FIG. 1

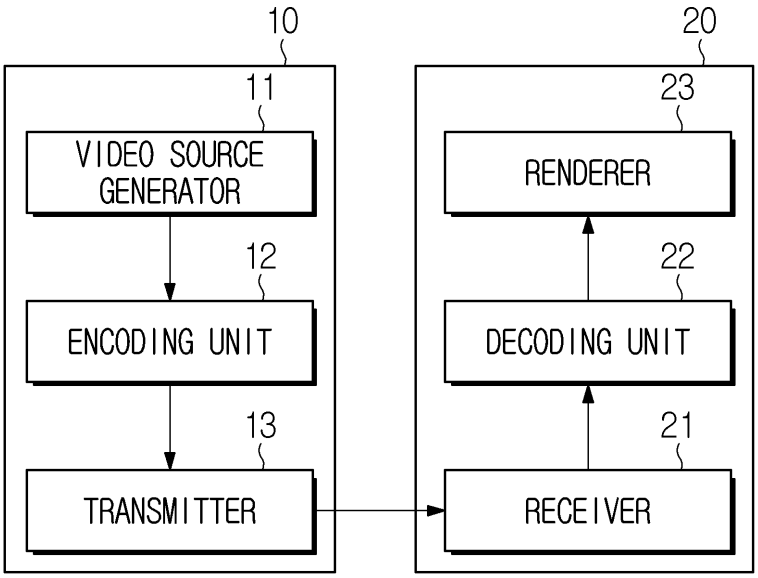

10
20
11
23
VIDEO SOURCE GENERATOR
RENDERER
12
22
ENCODING UNIT
DECODING UNIT
13
21
TRANSMITTER
RECEIVER

FIG. 2

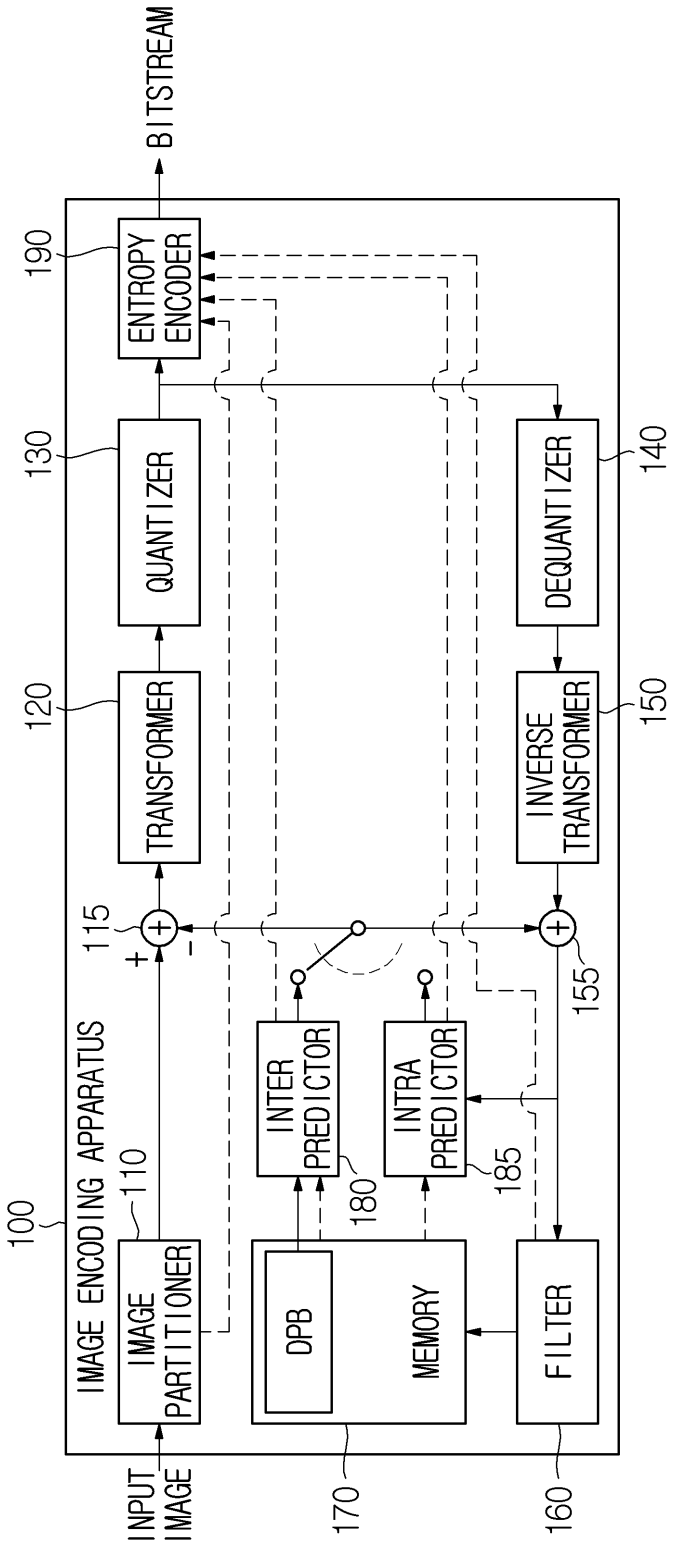

BITSTREAM
ENTROPY ENCODER
190
QUANTIZER
130
TRANSFORMER
120
115
+
-
IMAGE ENCODING APPARATUS
100
IMAGE PARTITIONER
110
INPUT IMAGE
