

FIG. 1A

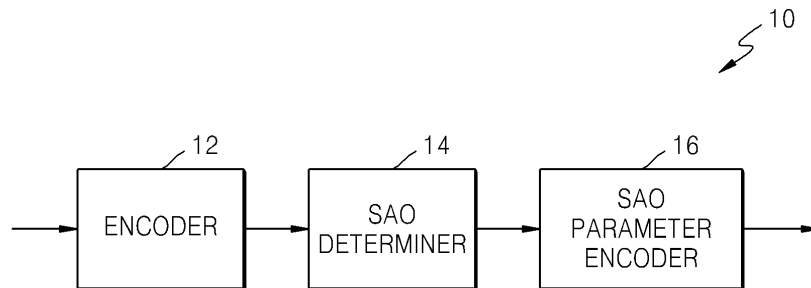


FIG. 1B

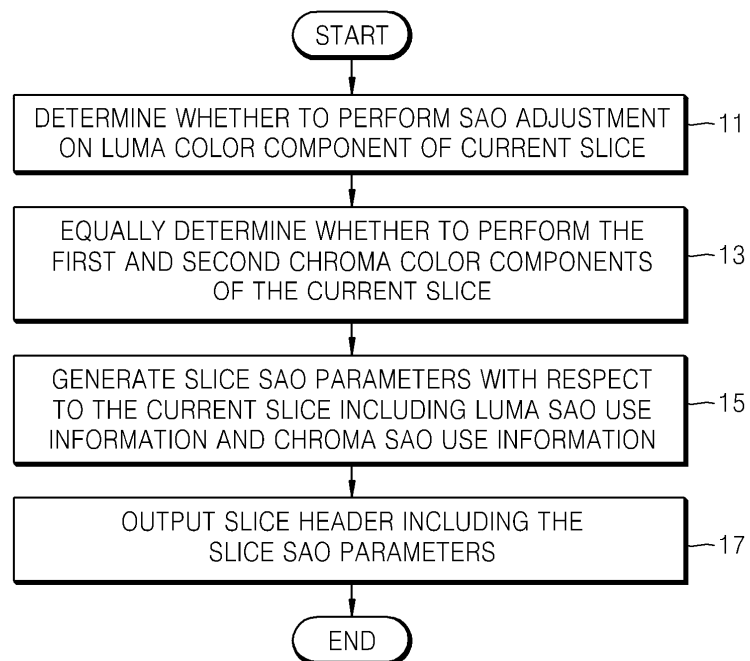


FIG. 2A

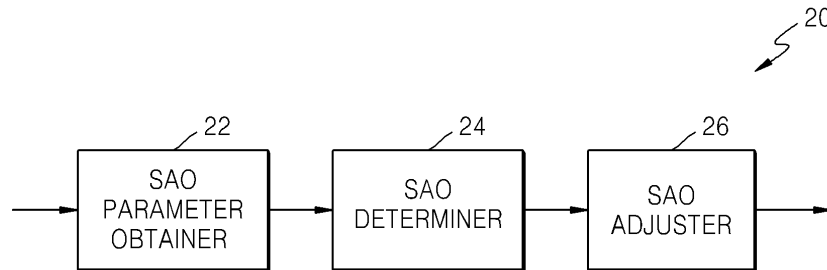


FIG. 2B

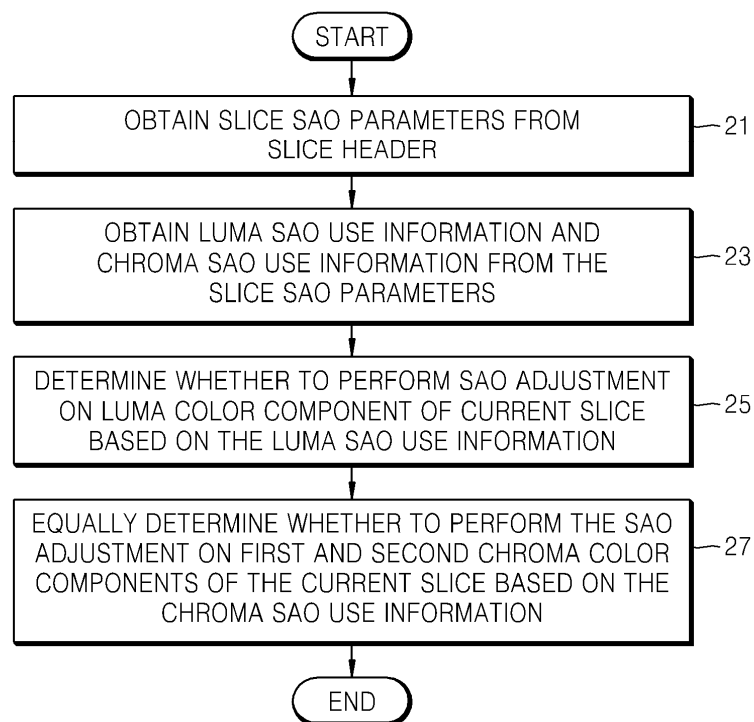


FIG. 3

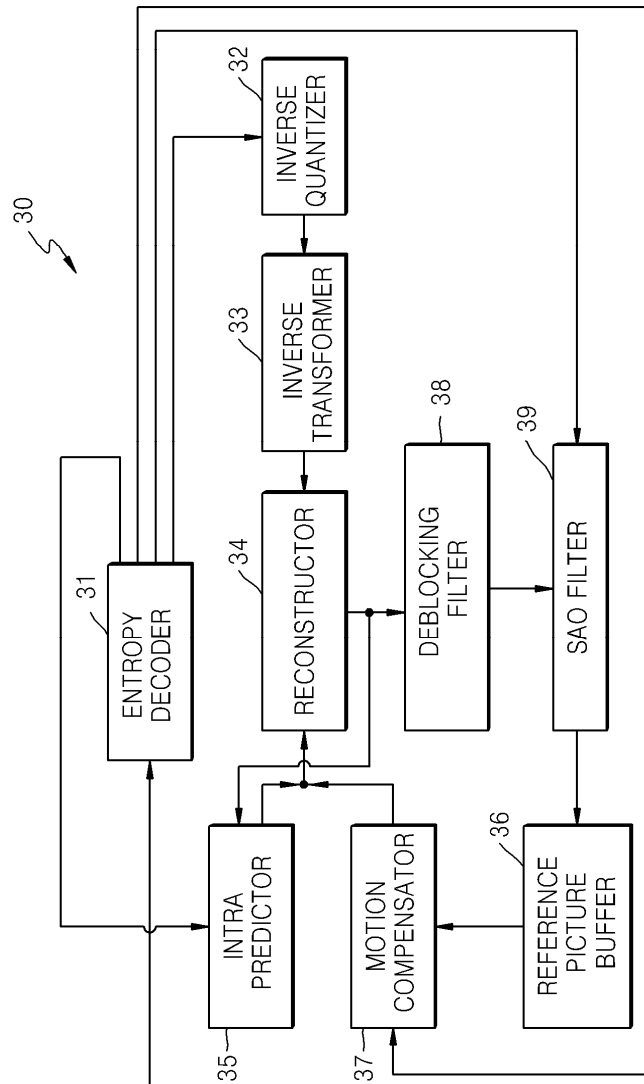


FIG. 4

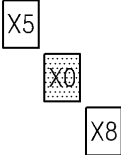
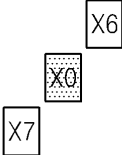
41	42	43	44
EO CLASS = 0	EO CLASS = 1	EO CLASS = 2	EO CLASS = 3
HORIZONTAL	VERTICAL	135° DIAGONAL	45° DIAGONAL
			

FIG. 5A

CATEGORY	CONDITION
1	$X_c < X_a \ \&\& \ X_c < X_b$
2	$(X_c < X_a \ \&\& \ X_c == X_b) \parallel (X_c == X_a \ \&\& \ X_c < X_b)$
3	$(X_c > X_a \ \&\& \ X_c == X_b) \parallel (X_c == X_a \ \&\& \ X_c > X_b)$
4	$X_c > X_a \ \&\& \ X_c > X_b$
0	IF CONDITIONS FOR CATEGORIES 1, 2, 3, AND 4 ARE NOT SATISFIED

