



- (12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẢNG ĐỘC QUYỀN SÁNG CHẾ
(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11) 
CỤC SỞ HỮU TRÍ TUỆ 1-0053272
(51)^{2022.01} H04N 19/103; H04N 19/70; H04N 19/174; H04N 19/157; H04N 19/172 (13) B
-

- (21) 1-2022-05590 (22) 24/02/2021
(86) PCT/CN2021/077560 24/02/2021 (87) WO 2021/169969 02/09/2021
(30) PCT/CN2020/077331 29/02/2020 CN
(45) 25/11/2025 452 (43) 26/12/2022 417A
(73) 1. BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD. (CN)
Room B-0035, 2/F, No.3 Building No.30, Shixing Road, Shijingshan District Beijing
100041, China
2. BYTEDANCE INC. (US)
12655 West Jefferson Boulevard Sixth Floor, Suite No. 137 Los Angeles, California
90066, USA
(72) WANG, Ye-Kui (US); DENG, Zhipin (CN); ZHANG, Li (CN); ZHANG, Kai (CN);
XU, Jizheng (CN).
(74) Công ty Luật TNHH Phạm và Liên danh (PHAM & ASSOCIATES)
-

- (54) PHƯƠNG PHÁP VÀ THIẾT BỊ XỬ LÝ DỮ LIỆU VIDEO, VẬT GHI MÁY TÍNH ĐỌC ĐƯỢC, VÀ PHƯƠNG PHÁP LƯU TRỮ DÒNG BIT

(21) 1-2022-05590

(57) Sáng chế đề xuất phương pháp xử lý video mà bao gồm bước thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh và dòng bit của video; trong đó dòng bit tương thích với quy tắc định dạng; trong đó quy tắc định dạng chỉ ra rằng một hoặc nhiều phần tử cú pháp chỉ báo về thông tin tạo mã được bao gồm theo cách có lựa chọn trong tập thông số ảnh của dòng bit đáp lại việc liệu có được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát.

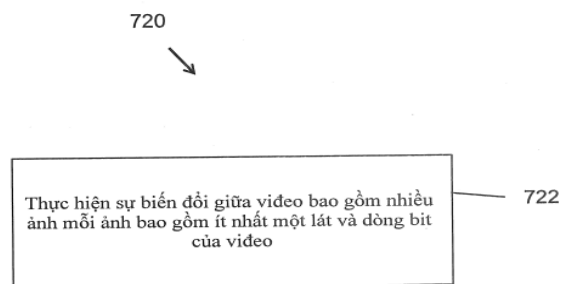


FIG. 7B

Lĩnh vực kỹ thuật được đề cập

Sáng chế liên quan tới việc tạo mã và giải mã hình ảnh và video.

Tình trạng kỹ thuật của sáng chế

Video kỹ thuật số sử dụng băng thông lớn nhất trên internet và các mạng truyền thông kỹ thuật số khác. Do số lượng các thiết bị người dùng được kết nối có khả năng nhận và hiển thị video tăng lên, dự kiến rằng yêu cầu về băng thông dành cho việc sử dụng video kỹ thuật số sẽ tiếp tục tăng trưởng.

Bản chất kỹ thuật của sáng chế

Sáng chế đề xuất các kỹ thuật mà có thể được sử dụng bởi các bộ mã hóa và các bộ giải mã video để xử lý sự biểu diễn được tạo mã của video bằng cách sử dụng thông tin điều khiển hữu ích để giải mã sự biểu diễn được tạo mã.

Theo một khía cạnh ví dụ, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video có một hoặc nhiều ảnh và sự biểu diễn được tạo mã của video, trong đó mỗi ảnh trong số một hoặc nhiều ảnh bao gồm chính xác một lát; trong đó sự biểu diễn được tạo mã tương thích với quy tắc định dạng; trong đó quy tắc định dạng chỉ ra rằng các cờ tập thông số ảnh được loại trừ khỏi sự biểu diễn được tạo mã đối với ảnh có chính xác một lát.

Theo khía cạnh ví dụ khác, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa lát video của video và sự biểu diễn được tạo mã của video, trong đó sự biểu diễn được tạo mã tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra việc bao gồm trường thứ nhất chỉ báo về sự hiện diện của trường thứ hai mà nhận dạng chỉ số lớn nhất đối với ảnh tham chiếu được sử dụng để tạo mã lát video dựa trên điều

kiện; trong đó điều kiện có thể được biểu diễn dưới dạng biểu thức logic “A & (B || C)”, mà trong đó A chỉ báo “rpl_info_in_ph_flag = 1, hoặc kiểu đơn vị NAL không bằng IDR, hoặc sps_idr_rpl_present_flag là bằng 1”, và “B” chỉ báo “slice_type là bằng P hoặc B, và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 0 là lớn hơn 1”, và “C” chỉ báo “slice_type là bằng B và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 1 là lớn hơn 1” và “&” là phép toán “và” logic và “||” là phép toán “hoặc” logic.

Theo khía cạnh ví dụ khác, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa vùng video của video và sự biểu diễn được tạo mã của video, trong đó sự biểu diễn được tạo mã tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra điều kiện mà điều khiển sự sử dụng phần tử cú pháp chỉ báo về đếm thứ tự ảnh đối với ảnh tham chiếu dài hạn được sử dụng để biến đổi vùng video

Theo khía cạnh ví dụ khác, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa vùng video của video và sự biểu diễn được tạo mã của video, trong đó sự biểu diễn được tạo mã tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra rằng các cấu trúc cú pháp liên quan đến ảnh tham chiếu được bao gồm nếu và chỉ nếu biến số X là bằng giá trị cụ thể, trong đó X phụ thuộc vào kiểu lát của vùng video.

Theo khía cạnh ví dụ khác, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh và dòng bit của video; trong đó dòng bit tương thích với quy tắc định dạng; trong đó quy tắc định dạng chỉ ra rằng một hoặc nhiều phần tử cú pháp chỉ báo về thông tin tạo mã được bao gồm theo cách có lựa chọn trong tập thông số ảnh của dòng bit đáp lại việc liệu có được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát.

Theo khía cạnh ví dụ khác, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video bao gồm nhiều ảnh mỗi ảnh bao gồm ít nhất một lát và dòng bit của video, trong đó dòng bit tương thích với quy tắc định dạng; trong đó quy tắc định dạng chỉ ra rằng cờ thứ nhất trong dòng bit chỉ báo việc liệu cấu trúc cú pháp tiêu đề ảnh của ảnh có được

bao gồm trong tiêu đề lát của lát của ảnh; và trong đó quy tắc định dạng chỉ ra rằng, đáp lại cờ bất kỳ trong số sáu cờ trong tập thông số ảnh liên kết với ảnh là bằng 1, cờ thứ nhất là bằng 0.

Theo khía cạnh ví dụ khác, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh, mỗi ảnh bao gồm ít nhất một lát, và dòng bit của video, trong đó dòng bit tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra việc bao gồm trường thứ nhất chỉ báo về sự hiện diện của trường thứ hai mà nhận dạng chỉ số lớn nhất đối với ảnh tham chiếu được sử dụng để tạo mã lát dựa trên điều kiện; trong đó điều kiện có thể được biểu diễn dưới dạng biểu thức logic “A & (B || C)”, mà trong đó: A chỉ báo rằng cờ thứ nhất mà chỉ báo việc liệu thông tin danh sách ảnh tham chiếu có được bao gồm trong cấu trúc cú pháp tiêu đề ảnh là bằng 1, hoặc kiểu đơn vị lớp trừu tượng mạng (NAL) không bằng kiểu làm mới giải mã tức thời (IDR – instantaneous decoding refresh), hoặc cờ thứ hai mà chỉ báo việc liệu thông tin danh sách ảnh tham chiếu có thể hiện diện trong các tiêu đề lát của các lát với các kiểu đơn vị NAL IDR là bằng 1, B chỉ báo rằng kiểu lát là bằng P hoặc B, và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 0 là lớn hơn 1, C chỉ báo kiểu lát là bằng B và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 1 là lớn hơn 1, và trong đó “&” là phép toán “và” logic và “||” là phép toán “hoặc” logic.

Theo khía cạnh ví dụ khác, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video và dòng bit của video, trong đó dòng bit tương thích với quy tắc định dạng, và trong đó quy tắc định dạng chỉ ra điều kiện mà điều khiển việc liệu hoặc cách thức một hoặc nhiều phần tử cú pháp liên quan tới các số đếm thứ tự ảnh đối với ảnh tham chiếu dài hạn được bao gồm trong cấu trúc cú pháp danh sách ảnh tham chiếu trong dòng bit.

Theo khía cạnh ví dụ khác, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video và dòng bit của video, trong đó dòng bit tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra việc liệu phần tử cú pháp liên quan tới các ảnh tham chiếu trong

danh sách ảnh tham chiếu có được bao gồm trong cấu trúc cú pháp danh sách ảnh tham chiếu.

Theo khía cạnh ví dụ khác nữa, thiết bị mã hóa video được bộc lộ. Bộ mã hóa video bao gồm bộ xử lý được tạo cấu hình để thực hiện các phương pháp nêu trên.

Theo khía cạnh ví dụ khác nữa, thiết bị giải mã video được bộc lộ. Bộ giải mã video bao gồm bộ xử lý được tạo cấu hình để thực hiện các phương pháp nêu trên.

Theo khía cạnh ví dụ khác nữa, dòng bit được tạo ra theo phương pháp được mô tả ở đây được bộc lộ. Dòng bit này có thể được lưu trong vật lưu trữ máy tính đọc được.

Theo khía cạnh ví dụ khác nữa, môi trường máy tính đọc được có mã được lưu trong đó được bộc lộ. Mã thực hiện một phương pháp trong số các phương pháp được mô tả ở đây dưới dạng mã bộ xử lý thực thi được.

Các dấu hiệu này và khác nữa được mô tả trong suốt bản mô tả này.

Mô tả vắn tắt các hình vẽ

FIG. 1 là sơ đồ khối của hệ thống xử lý video ví dụ.

FIG. 2 là sơ đồ khối của thiết bị xử lý video.

FIG. 3 là lưu đồ của phương pháp xử lý video ví dụ.

FIG. 4 là sơ đồ khối mà minh họa hệ thống tạo mã video theo một số phương án của sáng chế.

FIG. 5 là sơ đồ khối mà minh họa bộ mã hóa theo một số phương án của sáng chế.

FIG. 6 là sơ đồ khối mà minh họa bộ giải mã theo một số phương án của sáng chế.

Các hình vẽ từ FIG. 7A tới FIG. 7D là các lưu đồ của các phương pháp xử lý video ví dụ.

Mô tả chi tiết sáng chế

Các tiêu đề của các phần được sử dụng trong bản mô tả để làm cho dễ hiểu và không giới hạn khả năng ứng dụng của các kỹ thuật và các phương án được mô tả trong mỗi phần chỉ trong phần đó. Hơn nữa, thuật ngữ của chuẩn H.266 được sử dụng trong một số phần mô tả chỉ để làm cho dễ hiểu và không nhằm giới hạn phạm vi của các kỹ thuật được mô tả. Như vậy, các kỹ thuật được mô tả ở đây cũng có thể được áp dụng cho các giao thức tạo mã-giải mã video và các thiết kế khác.

1. Mở đầu

Phần mô tả này liên quan tới các công nghệ tạo mã video. Cụ thể là, về thiết kế của cú pháp PPS, PH, và SH trong tạo mã video. Các ý tưởng có thể được áp dụng một cách riêng rẽ hoặc theo cách kết hợp khác nhau, đối với chuẩn tạo mã video bất kỳ hoặc bộ tạo mã-giải mã video phi chuẩn mà hỗ trợ tạo mã video nhiều lớp, ví dụ, chuẩn tạo mã video vạn năng (VVC – Versatile Video Coding) đang được phát triển.

2. Các chữ viết tắt

APS	(Adaptation Parameter Set)	Tập thông số thích ứng
AU	(Access Unit)	Đơn vị truy cập
AUD	(Access Unit Delimiter)	Dấu cách đơn vị truy cập
AVC	(Advanced Video Coding)	Tạo mã video cao cấp
CLVS	(Coded Layer Video Sequence)	Dãy video lớp được tạo mã
CPB	(Coded Picture Buffer)	Bộ đệm ảnh được tạo mã
CRA	(Clean Random Access)	Truy cập ngẫu nhiên sạch
CTU	(Coding Tree Unit)	Đơn vị cây tạo mã
CVS	(Coded Video Sequence)	Dãy video được tạo mã
DPB	(Decoded Picture Buffer)	Bộ đệm ảnh được giải mã
DPS	(Decoding Parameter Set)	Tập thông số giải mã
EOB	(End Of Bitstream)	Cuối dòng bit
EOS	(End Of Sequence)	Cuối dãy
GDR	(Gradual Decoding Refresh)	Làm mới giải mã dần

HEVC(High Efficiency Video Coding) Tạo mã video hiệu suất cao

HRD (Hypothetical Reference Decoder) Bộ giải mã tham chiếu giả định

IDR (Instantaneous Decoding Refresh) Làm mới giải mã tức thời

JEM (Joint Exploration Model) Mô hình tìm kiếm chung

LTRP (Long-Term Reference Picture) Ảnh tham chiếu dài hạn

MCTS(Motion-Constrained Tile Sets) Các tập ô ảnh ràng buộc chuyển động

NAL (Network Abstraction Layer) Lớp trừu tượng mạng

OLS (Output Layer Set) Tập lớp đầu ra

PH (Picture Header) Tiêu đề ảnh

PPS (Picture Parameter Set) Tập thông số ảnh

PTL (Profile, Tier and Level) Tiểu sử, Cấp và Bậc

PU (Picture Unit) Đơn vị ảnh

RBSP (Raw Byte Sequence Payload) Tải dữ liệu byte thô

SEI (Supplemental Enhancement Information) Thông tin tăng cường bổ sung

SH (Slice Header) Tiêu đề lát

SPS (Sequence Parameter Set) Tập thông số dãy

STRP (Short-Term Reference Picture) Ảnh tham chiếu ngắn hạn

SVC (Scalable Video Coding) Tạo mã video khả mở rộng

VCL (Video Coding Layer) Lớp tạo mã video

VPS (Video Parameter Set) Tập thông số video

VTM (VVC Test Model) Mô hình thử VVC

VUI (Video Usability Information) Thông tin tính khả dụng video

VVC (Versatile Video Coding) Tạo mã video vạn năng

3. Nhập đề xử lý video

Các chuẩn tạo mã video đã tiến hóa chủ yếu thông qua sự phát triển của các chuẩn ITU-T và ISO/IEC đã biết. ITU-T đã đề xuất H.261 và H.263, ISO/IEC đã đề xuất MPEG-1 và MPEG-4 Visual, và hai tổ chức này cùng đề xuất các chuẩn H.262/MPEG-2 Video và H.264/MPEG-4 Advanced Video Coding (AVC) và H.265/HEVC. Kể từ chuẩn H.262, các chuẩn tạo mã video là dựa trên cấu trúc tạo mã video lai trong đó việc tạo mã dự báo theo thời gian cộng chuyển đổi được sử dụng. Để tìm kiếm các công nghệ tạo mã video tương lai sau HEVC, Joint Video Exploration Team (JVET) đã được đồng thành lập bởi VCEG và MPEG trong năm 2015. Kể từ đó, nhiều phương pháp mới đã được chấp nhận bởi JVET và được đưa vào phần mềm tham chiếu được gọi là Joint Exploration Model (JEM). Hội nghị JVET được đồng thời tổ chức mỗi lần một quý, và chuẩn tạo mã mới nhằm giảm 50% tốc độ bit so với HEVC. Chuẩn tạo mã video mới được đặt tên chính thức là chuẩn Tạo mã video vận năng (VVC) tại hội nghị JVET vào tháng tư năm 2018, và phiên bản thứ nhất của mô hình thử VVC (VTM) đã được phát hành vào lúc đó. Do có các nỗ lực không ngừng nghỉ đóng góp vào chuẩn hóa VVC, các kỹ thuật tạo mã mới được chấp nhận cho chuẩn VVC trong mỗi hội nghị JVET. Dự thảo làm việc VVC và mô hình thử VTM được cập nhật sau mỗi hội nghị. Dự án VVC nhằm hoàn thiện kỹ thuật (FDIS) tại hội nghị tháng 7 năm 2020.

3.1. Cú pháp PPS và các ngữ nghĩa

Trong văn bản dự thảo VVC cuối cùng, cú pháp PPS và các ngữ nghĩa là như sau:

	Mô tả
<code>pic parameter set rbsp() {</code>	
<code>pps_pic_parameter_set_id</code>	ue(v)
<code>pps_seq_parameter_set_id</code>	u(4)
<code>mixed_nalu_types_in_pic_flag</code>	u(1)
<code>pic_width_in_luma_samples</code>	ue(v)
<code>pic_height_in_luma_samples</code>	ue(v)
<code>pps_conformance_window_flag</code>	u(1)
<code>if(pps_conformance_window_flag) {</code>	
<code>pps_conf_win_left_offset</code>	ue(v)
<code>pps_conf_win_right_offset</code>	ue(v)
<code>pps_conf_win_top_offset</code>	ue(v)
<code>pps_conf_win_bottom_offset</code>	ue(v)
<code>}</code>	

scaling_window_explicit_signalling_flag	u(1)
if(scaling_window_explicit_signalling_flag) {	
scaling_win_left_offset	ue(v)
scaling_win_right_offset	ue(v)
scaling_win_top_offset	ue(v)
scaling_win_bottom_offset	ue(v)
}	
output_flag_present_flag	u(1)
subpic_id_mapping_in_pps_flag	u(1)
if(subpic_id_mapping_in_pps_flag) {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpic_minus1; i++)	
pps_subpic_id[i]	u(v)
}	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
pps_log2_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
if(NumTilesInPic > 1)	
rect_slice_flag	u(1)
if(rect_slice_flag)	
single_slice_per_subpic_flag	u(1)
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
if(num_slices_in_pic_minus1 > 0)	
tile_idx_delta_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
if(NumTileColumns > 1)	
slice_width_in_tiles_minus1[i]	ue(v)
if(NumTileRows > 1 &&	
(tile_idx_delta_present_flag tileIdx % NumTileColumns == 0))	
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 &&	
slice_height_in_tiles_minus1[i] == 0 &&	
RowHeight[SliceTopLeftTileIdx[i]/NumTileColumns] > 1) {	
num_exp_slices_in_tile[i]	ue(v)
for(j = 0; j < num_exp_slices_in_tile[i]; j++)	
exp_slice_height_in_ctus_minus1[j]	ue(v)
i += NumSlicesInTiles[i] - 1	
}	
}	
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1)	
tile_idx_delta[i]	se(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
}	

<u>cabac_init_present_flag</u>	u(1)
for(i = 0; i < 2; i++)	
<u>num_ref_idx_default_active_minus1[i]</u>	ue(v)
<u>rpl1_idx_present_flag</u>	u(1)
<u>init_qp_minus26</u>	se(v)
<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)
<u>pps_cu_chroma_qp_offset_list_enabled_flag</u>	u(1)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
<u>chroma_qp_offset_list_len_minus1</u>	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
<u>cb_qp_offset_list[i]</u>	se(v)
<u>cr_qp_offset_list[i]</u>	se(v)
if(pps_joint_cbr_qp_offset_present_flag) {	
<u>joint_cbr_qp_offset_list[i]</u>	se(v)
}	
}	
}	
<u>pps_weighted_pred_flag</u>	u(1)
<u>pps_weighted_bipred_flag</u>	u(1)
<u>deblocking_filter_control_present_flag</u>	u(1)
if(deblocking_filter_control_present_flag) {	
<u>deblocking_filter_override_enabled_flag</u>	u(1)
<u>pps_deblocking_filter_disabled_flag</u>	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
<u>pps_beta_offset_div2</u>	se(v)
<u>pps_tc_offset_div2</u>	se(v)
<u>pps_cb_beta_offset_div2</u>	se(v)
<u>pps_cb_tc_offset_div2</u>	se(v)
<u>pps_cr_beta_offset_div2</u>	se(v)
<u>pps_cr_tc_offset_div2</u>	se(v)
}	
}	
<u>rpl_info_in_ph_flag</u>	u(1)
if(deblocking_filter_override_enabled_flag) {	
<u>dbf_info_in_ph_flag</u>	u(1)
<u>sao_info_in_ph_flag</u>	u(1)
<u>alf_info_in_ph_flag</u>	u(1)
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && rpl_info_in_ph_flag) {	
<u>wp_info_in_ph_flag</u>	u(1)
<u>qp_delta_info_in_ph_flag</u>	u(1)
<u>pps_ref_wraparound_enabled_flag</u>	u(1)
if(pps_ref_wraparound_enabled_flag) {	
<u>pps_ref_wraparound_offset</u>	ue(v)
<u>picture_header_extension_present_flag</u>	u(1)

slice_header_extension_present_flag	u(1)
pps_extension_flag	u(1)
if(pps_extension_flag)	
while(more_rbsp_data())	
pps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

RBSP PPS phải khả dụng cho quá trình giải mã trước khi nó được tham chiếu, được bao gồm trong ít nhất một AU với TemporalId nhỏ hơn hoặc bằng TemporalId của đơn vị NAL PPS hoặc được cấp thông qua phương tiện ngoại vi.

