

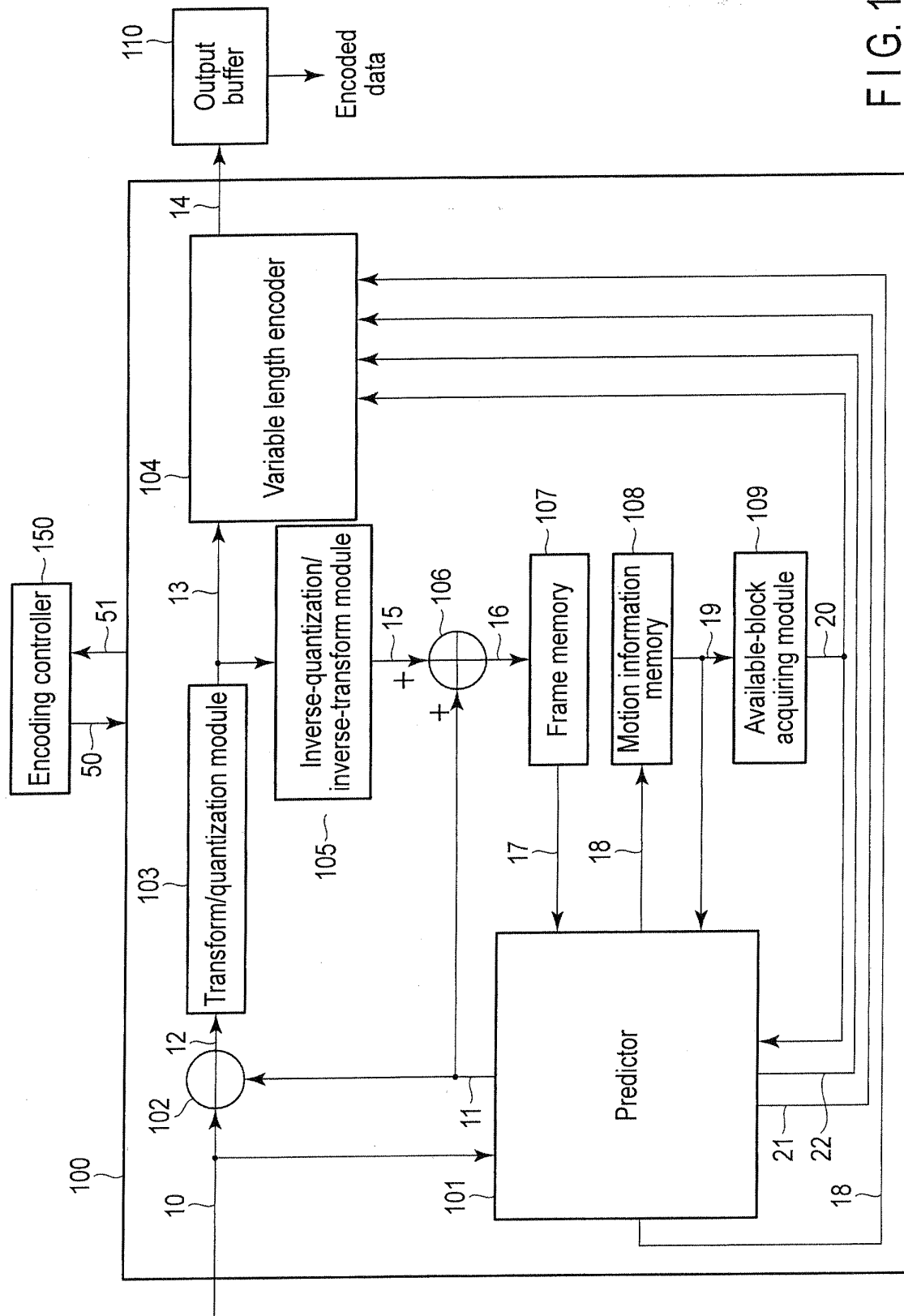


FIG. 1

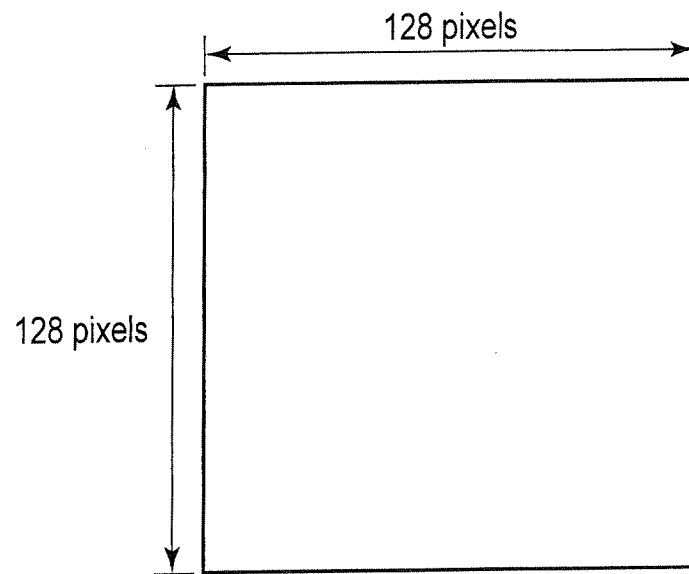


FIG. 2A

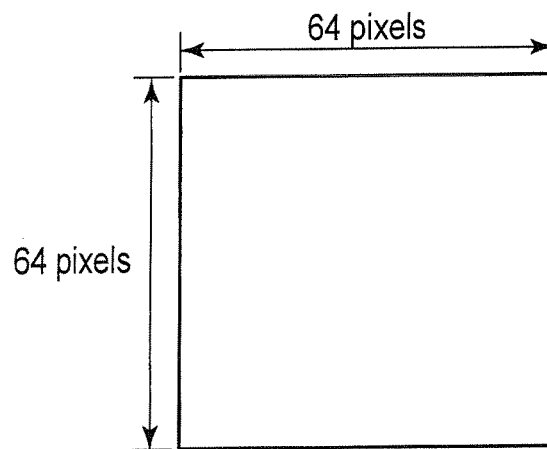


FIG. 2B

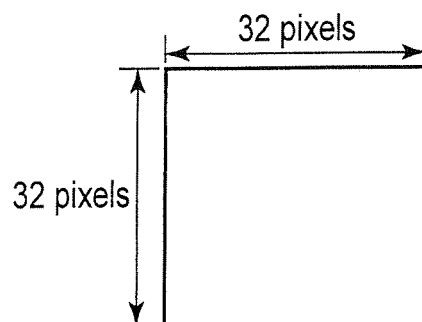


FIG. 2C

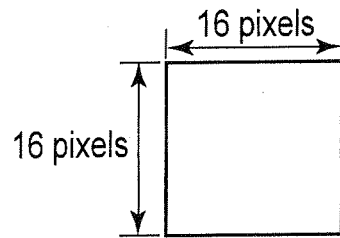


FIG. 2D

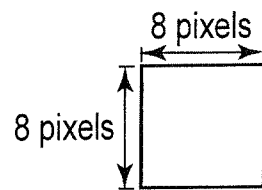


FIG. 2E

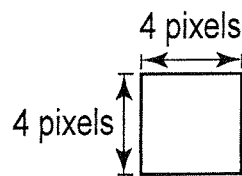


FIG. 2F

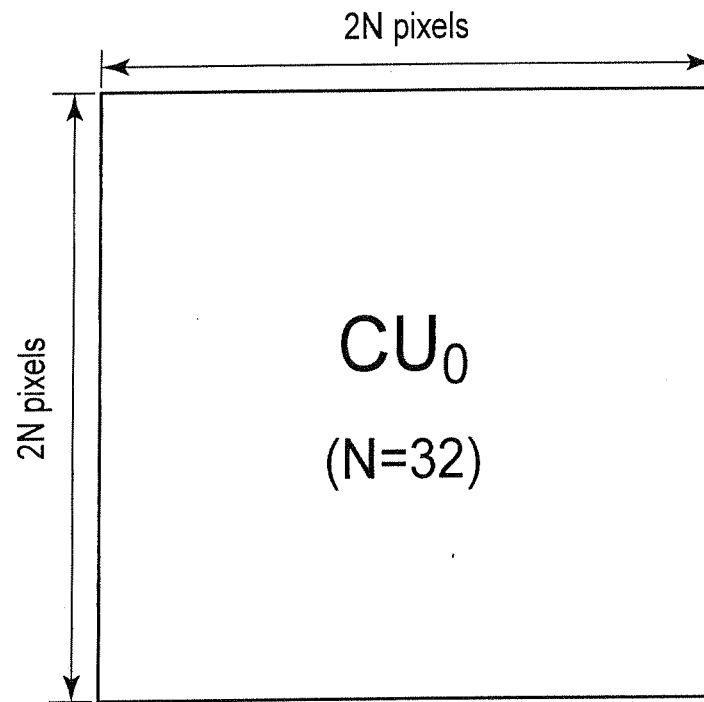


FIG. 3A

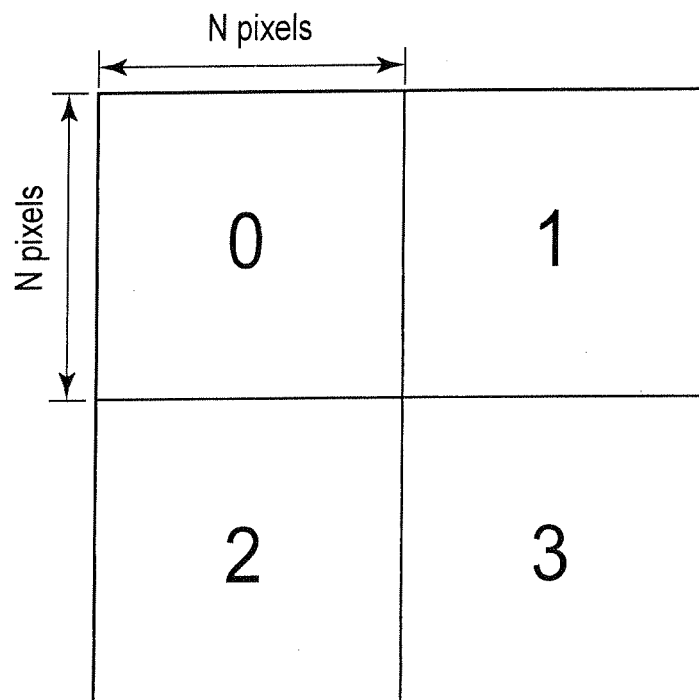


