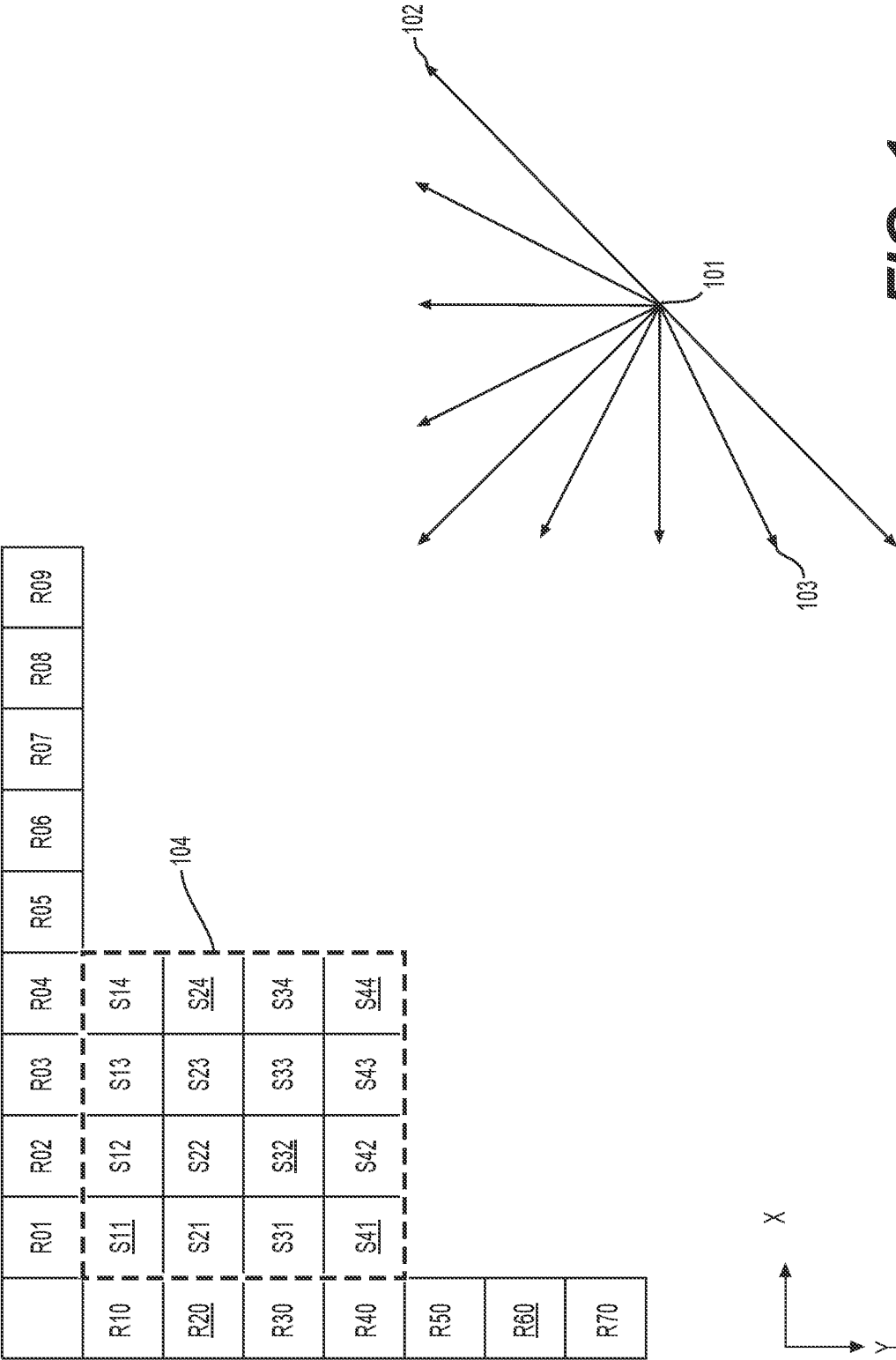




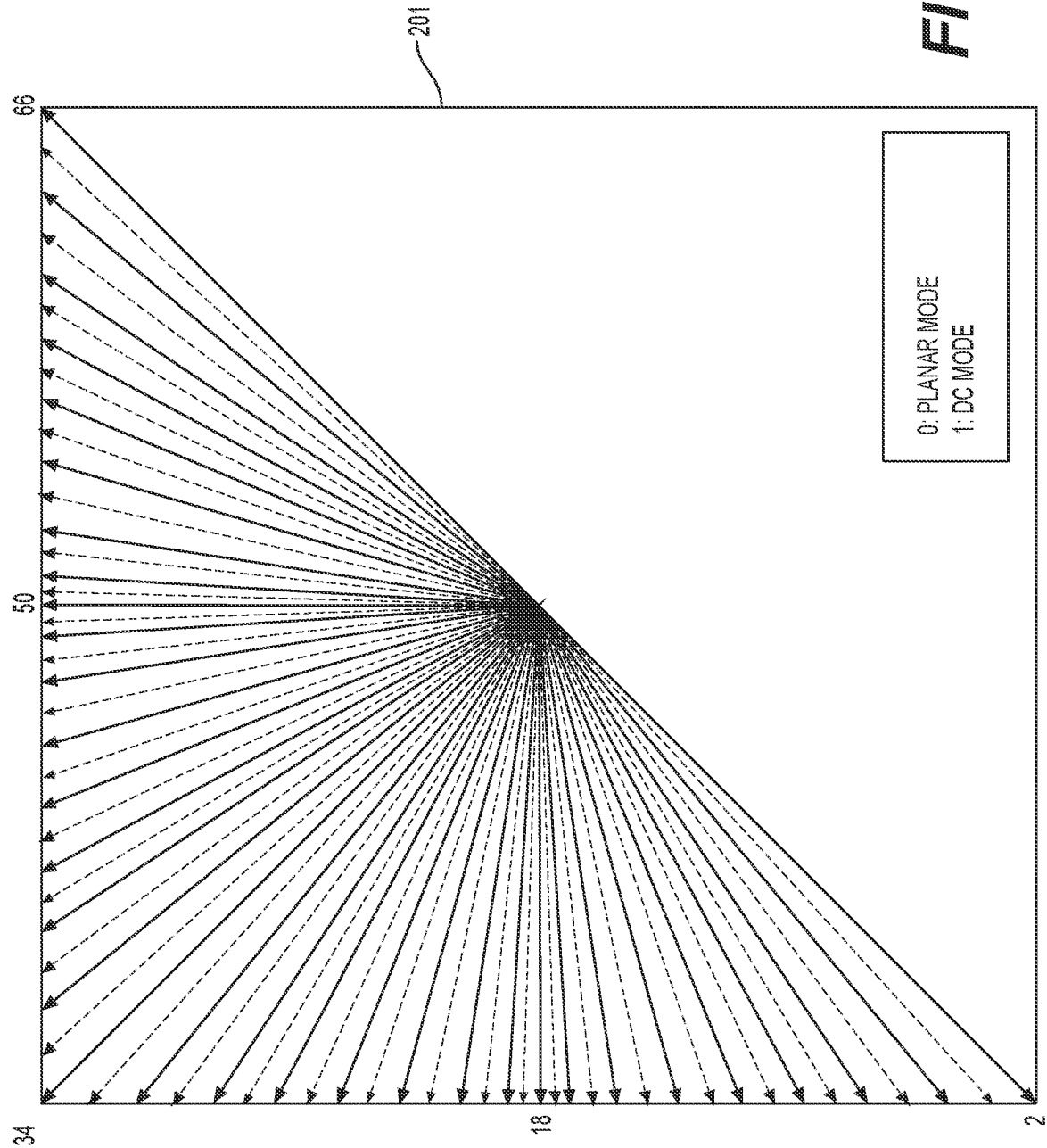


