

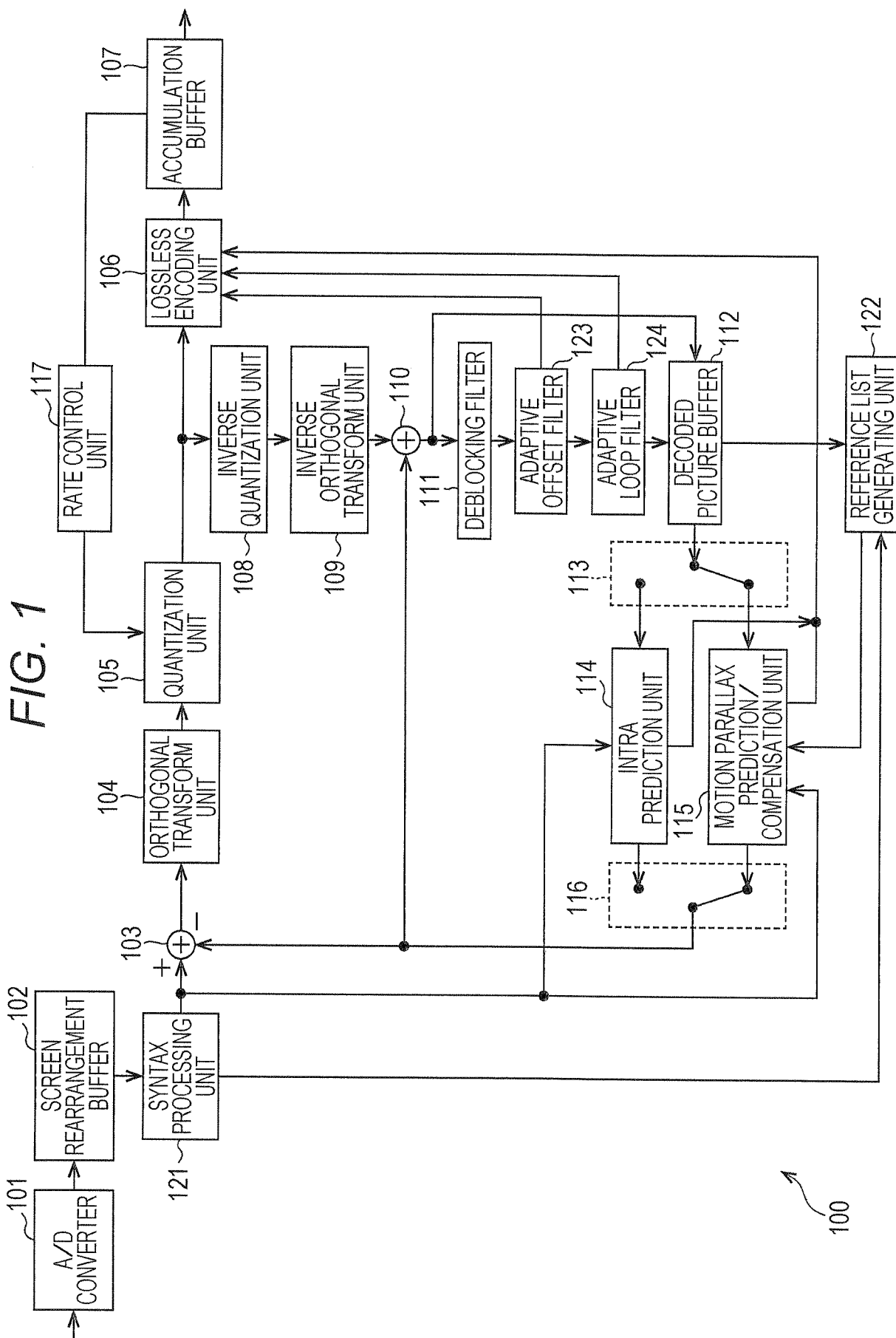


FIG. 2

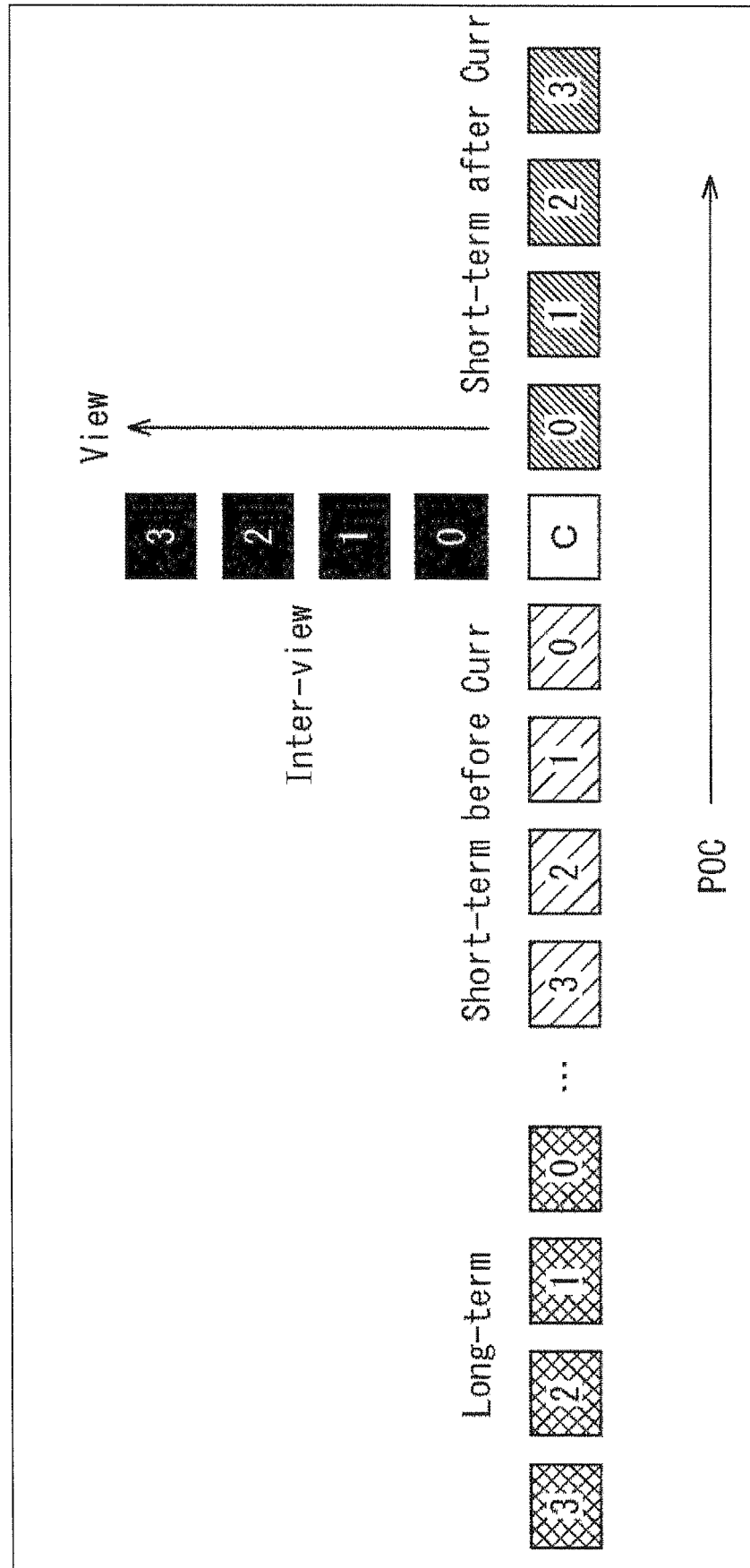


FIG. 3

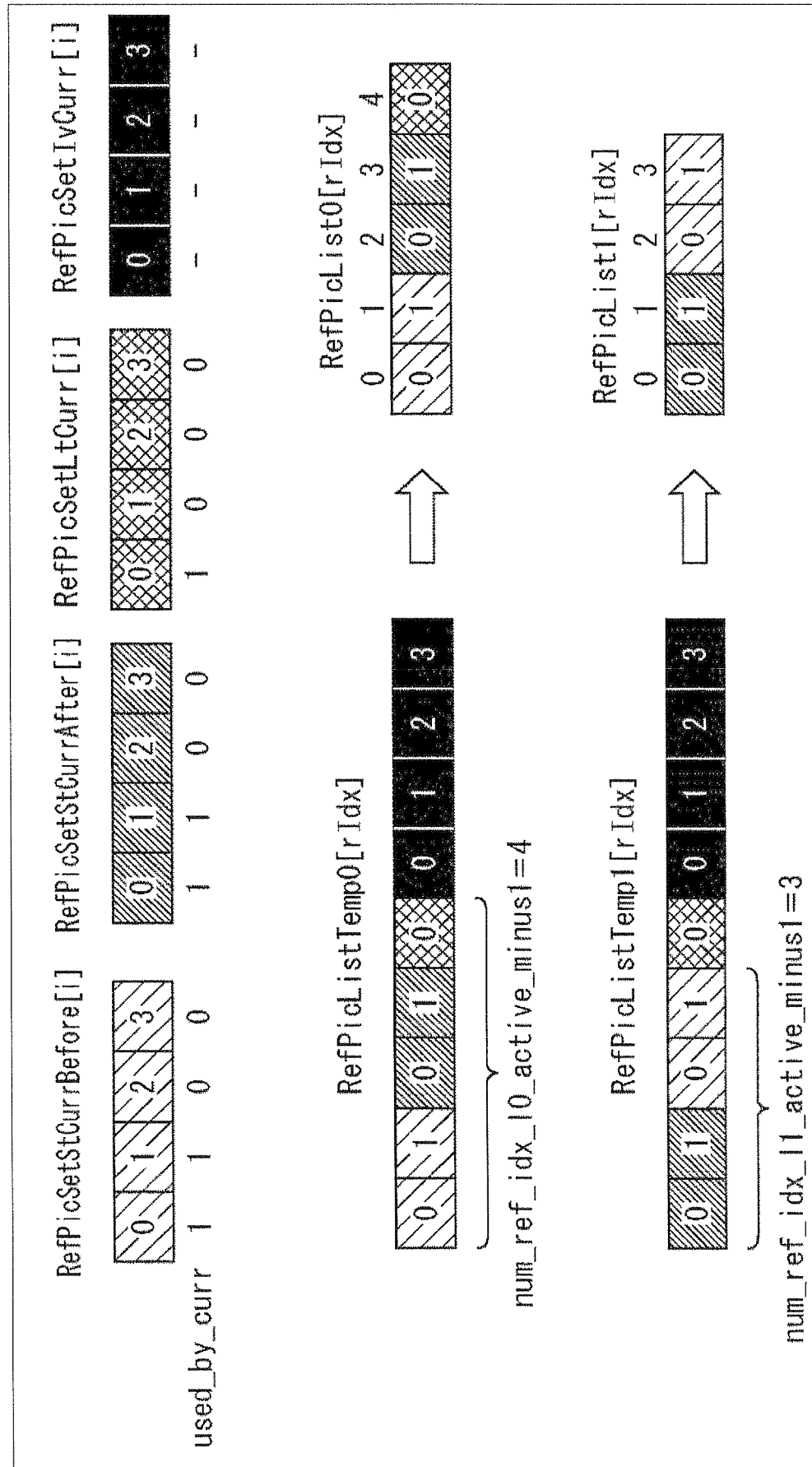




FIG. 5

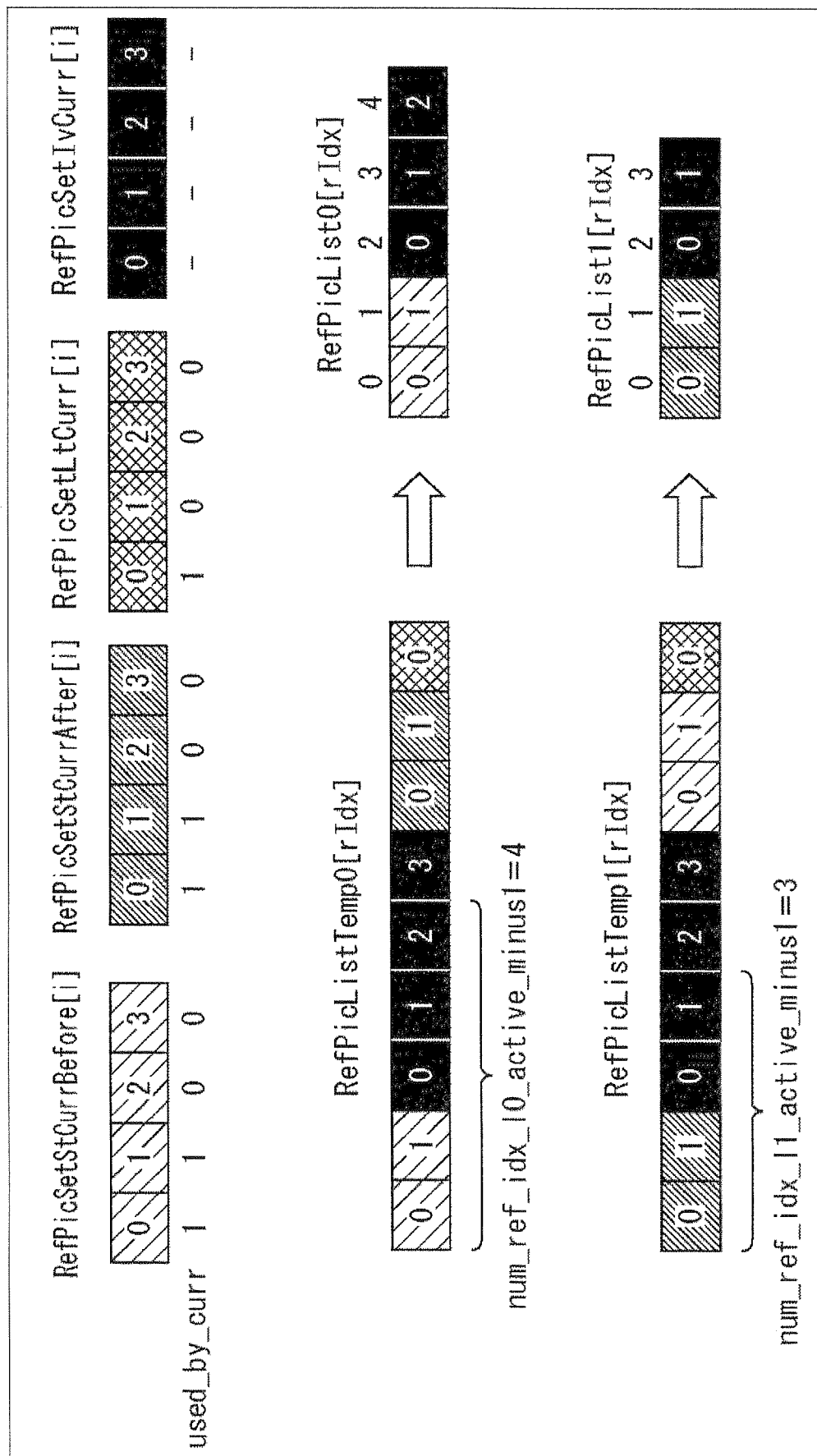


FIG. 6

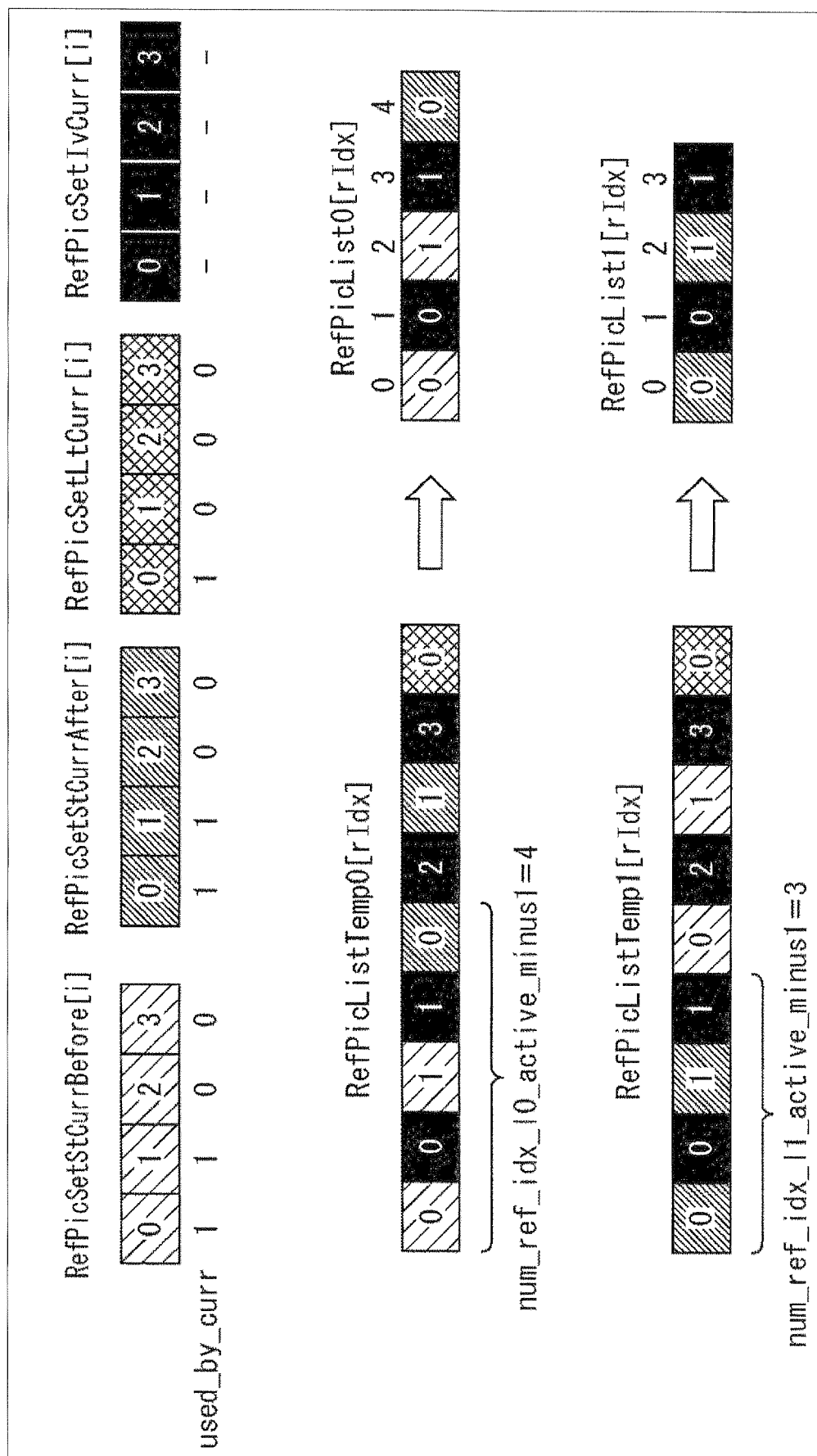


FIG. 7

|                      |           | CONVENTIONAL TECHNOLOGY 1                                                                                                                                                                                                                                                         | CONVENTIONAL TECHNOLOGY 2                                                                                                                                                                                                          | PRESENT TECHNOLOGY 1                                                                                                                                                                                                                                                           |
|----------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PRESENT TECHNOLOGY 1 | ADVANTAGE | <ul style="list-style-type: none"> <li>• CODING EFFICIENCY CAN BE IMPROVED FOR SEQUENCE HAVING HIGH INTER-VIEW CORRELATION (GENERAL CASE).</li> <li>• WHEN NUMBER OF STbC AND STaC IS SMALL, NUMBER OF UNIQUE Pictures CAN BE INCREASED IN WHOLE REFERENCE LIST L0/L1.