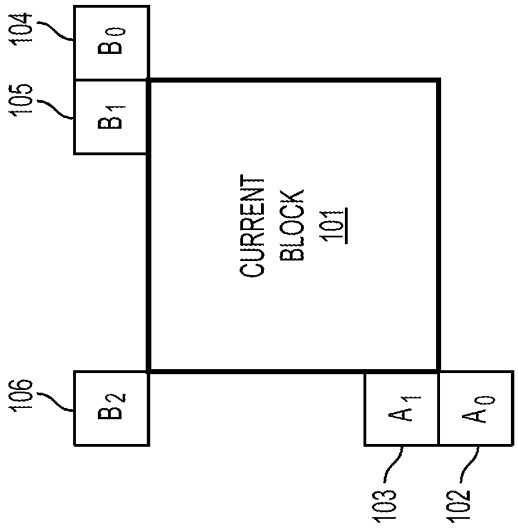

FIG. 1
(Related Art)

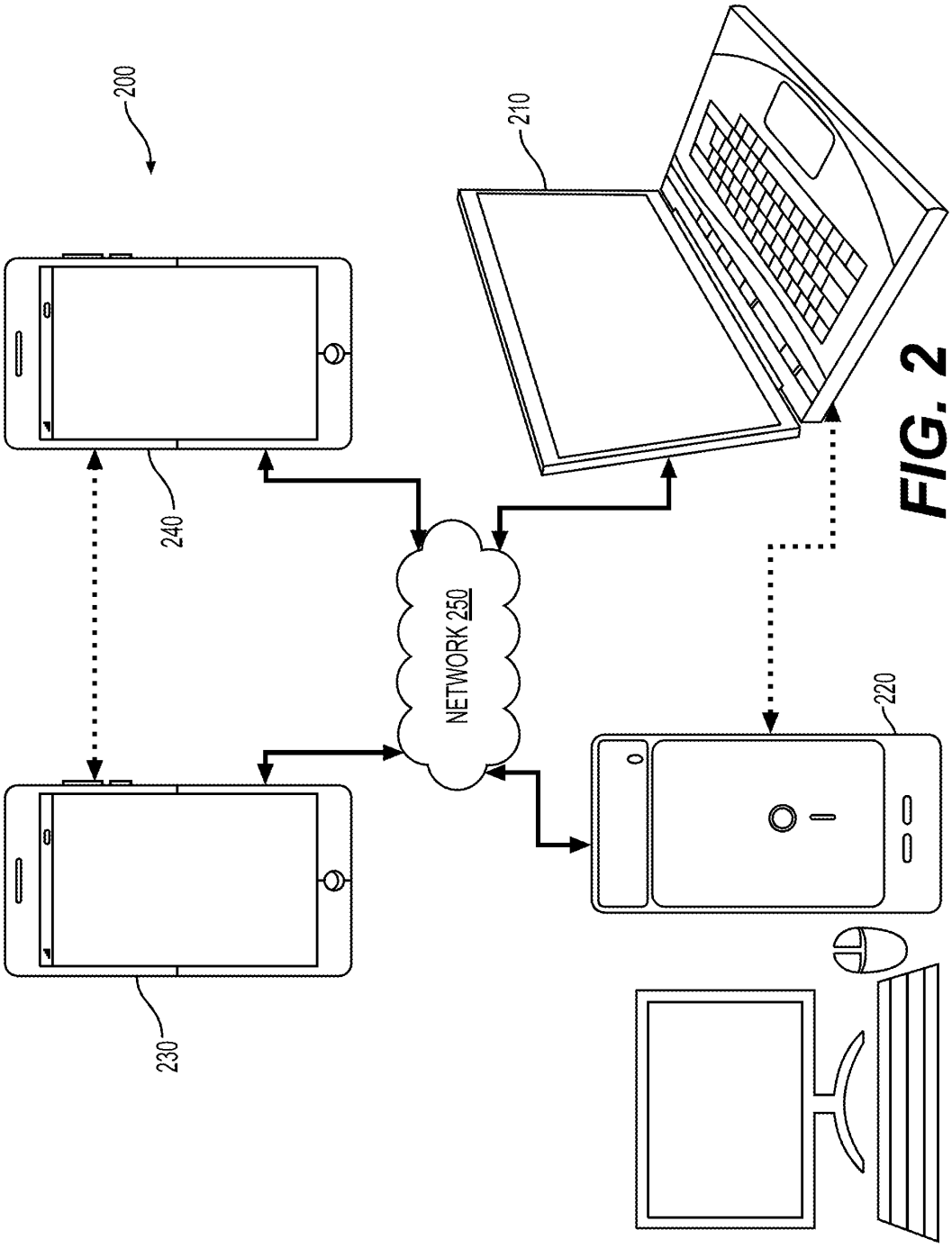

FIG. 2

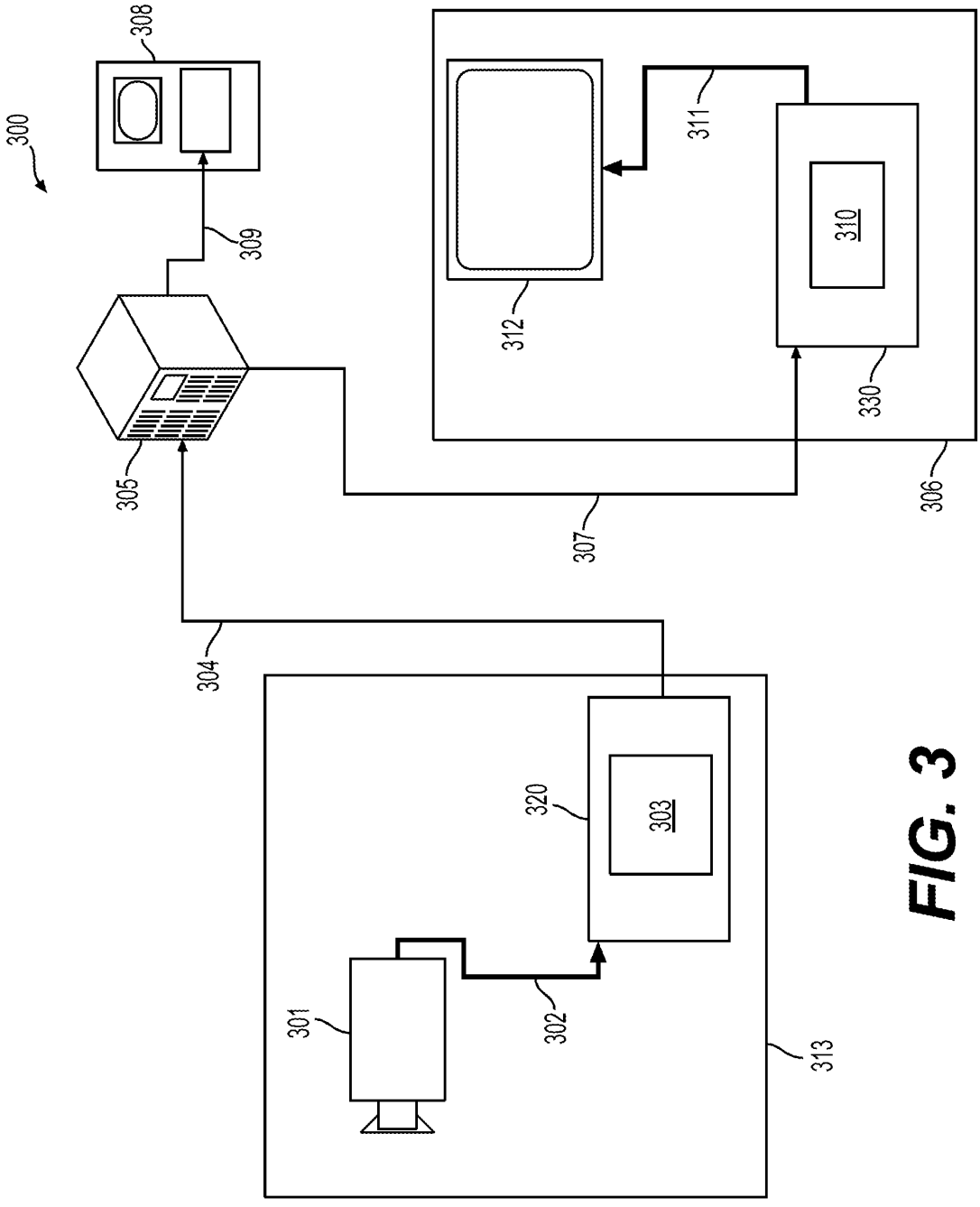

FIG. 3

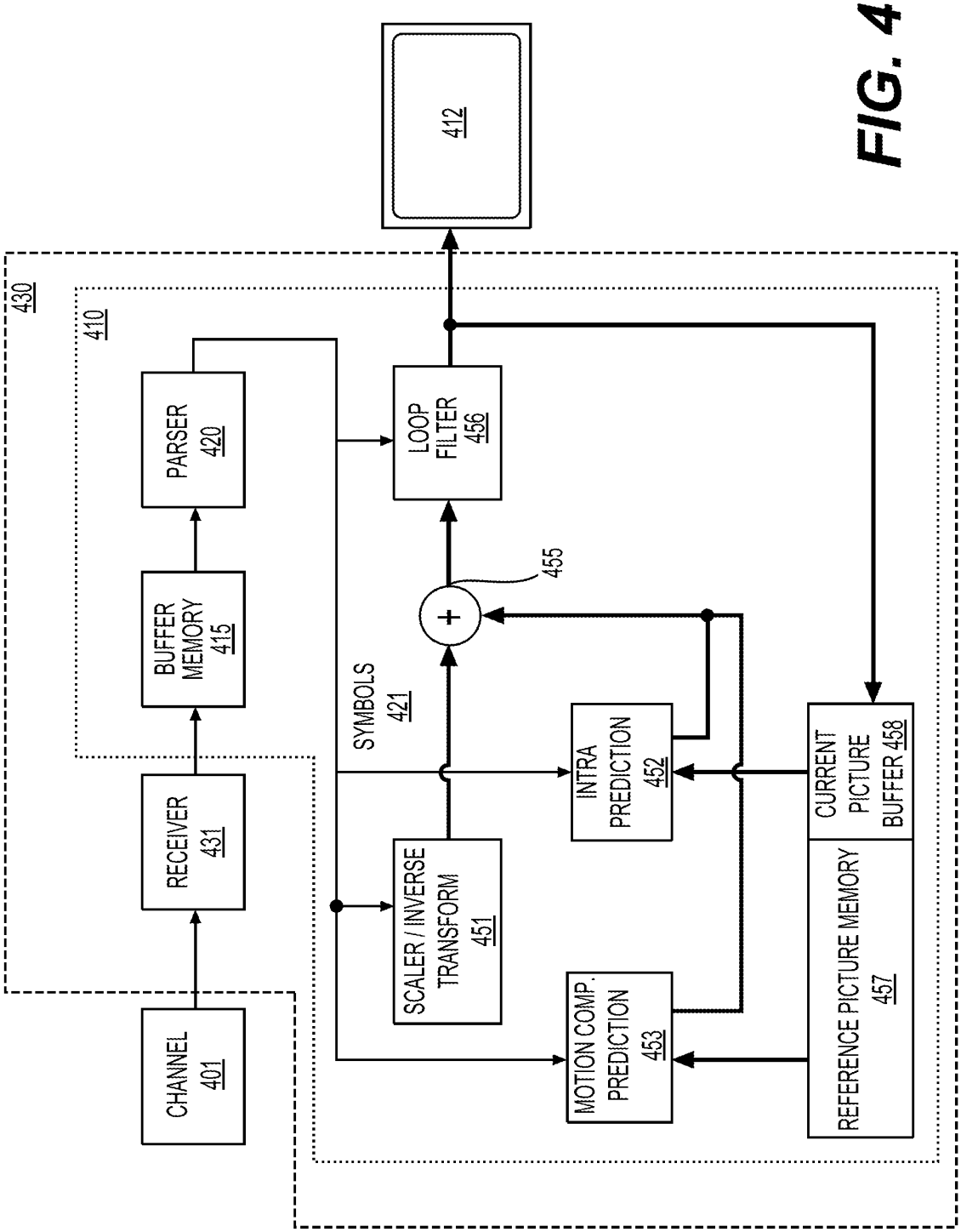
412
430
410
PARSER 420
LOOP FILTER 456
BUFFER MEMORY 415
455
SYMBOLS 421
RECEIVER 431
INTRA PREDICTION 452
CURRENT PICTURE BUFFER 458
SCALER / INVERSE TRANSFORM 451
REFERENCE PICTURE MEMORY 457
MOTION COMP. PREDICTION 453
CHANNEL 401

FIG. 4

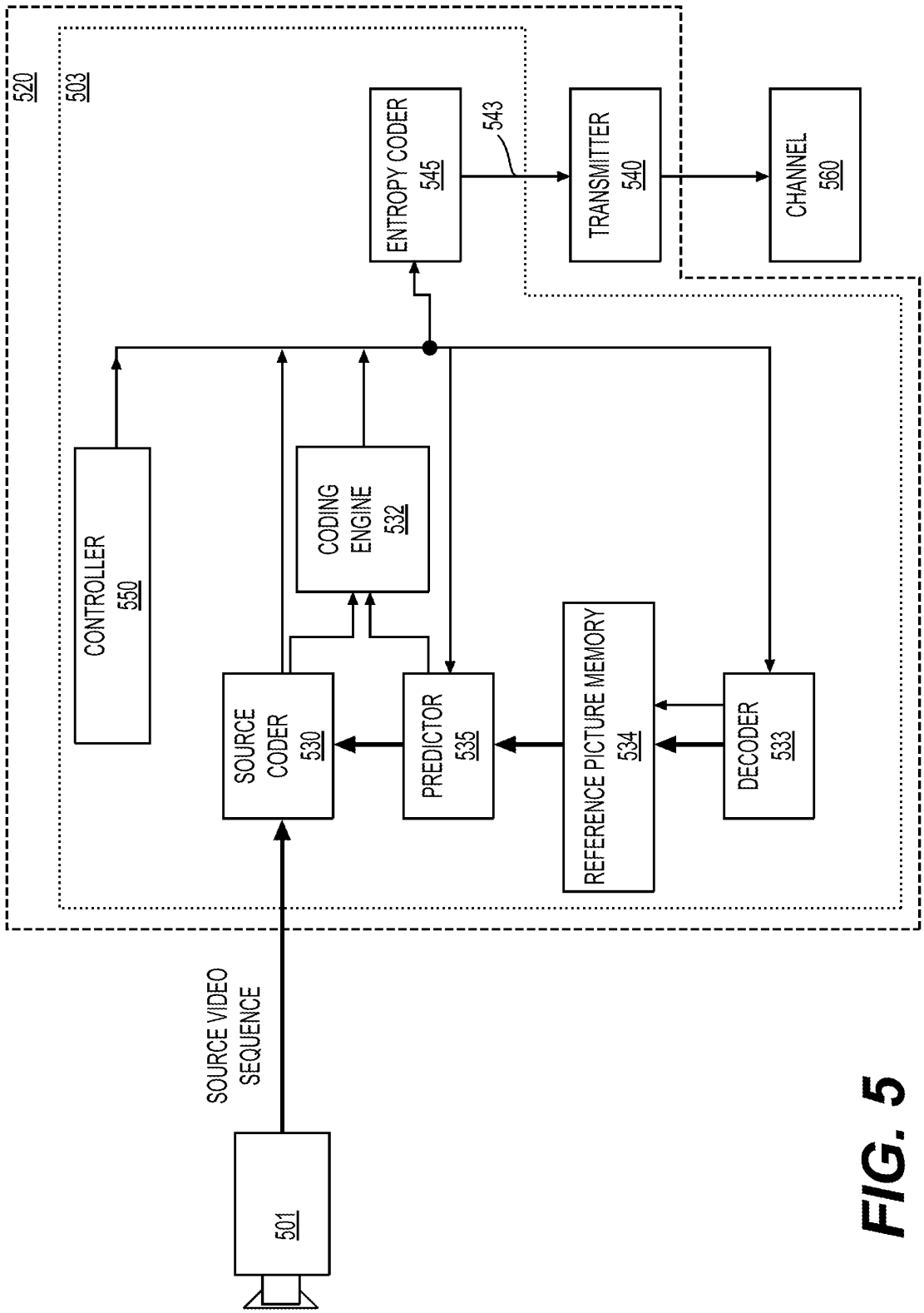
520
503
ENTROPY CODER
545
543
