

FIG. 1A

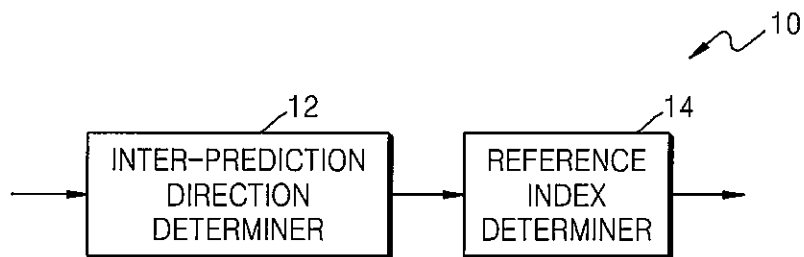


FIG. 1B

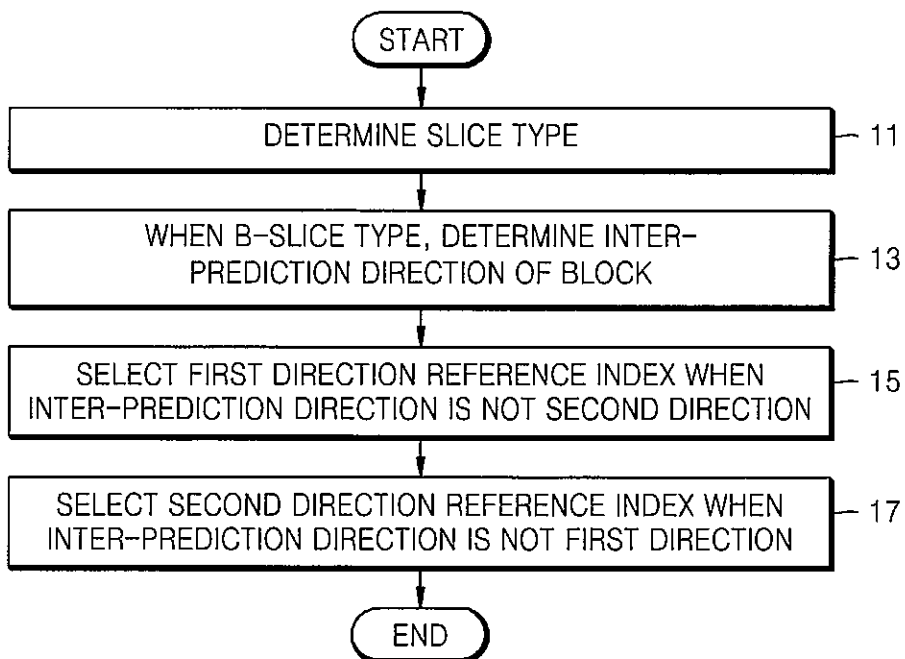


FIG. 2A

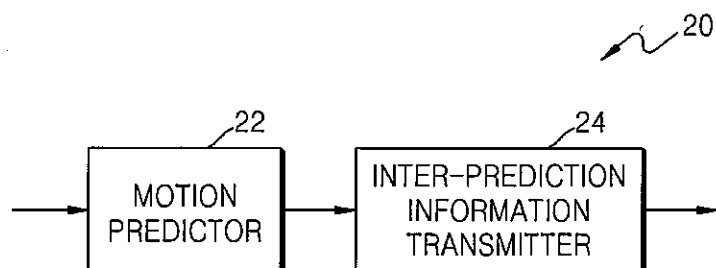


FIG. 2B

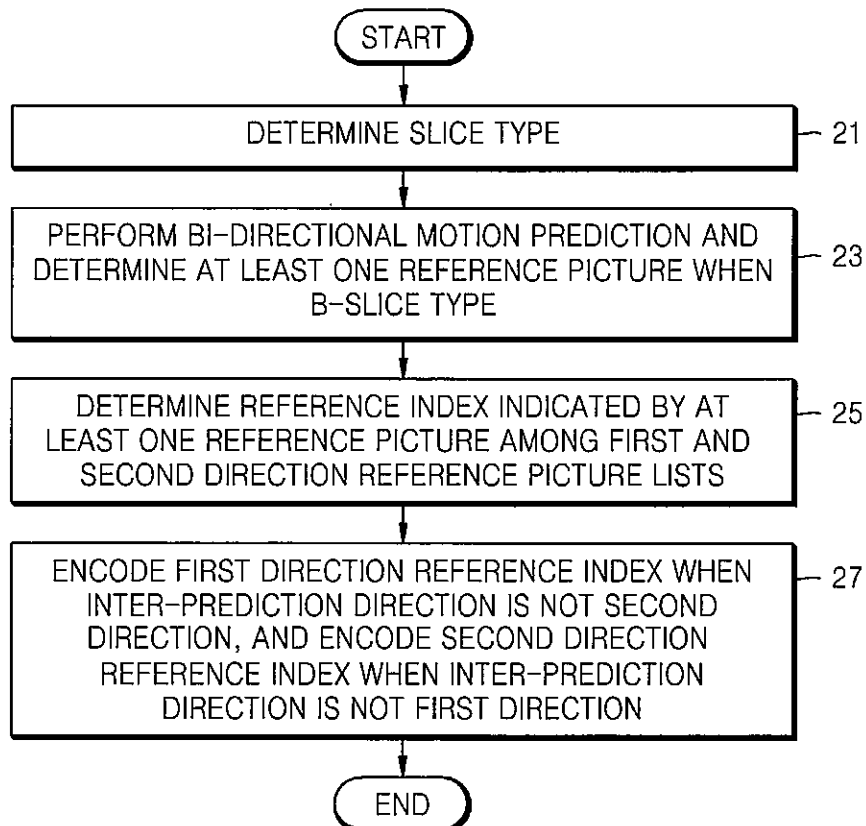


FIG. 3A

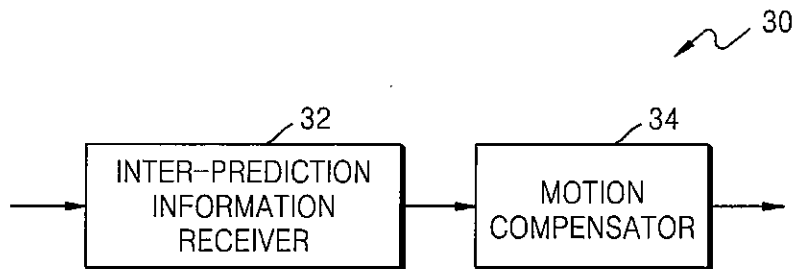


FIG. 3B

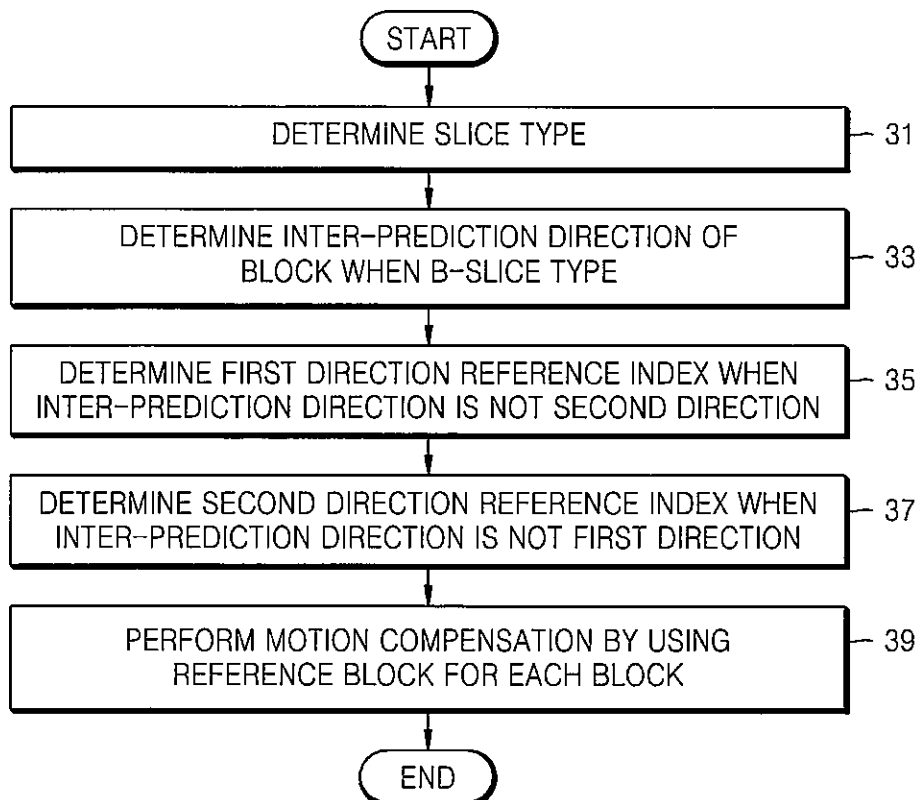


FIG. 4

inter_pred_flag	NAME OF inter_pred flag
INFERRED	Pred_L0
0	Pred_LC
1	Pred_BI



inter_pred_idc	NAME OF inter_pred_idc	
	(nPbW + nPbH) != 12	(nPbW + nPbH) == 12
0	PRED_L0	PRED_L0
1	PRED_L1	PRED_L1
2	PRED_B1	na

FIG. 5

50	slice_header() {
	...
52	if(lists_modification_present_flag) {
54	ref_pic_list_modification()
	ref_pic_list_combination()
	...

