

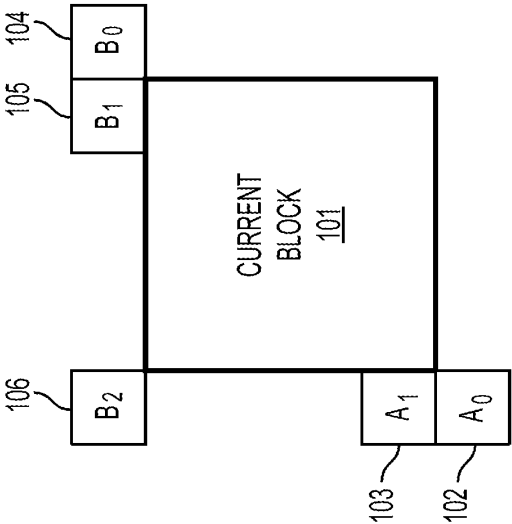


FIG. 1
(Related Art)

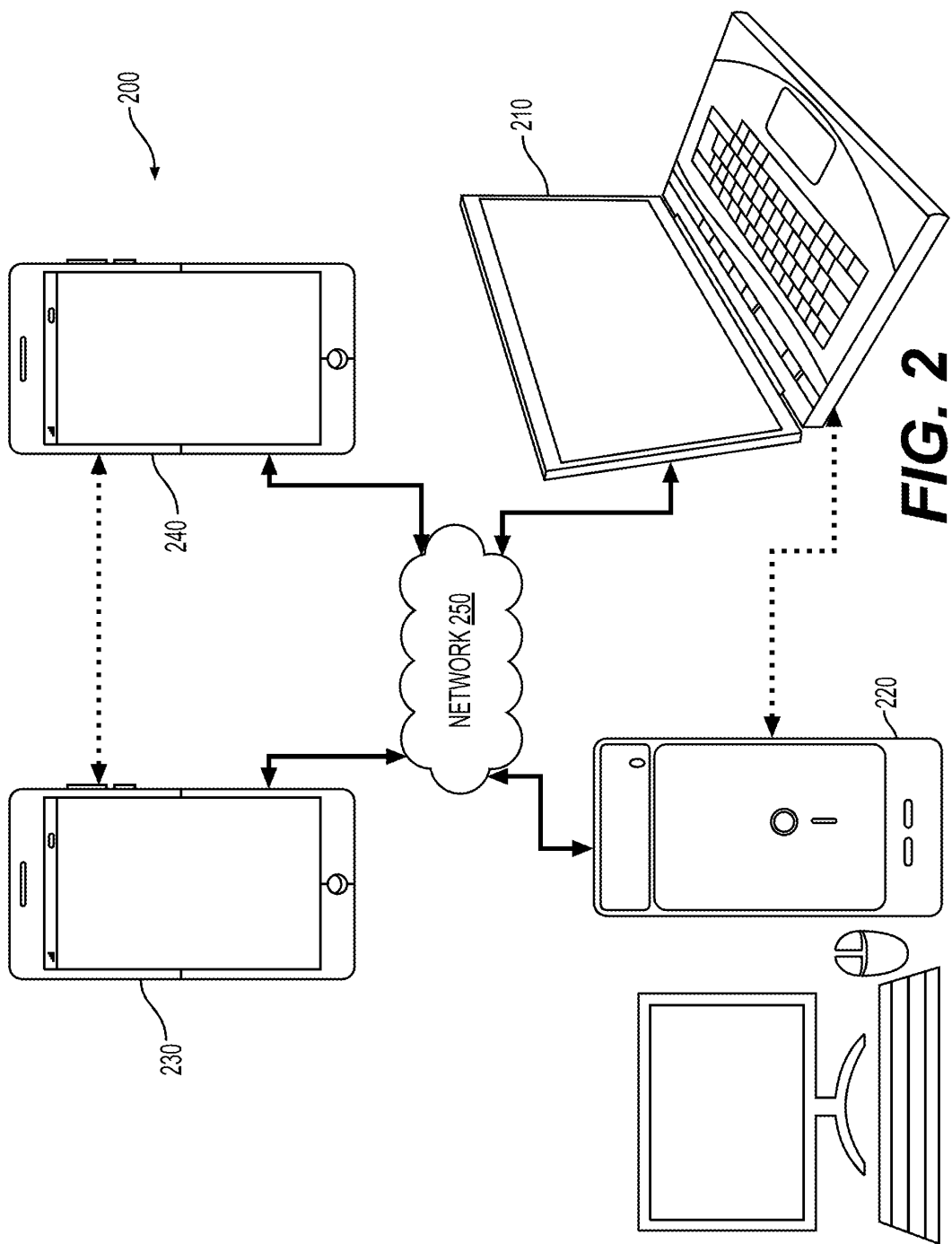


FIG. 2

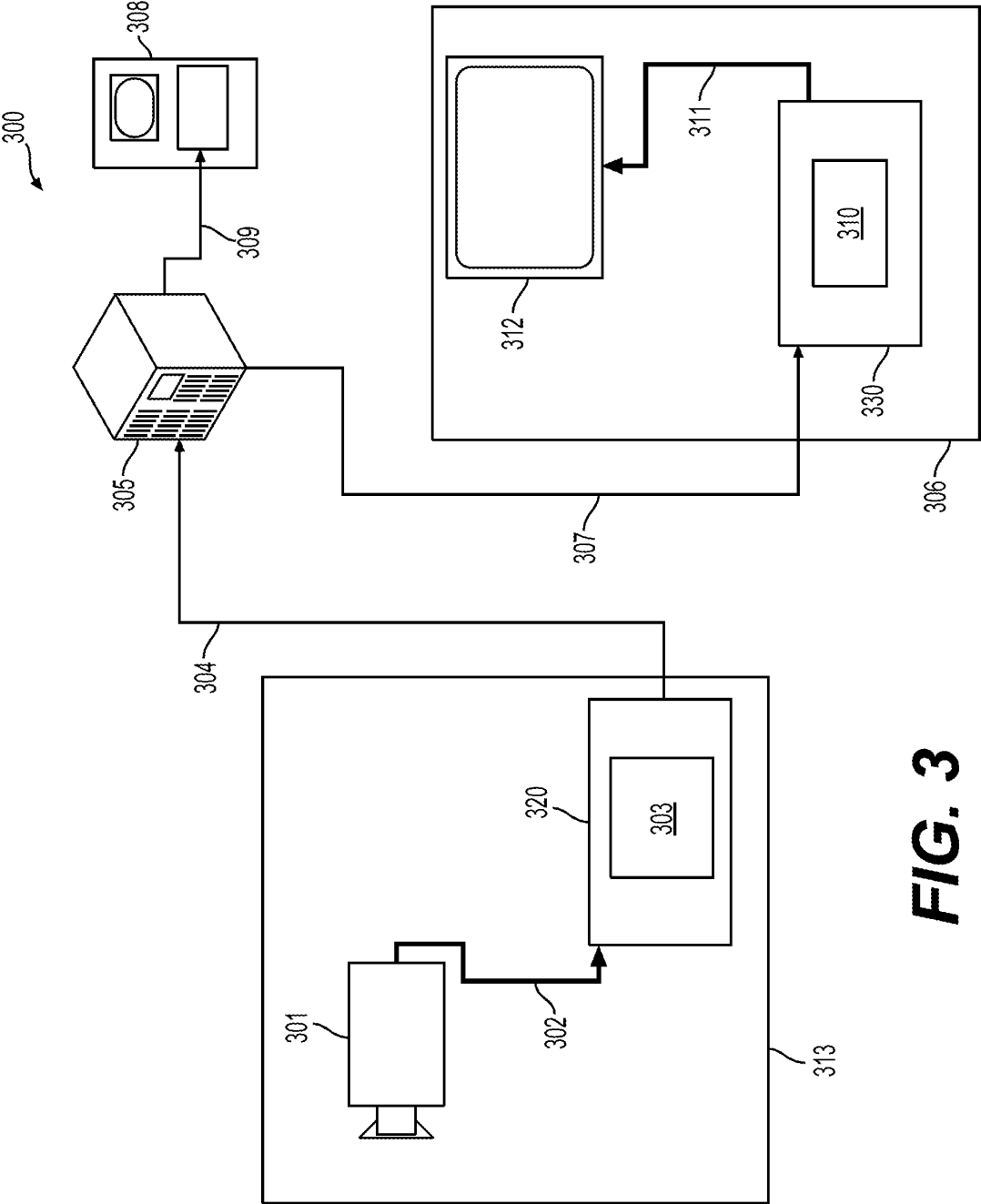
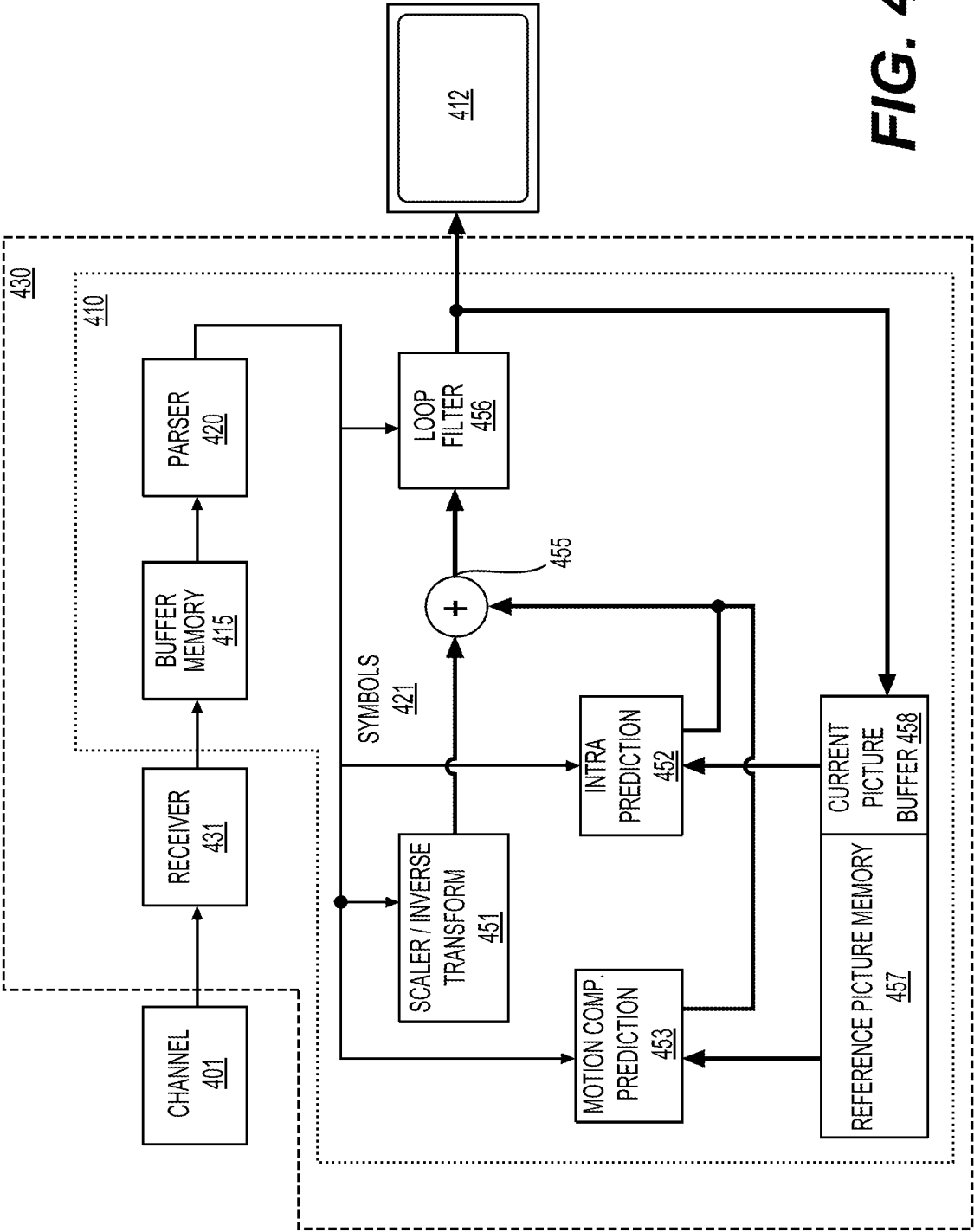
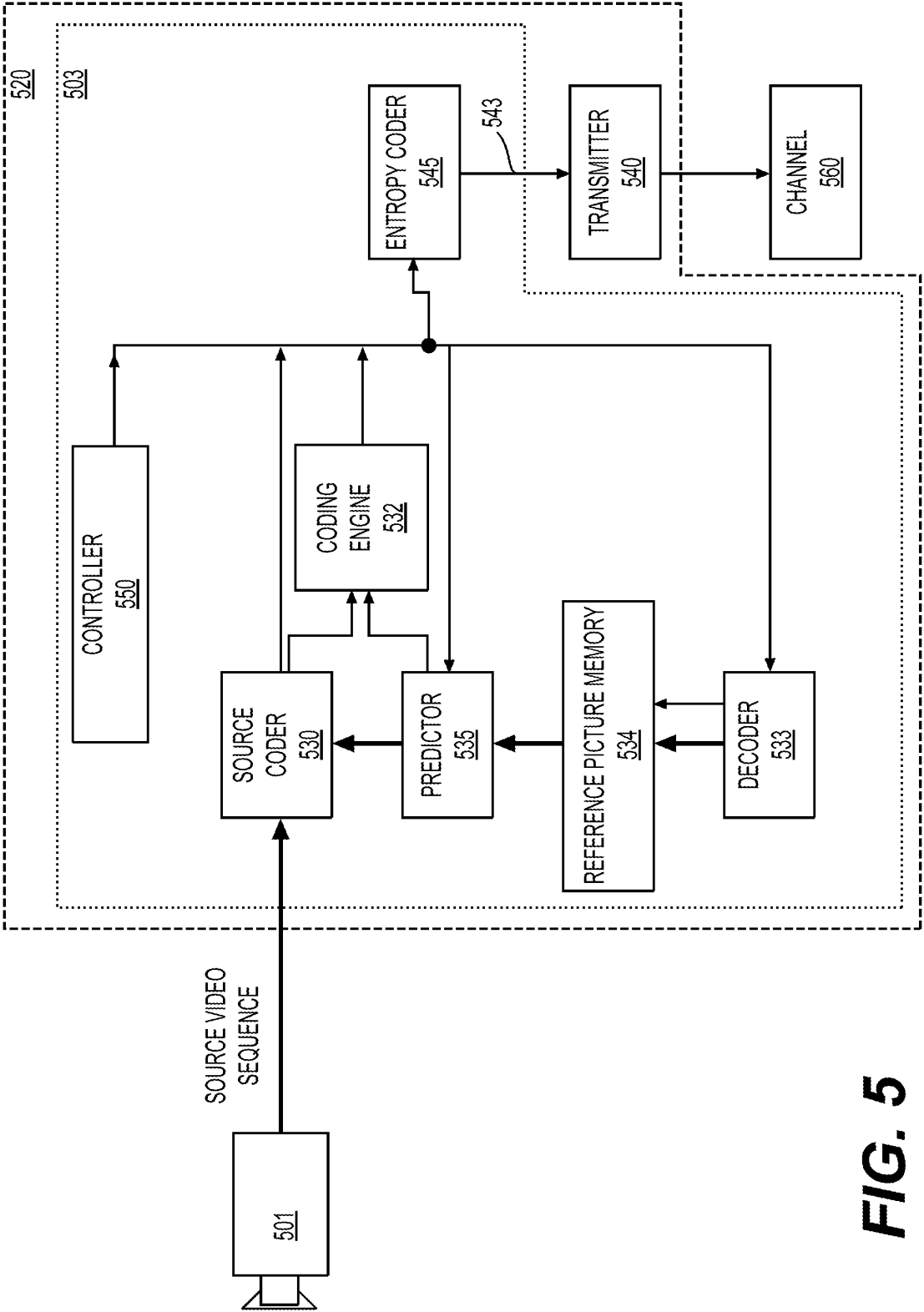