</li> </ul> | <ul style="list-style-type: none"> <li>• ADDITIONAL SYNTAX DOES NOT NEED TO BE TRANSMITTED.</li> <li>• PROCESSING AMOUNT IS SMALL OWING TO NO REARRANGEMENT OF TEMPORAL LIST.</li> </ul>                                           | <div style="border: 1px solid black; height: 100px; width: 100%;"></div>                                                                                                                                                                                                       |
| PRESENT TECHNOLOGY 2 | ADVANTAGE | <ul style="list-style-type: none"> <li>• WHEN TEMPORAL CORRELATION AND INTER-VIEW CORRELATION ARE ALTERNATELY ALIGNED, CODING EFFICIENCY CAN BE IMPROVED.</li> <li>• INTER-VIEW CAN BE ARRANGED AT SECOND POSITION FROM START OF REFERENCE LIST ALSO IN WORST CASE.</li> </ul>    | <ul style="list-style-type: none"> <li>• ADDITIONAL SYNTAX DOES NOT NEED TO BE TRANSMITTED.</li> <li>• WHEN TEMPORAL CORRELATION AND INTER-VIEW CORRELATION ARE ALTERNATELY ALIGNED, CODING EFFICIENCY CAN BE IMPROVED.</li> </ul> | <ul style="list-style-type: none"> <li>• WHEN TEMPORAL CORRELATION AND INTER-VIEW CORRELATION ARE ALTERNATELY ALIGNED, CODING EFFICIENCY CAN BE IMPROVED.</li> <li>• INTER-VIEW CAN BE ARRANGED AT SECOND POSITION FROM START OF REFERENCE LIST ALSO IN WORST CASE.</li> </ul> |

