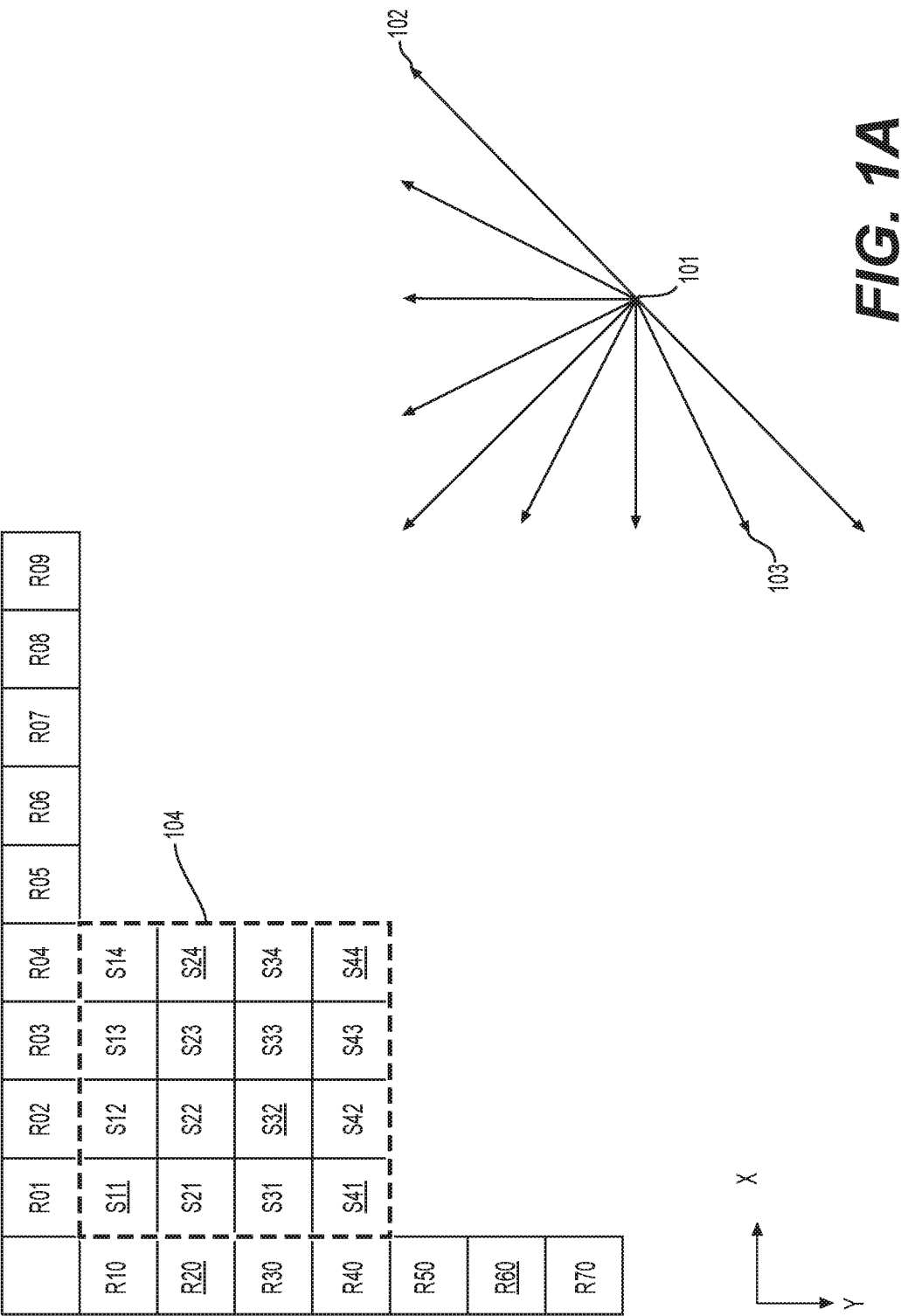




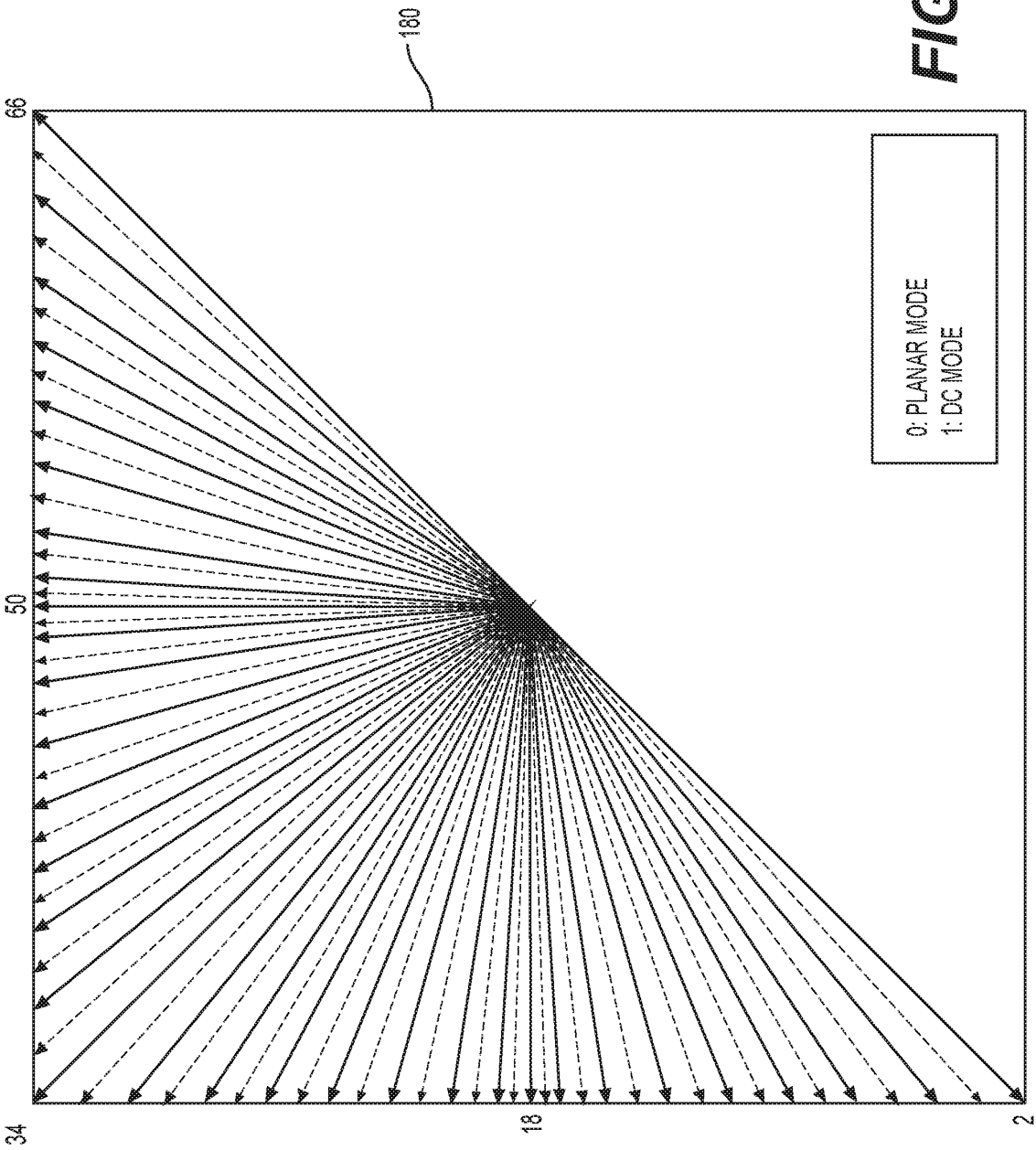


FIG. 1B

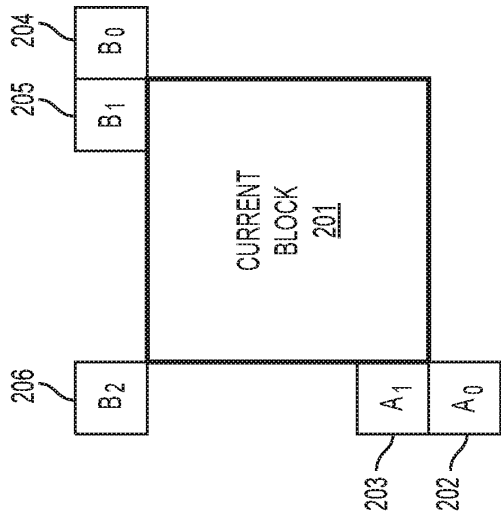


FIG. 2
(Related Art)