FIG. 3





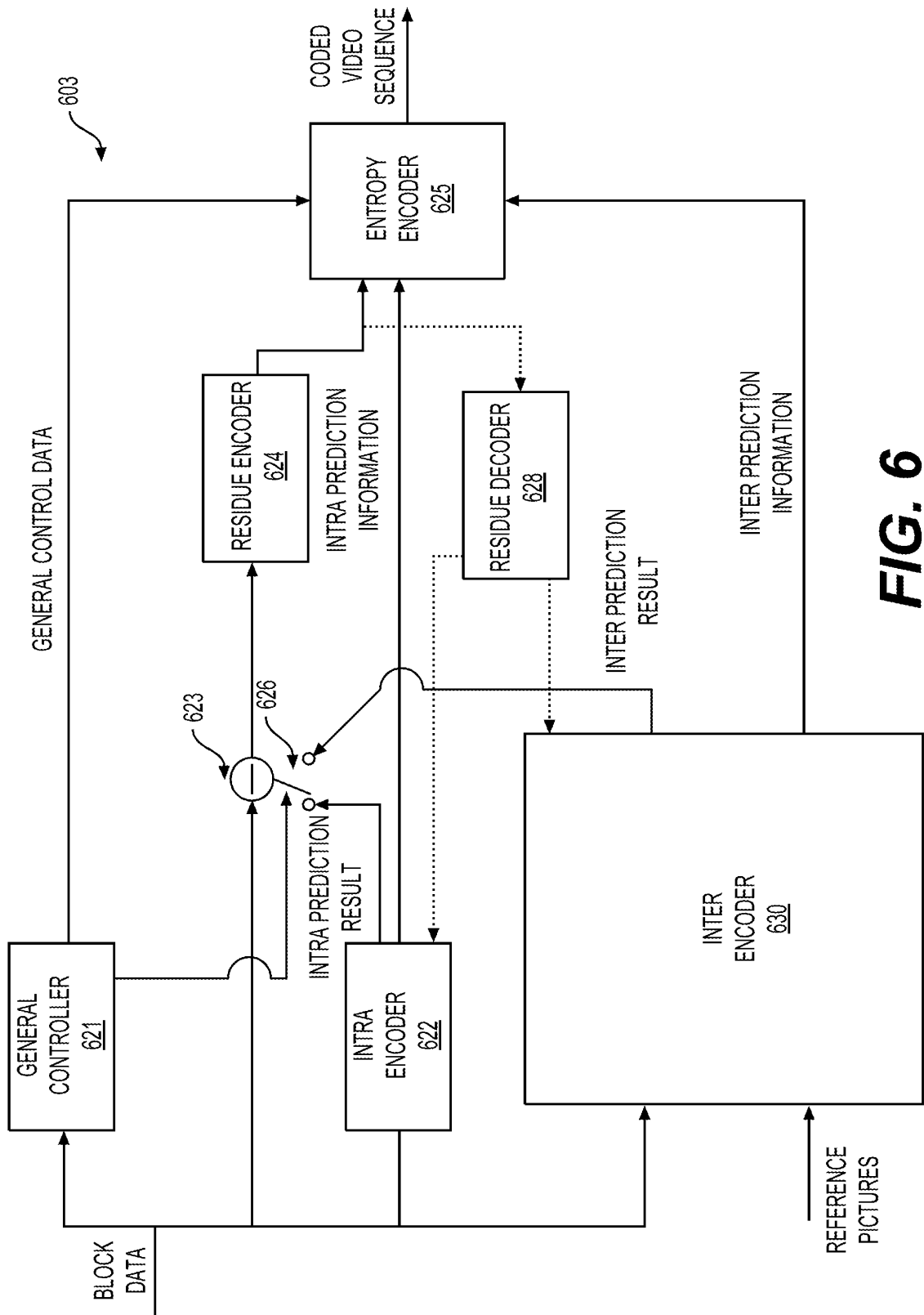


FIG. 6

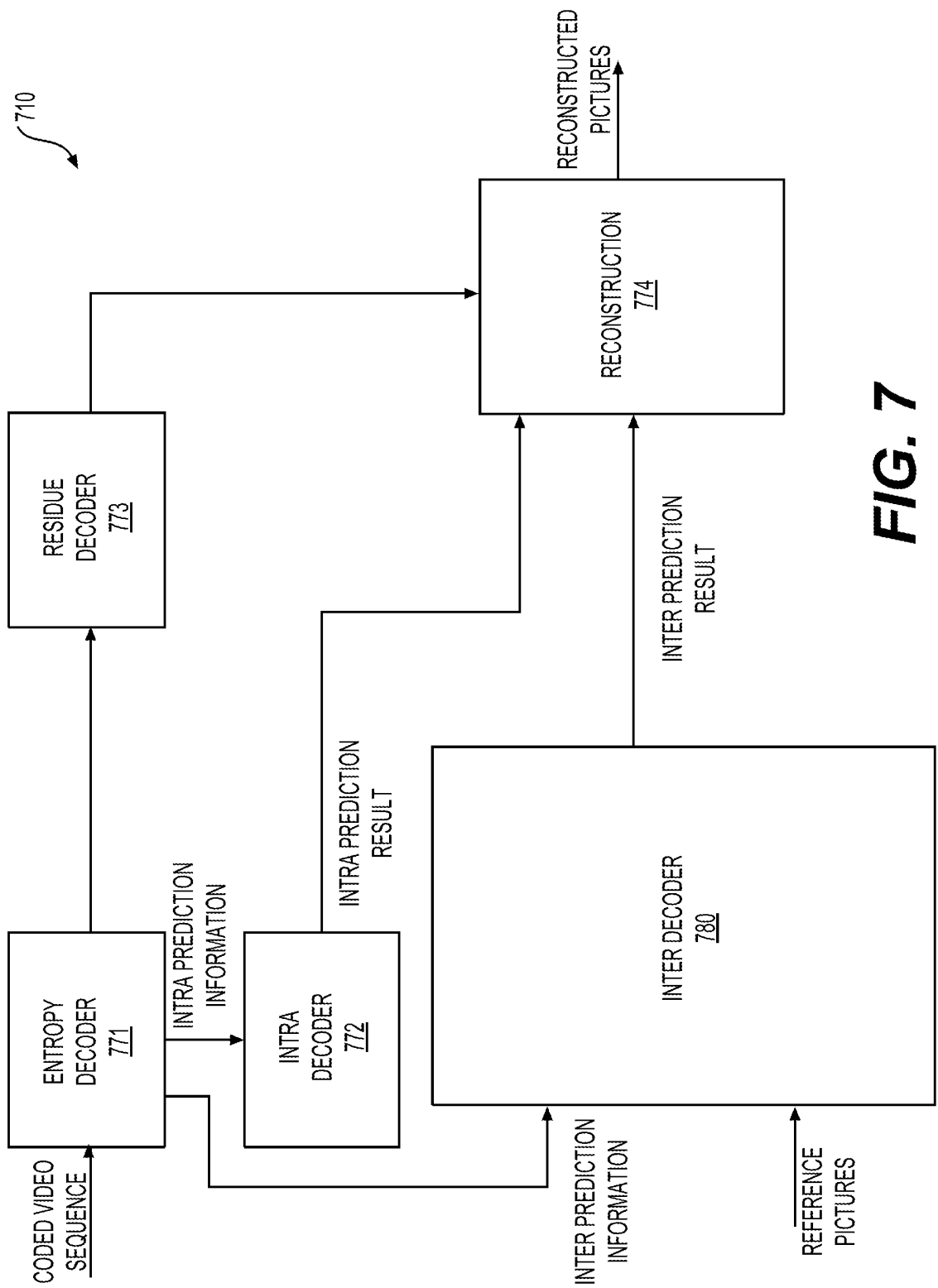


FIG. 7

4x4 transform

{64, 64, 64, 64}
{83, 36, -36, -83}
{64, -64, -64, 64}
{36, -83, 83, -36}

FIG. 8A

16x16 transform

{64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64}
{90 87 80 70 57 43 25 9 -9-25-43-57-70-80-87-90}
{89 75 50 18-18-50-75-89-89-75-50-18 18 50 75 89}
{87 57 9-43-80-90-70-25 25 70 90 80 43 -9-57-87}
{83 36-36-83-83-36 36 83 83 36-36-83-83-36 36 83}
{80 9-70-87-25 57 90 43-43-90-57 25 87 70 -9-80}
{75-18-89-50 50 89 18-75-75 18 89 50-50-89-18 75}
{70-43-87 9 90 25-80-57 57 80-25-90 -9 87 43-70}
{64-64-64 64 64-64-64 64 64-64-64 64 64 64-64-64 64}
{57-80-25 90 -9-87 43 70-70-43 87 9-90 25 80-57}
{50-89 18 75-75-18 89-50-50 89-18-75 75 18-89 50}
{43-90 57 25-87 70 9-80 80 -9-70 87-25-57 90-43}
{36-83 83-36-36 83-83 36 36-83 83-36-36 83-83 36}
{25-70 90-80 43 9-57 87-87 57 -9-43 80-90 70-25}
{18-50 75-89 89-75 50-18-18 50-75 89-89 75-50 18}
{ 9-25 43-57 70-80 87-90 90-87 80-70 57-43 25 -9}

8x8 transform

{64, 64, 64, 64, 64, 64, 64, 64}
{89, 75, 50, 18, -18, -50, -75, -89}
{83, 36, -36, -83, -83, -36, 36, 83}
{75, -18, -89, -50, 50, 89, 18, -75}
{64, -64, -64, 64, 64, -64, -64, 64}
{50, -89, 18, 75, -75, -18, 89, -50}
{36, -83, 83, -36, -36, 83, -83, 36}
{18, -50, 75, -89, 89, -75, 50, -18}

FIG. 8B

FIG. 8C

32x32 transform

```

{64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64}
{90 90 88 85 82 78 73 67 61 54 46 38 31 22 13 4 -4 -13 -22 -31 -38 -46 -54 -61 -67 -73 -78 -82 -85 -88 -90 -90}
{90 87 80 70 57 43 25 9 -9 -25 -43 -57 -70 -80 -87 -90 -87 -80 -70 -57 -43 -25 -9 9 25 43 57 70 80 87 90}
{90 82 67 46 22 -4 -31 -54 -73 -85 -90 -88 -78 -61 -38 -13 13 38 61 78 88 90 85 73 54 31 4 -22 -46 -67 -82 -90}
{89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89 89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89}
{88 67 31 -13 -54 -82 -90 -78 -46 -4 38 73 90 85 61 22 -22 -61 -85 -90 -73 -38 4 46 78 90 82 54 13 -31 -67 -88}
{87 57 9 -43 -80 -90 -70 -25 25 70 90 80 43 -9 -57 -87 -87 -57 -9 43 80 90 70 25 -25 -70 -90 -80 -43 9 57 87}
{85 46 -13 -67 -90 -73 -22 38 82 88 54 -4 -61 -90 -78 -31 31 78 90 61 4 -54 -88 -82 -38 22 73 90 67 13 -46 -85}
{83 36 -36 -83 -83 -36 36 83 36 -36 -83 -83 -36 36 83 36 -36 -83 -83 -36 36 83 36 -36 -83 -83 -36 36 83}
{82 22 -54 -90 -61 13 78 85 31 -46 -90 -67 4 73 88 38 -38 -88 -73 -4 67 90 46 -31 -85 -78 -13 61 90 54 -22 -82}
{80 9 -70 -87 -25 57 90 43 -43 -90 -57 25 87 70 -9 -80 -80 -9 70 87 25 -57 -90 -43 43 90 57 -25 -87 -70 9 80}
