



(51) International Patent Classification:  
H04N 19/46 (2014.01) H04N 19/70 (2014.01)

(21) International Application Number:  
PCT/EP2020/069304

(22) International Filing Date:  
08 July 2020 (08.07.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
62/873,812 12 July 2019 (12.07.2019) US

(71) Applicant: TELEFONAKTIEBOLAGET LM ERICSSON (PUBL) [SE/SE]; SE-164 83 Stockholm (SE).

(72) Inventors: PETERSSON, Martin; Stenvallavägen 10, 186 34 Vallentuna (SE). DAMGHANIAN, Mitra; Markörvägen 3, 19792 Upplands-Bro (SE). SJÖBERG, Rickard; Sankt Eriksgatan 84, 113 62 STOCKHOLM (SE).

(74) Agent: ERICSSON; Patent Development, Torshamnsgatan 21-23, 164 80 STOCKHOLM (SE).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,

(54) Title: SIGNALING OF ACCESS UNIT DELIMITER

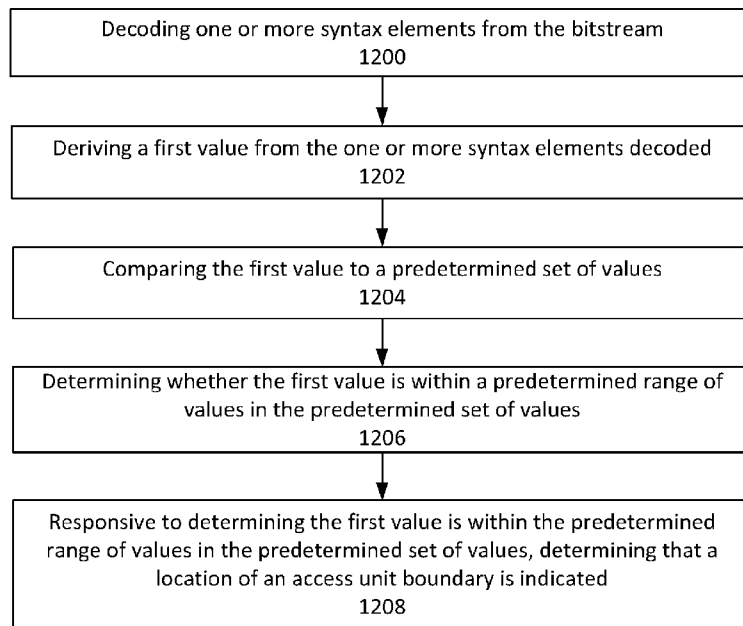


Figure 12

(57) Abstract: A method performed by a decoder to decode a bitstream and a decoder is provided. One or more syntax elements are decoded from the bitstream. A first value is derived from the one or more syntax elements. The first value is compared to a predetermined set of values that comprises at least two values. Whether the first value is within a predetermined range of values in the predetermined set of values is determined. Responsive to determining the first value is within the predetermined range of values in the predetermined set of values, it is determined that a location of an access unit boundary is indicated. Analogous methods performed by an encoder and an encoder are also provided.



MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,  
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,  
KM, ML, MR, NE, SN, TD, TG).

**Published:**

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

## SIGNALING OF ACCESS UNIT DELIMITER

### TECHNICAL FIELD

**[0001]** The application relates to methods and apparatuses for video encoding and decoding.

### BACKGROUND

**[0002]** HEVC and VVC

**[0003]** High Efficiency Video Coding (HEVC) is a block-based video codec standardized by the ITU-T (Telecommunication Standardization Sector of the International Telecommunications Union) and the MPEG (Moving Pictures Expert Group) that utilizes both temporal and spatial prediction. Spatial prediction is achieved using intra (I) prediction from within the current picture. Temporal prediction is achieved using uni-directional (P) or bi-directional inter (B) prediction on block level from previously decoded reference pictures. In the encoder, the difference between the original pixel data and the predicted pixel data, referred to as the residual, is transformed into the frequency domain, quantized and then entropy coded before transmitted together with necessary prediction parameters such as prediction mode and motion vectors, also entropy coded. The decoder performs entropy decoding, inverse quantization and inverse transformation to obtain the residual, and then adds the residual to an intra or inter prediction to reconstruct a picture.

**[0004]** The MPEG and the ITU-T are working on the successor to HEVC within the Joint Video Exploratory Team (JVET). The name of this video codec under development is Versatile Video Coding (VVC).

**[0005]** Components

**[0006]** A video sequence consists of a series of images where each image consists of one or more components. Each component can be described as a two-dimensional rectangular array of sample values. Typically, an image in a video sequence consists of three components: one luma component Y where the sample values are luma values and two chroma components Cb and Cr, where the sample values are chroma values. Typically, the dimensions of the chroma components are smaller than the luma components by a factor of two in each dimension. For example, the size of the luma component of a High Density (HD) image would be 1920x1080 and the chroma components would each have the dimension of 960x540. Chroma components are sometimes referred to as color components.

**[0007]** Blocks and units

**[0008]** A block is a two-dimensional array of samples. In video coding, each component is split into blocks and the coded video bitstream consists of a series of blocks. Typically, in video coding, the image is split into units that cover a specific area of the image. Each unit consists of all blocks from all components that make up that specific area and each block belongs to one unit. The macroblock in H.264 and the Coding Unit (CU) in High Efficiency Video Coding (HEVC) are examples of units.

**[0009]** A block can alternatively be defined as a two-dimensional array that a transform used in coding is applied to. These blocks are referred to under the name “transform blocks”. Alternatively, a block can be defined as a two-dimensional array that a single prediction mode is applied to. These blocks can be called “prediction blocks.” In this application, the term "block" is not tied to any particular one of these definitions as the description herein can apply to either definition.

**[0010]** NAL units

**[0011]** Both HEVC and VVC define a Network Abstraction Layer (NAL). All the data, i.e. both Video Coding Layer (VCL) or non-VCL data in HEVC and VVC is encapsulated in NAL units. A VCL NAL unit contains data that represents picture sample values. A non-VCL NAL unit contains additional associated data such as parameter sets and supplemental enhancement information (SEI) messages. The NAL unit in HEVC begins with a header called the NAL unit header. The syntax for the NAL unit header for HEVC is shown in Table 1 and starts with a forbidden\_zero\_bit that shall always be equal to 0 to prevent start code emulations. Without it, some MPEG systems might confuse the HEVC video bitstream with other data, but the 0 bit in the NAL unit header makes all possible HEVC bitstreams uniquely identifiable as HEVC bitstreams. The nal\_unit\_type, nuh\_layer\_id and nuh\_temporal\_id\_plus1 code words specify the NAL unit type of the NAL unit that identifies what type of data is carried in the NAL unit, the layer ID and the temporal ID for which the NAL unit belongs to. The NAL unit type indicates and specifies how the NAL unit should be parsed and decoded.

**[0012]** The rest of the bytes of the NAL unit is payload of the type indicated by the NAL unit type. A bitstream consists of a series of concatenated NAL units.

Nal_unit_header() {	Descriptor
<b>forbidden_zero_bit</b>	f(1)
<b>nal_unit_type</b>	u(6)

<b>nuh_layer_id</b>	u(6)
<b>nuh_temporal_id_plus1</b>	u(3)
}	

Table 1 – HEVC NAL unit header syntax (design A)

[0013] The syntax for the NAL unit header in the current version of the VVC draft, JVET-N1001-v8, is shown in Table 2.

Nal_unit_header() {	Descriptor
<b>zero_tid_required_flag</b>	u(1)
<b>nuh_temporal_id_plus1</b>	u(3)
<b>nal_unit_type_lsb</b>	u(4)
<b>nuh_layer_id</b>	u(7)
<b>nuh_reserved_zero_bit</b>	u(1)
}	

Table 2 – NAL unit header syntax for current version of VVC draft (design B)

[0014] At the 15<sup>th</sup> JVET meeting in Gothenburg, three different NAL unit header designs were presented and discussed. They are shown in Table 3, Table 4, and Table 5. The NAL unit header design in Table 4 was selected for the next version of the VVC draft.

Nal_unit_header() {	Descriptor
<b>nuh_layer_id</b>	u(8)
<b>nal_unit_type</b>	u(5)
<b>nuh_temporal_id_plus1</b>	u(3)
}	

Table 3 – NAL unit header syntax from JVET document JVET-O0179 (design C)

nal_unit_header() {	Descriptor
<b>forbidden_zero_bit</b>	f(1)
<b>nuh_reserved_zero_bit</b>	u(1)
<b>nuh_layer_id</b>	u(6)
<b>nal_unit_type</b>	u(5)
<b>nuh_temporal_id_plus1</b>	u(3)
}	

Table 4 – NAL unit header syntax from JVET document JVET-O0179 (design D)

nal_unit_header() {	Descriptor
<b>forbidden_zero_bit</b>	f(1)
<b>nuh_layer_id</b>	u(6)
<b>nal_unit_type</b>	u(5)
<b>nuh_temporal_id_plus1</b>	u(3)
<b>nuh_reserved_zero_bit</b>	u(1)
}	

Table 5 – NAL unit header syntax from JVET document JVET-O1037 (design E)

**[0015]** In all five NAL unit header designs presented above, design A to design E, the number of layers is limited. It may therefore be beneficial to incorporate an extension mechanism for the NAL unit header such that the number of layer ID values for the NAL units can be increased. This is provided for the NAL unit headers designs above as follows:

**[0016]** In design A, the nuh\_layer\_id value of 63 is reserved to provide a layer extension mechanism. The idea is that if the value of 63 is signaled, there is more data following the NAL unit header that contains a longer layer ID.

**[0017]** Design B is the same as in design A, but the value 127 for nuh\_layer\_id\_plus1 is reserved instead of 63.

**[0018]** Design C reserves values 127-187 that can be used to extend the number of layers.

**[0019]** Design D is similar but reserves values 32-63.

**[0020]** In Design E, it may be possible to use nuh\_reserved\_zero\_bit as an extension. The value of 1 is reserved and can be used to extend the range of layer\_id by adding additional NAL unit header extension bytes after then NAL unit header where the NAL unit header extension bytes are used to carry layer ID values.

**[0021]** A decoder or bitstream parser can conclude how the NAL unit should be handled, e.g. parsed and decoded, after looking at the NAL unit header. The rest of the bytes of the NAL unit is payload of the type indicated by the NAL unit type. A bitstream consists of a series of concatenated NAL units.

**[0022]** The NAL unit type indicates and defines how the NAL unit should be parsed and decoded. A VCL NAL unit provides information about the picture type of the current picture. The NAL unit types of the current version of the VVC draft are shown in Table 6.

<b>NalUnitType</b>	<b>Name of NalUnitType</b>	<b>Content of NAL unit and RBSP syntax structure</b>	<b>NAL unit type class</b>
0	PPS_NUT	Picture parameter set pic_parameter_set_rbsp()	non-VCL
1	AUD_NUT	Access unit delimiter access_unit_delimiter_rbsp()	non-VCL
2 3	PREFIX_SEI_N UT SUFFIX_SEI_N UT	Supplemental enhancement information sei_rbsp()	non-VCL
4	APS_NUT	Adaptation parameter set adaptation_parameter_set_rbsp()	non-VCL
5..7	RSV_NVCL5.. RSV_NVCL7	Reserved	non-VCL
8	TRAIL_NUT	Coded slice of a non-STSA trailing picture slice_layer_rbsp()	VCL
9	STSA_NUT	Coded slice of an STSA picture slice_layer_rbsp()	VCL
10	RADL_NUT	Coded slice of a RADL picture slice_layer_rbsp()	VCL
11	RASL_NUT	Coded slice of a RASL picture slice_layer_rbsp()	VCL
12..15	RSV_VCL_12.. RSV_VCL_15	Reserved non-IRAP VCL NAL unit types	VCL
16	DPS_NUT	Decoding parameter set decoding_parameter_set_rbsp()	non-VCL
17	SPS_NUT	Sequence parameter set seq_parameter_set_rbsp()	non-VCL

18	EOS_NUT	End of sequence end_of_seq_rbsp()	non-VCL
19	EOB_NUT	End of bitstream end_of_bitstream_rbsp()	non-VCL
20	VPS_NUT	Video parameter set video_parameter_set_rbsp()	non-VCL
21..23	RSV_NVCL21.. RSV_NVCL23	Reserved	non-VCL
24	IDR_W_RADL	Coded slice of an IDR picture slice_layer_rbsp()	VCL
25	IDR_N_LP		
26	CRA_NUT	Coded slice of a CRA picture slice_layer_rbsp()	VCL
27	GRA_NUT	Coded slice of a gradual random access picture slice_layer_rbsp()	VCL
28..31	UNSPEC28.. UNSPEC31	Unspecified	non-VCL

Table 6 – NAL unit types in the current version of the VVC draft

**[0023]** Temporal Layers

**[0024]** In HEVC, all pictures are associated with a TemporalId value which specified what temporal layer the picture belongs to. TemporalId values are decoded from the nuh\_temporal\_id\_plus1 syntax element in the NAL unit header. In HEVC, the encoder is required to set TemporalId values such that pictures belonging to a lower layer is perfectly decodable when higher temporal layers are discarded. Assume for instance that an encoded has output a bitstream using temporal layers 0, 1 and 2. Then removing all layer 2 NAL units or removing all layer 1 and 2 NAL units will result in bitstreams that can be decoded without problems. This is ensured by restrictions in the HEVC specification that the encoder must comply with. For instance, it is not allowed for a picture of a temporal layer to reference a picture of a higher temporal layer.

**[0025]** Layers – dependent and independent layers

**[0026]** Layers are defined in VVC as a set of VCL NAL units that all have a particular value of NuhLayerId and the associated non-VCL NAL units.



[0027] A layer access unit in VVC is defined as a set of NAL units for which the VCL NAL units all have a particular value of NuhLayerId, that are associated with each other according to a specified classification rule, that are consecutive in decoding order, and that contain exactly one coded picture.

[0028] A coded layer video sequence (CLVS) in the current version of VVC is defined as a sequence of layer access units that consists, in decoding order, of a CLVS start (CLVSS) layer access unit, followed by zero or more layer access units that are not CLVSS layer access units, including all subsequent layer access units up to but not including any subsequent layer access unit that is a CLVSS layer access unit.

[0029] The relation between the layer access units and coded layer video sequences is illustrated in figure 1.

[0030] In the current version of VVC, layers are coded independently from each other, i.e. a layer with e.g. NuhLayerId 0 may not predict video data from another layer with e.g. NuhLayerId 1. For the next version of the VVC draft, scalability support was adopted, which allows dependent coding between layers.

[0031] Access units and the access unit delimiter

[0032] For single layer coding in HEVC and the current VVC draft, an access unit (AU) is the coded representation of a single picture. An AU may consist of several video coding layer (VCL) NAL units as well as non-VCL NAL units. An access unit may optionally start with an access unit delimiter (AUD) NAL unit which indicates the start of the access unit and the type of the slices allowed in the picture, i.e. I, I-P or I-P-B. In the current version of the VVC draft, the access unit delimiter has NAL unit type 1 with the name AUD\_NUT. The tentative decision from the 15th JVET meeting in Gothenburg, is to mandate AUD for each picture, i.e. an AUD NAL unit must be sent as the first NAL unit in each access unit. The syntax and semantics for the access unit delimiter NAL unit in the current version of the VVC draft is shown below in Table 7.

access_unit_delimiter_rbsp( ) {	<b>Descriptor</b>
<b>pic_type</b>	u(3)
rbsp_trailing_bits( )	
}	

Table 7 Access unit delimiter RBSP syntax in the current version of the VVC draft

[0033] Access unit delimiter RBSP semantics

[0034] The access unit delimiter may be used to indicate the type of slices present in the coded pictures in the access unit containing the access unit delimiter NAL unit and to

simplify the detection of the boundary between access units. There is no normative decoding process associated with the access unit delimiter.

**[0035]** `pic_type` indicates that the `slice_type` values for all slices of the coded pictures in the access unit containing the access unit delimiter NAL unit are members of the set listed in Table 7-3 for the given value of `pic_type`. The value of `pic_type` shall be equal to 0, 1 or 2 in bitstreams conforming to this version of this Specification. Other values of `pic_type` are reserved for future use by ITU T | ISO/IEC. Decoders conforming to this version of this specification shall ignore reserved values of `pic_type`.

<code>pic_type</code>	<code>slice_type</code> values that may be present in the coded picture
0	I
1	P, I
2	B, P, I

Table 7-3 Interpretation of `pic_type`

**[0036]** Intra random access point (IRAP) pictures and the coded video sequence (CVS)

**[0037]** An intra random access point (IRAP) picture in HEVC is a picture that does not refer to any pictures other than itself for prediction in its decoding process. The first picture in the bitstream in decoding order in HEVC must be an IRAP picture but an IRAP picture may additionally also appear later in the bitstream. HEVC specifies three types of IRAP pictures, the broken link access (BLA) picture, the instantaneous decoder refresh (IDR) picture and the clean random access (CRA) picture

**[0038]** A coded video sequence (CVS) in HEVC is a series of access units starting at an IRAP access unit up to, but not including the next IRAP access unit in decoding order.

