



US 20250260818A1

(19) **United States**

(12) **Patent Application Publication**  
**DE LAGRANGE et al.**

(10) **Pub. No.: US 2025/0260818 A1**

(43) **Pub. Date: Aug. 14, 2025**

(54) **HIGH-LEVEL CONSTRAINT FLAG FOR  
LOCAL CHROMA QUANTIZATION  
PARAMETER CONTROL**

(30) **Foreign Application Priority Data**

Mar. 12, 2020 (EP) ..... 20305258.4

**Publication Classification**

(51) **Int. Cl.**

**H04N 19/124** (2014.01)

**H04N 19/176** (2014.01)

**H04N 19/70** (2014.01)

(52) **U.S. Cl.**

CPC ..... **H04N 19/124** (2014.11); **H04N 19/176**  
(2014.11); **H04N 19/70** (2014.11)

(71) Applicant: **INTERDIGITAL CE PATENT  
HOLDINGS, SAS**, Paris (FR)

(72) Inventors: **Philippe DE LAGRANGE**, Betton  
(FR); **Karam NASER**, Mouaze (FR);  
**Philippe BORDES**, Laille (FR);  
**Fabrice LELEANNEC**, Betton (FR)

(21) Appl. No.: **19/049,279**

(22) Filed: **Feb. 10, 2025**

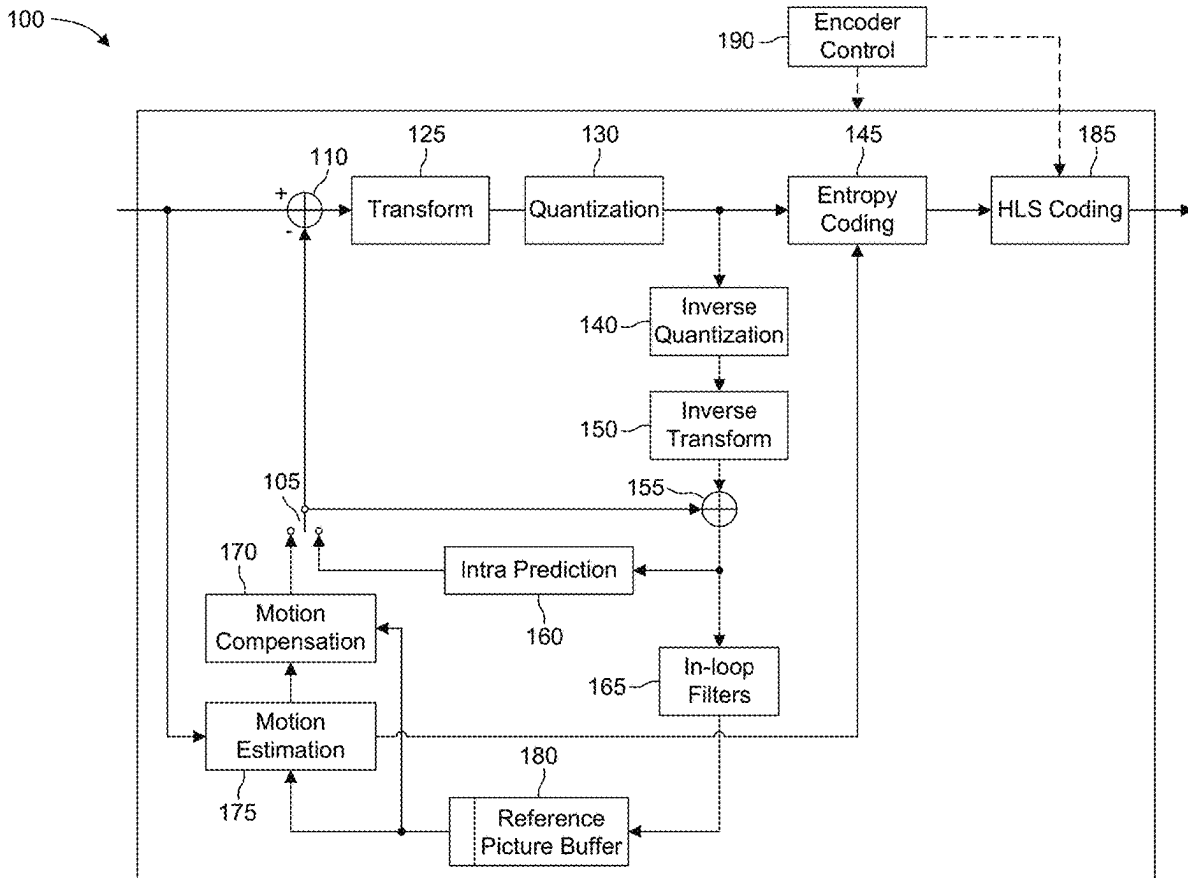
**Related U.S. Application Data**

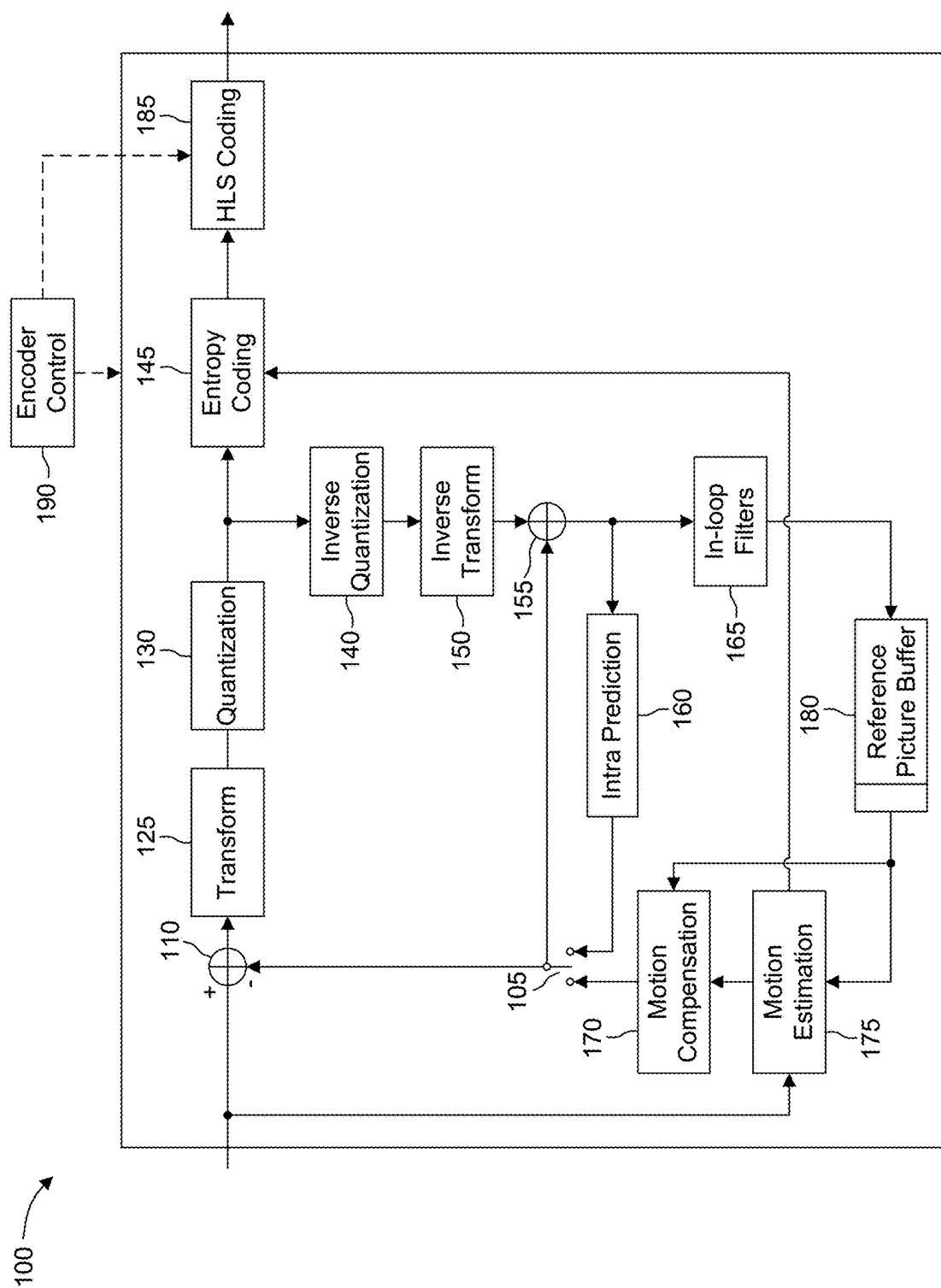
(63) Continuation of application No. 17/910,047, filed on  
Sep. 8, 2022, filed as application No. PCT/EP2021/  
055776 on Mar. 8, 2021.

(57)

**ABSTRACT**

A method and apparatus to improve compression efficiency in a video compression scheme enables separate control of a local chroma quantization parameter. The control can be used to disable a local chroma quantization parameter. One embodiment uses an existing local luma quantization parameter control flag for controlling local chroma quantization parameter. Another embodiment includes a separate local chroma quantization parameter control flag. Bitstream syntax is used to convey the flags.