TRANSMITTER
540
CHANNEL
560
CONTROLLER
550
CODING
ENGINE
532
SOURCE
CODER
530
PREDICTOR
535
REFERENCE PICTURE MEMORY
534
DECODER
533
SOURCE VIDEO
SEQUENCE
501

FIG. 5

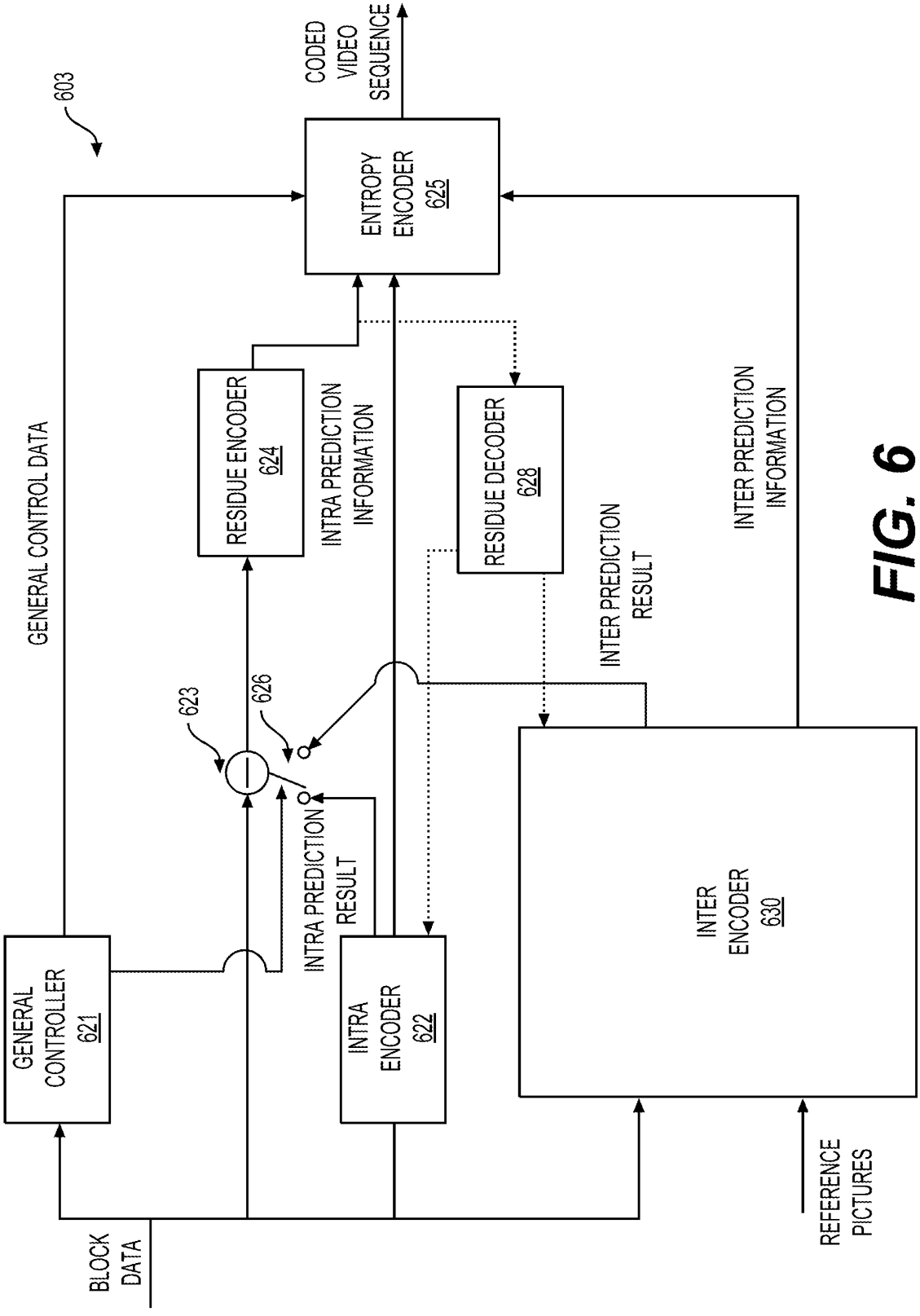
603
CODED VIDEO SEQUENCE
ENTROPY ENCODER 625
GENERAL CONTROL DATA
RESIDUE ENCODER 624
INTRA PREDICTION INFORMATION
RESIDUE DECODER 628
INTER PREDICTION RESULT
INTER PREDICTION INFORMATION
623
626
INTRA PREDICTION RESULT
GENERAL CONTROLLER 621
INTRA ENCODER 622
INTER ENCODER 630
BLOCK DATA
REFERENCE PICTURES