FIG. 8

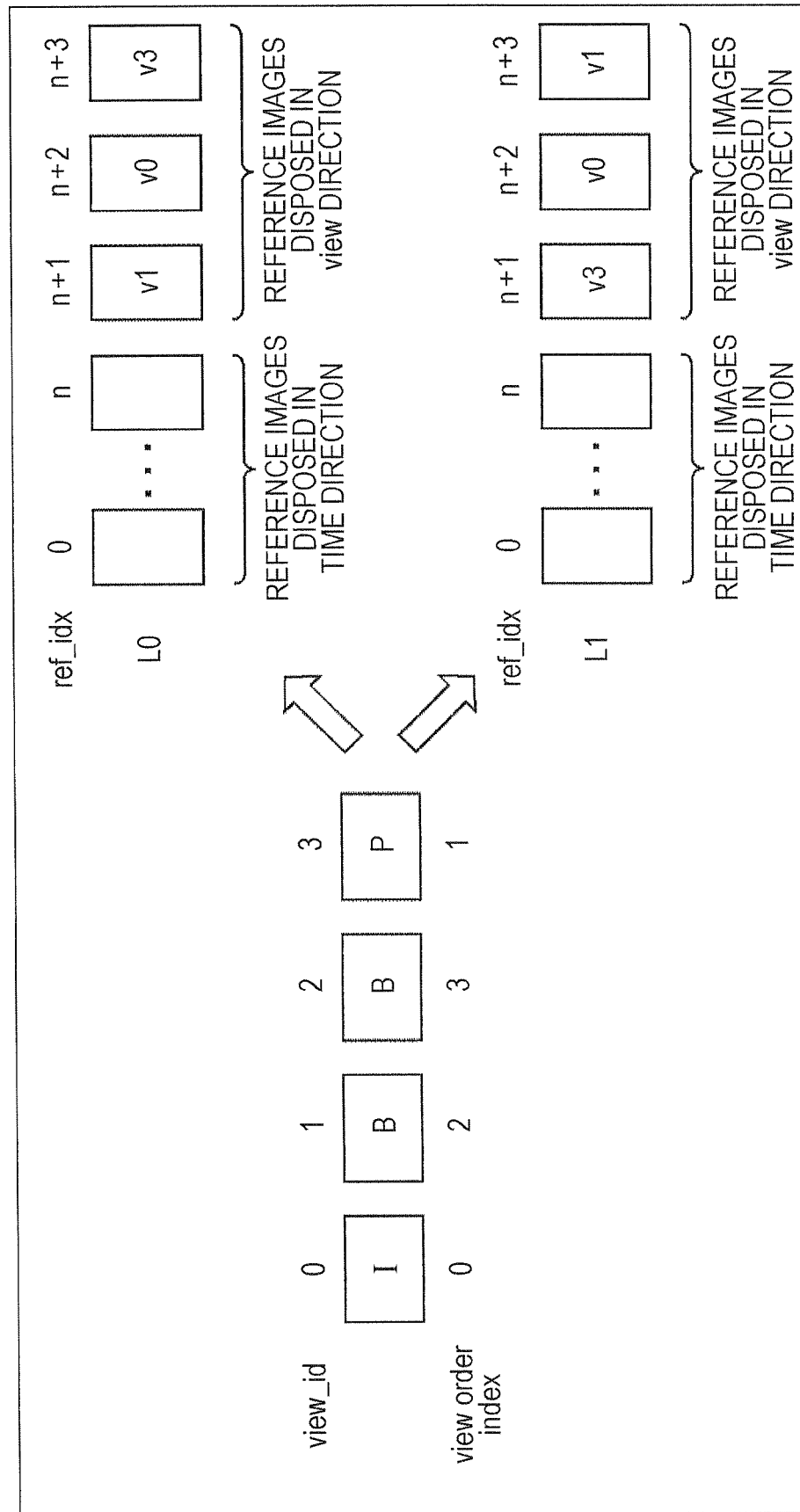


FIG. 9

```

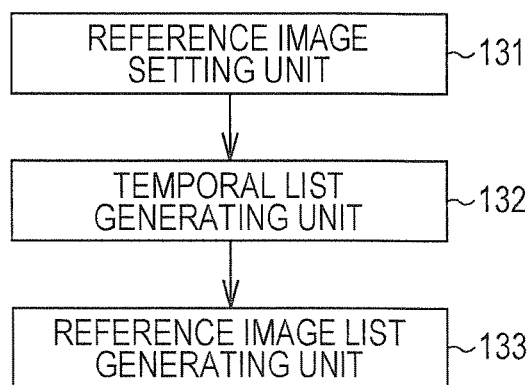
• The following applies:
• for( i = 0; i < NumIvCurr; i++ )
  RefPicSetIvCurr[i] = "inter-view reference picture with layer identifier equal to LayerIdInterView[i]"

• rIdx = 0
  while( rIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0; rIdx++, i++ )
      RefPicListTemp0[rIdx] = RefPicSetStCurrBefore[i]
    for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0; rIdx++, i++ )
      RefPicListTemp0[rIdx] = RefPicSetStCurrAfter[i]
    for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList0; rIdx++, i++ )
      RefPicListTemp0[rIdx] = RefPicSetLtCurr[i]
    for( i = 0; i < NumIvCurr && rIdx < NumRpsCurrTempList0; rIdx++, i++ )
      RefPicListTemp0[rIdx] = RefPicSetIvCurr[i]
  }

• rIdx = 0
  while( rIdx < NumRpsCurrTempList1 ) {
    for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
      RefPicListTemp1[rIdx] = RefPicSetStCurrAfter[i]
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
      RefPicListTemp1[rIdx] = RefPicSetStCurrBefore[i]
    for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
      RefPicListTemp1[rIdx] = RefPicSetLtCurr[i]
    for( i = 0; i < NumIvCurr && rIdx < NumRpsCurrTempList1; rIdx++, i++ )
      RefPicListTemp1[rIdx] = RefPicSetIvCurr[NumIvCurr-i]
  }

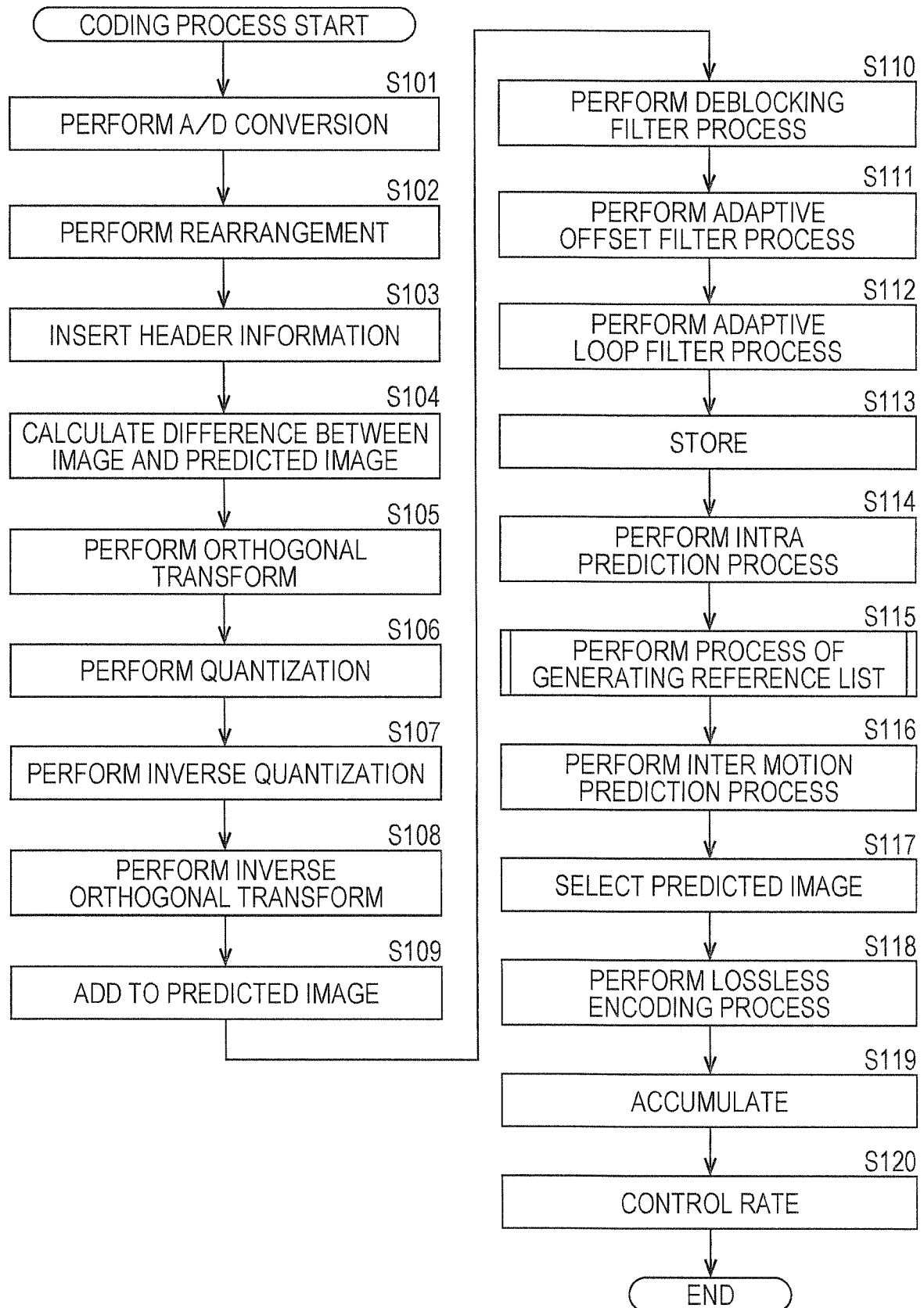
```