FIG. 3B

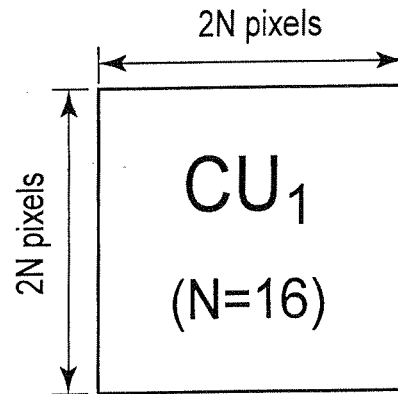


FIG. 3C

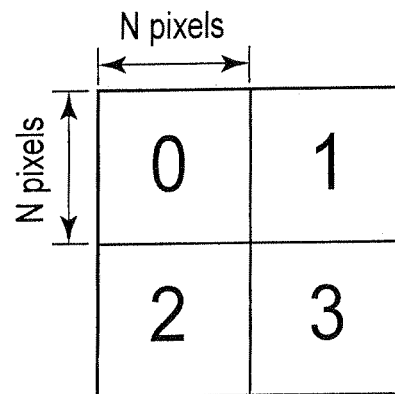


FIG. 3D

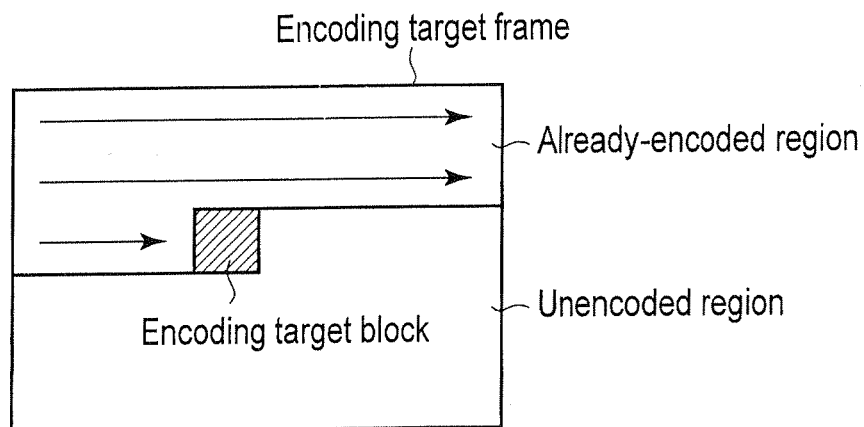


FIG. 4

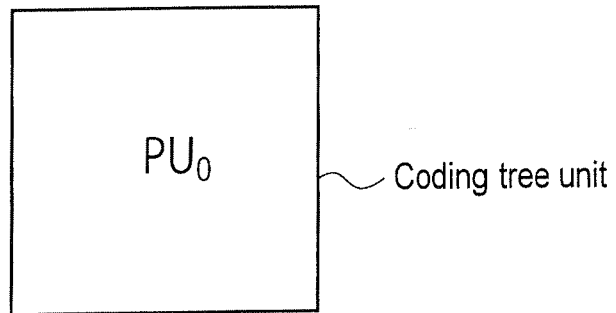


FIG. 5A

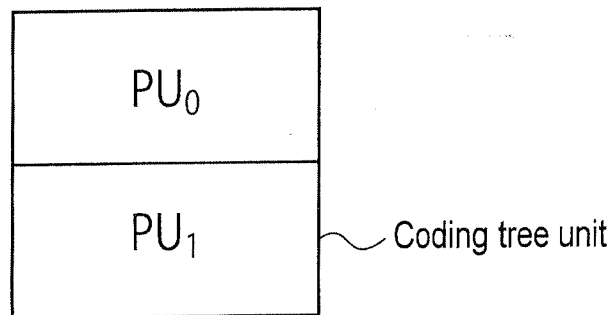


FIG. 5B

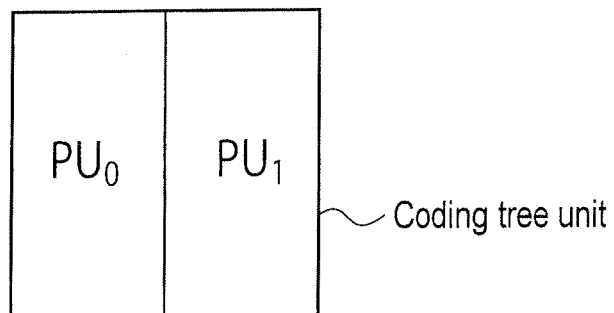


FIG. 5C

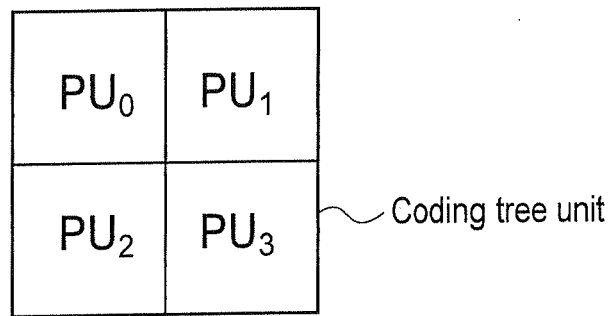


FIG. 5D

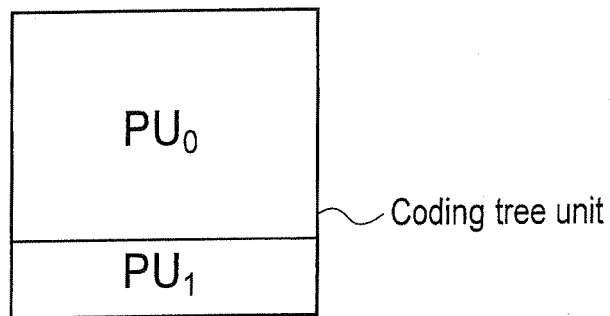


