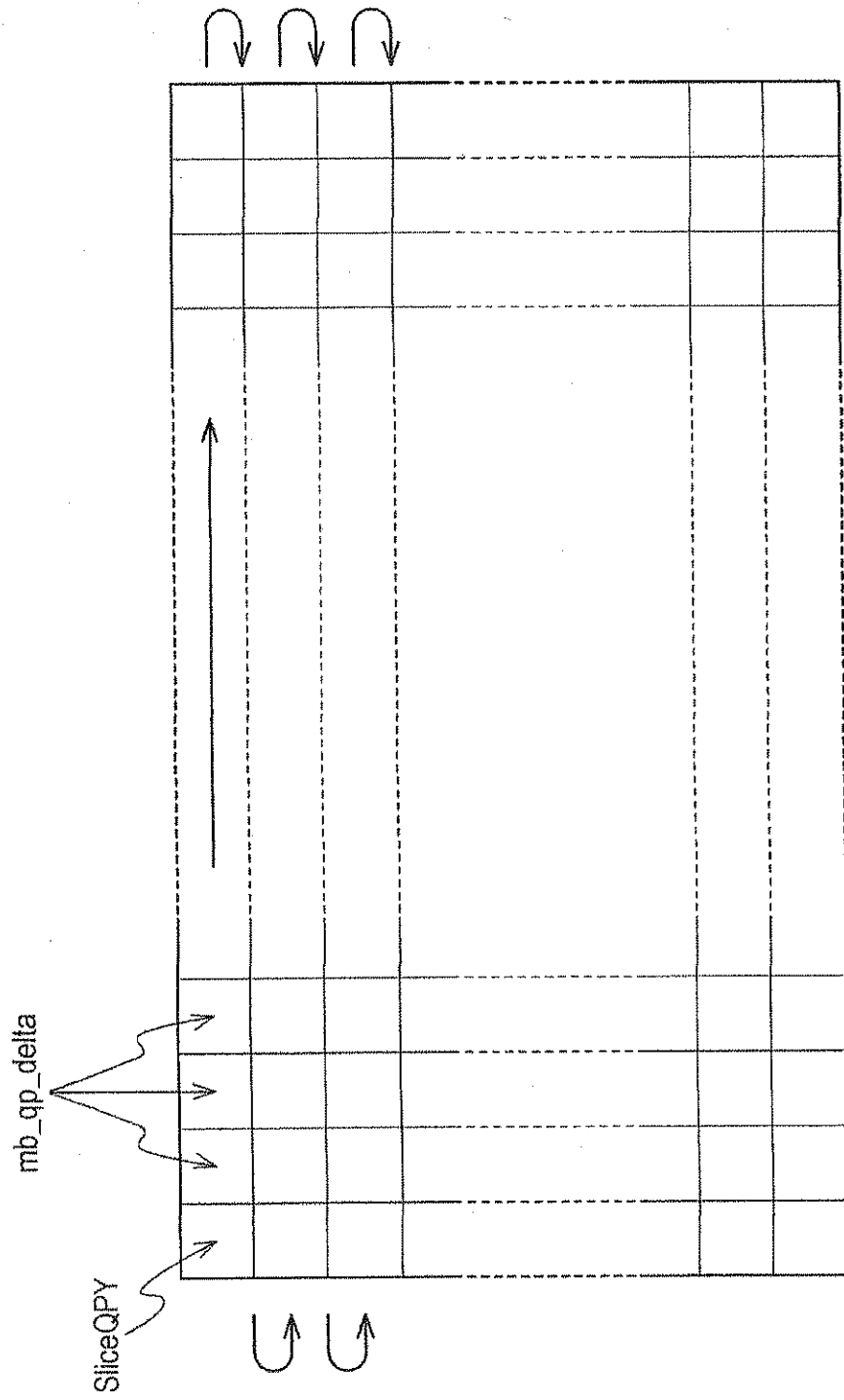


FIG. 1



25
G
L

0	1	4	5	16
2	3	6	7	
8	9	12	13	
10	11	14	15	

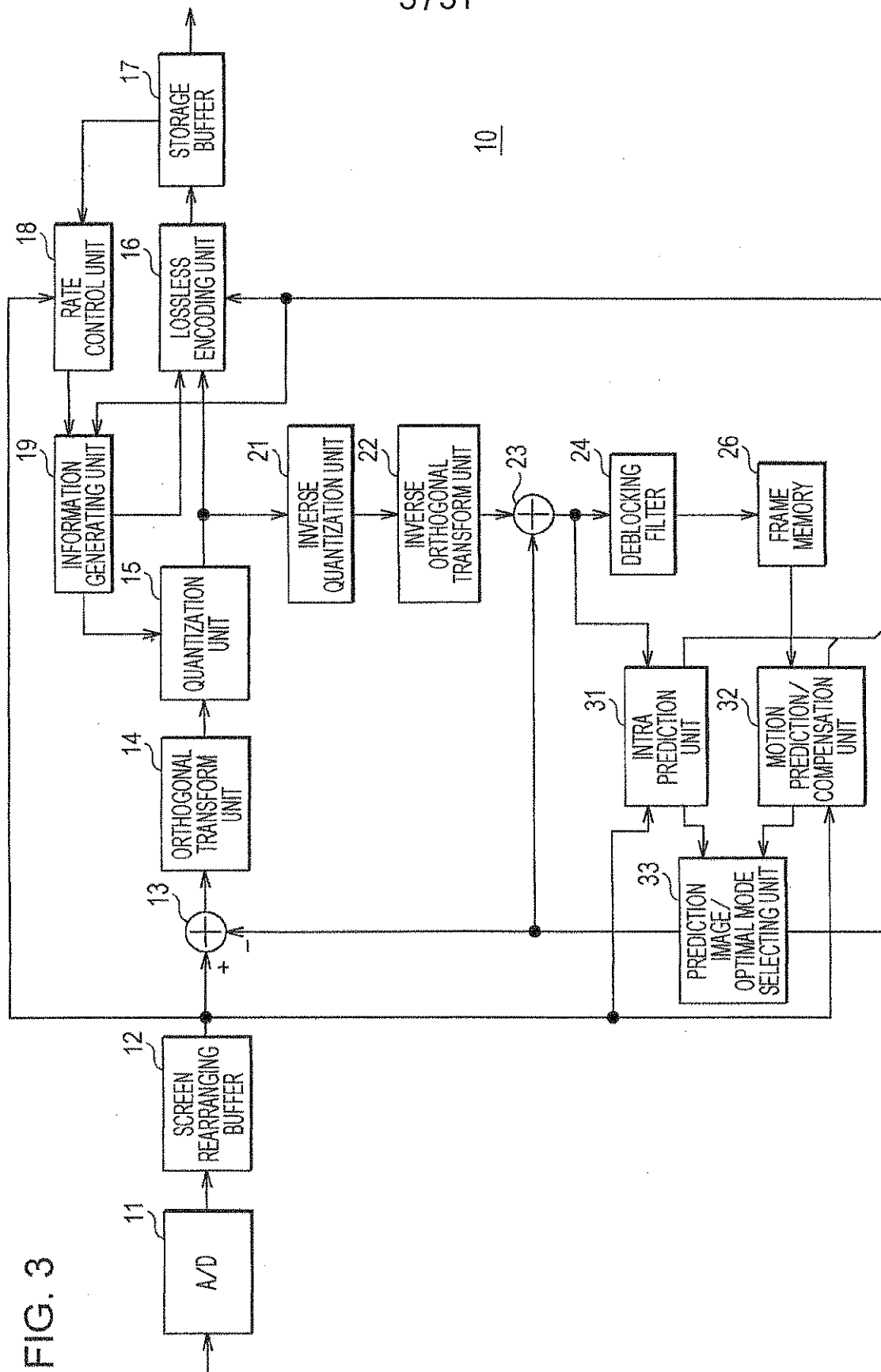


FIG. 4

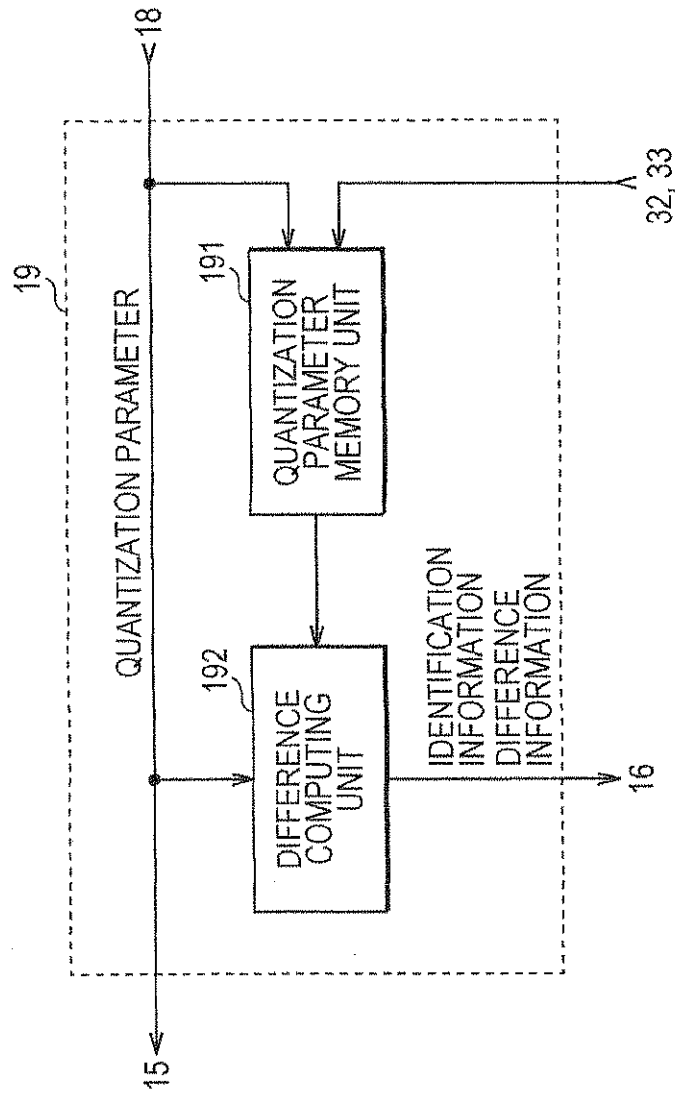


FIG. 5

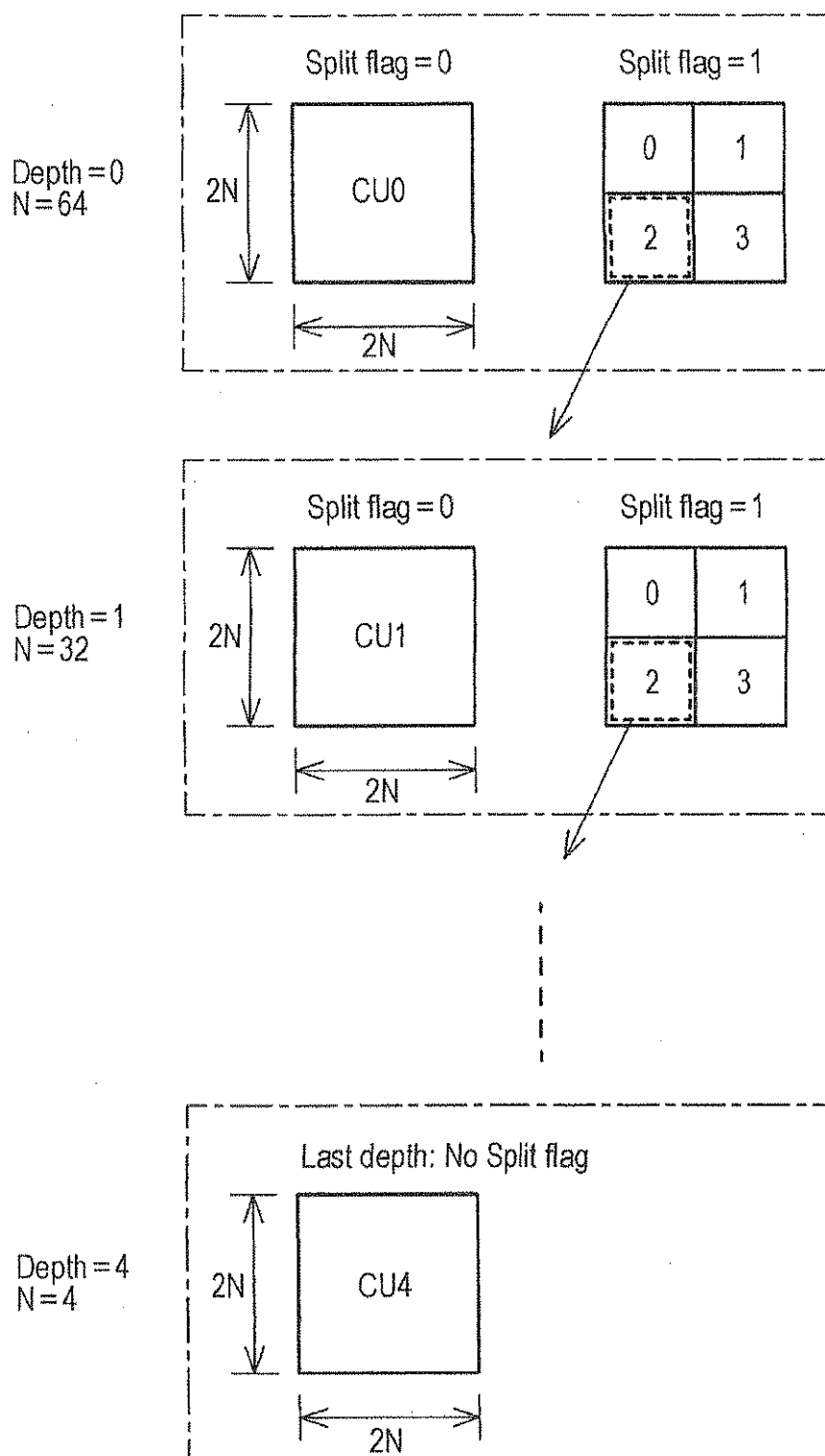


FIG. 6

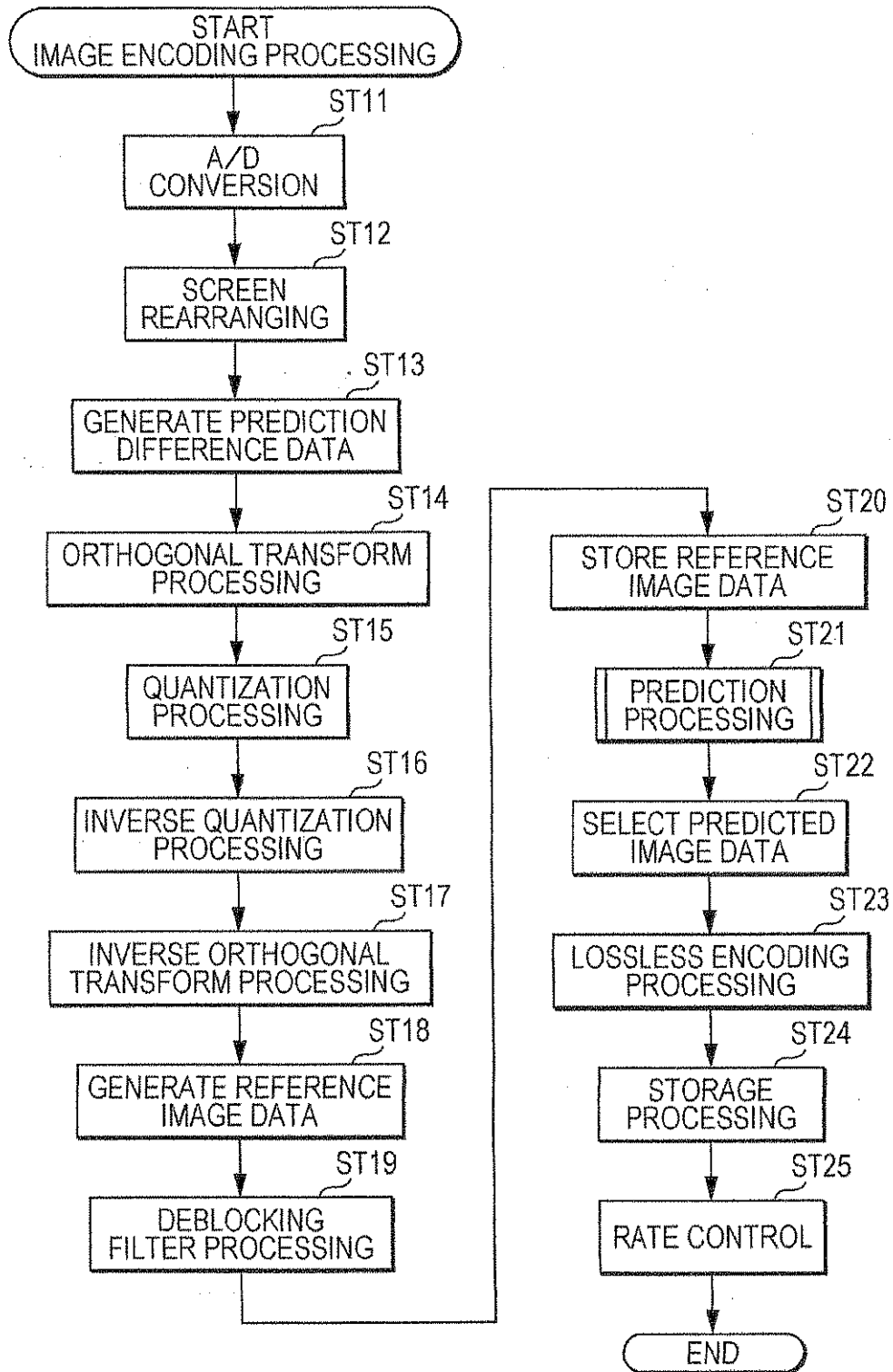


FIG. 7

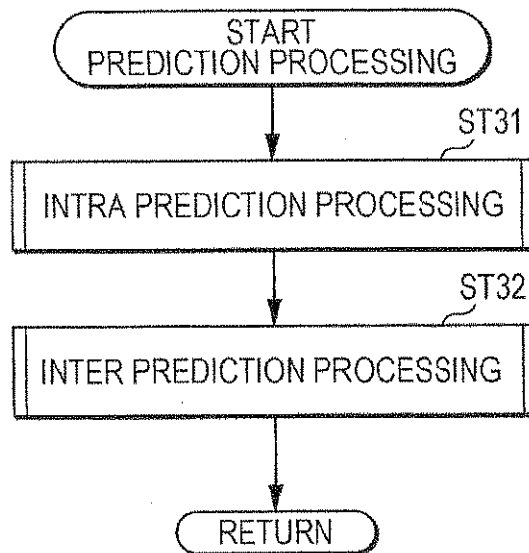


FIG. 8

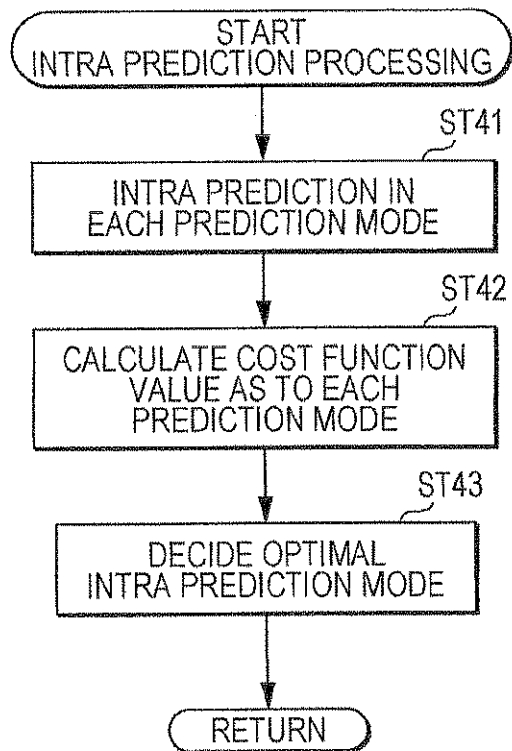


FIG. 9

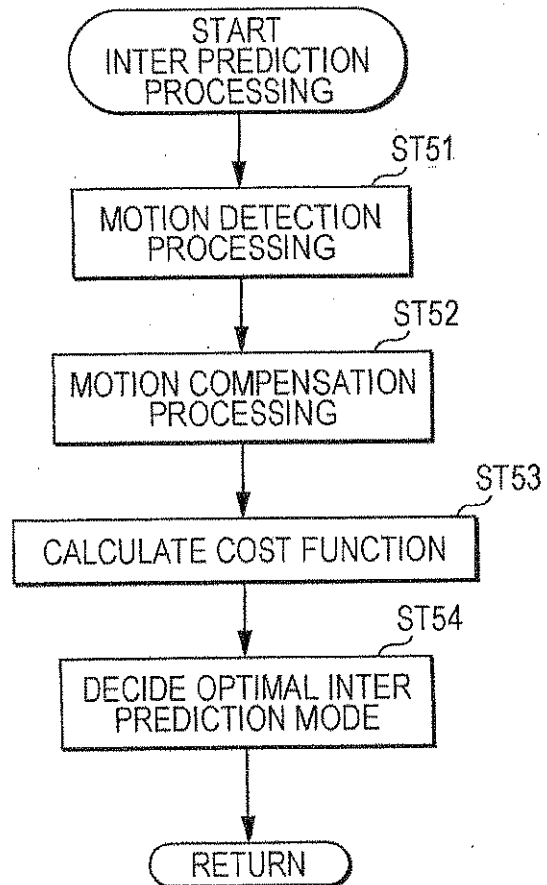
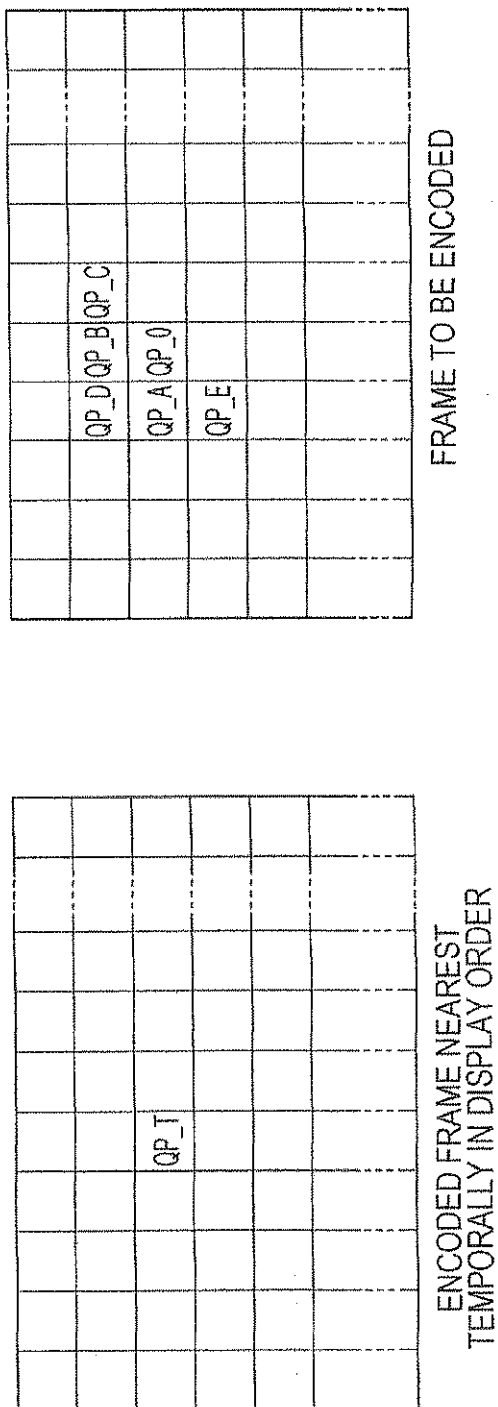


FIG. 10



1. 
 2. 
 3. 
 4. 
 5. 
 6. 
 7. 
 8. 
 9. 
 10. 
 11. 
 12. 
 13. 
 14. 
 15. 
 16. 
 17. 
 18. 
 19. 
 20. 
 21. 
 22. 
 23. 
 24. 
 25. 
 26. 
 27. 
 28. 
 29. 
 30. 
 31. 
 32. 
 33. 
 34. 
 35. 
 36. 
 37. 
 38. 
 39. 
 40. 
 41. 
 42. 
 43. 
 44. 
 45. 
 46. 
 47. 
 48. 
 49. 
 50. 
 51. 
 52. 
 53. 
 54. 
 55. 
 56. 
 57. 
 58. 
 59. 
 60. 
 61. 
 62. 
 63. 