**[0039]** IDR pictures always start a new CVS. An IDR picture may have associated random access decodable leading (RADL) pictures. An IDR picture does not have associated RASL pictures.

**[0040]** BLA pictures also starts a new CVS and has the same effect on the decoding process as an IDR picture. However, a BLA picture in HEVC may contain syntax elements that specify a non-empty set of reference pictures. A BLA picture may have associated RASL pictures, which are not output by the decoder and may not be decodable, as they may contain references to pictures that may not be present in the bitstream. A BLA

picture may also have associated RADL pictures, which are decoded. BLA pictures are not defined in the current version of VVC.

**[0041]** A CRA picture may have associated RADL or RASL pictures. As with a BLA picture, a CRA picture may contain syntax elements that specify a non-empty set of reference pictures. For CRA pictures, a flag can be set to specify that the associated RASL pictures are not output by the decoder, because they may not be decodable, as they may contain references to pictures that are not present in the bitstream. A CRA may start a CVS.

**[0042]** In the current version of the VVC draft, a CVS is started at a CVS start (CVSS) access unit, which may contain an IRAP picture, i.e., an IDR or a CRA picture, or a gradual decoding access (GRA) picture.

**[0043]** GRA pictures are essentially used for random access in bitstreams encoded for low-delay coding where a full IRAP picture would cause too much delay. A GRA picture may use gradual intra refresh that updates the video picture by picture where each picture is only partially intra coded. It is signaled with the GRA picture when the video is fully refreshed and ready for output, given that the bitstream was tuned into at the GRA picture. A GRA may start a CVS.

**[0044]** Parameter Sets

**[0045]** HEVC and VVC specifies three types of parameter sets, the picture parameter set (PPS), the sequence parameter set (SPS) and the video parameter set (VPS). The PPS contains data that is common for a whole picture, the SPS contains data that is common for a coded video sequence (CVS) and the VPS contains data that is common for multiple CVSs.

**[0046]** The current version of VVC also specifies two additional parameter sets, the adaptation parameter set (APS) and the decoder parameter set (DPS).

**[0047]** APS carries parameters needed for the adaptive loop filter (ALF) tool and the luma mapping and chroma scaling (LMCS) tool.

**[0048]** DPS specifies information that may not change during the decoding session and may be good for the decoder to know about, e.g. the maximum number of allowed sub-layers. The information in DPS is not necessary for operation of the decoding process.

## SUMMARY

**[0049]** In contrast to the current version of VVC where it was optional to start an access unit with an AUD NAL unit, the decision from the 15th JVET meeting is to mandate

an AUD NAL unit to start each access unit in the bitstream. This adds an additional bit cost to each picture in the bitstream. In the current version of the VVC draft, the AUD NAL unit has a 2 byte NAL unit header followed by the 3 bit pic\_type code word and 5 byte alignment bits. Thus, 3 bytes extra is needed for each access unit.

**[0050]** One problem is the case where there is only one slice per picture or, more specifically only one NAL unit per picture apart from the AUD NAL unit. There may also be only one picture in the access unit. It then seems unnecessary to send an AUD NAL unit for each picture.

**[0051]** According to some embodiments of inventive concepts, a method performed by a decoder to decode a bitstream is provided. The method includes decoding one or more syntax elements from the bitstream. The method includes deriving a first value from the one or more syntax elements. The method includes comparing the first value to a predetermined set of values that comprises at least two values. The method includes determining whether the first value is within a predetermined range of values in the predetermined set of values. The method includes responsive to determining the first value is within the predetermined range of values in the predetermined set of values, determining that a location of an access unit boundary is indicated. Analogous decoder embodiments are also provided.

**[0052]** According to other embodiments of inventive concepts, a method performed by a decoder is provided. The method includes decoding a codeword from a network abstraction layer (NAL) unit header of a first NAL unit. The method includes determining whether the codeword indicates a start of an access unit by: determining that the first NAL unit is a first NAL unit of an access unit responsive to the codeword being equal to one or more specified values, and determining that the first NAL unit is not a first NAL unit of an access unit responsive to the codeword being equal to one or more different specified values. Analogous decoder embodiments are also provided.

**[0053]** According to yet other embodiments of inventive concepts, a method performed by a decoder is provided. The method includes decoding a value from a syntax element in a parameter set, the value indicating whether or not there are access unit delimiters, AUD, network abstraction layer, NAL units in the access units following the current access unit in decoding order in the coded video stream, CVS. The method includes determining whether the value indicates that there are AUD NAL units in the access units following the current access unit in decoding order in the CVS.

**[0054]** One advantage that may be provided by the inventive concepts include saving bits as the AUD does not need to be signaled in all cases.

**[0055]** According to additional embodiments of inventive concepts, a method performed by an encoder is provided. The method includes determining whether an access unit boundary is being encoded. The method includes responsive to an access unit boundary being encoded, encoding one or more syntax values in a bitstream, with a first value of the one or more syntax values indicating a location of the access unit boundary.

**[0056]** According to yet other embodiments of inventive concepts, a method performed by an encoder is provided. The method includes determining whether access unit delimiter (AUD) network abstraction layer (NAL) units are to be present in access units following a current access unit in decoding order. The method includes responsive to determining that AUD NAL units are to be present, encoding a value of a NAL unit type in a NAL unit header of a NAL unit in a video stream being encoded to indicate that AUD NAL units are to be present in access units following the current access unit in decoding order.

**[0057]** According to further embodiments of inventive concepts, a further method performed by an encoder is provided. The method includes determining whether access unit delimiter (AUD) network abstraction layer (NAL) units are to be present in access units following a current access unit in decoding order in a coded video stream (CVS). The method includes responsive to determining that AUD NAL units are to be present in access units following the current access unit in decoding order, encoding a value of a syntax element in a parameter set of the coded video stream, the value indicating that there are AUD NAL units in access units following the current access unit in decoding order. Analogous encoder embodiments are also provided.

#### BRIEF DESCRIPTION OF THE DRAWINGS

**[0058]** The accompanying drawings, which are included to provide a further understanding of the disclosure and are incorporated in and constitute a part of this application, illustrate certain non-limiting embodiments of inventive concepts. In the drawings:

**[0059]** Figure 1 is a diagram illustrating the relation between layer access units and coded layer sequences (CLVS);

**[0060]** Figure 2 is a block diagram illustrating a bitstream and the relationship between a bitstream, NAL units, NAL unit headers, NAL unit payloads, and the first part and the extension of a NAL unit header according to some embodiments of inventive concepts;

[0061] Figure 3 is a block diagram illustrating a bitstream where the NAL units for which the value of the syntax element ADU flag in the NAL unit header is set to 1 starts an access unit according to some embodiments of inventive concepts;

[0062] Figure 4 is a block diagram illustrating an example of an environment of a system in which the encoder and decoder may be implemented according to some embodiments of inventive concepts;

[0063] Figure 5 is a block diagram illustrating an encoder according to some embodiments;

[0064] Figure 6 is a block diagram illustrating a decoder according to some embodiments;

[0065] Figures 7-13 are flow charts illustrating operations of a decoder in accordance with some embodiments of inventive concepts;

[0066] Figures 14-20 are flow charts illustrating operations of an encoder in accordance with some embodiments of inventive concepts;

[0067] Figure 21 is a flow chart illustrating operations of a decoder in accordance with some embodiments of inventive concepts; and

[0068] Figure 22 is a flow chart illustrating operations of an encoder in accordance with some embodiments of inventive concepts.

#### DETAILED DESCRIPTION

[0069] Inventive concepts will now be described more fully hereinafter with reference to the accompanying drawings, in which examples of embodiments of inventive concepts are shown. Inventive concepts may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein. Rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of present inventive concepts to those skilled in the art. It should also be noted that these embodiments are not mutually exclusive. Components from one embodiment may be tacitly assumed to be present/used in another embodiment.

[0070] The following description presents various embodiments of the disclosed subject matter. These embodiments are presented as teaching examples and are not to be construed as limiting the scope of the disclosed subject matter. For example, certain details of the described embodiments may be modified, omitted, or expanded upon without departing from the scope of the described subject matter.

**[0071]** In the disclosure that follows, codewords and syntax elements may be used interchangeably and the following terminology is used:

NAL unit header:	The entire NAL unit header, which may be one or more bytes
First part of the NAL unit header:	The part of the NAL unit header that is always present
NAL unit header extension:	The part of the NAL unit header that is not always present
Access unit delimiter	An indicator that indicates the beginning or the end of an access unit

**[0072]** This terminology is further illustrated in Figure 2. A bitstream 10 comprises one or more NAL units 11. Each NAL unit consists of a NAL unit header 12 and a NAL unit payload 13. A NAL unit may be either a VCL NAL unit where the payload comprises compressed video sample data, e.g. slice data, or a non-VCL NAL unit where the payload, if present, comprises non-VCL data that may be used in the decoding process, e.g. parameter set data, or be used to inform the decoder of things not related to the decoding process, e.g. SEI messages. A NAL unit may comprise a NAL unit header (14) that is always present and a second optional part, a NAL unit header extension (15) that is not always present. In the preferred solution a flag is signaled in the NAL unit header indicating whether the NAL unit is an access unit delimiter or not.

**[0073]** Prior to describing the embodiments in further detail, Figure 4 illustrates an example of an operating environment of an encoder 400 that may be used to encode bitstreams as described herein. The encoder 400 receives video from network 402 and/or from storage 404 and encodes the video into bitstreams as described below and transmits the encoded video to decoder 406 via network 408. Storage device 404 may be part of a storage depository of multi-channel audio signals such as a storage repository of a store or a streaming video service, a separate storage component, a component of a mobile device, etc. The decoder 406 may be part of a device 410 having a media player 412. The device 410 may be a mobile device, a set-top device, a desktop computer, and the like.

**[0074]** Figure 5 is a block diagram illustrating elements of encoder 400 configured to encode video frames according to some embodiments of inventive concepts. As shown, encoder 400 may include a network interface circuit 505 (also referred to as a network

interface) configured to provide communications with other devices/entities/functions/etc. The encoder 400 may also include a processor circuit 501 (also referred to as a processor) coupled to the network interface circuit 505, and a memory circuit 503 (also referred to as memory) coupled to the processor circuit. The memory circuit 503 may include computer readable program code that when executed by the processor circuit 501 causes the processor circuit to perform operations according to embodiments disclosed herein.

**[0075]** According to other embodiments, processor circuit 501 may be defined to include memory so that a separate memory circuit is not required. As discussed herein, operations of the encoder 400 may be performed by processor 501 and/or network interface 305. For example, processor 501 may control network interface 305 to transmit communications to decoder 406 and/or to receive communications through network interface 305 from one or more other network nodes/entities/servers such as other encoder nodes, depository servers, etc. Moreover, modules may be stored in memory 503, and these modules may provide instructions so that when instructions of a module are executed by processor 501, processor 501 performs respective operations.

**[0076]** Figure 6 is a block diagram illustrating elements of decoder 306 configured to decode video frames according to some embodiments of inventive concepts. As shown, decoder 406 may include a network interface circuit 605 (also referred to as a network interface) configured to provide communications with other devices/entities/functions/etc. The decoder 406 may also include a processor circuit 601 (also referred to as a processor) coupled to the network interface circuit 605, and a memory circuit 603 (also referred to as memory) coupled to the processor circuit. The memory circuit 603 may include computer readable program code that when executed by the processor circuit 601 causes the processor circuit to perform operations according to embodiments disclosed herein.

**[0077]** According to other embodiments, processor circuit 601 may be defined to include memory so that a separate memory circuit is not required. As discussed herein, operations of the decoder 406 may be performed by processor 601 and/or network interface 605. For example, processor 601 may control network interface 605 to receive communications from encoder 400. Moreover, modules may be stored in memory 603, and these modules may provide instructions so that when instructions of a module are executed by processor 601, processor 601 performs respective operations.

**[0078]** In contrast to the current version of VVC where it was optional to start an access unit with an AUD NAL unit, the decision from the 15th JVET meeting is to mandate an AUD NAL unit to start each access unit in the bitstream. This adds an additional bit cost to



each picture in the bitstream. In the current version of the VVC draft, the AUD NAL unit has a 2 byte NAL unit header followed by the 3 bit pic\_type code word and 5 byte alignment bits. Thus, 3 bytes extra is needed for each access unit.

**[0079]** Another problem is the case where there is only one slice per picture or, more specifically only one NAL unit per picture apart from the AUD NAL unit. There may also be only one picture in the access unit. It then seems unnecessary to send an AUD NAL unit for each picture.

**[0080]** The embodiments described herein describe methods of signaling the AUD that address the problems of the current solutions. It is to be understood by a person skilled in the art that two or more embodiments, or parts of embodiments, may be combined to form new solutions which are still covered by the invention described in this disclosure.

**[0081]** In a first embodiment the AUD may be signaled in the NAL unit header, e.g. as a flag. If the flag is set to 1 then the NAL unit is the first NAL unit of the access unit. If the flag is set to 0, then the NAL unit is not the first NAL unit header of the access unit.

**[0082]** In the first embodiment, the AUD may be signaled directly in the NAL unit header and the NAL unit has a type not solely defined for indicating an AUD NAL unit type with one codeword. If the codeword has one value, the NAL unit is the first NAL unit of the access unit. If the codeword has another value, the NAL unit is not the first NAL unit of the access unit. This embodiment is in contrast with the case where one NAL unit type is dedicated to indicate the start of an access unit in the sense that several NAL unit types can indicate the start of one access unit as long as the codeword is signaled in their NAL unit headers.

**[0083]** Example syntax for the NAL unit header comprising an AUD codeword is shown below.

nal_unit_header() {	Descriptor
...	
access_unit_delimiter	u(n)
...	
}	

**[0084]** *Access\_unit\_delimiter* equal to 1 specifies that the NAL unit is the first NAL unit of the access unit.

**[0085]** In a preferred version of this embodiment, the codeword is a flag. For instance, if the flag is equal to 1, then the NAL unit is the first NAL unit of the access unit. If

the flag is equal to 0, then the NAL unit is not the first NAL unit header of the access unit. Alternatively, a flag value of 0 means that the NAL unit is the first NAL unit of the access unit and a flag value of 1 means that it is not.

**[0086]** Example syntax and semantics for the NAL unit header comprising an AUD flag is shown below.

nal_unit_header() {	Descriptor
...	
<b>access_unit_delimiter_flag</b>	u(1)
...	
}	

**[0087]** *Access\_unit\_delimiter\_flag* equal to 1 specifies that the NAL unit is the first NAL unit of the access unit. The *access\_unit\_delimiter* flag equal to 0 specifies that the NAL unit is not the first NAL unit of the access unit. Alternatively, the *access\_unit\_delimiter* flag equal to 1 specifies that the NAL unit is not the first NAL unit of the access unit and the *access\_unit\_delimiter* equal to 0 specifies that the NAL unit is the first NAL unit of the access unit.

**[0088]** By signaling AUD directly in the NAL unit header by using a codeword in the NAL unit header where if the codeword has one or more specific values, the NAL unit is the first NAL unit of the access unit and if the codeword has one or more another values, the NAL unit is not the first NAL unit of the access unit, sending a separate NAL unit for the AUD is not needed.

**[0089]** Figure 3 illustrates an example of a bitstream where the NAL units for which the value of the syntax element *access\_unit\_delimiter\_flag* in their NAL unit header is equal to 1 starts an access unit and the NAL units that for which the value of the syntax element *access\_unit\_delimiter\_flag* is equal to 0 in their NAL unit headers are not starting a new access unit.

**[0090]** A decoder may perform all or a subset of the following steps for this embodiment:

- Decode a first NAL unit in a CVS
- Decode a codeword (e.g. a flag) from the NAL unit header of the first NAL unit
- Determine whether the codeword indicates a start of an access unit or not, wherein the decoder determines that the first NAL unit is a first NAL unit of

an access unit if the codeword is equal to one or more specific value(s) and the decoder determines that the first NAL unit is not the first NAL unit of an access unit if the codeword is equal to one or more other specific value(s)

**[0091]** An encoder may perform all or a subset of the following steps to encode a current access unit containing one or more pictures according to this embodiment:

- Encode and output all NAL units for the current access unit, wherein a codeword in the NAL unit header (e.g. a flag) of the first NAL unit of the current access unit is set to a value that indicates that the NAL unit is the first NAL unit of the access unit

**[0092]** The codewords (e.g. flags) in all NAL unit headers of all NAL units of the current access unit that are not the first NAL unit of the access unit are set to a value that indicates that the NAL units are not the first NAL unit of the access unit

**[0093]** In another version of the embodiment, the AUD codeword in the NAL unit header determines if the NAL unit is the first NAL unit of the access unit of the current layer. For instance, if the codeword is a flag and the flag is equal to 1 then the NAL unit is the first NAL unit of the access unit of the current layer. An access unit for a layer may be referred to as a layer access unit. If the flag is equal to 0, then the NAL unit is not the first NAL unit of the access unit of the current layer.