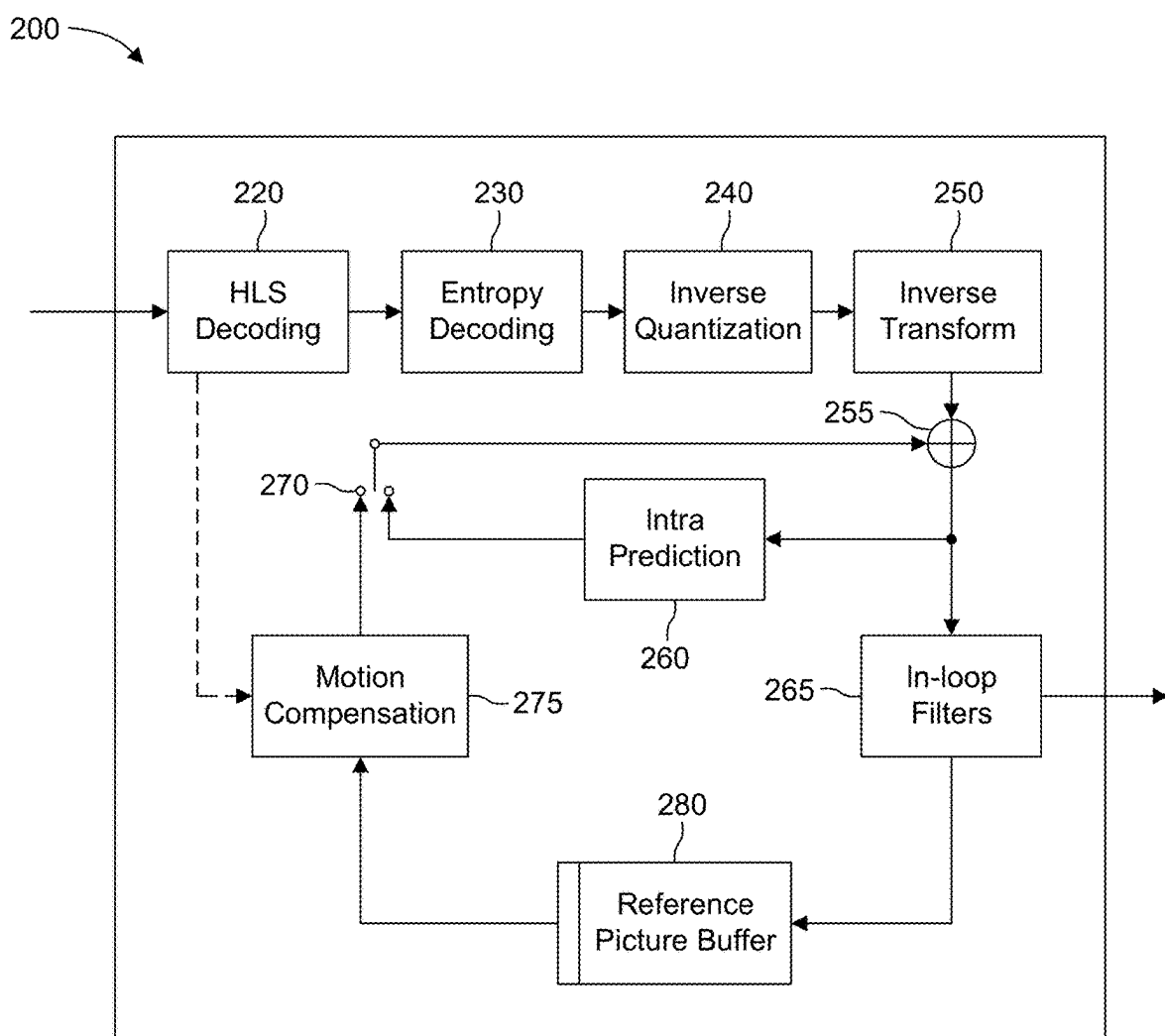


Figure 2

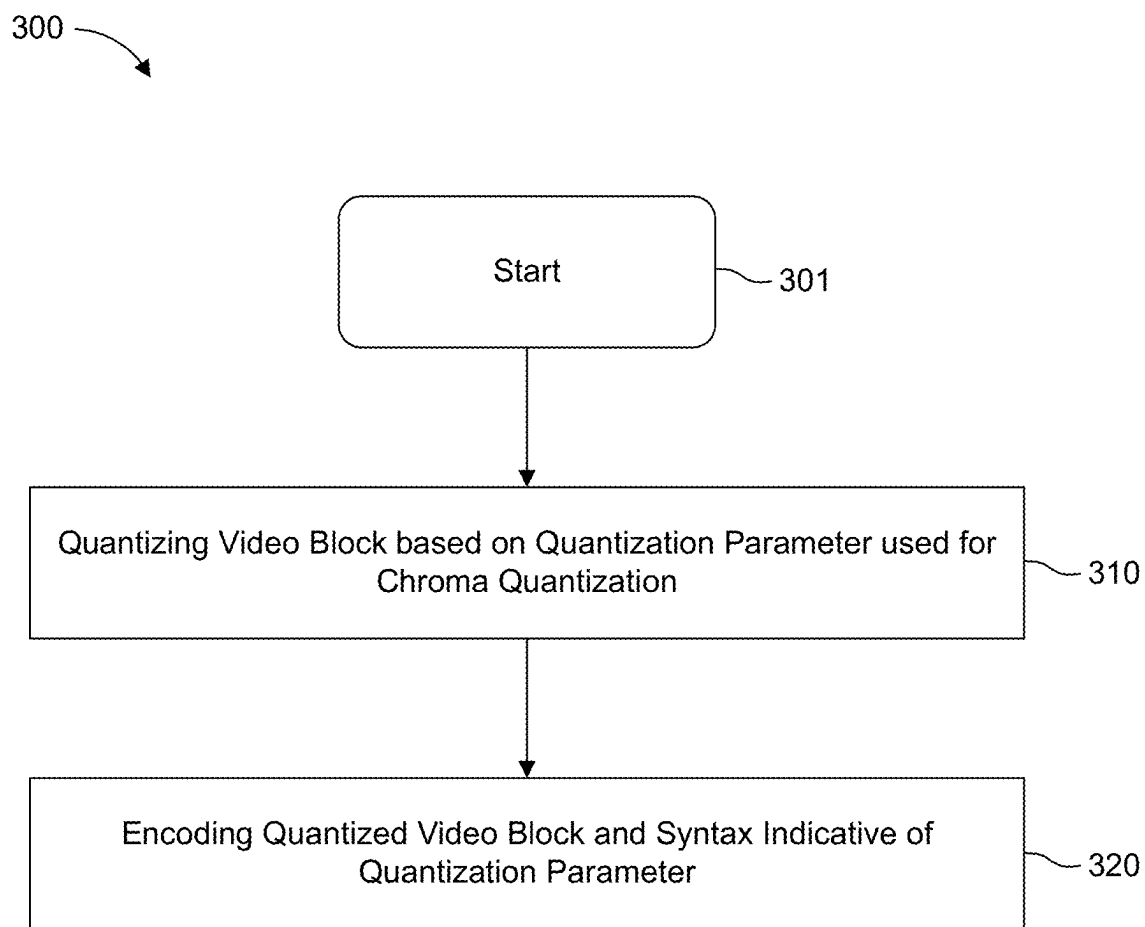


Figure 3

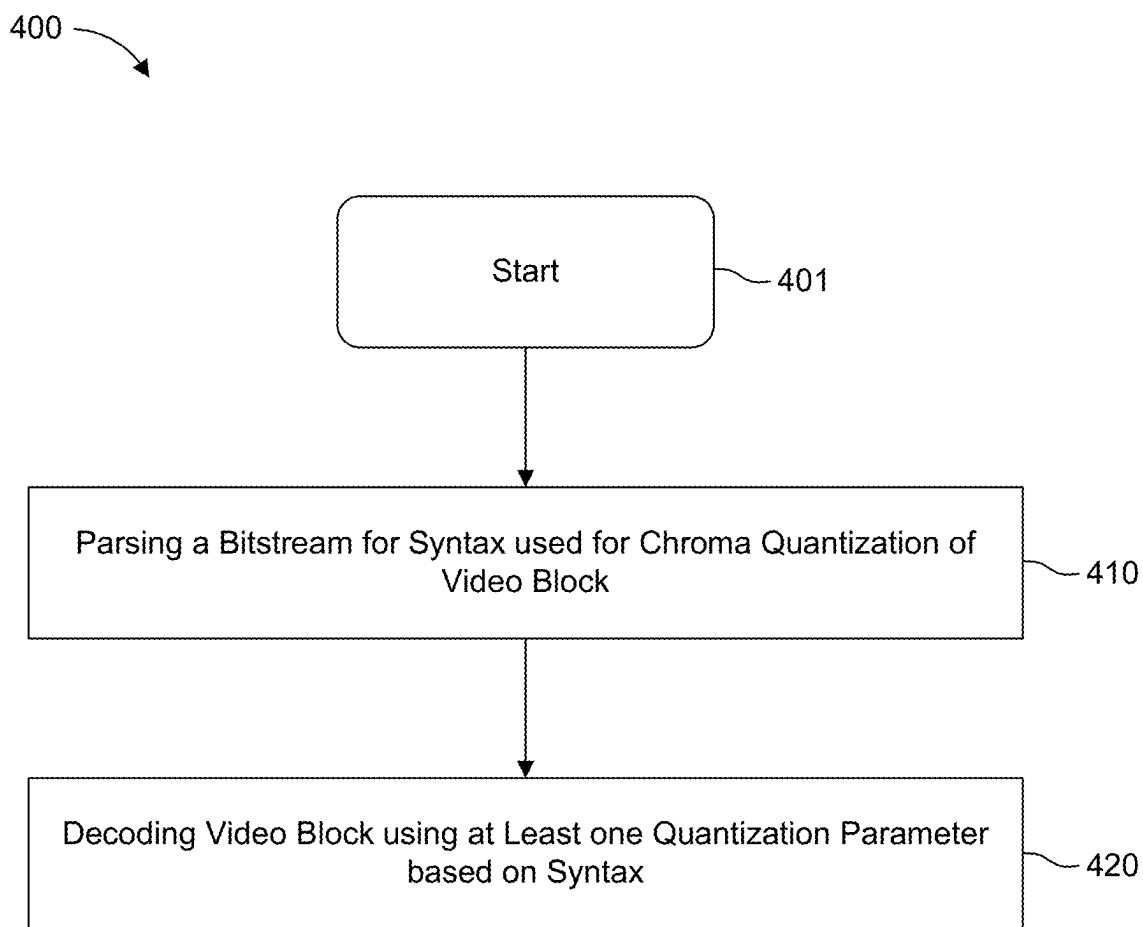


Figure 4

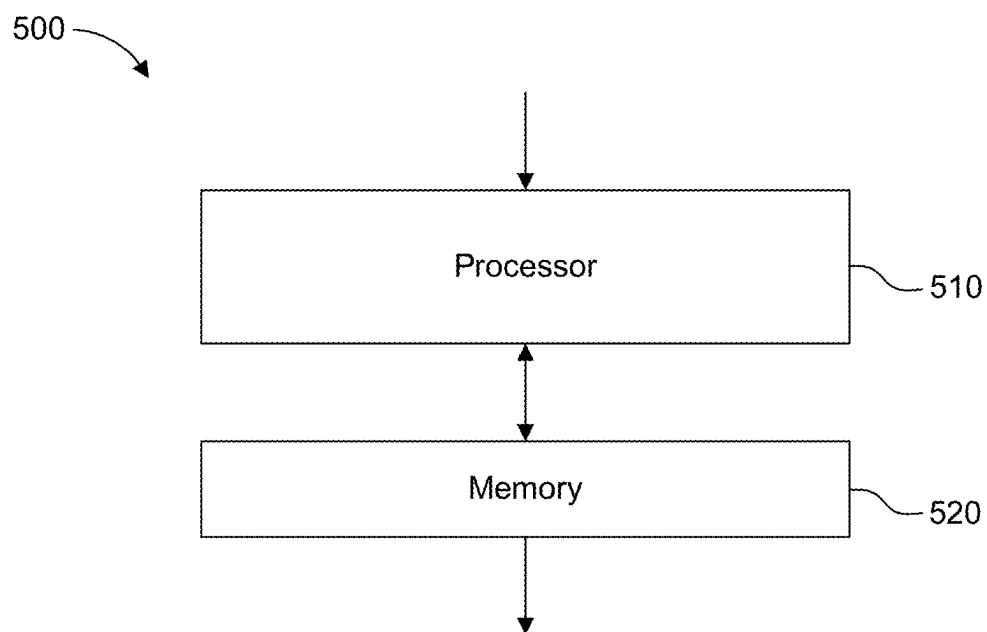


Figure 5

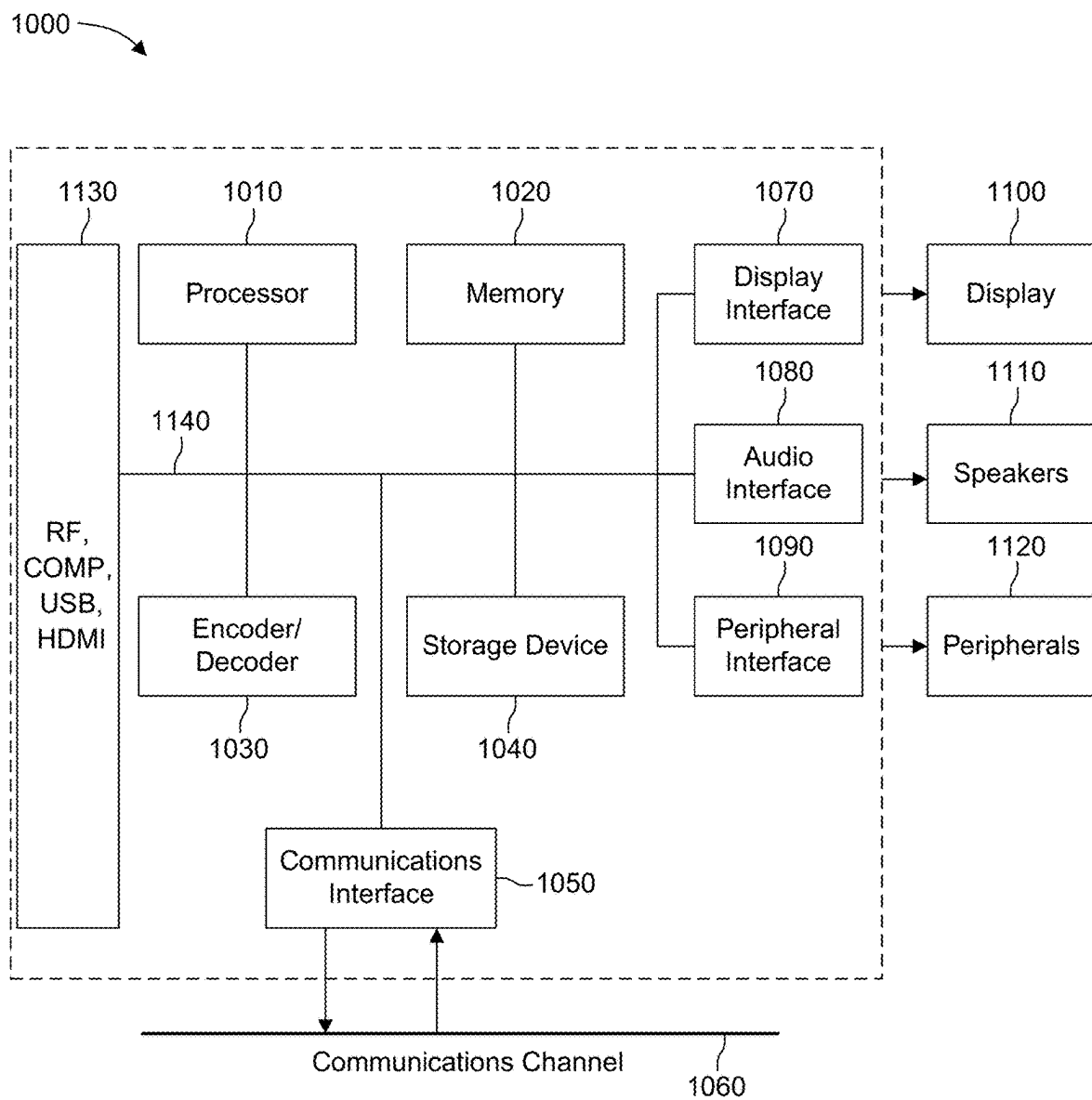


Figure 6

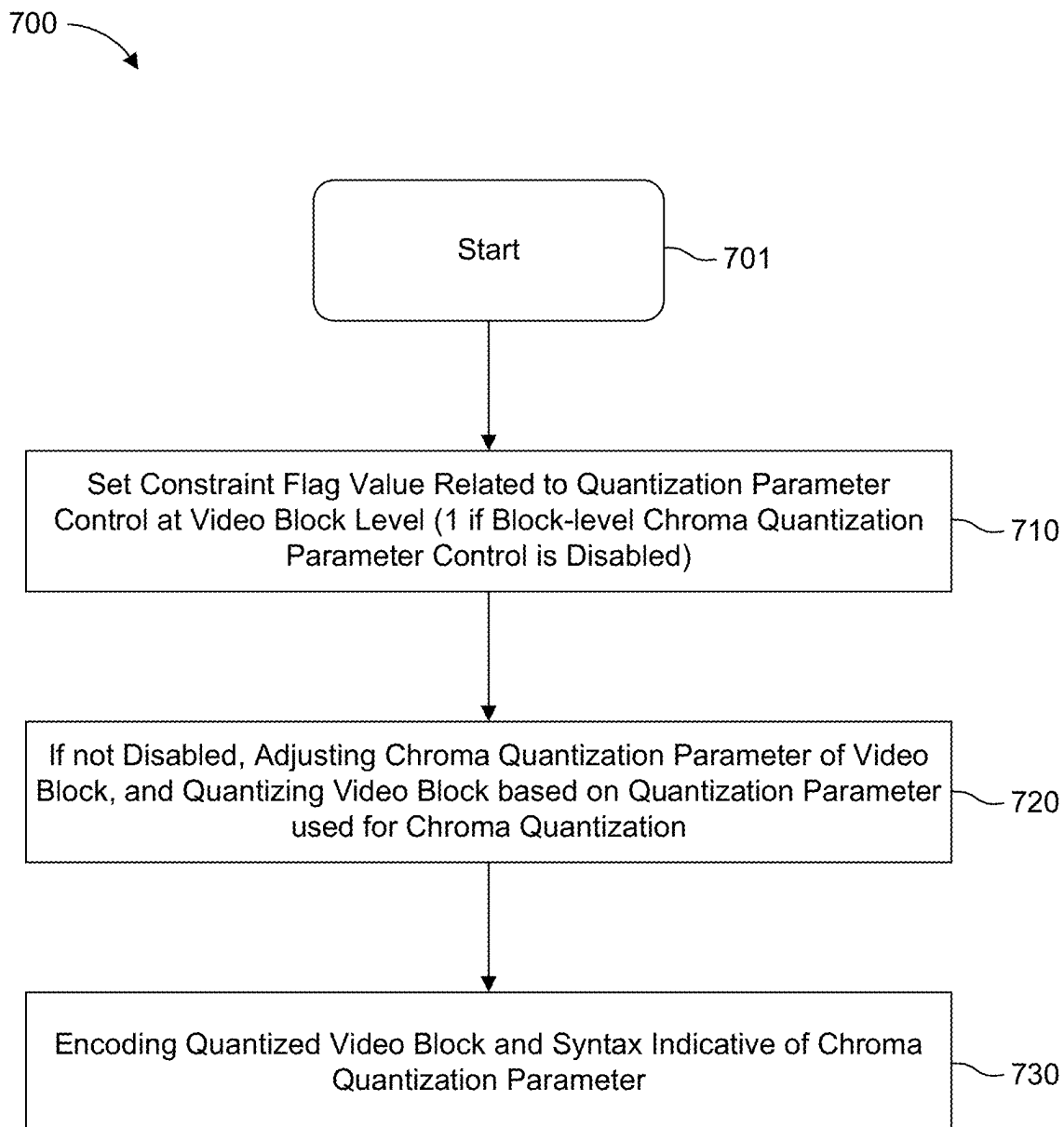


Figure 7



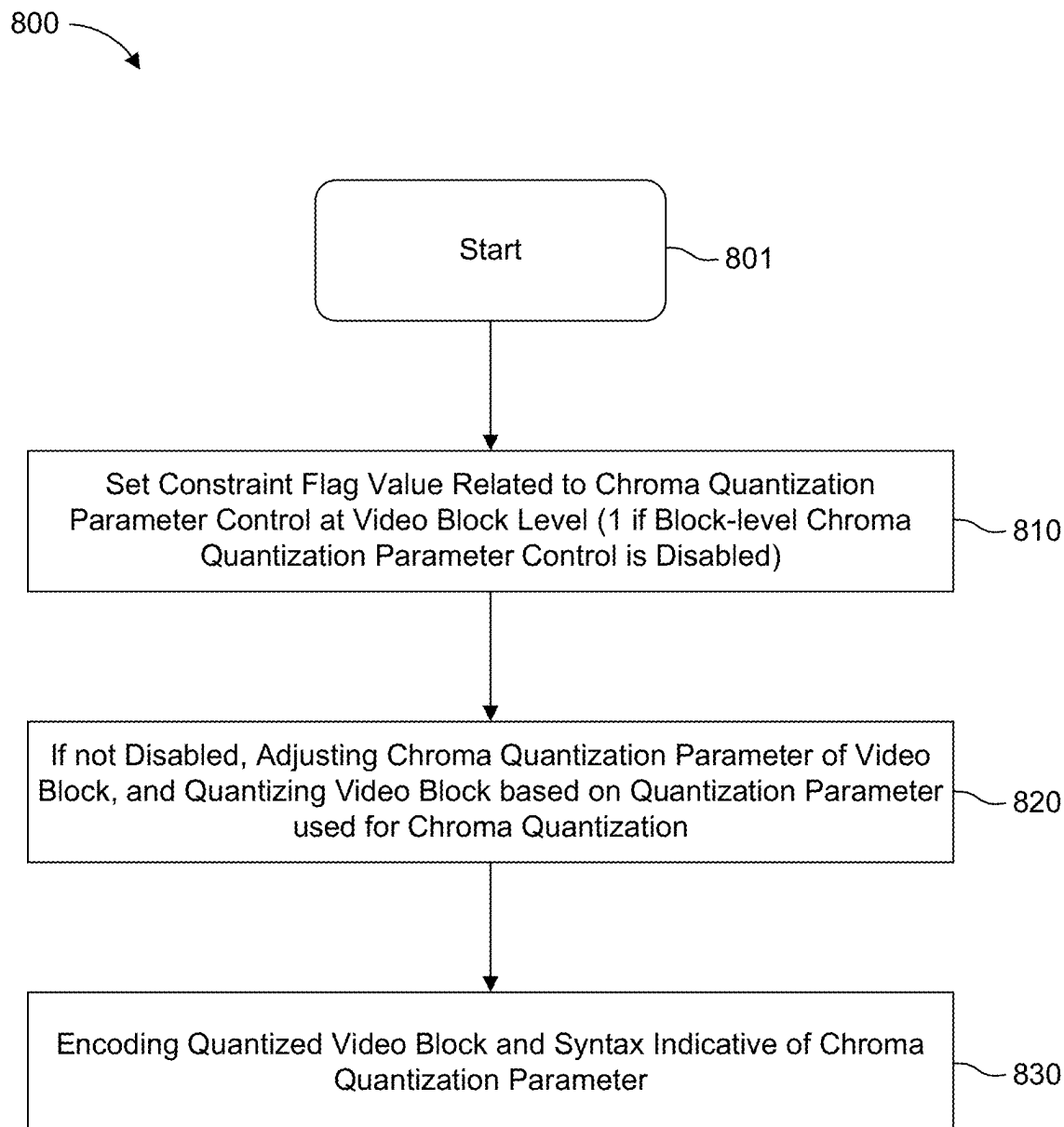


Figure 8

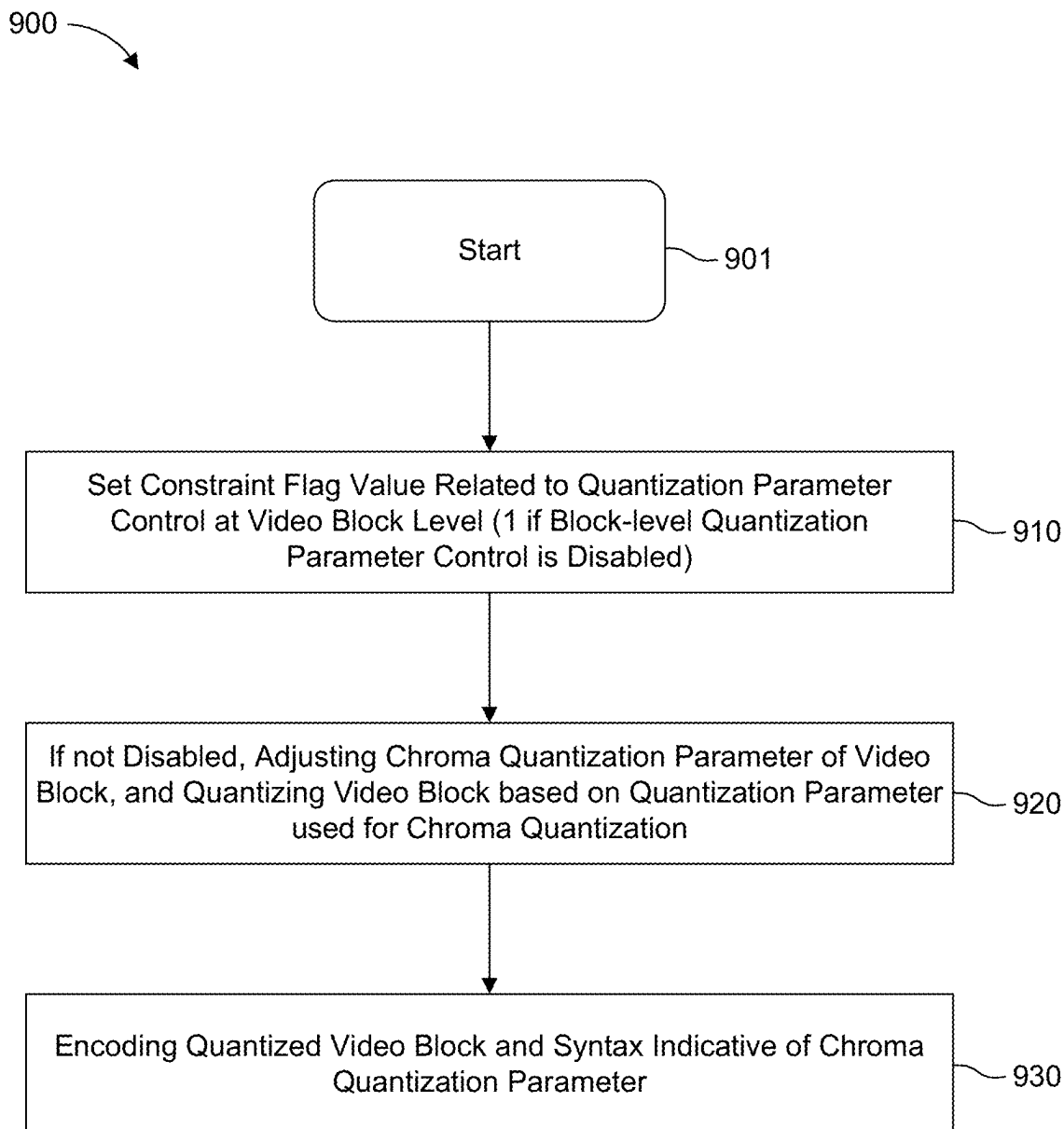


Figure 9

## HIGH-LEVEL CONSTRAINT FLAG FOR LOCAL CHROMA QUANTIZATION PARAMETER CONTROL

### TECHNICAL FIELD

[0001] At least one of the present embodiments generally relates to a method or an apparatus for video encoding or decoding.

### BACKGROUND

[0002] To achieve high compression efficiency, image and video coding schemes usually employ prediction, including spatial and/or motion vector prediction, and transforms to leverage spatial and temporal redundancy in the video content. Generally, intra or inter prediction is used to exploit the intra or inter frame correlation, then the differences between the original image and the predicted image, often denoted as prediction errors or prediction residuals, are transformed, quantized, and entropy coded. To reconstruct the video, the compressed data are decoded by inverse processes corresponding to the entropy coding, quantization, transform, and prediction. A number of coding tools can be used in the process of coding and decoding, including transforms and inverse transforms.

### SUMMARY

[0003] Drawbacks and disadvantages of the prior art may be addressed by the general aspects described herein, which are directed to constraint flags used to build profiles and/or chroma quantization control.

[0004] According to a first aspect, there is provided a method. The method comprises steps for quantizing at least one video block based on at least one quantization parameter that is used for chroma quantization; and, encoding said quantized at least one video block and syntax indicative of said at least one quantization parameter.