 64. 
 65. 
 66. 
 67. 
 68. 
 69. 
 70. 
 71. 
 72. 
 73. 
 74. 
 75. 
 76. 
 77. 
 78. 
 79. 
 80. 
 81. 
 82. 
 83. 
 84. 
 85. 
 86. 
 87. 
 88. 
 89. 
 90. 
 91. 
 92. 
 93. 
 94. 
 95. 
 96. 
 97. 
 98. 
 99.

31	-	31	31	31	31	31		
31	33	31	-	31	-	31		
31	31	-	31	31	31	33	31	
31	33	31	31	31	-	31	31	

ENCODED FRAME NEAREST

32	32	32	32	32	32	-	40	46
32	-	32	35	40	40	45	46	
-	32	32	32	BK0 33				
32	32	32	-					

FRAME TO BE ENCODED

FIG. 12

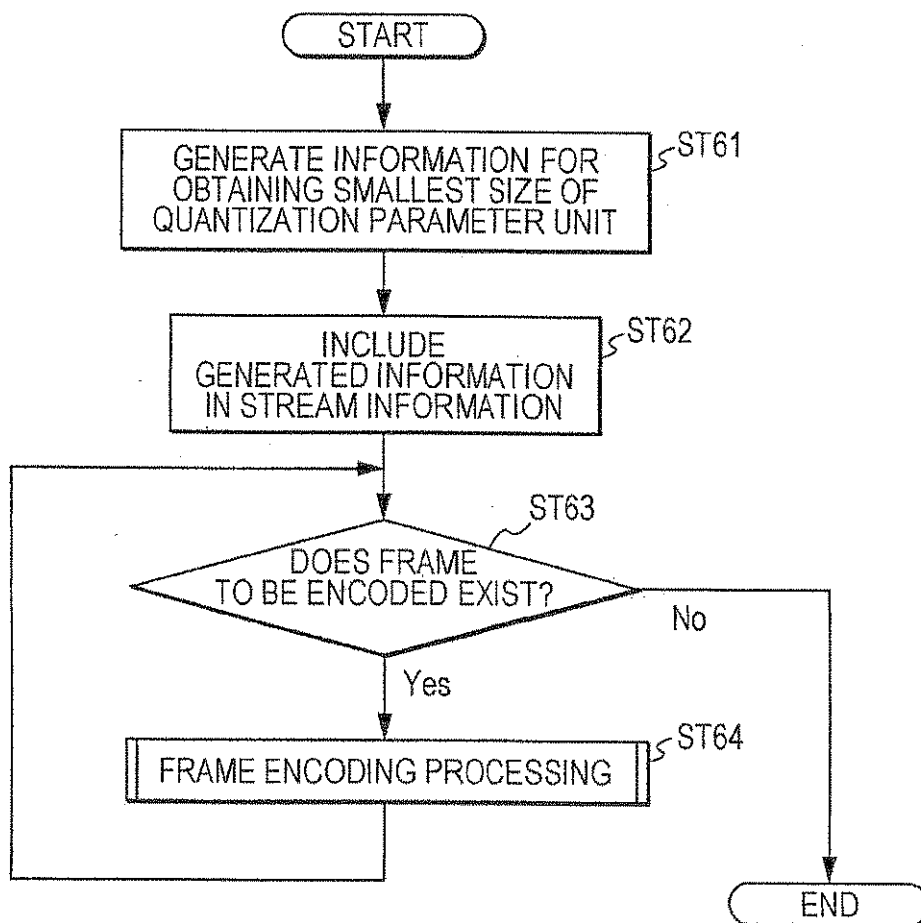


FIG. 13

seq_parameter_set_rbsp() {	Descriptor
profile_idc	u(8)
reserved_zero_8bits /* equal to 0 */	u(8)
level_idc	u(8)
seq_parameter_set_id	ue(v)
pic_width_in_luma_samples	u(16)
pic_height_in_luma_samples	u(16)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_frame_num_minus4	ue(v)
pic_order_cnt_type	ue(v)
if(pic_order_cnt_type == 0)	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