Tất cả các đơn vị NAL PPS với giá trị cụ thể của pps_pic_parameter_set_id bên trong PU phải có cùng nội dung.

pps_pic_parameter_set_id nhận dạng PPS để tham chiếu bởi các phần tử cú pháp khác. Giá trị của pps_pic_parameter_set_id phải nằm trong khoảng từ 0 tới 63, bao gồm các giá trị biên.

Các đơn vị NAL PPS, không phụ thuộc vào các giá trị nuh_layer_id, chia sẻ cùng khoảng giá trị của pps_pic_parameter_set_id.

Giả sử ppsLayerId là giá trị của nuh_layer_id của đơn vị NAL PPS cụ thể, và vcLLayerId là giá trị của nuh_layer_id của đơn vị NAL VCL cụ thể. Đơn vị NAL VCL cụ thể sẽ không tham chiếu tới đơn vị NAL PPS cụ thể trừ khi ppsLayerId là nhỏ hơn hoặc bằng vcLLayerId và lớp với nuh_layer_id bằng ppsLayerId được bao gồm trong ít nhất một OLS mà bao gồm lớp với nuh_layer_id bằng vcLLayerId.

pps_seq_parameter_set_id chỉ ra giá trị của sps_seq_parameter_set_id đối với SPS. Giá trị của pps_seq_parameter_set_id phải nằm trong khoảng từ 0 tới 15, bao gồm các giá trị biên. Giá trị của pps_seq_parameter_set_id phải là giống nhau trong tất cả các PPS mà được tham chiếu tới bởi các ảnh được tạo mã trong CLVS.

mixed_nalu_types_in_pic_flag bằng 1 chỉ ra rằng mỗi ảnh tham chiếu tới PPS có nhiều hơn một đơn vị NAL VCL, các đơn vị NAL VCL này không có cùng giá trị của nal_unit_type, và ảnh này không phải là ảnh IRAP. mixed_nalu_types_in_pic_flag bằng 0 chỉ ra rằng mỗi ảnh tham chiếu tới PPS

có một hoặc nhiều đơn vị NAL VCL và các đơn vị NAL VCL của mỗi ảnh tham chiếu tới PPS có cùng giá trị của `nal_unit_type`.

Khi `no_mixed_nalu_types_in_pic_constraint_flag` là bằng 1, giá trị của `mixed_nalu_types_in_pic_flag` sẽ bằng 0.

Đối với mỗi lát với giá trị `nal_unit_type` `nalUnitTypeA` trong khoảng từ `IDR_W_RADL` tới `CRA_NUT`, bao gồm các giá trị biên, trong ảnh `picA` mà còn chứa một hoặc nhiều lát với giá trị khác của `nal_unit_type` (tức là, giá trị của `mixed_nalu_types_in_pic_flag` đối với ảnh `picA` là bằng 1), điều sau đây được áp dụng:

- Lát phải thuộc về ảnh con `subpicA` mà đối với nó giá trị của `subpic_treated_as_pic_flag[i]` tương ứng là bằng 1.
- Lát sẽ không thuộc về ảnh con của `picA` mà chứa các đơn vị NAL VCL với `nal_unit_type` không bằng `nalUnitTypeA`.
- Nếu `nalUnitTypeA` là bằng `CRA`, đối với tất cả các PU sau đây mà theo sau ảnh hiện tại trong CLVS theo thứ tự giải mã và theo thứ tự xuất ra, không phải `RefPicList[0]` cũng không phải `RefPicList[1]` của lát trong `subpicA` trong các PU đó sẽ bao gồm ảnh bất kỳ đứng trước `picA` theo thứ tự giải mã trong mục nhập hoạt động.
- Ngược lại (tức là, `nalUnitTypeA` là bằng `IDR_W_RADL` hoặc `IDR_N_LP`), đối với tất cả các PU trong CLVS theo sau ảnh hiện tại theo thứ tự giải mã, không phải `RefPicList[0]` cũng không phải `RefPicList[1]` của lát trong `subpicA` trong các PU đó sẽ bao gồm ảnh bất kỳ đứng trước `picA` theo thứ tự giải mã trong mục nhập hoạt động.

LƯU Ý 1– `mixed_nalu_types_in_pic_flag` bằng 1 chỉ báo rằng các ảnh tham chiếu tới PPS chứa các lát với các kiểu đơn vị NAL khác nhau, ví dụ, các ảnh được tạo mã xuất phát từ hoạt động kết hợp dòng bit ảnh con mà đối với chúng các bộ mã hóa phải bảo đảm so khớp cấu trúc dòng bit và sự hiệu chỉnh tiếp sau của các thông số của các dòng bit gốc. Một ví dụ về các hiệu chỉnh này là như sau: Khi giá trị của `sps_idr_rpl_flag` là bằng 0 và `mixed_nalu_types_in_pic_flag` là bằng 1, ảnh tham chiếu tới

PPS không thể có các lát với nal_unit_type bằng IDR_W_RADL hoặc IDR_N_LP.

pic_width_in_luma_samples chỉ ra chiều rộng của mỗi ảnh được giải mã tham chiếu tới PPS trong các đơn vị của các mẫu độ sáng. **pic_width_in_luma_samples** phải không bằng 0, phải là bội số nguyên của $\text{Max}(8, \text{MinCbSizeY})$, và phải nhỏ hơn hoặc bằng **pic_width_max_in_luma_samples**.

Khi **res_change_in_clvs_allowed_flag** bằng 0, giá trị của **pic_width_in_luma_samples** phải bằng **pic_width_max_in_luma_samples**.

pic_height_in_luma_samples chỉ ra chiều cao của mỗi ảnh được giải mã tham chiếu tới PPS trong các đơn vị của các mẫu độ sáng. **pic_height_in_luma_samples** phải không bằng 0 và phải là bội số nguyên của $\text{Max}(8, \text{MinCbSizeY})$, và phải nhỏ hơn hoặc bằng **pic_height_max_in_luma_samples**.

Khi **res_change_in_clvs_allowed_flag** bằng 0, giá trị của **pic_height_in_luma_samples** phải bằng **pic_height_max_in_luma_samples**.

Các biến số **PicWidthInCtbsY**, **PicHeightInCtbsY**, **PicSizeInCtbsY**, **PicWidthInMinCbsY**, **PicHeightInMinCbsY**, **PicSizeInMinCbsY**, **PicSizeInSamplesY**, **PicWidthInSamplesC** và **PicHeightInSamplesC** được rút ra như sau:

$$\text{PicWidthInCtbsY} = \text{Ceil}(\text{pic_width_in_luma_samples} \div \text{CtbSizeY}) \quad (69)$$

$$\text{PicHeightInCtbsY} = \text{Ceil}(\text{pic_height_in_luma_samples} \div \text{CtbSizeY}) \quad (70)$$

$$\text{PicSizeInCtbsY} = \text{PicWidthInCtbsY} * \text{PicHeightInCtbsY} \quad (71)$$

$$\text{PicWidthInMinCbsY} = \text{pic_width_in_luma_samples} / \text{MinCbSizeY} \quad (72)$$

$$\text{PicHeightInMinCbsY} = \text{pic_height_in_luma_samples} / \text{MinCbSizeY} \quad (73)$$

$$\text{PicSizeInMinCbsY} = \text{PicWidthInMinCbsY} * \text{PicHeightInMinCbsY} \quad (74)$$

$$\begin{aligned} \text{PicSizeInSamplesY} = \\ \text{pic_width_in_luma_samples} * \text{pic_height_in_luma_samples} \quad (75) \\ \text{PicWidthInSamplesC} = \text{pic_width_in_luma_samples} / \text{SubWidthC} \quad (76) \end{aligned}$$

$$\text{PicHeightInSamplesC} = \text{pic_height_in_luma_samples} / \text{SubHeightC} \quad (77)$$

pps_conformance_window_flag bằng 1 chỉ báo rằng các thông số dịch vị cửa sổ xén tương thích tiếp ngay sau trong PPS. **pps_conformance_window_flag** bằng 0 chỉ báo rằng các thông số dịch vị cửa sổ xén tương thích là không hiện diện trong PPS.

pps_conf_win_left_offset, **pps_conf_win_right_offset**, **pps_conf_win_top_offset**, và **pps_conf_win_bottom_offset** chỉ ra các mẫu của các ảnh trong CLVS mà được xuất ra từ quá trình giải mã, theo vùng hình chữ nhật được chỉ ra trong các tọa độ ảnh để xuất ra. Khi **pps_conformance_window_flag** là bằng 0, các giá trị của **pps_conf_win_left_offset**, **pps_conf_win_right_offset**, **pps_conf_win_top_offset**, và **pps_conf_win_bottom_offset** được suy ra là bằng 0.

Cửa sổ xén tương thích chứa các mẫu độ sáng với các tọa độ ảnh ngang từ $\text{SubWidthC} * \text{pps_conf_win_left_offset}$ tới $\text{pic_width_in_luma_samples} - (\text{SubWidthC} * \text{pps_conf_win_right_offset} + 1)$ và các tọa độ ảnh dọc từ $\text{SubHeightC} * \text{pps_conf_win_top_offset}$ tới $\text{pic_height_in_luma_samples} - (\text{SubHeightC} * \text{pps_conf_win_bottom_offset} + 1)$, bao gồm các giá trị biên.

Giá trị của $\text{SubWidthC} * (\text{pps_conf_win_left_offset} + \text{pps_conf_win_right_offset})$ phải nhỏ hơn $\text{pic_width_in_luma_samples}$, và giá trị của $\text{SubHeightC} * (\text{pps_conf_win_top_offset} + \text{pps_conf_win_bottom_offset})$ phải nhỏ hơn $\text{pic_height_in_luma_samples}$.

Khi ChromaArrayType không bằng 0, các mẫu được chỉ ra tương ứng của hai mảng sắc độ là các mẫu có các tọa độ ảnh ($x/\text{SubWidthC}$, $y/\text{SubHeightC}$), trong đó (x , y) là các tọa độ ảnh của các mẫu độ sáng được chỉ ra.

LƯU Ý 2 – các thông số dịch vị cửa sổ xén tương thích chỉ được áp dụng ở đầu ra. Tất cả các quá trình giải mã nội tại được áp dụng cho kích thước ảnh không bị xén.

Giả sử ppsA và ppsB là hai PPS bất kỳ tham chiếu tới cùng một SPS. Sự tương thích dòng bit yêu cầu rằng, khi ppsA và ppsB có các giá trị giống nhau lần lượt của `pic_width_in_luma_samples` và `pic_height_in_luma_samples`, thì ppsA và ppsB phải có các giá trị giống nhau lần lượt của `pps_conf_win_left_offset`, `pps_conf_win_right_offset`, `pps_conf_win_top_offset`, và `pps_conf_win_bottom_offset`.

Khi `pic_width_in_luma_samples` là bằng `pic_width_max_in_luma_samples` và `pic_height_in_luma_samples` là bằng `pic_height_max_in_luma_samples`, sự tương thích dòng bit yêu cầu rằng `pps_conf_win_left_offset`, `pps_conf_win_right_offset`, `pps_conf_win_top_offset`, và `pps_conf_win_bottom_offset`, lần lượt bằng `sps_conf_win_left_offset`, `sps_conf_win_right_offset`, `sps_conf_win_top_offset`, và `sps_conf_win_bottom_offset`.

`scaling_window_explicit_signalling_flag` bằng 1 chỉ ra rằng các thông số dịch vị cửa sổ chọn thang là hiện diện trong PPS. **`scaling_window_explicit_signalling_flag`** bằng 0 chỉ ra rằng các thông số dịch vị cửa sổ chọn thang là không hiện diện trong PPS. Khi **`res_change_in_clvs_allowed_flag`** là bằng 0, giá trị của **`scaling_window_explicit_signalling_flag`** phải bằng 0.

`scaling_win_left_offset`, **`scaling_win_right_offset`**, **`scaling_win_top_offset`**, và **`scaling_win_bottom_offset`** chỉ ra các dịch vị mà được áp dụng cho kích thước ảnh để tính toán hệ số chọn thang. Khi không hiện diện, các giá trị của **`scaling_win_left_offset`**, **`scaling_win_right_offset`**, **`scaling_win_top_offset`**, và **`scaling_win_bottom_offset`** được suy ra là lần lượt

bằng $\text{pps_conf_win_left_offset}$, $\text{pps_conf_win_right_offset}$, $\text{pps_conf_win_top_offset}$, và $\text{pps_conf_win_bottom_offset}$.

Giá trị của $\text{SubWidthC} * (\text{scaling_win_left_offset} + \text{scaling_win_right_offset})$ phải nhỏ hơn $\text{pic_width_in_luma_samples}$, và giá trị của $\text{SubHeightC} * (\text{scaling_win_top_offset} + \text{scaling_win_bottom_offset})$ phải nhỏ hơn $\text{pic_height_in_luma_samples}$.

Các biến số PicOutputWidthL và PicOutputHeightL được rút ra như sau:

$$\text{PicOutputWidthL} = \text{pic_width_in_luma_samples} - \quad (78)$$

$$\text{SubWidthC} * (\text{scaling_win_right_offset} + \text{scaling_win_left_offset})$$

$$\text{PicOutputHeightL} = \text{pic_height_in_luma_samples} - \quad (79)$$

$$\text{SubHeightC} * (\text{scaling_win_bottom_offset} + \text{scaling_win_top_offset})$$

Giả sử $\text{refPicOutputWidthL}$ và $\text{refPicOutputHeightL}$ lần lượt là PicOutputWidthL và PicOutputHeightL , của ảnh tham chiếu của ảnh hiện tại mà tham chiếu tới PPS này. Sự tương thích dòng bit yêu cầu rằng tất cả các điều kiện sau đây phải được thỏa mãn:

- $\text{PicOutputWidthL} * 2$ phải lớn hơn hoặc bằng $\text{refPicWidthInLumaSamples}$.

- $\text{PicOutputHeightL} * 2$ phải lớn hơn hoặc bằng $\text{refPicHeightInLumaSamples}$.

- PicOutputWidthL phải nhỏ hơn hoặc bằng $\text{refPicWidthInLumaSamples} * 8$.

- PicOutputHeightL phải nhỏ hơn hoặc bằng $\text{refPicHeightInLumaSamples} * 8$.

- $\text{PicOutputWidthL} * \text{pic_width_max_in_luma_samples}$ phải lớn hơn hoặc bằng $\text{refPicOutputWidthL} * (\text{pic_width_in_luma_samples} - \text{Max}(8, \text{MinCbSize} - Y))$.

- $\text{PicOutputHeightL} * \text{pic_height_max_in_luma_samples}$ phải lớn hơn hoặc bằng $\text{refPicOutputHeightL} * (\text{pic_height_in_luma_samples} - \text{Max}(8, \text{MinCbSize} - Y))$.

$\text{refPicOutputHeightL} * (\text{pic_height_in_luma_samples} - \text{Max}(8, \text{MinCbSize} \text{ Y}))$.

output_flag_present_flag bằng 1 chỉ báo rằng phần tử cú pháp **pic_output_flag** hiện diện trong các tiêu đề lát tham chiếu tới PPS. **output_flag_present_flag** bằng 0 chỉ báo rằng phần tử cú pháp **pic_output_flag** không hiện diện trong các tiêu đề lát tham chiếu tới PPS.

subpic_id_mapping_in_pps_flag bằng 1 chỉ ra rằng ánh xạ ID ảnh con được báo tín hiệu trong PPS. **subpic_id_mapping_in_pps_flag** bằng 0 chỉ ra rằng ánh xạ ID ảnh con không được báo tín hiệu trong PPS. Nếu **subpic_id_mapping_explicitly_signalled_flag** bằng 0 hoặc **subpic_id_mapping_in_sps_flag** là bằng 1, giá trị của **subpic_id_mapping_in_pps_flag** phải bằng 0. Ngược lại (**subpic_id_mapping_explicitly_signalled_flag** là bằng 1 và **subpic_id_mapping_in_sps_flag** là bằng 0), giá trị của **subpic_id_mapping_in_pps_flag** phải bằng 1.

pps_num_subpics_minus1 phải bằng **sps_num_subpics_minus1**.

pps_subpic_id_len_minus1 phải bằng **sps_subpic_id_len_minus1**.

pps_subpic_id[i] chỉ ra ID ảnh con của ảnh con thứ *i*. Chiều dài của phần tử cú pháp **pps_subpic_id[i]** bằng **pps_subpic_id_len_minus1 + 1** bit.

Biến số **SubpicIdVal[i]**, đối với mỗi giá trị của *i* nằm trong khoảng từ 0 tới **sps_num_subpics_minus1**, bao gồm các giá trị biên, được rút ra như sau:

```
for(i = 0; i <= sps_num_subpics_minus1; i++)
    if( subpic_id_mapping_explicitly_signalled_flag)
        SubpicIdVal[i] = subpic_id_mapping_in_pps_flag ?
            pps_subpic_id[i] : sps_subpic_id[i]          (80)
    else
        SubpicIdVal[i] = i
```

Sự tương thích dòng bit yêu cầu rằng cả hai điều kiện ràng buộc sau đây được áp dụng:

– Đối với hai giá trị khác nhau bất kỳ của i và j nằm trong khoảng từ 0 tới $\text{sps_num_subpics_minus1}$, bao gồm các giá trị biên, $\text{SubpicIdVal}[i]$ phải không bằng $\text{SubpicIdVal}[j]$.

– Khi ảnh hiện tại không phải là ảnh thứ nhất của CLVS, đối với mỗi giá trị của i nằm trong khoảng từ 0 tới $\text{sps_num_subpics_minus1}$, bao gồm các giá trị biên, nếu giá trị của $\text{SubpicIdVal}[i]$ không bằng giá trị của $\text{SubpicIdVal}[i]$ của ảnh trước theo thứ tự giải mã trong cùng lớp, nal_unit_type đối với tất cả các đơn vị NAL lát được tạo mã của ảnh con trong ảnh hiện tại với chỉ số ảnh con i phải bằng giá trị cụ thể trong khoảng từ IDR_W_RADL tới CRA_NUT , bao gồm các giá trị biên.

no_pic_partition_flag bằng 1 chỉ ra rằng sự phân vùng ảnh không được áp dụng cho mỗi ảnh tham chiếu tới PPS. **no_pic_partition_flag** bằng 0 chỉ ra mỗi ảnh tham chiếu tới PPS có thể được phân vùng thành nhiều hơn một ô ảnh hoặc lát.

Sự tương thích dòng bit yêu cầu rằng giá trị của **no_pic_partition_flag** phải là giống nhau đối với tất cả các PPS mà được tham chiếu tới bởi các ảnh được tạo mã bên trong CLVS.