*FIG. 10*



↑  
122

FIG. 11



12/28

**FIG. 12**

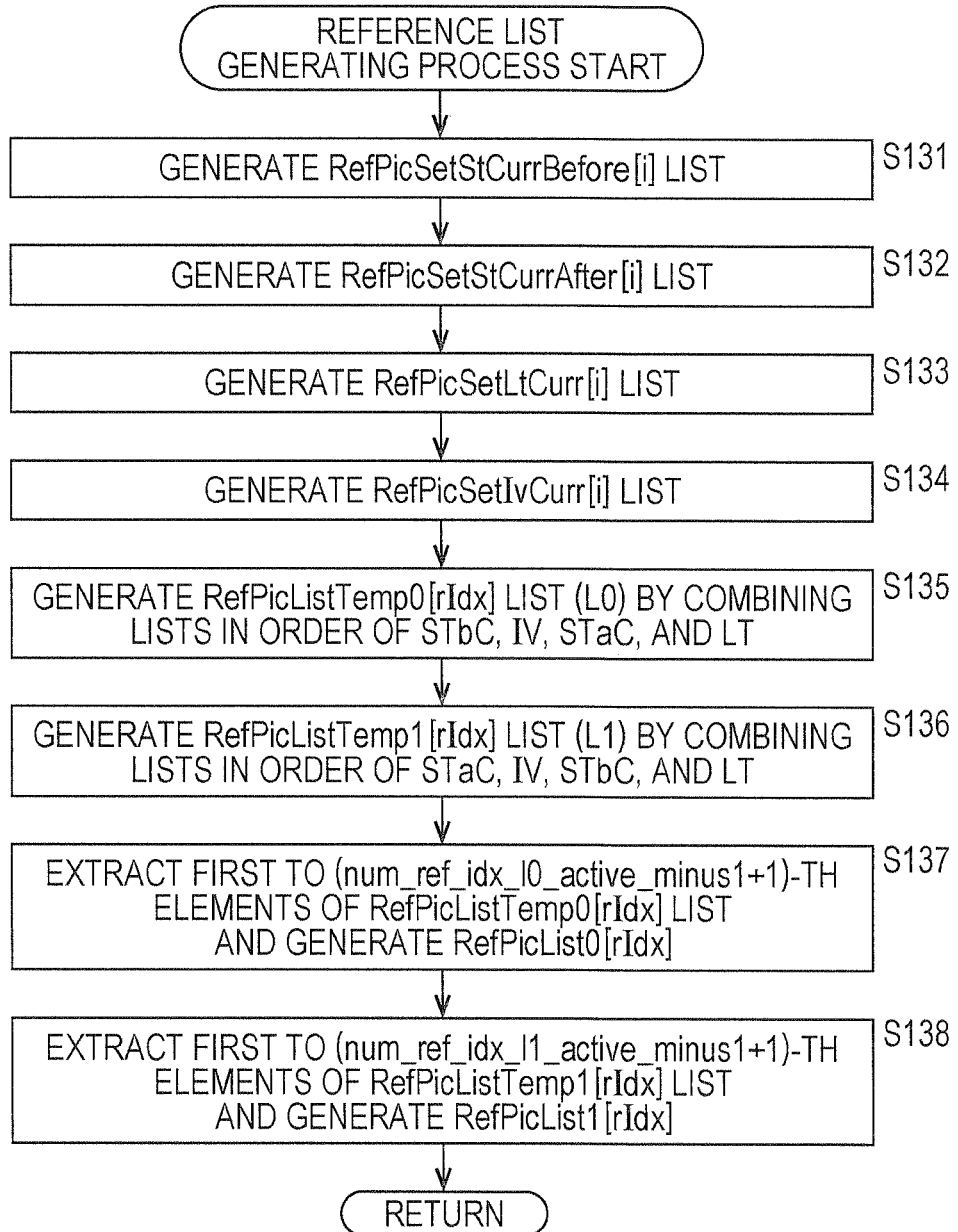


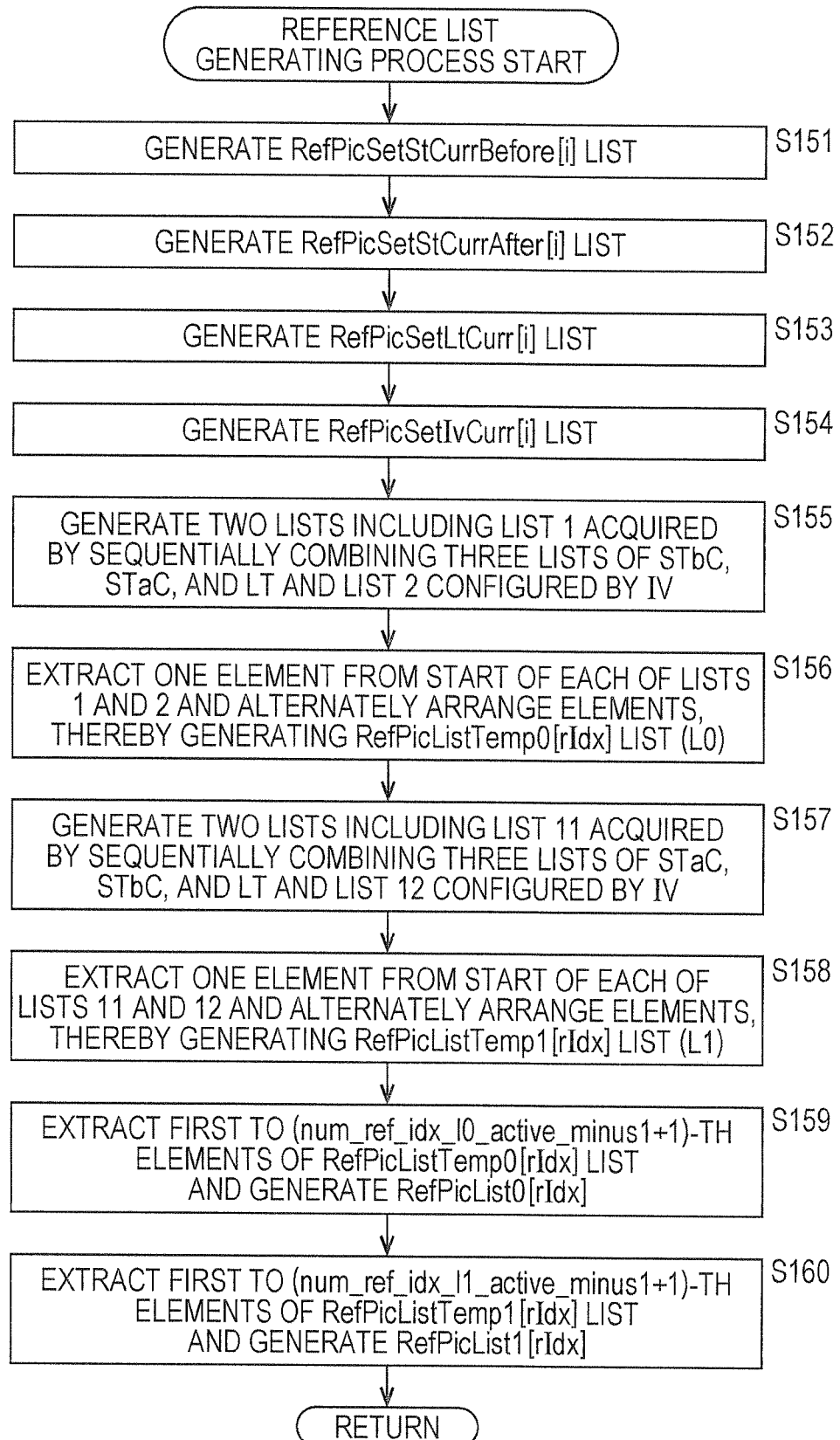
**FIG. 13**

FIG. 14