**[0094]** In other versions of this embodiment the AUD codeword in the NAL unit header determines one or more features of the access unit delimiter. The codeword could for instance indicate one or more of the following features:

1. The start of the access unit
2. The end of the access unit
3. The start of a CVS
4. The end of a CVS
5. The first VCL NAL unit in the access unit
6. The last VCL NAL unit in the access unit
7. The start of the access unit of the current layer
8. The end of the access unit of the current layer
9. The number of NAL units in the access unit
10. The number of VCL NAL units in the access unit
11. The number of non-VCL NAL units in the access unit

**[0095]** An AUD may be seen as a picture header comprising information that is valid for the whole picture. In one version of the embodiment, the NAL unit header could be

signaled with more than two bytes, e. g. three bytes. Applying the proposed method in a three byte NAL unit header scenario compared to the case the NAL unit header is signaled using two bytes but using other than the proposed method such as methods A to E (see Tables 1-5) would then save 1 byte per picture when there is one slice per picture, i.e. it would cost 3 bytes to signal the NAL unit header carrying both the single slice and the AUD codeword with the defined value that specifies that the NAL unit starts an access unit instead of 2 bytes for the NAL unit header carrying the AUD and another 2 bytes for the NAL unit header carrying the single slice. By signaling the NAL unit header with three bytes, it would be possible to fit more information in the NAL unit header. In this example the signaling overhead is reduced (by removing the dedicated AUD NAL unit which reduces the number of NAL units) although the length of the NAL unit header is increased.

**[0096]** A larger NAL unit header may for instance comprise one or more of the following:

1. More values for representing NAL unit types
2. More values for representing layer ID
3. More values for representing temporal ID
4. The start of the access unit
5. The end of the access unit
6. The start of a CVS
7. The end of a CVS
8. The first VCL NAL unit in the access unit
9. The last VCL NAL unit in the access unit
10. The start of the access unit of the current layer
11. The end of the access unit of the current layer
12. The number of NAL units in the access unit
13. The number of VCL NAL units in the access unit
14. The number of non-VCL NAL units in the access unit

**[0097]** Figure 7 illustrates operations of the decoder 406 in this embodiment. In operation 700, the processing circuitry 601 may decode a first network abstraction layer, NAL, unit in a coded video stream, CVS.

**[0098]** In operation 702, the processing circuitry 601 may decode a codeword from a NAL unit header of the first NAL unit. The codeword may be a flag. The codeword in an embodiment may determine one or more features of the access unit delimiter. The one or more features may be one or more of a start of the access unit, an end of the access unit, a

start of a CVS, an end of a CVS, a first VCL NAL unit in the access unit, a last VCL NAL unit in the access unit, a start of the access unit of the current layer, an end of the access unit of the current layer, a number of NAL units in the access unit, a number of VCL NAL units in the access unit, or a number of non-VCL NAL units in the access unit.

**[0099]** In operation 704, the processing circuitry 601 may determine whether the code word indicates a start of an access unit by: determining in operation 708 that the first NAL unit is a first NAL unit of an access unit responsive to the codeword being equal to one or more specified values and responsive to the codeword being equal to one or more other specified values, the processing circuitry 601 in operation 710 may determine that the first NAL unit is not the first NAL unit of an access unit.

**[0100]** Turning to Figure 14, operations of encoder 400 in an exemplary embodiment are illustrated. In operation 1400, the processing circuitry 501 may determine whether the network abstraction layer, NAL, unit is a first NAL unit of an access unit. In operation 1402, responsive to determining the NAL unit is the first NAL unit of the access unit, the processing circuitry 501 may encode a codeword in the NAL unit header indicating the NAL unit is the first NAL unit of the access unit. In operation 1404, responsive to determining the NAL unit is not the first NAL unit of the access unit, the processing circuitry 501 may encode a different codeword in the NAL unit header indicating the NAL unit head is not the first NAL unit of the access unit.

**[0101]** In a second embodiment the AUD information may be signaled in the NAL unit payload. In one example of this second embodiment, a first value may be derived from one or more codewords in the payload of the NAL unit such that if the first value is equal to one value or equal to a value in a set of values, the NAL unit is the first NAL unit of the access unit and if the first value is equal to another one value or equal to a value in another set of values, the NAL unit is not the first NAL unit of the access unit. The one or more codewords in the payload may be for example a single flag and the first value may be the value of the flag.

**[0102]** In an example of this embodiment, a fixed position flag in the payload of the NAL unit is used and if the fixed position flag is equal to 1 then the NAL unit starts an access unit and if the fixed position flag is equal to 0 then the NAL unit does not start an access unit in the bitstream.

**[0103]** In a variant of this embodiment, a second fixed position flag is present in the payload of all the NAL units except in the AU delimiter NAL unit and if this second flag

is equal to 1, then the NAL unit starts an access unit and if that flag is equal to 0 then the NAL unit does not start an access unit in the bitstream.

**[0104]** In yet another variation of this embodiment, the presence of the one or more codewords in the payload of the NAL unit is conditioned on the value of one or more codewords in other parts of the bit stream such as a flag in the SPS.

**[0105]** Figure 8 illustrates operations of the decoder 406 in this embodiment. In operation 800, the processing circuitry 601 may derive a first value from one or more codewords in a payload of a network abstraction layer, NAL, unit of a bitstream. The one or more codewords may be a flag and the first value may be a value of the flag. The presence of the one or more codewords in the payload of the NAL unit may be conditioned on a value of one or more codewords in at least one other part of the bit stream

**[0106]** Responsive to determining the first value is equal to a value in a set of values, the processing circuitry 601 in operation 802 may determine that the NAL unit is a first NAL unit of an access unit.

**[0107]** Responsive to determining the first value is not equal to the value in the set of values, the processing circuitry 601 in operation 804 may determine that the NAL unit is not the first NAL unit of the access unit.

**[0108]** Figure 15 illustrates operations of the encoder 400 in this embodiment. The processing circuitry 501 may determine in operation 1500 whether a network abstraction layer, NAL, unit is to be a first NAL unit of an access unit.

**[0109]** In operation 1502, the processing circuitry 501 may, responsive to determining the NAL unit is to be the first NAL unit, encode one or more codewords in a NAL unit header of the NAL unit to have a first value equal to a value in a set of values. The one or more codewords may be a flag and the value of the flag is the first value.

**[0110]** In operation 1504, the processing circuitry 501 may, responsive to determining the NAL unit is not to be the first NAL unit, encode the one or more codewords in the NAL unit header to have a first value not equal to the value in the set of values.

**[0111]** In a third embodiment, the presence of AUD NAL units in the rest of the CVS is signaled in the AUD NAL unit of the first access unit. In one version of this embodiment, the first NAL unit in the CVS is always an AUD NAL unit and a codeword, e.g. a flag, is used to determine if the following access units start with an AUD NAL unit or not.

**[0112]** In a preferred version the flag may only be set if each of the following access units in decoding order in the CVS only comprises one NAL unit. In another preferred version the flag is only present in the first NAL unit of the CVS. In another version the flag is

required to have a specific value if the NAL unit is not the first NAL unit of the CVS. In another version the flag may be set on or off for any AUD NAL unit in the CVS. In this case, the indication specifies the presence of access unit delimiters in the following NAL units in the decoding order in the CVS until the end of the CVS or a new AUD NAL unit is decoded with a new flag overriding the previous value of the flag.

**[0113]** Below is example syntax and semantics for this third embodiment.

access_unit_delimiter_rbsp() {	Descriptor
pic_type	u(3)
presence_of_access_unit_delimiters_in_following_access_units_in_cvs_flag	u(1)
rbp_trailing_bits()	
}	

**[0114]** The presence\_of\_access\_unit\_delimiters\_in\_following\_access\_units\_in\_cvs\_flag equal to 1 indicates that an access unit delimiter NAL unit is present for all access units in the CVS following the current access unit in decoding order. The presence\_of\_access\_unit\_delimiters\_in\_following\_access\_units\_in\_cvs\_flag equal to 0 indicates that an access unit delimiter NAL unit may or may not be present for access units in the CVS following the current access unit in decoding order.

**[0115]** One or several of the following restrictions may apply:

- It is a requirement for bitstream conformance that the first access unit in a CVS starts with an access unit delimiter NAL unit.
- An access unit comprising more than one NAL unit must start with an access unit delimiter.

**[0116]** A decoder may perform all or a subset of the following steps:

1. Decode a first NAL unit of a CVS, where the first NAL unit is an AUD NAL unit
2. Decode a codeword (e.g. a flag) from the first NAL unit
3. Determine an indication from the codeword that specifies whether access unit delimiters are present in the following access units in decoding order in the CVS or not
4. If the indication indicates that every access unit shall start with an AUD NAL unit
  - a. Determine the start of each access unit based on the NAL unit type, where a NAL unit type of AUD\_NUT determines that the NAL unit is the first NAL unit in the access unit

5. else if the indication indicates that every access unit may or may not start with an AUD NAL unit determine the start of each access unit by other means

**[0117]** Figure 9 illustrates operations of the decoder 406. In operation 900, the processing circuitry 601 may decode a codeword from a network abstraction layer, NAL, unit header of a first NAL unit of a coded video stream, CVS.

**[0118]** In operation 902, the processing circuitry 601 may determine whether an indication from the codeword specifies that access unit delimiters (AUDs) starts in every following access unit in decoding order in the CVS.

**[0119]** In operation 904, the processing circuitry 601 may, responsive to the indication indicating that every access unit start with an AUD NAL unit, determine a start of each access unit based on a NAL unit type, where a NAL unit type of AUD\_NUT (access unit delimiter NAL unit type) determines that the NAL unit is a first NAL unit in the access unit.

**[0120]** Figure 15 illustrates operations of the encoder 400. In operation 1500, the processing circuitry 501 may determine if every access unit starts with an access unit delimiter (AUD) NAL unit. Responsive to determining that every access unit starts with an AUD NAL unit, the processing circuitry 501 in operation 1502 may encode a codeword in a first NAL unit of a coded video stream that specifies that AUDs are present in every access unit in decoding order.

**[0121]** In a fourth embodiment, the presence of AUD NAL units in the following access units in decoding order in the rest of the CVS is indicated with a codeword in a parameter set. In the preferred version, the parameter set is SPS. In other versions the parameter set is a picture parameter set (PPS), an adaptive parameter set (APS), a decoder parameter set (DPS), or a picture header.

**[0122]** In this fourth embodiment, syntax is added to the parameter set that indicates whether there are AUD NAL units in the access units following the current access unit in decoding order in the CVS, or not. In one version of this embodiment the SPS NAL unit is the first NAL unit in the CVS. A second syntax may be present in the parameter set that indicates the number of NAL units in the first access unit. In another version of the embodiment, the AUD is the first NAL unit in the first access unit. In yet another version, an AUD NAL unit is the last NAL unit of the access unit. Below is example syntax for embodiment:



	Descriptor
seq_parameter_set( ) {	
...	
<b>access_unit_delimiters_is_present_in_following_access_units_in_cvs_flag</b>	u(1)
if (!access_unit_delimiters_is_present_in_following_access_units_in_cvs_flag )	
number_of_nal_units_in_current_access_unit	ue(v)
}	

**[0123]** In another version of the embodiment, the presence of AUD NAL units in the access units following the current access unit in decoding order in the CVS is indicated with a specific value of NAL unit type in a NAL unit header in a NAL unit, preferably the first NAL unit in the CVS.

**[0124]** For instance, if SPS is the first NAL unit in the CVS, one of two or more values of NAL unit type is used to express that the NAL unit is the NAL unit header where the two or more NAL unit types for SPS express whether the access units following the current access unit in decoding order in the CVS starts with an AUD NAL unit or not.

Example NAL unit types for SPS:

SPS\_AUD\_NALU\_IN\_ALL\_FOLLOWING\_AUS\_IN\_CVS\_NUT

SPS\_AUD\_NALU\_OPTIONAL\_IN\_CVS\_NUT

**[0125]** Figure 10 illustrates operations of the decoder 406. In operation 1000, the processing circuitry 601 may decode an value of a network abstraction layer, NAL unit type in a NAL unit header in a NAL unit. In operation 1002, the processing circuitry 601 may determine whether the value indicates that access unit delimiter (AUD) NAL units are present in access units following a current access unit in decoding order in the CVS.

**[0126]** Figure 17 illustrates operations of the encoder 400. In operation 1700, the encoder determines whether access unit delimiter, AUD, network abstraction layer, NAL, units are to be present in access units following a current access unit in decoding order.

**[0127]** In operation 1702, responsive to determining that AUD NAL units are to be present, the processing circuitry 501 encodes a value of a NAL unit type in a NAL unit header of a NAL unit in the video stream being encoded to indicate that AUD NAL units are to be present in access units following the current access unit in decoding order.

**[0128]** Figure 21 illustrates operations of the decoder 406 when a syntax element is added to a parameter set. In operation 2100, the processing circuitry 601 decodes a value of a syntax element in a parameter set, the value indicating whether there are AUD NAL units in access units following a current access unit in decoding order in a CVS. In operation 2102,

the processing circuitry 601 determines whether the value indicates that there are AUD NAL units in the access units following the current access unit in decoding order in the CVS. For example, the processing circuitry 601 can determine whether there are AUD NAL units in the access units following the current access unit in decoding order in the CVS based on a specific value of NAL unit type. In operation 2104, the processing circuitry 601 can determine from a syntax element in the parameter set a number of NAL units in a first access unit. Various operations from the flow chart of Figure 21 may be optional with respect to some embodiments of decoders and related methods. Regarding claims 9 and 22 (set forth below), for example, operations of block 2104 of Figure 21 may be optional.

**[0129]** Figure 22 illustrates operations of the encoder 400 when adding the syntax element to a parameter set. In operation 2200, the processing circuitry 501 determines whether access unit delimiter (AUD) network abstraction layer (NAL) units are to be present in access units following a current access unit in decoding order in a coded video stream (CVS). In operation 2202, the processing circuitry 501, responsive to determining that AUD NAL units are to be present in access units following the current access unit in decoding order, encodes a value of a syntax element in a parameter set of the coded video stream, the value indicating that there are AUD NAL units in access units following the current access unit in decoding order. For example, the processing circuitry 501 may, responsive to determining that AUD NAL units are to be present in access units following the current access unit in decoding order, encode a specific value of NAL unit type in a NAL unit header. In operation 2204, the processing circuitry 501 may encode a value of a second syntax element in the parameter set, the value of the second syntax element indicating a number of NAL units in a first access unit. Various operations from the flow chart of Figure 22 may be optional with respect to some embodiments of encoders and related methods. Regarding claims 29 and 37 (set forth below), for example, operations of block 2204 of Figure 22 may be optional.

**[0130]** In a fifth embodiment the AUD may be signaled in the NAL unit type codeword in the NAL unit header.

**[0131]** In one version of this embodiment, two NAL unit types are used for each current NAL unit type that may start an AU. For instance:

SPS\_AUD\_NUT

SPS\_NO\_AUD\_NUT

PPS\_AUD\_NUT

PPS\_NO\_AUD\_NUT

TRAIL\_AUD\_NUT

TRAIL\_NO\_AUD\_NUT

**[0132]** Any of the NAL unit types in Table 6 may be used to indicate AUD\_NUT or NO\_AUD\_NUT except for the NAL unit types 2 and 3 of Table 6. One of each of the two NAL\_unit\_type for each current NAL unit type indicates that the NAL unit type is an AUD. The other one of each of the two NAL\_unit\_type for each current NAL unit type indicates that the NAL unit type is not an AUD.

**[0133]** In a sixth embodiment, a set of values may be used to signal the AUD. In this embodiment, a decoder 406 derives the start of an access unit from one or more syntax elements in a video bitstream. First, the one or more syntax elements are decoded and a first value is derived from the decoded one or more syntax elements. Then, the first value is compared against the values of a predetermined set of values where the set consist of at least two values. If the first value is equal to one of the values in the predetermined set of values, the decoder determines that an access unit boundary is indicated.

**[0134]** In one version of this embodiment, the comparison of the first value against a predetermined set of values may consist of determining whether the first value is within a predetermined range of values. This determination may be done by one or both of the following:

- Determining whether the first value is larger than or equal to a first predetermined threshold value
- Determining whether the first value is smaller than or equal to a second predetermined threshold value

**[0135]** The one or more syntax elements may be syntax elements in a NAL unit header.

**[0136]** Figure 12 illustrates operation of decoder 406 in this sixth embodiment. In operation 1200, the processing circuitry 601 may decode one or more syntax elements from a received bitstream. The one or more syntax elements may be syntax elements in a NAL unit header. In operation 1202, the processing circuitry 601 may derive a first value from the one or more syntax elements decoded.

**[0137]** In operation 1204, the processing circuitry 601 may compare the first value to a predetermined set of values that comprises at least two values. The comparison of

the first value against a predetermined set of values may consist of the processing circuitry 601 determining whether the first value is within a predetermined range of values.