INTER PREDICTOR
180
INTRA PREDICTOR
185
DPB
MEMORY
170
DEQUANTIZER
140
INVERSE TRANSFORMER
150
155
FILTER
160

FIG. 3

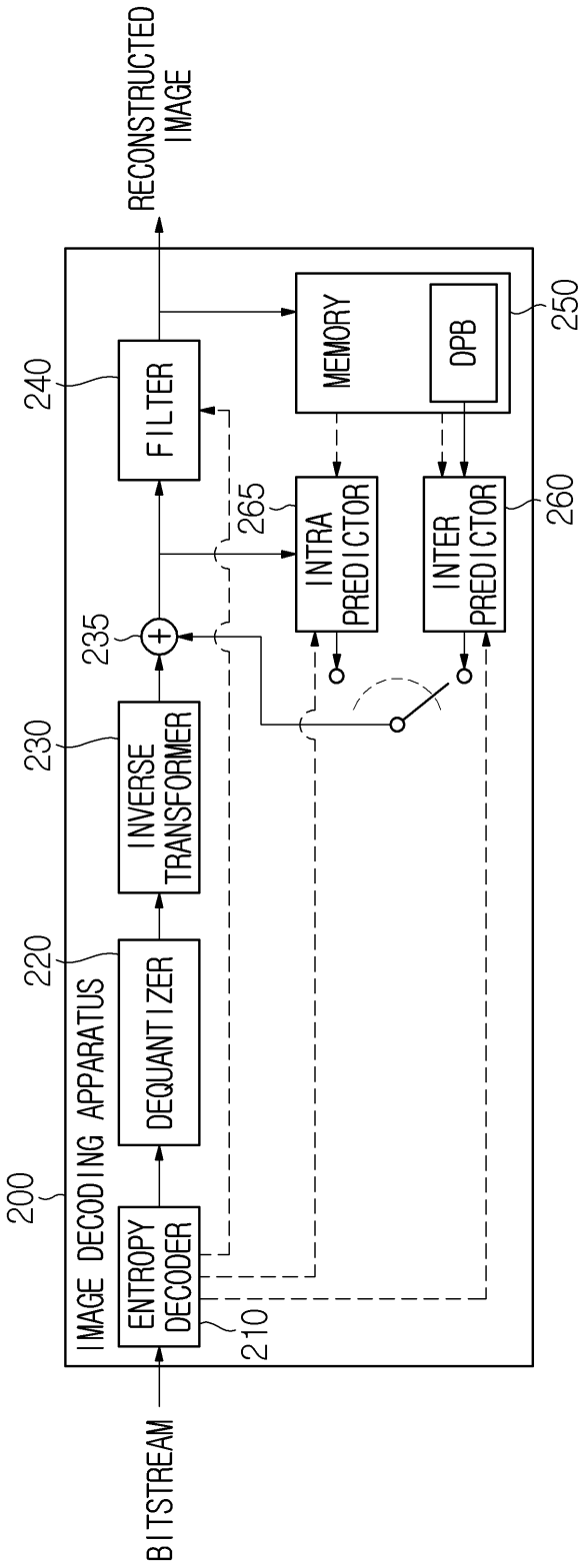

RECONSTRUCTED IMAGE
IMAGE DECODING APPARATUS
200
BITSTREAM
ENTROPY DECODER
210
DEQUANTIZER
220
INVERSE TRANSFORMER
230
235
FILTER
240
265
INTRA PREDICTOR
INTER PREDICTOR
260
MEMORY
DPB
250

