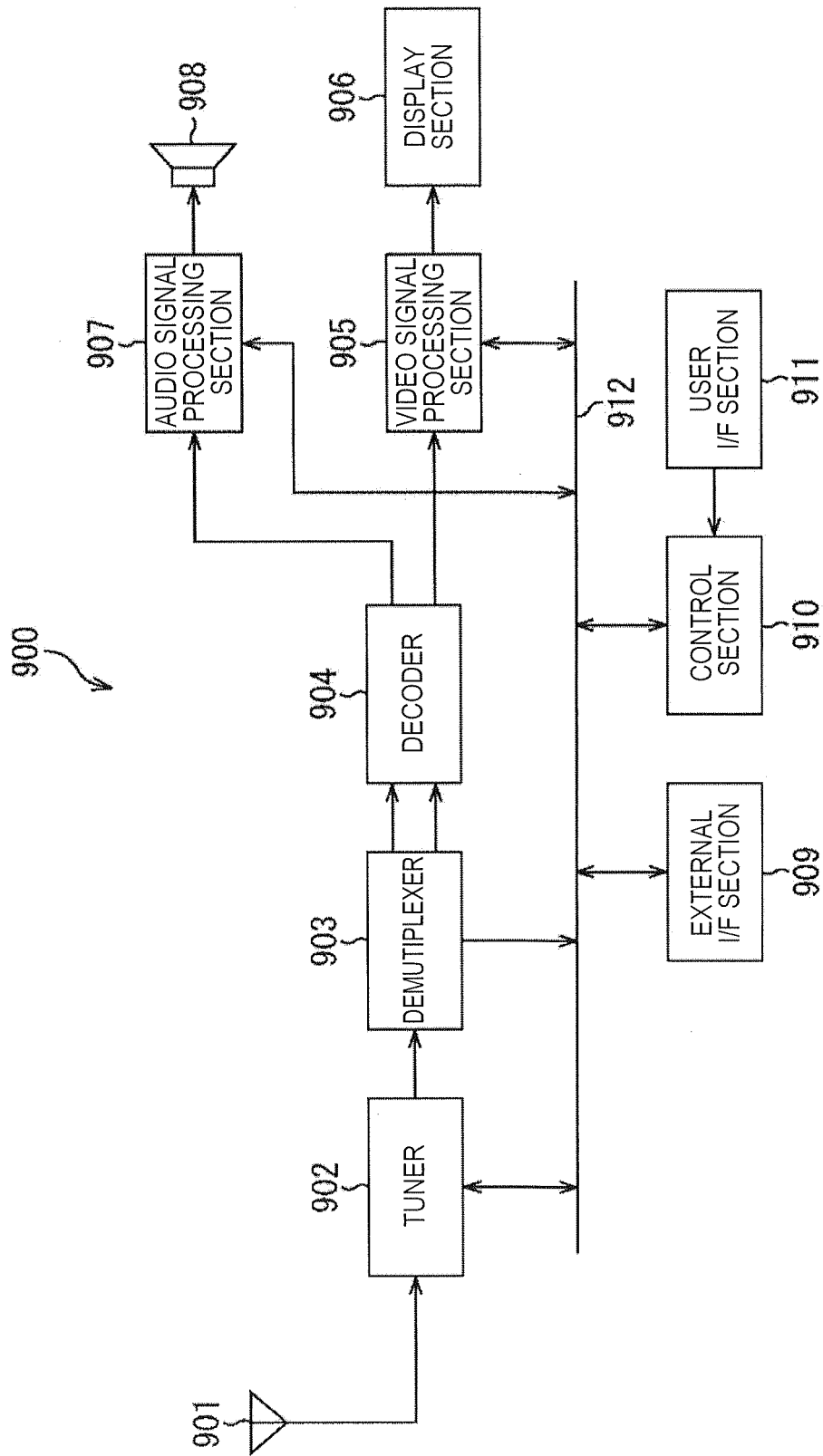