FIG. 5E

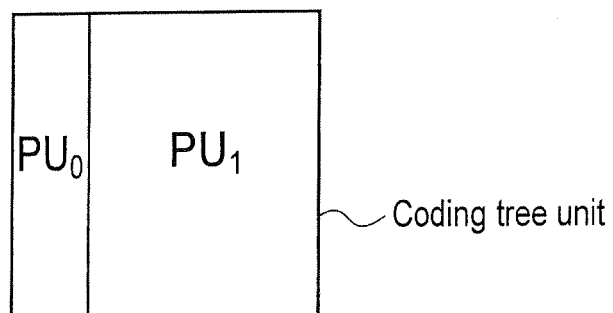


FIG. 5F

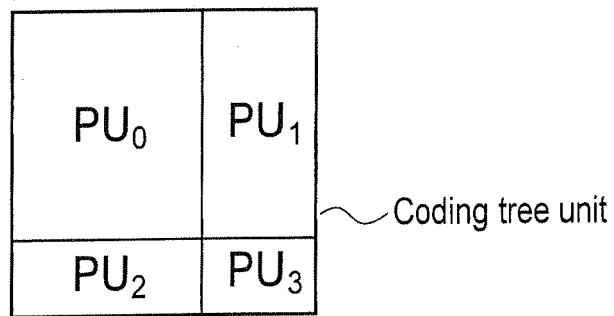


FIG. 5G

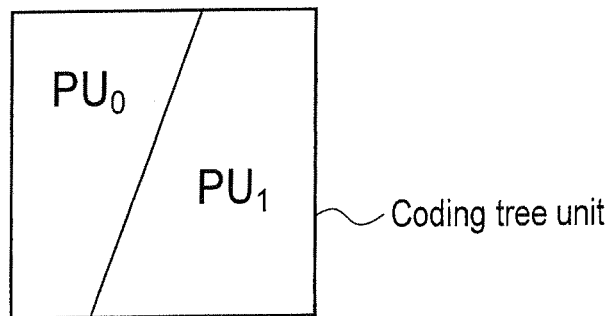


FIG. 5H

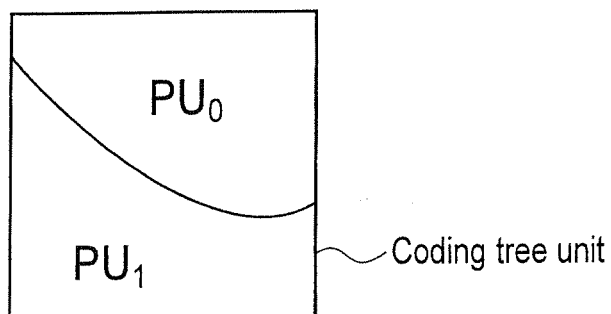


FIG. 5I

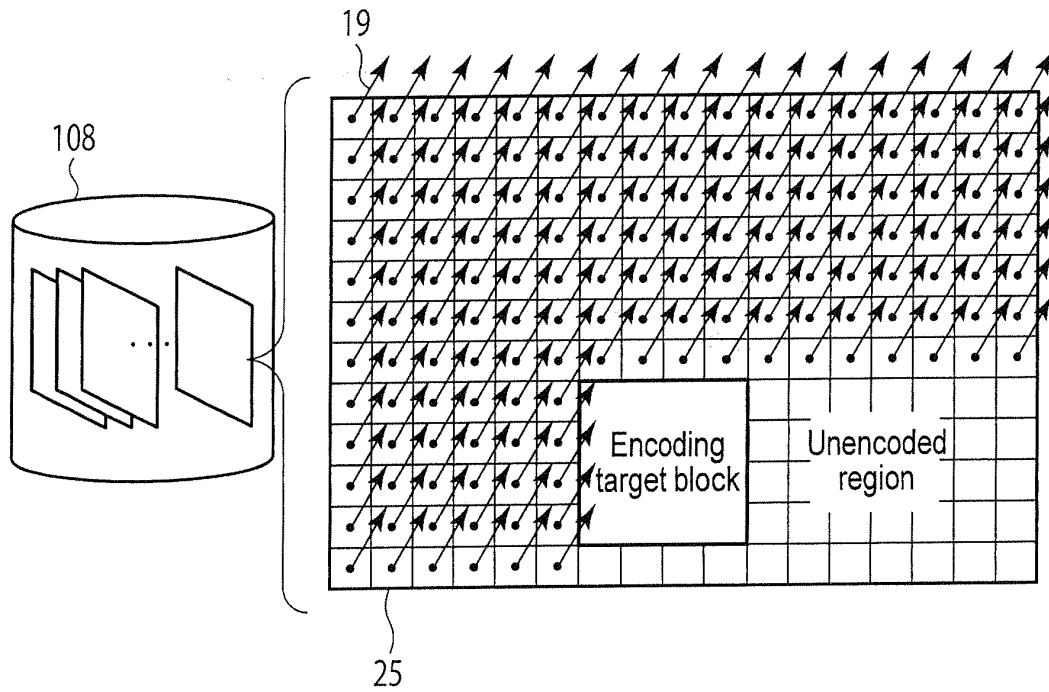


FIG. 6A

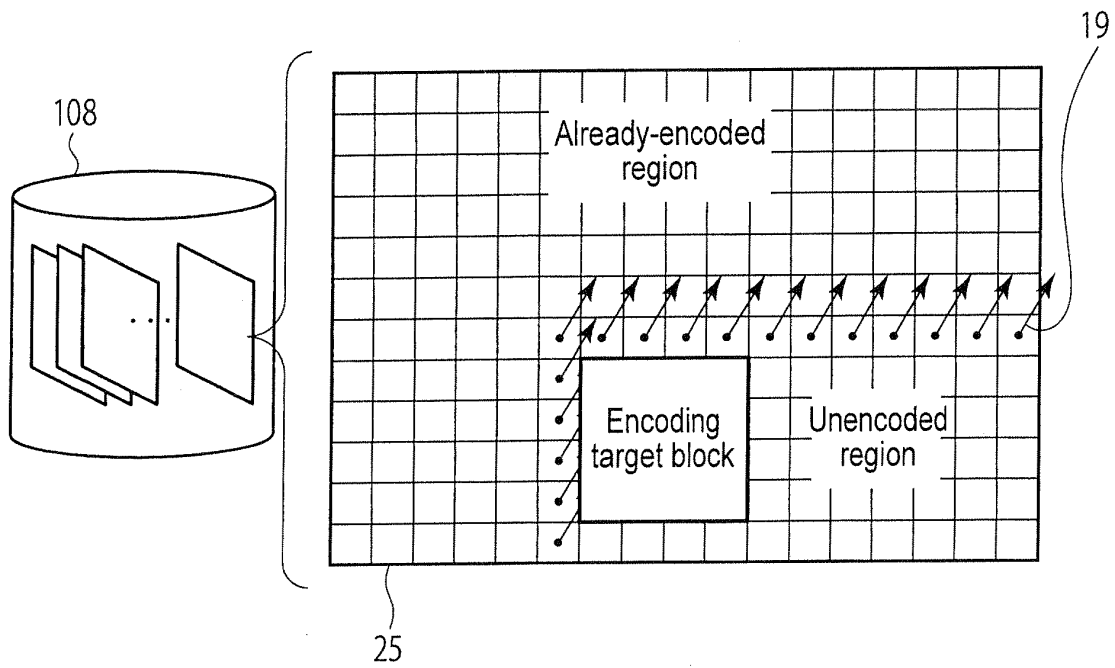


FIG. 6B

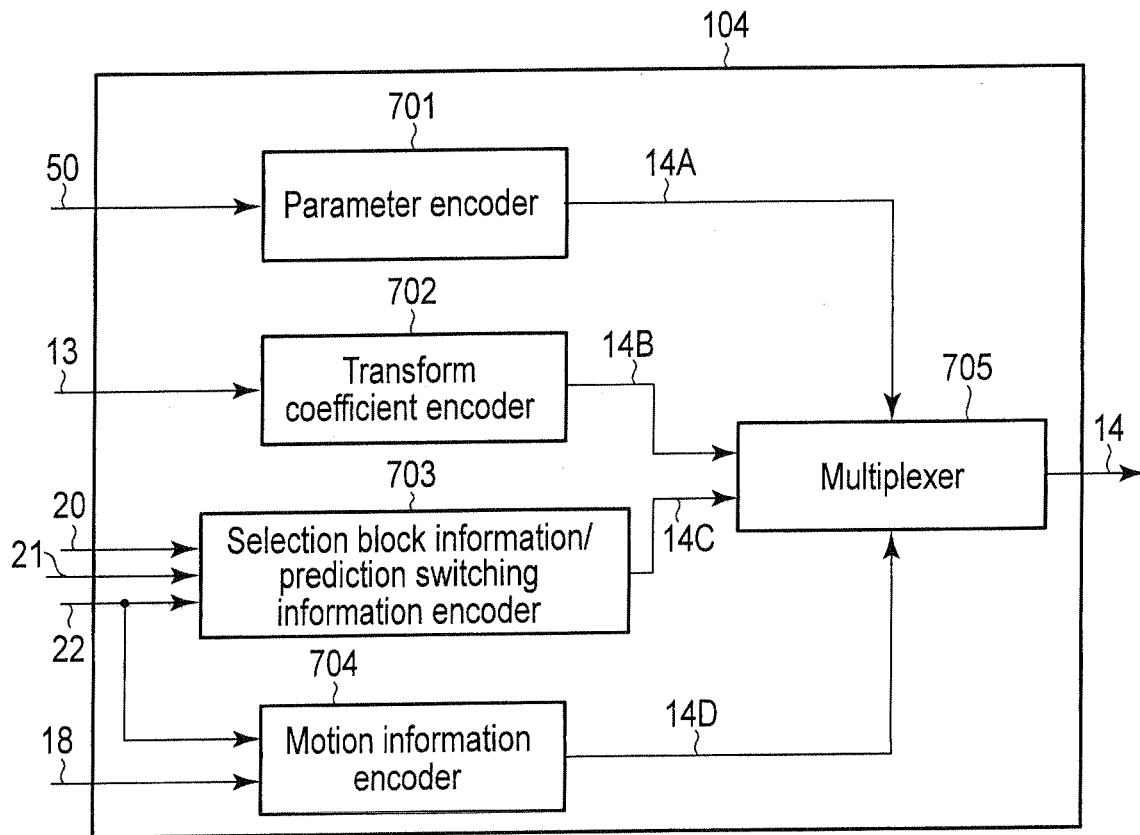


FIG. 7

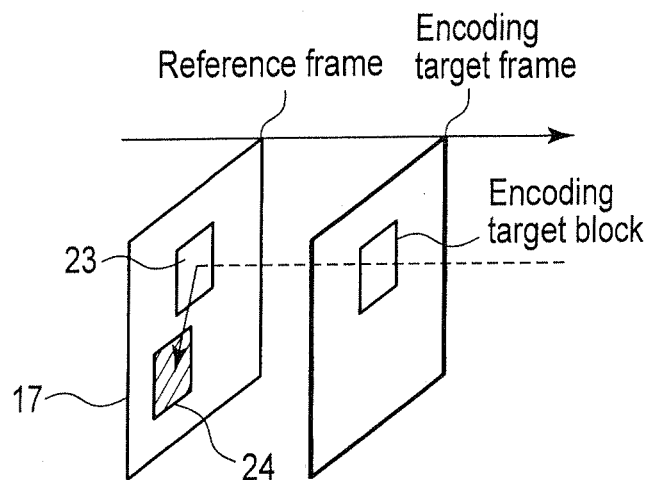


FIG. 8A