{78 -4 -82 -73 13 85 67 -22 -88 -61 31 90 54 -38 -90 -46 46 90 38 -54 -90 -31 61 88 22 -67 -85 -13 73 82 4 -78}
{75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75 75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75}
{73 -31 -90 -22 78 67 -38 -90 -13 82 61 -46 -88 -4 85 54 -54 -85 4 88 46 -61 -82 13 90 38 -67 -78 22 90 31 -73}
{70 -43 -87 9 90 25 -80 -57 57 80 -25 -90 -9 87 43 -70 -70 43 87 -9 -90 -25 80 57 -57 -80 25 90 9 -87 -43 70}
{67 -54 -78 38 85 -22 -90 4 90 13 -88 -31 82 46 -73 -61 61 73 -46 -82 31 88 -13 -90 -4 90 22 -85 -38 78 54 -67}
{64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64}
{61 -73 -46 82 31 -88 -13 90 -4 -90 22 85 -38 -78 54 67 -67 -54 78 38 -85 -22 90 4 -90 13 88 -31 -82 46 73 -61}
{57 -80 -25 90 -9 -87 43 70 -70 -43 87 9 -90 25 80 -57 -57 80 25 -90 9 87 -43 -70 70 43 -87 -9 90 -25 -80 57}
{54 -85 -4 88 -46 -61 82 13 -90 38 67 -78 -22 90 -31 -73 73 31 -90 22 78 -67 -38 90 -13 -82 61 46 -88 4 85 -54}
{50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50 50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50}
{46 -90 38 54 -90 31 61 -88 22 67 -85 13 73 -82 4 78 -78 -4 82 -73 -13 85 -67 -22 88 -61 -31 90 -54 -38 90 -46}
{43 -90 57 25 -87 70 9 -80 80 -9 -70 87 -25 -57 90 -43 -43 90 -57 -25 87 -70 -9 80 -80 9 70 -87 25 57 -90 43}
{38 -88 73 -4 -67 90 -46 -31 85 -78 13 61 -90 54 22 -82 82 -22 -54 90 -61 -13 78 -85 31 46 -90 67 4 -73 88 -38}
{36 -83 83 -36 -36 83 -36 36 -83 83 -36 -36 83 -36 36 -83 83 -36 -36 83 -36 36 -83 83 -36 -36 83 -36 36}
{31 -78 90 -61 4 54 -88 82 -38 -22 73 -90 67 -13 -46 85 -85 46 13 -67 90 -73 22 38 -82 88 -54 -4 61 -90 78 -31}
{25 -70 90 -80 43 9 -57 87 -87 57 -9 43 80 -90 70 -25 -25 70 -90 80 -43 -9 57 -87 87 -57 9 43 -80 90 -70 25}
{22 -61 85 -90 73 -38 -4 46 -78 90 -82 54 -13 -31 67 -88 88 -67 31 13 -54 82 -90 78 -46 4 38 -73 90 -85 61 -22}
{18 -50 75 -89 89 -75 50 -18 -18 50 -75 89 -89 75 -50 18 18 -50 75 -89 89 -75 50 -18 -18 50 -75 89 -89 75 -50 18}
{13 -38 61 -78 88 -90 85 -73 54 -31 4 22 -46 67 -82 90 -90 82 -67 46 -22 -4 31 -54 73 -85 90 -88 78 -61 38 -13}
{9 -25 43 -57 70 -80 87 -90 90 -87 80 -70 57 -43 25 -9 -9 25 -43 57 -70 80 -87 90 -90 87 -80 70 -57 43 -25 9}
{4 -13 22 -31 38 -46 54 -61 67 -73 78 -82 85 -88 90 -90 90 -90 88 -85 82 -78 73 -67 61 -54 46 -38 31 -22 13 -4}

```

FIG. 8D

FIG. 9A

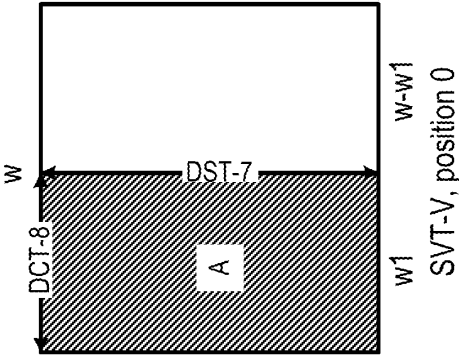


FIG. 9B

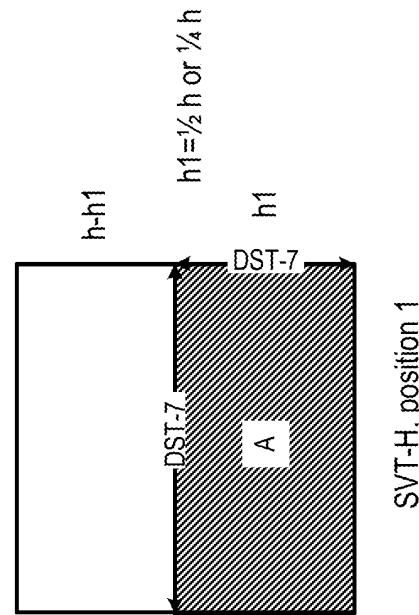
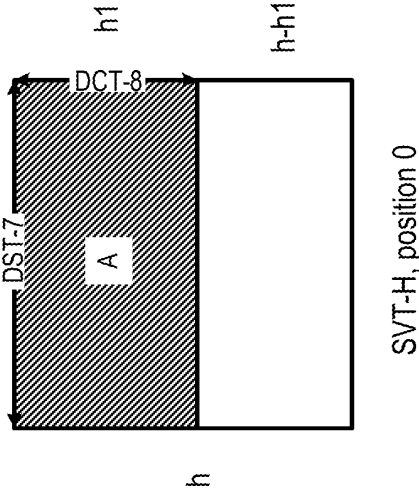
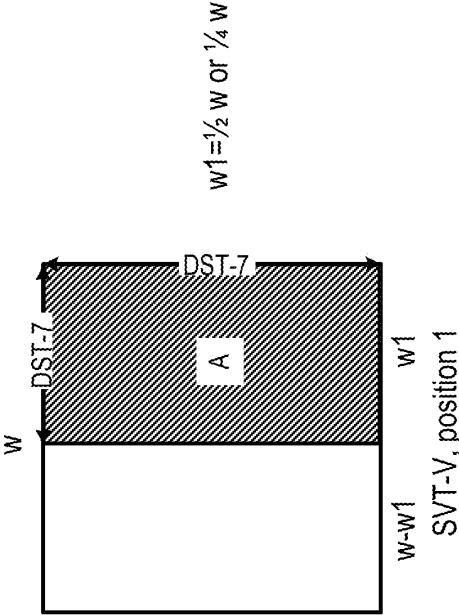


FIG. 9C

FIG. 9D

Number of sub-partitions depending on the block size

Block Size	Number of Sub-Partitions
4×4	Not divided
4×8 and 8×4	2
All other cases	4