FIG. 2

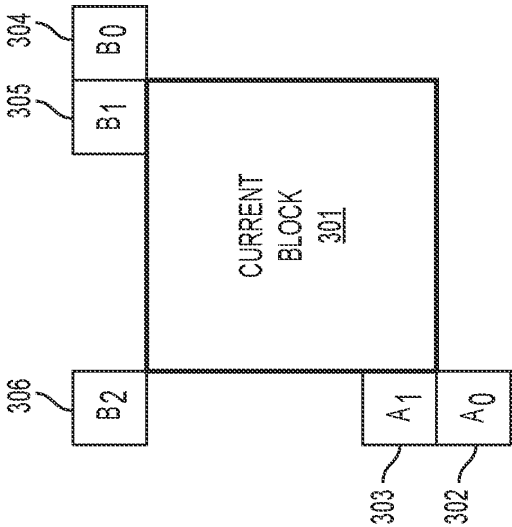


FIG. 3

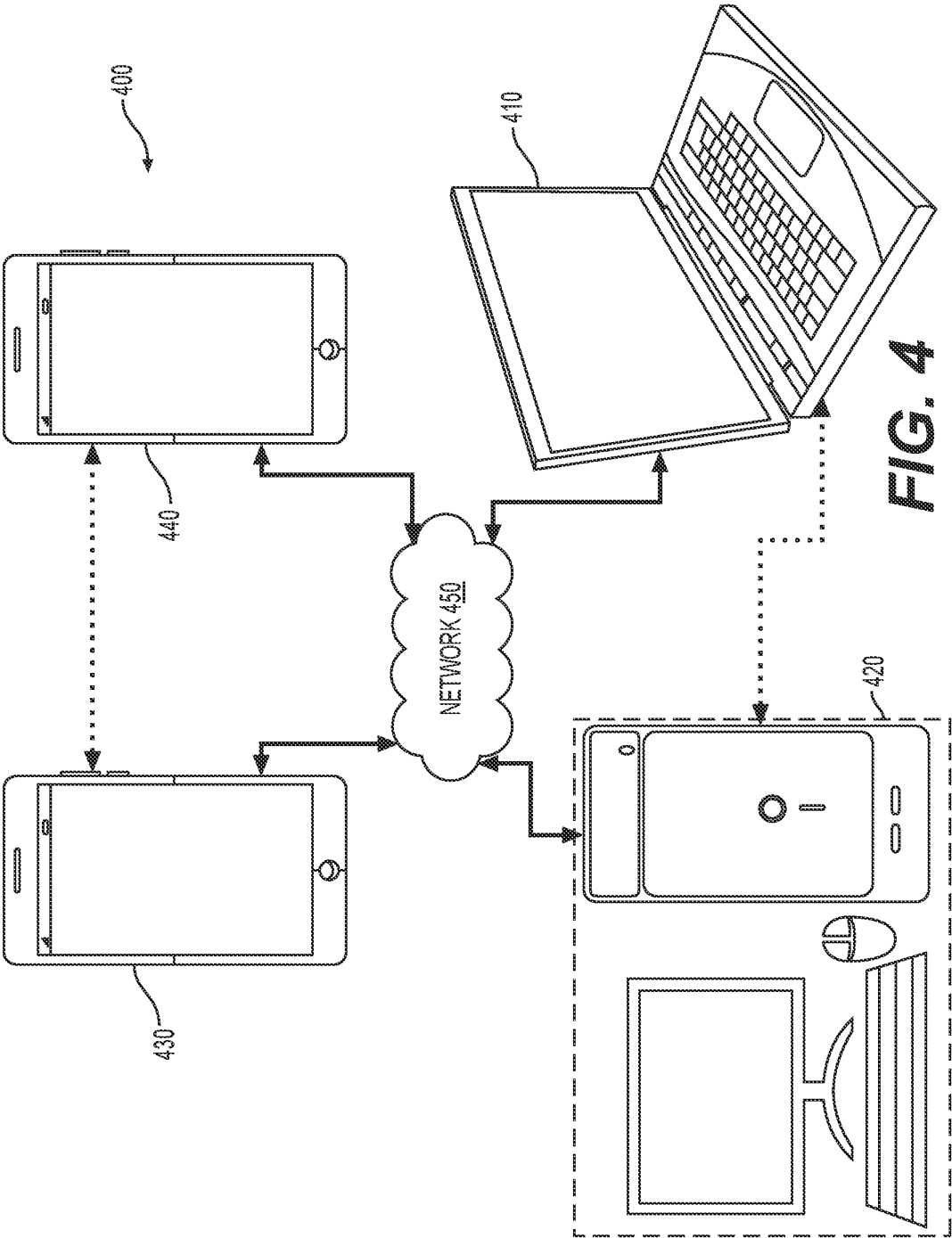


FIG. 4

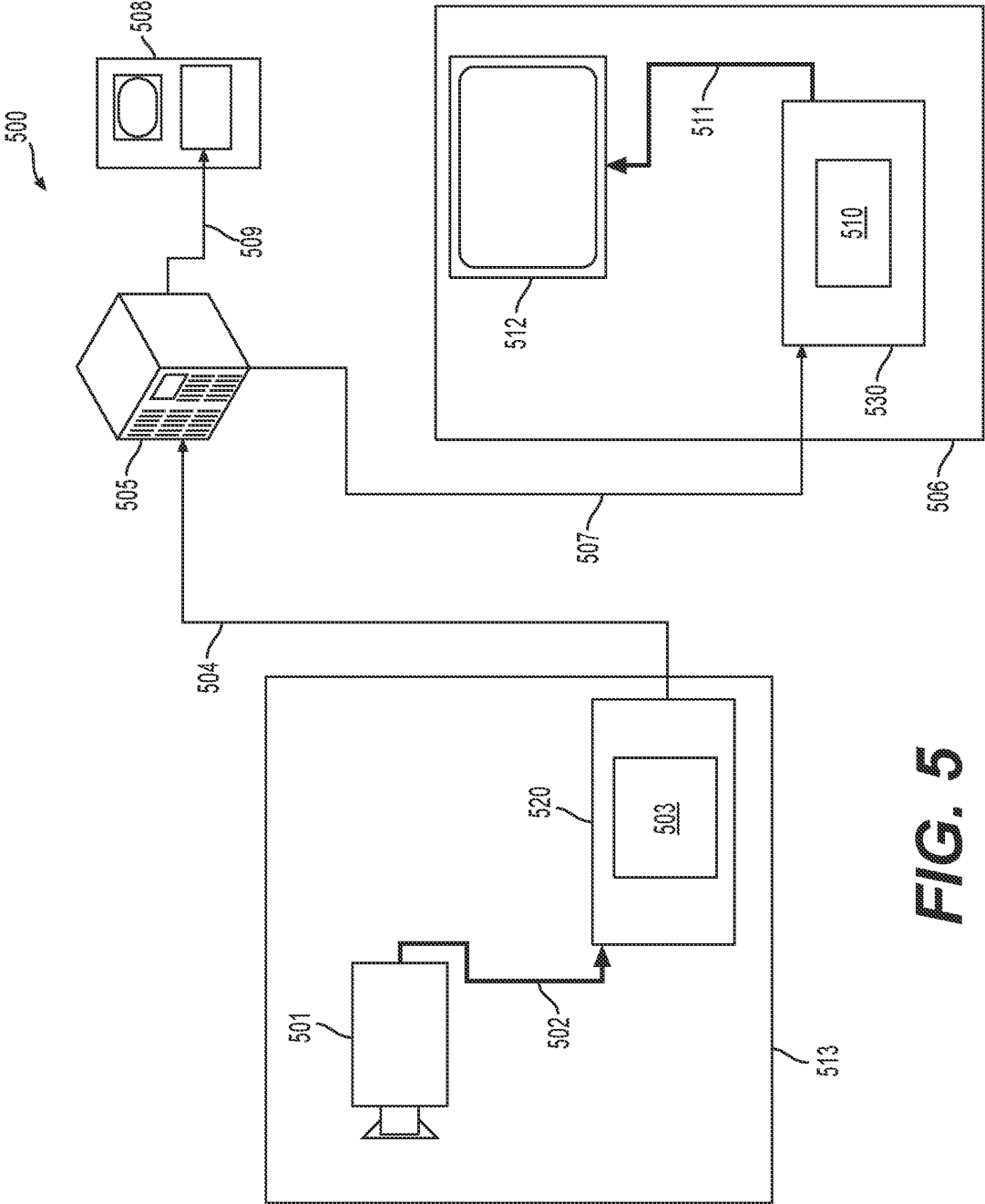