else if(pic_order_cnt_type == 1) {	
delta_pic_order_always_zero_flag	u(1)
offset_for_non_ref_pic	se(v)
num_ref_frames_in_pic_order_cnt_cycle	ue(v)
for(i = 0; i < num_ref_frames_in_pic_order_cnt_cycle; i++)	
offset_for_ref_frame[i]	se(v)
}	
max_num_ref_frames	ue(v)
gaps_in_frame_num_value_allowed_flag	u(1)
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
max_transform_hierarchy_depth_inter	ue(v)
max_transform_hierarchy_depth_intra	ue(v)
adaptive_loop_filter_enabled_flag	u(1)
cu_qp_delta_enabled_flag	u(1)
rbbsp_trailing_bits()	
}	

FIG. 14

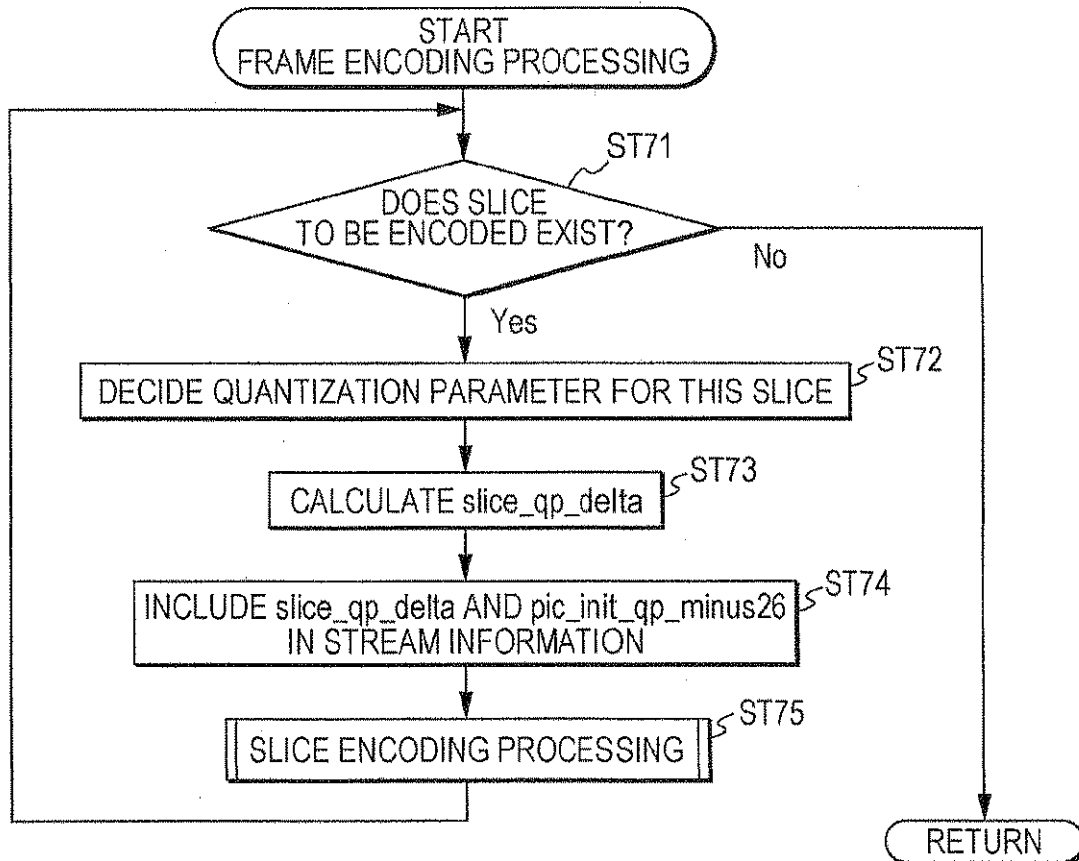


FIG. 15

pic_parameter_set_rbsp() {
pic_parameter_set_id
seq_parameter_set_id
entropy_coding_mode_flag
num_ref_idx_l0_default_active_minus1
num_ref_idx_l1_default_active_minus1
pic_init_qp_minus26 /* relative to 26 */
constrained_intra_pred_flag
rbsp_trailing_bits()
}