FIG. 10

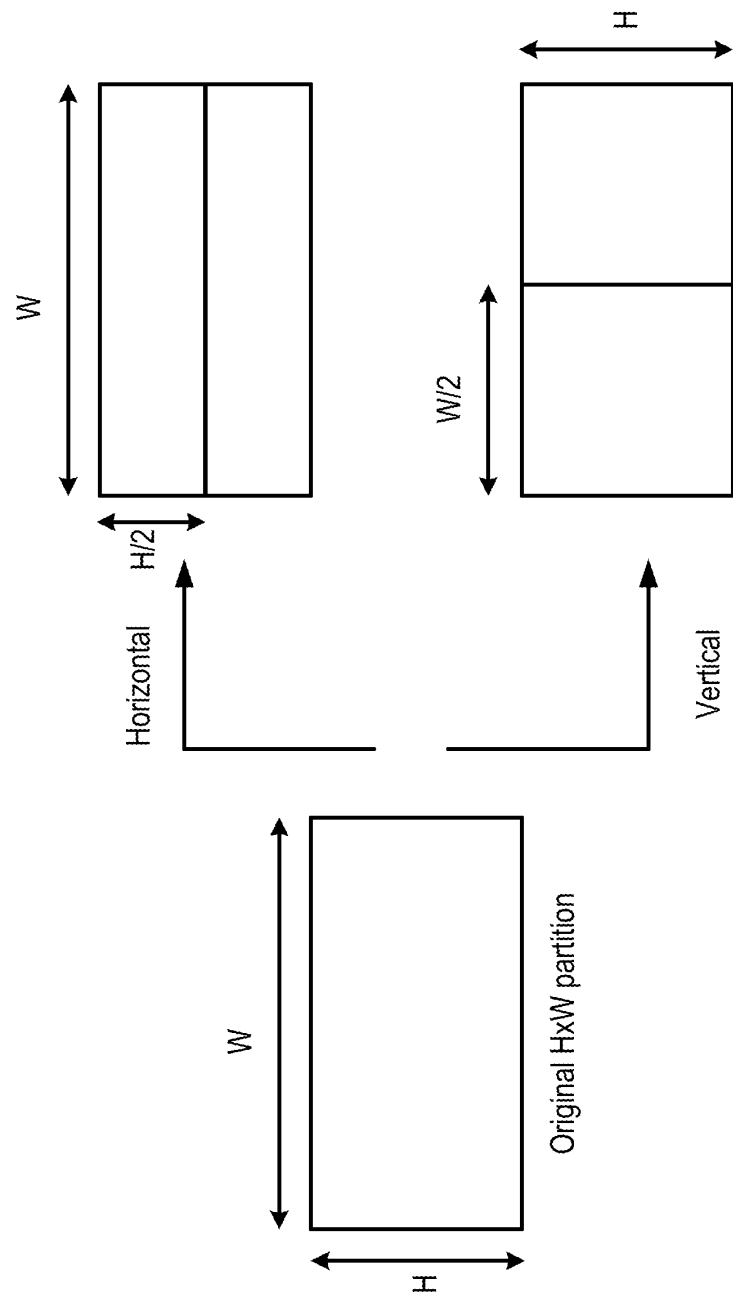


FIG. 11

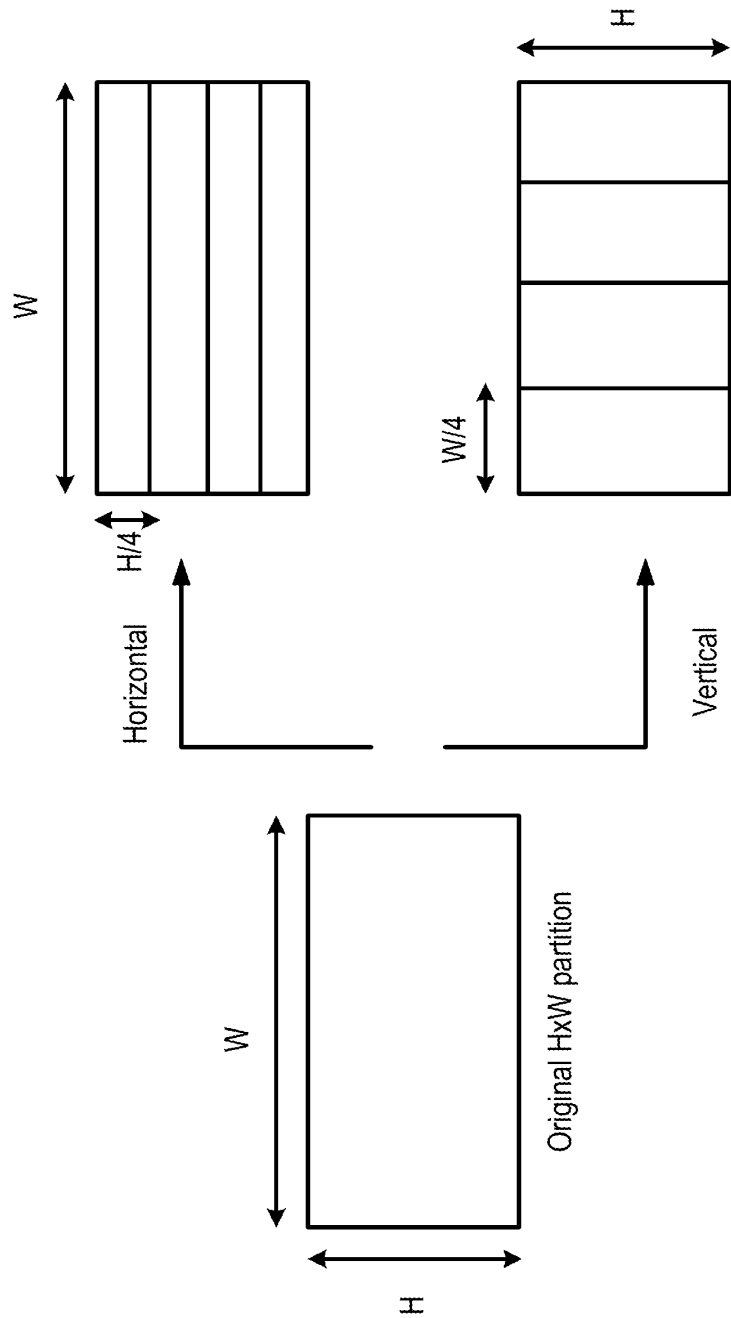


FIG. 12

[illegible]

FIG. 13C

Continue from FIG. 13D

```

{ ci, -cd, by, -br, bo, -bj, bf, -bk, bp, -bu, bz, -ce, cj, ch, -cc, bx, -bs, bn, -bi, bg, -bi, bq, -bv, ca, -cf, ck,
  cg, -cb, bw, -br, bm, -bn, bh, -bm, bt, -bw, cb, -cg, -ck, ci, -ca, bv, -bq, bi, -bg, bi, -bs, cc, -ch, -cj, ce,
  -bz, bu, -bp, bk, -bf, bj, -bo, bt, -by, cd, -ci }
{ ac, -an, am, -al, ak, -aj, ai, -ah, ah, -ai, aj, -ak, an, -am, al, -ak, aj, -ai, ah, -ah, ai, -
  aj, ak, -al, am, -an, ao, -aj, ai, -ah, ah, -ai, aj, -ak, an, -am, al, -am, ai, -ak,
  aj, -ai, ah, -ah, ai, -aj, ak, -al, am, -an, ao }
{ cj, -cg, cd, -ca, bx, -bu, br, -bo, bi, -bi, bf, -bh, bk, -bn, bq, -bt, bw, -bs, cc, -cf, ci, ck, -ch, bt, -bq, bn, -bk, bh,
  bv, -bs, -bp, km, -bj, bg, -bg, kj, -bm, bp, -bs, bv, -by, cb, -ca, ch, -ck, -ci, cf, -cc, bz, -bw, bt, -bq, bn, -bk, bh,
  -bf, bi, -bi, bo, -br, bu, -bx, ca, -cd, cg, -cj }
{ be, -bd, bc, -bb, ba, -az, ay, -ax, aw, -av, au, -at, as, -ar, ag, -aq, ar, -as, at, -au, av, -aw, ax, -ay,
  az, -ba, bb, -bc, bd, -be, bd, -bc, bb, -ba, az, -ay, ax, -aw, at, -as, ar, -ag, ap, -aq, ap, -ar, as, -at,
  au, -av, aw, -ax, ay, -az, ba, -bb, bc, -bd, be }
{ ck, -cj, ci, -ch, cg, -cf, ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca, -ca,
  bk, -bj, bi, -bi, bh, -bf, bg, -bf, bh, -bi, bj, -bk, bi, -bm, bn, -bo, bp, -bq, br, -bs, bt, -bu, bv, -bw, bx, -by, bz,
  -ca, ch, -cc, cd, -ce, cf, -cg, ch, -ci, cj, -ck }
}

```

where

```

{ aa, ab, ac, ad, ae, af, ag, ah, ai, aj, ak, al, am, an, ao, ap, aq, ar, as, at, au, av, aw, ax, ay, az, ba, bb,
  bc, bd, be, bf, bg, bh, bi, bj, bk, bl, bm, bn, bo, bp, bq, br, bs, bt, bu, bv, bw, bx, by, bz, ca, cb, cc, cd,
  ce, cf, cg, ch, ci, cj, ck } =
{ 64, 63, 36, 89, 75, 50, 18, 90, 87, 80, 70, 57, 43, 25, 9, 90, 90, 88, 85, 82, 78, 73, 67, 61, 54, 46, 38, 31, 22, 13, 4, 91, 90, 90, 90, 88, 87, 86,
  84, 83, 81, 79, 77, 73, 71, 69, 65, 62, 59, 56, 52, 48, 44, 41, 37, 33, 28, 24, 20, 15, 11, 7, 2 }

```

FIG. 13E

Transform basis functions of DCT-2, DST-7 and DCT-8 for N-point input

Transform Type	Basis function $T_i(j)$, $i, j=0, 1, \dots, N-1$
DCT-2	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-8	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \cos\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N + 2}\right)$
DST-7	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (j + 1)}{2N + 1}\right)$

FIG. 14

mts_idx	Transform Type Horizontal	Transform type Vertical	
-1	0	0	1501
0	1	1	1502
1	2	1	1503
2	1	2	1504
3	2	2	1505

