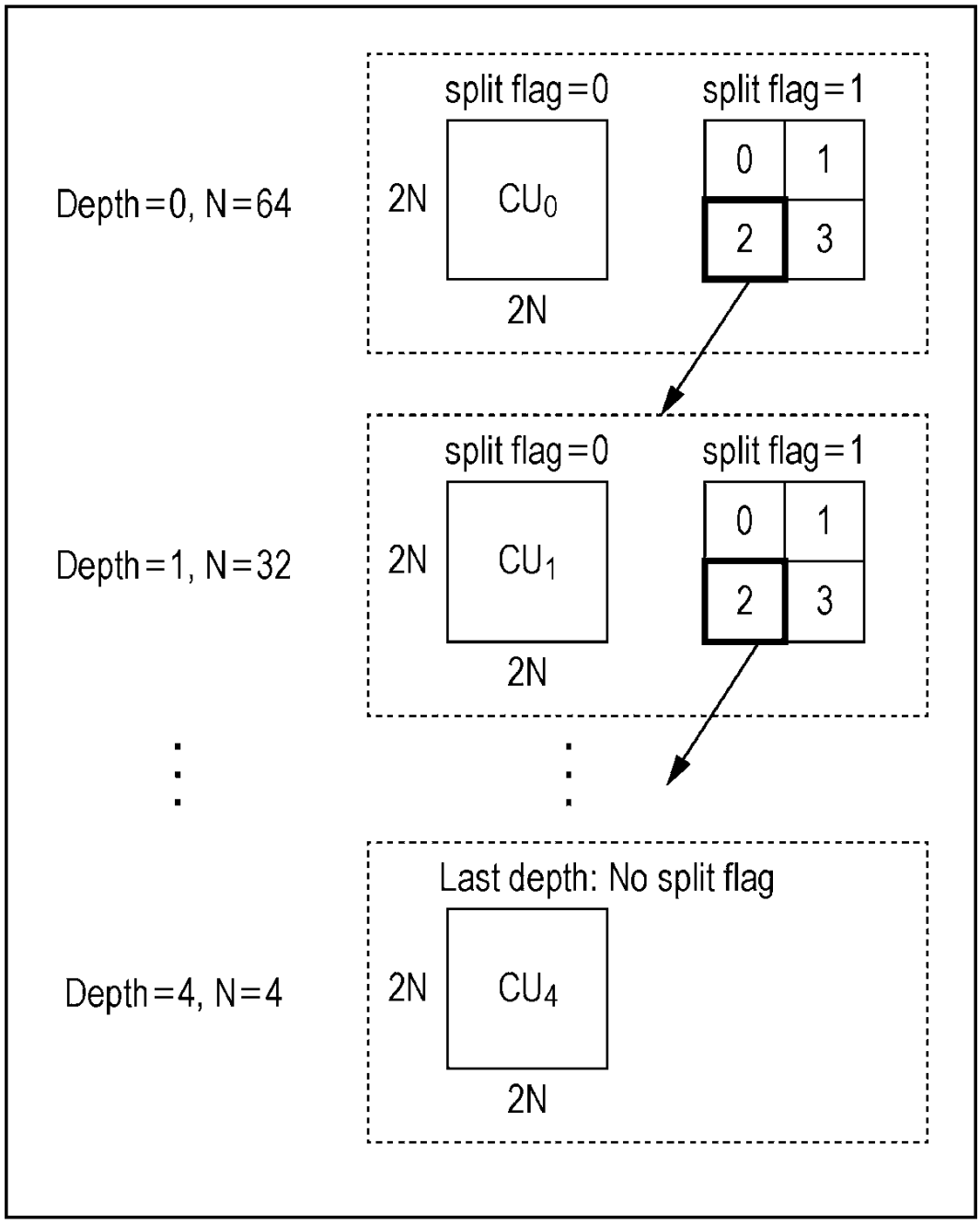
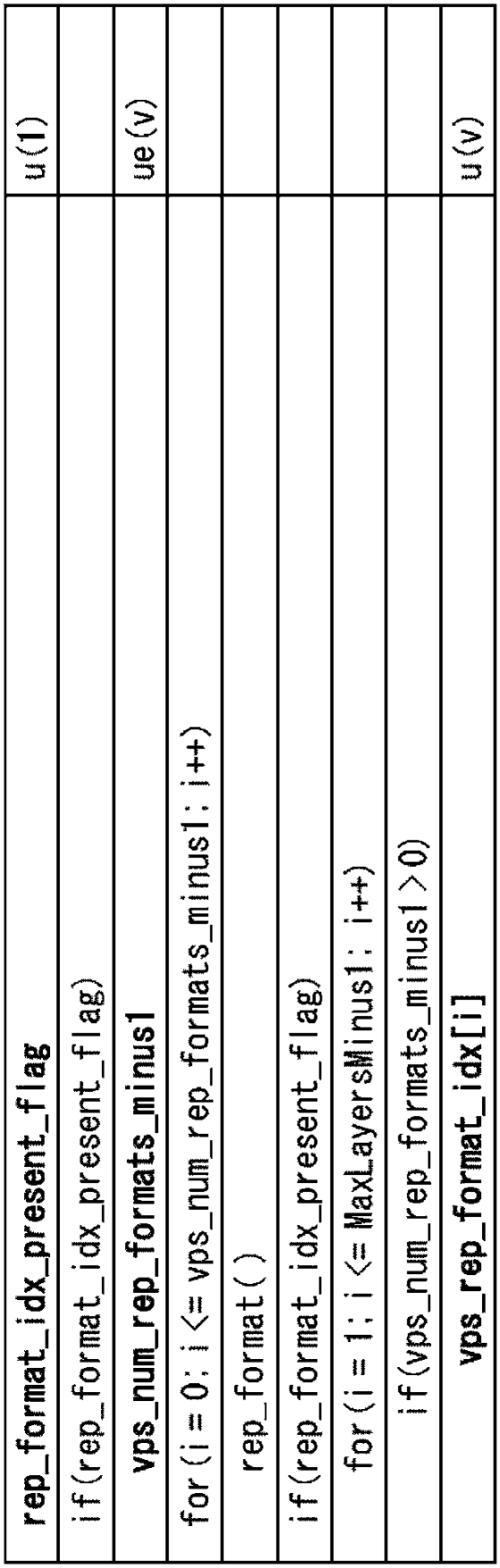
[Fig. 1]

FIG. 1


Depth=0, N=64
split flag=0
split flag=1
2N
CU0
2N
0
1
2
3
Depth=1, N=32
split flag=0
split flag=1
2N
CU1
2N
0
1
2
3
Last depth: No split flag
Depth=4, N=4
2N
CU4
2N

[Fig. 2]

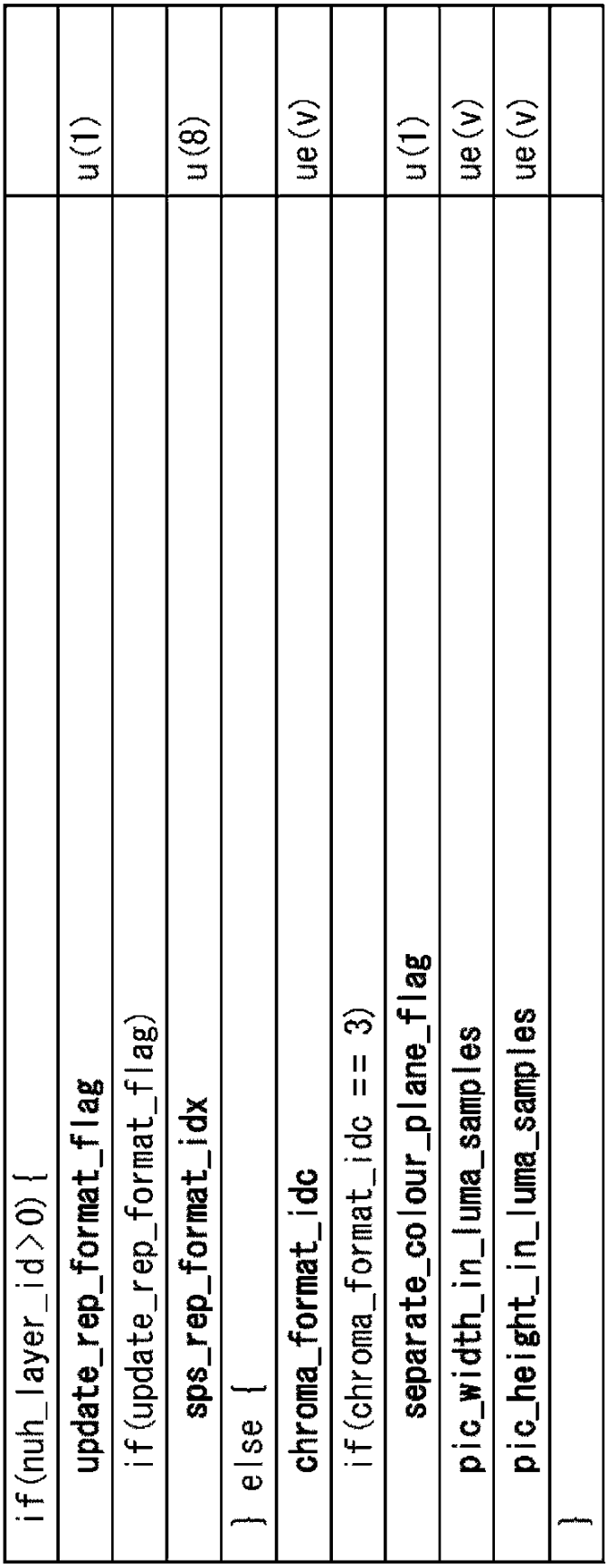
FIG. 2



| **rep_format_idx_present_flag** | u(1) |
|---|---|
| if(rep_format_idx_present_flag) | |
| **vps_num_rep_formats_minus1** | ue(v) |
| for (i = 0; i <= vps_num_rep_formats_minus1; i++) | |
| rep_format( ) | |
| if(rep_format_idx_present_flag) | |
| for (i = 1; i <= MaxLayersMinus1; i++) | |
| if(vps_num_rep_formats_minus1 > 0) | |
| **vps_rep_format_idx[i]** | u(v) |

[Fig. 3]

FIG. 3



| | |
|---|---|
| if(nuh_layer_id>0){ | |
| update_rep_format_flag | u(1) |
| if(update_rep_format_flag) | |
| sps_rep_format_idx | u(8) |
| } else { | |
| chroma_format_idc | ue(v) |
| if(chroma_format_idc == 3) | |
| separate_colour_plane_flag | u(1) |
| pic_width_in_luma_samples | ue(v) |
| pic_height_in_luma_samples | ue(v) |
| } | |

# FIG. 4

[Fig. 4]

**VPS**

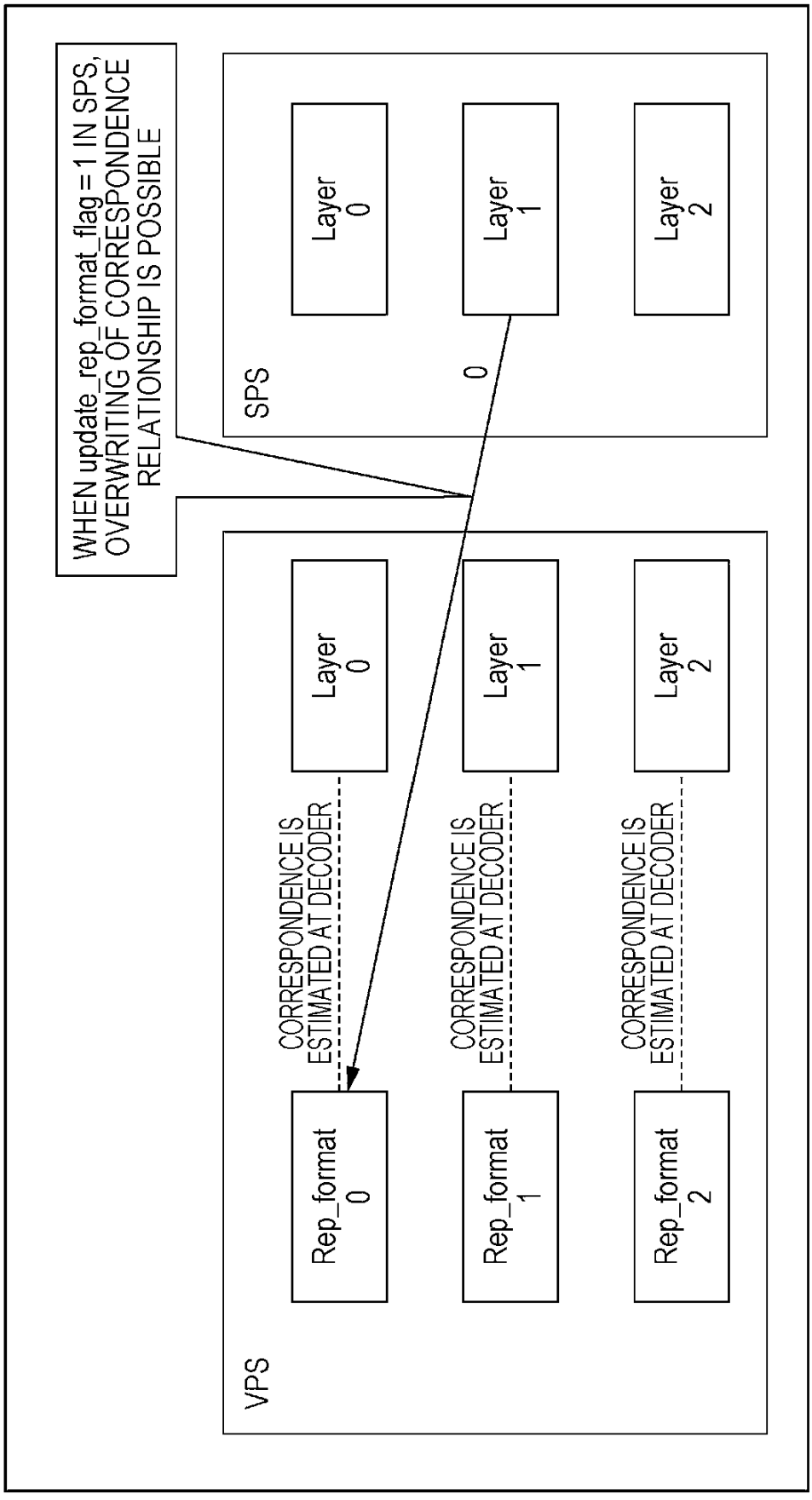
- **rep_format_idx_present_flag** equal to 1 specifies that the syntax elements vps_num_rep_formats_minus1 and vps_rep_format_idx[i] are present. rep_format_idx_present_flag equal to 0 specifies that the syntax elements vps_num_rep_formats_minus1 and vps_rep_format_idx[i] are not present.
- **vps_num_rep_formats_minus1** plus 1 specifies the number of the following rep_format( ) syntax structures in the VPS. When not present, the value of vps_num_rep_formats_minus1 is inferred to be equal to MaxLayersMinus1. The value of vps_num_rep_formats_minus1 shall be in the range of 0 to 255, inclusive.
- **vps_rep_format_idx**[i] specifies the index, into the list of rep_format( )syntax structures in the VPS, of the rep_format( )syntax structure that applies to the layer with nuh_layer_id equal to layer_id_in_nuh[i]. When not present, the value of vps_rep_format_idx[i] is inferred to be equal to (rep_format_idx_present_flag ? 0 : i). The value of vps_rep_format_idx[i] shall be in the range of 0 to vps_num_rep_formats_minus1, inclusive. The number of bits used for the representation of vps_rep_format_idx[i] is Ceil(Log2(vps_num_rep_formats_minus1+1)).

**SPS**

- **update_rep_format_flag** equal to 1 specifies that sps_rep_format_idx is present and that the sps_rep_format_idx-th rep_format( )syntax structures in the active VPS applies to the layers that refer to this SPS. update_rep_format_flag equal to 0 specifies that sps_rep_format_idx is not present. When not present, the value of update_rep_format_flag is inferred to be equal to 0.

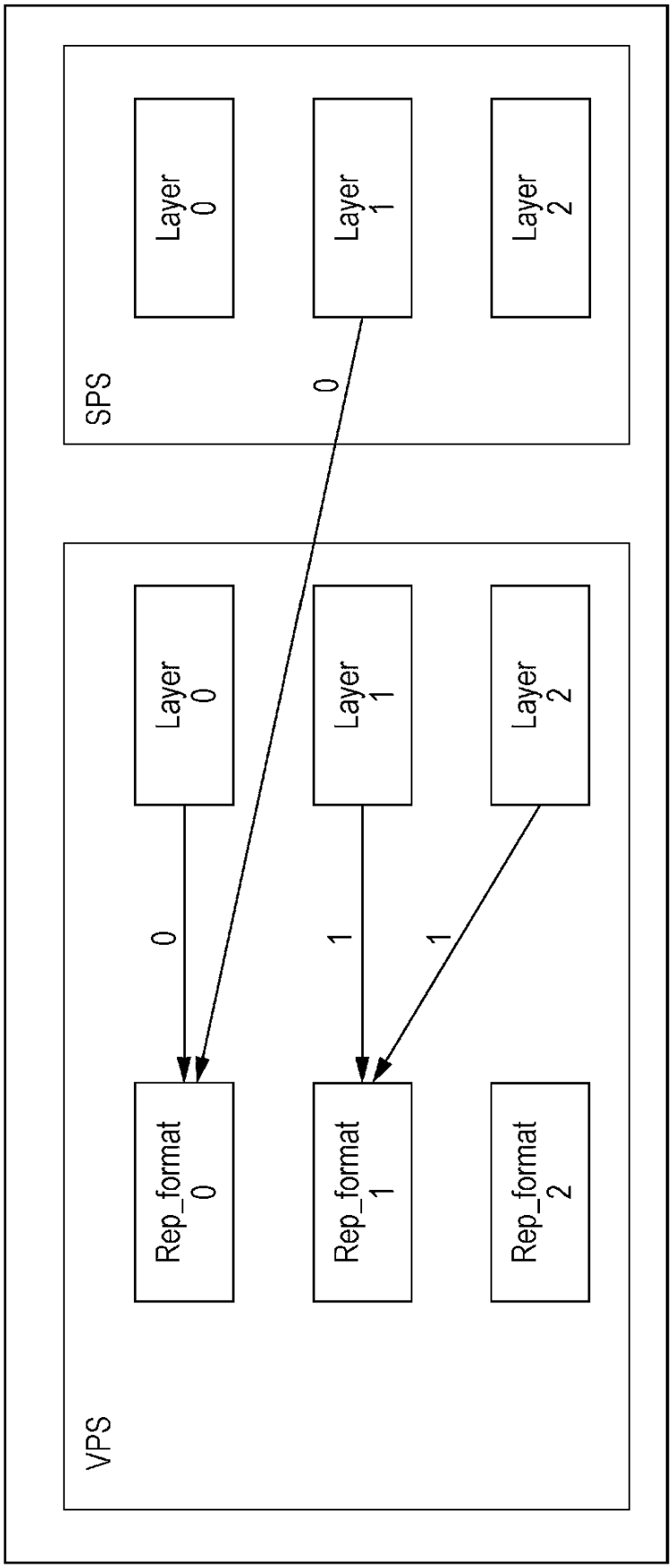
[Fig. 5]

FIG. 5


WHEN update_rep_format_flag = 1 IN SPS, OVERWRITING OF CORRESPONDENCE RELATIONSHIP IS POSSIBLE
SPS
Layer 0
Layer 1
Layer 2
0
VPS
Layer 0
Layer 1
Layer 2
CORRESPONDENCE IS ESTIMATED AT DECODER
CORRESPONDENCE IS ESTIMATED AT DECODER
CORRESPONDENCE IS ESTIMATED AT DECODER
Rep_format 0
Rep_format 1
Rep_format 2

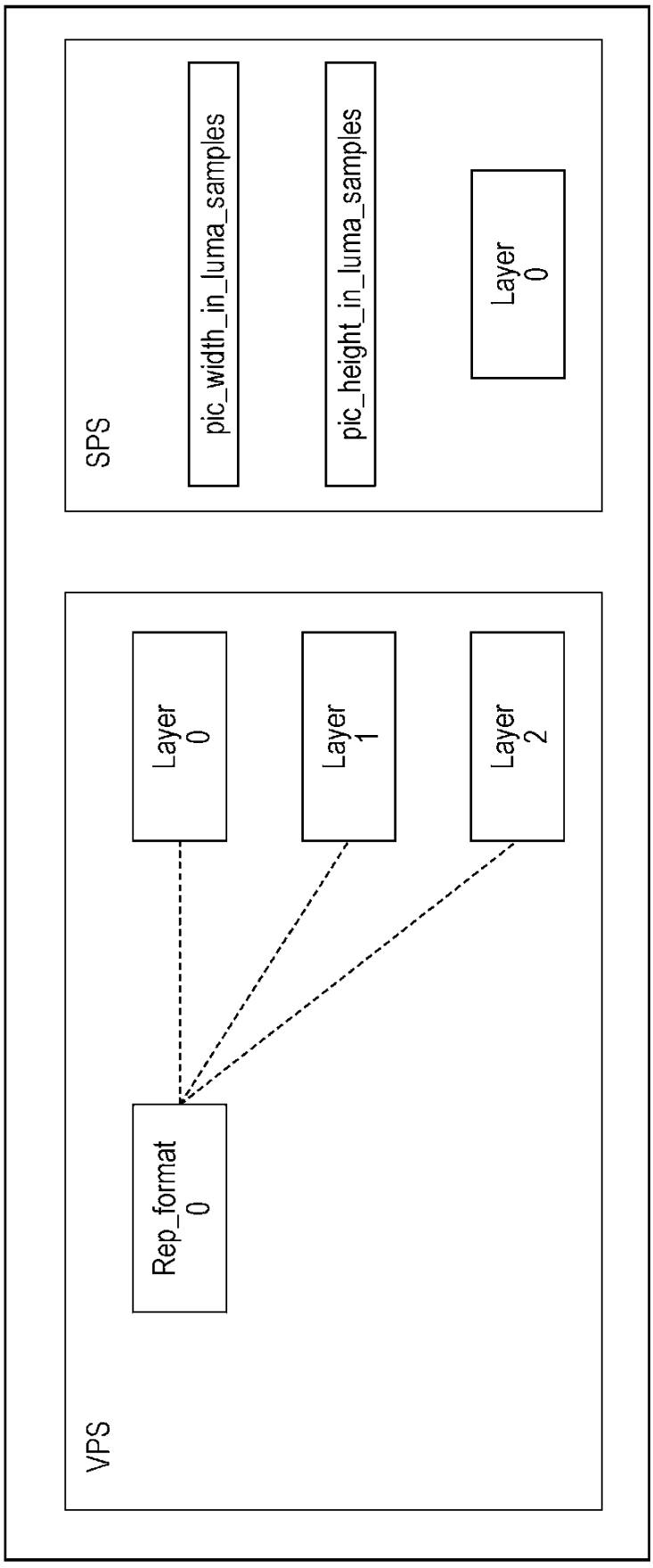
[Fig. 6]