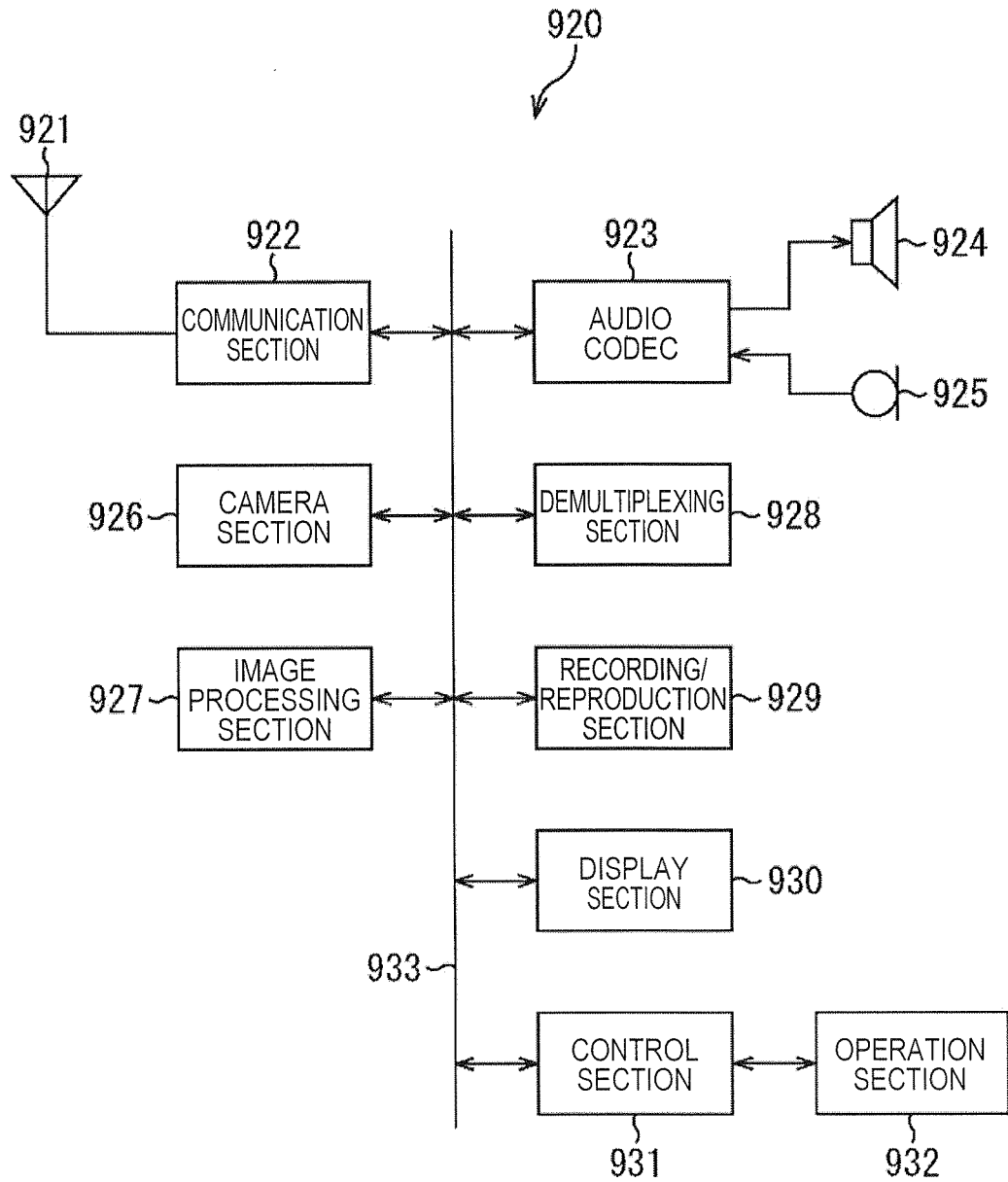
FIG. 51