**[0138]** In operation 1206, the processing circuitry 601 may determine whether the first value is within a predetermined range of values in the predetermined set of values. In determining the first value is within the predetermined range of values, the processing circuitry 601 in one embodiment may determine whether the first value is larger than or equal to a first predetermined threshold value. In another embodiment, the processing circuitry may determine whether the first value is smaller than or equal to a second predetermined threshold value.

**[0139]** In operation 1208, responsive to determining the first value is within the predetermined range of values in the predetermined set of values, the processing circuitry 601 may determine that a location of an access unit boundary is indicated.

**[0140]** Turning to Figure 19, operations of encoder 400 in an exemplary embodiment are illustrated. In operation 1900, the processing circuitry 501 may determine whether an access unit boundary is being encoded. In operation 1902, responsive to an access unit boundary is being encoded, the processing circuitry 501 may encode one or more syntax values in the bitstream, with a first value of the one or more syntax values indicating a location of the access unit boundary.

**[0141]** In a seventh embodiment, the AU delimiter NAL unit design may consist of signaling a NAL unit type in 1-byte NAL unit header. In this embodiment, the AU delimiter NAL unit design may consist of signaling a NAL unit type in a first part of the NAL unit header. If the NAL unit type is not an AU delimiter type, a NAL unit header extension is present.

**[0142]** One example of such a design on top of the HEVC NAL unit header design is shown in Table 8

	<b>Descriptor</b>
nal_unit_header() {	
<b>forbidden_zero_bit</b>	f(1)
<b>nal_unit_type</b>	u(6)
<b>nuh_layer_id_first_bit</b>	u(1)
if( nal_unit_type != AUD ) {	
<b>nuh_layer_id</b>	u(5)
<b>nuh_temporal_id_plus1</b>	u(3)
}	
}	

Table 8 – A 1-byte NAL unit header for AUD

**[0143]** A decoder may perform all of a subset of the following steps in this embodiment:

- derive a first value from one or more syntax element in a first part of a NAL unit header
- If the first value is equal to a value in a predetermined set of values, determine that a NAL unit header extension is not present and determine that the NAL unit containing the NAL unit header is the first NAL unit in an access unit
- If the first value is not equal to a value in a predetermined set of values, determine that a NAL unit header extension is present, decode the syntax element in the NAL unit header extension, and determine that the NAL unit containing the NAL unit header is not the first NAL unit in an access unit.

**[0144]** The predetermined set of values may in this embodiment be a set consisting of only one value.

**[0145]** Turning to Figure 13, operations of decoder 406 in this seventh embodiment are illustrated. In operation 1300, the processing circuitry 601 may derive a first value from one or more syntax elements in a first part of a network abstraction layer, NAL, unit header. In operation 1302, the processing circuitry 601 may determine whether the first value is equal to a value in a predetermined set of values.

**[0146]** Responsive to determining that the first value is equal to the value in the predetermined set of values, the processing circuitry 601 may determine in operation 1306 that a NAL unit header extension is not present and determine in operation 1308 that a NAL unit containing the NAL unit header is a first NAL unit in an access unit.

**[0147]** Responsive to determining that the first value is not equal to the value in the predetermined set of values, the processing circuitry 601 may determine in operation 1310 that a NAL unit header extension is present, determine in operation 1312 that a syntax element is in the NAL unit header extension, and determine in operation 1314 that the NAL unit containing the NAL unit header is not the first NAL unit in an access unit.

**[0148]** Turning to Figure 20, operations of encoder 400 in an exemplary embodiment are illustrated. In operation 2000, the processing circuitry 501 can determine whether a network abstraction layer, NAL, unit containing a NAL unit header is a first NAL unit in an access unit. In operation 2002, responsive to determining the NAL unit containing the NAL unit header is the first NAL unit in the access unit, the processing circuitry 501 may encode one or more syntax elements with a first value equal to a value in a predetermined set of values in a first part of the NAL unit header. In operation 2004, responsive to determining

the NAL unit containing the NAL unit header is not the first NAL unit in the access unit, the processing circuitry 501 may encode one or more syntax elements with a first value not equal to the value in a predetermined set of values in the first part of the NAL unit header.

**[0149]** Listing of embodiments:

Embodiment 1. A method performed by a decoder to decode a bitstream, the method comprising:

decoding (1000) one or more syntax elements from the bitstream;

deriving (1002) a first value from the one or more syntax elements;

comparing (1004) the first value to a predetermined set of values that comprises at least two values;

determining (1006) whether the first value is within a predetermined range of values in the predetermined set of values; and

responsive to determining the first value is within the predetermined range of values in the predetermined set of values, determining (1008) that a location of an access unit boundary is indicated.

Embodiment 2. The method of Embodiment 1 wherein determining whether the first value is within the predetermined range of values in the predetermined set of values comprises:

determining whether the first value is equal to a value in the predetermined range of values.

Embodiment 3. The method of any of Embodiments 1-2 wherein determining whether the first value is within the predetermined range of values comprises determining whether the first value is larger than or equal to a first predetermined threshold value.

Embodiment 4. The method of any of Embodiments 1-2 wherein determining whether the first value is within the predetermined range of values comprises determining whether the first value is smaller than or equal to a second predetermined threshold value.

Embodiment 5. A method performed by a decoder, the method comprising:  
deriving (1300) a first value from one or more syntax elements in a first part of a network abstraction layer, NAL, unit header;

determining (1302) whether the first value is equal to a value in a predetermined set of values;

responsive to determining (1304) the first value is equal to the value in the predetermined set of values:

determining (1306) that a NAL unit header extension is not present; and

determining (1308) that a NAL unit containing the NAL unit header is a first NAL unit in an access unit.

Embodiment 6. The method of Embodiment 5, further comprising:

responsive to determining (1304) the first value is not equal to the value in the predetermined set of values:

determining (1310) that a NAL unit header extension is present;

decoding (1312) a syntax element in the NAL unit header extension; and

determining (1314) that the NAL unit containing the NAL unit header is not the first NAL unit in an access unit.

Embodiment 7. A method performed by a decoder, the method comprising:

decoding (700) a codeword from a network abstraction layer, NAL, unit header of a first NAL unit; and

determining (704) whether the code word indicates a start of an access unit by:

determining (706) that the first NAL unit is a first NAL unit of an access unit responsive to the codeword being equal to one or more specified values; and

determining (708) that the first NAL unit is not a first NAL unit of an access unit responsive to the codeword being equal to one or more different specified values.

Embodiment 8. The method of Embodiment 7 wherein the codeword comprises a flag.

Embodiment 9. The method of any of Embodiments 7-8 wherein the codeword indicates one or more features of an access unit delimiter.

Embodiment 10. The method of Embodiment 9 where the one or more features may be one or more of a start of the access unit, an end of the access unit, a start of a CVS, an end of a CVS, a first VCL NAL unit in the access unit, a last VCL NAL unit in the access unit, a start of the access unit of the current layer, an end of the access unit of the current

layer, a number of NAL units in the access unit, a number of VCL NAL units in the access unit, or a number of non-VCL NAL units in the access unit.

Embodiment 11. A method performed by a decoder, the method comprising:  
deriving (800) a first value from one or more codewords in a payload of a network abstraction layer, NAL, unit of a bitstream;

responsive to determining the first value is equal to a value in a set of values,  
determining (802) that the NAL unit is a first NAL unit of an access unit; and

responsive to determining the first value is not equal to the value in the set of values,  
determining (804) that the NAL unit is not the first NAL unit of the access unit.

Embodiment 12. The method of Embodiment 11 wherein the one or more codewords comprises a flag and a value of the flag comprises the first value.

Embodiment 13. The method of any of Embodiments 11-12 wherein a presence of the one or more codewords in the payload of the NAL unit is conditioned on a value of one or more codewords in at least one other part of the bit stream.

Embodiment 14. A method performed by a decoder, the method comprising:  
decoding (900) a codeword from a network abstraction layer, NAL, unit header of a first NAL unit of a coded video stream, CVS;

determining (902) whether an indication from the codeword specifies that access unit delimiters (AUDs) start in every following access unit in decoding order in the CVS; and

responsive to the indication indicating that every access unit start with an AUD NAL unit, determining (904) a start of each access unit based on a NAL unit type, where a NAL unit type of access unit delimiter NAL unit type, AUD\_NUT, determines that the NAL unit is a first NAL unit in the access unit.

Embodiment 15. The method of Embodiment 14 wherein the codeword comprises a flag.

Embodiment 16. A method performed by a decoder, the method comprising:  
decoding (1000) a value of a network abstraction layer, NAL unit type in a NAL unit header in a NAL unit of a coded video stream, CVS; and

determining (1002) whether the value indicates that access unit delimiter (AUD) NAL units are present in access units following a current access unit in decoding order in the CVS.



Embodiment 17. A method performed by a decoder, the method comprising:  
decoding (1100) a codeword value of a network abstraction layer, NAL unit type in a NAL unit header in a NAL unit; and  
determining (1102) whether the value indicates that the NAL unit type is an access unit delimiter, AUD, or is not an AUD.

Embodiment 18. A decoder for a communication network, the decoder (406) comprising:  
a processor (601); and  
memory (603) coupled with the processor, wherein the memory comprises instructions that when executed by the processor cause the processor to perform operations according to any of Embodiments 1-17.

Embodiment 19. A computer program comprising computer-executable instructions configured to cause a device to perform the method according to any one of Embodiments 1-17, when the computer-executable instructions are executed on a processor (601) comprised in the device.

Embodiment 20. A computer program product comprising a computer-readable storage medium (603), the computer-readable storage medium having computer-executable instructions configured to cause a device to perform the method according to any one of Embodiments 1-17 when the computer-executable instructions are executed on a processor (601) comprised in the device.

Embodiment 21. An apparatus comprising:  
at least one processor (601);  
memory (603) communicatively coupled to the processor, said memory comprising instructions executable by the processor, which cause the processor to perform operations comprising operations according to any of Embodiments 1-17.

Embodiment 22. A method performed by an encoder, the method comprising:  
determining (1700) whether an access unit boundary is being encoded; and  
responsive to an access unit boundary being encoded, encoding (1702) one or more syntax values in the bitstream, with a first value of the one or more syntax values indicating a location of the access unit boundary.

Embodiment 23. A method performed by an encoder, the method comprising:  
determining (2000) whether a network abstraction layer, NAL, unit containing a NAL unit header is a first NAL unit in an access unit;

responsive to determining (2002) the NAL unit containing the NAL unit header is the first NAL unit in the access unit, encoding (2004) one or more syntax elements with a first value equal to a value in a predetermined set of values in a first part of the NAL unit header;  
and

responsive to determining (2002) the NAL unit containing the NAL unit header is not the first NAL unit in the access unit, encoding (2006) one or more syntax elements with the first value not equal to the value in a predetermined set of values in the first part of the NAL unit header and encoding syntax elements in a NAL unit header extension.

Embodiment 24. The method of Embodiment 23 wherein the predetermined set of values comprises only one value.

Embodiment 25. A method performed by an encoder, the method comprising:  
determining (1400) whether the network abstraction layer, NAL, unit is a first NAL unit of an access unit;

responsive to determining the NAL unit is the first NAL unit of the access unit, encoding (1402) one or more codewords in a payload of the NAL unit header to have a value first indicating the NAL unit is the first NAL unit of the access unit; and

responsive to determining the NAL unit is not the first NAL unit of the access unit, encoding (1404) a different codeword in the NAL unit header indicating the NAL unit head is not the first NAL unit of the access unit.

Embodiment 26. A method performed by an encoder, the method comprising:  
determining (1500) whether a network abstraction layer, NAL, unit is to be a first NAL unit of an access unit;

responsive to determining the NAL unit is to be the first NAL unit, encoding (1502) one or more codewords in a NAL unit header of the NAL unit to have a first value equal to a value in a set of values; and

responsive to determining the NAL unit is not to be the first NAL unit, encoding (1504) the one or more codewords in the NAL unit header to have a first value not equal to the value in the set of values.

Embodiment 27. The method of Embodiment 16 wherein the one or more codewords comprises a flag and the first value is a value of the flag.

Embodiment 28. A method performed by an encoder, the method comprising: determining (1600) whether every access unit starts with an access unit delimiter, AUD, network abstract layer, NAL, unit; and

responsive to determining that every access unit starts with an AUD NAL unit, encoding (1602) a codeword in a first NAL unit of a coded video stream that specifies that AUDs are present in every access unit in decoding order.

Embodiment 29. A method performed by an encoder, the method comprising: determining (1700) whether access unit delimiter, AUD, network abstraction layer, NAL, units are to be present in access units following a current access unit in decoding order; and

responsive to determining that AUD NAL units are to be present, encoding (1702) a value of a NAL unit type in a NAL unit header of a NAL unit in the video stream being encoded to indicate that AUD NAL units are to be present in access units following the current access unit in decoding order.

Embodiment 30. A method performed by an encoder, the method comprising: determining (1800) whether a network abstract layer, NAL, unit type of a NAL unit is to be an access unit delimiter, AUD, NAL, unit type; and

responsive to determining that the NAL unit type is to be an AUD NAL unit, encoding (1802) a codeword in a first NAL unit of a coded video stream that specifies that the NAL unit type is an AUD NAL unit type.

Embodiment 31. An encoder for a communication network, the encoder (200) comprising:

a processor (501); and

memory (503) coupled with the processor (501), wherein the memory (503) comprises instructions that when executed by the processor (501) cause the processor (501) to perform operations according to any of Embodiments 22-30.

Embodiment 32. A computer program comprising computer-executable instructions configured to cause a device to perform the method according to any one of

Embodiments 22-30, when the computer-executable instructions are executed on a processor (501) comprised in the device.

Embodiment 33. A computer program product comprising a computer-readable storage medium (503), the computer-readable storage medium having computer-executable instructions configured to cause a device to perform the method according to any one of Embodiments 22-30 when the computer-executable instructions are executed on a processor (501) comprised in the device.

Embodiment 34. An apparatus comprising:  
 at least one processor (501);  
 memory (503) communicatively coupled to the processor, said memory comprising instructions executable by the processor, which cause the processor to perform operations comprising operations according to any of Embodiments 22-30.

**[0150]** Explanations for abbreviations from the above disclosure are provided below.

Abbreviation	Explanation
AU	Access Unit
AUD	Access Unit Delimiter
APS	Adaptive Parameter Set
CVS	Coded Video Stream
DPS	Decoder Parameter Set
HEVC	High Efficiency Video Coding
PPS	Picture Parameter Set
NAL	Network Abstraction Layer
NALU	NAL unit
NUT	NAL unit type
SEI	Supplemental Enhancement Information
SPS	Sequence Parameter Set
VCL	Video Coding Layer
VVC	Versatile Video Coding

**[0151]** Further definitions are discussed below.

**[0152]** In the above-description of various embodiments of present inventive concepts, it is to be understood that the terminology used herein is for the purpose of

describing particular embodiments only and is not intended to be limiting of present inventive concepts. Unless otherwise defined, all terms (including technical and scientific terms) used herein have the same meaning as commonly understood by one of ordinary skill in the art to which present inventive concepts belong. It will be further understood that terms, such as those defined in commonly used dictionaries, should be interpreted as having a meaning that is consistent with their meaning in the context of this specification and the relevant art and will not be interpreted in an idealized or overly formal sense unless expressly so defined herein.

**[0153]** When an element is referred to as being "connected", "coupled", "responsive", or variants thereof to another element, it can be directly connected, coupled, or responsive to the other element or intervening elements may be present. In contrast, when an element is referred to as being "directly connected", "directly coupled", "directly responsive", or variants thereof to another element, there are no intervening elements present. Like numbers refer to like elements throughout. Furthermore, "coupled", "connected", "responsive", or variants thereof as used herein may include wirelessly coupled, connected, or responsive. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. Well-known functions or constructions may not be described in detail for brevity and/or clarity. The term "and/or" includes any and all combinations of one or more of the associated listed items.

**[0154]** It will be understood that although the terms first, second, third, etc. may be used herein to describe various elements/operations, these elements/operations should not be limited by these terms. These terms are only used to distinguish one element/operation from another element/operation. Thus a first element/operation in some embodiments could be termed a second element/operation in other embodiments without departing from the teachings of present inventive concepts. The same reference numerals or the same reference designators denote the same or similar elements throughout the specification.

**[0155]** As used herein, the terms "comprise", "comprising", "comprises", "include", "including", "includes", "have", "has", "having", or variants thereof are open-ended, and include one or more stated features, integers, elements, steps, components or functions but does not preclude the presence or addition of one or more other features, integers, elements, steps, components, functions or groups thereof. Furthermore, as used herein, the common abbreviation "e.g.", which derives from the Latin phrase "exempli

gratia," may be used to introduce or specify a general example or examples of a previously mentioned item, and is not intended to be limiting of such item. The common abbreviation "i.e.", which derives from the Latin phrase "id est," may be used to specify a particular item from a more general recitation.