FIG. 6


SPS
Layer 0
Layer 1
Layer 2
0
VPS
Layer 0
Layer 1
Layer 2
0
1
1
Rep_format 0
Rep_format 1
Rep_format 2

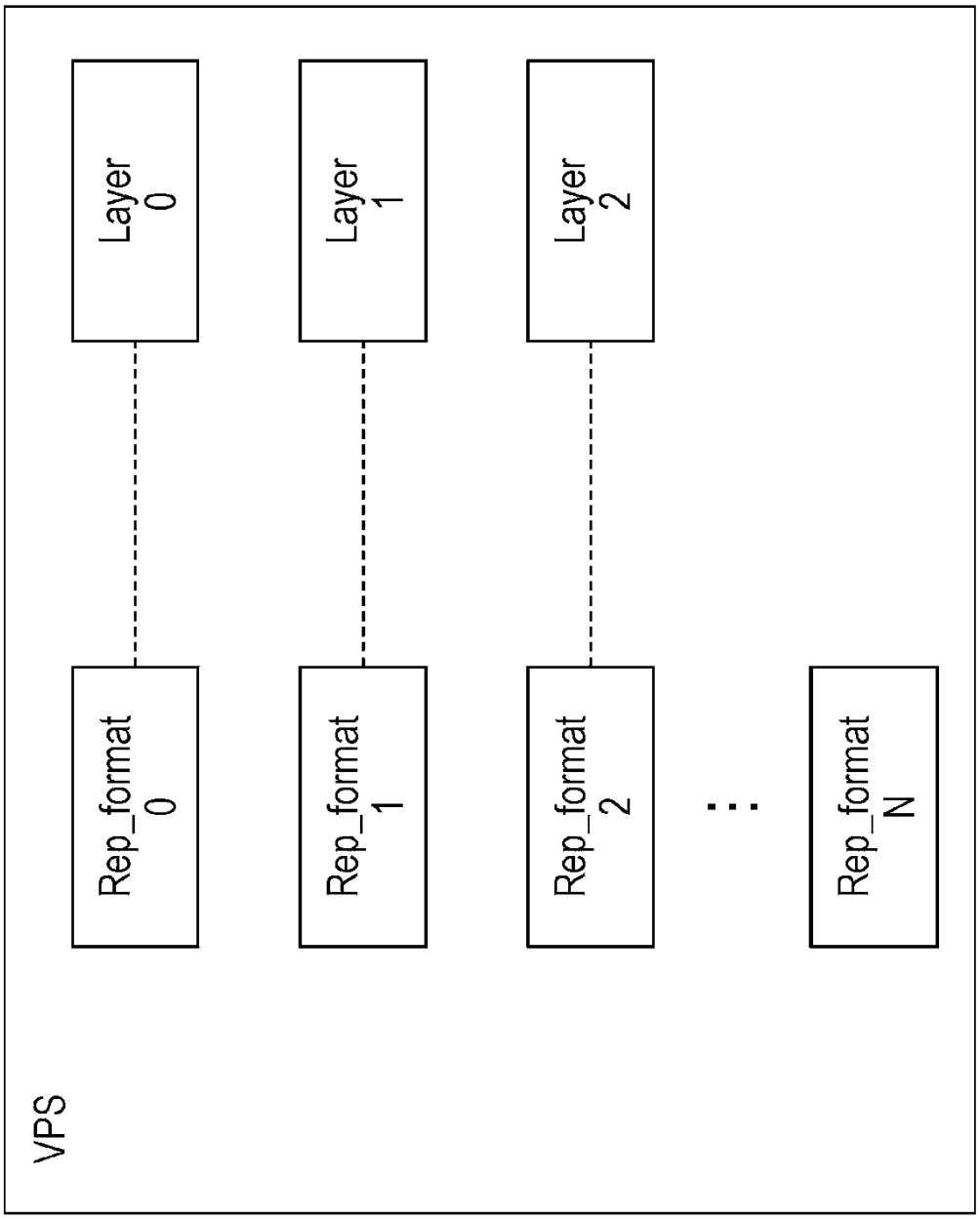
[Fig. 7]

FIG. 7


VPS
Rep_format 0
Layer 0
Layer 1
Layer 2
SPS
pic_width_in_luma_samples
pic_height_in_luma_samples
Layer 0

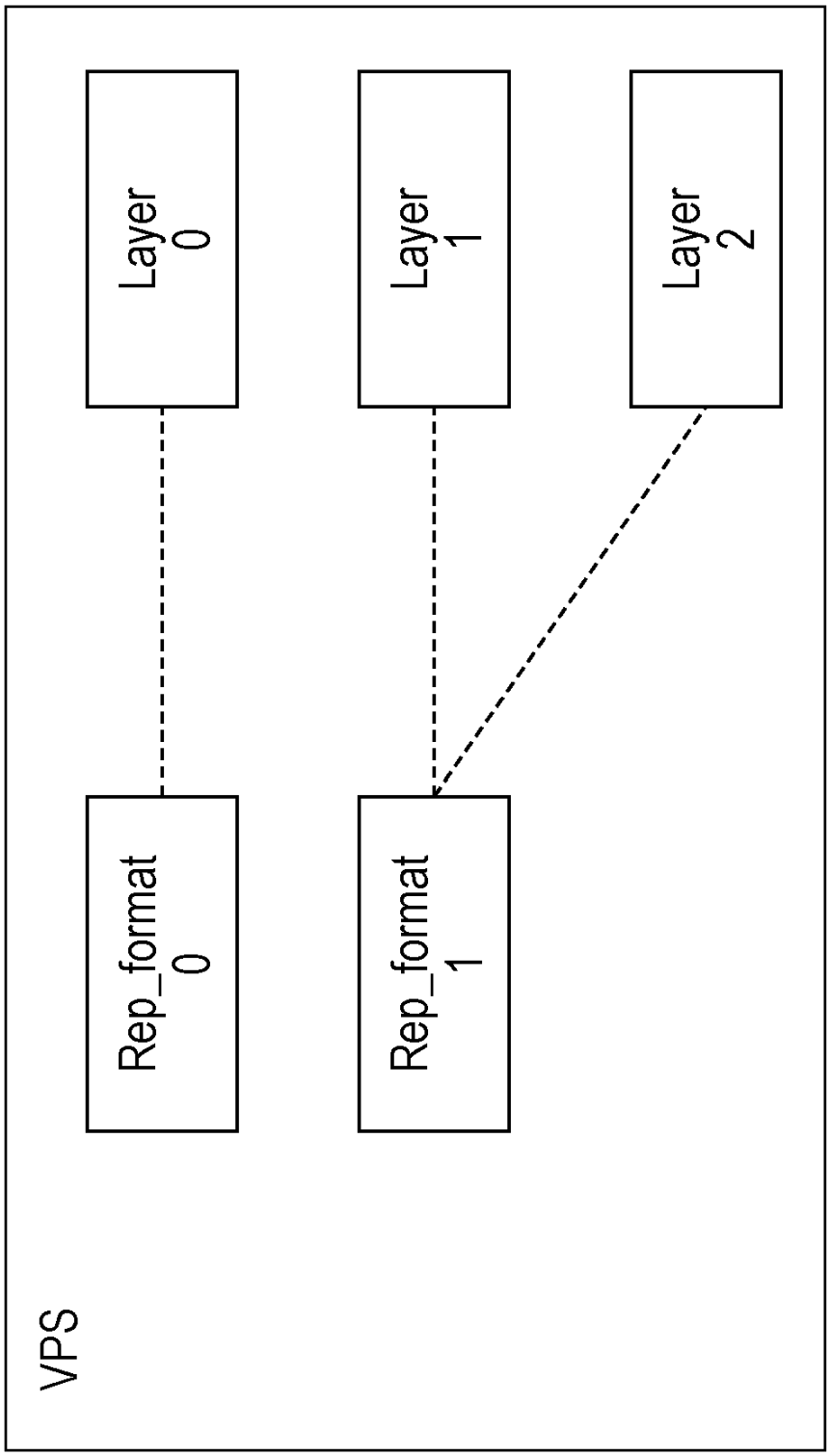
[Fig. 8]

FIG. 8


VPS
Layer 0
Layer 1
Layer 2
Rep_format 0
Rep_format 1
Rep_format 2
Rep_format N

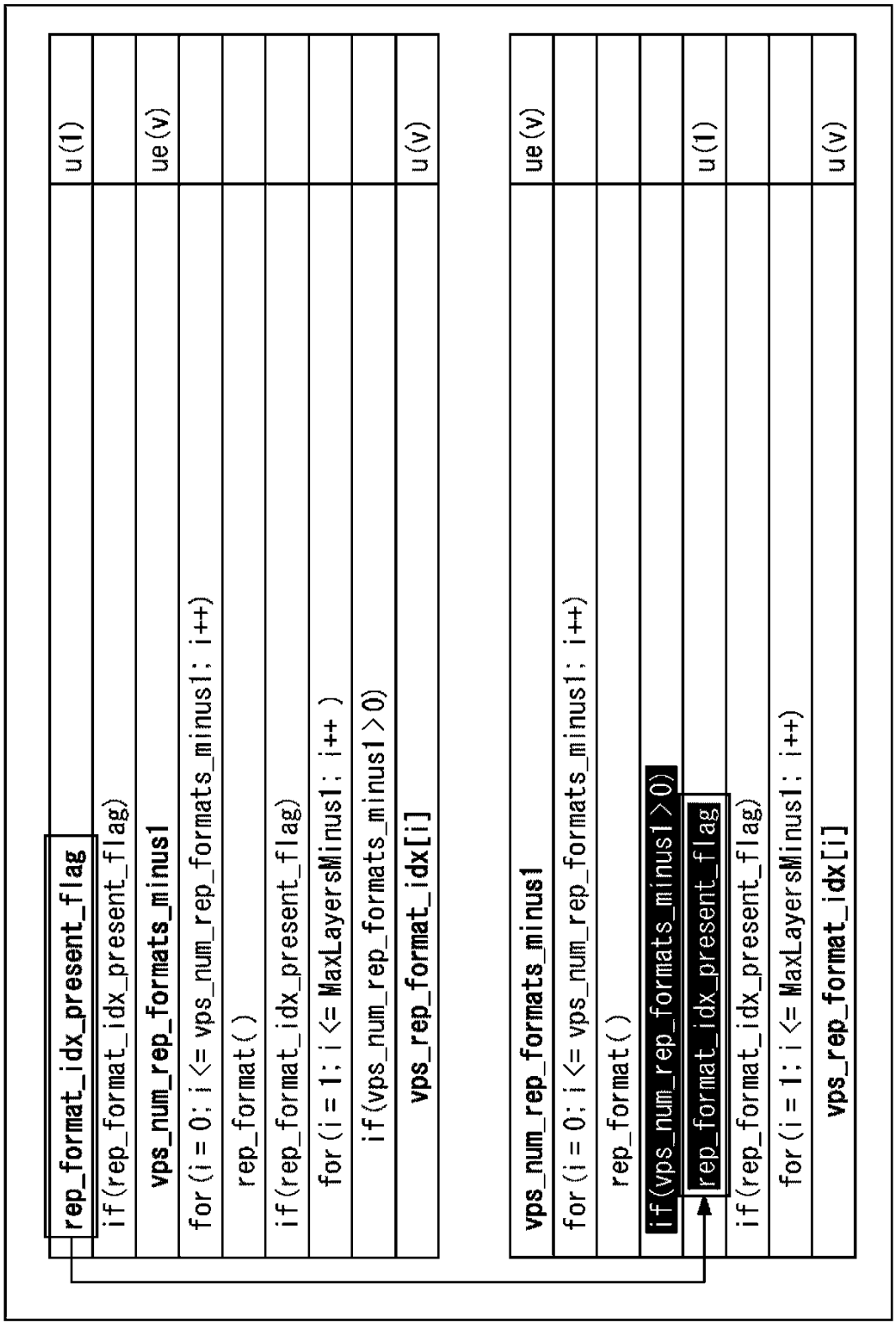
[Fig. 9]

FIG. 9



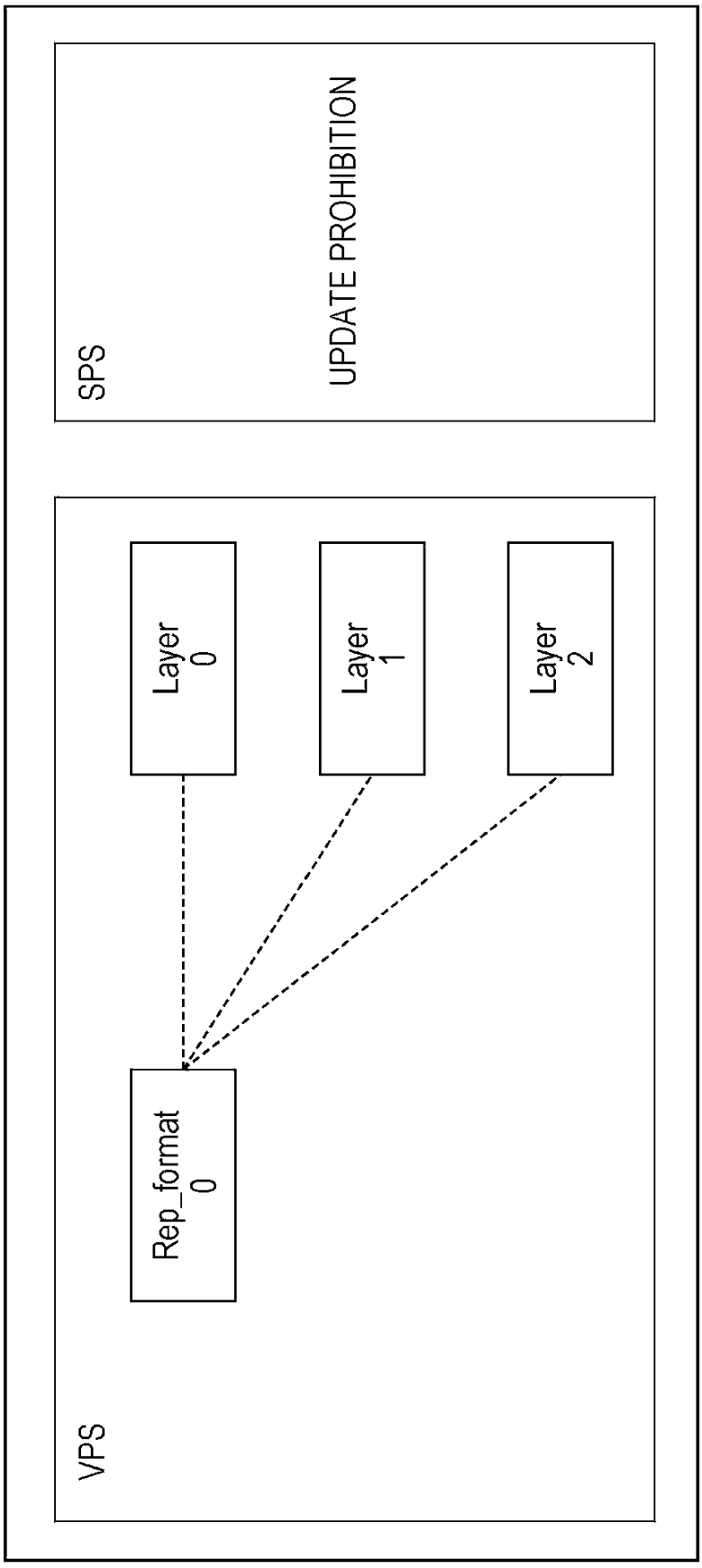
[Fig. 10]

FIG. 10


rep_format_idx_present_flag
u(1)
if(rep_format_idx_present_flag)
vps_num_rep_formats_minus1
ue(v)
for(i = 0; i <= vps_num_rep_formats_minus1; i++)
rep_format( )
if(rep_format_idx_present_flag)
for(i = 1; i <= MaxLayersMinus1; i++ )
if(vps_num_rep_formats_minus1 > 0)
vps_rep_format_idx[i]
u(v)
vps_num_rep_formats_minus1
ue(v)
for(i = 0; i <= vps_num_rep_formats_minus1; i++)
rep_format( )
if(vps_num_rep_formats_minus1 > 0)
rep_format_idx_present_flag
u(1)
if(rep_format_idx_present_flag)
for(i = 1; i <= MaxLayersMinus1; i++)
vps_rep_format_idx[i]
u(v)

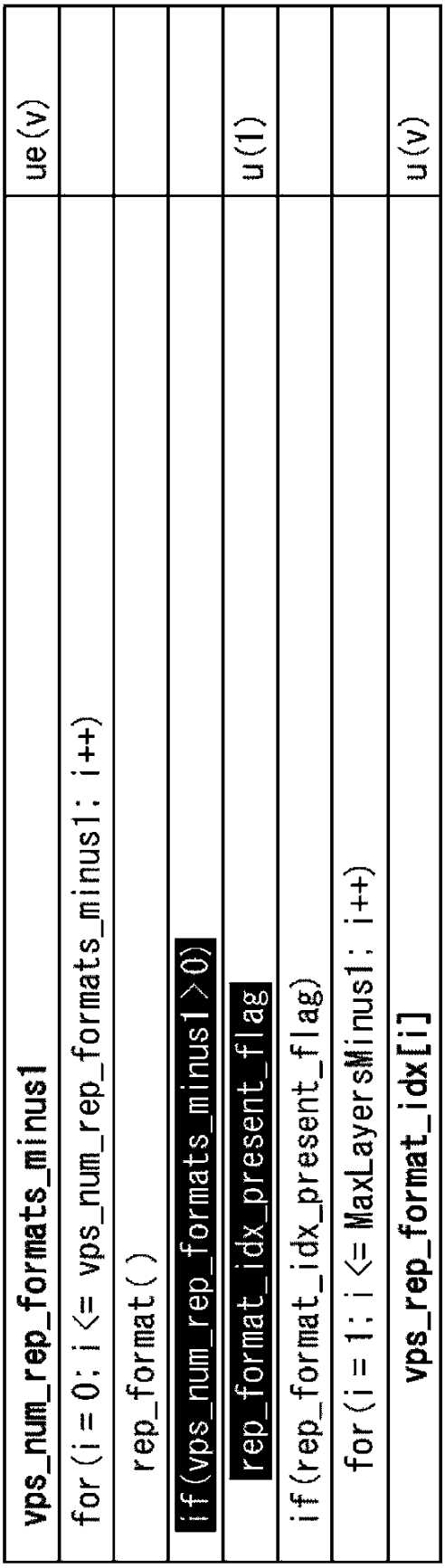
[Fig. 11]

FIG. 11


SPS
UPDATE PROHIBITION
VPS
Layer 0
Layer 1
Layer 2
Rep_format 0

[Fig. 12]

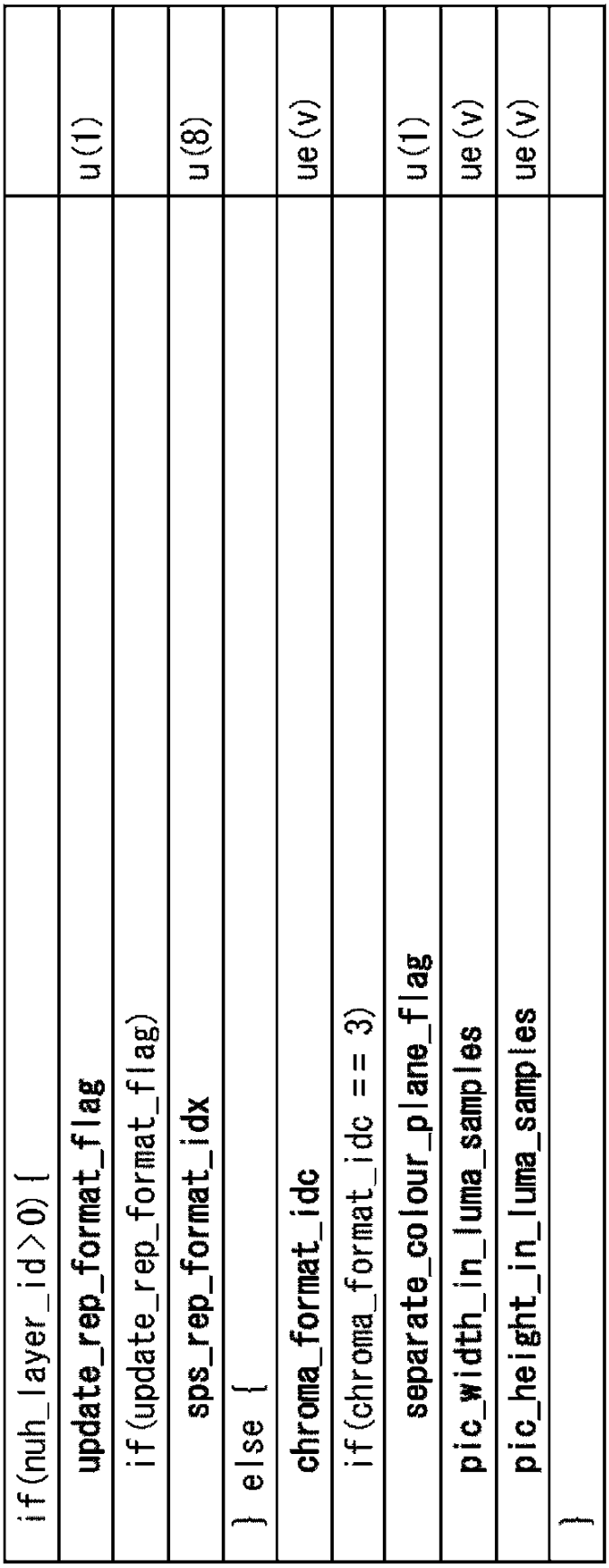
FIG. 12



| vps_num_rep_formats_minus1 | ue(v) |
|---|---|
| for (i = 0; i <= vps_num_rep_formats_minus1; i++) | |
| rep_format( ) | |
| if (vps_num_rep_formats_minus1 > 0) | |
| rep_format_idx_present_flag | u(1) |
| if (rep_format_idx_present_flag) | |
| for (i = 1; i <= MaxLayersMinus1; i++) | |
| vps_rep_format_idx[i] | u(v) |