FIG. 16

14 / 31

slice_header() {	Descriptor
first_tb_in_slice	ue(v)
entropy_slice_flag	u(1)
if(!entropy_slice_flag) {	
slice_type	ue(v)
pic_parameter_set_id	ue(v)
frame_num	u(v)
if(IdrPicFlag)	
idr_pic_id	ue(v)
if(pic_order_cnt_type == 0)	
pic_order_cnt_lsb /*	u(v)
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
}	
ref_pic_list_modification()	
ref_pic_list_combination()	
if(nal_ref_idc != 0)	
dec_ref_pic_marking()	
if(entropy_coding_mode_flag && slice_type != I)	
cabac_init_idc	ue(v)
slice_qp_delta	se(v)
if(adaptive_loop_filter_enabled_flag)	
alf_param()	
if(deblocking_filter_control_present_flag) {	
disable_deblocking_filter_idc	
if(disable_deblocking_filter_idc != 1) {	
slice_alpha_c0_offset_div2	
slice_beta_offset_div2	
}	
}	
if(slice_type == B)	
collocated_from_l0_flag	u(1)
} else	
if(entropy_coding_mode_flag && slice_type != I)	
cabac_init_idc	ue(v)
}	

FIG. 17

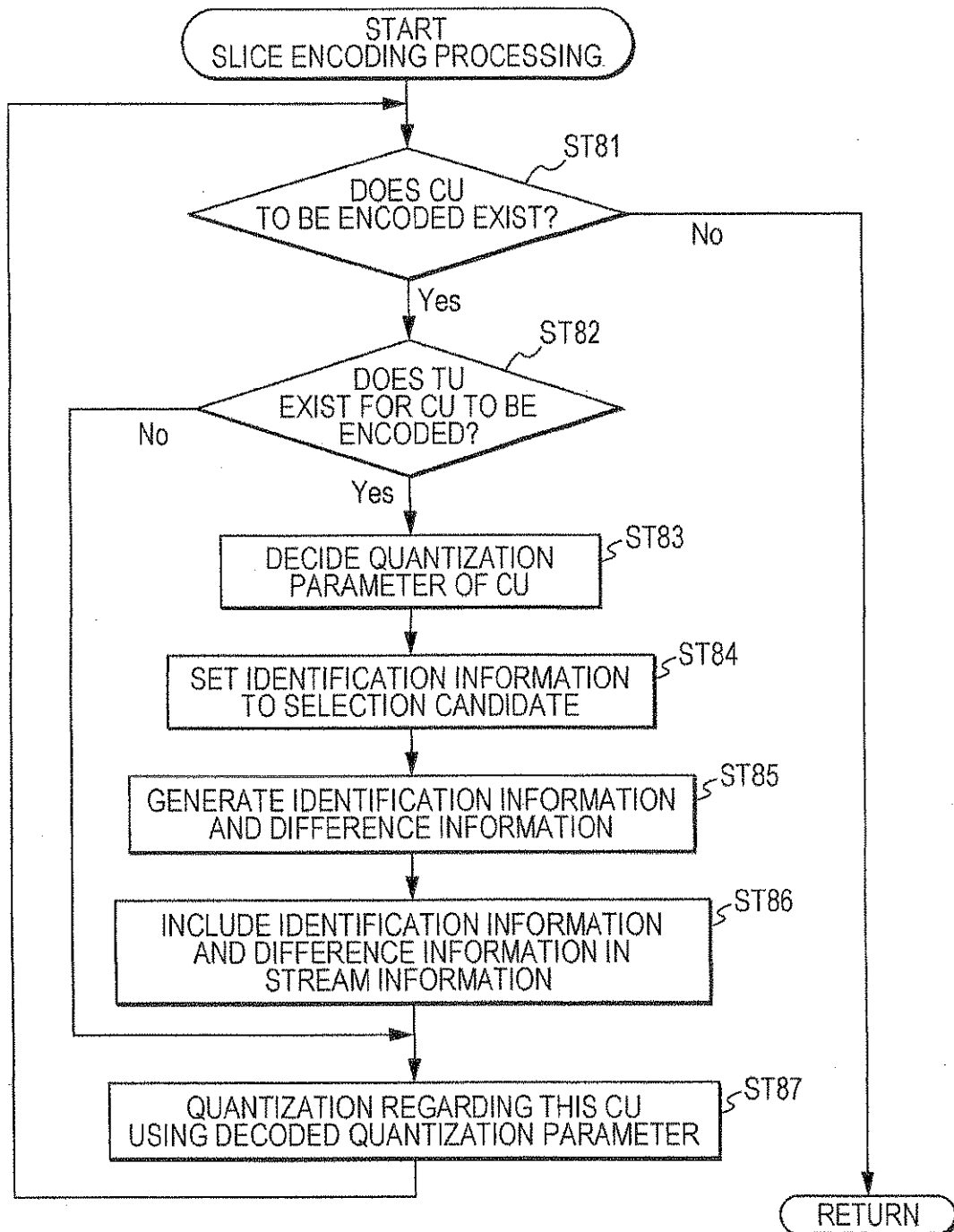


FIG. 18

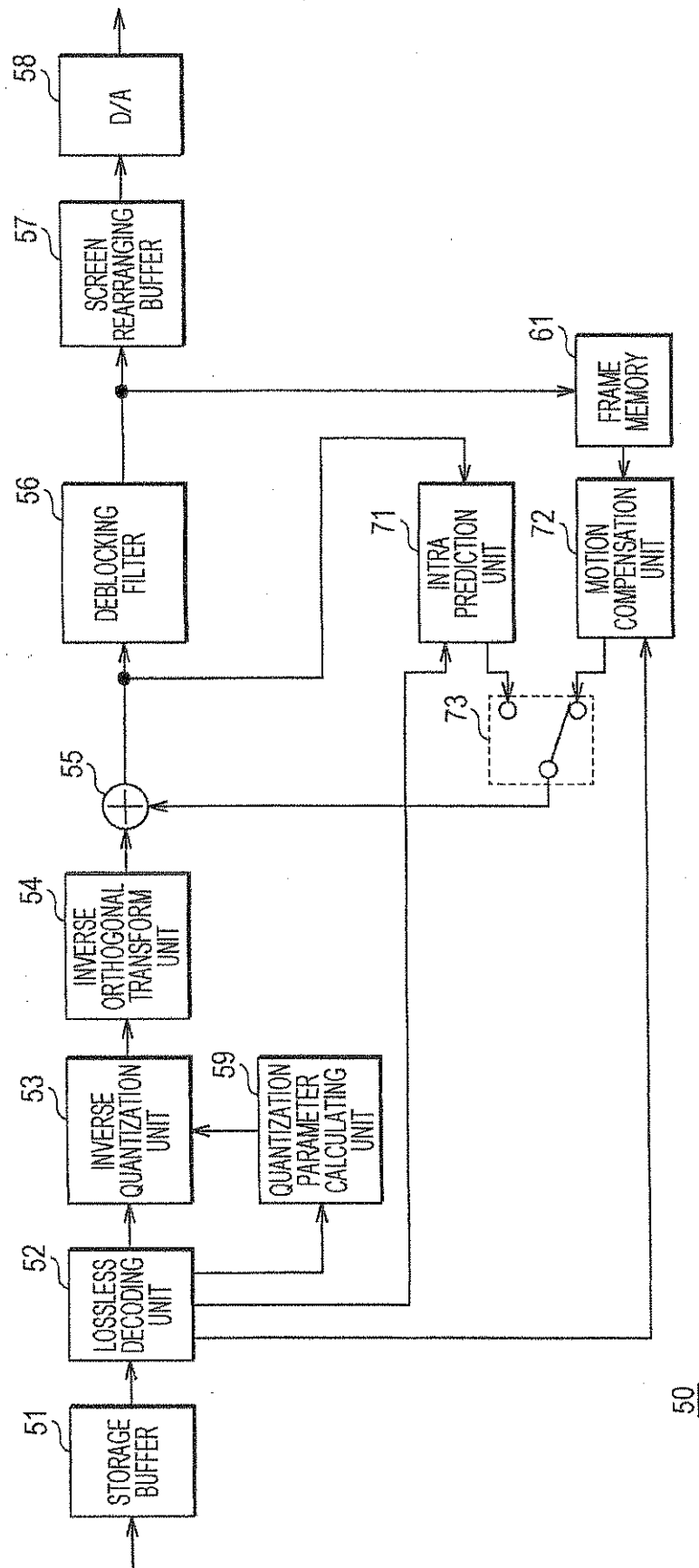


FIG. 19

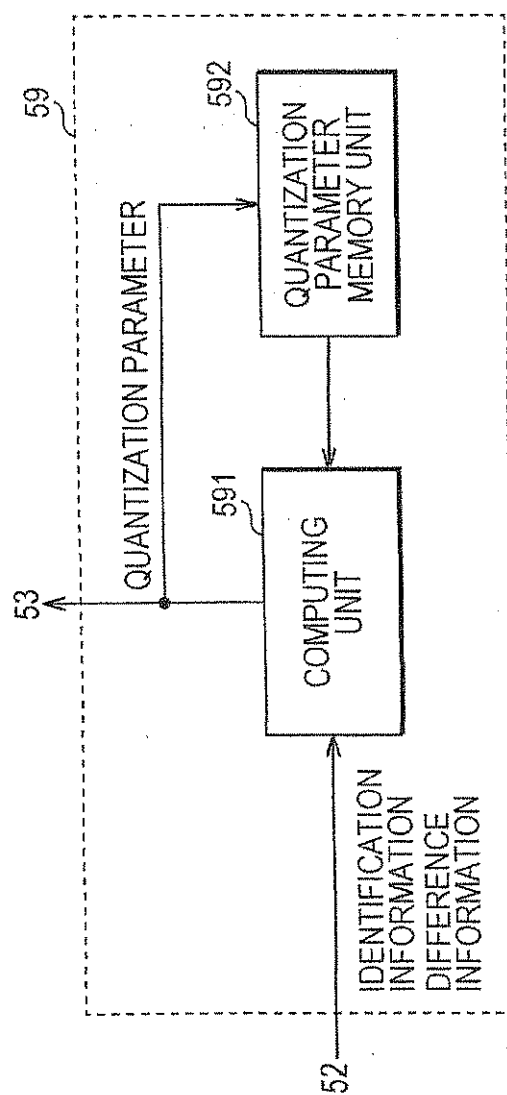


FIG. 20

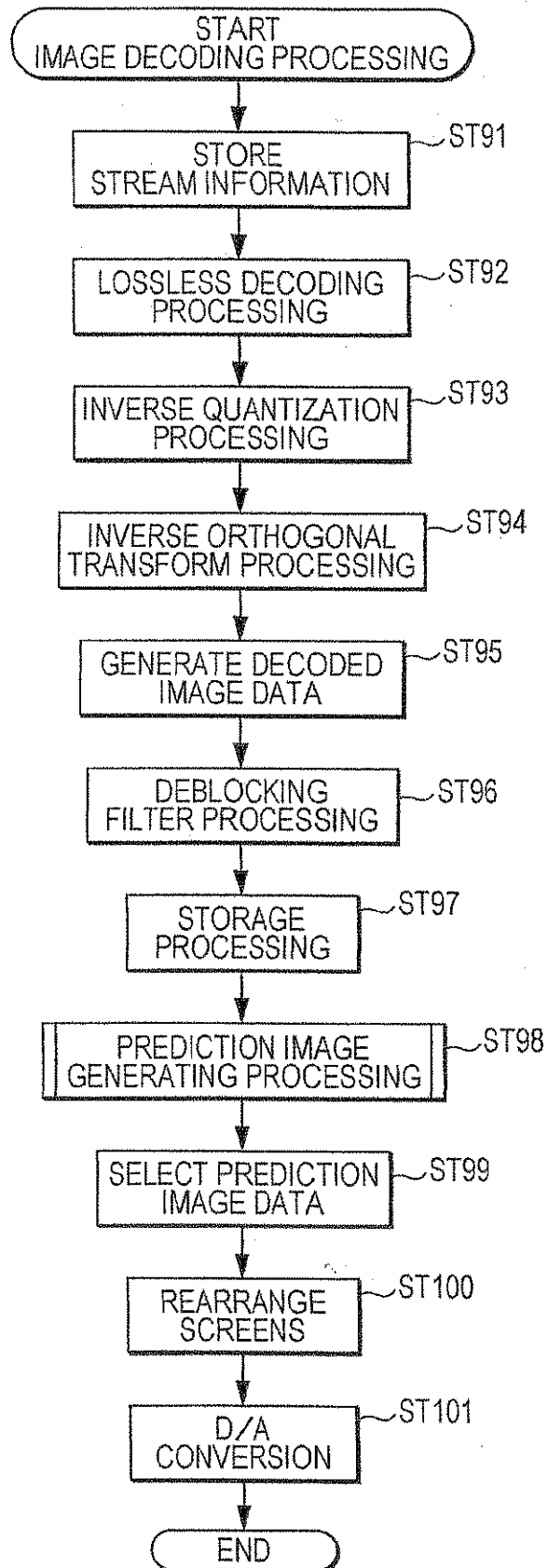


FIG. 21

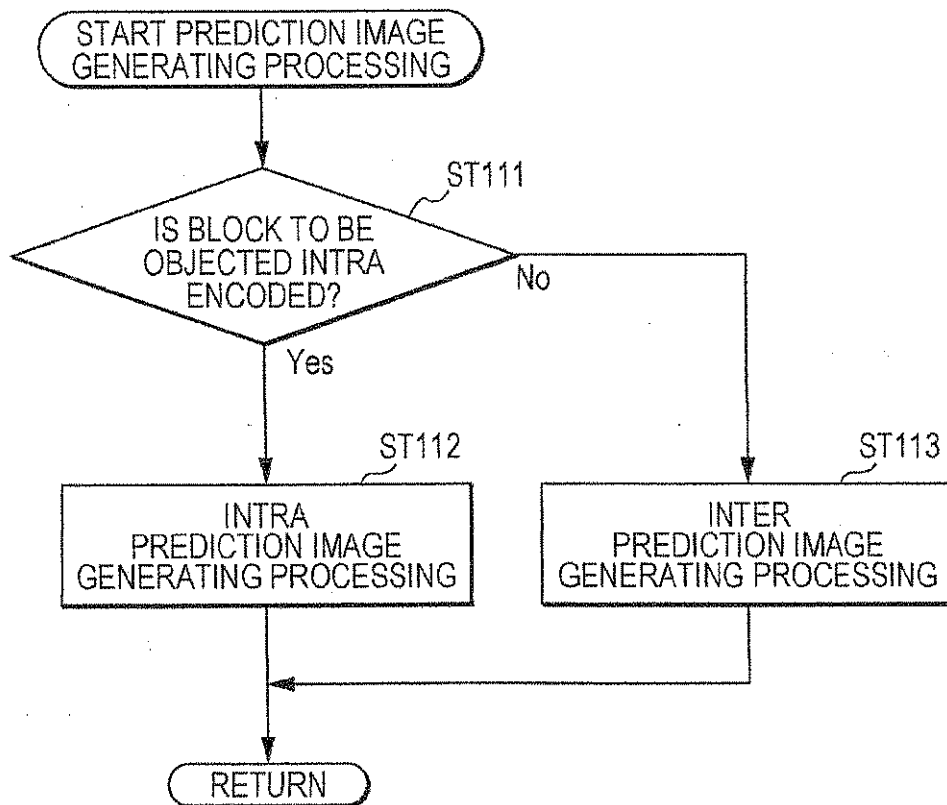
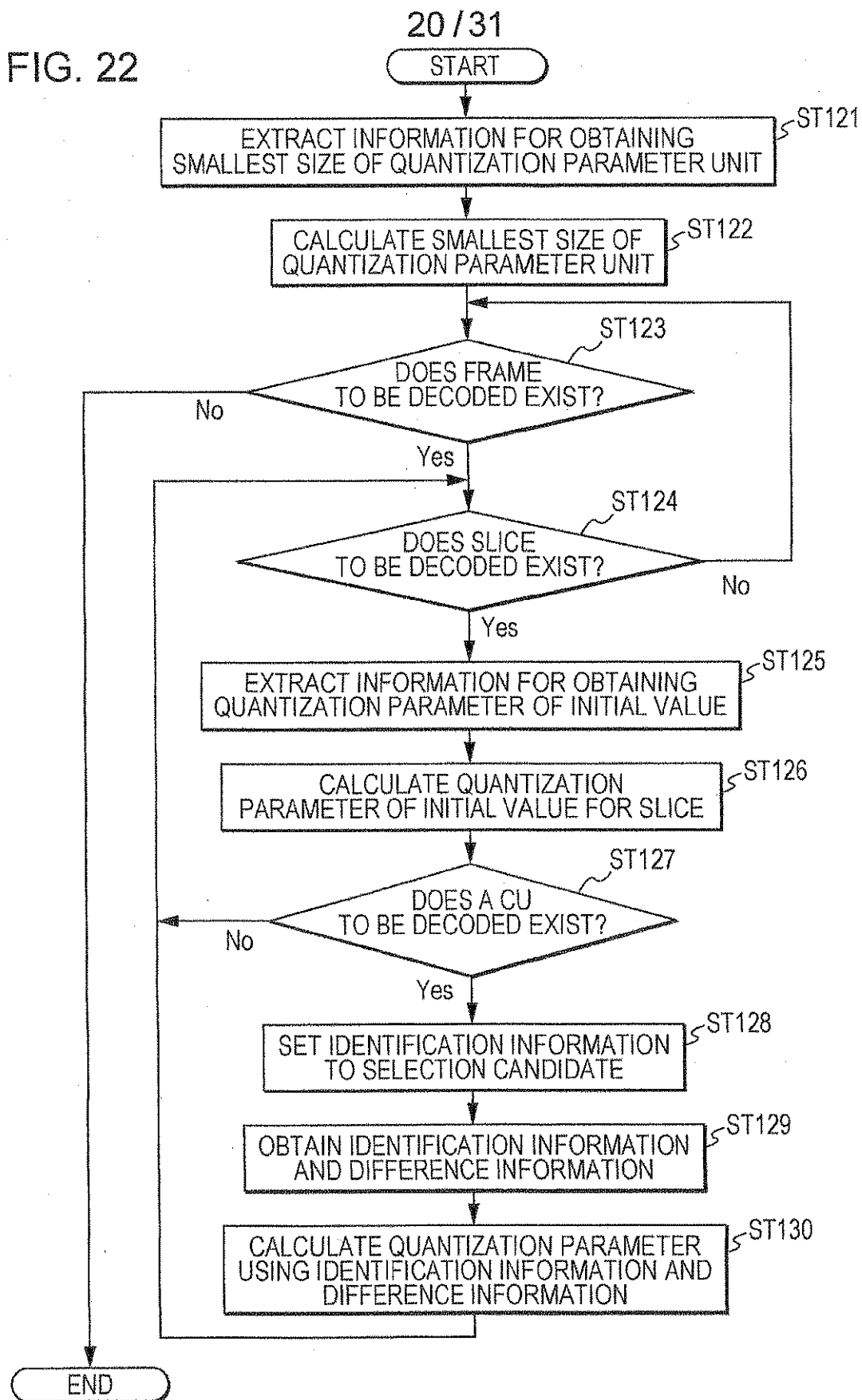


