

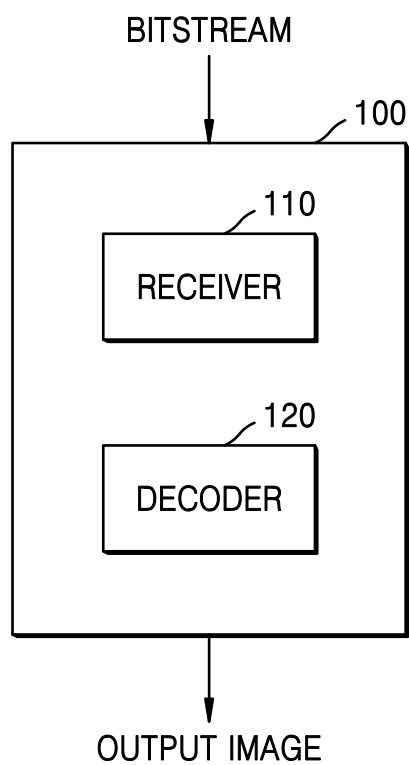
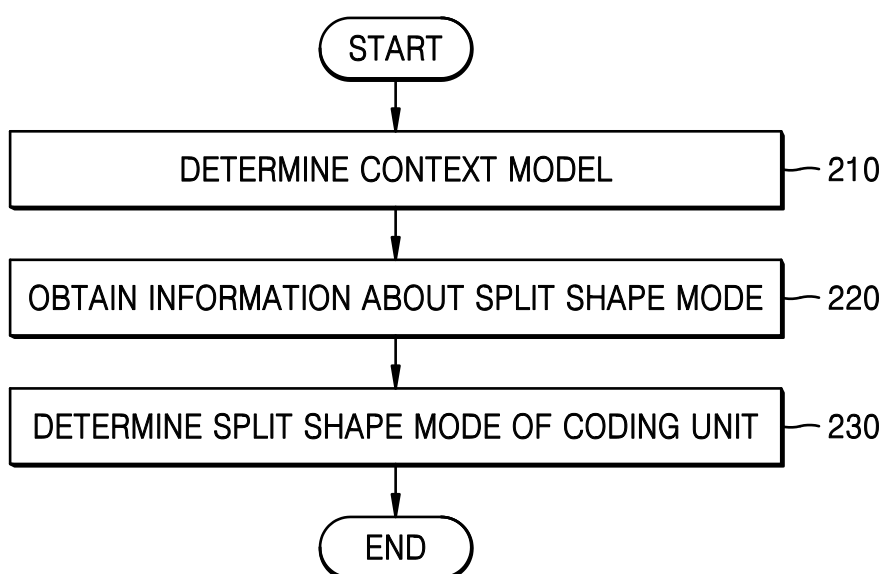
FIG. 1**FIG. 2**