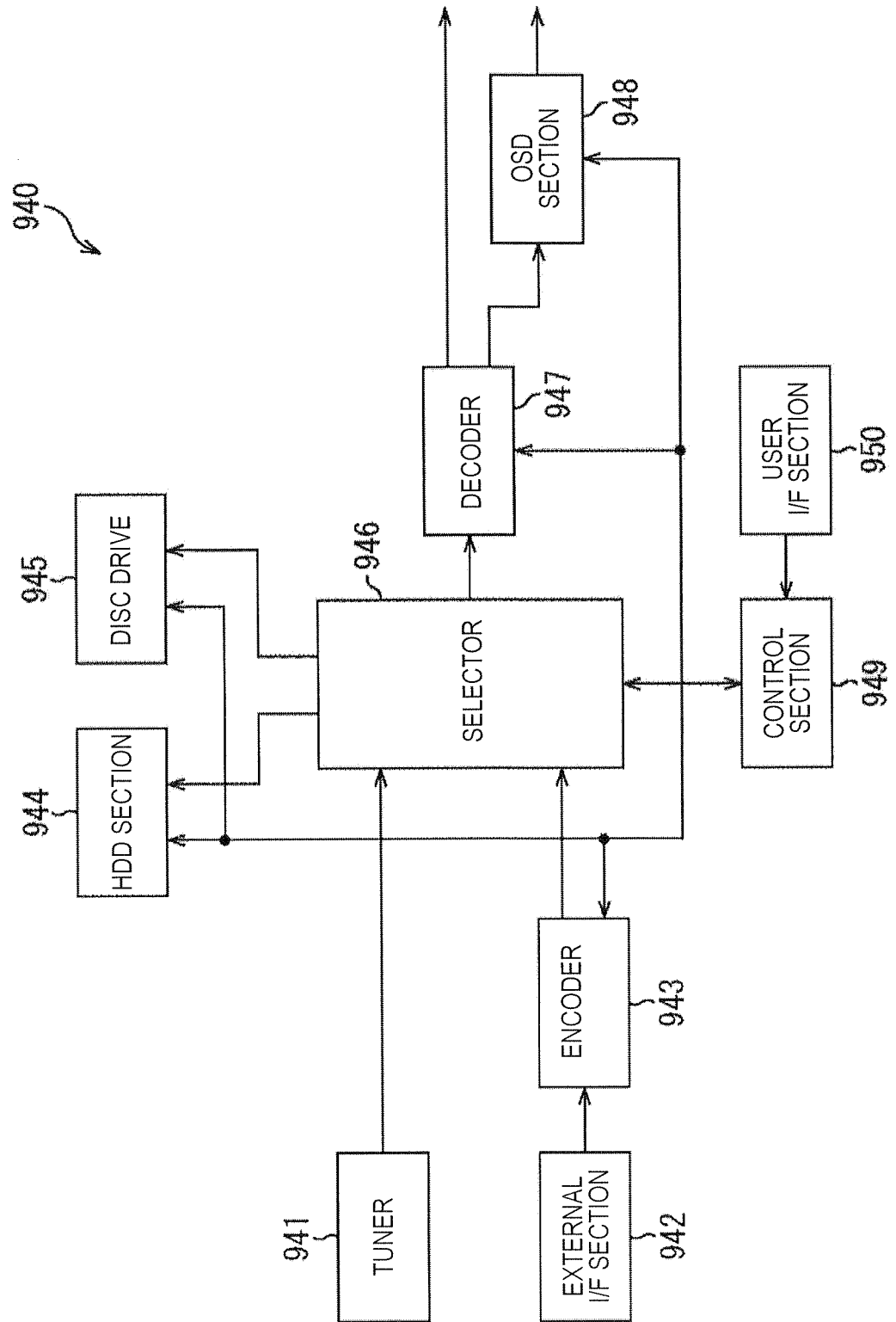
FIG. 52

FIG. 53