FIG. 5B

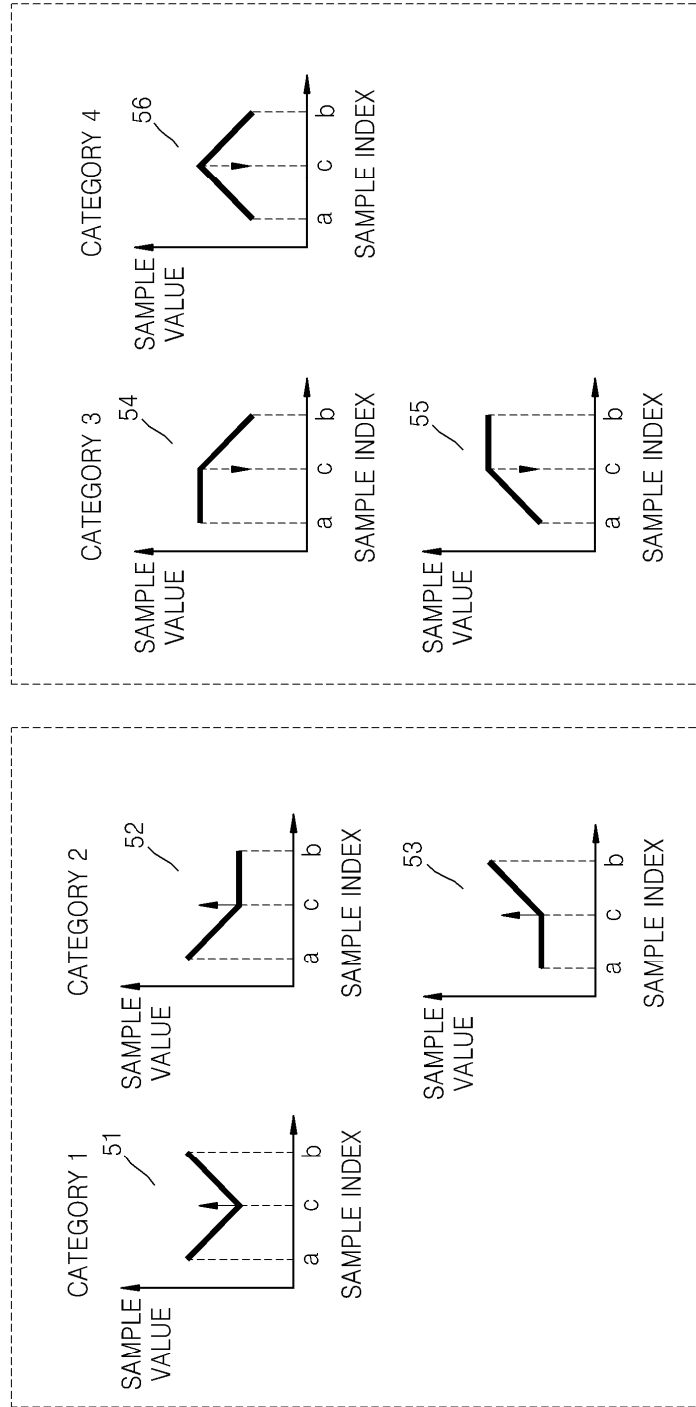


FIG. 6A

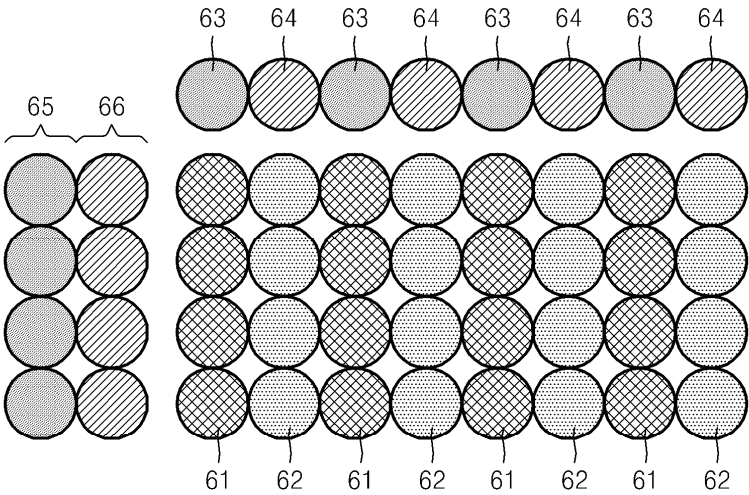


FIG. 6B

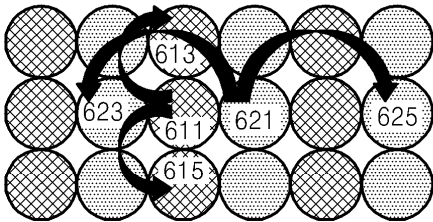


FIG. 6C

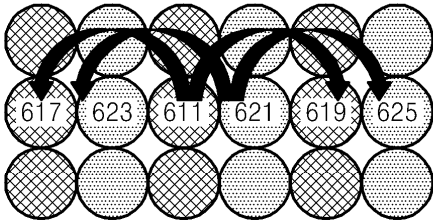


FIG. 7A

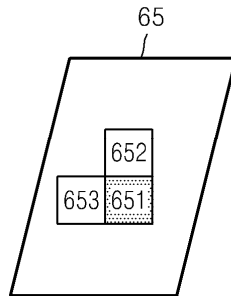


FIG. 7B

700	slice_header() {
	...
	if(sample_adaptive_offset_enabled_flag) {
	slice_sample_adaptive_offset_flag[0] ~ 701
	if(slice_sample_adaptive_offset_flag[0]) {
	slice_sample_adaptive_offset_flag[1] ~ 702
	slice_sample_adaptive_offset_flag[2] = slice_sample_adaptive_offset_flag[1]
	} 703
	}
	
	}
705	slice_data() {...
	for(cldx = 0; cldx < 3; cldx++)
	if(slice_sample_adaptive_offset_flag[cldx])
	sao_unit_cabac(xCtb, yCtb, cldx) ~ 706
	...}

FIG. 7C

706	sao_unit_cabac(rx, ry, cldx){
	if(rx > 0) {
	if(CtbAddrInSlice != 0 &&
	TileId[CtbAddrTS] == TileId[CtbAddrRStoTS[CtbAddrRS - 1]])
	sao_merge_left_flag [rx][ry] — 707
	}
	if(!sao_merge_left_flag [rx][ry] {
	if(ry > 0) {
	if(((CtbAddrTS - CtbAddrRStoTS [CtbAddrRS- PicWidthInCtbs]) <=
	CtbAddrInSlice) &&
	(TileId[CtbAddrTS] == TileId[CtbAddrRStoTS[CtbAddrRS-
	PicWidthInCtbs]]))
	sao_merge_up_flag [rx][ry] — 708
	}
	if(!sao_merge_up_flag [rx][ry])
	sao_offset_cabac(rx, ry, cldx) — 709
	}
	}

FIG. 7D

sao_offset_cabac(rx, ry, cldx) {
sao_type_idx[cldx][rx][ry] —711
...
if(sao_type_idx[cldx][rx][ry] != 0) {
for(i = 0; i < 4; i++)
sao_offset[cldx][rx][ry][i] —713
}
if(sao_type_idx[cldx][rx][ry] == 5) {
for(i = 0; i < 4; i++) {
if(sao_offset[cldx][rx][ry] != 0)
sao_offset_sign[cldx][rx][ry][i] —715
}
}
}

FIG. 7E

if(sao_type_idx[cldx][rx][ry] != 0) {
for(i = 0; i < 4; i++)
sao_offset_abs_1st_bin[cldx][rx][ry][i] —721
for(i = 0; i < 4; i++)
if(sao_offset_abs_1st_bin[cldx][rx][ry][i])
sao_offset_abs_remain_bins[cldx][rx][ry][i] —723
}

FIG. 7F

sao_unit_cabac(rx, ry, cldx)	706
...	
if(!sao_merge_up_flag){	
sao_type_idx[cldx][rx][ry]	711
if(sao_type_idx[cldx][rx][ry] != 0) {	
for(i = 0; i < 4; i++)	
sao_offset[cldx][rx][ry][i]	713
}	
if(sao_type_idx[cldx][rx][ry] == 5) {	
for(i = 0; i < 4; i++) {	
if(sao_offset[cldx][rx][ry] != 0)	
sao_offset_sign[cldx][rx][ry][i]	715
}	
sao_band_position[cldx][rx][ry]	717
}	
}	
}	