60

FIG. 6

60

```

ref_pic_list_combination() {
    if( slice_type == B ) {
        ref_pic_list_combination_flag
        if( ref_pic_list_combination_flag ) {
            num_ref_idx_l0_active_minus1
            ref_pic_list_modification_flag_l0
            if( ref_pic_list_modification_flag_l0 )
                for( i=0; i <= num_ref_idx_l0_active_minus1; i++ ) {
                    pic_from_list_0_flag
                    if( ( pic_from_list_0_flag &&
                        num_ref_idx_l0_active_minus1 > 0 ) ||
                        ( !pic_from_list_0_flag &&
                        num_ref_idx_l1_active_minus1 > 0 ) )
                        ref_idx_list_curr
                }
            }
        }
    }
}

```

FIG. 7

```

70      pred_weight_table( ) {
      ...
71      if( slice_type == B ) {
      if( ref_pic_list_combination_flag == 0 ) {
      for( i = 0; i <= num_ref_idx_l1_active_minus1; i++ ) {
      luma_weight_l1_flag
      if( luma_weight_l1_flag ) {
      delta_luma_weight_l1[ i ]
      luma_offset_l1[ i ]
73      }
      if( chroma_format_idc != 0 ) {
      chroma_weight_l1_flag
      if( chroma_weight_l1_flag )
      for( j = 0; j < 2; j++ ) {
      delta_chroma_weight_l1[ i ][ j ]
      delta_chroma_offset_l1[ i ][ j ]
      }
      }
      }
      } else { 
      for( i = 0; i <= num_ref_idx_lc_active_minus1; i++ ) {
      luma_weight_lc_flag
      if( luma_weight_l1_flag ) {
      delta_luma_weight_lc[ i ]
      luma_offset_lc[ i ]
75      }
      if( chroma_format_idc != 0 ) {
      chroma_weight_lc_flag
      if( chroma_weight_lc_flag )
      for( j = 0; j < 2; j++ ) {
      delta_chroma_weight_lc[ i ][ j ]
      delta_chroma_offset_lc[ i ][ j ]
      }
      }
      }
      }
      }
      }
      }
  
```

FIG. 8

```

80      prediction_unit( x0, y0, log2CbSize ) {
      ...
      } else {
81          if( slice_type == B )
82              inter_pred_flag[ x0 ][ y0 ]
83              {
                  if( inter_pred_flag[ x0 ][ y0 ] == Pred_LC ) {
                      if( num_ref_idx_lc_active_minus1 > 0 )
                      ref_idx_lc[ x0 ][ y0 ]
                      mvd_coding( mvd_lc[ x0 ][ y0 ][ 0 ],
                                mvd_lc[ x0 ][ y0 ][ 1 ] )
                     .mvp_lc_flag[ x0 ][ y0 ]
                  } else { /* Pred_L0 or Pred_BI */
84                      if( inter_pred_flag[ x0 ][ y0 ] != Pred_L1 ) {
85                          if( num_ref_idx_l0_active_minus1 > 0 )
86                          {
                              ref_idx_l0[ x0 ][ y0 ]
                              mvd_coding( mvd_l0[ x0 ][ y0 ][ 0 ],
                                          mvd_l0[ x0 ][ y0 ][ 1 ] )
                             .mvp_l0_flag[ x0 ][ y0 ]
                          }
87                          if( inter_pred_flag[ x0 ][ y0 ] == Pred_BI ) {
88                          if( inter_pred_flag[ x0 ][ y0 ] != Pred_L0 ) {
89                          if( num_ref_idx_l1_active_minus1 > 0 )
90                          {
                              ref_idx_l1[ x0 ][ y0 ]
                              if( mvd_l1_zero_flag ) {
                                  mvd_l1[ x0 ][ y0 ][ 0 ] = 0
                                  mvd_l1[ x0 ][ y0 ][ 1 ] = 0
                              } else
                                  mvd_coding( mvd_l1[ x0 ][ y0 ][ 0 ],
                                              mvd_l1[ x0 ][ y0 ][ 1 ] )
                             .mvp_l1_flag[ x0 ][ y0 ]
                          }
                      }
                  }
              }
      }
  