[0005] According to a second aspect, there is provided a method. The method comprises steps for parsing a video bitstream for syntax used to control at least one quantization parameter that is used for chroma quantization for at least one coded video block; and, decoding said at least one video block using the at least one quantization parameter based on the syntax

[0006] According to another aspect, there is provided an apparatus. The apparatus comprises a processor. The processor can be configured to encode a block of a video or decode a bitstream by executing any of the aforementioned methods.

[0007] According to another general aspect of at least one embodiment, there is provided a device comprising an apparatus according to any of the decoding embodiments; and at least one of (i) an antenna configured to receive a signal, the signal including the video block, (ii) a band limiter configured to limit the received signal to a band of frequencies that includes the video block, or (iii) a display configured to display an output representative of a video block.

[0008] According to another general aspect of at least one embodiment, there is provided a non-transitory computer readable medium containing data content generated according to any of the described encoding embodiments or variants.

[0009] According to another general aspect of at least one embodiment, there is provided a signal comprising video data generated according to any of the described encoding embodiments or variants.

[0010] According to another general aspect of at least one embodiment, a bitstream is formatted to include data content generated according to any of the described encoding embodiments or variants.

[0011] According to another general aspect of at least one embodiment, there is provided a computer program product comprising instructions which, when the program is executed by a computer, cause the computer to carry out any of the described decoding embodiments or variants.

[0012] These and other aspects, features and advantages of the general aspects will become apparent from the following detailed description of exemplary embodiments, which is to be read in connection with the accompanying drawings.