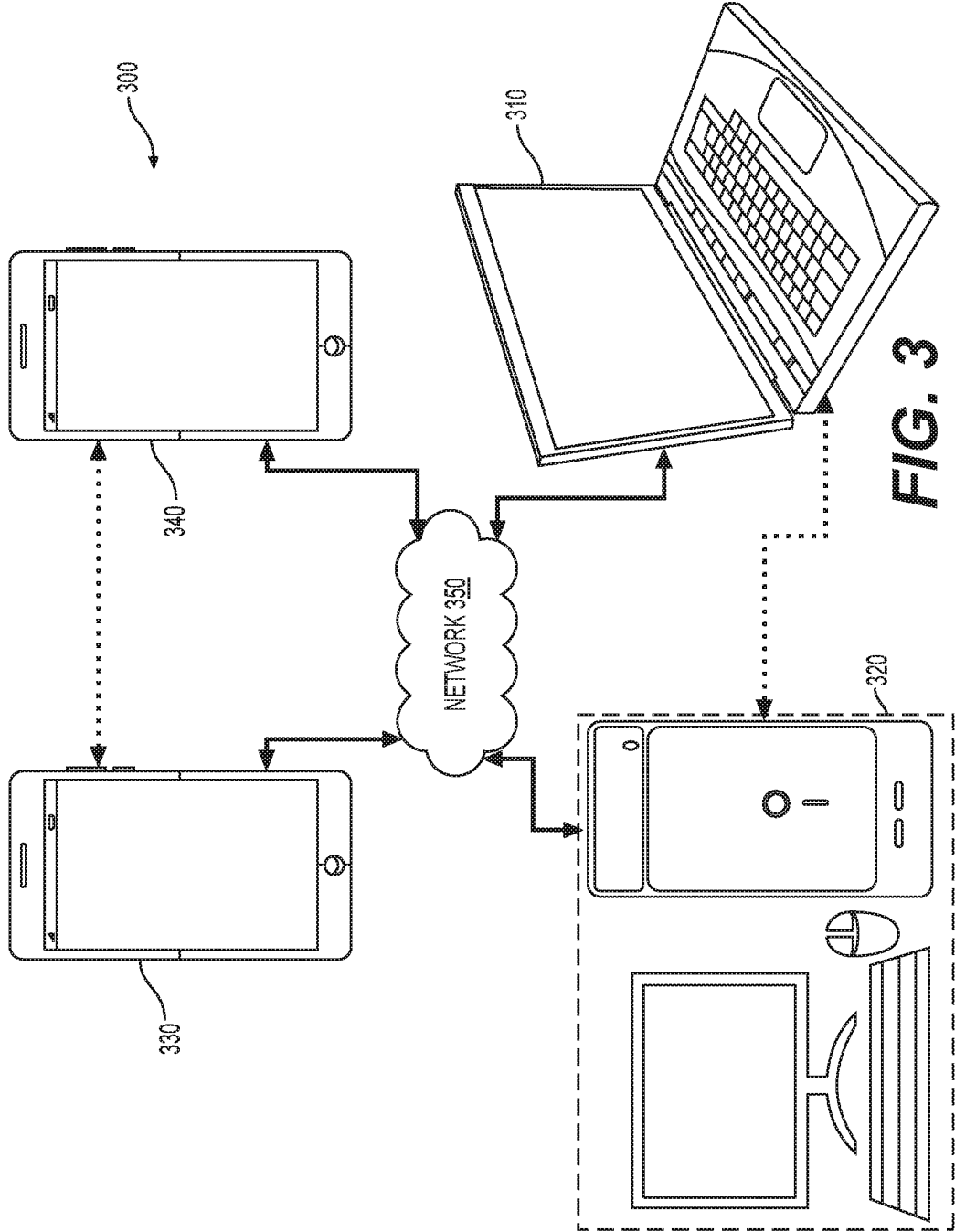


FIG. 3

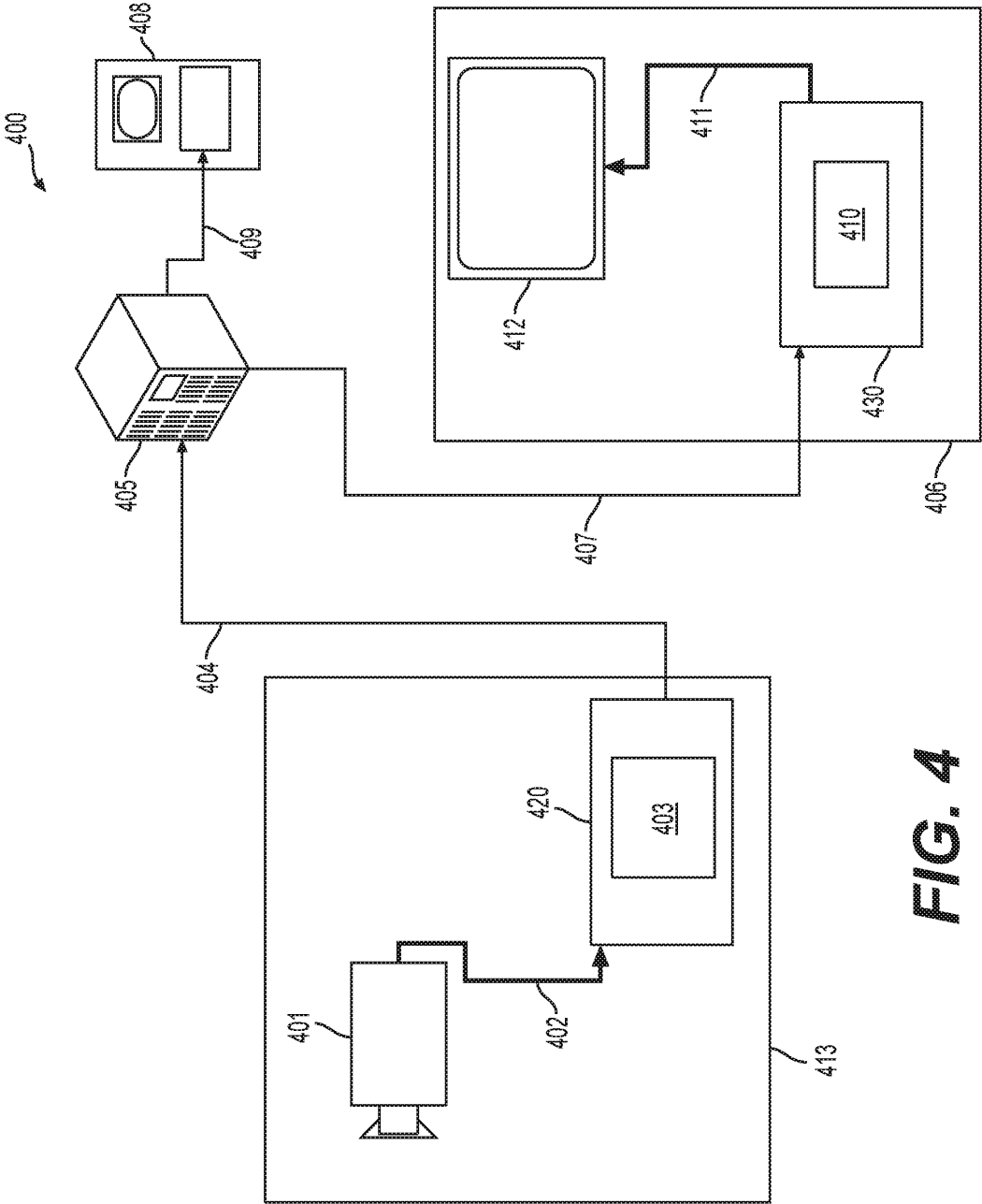


FIG. 4

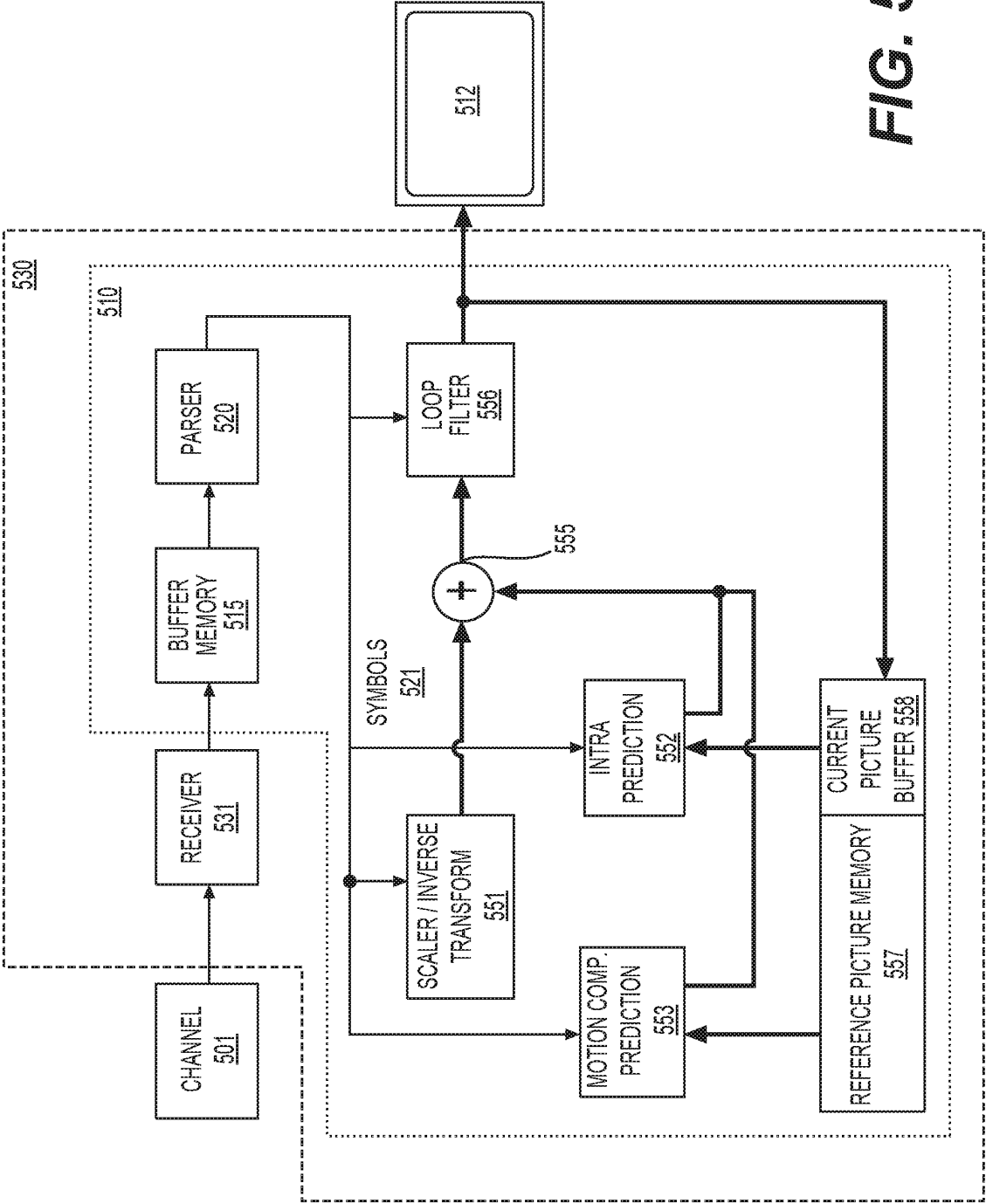


FIG. 5

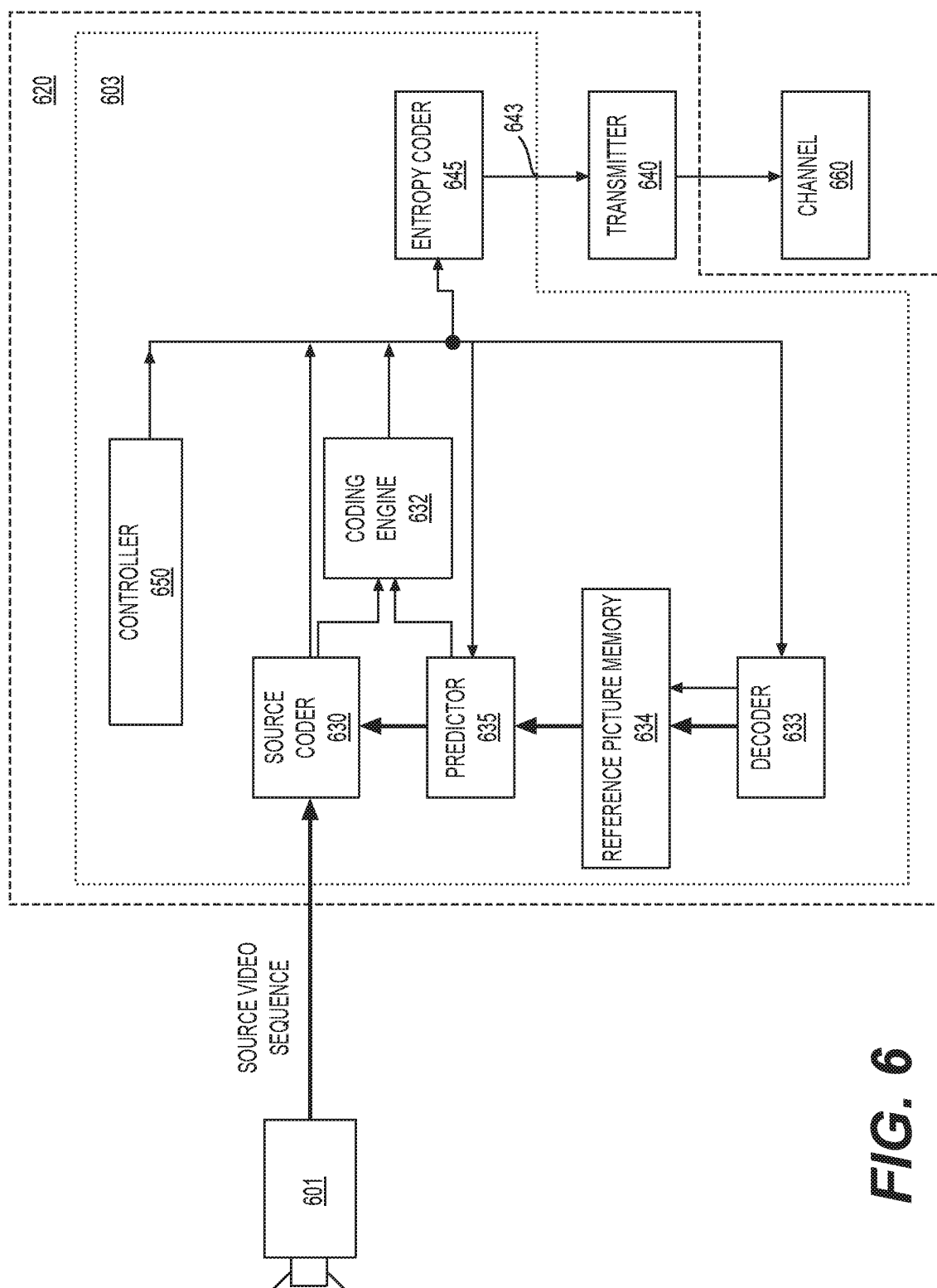


FIG. 6

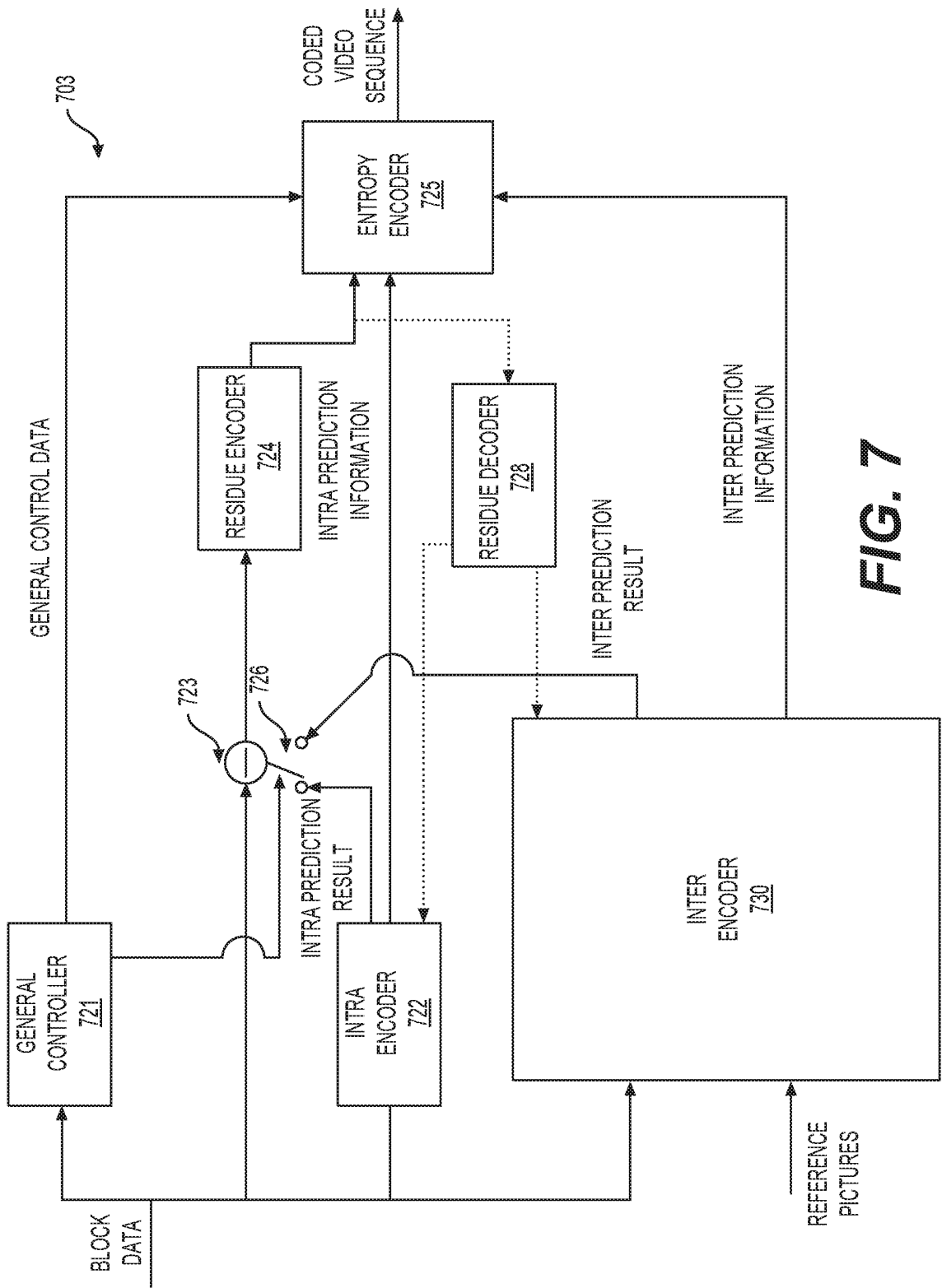


FIG. 7

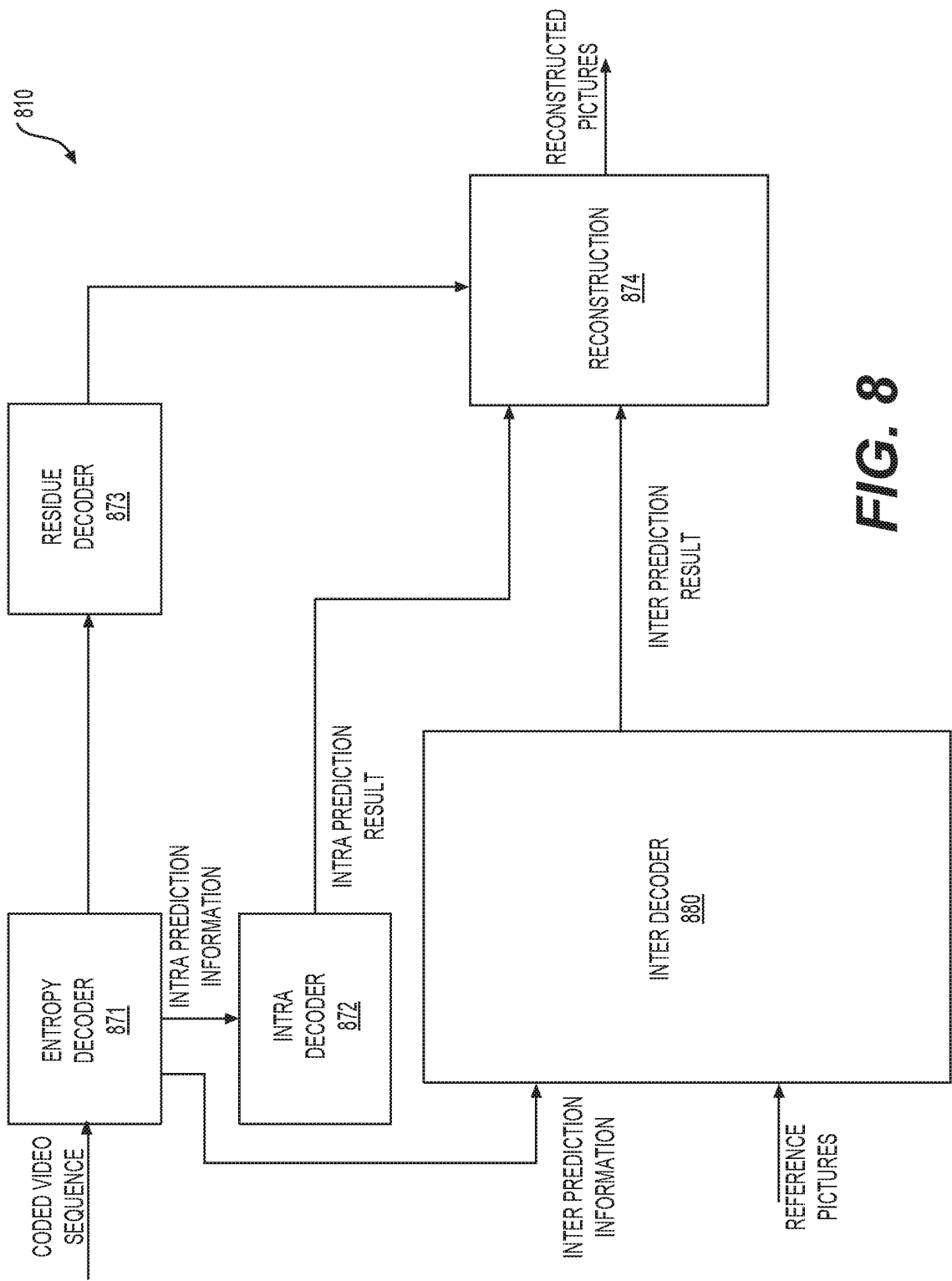


FIG. 8

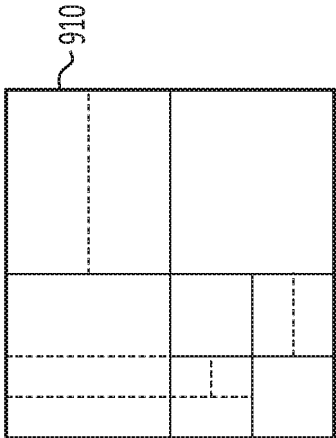


FIG. 9A

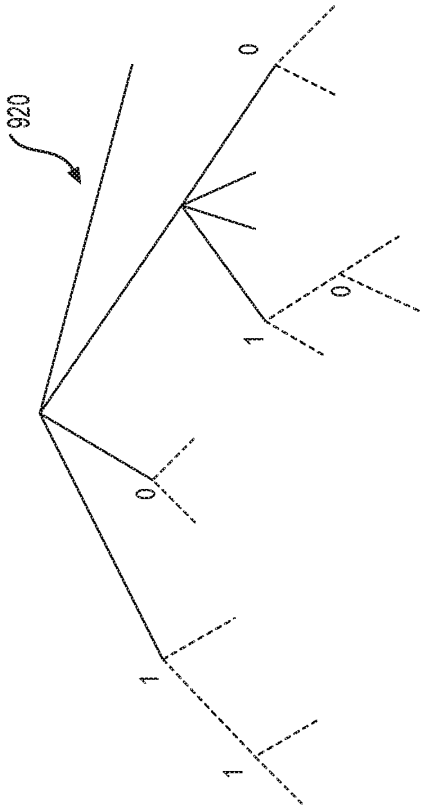


FIG. 9B

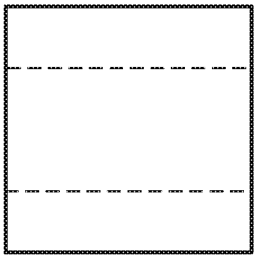


FIG. 9C

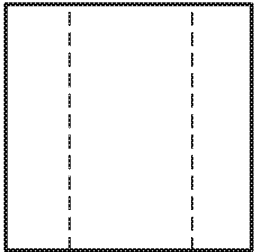


FIG. 9D

4x4 transform

16x16 transform

FIG. 10A

8x8 transform

FIG. 10B

FIG. 10C

Continue from FIG. 11A

bx,cj,bu,bf,bi,-by,-bj,-bp,-ce,cc,bn,bl,ca,-cg,-br,-bh,-bw,ck,bv,bg,bs,ch,-bz,-bk,-bo,-
 cd,cd,bo,bk,bz,-ch,-bs,-bg,-bv,-ck,bw,bh,br,cg,-ca,-bl,-bn,-cc,ce,bp,bj,by,-ci,-bt,-bf,-bu,-
 cj,bx,bi,bq,cf,-cb,-bm}{ab,ac,-ac,-ab,-ab,-ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-
 ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-
 ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab}

{bn,ce,-ca,-bj,-br,-ci,bw,bf,bv,-cj,-bs,-bi,-bz,cf,bo,bm,cd,-cb,-bk,-bq,-ch,bx,bg,bu,-ck,-bt,-