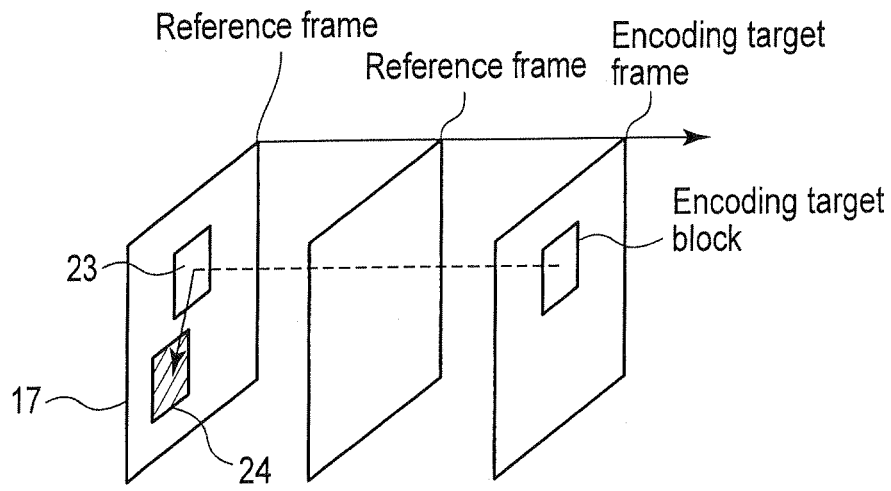


FIG. 8B

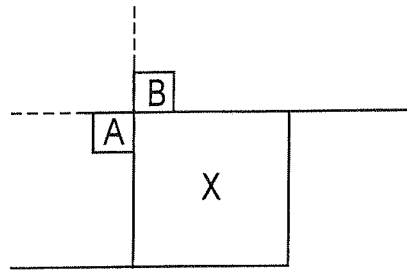


FIG. 9A

	dx	dy
a	-1	0
b	0	-1

FIG. 9B

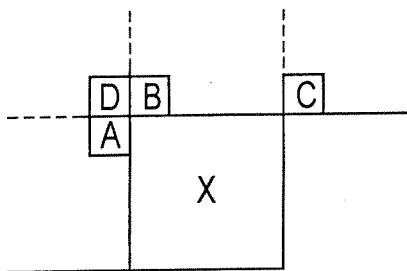


FIG. 9C

	dx	dy
a	-1	0
b	0	-1
c	N	-1
d	-1	-1

FIG. 9D

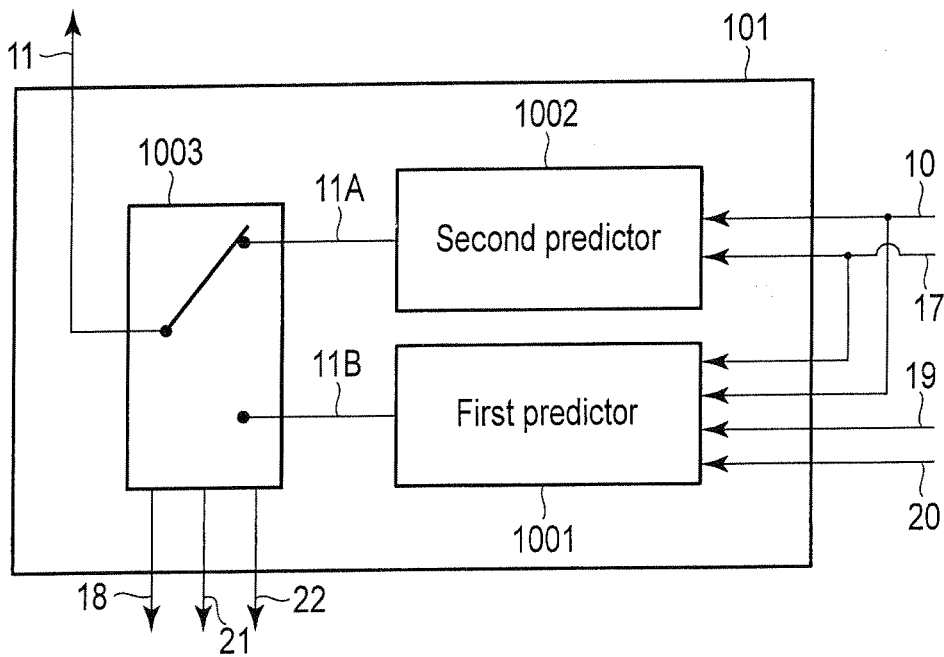


FIG. 10

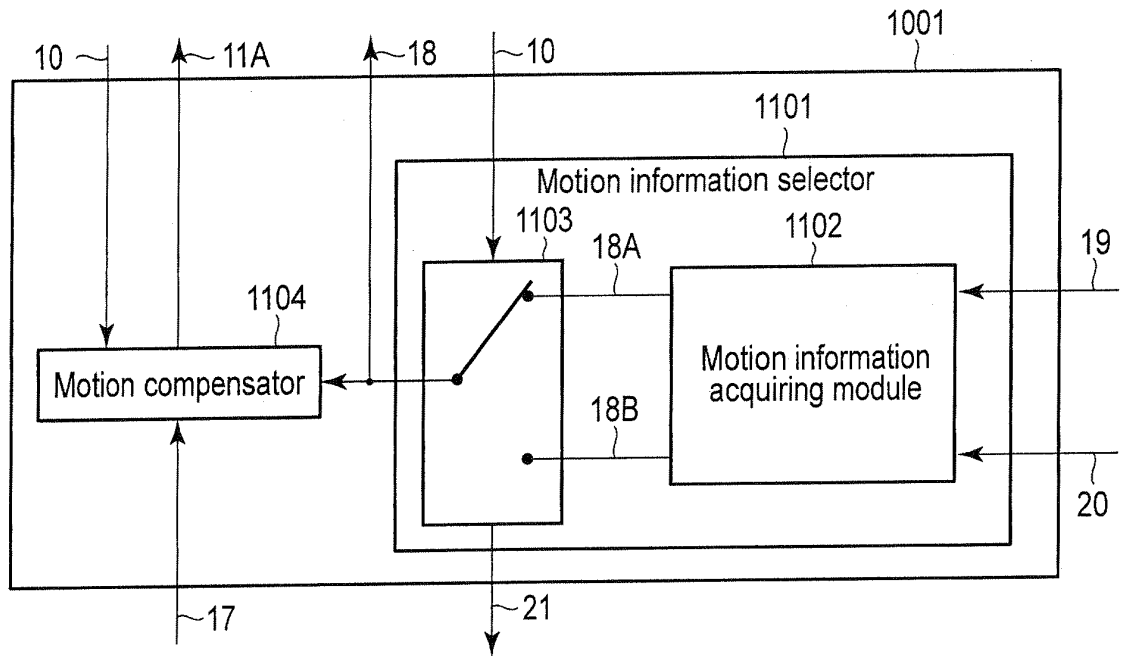


FIG. 11

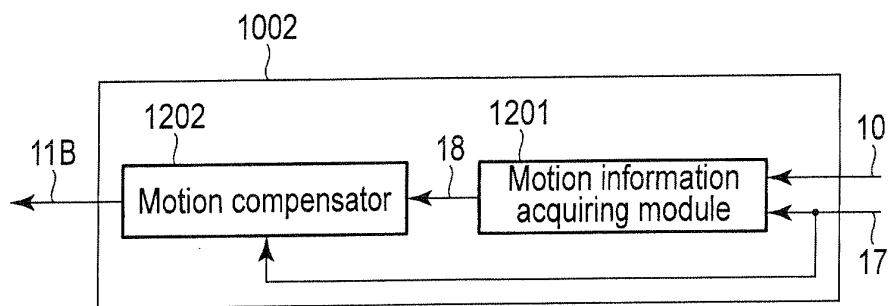


FIG. 12

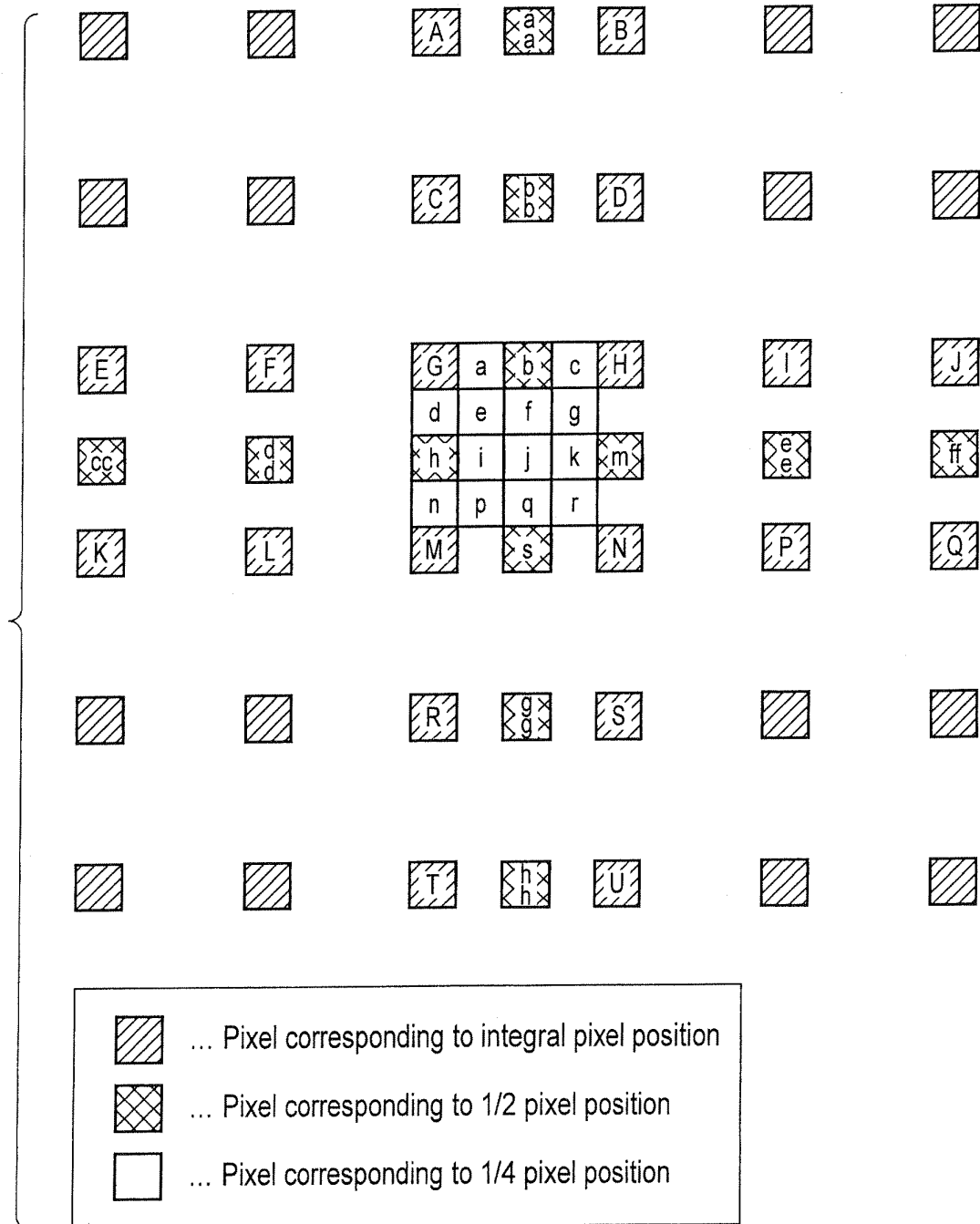
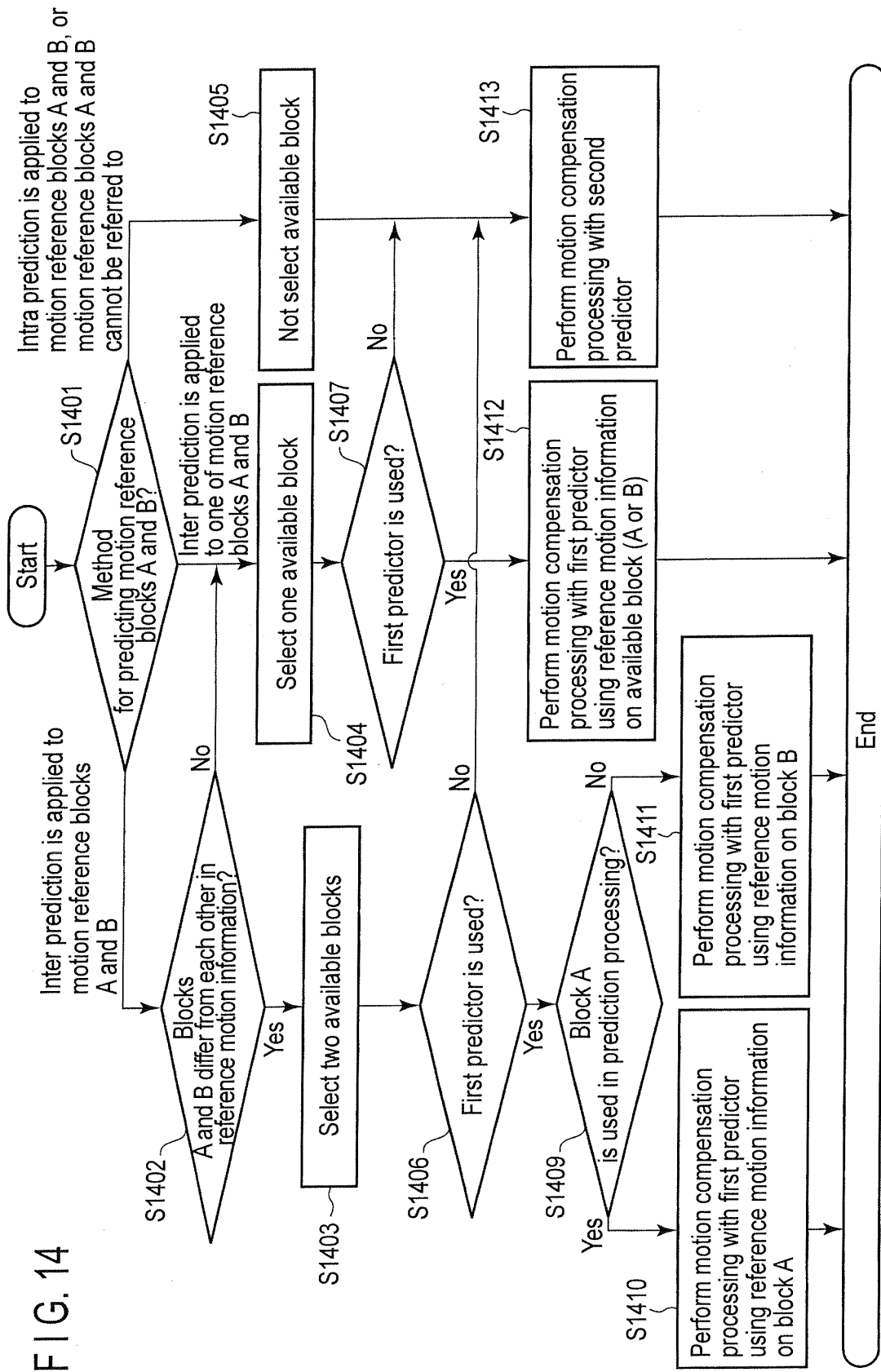