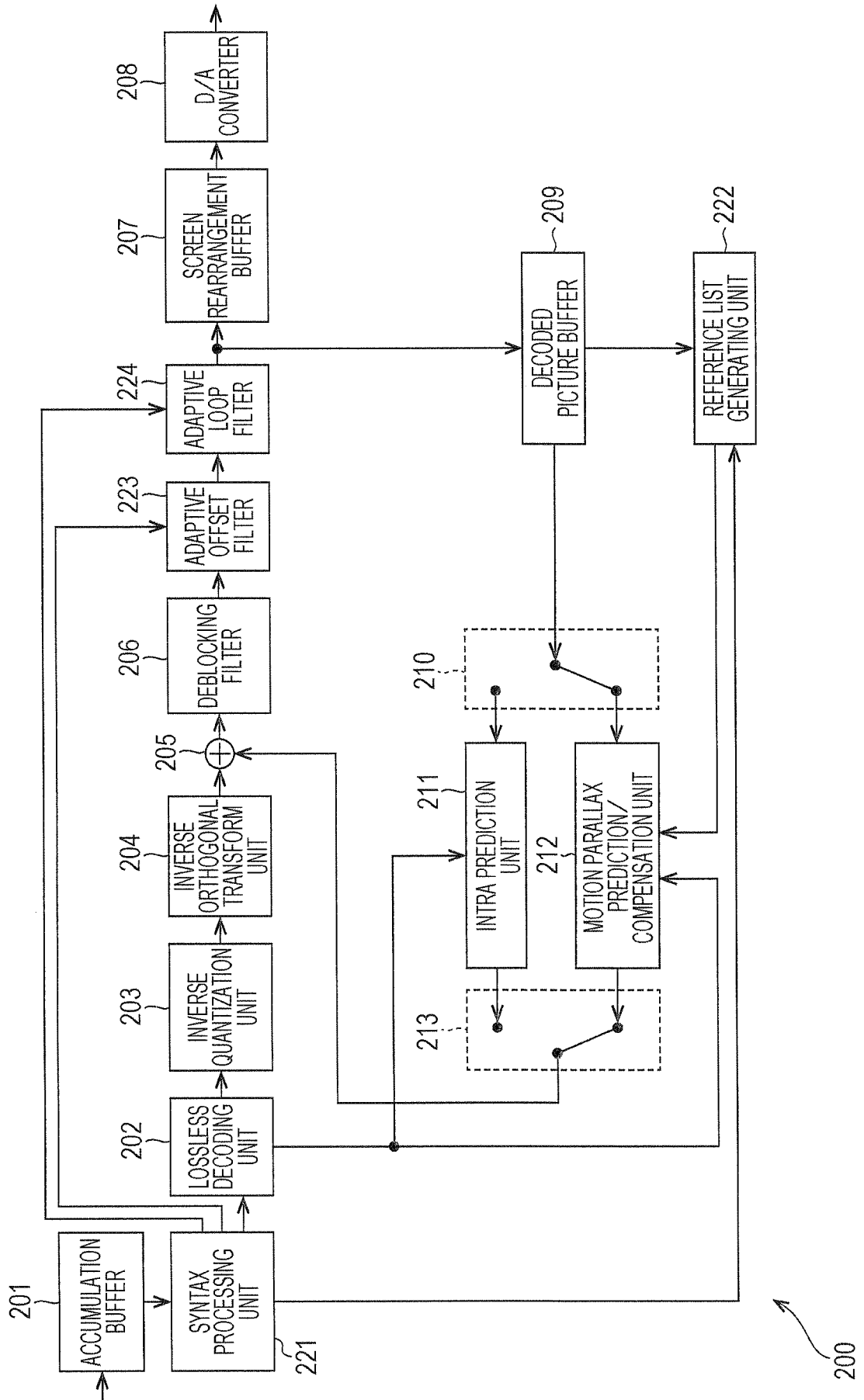


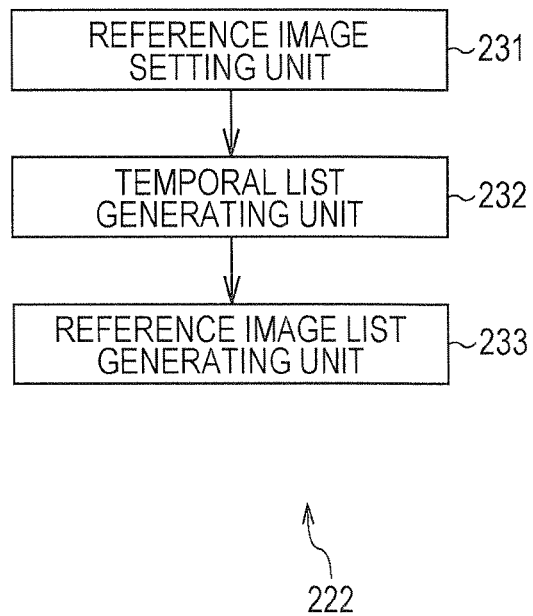
*FIG. 15*

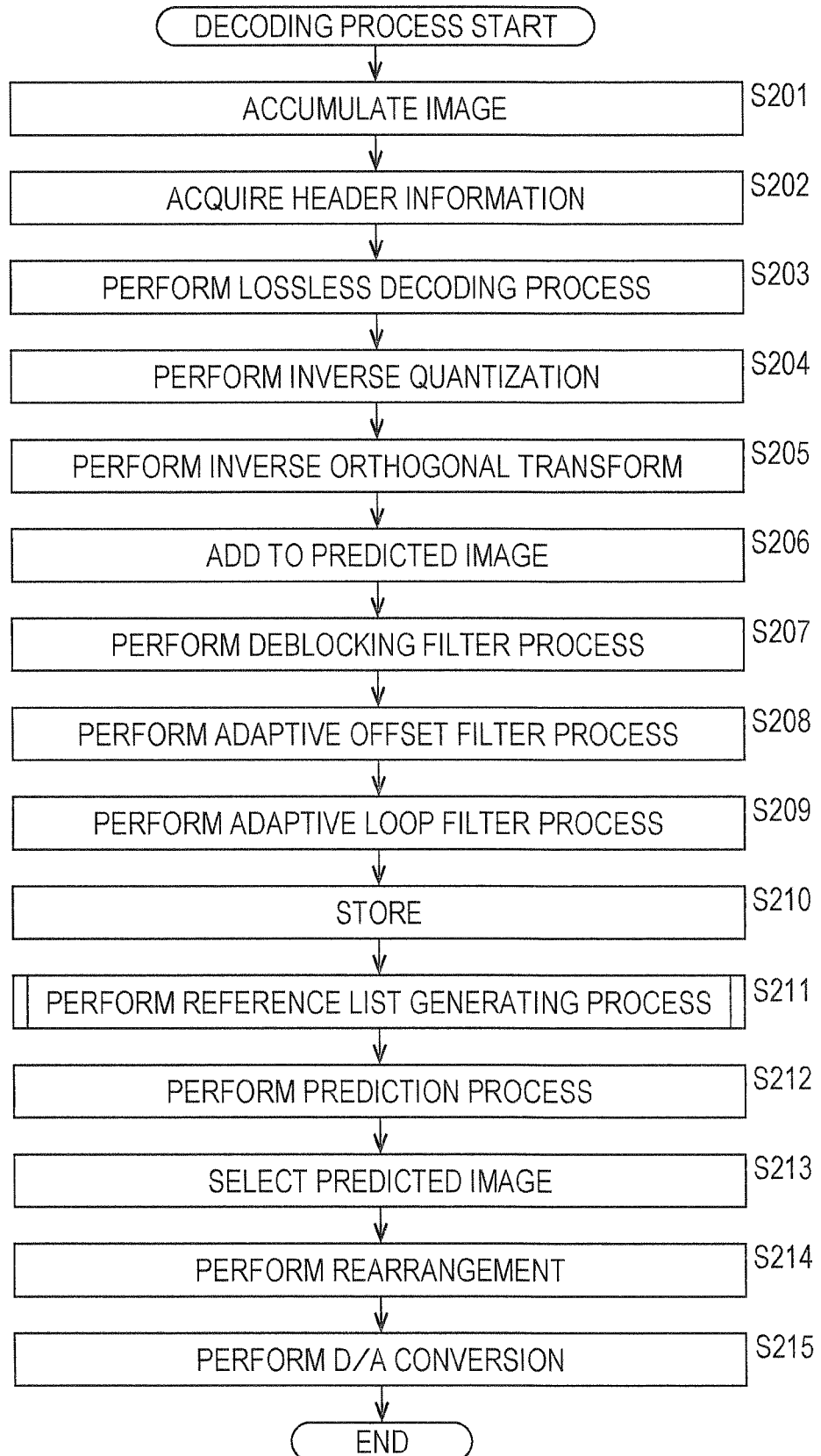
**FIG. 16**

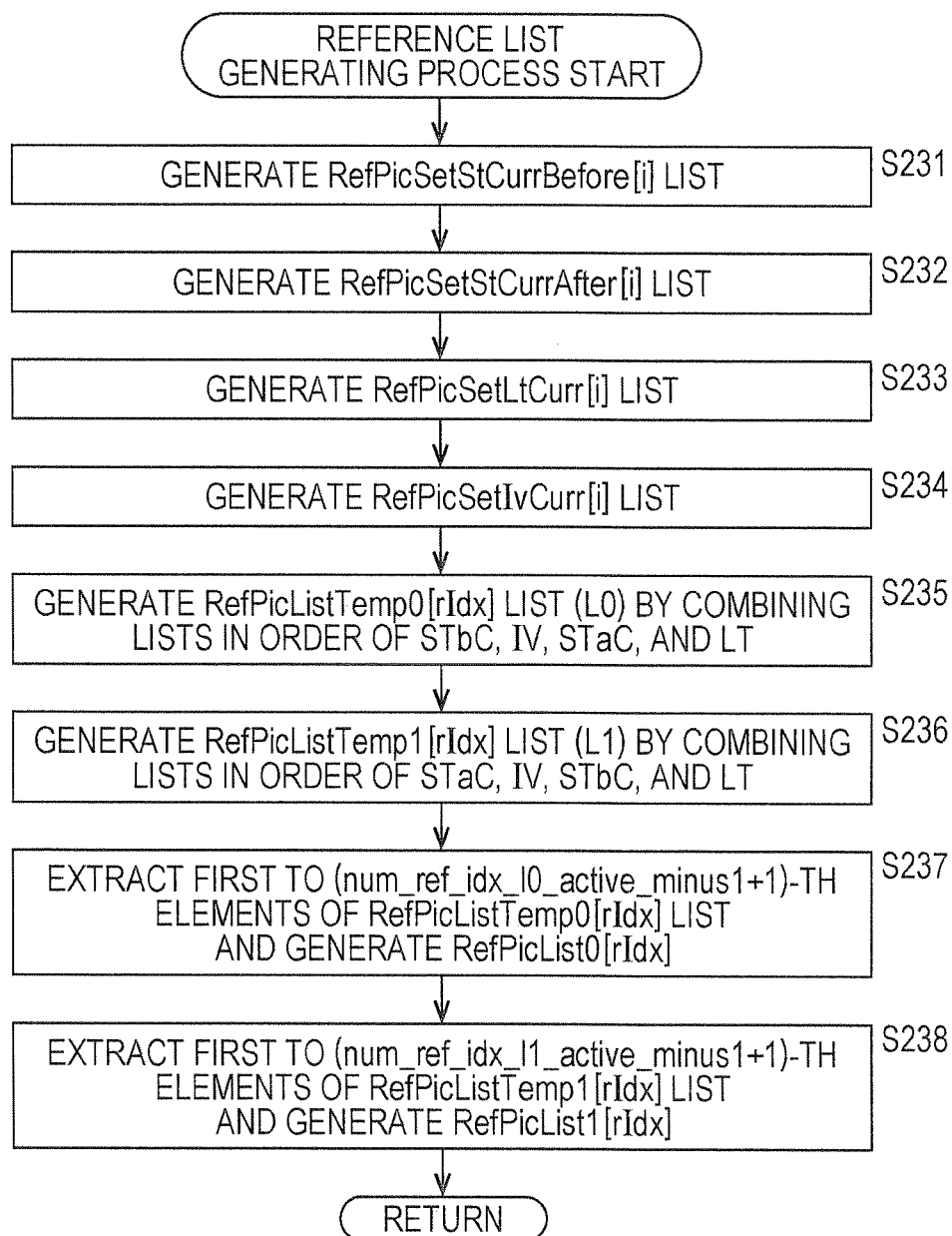