FIG. 6

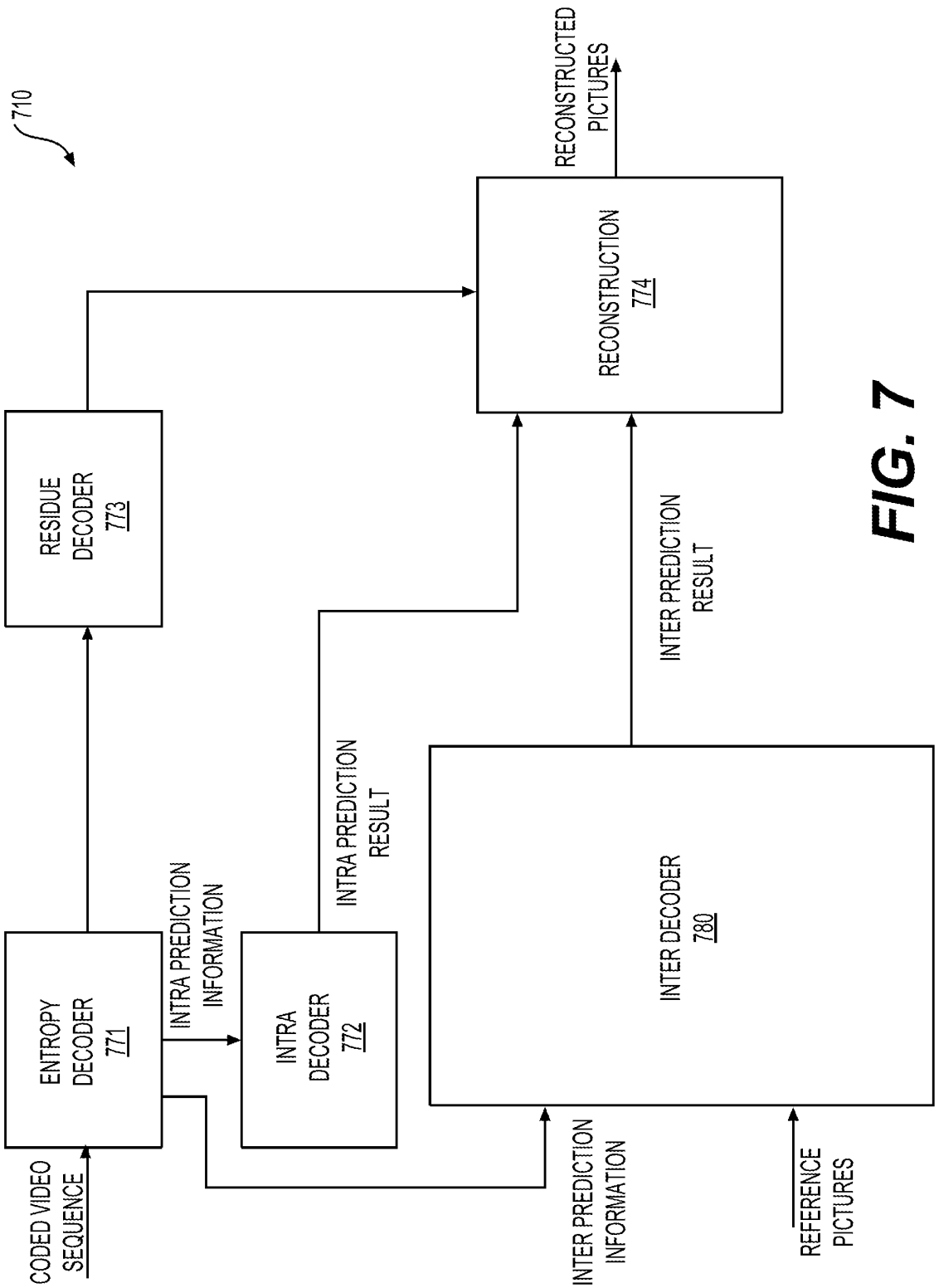
710
RECONSTRUCTED PICTURES
RECONSTRUCTION 774
RESIDUE DECODER 773
INTER PREDICTION RESULT
INTRA PREDICTION RESULT
INTER DECODER 780
ENTROPY DECODER 771
INTRA PREDICTION INFORMATION
INTRA DECODER 772
CODED VIDEO SEQUENCE
INTER PREDICTION INFORMATION
REFERENCE PICTURES