```

FIG. 9

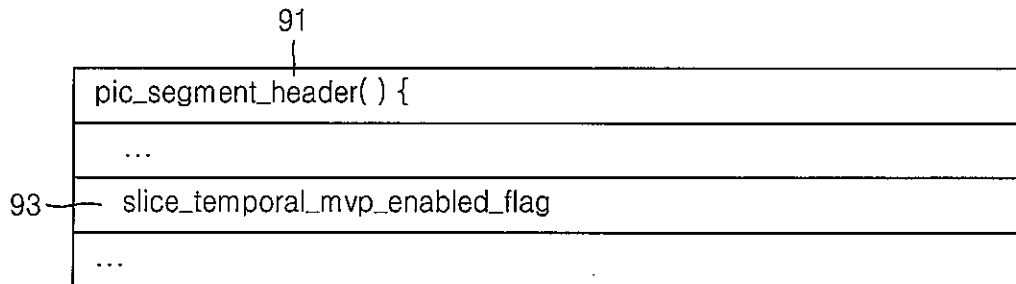


FIG. 10

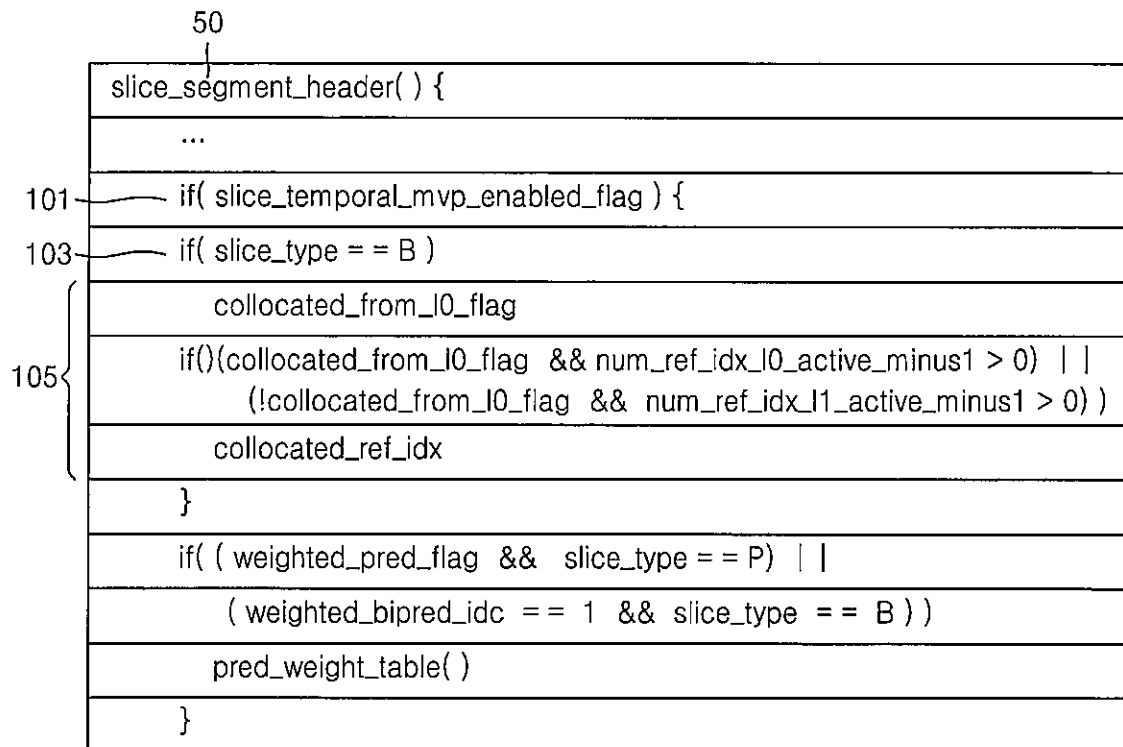


FIG. 11

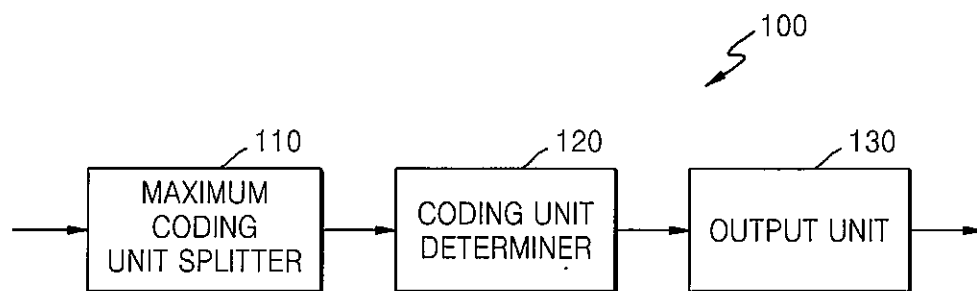


FIG. 12

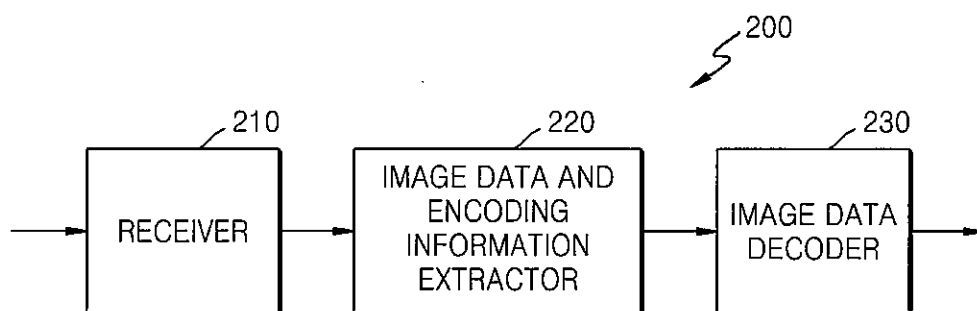


FIG. 13

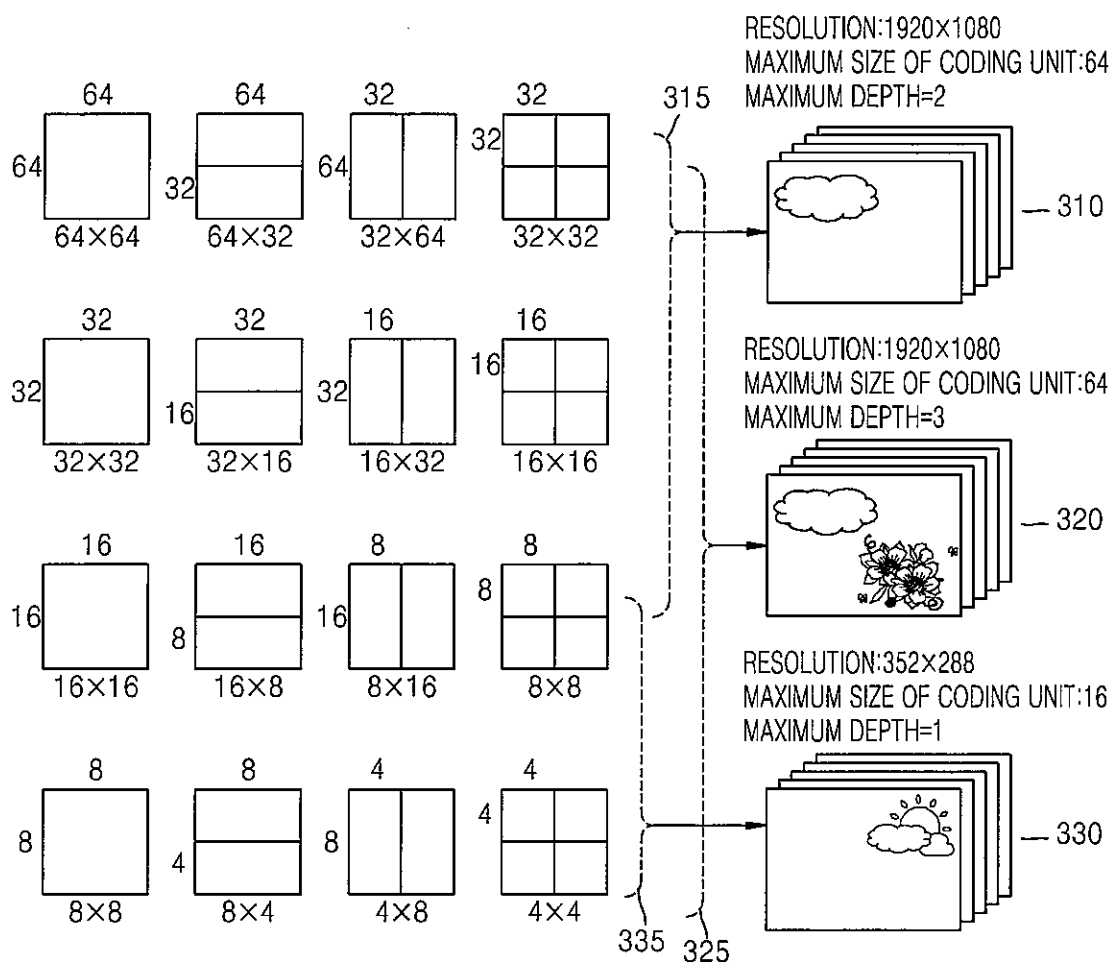


FIG. 14