FIG. 4

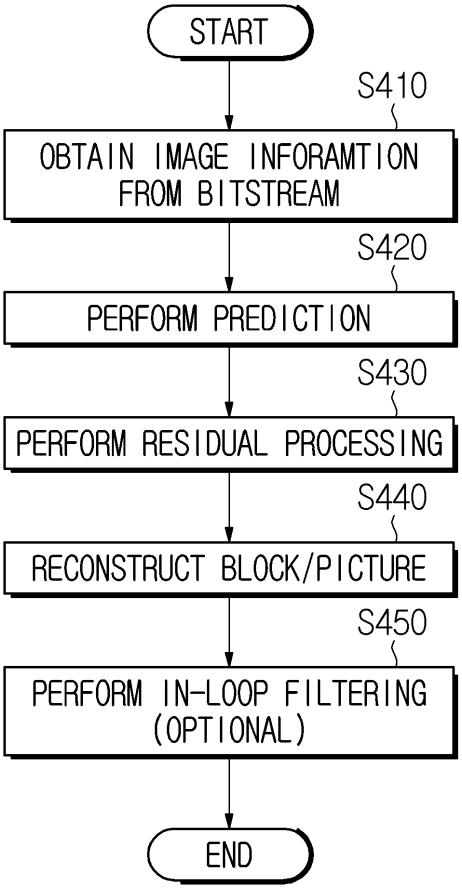

START
S410
OBTAIN IMAGE INFORAMTION FROM BITSTREAM
S420
PERFORM PREDICTION
S430
PERFORM RESIDUAL PROCESSING
S440
RECONSTRUCT BLOCK/PICTURE
S450
PERFORM IN-LOOP FILTERING (OPTIONAL)
END

FIG. 5

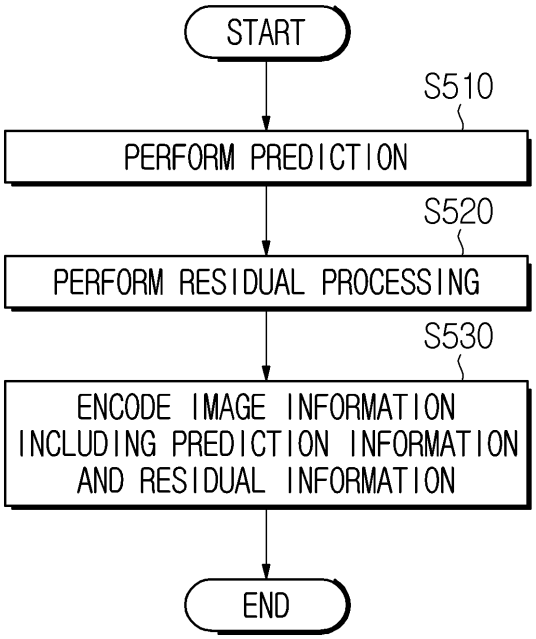

START
S510
PERFORM PREDICTION
S520
PERFORM RESIDUAL PROCESSING
S530
ENCODE IMAGE INFORMATION INCLUDING PREDICTION INFORMATION AND RESIDUAL INFORMATION
END