FIG. 5

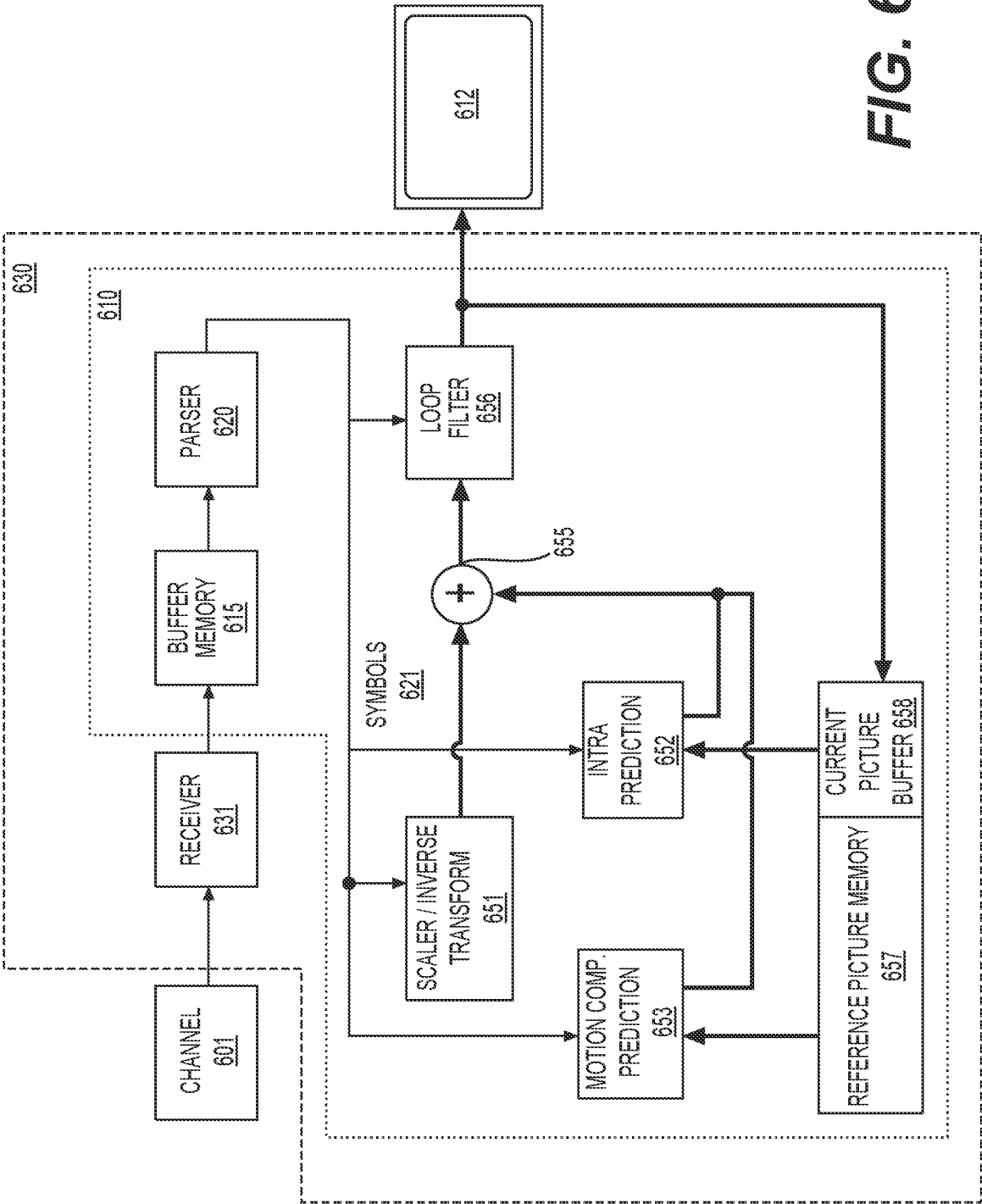


FIG. 6