FIG. 8

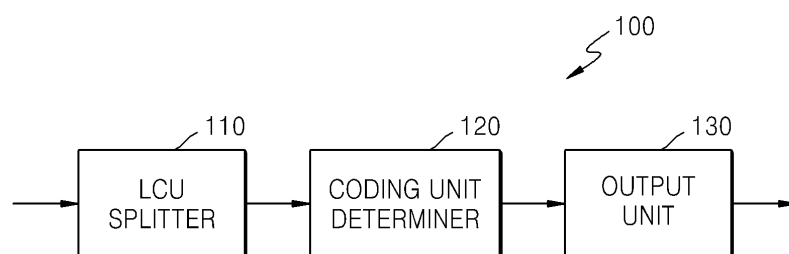


FIG. 9

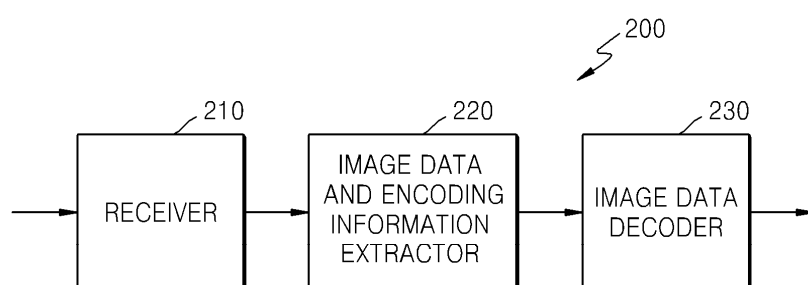


FIG. 10

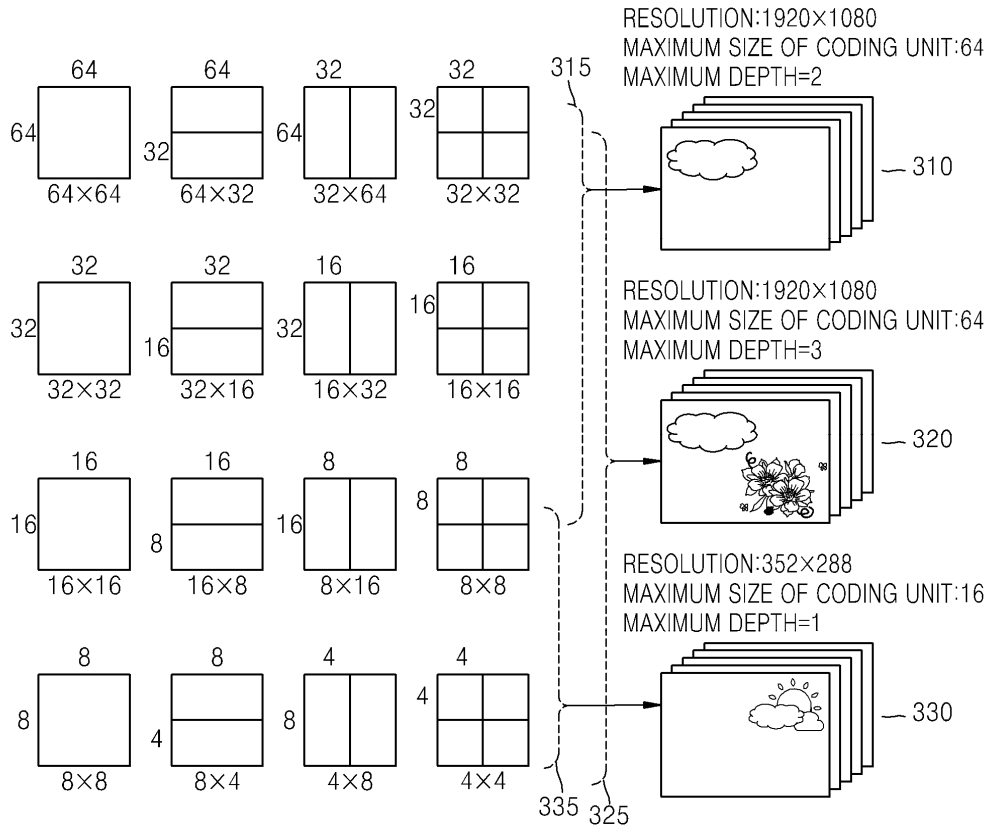