FIG. 6

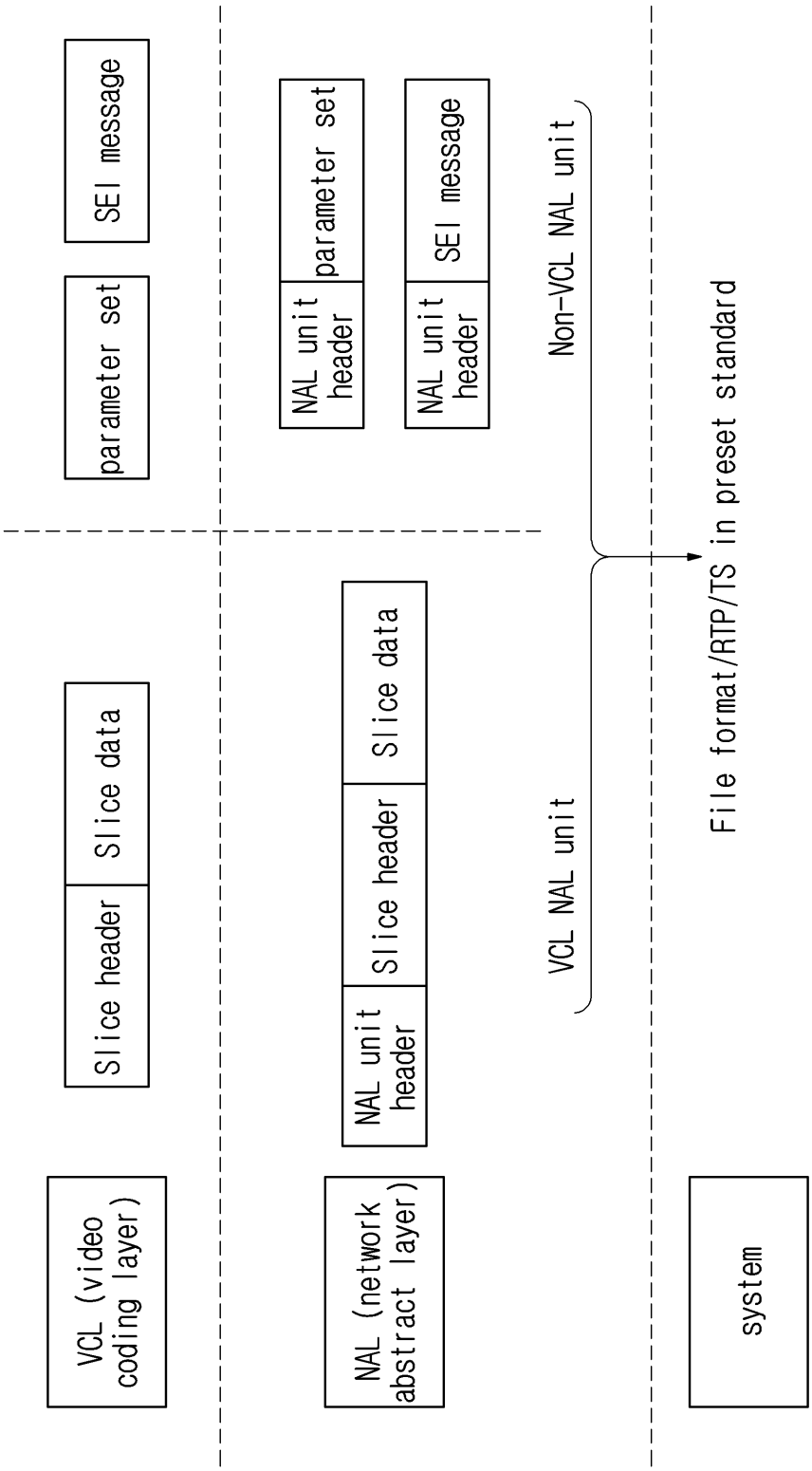

VCL (video coding layer)
NAL (network abstract layer)
system
Slice header
Slice data
NAL unit header
Slice header
Slice data
parameter set
SEI message
NAL unit header
parameter set
NAL unit header
SEI message
VCL NAL unit
Non-VCL NAL unit
File format/RTP/TS in preset standard

FIG. 7

pic_parameter_set_rbsp( ) {	Descriptor
…	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
…	
}	
entropy_coding_sync_enabled_flag	u(1)
if(!no_pic_partition_flag || entropy_coding_sync_enabled_flag)	
entry_point_offsets_present_flag	u(1)
…	u(1)
}	

FIG. 8

slice_header( ) {	Descriptor
…	
if(entry_point_offsets_present_flag && NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
…	
}	