[Fig. 13]

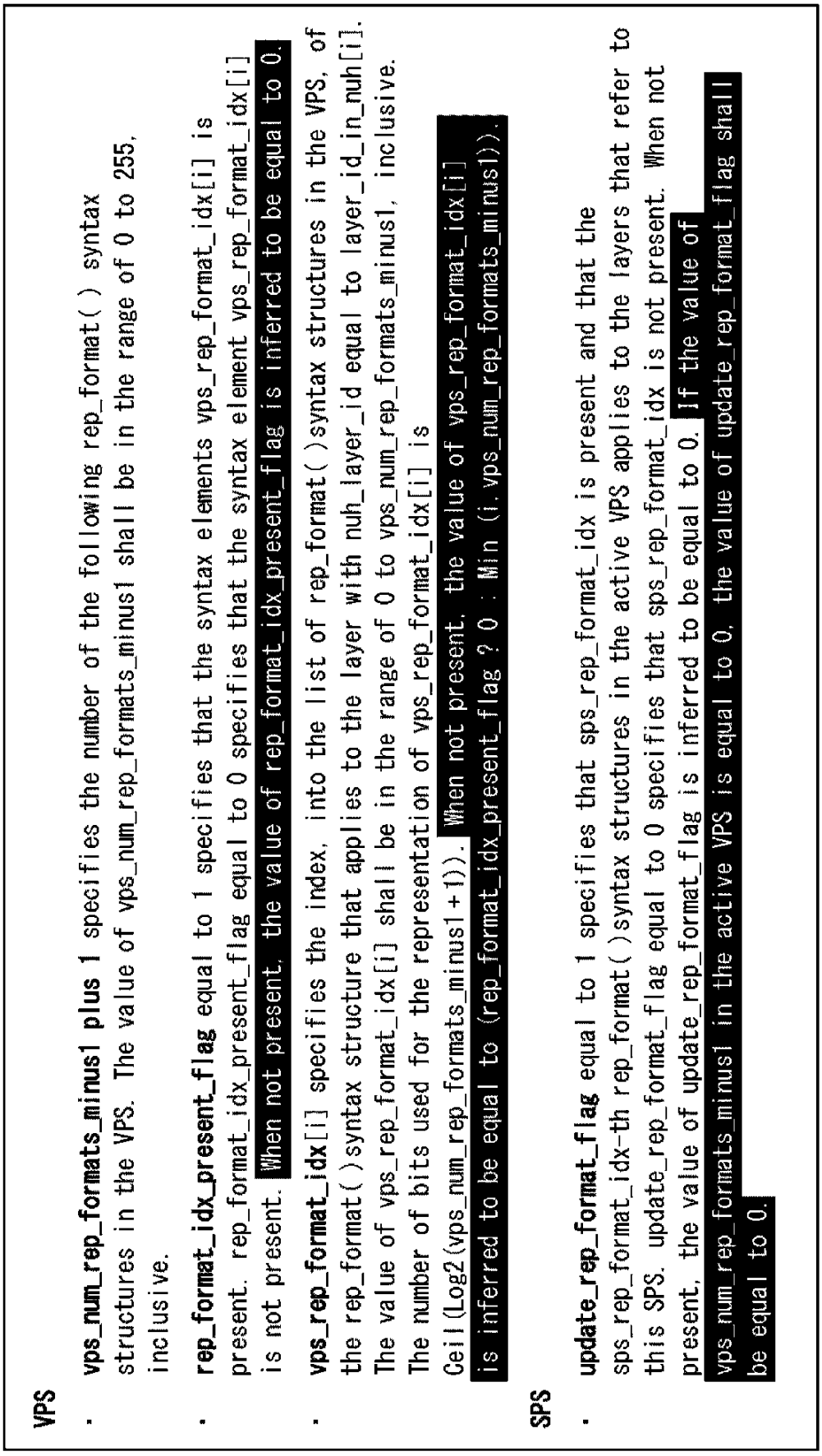
FIG. 13



| | |
|---|---|
| if(nuh_layer_id>0) { | |
| update_rep_format_flag | u(1) |
| if(update_rep_format_flag) | |
| sps_rep_format_idx | u(8) |
| } else { | |
| chroma_format_idc | ue(v) |
| if(chroma_format_idc == 3) | |
| separate_colour_plane_flag | u(1) |
| pic_width_in_luma_samples | ue(v) |
| pic_height_in_luma_samples | ue(v) |
| } | |

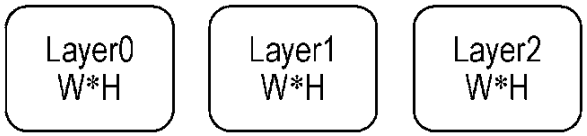
[Fig. 14]

FIG. 14


VPS
vps_num_rep_formats_minus1 plus 1 specifies the number of the following rep_format( ) syntax structures in the VPS. The value of vps_num_rep_formats_minus1 shall be in the range of 0 to 255, inclusive.
rep_format_idx_present_flag equal to 1 specifies that the syntax elements vps_rep_format_idx[i] is present. rep_format_idx_present_flag equal to 0 specifies that the syntax element vps_rep_format_idx[i] is not present. When not present, the value of rep_format_idx_present_flag is inferred to be equal to 0.
vps_rep_format_idx[i] specifies the index, into the list of rep_format( )syntax structures in the VPS, of the rep_format( )syntax structure that applies to the layer with nuh_layer_id equal to layer_id_in_nuh[i]. The value of vps_rep_format_idx[i] shall be in the range of 0 to vps_num_rep_formats_minus1, inclusive. The number of bits used for the representation of vps_rep_format_idx[i] is Ceil(Log2(vps_num_rep_formats_minus1+1)). When not present, the value of vps_rep_format_idx[i] is inferred to be equal to (rep_format_idx_present_flag ? 0 : Min (i, vps_num_rep_formats_minus1)).
SPS
update_rep_format_flag equal to 1 specifies that sps_rep_format_idx is present and that the sps_rep_format_idx-th rep_format( )syntax structures in the active VPS applies to the layers that refer to this SPS. update_rep_format_flag equal to 0 specifies that sps_rep_format_idx is not present. When not present, the value of update_rep_format_flag is inferred to be equal to 0. If the value of vps_num_rep_formats_minus1 in the active VPS is equal to 0, the value of update_rep_format_flag shall be equal to 0.

# FIG. 15

SNR/VIEW SCALABILITY (ALL LAYERS HAVE THE SAME RESOLUTION), 3 LAYER

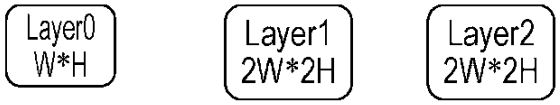



VALUE TO SEND ("–" INDICATES THAT DECODER PERFORMS ESTIMATION WITHOUT VALUE BEING TRANSFERRED)

| Syntax | METHOD 1 | METHOD 2 | METHOD 3 |
|---|---|---|---|
| rep_format_idx_present_flag | 1 | 1 | –<br>NO TRANSFER |
| vps_num_rep_formats_minus1 | 0 | (MaxLayersMinus1 = 3 – 1 = 2)<br>(ESTIMATION AT DECODER SIDE) | 0 |
| 0th rep_format() | W*H | W*H | W*H |
| 1st rep_format() | – | W*H | – |
| 2nd rep_format() | – | W*H | – |
| vps_rep_format_idx[0] | –<br>(DECODER ESTIMATES THAT Layer0 USES 0th rep_format) | –<br>(DECODER ESTIMATES THAT Layer0 USES 0th rep_format) | –<br>(DECODER ESTIMATES THAT Layer0 USES 0th rep_format) |
| vps_rep_format_idx[1] | –<br>(DECODER ESTIMATES THAT Layer1 USES 0th rep_format) | –<br>(DECODER ESTIMATES THAT Layer1 USES 1st rep_format) | –<br>(DECODER ESTIMATES THAT Layer1 USES 0th rep_format) |
| vps_rep_format_idx[2] | –<br>(DECODER ESTIMATES THAT Layer2 USES 0th rep_format) | –<br>(DECODER ESTIMATES THAT Layer2 USES 3rd rep_format) | –<br>(DECODER ESTIMATES THAT Layer2 USES 0th rep_format) |

# FIG. 16

SPATIAL AND SNR SCALABILITY, 3 LAYER

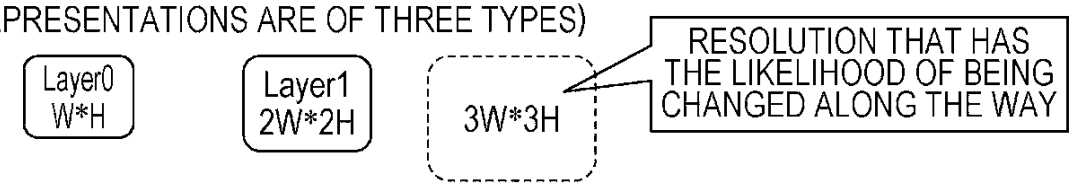



VALUE TO SEND ("–" INDICATES THAT DECODER PERFORMS ESTIMATION WITHOUT VALUE BEING TRANSFERRED)

| Syntax | METHOD 1 | METHOD 2 | METHOD 3 |
|---|---|---|---|
| rep_format_idx_present_flag | 1 | 0 | 0 |
| vps_num_rep_formats_minus1 | 1 | (MaxLayersMinus1 = 3 − 1 = 2)<br>(ESTIMATION AT DECODER SIDE) | 1 |
| 0th rep_format() | W*H | W*H | W*H |
| 1st rep_format() | 2W*2H | 2W*2H | 2W*2H |
| 2nd rep_format() | | 2W*2H | – |
| vps_rep_format_idx[0] | –<br>(DECODER ESTIMATES THAT Layer0 USES 0th rep_format) | –<br>(DECODER ESTIMATES THAT Layer0 USES 0th rep_format) | –<br>(DECODER ESTIMATES THAT Layer0 USES 0th rep_format) |
| vps_rep_format_idx[1] | 1 | –<br>(DECODER ESTIMATES THAT Layer1 USES 1st rep_format) | –<br>(DECODER ESTIMATES THAT Layer1 USES 1st rep_format) |
| vps_rep_format_idx[2] | 1 | –<br>(DECODER ESTIMATES THAT Layer2 USES 3rd rep_format) | –<br>(DECODER ESTIMATES THAT Layer2 USES 1st rep_format) |

# FIG. 17

SPATIAL SCALABILITY, 2 LAYER (REPRESENTATIONS ARE OF THREE TYPES)

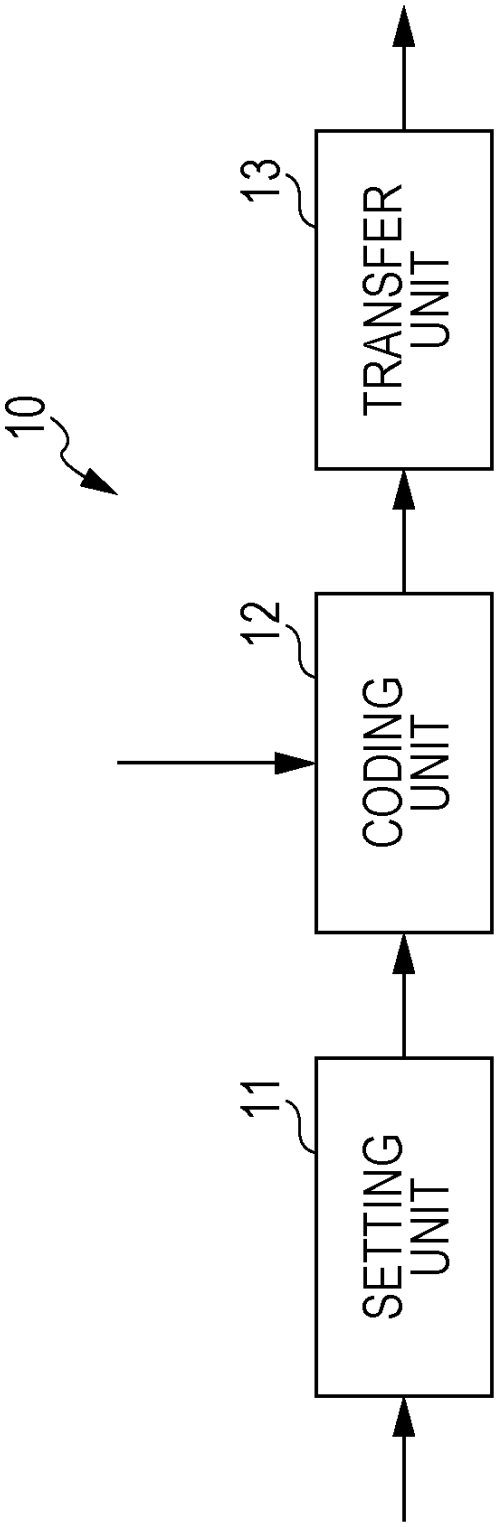



VALUE TO SEND ("–" INDICATES THAT DECODER PERFORMS ESTIMATION WITHOUT VALUE BEING TRANSFERRED)

| Syntax | METHOD 1 | METHOD 2 | METHOD 3 |
|---|---|---|---|
| rep_format_idx_present_flag | 1 | 0 | 0 |
| vps_num_rep_formats_minus1 | 2 | NO EXPRESSION | 2 |
| 0th rep_format() | W*H | | W*H |
| 1st rep_format() | 2W*2H | | 2W*2H |
| 2nd rep_format() | 3W*3H | | 3W*3H |
| vps_rep_format_idx[0] | – (DECODER ESTIMATES THAT Layer0 USES 0th rep_format) | | – (DECODER ESTIMATES THAT Layer0 USES 0th rep_format) |
| vps_rep_format_idx[1] | 1 | | – (DECODER ESTIMATES THAT Layer1 USES 1st rep_format) |
| vps_rep_format_idx[2] | – | | – |

[Fig. 18]

FIG. 18



[Fig. 19]

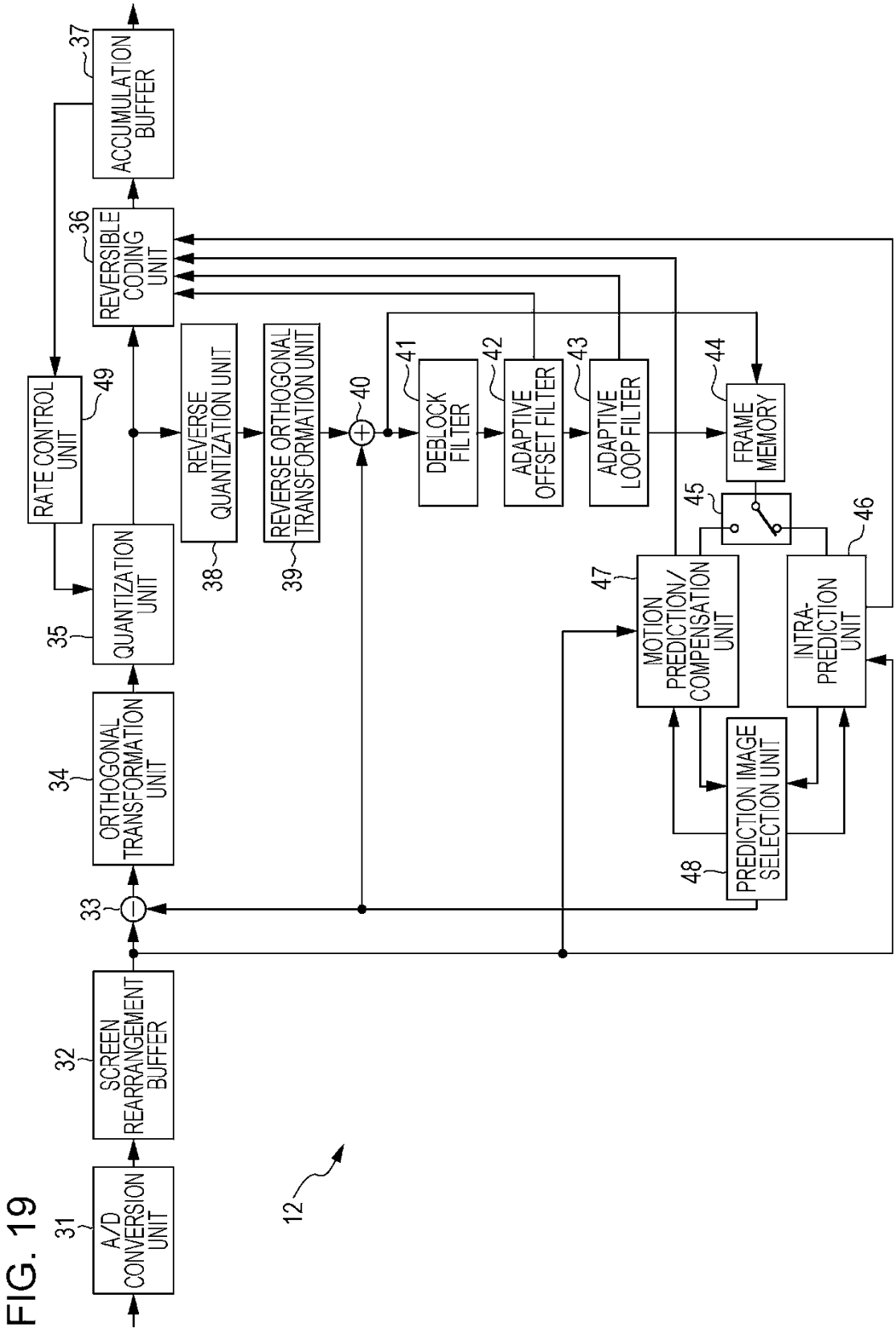
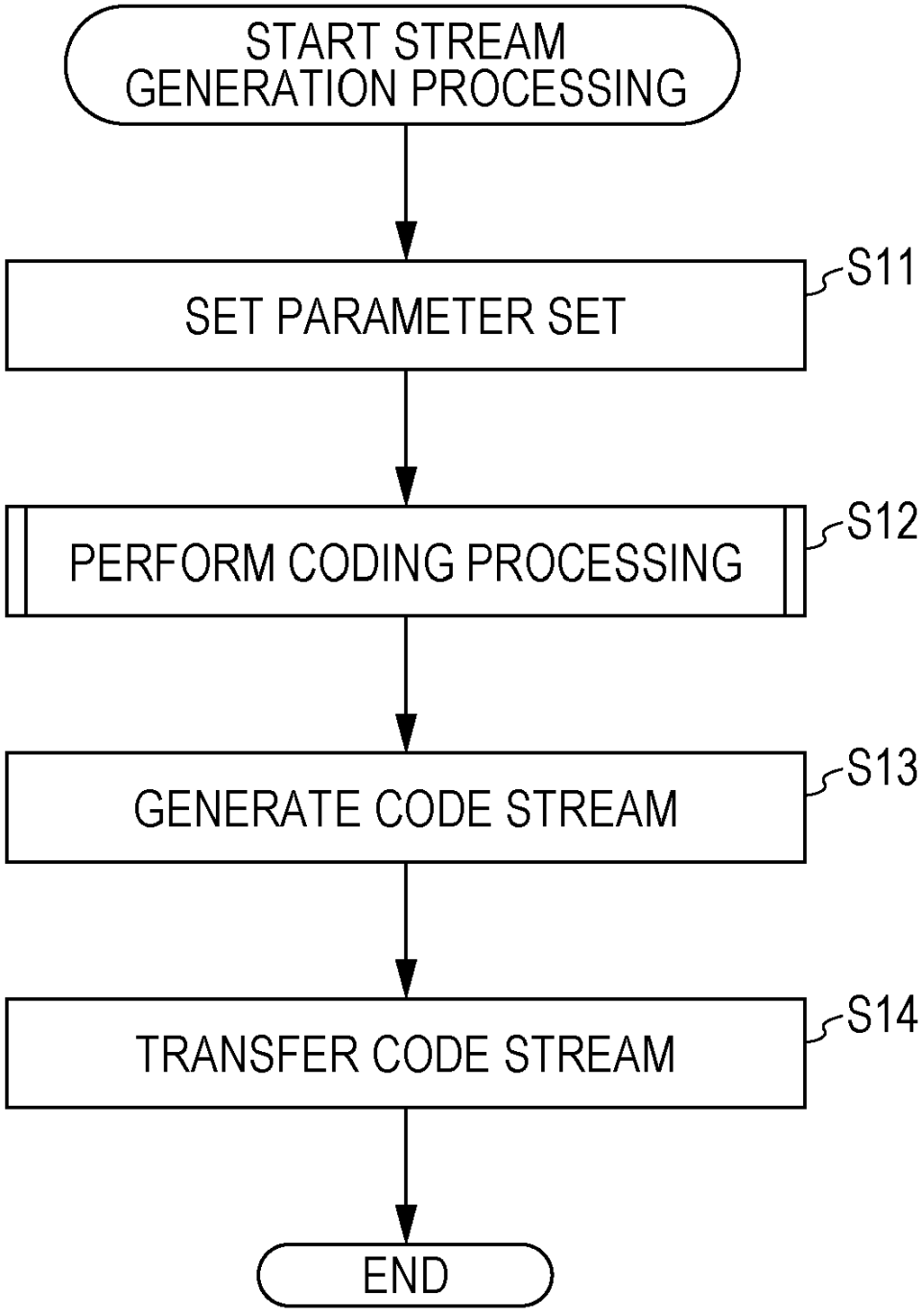

A/D CONVERSION UNIT
31
SCREEN REARRANGEMENT BUFFER
32
33
ORTHOGONAL TRANSFORMATION UNIT
34
QUANTIZATION UNIT
35
REVERSIBLE CODING UNIT
36
ACCUMULATION BUFFER
37
RATE CONTROL UNIT
49
REVERSE QUANTIZATION UNIT
38
REVERSE ORTHOGONAL TRANSFORMATION UNIT
39
40
DEBLOCK FILTER
41
ADAPTIVE OFFSET FILTER
42
ADAPTIVE LOOP FILTER
43
FRAME MEMORY
44
45
46
MOTION PREDICTION/ COMPENSATION UNIT
47
INTRA-PREDICTION UNIT
PREDICTION IMAGE SELECTION UNIT
48
12