Sự tương thích dòng bit yêu cầu rằng giá trị của **no_pic_partition_flag** phải không bằng 1 khi giá trị của $\text{sps_num_subpics_minus1} + 1$ là lớn hơn 1.

pps_log2_ctu_size_minus5 plus 5 chỉ ra kích thước khối cây tạo mã độ sáng của mỗi CTU. **pps_log2_ctu_size_minus5** phải bằng $\text{sps_log2_ctu_size_minus5}$.

num_exp_tile_columns_minus1 plus 1 chỉ ra số lượng các chiều rộng cột ô ảnh được tạo ra một cách rõ ràng. Giá trị của **num_exp_tile_columns_minus1** phải nằm trong khoảng từ 0 tới $\text{PicWidthInCtbsY} - 1$, bao gồm các giá trị biên. Khi **no_pic_partition_flag** là bằng 1, giá trị của **num_exp_tile_columns_minus1** được suy ra là bằng 0.

num_exp_tile_rows_minus1 plus 1 chỉ ra số lượng các chiều cao hàng ô ảnh được tạo ra một cách rõ ràng. Giá trị của **num_exp_tile_rows_minus1** phải nằm trong khoảng từ 0 tới $\text{PicHeightInCtbsY} - 1$, bao gồm các giá trị biên. Khi **no_pic_partition_flag** là bằng 1, giá trị của **num_exp_tile_rows_minus1** được suy ra là bằng 0.

tile_column_width_minus1[i] plus 1 chỉ ra chiều rộng của cột ô ảnh thứ *i* trong các đơn vị của các CTB đối với *i* nằm trong khoảng từ 0 tới **num_exp_tile_columns_minus1** – 1, bao gồm các giá trị biên. **tile_column_width_minus1[num_exp_tile_columns_minus1]** được sử dụng để rút ra chiều rộng của các cột ô ảnh với chỉ số lớn hơn hoặc bằng **num_exp_tile_columns_minus1** như được chỉ ra trong mục 6.5.1. Giá trị của **tile_column_width_minus1[i]** phải nằm trong khoảng từ 0 tới **PicWidthInCtbsY** – 1, bao gồm các giá trị biên. Khi không hiện diện, giá trị của **tile_column_width_minus1[0]** được suy ra là bằng **PicWidthInCtbsY** – 1.

tile_row_height_minus1[i] plus 1 chỉ ra chiều cao của hàng ô ảnh thứ *i* trong các đơn vị của các CTB đối với *i* nằm trong khoảng từ 0 tới **num_exp_tile_rows_minus1** – 1, bao gồm các giá trị biên. **tile_row_height_minus1[num_exp_tile_rows_minus1]** được sử dụng để rút ra chiều cao của các hàng ô ảnh với chỉ số lớn hơn hoặc bằng **num_exp_tile_rows_minus1** như được chỉ ra trong mục 6.5.1. Giá trị của **tile_row_height_minus1[i]** phải nằm trong khoảng từ 0 tới **PicHeightInCtbsY** – 1, bao gồm các giá trị biên. Khi không hiện diện, giá trị của **tile_row_height_minus1[0]** được suy ra là bằng **PicHeightInCtbsY** – 1.

rect_slice_flag bằng 0 chỉ ra rằng các ô ảnh bên trong mỗi lát là theo thứ tự quét mảnh và thông tin lát không được báo tín hiệu trong PPS. **rect_slice_flag** bằng 1 chỉ ra rằng các ô ảnh bên trong mỗi lát bao phủ vùng hình chữ nhật của ảnh và thông tin lát được báo tín hiệu trong PPS. Khi không hiện diện, **rect_slice_flag** được suy ra là bằng 1. Khi **subpic_info_present_flag** là bằng 1, giá trị của **rect_slice_flag** phải bằng 1.

single_slice_per_subpic_flag bằng 1 chỉ ra rằng mỗi ảnh con chứa một và chỉ một lát hình chữ nhật. **single_slice_per_subpic_flag** bằng 0 chỉ ra rằng mỗi ảnh con có thể chứa một hoặc nhiều lát hình chữ nhật. Khi **single_slice_per_subpic_flag** là bằng 1, **num_slices_in_pic_minus1** được suy ra là bằng **sps_num_subpics_minus1**. Khi không hiện diện, giá trị của **single_slice_per_subpic_flag** được suy ra là bằng 0.

num_slices_in_pic_minus1 plus 1 chỉ ra số lượng các lát hình chữ nhật trong mỗi ảnh tham chiếu tới PPS. Giá trị của **num_slices_in_pic_minus1** phải nằm trong khoảng từ 0 tới **MaxSlicesPerPicture** – 1, bao gồm các giá trị biên, trong đó **MaxSlicesPerPicture** được chỉ ra trong Phụ lục A. Khi **no_pic_partition_flag** là bằng 1, giá trị của **num_slices_in_pic_minus1** được suy ra là bằng 0.

tile_idx_delta_present_flag bằng 0 chỉ ra rằng các giá trị **tile_idx_delta** là không hiện diện trong PPS và tất cả các lát hình chữ nhật trong các ảnh tham chiếu tới PPS được chỉ ra theo thứ tự mảnh theo quy trình được xác định trong mục 6.5.1. **tile_idx_delta_present_flag** bằng 1 chỉ ra rằng các giá trị **tile_idx_delta** có thể hiện diện trong PPS và tất cả các lát hình chữ nhật trong các ảnh tham chiếu tới PPS được chỉ ra theo thứ tự được chỉ báo bởi các giá trị của **tile_idx_delta**. Khi không hiện diện, giá trị của **tile_idx_delta_present_flag** được suy ra là bằng 0.

slice_width_in_tiles_minus1[i] plus 1 chỉ ra chiều rộng của lát hình chữ nhật thứ *i* trong các đơn vị của các cột ô ảnh. Giá trị của **slice_width_in_tiles_minus1[i]** phải nằm trong khoảng từ 0 tới **NumTileColumns** – 1, bao gồm các giá trị biên.

Khi **slice_width_in_tiles_minus1[i]** không hiện diện, điều sau đây được áp dụng:

- Nếu **NumTileColumns** là bằng 1, giá trị của **slice_width_in_tiles_minus1[i]** được suy ra là bằng 0.
- Ngược lại, giá trị của **slice_width_in_tiles_minus1[i]** được suy ra như được chỉ ra trong mục 6.5.1.

slice_height_in_tiles_minus1[i] plus 1 chỉ ra chiều cao của lát hình chữ nhật thứ *i* trong các đơn vị của các hàng ô ảnh. Giá trị của **slice_height_in_tiles_minus1[i]** phải nằm trong khoảng từ 0 tới **NumTileRows** – 1, bao gồm các giá trị biên.

Khi **slice_height_in_tiles_minus1[i]** không hiện diện, điều sau đây được áp dụng:

– Nếu NumTileRows là bằng 1, hoặc tile_idx_delta_present_flag là bằng 0 và tileIdx % NumTileColumns là lớn hơn 0), giá trị của slice_height_in_tiles_minus1[i] được suy ra là bằng 0.

– Ngược lại (NumTileRows không bằng 1, và tile_idx_delta_present_flag là bằng 1 hoặc tileIdx % NumTileColumns là bằng 0), khi tile_idx_delta_present_flag là bằng 1 hoặc tileIdx % NumTileColumns là bằng 0, giá trị của slice_height_in_tiles_minus1[i] được suy ra là bằng slice_height_in_tiles_minus1[i - 1].

num_exp_slices_in_tile[i] chỉ ra số lượng các chiều cao lát được tạo ra một cách rõ ràng trong ô ảnh hiện tại mà chứa nhiều hơn một lát hình chữ nhật. Giá trị của num_exp_slices_in_tile[i] phải nằm trong khoảng từ 0 tới RowHeight[tileY] - 1, bao gồm các giá trị biên, trong đó tileY là chỉ số hàng ô ảnh chứa lát thứ i. Khi không hiện diện, giá trị của num_exp_slices_in_tile[i] được suy ra là bằng 0. Khi num_exp_slices_in_tile[i] là bằng 0, giá trị của biến số NumSlicesInTiles[i] được rút ra là bằng 1.

exp_slice_height_in_ctus_minus1[j] plus 1 chỉ ra chiều cao của lát hình chữ nhật thứ j trong ô ảnh hiện tại trong các đơn vị của các hàng CTU. Giá trị của exp_slice_height_in_ctus_minus1[j] phải nằm trong khoảng từ 0 tới RowHeight[tileY] - 1, bao gồm các giá trị biên, trong đó tileY là chỉ số hàng ô ảnh của ô ảnh hiện tại.

Khi num_exp_slices_in_tile[i] là lớn hơn 0, biến số NumSlicesInTiles[i] và SliceHeightInCtusMinus1[i + k] đối với k nằm trong khoảng từ 0 tới NumSlicesInTiles[i] - 1 được rút ra như sau:

```

remainingHeightInCtbsY =
RowHeight[SliceTopLeftTileIdx[i]/NumTileColumns]
numExpSliceInTile = num_exp_slices_in_tile[i]
for(j = 0; j < numExpSliceInTile - 1; j++) {
    SliceHeightInCtusMinus1[i++] = exp_slice_height_in_ctu_minus1[j]
    remainingHeightInCtbsY -= SliceHeightInCtusMinus1[j]
}

```

```

uniformSliceHeightMinus1 = SliceHeightInCtusMinus1[i - 1] (81)
while( remainingHeightInCtbsY >= (uniformSliceHeightMinus1 + 1))
{
    SliceHeightInCtusMinus1[i++] = uniformSliceHeightMinus1
    remainingHeightInCtbsY -= (uniformSliceHeightMinus1 + 1)
    j++
}
if( remainingHeightInCtbsY > 0) {
    SliceHeightInCtusMinus1[i++] = remainingHeightInCtbsY
    j++
}
NumSlicesInTiles[i] = j

```

tile_idx_delta[i] chỉ ra hiệu số giữa chỉ số ô ảnh của ô ảnh thứ nhất trong lát hình chữ nhật thứ i và chỉ số ô ảnh của ô ảnh thứ nhất trong lát hình chữ nhật thứ $(i + 1)$. Giá trị của **tile_idx_delta[i]** phải nằm trong khoảng từ $-\text{NumTilesInPic} + 1$ tới $\text{NumTilesInPic} - 1$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của **tile_idx_delta[i]** được suy ra là bằng 0. Khi hiện diện, giá trị của **tile_idx_delta[i]** phải không bằng 0.

loop_filter_across_tiles_enabled_flag bằng 1 chỉ ra rằng các hoạt động lọc trong vòng có thể được thực hiện ngang qua các biên ô ảnh trong các ảnh tham chiếu tới PPS. **loop_filter_across_tiles_enabled_flag** bằng 0 chỉ ra rằng các hoạt động lọc trong vòng không được thực hiện ngang qua các biên ô ảnh trong các ảnh tham chiếu tới PPS. Các hoạt động lọc trong vòng bao gồm bộ lọc giải khối, bộ lọc dịch vị thích ứng mẫu, và các hoạt động lọc vòng thích ứng. Khi không hiện diện, giá trị của **loop_filter_across_tiles_enabled_flag** được suy ra là bằng 1.

loop_filter_across_slices_enabled_flag bằng 1 chỉ ra rằng các hoạt động lọc trong vòng có thể được thực hiện ngang qua các biên lát trong các ảnh tham chiếu tới PPS. **loop_filter_across_slice_enabled_flag** bằng 0 chỉ ra rằng các hoạt động lọc trong vòng không được thực hiện ngang qua các biên lát trong các ảnh tham chiếu tới PPS. Các hoạt động lọc trong vòng bao gồm bộ lọc giải khối, bộ

lọc dịch vị thích ứng mẫu, và các hoạt động lọc vòng thích ứng. Khi không hiện diện, giá trị của `loop_filter_across_slices_enabled_flag` được suy ra là bằng 0.

`cabac_init_present_flag` bằng 1 chỉ ra rằng `cabac_init_flag` hiện diện trong các tiêu đề lát tham chiếu tới PPS. `cabac_init_present_flag` bằng 0 chỉ ra rằng `cabac_init_flag` không hiện diện trong các tiêu đề lát tham chiếu tới PPS.

`num_ref_idx_default_active_minus1[i]` plus 1, khi *i* bằng 0, chỉ ra giá trị được suy ra của biến số `NumRefIdxActive[0]` đối với các lát P hoặc B với `num_ref_idx_active_override_flag` bằng 0, và, khi *i* bằng 1, chỉ ra giá trị được suy ra của `NumRefIdxActive[1]` đối với các lát B với `num_ref_idx_active_override_flag` bằng 0. Giá trị của `num_ref_idx_default_active_minus1[i]` phải nằm trong khoảng từ 0 tới 14, bao gồm các giá trị biên.

`rpl1_idx_present_flag` bằng 0 chỉ ra rằng `ref_pic_list_sps_flag[1]` và `ref_pic_list_idx[1]` là không hiện diện trong các cấu trúc cú pháp PH hoặc các tiêu đề lát đối với các ảnh tham chiếu tới PPS. `rpl1_idx_present_flag` bằng 1 chỉ ra rằng `ref_pic_list_sps_flag[1]` và `ref_pic_list_idx[1]` có thể hiện diện trong các cấu trúc cú pháp PH hoặc các tiêu đề lát đối với các ảnh tham chiếu tới PPS.

`init_qp_minus26` plus 26 chỉ ra giá trị ban đầu của `SliceQpY` đối với mỗi lát tham chiếu tới PPS. Giá trị ban đầu của `SliceQpY` được cải biến ở bậc ảnh khi giá trị khác không của `ph_qp_delta` được giải mã hoặc ở bậc lát khi giá trị khác không của `slice_qp_delta` được giải mã. Giá trị của `init_qp_minus26` phải nằm trong khoảng từ $-(26 + QpBdOffset)$ tới +37, bao gồm các giá trị biên.

`cu_qp_delta_enabled_flag` bằng 1 chỉ ra rằng các phần tử cú pháp `ph_cu_qp_delta_subdiv_intra_slice` và `ph_cu_qp_delta_subdiv_inter_slice` là hiện diện trong các PH tham chiếu tới PPS và `cu_qp_delta_abs` có thể hiện diện trong cú pháp khối chuyển đổi. `cu_qp_delta_enabled_flag` bằng 0 chỉ ra rằng các phần tử cú pháp `ph_cu_qp_delta_subdiv_intra_slice` và `ph_cu_qp_delta_subdiv_inter_slice` là không hiện diện trong các PH tham chiếu tới PPS và `cu_qp_delta_abs` không hiện diện trong cú pháp khối chuyển đổi.

`pps_chroma_tool_offsets_present_flag` bằng 1 chỉ ra rằng các phần tử cú pháp liên quan đến các dịch vị công cụ sắc độ là hiện diện trong cấu trúc cú pháp

RBSP PPS. `pps_chroma_tool_offsets_present_flag` bằng 0 chỉ ra rằng các phần tử cú pháp liên quan đến các dịch vị công cụ sắc độ là không hiện diện trong cấu trúc cú pháp RBSP PPS. Khi `ChromaArrayType` là bằng 0, giá trị của `pps_chroma_tool_offsets_present_flag` phải bằng 0.

`pps_cb_qp_offset` và `pps_cr_qp_offset` chỉ ra các dịch vị tới thông số lượng tử hóa độ sáng Qp'_Y được sử dụng để lần lượt rút ra Qp'_{Cb} và Qp'_{Cr} . Các giá trị của `pps_cb_qp_offset` và `pps_cr_qp_offset` phải nằm trong khoảng từ -12 tới +12, bao gồm các giá trị biên. Khi `ChromaArrayType` bằng 0, `pps_cb_qp_offset` và `pps_cr_qp_offset` không được sử dụng trong quá trình giải mã và các bộ giải mã phải bỏ qua giá trị của chúng. Khi không hiện diện, các giá trị của `pps_cb_qp_offset` và `pps_cr_qp_offset` được suy ra là bằng 0.

`pps_joint_cbr_qp_offset_present_flag` bằng 1 chỉ ra rằng `pps_joint_cbr_qp_offset_value` và `joint_cbr_qp_offset_list[i]` là hiện diện trong cấu trúc cú pháp RBSP PPS. `pps_joint_cbr_qp_offset_present_flag` bằng 0 chỉ ra rằng `pps_joint_cbr_qp_offset_value` và `joint_cbr_qp_offset_list[i]` là không hiện diện trong cấu trúc cú pháp RBSP PPS. Khi `ChromaArrayType` là bằng 0 hoặc `sps_joint_cbr_enabled_flag` là bằng 0, giá trị của `pps_joint_cbr_qp_offset_present_flag` phải bằng 0. Khi không hiện diện, giá trị của `pps_joint_cbr_qp_offset_present_flag` được suy ra là bằng 0.

`pps_joint_cbr_qp_offset_value` chỉ ra dịch vị tới thông số lượng tử hóa độ sáng Qp'_Y được sử dụng để rút ra Qp'_{CbCr} . Giá trị của `pps_joint_cbr_qp_offset_value` phải nằm trong khoảng từ -12 tới +12, bao gồm các giá trị biên. Khi `ChromaArrayType` là bằng 0 hoặc `sps_joint_cbr_enabled_flag` là bằng 0, `pps_joint_cbr_qp_offset_value` không được sử dụng trong quá trình giải mã và các bộ giải mã phải bỏ qua giá trị của nó. Khi `pps_joint_cbr_qp_offset_present_flag` là bằng 0, `pps_joint_cbr_qp_offset_value` không hiện diện và được suy ra là bằng 0.

`pps_slice_chroma_qp_offsets_present_flag` bằng 1 chỉ ra rằng các phần tử cú pháp `slice_cb_qp_offset` và `slice_cr_qp_offset` là hiện diện trong các tiêu đề lát được liên kết. `pps_slice_chroma_qp_offsets_present_flag` bằng 0 chỉ ra rằng các phần tử cú pháp `slice_cb_qp_offset` và `slice_cr_qp_offset` là không hiện diện

trong các tiêu đề lát được liên kết. Khi không hiện diện, giá trị của `pps_slice_chroma_qp_offsets_present_flag` được suy ra là bằng 0.

`pps_cu_chroma_qp_offset_list_enabled_flag` bằng 1 chỉ ra rằng các phần tử cú pháp `ph_cu_chroma_qp_offset_subdiv_intra_slice` và `ph_cu_chroma_qp_offset_subdiv_inter_slice` là hiện diện trong các PH tham chiếu tới PPS và `cu_chroma_qp_offset_flag` có thể hiện diện trong cú pháp khối chuyển đổi và cú pháp tạo mã bảng màu. `pps_cu_chroma_qp_offset_list_enabled_flag` bằng 0 chỉ ra rằng các phần tử cú pháp `ph_cu_chroma_qp_offset_subdiv_intra_slice` và `ph_cu_chroma_qp_offset_subdiv_inter_slice` là không hiện diện trong các PH tham chiếu tới PPS và `cu_chroma_qp_offset_flag` không hiện diện trong cú pháp khối chuyển đổi và cú pháp tạo mã bảng màu. Khi không hiện diện, giá trị của `pps_cu_chroma_qp_offset_list_enabled_flag` được suy ra là bằng 0.

`chroma_qp_offset_list_len_minus1` plus 1 chỉ ra số lượng các phần tử cú pháp `cb_qp_offset_list[i]`, `cr_qp_offset_list[i]`, và `joint_cbc_r_qp_offset_list[i]` mà hiện diện trong cấu trúc cú pháp RBSP PPS. Giá trị của `chroma_qp_offset_list_len_minus1` phải nằm trong khoảng từ 0 tới 5, bao gồm các giá trị biên.

`cb_qp_offset_list[i]`, `cr_qp_offset_list[i]`, và `joint_cbc_r_qp_offset_list[i]`, chỉ ra các dịch vị được sử dụng trong khi lần lượt rút ra $Q_p'_{Cb}$, $Q_p'_{Cr}$, và $Q_p'_{CbCr}$. Các giá trị của `cb_qp_offset_list[i]`, `cr_qp_offset_list[i]`, và `joint_cbc_r_qp_offset_list[i]` phải nằm trong khoảng từ -12 tới +12, bao gồm các giá trị biên. Khi `pps_joint_cbc_r_qp_offset_present_flag` là bằng 0, `joint_cbc_r_qp_offset_list[i]` không hiện diện và nó được suy ra là bằng 0.