### BRIEF DESCRIPTION OF THE DRAWINGS

[0013] FIG. 1 shows a standard, generic video compression scheme. FIG. 2 shows a standard, generic, video compression scheme.

[0014] FIG. 3 shows one embodiment of an encoding method under the general aspects for encoding with control of a local chroma quantization parameter.

[0015] FIG. 4 shows another embodiment of a decoding method under the general aspects for decoding with control of a local chroma quantization parameter.

[0016] FIG. 5 shows one embodiment of an apparatus for encoding or decoding using the described aspects.

[0017] FIG. 6 shows a typical processor arrangement in which the described embodiments may be implemented.

[0018] FIG. 7 shows another embodiment of a method under the current aspects.

[0019] FIG. 8 shows another embodiment of a method under the current aspects.

[0020] FIG. 9 shows another embodiment of a method under the current aspects.

### DETAILED DESCRIPTION

[0021] To achieve high compression efficiency, image and video coding schemes usually employ prediction, including motion vector prediction, and transformations to leverage spatial and temporal redundancy in the video content. Generally, intra or inter prediction is used to exploit the intra or inter frame correlation, then the differences between the original image and the predicted image, often denoted as prediction errors or prediction residuals, are transformed, quantized, and entropy coded. To reconstruct the video, the compressed data are decoded by inverse processes corresponding to the entropy coding, quantization, transform, and prediction.

[0022] The following general aspects are in the field of video compression, more specifically the high-level syntax provided to disable local control of the quantization step of the video compression scheme. Video compression standards, such as High-Efficiency Video Coding (HEVC) and Versatile Video Coding (VVC) use quantization with quantization step sizes that are used to regulate bit-rate and/or video quality. The HEVC and VVC (draft 8) specifications make use of a quantization parameter QP (or qP) to define a current quantization step.