**[0156]** Example embodiments are described herein with reference to block diagrams and/or flowchart illustrations of computer-implemented methods, apparatus (systems and/or devices) and/or computer program products. It is understood that a block of the block diagrams and/or flowchart illustrations, and combinations of blocks in the block diagrams and/or flowchart illustrations, can be implemented by computer program instructions that are performed by one or more computer circuits. These computer program instructions may be provided to a processor circuit of a general purpose computer circuit, special purpose computer circuit, and/or other programmable data processing circuit to produce a machine, such that the instructions, which execute via the processor of the computer and/or other programmable data processing apparatus, transform and control transistors, values stored in memory locations, and other hardware components within such circuitry to implement the functions/acts specified in the block diagrams and/or flowchart block or blocks, and thereby create means (functionality) and/or structure for implementing the functions/acts specified in the block diagrams and/or flowchart block(s).

**[0157]** These computer program instructions may also be stored in a tangible computer-readable medium that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable medium produce an article of manufacture including instructions which implement the functions/acts specified in the block diagrams and/or flowchart block or blocks. Accordingly, embodiments of present inventive concepts may be embodied in hardware and/or in software (including firmware, resident software, micro-code, etc.) that runs on a processor such as a digital signal processor, which may collectively be referred to as "circuitry," "a module" or variants thereof.

**[0158]** It should also be noted that in some alternate implementations, the functions/acts noted in the blocks may occur out of the order noted in the flowcharts. For example, two blocks shown in succession may in fact be executed substantially concurrently or the blocks may sometimes be executed in the reverse order, depending upon the functionality/acts involved. Moreover, the functionality of a given block of the flowcharts and/or block diagrams may be separated into multiple blocks and/or the functionality of two or more blocks of the flowcharts and/or block diagrams may be at least partially integrated.

Finally, other blocks may be added/inserted between the blocks that are illustrated, and/or blocks/operations may be omitted without departing from the scope of inventive concepts. Moreover, although some of the diagrams include arrows on communication paths to show a primary direction of communication, it is to be understood that communication may occur in the opposite direction to the depicted arrows.

**[0159]** Many variations and modifications can be made to the embodiments without substantially departing from the principles of the present inventive concepts. All such variations and modifications are intended to be included herein within the scope of present inventive concepts. Accordingly, the above disclosed subject matter is to be considered illustrative, and not restrictive, and the examples of embodiments are intended to cover all such modifications, enhancements, and other embodiments, which fall within the spirit and scope of present inventive concepts. Thus, to the maximum extent allowed by law, the scope of present inventive concepts are to be determined by the broadest permissible interpretation of the present disclosure including the examples of embodiments and their equivalents, and shall not be restricted or limited by the foregoing detailed description.

**[0160]** Generally, all terms used herein are to be interpreted according to their ordinary meaning in the relevant technical field, unless a different meaning is clearly given and/or is implied from the context in which it is used. All references to a/an/the element, apparatus, component, means, step, etc. are to be interpreted openly as referring to at least one instance of the element, apparatus, component, means, step, etc., unless explicitly stated otherwise. The steps of any methods disclosed herein do not have to be performed in the exact order disclosed, unless a step is explicitly described as following or preceding another step and/or where it is implicit that a step must follow or precede another step. Any feature of any of the embodiments disclosed herein may be applied to any other embodiment, wherever appropriate. Likewise, any advantage of any of the embodiments may apply to any other embodiments, and vice versa. Other objectives, features and advantages of the enclosed embodiments will be apparent from the following description.

**[0161]** Some of the embodiments contemplated herein will now be described more fully with reference to the accompanying drawings. Other embodiments, however, are contained within the scope of the subject matter disclosed herein, the disclosed subject matter should not be construed as limited to only the embodiments set forth herein; rather, these embodiments are provided by way of example to convey the scope of the subject matter to those skilled in the art.

**[0162]** Any appropriate steps, methods, features, functions, or benefits disclosed herein may be performed through one or more functional units or modules of one or more virtual apparatuses. Each virtual apparatus may comprise a number of these functional units. These functional units may be implemented via processing circuitry, which may include one or more microprocessor or microcontrollers, as well as other digital hardware, which may include digital signal processors (DSPs), special-purpose digital logic, and the like. The processing circuitry may be configured to execute program code stored in memory, which may include one or several types of memory such as read-only memory (ROM), random-access memory (RAM), cache memory, flash memory devices, optical storage devices, etc. Program code stored in memory includes program instructions for executing one or more telecommunications and/or data communications protocols as well as instructions for carrying out one or more of the techniques described herein. In some implementations, the processing circuitry may be used to cause the respective functional unit to perform corresponding functions according one or more embodiments of the present disclosure.

**[0163]** The term unit may have conventional meaning in the field of electronics, electrical devices and/or electronic devices and may include, for example, electrical and/or electronic circuitry, devices, modules, processors, memories, logic solid state and/or discrete devices, computer programs or instructions for carrying out respective tasks, procedures, computations, outputs, and/or displaying functions, and so on, as such as those that are described herein.



## CLAIMS

1. A method performed by a decoder to decode a bitstream, the method comprising:
  - decoding (1200) one or more syntax elements from the bitstream;
  - deriving (1202) a first value from the one or more syntax elements;
  - comparing (1204) the first value to a predetermined set of values that comprises at least two values;
  - determining (1206) whether the first value is within a predetermined range of values in the predetermined set of values; and
  - responsive to determining the first value is within the predetermined range of values in the predetermined set of values, determining (1208) that a location of an access unit boundary is indicated.
2. The method of Claim 1 wherein determining whether the first value is within the predetermined range of values in the predetermined set of values comprises:
  - determining whether the first value is equal to a value in the predetermined range of values.
3. The method of any of Claims 1-2 wherein determining whether the first value is within the predetermined range of values comprises determining whether the first value is larger than or equal to a first predetermined threshold value.
4. The method of any of Claims 1-2 wherein determining whether the first value is within the predetermined range of values comprises determining whether the first value is smaller than or equal to a second predetermined threshold value.
5. A method performed by a decoder, the method comprising:
  - decoding (700) a codeword from a network abstraction layer, NAL, unit header of a first NAL unit; and
  - determining (704) whether the codeword indicates a start of an access unit by:
    - determining (706) that the first NAL unit is a first NAL unit of an access unit responsive to the codeword being equal to one or more specified values; and
    - determining (708) that the first NAL unit is not a first NAL unit of an access unit responsive to the codeword being equal to one or more different specified values.

6. The method of Claim 5 wherein the codeword comprises a flag.
7. The method of any of Claims 5-6 wherein the codeword indicates one or more features of an access unit delimiter.
8. The method of Claim 7 where the one or more features may be one or more of a start of the access unit, an end of the access unit, a start of a CVS, an end of a CVS, a first VCL NAL unit in the access unit, a last VCL NAL unit in the access unit, a start of the access unit of a current layer, an end of the access unit of the current layer, a number of NAL units in the access unit, a number of VCL NAL units in the access unit, or a number of non-VCL NAL units in the access unit.
9. A method performed by a decoder, the method comprising:
  - decoding (2100) a value from a syntax element in a parameter set, the value indicating whether or not there are access unit delimiter, AUD, network abstraction layer, NAL units in the access units following the current access unit in decoding order in the coded video stream, CVS; and
  - determining (2102) whether the value indicates that there are AUD NAL units in the access units following the current access unit in decoding order in the CVS.
10. The method of Claim 9, further comprising determining (2104) from a syntax element in the parameter set a number of NAL units in a first access unit.
11. The method of Claim 9, further comprising determining whether there are AUD NAL units in the access units following the current access unit in decoding order in the CVS based on a specific value of NAL unit type
12. A decoder for a communication network, the decoder (406) comprising:
  - a processor (601); and
  - memory (603) coupled with the processor, wherein the memory comprises instructions that when executed by the processor cause the processor to perform operations according to any of Claims 1-11.
13. An apparatus comprising:
  - at least one processor (601);

memory (603) communicatively coupled to the processor, said memory comprising instructions executable by the processor, which cause the processor to perform operations comprising operations according to any of Claims 1-11.

14. A decoder comprising:

at least one processor (601);

memory (603) communicatively coupled to the processor, said memory comprising instructions executable by the processor, which cause the processor to perform operations comprising:

decoding (1200) one or more syntax elements from the bitstream;

deriving (1202) a first value from the one or more syntax elements;

comparing (1204) the first value to a predetermined set of values that comprises at least two values;

determining (1206) whether the first value is within a predetermined range of values in the predetermined set of values; and

responsive to determining the first value is within the predetermined range of values in the predetermined set of values, determining (1208) that a location of an access unit boundary is indicated.

15. The method of Claim 14 wherein determining whether the first value is within the predetermined range of values in the predetermined set of values comprises:

determining whether the first value is equal to a value in the predetermined range of values.

16. The method of any of Claims 14-15 wherein determining whether the first value is within the predetermined range of values comprises determining whether the first value is larger than or equal to a first predetermined threshold value.

17. The method of any of Claims 14-15 wherein determining whether the first value is within the predetermined range of values comprises determining whether the first value is smaller than or equal to a second predetermined threshold value.

18. A decoder (406) comprising:

at least one processor (601);

memory (603) communicatively coupled to the processor, said memory comprising instructions executable by the processor, which cause the processor to perform operations comprising:

decoding (700) a codeword from a network abstraction layer, NAL, unit header of a first NAL unit; and

determining (704) whether the codeword indicates a start of an access unit by:

determining (706) that the first NAL unit is a first NAL unit of an access unit responsive to the codeword being equal to one or more specified values; and

determining (708) that the first NAL unit is not a first NAL unit of an access unit responsive to the codeword being equal to one or more different specified values.

19. The decoder (406) of Claim 18 wherein the codeword comprises a flag.

20. The decoder (406) of any of Claims 18-19 wherein the codeword indicates one or more features of an access unit delimiter.

21. The decoder (406) of Claim 20 where the one or more features may be one or more of a start of the access unit, an end of the access unit, a start of a CVS, an end of a CVS, a first VCL NAL unit in the access unit, a last VCL NAL unit in the access unit, a start of the access unit of a current layer, an end of the access unit of the current layer, a number of NAL units in the access unit, a number of VCL NAL units in the access unit, or a number of non-VCL NAL units in the access unit.

22. A decoder (406) comprising:

at least one processor (601);

memory (603) communicatively coupled to the processor, said memory comprising instructions executable by the processor, which cause the processor to perform operations comprising:

decoding (2100) a value from a syntax element in a parameter set, the value indicating whether or not there are access unit delimiter, AUD, network abstraction layer, NAL units in the access units following the current access unit in decoding order in the coded video stream, CVS; and

determining (2102) whether the value indicates that there are AUD NAL units in the access units following the current access unit in decoding order in the CVS.

23. The decoder (406) of Claim 22, further comprising determining (2104) from a syntax element in the parameter set a number of NAL units in a first access unit.
24. The decoder (406) of Claim 22, further comprising determining whether there are AUD NAL units in the access units following the current access unit in decoding order in the CVS based on a specific value of NAL unit type.
25. A method performed by an encoder (400), the method comprising:  
determining (1700) whether an access unit boundary is being encoded; and  
responsive to an access unit boundary being encoded, encoding (1702) one or more syntax values in a bitstream, with a first value of the one or more syntax values indicating a location of the access unit boundary.
26. The method of Claim 25, wherein the first value is within a predetermined range of values.
27. The method of any of Claims 25-26, wherein the first value is larger than or equal to a first predetermined threshold value.
28. A method performed by an encoder (400), the method comprising:  
determining (1700) whether access unit delimiter, AUD, network abstraction layer, NAL, units are to be present in access units following a current access unit in decoding order; and  
responsive to determining that AUD NAL units are to be present, encoding (1702) a value of a NAL unit type in a NAL unit header of a NAL unit in a video stream being encoded to indicate that AUD NAL units are to be present in access units following the current access unit in decoding order.
29. A method performed by an encoder (400), the method comprising:  
determining (2200) whether access unit delimiter, AUD, network abstraction layer, NAL, units are to be present in access units following a current access unit in decoding order in a coded video stream, CVS; and  
responsive to determining that AUD NAL units are to be present in access units following the current access unit in decoding order, encoding (2202) a value of a syntax element in a parameter set of the coded video stream, the value indicating that there are AUD NAL units in access units following the current access unit in decoding order.

30. The method of Claim 29, further comprising encoding (2204) a value of a second syntax element in the parameter set, the value of the second syntax element indicating a number of NAL units in a first access unit.
31. The method of Claim 29, further comprising responsive to determining that AUD NAL units are to be present in access units following the current access unit in decoding order, encoding a specific value of NAL unit type in a NAL unit header.
32. An encoder (400) for a communication network, the encoder (400) comprising:  
a processor (501); and  
memory (503) coupled with the processor (501), wherein the memory (503) comprises instructions that when executed by the processor (501) cause the processor (501) to perform operations according to any of Claims 25-31.
33. An encoder (400) comprising:  
at least one processor (501);  
memory (503) communicatively coupled to the processor, said memory comprising instructions executable by the processor, which cause the processor to perform operations comprising:  
determining (1900) whether an access unit boundary is being encoded; and  
responsive to an access unit boundary being encoded, encoding (1902) one or more syntax values in a bitstream, with a first value of the one or more syntax values indicating a location of the access unit boundary.
34. The encoder (400) of Claim 33, wherein the first value is within a predetermined range of values.
35. The method of any of Claims 33-34, wherein the first value is larger than or equal to a first predetermined threshold value.
36. An encoder (400) comprising:  
at least one processor (501);  
memory (503) communicatively coupled to the processor, said memory comprising instructions executable by the processor, which cause the processor to perform operations comprising:

determining (1700) whether access unit delimiter, AUD, network abstraction layer, NAL, units are to be present in access units following a current access unit in decoding order; and

responsive to determining that AUD NAL units are to be present, encoding (1702) a value of a NAL unit type in a NAL unit header of a NAL unit in a video stream being encoded to indicate that AUD NAL units are to be present in access units following the current access unit in decoding order.

37. An encoder (400) comprising:

at least one processor (501);

memory (503) communicatively coupled to the processor, said memory comprising instructions executable by the processor, which cause the processor to perform operations comprising:

determining (2200) whether access unit delimiter, AUD, network abstraction layer, NAL, units are to be present in access units following a current access unit in decoding order in a coded video stream, CVS; and

responsive to determining that AUD NAL units are to be present in access units following the current access unit in decoding order, encoding (2202) a value of a syntax element in a parameter set of the CVS, the value indicating that there are AUD NAL units in access units following the current access unit in decoding order.

38. The encoder (400) of Claim 37, wherein the memory comprises further instructions executable by the processor, which when executed, cause the processor to perform operations further comprising encoding (2204) a value of a second syntax element in the parameter set, the value of the second syntax element indicating a number of NAL units in a first access unit.

39. The encoder (400) of Claim 37, wherein the memory comprises further instructions executable by the processor, which when executed, cause the processor to perform operations further comprising responsive to determining that AUD NAL units are to be present in access units following the current access unit in decoding order, encoding a specific value of NAL unit type in a NAL unit header.