`pps_weighted_pred_flag` bằng 0 chỉ ra rằng dự báo có trọng số không được áp dụng cho các lát P tham chiếu tới PPS. `pps_weighted_pred_flag` bằng 1 chỉ ra rằng dự báo có trọng số được áp dụng cho các lát P tham chiếu tới PPS. Khi `sps_weighted_pred_flag` là bằng 0, giá trị của `pps_weighted_pred_flag` phải bằng 0.

`pps_weighted_bipred_flag` bằng 0 chỉ ra rằng dự báo có trọng số rõ ràng không được áp dụng cho các lát B tham chiếu tới PPS.

pps_weighted_bipred_flag bằng 1 chỉ ra rằng dự báo có trọng số rõ ràng được áp dụng cho các lát B tham chiếu tới PPS. Khi **sps_weighted_bipred_flag** là bằng 0, giá trị của **pps_weighted_bipred_flag** phải bằng 0.

deblocking_filter_control_present_flag bằng 1 chỉ ra sự hiện diện của các phần tử cú pháp điều khiển lọc giải khối trong PPS. **deblocking_filter_control_present_flag** bằng 0 chỉ ra sự không hiện diện của các phần tử cú pháp điều khiển lọc giải khối trong PPS.

deblocking_filter_override_enabled_flag bằng 1 chỉ ra sự hiện diện của **ph_deblocking_filter_override_flag** trong các PH tham chiếu tới PPS hoặc **slice_deblocking_filter_override_flag** trong các tiêu đề lát tham chiếu tới PPS. **deblocking_filter_override_enabled_flag** bằng 0 chỉ ra sự không hiện diện của **ph_deblocking_filter_override_flag** trong các PH tham chiếu tới PPS hoặc **slice_deblocking_filter_override_flag** trong các tiêu đề lát tham chiếu tới PPS. Khi không hiện diện, giá trị của **deblocking_filter_override_enabled_flag** được suy ra là bằng 0.

pps_deblocking_filter_disabled_flag bằng 1 chỉ ra rằng hoạt động của bộ lọc giải khối không được áp dụng cho các lát tham chiếu tới PPS mà trong đó **slice_deblocking_filter_disabled_flag** không hiện diện. **pps_deblocking_filter_disabled_flag** bằng 0 chỉ ra rằng hoạt động của bộ lọc giải khối được áp dụng cho các lát tham chiếu tới PPS mà trong đó **slice_deblocking_filter_disabled_flag** không hiện diện. Khi không hiện diện, giá trị của **pps_deblocking_filter_disabled_flag** được suy ra là bằng 0.

pps_beta_offset_div2 và **pps_tc_offset_div2** chỉ ra các dịch vị thông số giải khối mặc định đối với β và tC (được chia cho 2) mà được áp dụng cho thành phần độ sáng đối với các lát tham chiếu tới PPS, trừ khi các dịch vị thông số giải khối mặc định được ghi đè bởi các dịch vị thông số giải khối hiện diện trong các tiêu đề ảnh hoặc các tiêu đề lát của các lát tham chiếu tới PPS. Cả hai giá trị của **pps_beta_offset_div2** và **pps_tc_offset_div2** phải nằm trong khoảng từ -12 tới 12, bao gồm các giá trị biên. Khi không hiện diện, cả hai giá trị của **pps_beta_offset_div2** và **pps_tc_offset_div2** đều được suy ra là bằng 0.

pps_cb_beta_offset_div2 và **pps_cb_tc_offset_div2** chỉ ra các dịch vị thông số giải khối mặc định đối với β và tC (được chia cho 2) mà được áp dụng cho thành phần Cb đối với các lát tham chiếu tới PPS, trừ khi các dịch vị thông số giải khối mặc định được ghi đè bởi các dịch vị thông số giải khối hiện diện trong các tiêu đề ảnh hoặc các tiêu đề lát của các lát tham chiếu tới PPS. Cả hai giá trị của **pps_cb_beta_offset_div2** và **pps_cb_tc_offset_div2** đều phải nằm trong khoảng từ -12 tới 12, bao gồm các giá trị biên. Khi không hiện diện, cả hai giá trị của **pps_cb_beta_offset_div2** và **pps_cb_tc_offset_div2** đều được suy ra là bằng 0.

pps_cr_beta_offset_div2 và **pps_cr_tc_offset_div2** chỉ ra các dịch vị thông số giải khối mặc định đối với β và tC (được chia cho 2) mà được áp dụng cho thành phần Cr đối với các lát tham chiếu tới PPS, trừ khi các dịch vị thông số giải khối mặc định được ghi đè bởi các dịch vị thông số giải khối hiện diện trong các tiêu đề ảnh hoặc các tiêu đề lát của các lát tham chiếu tới PPS. Cả hai giá trị của **pps_cr_beta_offset_div2** và **pps_cr_tc_offset_div2** đều phải nằm trong khoảng từ -12 tới 12, bao gồm các giá trị biên. Khi không hiện diện, cả hai giá trị của **pps_cr_beta_offset_div2** và **pps_cr_tc_offset_div2** đều được suy ra là bằng 0.

rpl_info_in_ph_flag bằng 1 chỉ ra rằng thông tin danh sách ảnh tham chiếu là hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. **rpl_info_in_ph_flag** bằng 0 chỉ ra rằng thông tin danh sách ảnh tham chiếu không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH.

dbf_info_in_ph_flag bằng 1 chỉ ra rằng thông tin lọc giải khối là hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. **dbf_info_in_ph_flag** bằng 0 chỉ ra rằng thông tin lọc giải khối không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. Khi không hiện diện, giá trị của **dbf_info_in_ph_flag** được suy ra là bằng 0.

sao_info_in_ph_flag bằng 1 chỉ ra rằng thông tin lọc SAO là hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. **sao_info_in_ph_flag** bằng 0 chỉ ra rằng thông tin lọc SAO không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH.

alf_info_in_ph_flag bằng 1 chỉ ra rằng thông tin ALF là hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. **alf_info_in_ph_flag** bằng 0 chỉ ra rằng thông tin ALF không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH.

wp_info_in_ph_flag bằng 1 chỉ ra rằng thông tin dự báo có trọng số có thể hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. **wp_info_in_ph_flag** bằng 0 chỉ ra rằng thông tin dự báo có trọng số không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. Khi không hiện diện, giá trị của **wp_info_in_ph_flag** được suy ra là bằng 0.

qp_delta_info_in_ph_flag bằng 1 chỉ ra rằng thông tin delta QP là hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. **qp_delta_info_in_ph_flag** bằng 0 chỉ ra rằng thông tin delta QP không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH.

pps_ref_wraparound_enabled_flag bằng 1 chỉ ra rằng bù chuyển động vòng lại ngang được áp dụng trong dự báo ngoài. **pps_ref_wraparound_enabled_flag** bằng 0 chỉ ra rằng bù chuyển động vòng lại ngang không được áp dụng. Khi giá trị của $\text{CtbSizeY}/\text{MinCbSizeY} + 1$ là lớn hơn $\text{pic_width_in_luma_samples}/\text{MinCbSizeY} - 1$, giá trị của **pps_ref_wraparound_enabled_flag** phải bằng 0. Khi **sps_ref_wraparound_enabled_flag** là bằng 0, giá trị của **pps_ref_wraparound_enabled_flag** phải bằng 0.

pps_ref_wraparound_offset plus $(CtbSizeY/MinCbSizeY) + 2$ chỉ ra dịch vị được sử dụng để tính vị trí vòng lại ngang trong các đơn vị của các mẫu độ sáng $MinCbSizeY$. Giá trị của **pps_ref_wraparound_offset** phải nằm trong khoảng từ 0 tới $(pic_width_in_luma_samples/MinCbSizeY) - (CtbSizeY/MinCbSizeY) - 2$, bao gồm các giá trị biên.

Biến số **PpsRefWraparoundOffset** được thiết lập bằng **pps_ref_wraparound_offset** + $(CtbSizeY/MinCbSizeY) + 2$.

picture_header_extension_present_flag bằng 0 chỉ ra rằng không có các phần tử cú pháp mở rộng PH hiện diện trong các PH tham chiếu tới PPS. **picture_header_extension_present_flag** bằng 1 chỉ ra rằng các phần tử cú pháp mở rộng PH là hiện diện trong các PH tham chiếu tới PPS. **picture_header_extension_present_flag** phải bằng 0 trong các dòng bit tương thích với phiên bản này của Đặc tả này.

slice_header_extension_present_flag bằng 0 chỉ ra rằng không có các phần tử cú pháp mở rộng tiêu đề lát hiện diện trong các tiêu đề lát đối với các ảnh được tạo mã tham chiếu tới PPS. **slice_header_extension_present_flag** bằng 1 chỉ ra rằng các phần tử cú pháp mở rộng tiêu đề lát là hiện diện trong các tiêu đề lát đối với các ảnh được tạo mã tham chiếu tới PPS. **slice_header_extension_present_flag** phải bằng 0 trong các dòng bit tương thích với phiên bản này của Đặc tả này.

pps_extension_flag bằng 0 chỉ ra rằng không có các phần tử cú pháp **pps_extension_data_flag** hiện diện trong cấu trúc cú pháp RBSP PPS. **pps_extension_flag** bằng 1 chỉ ra rằng có các phần tử cú pháp **pps_extension_data_flag** hiện diện trong cấu trúc cú pháp RBSP PPS.

pps_extension_data_flag có thể có giá trị bất kỳ. Sự hiện diện và giá trị của nó không ảnh hưởng đến sự tương thích của bộ giải mã với các tiêu sử được chỉ ra trong phiên bản này của Đặc tả này. Các bộ giải mã tương thích với phiên bản này của Đặc tả này phải bỏ qua tất cả các phần tử cú pháp **pps_extension_data_flag**.

3.2. Cú pháp PH và các ngữ nghĩa

Trong văn bản dự thảo VVC cuối cùng, cú pháp PH và các ngữ nghĩa là như sau:

picture header rbsp() {	Mô tả
picture_header_structure()	
rbsp_trailing_bits()	
}	

PH RBSP chứa cấu trúc cú pháp PH, tức là, picture_header_structure().

picture_header_structure() {	Mô tả
gdr_or_irap_pic_flag	u(1)
if(gdr_or_irap_pic_flag)	
gdr_pic_flag	u(1)
ph_inter_slice_allowed_flag	u(1)
if(ph_inter_slice_allowed_flag)	
ph_intra_slice_allowed_flag	u(1)
non_reference_picture_flag	u(1)
ph_pic_parameter_set_id	ue(v)
ph_pic_order_cnt_lsb	u(v)
if(gdr_or_irap_pic_flag)	
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
for(i = 0; i < NumExtraPhBits; i++)	
ph_extra_bit[i]	u(1)
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
if(sps_alf_enabled_flag && alf_info_in_ph_flag) {	
ph_alf_enabled_flag	u(1)
if(ph_alf_enabled_flag) {	
ph_num_alf_aps_ids_luma	u(3)
for(i = 0; i < ph_num_alf_aps_ids_luma; i++)	
ph_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
ph_alf_chroma_idc	u(2)
if(ph_alf_chroma_idc > 0)	
ph_alf_aps_id_chroma	u(3)
if(sps_ccalf_enabled_flag) {	
ph_cc_alf_cb_enabled_flag	u(1)
if(ph_cc_alf_cb_enabled_flag)	
ph_cc_alf_cb_aps_id	u(3)
ph_cc_alf_cr_enabled_flag	u(1)
if(ph_cc_alf_cr_enabled_flag)	
ph_cc_alf_cr_aps_id	u(3)
}	
}	
}	
}	
if(sps_lmcs_enabled_flag) {	

ph lmcs_enabled_flag	u(1)
if(ph lmcs_enabled_flag) {	
ph lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
ph chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
ph scaling_list_present_flag	u(1)
if(ph_scaling_list_present_flag)	
ph scaling_list_aps_id	u(3)
}	
if(sps_virtual_boundaries_enabled_flag && !sps_virtual_boundaries_present_flag) {	
ph virtual_boundaries_present_flag	u(1)
if(ph_virtual_boundaries_present_flag) {	
ph num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph virtual_boundaries_pos_x[i]	u(13)
ph num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(output_flag_present_flag)	
pic_output_flag	u(1)
if(rpl_info_in_ph_flag)	
ref_pic_lists()	
if(partition_constraints_override_enabled_flag)	
partition_constraints_override_flag	u(1)
if(ph_intra_slice_allowed_flag) {	
if(partition_constraints_override_flag) {	
ph log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
ph max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(ph_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
ph log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
ph log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	
ph log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
ph max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(ph_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
ph log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
ph log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
if(cu_qp_delta_enabled_flag)	
ph cu_qp_delta_subdiv_intra_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
ph cu_chroma_qp_offset_subdiv_intra_slice	ue(v)
}	
if(ph_inter_slice_allowed_flag) {	

if(partition constraints override flag) {	
ph_log2_diff_min_qt_min_cb_inter_slice	ue(v)
ph_max_mtt_hierarchy_depth_inter_slice	ue(v)
if(ph_max_mtt_hierarchy_depth_inter_slice != 0) {	
ph_log2_diff_max_bt_min_qt_inter_slice	ue(v)
ph_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
}	
if(cu_qp_delta_enabled_flag)	
ph_cu_qp_delta_subdiv_inter_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
ph_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
if(sps_temporal_mvp_enabled_flag) {	
ph_temporal_mvp_enabled_flag	u(1)
if(ph_temporal_mvp_enabled_flag && rpl_info_in_ph_flag) {	
ph_collocated_from_l0_flag	u(1)
if((ph_collocated_from_l0_flag && num_ref_entries[0][RplIdx[0]] > 1) (!ph_collocated_from_l0_flag && num_ref_entries[1][RplIdx[1]] > 1))	
ph_collocated_ref_idx	ue(v)
}	
}	
mvd_l1_zero_flag	u(1)
if(sps_fpel_mmvd_enabled_flag)	
ph_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
ph_disable_bdof_flag	u(1)
if(sps_dmvr_pic_present_flag)	
ph_disable_dmvr_flag	u(1)
if(sps_prof_pic_present_flag)	
ph_disable_prof_flag	u(1)
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && wp_info_in_ph_flag)	
pred_weight_table()	
}	
if(qp_delta_info_in_ph_flag)	
ph_qp_delta	se(v)
if(sps_joint_cbr_enabled_flag)	
ph_joint_cbr_sign_flag	u(1)
if(sps_sao_enabled_flag && sao_info_in_ph_flag) {	
ph_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
ph_sao_chroma_enabled_flag	u(1)
}	
if(sps_dep_quant_enabled_flag)	
ph_dep_quant_enabled_flag	u(1)
if(sps_sign_data_hiding_enabled_flag && !ph_dep_quant_enabled_flag)	
pic_sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag && dbf_info_in_ph_flag) {	
ph_deblocking_filter_override_flag	u(1)
if(ph_deblocking_filter_override_flag) {	
ph_deblocking_filter_disabled_flag	u(1)

if(!ph deblocking filter disabled flag) {	
ph_beta_offset_div2	se(v)
ph_tc_offset_div2	se(v)
ph_cb_beta_offset_div2	se(v)
ph_cb_tc_offset_div2	se(v)
ph_cr_beta_offset_div2	se(v)
ph_cr_tc_offset_div2	se(v)
}	
}	
}	
if(picture header extension present flag) {	
ph_extension_length	ue(v)
for(i = 0; i < ph_extension_length; i++)	
ph_extension_data_byte[i]	u(8)
}	
}	

Cấu trúc cú pháp PH chứa thông tin mà là chung đối với tất cả các lát của ảnh được tạo mã liên kết với cấu trúc cú pháp PH.

gdr_or_irap_pic_flag bằng 1 chỉ ra rằng ảnh hiện tại là ảnh GDR hoặc ảnh IRAP. **gdr_or_irap_pic_flag** bằng 0 chỉ ra rằng ảnh hiện tại có thể là hoặc có thể không là ảnh GDR hoặc ảnh IRAP.

gdr_pic_flag bằng 1 chỉ ra ảnh liên kết với PH là ảnh GDR. **gdr_pic_flag** bằng 0 chỉ ra rằng ảnh liên kết với PH không phải ảnh GDR. Khi không hiện diện, giá trị của **gdr_pic_flag** được suy ra là bằng 0. Khi **gdr_enabled_flag** là bằng 0, giá trị của **gdr_pic_flag** phải bằng 0.

ph_inter_slice_allowed_flag bằng 0 chỉ ra rằng tất cả các lát được tạo mã của ảnh có **slice_type** bằng 2. **ph_inter_slice_allowed_flag** bằng 1 chỉ ra rằng có thể có hoặc có thể không có một hoặc nhiều lát được tạo mã trong ảnh mà có **slice_type** bằng 0 hoặc 1.

ph_intra_slice_allowed_flag bằng 0 chỉ ra rằng tất cả các lát được tạo mã của ảnh có **slice_type** bằng 0 hoặc 1. **ph_intra_slice_allowed_flag** bằng 1 chỉ ra rằng có thể có hoặc có thể không có một hoặc nhiều lát được tạo mã trong ảnh mà có **slice_type** bằng 2. Khi không hiện diện, giá trị của **ph_intra_slice_allowed_flag** được suy ra là bằng 1.

LƯU Ý 1 – Đối với các dòng bit mà phải làm việc với ảnh con dựa trên sự kết hợp dòng bit mà không cần phải thay đổi các đơn vị NAL PH, bộ mã

hóa được dự kiến thiết lập các giá trị của cả `ph_inter_slice_allowed_flag` lẫn `ph_intra_slice_allowed_flag` bằng 1.

`non_reference_picture_flag` bằng 1 chỉ ra ảnh liên kết với PH sẽ không bao giờ được sử dụng làm ảnh tham chiếu. `non_reference_picture_flag` bằng 0 chỉ ra ảnh liên kết với PH có thể có hoặc có thể không được sử dụng làm ảnh tham chiếu.

`ph_pic_parameter_set_id` chỉ ra giá trị của `pps_pic_parameter_set_id` đối với PPS đang được sử dụng. Giá trị của `ph_pic_parameter_set_id` phải nằm trong khoảng từ 0 tới 63, bao gồm các giá trị biên.

Sự tương thích dòng bit yêu cầu rằng giá trị của `TemporalId` của PH phải lớn hơn hoặc bằng giá trị của `TemporalId` của PPS mà có `pps_pic_parameter_set_id` bằng `ph_pic_parameter_set_id`.

`ph_pic_order_cnt_lsb` chỉ ra môđun đếm thứ tự ảnh `MaxPicOrderCntLsb` đối với ảnh hiện tại. Chiều dài của phần tử cú pháp `ph_pic_order_cnt_lsb` bằng $\log_2_max_pic_order_cnt_lsb_minus4 + 4$ bit. Giá trị của `ph_pic_order_cnt_lsb` phải nằm trong khoảng từ 0 tới $MaxPicOrderCntLsb - 1$, bao gồm các giá trị biên.

`no_output_of_prior_pics_flag` ảnh hưởng tới đầu ra của các ảnh được giải mã trước đó trong DPB sau khi giải mã ảnh CLVSS mà không phải là ảnh thứ nhất trong dòng bit như được chỉ ra trong Phụ lục C.

`recovery_poc_cnt` chỉ ra điểm phục hồi của các ảnh được giải mã theo thứ tự xuất ra. Nếu ảnh hiện tại là ảnh GDR mà được liên kết với PH, và có ảnh `picA` mà theo sau ảnh GDR hiện tại theo thứ tự giải mã trong CLVS mà có `PicOrderCntVal` bằng `PicOrderCntVal` của ảnh GDR hiện tại cộng với giá trị của `recovery_poc_cnt`, thì ảnh `picA` được gọi là ảnh điểm phục hồi. Ngược lại, ảnh thứ nhất theo thứ tự xuất ra mà có `PicOrderCntVal` lớn hơn `PicOrderCntVal` của ảnh hiện tại cộng với giá trị của `recovery_poc_cnt` được gọi là ảnh điểm phục hồi. Ảnh điểm phục hồi sẽ không đứng trước ảnh GDR hiện tại theo thứ tự giải mã. Giá trị của `recovery_poc_cnt` phải nằm trong khoảng từ 0 tới $MaxPicOrderCntLsb - 1$, bao gồm các giá trị biên.