FIG. 7

4x4 transform

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

FIG. 8A

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

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}
{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}
```

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}
{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 -9-25-43-57-70-80-87-90-90-87-80-70-57-43-25 -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 83 36-36-83-83-36 36 83 83 36-36-83-83-36 36 83 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 64-64-64 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 83-83 36 36-83 83-36-36 83-83 36 36-83 83-36-36 83-83 36 36-83 83-36-36 83-83 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

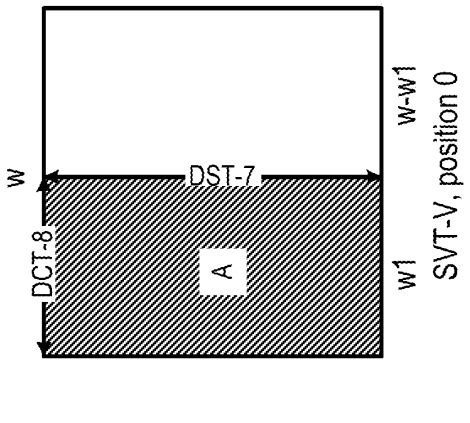

FIG. 9A

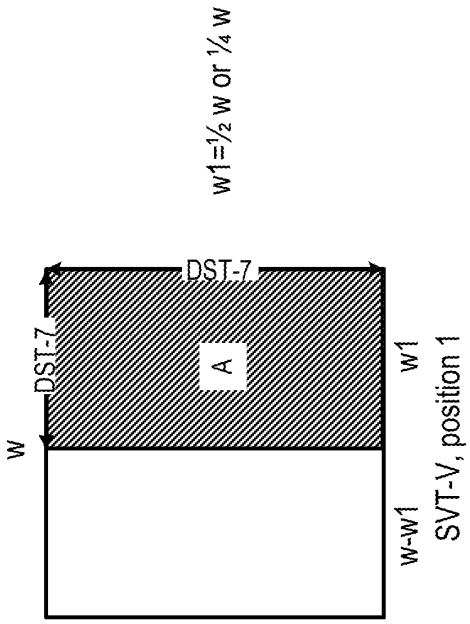

FIG. 9B

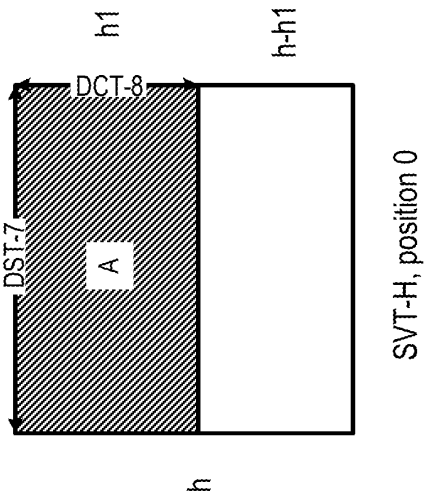

FIG. 9C

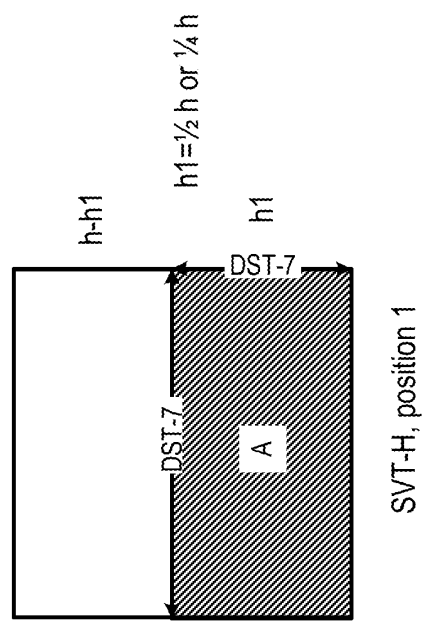

FIG. 9D

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

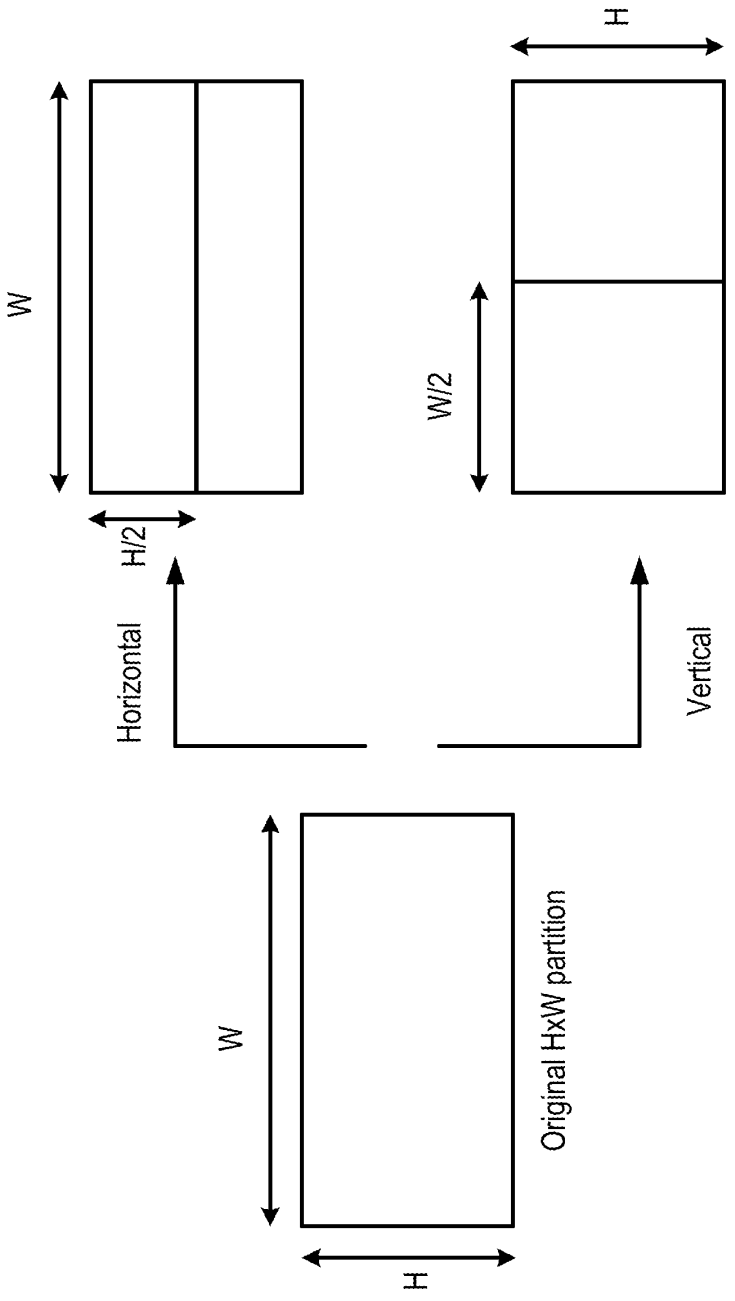

FIG. 11

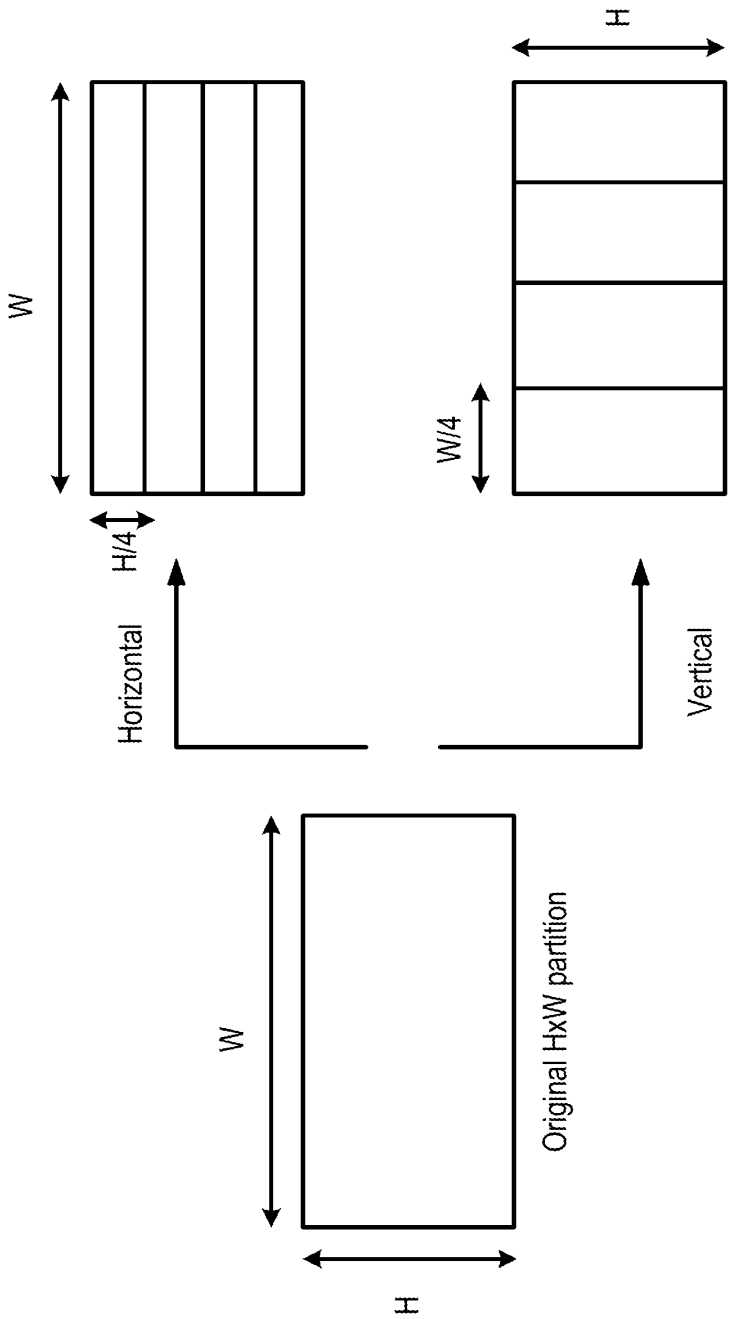

FIG. 12

```
{
    { 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,
    
```

Continue from FIG. 13A

{ 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, -bb, -au, -aq, -ax, -be, ay, ar, at, ba, -bc, -av, -ap, -aw, -bd, az, as }

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

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

{ 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, bt, -cb, -bj, -ck, bk, ca, -bu, -bq, ce, bg, ch, -bm, -bx, bx, bm, -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, 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 }

FIG. 13B

Continue from FIG. 13B

```
{ 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, -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 }
{ 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, 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, -ca, -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,
as, 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, -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 }
```

FIG. 13C

Continue from FIG. 13C

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

FIG. 13D

Continue from FIG. 13D

```
  { 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, -bz,  bm, -bh,  bh, -bm,  bz, -bw,  cb, -cg, -ck,  cf, -ca,  bv, -bq,  bi, -bg,  bl, -bn,  bs, -bx,  cc, -ch, -cj,  ce,
-bz,  bm, -bp,  bk, -bf,  bj, -bo,  bt, -by,  cd, -ci }
  { ao, -am,  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, -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, -bm,  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. 13E

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

Transform Type	Basis function $T_i(j)$, $i, j=0, 1, \ldots, N-1$
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

1500

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

Transform Type
0 – DCT2
1 – DST7
2 – DCT8

FIG. 15

4-point DST-7:

```
{  a,  b,  c,  d }
{  c,  c,  0, -c }
{  d, -a, -c,  b }
{  b, -d,  c, -a }
```

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

FIG. 16A

8-point DST-7:

```
{  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,}
```

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

FIG. 16B

16-point DST-7:

```
{  a,  b,  c,  d,  e,  f,  g,  h,  i,  j,  k,  l,  m,  n,  o,  p,}
{  c,  f,  i,  l,  o,  o,  l,  i,  f,  c,  0, -c, -f, -i, -l, -o,}
{  e,  j,  o,  m,  h,  c, -b, -g, -l, -p, -k, -f, -a,  d,  i,  n,}
{  g,  n,  l,  e, -b, -i, -p, -j, -c,  d,  k,  o,  h,  a, -f, -m,}
{  i,  o,  f, -c, -l, -l, -c,  f,  o,  i,  0, -i, -o, -f,  c,  l,}
{  k,  k,  0, -k, -k,  0,  k,  k,  0, -k, -k,  0,  k,  k,  0, -k,}
{  m,  g, -f, -n, -a,  l,  h, -e, -o, -b,  k,  i, -d, -p, -c,  j,}
{  o,  c, -l, -f,  i,  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,  o, -c, -i,  l,  0, -l,  i,  c, -o,  f,}
{  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,}
```

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, 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. 16D

4-point DCT-8:

```
{  a,  b,  c,  d,}
{  b,  0, -b, -b,}
{  c, -b, -d,  a,}
{  d, -b,  a, -c,}
```