FIG. 19

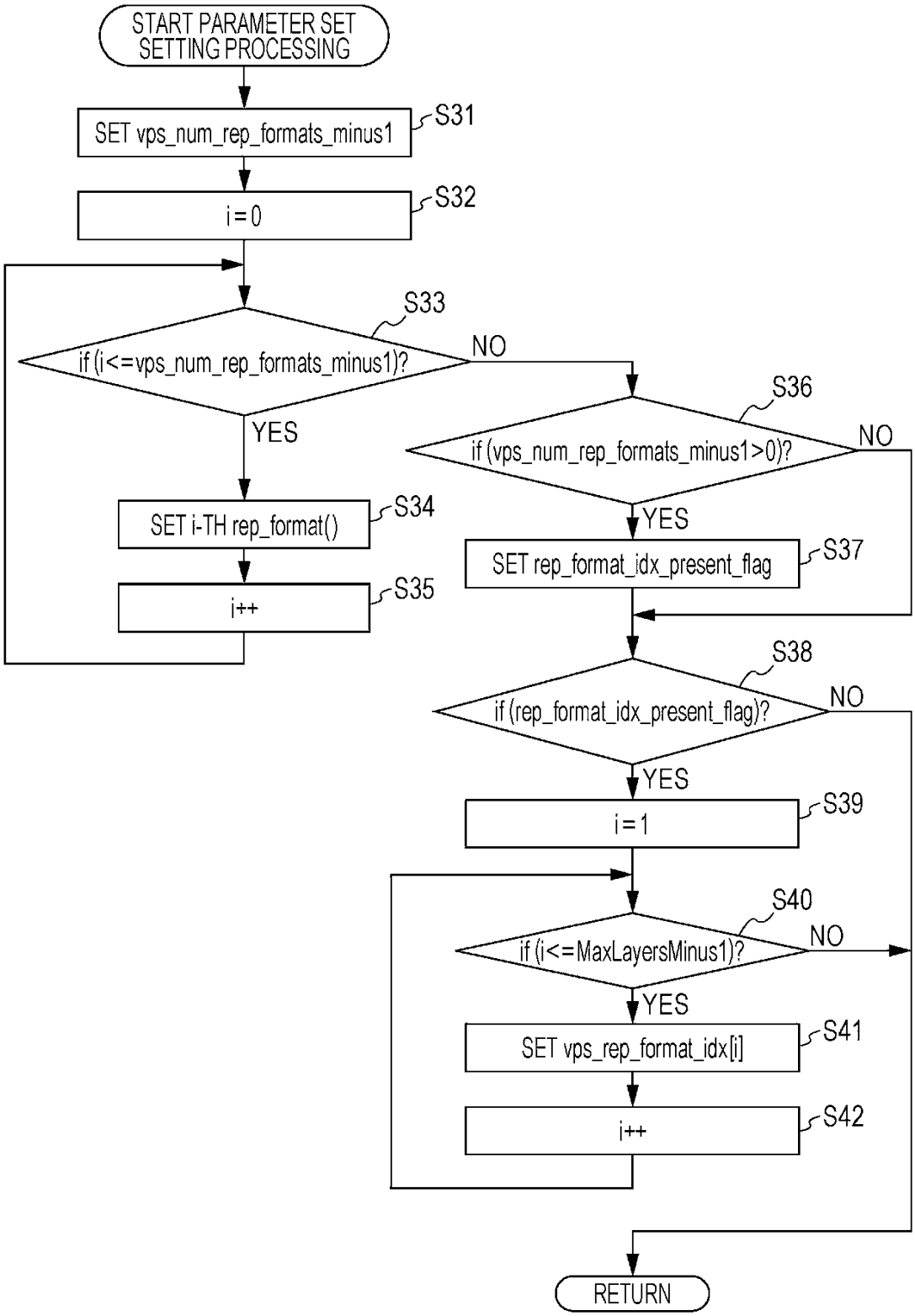
[Fig. 20]

FIG. 20


START STREAM GENERATION PROCESSING
SET PARAMETER SET
S11
PERFORM CODING PROCESSING
S12
GENERATE CODE STREAM
S13
TRANSFER CODE STREAM
S14
END

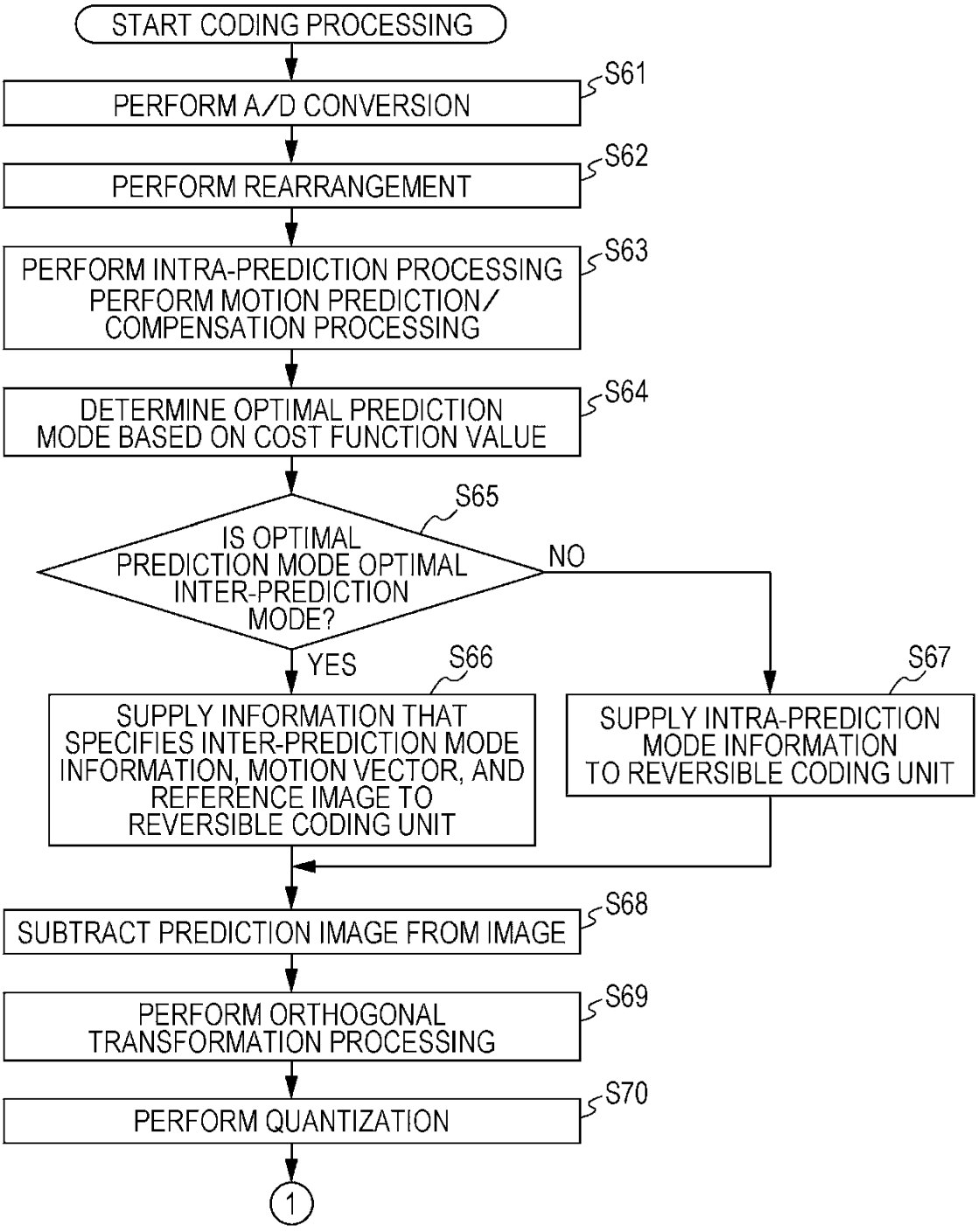
[Fig. 21]

FIG. 21


START PARAMETER SET SETTING PROCESSING
SET vps_num_rep_formats_minus1
S31
i = 0
S32
S33
if (i<=vps_num_rep_formats_minus1)?
NO
YES
SET i-TH rep_format()
S34
i++
S35
S36
if (vps_num_rep_formats_minus1>0)?
NO
YES
SET rep_format_idx_present_flag
S37
S38
if (rep_format_idx_present_flag)?
NO
YES
i = 1
S39
S40
if (i<=MaxLayersMinus1)?
NO
YES
SET vps_rep_format_idx[i]
S41
i++
S42
RETURN

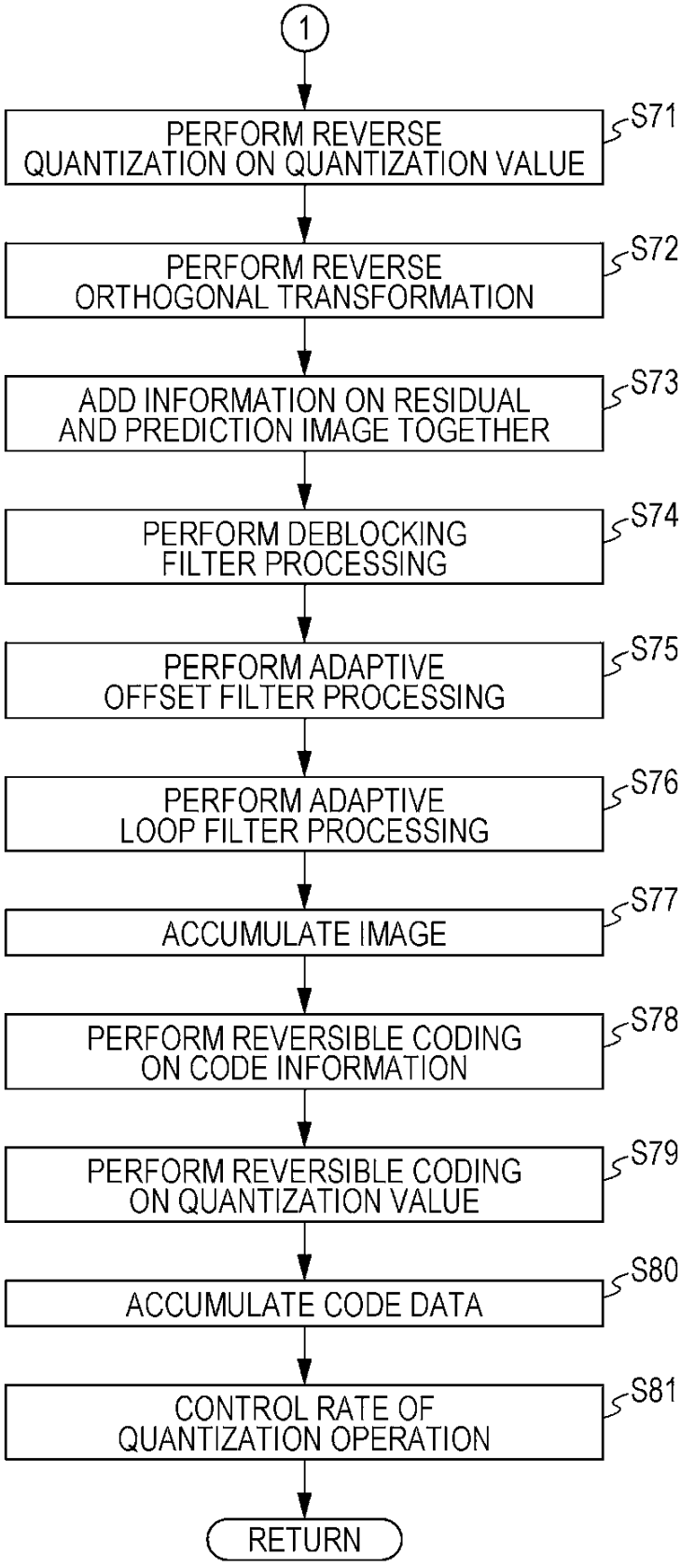
[Fig. 22]

FIG. 22


START CODING PROCESSING
S61
PERFORM A/D CONVERSION
S62
PERFORM REARRANGEMENT
S63
PERFORM INTRA-PREDICTION PROCESSING
PERFORM MOTION PREDICTION/
COMPENSATION PROCESSING
S64
DETERMINE OPTIMAL PREDICTION
MODE BASED ON COST FUNCTION VALUE
S65
IS OPTIMAL
PREDICTION MODE OPTIMAL
INTER-PREDICTION
MODE?
NO
YES
S66
SUPPLY INFORMATION THAT
SPECIFIES INTER-PREDICTION MODE
INFORMATION, MOTION VECTOR, AND
REFERENCE IMAGE TO
REVERSIBLE CODING UNIT
S67
SUPPLY INTRA-PREDICTION
MODE INFORMATION
TO REVERSIBLE CODING UNIT
S68
SUBTRACT PREDICTION IMAGE FROM IMAGE
S69
PERFORM ORTHOGONAL
TRANSFORMATION PROCESSING
S70
PERFORM QUANTIZATION
1

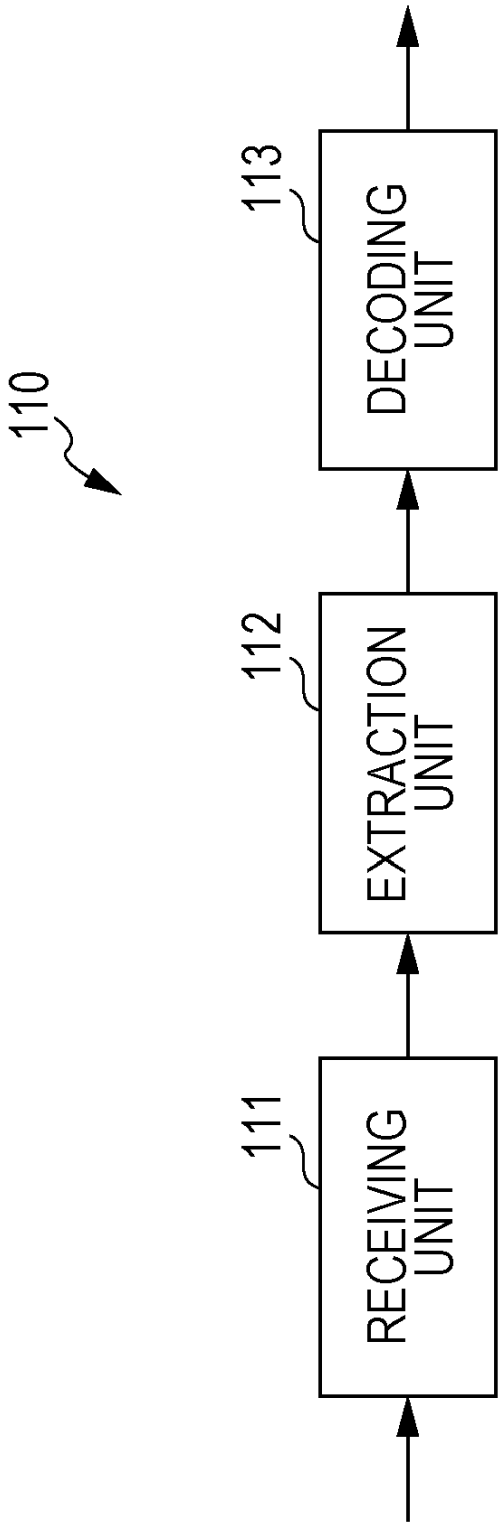
[Fig. 23]

FIG. 23


1
PERFORM REVERSE QUANTIZATION ON QUANTIZATION VALUE
S71
PERFORM REVERSE ORTHOGONAL TRANSFORMATION
S72
ADD INFORMATION ON RESIDUAL AND PREDICTION IMAGE TOGETHER
S73
PERFORM DEBLOCKING FILTER PROCESSING
S74
PERFORM ADAPTIVE OFFSET FILTER PROCESSING
S75
PERFORM ADAPTIVE LOOP FILTER PROCESSING
S76
ACCUMULATE IMAGE
S77
PERFORM REVERSIBLE CODING ON CODE INFORMATION
S78
PERFORM REVERSIBLE CODING ON QUANTIZATION VALUE
S79
ACCUMULATE CODE DATA
S80
CONTROL RATE OF QUANTIZATION OPERATION
S81
RETURN

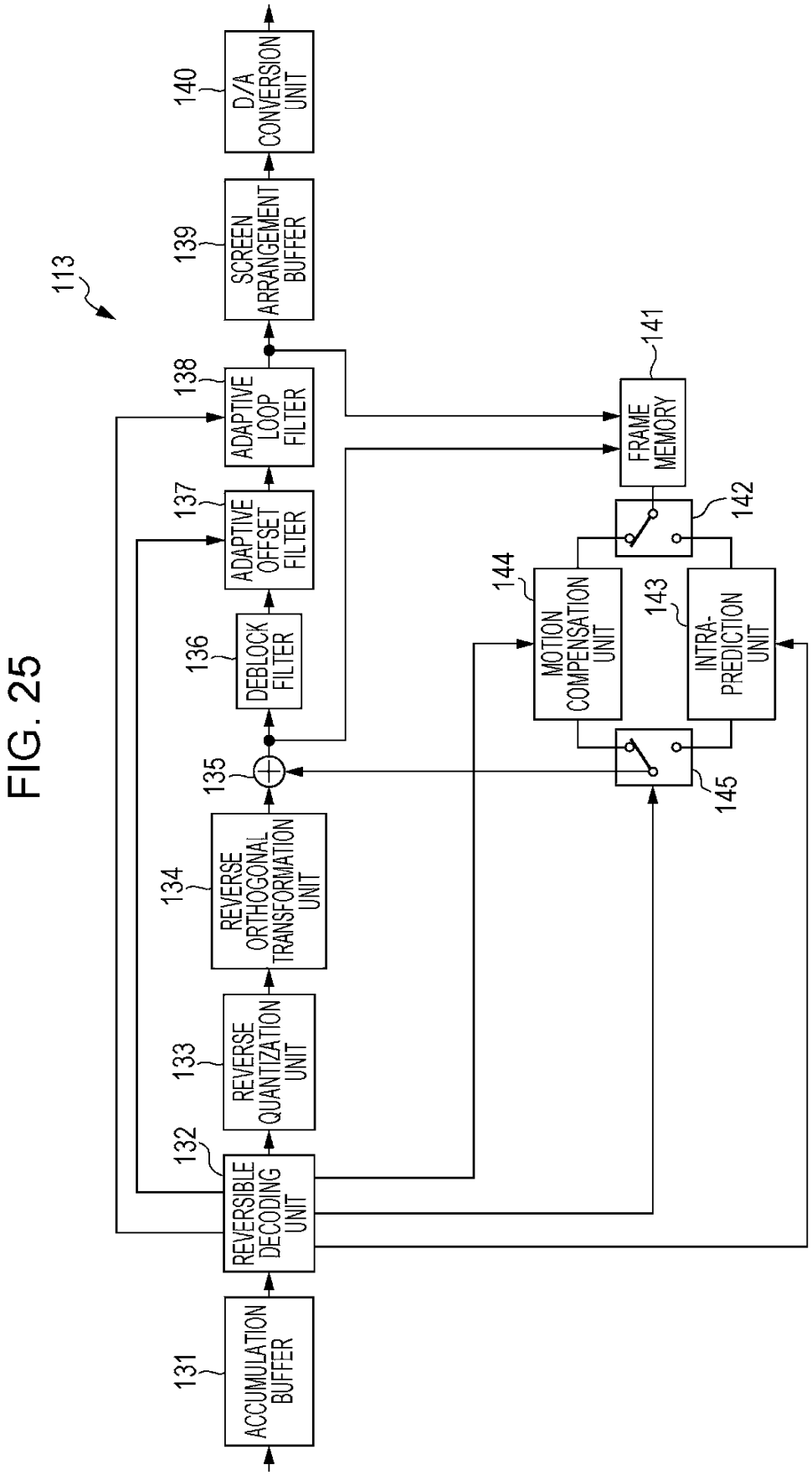
[Fig. 24]

FIG. 24


110
111
RECEIVING UNIT
112
EXTRACTION UNIT
113
DECODING UNIT

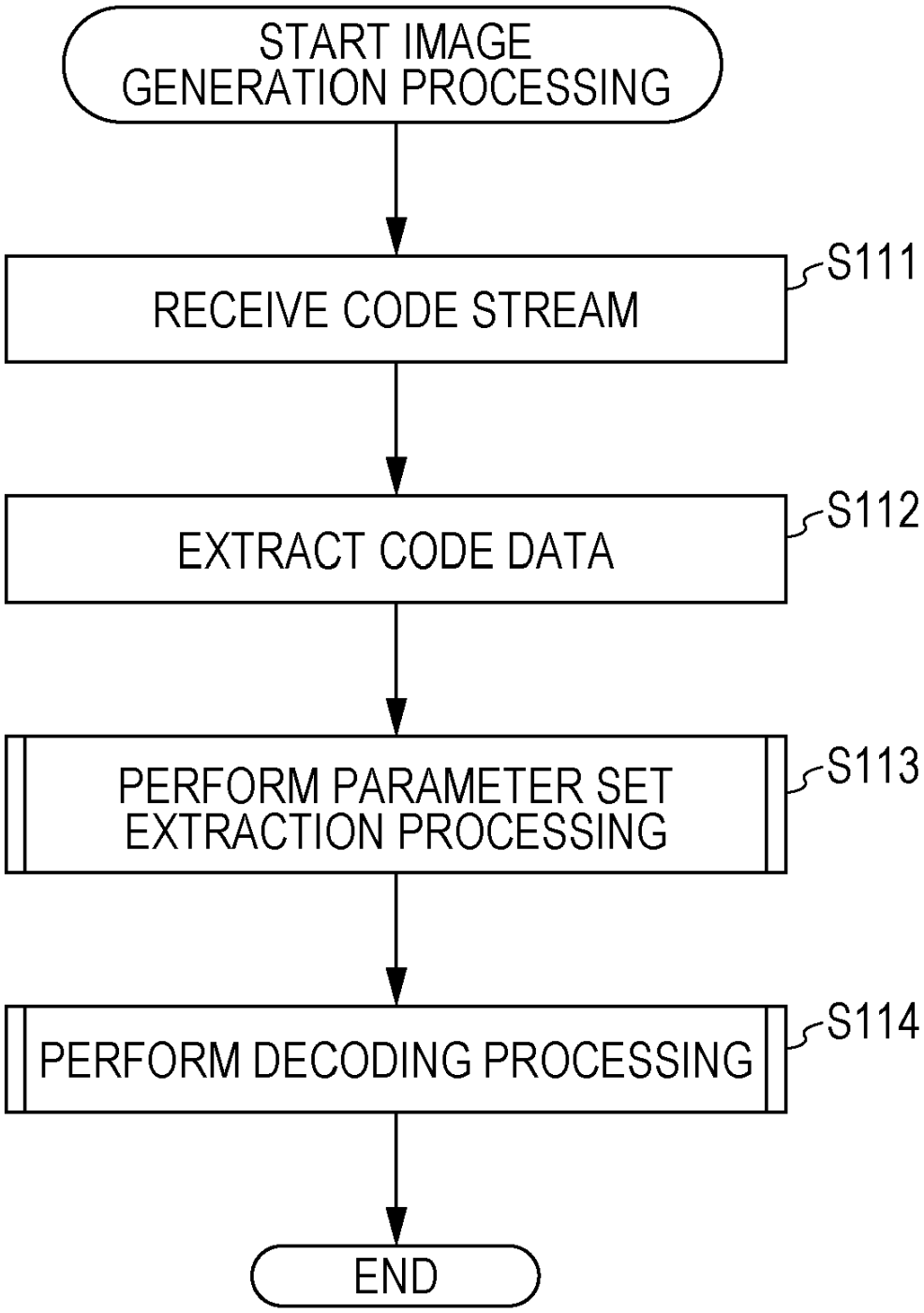
[Fig. 25]