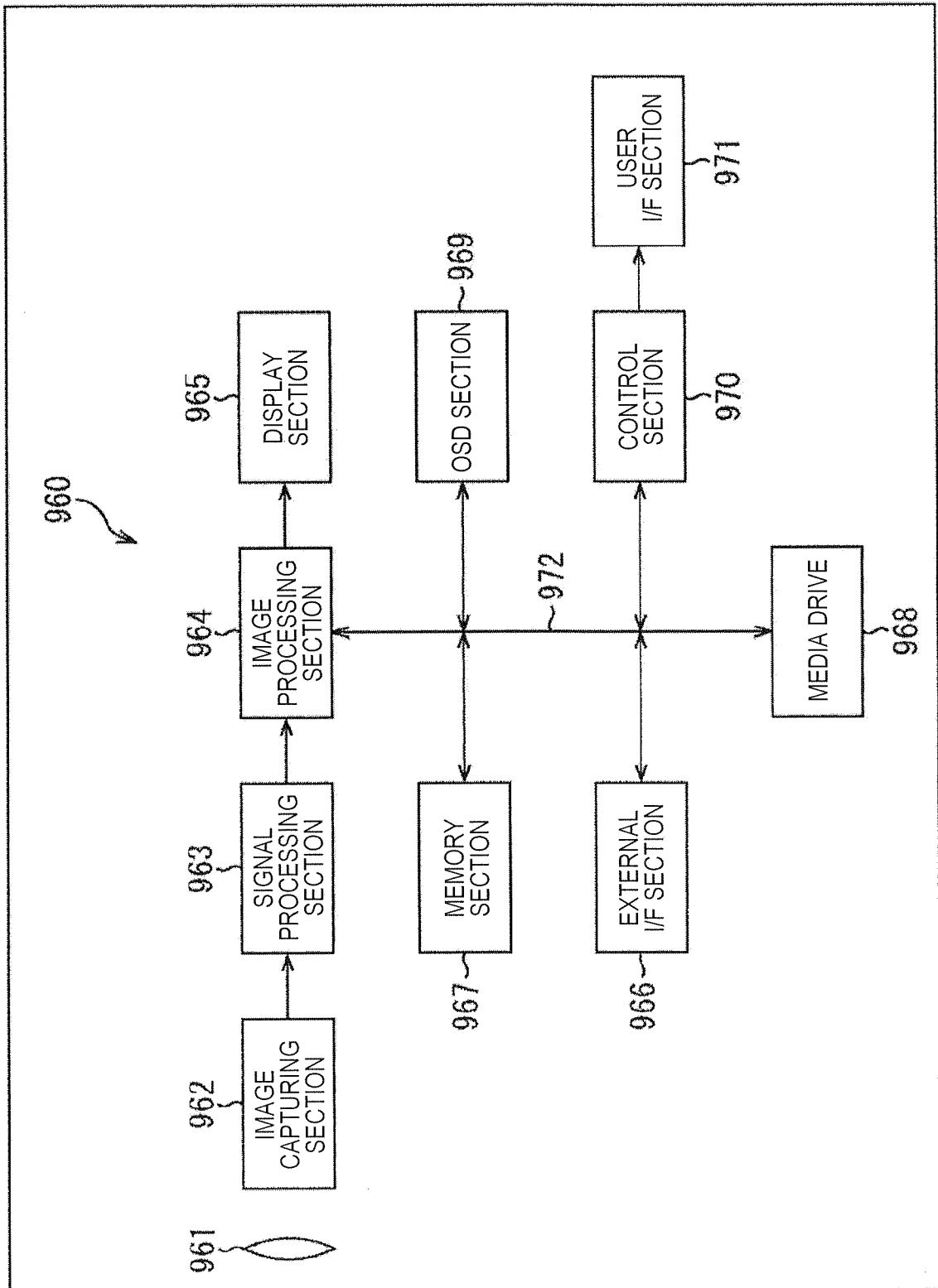


FIG. 54