 bh,-by,cg,bp,bl,cc,-cc,-bl,-bp,-cg,by,bh,bi,ck,-bu,-bg,-bx,cb,bq,bk,cb,-cd,-bm,-bo,-
 cf,bz,bi,bs,cj,-bv,-bf,-bw,ci,br,bj,ca,-ce,-bn}{at,bc,-ay,-ap,-ax,bd,au,as,bb,-az,-aq,-
 aw,be,av,ar,ba,-ba,-ar,-av,-be,aw,aq,az,-bb,-as,-au,-bd,ax,ap,ay,-bc,-at,-at,-bc,ay,ap,ax,-
 bd,-au,-as,-bb,az,aq,aw,-be,-av,-ar,-ba,ba,ar,av,be,-aw,-aq,-az,bb,as,au,bd,-ax,-ap,-
 ay,bc,at}

{bo,ch,-bv,-bh,-ca,cc,bj,bi,-cj,-bq,-bm,-cf,bx,bf,by,-ce,-bl,-br,-ck,bs,bk,cd,-bz,-bg,-
 bw,cg,bn,bp,ci,-bu,-bi,-cb,cb,bi,bu,-ci,-bp,-bn,-cg,bw,bg,bz,-cd,-bk,-bs,ck,br,bl,ce,-by,-bf,-
 bx,cf,bm,bq,cj,-bt,-bj,-cc,ca,bh,bv,-ch,-bo}{aj,ao,-ak,-ai,-an,al,ah,am,-am,-ah,-al,an,ai,ak,-
 ao,-aj,-aj,-ao,ak,ai,an,-al,-ah,-am,am,ah,al,-an,-ai,-ak,ao,aj,aj,ao,-ak,-ai,-an,al,ah,am,-am,-
 ah,-al,an,ai,ak,-ao,-aj,-aj,-ao,ak,ai,an,-al,-ah,-am,am,ah,al,-an,-ai,-ak,ao,aj}{bp,ck,-bq,-bo,-
 cj,br,bn,ci,-bs,-bm,-ch,bi,bl,cg,-bu,-bk,-cf,bv,bj,ce,-bw,-bi,-cd,bx,bh,cc,-by,-bg,-cb,bz,bf,ca,-
 ca,-bf,-bz,cb,bg,by,-cc,-bh,-bx,cd,bi,bw,-ce,-bj,-bv,cf,bk,bu,-cg,-bl,-bt,cb,bm,bs,-ci,-bn,-
 br,cj,bo,bq,-ck,-bp}