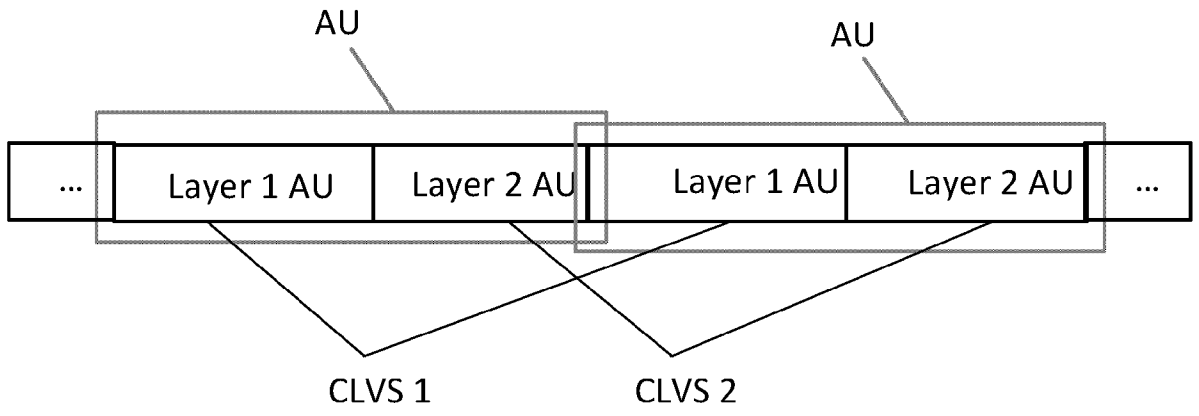


Figure 1

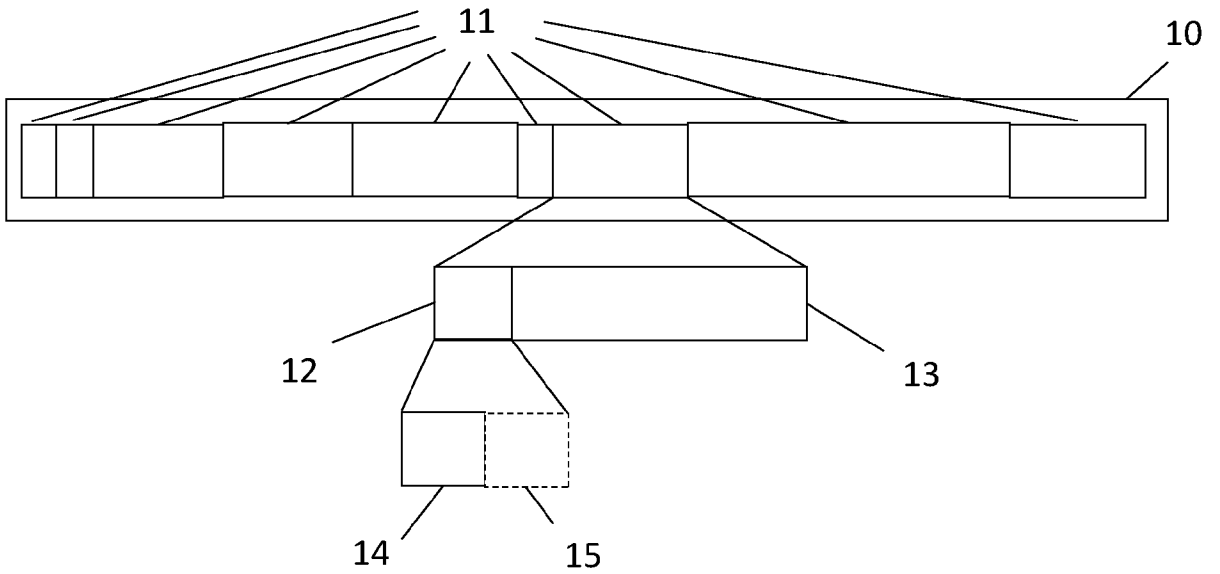


Figure 2

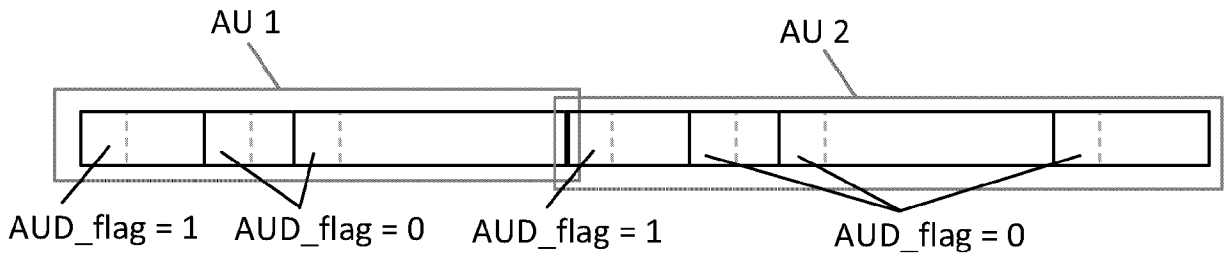


Figure 3



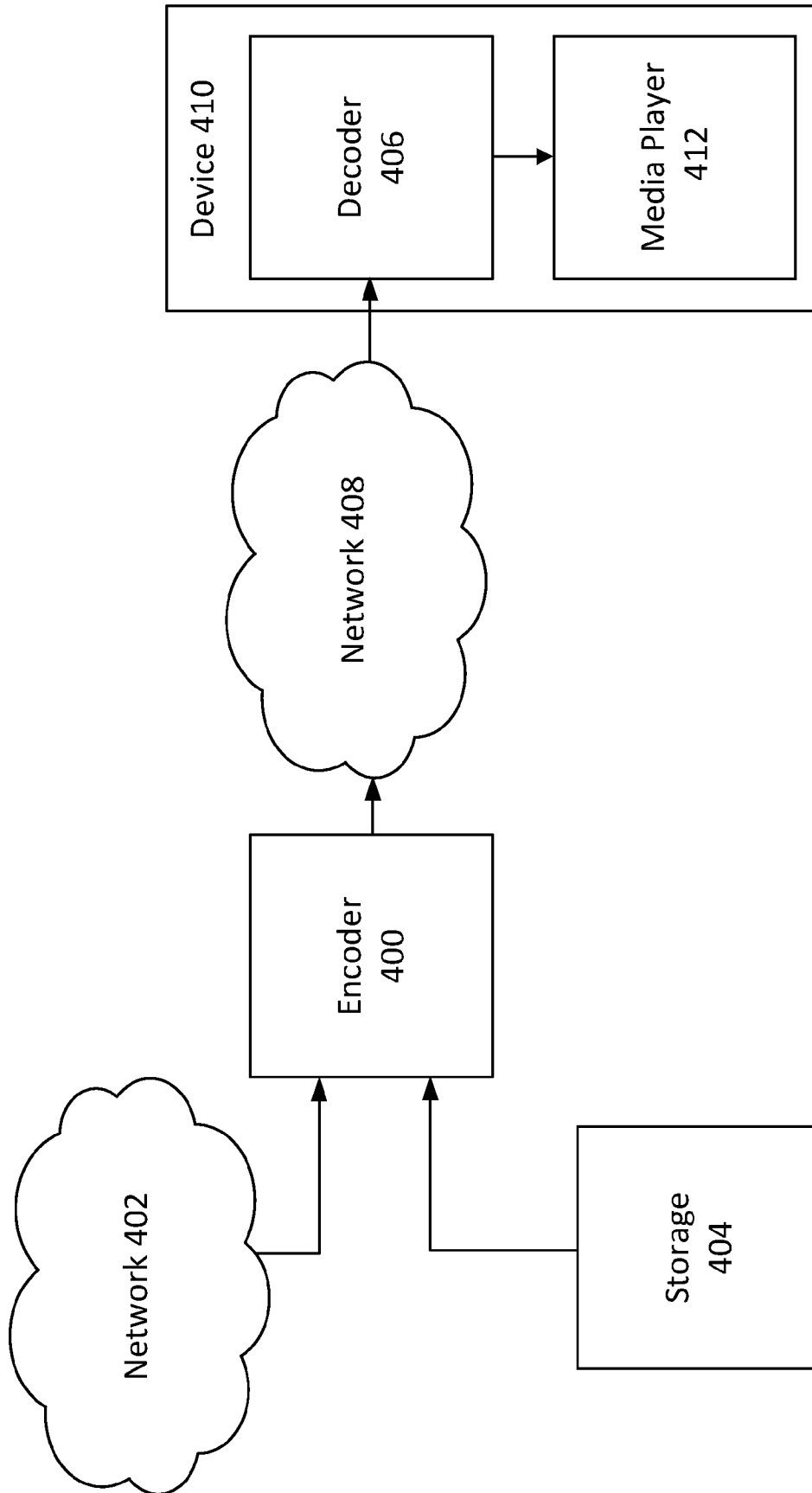


Figure 4

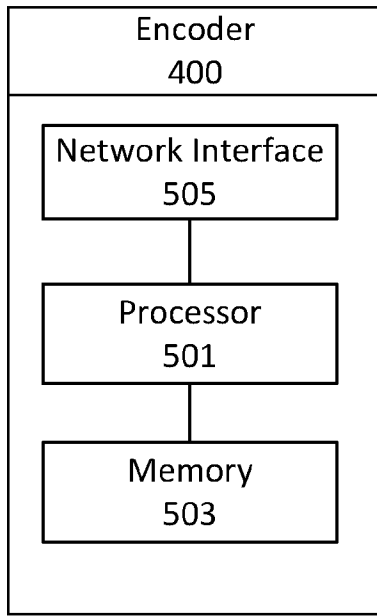


Figure 5

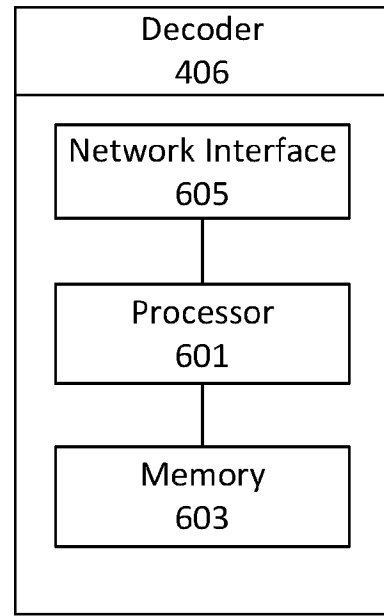


Figure 6

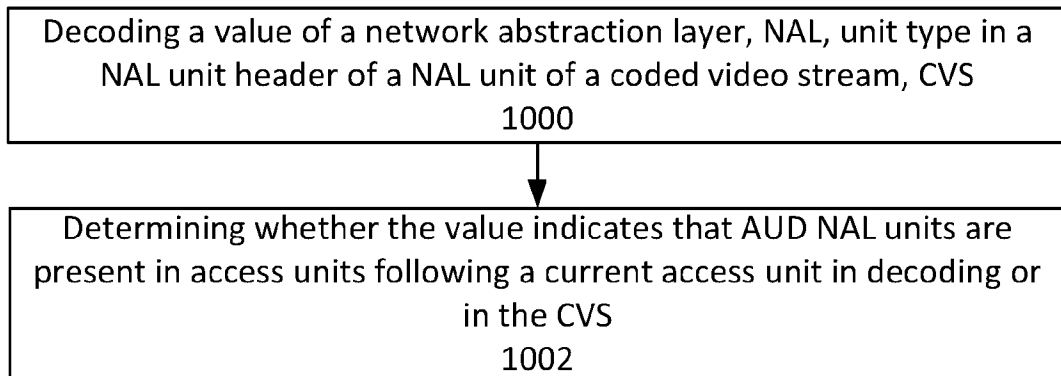


Figure 10

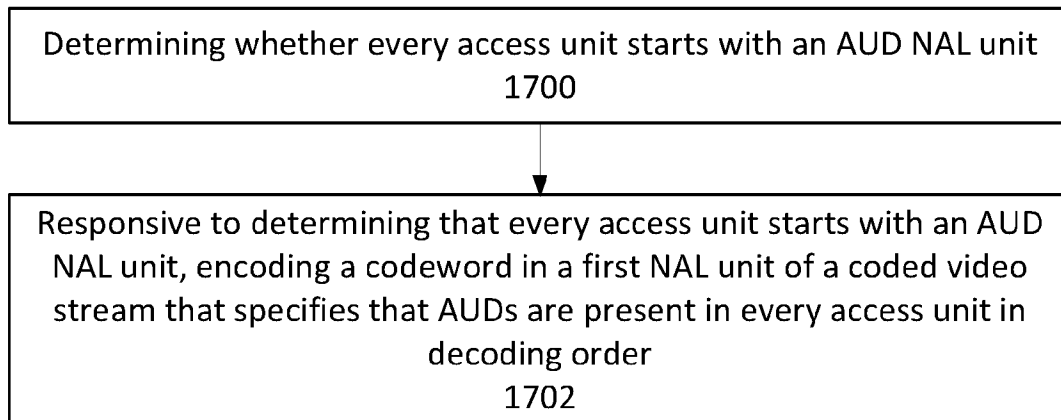


Figure 17

4/10

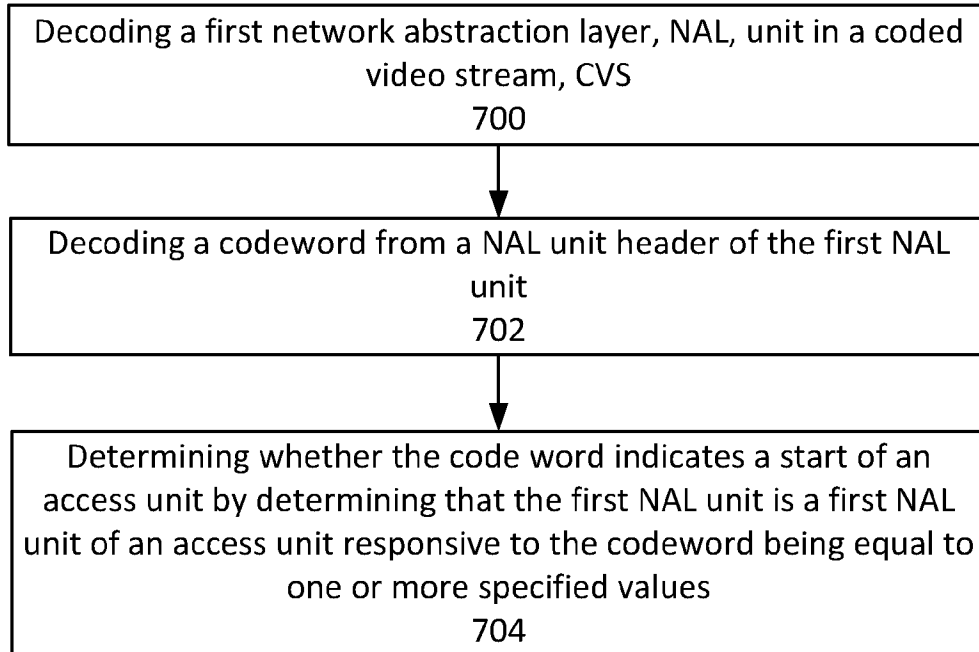


Figure 7

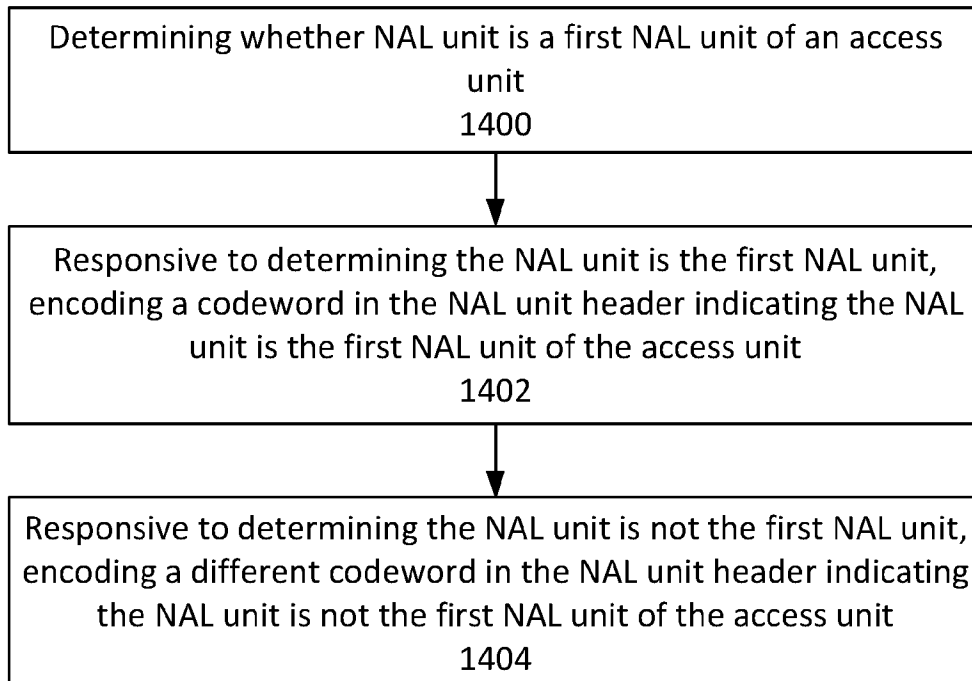


Figure 14

5/10

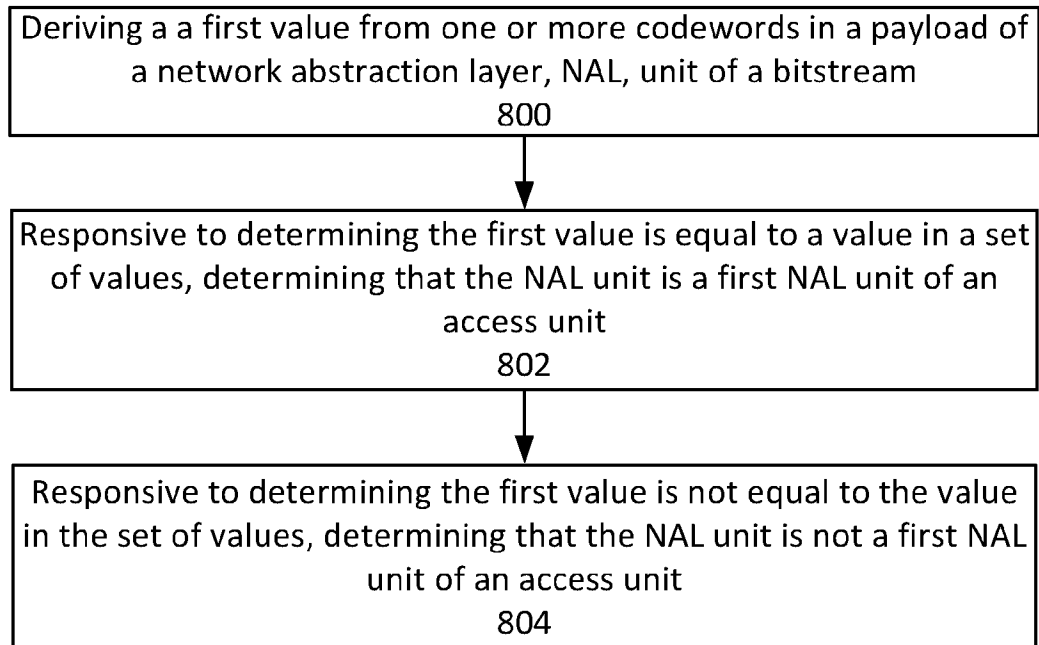


Figure 8

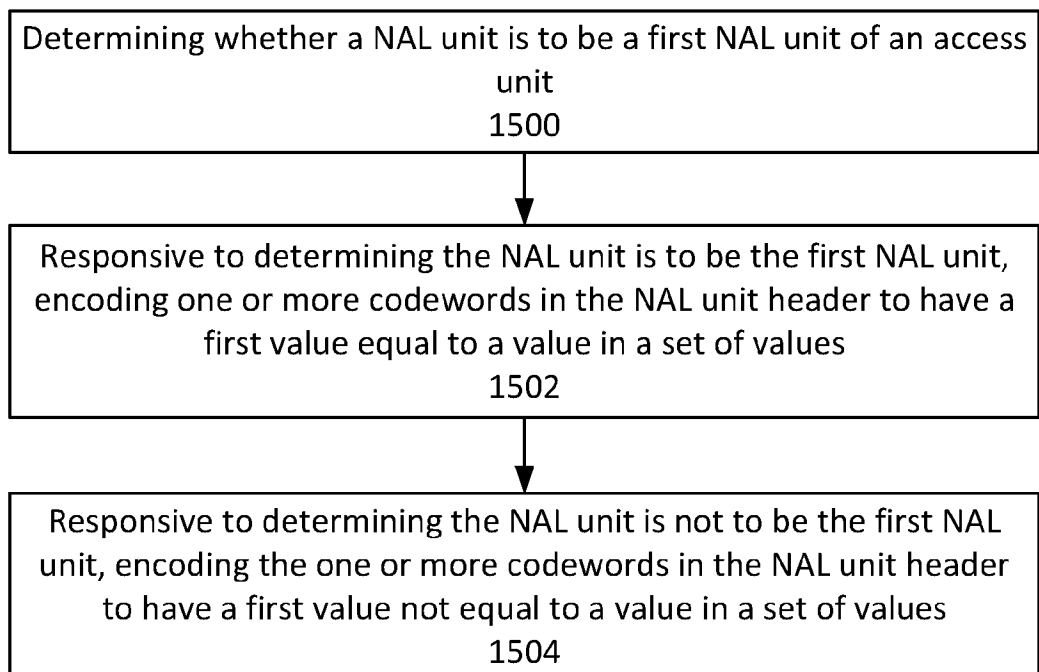


Figure 15

6/10

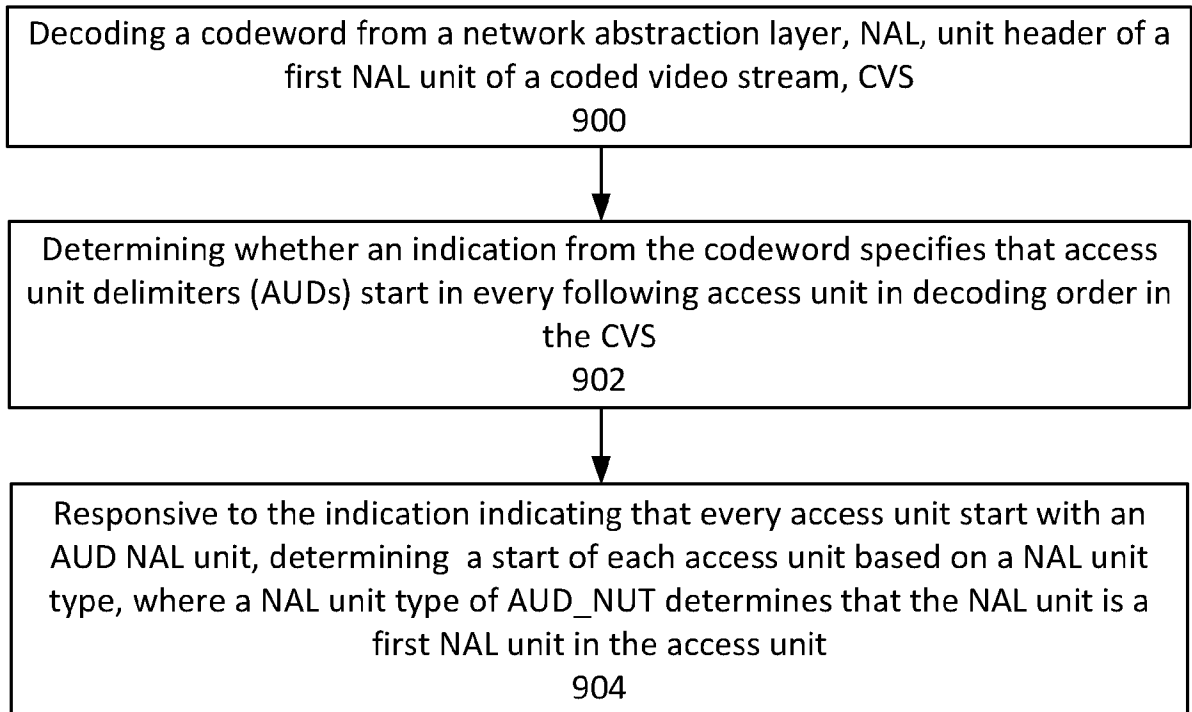


Figure 9

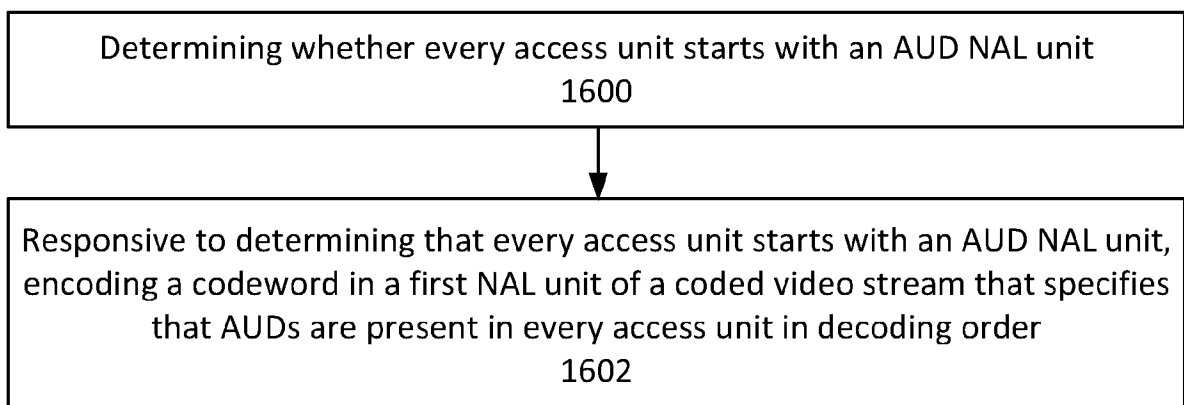


Figure 16

7/10

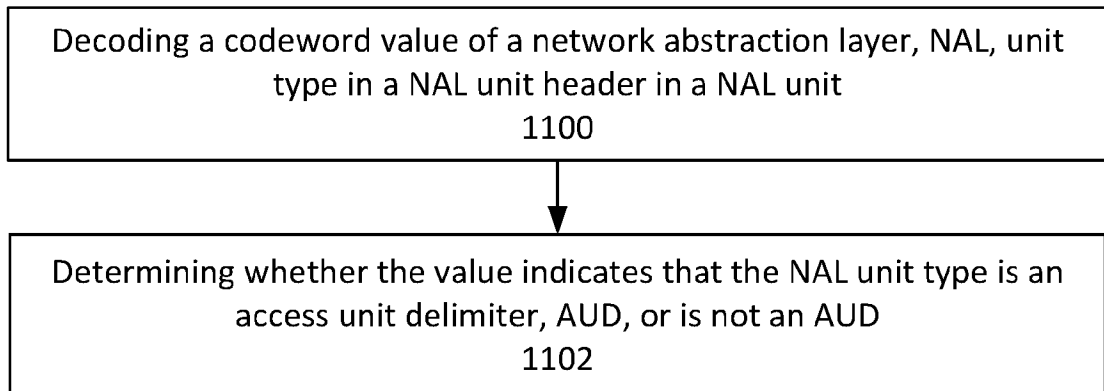


Figure 11

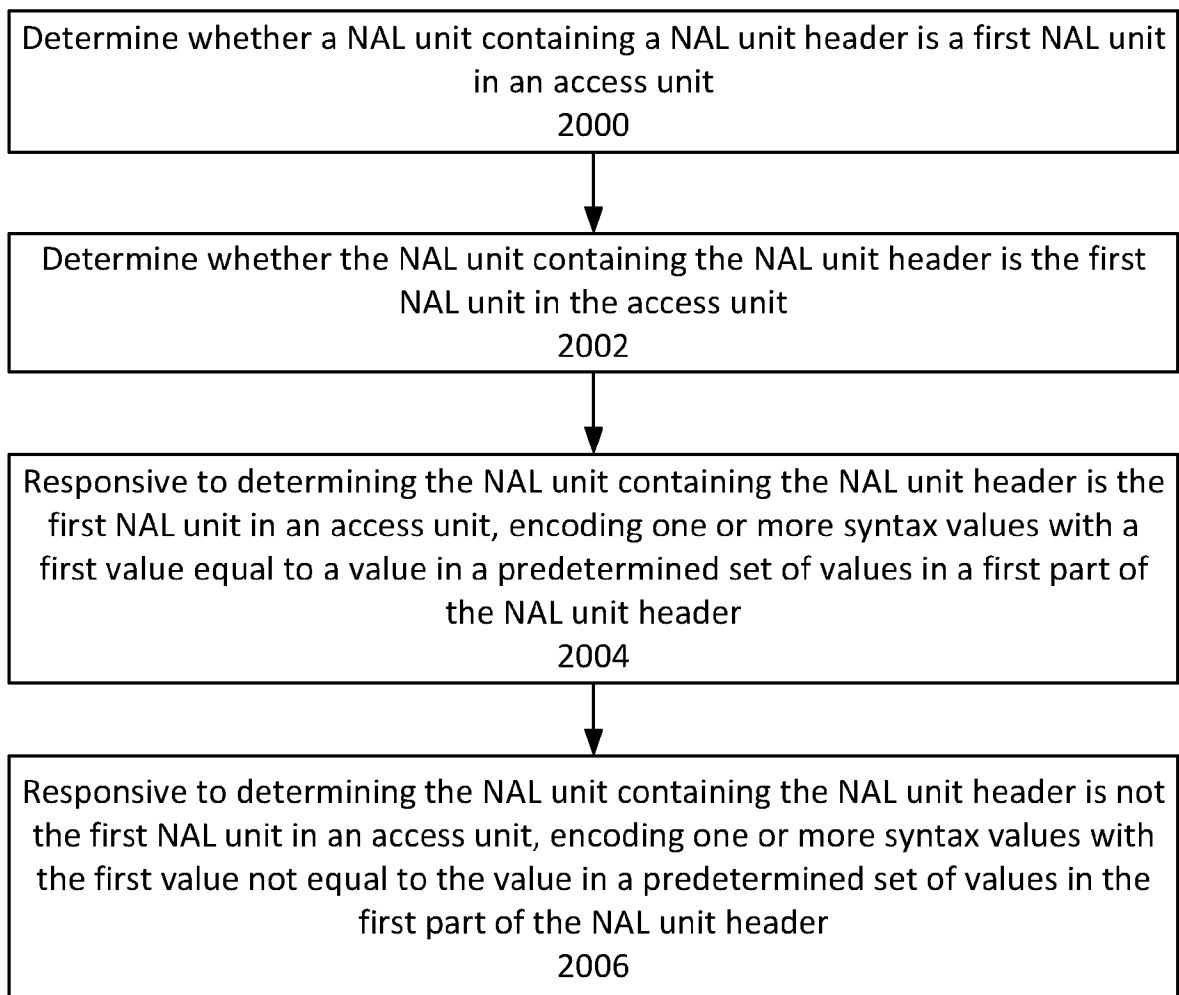


Figure 20

8/10

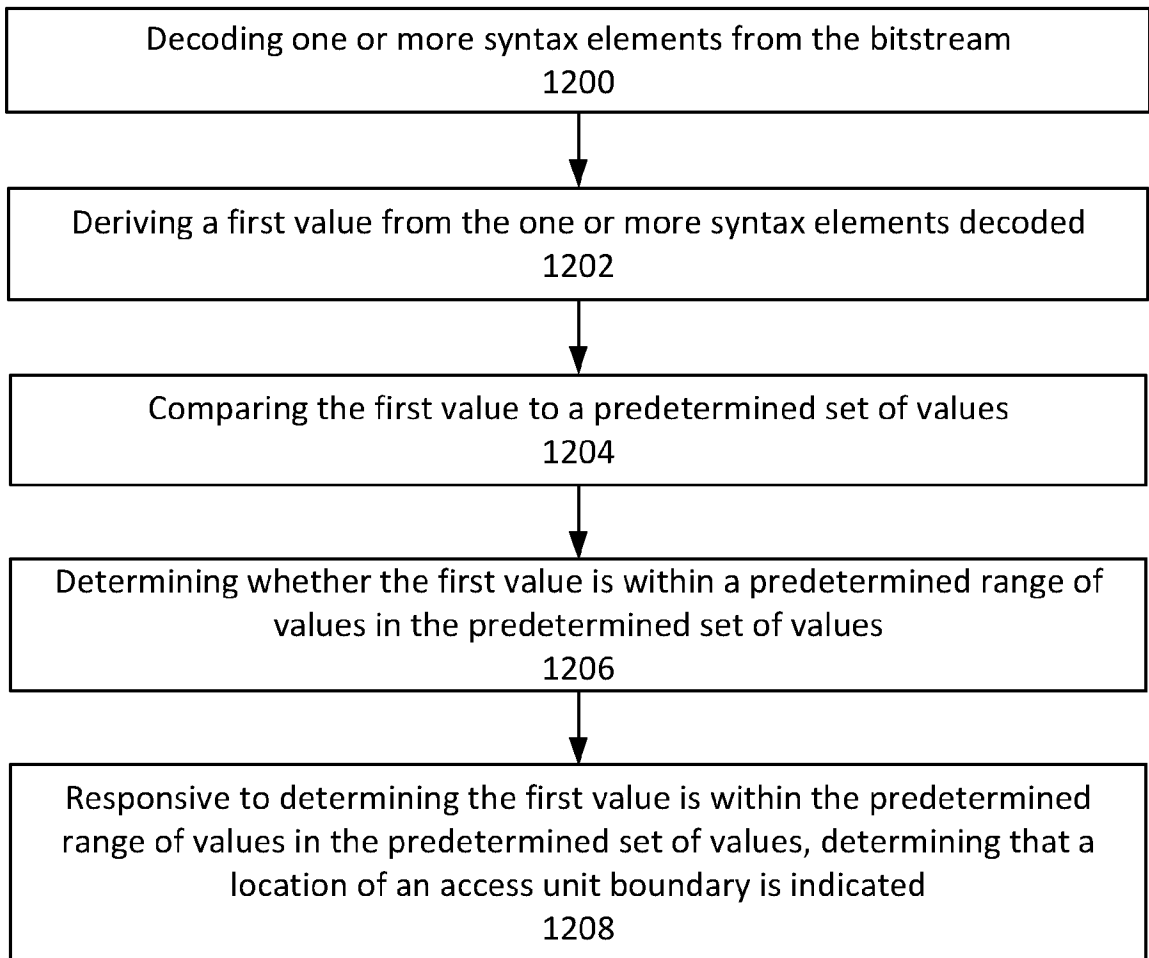


Figure 12

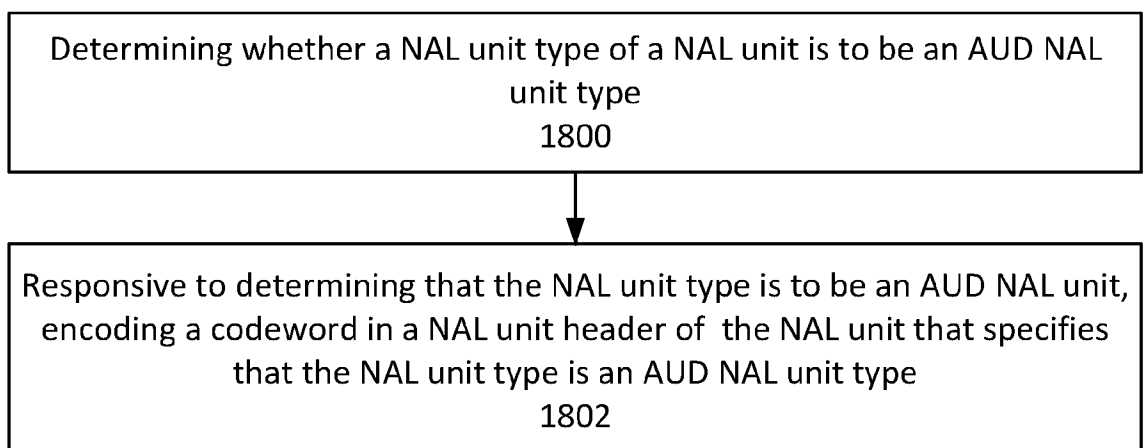


Figure 18

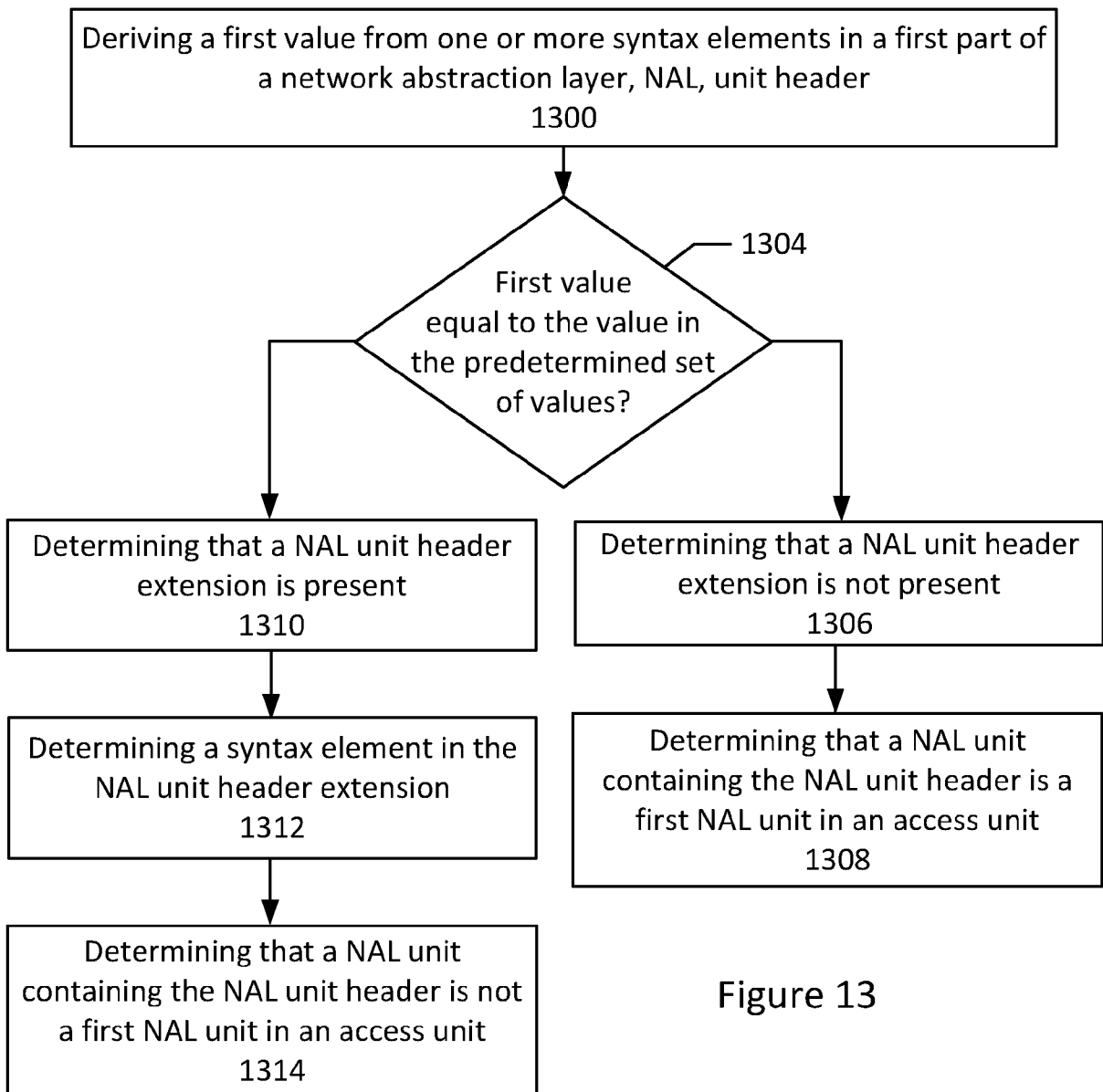


Figure 13

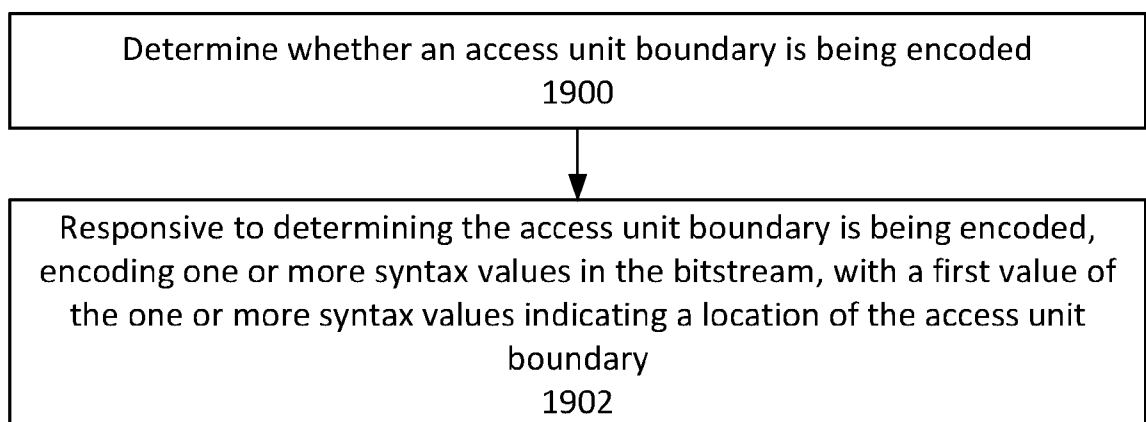


Figure 19



10/10

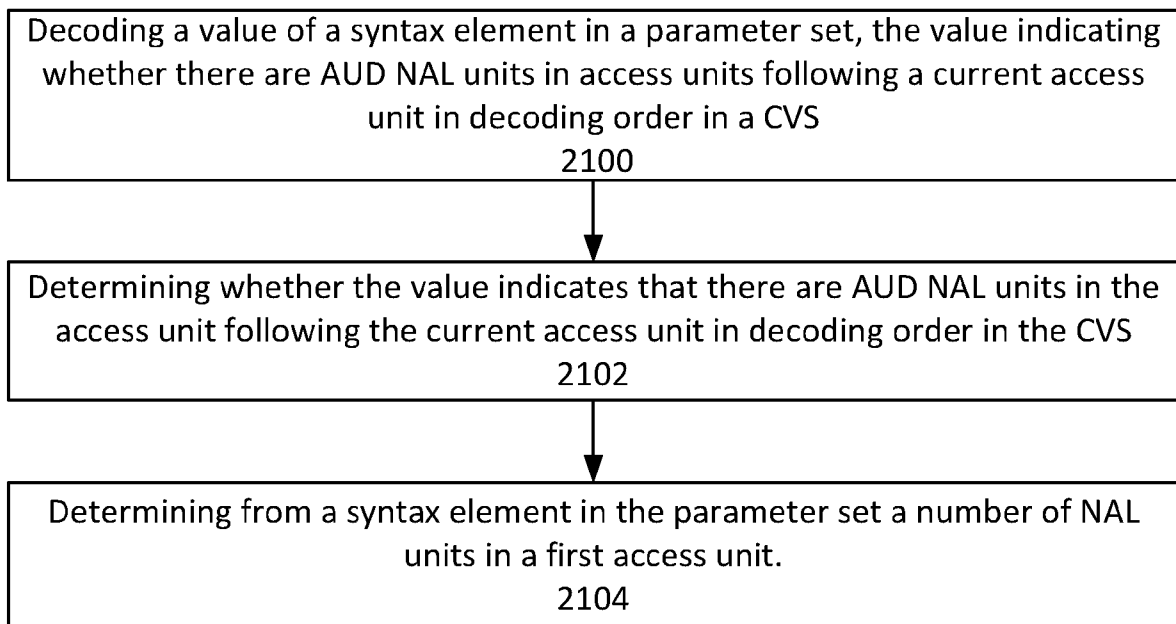


Figure 21

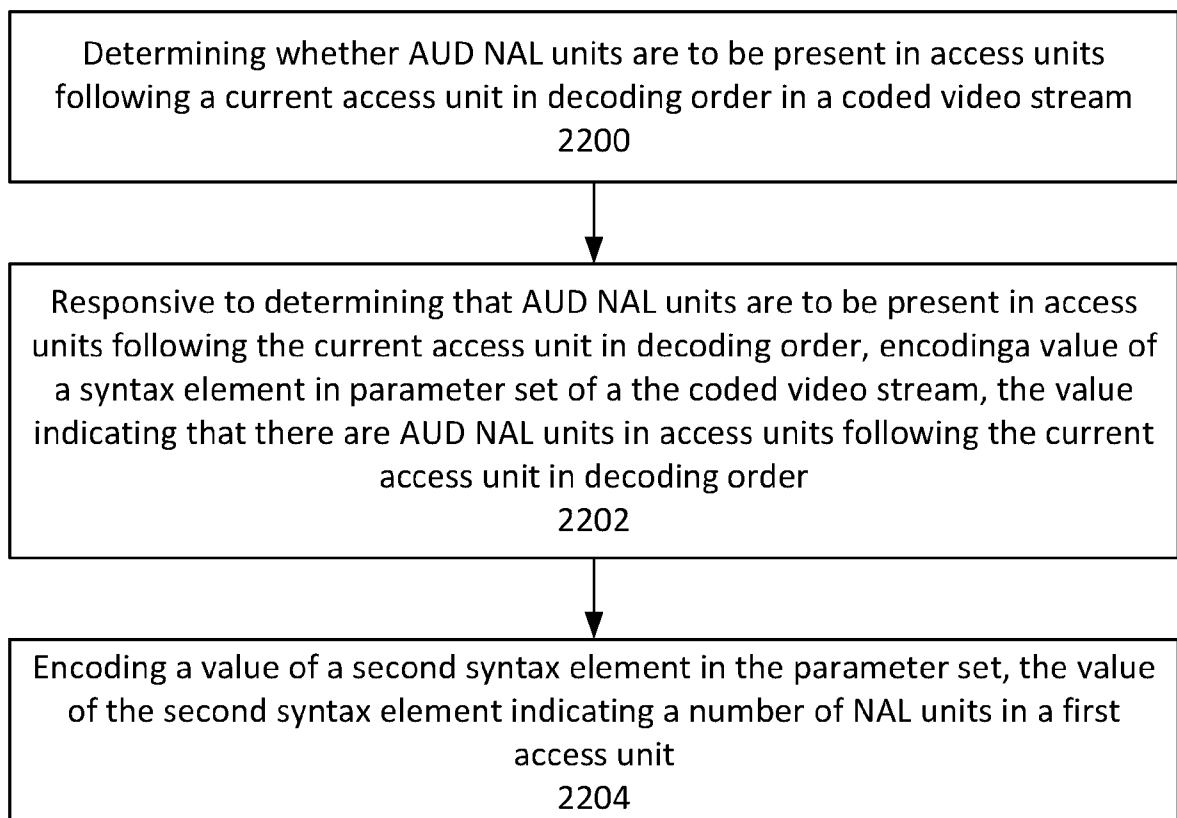


Figure 22

INTERNATIONAL SEARCH REPORT

International application No  
PCT/EP2020/069304

A. CLASSIFICATION OF SUBJECT MATTER  
INV. H04N19/46 H04N19/70  
ADD.  
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED  
Minimum documentation searched (classification system followed by classification symbols)  
H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)  
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	"Working Draft 5 of Versatile Video Coding", 126. MPEG MEETING; 20190325 - 20190329; GENEVA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), , no. n18370 7 July 2019 (2019-07-07), XP030222340, Retrieved from the Internet: URL:http://phenix.int-evry.fr/mpeg/doc_end_user/documents/126_Geneva/wg11/w18370.zip w18370.docx [retrieved on 2019-07-07]	1-5, 12-18, 25-27, 32-35
Y	pages 70-73  -----  -/--	6-8, 19-21

Further documents are listed in the continuation of Box C.

See patent family annex.

\* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search  6 November 2020	Date of mailing of the international search report  17/11/2020
--	--

Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer  Sanahuja, Francesc
--	--

## INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2020/069304

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	HENG (BROADCOM) B ET AL: "AHG17: Association of VCL NAL units to coded pictures", 15. JVET MEETING; 20190703 - 20190712; GOTHENBURG; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16 ), , no. JVET-00610 26 June 2019 (2019-06-26), XP030220036, Retrieved from the Internet: URL:http://phenix.int-evry.fr/jvet/doc_end_user/documents/15_Gothenburg/wg11/JVET-00610-v1.zip JVET-00610.DOCX [retrieved on 2019-06-26]	1-4, 15-17, 26,27, 34,35
Y	Sections 1 and 2	6-8, 19-21
-----		
X	HANNUKSELA (NOKIA) M M: "Version 1/MV-HEVC/SHVC HLS: Access unit boundary detection", 7. JCT-3V MEETING; 20140111 - 20140117; SAN JOSE; (THE JOINT COLLABORATIVE TEAM ON 3D VIDEO CODING EXTENSION DEVELOPMENT OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16 ), , no. JCT3V-G0032 19 December 2013 (2013-12-19), XP030131775, Retrieved from the Internet: URL:http://phenix.int-evry.fr/jct2/doc_end_user/documents/7_San%20Jose/wg11/JCT3V-G0032-v1.zip JCTVC-P0043 JCT3V-G0032.doc [retrieved on 2014-01-13]	1-4, 15-17, 26,27, 34,35
Y	Sections 1-5	6-8, 19-21
-----		
X	SJÖBERG (ERICSSON) R ET AL: "AHG17: Picture header NAL unit type", 13. JVET MEETING; 20190109 - 20190118; MARRAKECH; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16 ), , no. JVET-M0377 12 January 2019 (2019-01-12), XP030201783, Retrieved from the Internet: URL:http://phenix.int-evry.fr/jvet/doc_end_user/documents/13_Marrakech/wg11/JVET-M0377-v3.zip JVET-M0377-v2.docx [retrieved on 2019-01-12]	9,22,29, 37
Y	Abstract, sections 2 and 3	10,23, 30,38
A		24,28, 31,36,39
-----		
	-/--	

## INTERNATIONAL SEARCH REPORT

International application No  
PCT/EP2020/069304

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	US 2016/156968 A1 (TOMA TADAMASA [JP] ET AL) 2 June 2016 (2016-06-02) paragraphs [0159] - [0162]	10,23, 30,38 24,28, 31,36,39