FIG. 13



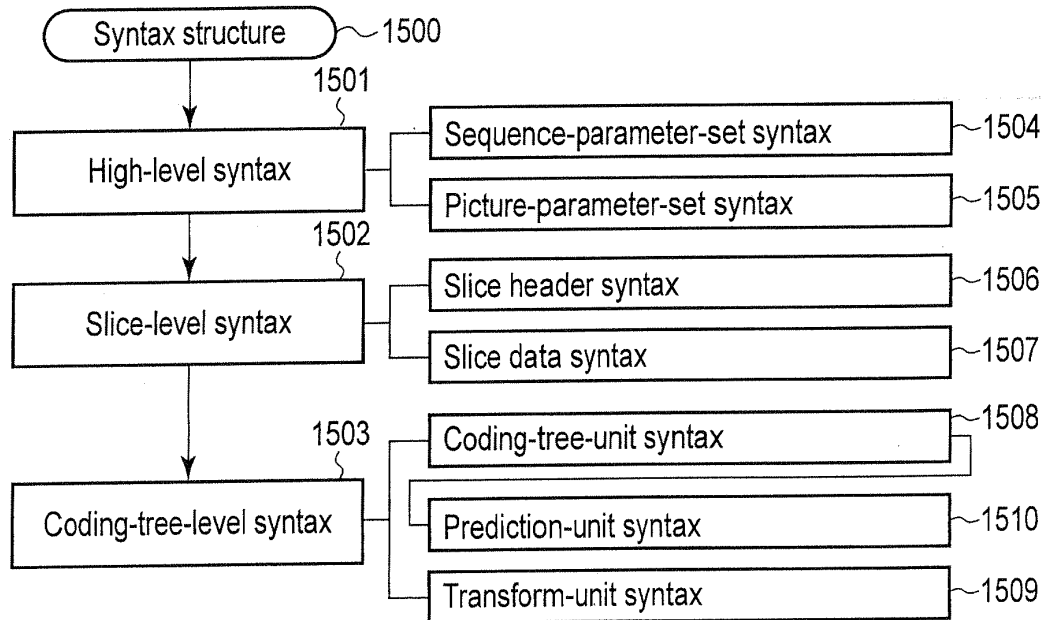


FIG. 15

seq_parameter_set_rbsp () {
...
inter_merge_flag
...
}

FIG. 16

slice_header() {
...
slice_inter_merge_flag
...
}

FIG. 17

coding_unit(x0, y0, currCodingUnitSize) {
if(x0 + currCodingUnitSize < PicWidthInSamplesL &&
y0 + currCodingUnitSize < PicHeightInSamplesL &&
currCodingUnitSize > MinCodingUnitSize)
split_coding_unit_flag
if(split_coding_unit_flag) {
splitCodingUnitSize = currCodingUnitSize >> 1
...
coding_unit(x0, y0, splitCodingUnitSize)
...
coding_unit(x1, y0, splitCodingUnitSize)
...
coding_unit(x0, y1, splitCodingUnitSize)
...
coding_unit(x1, y1, splitCodingUnitSize)
} else {
if(NumMergeCandidates > 0) {
merge_flag
if(merge_flag && NumMergeCandidates > 1)
merge_left_flag
}
}
if(!merge_flag) {
prediction_unit(x0, y0, currCodingUnitSize)
}
if(PredMode != MODE_SKIP) {
...
}
}
}