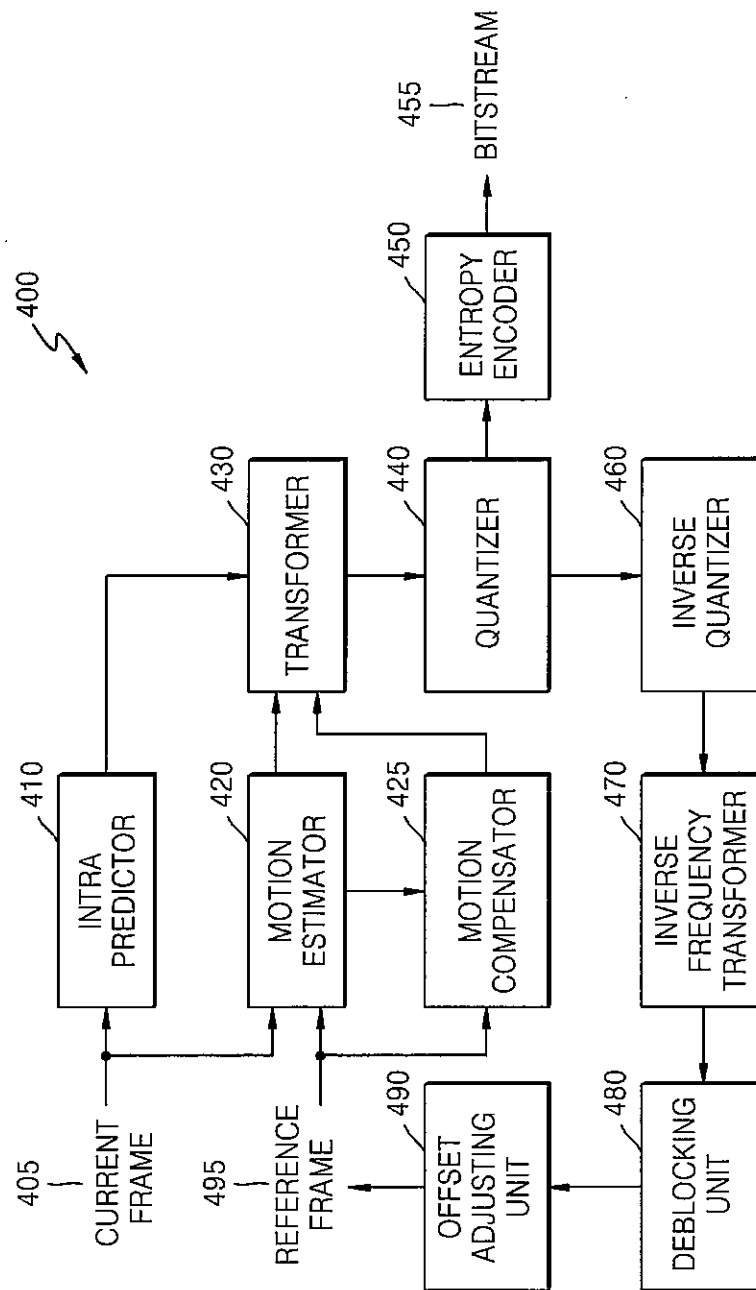


FIG. 15

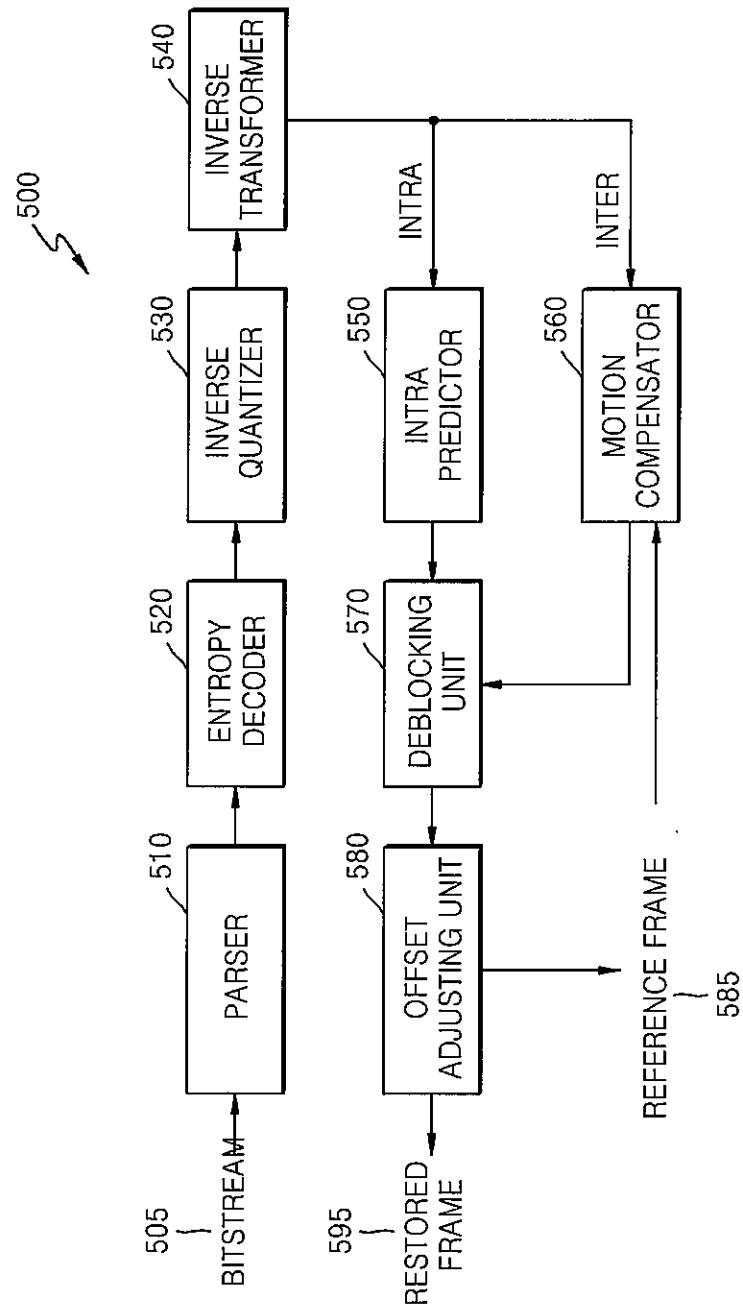


FIG. 16

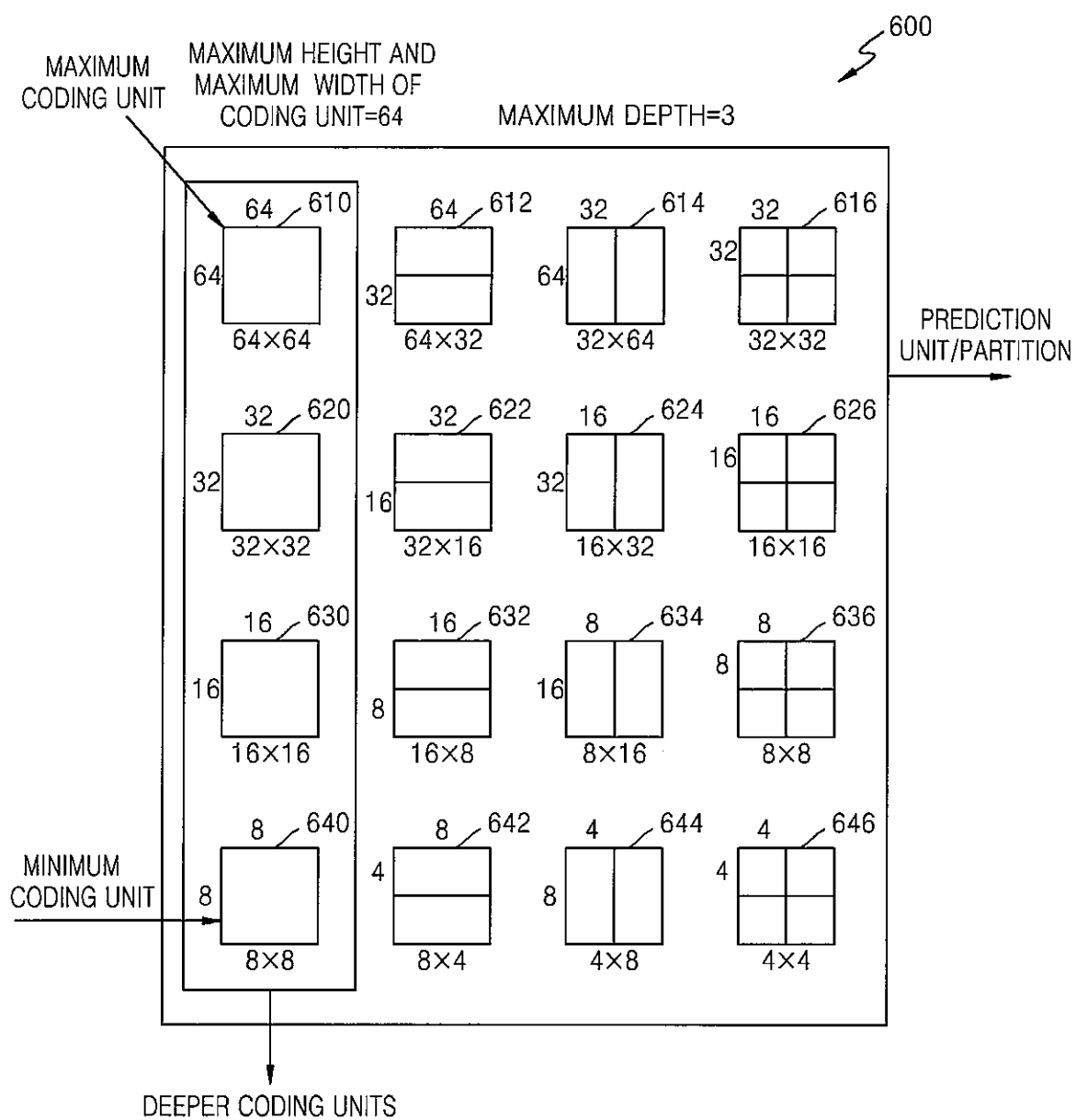


FIG. 17

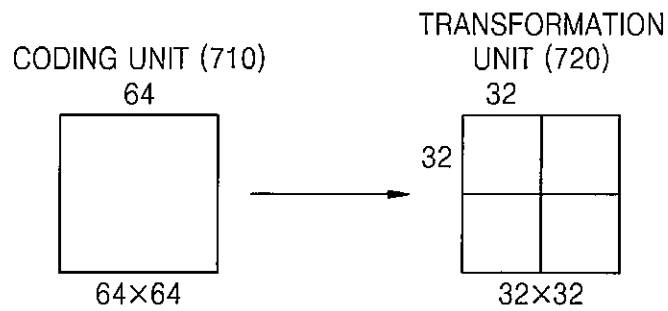
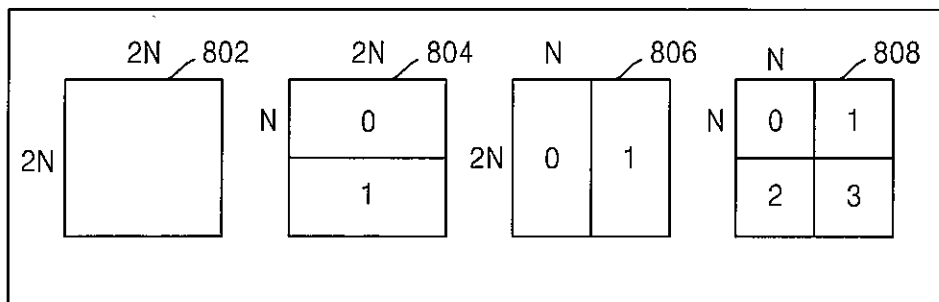
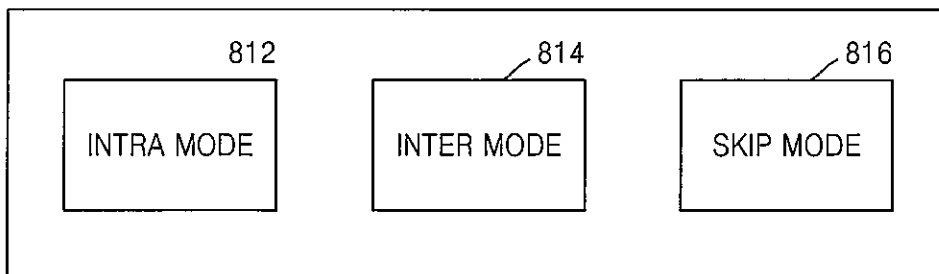