1500

Transform Type
0 – DCT2
1 – DST7
2 – DCT8

FIG. 15

4-point DST-7:

$$\begin{aligned} &\{a, b, c, d\} \\ &\{c, c, 0, -c\} \\ &\{d, -a, -c, b\} \\ &\{b, -d, c, -a\} \end{aligned}$$

where $\{a, b, c, d\} = \{29, 55, 74, 84\}$

FIG. 16A8-point DST-7:

$$\begin{aligned} &\{a, b, c, d, e, f, g, h\} \\ &\{c, f, h, e, b, -a, -d, -g\} \\ &\{e, g, b, -c, -h, -d, a, f\} \\ &\{g, c, -d, -f, a, h, b, -e\} \\ &\{h, -a, -g, b, f, -c, -e, d\} \\ &\{f, -e, -a, g, -d, -b, h, -c\} \\ &\{d, -h, e, -a, -c, g, -f, b\} \\ &\{b, -d, f, -h, g, -e, c, -a\} \end{aligned}$$

where $\{a, b, c, d, e, f, g, h\} = \{17, 32, 46, 60, 71, 78, 85, 86\}$

FIG. 16B16-point DST-7:

$$\begin{aligned} &\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p\} \\ &\{c, f, i, l, o, h, c, -b, -g, -l, i, f, c, 0, -c, -f, -a, d, i, n\} \\ &\{e, j, o, m, h, c, -b, -g, -l, i, f, c, 0, -c, -f, -a, d, i, n\} \\ &\{g, n, l, e, -b, -l, -c, f, o, i, 0, -i, -o, -f, c, l\} \\ &\{i, o, f, -c, -l, -k, 0, k, 0, -k, 0, k, 0, -k\} \\ &\{k, k, 0, -k, 0, k, i, -d, -p, -c, j\} \\ &\{o, g, -f, -n, -a, l, i, -f, -l, c, o, 0, -o, -c, l, f, -i\} \\ &\{p, -a, -o, b, n, -c, -m, d, l, -e, -k, f, j, -g, -i, h\} \\ &\{n, -e, -i, j, d, -o, a, m, -f, -h, k, c, -p, b, l, -g\} \\ &\{l, -i, -c, o, -f, f, d, -n, i, a, -k, l, -b, -h, o, -e\} \\ &\{j, -m, c, g, -p, f, d, -n, i, a, -k, l, -b, -h, o, -e\} \\ &\{h, -p, i, -a, -g, o, -j, b, f, -n, k, -c, -e, m, -l, d\} \\ &\{f, -l, o, -i, c, c, -i, o, -l, f, 0, -f, l, -o, i, -c\} \\ &\{d, -h, l, -p, m, -i, e, -a, -c, g, -k, o, -n, j, -f, b\} \\ &\{b, -d, f, -h, j, -l, n, -p, o, -m, k, -i, g, -e, c, -a\} \end{aligned}$$

where $\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p\} = \{9, 17, 25, 33, 41, 49, 56, 62, 66, 72, 77, 81, 83, 87, 89, 90\}$

FIG. 16C

32-point DST-7:

{ a,	b,	c,	d,	e,	f,	g,	h,	i,	j,	k,	l,	m,	n,	o,	p,	q,	r,	s,	t,	u,	v,	w,	x,	y,	z,	A,	B,	C,	D,	E,	F,	
{ c,	f,	i,	l,	o,	t,	y,	d,	g,	a,	e,	h,	k,	f,	j,	m,	r,	u,	x,	b,	i,	n,	q,	v,	z,	A,	B,	C,	D,	E,	F,		
{ e,	j,	o,	b,	w,	p,	i,	b,	y,	e,	l,	s,	z,	u,	h,	a,	c,	f,	o,	t,	y,	d,	g,	a,	e,	h,	k,	f,	j,	m,	r,	u,	
{ i,	x,	A,	C,	t,	k,	g,	h,	q,	p,	-a,	-v,	-m,	-d,	-e,	-j,	-o,	-b,	-w,	-p,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,		
{ k,	v,	F,	u,	j,	-a,	-l,	-w,	-z,	-m,	-o,	-i,	-h,	-q,	-p,	-e,	-t,	-y,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	
{ m,	z,	z,	e,	-j,	-v,	-k,	f,	w,	x,	f,	-o,	-h,	-q,	-p,	-e,	-t,	-y,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	
{ o,	d,	n,	-c,	-t,	-e,	-j,	-v,	-k,	f,	w,	x,	f,	-o,	-h,	-q,	-p,	-e,	-t,	-y,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	
{ q,	z,	n,	-k,	-y,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	-r,	
{ s,	A,	h,	-k,	-y,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	-r,	
{ u,	w,	h,	-k,	-y,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	-r,	
{ w,	s,	-d,	-A,	-o,	-e,	-t,	-y,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	
{ Y,	o,	-j,	-D,	-e,	-t,	-y,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,		
{ A,	k,	-p,	-v,	-e,	-t,	-y,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,		
{ C,	g,	-v,	-f,	-y,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	-r,	
{ E,	a,	-e,	-b,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	-r,		
{ F,	-a,	-e,	-b,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	-r,		
{ D,	-e,	-b,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	-r,			
{ B,	-i,	-s,	r,	j,	-t,	-y,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,		
{ z,	m,	-m,	z,	o,	-z,	m,	-m,	z,	o,	-z,	m,	-m,	z,	o,	-z,	m,	-m,	z,	o,	-z,	m,	-m,	z,	o,	-z,	m,	-m,	z,	o,	-z,	m,	
{ x,	-g,	-y,	E,	-j,	-b,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	
{ v,	-u,	-a,	w,	-t,	-b,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	
{ t,	-y,	e,	o,	-d,	j,	-b,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,
{ p,	-e,	k,	g,	-y,	v,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	
{ r,	-e,	k,	g,	-y,	v,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	-n,	-q,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	
{ n,	-B,	w,	-i,	-e,	s,	-e,	s,	-e,	s,	-e,	s,	-e,	s,	-e,	s,	-e,	s,	-e,	s,	-e,	s,	-e,	s,	-e,	s,	-e,	s,	-e,	s,	-e,	s,	
{ l,	-x,	C,	-q,	e,	g,	-e,	g,	-e,	g,	-e,	g,	-e,	g,	-e,	g,	-e,	g,	-e,	g,	-e,	g,	-e,	g,	-e,	g,	-e,	g,	-e,	g,	-e,	g,	
{ j,	-t,	D,	-y,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	
{ h,	-p,	x,	-f,	Y,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	-e,	o,	
{ f,	-l,	r,	-s,	D,	-c,	w,	-g,	k,	-e,	-a,	g,	-m,	s,	-y,	E,	-j,	-b,	-d,	-g,	-a,	-e,	-h,	-k,	-f,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	
{ d,	-h,	l,	-p,	t,	-x,	B,	-E,	C,	-y,	u,	-q,	m,	-i,	e,	-a,	z,	-v,	-z,	-A,	-B,	-C,	-D,	-E,	-F,	-j,	-m,	-r,	-u,	-x,	-b,	-i,	
{ b,	-d,	f,	-b,	j,	-l,	n,	-p,	r,	-t,	v,	-x,	z,	-B,	-E,	-C,	A,	-y,	w,	-u,	-q,	o,	-s,	-g,	o,	-m,	k,	-i,	g,	-e,	c,	-a,	

where {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F} = {4, 9, 13, 17, 21, 26, 30, 34, 38, 42, 45, 50, 53, 56, 60, 63, 66, 68, 72, 74, 77, 78, 80, 82, 84, 85, 86, 88, 89, 90, 90 }

FIG. 16D

32-point DCT-8:

{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F } =
 { b, e, h, k, n, q, t, w, z, A, B, C, D, E, F }
 { c, h, m, r, y, F, A, -t, -w, -z, -A, -B, -C, -D, -E, -F }
 { d, k, r, y, F, A, -t, -w, -z, -A, -B, -C, -D, -E, -F }
 { e, n, w, F, A, -t, -w, -z, -A, -B, -C, -D, -E, -F }
 { f, q, B, -a, -p, -e, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { g, t, -t, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { h, w, -B, -a, -p, -e, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { i, z, -w, -f, -i, -l, -o, -r, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { j, C, -t, -w, -f, -i, -l, -o, -r, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { k, F, -m, -i, -d, -o, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { l, -E, -h, -p, A, d, t, -w, -z, -A, -B, -C, -D, -E, -F }
 { m, -y, -C, -D, i, s, -t, -w, -z, -A, -B, -C, -D, -E, -F }
 { n, -y, -C, -D, i, s, -t, -w, -z, -A, -B, -C, -D, -E, -F }
 { o, -v, -h, C, a, B, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { p, -s, -m, v, j, -y, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { q, -p, -r, o, s, -t, -w, -z, -A, -B, -C, -D, -E, -F }
 { r, -m, -w, h, B, -c, -o, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { s, -j, -B, a, -C, -i, -t, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { t, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { u, -d, B, n, -k, -E, g, -r, -z, -A, -B, -C, -D, -E, -F }
 { v, -a, w, u, B, -h, -c, -o, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { w, -c, r, B, -h, -c, -o, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { x, -f, m, -E, -q, -z, -j, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { y, -l, h, -x, -z, -j, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { z, -l, C, -g, E, u, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { A, -o, C, -j, v, F, -t, -w, -z, -A, -B, -C, -D, -E, -F }
 { B, -r, h, -c, m, -w, -z, -j, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { C, -u, m, -e, d, -l, -t, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { D, -x, z, -l, f, -a, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { E, -A, w, -s, o, -k, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }
 { F, -D, B, -z, x, -v, -t, -g, -t, -u, -v, -w, -x, -y, -z, -A, -B, -C, -D, -E, -F }

where { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F } =
 { 90, 90, 89, 88, 86, 85, 84, 82, 80, 78, 77, 74, 72, 68, 66, 63, 60, 56, 53, 50, 45, 42, 38, 34, 30, 26, 21, 17, 13, 9, 4 }

FIG. 17D

seq_parameter_set_rbsp() { sps_mts_enabled_flag if(sps_mts_enabled_flag) { sps_explicit_mts_intra_enabled_flag sps_explicit_mts_inter_enabled_flag }	Descriptor u(1) u(1) u(1)
--	--

FIG. 18

Mapping from intra prediction mode to transform set index

1900

intra mode	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
set	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
intra mode	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67 (LM)
set	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	NULL