**FIG. 17**

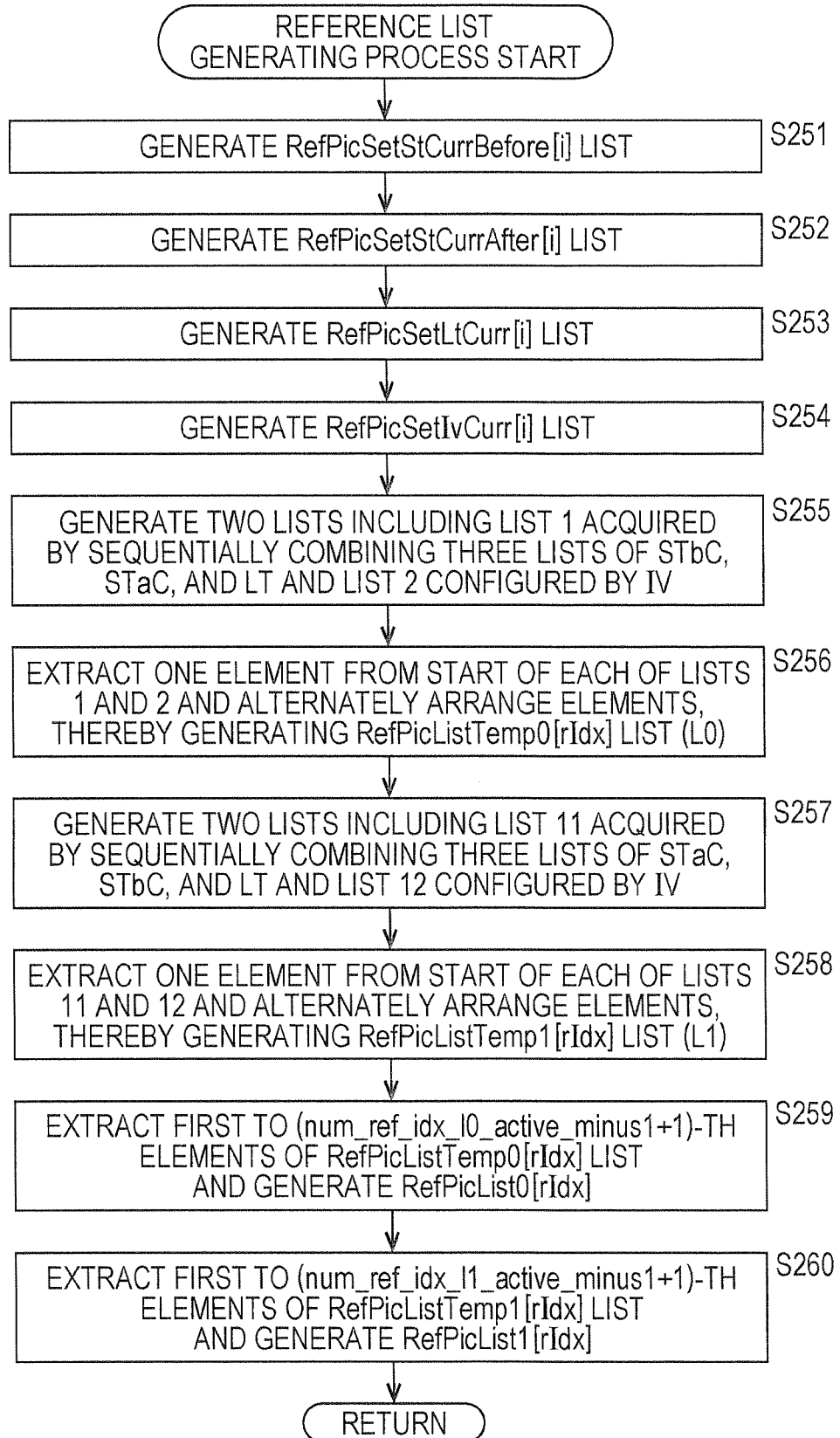
**FIG. 18**



FIG. 20

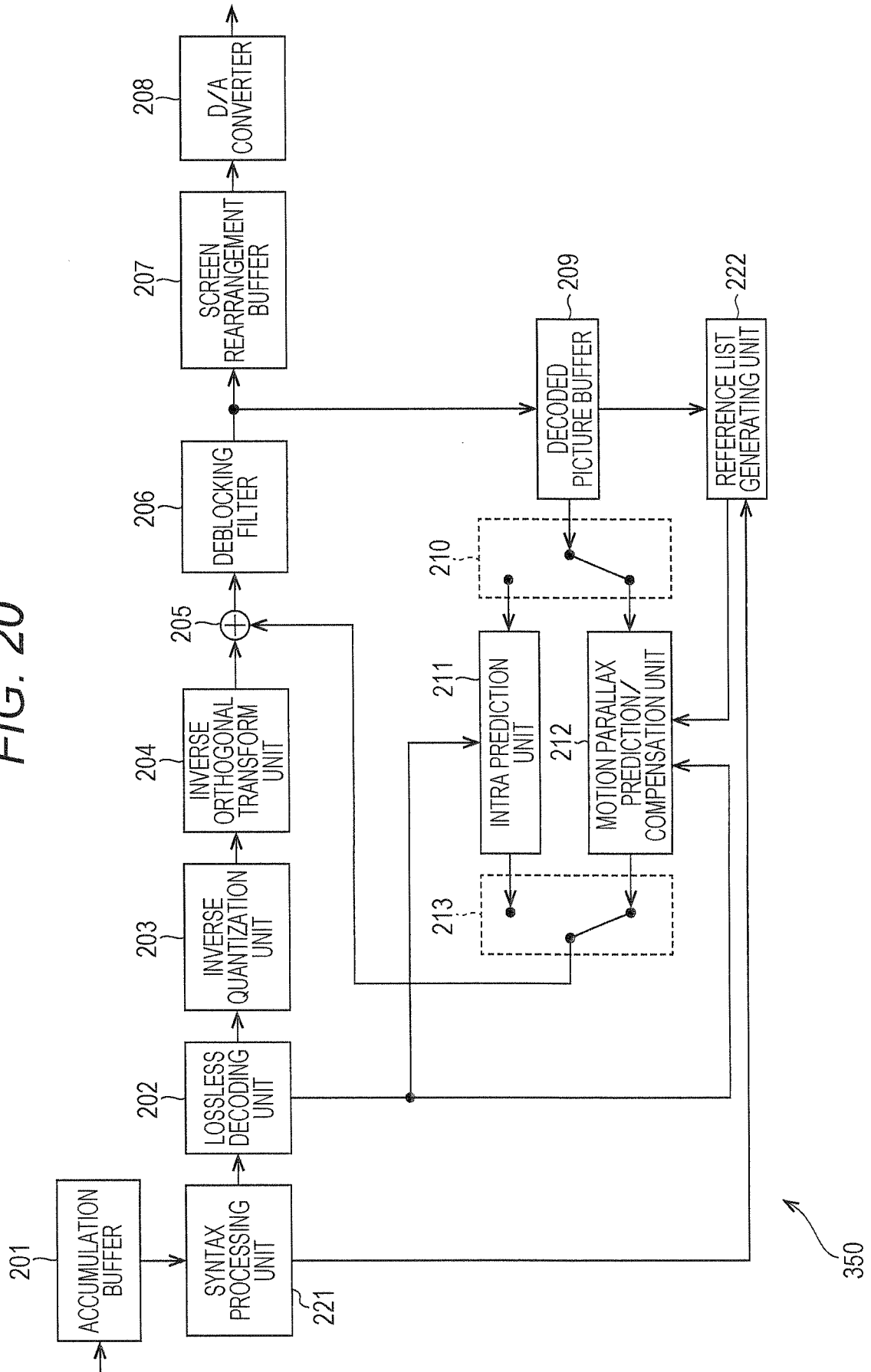


FIG. 21

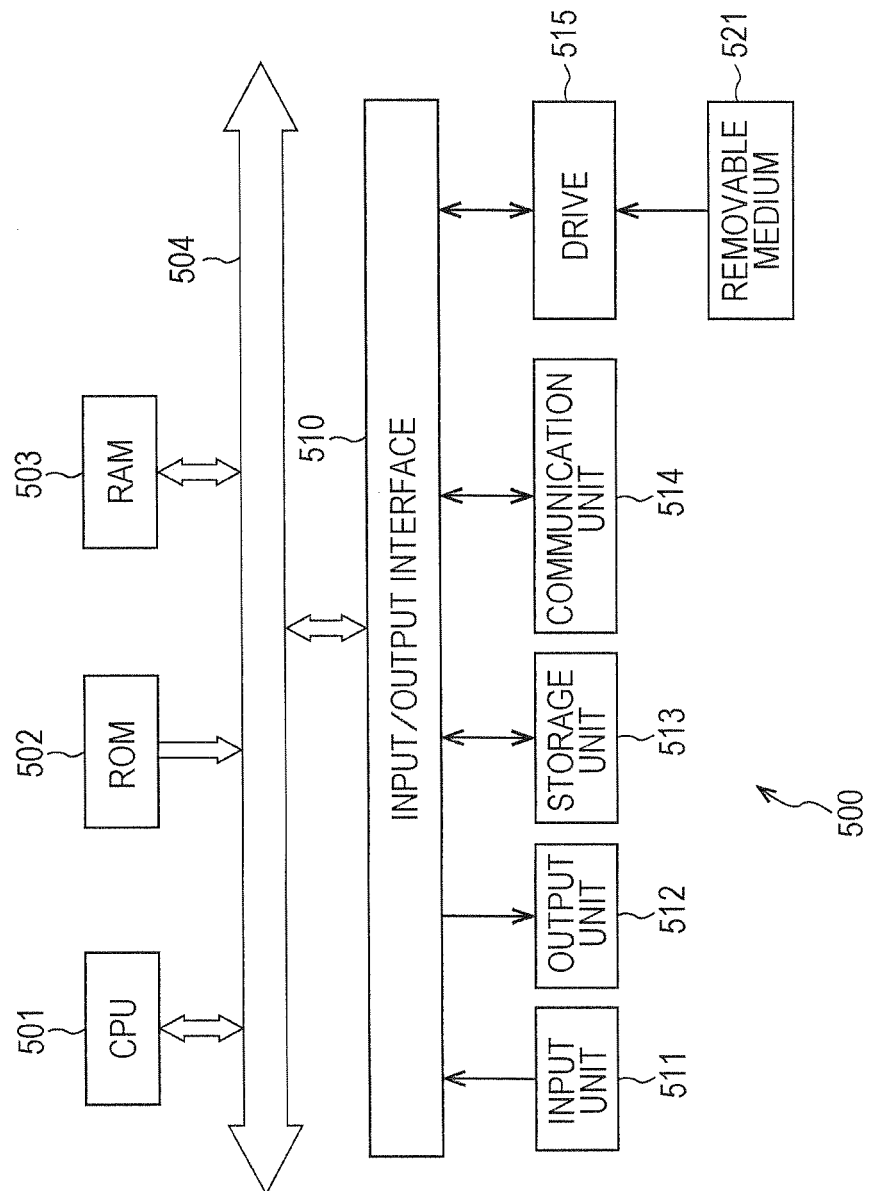