FIG. 18

PARTITION TYPE (800)



PREDICTION MODE (810)



SIZE OF TRANSFORMATION UNIT (820)

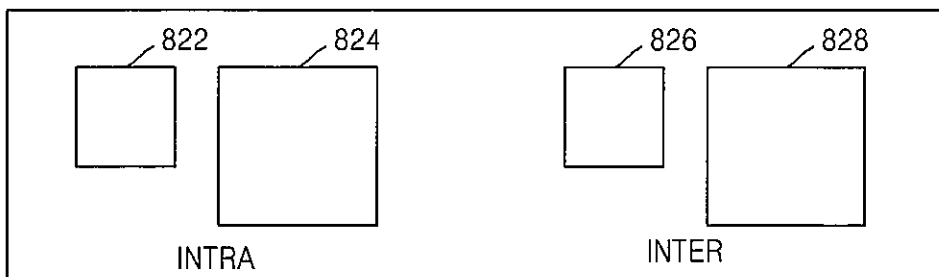


FIG. 19

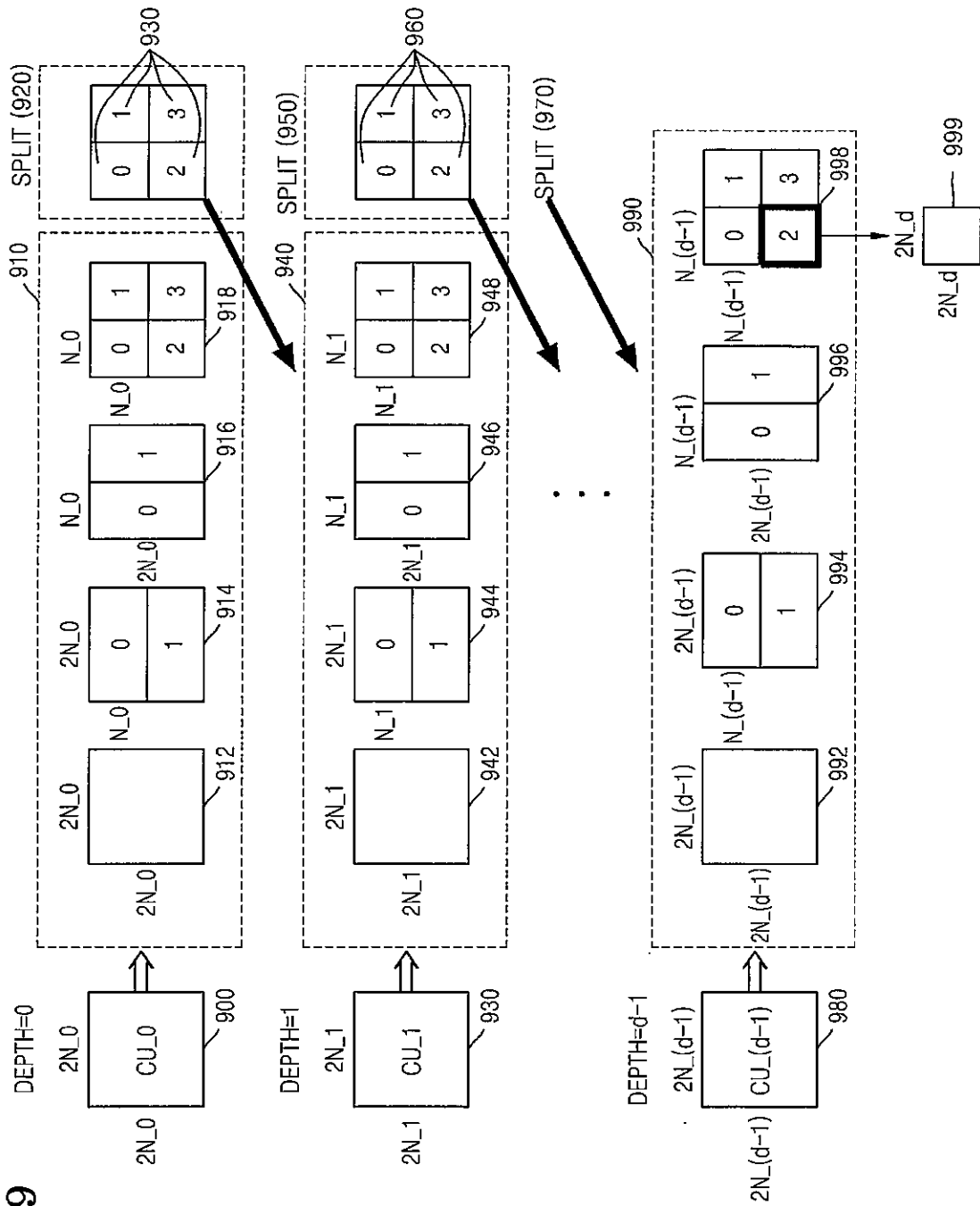
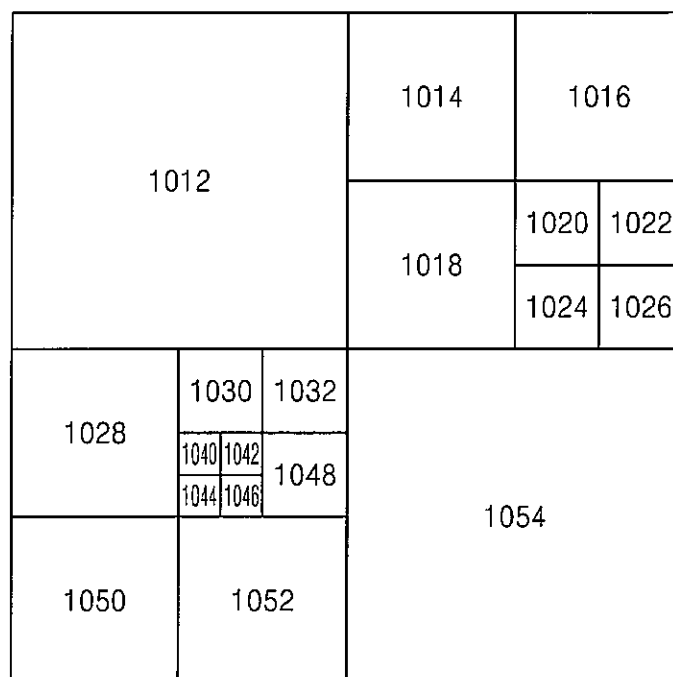
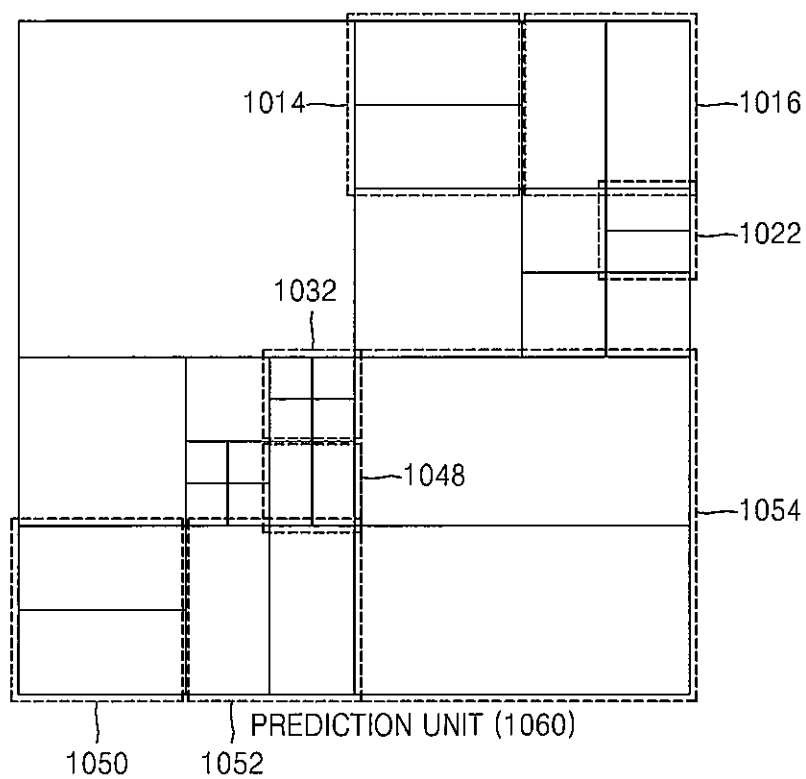