**[0023]** The HEVC specification uses a dequantization process where coded block frequency-transformed coefficients *TransCoeffLevel* are scaled by the current quantization step (*levelScale* [*qP %6*] << (*qP/6*)) and further scaled by a quantization matrix *m* as follows:

$$d[x][y] = \text{Clip3}(\text{coeffMin}, \text{coeffMax}, ((\text{TransCoeffLevel}[xTbY][yTbY][cIdx][x][y] * m[x][y] * \text{levelScale}[qP \%6] \ll (qP/6)) + (1 \ll (bdShift - 1))) \gg bdShift)$$

Where:

**[0024]** *TransCoeffLevel*[ . . . ] are the transformed coefficients' absolute values for the current block identified by its spatial coordinates *xTbY*, *yTbY* and its component index *cIdx*.

**[0025]** *x* and *y* are the horizontal/vertical frequency indices.

**[0026]** *qP* is the current quantization parameter.

**[0027]** the multiplication by *levelScale* [*qP %6*] and left shift by (*qP/6*) is the equivalent of the multiplication by quantization step *qStep*=(*levelScale* [*qP %6*] << (*qP/6*))

**[0028]** *m*[ . . . ][ . . . ] is the two-dimensional quantization matrix

**[0029]** *bdShift* is an additional scaling factor to account for image sample bit depth. The term ( $1 \ll (bdShift - 1)$ ) serves the purpose of rounding to the nearest integer.

**[0030]** *d*[ . . . ] are the resulting dequantized transformed coefficients absolute values.

**[0031]** In HEVC (with range extension) and VVC, it is possible to adjust the QP locally, using specific syntax. A luma QP offset can be specified, and, optionally, a chroma QP offset can also be specified. Related syntax present in the VVC transform unit syntax structure is illustrated below:

transform_unit( x0, y0, tbWidth, tbHeight, treeType, subTuIndex, chType ) {	Descriptor
[... (cbf flags)]	
if( ( CbWidth[ chType ][ x0 ][ y0 ] > 64    CbHeight[ chType ][ x0 ][ y0 ] > 64    tu_cbf_luma[ x0 ][ y0 ]    ( chromaAvailable && ( tu_cbf_cb[ xC ][ yC ]    tu_cbf_cr[ xC ][ yC ] ) ) && treeType != DUAL_TREE_CHROMA && cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) {	
<b>cu_qp_delta_abs</b>	ae(v)
if( cu_qp_delta_abs )	
<b>cu_qp_delta_sign_flag</b>	ae(v)
}	
if( ( CbWidth[ chType ][ x0 ][ y0 ] > 64    CbHeight[ chType ][ x0 ][ y0 ] > 64    ( chromaAvailable && ( tu_cbf_cb[ xC ][ yC ]    tu_cbf_cr[ xC ][ yC ] ) ) ) && treeType != DUAL_TREE_LUMA && cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded ) {	
<b>cu_chroma_qp_offset_flag</b>	ae(v)
if( cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0 )	
<b>cu_chroma_qp_offset_idx</b>	ae(v)
}	
[...] (other flags, residuals)]	
}	

*cu\_qp\_delta\_abs* and *cu\_qp\_delta\_sign\_flag* are used to compute *CuQpDeltaVal*, which is the difference between the luma QP of the current coding unit (CU) and the luma QP prediction. The luma QP prediction is based on the luma QP of blocks on top and left of the current Quantization Group, which will not be discussed here.

*cu\_chroma\_qp\_offset\_flag* and *cu\_chroma\_qp\_offset\_idx* are used to derive chroma QP offsets. *cu\_chroma\_qp\_offset\_flag* enables the offset feature (otherwise offsets are zero), and *cu\_chroma\_qp\_offset\_idx* specifies an index in a lookup table, where offsets for every chroma component is found for every index, as illustrated below:

Index	Cb QP offset	Cr QP offset	Joint CbCr QP offset
0	1	0	1
1	-1	-1	0
[ . . . ]	[ . . . ]	[ . . . ]	[ . . . ]

**[0032]** In VVC, a syntax structure called “general\_constraint\_info” (referred to by the *profile\_tier\_level* syntax structure) contains a number of flags that can be used to disable some coding tools. This can be used to disable some tools in a given profile, so that decoders compliant with this profile do not need to implement those disabled tools.

**[0033]** For example, this makes it possible to specify low-complexity decoders, compliant with a subset of VVC.

**[0034]** The syntax and semantics of the constraint flag related to local QP control is shown below:

general_constraint_info( ) {	Descriptor
[...]	
<b>no_transform_skip_constraint_flag</b>	u(1)
<b>no_bdpcm_constraint_flag</b>	u(1)
<b>no_qp_delta_constraint_flag</b>	u(1)
[...]	
}	

**[0035]** *no\_qp\_delta\_constraint\_flag* equal to 1 specifies that it is a requirement of bitstream *cu\_qp\_delta\_*

*enabled\_flag* shall be equal to 0. *no\_qp\_delta\_constraint\_flag* equal to 0 does not impose such a constraint.

**[0036]** Note: *cu\_qp\_delta\_enabled\_flag* is found in the PPS syntax structure, and conditions all syntax related to

local luma QP control. Forcing it to zero means that no local luma QP control syntax is present in the bitstream, disabling the functionality.

**[0037]** In current VVC draft, local chroma QP control cannot be disabled with a constraint flag. This prevents defining profiles without local chroma QP control.

**[0038]** At least one embodiment disclosed herein enables disabling of local chroma QP control with a high-level constraint flag.

Method 1: Use No\_Qp\_Delta\_Constraint\_Flag to Disable Both Luma and Chroma Local QP Control

**[0039]** One method to disable local chroma QP control with a high-level constraint flag is to reuse the flag designed to disable local luma QP control, and make it disable both luma and chroma local QP control. This approach can be used in an encoder only (conformance).

**[0040]** For an encoder only embodiment, syntax is unchanged for a decoder and ignores the value of constraint flag (except potentially to check conformance with supported profile, and crash otherwise; can also detect and crash on invalid PPS flags). An encoder has to comply with conformance constraint by actually disabling chroma QP control when the constraint flag is 1.

**[0041]** Example syntax and semantics in the context of VVC draft 8 are shown below.

Syntax (Unchanged):

general_constraint_info( ) {	Descriptor
[...]	
<b>no_transform_skip_constraint_flag</b>	u(1)
<b>no_bdpcm_constraint_flag</b>	u(1)
<b>no_qp_delta_constraint_flag</b>	u(1)
[...]	
}	

Semantics (with Additions Underlined):

**[0042]** no\_qp\_delta\_constraint\_flag equal to 1 specifies that it is a requirement of bitstream conformance that both cu\_qp\_delta\_enabled\_flag and pps\_cu\_chroma\_qp\_offset\_list\_enabled\_flag shall be equal to 0. no\_qp\_delta\_constraint\_flag equal to 0 does not impose such a constraint.

Note: pps\_cu\_chroma\_qp\_offset\_list\_enabled\_flag is found in the PPS syntax structure, and conditions all syntax related to local chroma QP control. Forcing it to zero means that no local chroma QP control syntax is present in the bitstream, disabling the functionality.

Method 2: Add a Specific Constraint Flag to Disable Local Chroma QP Control

**[0043]** Another method to disable local chroma QP control with a high-level constraint flag is to add a specific constraint flag. This could be useful to separately control the functionalities in different profiles. For example, in HEVC, local luma QP control is in the core specification but local chroma QP control is in the range extension, which means that in HEVC without range extension, local luma QP control is possible but there is no local chroma QP control. Decoder has to parse the new conformance flag but ignores its value.

**[0044]** An encoder has to comply with conformance constraint by actually disabling chroma QP control when the constraint flag is 1.

**[0045]** Example syntax and semantics in the context of VVC draft 8 are shown below:

Syntax (Additions Underlined):

general_constraint_info( ) {	Descriptor
[...]	
<b>no_transform_skip_constraint_flag</b>	u(1)
<b>no_bdpcm_constraint_flag</b>	u(1)
<b>no_qp_delta_constraint_flag</b>	u(1)
<b><u>no_chroma_qp_offsets_constraint_flag</u></b>	<u>u(1)</u>
[...]	
}	

Semantics (Additions Underlined):

**[0046]** no\_qp\_delta\_constraint\_flag equal to 1 specifies that it is a requirement of bitstream conformance that cu\_qp\_delta\_enabled\_flag shall be equal to 0. no\_qp\_delta\_constraint\_flag equal to 0 does not impose such a constraint.

**[0047]** no\_chroma\_qp\_offset\_constraint\_flag equal to 1 specifies that it is a requirement of bitstream conformance that pps\_cu\_chroma\_qp\_offset\_list\_enabled\_flag shall be equal to 0. no\_chroma\_qp\_offset\_constraint\_flag equal to 0 does not impose such a constraint.

Method 3: Make Chroma QP Control Depend on Use of Luma QP Control

**[0048]** Another method to disable local chroma QP control with a high-level constraint flag is to use the flag designed to disable local luma QP control, and condition chroma QP control syntax to activation of luma QP control (at SPS level).

**[0049]** Example syntax and semantics in the context of VVC draft 8 are shown below:

Syntax (Additions Underlined):

pic_parameter_set_rbsp( ) {	Descriptor
[...]	
<b>cu_qp_delta_enabled_flag</b>	u(1)
<b>pps_chroma_tool_offsets_present_flag</b>	u(1)
if( pps_chroma_tool_offsets_present_flag ) {	
<b>pps_cb_qp_offset</b>	se(v)
<b>pps_cr_qp_offset</b>	se(v)
<b>pps_joint_cbr_qp_offset_present_flag</b>	u(1)
if( pps_joint_cbr_qp_offset_present_flag )	
<b>pps_joint_cbr_qp_offset_value</b>	se(v)
<b>pps_slice_chroma_qp_offsets_present_flag</b>	u(1)
if( <u>cu_qp_delta_enabled_flag</u> )	
<b>pps_cu_chroma_qp_offset_list_enabled_flag</b>	u(1)
}	
[...]	
}	

Semantics: Unchanged

Method 4: Condition Both Luma and Chroma QP Control to an Additional Flag

**[0050]** Another method to disable local chroma QP control with a high-level constraint flag is to use a single constraint

flag, that disables a new flag (at PPS level), with that new flag conditioning all QP control related syntax (both luma and chroma).