FIG. 19

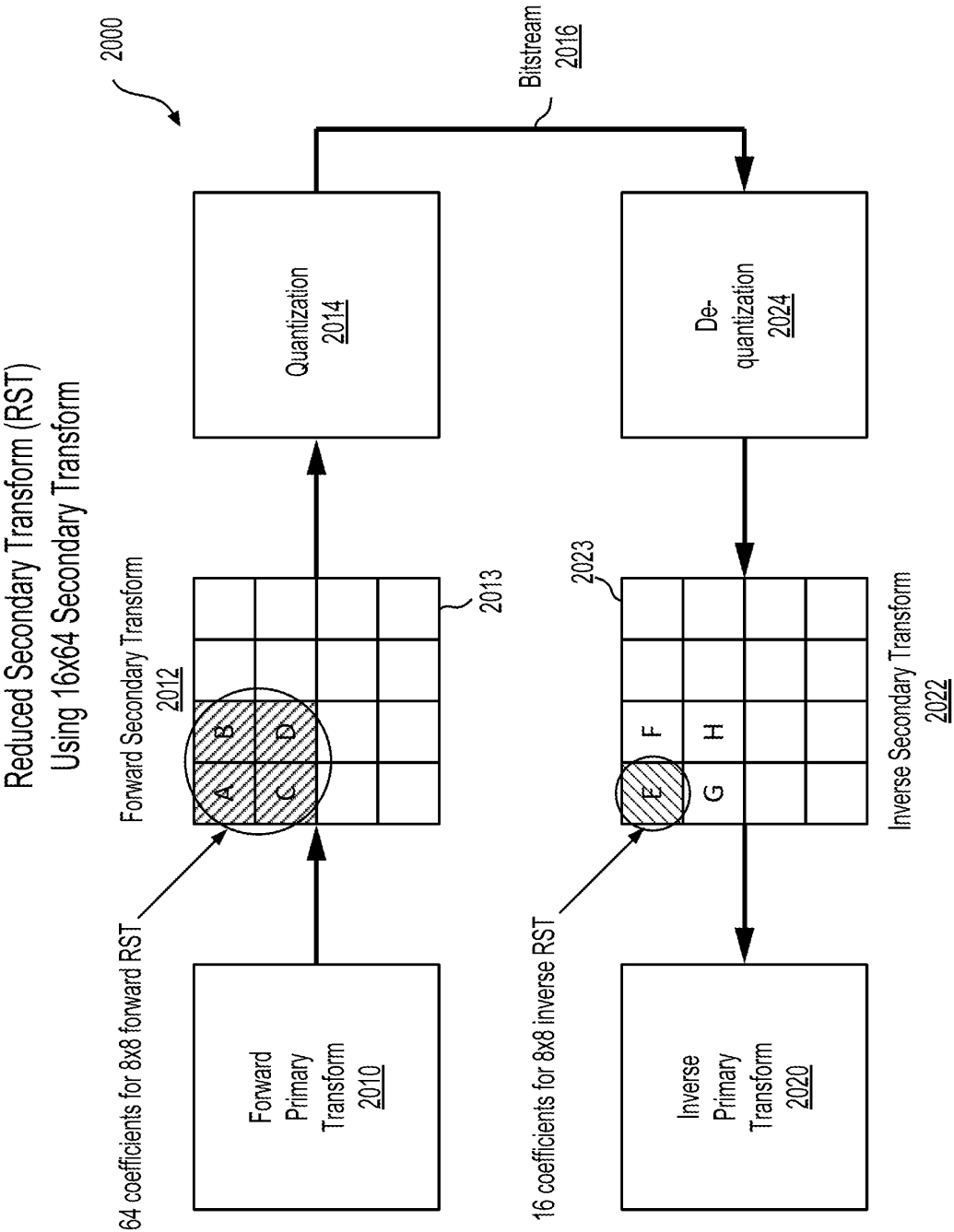


FIG. 20

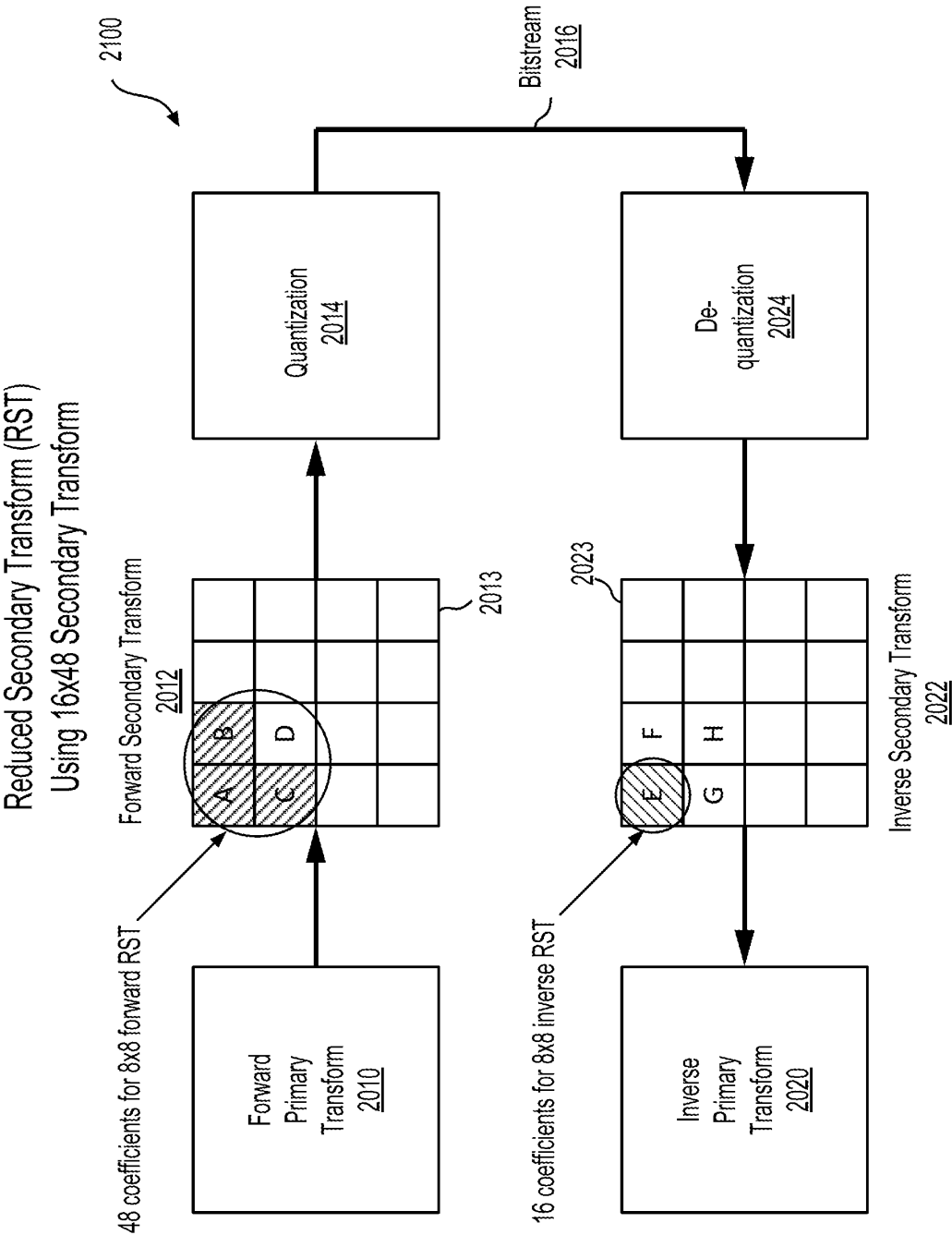


FIG. 21

Descriptor	
	coding_init(x0, y0, cbWidth, cbHeight, treeType) {
	numSigCoeff = 0
	numZeroOutSigCoeff = 0
	transform_tree(x0, y0, cbWidth, cbHeight, treeType)
	lfstWidth = (treeType == DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : cbWidth
	lfstHeight = (treeType == DUAL_TREE_CHROMA) ? cbHeight / SubHeightC : cbHeight
	if(Min(lfstWidth, lfstHeight) >= 4 && sps_lfst_enabled_flag == 1 && CuPredMode[x0][y0] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT && !intra_mip_flag[x0][y0]) {
	if((numSigCoeff > ((treeType == SINGLE_TREE) ? 2 : 1)) && numZeroOutSigCoeff == 0)
	lfst_idx [x0][y0]
	}
	}
	}
	ae(v)

2200

FIG. 22



Forward and Inverse Reduced Transform

FIG. 23

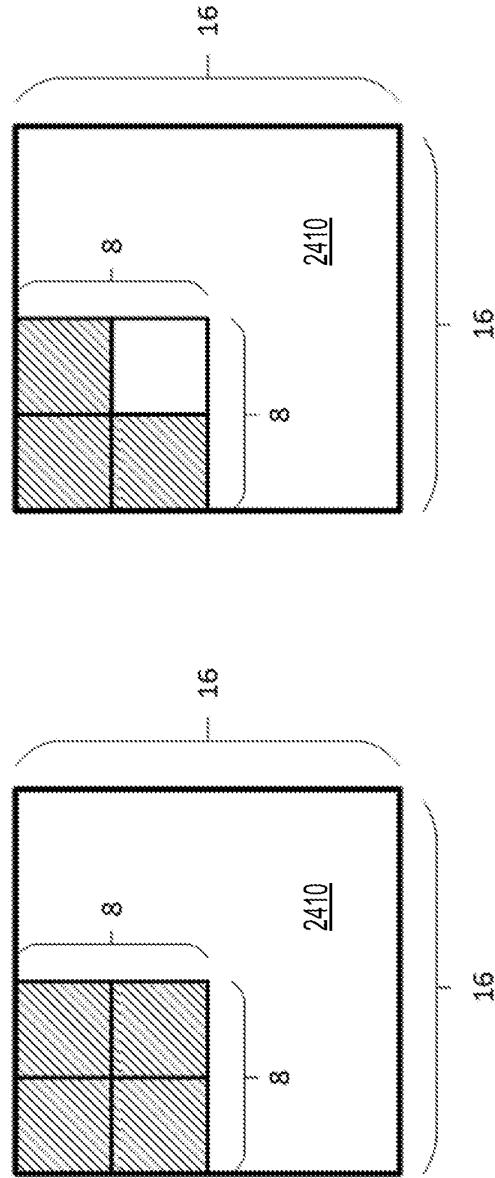


FIG. 24A

FIG. 24B

Transform Set Selection Table

IntraPredMode	Tr. set index
IntraPredMode < 0	1
0 <= IntraPredMode <= 1	0
2 <= IntraPredMode <= 12	1
13 <= IntraPredMode <= 23	2
24 <= IntraPredMode <= 44	3
45 <= IntraPredMode <= 55	2
56 <= IntraPredMode	1