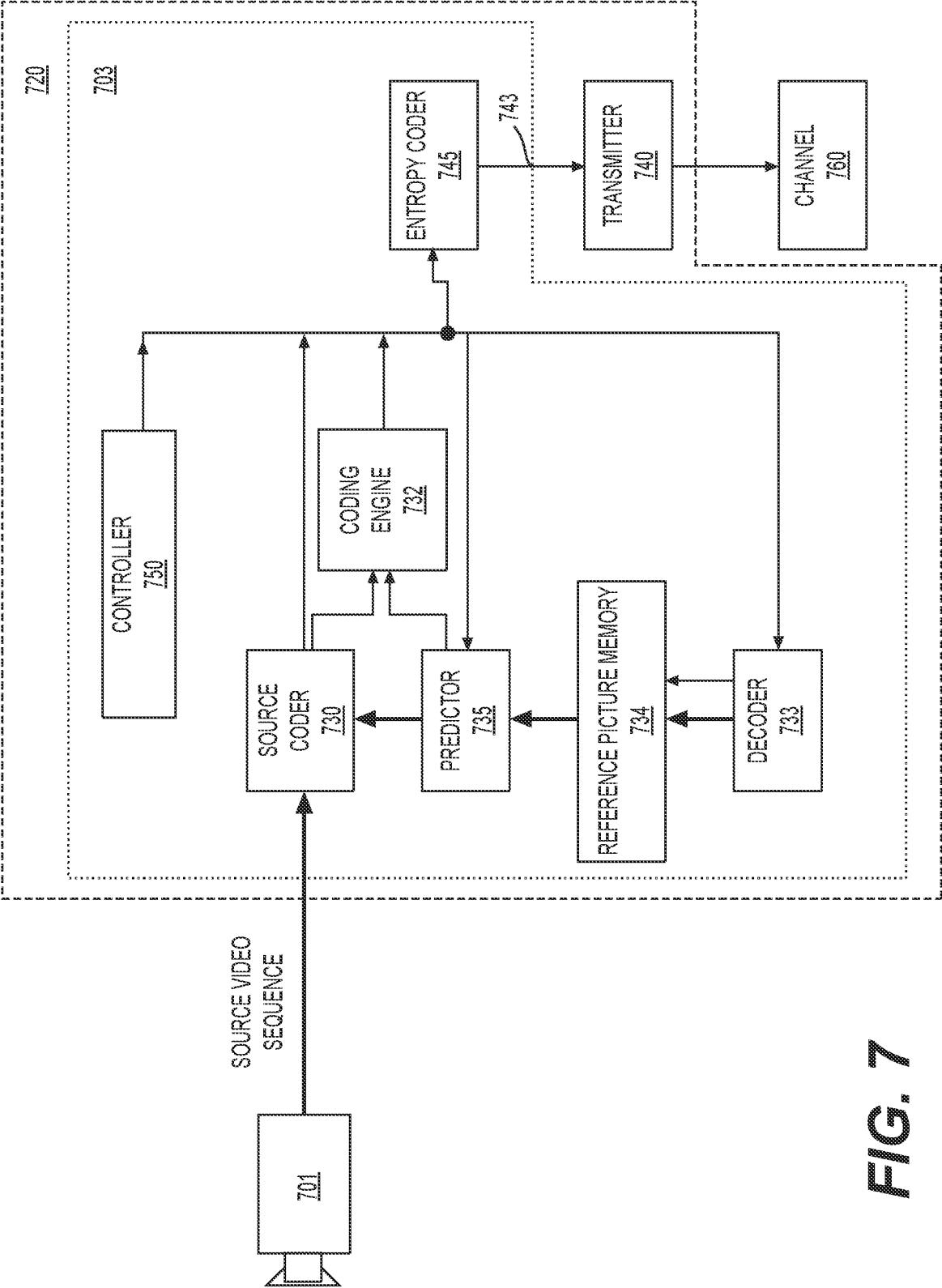
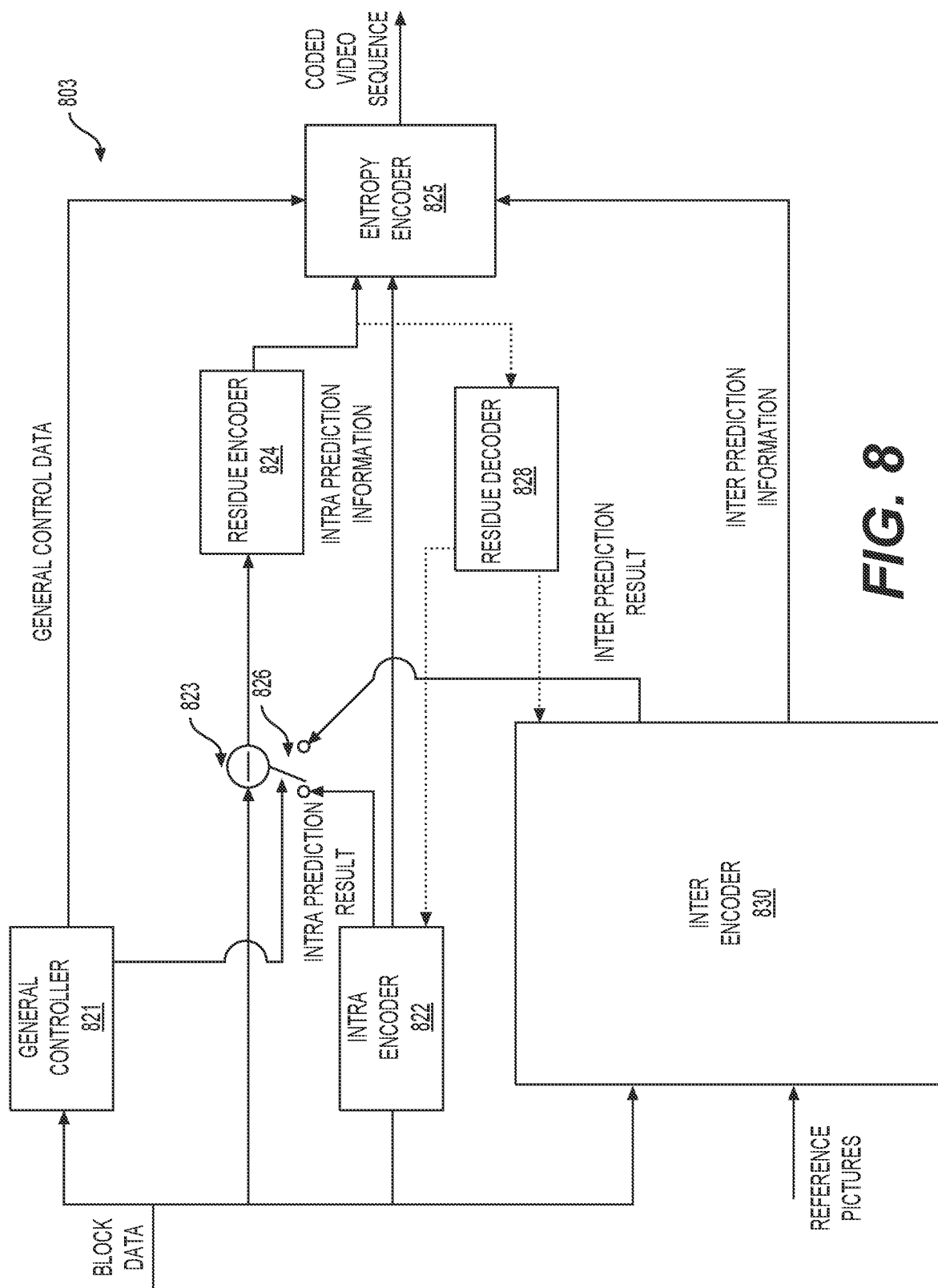