FIG. 9

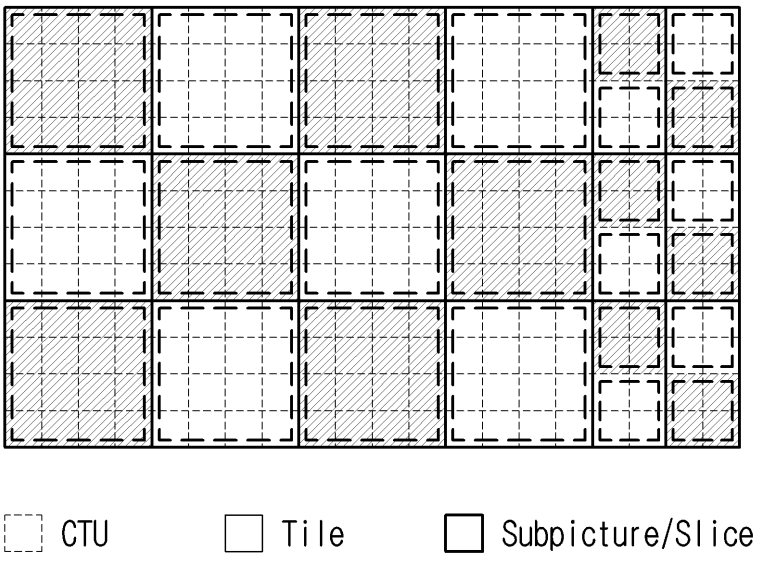


FIG. 10

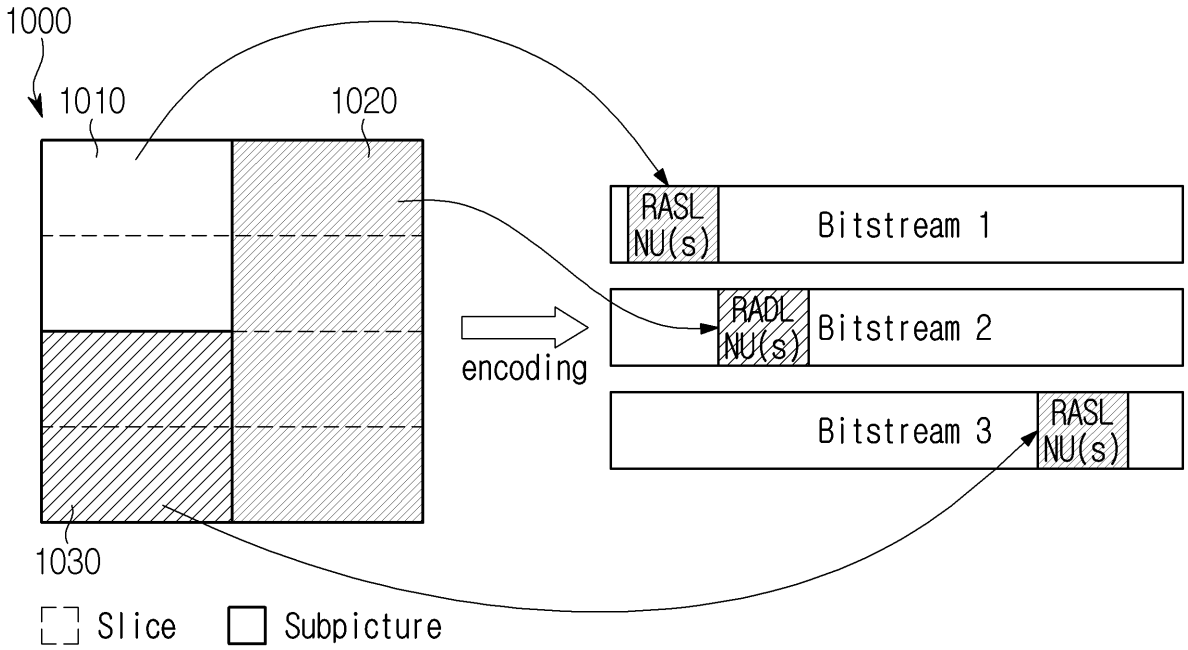


FIG. 11

pic_parameter_set_rbsp( ) {	**Descriptor**
…	
pps_mixed_nalu_types_in_pic_flag	
…	
pps_no_pic_partition_flag	u(1)
pps_subpic_id_mapping_present_flag	u(1)
if(pps_subpic_id_mapping_present_flag) {	
if(!pps_no_pic_partition_flag)	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpics_minus1; i++)	
pps_subpic_id[i]	u(v)
}	
…	

FIG. 12

seq_parameter_set_rbsp( ) {	**Descriptor**
…	
sps_subpic_info_present_flag	u(1)
if(sps_subpic_info_present_flag) {	
sps_num_subpics_minus1	ue(v)
if(sps_num_subpics_minus1 > 0) {	
sps_independent_subpics_flag	u(1)
sps_subpic_same_size_flag	u(1)
}	
…	
if(!sps_independent_subpics_flag) {	
sps_subpic_treated_as_pic_flag[i]	u(1)
…	