FIG. 11

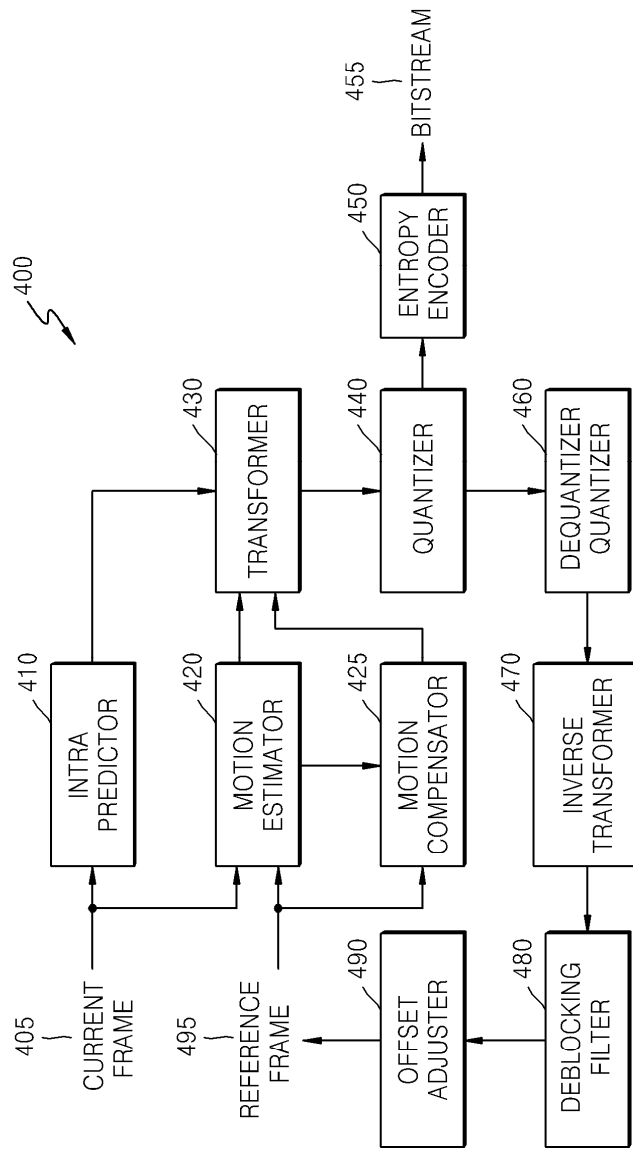


FIG. 12

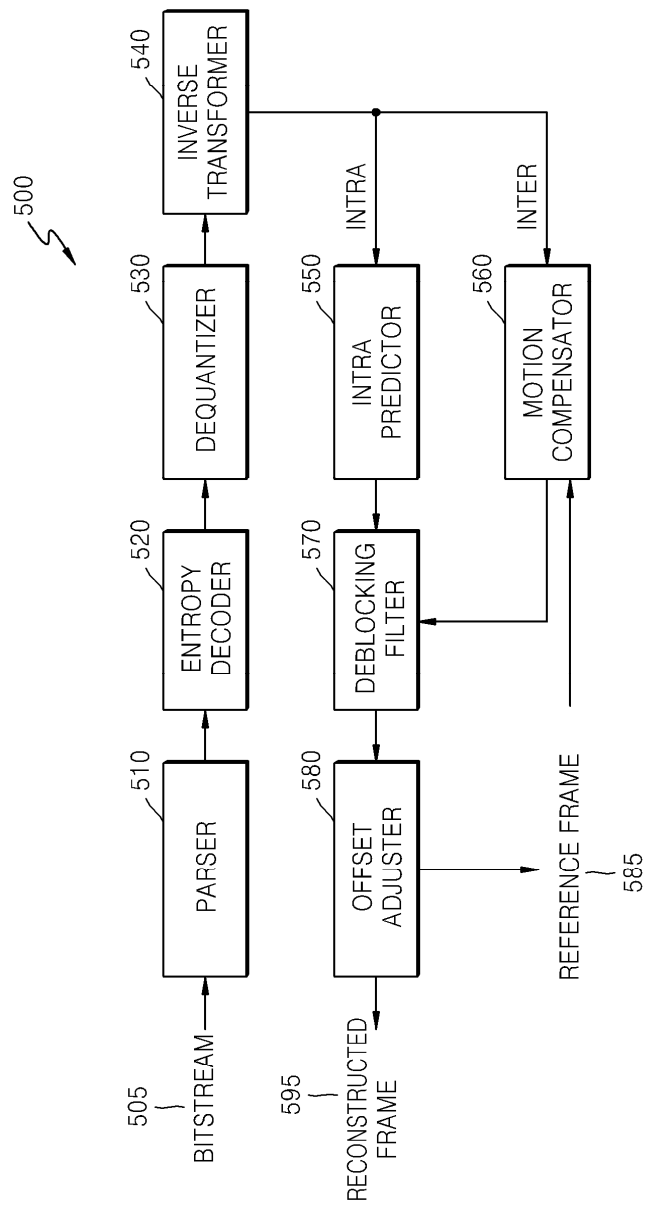


FIG. 13

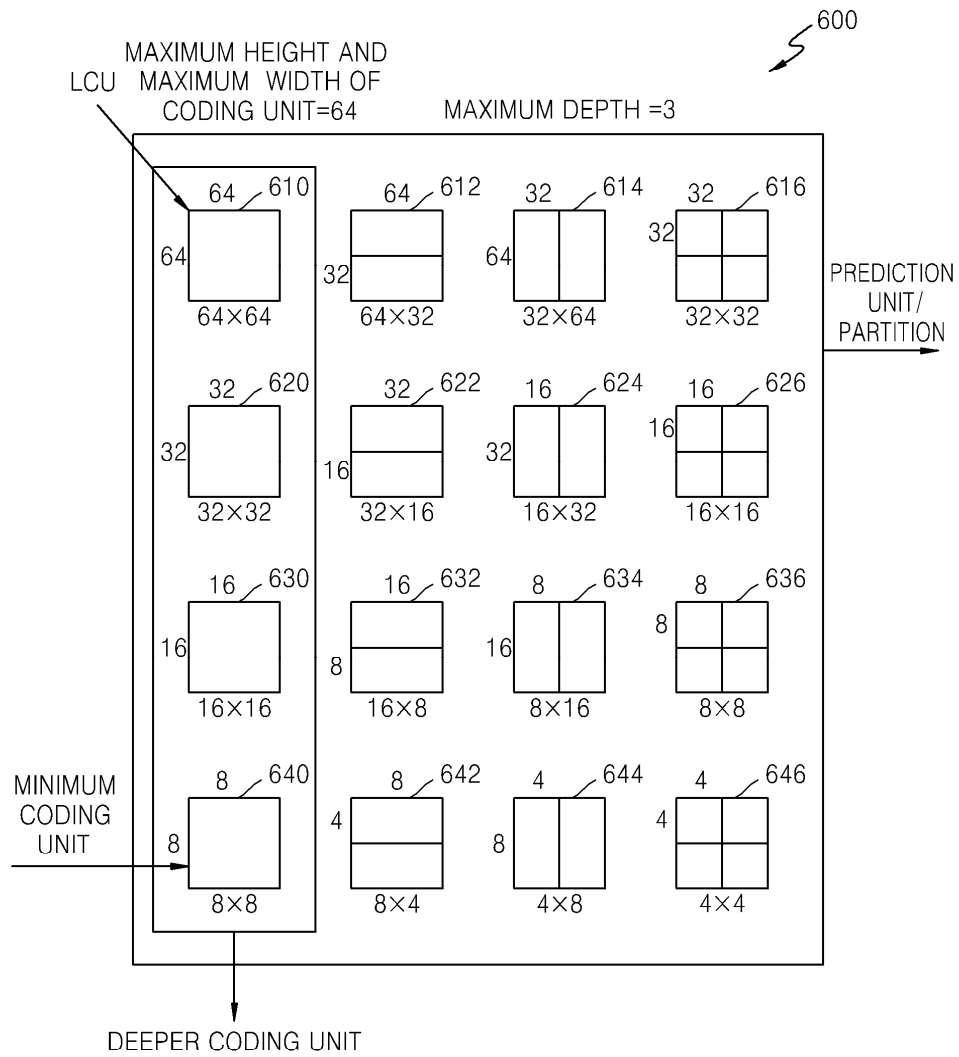