FIG. 18

```

prediction_unit( x0, y0, currPredUnitSize ) {
    ...
    if( slice_type != I && NumMergeCandidates > 0 ) {
        merge_flag
        if( merge_flag && NumMergeCandidates > 1 )
            merge_left_flag
    }
}
if( !merge_flag ) {
    ...
    pred_mode
    ...
    if( PredMode == MODE_INTRA ) {
        ...
    }
    else if( PredMode == MODE_INTER ) {
        ...
        inter_partitioning_idc
        for( i = 0 ; i < NumPuParts( inter_partitioning_idc ); i++ ) {
            ...
            ref_idx_10[i]
            ...
            ref_idx_11[i]
            ...
            mvd_10[i][0]
            ...
            mvd_11[i][1]
            ...
        }
    }
    ...
}
}

```

FIG. 19

coding_unit(x0, y0, currCodingUnitSize) {
if(x0 + currCodingUnitSize < PicWidthInSamplesL &&
y0 + rCodingUnitSize < PicHeightInSamplesL &&
currCodingUnitSize > MinCodingUnitSize)
split_coding_unit_flag
if(split_coding_unit_flag) {
split_CodingUnitSize = currCodingUnitSize >> 1
...
coding_unit(x0, y0, splitCodingUnitSize)
...
coding_unit(x1, y0, splitCodingUnitSize)
...
coding_unit(x0, y1, splitCodingUnitSize)
...
coding_unit(x1, y1, splitCodingUnitSize)
} else {
if(merge_available_size(currCodingUnitSize))
if(NumMergeCandidates > 0) {
merge_flag
if(merge_flag && NumMergeCandidates > 1)
merge_left_flag
}
}
}
if(!merge_flag) {
prediction_unit(x0, y0, currCodingUnitSize)
}
if(PredMode != MODE_SKIP) {
...
}
}

FIG. 20

prediction_unit(x0, y0, currPredUnitSize) {
if(slice_type != I)
skip_flag
if(skip_flag) {
...
}
else {
...
pred_mode
...
if(PredMode == MODE_INTRA) {
...
}
else if(PredMode == MODE_INTER) {
...
inter_partitioning_idc
for(i = 0 ; i < NumPuParts(inter_partitioning_idc); i++) {
if(NumMergeCandidates > 0) {
merge_flag[i]
if(merge_flag[i] && NumMergeCandidates > 1)
merge_left_flag[i]
}
if(!merge_flag[i]) {
...
ref_idx_10[i]
...
ref_idx_11[i]
...
mvd_10[i][0]
...
mvd_11[i][1]
...
}
}
}
}
...
}
}