FIG. 3

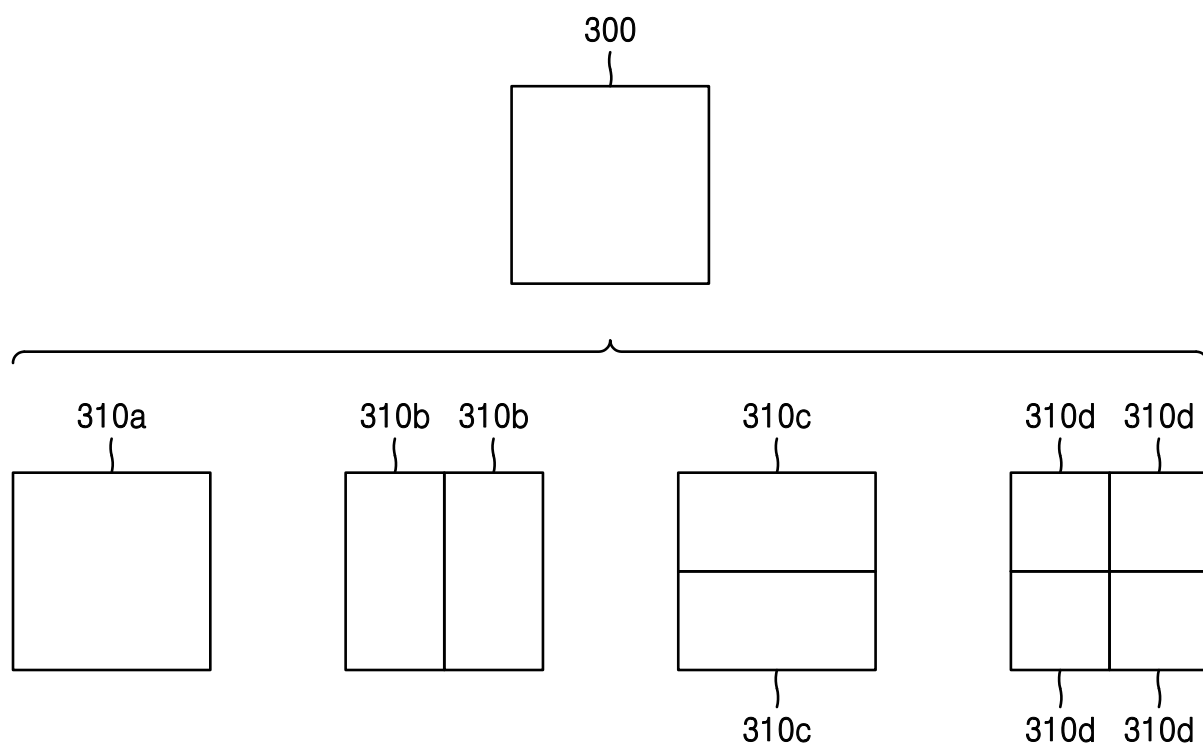


FIG. 4

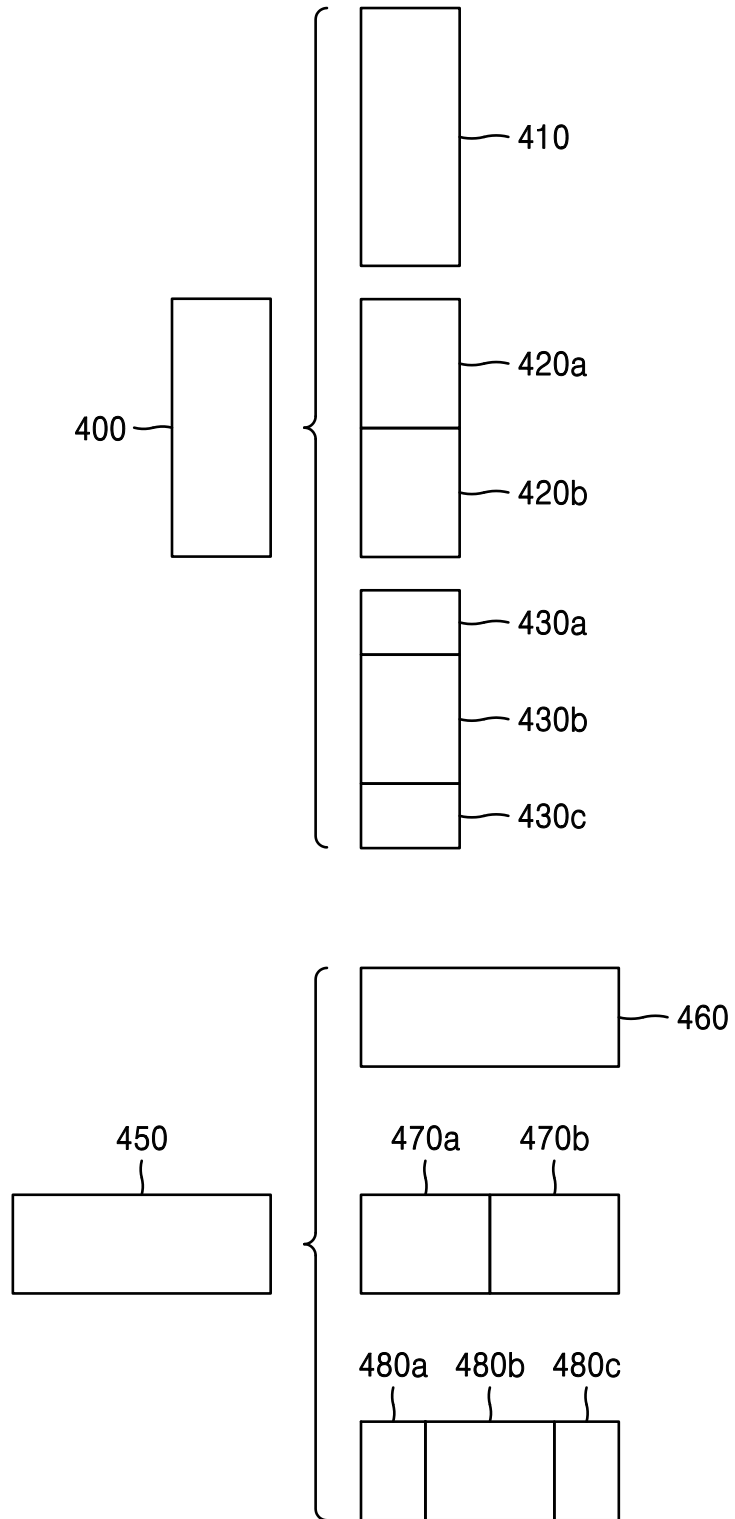


FIG. 5

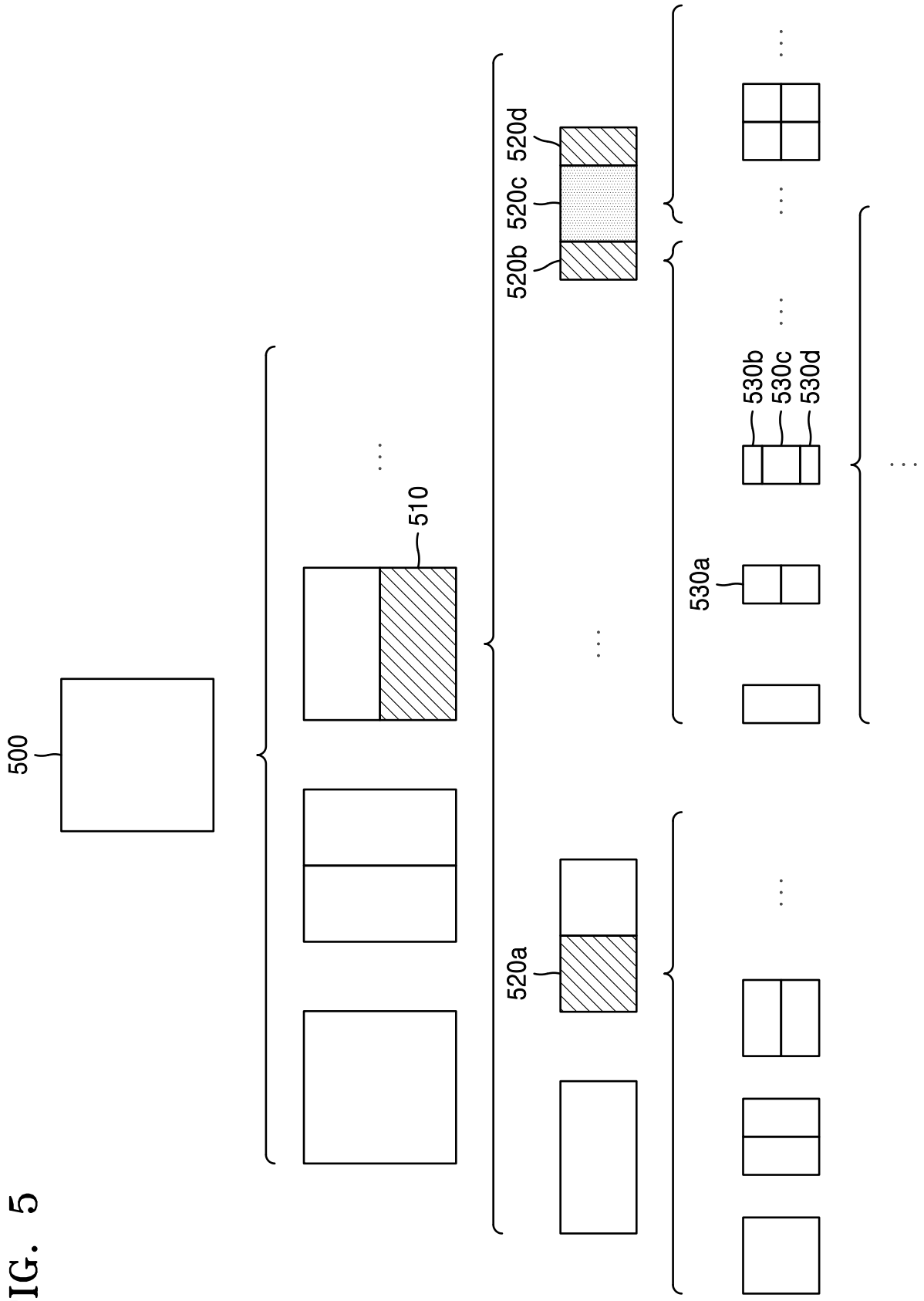


FIG. 6

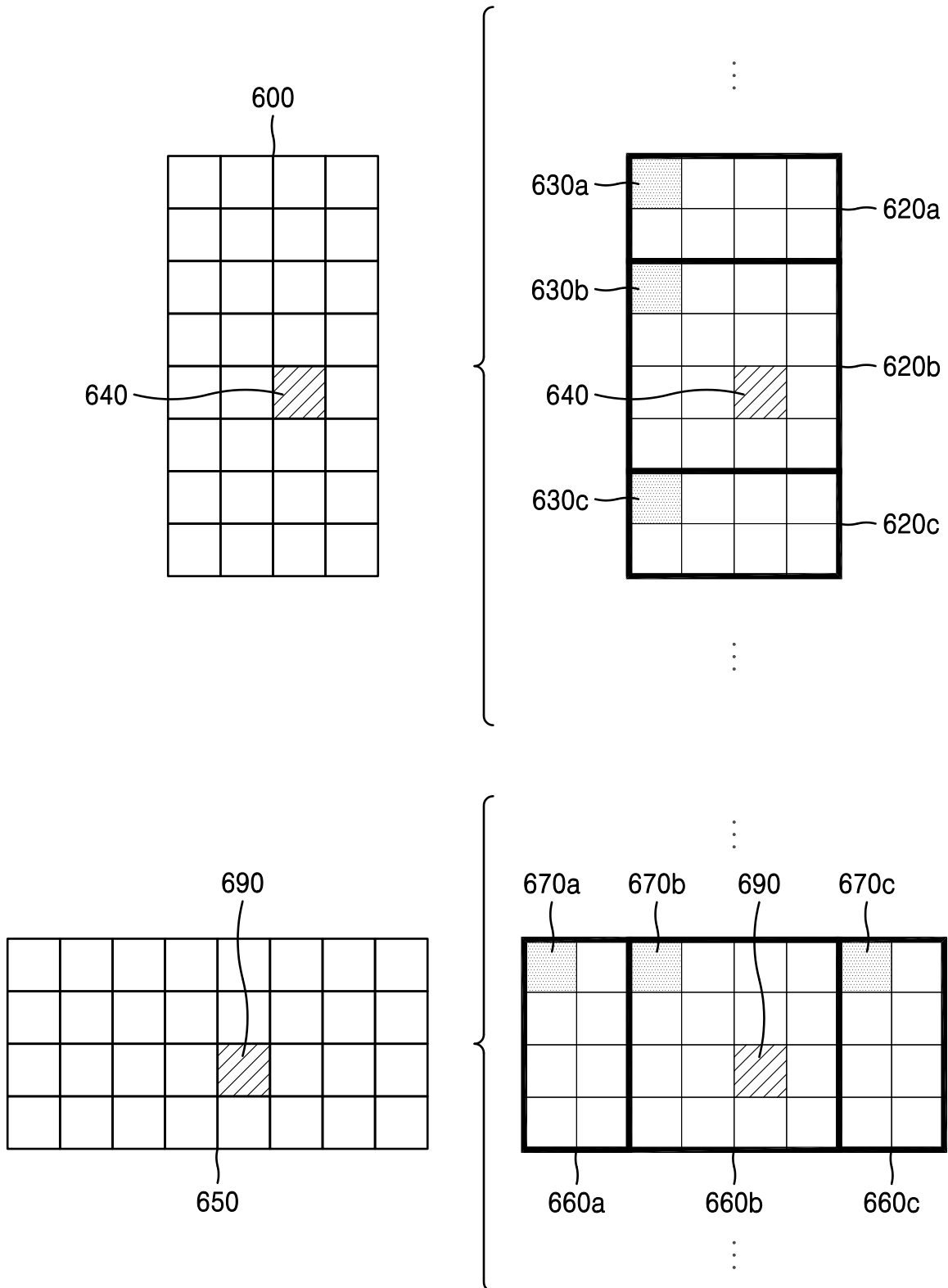


FIG. 7