{au,-be,-at,-av,bd,as,aw,-bc,-ar,-ax,bb,aq,ay,-ba,-ap,-az,az,ap,ba,-ay,-aq,-bb,ax,ar,bc,-
 aw,-as,-bd,av,at,be,-au,-au,be,at,av,-bd,-as,-aw,bc,ar,ax,-bb,-aq,-ay,ba,ap,az,-az,-ap,-
 ba,ay,aq,bb,-ax,-ar,-bc,aw,as,bd,-av,-at,-be,au}{bq,-ci,-bl,-bv,cd,bg,ca,-by,-bi,-cf,bi,bn,ck,-
 bo,-bs,cg,bj,bx,-cb,-bf,-cc,bw,bk,cb,-br,-bp,cj,bm,bu,-ce,-bh,-bz,bz,bh,ce,-bu,-bm,-cj,bp,br,-
 ch,-bk,-bw,cc,bf,cb,-bx,-bj,-cg,bs,bo,-ck,-bn,-bt,cf,bi,by,-ca,-bg,-cd,bv,bl,ci,-bq}

{ae,-ag,-ad,-af,af,ad,ag,-ae,-ae,ag,ad,af,-af,-ad,-ag,ae,ae,-ag,-ad,-af,af,ad,ag,-ae,-
 ae,ag,ad,af,-af,-ad,-ag,ae,ae,-ag,-ad,-af,af,ad,ag,-ae,-ae,ag,ad,af,-af,-ad,-ag,ae,ae,-ag,-
 ad,-af,af,ad,ag,-ae,-ae,ag,ad,af,-af,-ad,-ag,ae}{br,-cf,-bg,-cc,bu,bo,-ci,-bj,-bz,bx,bl,ck,-bm,-
 bw,ca,bi,ci,-bp,-bt,cd,bf,ce,-bs,-bq,cg,bh,cb,-bv,-bn,cj,bk,by,-by,-bk,-cj,bn,bv,-cb,-bh,-
 cg,bq,bs,-ce,-bf,-cd,bt,bp,-ch,-bi,-ca,bw,bm,-ck,-bl,-bx,bz,bj,ci,-bo,-bu,cc,bg,cf,-br}

{av,-bb,-ap,-bc,au,aw,-ba,-aq,-bd,at,ax,-az,-ar,-be,as,ay,-ay,-as,be,ar,az,-ax,-at,bd,aq,ba,-
 aw,-au,bc,ap,bb,-av,-av,bb,ap,bc,-au,-aw,ba,aq,bd,-at,-ax,az,ar,be,-as,-ay,ay,as,-be,-ar,-
 az,ax,at,-bd,-aq,-ba,aw,au,-bc,-ap,-bb,av}{bs,-cc,-bi,-cj,bl,bz,-bv,-bp,cf,bf,cg,-bo,-
 bw,by,bm,-ci,-bh,-cd,br,bi,-cb,-bj,-ck,bk,ca,-bu,-bq,ce,bg,cb,-bn,-bx,bx,bn,-ch,-bg,-
 ce,bq,bu,-ca,-bk,ck,bj,cb,-bt,-br,cd,bh,ci,-bm,-by,bw,bo,-cg,-bf,-cf,bp,bv,-bz,-bl,cj,bi,cc,-bs}

{ak,-am,-ai,ao,ah,an,-aj,-al,aj,-an,-ah,-ao,ai,am,-ak,-ak,am,ai,-ao,-ah,-an,aj,ai,-al,-
 aj,an,ah,ao,-ai,-am,ak,ak,-am,-ai,ao,ah,an,-aj,-al,aj,-an,-ah,-ao,ai,am,-ak,-ak,am,ai,-ao,-
 ah,-an,aj,ai,-al,-aj,an,ah,ao,-ai,-am,ak}{bt,-bz,-bn,cf,bh,ck,-bi,-ce,bo,by,-bu,-bs,ca,bm,-cg,-
 bg,-cj,bj,cd,-bp,-bx,bv,br,-cb,-bl,cb,bf,ci,-bk,-cc,bq,bw,-bw,-bq,cc,bk,-ci,-bf,-ch,bl,cb,-br,-
 bv,bx,bp,-cd,-bj,cj,bg,cg,-bm,-ca,bs,bu,-by,-bo,ce,bi,-ck,-bh,-cf,bn,bz,-bt}

FIG. 11B

Continue from FIG. 11B

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

FIG. 11C

Continue from FIG. 11C

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

FIG. 11D

Continue from FIG. 11D

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

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}=

{64,83,36,89,75,50,18,90,87,80,70,57,43,25,9,90,90,88,85,82,78,73,67,61,54,46,38,31,22,13,4,91,90,90,90,88,87,86,84,83,81,79,77,73,71,69,65,62,59,56,52,48,44,41,37,33,28,24,20,15,11,7,2}

FIG. 11E

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. 12

mts_idx	Transform Type Horizontal	Transform type Vertical	
-1	0	0	1301
0	1	1	1302
1	2	1	1303
2	1	2	1304
3	2	2	1305

1300

Transform Type
0 - DCT2
1 - DST7
2 - DCT8

FIG. 13

4-point DST-7:

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

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

FIG. 14A8-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. 14B16-point DST-7:

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

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. 14C

32-pointDST-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,r,u,x,A,D,F,C,z,w,t,q,n,k,h,e,b,-a,-d,-g,-j,-m,-p,-s,-v,-y,-B,-
 E,}{e,j,o,t,y,D,D,y,t,o,j,e,0,-e,-j,-o,-t,-y,-D,-D,-y,-t,-o,-j,-
 e,0,e,j,o,t,y,D,}{g,n,u,B,D,w,p,i,b,-e,-l,-s,-z,-F,-y,-r,-k,-d,c,j,q,x,E,A,t,m,f,-a,-h,-o,-
 v,-C,}{i,r,A,C,t,k,b,-g,-p,-y,-E,-v,-m,-d,e,n,w,F,x,o,f,-c,-l,-u,-D,-z,-q,-
 h,a,j,s,B,}{k,v,F,u,j,-a,-l,-w,-E,-t,-i,b,m,x,D,s,h,-c,-n,-y,-C,-r,-g,d,o,z,B,q,f,-e,-p,-A,}
 {m,z,z,m,0,-m,-z,-z,-m,0,m,z,z,m,0,-m,-z,-z,-m,0,m,z,z,m,0,-m,-z,-z,-m,0,m,z,}
 {o,D,t,e,-j,-y,-y,-j,e,t,D,o,0,-o,-D,-t,-e,j,y,y,j,-e,-t,-D,-o,0,o,D,t,e,-j,-y,}
 {q,E,n,-c,-t,-B,-k,f,w,y,h,-i,-z,-v,-e,l,C,s,b,-o,-F,-p,a,r,D,m,-d,-u,-A,-j,g,x,}
 {s,A,h,-k,-D,-p,c,v,x,e,-n,-F,-m,f,y,u,b,-q,-C,-j,i,B,r,-a,-t,-z,-g,l,E,o,-d,-w,}
 {u,w,b,-s,-y,-d,q,A,f,-o,-C,-h,m,E,j,-k,-F,-l,i,D,n,-g,-B,-p,e,z,r,-c,-x,-t,a,v,}
 {w,s,-d,-A,-o,h,E,k,-l,-D,-g,p,z,c,-t,-v,a,x,r,-e,-B,-n,i,F,j,-m,-C,-f,q,y,b,-u,}
 {y,o,-j,-D,-e,t,t,-e,-D,-j,o,y,0,-y,-o,j,D,e,-t,-t,e,D,j,-o,-y,0,y,o,-j,-D,-e,t,}
 {A,k,-p,-v,e,F,f,-u,-q,j,B,a,-z,-l,o,w,-d,-E,-g,t,r,-i,-C,-b,y,m,-n,-x,c,D,h,-s,}
 {C,g,-v,-n,o,u,-h,-B,a,D,f,-w,-m,p,t,-i,-A,b,E,e,-x,-l,q,s,-j,-z,c,F,d,-y,-k,r,}
 {E,c,-B,-f,y,i,-v,-l,s,o,-p,-r,m,u,-j,-x,g,A,-d,-D,a,F,b,-C,-e,z,h,-w,-k,t,n,-q,}
 {F,-a,-E,b,D,-c,-C,d,B,-e,-A,f,z,-g,-y,h,x,-i,-w,j,v,-k,-u,l,t,-m,-s,n,r,-o,-q,p,}
 {D,-e,-y,j,t,-o,-o,t,j,-y,-e,D,0,-D,e,y,-j,-t,o,o,-t,-j,y,e,-D,0,D,-e,-y,j,t,-o,}
 {B,-i,-s,r,j,-A,-a,C,-h,-t,q,k,-z,-b,D,-g,-u,p,l,-y,-c,E,-f,-v,o,m,-x,-d,F,-e,-w,n,}
 {z,-m,-m,z,0,-z,m,m,-z,0,z,-m,-m,z,0,-z,m,m,-z,0,z,-m,-m,z,0,-z,m,m,-z,0,z,-m,}
 {x,-q,-g,E,-j,-n,A,-c,-u,t,d,-B,m,k,-D,f,r,-w,-a,y,-p,-h,F,-i,-o,z,-b,-v,s,e,-C,l,}
 {v,-u,-a,w,-t,-b,x,-s,-c,y,-r,-d,z,-q,-e,A,-p,-f,B,-o,-g,C,-n,-h,D,-m,-i,E,-l,-j,F,-k,}
 {t,-y,e,o,-D,j,j,-D,o,e,-y,t,0,-t,y,-e,-o,D,-j,-j,D,-o,-e,y,-t,0,t,-y,e,o,-D,j,}
 {r,-C,k,g,-y,v,-d,-n,F,-o,-c,u,-z,h,j,-B,s,-a,-q,D,-l,-f,x,-w,e,m,-E,p,b,-t,A,-i,}
 {p,-F,q,-a,-o,E,-r,b,n,-D,s,-c,-m,C,-t,d,l,-B,u,-e,-k,A,-v,f,j,-z,w,-g,-i,y,-x,h,}
 {n,-B,w,-i,-e,s,-F,r,-d,-j,x,-A,m,a,-o,C,-v,h,f,-t,E,-q,c,k,-y,z,-l,-b,p,-D,u,-g,}
 {l,-x,C,-q,e,g,-s,E,-v,j,b,-n,z,-A,o,-c,-i,u,-F,t,-h,-d,p,-B,y,-m,a,k,-w,D,-r,f,}
 {j,-t,D,-y,o,-e,-e,o,-y,D,-t,j,0,-j,t,-D,y,-o,e,e,-o,y,-D,t,-j,0,j,-t,D,-y,o,-e,}
 {h,-p,x,-F,y,-q,i,-a,-g,o,-w,E,-z,r,-j,b,f,-n,v,-D,A,-s,k,-c,-e,m,-u,C,-B,t,-l,d,}
 {f,-l,r,-x,D,-C,w,-q,k,-e,-a,g,-m,s,-y,E,-B,v,-p,j,-d,-b,h,-n,t,-z,F,-A,u,-o,i,-c,}
 {d,-h,l,-p,t,-x,B,-F,C,-y,u,-q,m,-i,e,-a,-c,g,-k,o,-s,w,-A,E,-D,z,-v,r,-n,j,-f,b,}
 {b,-d,f,-h,j,-l,n,-p,r,-t,v,-x,z,-B,D,-F,E,-C,A,-y,w,-u,s,-q,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,
 26,30,34,38,42,45,50,53,56,60,63,66,68,72,74,77,78,80,82,84,85,86,88,88,89,90
 ,90}

FIG. 14D

4-point DCT-8:

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

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

FIG. 15A8-point DCT-8:

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

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

FIG. 15B16-point DCT-8:

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

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

FIG. 15C

32-pointDCT-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,C,F,-E,-B,-y,-v,-s,-p,-m,-j,-g,-d,-a,-c,-f,-i,-l,-o,-r,-u,-x,-A,-D,}
 {c,h,m,r,w,B,0,-B,-w,-r,-m,-h,-c,-c,-h,-m,-r,-w,-B,0,B,w,r,m,h,c,c,h,m,r,w,B,}
 {d,k,r,y,F,-A,-t,-m,-f,-b,-i,-p,-w,-D,C,v,o,h,a,g,n,u,B,-E,-x,-q,-j,-c,-e,-l,-s,-z,}
 {e,n,w,F,-y,-p,-g,-c,-l,-u,-D,A,r,i,a,j,s,B,-C,-t,-k,-b,-h,-q,-z,E,v,m,d,f,o,x,}
 {f,q,B,-A,-p,-e,-g,-r,-C,z,o,d,h,s,D,-y,-n,-c,-i,-t,-E,x,m,b,j,u,F,-w,-l,-a,-k,-v,}
 {g,t,0,-t,-g,-g,-t,0,t,g,g,t,0,-t,-g,-g,-t,0,t,g,g,t,0,-t,-g,-g,-t,0,t,g,g,t,}
 {h,w,-B,-m,-c,-r,0,r,c,m,B,-w,-h,-h,-w,B,m,c,r,0,-r,-c,-m,-B,w,h,h,w,-B,-m,-c,-r,}
 {i,z,-w,-f,-l,-C,t,c,o,F,-q,-a,-r,E,n,d,u,-B,-k,-g,-x,y,h,j,A,-v,-e,-m,-D,s,b,p,}
 {j,C,-r,-b,-u,z,g,m,F,-o,-e,-x,w,d,p,-E,-l,-h,-A,t,a,s,-B,-i,-k,-D,q,c,v,-y,-f,-n,}
 {k,F,-m,-i,-D,o,g,B,-q,-e,-z,s,c,x,-u,-a,-v,w,b,t,-y,-d,-r,A,f,p,-C,-h,-n,E,j,l,}
 {l,-E,-h,-p,A,d,t,-w,-a,-x,s,e,B,-o,-i,-F,k,m,-D,-g,-q,z,c,u,-v,-b,-y,r,f,C,-n,-j,}
 {m,-B,-c,-w,r,h,0,-h,-r,w,c,B,-m,-m,B,c,w,-r,-h,0,h,r,-w,-c,-B,m,m,-B,-c,-w,r,h,}
 {n,-y,-c,-D,i,s,-t,-h,E,d,x,-o,-m,z,b,C,-j,-r,u,g,-F,-e,-w,p,l,-A,-a,-B,k,q,-v,-f,}
 {o,-v,-h,C,a,D,-g,-w,n,p,-u,-i,B,b,E,-f,-x,m,q,-t,-j,A,c,F,-e,-y,l,r,-s,-k,z,d,}
 {p,-s,-m,v,j,-y,-g,B,d,-E,-a,-F,c,C,-f,-z,i,w,-l,-t,o,q,-r,-n,u,k,-x,-h,A,e,-D,-b,}
 {q,-p,-r,o,s,-n,-t,m,u,-l,-v,k,w,-j,-x,i,y,-h,-z,g,A,-f,-B,e,C,-d,-D,c,E,-b,-F,a,}
 {r,-m,-w,h,B,-c,0,c,-B,-h,w,m,-r,-r,m,w,-h,-B,c,0,-c,B,h,-w,-m,r,r,-m,-w,h,B,-c,}
 {s,-j,-B,a,-C,-i,t,r,-k,-A,b,-D,-h,u,q,-l,-z,c,-E,-g,v,p,-m,-y,d,-F,-f,w,o,-n,-x,e,}
 {t,-g,0,g,-t,-t,g,0,-g,t,t,-g,0,g,-t,-t,g,0,-g,t,t,-g,0,g,-t,-t,g,0,-g,t,t,-g,}
 {u,-d,B,n,-k,-E,g,-r,-x,a,-y,-q,h,-F,-j,o,A,-c,v,t,-e,C,m,-l,-D,f,-s,-w,b,-z,-p,i,}
 {v,-a,w,u,-b,x,t,-c,y,s,-d,z,r,-e,A,q,-f,B,p,-g,C,o,-h,D,n,-i,E,m,-j,F,l,-k,}
 {w,-c,r,B,-h,m,0,-m,h,-B,-r,c,-w,-w,c,-r,-B,h,-m,0,m,-h,B,r,-c,w,w,-c,r,B,-h,m,}
 {x,-f,m,-E,-q,b,-t,-B,j,-i,A,u,-c,p,F,-n,e,-w,-y,g,-l,D,r,-a,s,C,-k,h,-z,-v,d,-o,}
 {y,-i,h,-x,-z,j,-g,w,A,-k,f,-v,-B,l,-e,u,C,-m,d,-t,-D,n,-c,s,E,-o,b,-r,-F,p,-a,q,}
 {z,-l,c,-q,E,u,-g,h,-v,-D,p,-b,m,-A,-y,k,-d,r,-f,g,-s,E,w,-k,b,-n,z,B,-p,d,-i,u,}
 {A,-o,c,-j,v,F,-t,h,-e,q,-C,-y,m,-a,l,-x,-D,r,-f,g,-s,E,w,-k,b,-n,z,B,-p,d,-i,u,}
 {B,-r,h,-c,m,-w,0,w,-m,c,-h,r,-B,-B,r,-h,c,-m,w,0,-w,m,-c,h,-r,B,B,-r,h,-c,m,-w,}
 {C,-u,m,-e,d,-l,t,-B,-D,v,-n,f,-c,k,-s,A,E,-w,o,-g,b,-j,r,-z,-F,x,-p,h,-a,i,-q,y,}
 {D,-x,r,-l,f,-a,g,-m,s,-y,E,C,-w,q,-k,e,-b,h,-n,t,-z,F,B,-v,p,-j,d,-c,i,-o,u,-A,}
 {E,-A,w,-s,o,-k,g,-c,b,-f,j,-n,r,-v,z,-D,-F,B,-x,t,-p,l,-h,d,-a,e,-i,m,-q,u,-y,C,}
 {F,-D,B,-z,x,-v,t,-r,p,-n,l,-j,h,-f,d,-b,a,-c,e,-g,i,-k,m,-o,q,-s,u,-w,y,-A,C,-E,}

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,
 88,86,85,84,82,80,78,77,74,72,68,66,63,60,56,53,50,45,42,38,34,30,26,21,17,1
 3,9,4}

FIG. 15D

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. 16

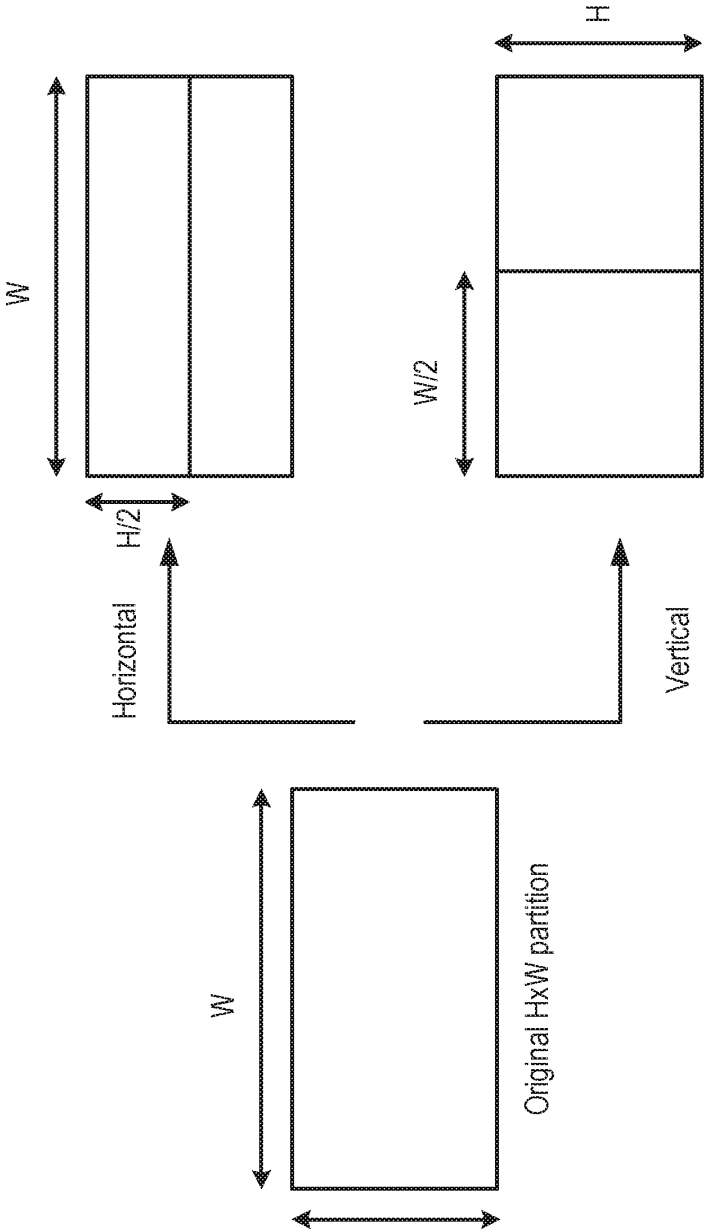


FIG. 17

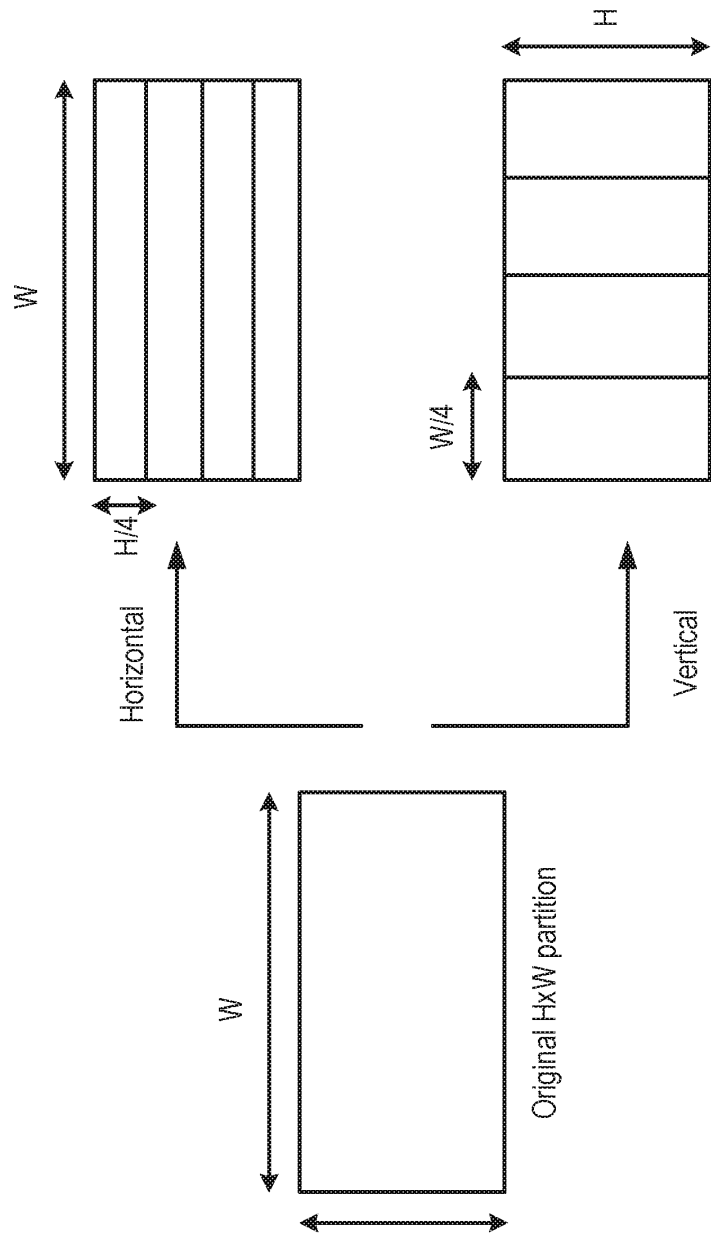


FIG. 18

1900

Descriptor	
	coding_unit(x0, y0, cbWidth, cbHeight, treeType) {
	if(tile_group_type != I sps_ibc_enabled_flag) {
	if(treeType != DUAL_TREE_CHROMA)
ae(v)	cu_skip_flag[x0][y0]
	if(cu_skip_flag[x0][y0] == 0 && tile_group_type != I)
ae(v)	pred_mode_flag
	if(((tile_group_type == I && cu_skip_flag[x0][y0] == 0)
	(tile_group_type != I && CuPredMode[x0][y0] != MODE_INTRA)) &&
	sps_ibc_enabled_flag)
ae(v)	pred_mode_ibc_flag
	}
	if(CuPredMode[x0][y0] == MODE_INTRA) {
	if(sps_pcm_enabled_flag &&
	cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY &&
	cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)
ae(v)	pcm_flag[x0][y0]
	if(pcm_flag[x0][y0]) {
	while(!byte_aligned())
l(1)	pcm_alignment_zero_bit
	pcm_sample(cbWidth, cbHeight, treeType)
	} else {
	if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {
	if((y0 % CtbSizeY) > 0)
ae(v)	intra_luma_ref_idx[x0][y0]

FIG. 19A

1910

Continue from FIG. 19A

if (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])	
intra_luma_mpm_idx[x0][y0]	ae(v)
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
if (treeType = = SINGLE_TREE treeType = = DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
} else if (treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
.....	
}	