Khi ảnh hiện tại là ảnh GDR, biến số RpPicOrderCntVal được rút ra như sau:

$$\text{RpPicOrderCntVal} = \text{PicOrderCntVal} + \text{recovery_poc_cnt} \quad (82)$$

LƯU Ý 2 – Khi gdr_enabled_flag là bằng 1 và PicOrderCntVal của ảnh hiện tại là lớn hơn hoặc bằng RpPicOrderCntVal của ảnh GDR được liên kết, các ảnh được giải mã hiện tại và tiếp theo theo thứ tự xuất ra là so khớp chính xác với các ảnh tương ứng được tạo ra bằng cách bắt đầu quá trình giải mã từ ảnh IRAP trước đó, khi hiện diện, mà đứng trước ảnh GDR được liên kết theo thứ tự giải mã.

ph_extra_bit[i] có thể bằng 1 hoặc 0. Các bộ giải mã tương thích với phiên bản này của Đặc tả này phải bỏ qua giá trị của **ph_extra_bit[i]**. Giá trị của nó không ảnh hưởng tới sự tương thích của bộ giải mã với các tiêu sử được chỉ ra trong phiên bản này của đặc tả.

ph_poc_msb_present_flag bằng 1 chỉ ra rằng phần tử cú pháp poc_msb_val là hiện diện trong PH. **ph_poc_msb_present_flag** bằng 0 chỉ ra rằng phần tử cú pháp poc_msb_val không hiện diện trong PH. Khi vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]] là bằng 0 và có ảnh trong AU hiện tại trong lớp tham chiếu của lớp hiện tại, giá trị của **ph_poc_msb_present_flag** phải bằng 0.

poc_msb_val chỉ ra giá trị MSB POC của ảnh hiện tại. Chiều dài của phần tử cú pháp poc_msb_val bằng poc_msb_len_minus1 + 1 bit.

ph_alf_enabled_flag bằng 1 chỉ ra rằng bộ lọc vòng thích ứng là được phép đối với tất cả các lát liên kết với PH và có thể được áp dụng cho thành phần màu Y, Cb, hoặc Cr trong các lát. **ph_alf_enabled_flag** bằng 0 chỉ ra rằng bộ lọc vòng thích ứng có thể không được phép đối với một, hoặc nhiều, hoặc tất cả các lát liên kết với PH. Khi không hiện diện, **ph_alf_enabled_flag** được suy ra là bằng 0.

ph_num_alf_aps_ids_luma chỉ ra số lượng các APS ALF mà các lát liên kết với PH tham chiếu tới.

ph_alf_aps_id_luma[i] chỉ ra adaptation_parameter_set_id của APS ALF thứ i mà thành phần độ sáng của các lát liên kết với PH tham chiếu tới.

Giá trị của `alf_luma_filter_signal_flag` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `ph_alf_aps_id_luma[i]` phải bằng 1.

`TemporalId` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `ph_alf_aps_id_luma[i]` phải nhỏ hơn hoặc bằng `TemporalId` của ảnh liên kết với PH.

`ph_alf_chroma_idc` bằng 0 chỉ ra rằng bộ lọc vòng thích ứng không được áp dụng cho các thành phần màu Cb và Cr. `ph_alf_chroma_idc` bằng 1 chỉ báo rằng bộ lọc vòng thích ứng được áp dụng cho thành phần màu Cb. `ph_alf_chroma_idc` bằng 2 chỉ báo rằng bộ lọc vòng thích ứng được áp dụng cho thành phần màu Cr. `ph_alf_chroma_idc` bằng 3 chỉ báo rằng bộ lọc vòng thích ứng được áp dụng cho các thành phần màu Cb và Cr. Khi `ph_alf_chroma_idc` không hiện diện, nó được suy ra là bằng 0.

`ph_alf_aps_id_chroma` chỉ ra `adaptation_parameter_set_id` của APS ALF mà thành phần sắc độ của các lát liên kết với PH tham chiếu tới.

Giá trị của `alf_chroma_filter_signal_flag` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `ph_alf_aps_id_chroma` phải bằng 1.

`TemporalId` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `ph_alf_aps_id_chroma` phải nhỏ hơn hoặc bằng `TemporalId` của ảnh liên kết với PH.

`ph_cc_alf_cb_enabled_flag` bằng 1 chỉ ra rằng bộ lọc giao thành phần đối với thành phần màu Cb là được phép đối với tất cả các lát liên kết với PH và có thể được áp dụng cho thành phần màu Cb trong các lát. `ph_cc_alf_cb_enabled_flag` bằng 0 chỉ ra rằng bộ lọc giao thành phần đối với thành phần màu Cb có thể là không được phép đối với một, hoặc nhiều, hoặc tất cả các lát liên kết với PH. Khi không hiện diện, `ph_cc_alf_cb_enabled_flag` được suy ra là bằng 0.

`ph_cc_alf_cb_aps_id` chỉ ra `adaptation_parameter_set_id` của APS ALF mà thành phần màu Cb của các lát liên kết với PH tham chiếu tới.

Giá trị của `alf_cc_cb_filter_signal_flag` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `ph_cc_alf_cb_aps_id` phải bằng 1.

`TemporalId` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `ph_cc_alf_cb_aps_id` phải nhỏ hơn hoặc bằng `TemporalId` của ảnh liên kết với PH.

`ph_cc_alf_cr_enabled_flag` bằng 1 chỉ ra rằng bộ lọc giao thành phần đối với thành phần màu Cr là được phép đối với tất cả các lát liên kết với PH và có thể được áp dụng cho thành phần màu Cr trong các lát. `ph_cc_alf_cr_enabled_flag` bằng 0 chỉ ra rằng bộ lọc giao thành phần đối với thành phần màu Cr có thể là không được phép đối với một, hoặc nhiều, hoặc tất cả các lát liên kết với PH. Khi không hiện diện, `ph_cc_alf_cr_enabled_flag` được suy ra là bằng 0.

`ph_cc_alf_cr_aps_id` chỉ ra `adaptation_parameter_set_id` của APS ALF mà thành phần màu Cr của các lát liên kết với PH tham chiếu tới.

Giá trị của `alf_cc_cr_filter_signal_flag` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `ph_cc_alf_cr_aps_id` phải bằng 1.

`TemporalId` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `ph_cc_alf_cr_aps_id` phải nhỏ hơn hoặc bằng `TemporalId` của ảnh liên kết với PH.

`ph_lmcs_enabled_flag` bằng 1 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ là được phép đối với tất cả các lát liên kết với PH. `ph_lmcs_enabled_flag` bằng 0 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ có thể là không được phép đối với một, hoặc nhiều, hoặc tất cả các lát liên kết với PH. Khi không hiện diện, giá trị của `ph_lmcs_enabled_flag` được suy ra là bằng 0.

`ph_lmcs_aps_id` chỉ ra `adaptation_parameter_set_id` của APS LMCS mà các lát liên kết với PH tham chiếu tới. `TemporalId` của đơn vị NAL APS có `aps_params_type` bằng `LMCS_APS` và `adaptation_parameter_set_id` bằng `ph_lmcs_aps_id` phải nhỏ hơn hoặc bằng `TemporalId` của ảnh liên kết với PH.

ph_chroma_residual_scale_flag bằng 1 chỉ ra rằng chọn thang dư sắc độ là được phép đối với tất cả các lát liên kết với PH. **ph_chroma_residual_scale_flag** bằng 0 chỉ ra rằng chọn thang dư sắc độ có thể là không được phép đối với một, hoặc nhiều, hoặc tất cả các lát liên kết với PH. Khi **ph_chroma_residual_scale_flag** không hiện diện, nó được suy ra là bằng 0.

ph_scaling_list_present_flag bằng 1 chỉ ra rằng dữ liệu danh sách chọn thang được sử dụng đối với các lát được liên kết với PH được rút ra dựa trên dữ liệu danh sách chọn thang được chứa trong danh sách chọn thang APS tham chiếu. **ph_scaling_list_present_flag** bằng 0 chỉ ra rằng dữ liệu danh sách chọn thang được sử dụng đối với các lát liên kết với PH được thiết lập bằng 16. Khi không hiện diện, giá trị của **ph_scaling_list_present_flag** được suy ra là bằng 0.

ph_scaling_list_aps_id chỉ ra **adaptation_parameter_set_id** của danh sách chọn thang APS. **TemporalId** của đơn vị NAL APS có **aps_params_type** bằng **SCALING_APS** và **adaptation_parameter_set_id** bằng **ph_scaling_list_aps_id** phải nhỏ hơn hoặc bằng **TemporalId** của ảnh liên kết với PH.

ph_virtual_boundaries_present_flag bằng 1 chỉ ra rằng thông tin về các biên ảo được báo tín hiệu trong PH. **ph_virtual_boundaries_present_flag** bằng 0 chỉ ra rằng thông tin về các biên ảo không được báo tín hiệu trong PH. Khi có một hoặc nhiều hơn một các biên ảo được báo tín hiệu trong PH, các hoạt động lọc trong vòng là không được phép ngang qua các biên ảo trong ảnh. Các hoạt động lọc trong vòng bao gồm bộ lọc giải khối, bộ lọc dịch vị thích ứng mẫu, và các hoạt động lọc vòng thích ứng. Khi không hiện diện, giá trị của **ph_virtual_boundaries_present_flag** được suy ra là bằng 0.

Sự tương thích dòng bit yêu cầu rằng, khi **subpic_info_present_flag** là bằng 1, giá trị của **ph_virtual_boundaries_present_flag** phải bằng 0.

Biến số **VirtualBoundariesPresentFlag** được rút ra như sau:

```
VirtualBoundariesPresentFlag = 0
if( sps_virtual_boundaries_enabled_flag )
    VirtualBoundariesPresentFlag =
        sps_virtual_boundaries_present_flag ||
```

$$\begin{aligned} & \text{ph_virtual_boundaries_present_flag} & (83) \\ & (83) \end{aligned}$$

ph_num_ver_virtual_boundaries chỉ ra số lượng của các phần tử cú pháp **ph_virtual_boundaries_pos_x[i]** mà hiện diện trong PH. Khi **ph_num_ver_virtual_boundaries** không hiện diện, nó được suy ra là bằng 0.

Biến số NumVerVirtualBoundaries được rút ra như sau:

$$\begin{aligned} & \text{NumVerVirtualBoundaries} = 0 \\ & \text{if(sps_virtual_boundaries_enabled_flag)} \\ & \quad \text{NumVerVirtualBoundaries} = \text{sps_virtual_boundaries_present_flag ?} \\ & \quad \text{sps_num_ver_virtual_boundaries :} \\ & \quad \text{ph_num_ver_virtual_boundaries} & (84) \end{aligned}$$

ph_virtual_boundaries_pos_x[i] chỉ ra vị trí của đường biên ảo dọc thứ *i* trong các đơn vị của các mẫu độ sáng được chia cho 8. Giá trị của **ph_virtual_boundaries_pos_x[i]** phải nằm trong khoảng từ 1 tới $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, bao gồm các giá trị biên.

Danh sách VirtualBoundariesPosX[i] đối với *i* nằm trong khoảng từ 0 tới NumVerVirtualBoundaries - 1, bao gồm các giá trị biên, trong các đơn vị của các mẫu độ sáng, chỉ ra các vị trí của các biên ảo dọc, được rút ra như sau:

$$\begin{aligned} & \text{for}(i = 0; i < \text{NumVerVirtualBoundaries}; i++) \\ & \quad \text{VirtualBoundariesPosX}[i] = (\text{sps_virtual_boundaries_present_flag ?} \\ & \quad \text{sps_virtual_boundaries_pos_x}[i] : \\ & \quad \text{ph_virtual_boundaries_pos_x}[i]) * 8 & (85) \end{aligned}$$

Khoảng cách giữa hai biên ảo dọc bất kỳ phải lớn hơn hoặc bằng các mẫu độ sáng CtbSizeY.

ph_num_hor_virtual_boundaries chỉ ra số lượng của các phần tử cú pháp **ph_virtual_boundaries_pos_y[i]** mà hiện diện trong PH. Khi **ph_num_hor_virtual_boundaries** không hiện diện, nó được suy ra là bằng 0.

Thông số NumHorVirtualBoundaries được rút ra như sau:

$$\begin{aligned} & \text{NumHorVirtualBoundaries} = 0 \\ & \text{if(sps_virtual_boundaries_enabled_flag)} \\ & \quad \text{NumHorVirtualBoundaries} = \text{sps_virtual_boundaries_present_flag ?} \end{aligned}$$

`sps_num_hor_virtual_boundaries` :

`ph_num_hor_virtual_boundaries` (86)

Khi `sps_virtual_boundaries_enabled_flag` là bằng 1 và `ph_virtual_boundaries_present_flag` là bằng 1, tổng của `ph_num_ver_virtual_boundaries` và `ph_num_hor_virtual_boundaries` phải lớn hơn 0.

`ph_virtual_boundaries_pos_y[i]` chỉ ra vị trí của đường biên ảo ngang thứ `i` trong các đơn vị của các mẫu độ sáng được chia cho 8. Giá trị của `ph_virtual_boundaries_pos_y[i]` phải nằm trong khoảng từ 1 tới $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, bao gồm các giá trị biên.

Danh sách `VirtualBoundariesPosY[i]` đối với `i` nằm trong khoảng từ 0 tới `NumHorVirtualBoundaries - 1`, bao gồm các giá trị biên, trong các đơn vị của các mẫu độ sáng, chỉ ra các vị trí của các biên ảo ngang, được rút ra như sau:

`for(i = 0; i < NumHorVirtualBoundaries; i++)`

`VirtualBoundariesPosY[i] = (sps_virtual_boundaries_present_flag ?`

`sps_virtual_boundaries_pos_y[i] :`

`ph_virtual_boundaries_pos_y[i]) * 8` (87)

Khoảng cách giữa hai biên ảo ngang bất kỳ phải lớn hơn hoặc bằng các mẫu độ sáng `CtbSizeY`.

`pic_output_flag` ảnh hưởng tới các quá trình xuất ra và loại bỏ ảnh được giải mã như được chỉ ra trong Phụ lục C. Khi `pic_output_flag` không hiện diện, nó được suy ra là bằng 1.

`partition_constraints_override_flag` bằng 1 chỉ ra rằng các thông số điều kiện ràng buộc phân vùng là hiện diện trong PH. `partition_constraints_override_flag` bằng 0 chỉ ra rằng các thông số điều kiện ràng buộc phân vùng là không hiện diện trong PH. Khi không hiện diện, giá trị của `partition_constraints_override_flag` được suy ra là bằng 0.

`ph_log2_diff_min_qt_min_cb_intra_slice_luma` chỉ ra hiệu số giữa lôgarit cơ số 2 của kích thước nhỏ nhất trong các mẫu độ sáng của khối dạng lá độ sáng thu được từ sự tách cây tứ phân của CTU và lôgarit cơ số 2 của kích thước khối tạo mã nhỏ nhất trong các mẫu độ sáng đối với các CU độ sáng trong

các lát với `slice_type` bằng 2 (I) liên kết với PH. Giá trị của `ph_log2_diff_min_qt_min_cb_intra_slice_luma` phải nằm trong khoảng từ 0 tới $\text{CtbLog2SizeY} - \text{MinCbLog2SizeY}$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của `ph_log2_diff_min_qt_min_cb_luma` được suy ra là bằng `sps_log2_diff_min_qt_min_cb_intra_slice_luma`.

`ph_max_mtt_hierarchy_depth_intra_slice_luma` chỉ ra chiều sâu phân cấp lớn nhất đối với các đơn vị tạo mã thu được từ việc tách cây nhiều kiểu của lá cây tứ phân trong các lát với `slice_type` bằng 2 (I) liên kết với PH. Giá trị của `ph_max_mtt_hierarchy_depth_intra_slice_luma` phải nằm trong khoảng từ 0 tới $2 * (\text{CtbLog2SizeY} - \text{MinCbLog2SizeY})$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của `ph_max_mtt_hierarchy_depth_intra_slice_luma` được suy ra là bằng `sps_max_mtt_hierarchy_depth_intra_slice_luma`.

`ph_log2_diff_max_bt_min_qt_intra_slice_luma` chỉ ra hiệu số giữa lôgarit cơ số 2 của kích thước lớn nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối tạo mã độ sáng mà có thể được tách bằng cách sử dụng việc tách nhị phân và kích thước nhỏ nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối dạng lá độ sáng thu được từ sự tách cây tứ phân của CTU trong các lát với `slice_type` bằng 2 (I) liên kết với PH. Giá trị của `ph_log2_diff_max_bt_min_qt_intra_slice_luma` phải nằm trong khoảng từ 0 tới $\text{CtbLog2SizeY} - \text{MinQtLog2SizeIntraY}$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của `ph_log2_diff_max_bt_min_qt_intra_slice_luma` được suy ra là bằng `sps_log2_diff_max_bt_min_qt_intra_slice_luma`.

`ph_log2_diff_max_tt_min_qt_intra_slice_luma` chỉ ra hiệu số giữa lôgarit cơ số 2 của kích thước lớn nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối tạo mã độ sáng mà có thể được tách bằng cách sử dụng việc tách tam phân và kích thước nhỏ nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối dạng lá độ sáng thu được từ sự tách cây tứ phân của CTU trong các lát với `slice_type` bằng 2 (I) liên kết với PH. Giá trị của `ph_log2_diff_max_tt_min_qt_intra_slice_luma` phải nằm trong khoảng từ 0 tới $\text{CtbLog2SizeY} - \text{MinQtLog2SizeIntraY}$, bao gồm các giá trị biên. Khi không

hiện diện, giá trị của `ph_log2_diff_max_tt_min_qt_intra_slice_luma` được suy ra là bằng `sps_log2_diff_max_tt_min_qt_intra_slice_luma`.

`ph_log2_diff_min_qt_min_cb_intra_slice_chroma` chỉ ra hiệu số giữa lôgarit cơ số 2 của kích thước nhỏ nhất trong các mẫu độ sáng của khối dạng lá sắc độ thu được từ sự tách cây tứ phân của CTU sắc độ với `treeType` bằng `DUAL_TREE_CHROMA` và lôgarit cơ số 2 của kích thước khối tạo mã nhỏ nhất trong các mẫu độ sáng đối với các CU sắc độ với `treeType` bằng `DUAL_TREE_CHROMA` trong các lát với `slice_type` bằng 2 (I) liên kết với PH. Giá trị của `ph_log2_diff_min_qt_min_cb_intra_slice_chroma` phải nằm trong khoảng từ 0 tới $\text{CtbLog2SizeY} - \text{MinCbLog2SizeY}$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của `ph_log2_diff_min_qt_min_cb_intra_slice_chroma` được suy ra là bằng `sps_log2_diff_min_qt_min_cb_intra_slice_chroma`.

`ph_max_mtt_hierarchy_depth_intra_slice_chroma` chỉ ra chiều sâu phân cấp lớn nhất đối với các đơn vị tạo mã sắc độ thu được từ việc tách cây nhiều kiểu của lá cây tứ phân sắc độ với `treeType` bằng `DUAL_TREE_CHROMA` trong các lát với `slice_type` bằng 2 (I) liên kết với PH. Giá trị của `ph_max_mtt_hierarchy_depth_intra_slice_chroma` phải nằm trong khoảng từ 0 tới $2 * (\text{CtbLog2SizeY} - \text{MinCbLog2SizeY})$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của `ph_max_mtt_hierarchy_depth_intra_slice_chroma` được suy ra là bằng `sps_max_mtt_hierarchy_depth_intra_slice_chroma`.