FIG. 14

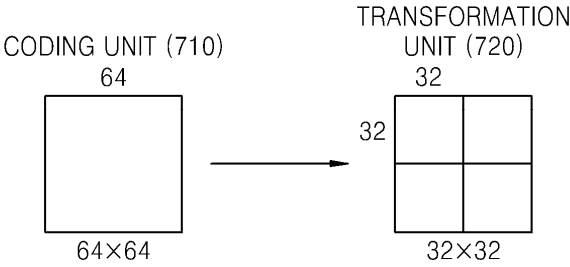


FIG. 15

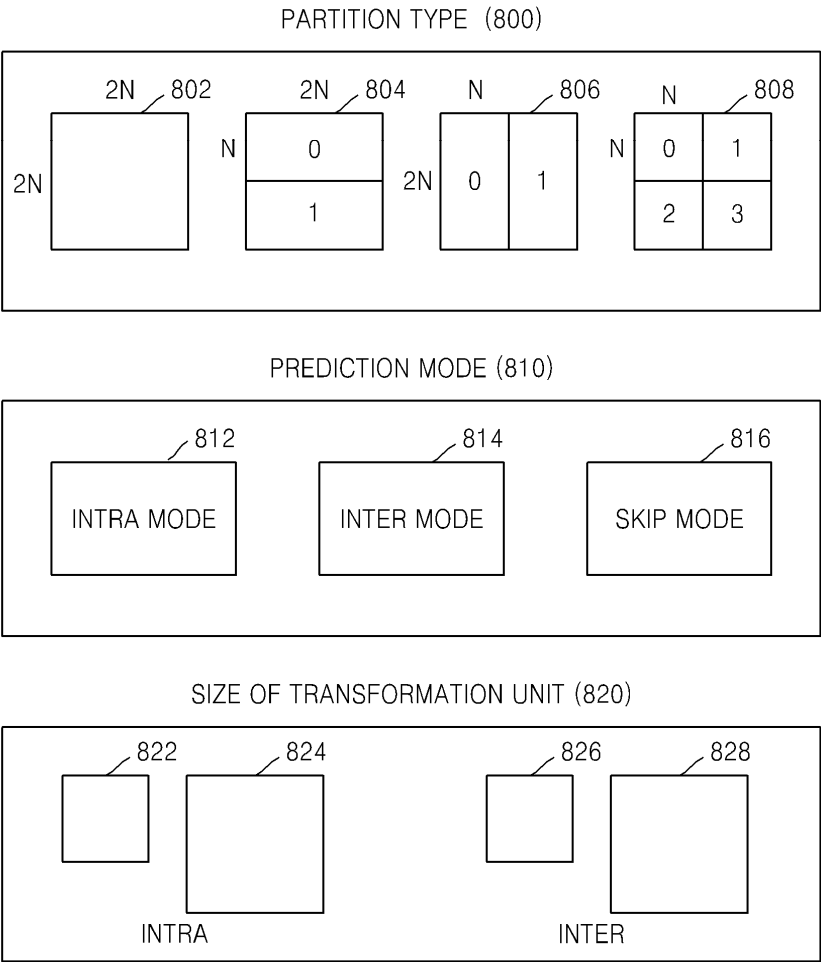
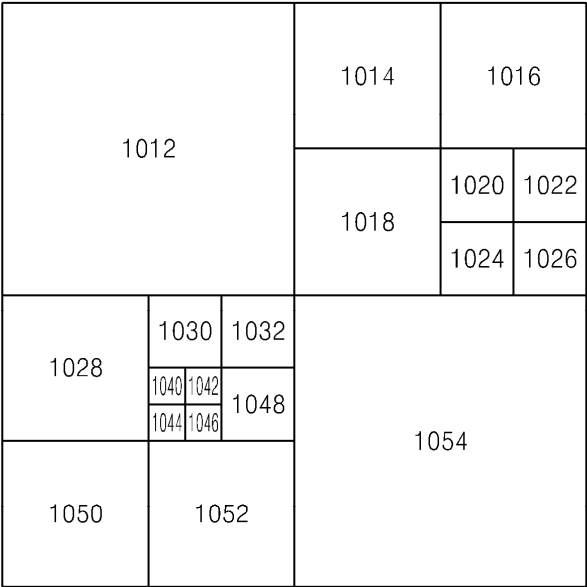