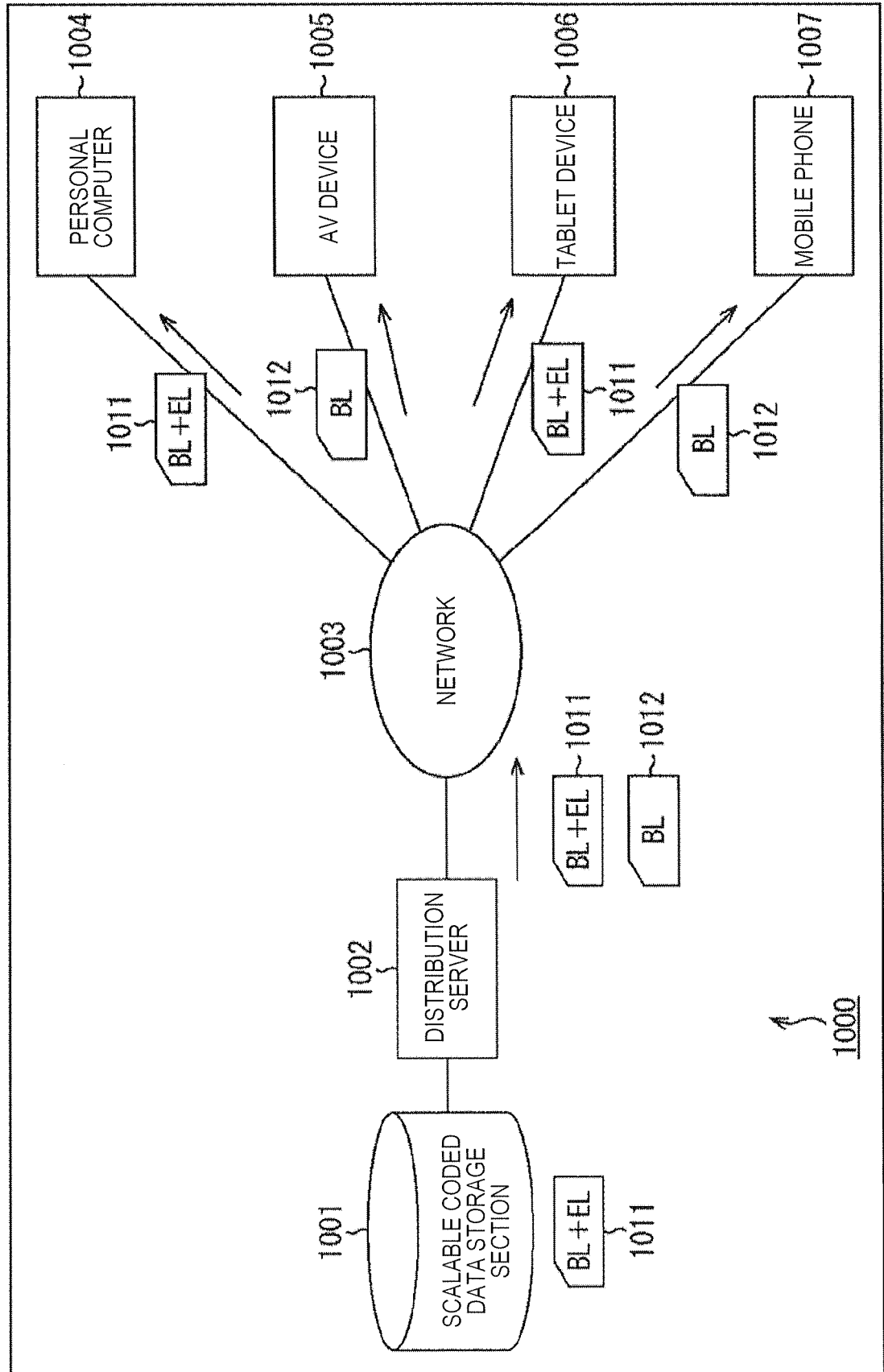


FIG. 55