FIG. 22

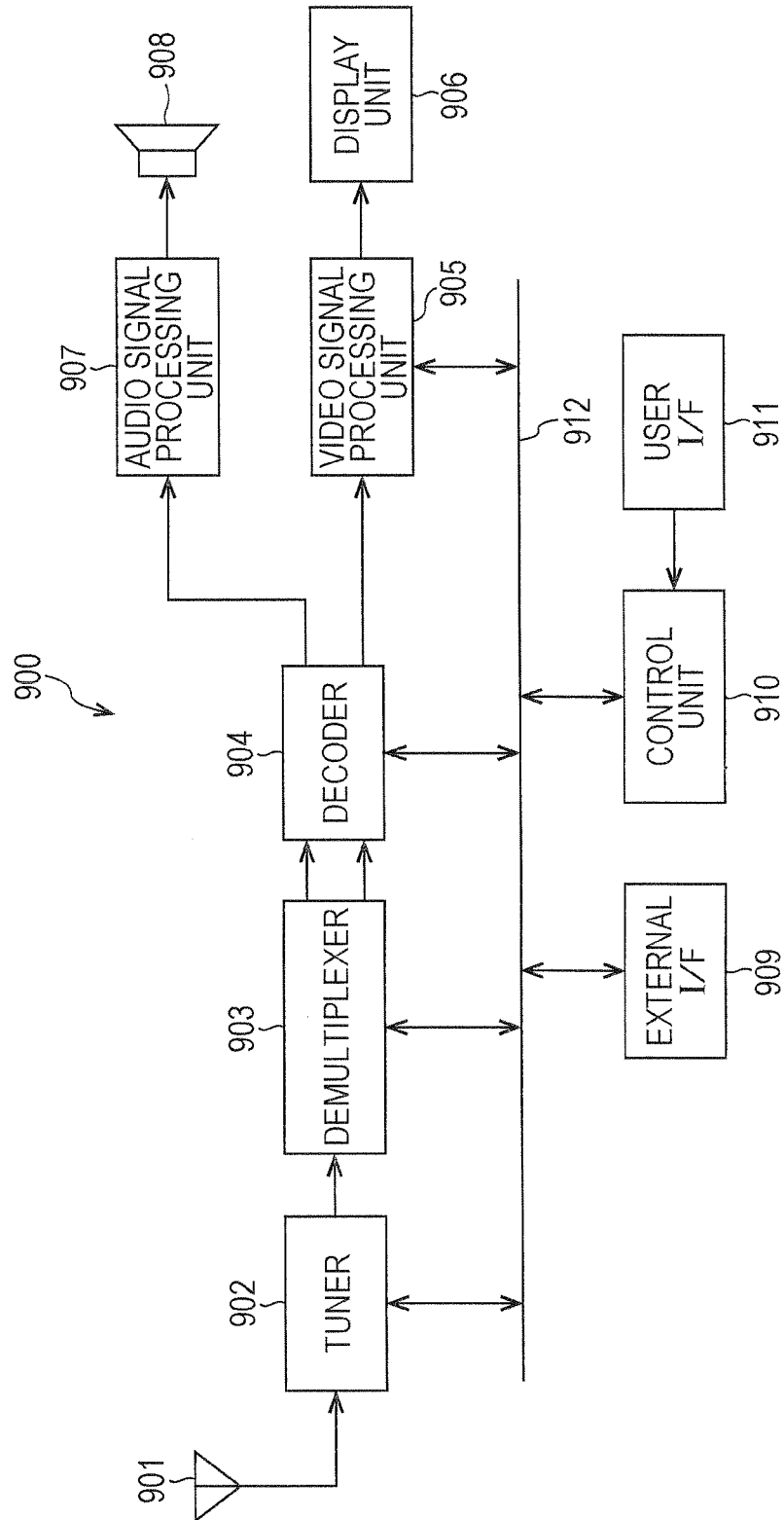


FIG. 23

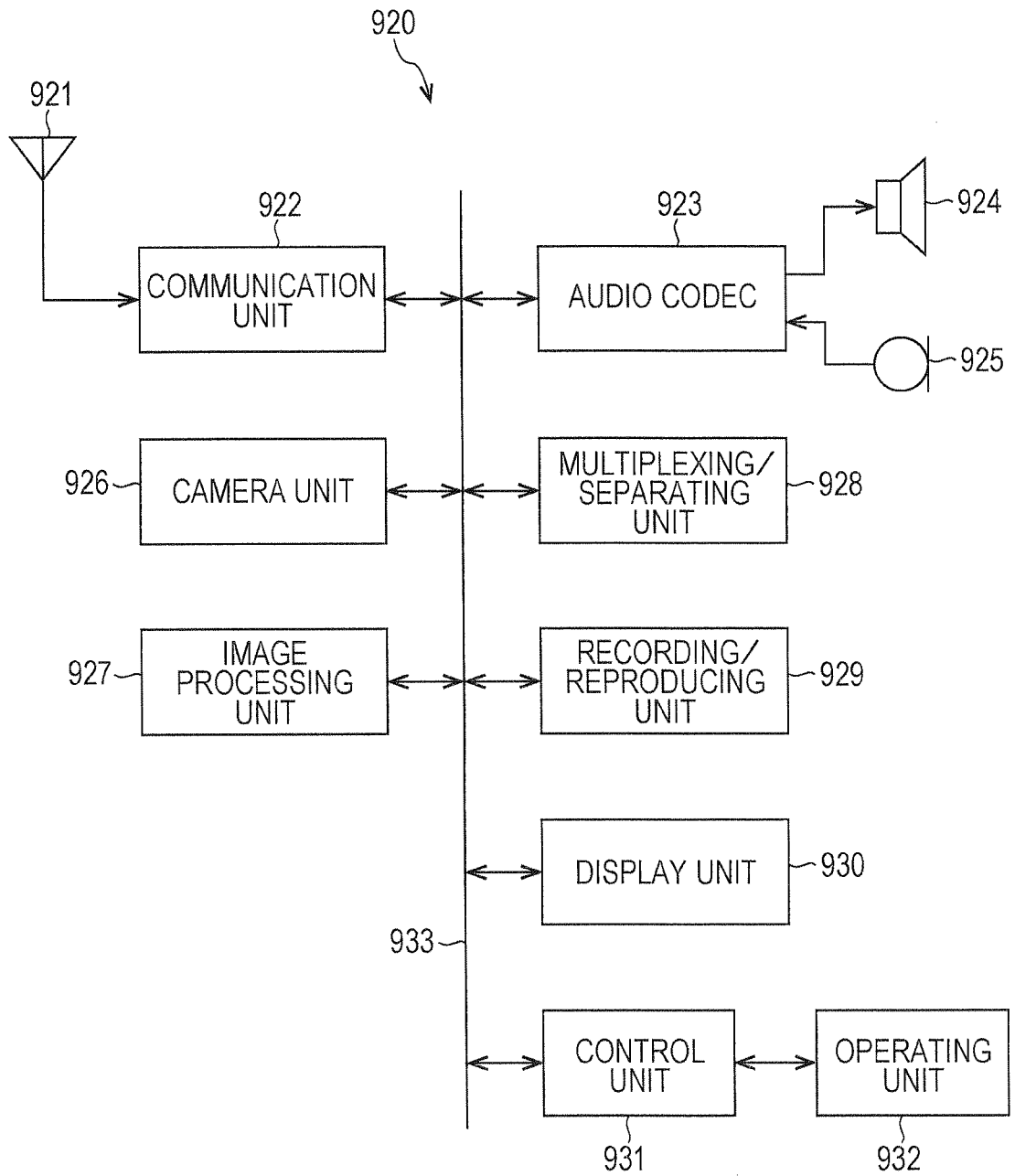


FIG. 24

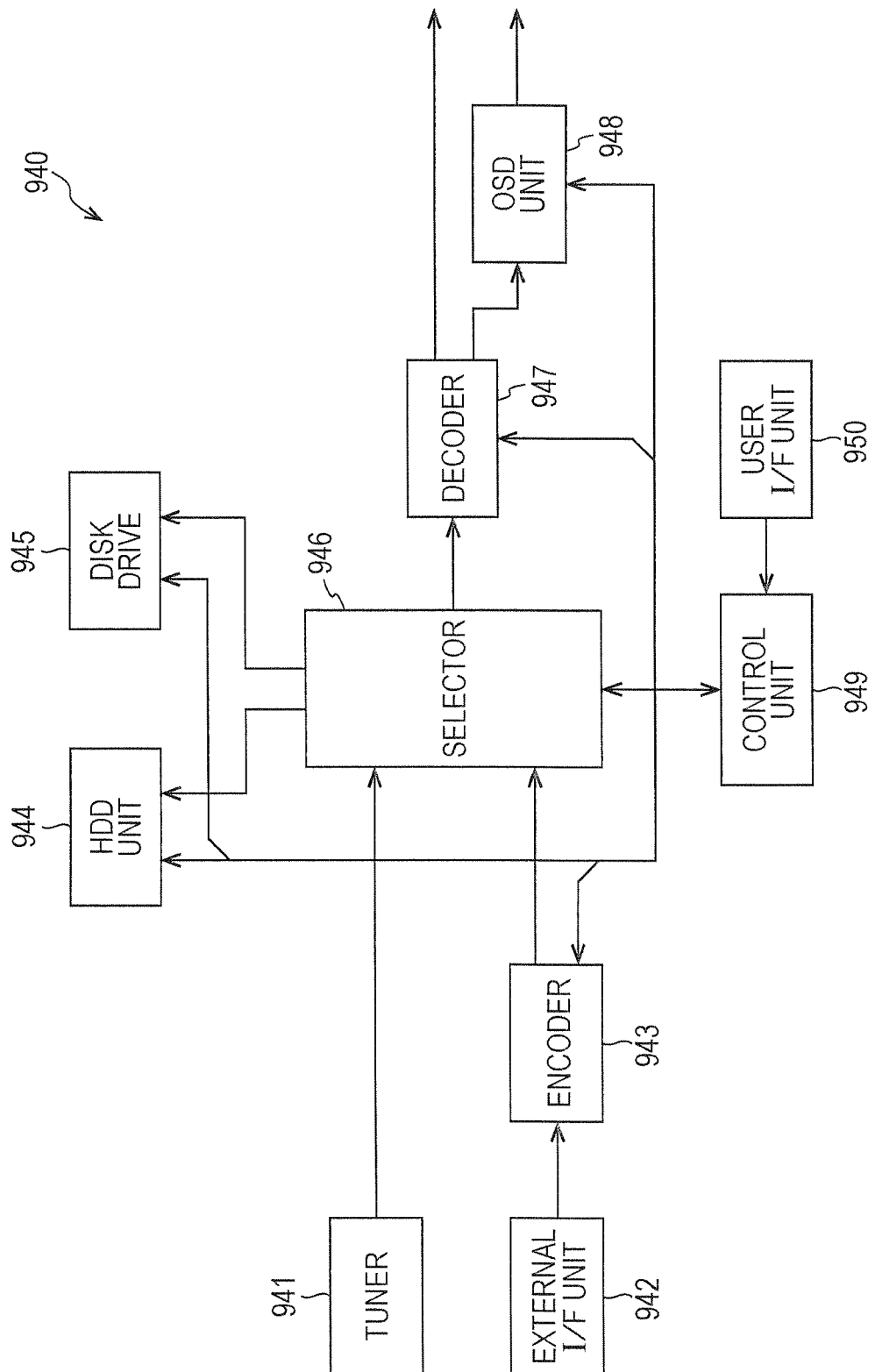