FIG. 17



CODING UNIT (1010)

FIG. 18

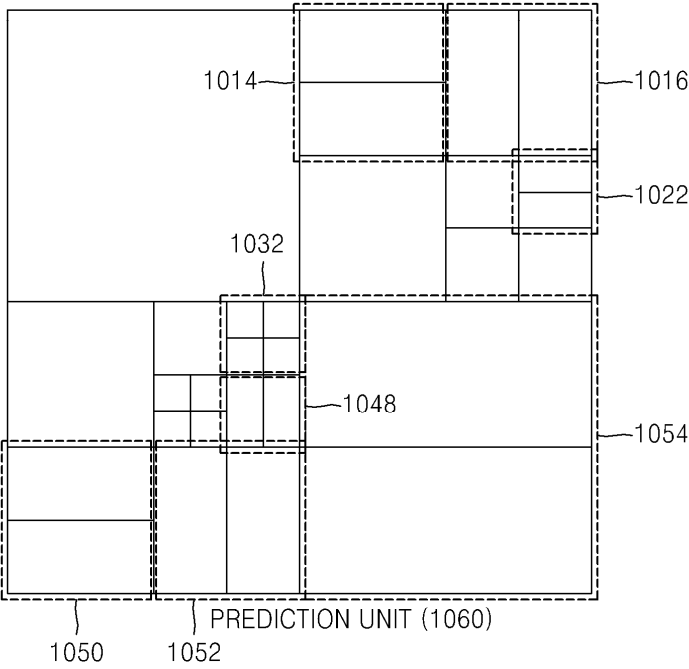


FIG. 19

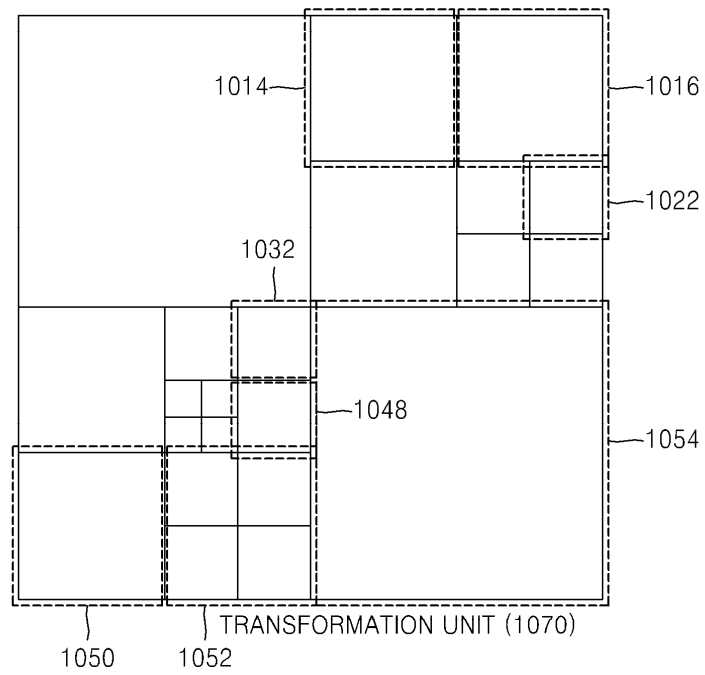


FIG. 20

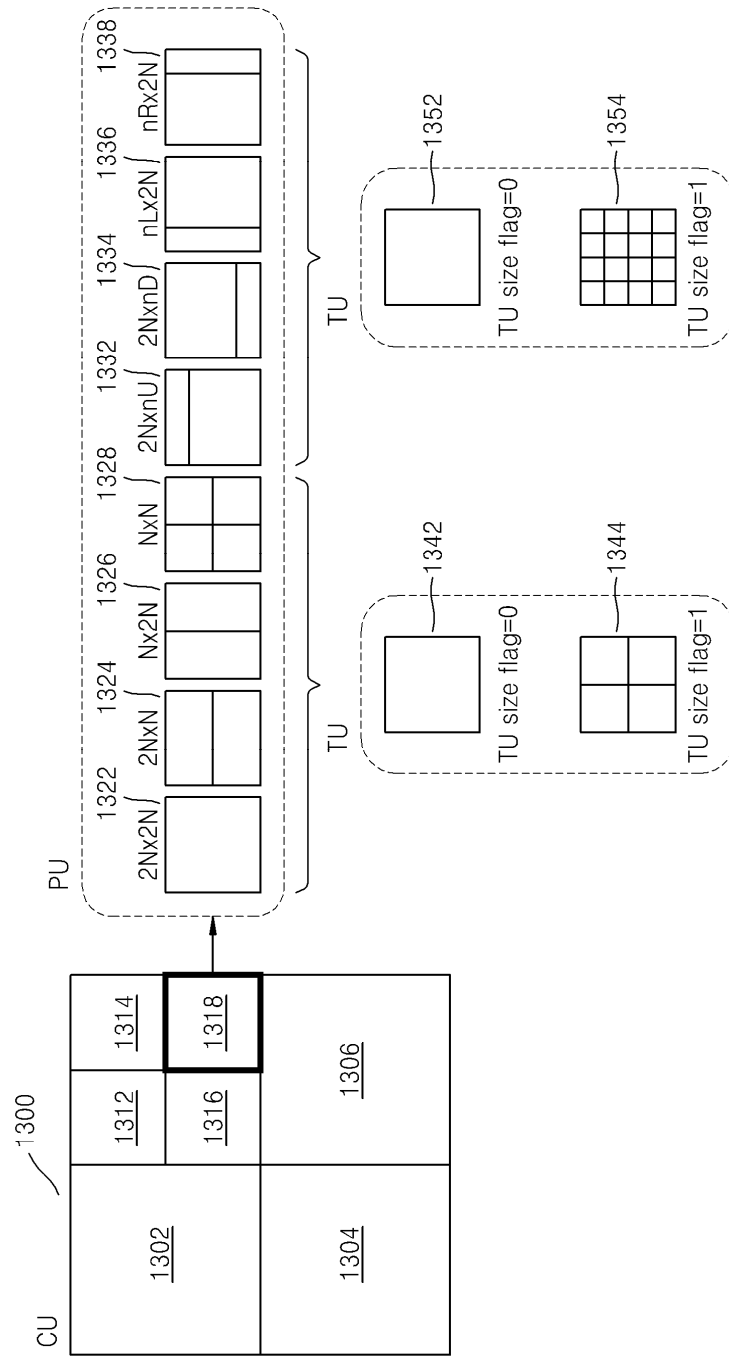


FIG. 21

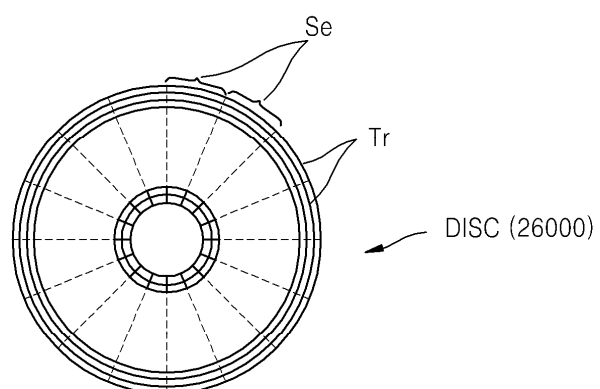
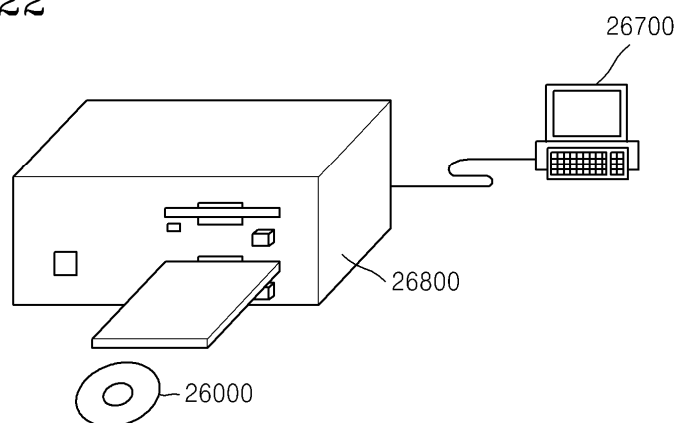