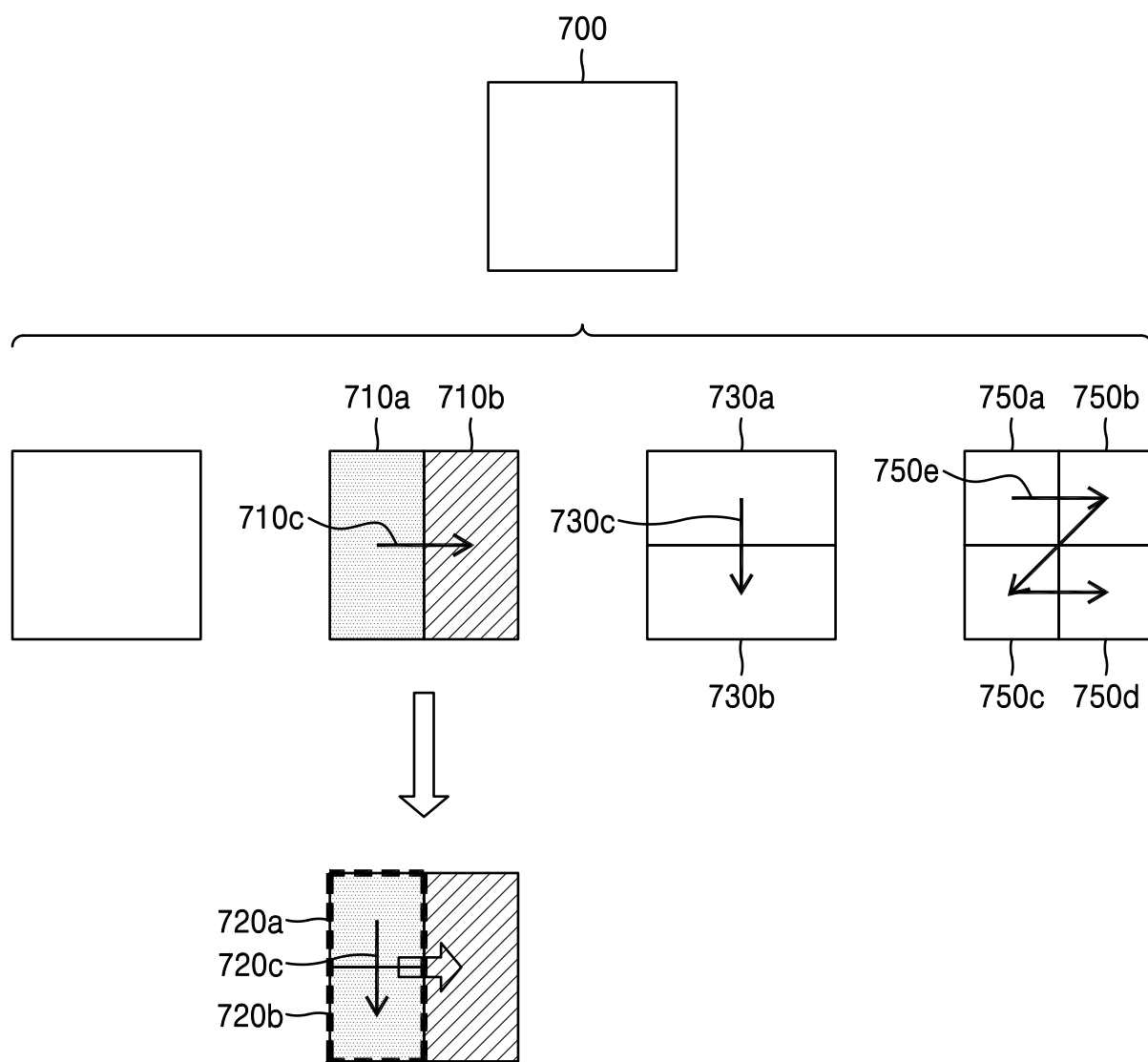


FIG. 8

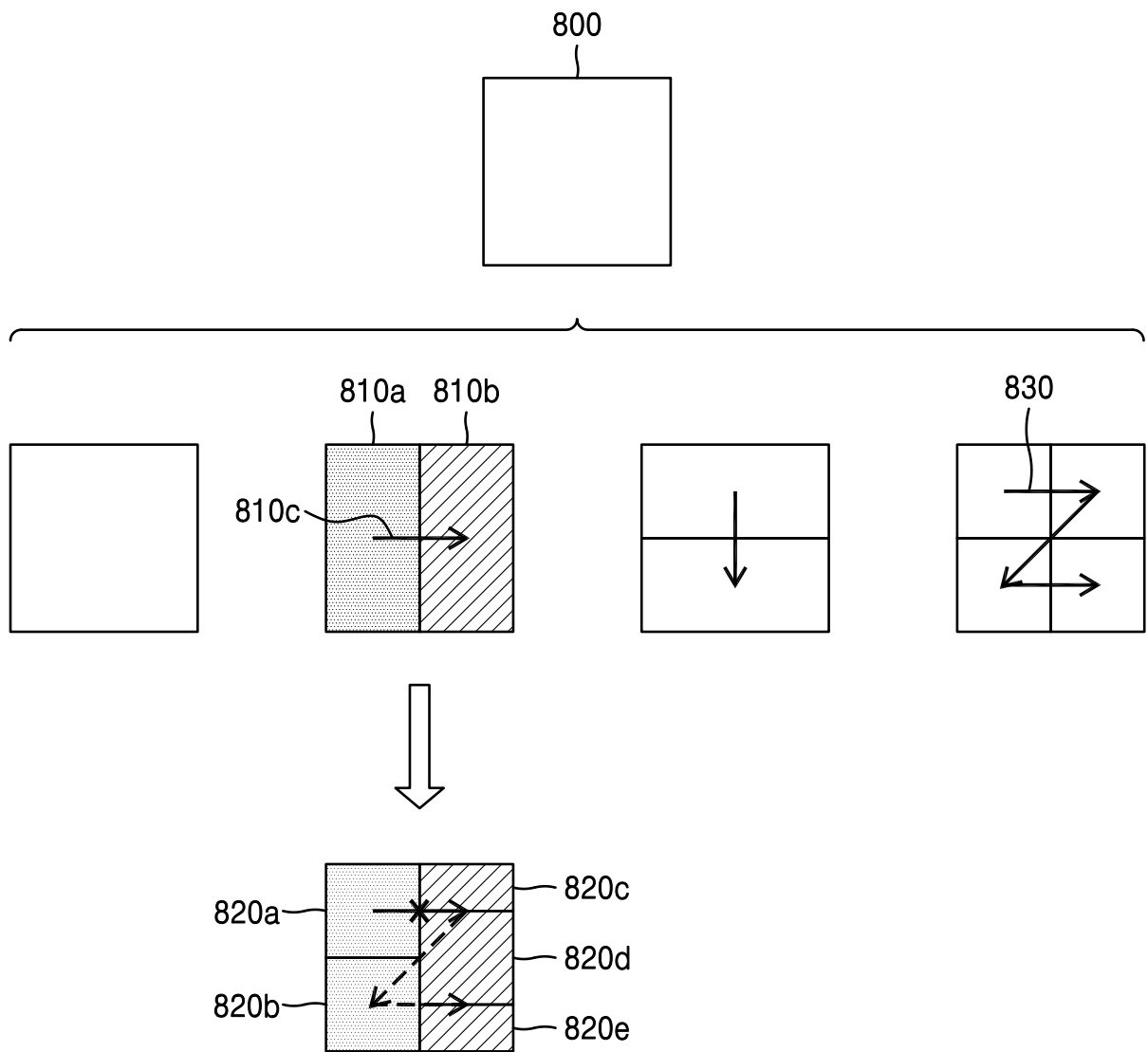


FIG. 9

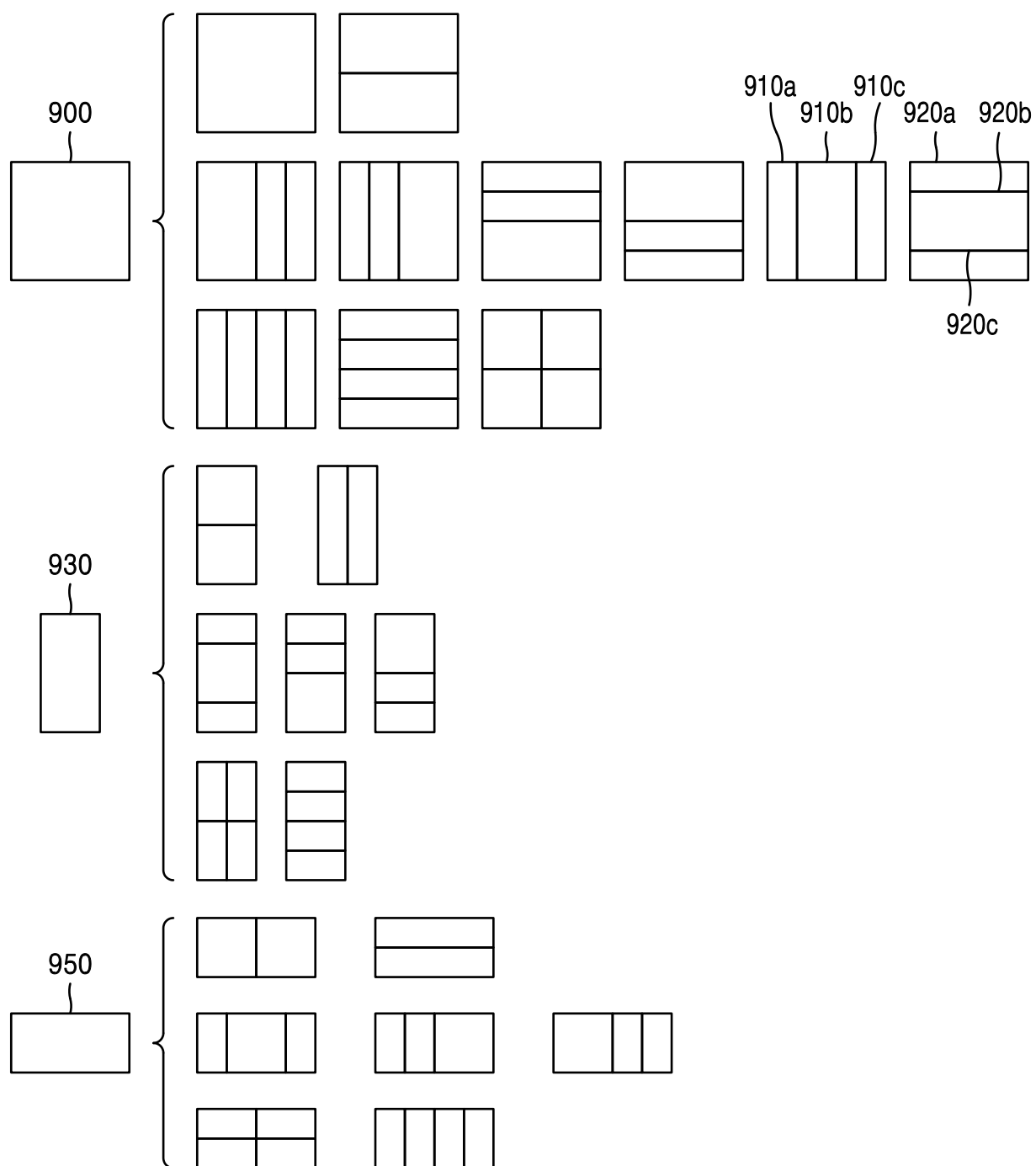


FIG. 11

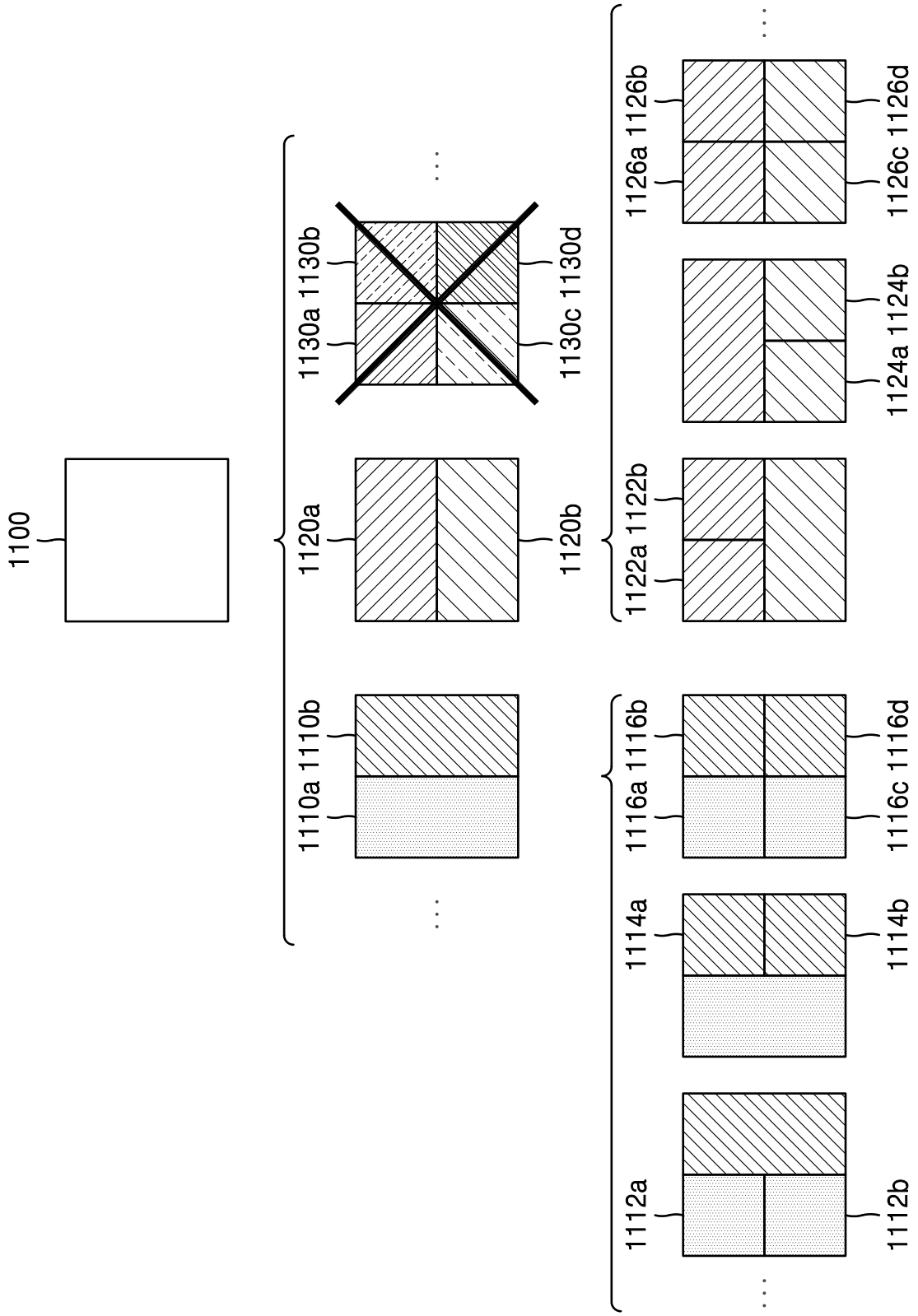


FIG. 12