FIG. 22



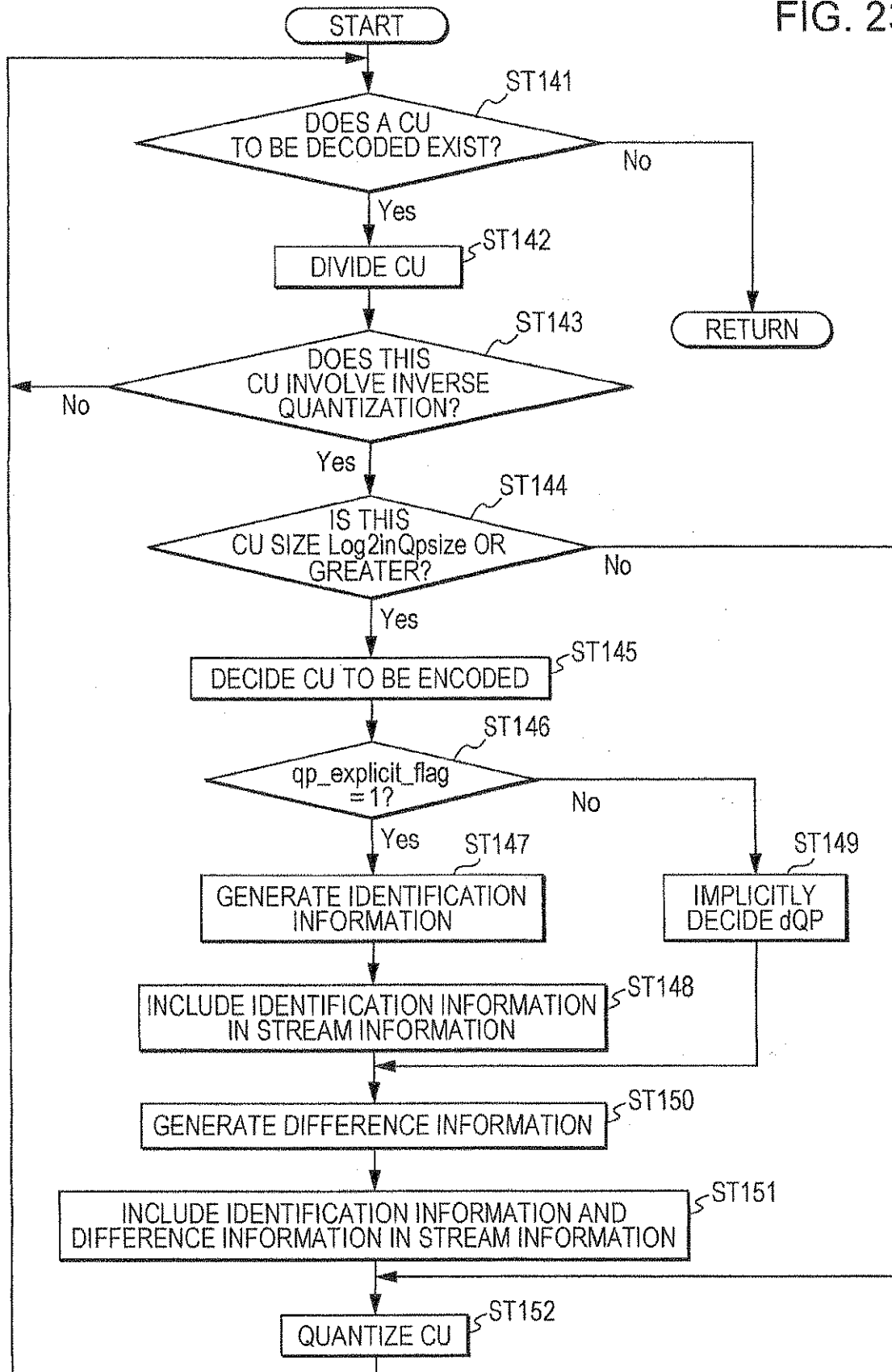


FIG. 24

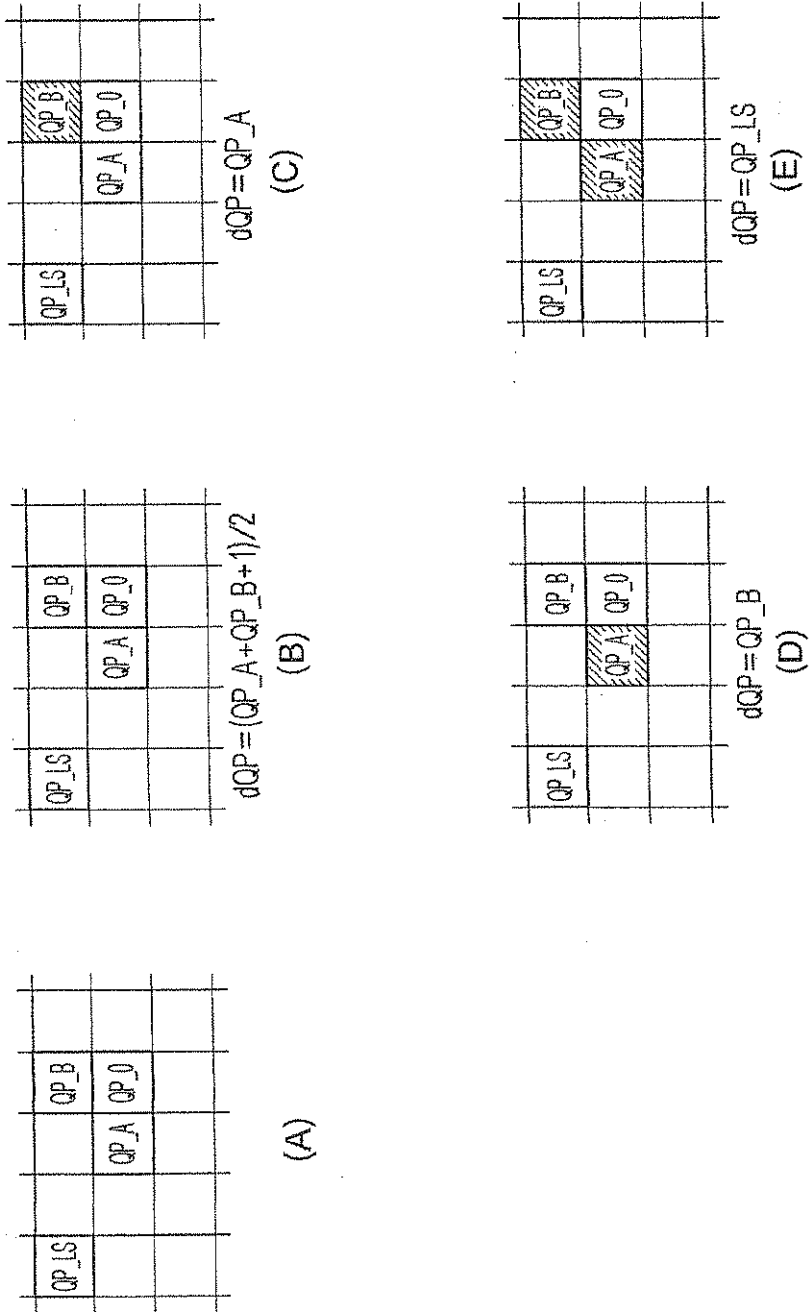


FIG. 25

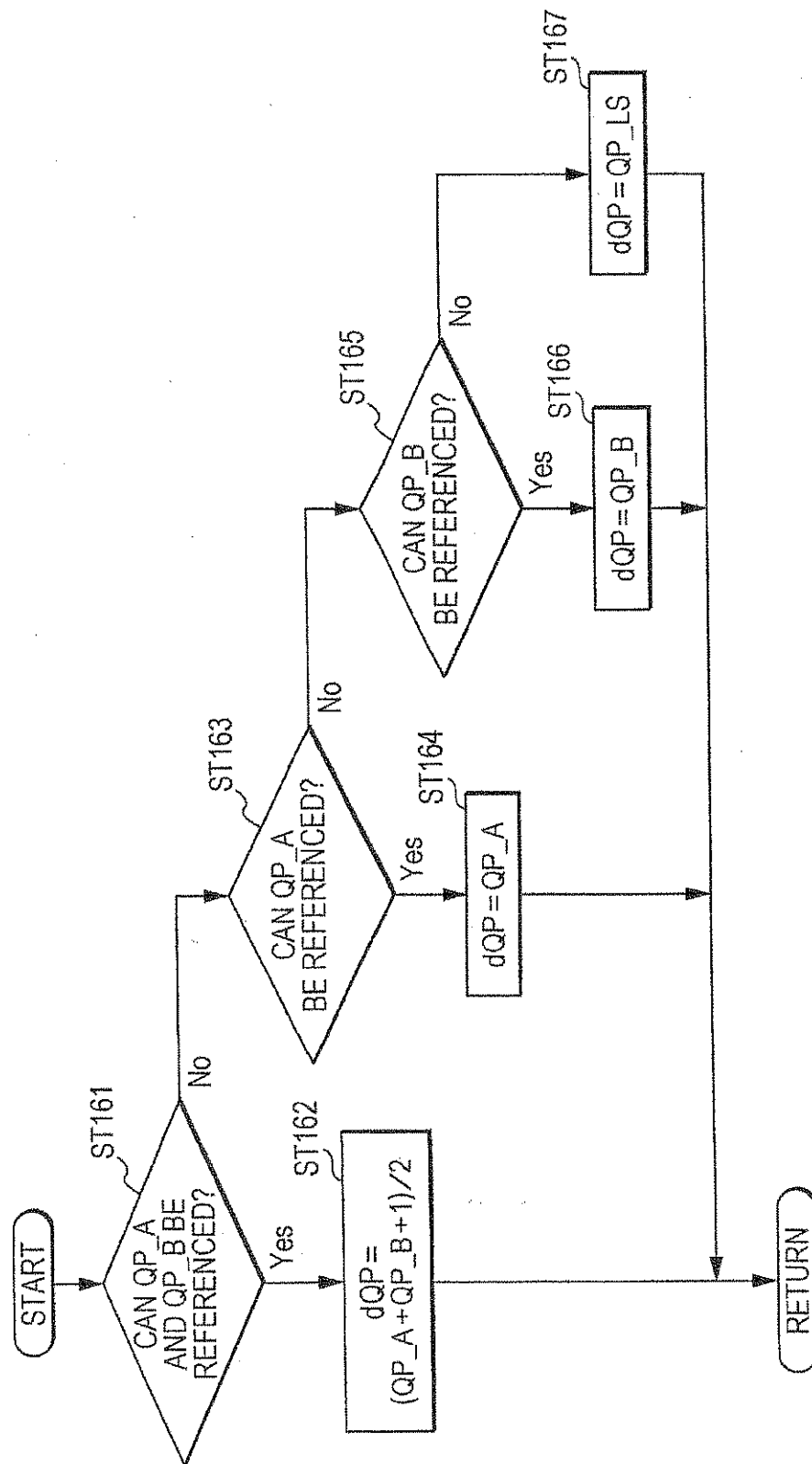


FIG. 26

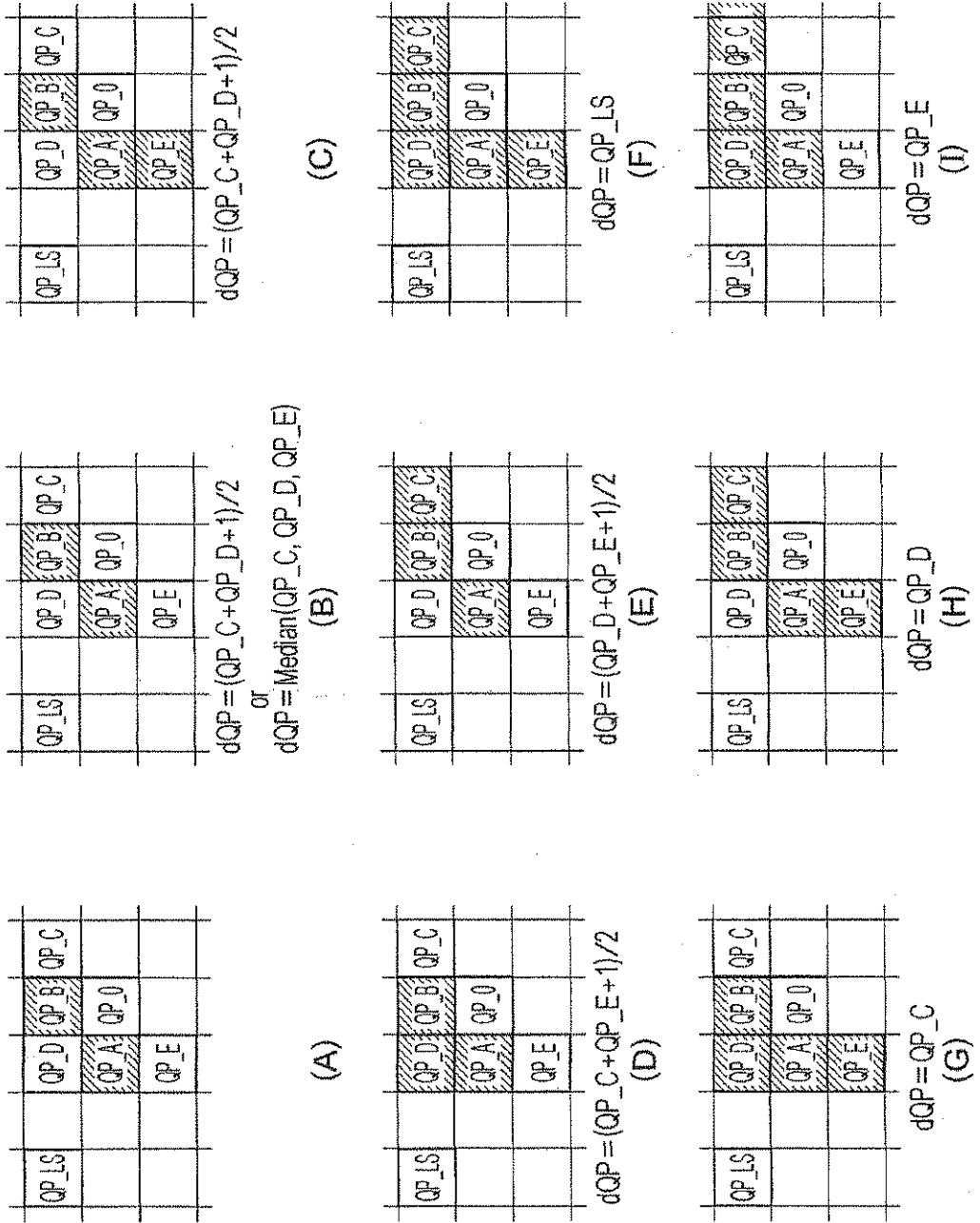


FIG. 27

```

if ( left & above ) dQP = Avg.(QP_A, QP_B)
else if ( left ) dQP = QP_A
else if ( above ) dQP = QP_B
else if (above-left & above-right & below-left )
    dQP = Median or Avg.(QP_C, QP_D))
else if (above-left & above-right ) dQP = Avg.(QP_C, QP_D)
else if (above-right & below-left ) dQP = Avg.(QP_C, QP_E)
else if (above-left & below-left ) dQP = Avg.(QP_D, QP_E)
else dQP = QP_LS