`ph_log2_diff_max_bt_min_qt_intra_slice_chroma` chỉ ra hiệu số giữa lôgarit cơ số 2 của kích thước lớn nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối tạo mã sắc độ mà có thể được tách bằng cách sử dụng việc tách nhị phân và kích thước nhỏ nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối dạng lá sắc độ thu được từ sự tách cây tứ phân của CTU sắc độ với `treeType` bằng `DUAL_TREE_CHROMA` trong các lát với `slice_type` bằng 2 (I) liên kết với PH. Giá trị của `ph_log2_diff_max_bt_min_qt_intra_slice_chroma` phải nằm trong khoảng từ 0 tới $\text{CtbLog2SizeY} - \text{MinQtLog2SizeIntraC}$, bao gồm các giá trị biên. Khi không

hiện diện, giá trị của `ph_log2_diff_max_bt_min_qt_intra_slice_chroma` được suy ra là bằng `sps_log2_diff_max_bt_min_qt_intra_slice_chroma`.

`ph_log2_diff_max_tt_min_qt_intra_slice_chroma` chỉ ra hiệu số giữa lôgarit cơ số 2 của kích thước lớn nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối tạo mã sắc độ mà có thể được tách bằng cách sử dụng việc tách tam phân và kích thước nhỏ nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối dạng lá sắc độ thu được từ sự tách cây tứ phân của CTU sắc độ với `treeType` bằng `DUAL_TREE_CHROMA` trong các lát với `slice_type` bằng 2 (I) liên kết với PH. Giá trị của `ph_log2_diff_max_tt_min_qt_intra_slice_chroma` phải nằm trong khoảng từ 0 tới $\text{CtbLog2SizeY} - \text{MinQtLog2SizeIntraC}$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của `ph_log2_diff_max_tt_min_qt_intra_slice_chroma` được suy ra là bằng `sps_log2_diff_max_tt_min_qt_intra_slice_chroma`.

`ph_cu_qp_delta_subdiv_intra_slice` chỉ ra giá trị `cbSubdiv` lớn nhất của các đơn vị tạo mã trong lát trong mà chuyển `cu_qp_delta_abs` và `cu_qp_delta_sign_flag`. Giá trị của `ph_cu_qp_delta_subdiv_intra_slice` phải nằm trong khoảng từ 0 tới $2 * (\text{CtbLog2SizeY} - \text{MinQtLog2SizeIntraY} + \text{ph_max_mtt_hierarchy_depth_intra_slice_luma})$, bao gồm các giá trị biên.

Khi không hiện diện, giá trị của `ph_cu_qp_delta_subdiv_intra_slice` được suy ra là bằng 0.

`ph_cu_chroma_qp_offset_subdiv_intra_slice` chỉ ra giá trị `cbSubdiv` lớn nhất của các đơn vị tạo mã trong lát trong mà chuyển `cu_chroma_qp_offset_flag`. Giá trị của `ph_cu_chroma_qp_offset_subdiv_intra_slice` phải nằm trong khoảng từ 0 tới $2 * (\text{CtbLog2SizeY} - \text{MinQtLog2SizeIntraY} + \text{ph_max_mtt_hierarchy_depth_intra_slice_luma})$, bao gồm các giá trị biên.

Khi không hiện diện, giá trị của `ph_cu_chroma_qp_offset_subdiv_intra_slice` được suy ra là bằng 0.

`ph_log2_diff_min_qt_min_cb_inter_slice` chỉ ra hiệu số giữa lôgarit cơ số 2 của kích thước nhỏ nhất trong các mẫu độ sáng của khối dạng lá độ sáng thu

được từ sự tách cây tứ phân của CTU và lôgarit cơ số 2 của kích thước khối tạo mã độ sáng nhỏ nhất trong các mẫu độ sáng đối với các CU độ sáng trong các lát với `slice_type` bằng 0 (B) hoặc 1 (P) liên kết với PH. Giá trị của `ph_log2_diff_min_qt_min_cb_inter_slice` phải nằm trong khoảng từ 0 tới $\text{CtbLog2SizeY} - \text{MinCbLog2SizeY}$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của `ph_log2_diff_min_qt_min_cb_luma` được suy ra là bằng `sps_log2_diff_min_qt_min_cb_inter_slice`.

`ph_max_mtt_hierarchy_depth_inter_slice` chỉ ra chiều sâu phân cấp lớn nhất đối với các đơn vị tạo mã thu được từ việc tách cây nhiều kiểu của lá cây tứ phân trong các lát với `slice_type` bằng 0 (B) hoặc 1 (P) liên kết với PH. Giá trị của `ph_max_mtt_hierarchy_depth_inter_slice` phải nằm trong khoảng từ 0 tới $2 * (\text{CtbLog2SizeY} - \text{MinCbLog2SizeY})$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của `ph_max_mtt_hierarchy_depth_inter_slice` được suy ra là bằng `sps_max_mtt_hierarchy_depth_inter_slice`.

`ph_log2_diff_max_bt_min_qt_inter_slice` chỉ ra hiệu số giữa lôgarit cơ số 2 của kích thước lớn nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối tạo mã độ sáng mà có thể được tách bằng cách sử dụng việc tách nhị phân và kích thước nhỏ nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối dạng lá độ sáng thu được từ sự tách cây tứ phân của CTU trong các lát với `slice_type` bằng 0 (B) hoặc 1 (P) liên kết với PH. Giá trị của `ph_log2_diff_max_bt_min_qt_inter_slice` phải nằm trong khoảng từ 0 tới $\text{CtbLog2SizeY} - \text{MinQtLog2SizeInterY}$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của `ph_log2_diff_max_bt_min_qt_inter_slice` được suy ra là bằng `sps_log2_diff_max_bt_min_qt_inter_slice`.

`ph_log2_diff_max_tt_min_qt_inter_slice` chỉ ra hiệu số giữa lôgarit cơ số 2 của kích thước lớn nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối tạo mã độ sáng mà có thể được tách bằng cách sử dụng việc tách tam phân và kích thước nhỏ nhất (chiều rộng hoặc chiều cao) trong các mẫu độ sáng của khối dạng lá độ sáng thu được từ sự tách cây tứ phân của CTU trong các lát với `slice_type` bằng 0 (B) hoặc 1 (P) liên kết với PH. Giá trị của `ph_log2_diff_max_tt_min_qt_inter_slice` phải nằm trong khoảng từ 0 tới

$CtbLog2SizeY - MinQtLog2SizeInterY$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của $ph_log2_diff_max_tt_min_qt_inter_slice$ được suy ra là bằng $sps_log2_diff_max_tt_min_qt_inter_slice$.

$ph_cu_qp_delta_subdiv_inter_slice$ chỉ ra giá trị $cbSubdiv$ lớn nhất của các đơn vị tạo mã mà trong lát ngoài chuyển $cu_qp_delta_abs$ và $cu_qp_delta_sign_flag$. Giá trị của $ph_cu_qp_delta_subdiv_inter_slice$ phải nằm trong khoảng từ 0 tới $2 * (CtbLog2SizeY - MinQtLog2SizeInterY + ph_max_mtt_hierarchy_depth_inter_slice)$, bao gồm các giá trị biên.

Khi không hiện diện, giá trị của $ph_cu_qp_delta_subdiv_inter_slice$ được suy ra là bằng 0.

$ph_cu_chroma_qp_offset_subdiv_inter_slice$ chỉ ra giá trị $cbSubdiv$ lớn nhất của các đơn vị tạo mã trong lát ngoài mà chuyển $cu_chroma_qp_offset_flag$. Giá trị của $ph_cu_chroma_qp_offset_subdiv_inter_slice$ phải nằm trong khoảng từ 0 tới $2 * (CtbLog2SizeY - MinQtLog2SizeInterY + ph_max_mtt_hierarchy_depth_inter_slice)$, bao gồm các giá trị biên.

Khi không hiện diện, giá trị của $ph_cu_chroma_qp_offset_subdiv_inter_slice$ được suy ra là bằng 0.

$ph_temporal_mvp_enabled_flag$ chỉ ra việc liệu các bộ dự báo vector chuyển động theo thời gian có thể được sử dụng để dự báo ngoài đối với các lát liên kết với PH hay không. Nếu $ph_temporal_mvp_enabled_flag$ là bằng 0, các phần tử cú pháp của các lát liên kết với PH phải được ràng buộc sao cho không có bộ dự báo vector chuyển động theo thời gian nào được sử dụng trong việc giải mã các lát. Ngược lại ($ph_temporal_mvp_enabled_flag$ là bằng 1), các bộ dự báo vector chuyển động theo thời gian có thể được sử dụng trong việc giải mã các lát liên kết với PH. Khi không hiện diện, giá trị của $ph_temporal_mvp_enabled_flag$ được suy ra là bằng 0. Khi không có ảnh tham chiếu trong DPB có độ phân giải không gian giống như ảnh hiện tại, giá trị của $ph_temporal_mvp_enabled_flag$ phải bằng 0.

Số lượng lớn nhất của các ứng viên MVP kết hợp dựa trên khối con, `MaxNumSubblockMergeCand`, được rút ra như sau:

```

if( sps_affine_enabled_flag)
    MaxNumSubblockMergeCand = 5 -
    five_minus_max_num_subblock_merge_cand          (88)
else
    MaxNumSubblockMergeCand =

```

```

    sps_sbtmvp_enabled_flag && ph_temporal_mvp_enable_flag

```

Giá trị của `MaxNumSubblockMergeCand` phải nằm trong khoảng từ 0 tới 5, bao gồm các giá trị biên.

`ph_collocated_from_l0_flag` bằng 1 chỉ ra rằng ảnh được sắp xếp được sử dụng để dự báo vector chuyển động theo thời gian được rút ra từ danh sách ảnh tham chiếu 0. `ph_collocated_from_l0_flag` bằng 0 chỉ ra rằng ảnh được sắp xếp được sử dụng để dự báo vector chuyển động theo thời gian được rút ra từ danh sách ảnh tham chiếu 1.

`ph_collocated_ref_idx` chỉ ra chỉ số tham chiếu của ảnh được sắp xếp được sử dụng để dự báo vector chuyển động theo thời gian.

Khi `ph_collocated_from_l0_flag` là bằng 1, `ph_collocated_ref_idx` tham chiếu tới mục nhập trong danh sách ảnh tham chiếu 0, và giá trị của `ph_collocated_ref_idx` phải nằm trong khoảng từ 0 tới `num_ref_entries[0][RplIdx[0]] - 1`, bao gồm các giá trị biên.

Khi `ph_collocated_from_l0_flag` là bằng 0, `ph_collocated_ref_idx` tham chiếu tới mục nhập trong danh sách ảnh tham chiếu 1, và giá trị của `ph_collocated_ref_idx` phải nằm trong khoảng từ 0 tới `num_ref_entries[1][RplIdx[1]] - 1`, bao gồm các giá trị biên.

Khi không hiện diện, giá trị của `ph_collocated_ref_idx` được suy ra là bằng 0.

`mvd_l1_zero_flag` bằng 1 chỉ báo rằng cấu trúc cú pháp `mvd_coding(x0, y0, 1)` không được phân tích và `MvdL1[x0][y0][compIdx]` và `MvdCpL1[x0][y0][cpIdx][compIdx]` được thiết lập bằng bằng 0 đối với

$\text{compIdx} = 0..1$ và $\text{cpIdx} = 0..2$. mvd_l1_zero_flag bằng 0 chỉ báo rằng cấu trúc cú pháp $\text{mvd_coding}(x0, y0, 1)$ được phân tích.

ph_fpel_mmvd_enabled_flag bằng 1 chỉ ra rằng chế độ kết hợp với hiệu số vectơ chuyển động sử dụng độ chính xác mẫu số nguyên trong các lát liên kết với PH. **ph_fpel_mmvd_enabled_flag** bằng 0 chỉ ra rằng chế độ kết hợp với hiệu số vectơ chuyển động có thể sử dụng độ chính xác mẫu phân số trong các lát liên kết với PH. Khi không hiện diện, giá trị của **ph_fpel_mmvd_enabled_flag** được suy ra bằng 0.

ph_disable_bdof_flag bằng 1 chỉ ra rằng dự báo ngoài kép dựa trên dự báo ngoài luồng quang học hai hướng là không được phép trong các lát liên kết với PH. **ph_disable_bdof_flag** bằng 0 chỉ ra rằng dự báo ngoài kép dựa trên dự báo ngoài luồng quang học hai hướng có thể có hoặc có thể không được phép trong các lát liên kết với PH.

Khi **ph_disable_bdof_flag** không hiện diện, điều sau đây được áp dụng:

- Nếu **sps_bdof_enabled_flag** là bằng 1, giá trị của **ph_disable_bdof_flag** được suy ra là bằng 0.
- Ngược lại (**sps_bdof_enabled_flag** là bằng 0), giá trị của **ph_disable_bdof_flag** được suy ra là bằng 1.

ph_disable_dmvr_flag bằng 1 chỉ ra rằng dự báo ngoài kép dựa trên làm mịn bộ giải mã vectơ chuyển động là không được phép trong các lát liên kết với PH. **ph_disable_dmvr_flag** bằng 0 chỉ ra rằng dự báo ngoài kép dựa trên làm mịn bộ giải mã vectơ chuyển động có thể có hoặc có thể không được phép trong các lát liên kết với PH.

Khi **ph_disable_dmvr_flag** không hiện diện, điều sau đây được áp dụng:

- Nếu **sps_dmvr_enabled_flag** là bằng 1, giá trị của **ph_disable_dmvr_flag** được suy ra là bằng 0.
- Ngược lại (**sps_dmvr_enabled_flag** là bằng 0), giá trị của **ph_disable_dmvr_flag** được suy ra là bằng 1.

ph_disable_prof_flag bằng 1 chỉ ra rằng làm mịn dự báo với luồng quang học là không được phép trong các lát liên kết với PH. **ph_disable_prof_flag** bằng

0 chỉ ra rằng làm mịn dự báo với luồng quang học có thể có hoặc có thể không được phép trong các lát liên kết với PH.

Khi `ph_disable_prof_flag` không hiện diện, điều sau đây được áp dụng:

- Nếu `sps_affine_prof_enabled_flag` là bằng 1, giá trị của `ph_disable_prof_flag` được suy ra là bằng 0.
- Ngược lại (`sps_affine_prof_enabled_flag` là bằng 0), giá trị của `ph_disable_prof_flag` được suy ra là bằng 1.

`ph_qp_delta` chỉ ra giá trị ban đầu của `QpY` được sử dụng đối với các khối tạo mã trong ảnh cho đến khi được cải biến bởi giá trị của `CuQpDeltaVal` trong lớp đơn vị tạo mã.

Khi `qp_delta_info_in_ph_flag` là bằng 1, giá trị ban đầu của thông số lượng tử hóa `QpY` đối với tất cả các lát của ảnh, `SliceQpY`, được rút ra như sau:

$$\text{SliceQpY} = 26 + \text{init_qp_minus26} + \text{ph_qp_delta} \quad (89)$$

Giá trị của `SliceQpY` phải nằm trong khoảng từ `-QpBdOffset` tới `+63`, bao gồm các giá trị biên.

`ph_joint_cbr_sign_flag` chỉ ra việc liệu, trong các đơn vị chuyển đổi với `tu_joint_cbr_residual_flag[x0][y0]` bằng 1, các mẫu dư được sắp xếp của cả hai thành phần sắc độ có mang các dấu đảo ngược hay không. Khi `tu_joint_cbr_residual_flag[x0][y0]` bằng 1 đối với khối chuyển đổi, `ph_joint_cbr_sign_flag` bằng 0 chỉ ra rằng dấu của mỗi mẫu dư của thành phần `Cr` (hoặc `Cb`) là giống như dấu của mẫu dư `Cb` (hoặc `Cr`) được sắp xếp và `ph_joint_cbr_sign_flag` bằng 1 chỉ ra rằng dấu của mỗi mẫu dư của thành phần `Cr` (hoặc `Cb`) được cho bởi dấu đảo ngược của mẫu dư `Cb` (hoặc `Cr`) được sắp xếp.

`ph_sao_luma_enabled_flag` bằng 1 chỉ ra rằng SAO là được phép đối với thành phần độ sáng trong tất cả các lát liên kết với PH; `ph_sao_luma_enabled_flag` bằng 0 chỉ ra rằng SAO đối với thành phần độ sáng có thể không được phép đối với một, hoặc nhiều, hoặc tất cả các lát liên kết với PH. Khi `ph_sao_luma_enabled_flag` không hiện diện, nó được suy ra là bằng 0.

ph_sao_chroma_enabled_flag bằng 1 chỉ ra rằng SAO là được phép đối với thành phần sắc độ trong tất cả các lát liên kết với PH; **ph_sao_chroma_enabled_flag** bằng 0 chỉ ra rằng SAO đối với thành phần sắc độ có thể không được phép đối với một, hoặc nhiều, hoặc tất cả các lát liên kết với PH. Khi **ph_sao_chroma_enabled_flag** không hiện diện, nó được suy ra là bằng 0.

ph_dep_quant_enabled_flag bằng 0 chỉ ra rằng lượng tử hóa phụ thuộc là không được phép đối với ảnh hiện tại. **ph_dep_quant_enabled_flag** bằng 1 chỉ ra rằng lượng tử hóa phụ thuộc là được phép đối với ảnh hiện tại. Khi **ph_dep_quant_enabled_flag** không hiện diện, nó được suy ra là bằng 0.

pic_sign_data_hiding_enabled_flag bằng 0 chỉ ra rằng che bit dấu là không được phép đối với ảnh hiện tại. **pic_sign_data_hiding_enabled_flag** bằng 1 chỉ ra rằng che bit dấu là được phép đối với ảnh hiện tại. Khi **pic_sign_data_hiding_enabled_flag** không hiện diện, nó được suy ra là bằng 0.

ph_deblocking_filter_override_flag bằng 1 chỉ ra rằng các thông số giải khối là hiện diện trong PH. **ph_deblocking_filter_override_flag** bằng 0 chỉ ra rằng các thông số giải khối là không hiện diện trong PH. Khi không hiện diện, giá trị của **ph_deblocking_filter_override_flag** được suy ra là bằng 0.

ph_deblocking_filter_disabled_flag bằng 1 chỉ ra rằng hoạt động của bộ lọc giải khối không được áp dụng đối với các lát liên kết với PH. **ph_deblocking_filter_disabled_flag** bằng 0 chỉ ra rằng hoạt động của bộ lọc giải khối được áp dụng đối với các lát liên kết với PH. Khi **ph_deblocking_filter_disabled_flag** không hiện diện, nó được suy ra là bằng **pps_deblocking_filter_disabled_flag**.

ph_beta_offset_div2 và **ph_tc_offset_div2** chỉ ra các dịch vị thông số giải khối đối với β và tC (được chia cho 2) mà được áp dụng cho thành phần độ sáng đối với các lát liên kết với PH. Cả hai giá trị của **ph_beta_offset_div2** và **ph_tc_offset_div2** đều phải nằm trong khoảng từ -12 tới 12, bao gồm các giá trị biên. Khi không hiện diện, các giá trị của **ph_beta_offset_div2** và **ph_tc_offset_div2** lần lượt được suy ra là bằng **pps_beta_offset_div2** và **pps_tc_offset_div2**.

ph_cb_beta_offset_div2 và **ph_cb_tc_offset_div2** chỉ ra các dịch vị thông số giải khối đối với β và tC (được chia cho 2) mà được áp dụng cho thành phần Cb đối với các lát liên kết với PH. Cả hai giá trị của **ph_cb_beta_offset_div2** và **ph_cb_tc_offset_div2** đều phải nằm trong khoảng từ -12 tới 12, bao gồm các giá trị biên. Khi không hiện diện, các giá trị của **ph_cb_beta_offset_div2** và **ph_cb_tc_offset_div2** lần lượt được suy ra là bằng **pps_cb_beta_offset_div2** và **pps_cb_tc_offset_div2**.

ph_cr_beta_offset_div2 và **ph_cr_tc_offset_div2** chỉ ra các dịch vị thông số giải khối đối với β và tC (được chia cho 2) mà được áp dụng cho thành phần Cr đối với các lát liên kết với PH. Cả hai giá trị của **ph_cr_beta_offset_div2** và **ph_cr_tc_offset_div2** đều phải nằm trong khoảng từ -12 tới 12, bao gồm các giá trị biên. Khi không hiện diện, các giá trị của **ph_cr_beta_offset_div2** và **ph_cr_tc_offset_div2** lần lượt được suy ra là bằng **pps_cr_beta_offset_div2** và **pps_cr_tc_offset_div2**.

ph_extension_length chỉ ra chiều dài của dữ liệu mở rộng PH tính bằng byte, không bao gồm các bit được sử dụng để báo tín hiệu bản thân **ph_extension_length**. Giá trị của **ph_extension_length** phải nằm trong khoảng từ 0 tới 256, bao gồm các giá trị biên. Khi không hiện diện, giá trị của **ph_extension_length** được suy ra là bằng 0.

ph_extension_data_byte có thể có giá trị bất kỳ. Các bộ giải mã tương thích với phiên bản này của Đặc tả này phải bỏ qua giá trị của **ph_extension_data_byte**. Giá trị của nó không ảnh hưởng tới sự tương thích của bộ giải mã với các tiêu sử được chỉ ra trong phiên bản này của đặc tả.