FIG. 25

2600

Linear Interpolation
2603

Matrix Vector Multiplication
2602

Averaging
2601

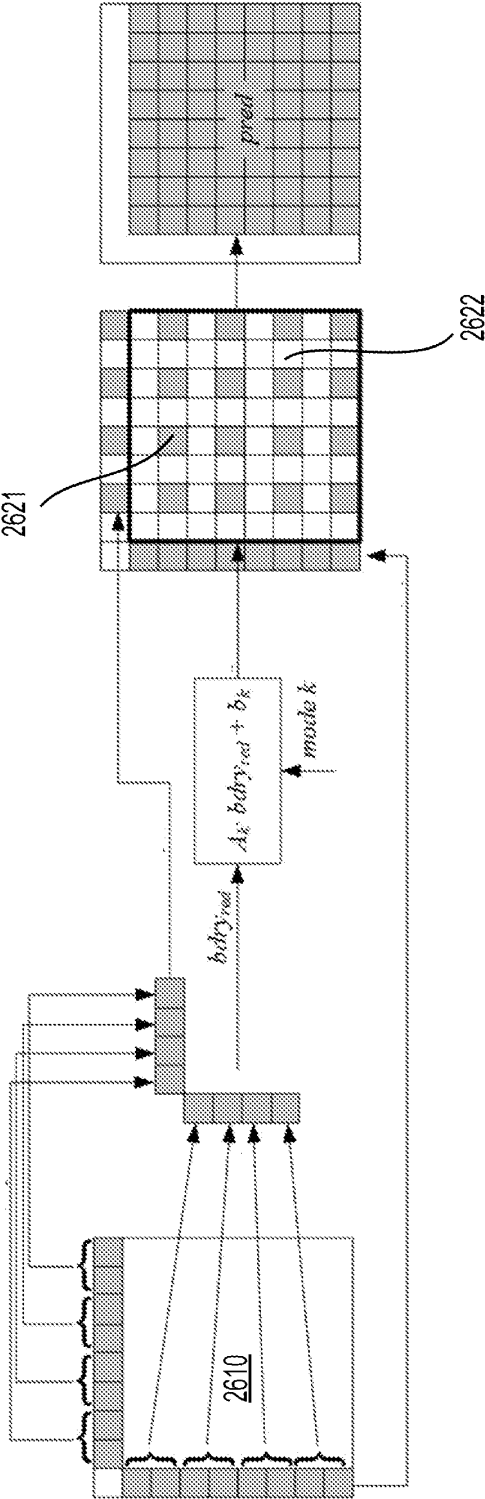
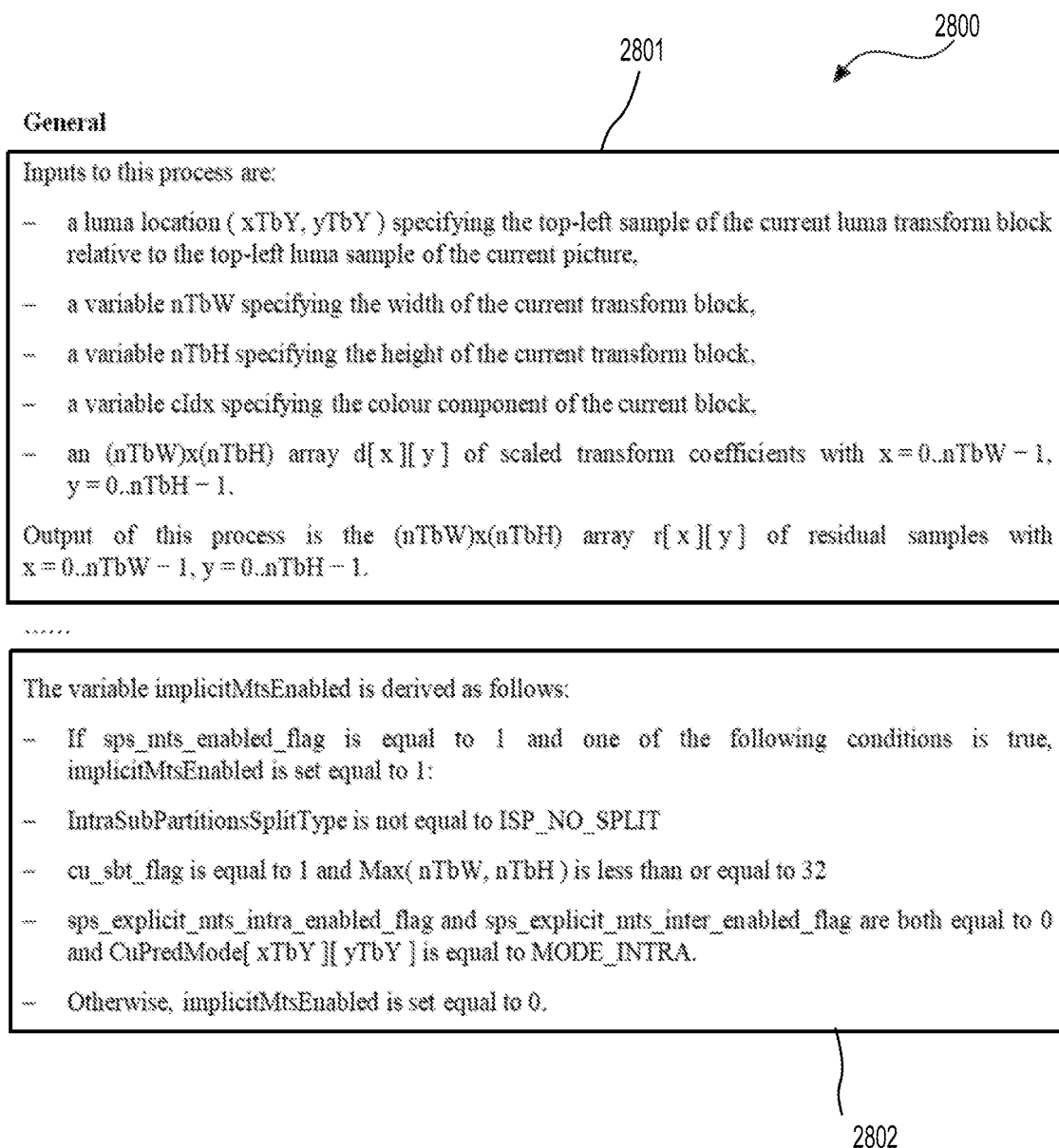


FIG. 26

2701

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
.....	
if(sps_mip_enabled_flag && (Abs(Log2(cbWidth) - Log2(cbHeight)) <= 2) && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_mip_flag[x0][y0]	ae(v)
if(intra_mip_flag[x0][y0]) {	
intra_mip_mpm_flag[x0][y0]	ae(v)
if(intra_mip_mpm_flag[x0][y0])	
intra_mip_mpm_idx[x0][y0]	ae(v)
else	
intra_mip_mpm_remainder[x0][y0]	ae(v)
} else {	
if(sps_mrl_enabled_flag && ((y0 % CtbSizeY) > 0))	
intra_luma_ref_idx[x0][y0]	ae(v)
if(sps_isp_enabled_flag && intra_luma_ref_idx[x0][y0] == 0 && (cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag[x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] == 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag[x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0]) {	
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_not_planar_flag[x0][y0]	ae(v)
if(intra_luma_not_planar_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
}	
}	
.....	

FIG. 27

**FIG. 28A**

2803

Continue from FIG. 28A

2800

The variable trTypeHor specifying the horizontal transform kernel and the variable trTypeVer specifying the vertical transform kernel are derived as follows:

- If cldx is greater than 0, trTypeHor and trTypeVer are set equal to 0.
- Otherwise, if implicitMtsEnabled is equal to 1, the following applies:
- If IntraSubPartitionsSplitType is not equal to ISP_NO_SPLIT or both sps_explicit_mts_intra_enabled_flag and sps_explicit_mts_inter_enabled_flag are equal to 0 and CuPredMode[xTbY][yTbY] is equal to MODE_INTRA, trTypeHor and trTypeVer are derived as follows:.

$$\text{trTypeHor} = (\text{nTbW} \geq 4 \ \&\& \ \text{nTbW} \leq 16) ? 1 : 0 \quad (3-1)$$

$$\text{trTypeVer} = (\text{nTbH} \geq 4 \ \&\& \ \text{nTbH} \leq 16) ? 1 : 0 \quad (3-2)$$

- Otherwise (cu_sbt_flag is equal to 1), trTypeHor and trTypeVer are specified in Table II depending on cu_sbt_horizontal_flag and cu_sbt_pos_flag.
- Otherwise, trTypeHor and trTypeVer are specified in Table I depending on tu_mts_idx[xTbY][yTbY].

Table I – Specification of trTypeHor and trTypeVer depending on tu_mts_idx[x][y]

tu_mts_idx[x0][y0]	0	1	2	3	4
trTypeHor	0	1	2	1	2
trTypeVer	0	1	1	2	2

Table II – Specification of trTypeHor and trTypeVer depending on cu_sbt_horizontal_flag and cu_sbt_pos_flag

cu_sbt_horizontal_flag	cu_sbt_pos_flag	trTypeHor	trTypeVer
0	0	2	1
0	1	1	1
1	0	1	2
1	1	1	1

FIG. 28B

2900

The variable `implicitMtsEnabled` is derived as follows:

- If `sps_mts_enabled_flag` is equal to 1 and one of the following conditions is true, `implicitMtsEnabled` is set equal to 1:
- `IntraSubPartitionsSplitType` is not equal to `ISP_NO_SPLIT`
- `cu_sbt_flag` is equal to 1 and `Max( nTbW, nTbH )` is less than or equal to 32
- `sps_explicit_mts_intra_enabled_flag` and `sps_explicit_mts_inter_enabled_flag` are both equal to 0 and `CuPredMode[ xTbY ][ yTbY ]` is equal to `MODE_INTRA`.
- Otherwise, `implicitMtsEnabled` is set equal to 0.

FIG. 29

3000

The variable `implicitMtsEnabled` is derived as follows:

- If `sps_mts_enabled_flag` is equal to 1 and one of the following conditions is true, `implicitMtsEnabled` is set equal to 1:
- `IntraSubPartitionsSplitType` is not equal to `ISP_NO_SPLIT`
- `cu_sbt_flag` is equal to 1 and `Max( nTbW, nTbH )` is less than or equal to 32
- `sps_explicit_mts_intra_enabled_flag` and `sps_explicit_mts_inter_enabled_flag` are both equal to 0 and `CuPredMode[ xTbY ][ yTbY ]` is equal to `MODE_INTRA` and `sps_first_enabled_flag` is equal to 0.
- Otherwise, `implicitMtsEnabled` is set equal to 0.

FIG. 30

3100

The variable `implicitMtsEnabled` is derived as follows:

- If `sps_mts_enabled_flag` is equal to 1 and one of the following conditions is true, `implicitMtsEnabled` is set equal to 1:
 - `IntraSubPartitionsSplitType` is not equal to `ISP_NO_SPLIT`
 - `cu_sbt_flag` is equal to 1 and `Max( nTbW, nTbH )` is less than or equal to 32
 - `sps_explicit_mts_intra_enabled_flag` and `sps_explicit_mts_inter_enabled_flag` are both equal to 0 and `CuPredMode[ xTbY ][ yTbY ]` is equal to `MODE_INTRA` and `sps_mip_enabled_flag` is equal to 0.
- Otherwise, `implicitMtsEnabled` is set equal to 0.

FIG. 31

3200

The variable `implicitMtsEnabled` is derived as follows:

- If `sps_mts_enabled_flag` is equal to 1 and one of the following conditions is true, `implicitMtsEnabled` is set equal to 1:
 - `IntraSubPartitionsSplitType` is not equal to `ISP_NO_SPLIT`
 - `cu_sbt_flag` is equal to 1 and `Max( nTbW, nTbH )` is less than or equal to 32
 - `sps_explicit_mts_intra_enabled_flag` and `sps_explicit_mts_inter_enabled_flag` are both equal to 0 and `CuPredMode[ xTbY ][ yTbY ]` is equal to `MODE_INTRA`.
 - `tu_mts_idx[ x0 ][ y0 ]` and `lfmst_idx[ x0 ][ y0 ]` are both equal to 0.
- Otherwise, `implicitMtsEnabled` is set equal to 0.

FIG. 32

The variable `implicitMtsEnabled` is derived as follows:

- If `sps_mts_enabled_flag` is equal to 1 and one of the following conditions is true, `implicitMtsEnabled` is set equal to 1:
 - `IntraSubPartitionsSplitType` is not equal to `ISP_NO_SPLIT`
 - `cu_sbt_flag` is equal to 1 and $\text{Max}(nTbW, nTbH)$ is less than or equal to 32
 - ~~`sps_explicit_mts_intra_enabled_flag` and `sps_explicit_mts_inter_enabled_flag` are both equal to 0 and `CuPredMode[xTbY][yTbY]` is equal to `MODE_INTRA`.~~
 - `tu_mts_idx[x0][y0]`, `intra_mip_flag[x0][y0]` and `lfst_idx[x0][y0]` are all equal to 0.
- Otherwise, `implicitMtsEnabled` is set equal to 0.