```

FIG. 28

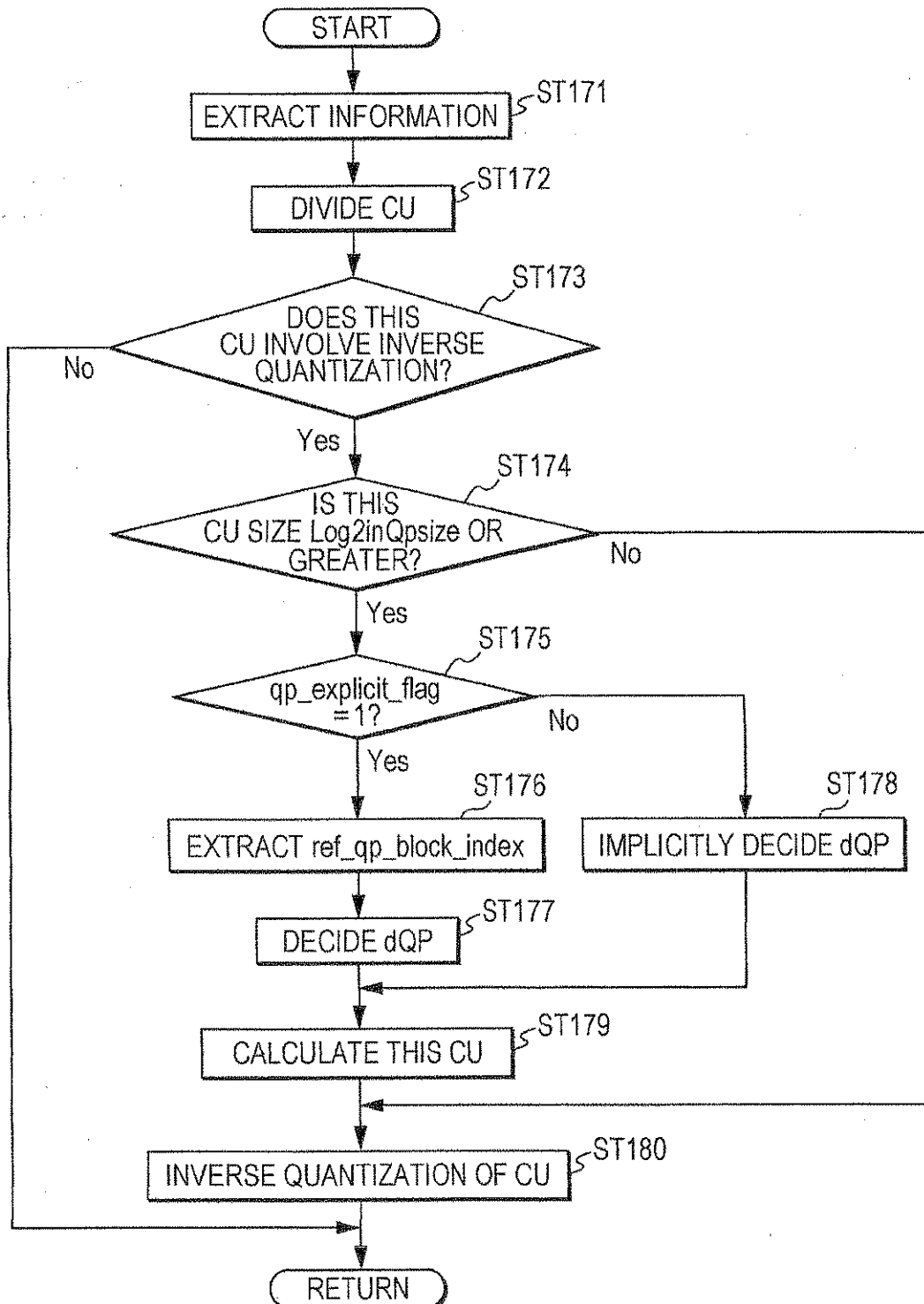


FIG. 29

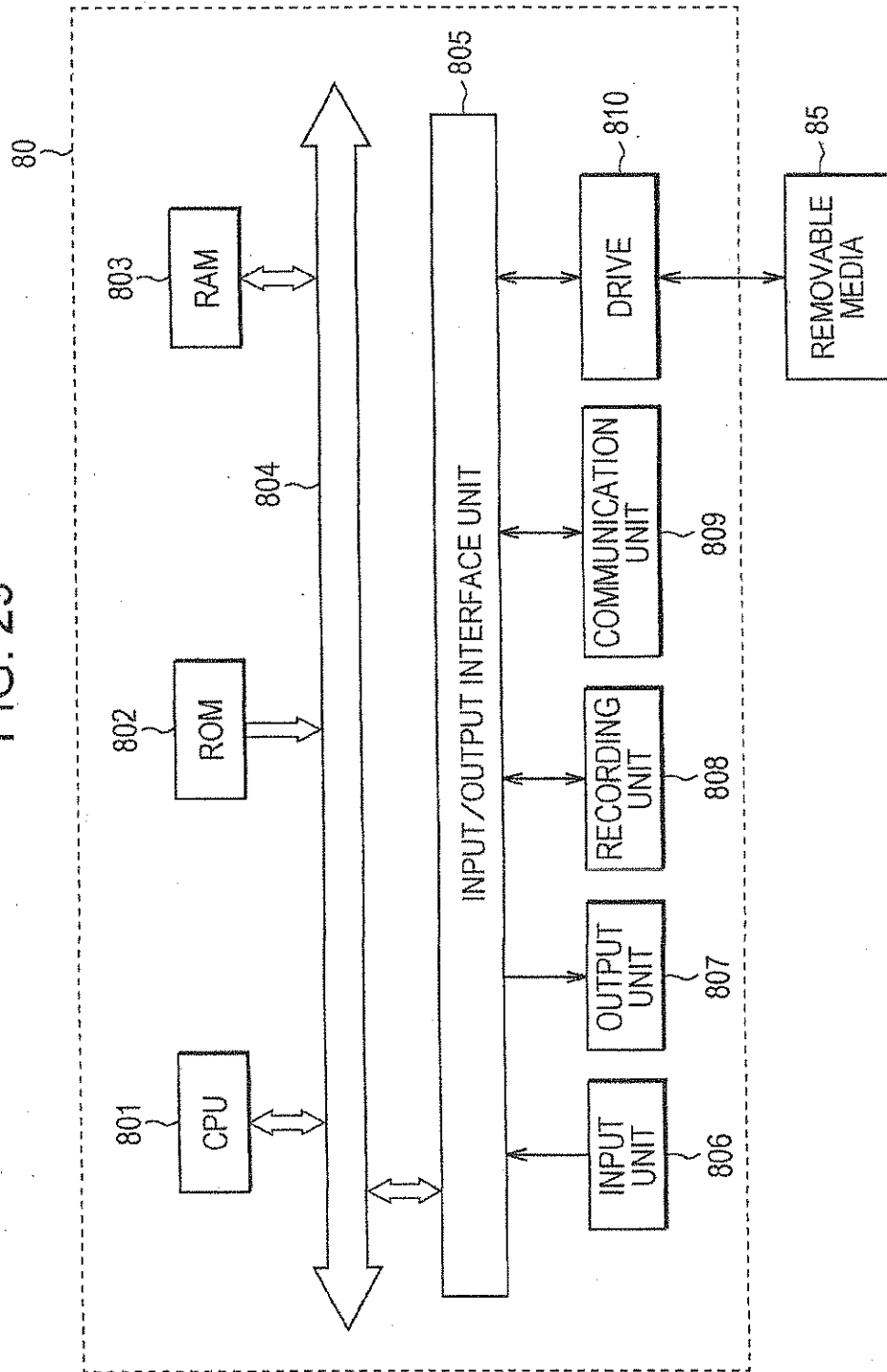


FIG. 30