3.3. Cú pháp SH và các ngữ nghĩa

Trong văn bản dự thảo VVC cuối cùng, cú pháp SH và các ngữ nghĩa là như sau:

	Mô tả
slice_header() {	
picture_header_in_slice_header_flag	u(1)
if (picture_header_in_slice_header_flag)	
picture_header_structure()	
if (subpic_info_present_flag)	
slice_subpic_id	u(v)
if ((rect_slice_flag && NumSlicesInSubpic[CurrSubpicIdx] > 1) (!rect_slice_flag && NumTilesInPic > 1))	

slice_address	u(v)
for(i = 0; i < NumExtraShCac bit; i++)	
sh_extra_bit[i]	u(1)
if(!rect slice flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)
if(ph inter slice allowed flag)	
slice_type	ue(v)
if(sps_alf_enabled_flag && !alf_info_in_ph_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
if(sps_ccalf_enabled_flag) {	
slice_cc_alf_cb_enabled_flag	u(1)
if(slice_cc_alf_cb_enabled_flag)	
slice_cc_alf_cb_aps_id	u(3)
slice_cc_alf_cr_enabled_flag	u(1)
if(slice_cc_alf_cr_enabled_flag)	
slice_cc_alf_cr_aps_id	u(3)
}	
}	
}	
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(!rpl_info_in_ph_flag && ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag))	
ref_pic_lists()	
if((rpl_info_in_ph_flag ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag)) && (slice_type != I && num_ref_entries[0][RplIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
if(num_ref_entries[i][RplIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
if(slice_type != I) {	
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(ph_temporal_mvp_enabled_flag && !rpl_info_in_ph_flag) {	
if(slice_type == B)	
slice_collocated_from_10_flag	u(1)
if((slice_collocated_from_10_flag && NumRefIdxActive[0] > 1) (!slice_collocated_from_10_flag && NumRefIdxActive[1] > 1))	
slice_collocated_ref_idx	ue(v)

}	
if(!wp_info_in_ph_flag && ((pps_weighted_pred_flag && slice_type == P)	
(pps weighted bipred flag && slice type == B)))	
pred_weight_table()	
}	
if(!qp_delta_info_in_ph_flag)	
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_qp_offset	se(v)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(sps_sao_enabled_flag && !sao_info_in_ph_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(deblocking_filter_override_enabled_flag && !dbf_info_in_ph_flag)	
slice_deblocking_filter_override_flag	u(1)
if(slice_deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
slice_cb_beta_offset_div2	se(v)
slice_cb_tc_offset_div2	se(v)
slice_cr_beta_offset_div2	se(v)
slice_cr_tc_offset_div2	se(v)
}	
}	
slice_ts_residual_coding_disabled_flag	u(1)
if(ph_lmcs_enabled_flag)	
slice_lmcs_enabled_flag	u(1)
if(ph_scaling_list_enabled_flag)	
slice_scaling_list_present_flag	u(1)
if(NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

Biến số `CuQpDeltaVal`, chỉ ra hiệu số giữa thông số lượng tử hóa độ sáng đối với đơn vị tạo mã chứa `cu_qp_delta_abs` và dự báo của nó, được thiết lập bằng 0. Tất cả các biến số `CuQpOffsetCb`, `CuQpOffsetCr`, và `CuQpOffsetCbCr`, chỉ ra các giá trị được sử dụng khi xác định các giá trị tương ứng của các thông số lượng tử hóa `Qp'Cb`, `Qp'Cr`, và `Qp'CbCr` đối với đơn vị tạo mã chứa `cu_chroma_qp_offset_flag`, đều được thiết lập bằng 0.

`picture_header_in_slice_header_flag` bằng 1 chỉ ra rằng cấu trúc cú pháp PH hiện diện trong tiêu đề lát. `picture_header_in_slice_header_flag` bằng 0 chỉ ra rằng cấu trúc cú pháp PH không hiện diện trong tiêu đề lát.

Sự tương thích dòng bit yêu cầu rằng giá trị của `picture_header_in_slice_header_flag` phải là giống nhau trong tất cả các lát được tạo mã trong CLVS.

Khi `picture_header_in_slice_header_flag` là bằng 1 đối với lát được tạo mã, sự tương thích dòng bit yêu cầu rằng không có đơn vị NAL VCL với `nal_unit_type` bằng PH_NUT sẽ hiện diện trong CLVS.

Khi `picture_header_in_slice_header_flag` là bằng 0, tất cả các lát được tạo mã trong ảnh hiện tại phải có `picture_header_in_slice_header_flag` là bằng 0, và PU hiện tại phải có đơn vị NAL PH.

`slice_subpic_id` chỉ ra ID ảnh con của ảnh con mà chứa lát. Nếu `slice_subpic_id` là hiện diện, giá trị của biến số `CurrSubpicIdx` được rút ra sao cho `SubpicIdVal[CurrSubpicIdx]` là bằng `slice_subpic_id`. Ngược lại (`slice_subpic_id` không hiện diện), `CurrSubpicIdx` được rút ra là bằng 0. Chiều dài của `slice_subpic_id` bằng `sps_subpic_id_len_minus1 + 1` bit.

`slice_address` chỉ ra địa chỉ lát của lát. Khi không hiện diện, giá trị của `slice_address` được suy ra là bằng 0. Khi `rect_slice_flag` là bằng 1 và `NumSlicesInSubpic[CurrSubpicIdx]` là bằng 1, giá trị của `slice_address` được suy ra là bằng 0.

Nếu `rect_slice_flag` là bằng 0, điều sau đây được áp dụng:

- Địa chỉ lát bằng chỉ số ô ảnh quét màn.
- Chiều dài của `slice_address` bằng $\text{Ceil}(\text{Log2}(\text{NumTilesInPic}))$ bit.

- Giá trị của `slice_address` phải nằm trong khoảng từ 0 tới `NumTilesInPic - 1`, bao gồm các giá trị biên.

Ngược lại (`rect_slice_flag` là bằng 1), điều sau đây được áp dụng:

- Địa chỉ lát bằng chỉ số lát bậc ảnh con của lát.
- Chiều dài của `slice_address` bằng $\text{Ceil}(\text{Log2}(\text{NumSlicesInSubpic}[\text{CurrSubpicIdx}]))$ bit.

- Giá trị của `slice_address` phải nằm trong khoảng từ 0 tới `NumSlicesInSubpic[CurrSubpicIdx] - 1`, bao gồm các giá trị biên.

Sự tương thích dòng bit yêu cầu rằng các điều kiện ràng buộc sau đây được áp dụng:

- Nếu `rect_slice_flag` là bằng 0 hoặc `subpic_info_present_flag` là bằng 0, giá trị của `slice_address` phải không bằng giá trị của `slice_address` của đơn vị NAL lát được tạo mã bất kỳ khác của cùng ảnh được tạo mã.

- Ngược lại, cặp giá trị `slice_subpic_id` và `slice_address` phải không bằng cặp giá trị `slice_subpic_id` và `slice_address` của đơn vị NAL lát được tạo mã bất kỳ khác của cùng ảnh được tạo mã.

- Các hình dạng của các lát của ảnh phải sao cho mỗi CTU, khi được giải mã, phải có toàn bộ đường biên trái và toàn bộ đường biên đỉnh của nó chứa đường biên ảnh hoặc chứa các biên của (các) CTU được giải mã trước đó.

`sh_extra_bit[i]` có thể bằng 1 hoặc 0. Các bộ giải mã tương thích với phiên bản này của Đặc tả này phải bỏ qua giá trị của `sh_extra_bit[i]`. Giá trị của nó không ảnh hưởng tới sự tương thích của bộ giải mã với các tiêu sử được chỉ ra trong phiên bản này của đặc tả.

`num_tiles_in_slice_minus1` plus 1, khi hiện diện, chỉ ra số lượng của các ô ảnh trong lát. Giá trị của `num_tiles_in_slice_minus1` phải nằm trong khoảng từ 0 tới `NumTilesInPic - 1`, bao gồm các giá trị biên.

Biến số `NumCtusInCurrSlice`, mà chỉ ra số lượng của các CTU trong lát hiện tại, và danh sách `CtbAddrInCurrSlice[i]`, đối với `i` nằm trong khoảng từ 0 tới

NumCtusInCurrSlice - 1, bao gồm các giá trị biên, chỉ ra địa chỉ quét mảnh ảnh của CTB thứ i bên trong lát, được rút ra như sau:

```

    if( rect_slice_flag) {
        picLevelSliceIdx = slice_address
        for(j = 0; j < CurrSubpicIdx; j++)
            picLevelSliceIdx += NumSlicesInSubpic[j]
        NumCtusInCurrSlice = NumCtusInSlice[picLevelSliceIdx]
        for(i = 0; i < NumCtusInCurrSlice; i++)
            CtbAddrInCurrSlice[i] = CtbAddrInSlice[picLevelSliceIdx][i]
(117)
    } else {
        NumCtusInCurrSlice = 0
        for(tileIdx = slice_address; tileIdx <=
slice_address + num_tiles_in_slice_minus1; tileIdx++) {
            tileX = tileIdx % NumTileColumns
            tileY = tileIdx/NumTileColumns
            for(ctbY = tileRowBd[tileY]; ctbY < tileRowBd[tileY + 1];
ctbY++) {
                for(ctbX = tileColBd[tileX]; ctbX < tileColBd[tileX + 1];
ctbX++) {
                    CtbAddrInCurrSlice[NumCtusInCurrSlice] =
ctbY * PicWidthInCtb + ctbX
                    NumCtusInCurrSlice++
                }
            }
        }
    }

```

Các biến số SubpicLeftBoundaryPos, SubpicTopBoundaryPos, SubpicRightBoundaryPos, và SubpicBotBoundaryPos được rút ra như sau:

```

    if( subpic_treated_as_pic_flag[CurrSubpicIdx]) {
        SubpicLeftBoundaryPos =

```

```

subpic_ctu_top_left_x[CurrSubpicIdx] * CtbSizeY
SubpicRightBoundaryPos =
Min(pic_width_max_in_luma_samples - 1,
    (subpic_ctu_top_left_x[CurrSubpicIdx] +
    subpic_width_minus1[CurrSubpicIdx] + 1) * CtbSizeY - 1)
SubpicTopBoundaryPos =
subpic_ctu_top_left_y[CurrSubpicIdx] * CtbSizeY      (118)
SubpicBotBoundaryPos =
Min(pic_height_max_in_luma_samples - 1,
    (subpic_ctu_top_left_y[CurrSubpicIdx] +
    subpic_height_minus1[CurrSubpicIdx] + 1) * CtbSizeY - 1)
}

```

slice_type chỉ ra kiểu tạo mã của lát theo Bảng 9.

Bảng 9 – Liên kết tên với slice_type

slice_type	Tên của slice_type
0	B (lát B)
1	P (lát P)
2	I (lát I)

Khi không hiện diện, giá trị của slice_type được suy ra là bằng 2.

Khi ph_intra_slice_allowed_flag là bằng 0, giá trị của slice_type phải bằng 0 hoặc 1. Khi nal_unit_type nằm trong khoảng từ IDR_W_RADL tới CRA_NUT, bao gồm các giá trị biên, và vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]] là bằng 1, slice_type phải bằng 2.

Các biến số MinQtLog2SizeY, MinQtLog2SizeC, MinQtSizeY, MinQtSizeC, MaxBtSizeY, MaxBtSizeC, MinBtSizeY, MaxTtSizeY, MaxTtSizeC, MinTtSizeY, MaxMttDepthY và MaxMttDepthC được rút ra như sau:

- Nếu slice_type bằng 2 (I), điều sau đây được áp dụng:

```

MinQtLog2SizeY =
MinCbLog2SizeY + ph_log2_diff_min_qt_min_cb_intra_slice_luma
(119)

```

$$\begin{aligned} \text{MinQtLog2SizeC} = \\ \text{MinCbLog2SizeY} + \text{ph_log2_diff_min_qt_min_cb_intra_slice_chroma} \end{aligned} \quad (120)$$

$$\begin{aligned} \text{MaxBtSizeY} = 1 \ll \\ (\text{MinQtLog2SizeY} + \text{ph_log2_diff_max_bt_min_qt_intra_slice_luma}) \end{aligned} \quad (121)$$

$$\begin{aligned} \text{MaxBtSizeC} = 1 \ll \\ (\text{MinQtLog2SizeC} + \text{ph_log2_diff_max_bt_min_qt_intra_slice_chroma}) \end{aligned} \quad (122)$$

$$\begin{aligned} \text{MaxTtSizeY} = 1 \ll \\ (\text{MinQtLog2SizeY} + \text{ph_log2_diff_max_tt_min_qt_intra_slice_luma}) \end{aligned} \quad (123)$$

$$\begin{aligned} \text{MaxTtSizeC} = 1 \ll \\ (\text{MinQtLog2SizeC} + \text{ph_log2_diff_max_tt_min_qt_intra_slice_chroma}) \end{aligned} \quad (124)$$

$$\text{MaxMttDepthY} = \text{ph_max_mtt_hierarchy_depth_intra_slice_luma} \quad (125)$$

$$\begin{aligned} \text{MaxMttDepthC} = \\ \text{ph_max_mtt_hierarchy_depth_intra_slice_chroma} \end{aligned} \quad (126)$$

$$\text{CuQpDeltaSubdiv} = \text{ph_cu_qp_delta_subdiv_intra_slice} \quad (127)$$

$$\begin{aligned} \text{CuChromaQpOffsetSubdiv} = \\ \text{ph_cu_chroma_qp_offset_subdiv_intra_slice} \end{aligned} \quad (128)$$

– Ngược lại (slice_type bằng 0 (B) hoặc 1 (P)), điều sau đây được áp dụng:

$$\begin{aligned} \text{MinQtLog2SizeY} = \\ \text{MinCbLog2SizeY} + \text{ph_log2_diff_min_qt_min_cb_inter_slice} \end{aligned} \quad (129)$$

$$\begin{aligned} \text{MinQtLog2SizeC} = \\ \text{MinCbLog2SizeY} + \text{ph_log2_diff_min_qt_min_cb_inter_slice} \end{aligned} \quad (130)$$

$$\begin{aligned} \text{MaxBtSizeY} = 1 \ll \\ (\text{MinQtLog2SizeY} + \text{ph_log2_diff_max_bt_min_qt_inter_slice}) \end{aligned} \quad (131)$$

$$\begin{aligned} \text{MaxBtSizeC} &= 1 \ll \\ (\text{MinQtLog2SizeC} + \text{ph_log2_diff_max_bt_min_qt_inter_slice}) & \quad (132) \end{aligned}$$

$$\begin{aligned} \text{MaxTtSizeY} &= 1 \ll \\ (\text{MinQtLog2SizeY} + \text{ph_log2_diff_max_tt_min_qt_inter_slice}) & \quad (133) \end{aligned}$$

$$\begin{aligned} \text{MaxTtSizeC} &= 1 \ll \\ (\text{MinQtLog2SizeC} + \text{ph_log2_diff_max_tt_min_qt_inter_slice}) & \quad (134) \end{aligned}$$

$$\begin{aligned} \text{MaxMttDepthY} &= \text{ph_max_mtt_hierarchy_depth_inter_slice} \\ (135) \end{aligned}$$

$$\begin{aligned} \text{MaxMttDepthC} &= \text{ph_max_mtt_hierarchy_depth_inter_slice} \\ (136) \end{aligned}$$

$$\begin{aligned} \text{CuQpDeltaSubdiv} &= \text{ph_cu_qp_delta_subdiv_inter_slice} \\ (137) \end{aligned}$$

$$\begin{aligned} \text{CuChromaQpOffsetSubdiv} &= \\ \text{ph_cu_chroma_qp_offset_subdiv_inter_slice} & \quad (138) \end{aligned}$$

– Điều sau đây được áp dụng:

$$\begin{aligned} \text{MinQtSizeY} &= 1 \ll \text{MinQtLog2SizeY} \\ (139) \end{aligned}$$

$$\begin{aligned} \text{MinQtSizeC} &= 1 \ll \text{MinQtLog2Size} \\ (140) \end{aligned}$$

$$\begin{aligned} \text{MinBtSizeY} &= 1 \ll \text{MinCbLog2SizeY} \\ (141) \end{aligned}$$

$$\begin{aligned} \text{MinTtSizeY} &= 1 \ll \text{MinCbLog2SizeY} \\ (142) \end{aligned}$$

slice_alf_enabled_flag bằng 1 chỉ ra rằng bộ lọc vòng thích ứng là được phép và có thể được áp dụng cho thành phần màu Y, Cb, hoặc Cr trong lát. **slice_alf_enabled_flag** bằng 0 chỉ ra rằng bộ lọc vòng thích ứng là không được phép đối với tất cả các thành phần màu trong lát. Khi không hiện diện, giá trị của **slice_alf_enabled_flag** được suy ra là bằng **ph_alf_enabled_flag**.

slice_num_alf_aps_ids_luma chỉ ra số lượng các APS ALF mà lát tham chiếu tới. Khi **slice_alf_enabled_flag** là bằng 1 và **slice_num_alf_aps_ids_luma**

không hiện diện, giá trị của `slice_num_alf_aps_ids_luma` được suy ra là bằng giá trị của `ph_num_alf_aps_ids_luma`.

`slice_alf_aps_id_luma[i]` chỉ ra `adaptation_parameter_set_id` của APS ALF thứ *i* mà thành phần độ sáng của lát tham chiếu tới. `TemporalId` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `slice_alf_aps_id_luma[i]` phải nhỏ hơn hoặc bằng `TemporalId` của đơn vị NAL lát được tạo mã. Khi `slice_alf_enabled_flag` là bằng 1 và `slice_alf_aps_id_luma[i]` không hiện diện, giá trị của `slice_alf_aps_id_luma[i]` được suy ra là bằng giá trị của `ph_alf_aps_id_luma[i]`.

Giá trị của `alf_luma_filter_signal_flag` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `slice_alf_aps_id_luma[i]` phải bằng 1.

`slice_alf_chroma_idc` bằng 0 chỉ ra rằng bộ lọc vòng thích ứng không được áp dụng cho các thành phần màu Cb và Cr. `slice_alf_chroma_idc` bằng 1 chỉ báo rằng bộ lọc vòng thích ứng được áp dụng cho thành phần màu Cb. `slice_alf_chroma_idc` bằng 2 chỉ báo rằng bộ lọc vòng thích ứng được áp dụng cho thành phần màu Cr. `slice_alf_chroma_idc` bằng 3 chỉ báo rằng bộ lọc vòng thích ứng được áp dụng cho các thành phần màu Cb và Cr. Khi `slice_alf_chroma_idc` không hiện diện, nó được suy ra là bằng `ph_alf_chroma_idc`.