1900

FIG. 19B

FIG. 20A

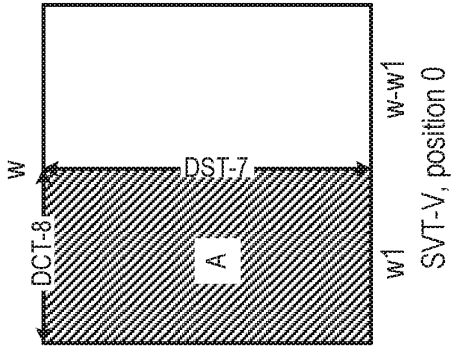


FIG. 20B

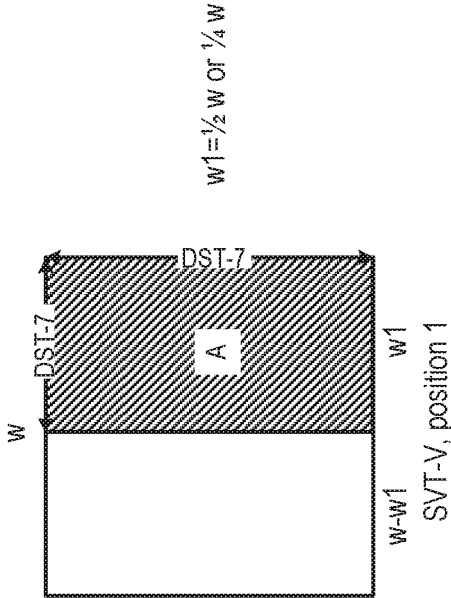


FIG. 20C

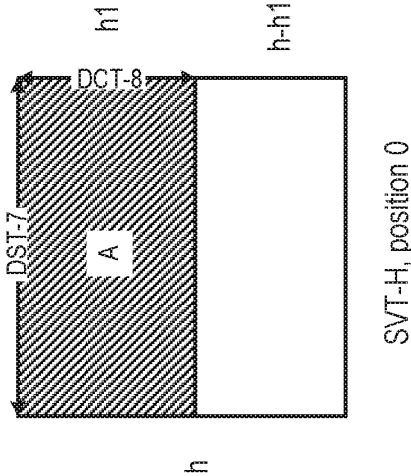
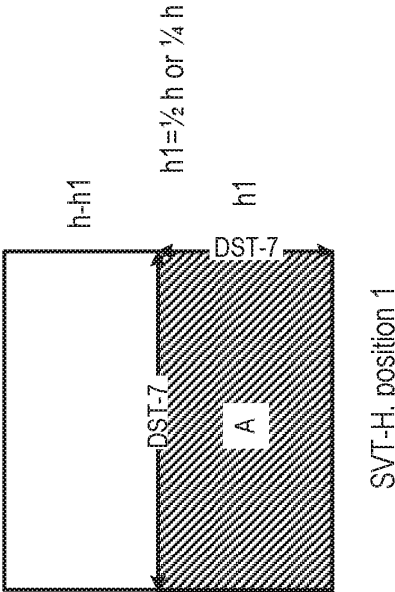


FIG. 20D



Sequence parameter set RBSP syntax	
seq_parameter_set_rbsp() {	
sps_seq_parameter_set_id	ue(v)
...	
sps_mts_intra_enabled_flag	u(1)
sps_mts_inter_enabled_flag	u(1)
sps_sbt_enable_flag	u(1)
rbsp_trailing_bits()	
}	

FIG. 21A

General slice header syntax

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I) {	
log2_diff_ctu_max_bt_size	ue(v)
if(sps_sbtmvp_enabled_flag) {	
sbtmvp_size_override_flag	u(1)
if(sbtmvp_size_override_flag)	
log2_sbtmvp_active_size_minus2	u(3)
}	
if(sps_temporal_mvp_enabled_flag)	
slice_temporal_mvp_enabled_flag	u(1)
if(slice_type == B)	
mvd_l1_zero_flag	u(1)
if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B)	
collocated_from_l0_flag	u(1)
}	
six_minus_max_num_merge_cand	ue(v)
if(sps_sbt_enable_flag)	
slice_max_sbt_size_64_flag	u(1)
}	
if (sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag)	
alf_data()	
}	
dep_quant_enabled_flag	u(1)
if(!dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
byte_alignment()	
}	

FIG. 21B

2102

Coding unit syntax

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
...	
if(CuPredMode[x0][y0] != MODE_INTRA && cu_skip_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
if(CuPredMode[x0][y0] != MODE_INTRA && sps_sbt_enable_flag) {	
if(cbWidth <= maxSbtSize && cbHeight <= maxSbtSize) {	
allowSbtVerHalf = cbWidth >= 8	
allowSbtVerQuad = cbWidth >= 16	
allowSbtHorHalf = cbHeight >= 8	
allowSbtHorQuad = cbHeight >= 16	
if(allowSbtVerHalf allowSbtHorHalf allowSbtVerQuad allowSbtHorQuad)	
cu_sbt_flag[x0][y0]	ae(v)
)	
if(cu_sbt_flag[x0][y0]) {	
if((allowSbtVerHalf allowSbtHorHalf) && (allowSbtVerQuad allowSbtHorQuad))	
cu_sbt_quad_flag[x0][y0]	ae(v)
if((cu_sbt_quad_flag[x0][y0] && allowSbtVerQuad && allowSbtHorQuad) (!cu_sbt_quad_flag[x0][y0] && allowSbtVerHalf && allowSbtHorHalf))	
cu_sbt_horizontal_flag[x0][y0]	ae(v)
cu_sbt_pos_flag[x0][y0]	ae(v)
)	
}	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
}	
}	

FIG. 21C

2103

Transform tree syntax

transform_tree(x0, y0, tbWidth, tbHeight, treeType) {	Descriptor
if(tbWidth > MaxTbSizeY tbHeight > MaxTbSizeY) {	
trafoWidth = (tbWidth > MaxTbSizeY) ? (tbWidth / 2) : tbWidth	
trafoHeight = (tbHeight > MaxTbSizeY) ? (tbHeight / 2) : tbHeight	
transform_tree(x0, y0, trafoWidth, trafoHeight)	
if(tbWidth > MaxTbSizeY)	
transform_tree(x0 + trafoWidth, y0, trafoWidth, trafoHeight, treeType)	
if(tbHeight > MaxTbSizeY)	
transform_tree(x0, y0 + trafoHeight, trafoWidth, trafoHeight, treeType)	
if(tbWidth > MaxTbSizeY && tbHeight > MaxTbSizeY)	
transform_tree(x0 + trafoWidth, y0 + trafoHeight, trafoWidth, trafoHeight, treeType)	
} else if(cu_sbt_flag[x0][y0])	
factorTb0 = cu_sbt_quad_flag[x0][y0] ? 1 : 2	
factorTb0 = cu_sbt_pos_flag[x0][y0] ? (4 - factorTb0) : factorTb0	
noResiTb0 = cu_sbt_pos_flag[x0][y0] ? 1 : 0	
if(!cu_sbt_horizontal_flag[x0][y0]) {	
trafoWidth = tbWidth * factorTb0 / 4	
transform_tree(x0, y0, trafoWidth, tbHeight, treeType, noResiTb0)	
transform_tree(x0 + trafoWidth, y0, tbWidth - trafoWidth, tbHeight, treeType, !noResiTb0)	
}	
else {	
trafoHeight = tbHeight * factorTb0 / 4	
transform_tree(x0, y0, tbWidth, trafoHeight, treeType, noResiTb0)	
transform_tree(x0, y0 + trafoHeight, tbWidth, tbHeight - trafoHeight, treeType, !noResiTb0)	
}	
} else {	
transform_unit(x0, y0, tbWidth, tbHeight, treeType, 0)	
}	
}	

FIG. 21D

2104

Transform unit syntax			Descriptor
transform_unit(x0, y0, tbWidth, tbHeight, treeType, noResi) {			
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {			
tu_cbf_luma[x0][y0]			ae(v)
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) {			
tu_cbf_cb[x0][y0]			ae(v)
tu_cbf_cr[x0][y0]			ae(v)
}			
if((((CuPredMode[x0][y0] == MODE_INTRA) && sps_mts_intra_enabled_flag) (((CuPredMode[x0][y0] == MODE_INTER) && sps_mts_inter_enabled_flag) && tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA && (tbWidth <= 32) && (tbHeight <= 32) && !cu_sbt_flag[x0][y0]))			
cu_mts_flag[x0][y0]			ae(v)
if(tu_cbf_luma[x0][y0])			
residual_coding(x0, y0, log2(tbWidth), log2(tbHeight), 0)			
if(tu_cbf_cb[x0][y0])			
residual_coding(x0, y0, log2(tbWidth / 2), log2(tbHeight / 2), 1)			
if(tu_cbf_cr[x0][y0])			
residual_coding(x0, y0, log2(tbWidth / 2), log2(tbHeight / 2), 2)			
}			

FIG. 21E

Sequence parameter set RBSP semantics

sps_sbt_enabled_flag equal to 0 specifies that sub-block transform for inter-predicted CU is disabled. **sps_sbt_enabled_flag** equal to 1 specifies that sub-block transform for inter-predicted CU is enabled. 2108

General slice header semantics

slice_sbt_max_size_64_flag equal to 0 specifies that the maximum CU width and height for allowing sub-block transform is 32. **slice_sbt_max_size_64_flag** equal to 1 specifies that the maximum CU width and height for allowing sub-block transform is 64. 2109
maxSbtSize = **slice_sbt_max_size_64_flag**? 64 : 32

FIG. 21F

Coding unit semantics

cu_sbt_flag[x0][y0] equal to 1 specifies that for the current coding unit, sub-block transform is used. **cu_sbt_flag**[x0][y0] equal to 0 specifies that for the current coding unit, the sub-block transform is not used.

When **cu_sbt_flag**[x0][y0] is not present, its value is inferred to be equal to 0.

NOTE – : When sub-block transform is used, a coding unit is tiled into two transform units, one transform unit has residual, the other does not have residual.

cu_sbt_quad_flag[x0][y0] equal to 1 specifies that for the current coding unit, the sub-block transform include a transform unit of 1/4 size of the current coding unit. **cu_sbt_quad_flag**[x0][y0] equal to 0 specifies that for the current coding unit the sub-block transform include a transform unit of 1/2 size of the current coding unit.

When **cu_sbt_quad_flag**[x0][y0] is not present, its value is inferred to be equal to 0.

cu_sbt_horizontal_flag[x0][y0] equal to 1 specifies that the current coding unit is tiled into 2 transform units by a horizontal split. **cu_sbt_horizontal_flag**[x0][y0] equal to 0 specifies that the current coding unit is tiled into 2 transform units by a vertical split.

When **cu_sbt_horizontal_flag**[x0][y0] is not present, its value is derived as follows:

- If **cu_sbt_quad_flag**[x0][y0] is equal to 1, **cu_sbt_horizontal_flag**[x0][y0] is set to be equal to `allowSbtHoriQuad`.
- Otherwise (**cu_sbt_quad_flag**[x0][y0] is equal to 0), **cu_sbt_horizontal_flag**[x0][y0] is set to be equal to `allowSbtHoriHalf`.

cu_sbt_pos_flag[x0][y0] equal to 1 specifies that the **tu_cbf_luma**, **tu_cbf_cb** and **tu_cbf_cr** of the first transform unit in the current coding unit are not present in the bitstream. **cu_sbt_pos_flag**[x0][y0] equal to 0 specifies that the **tu_cbf_luma**, **tu_cbf_cb** and **tu_cbf_cr** of the second transform unit in the current coding unit are not present in the bitstream.

FIG. 21G

2110

Transformation process for scaled transform coefficients

General

Inputs to this process are:

- a luma location ($xTbY, yTbY$) specifying the top-left sample of the current luma transform block relative to the top-left luma sample of the current picture,
- a variable $nTbW$ specifying the width of the current transform block,
- a variable $nTbH$ specifying the height of the current transform block,
- a variable $cldx$ specifying the colour component of the current block,
- an $(nTbW) \times (nTbH)$ array $d[x][y]$ of scaled transform coefficients with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$.

Output of this process is the $(nTbW) \times (nTbH)$ array $r[x][y]$ of residual samples with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$.

2112

If $cu_sbt_flag[xTbY][yTbY]$ is equal to 1, the variable $trTypeHor$ specifying the horizontal transform kernel and the variable $trTypeVer$ specifying the vertical transform kernel are derived in Table 8-X depending on $cu_sbt_horizontal_flag[xTbY][yTbY]$ and $cu_sbt_pos_flag[xTbY][yTbY]$.