FIG. 25


113
131
ACCUMULATION BUFFER
132
REVERSIBLE DECODING UNIT
133
REVERSE QUANTIZATION UNIT
134
REVERSE ORTHOGONAL TRANSFORMATION UNIT
135
136
DEBLOCK FILTER
137
ADAPTIVE OFFSET FILTER
138
ADAPTIVE LOOP FILTER
139
SCREEN ARRANGEMENT BUFFER
140
D/A CONVERSION UNIT
141
FRAME MEMORY
142
143
INTRA-PREDICTION UNIT
144
MOTION COMPENSATION UNIT
145

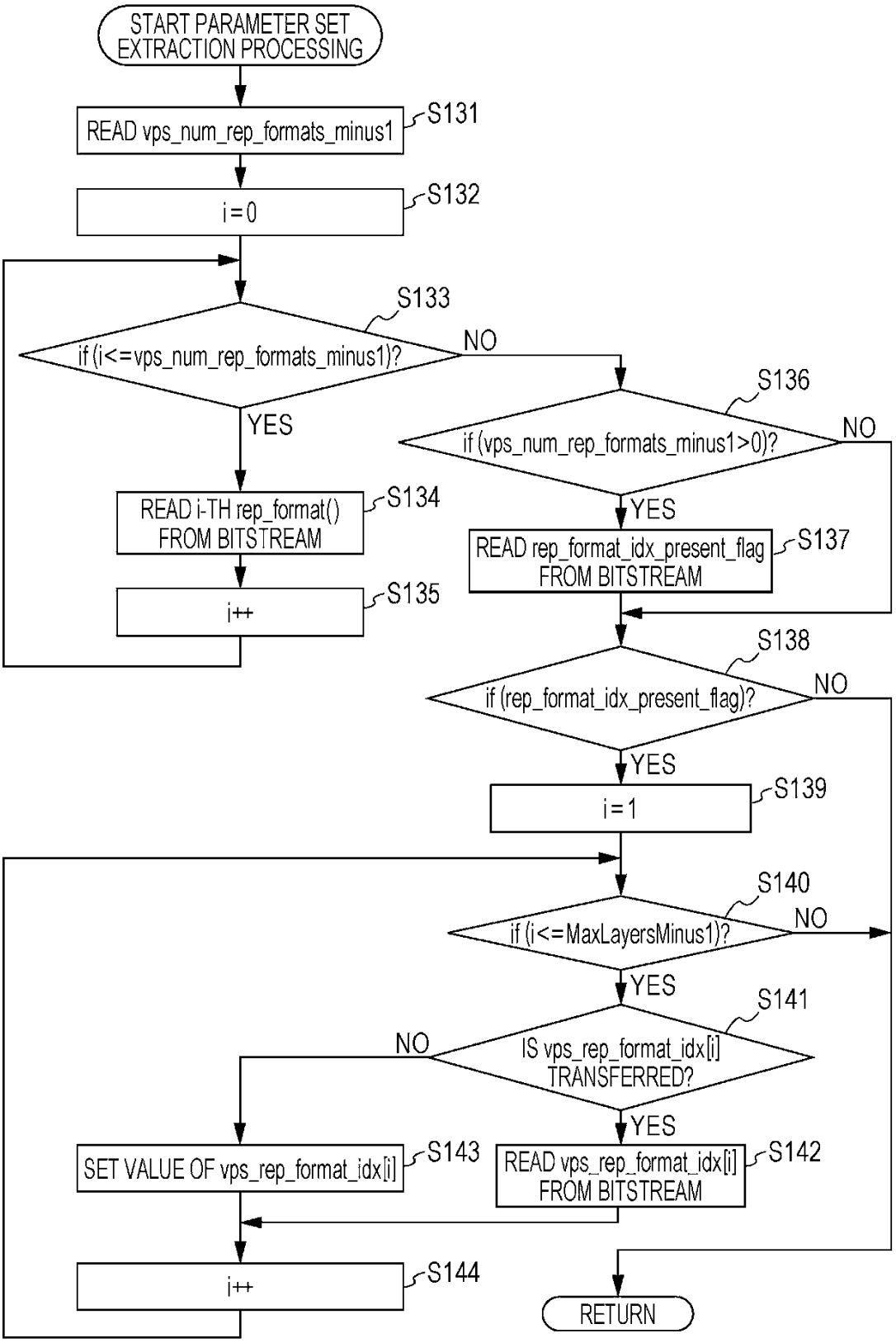
[Fig. 26]

FIG. 26


START IMAGE GENERATION PROCESSING
RECEIVE CODE STREAM
S111
EXTRACT CODE DATA
S112
PERFORM PARAMETER SET EXTRACTION PROCESSING
S113
PERFORM DECODING PROCESSING
S114
END

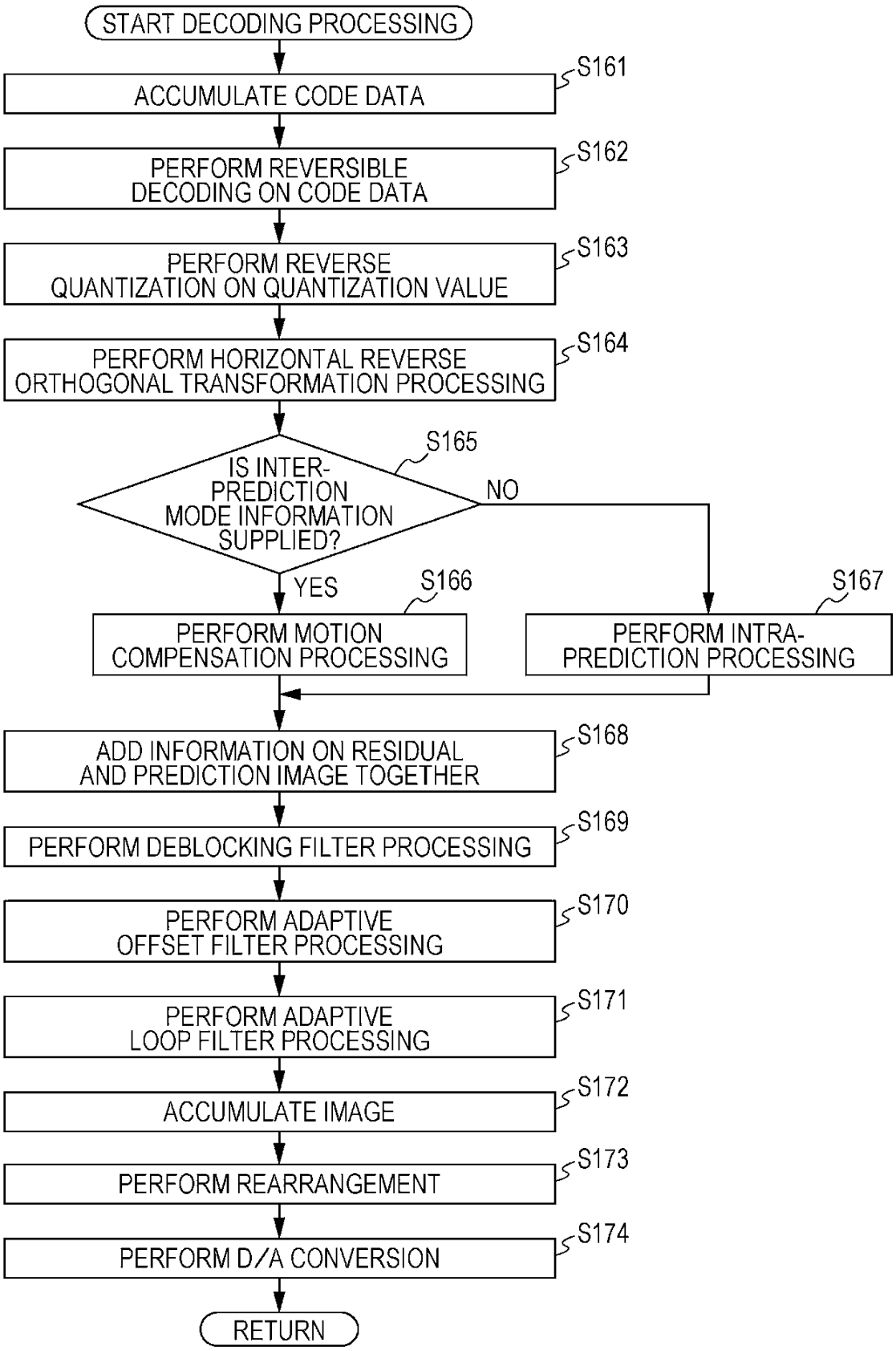
[Fig. 27]

FIG. 27


START PARAMETER SET EXTRACTION PROCESSING
READ vps_num_rep_formats_minus1
S131
i=0
S132
S133
if (i<=vps_num_rep_formats_minus1)?
NO
YES
READ i-TH rep_format() FROM BITSTREAM
S134
i++
S135
S136
if (vps_num_rep_formats_minus1>0)?
NO
YES
READ rep_format_idx_present_flag FROM BITSTREAM
S137
S138
if (rep_format_idx_present_flag)?
NO
YES
i=1
S139
S140
if (i<=MaxLayersMinus1)?
NO
YES
S141
IS vps_rep_format_idx[i] TRANSFERRED?
NO
YES
SET VALUE OF vps_rep_format_idx[i]
S143
READ vps_rep_format_idx[i] FROM BITSTREAM
S142
i++
S144
RETURN

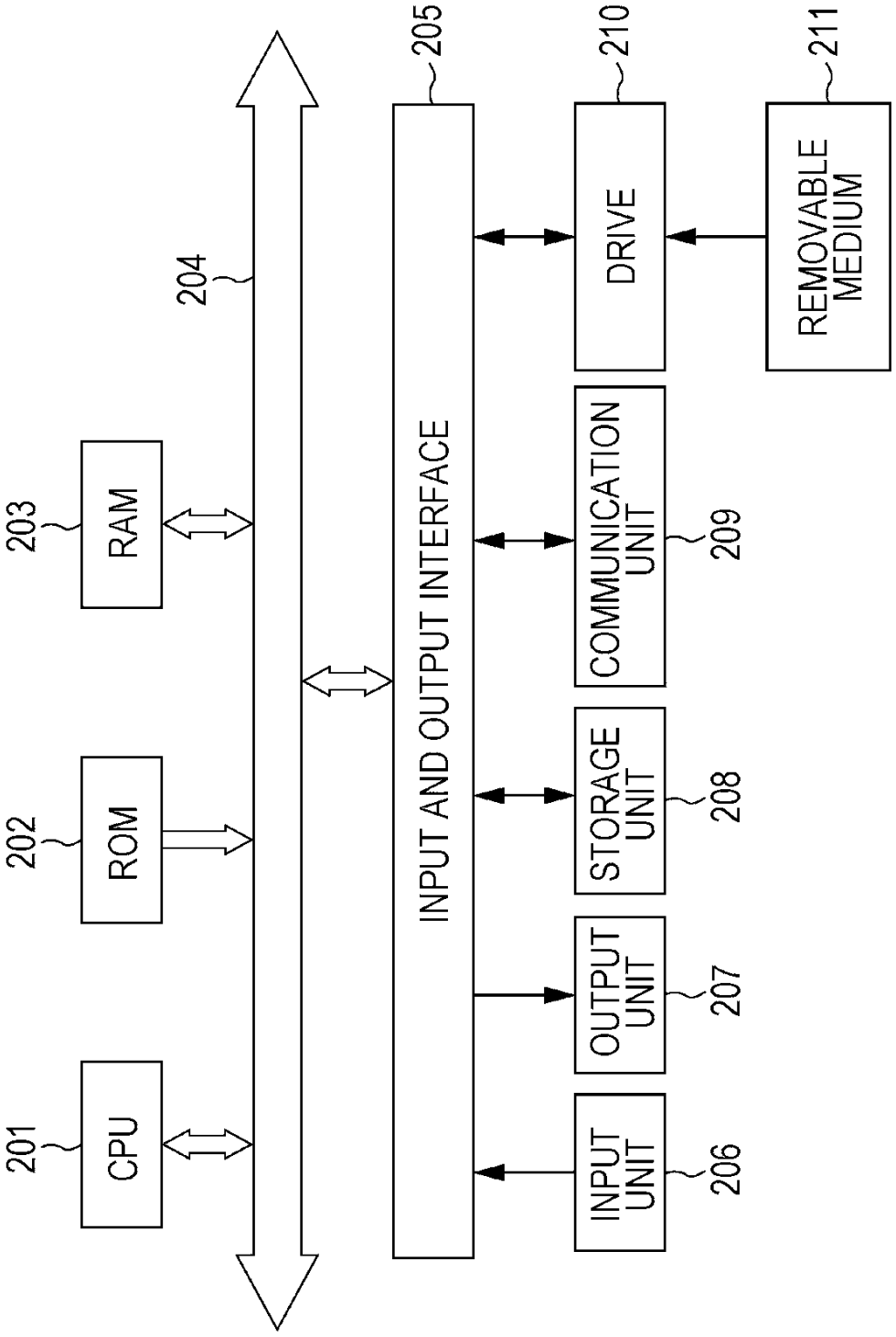
[Fig. 28]

FIG. 28


START DECODING PROCESSING
ACCUMULATE CODE DATA
S161
PERFORM REVERSIBLE DECODING ON CODE DATA
S162
PERFORM REVERSE QUANTIZATION ON QUANTIZATION VALUE
S163
PERFORM HORIZONTAL REVERSE ORTHOGONAL TRANSFORMATION PROCESSING
S164
IS INTER-PREDICTION MODE INFORMATION SUPPLIED?
S165
NO
YES
PERFORM MOTION COMPENSATION PROCESSING
S166
PERFORM INTRA-PREDICTION PROCESSING
S167
ADD INFORMATION ON RESIDUAL AND PREDICTION IMAGE TOGETHER
S168
PERFORM DEBLOCKING FILTER PROCESSING
S169
PERFORM ADAPTIVE OFFSET FILTER PROCESSING
S170
PERFORM ADAPTIVE LOOP FILTER PROCESSING
S171
ACCUMULATE IMAGE
S172
PERFORM REARRANGEMENT
S173
PERFORM D/A CONVERSION
S174
RETURN

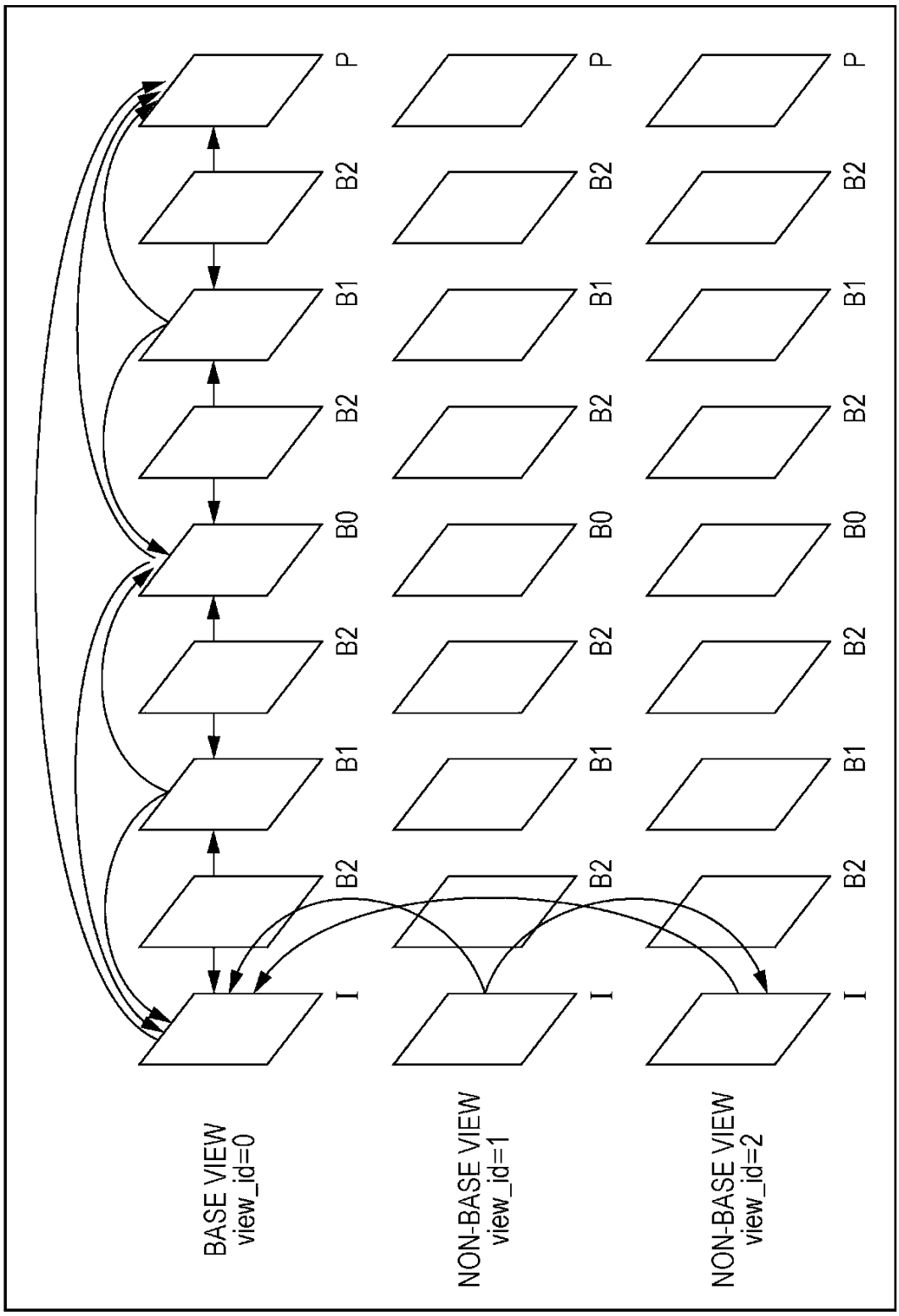
[Fig. 29]

FIG. 29


201
CPU
202
ROM
203
RAM
204
205
INPUT AND OUTPUT INTERFACE
206
INPUT UNIT
207
OUTPUT UNIT
208
STORAGE UNIT
209
COMMUNICATION UNIT
210
DRIVE
211
REMOVABLE MEDIUM

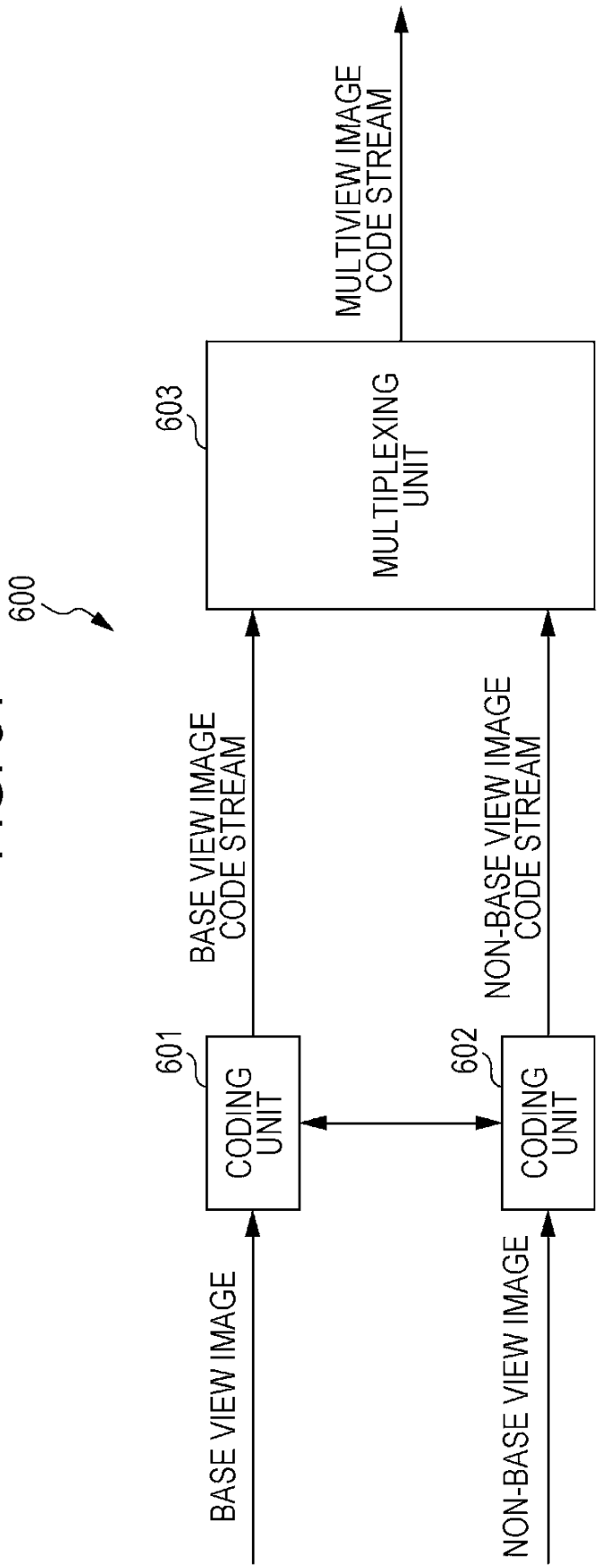
[Fig. 30]

FIG. 30


BASE VIEW
view_id=0
NON-BASE VIEW
view_id=1
NON-BASE VIEW
view_id=2
I
B2
B1
B2
B0
B2
B1
B2
P