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

FIG. 17A

8-point DCT-8:

```
{  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,}
```

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

FIG. 17B

16-point DCT-8:

```
{  a,  b,  c,  d,  e,  f,  g,  h,  i,  j,  k,  l,  m,  n,  o,  p,}
{  b,  e,  h,  k,  n,  0, -n, -k, -h, -e, -b, -b, -e, -h, -k, -n,}
{  c,  h,  m, -p, -k, -f, -a, -e, -j, -o,  n,  i,  d,  b,  g,  l,}
{  d,  k, -p, -i, -b, -f, -m,  n,  g,  a,  h,  o, -l, -e, -c, -j,}
{  e,  n, -k, -b, -h,  0,  h,  b,  k, -n, -e, -e, -n,  k,  b,  h,}
{  f,  0, -f, -f,  0,  f,  f,  0, -f, -f,  0,  f,  f,  0, -f, -f,}
{  g, -n, -a, -m,  h,  f, -o, -b, -l,  i,  e, -p, -c, -k,  j,  d,}
{  h, -k, -e,  n,  b,  0, -b, -n,  e,  k, -h, -h,  k,  e, -n, -b,}
{  i, -h, -j,  g,  k, -f, -l,  e,  m, -d, -n,  c,  o, -b, -p,  a,}
{  j, -e, -o,  a, -n, -f,  i,  k, -d, -p,  b, -m, -g,  h,  l, -c,}
{  k, -b,  n,  h, -e,  0,  e, -h, -n,  b, -k, -k,  b, -n, -h,  e,}
{  l, -b,  i,  o, -e,  f, -p, -h,  c, -m, -k,  a, -j, -n,  d, -g,}
{  m, -e,  d, -l, -n,  f, -c,  k,  o, -g,  b, -j, -p,  h, -a,  i,}
{  n, -h,  b, -e,  k,  0, -k,  e, -b,  h, -n, -n,  h, -b,  e, -k,}
{  o, -k,  g, -c,  b, -f,  j, -n, -p,  l, -h,  d, -a,  e, -i,  m,}
{  p, -n,  l, -j,  h, -f,  d, -b,  a, -c,  e, -g,  i, -k,  m, -o,}
```

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. 17C

32-point DCT-8:

$$\begin{Bmatrix}
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 \\
a, & e, & h, & i, & l, & o, & p, & s, & v, & y, & -B, & -E, & -H, & -K, & -N, & -Q, & -T, & -W, & -Z, & -C, & -F, & -I, & -L, & -O, & -R, & -U, & -X, & -A, & -D, & -G, & -J, & -M \\
a, & h, & k, & n, & q, & t, & w, & z, & C, & F, & -I, & -L, & -O, & -R, & -U, & -X, & -A, & -D, & -G, & -J, & -M, & -P, & -S, & -V, & -Y, & -B, & -E, & -H, & -K, & -N, & -Q, & -T \\
a, & i, & n, & q, & v, & y, & -B, & -E, & -H, & -K, & -N, & -Q, & -T, & -W, & -Z, & -C, & -F, & -I, & -L, & -O, & -R, & -U, & -X, & -A, & -D, & -G, & -J, & -M, & -P, & -S, & -V \\
a, & l, & q, & v, & A, & F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A \\
a, & o, & t, & y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D \\
a, & r, & w, & B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G \\
a, & u, & z, & E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J \\
a, & x, & C, & H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M \\
a, & A, & F, & K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P \\
a, & B, & G, & L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q \\
a, & C, & H, & M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R \\
a, & D, & I, & N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S \\
a, & E, & J, & O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T \\
a, & F, & K, & P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U \\
a, & G, & L, & Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V \\
a, & H, & M, & R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W \\
a, & I, & N, & S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X \\
a, & J, & O, & T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y \\
a, & K, & P, & U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z \\
a, & L, & Q, & V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A \\
a, & M, & R, & W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B \\
a, & N, & S, & X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C \\
a, & O, & T, & Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D \\
a, & P, & U, & Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E \\
a, & Q, & V, & A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F \\
a, & R, & W, & B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G \\
a, & S, & X, & C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H \\
a, & T, & Y, & D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I \\
a, & U, & Z, & E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J \\
a, & V, & A, & F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K \\
a, & W, & B, & G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L \\
a, & X, & C, & H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M \\
a, & Y, & D, & I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N \\
a, & Z, & E, & J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O \\
a, & A, & G, & M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R \\
a, & B, & H, & N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S \\
a, & C, & I, & O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T \\
a, & D, & J, & P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V, & -A, & -F, & -K, & -P, & -U \\
a, & E, & K, & Q, & -V, & -A, & -F, & -K, & -P, & -U, & -Z, & -E, & -J, & -O, & -T, & -Y, & -D, & -I, & -N, & -S, & -X, & -C, & -H, & -M, & -R, & -W, & -B, & -G, & -L, & -Q, & -V
\end{Bmatrix}$$

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 } = {90,90,89,88,88,86,85,84,82,80,78,77,74,72,68,66,63,60,56,53,50,45,42,38,34,30,26,21,17,13,9,4}

FIG. 17D

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

FIG. 18

1900

Mapping from intra prediction mode to transform set index

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

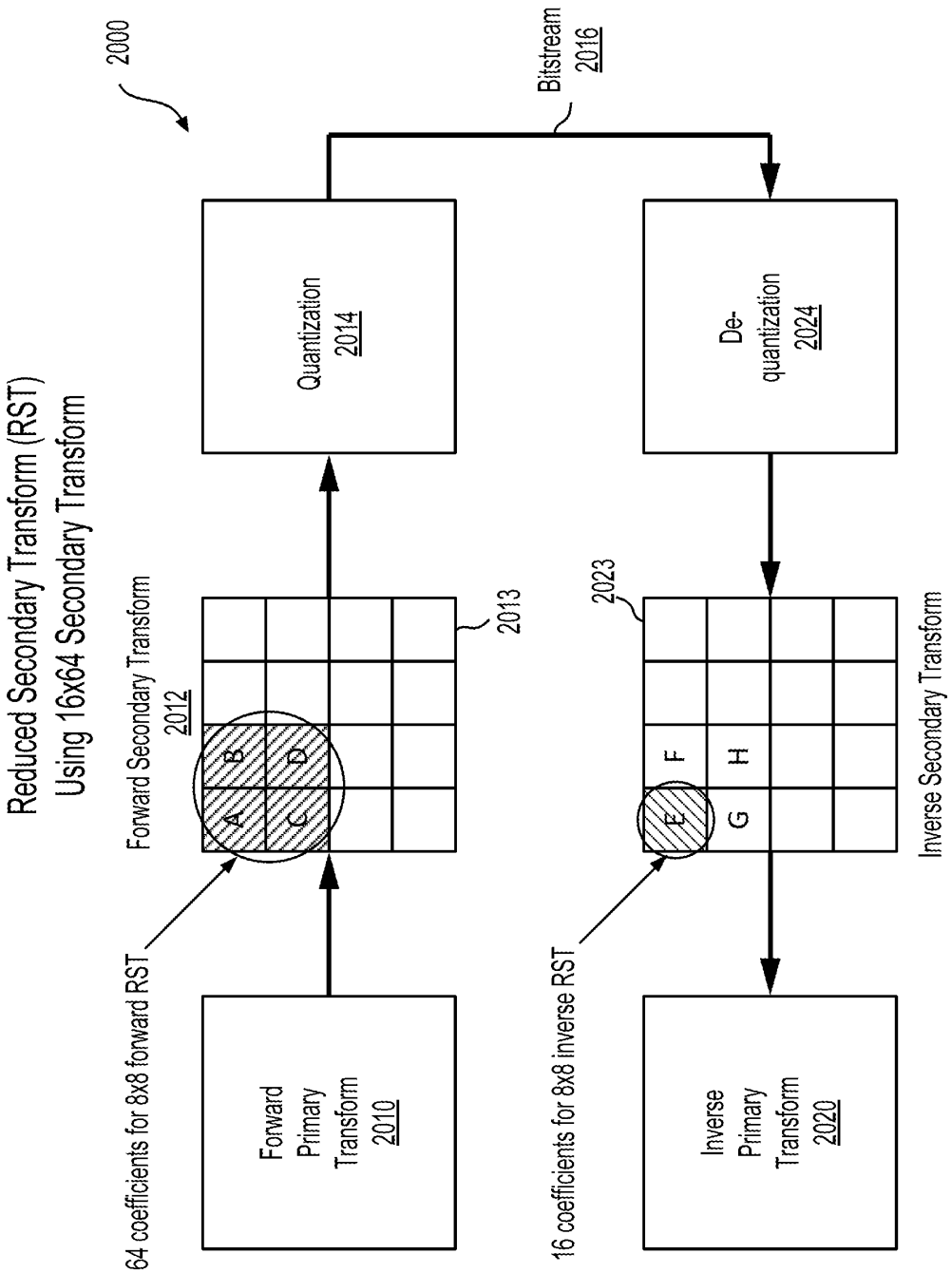
Reduced Secondary Transform (RST)
Using 16x64 Secondary Transform
2000
Bitstream
2016
Quantization
2014
De-
quantization
2024
Forward Secondary Transform
2012
2013
2023
A
B
C
D
E
F
G
H
64 coefficients for 8x8 forward RST
16 coefficients for 8x8 inverse RST
Forward
Primary
Transform
2010
Inverse
Primary
Transform
2020
Inverse Secondary Transform
2022