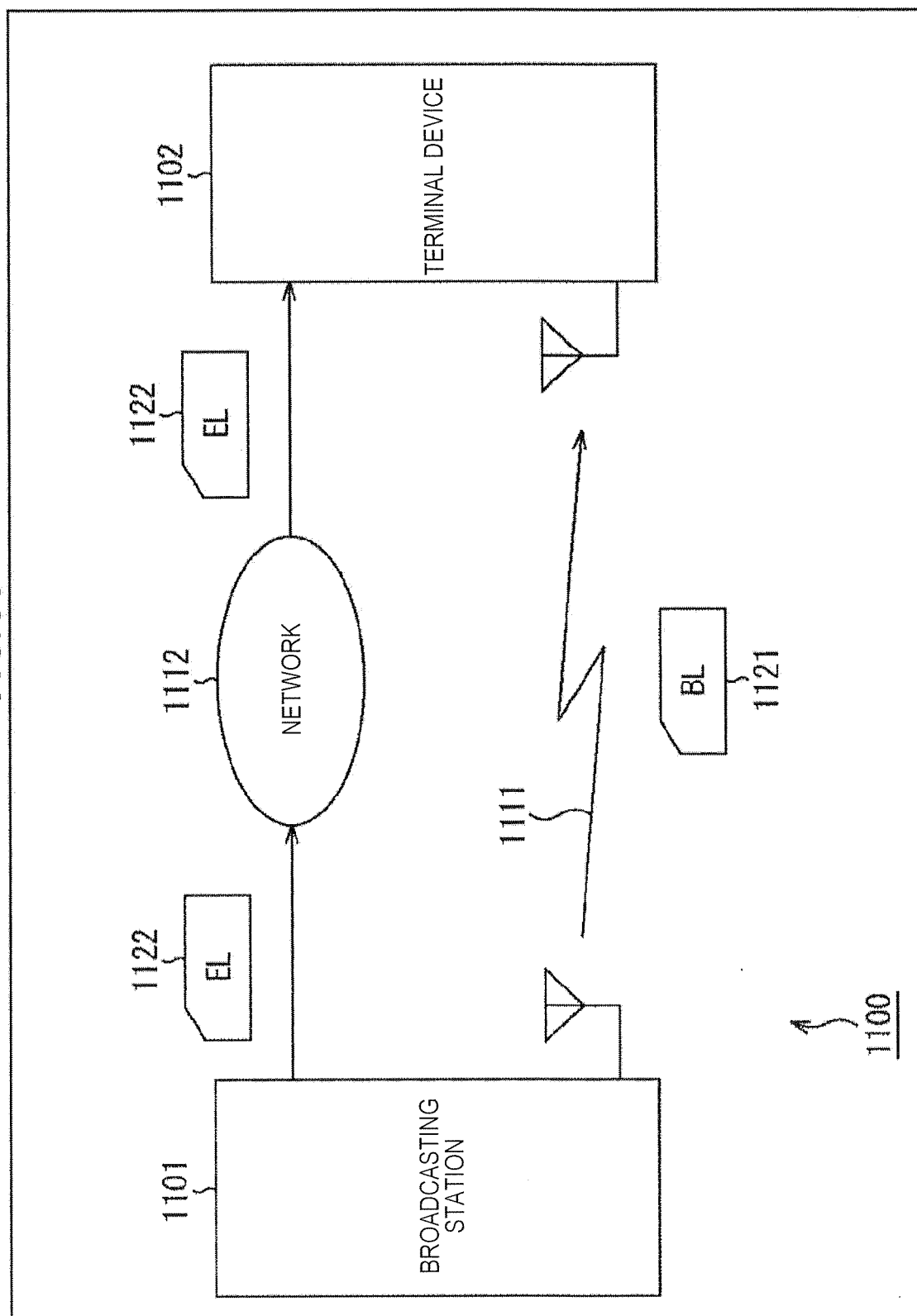


FIG. 56