FIG. 20



CODING UNIT (1010)

FIG. 21



PREDICTION UNIT (1060)

FIG. 22

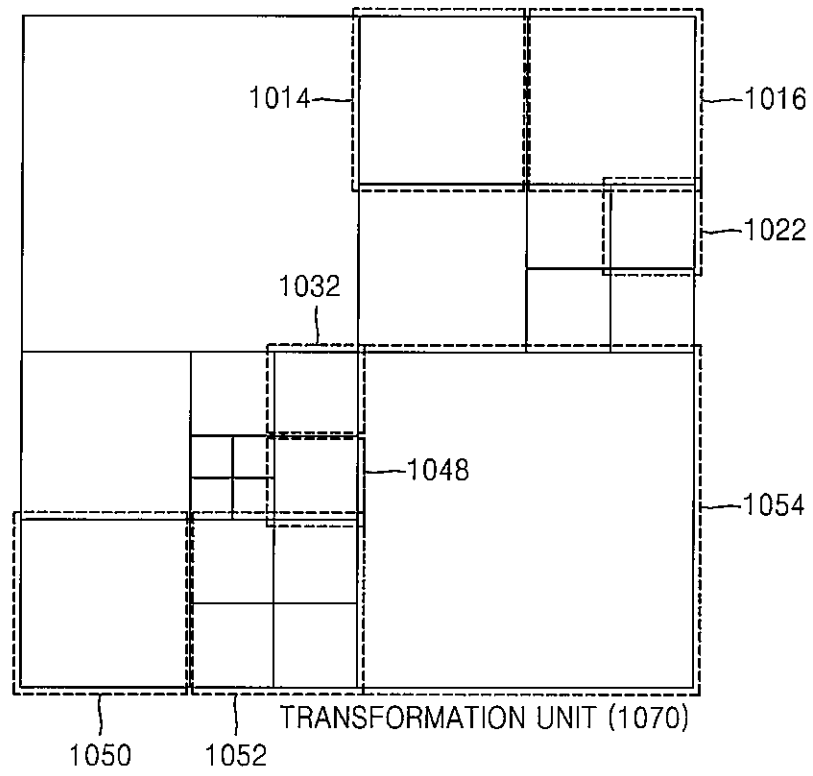


FIG. 23

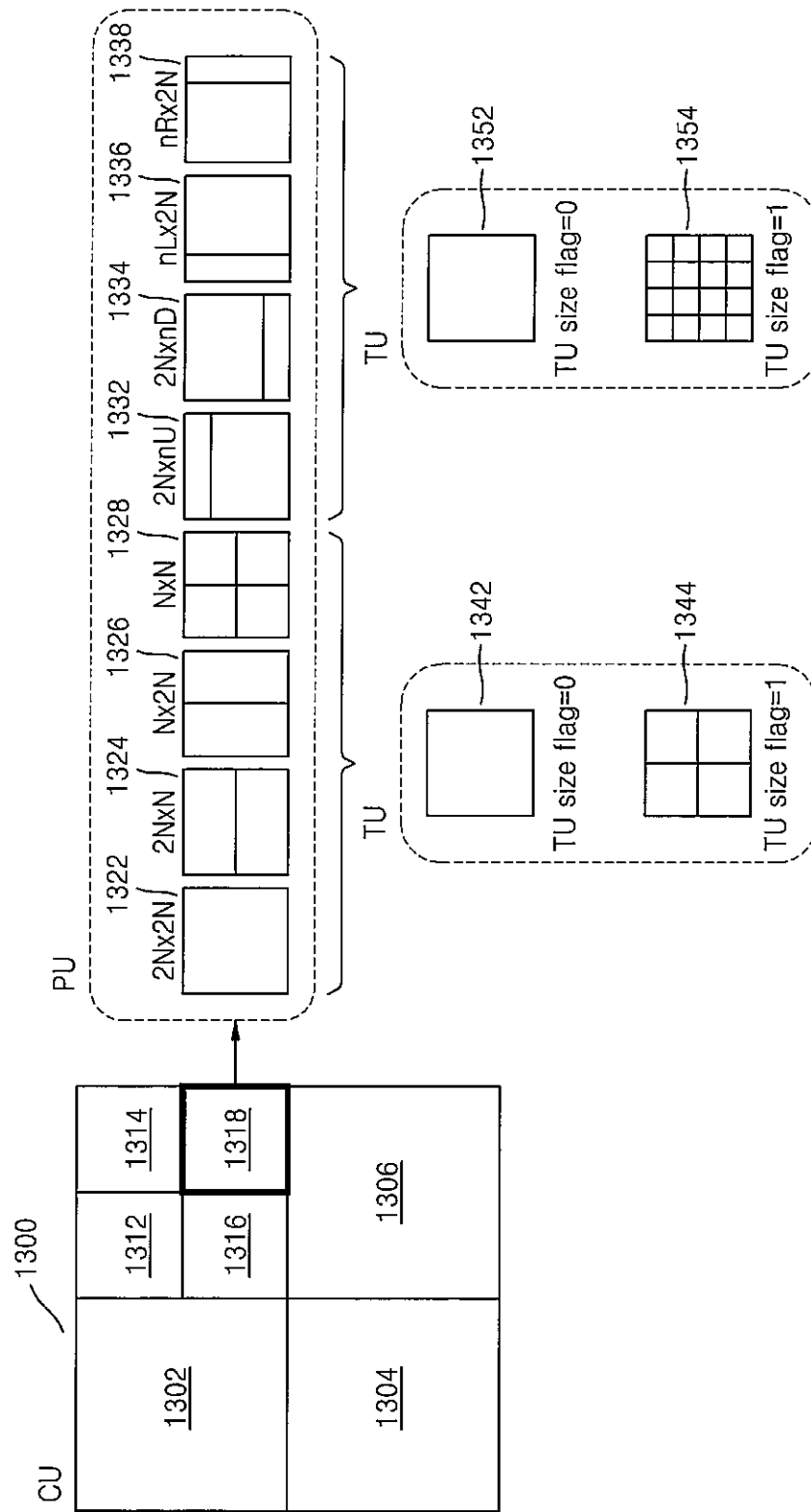