FIG. 20

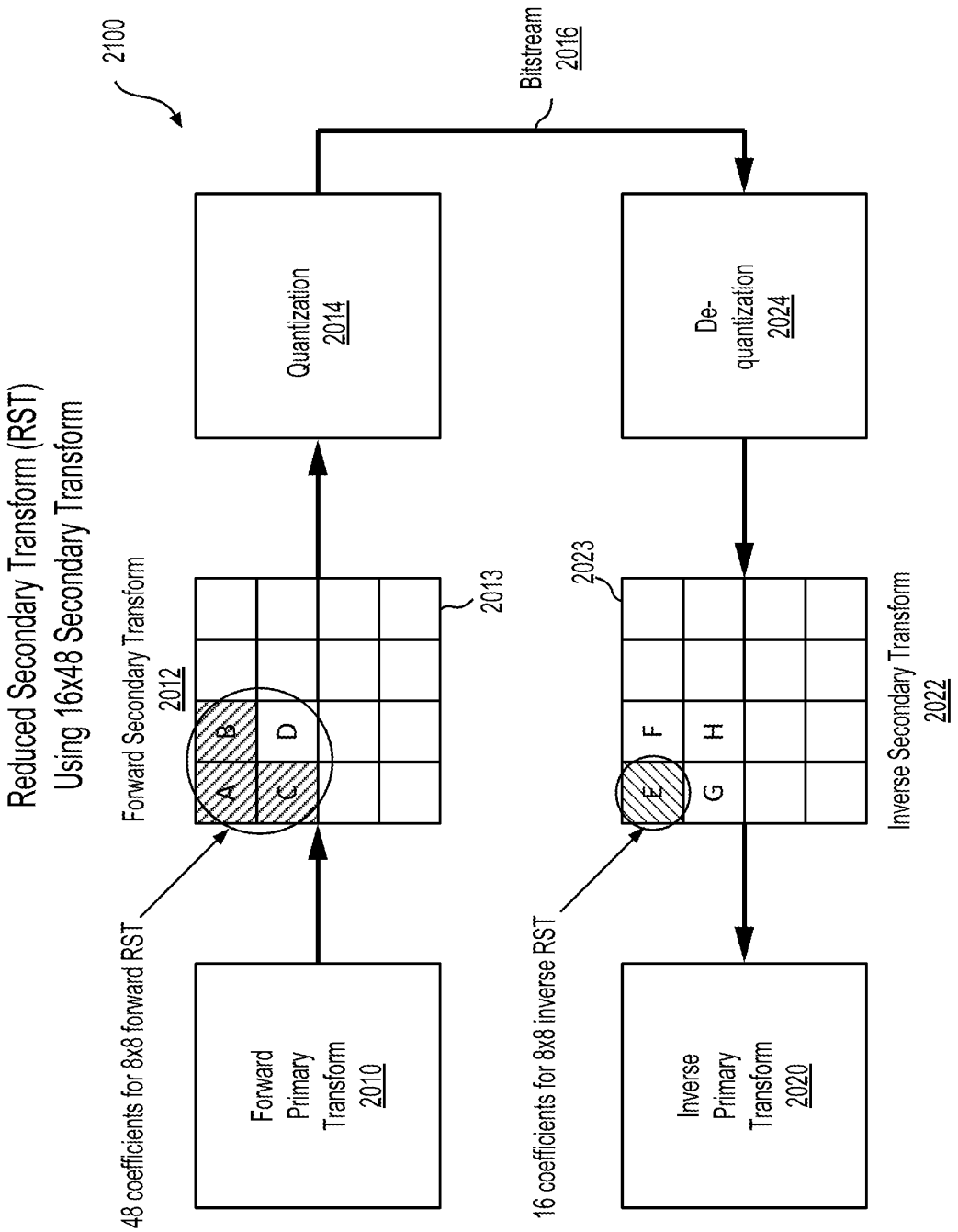
Reduced Secondary Transform (RST)
Using 16x48 Secondary Transform
2100
Bitstream
2016
Quantization
2014
De-
quantization
2024
Forward Secondary Transform
2012
2013
2023
A
B
C
D
E
F
G
H
48 coefficients for 8x8 forward RST
16 coefficients for 8x8 inverse RST
Forward
Primary
Transform
2010
Inverse
Primary
Transform
2020
Inverse Secondary Transform
2022

FIG. 21

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	**Descriptor**
.....	
numSigCoeff = 0	
numZeroOutSigCoeff = 0	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
lfnstWidth = (treeType == DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : cbWidth	
lfnstHeight = (treeType == DUAL_TREE_CHROMA) ? cbHeight / SubHeightC : cbHeight	
if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_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)	
lfnst_idx[x0][y0]	ae(v)
}	
}	
}	
}	

2200

FIG. 22

Forward and Inverse Reduced Transform

FIG. 23

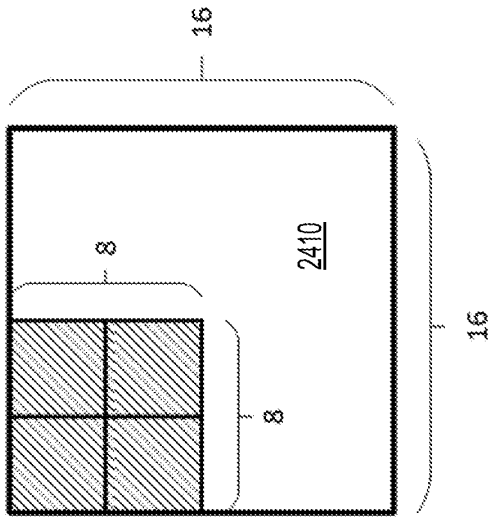

FIG. 24B

FIG. 24A

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

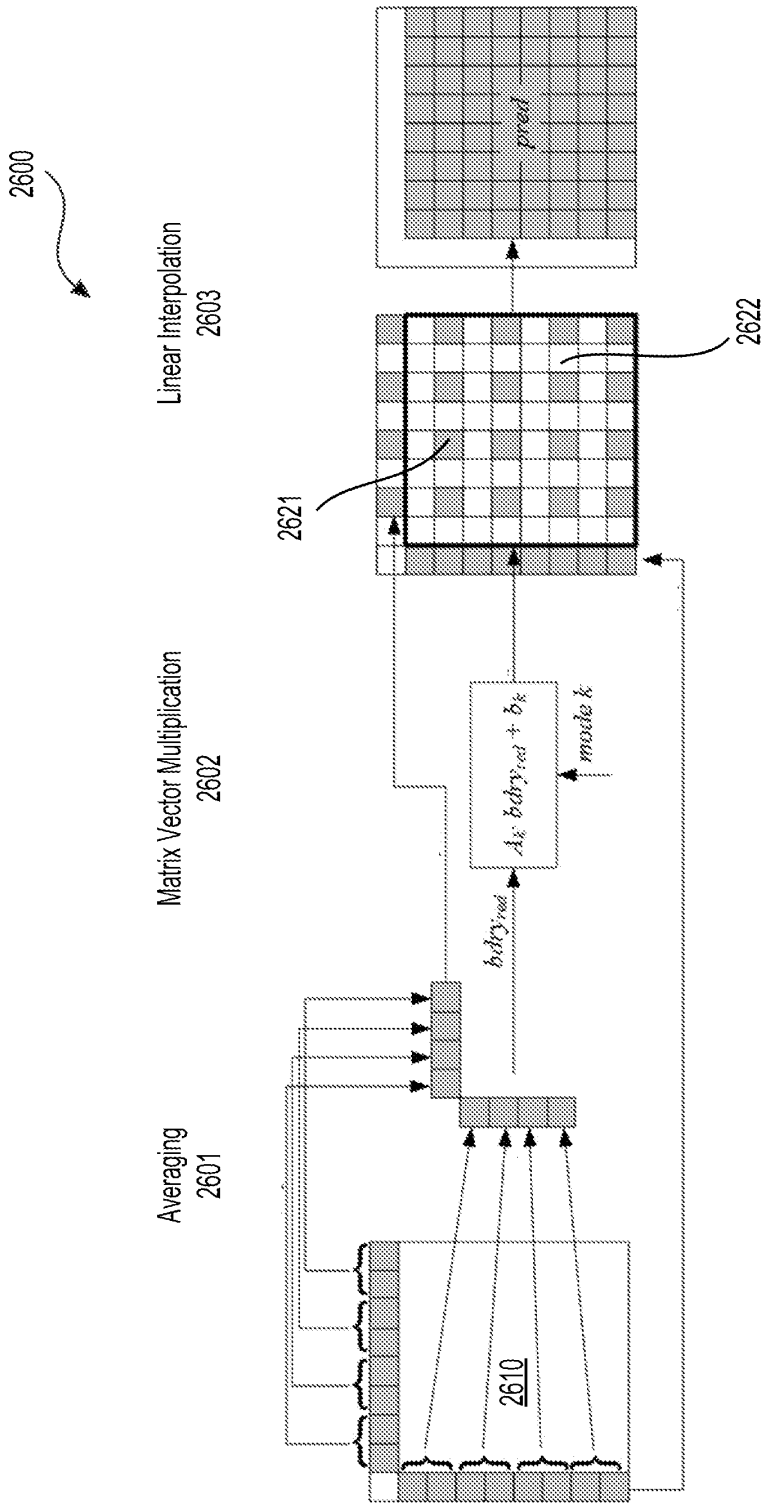
2600
Averaging
2601
Matrix Vector Multiplication
2602
Linear Interpolation
2603
2610
2621
2622
pred
mode k