FIG. 21

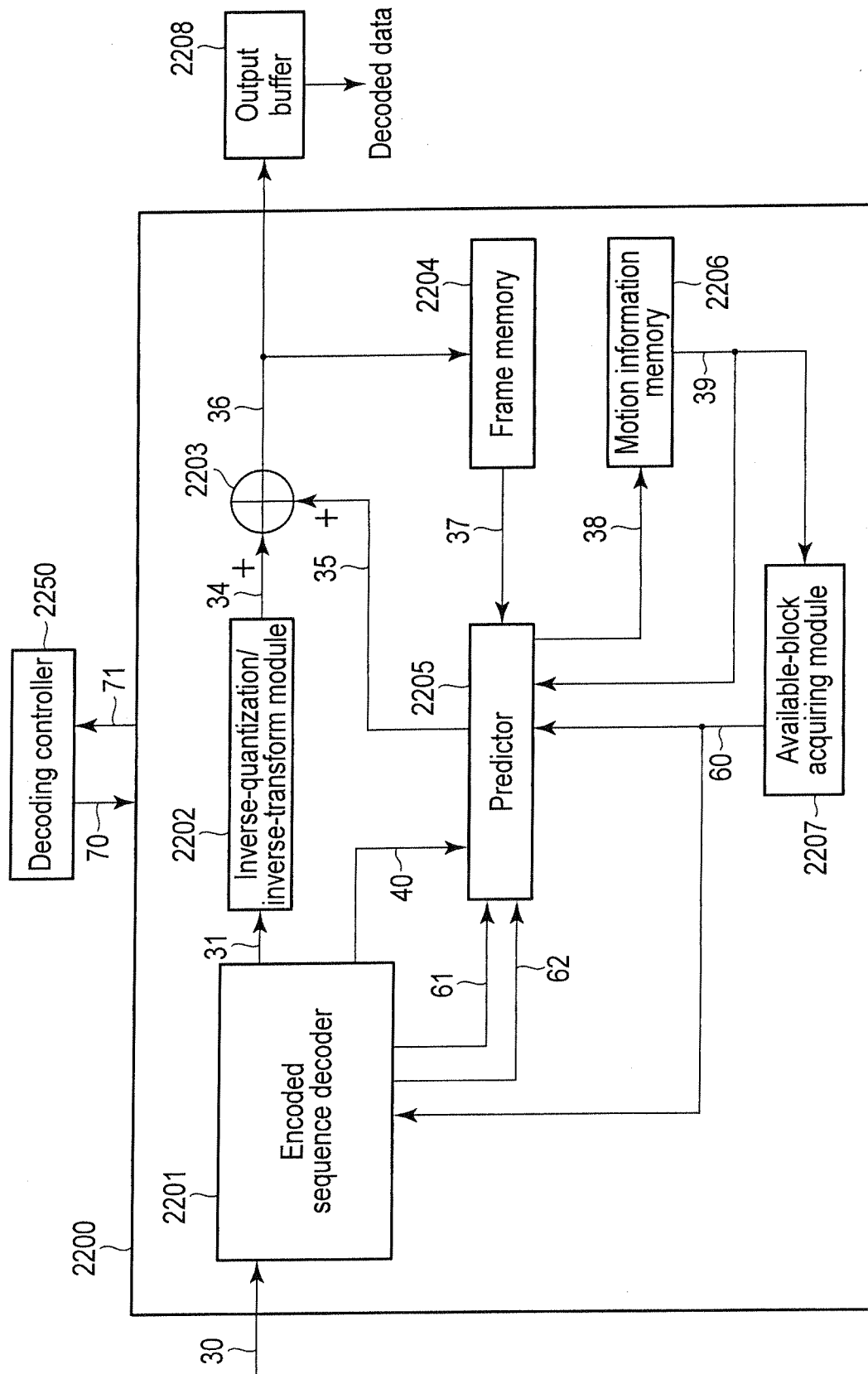


FIG. 22

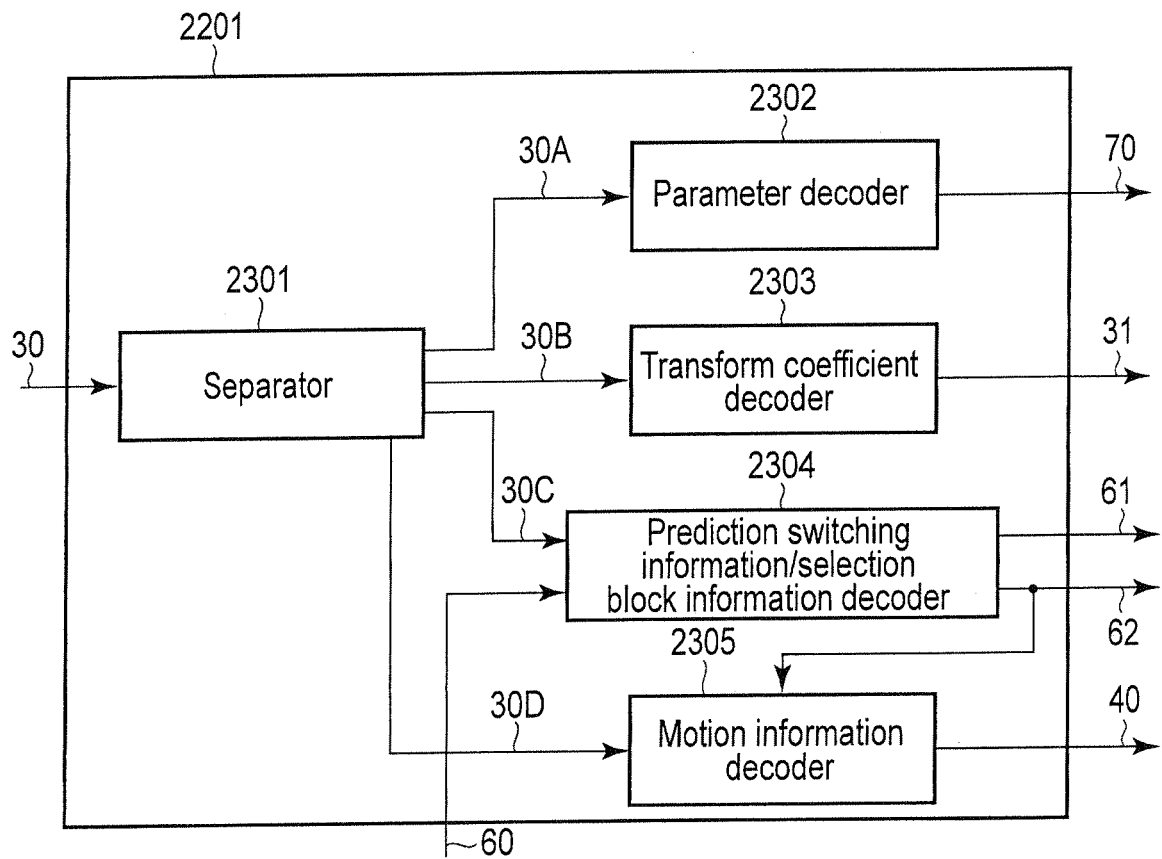


FIG. 23

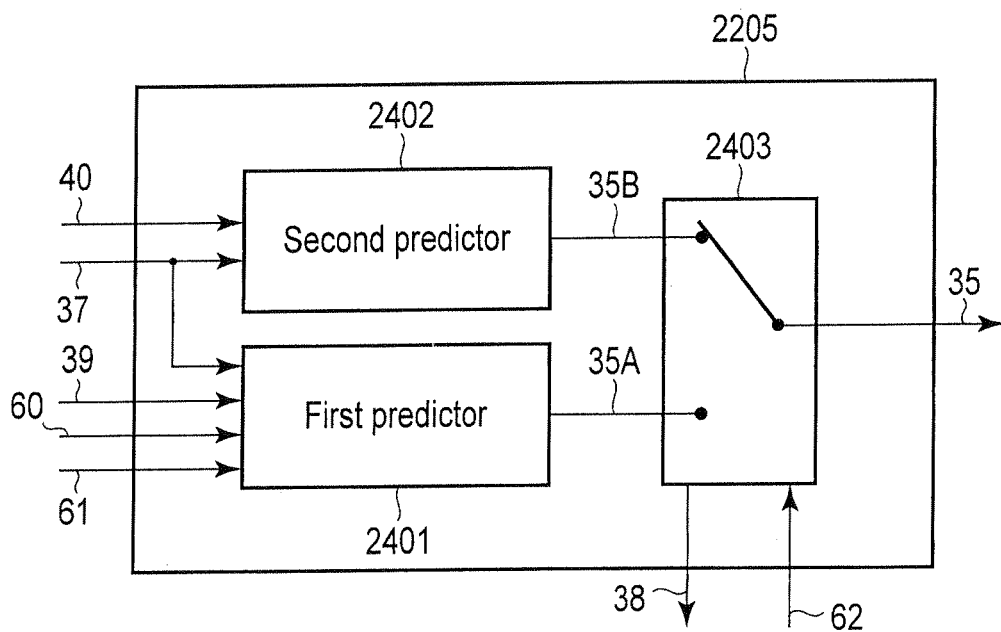


FIG. 24

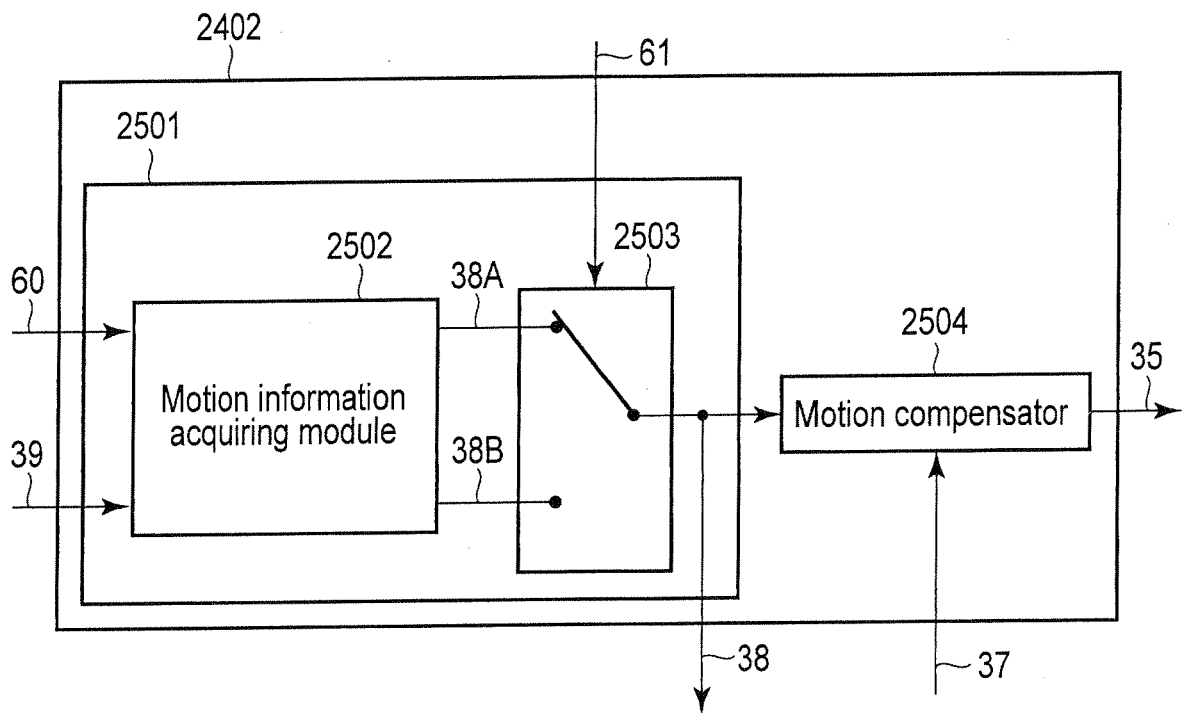


FIG. 25

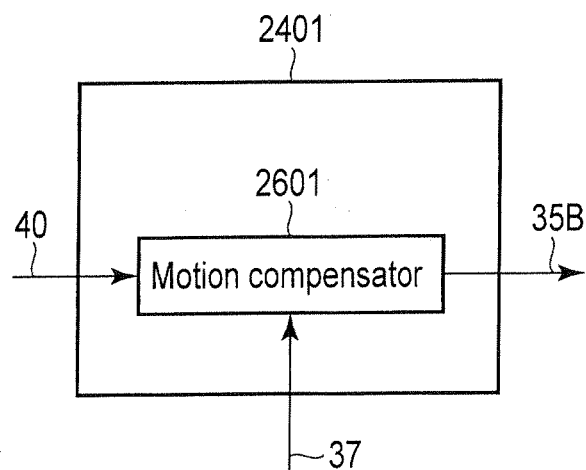


FIG. 26