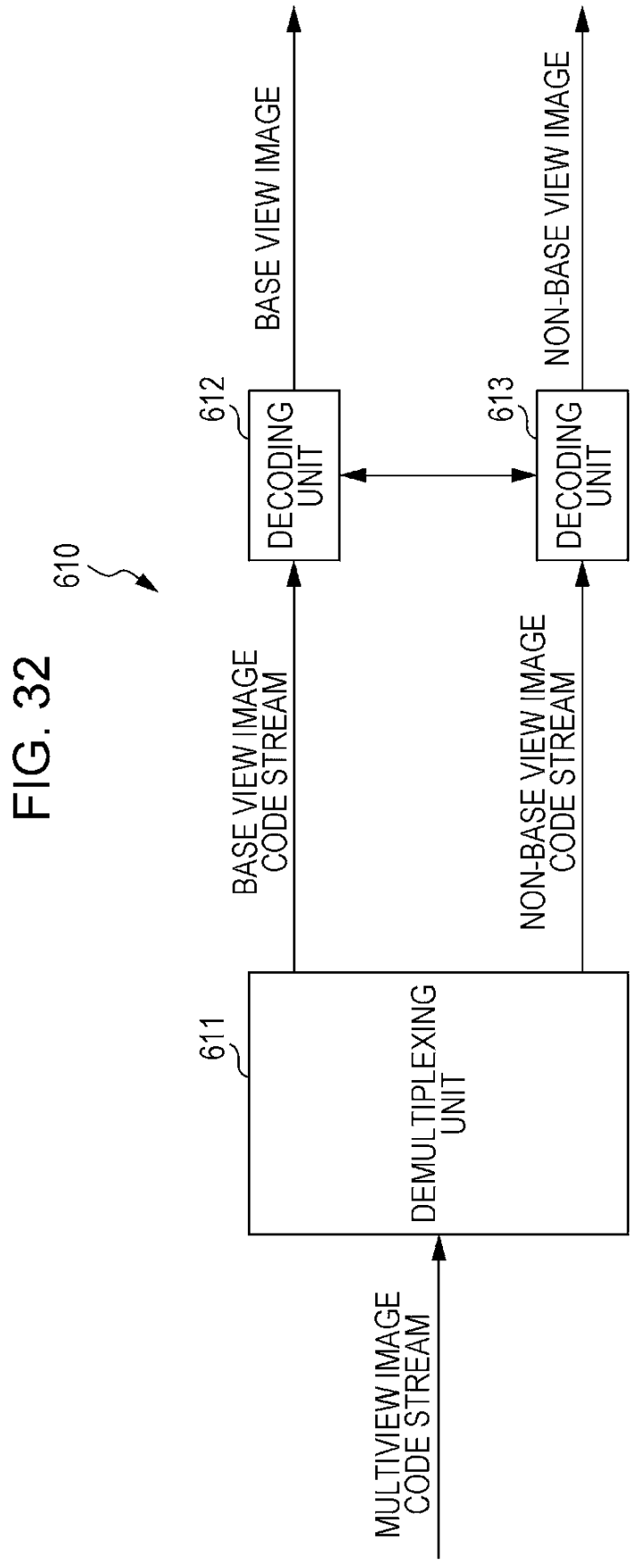
[Fig. 31]

FIG. 31


600
603
MULTIPLEXING UNIT
MULTIVIEW IMAGE CODE STREAM
BASE VIEW IMAGE CODE STREAM
NON-BASE VIEW IMAGE CODE STREAM
601
602
CODING UNIT
CODING UNIT
BASE VIEW IMAGE
NON-BASE VIEW IMAGE

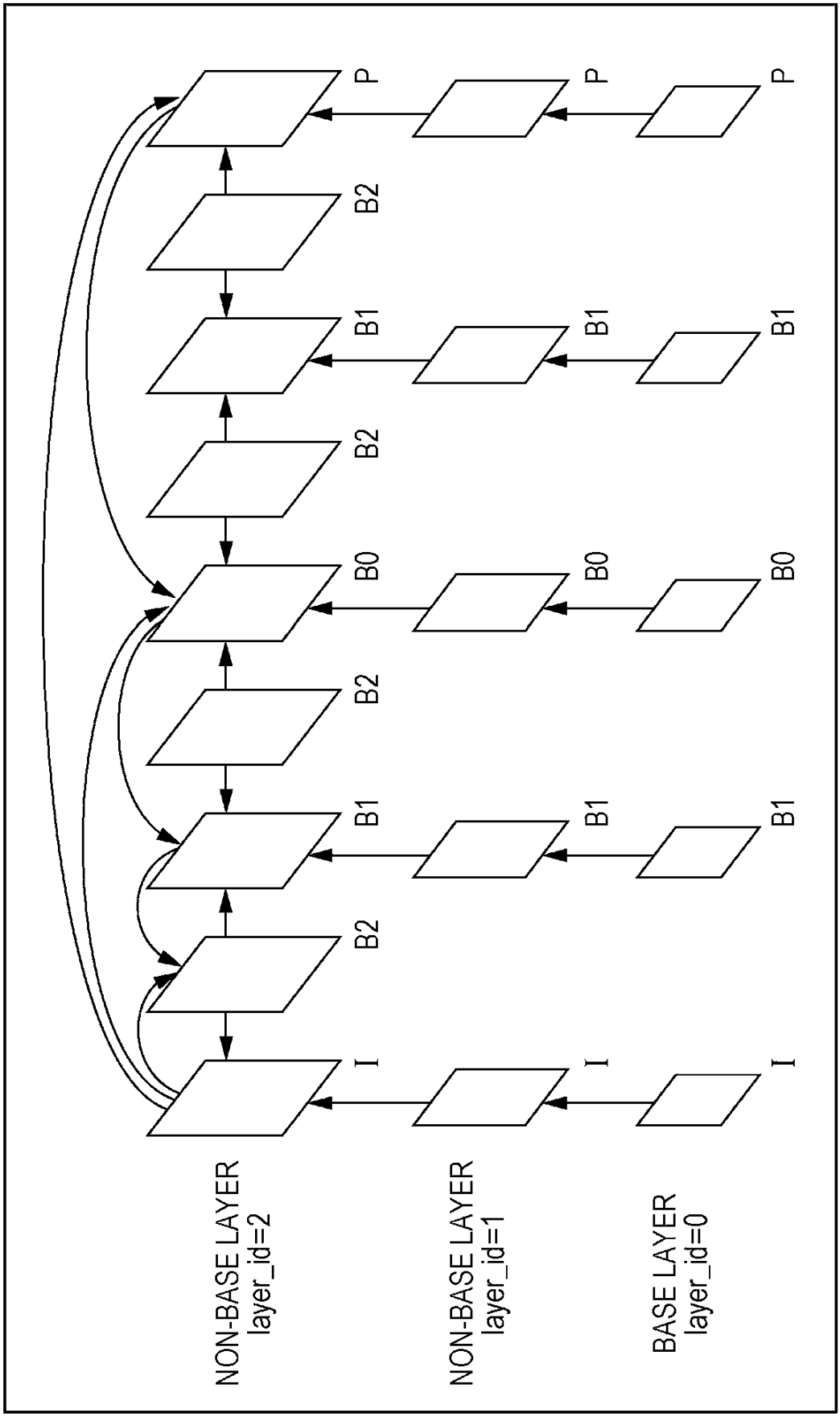
[Fig. 32]

FIG. 32


610
611
DEMULTIPLEXING UNIT
612
DECODING UNIT
613
DECODING UNIT
MULTIVIEW IMAGE CODE STREAM
BASE VIEW IMAGE CODE STREAM
NON-BASE VIEW IMAGE CODE STREAM
BASE VIEW IMAGE
NON-BASE VIEW IMAGE

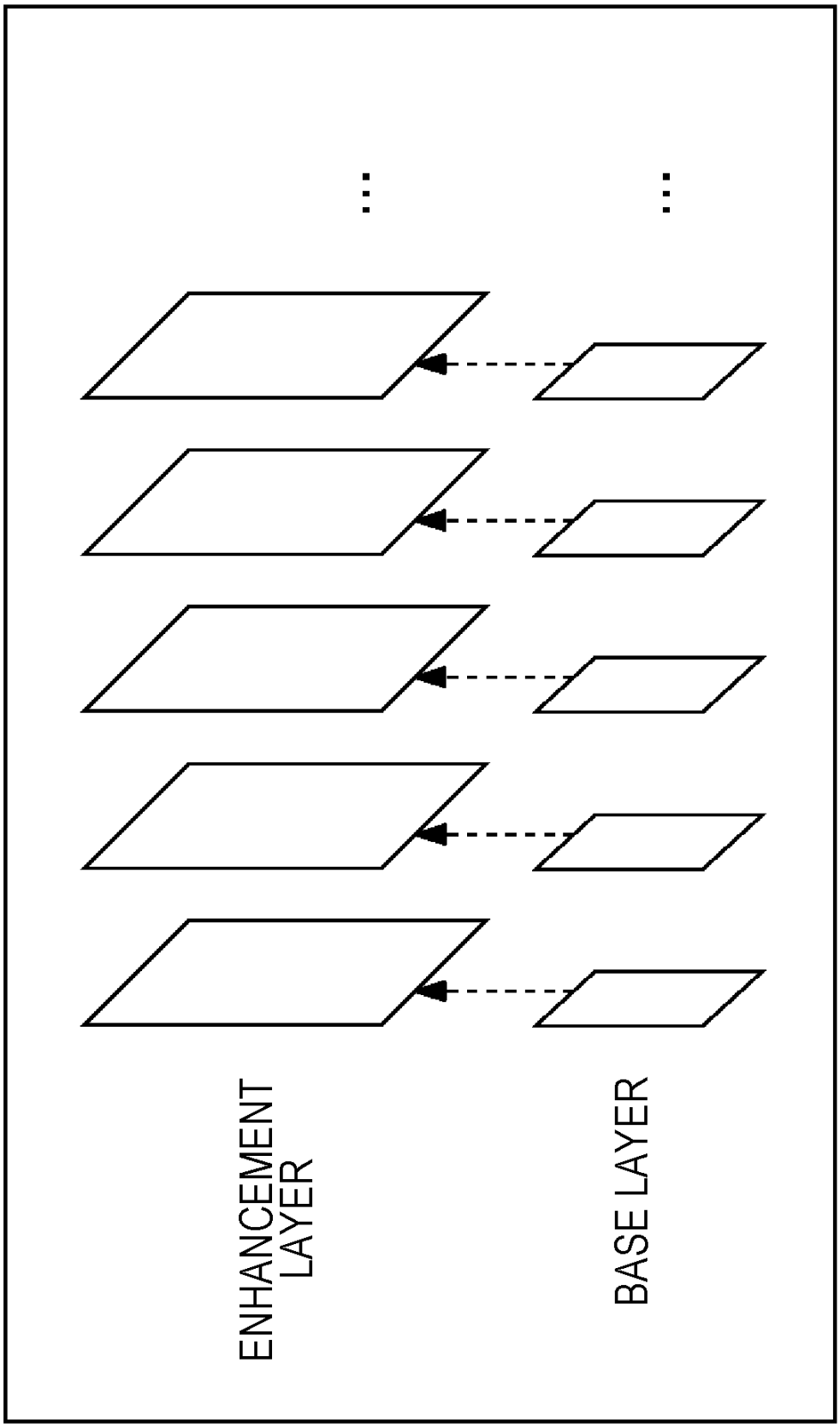
[Fig. 33]

FIG. 33


NON-BASE LAYER
layer_id=2
NON-BASE LAYER
layer_id=1
BASE LAYER
layer_id=0
I
B2
B1
B2
B0
B2
B1
B2
P
B2
B1
B2
B0
B2
B1
B2
P
B2
B1
B2
B0
B2
B1
B2
P
B2
B1
B2
B0
B2
B1
B2

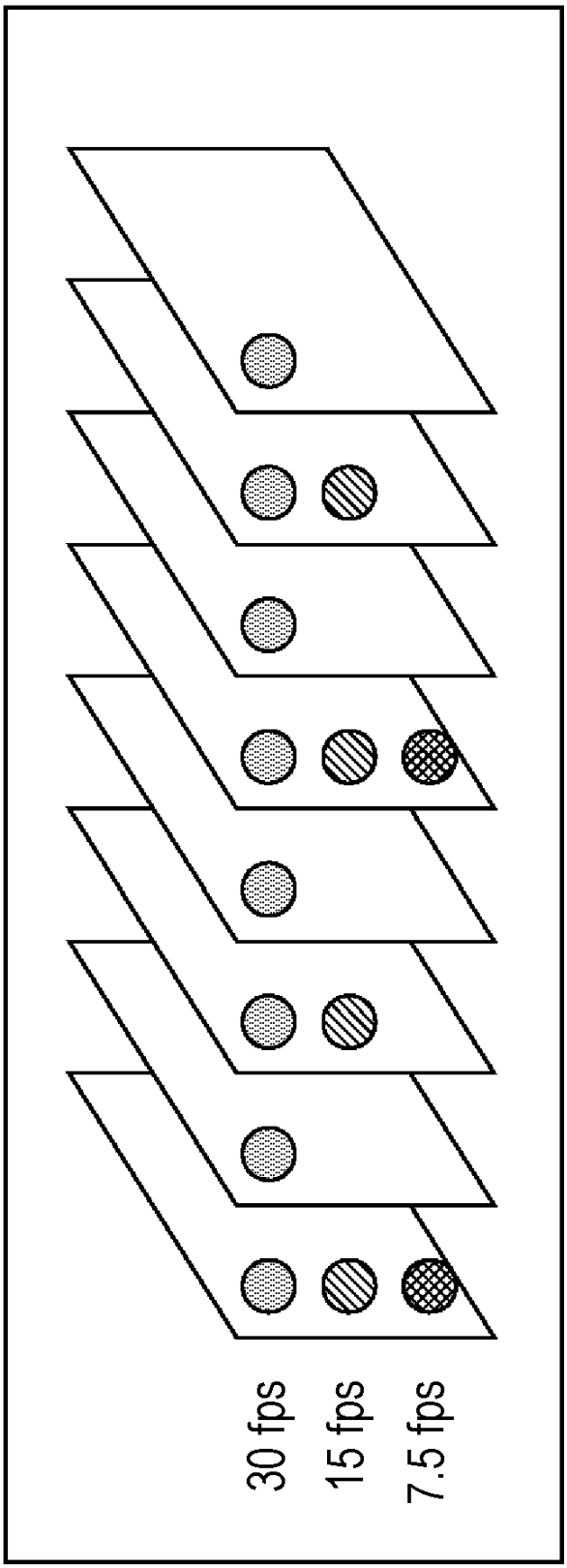
[Fig. 34]

FIG. 34



[Fig. 35]

FIG. 35



[Fig. 36]

FIG. 36

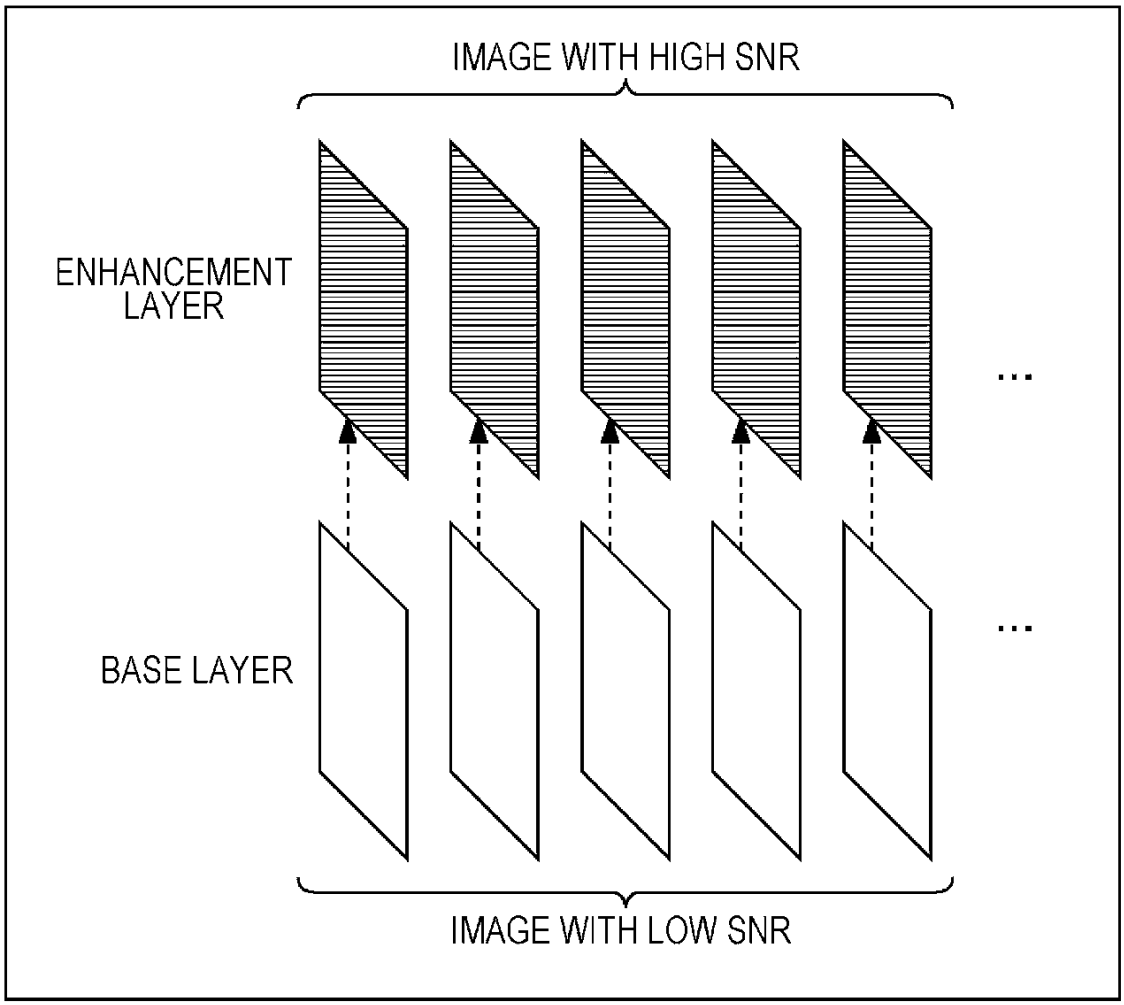
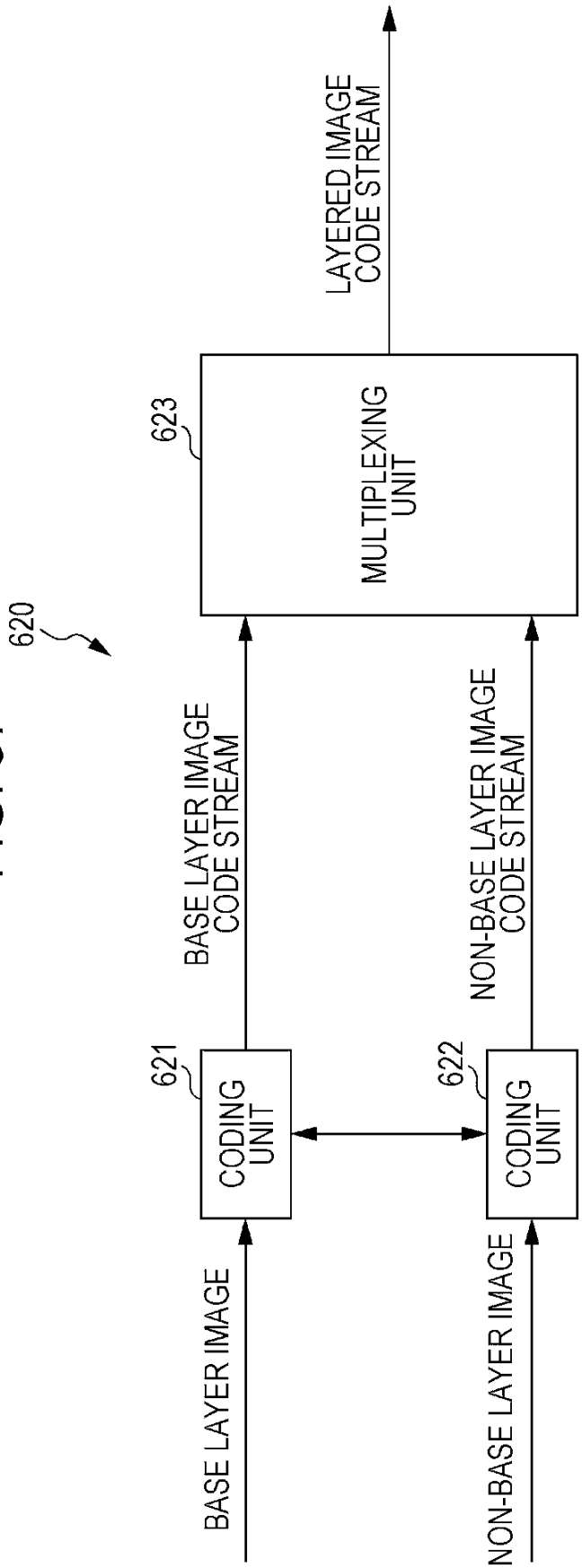