Otherwise ($cu_sbt_flag[xTbY][yTbY]$ is equal to 0), the variable $trTypeHor$ specifying the horizontal transform kernel and the variable $trTypeVer$ specifying the vertical transform kernel are derived in Table 8-9 depending on $mts_idx[xTbY][yTbY]$ and $CuPredMode[xTbY][yTbY]$.

The $(nTbW) \times (nTbH)$ array r of residual samples is derived as follows:

1. Each (vertical) column of scaled transform coefficients $d[x][y]$ with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ is transformed to $e[x][y]$ with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ by invoking the one-dimensional transformation process for each column $x = 0..nTbW - 1$ with the height of the transform block $nTbH$, the list $d[x][y]$ with $y = 0..nTbH - 1$ and the transform type variable $trType$ set equal to $trTypeVer$ as inputs, and the output is the list $e[x][y]$ with $y = 0..nTbH - 1$.
2. The intermediate sample values $g[x][y]$ with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ are derived as follows:

$$g[x][y] = \text{Clip3}(\text{CoeffMin}, \text{CoeffMax}, (e[x][y] + 256) \gg 9)$$

FIG. 21H

Continue from FIG. 21H

3. Each (horizontal) row of the resulting array $g[x][y]$ with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ is transformed to $r[x][y]$ with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ by invoking the one-dimensional transformation process for each row $y = 0..nTbH - 1$ with the width of the transform block $nTbW$, the list $g[x][y]$ with $x = 0..nTbW - 1$ and the transform type variable $trType$ set equal to $trTypeHor$ as inputs, and the output is the list $r[x][y]$ with $x = 0..nTbW - 1$.

Table 8-X – Specification of $trTypeHor$ and $trTypeVer$ depending on $mts_idx[x][y]$ and $CuPredMode[x][y]$

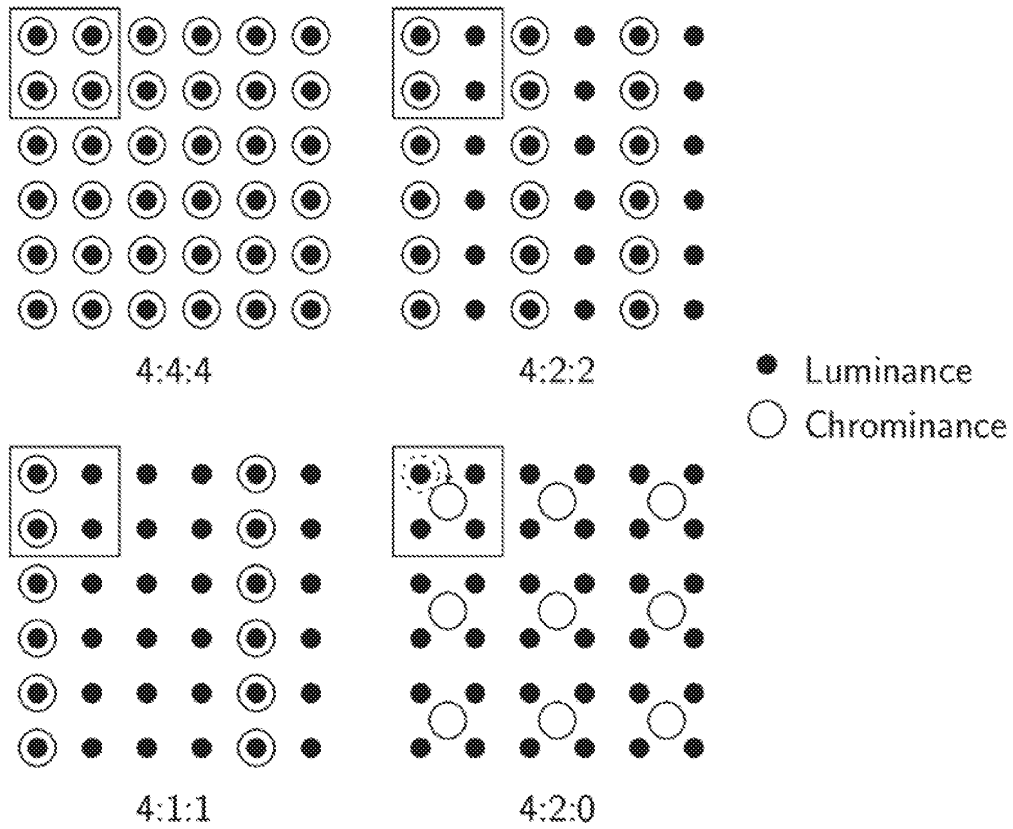
	cu_sbt_horizontal_flag [xTbY][yTbY] == 0		cu_sbt_horizontal_flag [xTbY][yTbY] == 1	
	cu_sbt_pos_flag [xTbY][yTbY] == 0	cu_sbt_pos_flag [xTbY][yTbY] == 1	cu_sbt_pos_flag [xTbY][yTbY] == 0	cu_sbt_pos_flag [xTbY][yTbY] == 1
$trTypeHor$	2	1	$nTbW > 32 ? 0 : 1$	$nTbW > 32 ? 0 : 1$
$trTypeVer$	$nTbH > 32 ? 0 : 1$	$nTbH > 32 ? 0 : 1$	2	1

2113

Table 8-9 – Specification of $trTypeHor$ and $trTypeVer$ depending on $mts_idx[x][y]$ and $CuPredMode[x][y]$

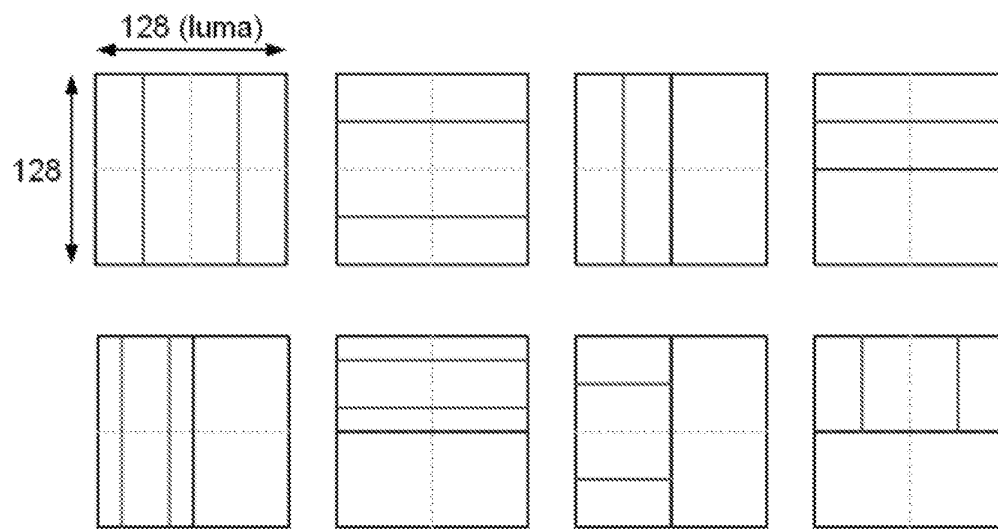
$mts_idx[xTbY][yTbY]$	$CuPredMode[xTbY][yTbY] == MODE_INTRA$		$CuPredMode[xTbY][yTbY] == MODE_INTER$	
	$trTypeHor$	$trTypeVer$	$trTypeHor$	$trTypeVer$
-1 (inferred)	0	0	0	0
0 (00)	1	1	2	2
1 (01)	2	1	1	2
2 (10)	1	2	2	1
3 (11)	2	2	1	1

FIG. 21I



Different YUV formats

FIG. 22



Examples of disallowed TT and BT partitioning

FIG. 23

transform_tree(x0, y0, tbWidth, tbHeight, treeType) {	Descriptor
InferTuCbFLuma = 1	
if(IntraSubPartSplitType == NO_ISP_SPLIT) {	
MaxSbSizeY = (tbWidth > 64 tbHeight > 64) ? 64 : MaxTbSizeY	
if(tbWidth > MaxSbSizeY tbHeight > MaxSbSizeY) {	
trafoWidth = (tbWidth > MaxSbSizeY) ? (tbWidth / 2) : tbWidth	
trafoHeight = (tbHeight > MaxSbSizeY) ? (tbHeight / 2) : tbHeight	
transform_tree(x0, y0, trafoWidth, trafoHeight)	
if(tbWidth > MaxSbSizeY)	
transform_tree(x0 + trafoWidth, y0, trafoWidth, trafoHeight, treeType)	
if(tbHeight > MaxSbSizeY)	
transform_tree(x0, y0 + trafoHeight, trafoWidth, trafoHeight, treeType)	
if(tbWidth > MaxSbSizeY && tbHeight > MaxSbSizeY)	
transform_tree(x0 + trafoWidth, y0 + trafoHeight, trafoWidth, trafoHeight, treeType)	
} else {	
transform_unit(x0, y0, tbWidth, tbHeight, treeType, 0)	
}	
} else if(cu_sbt_flag) {	
if(!cu_sbt_horizontal_flag) {	
trafoWidth = tbWidth * SbtNumFourthsTb0 / 4	
transform_unit(x0, y0, trafoWidth, tbHeight, treeType, 0)	
transform_unit(x0 + trafoWidth, y0, tbWidth - trafoWidth, tbHeight, treeType, 1)	
} else {	
trafoHeight = tbHeight * SbtNumFourthsTb0 / 4	
transform_unit(x0, y0, tbWidth, trafoHeight, treeType, 0)	
transform_unit(x0, y0 + trafoHeight, tbWidth, tbHeight - trafoHeight, treeType, 1)	
}	
} else if(IntraSubPartitionsSplitType == ISP_HOR_SPLIT) {	
trafoHeight = tbHeight / NumIntraSubPartitions	
for(partIdx = 0; partIdx < NumIntraSubPartitions; partIdx++)	
transform_unit(x0, y0 + trafoHeight * partIdx, tbWidth, trafoHeight, treeType, partIdx)	
} else if(IntraSubPartitionsSplitType == ISP_VER_SPLIT) {	
trafoWidth = tbWidth / NumIntraSubPartitions	
for(partIdx = 0; partIdx < NumIntraSubPartitions; partIdx++)	
transform_unit(x0 + trafoWidth * partIdx, y0, trafoWidth, tbHeight, treeType, partIdx)	
}	
}	