A	<p style="text-align: center;">-----</p> <p>"Study Text of ISO/IEC 13818-1:200/FPDAM3", 64. MPEG MEETING; 20030310 - 20030314; PATTAYA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), , no. n5605 10 April 2003 (2003-04-10), XP030279255, Retrieved from the Internet: URL:<a href="http://phenix.int-evry.fr/mpeg/doc_end_user/documents/64_Pattaya/wg11/w5605.zip">http://phenix.int-evry.fr/mpeg/doc_end_user/documents/64_Pattaya/wg11/w5605.zip</a> w5605.doc [retrieved on 2010-08-27] Pages 9-10, clause 2.12</p> <p style="text-align: center;">-----</p>	9-11, 22-24, 28-31, 36-39

# INTERNATIONAL SEARCH REPORT

International application No.  
PCT/EP2020/069304

## Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1.  Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
  
2.  Claims Nos.:  
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
  
3.  Claims Nos.:  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

## Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1.  As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2.  As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.
3.  As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
  
4.  No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

### Remark on Protest

- The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- No protest accompanied the payment of additional search fees.

**FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210**

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-8, 12-21, 25-27, 32-35

Method for identifying the location of a current access unit.

---

2. claims: 9-11, 22-24, 28-31, 36-39

Method for determining whether subsequent access units will be signalled by an access unit delimiter (AUD).

---

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2020/069304

Patent document cited in search report	Publication date	Patent family member(s)	Publication date	
US 2016156968	A1	02-06-2016	CN 105210365 A	30-12-2015
			EP 3070933 A1	21-09-2016
			JP 6411862 B2	24-10-2018
			JP 2015097386 A	21-05-2015
			JP 2019024229 A	14-02-2019
			US 2016156968 A1	02-06-2016
			US 2017214969 A1	27-07-2017
			US 2018302674 A1	18-10-2018
-----				