FIG. 13

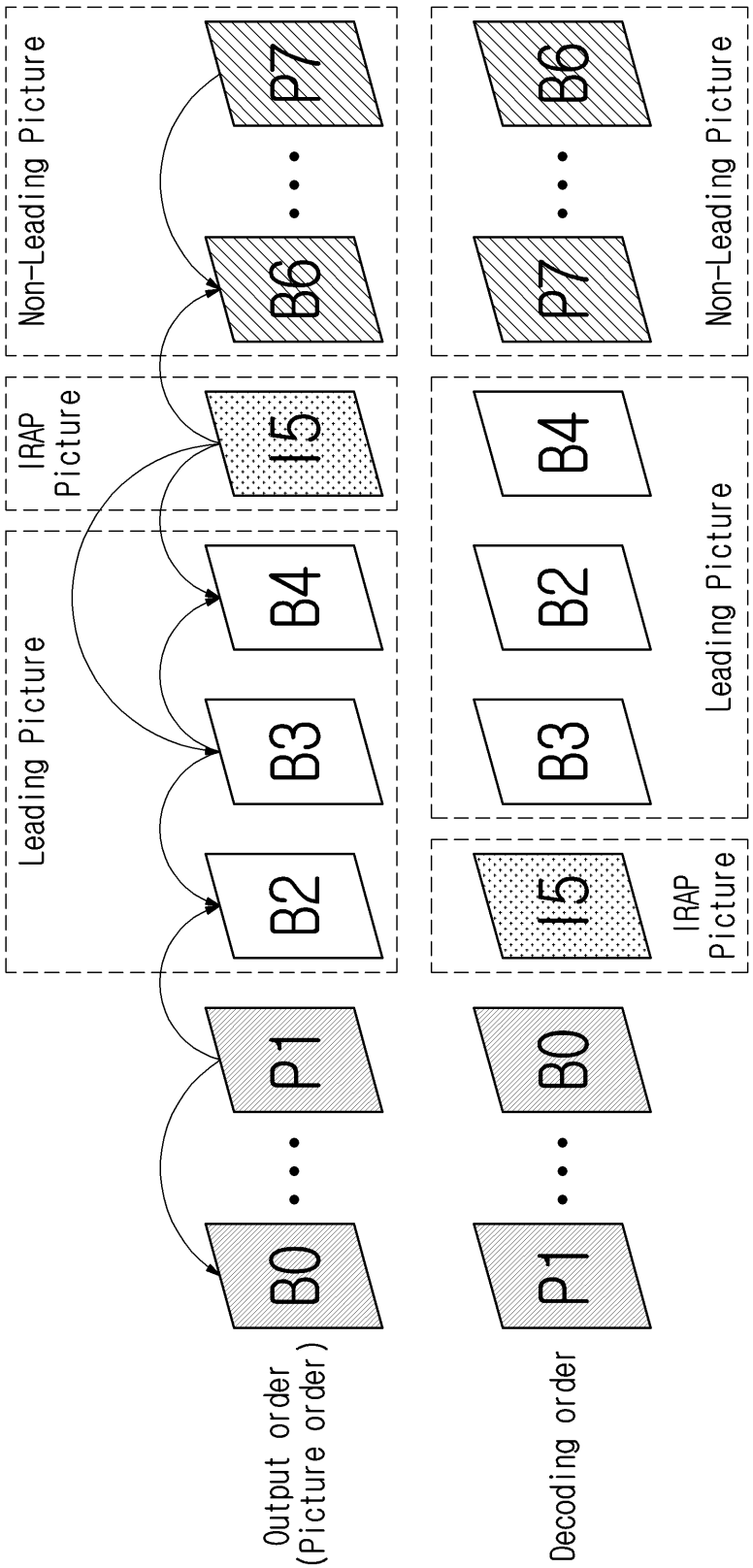

Output order
(Picture order)
B0
P1
B2
B3
B4
I5
B6
P7
Leading Picture
IRAP
Picture
Non-Leading Picture
Decoding order
P1
B0
I5
B3
B2
B4
P7
B6
IRAP
Picture
Leading Picture
Non-Leading Picture