FIG. 24

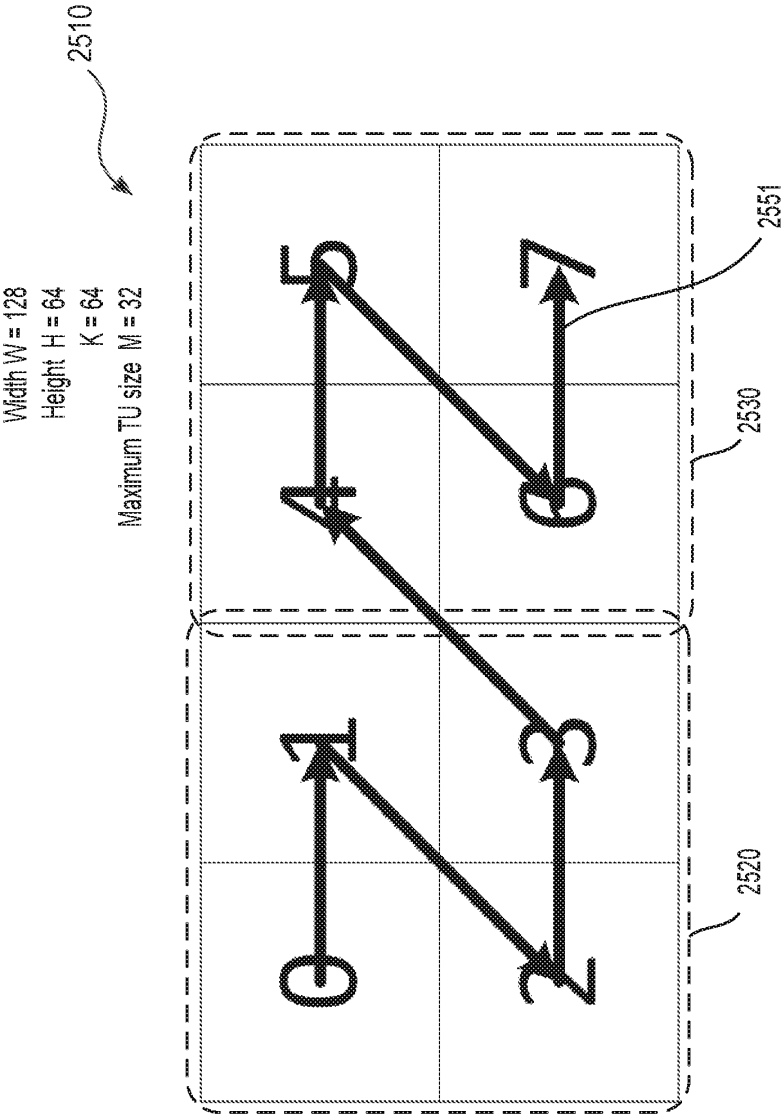


FIG. 25

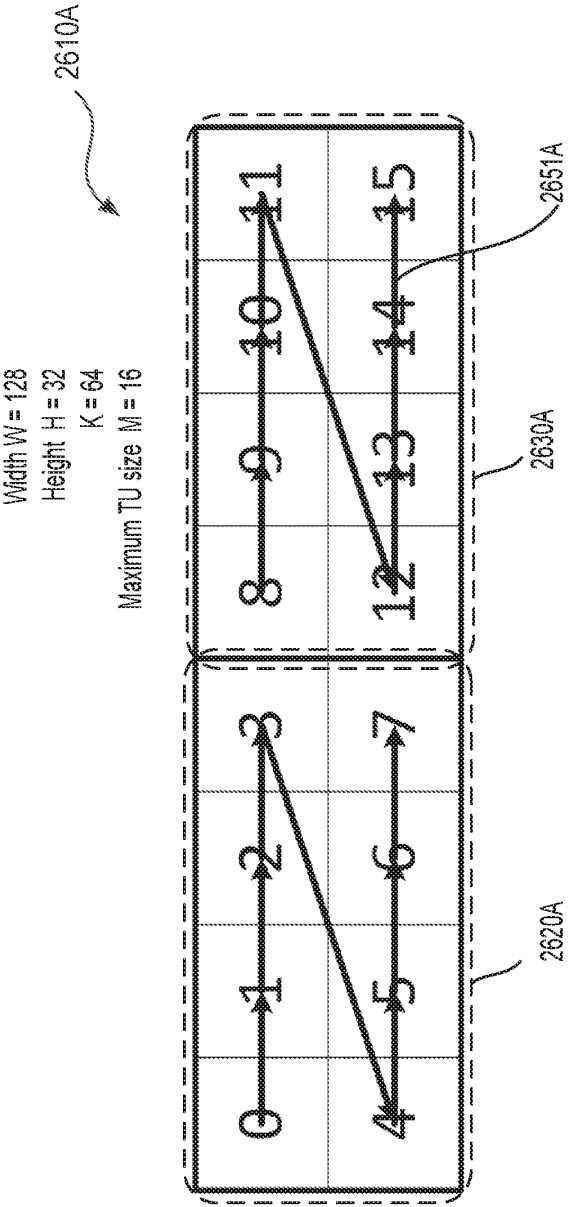


FIG. 26A

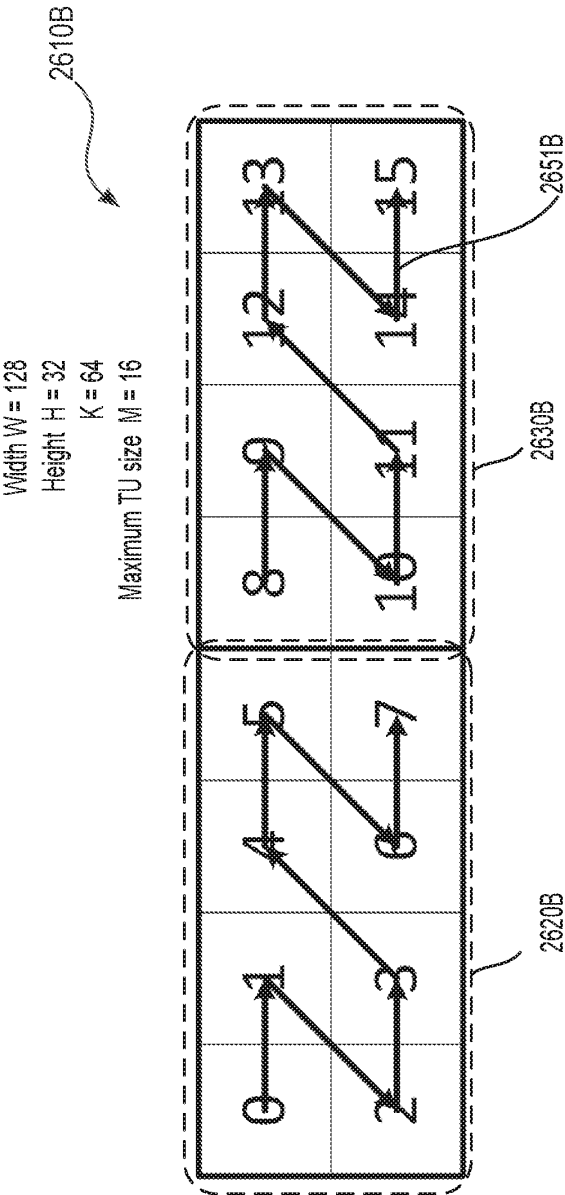
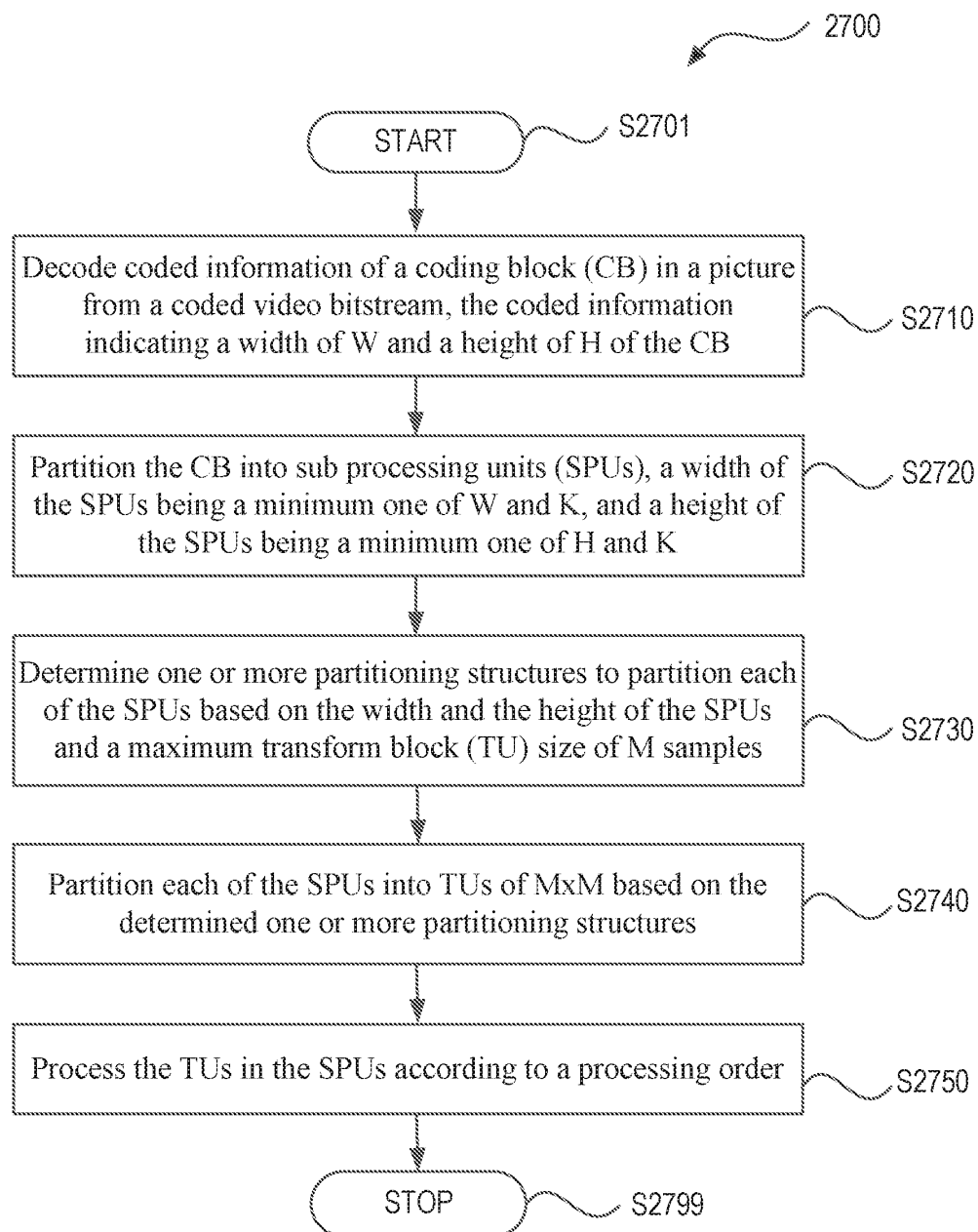


FIG. 26B

**FIG. 27**

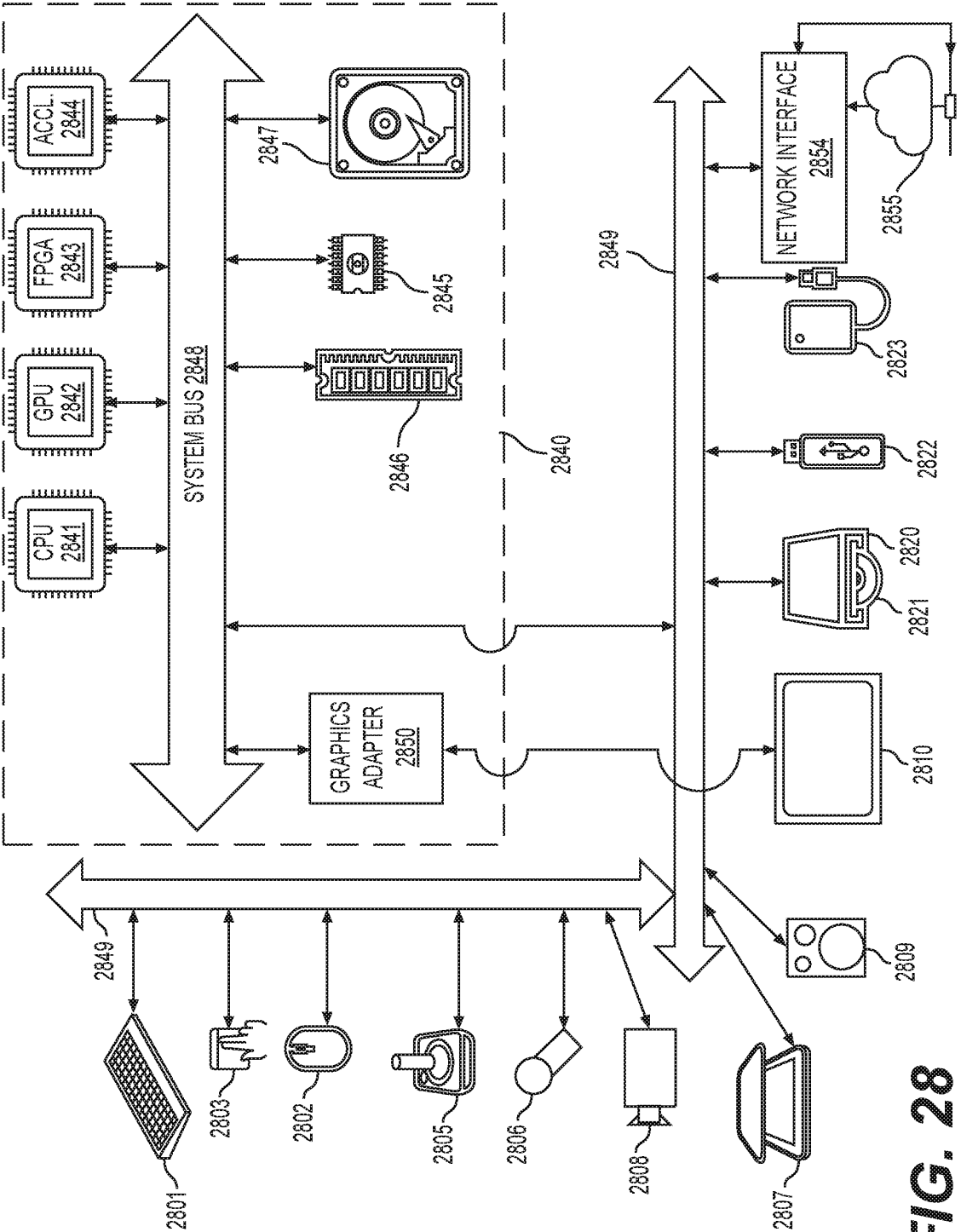


FIG. 28