FIG. 22



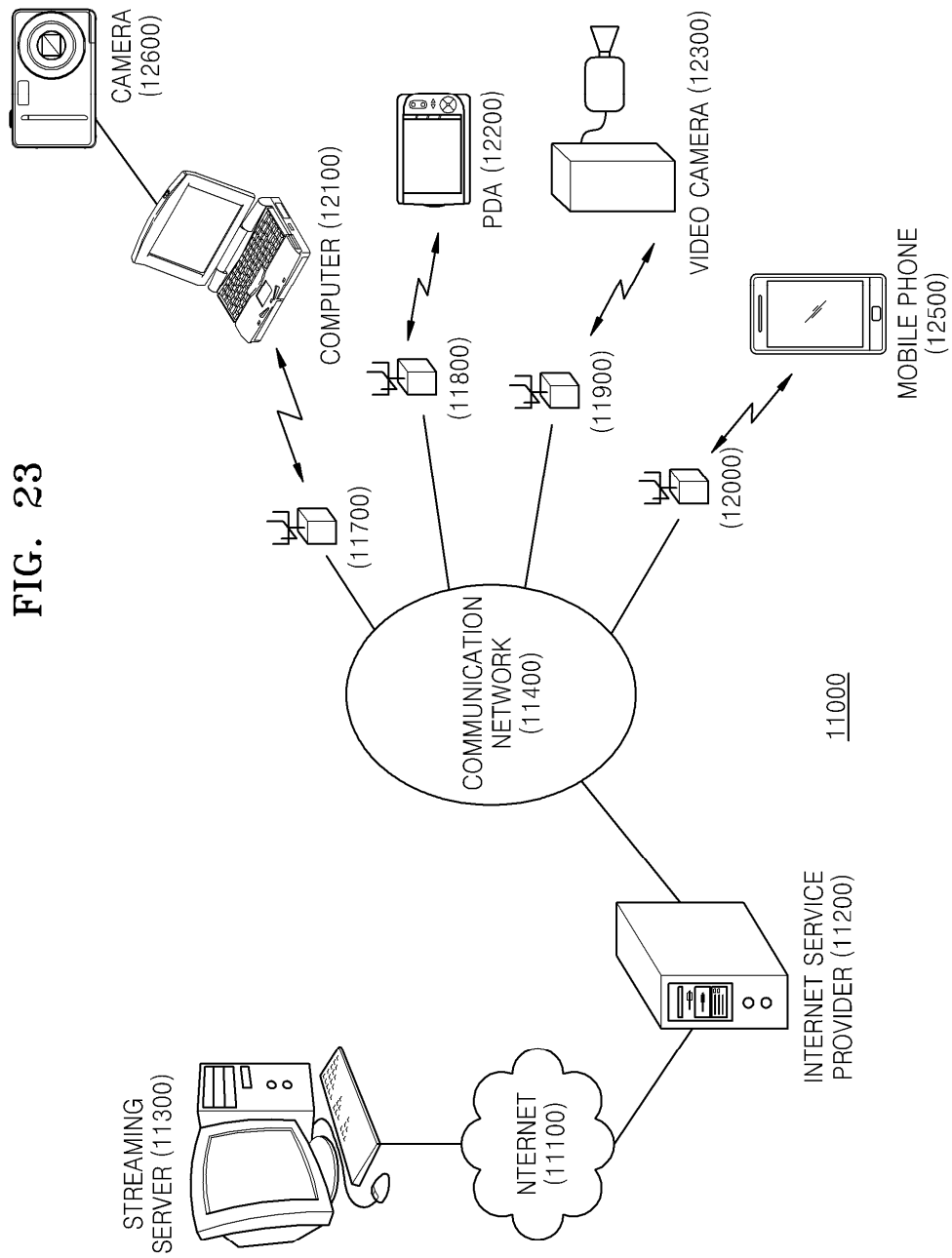


FIG. 23

FIG. 24