FIG. 7



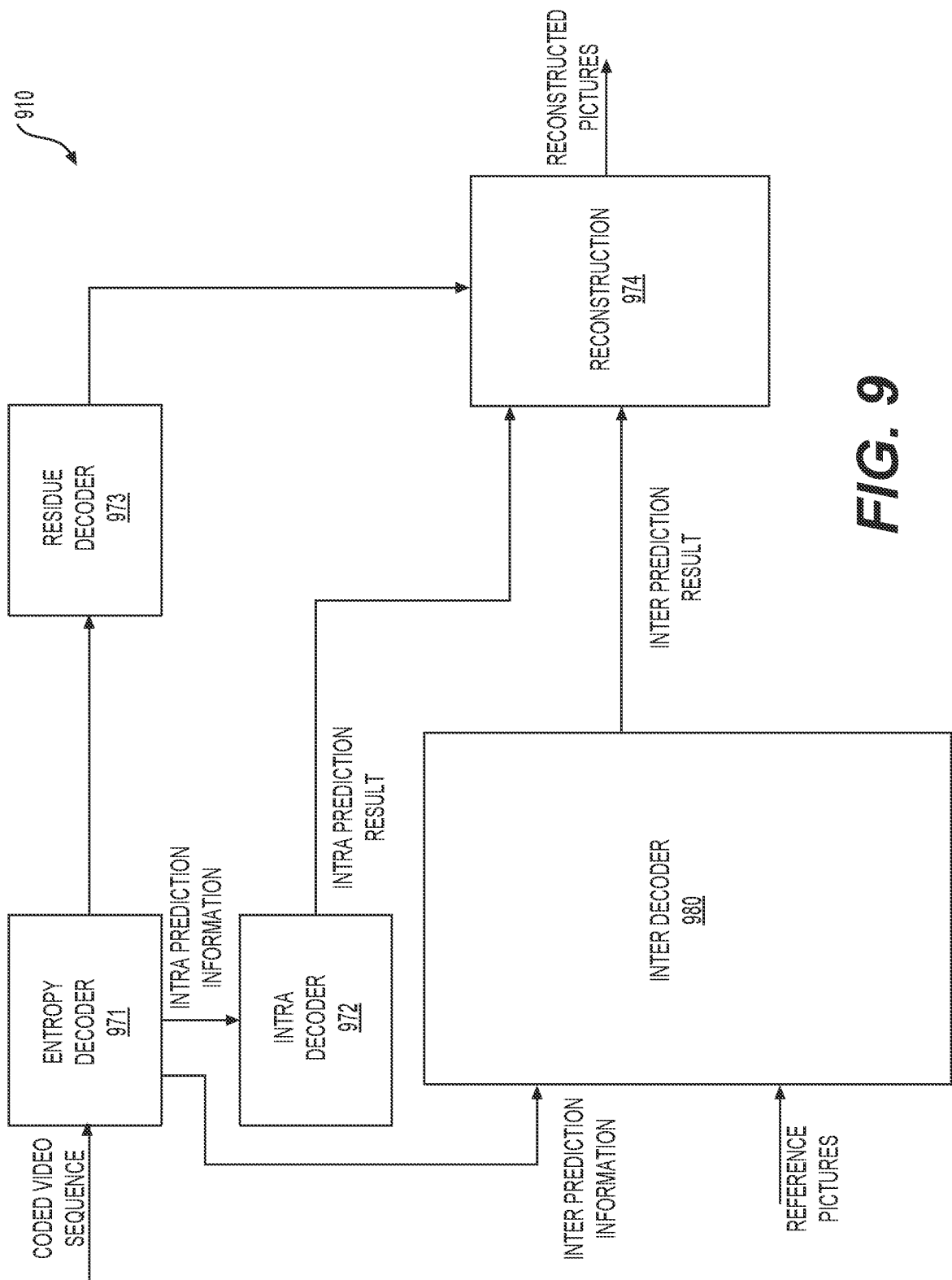


FIG. 9

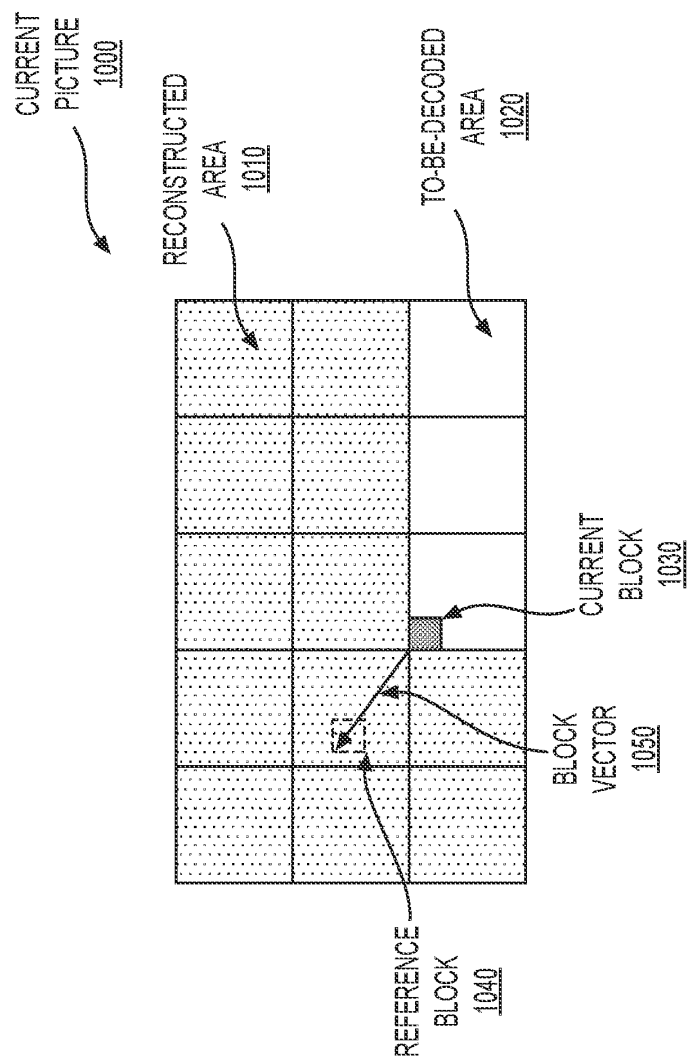


FIG. 10

1100		Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {		
chType = treeType == DUAL_TREE_CHROMA ? 1 : 0		
if(slice_type != I sps_ibc_enabled_flag sps_palette_enabled_flag) {		
-- if(treeType != DUAL_TREE_CHROMA) &&		
!(((cbWidth == 4 && cbHeight == 4) modeType ==		1110
MODE_TYPE_INTRA)		
&& !sps_ibc_enabled_flag))		
cu_skip_flag[x0][y0]		1101
if(cu_skip_flag[x0][y0] == 0 && slice_type != I		
&& !((cbWidth == 4 && cbHeight == 4) && modeType ==		
MODE_TYPE_ALL)		
pred_mode_flag		1102
if(((slice_type == I && cu_skip_flag[x0][y0] == 0)		
(slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA		
(cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] ==		
0))) &&		
cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER		
&&		
sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)		
pred_mode_ibc_flag		1103
		ae(v)

FIG. 11

1200

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr) {	Descriptor
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor allowSplitQT) &&(x0 + cbWidth <= pic_width_in_luma_samples) &&(y0 + cbHeight <= pic_height_in_luma_samples))	
split_cu_flag	ae(v)
if(cu_qp_delta_enabled_flag && qgOnY && cbSubdiv <= CuQpDeltaSubdiv) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
CuQgTopLeftX = x0	
CuQgTopLeftY = y0	
}	
if(cu_chroma_qp_offset_enabled_flag && qgOnC && cbSubdiv <= CuChromaQpOffsetSubdiv)	
IsCuChromaQpOffsetCoded = 0	
if(split_cu_flag) {	
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && allowSplitQT)	
split_qt_flag	ae(v)
if(!split_qt_flag) {	
if((allowSplitBtHor allowSplitTtHor) && (allowSplitBtVer allowSplitTtVer))	
mtt_split_cu_vertical_flag	ae(v)
if((allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag) (allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag))	
mtt_split_cu_binary_flag	ae(v)
}	
if(modeTypeCondition == 1)	
modeType = MODE_TYPE_INTRA	

FIG. 12A

Continue from FIG. 12A

1200

else if(modeTypeCondition == 2) {	
mode_constraint_flag	ae(v)
modeType = mode_constraint_flag ? MODE_TYPE_INTRA : MODE_TYPE_INTER	
} else {	
modeType = modeTypeCurr	
}	
treeType = (modeType == MODE_TYPE_INTRA) ? DUAL_TREE_LUMA : treeTypeCurr	
if(!split_qt_flag) {	
if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {	
depthOffset += (x0 + cbWidth > pic_width_in_luma_samples) ? 1 : 0	
x1 = x0 + (cbWidth / 2)	
coding_tree(x0, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC, cbSubdiv + 1,	
cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
if(x1 < pic_width_in_luma_samples)	
coding_tree(x1, y0, cbWidth / 2, cbHeightY, qgOnY, qgOnC, cbSubdiv + 1,	
cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	