[0051] Example syntax and semantics in the context of VVC draft 8 are shown below:

Syntax (Deletions Strikethrough, Additions Underlined):

general_constraint_info( ) {	Descriptor
[...]	
<u>no_transform_skip_constraint_flag</u>	u(1)
<u>no_bdpcm_constraint_flag</u>	u(1)
<u>no_cu_qp_constraint_flag</u>	u(1)
[...]	
}	

pic_parameter_set_rbsp( ) {	Descriptor
[...]	
<u>cu_qp_enabled_flag</u>	u(1)
if (cu_qp_enabled_flag)	
<u>cu_qp_delta_enabled_flag</u>	u(1)
<u>pps_chroma_tool_offsets_present_flag</u>	u(1)
if( pps_chroma_tool_offsets_present_flag ) {	
<u>pps_cb_qp_offset</u>	se(v)
<u>pps_cr_qp_offset</u>	se(v)
<u>pps_joint_cbr_qp_offset_present_flag</u>	u(1)
if( pps_joint_cbr_qp_offset_present_flag )	
<u>pps_joint_cbr_qp_offset_value</u>	se(v)
<u>pps_slice_chroma_qp_offsets_present_flag</u>	u(1)
if (cu_qp_enabled_flag)	
<u>pps_cu_chroma_qp_offset_list_enabled_flag</u>	u(1)
}	
[...]	
}	

Semantics (Deletions Strikethrough, Additions Underlined):

[0052] ~~no\_qp\_delta\_constraint\_flag~~ ~~no\_cu\_qp\_constraint\_flag~~ equal to 1 specifies that it is a requirement of bitstream conformance that ~~cu\_qp\_delta\_enabled\_flag~~ cu\_qp\_delta\_enabled\_flag shall be equal to 0. ~~no\_qp\_delta\_constraint\_flag~~ ~~no\_cu\_qp\_constraint\_flag~~ equal to 0 does not impose such a constraint.

[0053] cu\_qp\_enabled\_flag equal to 1 specifies that the cu\_qp\_delta\_enabled\_flag and pps\_cu\_chroma\_qp\_offset\_list\_enabled\_flag syntax elements are present in the PPS. cu\_qp\_enabled\_flag equal to 0 specifies that the cu\_qp\_delta\_enabled\_flag and pps\_cu\_chroma\_qp\_offset\_list\_enabled\_flag syntax elements are not present in the PPS.

[0054] Note: these examples are not limitative. Although in these examples, the flags that are disabled by constraints flags are present in the PPS syntax structure, the method described here would still be valid if those flags (potentially disabled by constraint flags) were moved to an other syntax structure like SPS, or if the constraint flag related to local chroma QP control was constraining (disabling) flags other than pps\_cu\_chroma\_qp\_offset\_list\_enabled\_flag, potentially in syntax structures other than PPS (e.g. SPS).

[0055] One embodiment of a method 300 under the general aspects described here is shown in FIG. 3. The method commences at start block 301 and control proceeds to block 310 for quantizing at least one video block based on at least

one quantization parameter that is used for chroma quantization. Control proceeds from block 310 to block 320 for encoding the quantized at least one video block and syntax indicative of the at least one quantization parameter.

[0056] One embodiment of a method 400 under the general aspects described here is shown in FIG. 4. The method commences at start block 401 and control proceeds to block

410 for parsing a video bitstream for syntax used to control at least one quantization parameter that is used for chroma quantization for at least one coded video block. Control proceeds from block 410 to block 420 for decoding the at least one video block using the at least one quantization parameter based on the syntax.

[0057] FIG. 5 shows one embodiment of an apparatus 500 for compressing, encoding or decoding video with ability to disable local chroma quantization parameter control. The apparatus comprises Processor 510 and can be interconnected to a memory 520 through at least one port. Both Processor 510 and memory 520 can also have one or more additional interconnections to external connections.

[0058] Processor 510 is also configured to either insert or receive information in a bitstream and, either compressing, encoding or decoding using various coding tools.

[0059] This application describes a variety of aspects, including tools, features, embodiments, models, approaches, etc. Many of these aspects are described with specificity and, at least to show the individual characteristics, are often described in a manner that may sound limiting. However, this is for purposes of clarity in description, and does not limit the application or scope of those aspects. Indeed, all of the different aspects can be combined and interchanged to provide further aspects. Moreover, the aspects can be combined and interchanged with aspects described in earlier filings as well.

[0060] The aspects described and contemplated in this application can be implemented in many different forms. FIGS. 1, 2, and 6 provide some embodiments, but other embodiments are contemplated and the discussion of FIGS. 1, 2, and 6 does not limit the breadth of the implementations. At least one of the aspects generally relates to video encoding and decoding, and at least one other aspect generally relates to transmitting a bitstream generated or encoded. These and other aspects can be implemented as a method, an apparatus, a computer readable storage medium having stored thereon instructions for encoding or decoding video data according to any of the methods described, and/or a computer readable storage medium having stored thereon a bitstream generated according to any of the methods described.

[0061] In the present application, the terms “reconstructed” and “decoded” may be used interchangeably, the terms “pixel” and “sample” may be used interchangeably, the terms “image,” “picture” and “frame” may be used

interchangeably. Usually, but not necessarily, the term “reconstructed” is used at the encoder side while “decoded” is used at the decoder side.

**[0062]** Various methods are described herein, and each of the methods comprises one or more steps or actions for achieving the described method. Unless a specific order of steps or actions is required for proper operation of the method, the order and/or use of specific steps and/or actions may be modified or combined.

**[0063]** Various methods and other aspects described in this application can be used to modify modules, for example, the intra prediction, entropy coding, and/or decoding modules (160, 360, 145, 330), of a video encoder 100 and decoder 200 as shown in FIG. 1 and FIG. 2. Moreover, the present aspects are not limited to VVC or HEVC, and can be applied, for example, to other standards and recommendations, whether pre-existing or future-developed, and extensions of any such standards and recommendations (including VVC and HEVC). Unless indicated otherwise, or technically precluded, the aspects described in this application can be used individually or in combination.

**[0064]** Various numeric values are used in the present application. The specific values are for example purposes and the aspects described are not limited to these specific values.

**[0065]** FIG. 1 illustrates an encoder 100. Variations of this encoder 100 are contemplated, but the encoder 100 is described below for purposes of clarity without describing all expected variations.

**[0066]** Before being encoded, the video sequence may go through pre-encoding processing (101), for example, applying a color transform to the input color picture (e.g., conversion from RGB 4:4:4 to YCbCr 4:2:0), or performing a remapping of the input picture components in order to get a signal distribution more resilient to compression (for instance using a histogram equalization of one of the color components). Metadata can be associated with the pre-processing and attached to the bitstream.

**[0067]** In the encoder 100, a picture is encoded by the encoder elements as described below. The picture to be encoded is partitioned (102) and processed in units of, for example, CUs. Each unit is encoded using, for example, either an intra or inter mode. When a unit is encoded in an intra mode, it performs intra prediction (160). In an inter mode, motion estimation (175) and compensation (170) are performed. The encoder decides (105) which one of the intra mode or inter mode to use for encoding the unit, and indicates the intra/inter decision by, for example, a prediction mode flag. Prediction residuals are calculated, for example, by subtracting (110) the predicted block from the original image block.

**[0068]** The prediction residuals are then transformed (125) and quantized (130). The quantized transform coefficients, as well as motion vectors and other syntax elements, are entropy coded (145) to output a bitstream. The encoder can skip the transform and apply quantization directly to the non-transformed residual signal. The encoder can bypass both transform and quantization, i.e., the residual is coded directly without the application of the transform or quantization processes.

**[0069]** The encoder decodes an encoded block to provide a reference for further predictions. The quantized transform coefficients are de-quantized (140) and inverse transformed (150) to decode prediction residuals. Combining (155) the

decoded prediction residuals and the predicted block, an image block is reconstructed. In-loop filters (165) are applied to the reconstructed picture to perform, for example, deblocking/SAO (Sample Adaptive Offset) filtering to reduce encoding artifacts. The filtered image is stored at a reference picture buffer (180).

**[0070]** FIG. 2 illustrates a block diagram of a video decoder 200. In the decoder 200, a bitstream is decoded by the decoder elements as described below. Video decoder 200 generally performs a decoding pass reciprocal to the encoding pass as described in FIG. 1. The encoder 100 also generally performs video decoding as part of encoding video data.

**[0071]** In particular, the input of the decoder includes a video bitstream, which can be generated by video encoder 100. The bitstream is first entropy decoded (230) to obtain transform coefficients, motion vectors, and other coded information. The picture partition information indicates how the picture is partitioned. The decoder may therefore divide (235) the picture according to the decoded picture partitioning information. The transform coefficients are de-quantized (240) and inverse transformed (250) to decode the prediction residuals. Combining (255) the decoded prediction residuals and the predicted block, an image block is reconstructed. The predicted block can be obtained (270) from intra prediction (260) or motion-compensated prediction (i.e., inter prediction) (275). In-loop filters (265) are applied to the reconstructed image. The filtered image is stored at a reference picture buffer (280).

**[0072]** The decoded picture can further go through post-decoding processing (285), for example, an inverse color transform (e.g. conversion from YCbCr 4:2:0 to RGB 4:4:4) or an inverse remapping performing the inverse of the remapping process performed in the pre-encoding processing (101). The post-decoding processing can use metadata derived in the pre-encoding processing and signaled in the bitstream.

**[0073]** FIG. 6 illustrates a block diagram of an example of a system in which various aspects and embodiments are implemented. System 1000 can be embodied as a device including the various components described below and is configured to perform one or more of the aspects described in this document. Examples of such devices include, but are not limited to, various electronic devices such as personal computers, laptop computers, smartphones, tablet computers, digital multimedia set top boxes, digital television receivers, personal video recording systems, connected home appliances, and servers. Elements of system 1000, singly or in combination, can be embodied in a single integrated circuit (IC), multiple ICs, and/or discrete components. For example, in at least one embodiment, the processing and encoder/decoder elements of system 1000 are distributed across multiple ICs and/or discrete components. In various embodiments, the system 1000 is communicatively coupled to one or more other systems, or other electronic devices, via, for example, a communications bus or through dedicated input and/or output ports. In various embodiments, the system 1000 is configured to implement one or more of the aspects described in this document.

**[0074]** The system 1000 includes at least one processor 1010 configured to execute instructions loaded therein for implementing, for example, the various aspects described in this document. Processor 1010 can include embedded memory, input output interface, and various other circuitries

as known in the art. The system **1000** includes at least one memory **1020** (e.g., a volatile memory device, and/or a non-volatile memory device). System **1000** includes a storage device **1040**, which can include non-volatile memory and/or volatile memory, including, but not limited to, Electrically Erasable Programmable Read-Only Memory (EEPROM), Read-Only Memory (ROM), Programmable Read-Only Memory (PROM), Random Access Memory (RAM), Dynamic Random Access Memory (DRAM), Static Random Access Memory (SRAM), flash, magnetic disk drive, and/or optical disk drive. The storage device **1040** can include an internal storage device, an attached storage device (including detachable and non-detachable storage devices), and/or a network accessible storage device, as non-limiting examples.