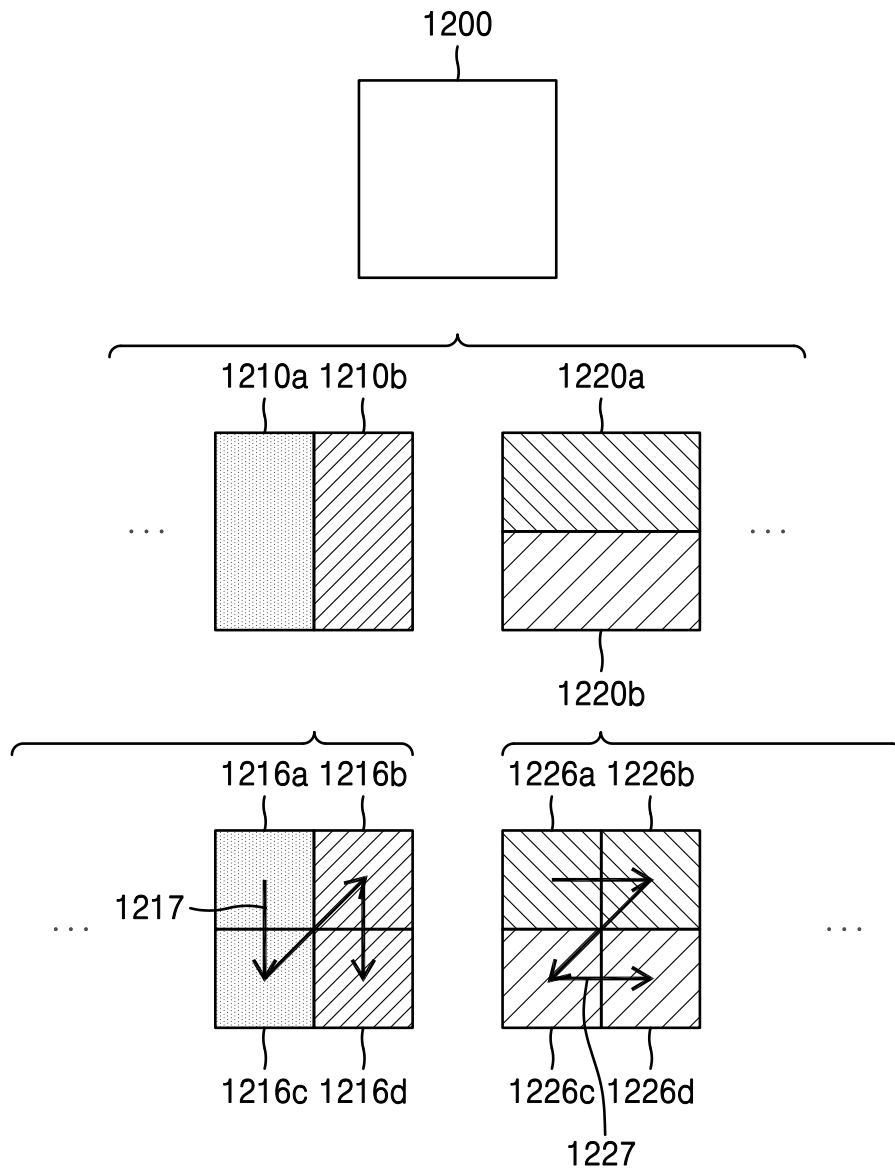


FIG. 13

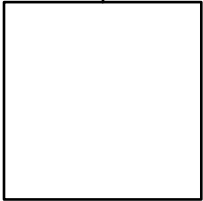
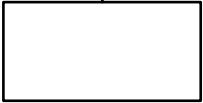
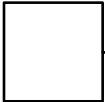
<div>BLOCK SHAPE</div> <div>DEPTH</div>	0: SQUARE	1: NS_VER	2: NS_HOR
DEPTH D	<div>1300</div> 	 <div>1310</div>	<div>1320</div> 
DEPTH D+1	 <div>1302</div>	 <div>1312</div>	 <div>1322</div>
DEPTH D+2	 <div>1304</div>	 <div>1314</div>	 <div>1324</div>
...	...	...	...