FIG. 12B

Continue from FIG. 12B

1200

<pre> } else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) { </pre>	
<pre> depthOffset += (y0 + cbHeight > pic_height_in_luma_samples) ? 1 : 0 </pre>	
<pre> y1 = y0 + (cbHeight / 2) </pre>	
<pre> coding_tree(x0, y0, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, </pre>	
<pre> cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType) </pre>	
<pre> if(y1 < pic_height_in_luma_samples) </pre>	
<pre> coding_tree(x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, </pre>	
<pre> cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType) </pre>	
<pre> } else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER) { </pre>	
<pre> x1 = x0 + (cbWidth / 4) </pre>	
<pre> x2 = x0 + (3 * cbWidth / 4) </pre>	
<pre> qgOnY = qgOnY && (cbSubdiv + 2 <= CuQpDeltaSubdiv) </pre>	
<pre> qgOnC = qgOnC && (cbSubdiv + 2 <= CuChromaQpOffsetSubdiv) </pre>	
<pre> coding_tree(x0, y0, cbWidth / 4, cbHeight, qgOnY, qgOnC, cbSubdiv + 2, </pre>	
<pre> cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType) </pre>	
<pre> coding_tree(x1, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC, cbSubdiv + 1, </pre>	
<pre> cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType) </pre>	
<pre> coding_tree(x2, y0, cbWidth / 4, cbHeight, qgOnY, qgOnC, cbSubdiv + 2, </pre>	
<pre> cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType) </pre>	

FIG. 12C

Continue from FIG. 12C

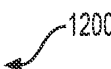
1200

<code>} else { /* SPLIT TT HOR */</code>	
<code> y1 = y0 + (cbHeight / 4)</code>	
<code> y2 = y0 + (3 * cbHeight / 4)</code>	
<code> qgOnY = qgOnY && (cbSubdiv + 2 <= CuQpDeltaSubdiv)</code>	
<code> qgOnC = qgOnC && (cbSubdiv + 2 <=</code> <code>CuChromaQpOffsetSubdiv)</code>	
<code> coding_tree(x0, y0, cbWidth, cbHeight / 4, qgOnY, qgOnC,</code> <code> cbSubdiv + 2,</code>	
<code> cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)</code>	
<code> coding_tree(x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC,</code> <code> cbSubdiv + 1,</code>	
<code> cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)</code>	
<code> coding_tree(x0, y2, cbWidth, cbHeight / 4, qgOnY, qgOnC,</code> <code> cbSubdiv + 2,</code>	
<code> cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)</code>	
<code> }</code>	
<code> } else {</code>	
<code> x1 = x0 + (cbWidth / 2)</code>	
<code> y1 = y0 + (cbHeight / 2)</code>	
<code> coding_tree(x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC,</code> <code> cbSubdiv + 2,</code>	
<code> cqtDepth + 1, 0, 0, 0, treeType, modeType)</code>	
<code> if(x1 < pic_width_in_luma_samples)</code>	
<code> coding_tree(x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC,</code> <code> cbSubdiv + 2,</code>	
<code> cqtDepth + 1, 0, 0, 1, treeType, modeType)</code>	
<code> if(y1 < pic_height_in_luma_samples)</code>	
<code> coding_tree(x0, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC,</code> <code> cbSubdiv + 2,</code>	
<code> cqtDepth + 1, 0, 0, 2, treeType, modeType)</code>	
<code> if(y1 < pic_height_in_luma_samples && x1 <</code> <code> pic_width_in_luma_samples)</code>	

FIG. 12D

Continue from FIG. 12D

1200



coding_tree(x1, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 3, treeType, modeType)	
}	
if(modeTypeCur == MODE_TYPE_ALL && modeType == MODE_TYPE_INTRA) {	
coding_tree(x0, y0, cbWidth, cbHeight, 0, qgOnC, cbSubdiv, cqtDepth, mttDepth, 0, 0 DUAL_TREE_CHROMA , modeType)	
}	
} else	
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeTypeCurr , mode TypeCurr)	
}	

FIG. 12E

1300

	Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	
chType = treeType == DUAL_TREE_CHROMA ? 1 : 0	
if(slice_type != I sps_ibc_enabled_flag sps_palette_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && ! ((cbWidth == 4 && cbHeight == 4) modeType == MODE_TYPE_INTRA) && ! sps_ibc_enabled_flag) { && !(slice_type = I && (cbWidth > 64 cbHeight > 64)))	1310
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
pred_mode_flag	ae(v)
if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0)))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)

FIG. 13

Descriptor	
	coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr) {
	if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor allowSplitQT) &&(x0 + cbWidth <= pic_width_in_luma_samples) &&(y0 + cbHeight <= pic_height_in_luma_samples))
ae(v)	split_cu_flag
	if(cu_qp_delta_enabled_flag && qgOnY && cbSubdiv <= CuQpDeltaSubdiv) {
	IsCuQpDeltaCoded = 0
	CuQpDeltaVal = 0
	CuQgTopLeftX = x0
	CuQgTopLeftY = y0
	}
	if(cu_chroma_qp_offset_enabled_flag && qgOnC && cbSubdiv <= CuChromaQpOffsetSubdiv)
	IsCuChromaQpOffsetCoded = 0
	if(split_cu_flag) {
	
	} else {
	if(slice_type == I && (cbWidth > 64 cbHeight > 64))
	modeTypeCurr = MODE_TYPE_INTRA
	coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeTypeCurr, modeTypeCurr)
	}
	}