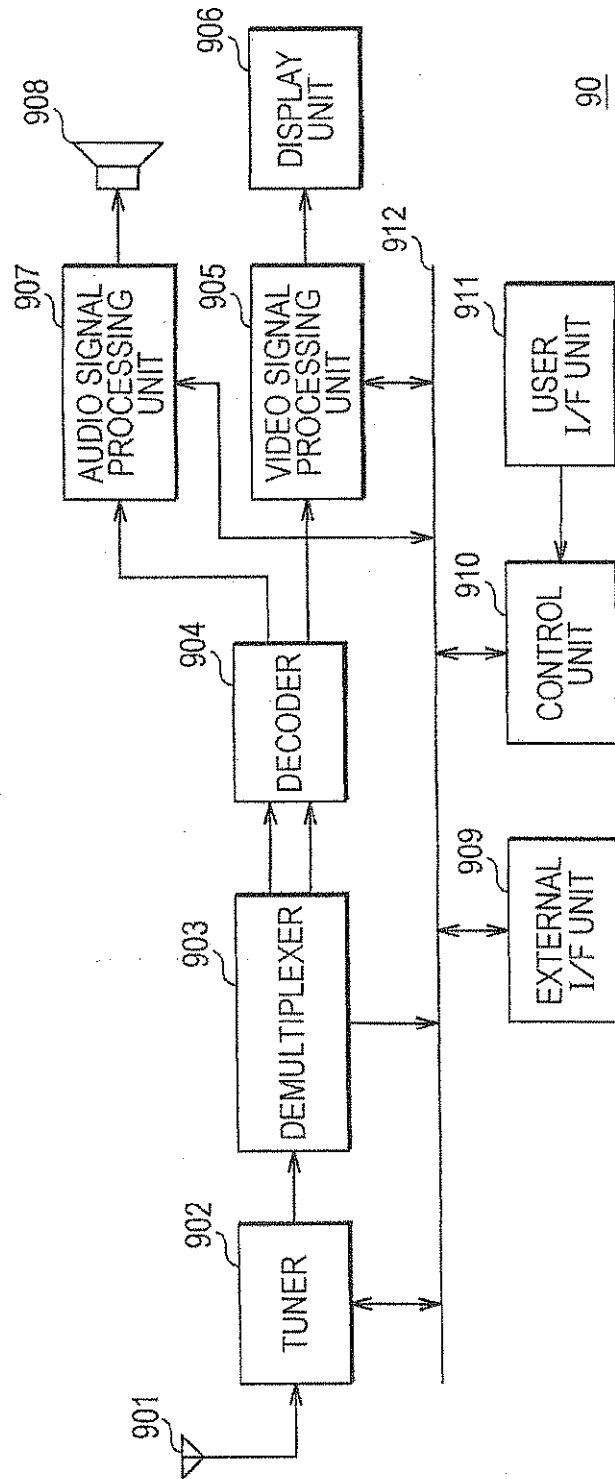


FIG. 31

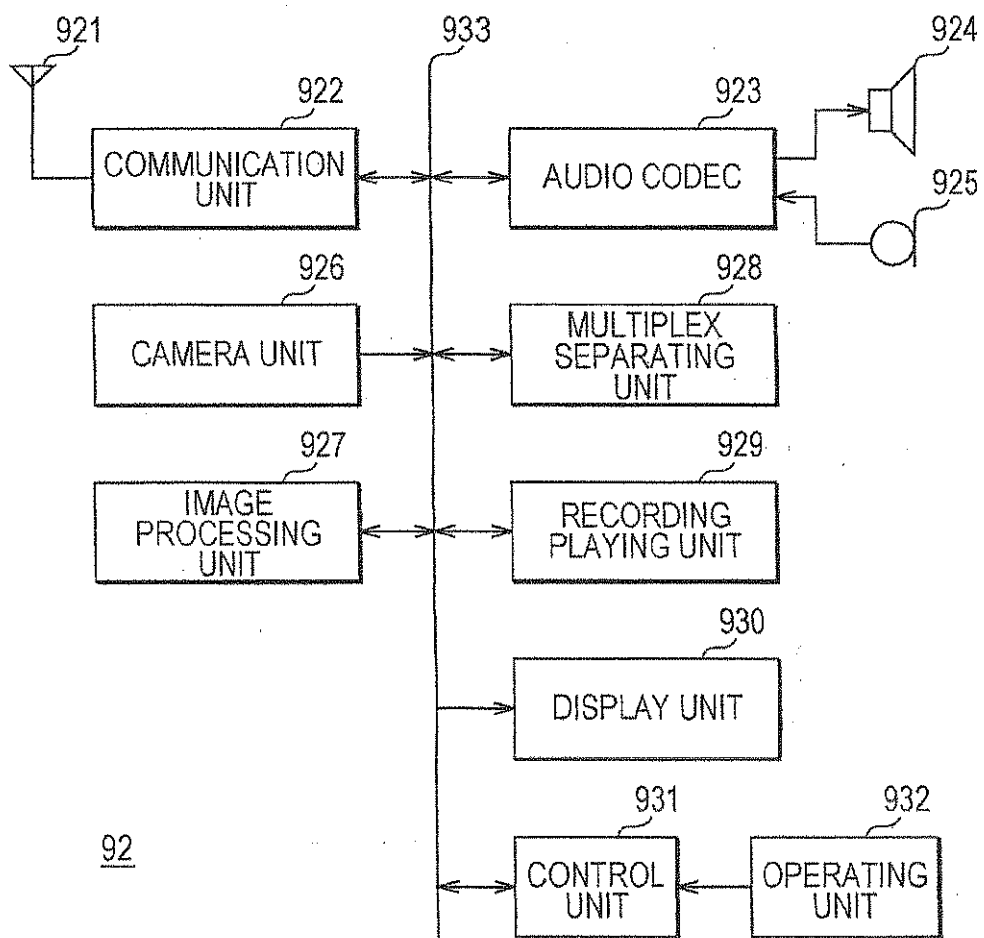


FIG. 32

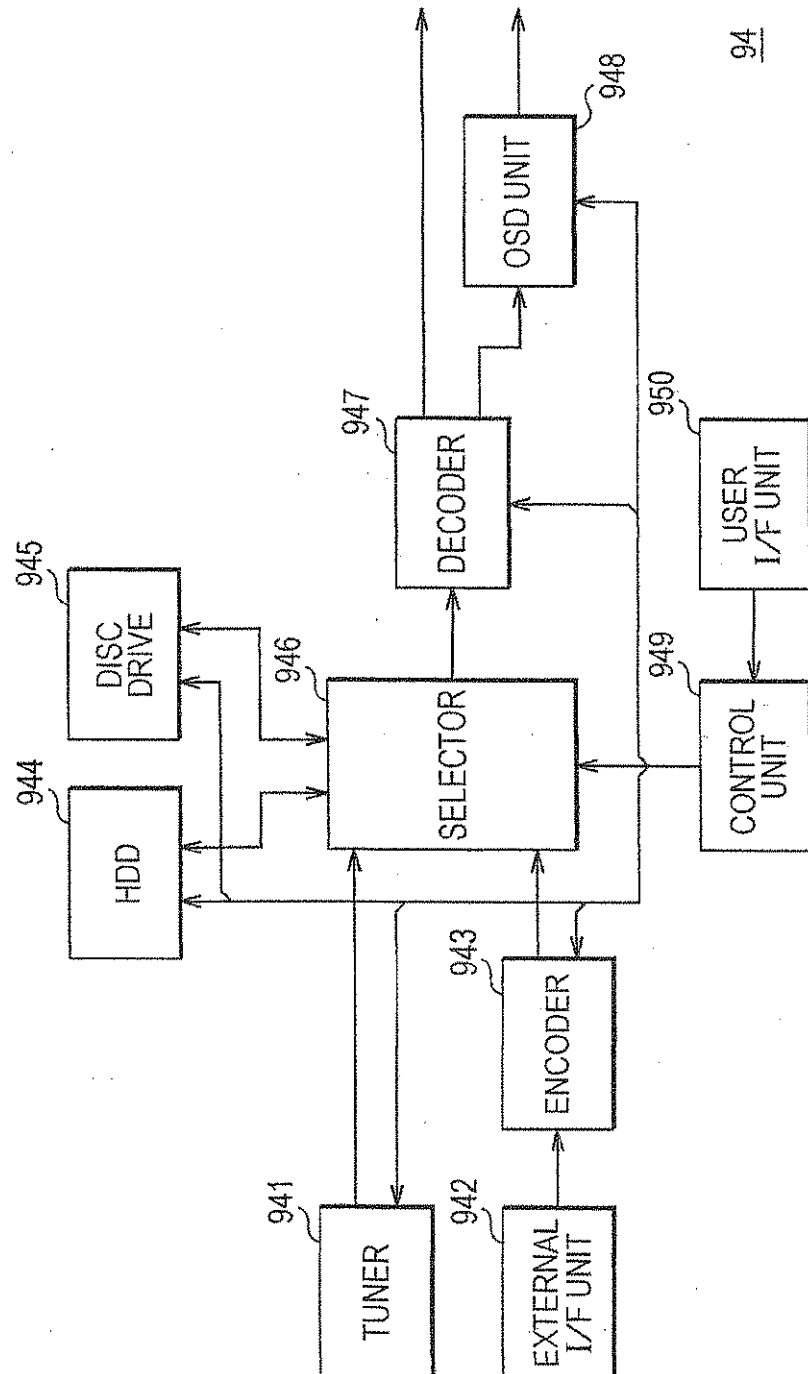


FIG. 33