FIG. 14

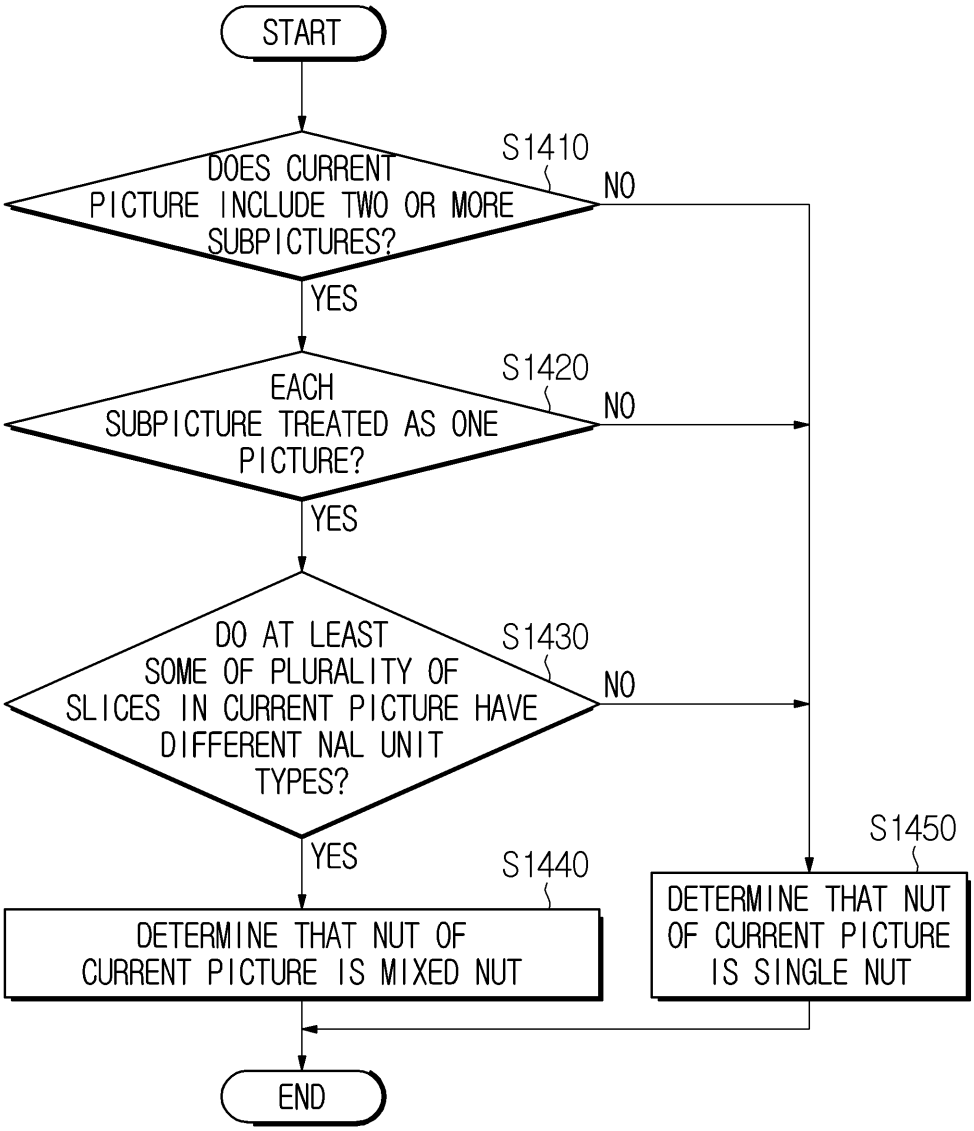

START
S1410
DOES CURRENT PICTURE INCLUDE TWO OR MORE SUBPICTURES?
NO
YES
S1420
EACH SUBPICTURE TREATED AS ONE PICTURE?
NO
YES
S1430
DO AT LEAST SOME OF PLURALITY OF SLICES IN CURRENT PICTURE HAVE DIFFERENT NAL UNIT TYPES?
NO
YES
S1440
DETERMINE THAT NUT OF CURRENT PICTURE IS MIXED NUT
S1450
DETERMINE THAT NUT OF CURRENT PICTURE IS SINGLE NUT
END

FIG. 15

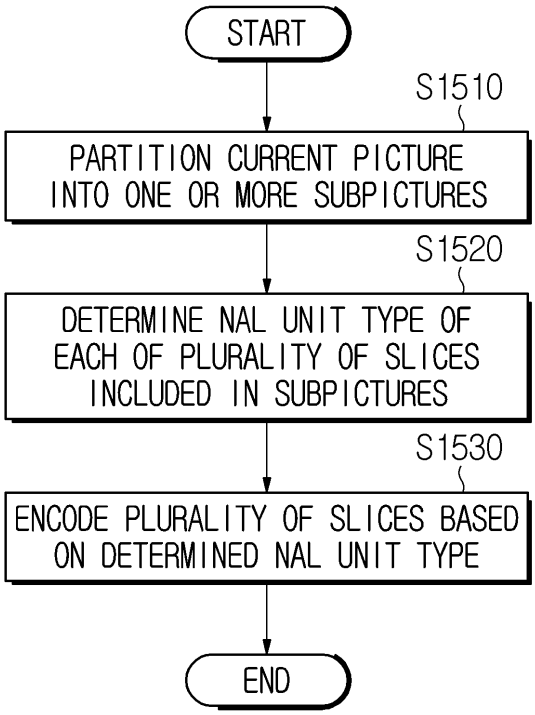

START
S1510
PARTITION CURRENT PICTURE INTO ONE OR MORE SUBPICTURES
S1520
DETERMINE NAL UNIT TYPE OF EACH OF PLURALITY OF SLICES INCLUDED IN SUBPICTURES
S1530
ENCODE PLURALITY OF SLICES BASED ON DETERMINED NAL UNIT TYPE
END