FIG. 14

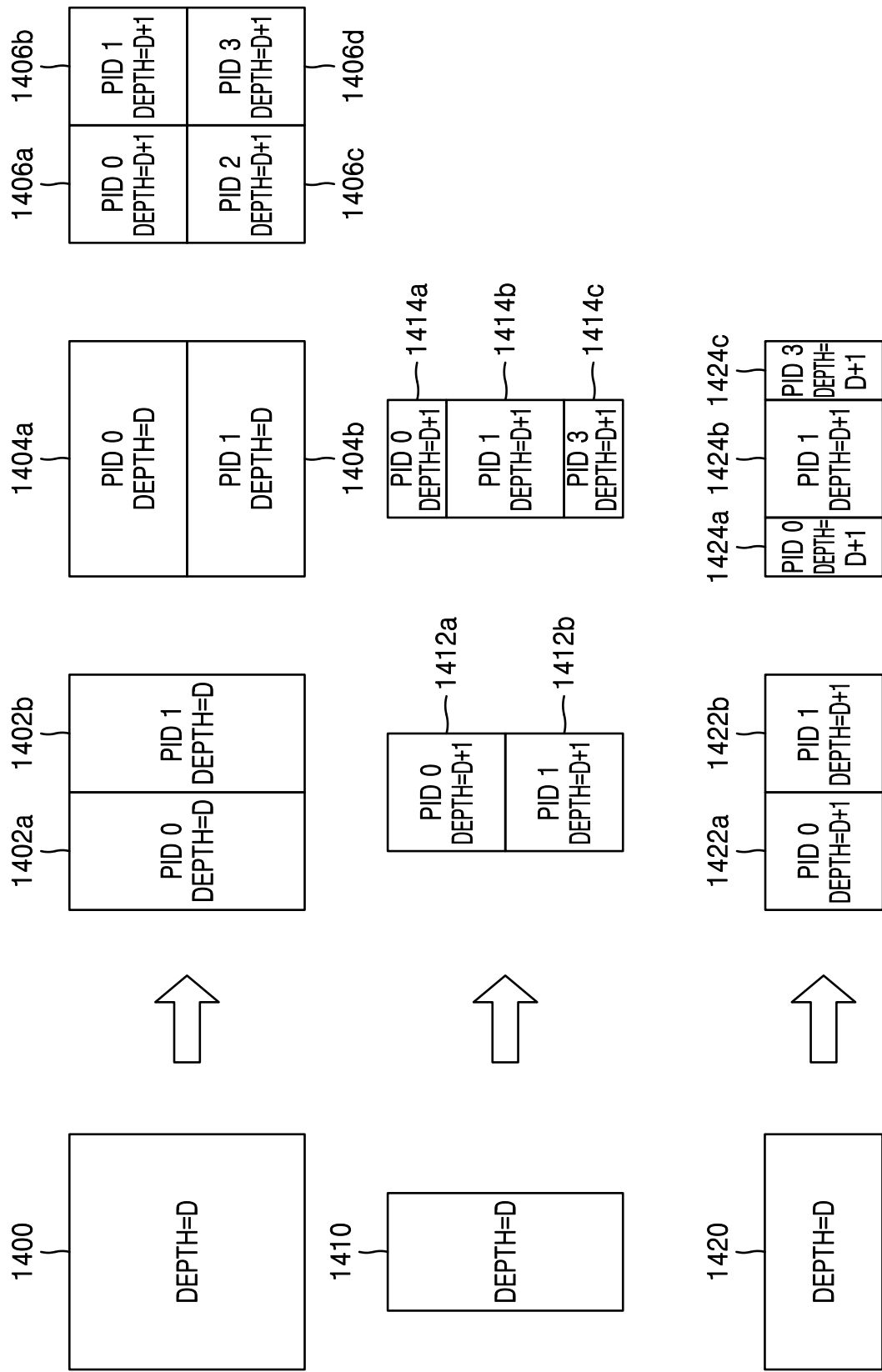


FIG. 15

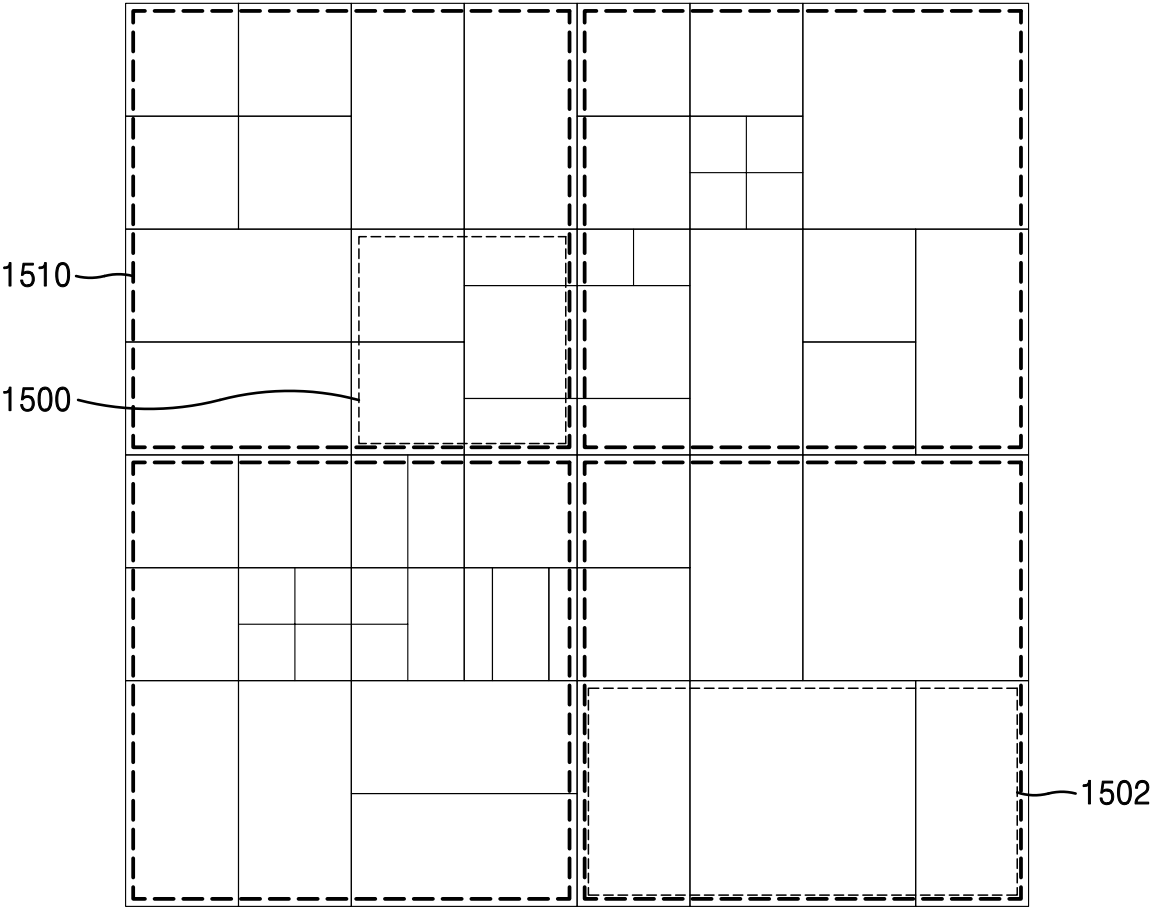


FIG. 16

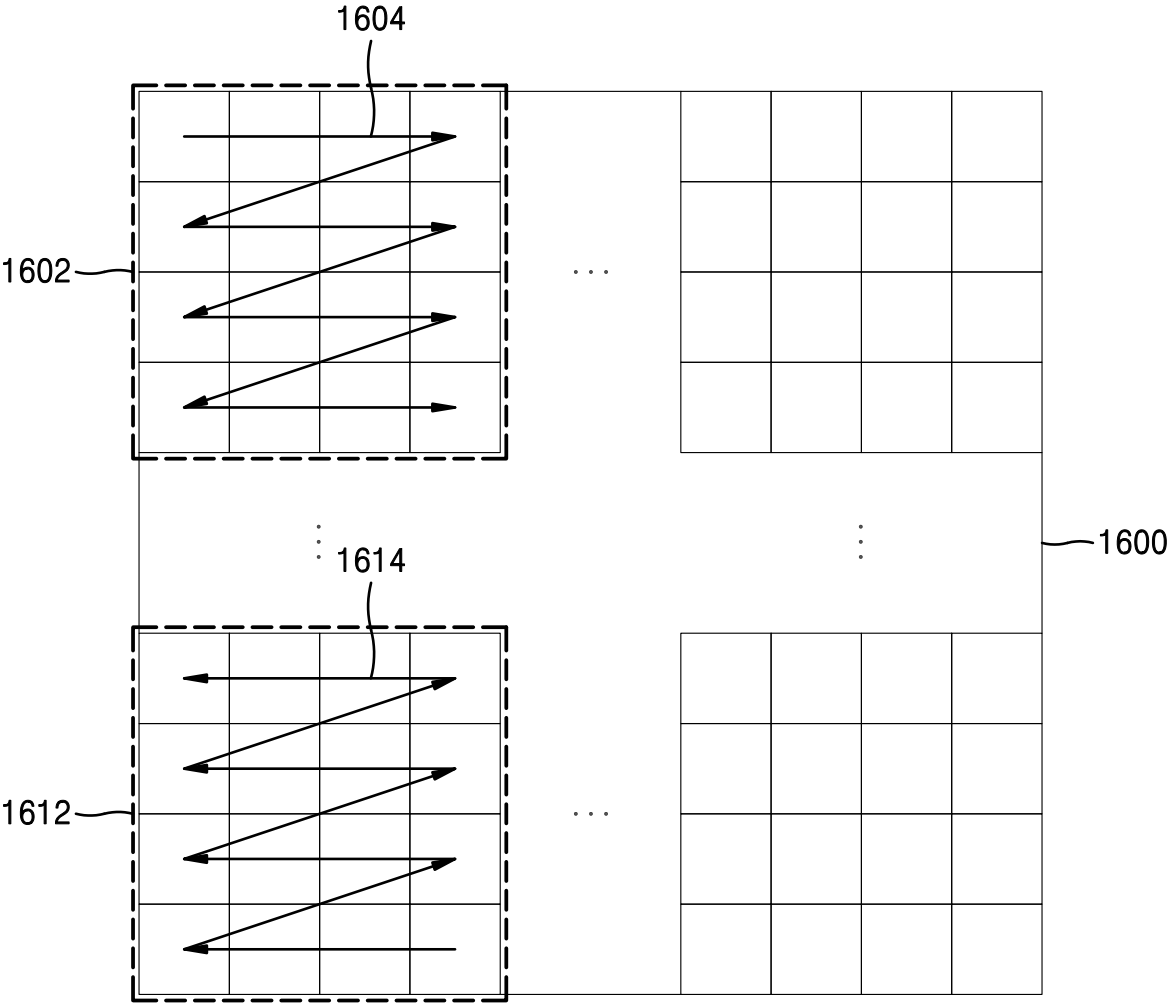


FIG. 17

1700

SHAPE OF CODING UNIT	RATIO OF WIDTH AND HEIGHT OF CODING UNIT	DIRECTION OF CODING UNIT
Square	1:1	—
Non-square	1:2	VERTICAL DIRECTION
	2:1	HORIZONTAL DIRECTION
	1:4	VERTICAL DIRECTION
	4:1	HORIZONTAL DIRECTION
	1:8	VERTICAL DIRECTION
	8:1	HORIZONTAL DIRECTION

FIG. 18

1800

		LENGTH OF WIDTH					
		4	8	16	32	64	128
RATIO	8:1	X	X	X	32x4	64x8	128x16
	4:1	X	X	16x4	32x8	64x16	128x32
	2:1	X	8x4	16x8	32x16	64x32	128x64
	1:1	4x4	8x8	16x16	32x32	64x64	128x128
	1:2	4x8	8x16	16x32	32x64	64x128	X
	1:4	4x16	8x32	16x64	32x128	X	X
	1:8	4x32	8x64	16x128	X	X	X

FIG. 19

1910

LENGTH OF WIDTH

	4	8	16	32	64	128
8:1	X	X	X	32x4	64x8	128x16
4:1	X	X	16x4	32x8	64x16	128x32
2:1	X	8x4	16x8	32x16	64x32	128x64
1:1	4x4	8x8	16x16	32x32	64x64	128x128
1:2	4x8	8x16	16x32	32x64	64x128	X
1:4	4x16	8x32	16x64	32x128	X	X
1:8	4x32	8x64	16x128	X	X	X

RATIO

1920

LENGTH OF WIDTH

	4	8	16	32	64	128
8:1	X	X	X	32x4	64x8	128x16
4:1	X	X	16x4	32x8	64x16	128x32
2:1	X	8x4	16x8	32x16	64x32	128x64
1:1	4x4	8x8	16x16	32x32	64x64	128x128
1:2	4x8	8x16	16x32	32x64	64x128	X
1:4	4x16	8x32	16x64	32x128	X	X
1:8	4x32	8x64	16x128	X	X	X