FIG. 25

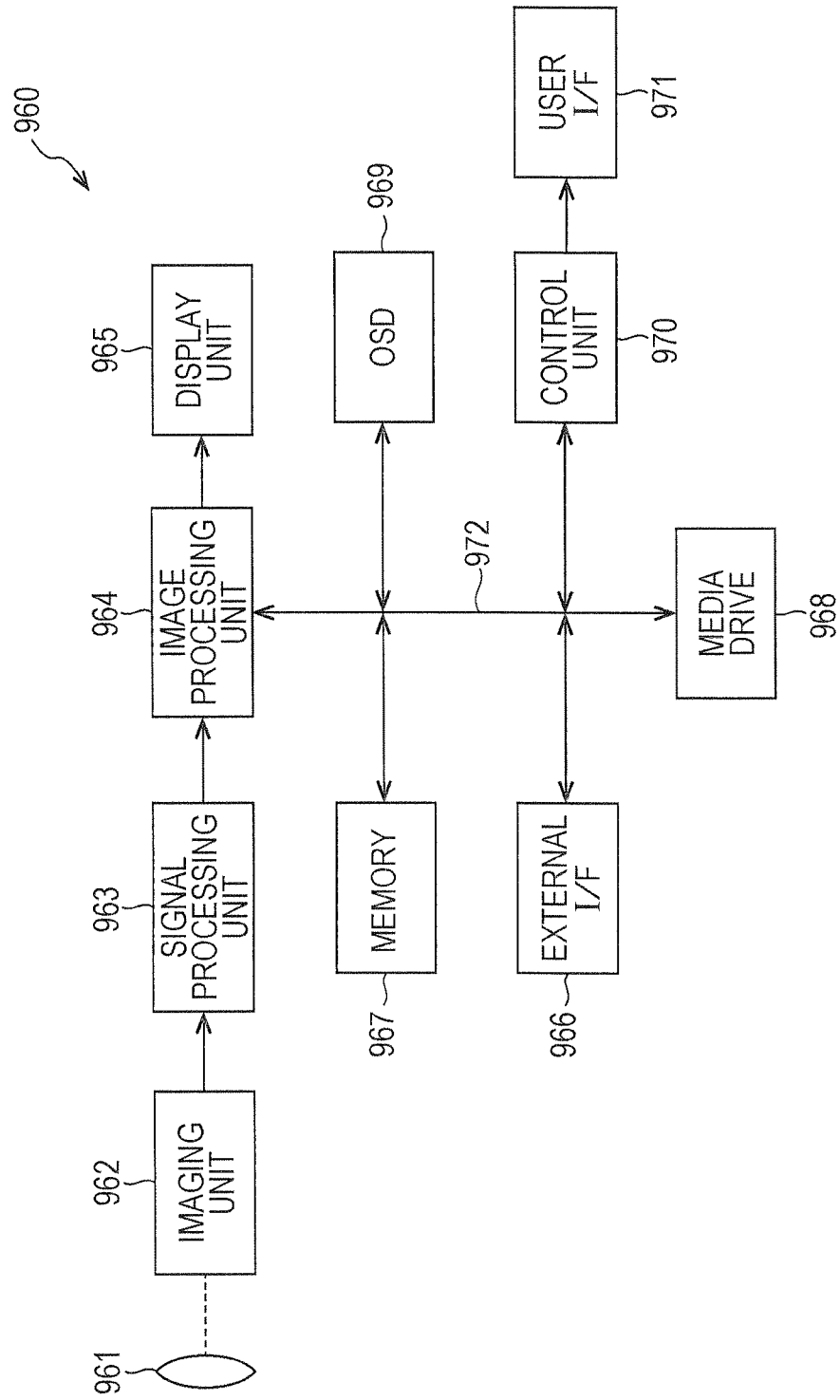


FIG. 26

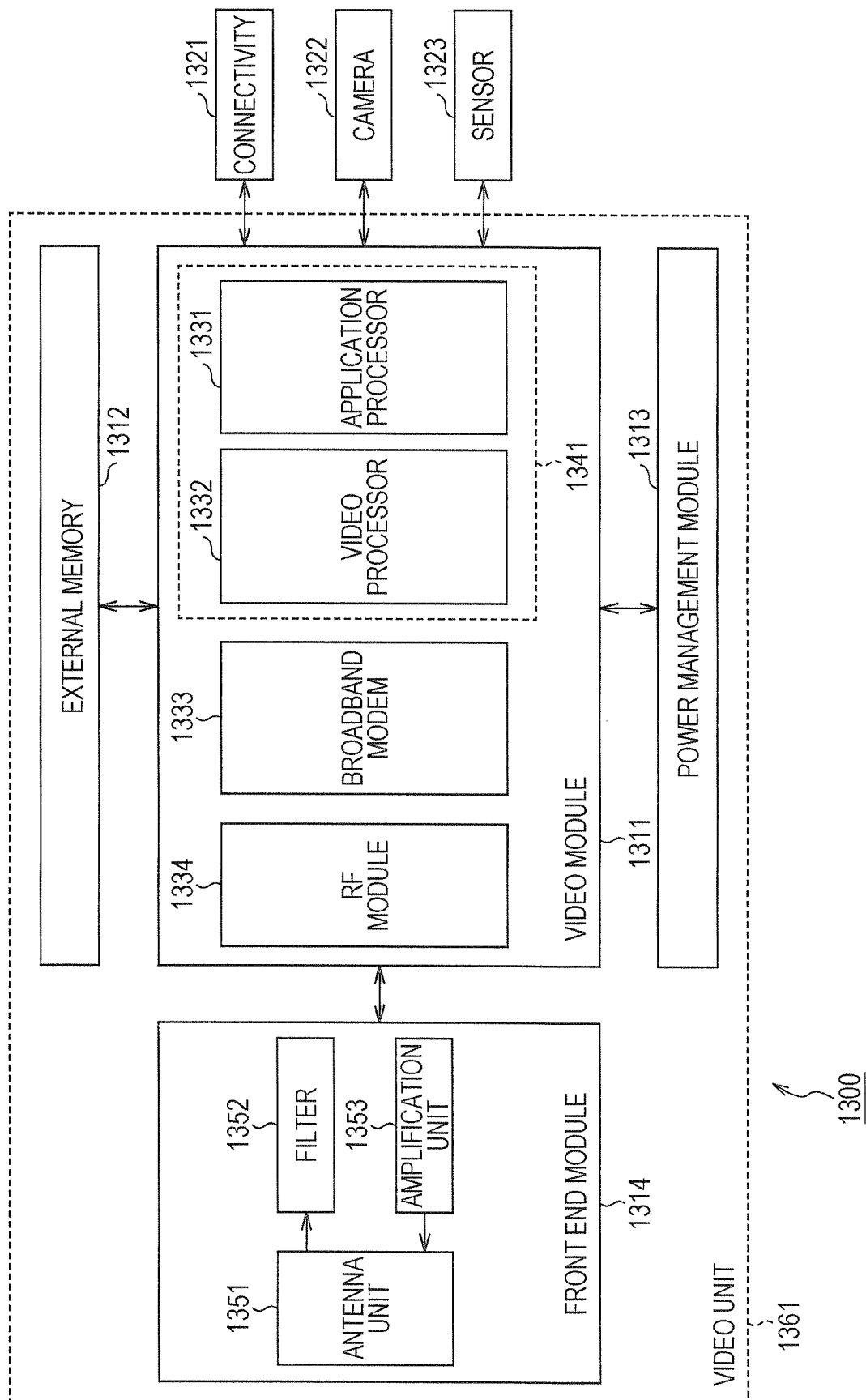


FIG. 27

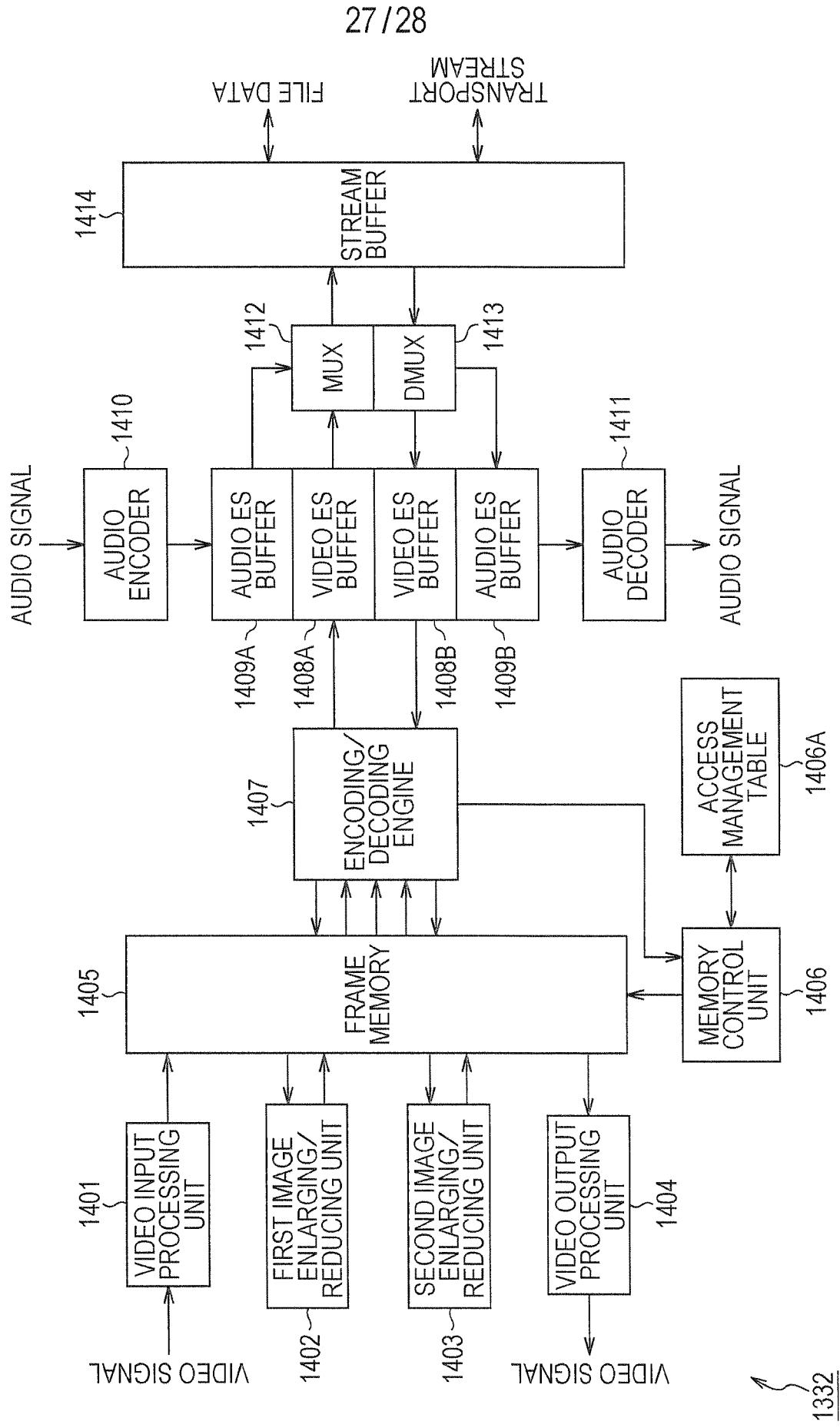


FIG. 28