IMAGE WITH HIGH SNR
ENHANCEMENT LAYER
...
BASE LAYER
...
IMAGE WITH LOW SNR

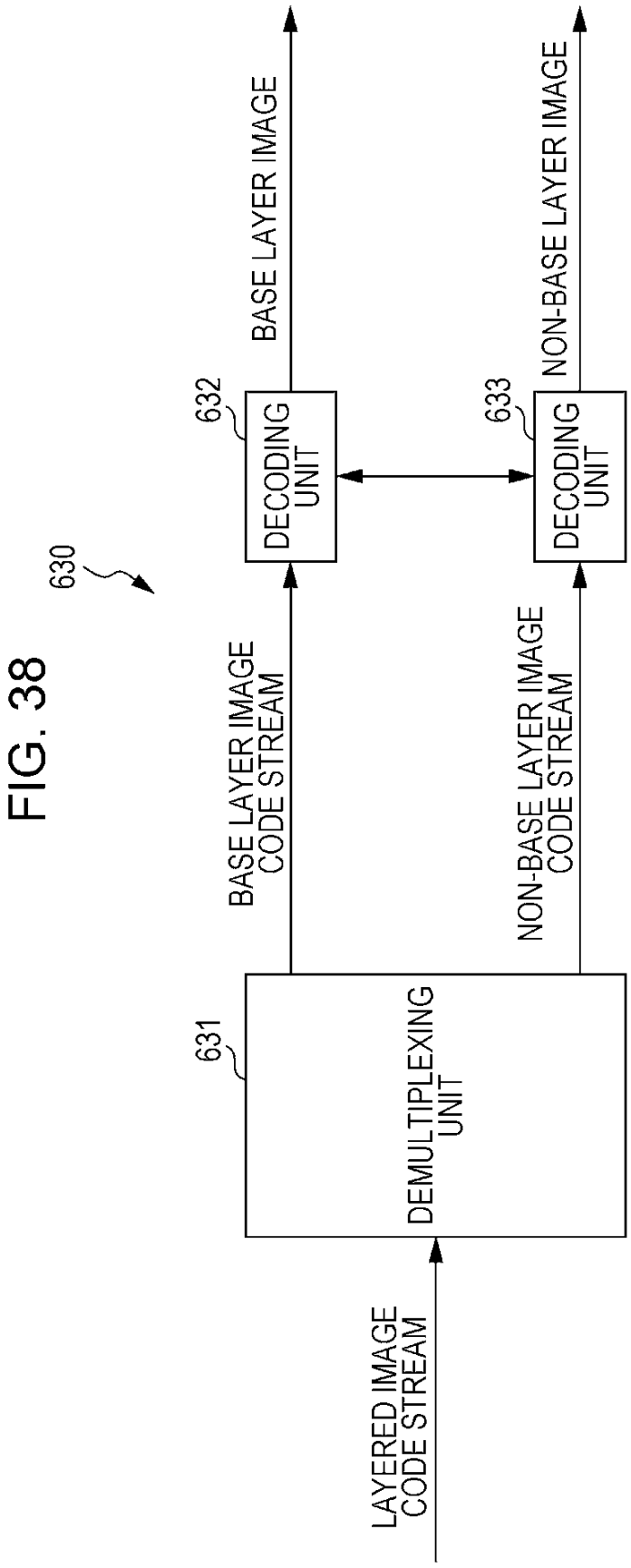
[Fig. 37]

FIG. 37


620
623
MULTIPLEXING UNIT
LAYERED IMAGE CODE STREAM
BASE LAYER IMAGE CODE STREAM
NON-BASE LAYER IMAGE CODE STREAM
621
622
CODING UNIT
CODING UNIT
BASE LAYER IMAGE
NON-BASE LAYER IMAGE

[Fig. 38]

FIG. 38


630
631
DEMULTIPLEXING UNIT
632
DECODING UNIT
633
DECODING UNIT
LAYERED IMAGE CODE STREAM
BASE LAYER IMAGE CODE STREAM
NON-BASE LAYER IMAGE CODE STREAM
BASE LAYER IMAGE
NON-BASE LAYER IMAGE

[Fig. 39]

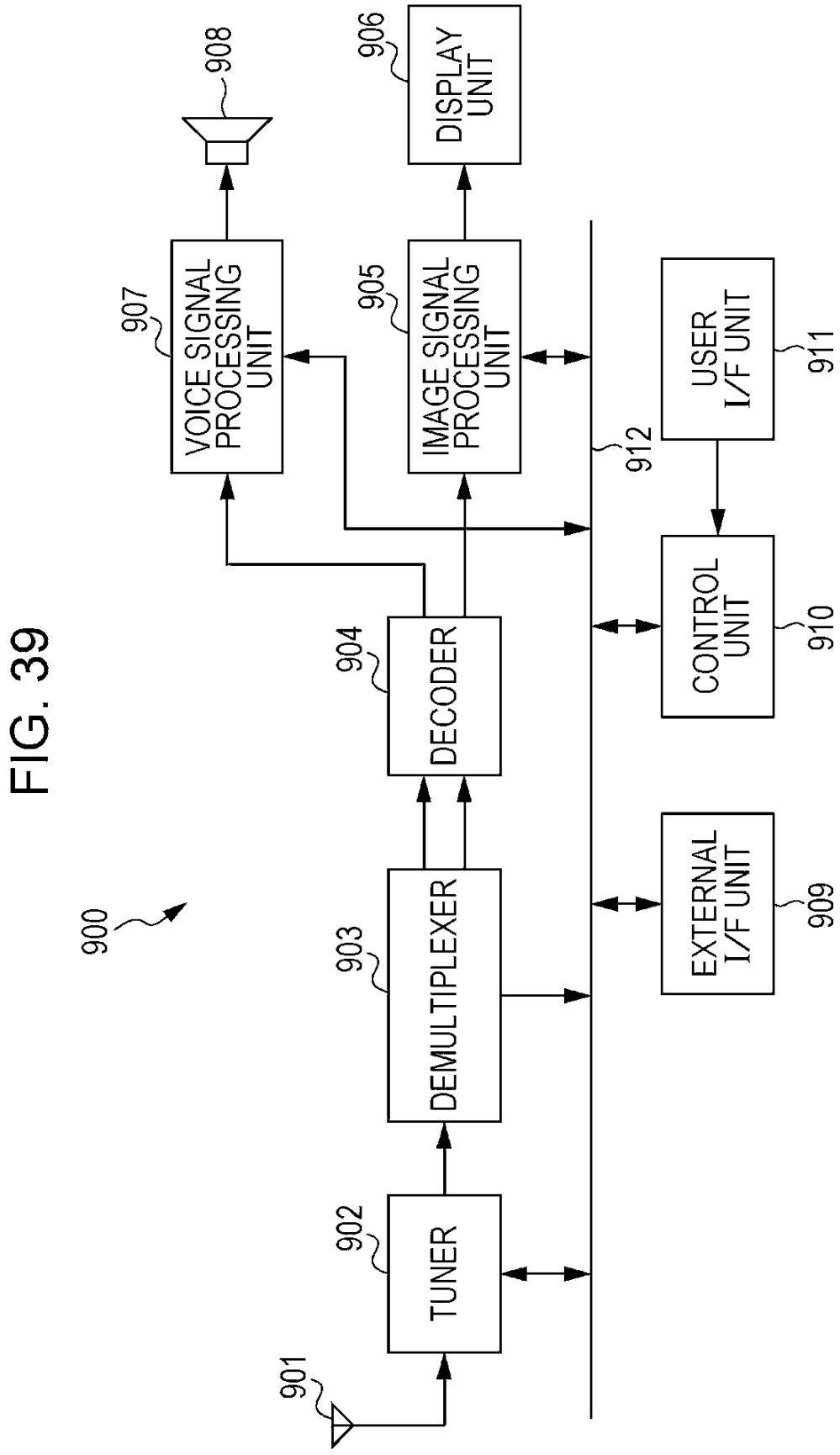
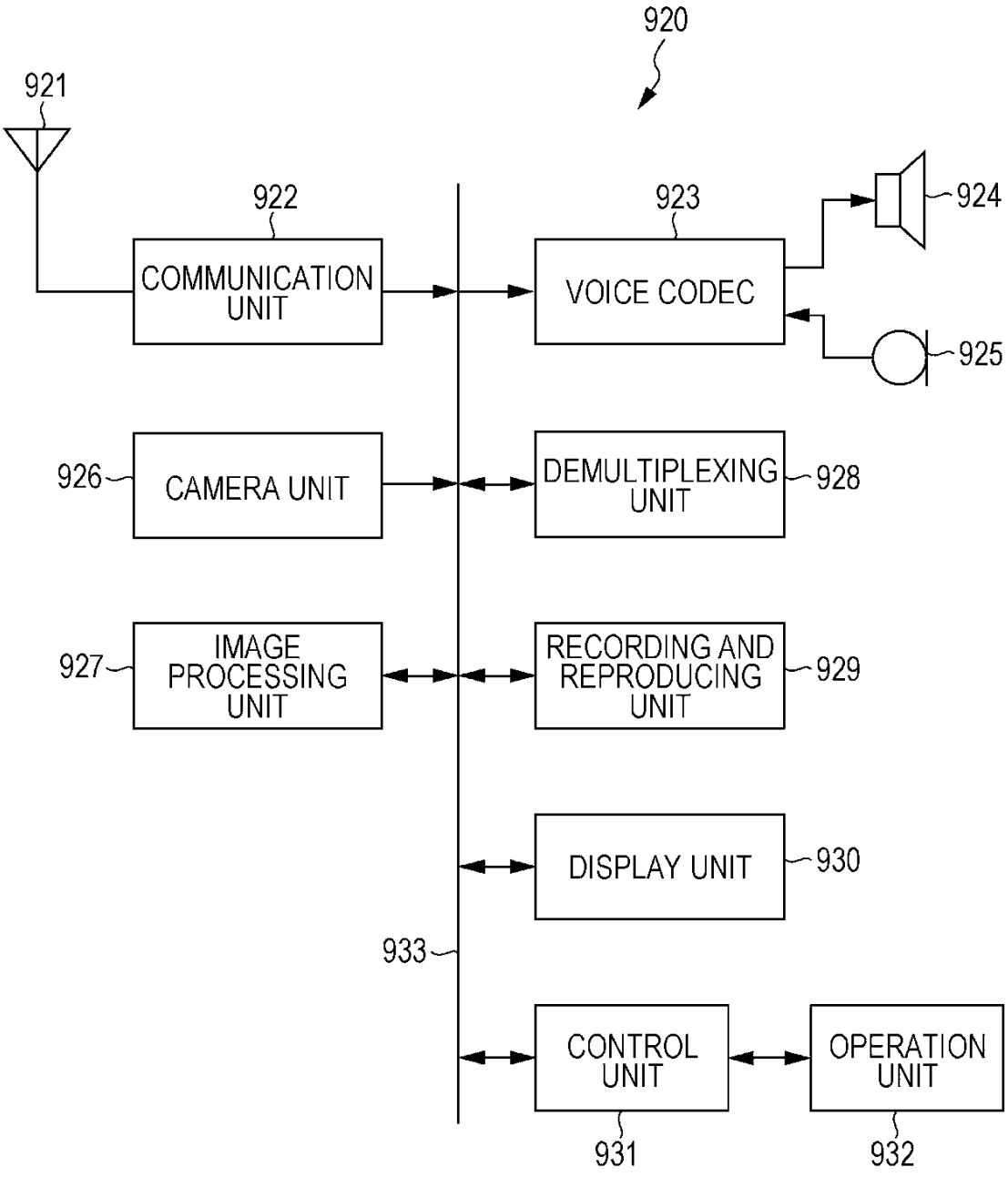

900
901
902
TUNER
903
DEMULTIPLEXER
904
DECODER
905
IMAGE SIGNAL PROCESSING UNIT
906
DISPLAY UNIT
907
VOICE SIGNAL PROCESSING UNIT
908
909
EXTERNAL I/F UNIT
910
CONTROL UNIT
911
USER I/F UNIT
912

FIG. 39

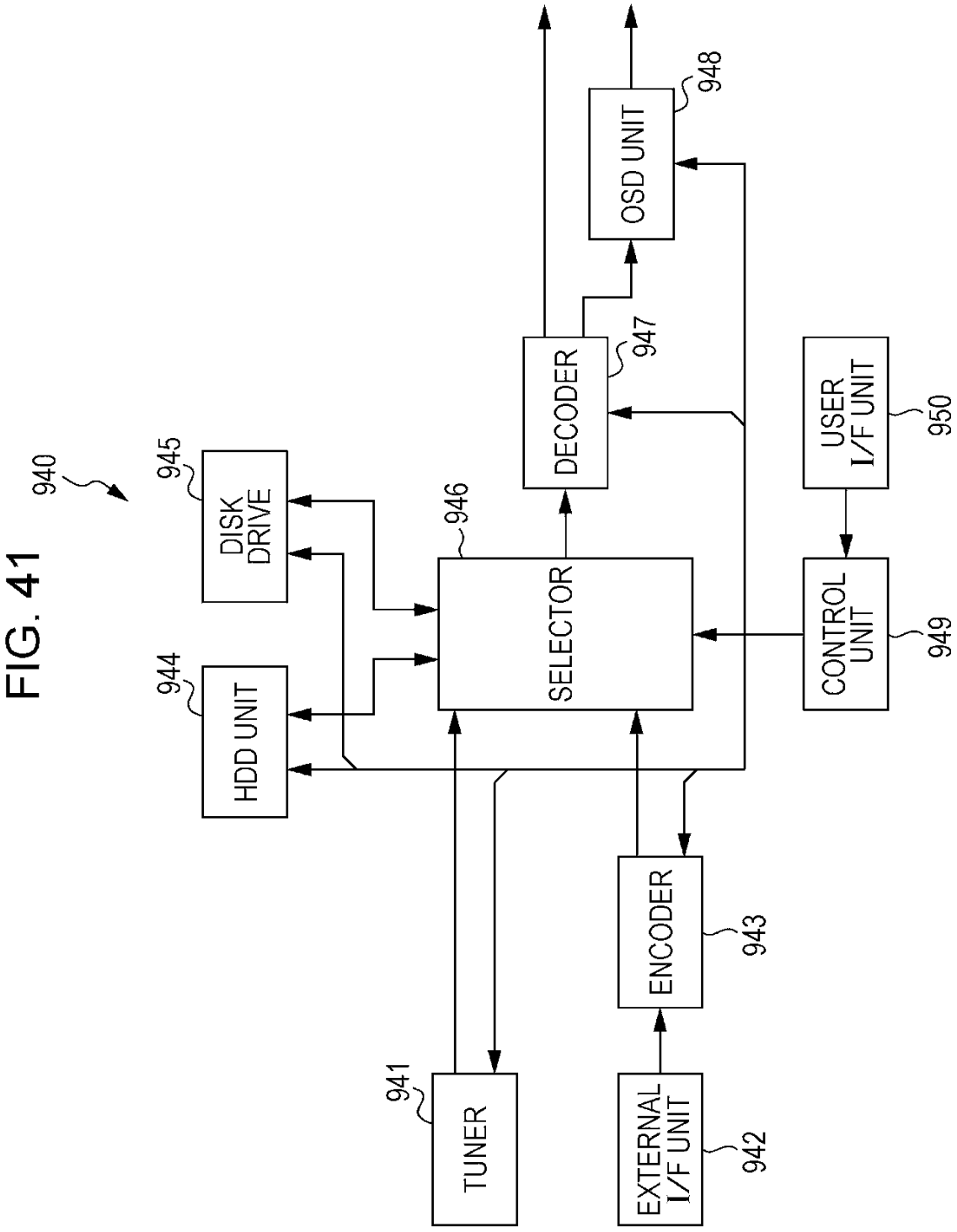
[Fig. 40]

FIG. 40


920
921
922
COMMUNICATION UNIT
923
VOICE CODEC
924
925
926
CAMERA UNIT
DEMULTIPLEXING UNIT
928
927
IMAGE PROCESSING UNIT
RECORDING AND REPRODUCING UNIT
929
DISPLAY UNIT
930
933
CONTROL UNIT
OPERATION UNIT
931
932

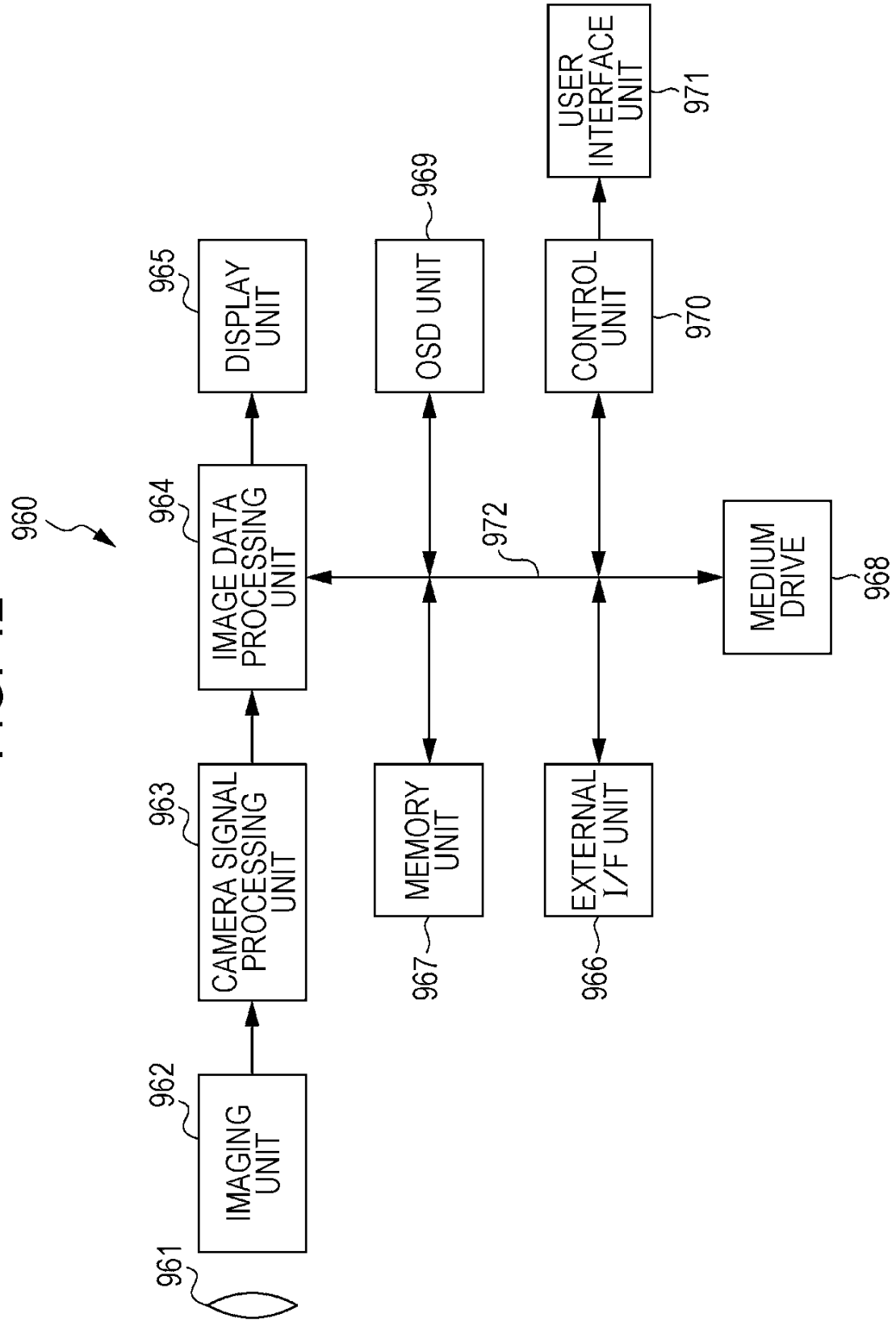
[Fig. 41]

FIG. 41


940
945
DISK DRIVE
944
HDD UNIT
946
SELECTOR
DECODER
947
OSD UNIT
948
USER I/F UNIT
950
CONTROL UNIT
949
941
TUNER
942
EXTERNAL I/F UNIT
ENCODER
943

[Fig. 42]

FIG. 42


960
961
962
IMAGING UNIT
963
CAMERA SIGNAL PROCESSING UNIT
964
IMAGE DATA PROCESSING UNIT
965
DISPLAY UNIT
966
EXTERNAL I/F UNIT
967
MEMORY UNIT
968
MEDIUM DRIVE
969
OSD UNIT
970
CONTROL UNIT
971
USER INTERFACE UNIT
972

[Fig. 43]