FIG. 16

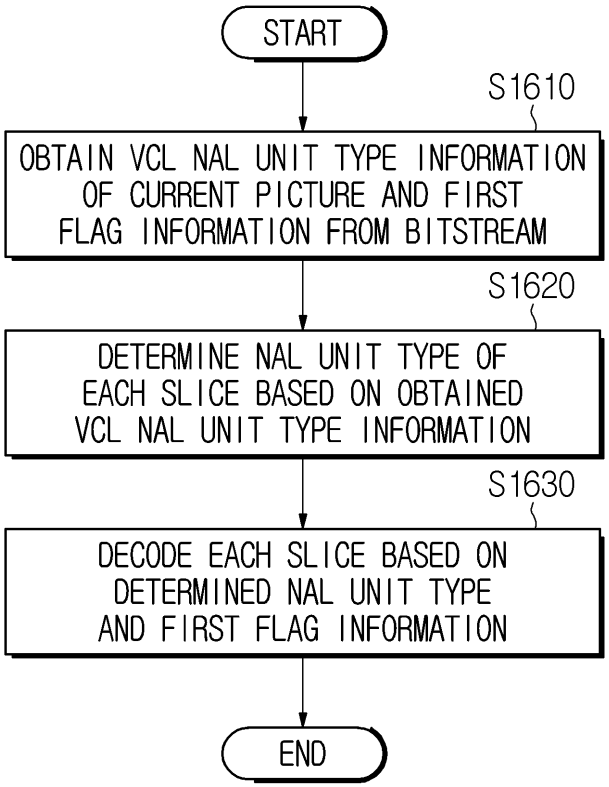
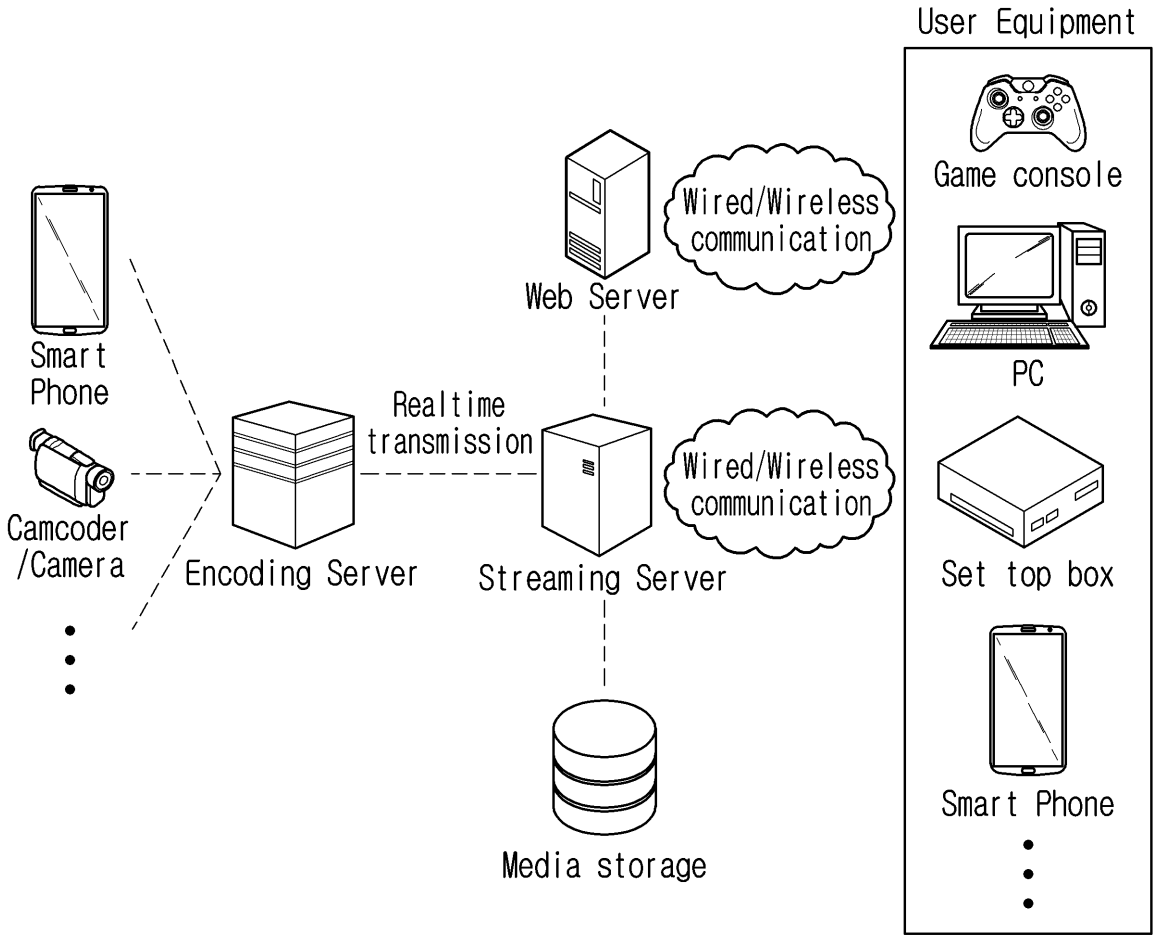

START
S1610
OBTAIN VCL NAL UNIT TYPE INFORMATION OF CURRENT PICTURE AND FIRST FLAG INFORMATION FROM BITSTREAM
S1620
DETERMINE NAL UNIT TYPE OF EACH SLICE BASED ON OBTAINED VCL NAL UNIT TYPE INFORMATION
S1630
DECODE EACH SLICE BASED ON DETERMINED NAL UNIT TYPE AND FIRST FLAG INFORMATION
END

FIG. 17


User Equipment
Game console
Wired/Wireless communication
Web Server
Smart Phone
PC
Realtime transmission
Camcoder /Camera
Encoding Server
Streaming Server
Wired/Wireless communication
Set top box
Smart Phone
Media storage