**[0075]** System **1000** includes an encoder/decoder module **1030** configured, for example, to process data to provide an encoded video or decoded video, and the encoder/decoder module **1030** can include its own processor and memory. The encoder/decoder module **1030** represents module(s) that can be included in a device to perform the encoding and/or decoding functions. As is known, a device can include one or both of the encoding and decoding modules. Additionally, encoder/decoder module **1030** can be implemented as a separate element of system **1000** or can be incorporated within processor **1010** as a combination of hardware and software as known to those skilled in the art.

**[0076]** Program code to be loaded onto processor **1010** or encoder/decoder **1030** to perform the various aspects described in this document can be stored in storage device **1040** and subsequently loaded onto memory **1020** for execution by processor **1010**. In accordance with various embodiments, one or more of processor **1010**, memory **1020**, storage device **1040**, and encoder/decoder module **1030** can store one or more of various items during the performance of the processes described in this document. Such stored items can include, but are not limited to, the input video, the decoded video or portions of the decoded video, the bitstream, matrices, variables, and intermediate or final results from the processing of equations, formulas, operations, and operational logic.

**[0077]** In some embodiments, memory inside of the processor **1010** and/or the encoder/decoder module **1030** is used to store instructions and to provide working memory for processing that is needed during encoding or decoding. In other embodiments, however, a memory external to the processing device (for example, the processing device can be either the processor **1010** or the encoder/decoder module **1030**) is used for one or more of these functions. The external memory can be the memory **1020** and/or the storage device **1040**, for example, a dynamic volatile memory and/or a non-volatile flash memory. In several embodiments, an external non-volatile flash memory is used to store the operating system of, for example, a television. In at least one embodiment, a fast external dynamic volatile memory such as a RAM is used as working memory for video coding and decoding operations, such as for MPEG-2 (MPEG refers to the Moving Picture Experts Group, MPEG-2 is also referred to as ISO/IEC 13818, and 13818-1 is also known as H.222, and 13818-2 is also known as H.262), HEVC (HEVC refers to High Efficiency Video Coding, also known as H.265 and MPEG-H Part 2), or VVC (Versatile Video Coding, a new standard being developed by JVET, the Joint Video Experts Team).

**[0078]** The input to the elements of system **1000** can be provided through various input devices as indicated in block **1130**. Such input devices include, but are not limited to, (i) a radio frequency (RF) portion that receives an RF signal transmitted, for example, over the air by a broadcaster, (ii) a Component (COMP) input terminal (or a set of COMP input terminals), (iii) a Universal Serial Bus (USB) input terminal, and/or (iv) a High Definition Multimedia Interface (HDMI) input terminal. Other examples, not shown in FIG. 6, include composite video.

**[0079]** In various embodiments, the input devices of block **1130** have associated respective input processing elements as known in the art. For example, the RF portion can be associated with elements suitable for (i) selecting a desired frequency (also referred to as selecting a signal, or band-limiting a signal to a band of frequencies), (ii) downconverting the selected signal, (iii) band-limiting again to a narrower band of frequencies to select (for example) a signal frequency band which can be referred to as a channel in certain embodiments, (iv) demodulating the downconverted and band-limited signal, (v) performing error correction, and (vi) demultiplexing to select the desired stream of data packets. The RF portion of various embodiments includes one or more elements to perform these functions, for example, frequency selectors, signal selectors, band-limiters, channel selectors, filters, downconverters, demodulators, error correctors, and demultiplexers. The RF portion can include a tuner that performs various of these functions, including, for example, downconverting the received signal to a lower frequency (for example, an intermediate frequency or a near-baseband frequency) or to baseband. In one set-top box embodiment, the RF portion and its associated input processing element receives an RF signal transmitted over a wired (for example, cable) medium, and performs frequency selection by filtering, downconverting, and filtering again to a desired frequency band. Various embodiments rearrange the order of the above-described (and other) elements, remove some of these elements, and/or add other elements performing similar or different functions. Adding elements can include inserting elements in between existing elements, such as, for example, inserting amplifiers and an analog-to-digital converter. In various embodiments, the RF portion includes an antenna.

**[0080]** Additionally, the USB and/or HDMI terminals can include respective interface processors for connecting system **1000** to other electronic devices across USB and/or HDMI connections. It is to be understood that various aspects of input processing, for example, Reed-Solomon error correction, can be implemented, for example, within a separate input processing IC or within processor **1010** as necessary. Similarly, aspects of USB or HDMI interface processing can be implemented within separate interface ICs or within processor **1010** as necessary. The demodulated, error corrected, and demultiplexed stream is provided to various processing elements, including, for example, processor **1010**, and encoder/decoder **1030** operating in combination with the memory and storage elements to process the datastream as necessary for presentation on an output device.

**[0081]** Various elements of system **1000** can be provided within an integrated housing. Within the integrated housing, the various elements can be interconnected and transmit data therebetween using suitable connection arrangement, for



example, an internal bus as known in the art, including the Inter-IC (I2C) bus, wiring, and printed circuit boards.

[0082] The system 1000 includes communication interface 1050 that enables communication with other devices via communication channel 1060. The communication interface 1050 can include, but is not limited to, a transceiver configured to transmit and to receive data over communication channel 1060. The communication interface 1050 can include, but is not limited to, a modem or network card and the communication channel 1060 can be implemented, for example, within a wired and/or a wireless medium.

[0083] Data is streamed, or otherwise provided, to the system 1000, in various embodiments, using a wireless network such as a Wi-Fi network, for example IEEE 802.11 (IEEE refers to the Institute of Electrical and Electronics Engineers). The Wi-Fi signal of these embodiments is received over the communications channel 1060 and the communications interface 1050 which are adapted for Wi-Fi communications. The communications channel 1060 of these embodiments is typically connected to an access point or router that provides access to external networks including the Internet for allowing streaming applications and other over-the-top communications. Other embodiments provide streamed data to the system 1000 using a set-top box that delivers the data over the HDMI connection of the input block 1130. Still other embodiments provide streamed data to the system 1000 using the RF connection of the input block 1130. As indicated above, various embodiments provide data in a non-streaming manner. Additionally, various embodiments use wireless networks other than Wi-Fi, for example a cellular network or a Bluetooth network.

[0084] The system 1000 can provide an output signal to various output devices, including a display 1100, speakers 1110, and other peripheral devices 1120. The display 1100 of various embodiments includes one or more of, for example, a touchscreen display, an organic light-emitting diode (OLED) display, a curved display, and/or a foldable display. The display 1100 can be for a television, a tablet, a laptop, a cell phone (mobile phone), or other device. The display 1100 can also be integrated with other components (for example, as in a smart phone), or separate (for example, an external monitor for a laptop). The other peripheral devices 1120 include, in various examples of embodiments, one or more of a stand-alone digital video disc (or digital versatile disc) (DVR, for both terms), a disk player, a stereo system, and/or a lighting system. Various embodiments use one or more peripheral devices 1120 that provide a function based on the output of the system 1000. For example, a disk player performs the function of playing the output of the system 1000.

[0085] In various embodiments, control signals are communicated between the system 1000 and the display 1100, speakers 1110, or other peripheral devices 1120 using signaling such as AV.Link, Consumer Electronics Control (CEC), or other communications protocols that enable device-to-device control with or without user intervention. The output devices can be communicatively coupled to system 1000 via dedicated connections through respective interfaces 1070, 1080, and 1090. Alternatively, the output devices can be connected to system 1000 using the communications channel 1060 via the communications interface 1050. The display 1100 and speakers 1110 can be integrated in a single unit with the other components of system 1000 in an electronic device such as, for example, a television. In

various embodiments, the display interface 1070 includes a display driver, such as, for example, a timing controller (T Con) chip.

[0086] The display 1100 and speaker 1110 can alternatively be separate from one or more of the other components, for example, if the RF portion of input 1130 is part of a separate set-top box. In various embodiments in which the display 1100 and speakers 1110 are external components, the output signal can be provided via dedicated output connections, including, for example, HDMI ports, USB ports, or COMP outputs.

[0087] The embodiments can be carried out by computer software implemented by the processor 1010 or by hardware, or by a combination of hardware and software. As a non-limiting example, the embodiments can be implemented by one or more integrated circuits. The memory 1020 can be of any type appropriate to the technical environment and can be implemented using any appropriate data storage technology, such as optical memory devices, magnetic memory devices, semiconductor-based memory devices, fixed memory, and removable memory, as non-limiting examples. The processor 1010 can be of any type appropriate to the technical environment, and can encompass one or more of microprocessors, general purpose computers, special purpose computers, and processors based on a multi-core architecture, as non-limiting examples.

[0088] Various implementations involve decoding. “Decoding”, as used in this application, can encompass all or part of the processes performed, for example, on a received encoded sequence to produce a final output suitable for display. In various embodiments, such processes include one or more of the processes typically performed by a decoder, for example, entropy decoding, inverse quantization, inverse transformation, and differential decoding. In various embodiments, such processes also, or alternatively, include processes performed by a decoder of various implementations described in this application.

[0089] As further examples, in one embodiment “decoding” refers only to entropy decoding, in another embodiment “decoding” refers only to differential decoding, and in another embodiment “decoding” refers to a combination of entropy decoding and differential decoding. Whether the phrase “decoding process” is intended to refer specifically to a subset of operations or generally to the broader decoding process will be clear based on the context of the specific descriptions and is believed to be well understood by those skilled in the art.

[0090] Various implementations involve encoding. In an analogous way to the above discussion about “decoding”, “encoding” as used in this application can encompass all or part of the processes performed, for example, on an input video sequence to produce an encoded bitstream. In various embodiments, such processes include one or more of the processes typically performed by an encoder, for example, partitioning, differential encoding, transformation, quantization, and entropy encoding. In various embodiments, such processes also, or alternatively, include processes performed by an encoder of various implementations described in this application.

[0091] As further examples, in one embodiment “encoding” refers only to entropy encoding, in another embodiment “encoding” refers only to differential encoding, and in another embodiment “encoding” refers to a combination of differential encoding and entropy encoding. Whether the

phrase “encoding process” is intended to refer specifically to a subset of operations or generally to the broader encoding process will be clear based on the context of the specific descriptions and is believed to be well understood by those skilled in the art.

**[0092]** Note that the syntax elements as used herein are descriptive terms. As such, they do not preclude the use of other syntax element names.