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

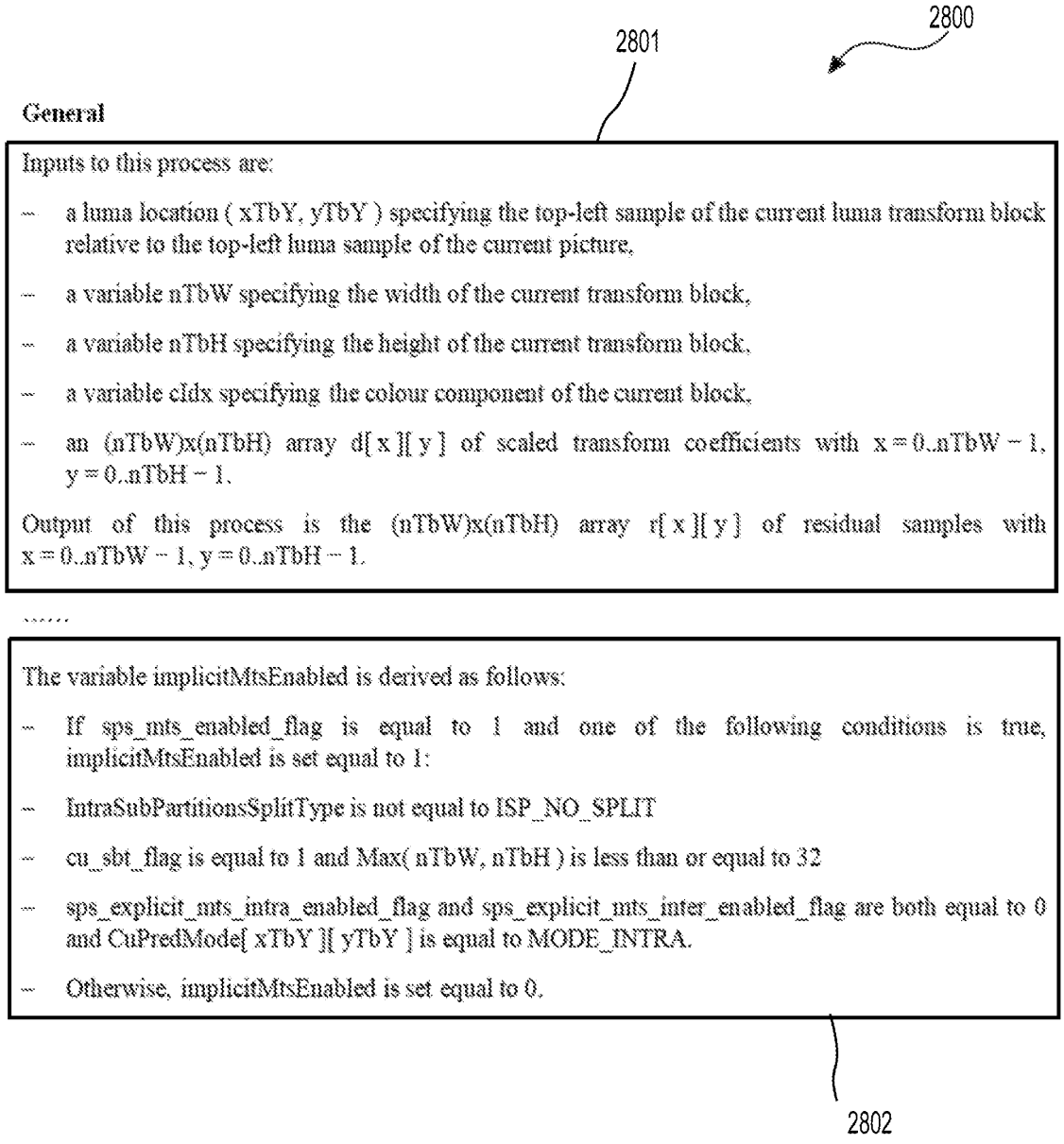
2800
2801
General
Inputs to this process are:
– a luma location (xTbY, yTbY) specifying the top-left sample of the current luma transform block relative to the top-left luma sample of the current picture,
– a variable nTbW specifying the width of the current transform block,
– a variable nTbH specifying the height of the current transform block,
– a variable cIdx specifying the colour component of the current block,
– an (nTbW)x(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)x(nTbH) array r[x][y] of residual samples with x = 0..nTbW − 1, y = 0..nTbH − 1.
.....
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.
2802

FIG. 28A

2800

2803 *Continue from FIG. 28A*

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

- If cIdx 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:.

  trTypeHor = (nTbW >= 4 && nTbW <= 16) ? 1 : 0 (3-1)

  trTypeVer = (nTbH >= 4 && nTbH <= 16) ? 1 : 0 (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]

2804

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

2805

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~~ is 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_lfnst_enabled_flag is equal to 0.
- Otherwise, implicitMtsEnabled is set equal to 0.

FIG. 30

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

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 lfnst_idx[x0][y0] are both equal to 0.
- Otherwise, implicitMtsEnabled is set equal to 0.

FIG. 32

3300

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], intra_mip_flag[x0][y0] and lfnst_idx[x0][y0] are all equal to 0.
- Otherwise, implicitMtsEnabled is set equal to 0.