RATIO

1930

LENGTH OF WIDTH

	4	8	16	32	64	128
8:1	X	X	X	32x4	64x8	128x16
4:1	X	X	16x4	32x8	64x16	128x32
2:1	X	8x4	16x8	32x16	64x32	128x64
1:1	4x4	8x8	16x16	32x32	64x64	128x128
1:2	4x8	8x16	16x32	32x64	64x128	X
1:4	4x16	8x32	16x64	32x128	X	X
1:8	4x32	8x64	16x128	X	X	X

RATIO

1940

LENGTH OF HEIGHT

	4	8	16	32	64	128
4	4x4	4x8	4x16	4x32	X	X
8	8x4	8x8	8x16	8x32	8x64	X
16	16x4	16x8	16x16	16x32	16x64	16x128
32	32x4	32x8	32x16	32x32	32x64	32x128
64	X	64x8	64x16	64x32	64x64	64x128
128	X	X	128x16	128x32	128x64	128x128

LENGTH OF WIDTH

FIG. 20

2000

LENGTH OF HEIGHT						
	4	8	16	32	64	128
4	X	4	4	X	X	X
8	4	4	3	3	X	X
16	4	3	3	2	2	X
32	X	3	2	2	1	1
64	X	X	2	1	1	0
128	X	X	X	1	0	0

LENGTH OF WIDTH

FIG. 21

```

#define NA 255 //never split
#define NB 14  //not reach in current setting of max AR 1:4
#define NC 15  //not reach in current setting of max AR 1:4

const u8 ifvc_tbl_split_flag_ctx[6][6] =
{
    {NA, 4, 4, NB, NC, NC},
    { 4, 4, 3, 3, NB, NC},
    { 4, 3, 3, 2, 2, NB},
    {NB, 3, 2, 2, 1, 1},
    {NC, NB, 2, 1, 1, 0},
    {NC, NC, NB, 1, 0, 0},
};

```

2110

```

for(i=0; i<3; i++) /* 0: Up, 1: Left, 2: Right */
{
    if(avail[i])
    {
        w[i] = width_of_block[i];
        h[i] = height_of_block[i];
        if(i==0)
            smaller[i] = w[i] < current_block_width;
        else
            smaller[i] = h[i] < current_block_height;
    }
}
ctx = min(smaller[0] + smaller[1] + smaller[2], 2);
ctx = ctx + 3 * ifvc_tbl_split_flag_ctx[log2_cuw-2][log2_cuh-2];

```

2120

```
u8 ctx_dir = (log2_cuw-log2_cuh+2) + dir_ctx_offset;
```

2130

```
u8 ctx_typ = 0 + typ_ctx_offset;
```

2140

FIG. 22

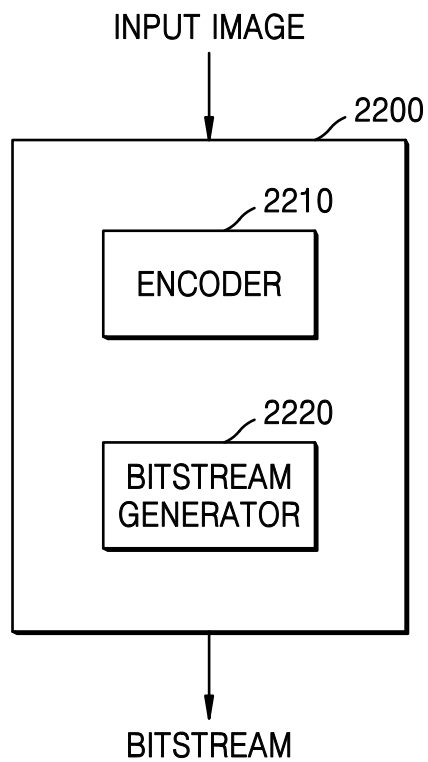


FIG. 23

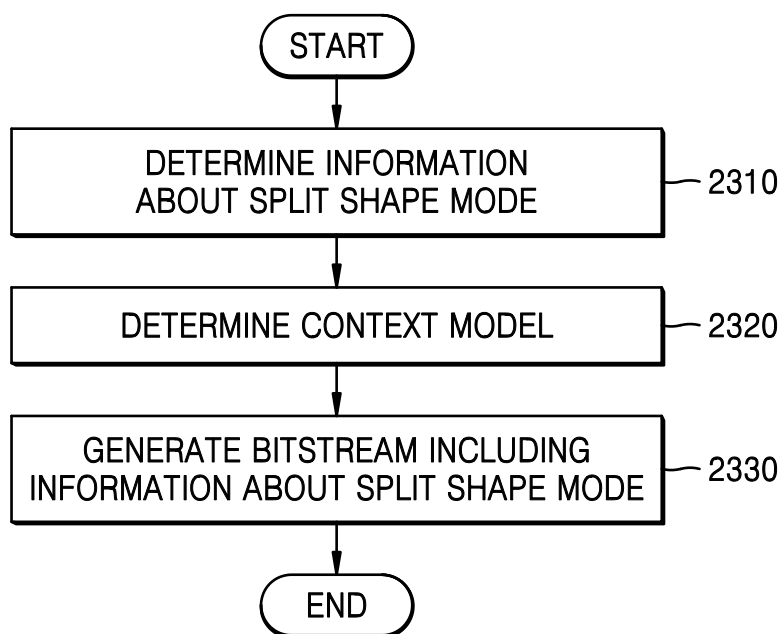


FIG. 24

split_unit() {	Descriptor
split_mode	ae(v)
if(split_mode == QUAD_SPLIT split_mode == BI_VER_SPLIT	
split_mode == TRI_VER_SPLIT) {	
suco_info	ae(v)
}	
if(split_info == QUAD_SPLIT) {	
if(suco_info == 0) {	
split_unit(sub_pos_0)	
split_unit(sub_pos_1)	
split_unit(sub_pos_2)	
split_unit(sub_pos_3)	
}	
else {	
split_unit(sub_pos_1)	
split_unit(sub_pos_0)	
split_unit(sub_pos_3)	
split_unit(sub_pos_2)	
}	
}	
else if(split_mode == BI_VER_SPLIT) {	
if(suco_info == 0) {	
split_unit(sub_pos_0)	
split_unit(sub_pos_1)	
}	
else {	
split_unit(sub_pos_1)	
split_unit(sub_pos_0)	
}	
}	
else if(split_mode == BI_HOR_SPLIT) {	
split_unit(sub_pos_0)	
split_unit(sub_pos_1)	
}	
else if(split_mode == TRI_VER_SPLIT) {	
if(suco_info == 0) {	
split_unit(sub_pos_0)	
split_unit(sub_pos_1)	
split_unit(sub_pos_2)	
}	
else {	
split_unit(sub_pos_2)	
split_unit(sub_pos_1)	
split_unit(sub_pos_0)	
}	
}	
else if(split_mode == TRI_HOR_SPLIT) {	
split_unit(sub_pos_0)	
split_unit(sub_pos_1)	
split_unit(sub_pos_2)	
}	
else { // split_mode == NO_SPLIT	
coding_unit(...)	
}	
}	