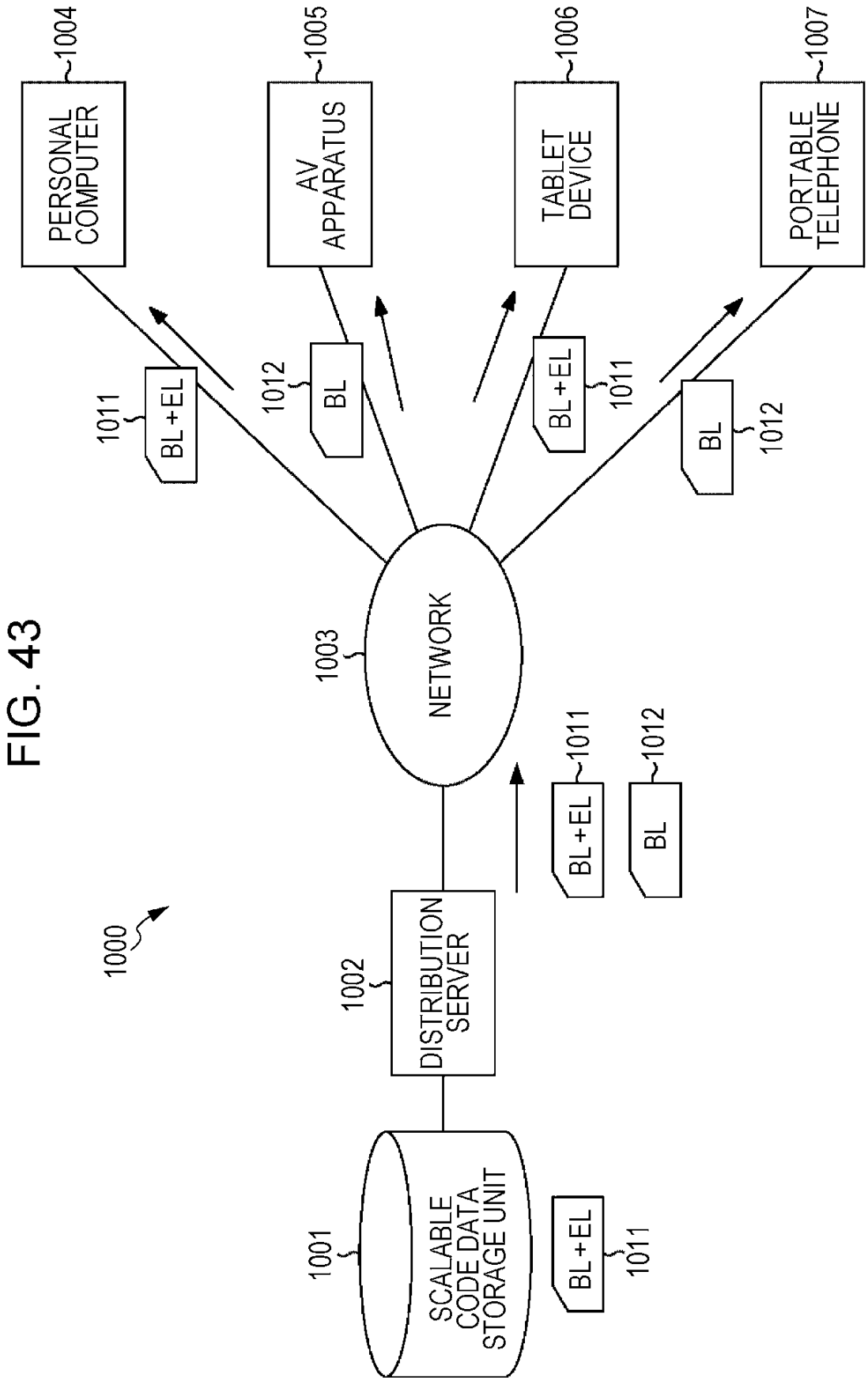
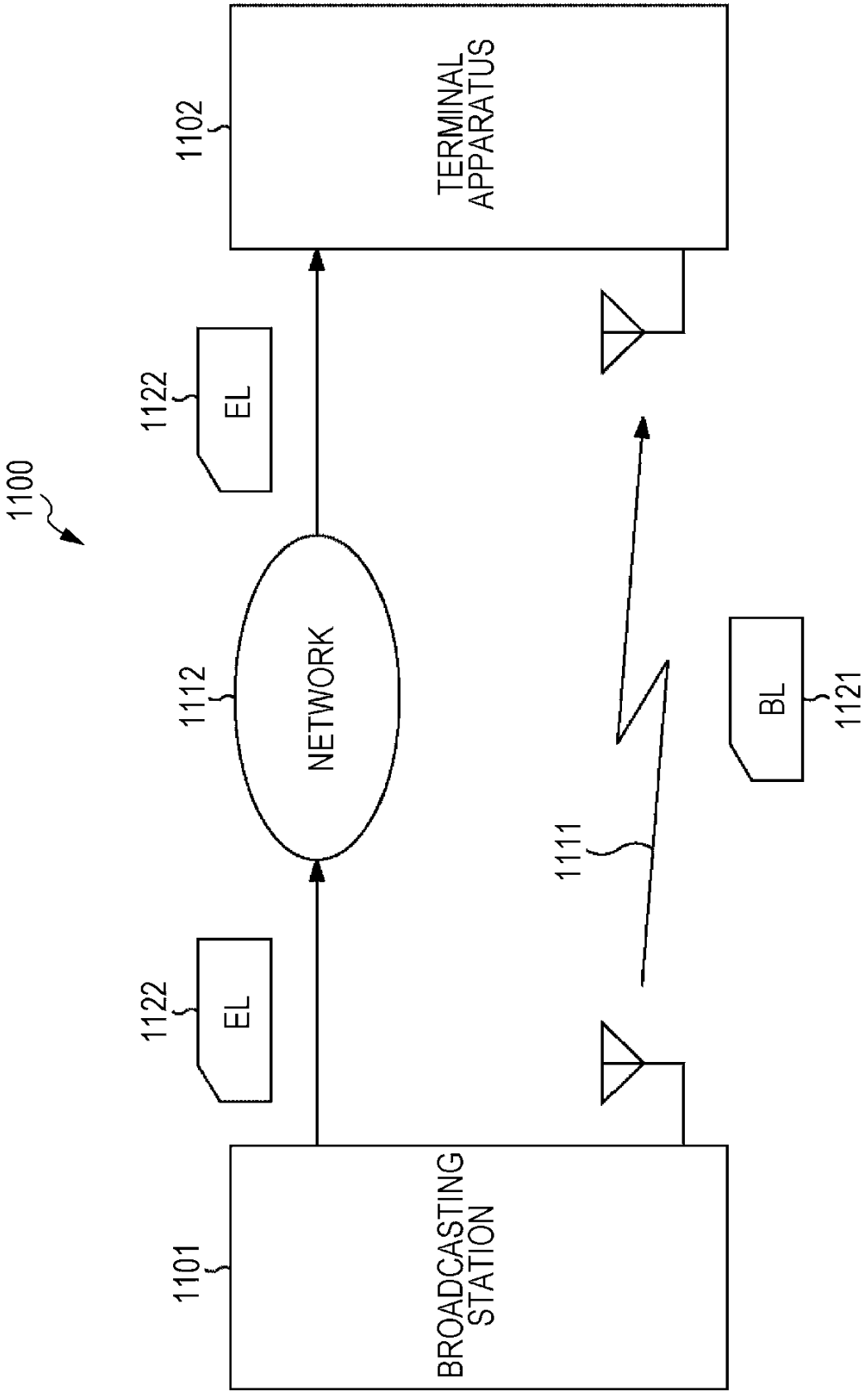

1000
SCALABLE CODE DATA STORAGE UNIT
1001
BL+EL
1011
DISTRIBUTION SERVER
1002
NETWORK
1003
BL+EL
1011
BL
1012
PERSONAL COMPUTER
1004
AV APPARATUS
1005
TABLET DEVICE
1006
PORTABLE TELEPHONE
1007
BL+EL
1011
BL
1012
BL+EL
1011
BL
1012

FIG. 43

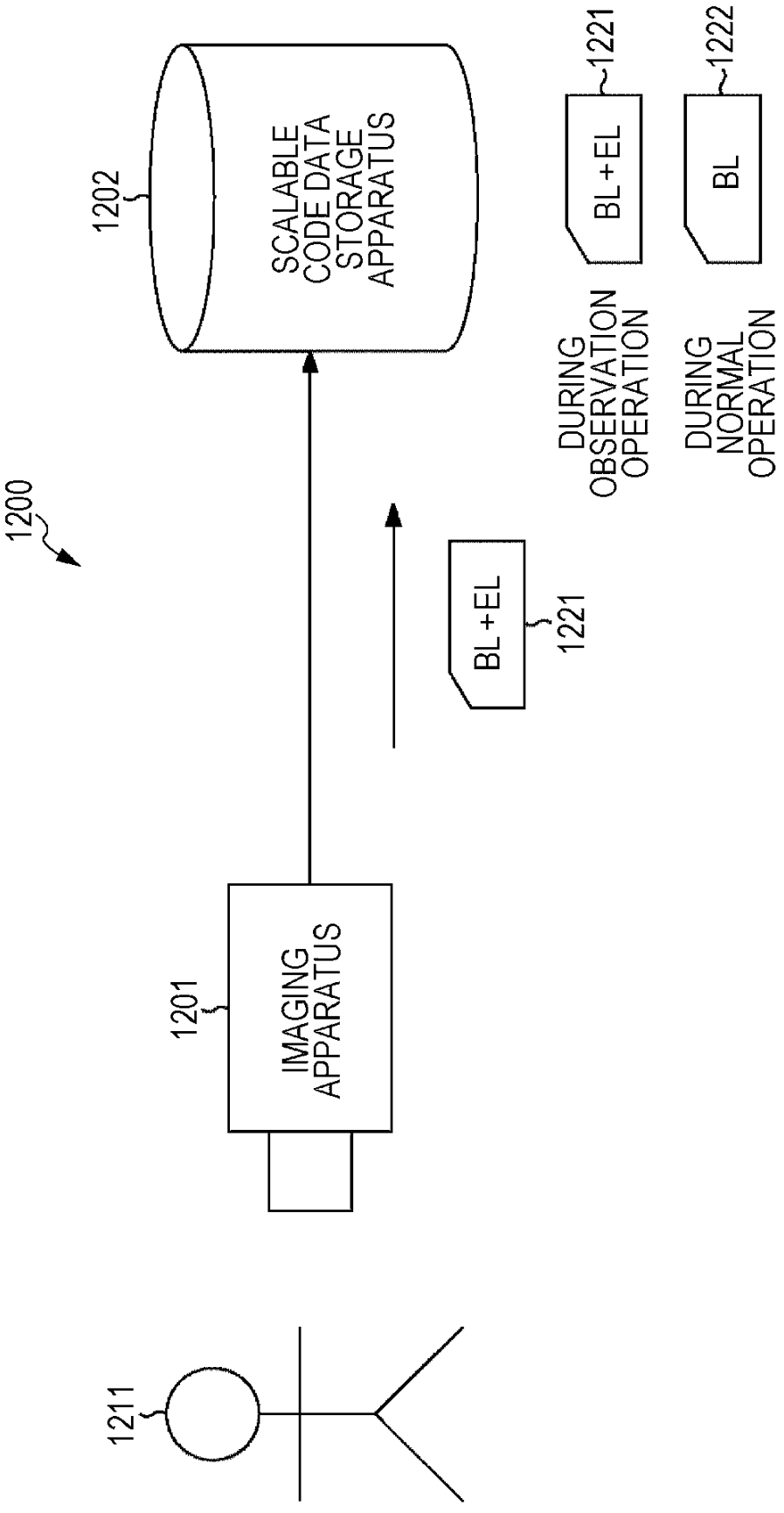
[Fig. 44]

FIG. 44


1100
1101
BROADCASTING STATION
1102
TERMINAL APPARATUS
1111
1112
NETWORK
1121
BL
1122
EL
1122
EL

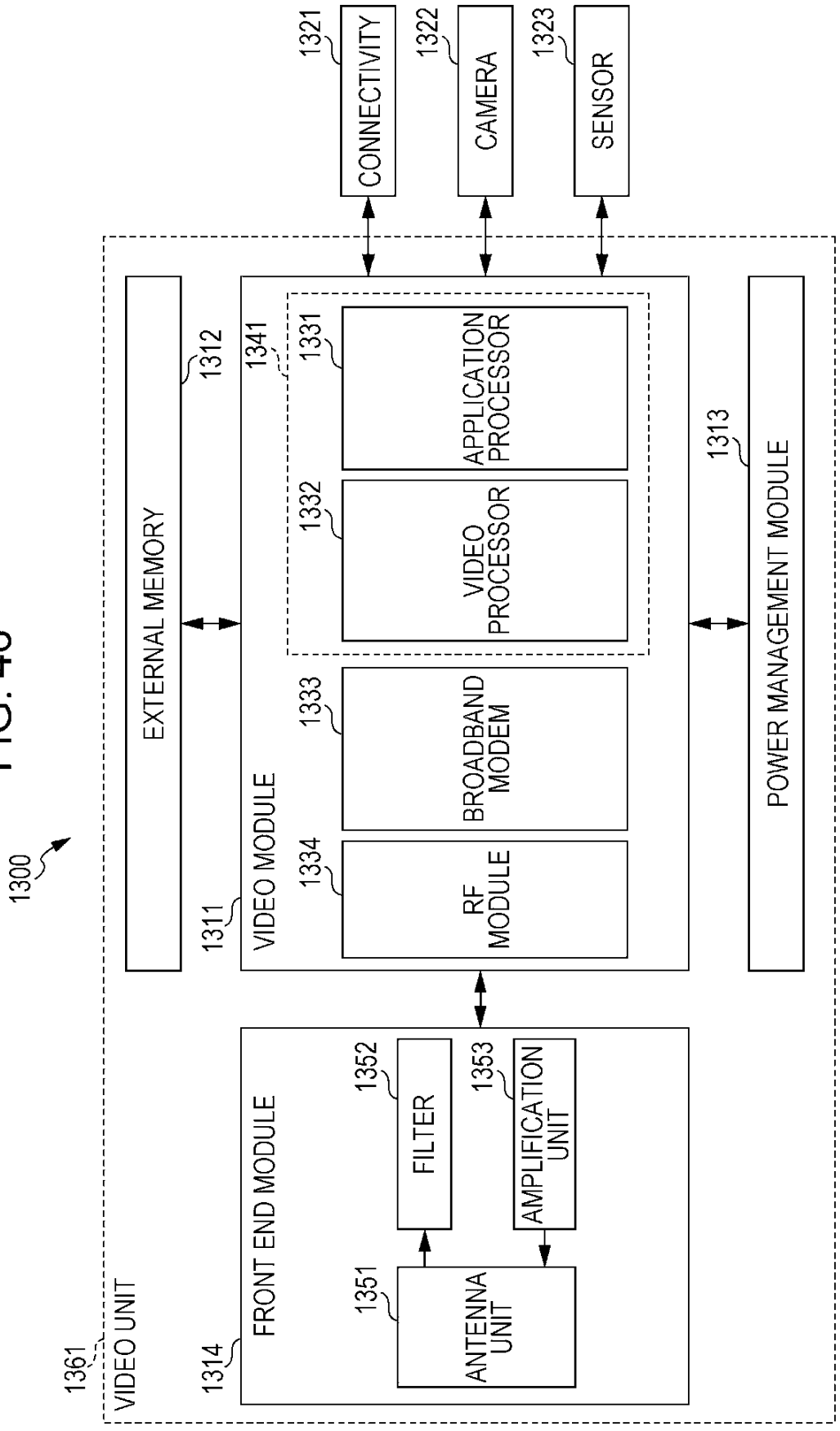
[Fig. 45]

FIG. 45


1200
1202
SCALABLE CODE DATA STORAGE APPARATUS
1201
IMAGING APPARATUS
1211
BL+EL
1221
DURING OBSERVATION OPERATION
BL+EL
1221
DURING NORMAL OPERATION
BL
1222

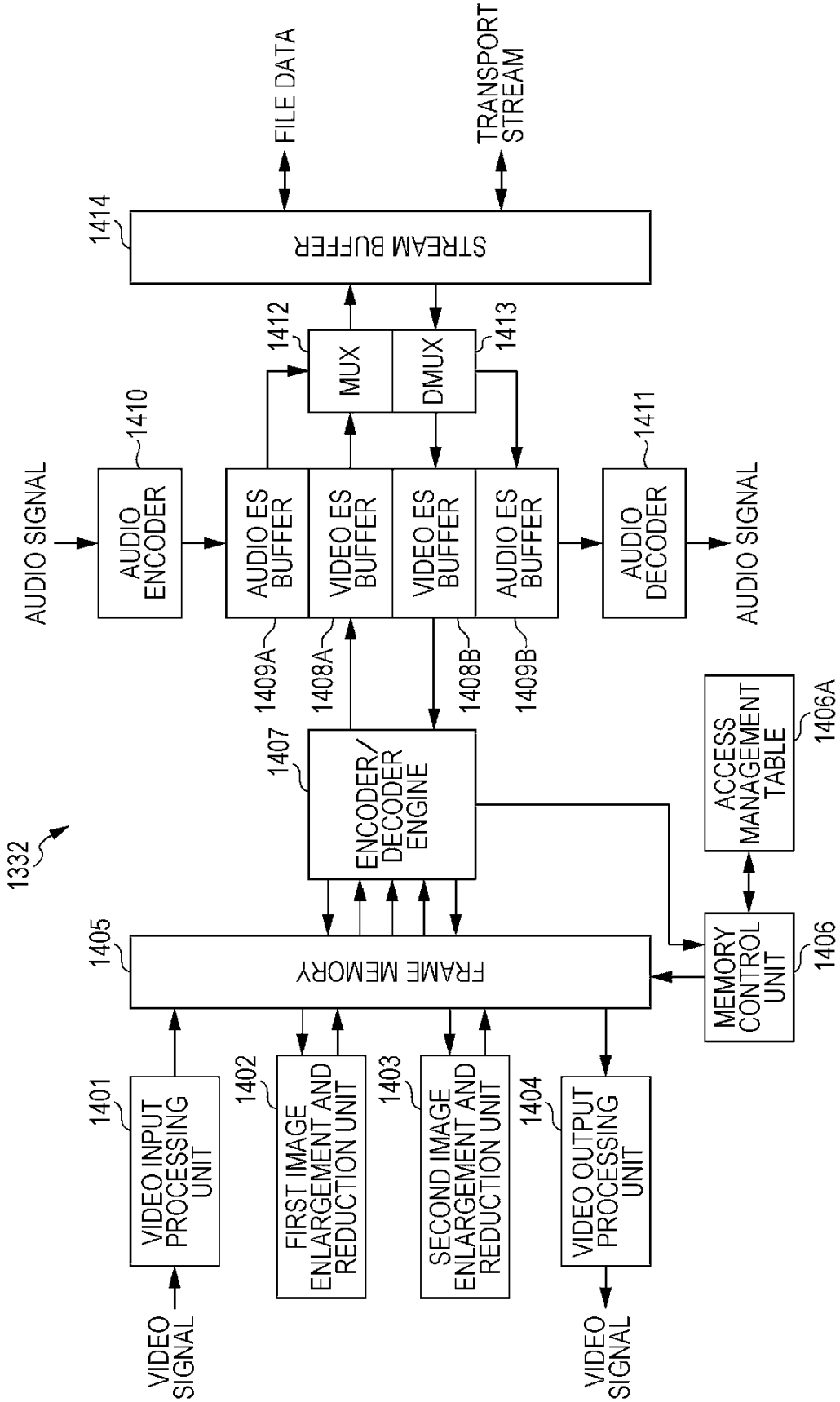
[Fig. 46]

FIG. 46


1300
VIDEO UNIT
1361
EXTERNAL MEMORY
1312
VIDEO MODULE
1311
1341
APPLICATION PROCESSOR
1331
VIDEO PROCESSOR
1332
BROADBAND MODEM
1333
RF MODULE
1334
POWER MANAGEMENT MODULE
1313
FRONT END MODULE
1314
FILTER
1352
AMPLIFICATION UNIT
1353
ANTENNA UNIT
1351
CONNECTIVITY
1321
CAMERA
1322
SENSOR
1323

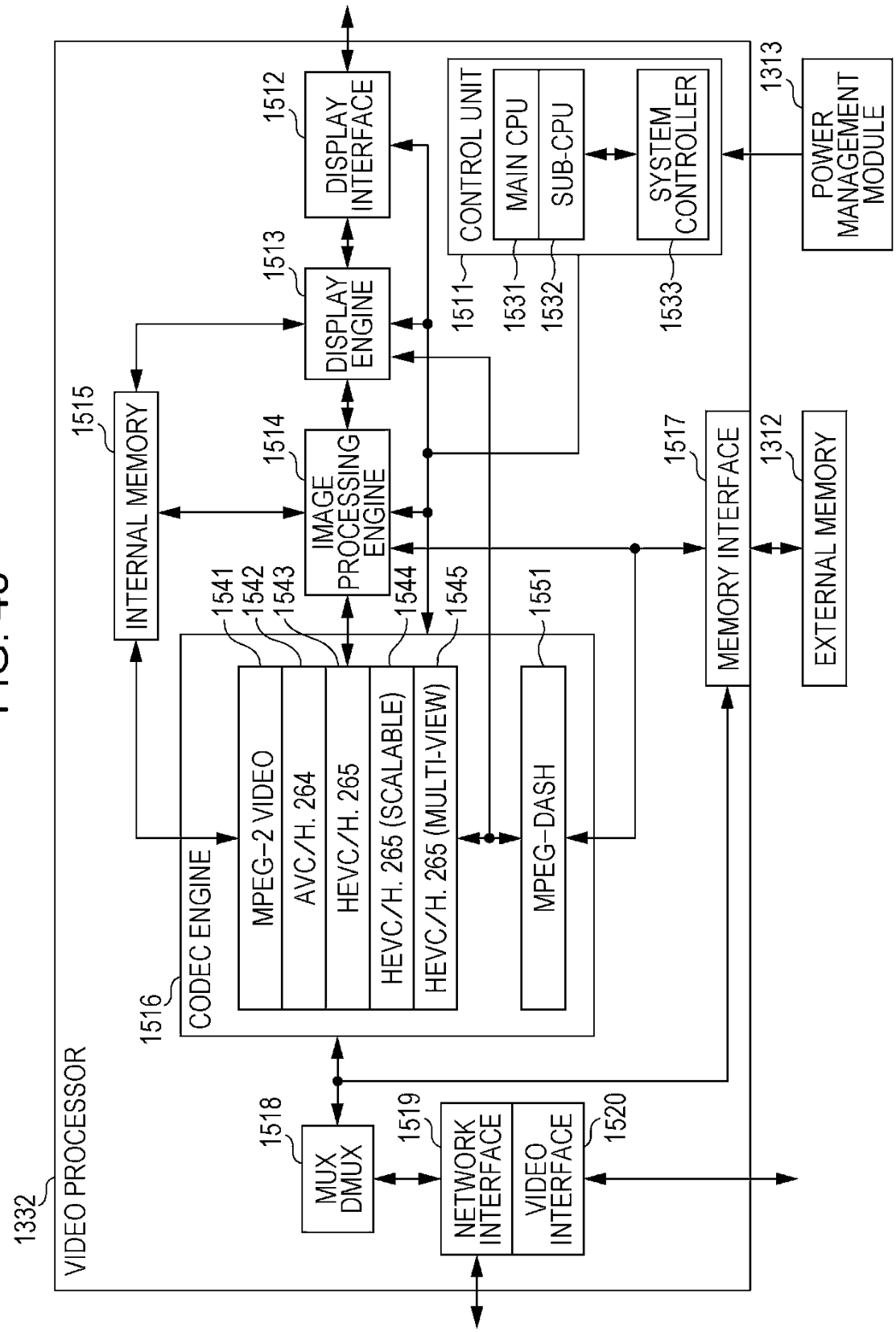
[Fig. 47]

FIG. 47


1332
VIDEO SIGNAL
1401
VIDEO INPUT PROCESSING UNIT
1402
FIRST IMAGE ENLARGEMENT AND REDUCTION UNIT
1403
SECOND IMAGE ENLARGEMENT AND REDUCTION UNIT
1404
VIDEO OUTPUT PROCESSING UNIT
VIDEO SIGNAL
1405
FRAME MEMORY
1406
MEMORY CONTROL UNIT
1406A
ACCESS MANAGEMENT TABLE
1407
ENCODER/ DECODER ENGINE
1408A
1408B
1409A
1409B
AUDIO SIGNAL
1410
AUDIO ENCODER
AUDIO ES BUFFER
VIDEO ES BUFFER
VIDEO ES BUFFER
AUDIO ES BUFFER
1411
AUDIO DECODER
AUDIO SIGNAL
1412
MUX
DMUX
1413
1414
STREAM BUFFER
FILE DATA
TRANSPORT STREAM

[Fig. 48]

FIG. 48


VIDEO PROCESSOR
1332
INTERNAL MEMORY
1515
DISPLAY INTERFACE
1512
DISPLAY ENGINE
1513
IMAGE PROCESSING ENGINE
1514
CONTROL UNIT
1511
MAIN CPU
1531
SUB-CPU
1532
SYSTEM CONTROLLER
1533
POWER MANAGEMENT MODULE
1313
MEMORY INTERFACE
1517
EXTERNAL MEMORY
1312
CODEC ENGINE
1516
MPEG-2 VIDEO
1541
AVC/H. 264
1542
HEVC/H. 265
1543
HEVC/H. 265 (SCALABLE)
1544
HEVC/H. 265 (MULTI-VIEW)
1545
MPEG-DASH
1551
MUX DMUX
1518
NETWORK INTERFACE
1519
VIDEO INTERFACE
1520