FIG. 33

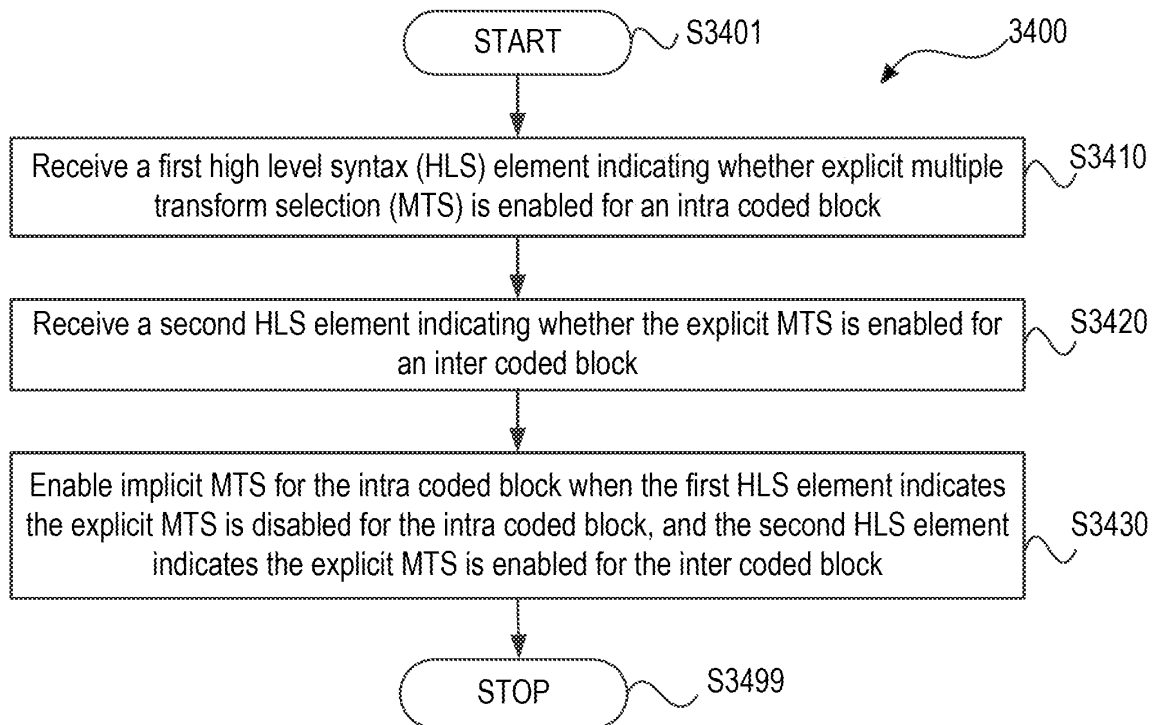
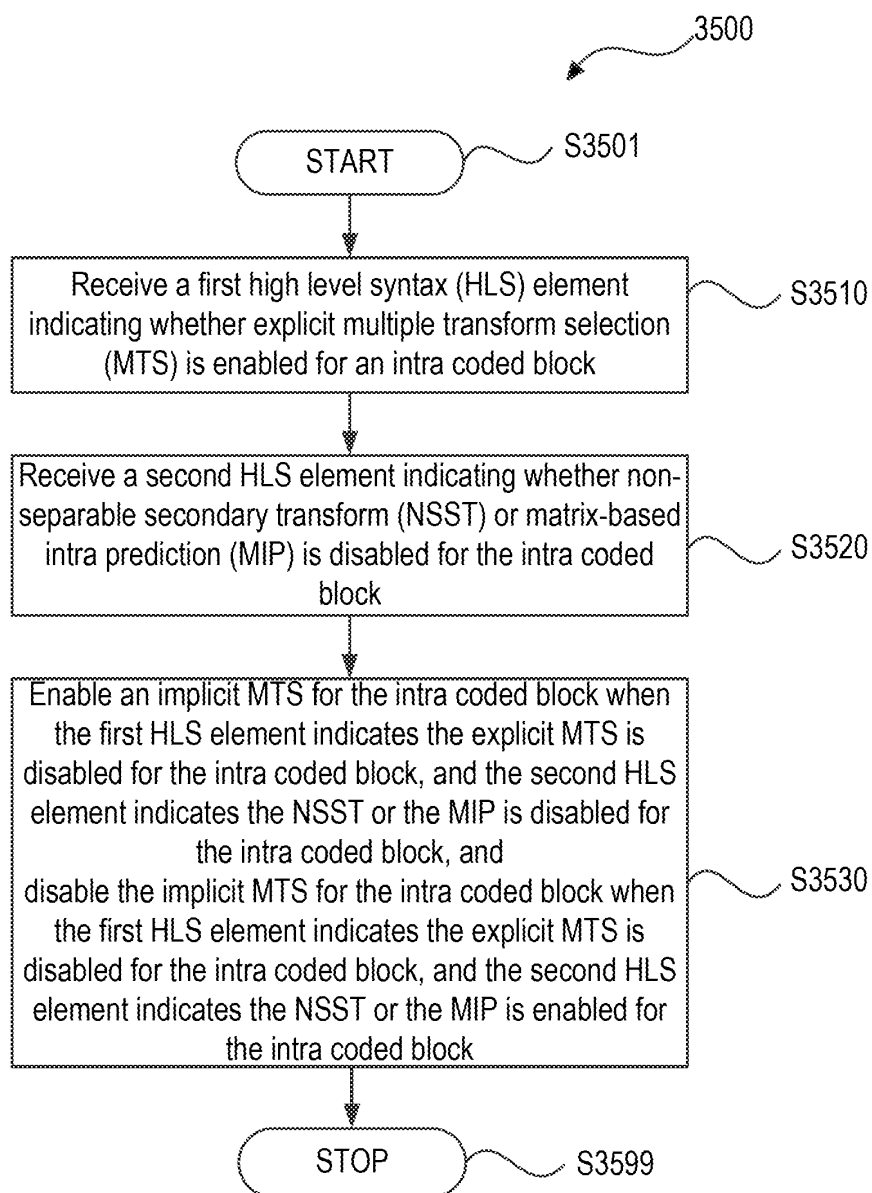
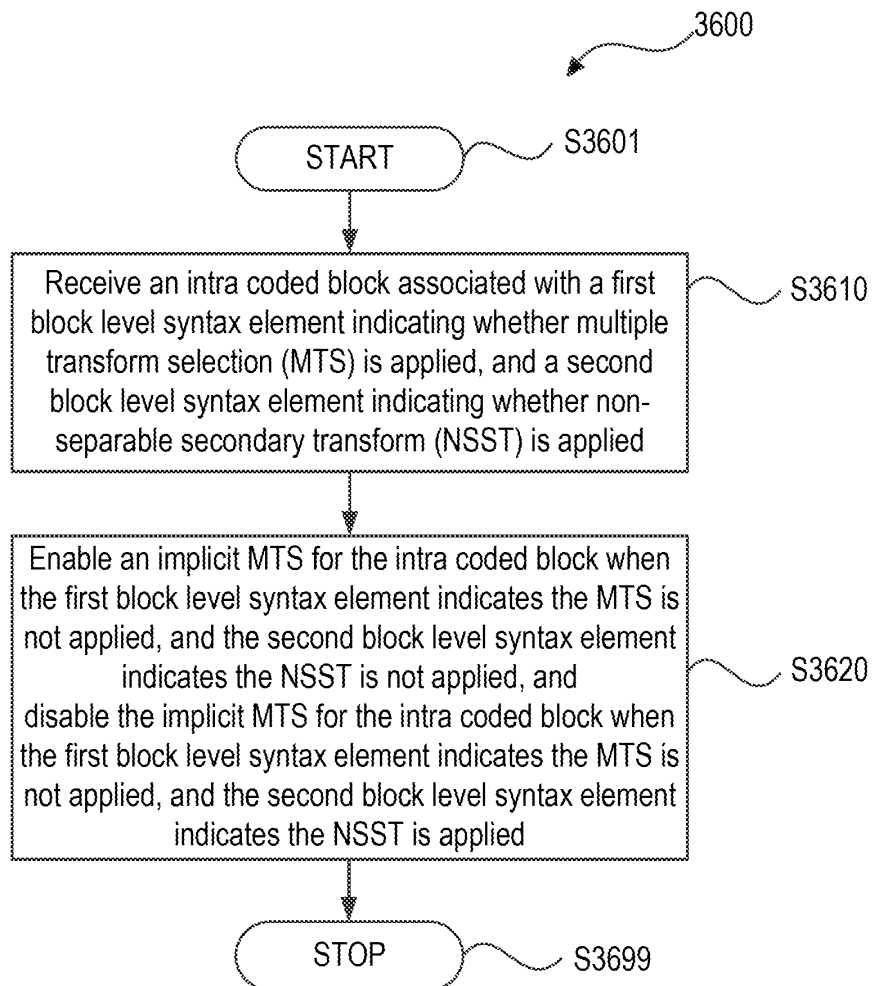


FIG. 34

**FIG. 35**

**FIG. 36**

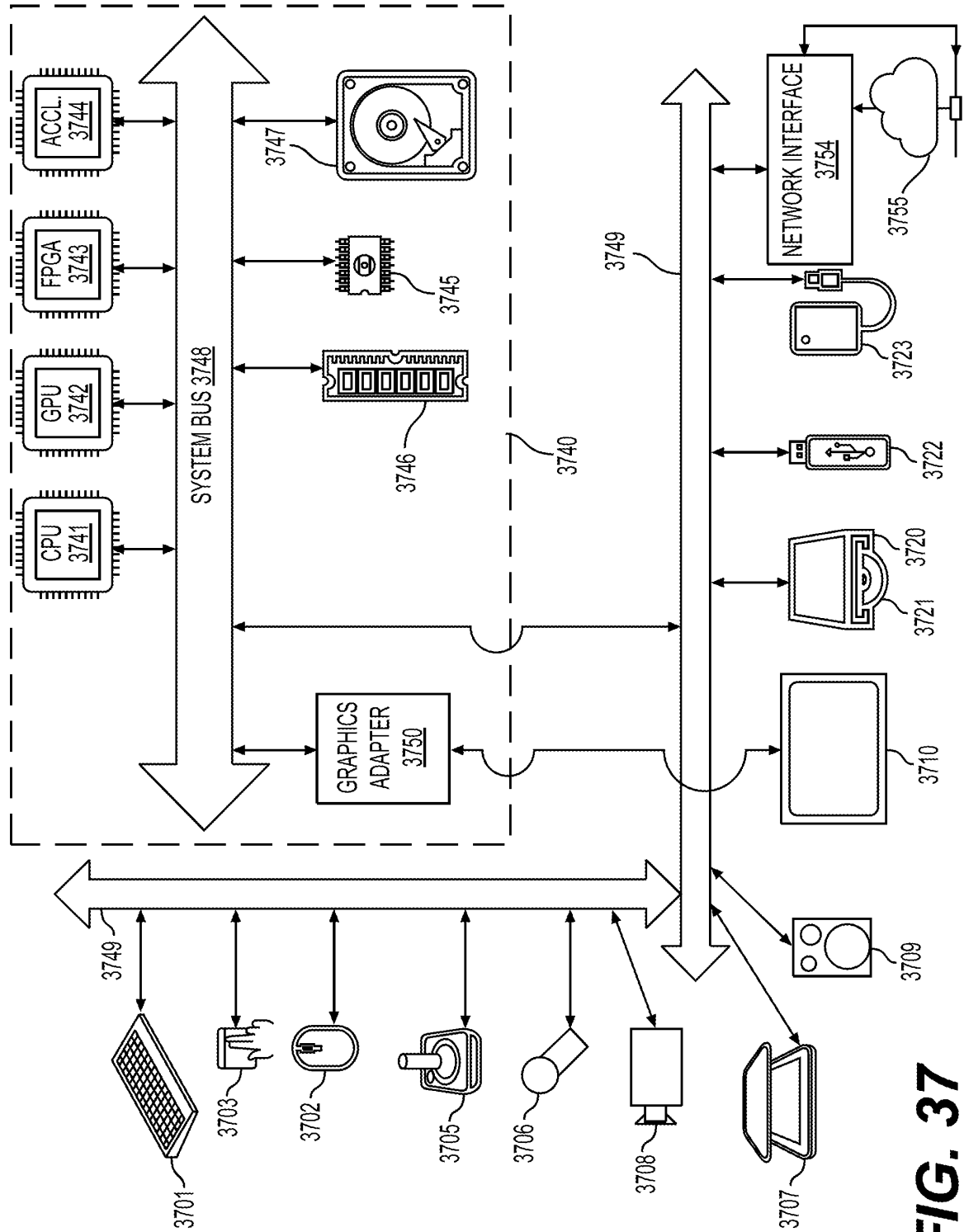


FIG. 37