FIG. 24

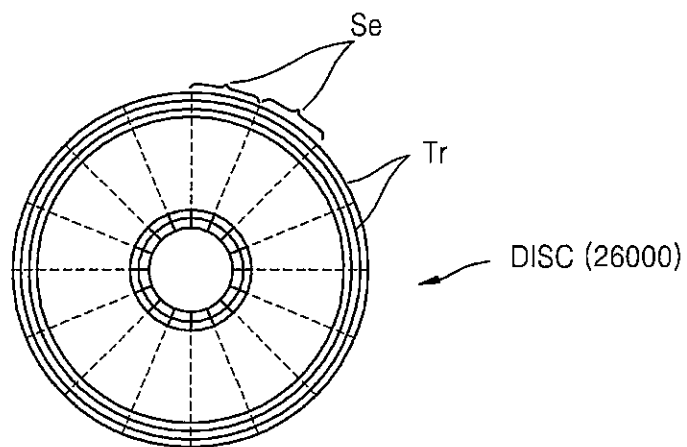


FIG. 25

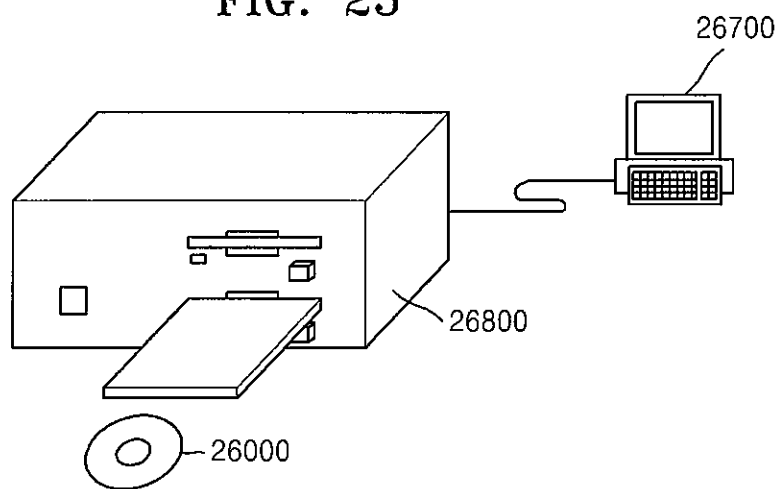


FIG. 26

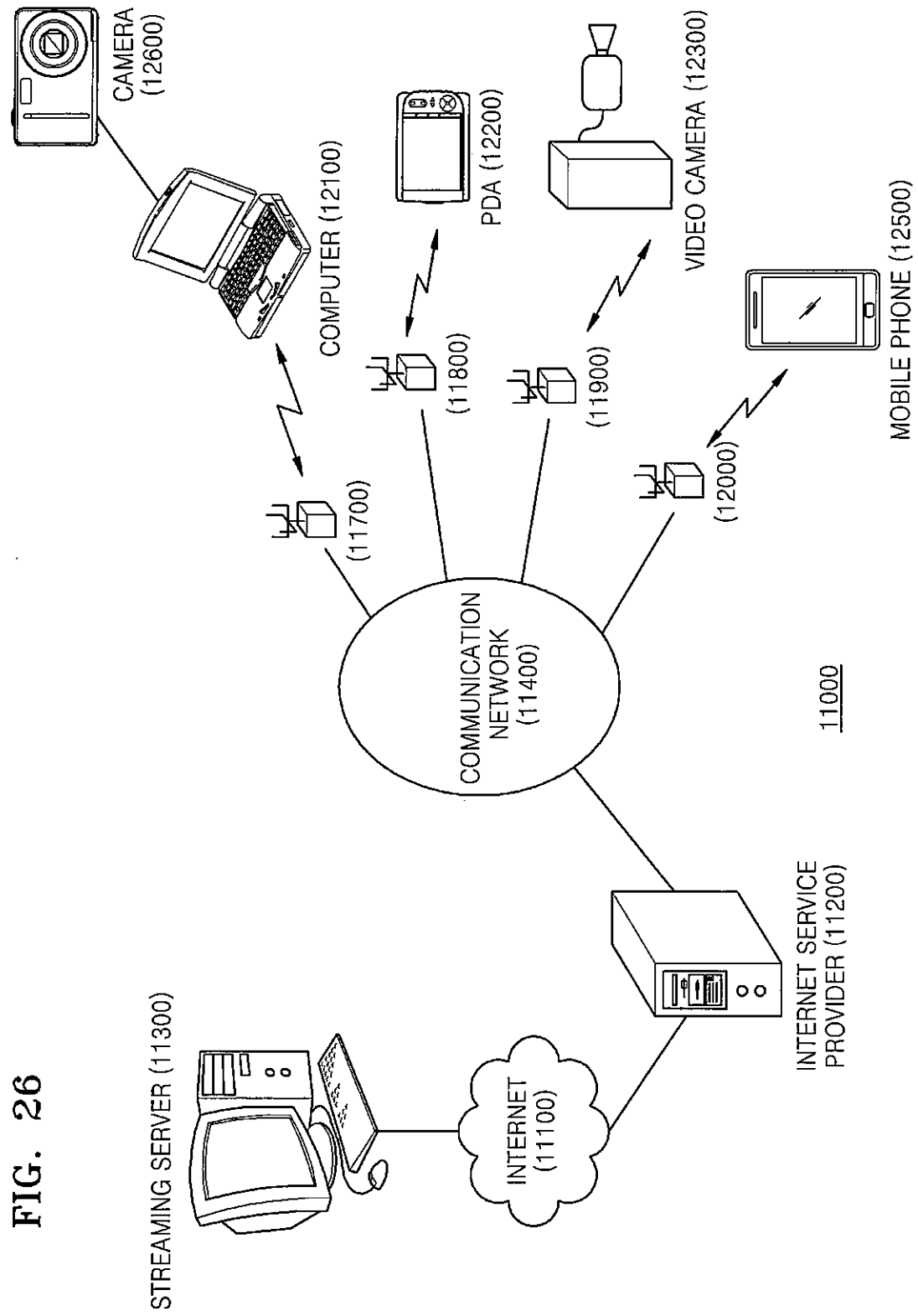