**[0093]** When a figure is presented as a flow diagram, it should be understood that it also provides a block diagram of a corresponding apparatus. Similarly, when a figure is presented as a block diagram, it should be understood that it also provides a flow diagram of a corresponding method/process.

**[0094]** Various embodiments may refer to parametric models or rate distortion optimization. In particular, during the encoding process, the balance or trade-off between the rate and distortion is usually considered, often given the constraints of computational complexity. It can be measured through a Rate Distortion Optimization (RDO) metric, or through Least Mean Square (LMS), Mean of Absolute Errors (MAE), or other such measurements. Rate distortion optimization is usually formulated as minimizing a rate distortion function, which is a weighted sum of the rate and of the distortion. There are different approaches to solve the rate distortion optimization problem. For example, the approaches may be based on an extensive testing of all encoding options, including all considered modes or coding parameters values, with a complete evaluation of their coding cost and related distortion of the reconstructed signal after coding and decoding. Faster approaches may also be used, to save encoding complexity, in particular with computation of an approximated distortion based on the prediction or the prediction residual signal, not the reconstructed one. Mix of these two approaches can also be used, such as by using an approximated distortion for only some of the possible encoding options, and a complete distortion for other encoding options. Other approaches only evaluate a subset of the possible encoding options. More generally, many approaches employ any of a variety of techniques to perform the optimization, but the optimization is not necessarily a complete evaluation of both the coding cost and related distortion.

**[0095]** The implementations and aspects described herein can be implemented in, for example, a method or a process, an apparatus, a software program, a data stream, or a signal. Even if only discussed in the context of a single form of implementation (for example, discussed only as a method), the implementation of features discussed can also be implemented in other forms (for example, an apparatus or program). An apparatus can be implemented in, for example, appropriate hardware, software, and firmware. The methods can be implemented in, for example, a processor, which refers to processing devices in general, including, for example, a computer, a microprocessor, an integrated circuit, or a programmable logic device. Processors also include communication devices, such as, for example, computers, cell phones, portable/personal digital assistants (“PDAs”), and other devices that facilitate communication of information between end-users.

**[0096]** Reference to “one embodiment” or “an embodiment” or “one implementation” or “an implementation”, as well as other variations thereof, means that a particular feature, structure, characteristic, and so forth described in

connection with the embodiment is included in at least one embodiment. Thus, the appearances of the phrase “in one embodiment” or “in an embodiment” or “in one implementation” or “in an implementation”, as well as any other variations, appearing in various places throughout this application are not necessarily all referring to the same embodiment.

**[0097]** Additionally, this application may refer to “determining” various pieces of information. Determining the information can include one or more of, for example, estimating the information, calculating the information, predicting the information, or retrieving the information from memory.

**[0098]** Further, this application may refer to “accessing” various pieces of information. Accessing the information can include one or more of, for example, receiving the information, retrieving the information (for example, from memory), storing the information, moving the information, copying the information, calculating the information, determining the information, predicting the information, or estimating the information.

**[0099]** Additionally, this application may refer to “receiving” various pieces of information. Receiving is, as with “accessing”, intended to be a broad term. Receiving the information can include one or more of, for example, accessing the information, or retrieving the information (for example, from memory). Further, “receiving” is typically involved, in one way or another, during operations such as, for example, storing the information, processing the information, transmitting the information, moving the information, copying the information, erasing the information, calculating the information, determining the information, predicting the information, or estimating the information.

**[0100]** It is to be appreciated that the use of any of the following “/”, “and/or”, and “at least one of”, for example, in the cases of “A/B”, “A and/or B” and “at least one of A and B”, is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further example, in the cases of “A, B, and/or C” and “at least one of A, B, and C”, such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C) only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended, as is clear to one of ordinary skill in this and related arts, for as many items as are listed.

**[0101]** Also, as used herein, the word “signal” refers to, among other things, indicating something to a corresponding decoder. For example, in certain embodiments the encoder signals a particular one of a plurality of transforms, coding modes or flags. In this way, in an embodiment the same transform, parameter, or mode is used at both the encoder side and the decoder side. Thus, for example, an encoder can transmit (explicit signaling) a particular parameter to the decoder so that the decoder can use the same particular parameter. Conversely, if the decoder already has the particular parameter as well as others, then signaling can be used without transmitting (implicit signaling) to simply allow the decoder to know and select the particular param-

eter. By avoiding transmission of any actual functions, a bit savings is realized in various embodiments. It is to be appreciated that signaling can be accomplished in a variety of ways. For example, one or more syntax elements, flags, and so forth are used to signal information to a corresponding decoder in various embodiments. While the preceding relates to the verb form of the word “signal”, the word “signal” can also be used herein as a noun.

**[0102]** As will be evident to one of ordinary skill in the art, implementations can produce a variety of signals formatted to carry information that can be, for example, stored or transmitted. The information can include, for example, instructions for performing a method, or data produced by one of the described implementations. For example, a signal can be formatted to carry the bitstream of a described embodiment. Such a signal can be formatted, for example, as an electromagnetic wave (for example, using a radio frequency portion of spectrum) or as a baseband signal. The formatting can include, for example, encoding a data stream and modulating a carrier with the encoded data stream. The information that the signal carries can be, for example, analog or digital information. The signal can be transmitted over a variety of different wired or wireless links, as is known. The signal can be stored on a processor-readable medium.

**[0103]** We describe a number of embodiments, across various claim categories and types. Features of these embodiments can be provided alone or in any combination. Further, embodiments can include one or more of the following features, devices, or aspects, alone or in any combination, across various claim categories and types:

**[0104]** A process or device to encode or decode video data using syntax that disables local chroma quantization parameter control.

**[0105]** A process or device to encode or decode video data using syntax that disables local chroma quantization parameter control, wherein local chroma quantization parameter control is controlled by a local luma quantization parameter.

**[0106]** A process or device to encode or decode video data using syntax that disables local chroma quantization parameter control, wherein local chroma quantization parameter control is controlled by a separate syntax different from local luma quantization parameter control.

**[0107]** One of the above processes or devices according to the HEVC or VVC video standard.

**[0108]** A bitstream or signal that includes one or more of the described syntax elements, or variations thereof.

**[0109]** A bitstream or signal that includes syntax conveying information generated according to any of the embodiments described.

**[0110]** Creating and/or transmitting and/or receiving and/or decoding according to any of the embodiments described.

**[0111]** A method, process, apparatus, medium storing instructions, medium storing data, or signal according to any of the embodiments described.

**[0112]** Inserting in the signaling syntax elements that enable the decoder to determine coding mode in a manner corresponding to that used by an encoder.

**[0113]** Creating and/or transmitting and/or receiving and/or decoding a bitstream or signal that includes one or more of the described syntax elements, or variations thereof.

**[0114]** A TV, set-top box, cell phone, tablet, or other electronic device that performs transform method(s) according to any of the embodiments described.

**[0115]** A TV, set-top box, cell phone, tablet, or other electronic device that performs transform method(s) determination according to any of the embodiments described, and that displays (e.g. using a monitor, screen, or other type of display) a resulting image.

**[0116]** A TV, set-top box, cell phone, tablet, or other electronic device that selects, bandlimits, or tunes (e.g. using a tuner) a channel to receive a signal including an encoded image, and performs transform method(s) according to any of the embodiments described.

**[0117]** A TV, set-top box, cell phone, tablet, or other electronic device that receives (e.g. using an antenna) a signal over the air that includes an encoded image, and performs transform method(s).

1. An apparatus, comprising:

a processor, configured to perform;

quantizing at least one video block based on at least one quantization parameter that is used for chroma quantization; and

encoding said quantized at least one video block and syntax indicative of said at least one quantization parameter, and said syntax comprises a flag to disable local chroma quantization parameter control, wherein said syntax comprises a no chroma qp offset constraint flag, which specifies for a value of 1 that a `pps_cu_chroma_qp_offset_list_enabled_flag` shall be equal to 0.

2. The apparatus of claim 1, wherein said syntax further comprises a flag to control local luma quantization parameter control.

3. The apparatus of claim 1, wherein said syntax comprises at least one flag to control local chroma quantization parameter control.

4. The apparatus of claim 1 wherein said syntax comprises a no gp delta constraint flag.

5. The apparatus of claim 1, wherein said syntax corresponds to a High Efficiency Video Coding H.265 standard.

6. The apparatus of claim 1, wherein said syntax corresponds to a Versatile Video Coding H.266 standard

7. The apparatus of claim 1, wherein said syntax is used for luma quantization control.

8. A method, comprising:

quantizing at least one video block based on at least one quantization parameter that is used for chroma quantization; and,

encoding said quantized at least one video block and syntax indicative of said at least one quantization parameter; wherein said syntax comprises a no chroma qp offset constraint flag, which specifies for a value of 1 that a `pps_cu_chroma_qp_offset_list_enabled_flag` shall be equal to 0.

9. An apparatus, comprising:

a processor, configured to perform;

parsing a video bitstream for syntax used to control at least one quantization parameter that is used for chroma quantization for at least one video block, and,

decoding said at least one video block using the at least one quantization parameter based on the syntax, wherein said syntax comprises a no chroma qp offset constraint flag, which specifies for a value of 1 that a pps cu chroma op offset list enabled flag shall be equal to 0.

**10.** A method, comprising:

parsing a video bitstream for syntax used to control at least one quantization parameter that is used for chroma quantization for at least one video block; and,

decoding said at least one video block using the at least one quantization parameter based on the syntax, and said syntax comprises a flag to disable local chroma quantization parameter control, wherein said syntax comprises a no chroma qp offset constraint flag, which specifies for a value of 1 that a pps cu chroma pp offset list enabled flag shall be equal to 0.

**11.** A device comprising:

the apparatus according to claim 9; and

at least one of (i) an antenna configured to receive a signal, the signal including a video block, (ii) a band

limiter configured to limit the received signal to a band of frequencies that includes the video block, and (iii) a display configured to display an output representative of the video block.

**12.** A non-transitory computer readable medium containing data content generated according to the method of claim 8, for playback using a processor.

**13.** A computer program product comprising instructions which, when the instructions are executed by a computer cause the computer to carry out the method of claim 10.

**14.** The method of claim 10, wherein said syntax comprises another flag to control local luma quantization parameter control

**15.** The method of claim 10, wherein said syntax comprises at least one flag to control local chroma quantization parameter control

**16.** The method of claim 8 wherein said syntax comprises a no\_qp\_delta\_constraint\_flag.

**17.** The apparatus of claim 9, wherein said syntax corresponds to a Versatile Video Coding H.266 standard.

\* \* \* \* \*