1400

1410

1400

1410

FIG. 14

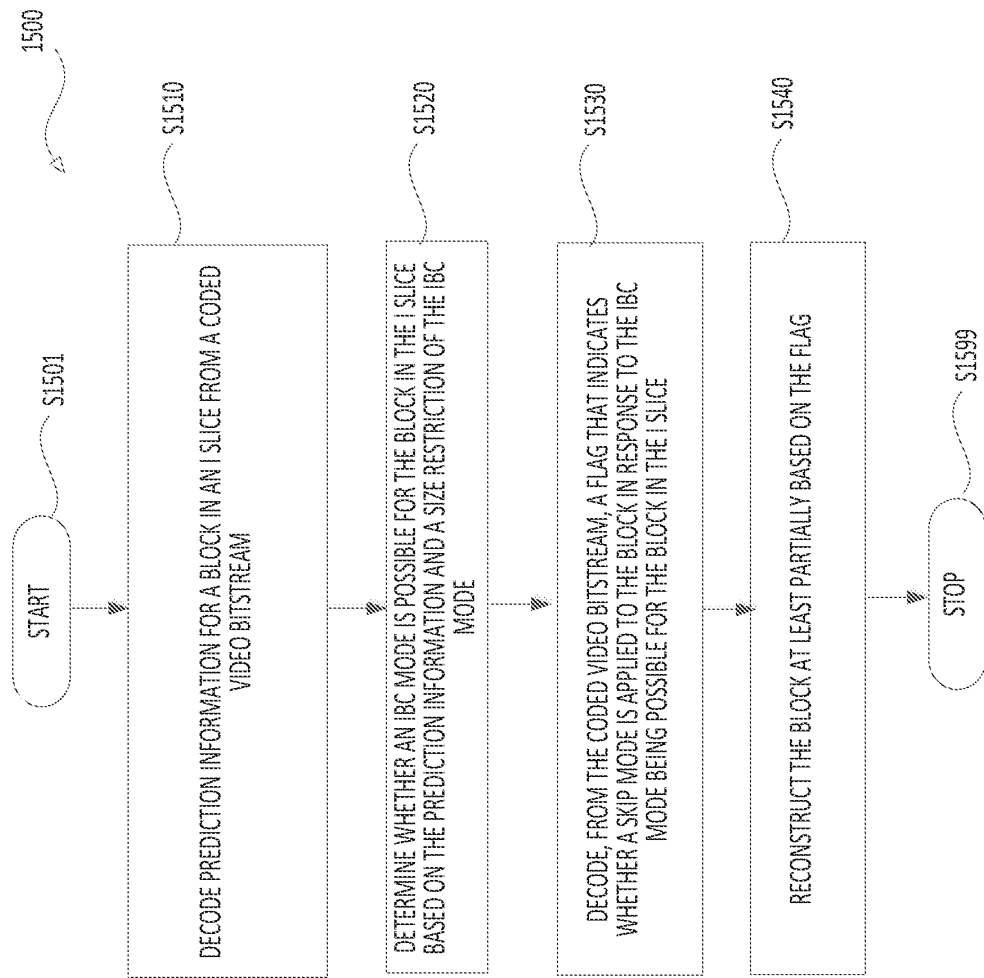


FIG. 15

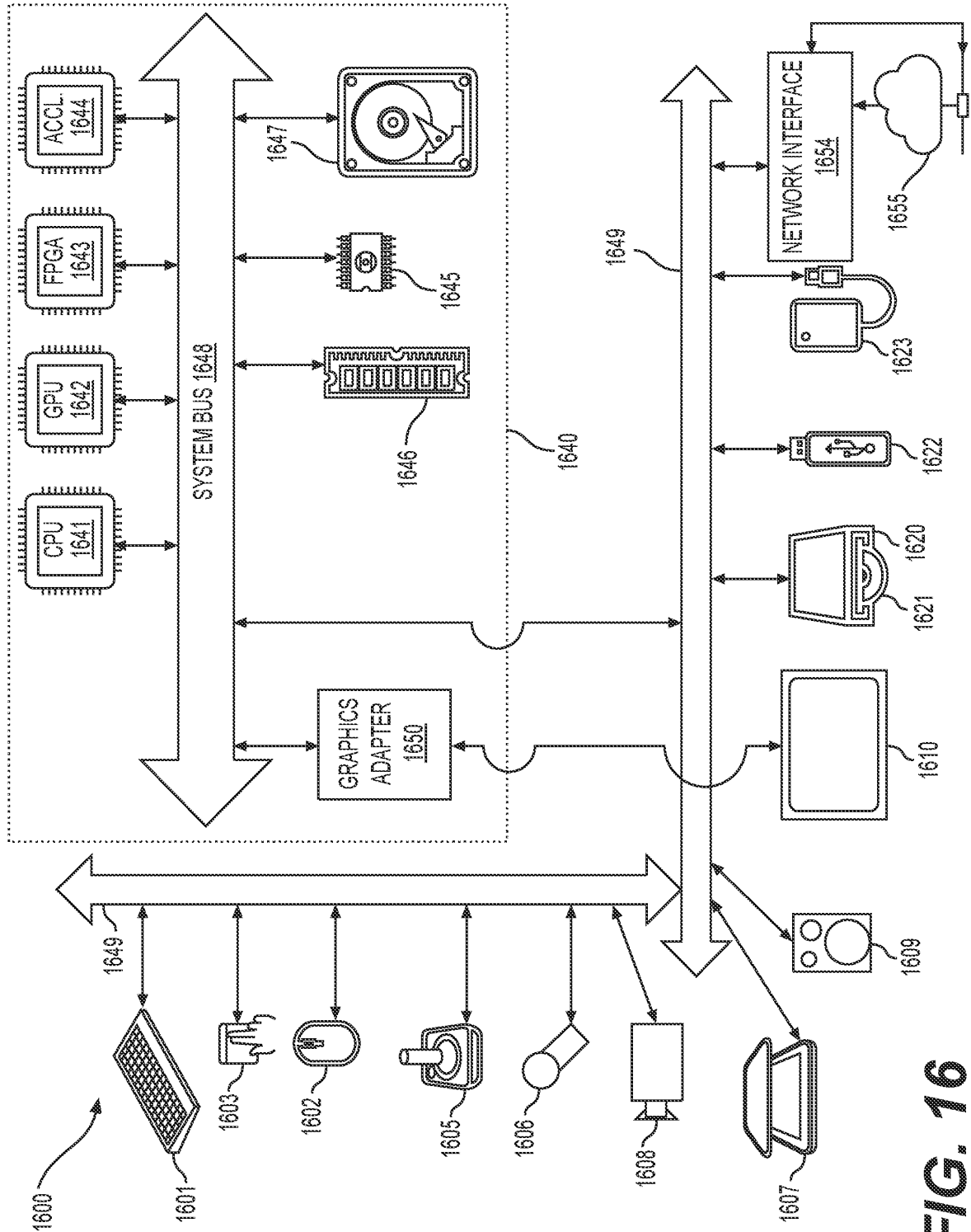


FIG. 16