FIG. 27

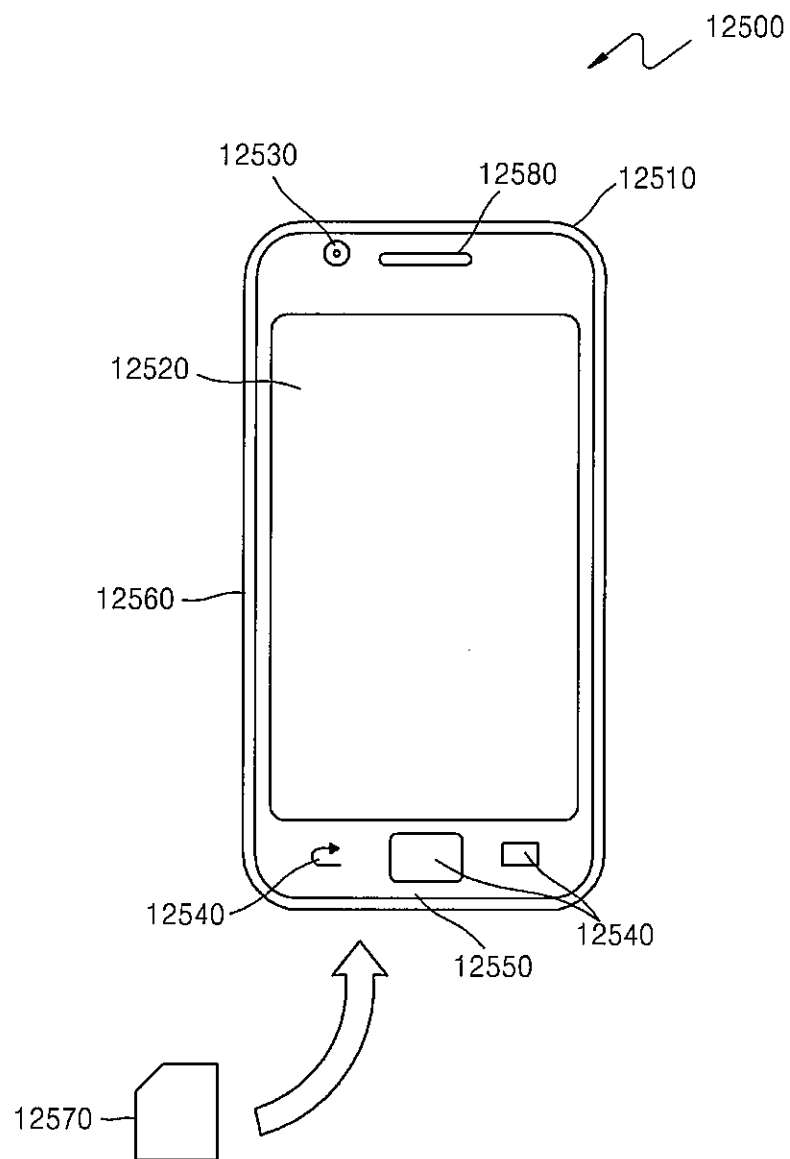


FIG. 28

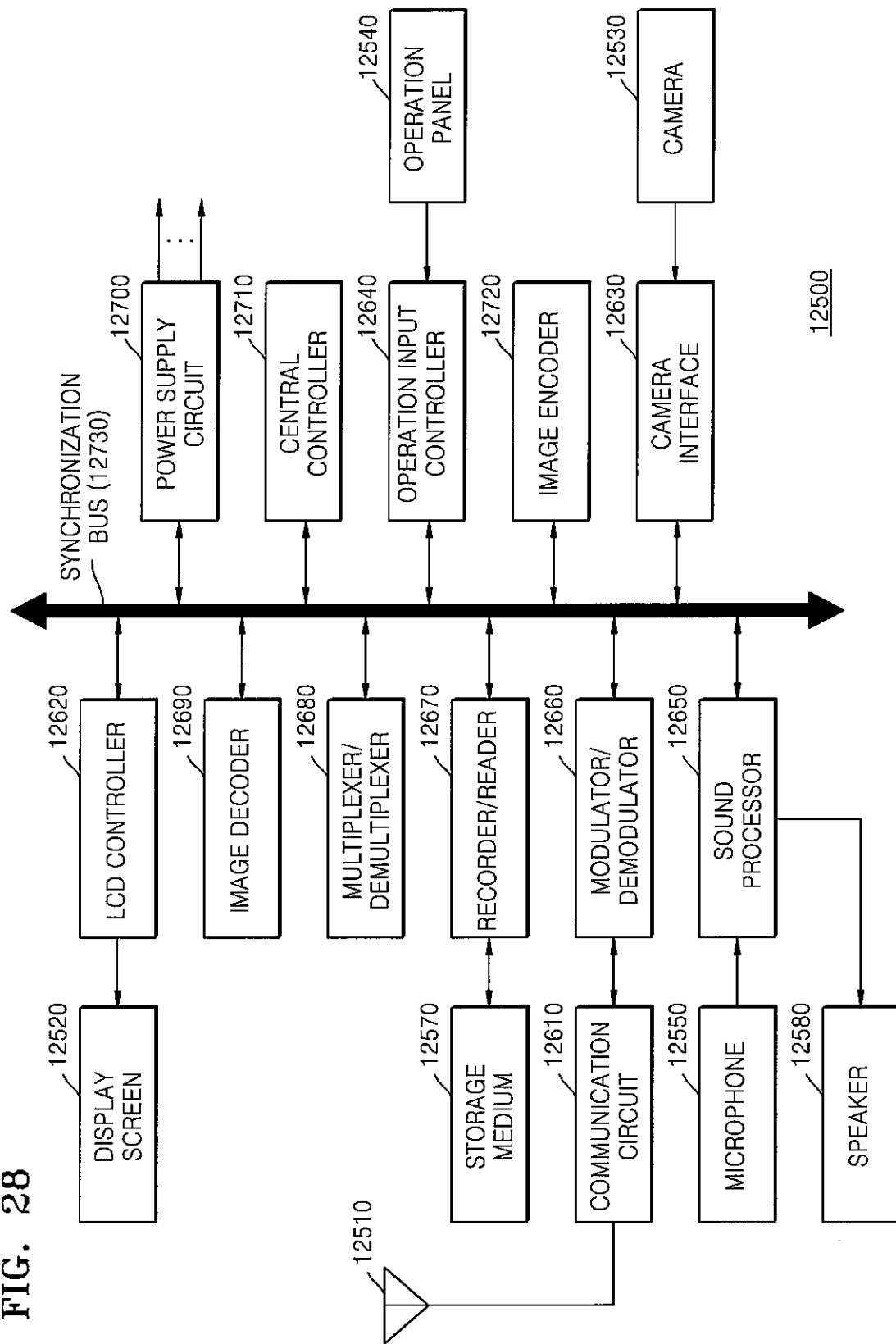


FIG. 29

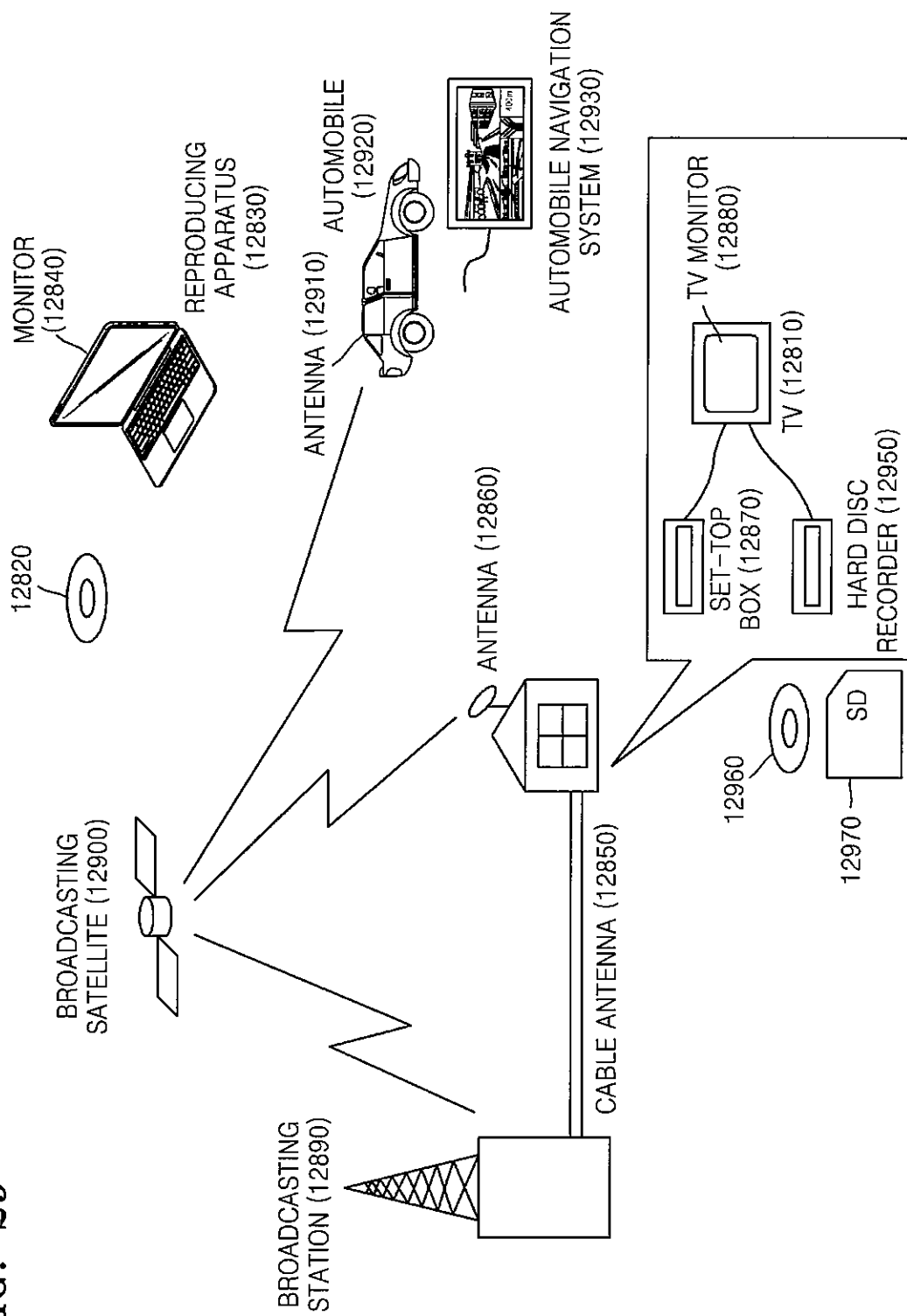


FIG. 30