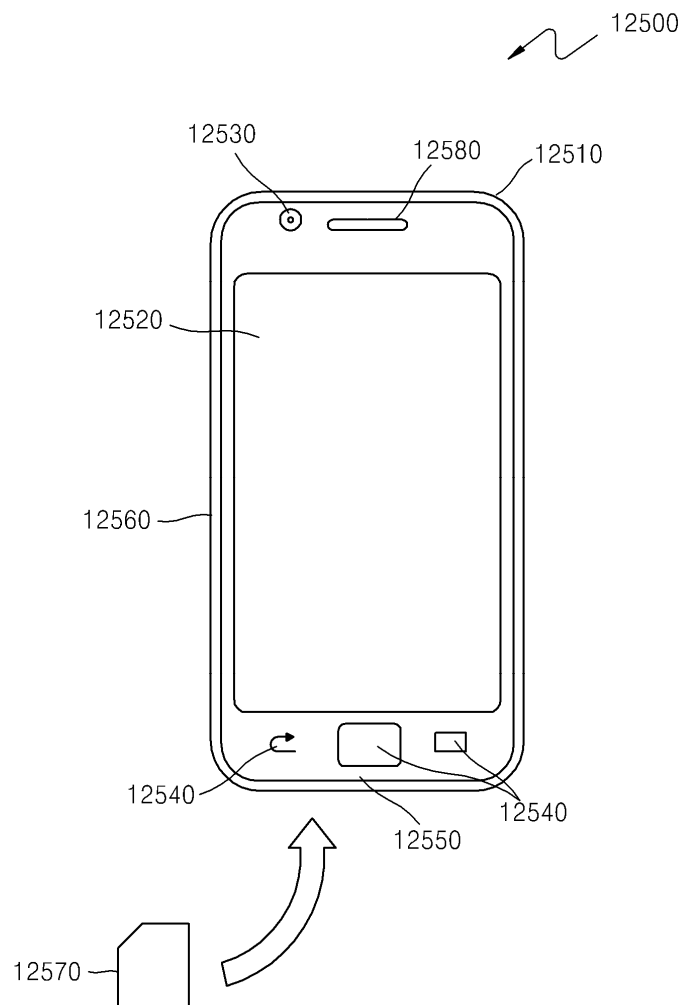


FIG. 25

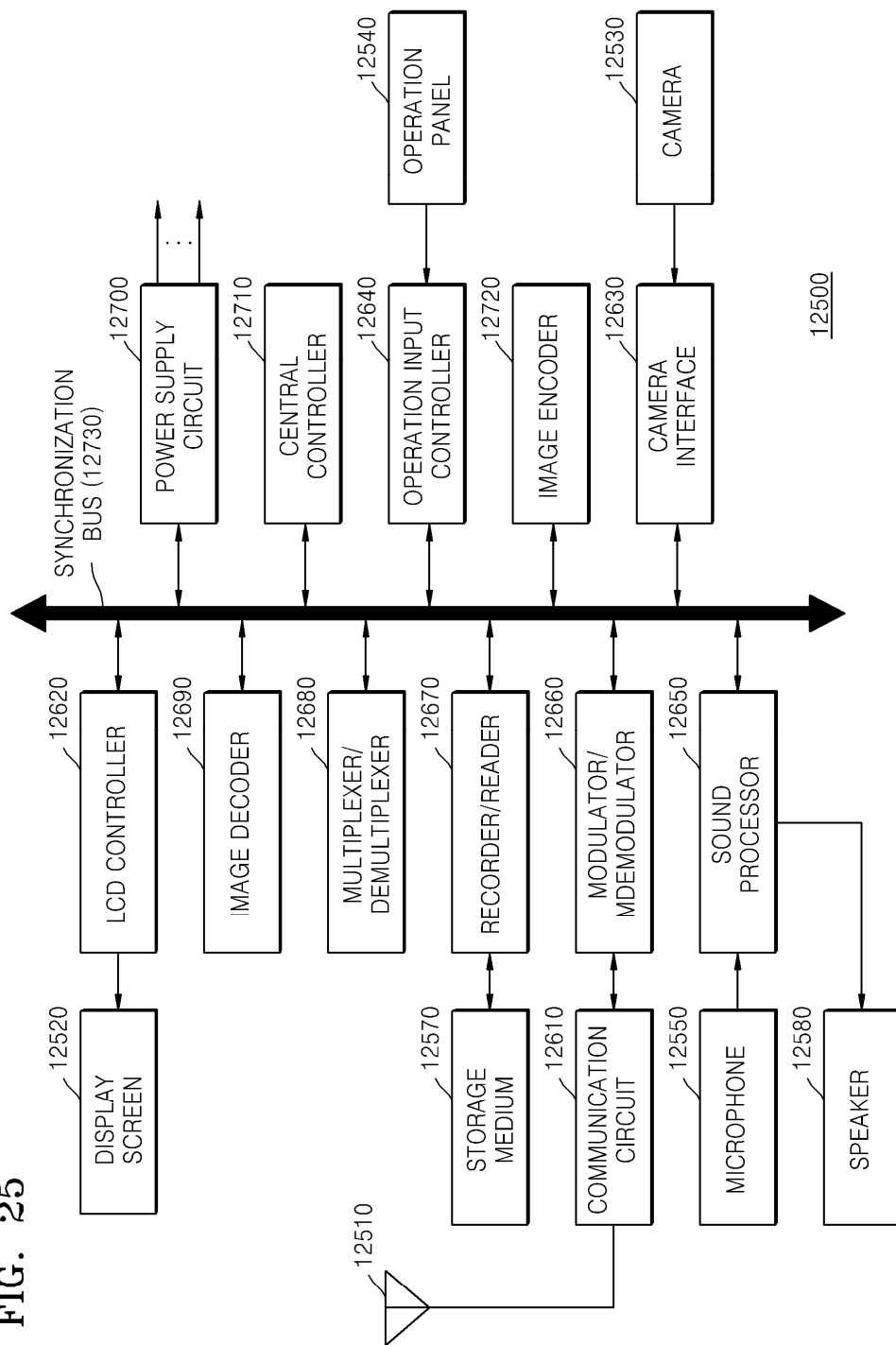


FIG. 26