FIG. 33

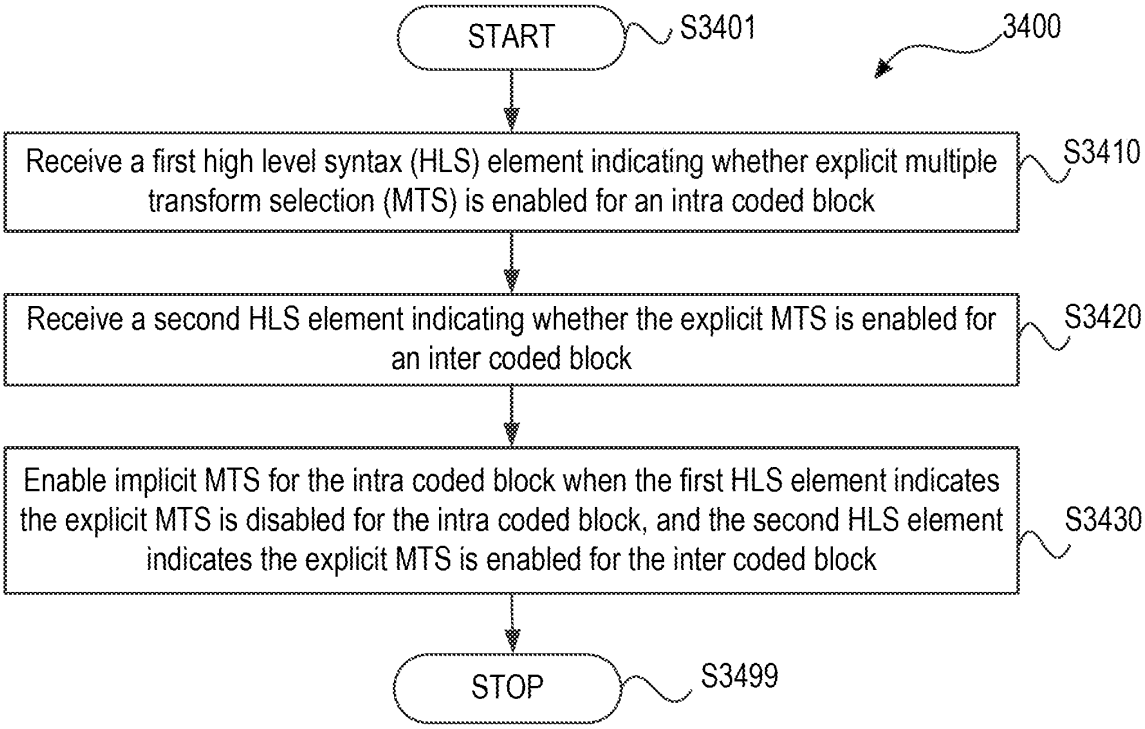

FIG. 34

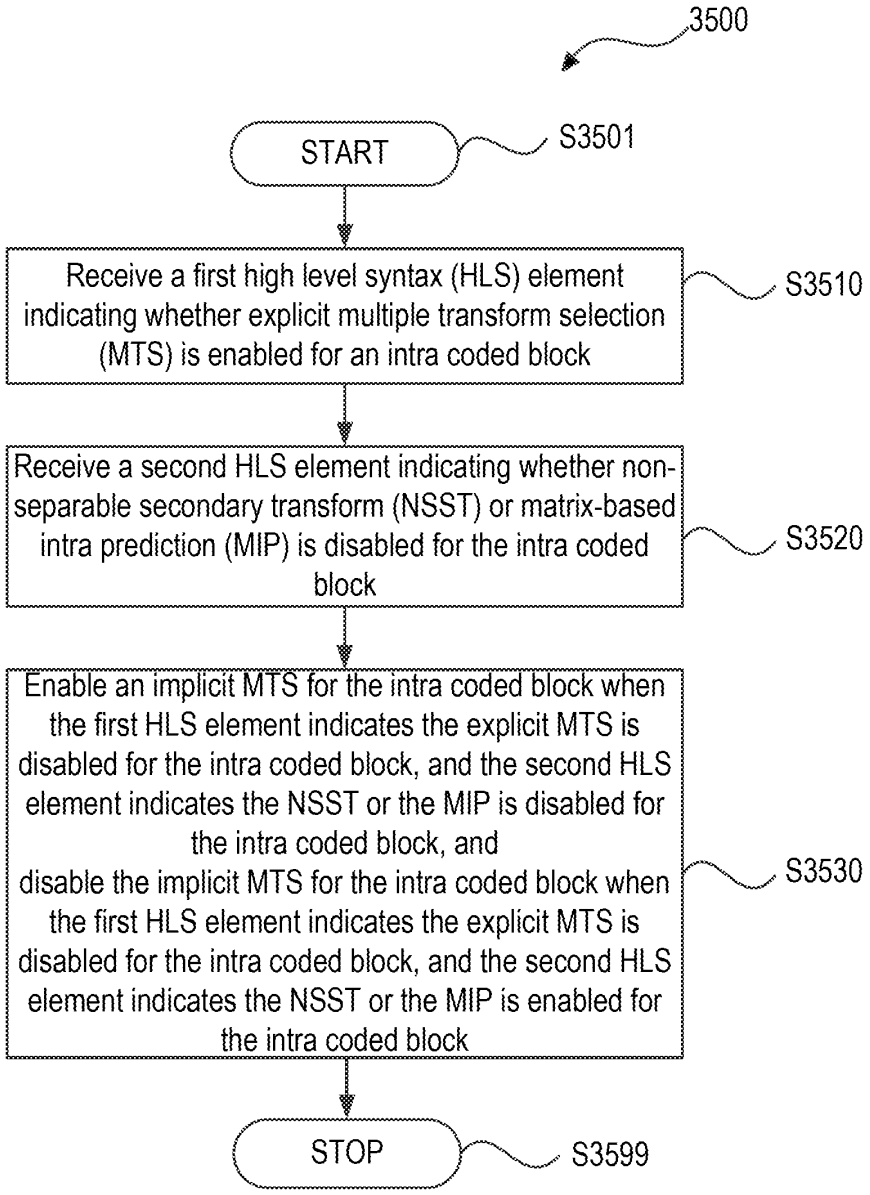
3500
START
S3501
Receive a first high level syntax (HLS) element indicating whether explicit multiple transform selection (MTS) is enabled for an intra coded block
S3510
Receive a second HLS element indicating whether non-separable secondary transform (NSST) or matrix-based intra prediction (MIP) is disabled for the intra coded block
S3520
Enable an implicit MTS for the intra coded block when the first HLS element indicates the explicit MTS is disabled for the intra coded block, and the second HLS element indicates the NSST or the MIP is disabled for the intra coded block, and
disable the implicit MTS for the intra coded block when the first HLS element indicates the explicit MTS is disabled for the intra coded block, and the second HLS element indicates the NSST or the MIP is enabled for the intra coded block
S3530
STOP
S3599

FIG. 35

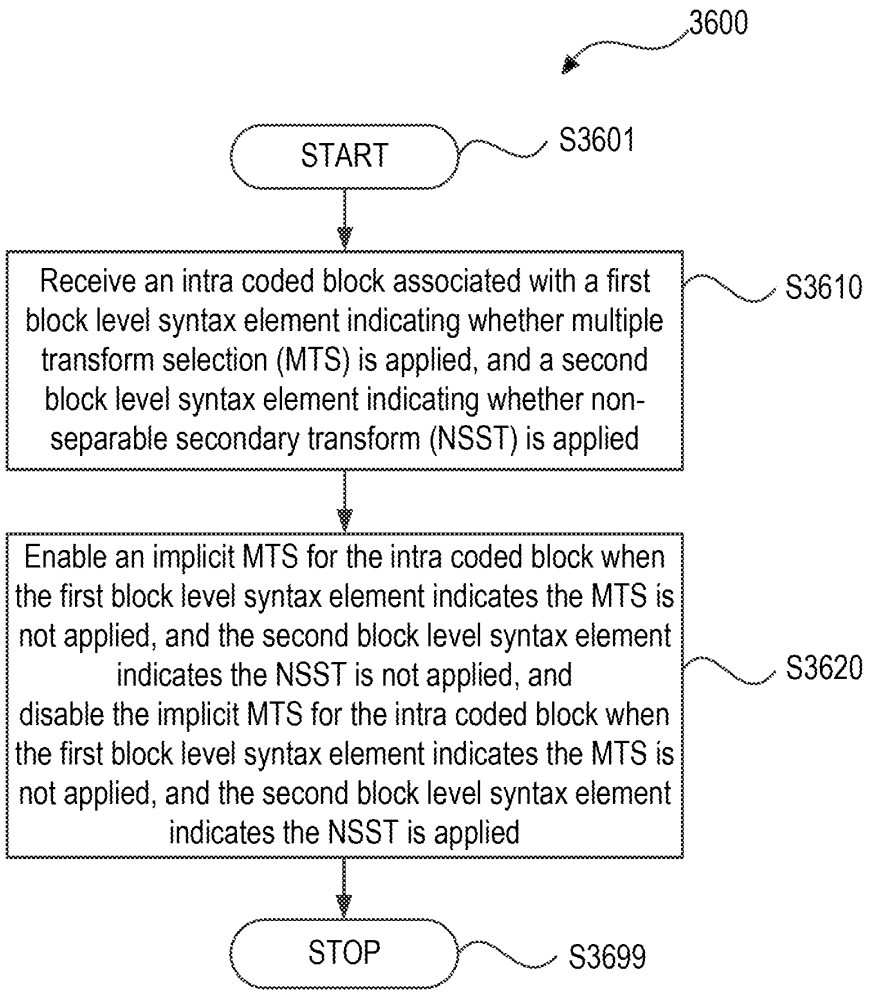
3600
START
S3601
Receive an intra coded block associated with a first block level syntax element indicating whether multiple transform selection (MTS) is applied, and a second block level syntax element indicating whether non-separable secondary transform (NSST) is applied
S3610
Enable an implicit MTS for the intra coded block when the first block level syntax element indicates the MTS is not applied, and the second block level syntax element indicates the NSST is not applied, and
disable the implicit MTS for the intra coded block when the first block level syntax element indicates the MTS is not applied, and the second block level syntax element indicates the NSST is applied
S3620
STOP
S3699

FIG. 36

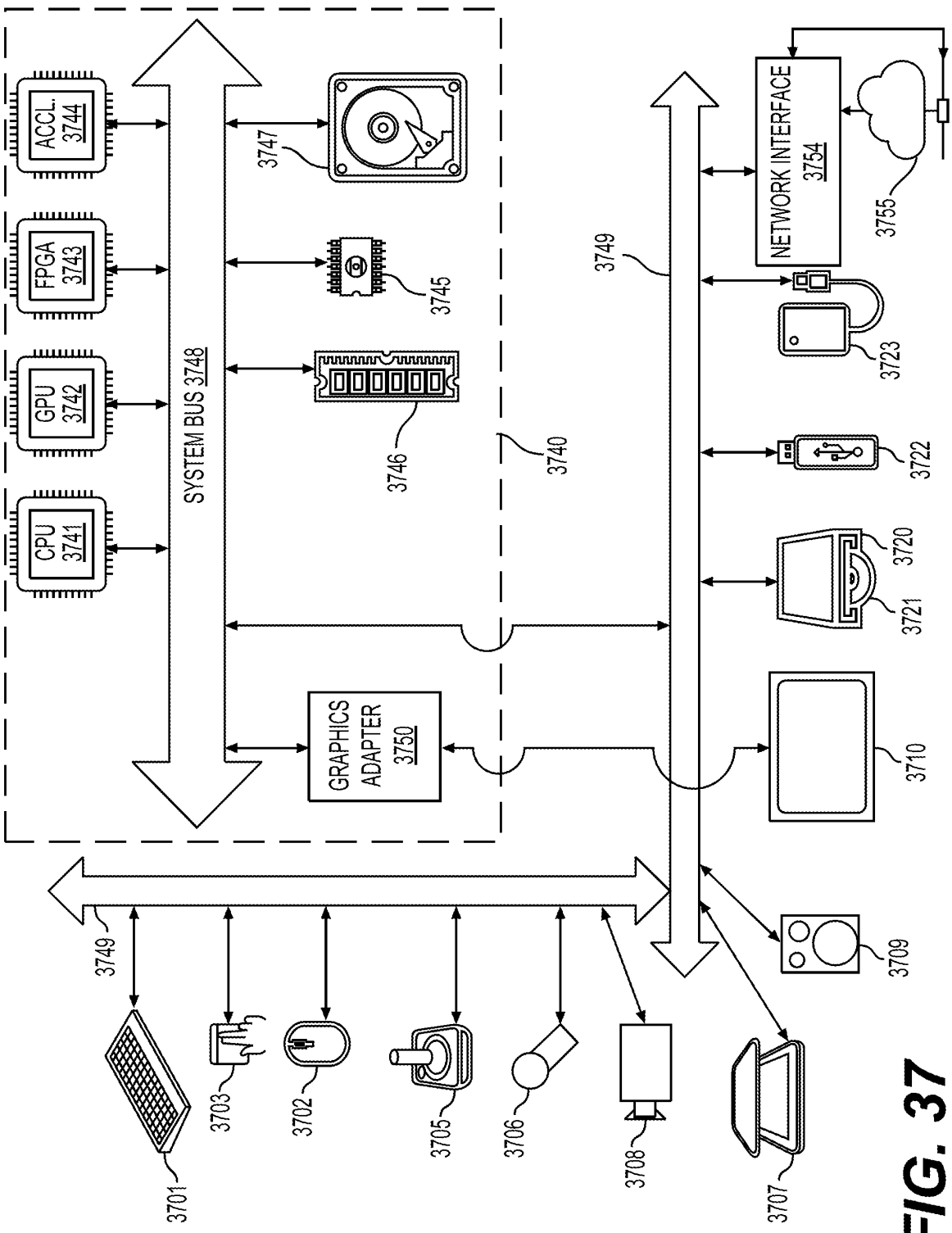
ACCL. 3744
FPGA 3743
GPU 3742
CPU 3741
SYSTEM BUS 3748
3747
3745
3746
3740
GRAPHICS ADAPTER 3750
3749
NETWORK INTERFACE 3754
3755
3723
3722
3720
3721
3710
3709
3701
3703
3702
3705
3706
3708
3707

FIG. 37