`slice_alf_aps_id_chroma` chỉ ra `adaptation_parameter_set_id` của APS ALF mà thành phần sắc độ của lát tham chiếu tới. `TemporalId` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `slice_alf_aps_id_chroma` phải nhỏ hơn hoặc bằng `TemporalId` của đơn vị NAL lát được tạo mã. Khi `slice_alf_enabled_flag` là bằng 1 và `slice_alf_aps_id_chroma` không hiện diện, giá trị của `slice_alf_aps_id_chroma` được suy ra là bằng giá trị của `ph_alf_aps_id_chroma`.

Giá trị của `alf_chroma_filter_signal_flag` của đơn vị NAL APS có `aps_params_type` bằng `ALF_APS` và `adaptation_parameter_set_id` bằng `slice_alf_aps_id_chroma` phải bằng 1.

slice_cc_alf_cb_enabled_flag bằng 0 chỉ ra rằng bộ lọc giao thành phần không được áp dụng cho thành phần màu Cb. **slice_cc_alf_cb_enabled_flag** bằng 1 chỉ báo rằng bộ lọc giao thành phần là được phép và có thể được áp dụng cho thành phần màu Cb. Khi **slice_cc_alf_cb_enabled_flag** không hiện diện, nó được suy ra là bằng **ph_cc_alf_cb_enabled_flag**.

slice_cc_alf_cb_aps_id chỉ ra **adaptation_parameter_set_id** mà thành phần màu Cb của lát tham chiếu tới.

TemporalId của đơn vị NAL APS có **aps_params_type** bằng **ALF_APS** và **adaptation_parameter_set_id** bằng **slice_cc_alf_cb_aps_id** phải nhỏ hơn hoặc bằng **TemporalId** của đơn vị NAL lát được tạo mã. Khi **slice_cc_alf_cb_enabled_flag** là bằng 1 và **slice_cc_alf_cb_aps_id** không hiện diện, giá trị của **slice_cc_alf_cb_aps_id** được suy ra là bằng giá trị của **ph_cc_alf_cb_aps_id**.

Giá trị của **alf_cc_cb_filter_signal_flag** của đơn vị NAL APS có **aps_params_type** bằng **ALF_APS** và **adaptation_parameter_set_id** bằng **slice_cc_alf_cb_aps_id** phải bằng 1.

slice_cc_alf_cr_enabled_flag bằng 0 chỉ ra rằng bộ lọc giao thành phần không được áp dụng cho thành phần màu Cr. **slice_cc_alf_cr_enabled_flag** bằng 1 chỉ báo rằng bộ lọc vòng thích ứng giao thành phần là được phép và có thể được áp dụng cho thành phần màu Cr. Khi **slice_cc_alf_cr_enabled_flag** không hiện diện, nó được suy ra là bằng **ph_cc_alf_cr_enabled_flag**.

slice_cc_alf_cr_aps_id chỉ ra **adaptation_parameter_set_id** mà thành phần màu Cr của lát tham chiếu tới. **TemporalId** của đơn vị NAL APS có **aps_params_type** bằng **ALF_APS** và **adaptation_parameter_set_id** bằng **slice_cc_alf_cr_aps_id** phải nhỏ hơn hoặc bằng **TemporalId** của đơn vị NAL lát được tạo mã. Khi **slice_cc_alf_cr_enabled_flag** là bằng 1 và **slice_cc_alf_cr_aps_id** không hiện diện, giá trị của **slice_cc_alf_cr_aps_id** được suy ra là bằng giá trị của **ph_cc_alf_cr_aps_id**.

Giá trị của **alf_cc_cr_filter_signal_flag** của đơn vị NAL APS có **aps_params_type** bằng **ALF_APS** và **adaptation_parameter_set_id** bằng **slice_cc_alf_cr_aps_id** phải bằng 1.

colour_plane_id nhận dạng mặt phẳng màu liên kết với lát hiện tại khi **separate_colour_plane_flag** là bằng 1. Giá trị của **colour_plane_id** phải nằm trong khoảng từ 0 tới 2, bao gồm các giá trị biên. Các giá trị **colour_plane_id** 0, 1 và 2 lần lượt tương ứng với các mặt phẳng Y, Cb và Cr. Giá trị 3 của **colour_plane_id** được dành lại để sử dụng trong tương lai bởi ITU-T | ISO/IEC.

LƯU Ý 1 – Không có sự phụ thuộc giữa các quá trình giải mã của các mặt phẳng màu khác nhau của một ảnh.

num_ref_idx_active_override_flag bằng 1 chỉ ra rằng phần tử cú pháp **num_ref_idx_active_minus1[0]** hiện diện đối với các lát P và B và phần tử cú pháp **num_ref_idx_active_minus1[1]** hiện diện đối với các lát B. **num_ref_idx_active_override_flag** bằng 0 chỉ ra rằng các phần tử cú pháp **num_ref_idx_active_minus1[0]** và **num_ref_idx_active_minus1[1]** là không hiện diện. Khi không hiện diện, giá trị của **num_ref_idx_active_override_flag** được suy ra là bằng 1.

num_ref_idx_active_minus1[i] được sử dụng để rút ra biến số **NumRefIdxActive[i]** như được chỉ ra bởi Phương trình 143. Giá trị của **num_ref_idx_active_minus1[i]** phải nằm trong khoảng từ 0 tới 14, bao gồm các giá trị biên.

Đối với *i* bằng 0 hoặc 1, khi lát hiện tại là lát B, **num_ref_idx_active_override_flag** là bằng 1, và **num_ref_idx_active_minus1[i]** không hiện diện, **num_ref_idx_active_minus1[i]** được suy ra là bằng 0.

Khi lát hiện tại là lát P, **num_ref_idx_active_override_flag** là bằng 1, và **num_ref_idx_active_minus1[0]** không hiện diện, **num_ref_idx_active_minus1[0]** được suy ra là bằng 0.

Biến số **NumRefIdxActive[i]** được rút ra như sau:

```

for(i = 0; i < 2; i++) {
  if(slice_type == B || (slice_type == P && i == 0)) {
    if(num_ref_idx_active_override_flag)
      NumRefIdxActive[i] = num_ref_idx_active_minus1[i] + 1
  }
  else {

```

(143)

```

        if( num_ref_entries[i][RplIdx[i]] >=
num_ref_idx_default_active_minus1[i] + 1)
            NumRefIdxActive[i] =
num_ref_idx_default_active_minus1[i] + 1
        else
            NumRefIdxActive[i] = num_ref_entries[i][RplIdx[i]]
    }
} else /* slice_type == I || (slice_type == P && i == 1) */
    NumRefIdxActive[i] = 0
}

```

Giá trị của NumRefIdxActive[i] - 1 chỉ ra chỉ số tham chiếu lớn nhất đối với danh sách ảnh tham chiếu i mà có thể được sử dụng để giải mã lát. Khi giá trị của NumRefIdxActive[i] là bằng 0, không có chỉ số tham chiếu nào đối với danh sách ảnh tham chiếu i có thể được sử dụng để giải mã lát.

Khi lát hiện tại là lát P, giá trị của NumRefIdxActive[0] phải lớn hơn 0.

Khi lát hiện tại là lát B, cả NumRefIdxActive[0] lẫn NumRefIdxActive[1] phải lớn hơn 0.

cabac_init_flag chỉ ra phương pháp để xác định bảng khởi tạo được sử dụng trong quá trình khởi tạo đối với các biến số ngữ cảnh. Khi cabac_init_flag không hiện diện, nó được suy ra là bằng 0.

slice_collocated_from_10_flag bằng 1 chỉ ra rằng ảnh được sắp xếp được sử dụng để dự báo vector chuyển động theo thời gian được rút ra từ danh sách ảnh tham chiếu 0. slice_collocated_from_10_flag bằng 0 chỉ ra rằng ảnh được sắp xếp được sử dụng để dự báo vector chuyển động theo thời gian được rút ra từ danh sách ảnh tham chiếu 1.

Khi slice_type là bằng B hoặc P, ph_temporal_mvp_enabled_flag là bằng 1, và slice_collocated_from_10_flag không hiện diện, điều sau đây được áp dụng:

- Nếu rpl_info_in_ph_flag là bằng 1, slice_collocated_from_10_flag được suy ra là bằng ph_collocated_from_10_flag.
- Ngược lại (rpl_info_in_ph_flag là bằng 0 và slice_type là bằng P), giá trị của slice_collocated_from_10_flag được suy ra là bằng 1.

`slice_collocated_ref_idx` chỉ ra chỉ số tham chiếu của ảnh được sắp xếp được sử dụng để dự báo vector chuyển động theo thời gian.

Khi `slice_type` là bằng P hoặc khi `slice_type` là bằng B và `slice_collocated_from_l0_flag` là bằng 1, `slice_collocated_ref_idx` tham chiếu tới mục nhập trong danh sách ảnh tham chiếu 0, và giá trị của `slice_collocated_ref_idx` phải nằm trong khoảng từ 0 tới `NumRefIdxActive[0] - 1`, bao gồm các giá trị biên.

Khi `slice_type` là bằng B và `slice_collocated_from_l0_flag` là bằng 0, `slice_collocated_ref_idx` tham chiếu tới mục nhập trong danh sách ảnh tham chiếu 1, và giá trị của `slice_collocated_ref_idx` phải nằm trong khoảng từ 0 tới `NumRefIdxActive[1] - 1`, bao gồm các giá trị biên.

Khi `slice_collocated_ref_idx` không hiện diện, điều sau đây được áp dụng:

- Nếu `rpl_info_in_ph_flag` là bằng 1, giá trị của `slice_collocated_ref_idx` được suy ra là bằng `ph_collocated_ref_idx`.
- Ngược lại (`rpl_info_in_ph_flag` là bằng 0), giá trị của `slice_collocated_ref_idx` được suy ra là bằng 0.

Sự tương thích dòng bit yêu cầu rằng ảnh được tham chiếu tới bởi `slice_collocated_ref_idx` phải là giống nhau đối với tất cả các lát của ảnh được tạo mã.

Sự tương thích dòng bit yêu cầu rằng các giá trị của `pic_width_in_luma_samples` và `pic_height_in_luma_samples` của ảnh tham chiếu được tham chiếu tới bởi `slice_collocated_ref_idx` phải lần lượt bằng các giá trị của `pic_width_in_luma_samples` và `pic_height_in_luma_samples`, của ảnh hiện tại, và `RprConstraintsActive[slice_collocated_from_l0_flag ? 0 : 1][slice_collocated_ref_idx]` phải bằng 0.

`slice_qp_delta` chỉ ra giá trị ban đầu của `QpY` được sử dụng để tạo mã các khối trong lát cho đến khi được cải biến bởi giá trị của `CuQpDeltaVal` trong lớp đơn vị tạo mã.

Khi `qp_delta_info_in_ph_flag` là bằng 0, giá trị ban đầu của thông số lượng tử hóa `QpY` đối với lát, `SliceQpY`, được rút ra như sau:

$$\text{SliceQpY} = 26 + \text{init_qp_minus26} + \text{slice_qp_delta} \quad (144)$$

Giá trị của SliceQpY phải nằm trong khoảng từ $-QpBdOffset$ tới $+63$, bao gồm các giá trị biên.

Khi một trong số các điều kiện sau đây là đúng:

- Giá trị của wp_info_in_ph_flag là bằng 1, pps_weighted_pred_flag là bằng 1, và slice_type là bằng P.

- Giá trị của wp_info_in_ph_flag là bằng 1, pps_weighted_bipred_flag là bằng 1, và slice_type là bằng B.

thì điều sau đây được áp dụng:

- Giá trị của NumRefIdxActive[0] phải nhỏ hơn hoặc bằng giá trị của NumWeightsL0.

- Đối với mỗi chỉ số ảnh tham chiếu RefPicList[0][i] đối với i nằm trong khoảng từ 0 tới NumRefIdxActive[0] - 1, bao gồm các giá trị biên, trọng số độ sáng, trọng số Cb, và trọng số Cr mà áp dụng cho chỉ số ảnh tham chiếu lần lượt là LumaWeightL0[i], ChromaWeightL0[0][i], và ChromaWeightL0[1][i].

Khi wp_info_in_ph_flag là bằng 1, pps_weighted_bipred_flag là bằng 1, và slice_type là bằng B, điều sau đây được áp dụng:

- Giá trị của NumRefIdxActive[1] phải nhỏ hơn hoặc bằng giá trị của NumWeightsL1.

- Đối với mỗi chỉ số ảnh tham chiếu RefPicList[1][i] đối với i nằm trong khoảng từ 0 tới NumRefIdxActive[1] - 1, bao gồm các giá trị biên, trọng số độ sáng, trọng số Cb, và trọng số Cr mà áp dụng cho chỉ số ảnh tham chiếu lần lượt là LumaWeightL1[i], ChromaWeightL1[0][i], và ChromaWeightL1[1][i].

slice_cb_qp_offset chỉ ra hiệu số cần được thêm vào giá trị của pps_cb_qp_offset khi xác định giá trị của thông số lượng tử hóa Qp'Cb. Giá trị của slice_cb_qp_offset phải nằm trong khoảng từ -12 tới $+12$, bao gồm các giá trị biên. Khi slice_cb_qp_offset không hiện diện, nó được suy ra là bằng 0. Giá trị của pps_cb_qp_offset + slice_cb_qp_offset phải nằm trong khoảng từ -12 tới $+12$, bao gồm các giá trị biên.

slice_cr_qp_offset chỉ ra hiệu số cần được thêm vào giá trị của **pps_cr_qp_offset** khi xác định giá trị của thông số lượng tử hóa Q_p' Cr. Giá trị của **slice_cr_qp_offset** phải nằm trong khoảng từ -12 tới +12, bao gồm các giá trị biên. Khi **slice_cr_qp_offset** không hiện diện, nó được suy ra là bằng 0. Giá trị của **pps_cr_qp_offset** + **slice_cr_qp_offset** phải nằm trong khoảng từ -12 tới +12, bao gồm các giá trị biên.

slice_joint_cbr_qp_offset chỉ ra hiệu số cần được thêm vào giá trị của **pps_joint_cbr_qp_offset_value** khi xác định giá trị của Q_p' cbCr. Giá trị của **slice_joint_cbr_qp_offset** phải nằm trong khoảng từ -12 tới +12, bao gồm các giá trị biên. Khi **slice_joint_cbr_qp_offset** không hiện diện, nó được suy ra là bằng 0. Giá trị của **pps_joint_cbr_qp_offset_value** + **slice_joint_cbr_qp_offset** phải nằm trong khoảng từ -12 tới +12, bao gồm các giá trị biên.

cu_chroma_qp_offset_enabled_flag bằng 1 chỉ ra rằng **cu_chroma_qp_offset_flag** có thể hiện diện trong khối chuyển đổi và cú pháp tạo mã bảng màu. **cu_chroma_qp_offset_enabled_flag** bằng 0 chỉ ra rằng **cu_chroma_qp_offset_flag** không hiện diện trong khối chuyển đổi hoặc cú pháp tạo mã bảng màu. Khi không hiện diện, giá trị của **cu_chroma_qp_offset_enabled_flag** được suy ra là bằng 0.

slice_sao_luma_flag bằng 1 chỉ ra rằng SAO là được phép đối với thành phần độ sáng trong lát hiện tại; **slice_sao_luma_flag** bằng 0 chỉ ra rằng SAO là không được phép đối với thành phần độ sáng trong lát hiện tại. Khi **slice_sao_luma_flag** không hiện diện, nó được suy ra là bằng **ph_sao_luma_enabled_flag**.

slice_sao_chroma_flag bằng 1 chỉ ra rằng SAO là được phép đối với thành phần sắc độ trong lát hiện tại; **slice_sao_chroma_flag** bằng 0 chỉ ra rằng SAO là không được phép đối với thành phần sắc độ trong lát hiện tại. Khi **slice_sao_chroma_flag** không hiện diện, nó được suy ra là bằng **ph_sao_chroma_enabled_flag**.

slice_deblocking_filter_override_flag bằng 1 chỉ ra rằng các thông số giải khối là hiện diện trong tiêu đề lát. **slice_deblocking_filter_override_flag** bằng 0

chỉ ra rằng các thông số giải khối là không hiện diện trong tiêu đề lát. Khi không hiện diện, giá trị của `slice_deblocking_filter_override_flag` được suy ra là bằng `ph_deblocking_filter_override_flag`.

`slice_deblocking_filter_disabled_flag` bằng 1 chỉ ra rằng hoạt động của bộ lọc giải khối không được áp dụng đối với lát hiện tại. `slice_deblocking_filter_disabled_flag` bằng 0 chỉ ra rằng hoạt động của bộ lọc giải khối được áp dụng đối với lát hiện tại. Khi `slice_deblocking_filter_disabled_flag` không hiện diện, nó được suy ra là bằng `ph_deblocking_filter_disabled_flag`.

`slice_beta_offset_div2` và `slice_tc_offset_div2` chỉ ra các dịch vị thông số giải khối đối với β và tC (được chia cho 2) mà được áp dụng cho thành phần độ sáng đối với lát hiện tại. Cả hai giá trị của `slice_beta_offset_div2` và `slice_tc_offset_div2` đều phải nằm trong khoảng từ -12 tới 12, bao gồm các giá trị biên. Khi không hiện diện, các giá trị của `slice_beta_offset_div2` và `slice_tc_offset_div2` lần lượt được suy ra là bằng `ph_beta_offset_div2` và `ph_tc_offset_div2`.

`slice_cb_beta_offset_div2` và `slice_cb_tc_offset_div2` chỉ ra các dịch vị thông số giải khối đối với β và tC (được chia cho 2) mà được áp dụng cho thành phần C_b đối với lát hiện tại. Cả hai giá trị của `slice_cb_beta_offset_div2` và `slice_cb_tc_offset_div2` đều phải nằm trong khoảng từ -12 tới 12, bao gồm các giá trị biên. Khi không hiện diện, các giá trị của `slice_cb_beta_offset_div2` và `slice_cb_tc_offset_div2` lần lượt được suy ra là bằng `ph_cb_beta_offset_div2` và `ph_cb_tc_offset_div2`.

`slice_cr_beta_offset_div2` và `slice_cr_tc_offset_div2` chỉ ra các dịch vị thông số giải khối đối với β và tC (được chia cho 2) mà được áp dụng cho thành phần C_r đối với lát hiện tại. Cả hai giá trị của `slice_cr_beta_offset_div2` và `slice_cr_tc_offset_div2` đều phải nằm trong khoảng từ -12 tới 12, bao gồm các giá trị biên. Khi không hiện diện, các giá trị của `slice_cr_beta_offset_div2` và `slice_cr_tc_offset_div2` lần lượt được suy ra là bằng `ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2`.

slice_ts_residual_coding_disabled_flag bằng 1 chỉ ra rằng cấu trúc cú pháp **residual_coding()** được sử dụng để phân tích các mẫu dư của khối bỏ qua chuyển đổi đối với lát hiện tại. **slice_ts_residual_coding_disabled_flag** bằng 0 chỉ ra rằng cấu trúc cú pháp **residual_ts_coding()** được sử dụng để phân tích các mẫu dư của khối bỏ qua chuyển đổi đối với lát hiện tại. Khi **slice_ts_residual_coding_disabled_flag** không hiện diện, nó được suy ra là bằng 0.

slice_lmcs_enabled_flag bằng 1 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ là được phép đối với lát hiện tại. **slice_lmcs_enabled_flag** bằng 0 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ không được phép đối với lát hiện tại. Khi **slice_lmcs_enabled_flag** không hiện diện, nó được suy ra là bằng 0.

slice_scaling_list_present_flag bằng 1 chỉ ra rằng dữ liệu danh sách chọn thang được sử dụng đối với lát hiện tại được rút ra dựa trên dữ liệu danh sách chọn thang được chứa trong danh sách chọn thang APS tham chiếu với **aps_params_type** bằng **SCALING_APS** và **adaptation_parameter_set_id** bằng **ph_scaling_list_aps_id**. **slice_scaling_list_present_flag** bằng 0 chỉ ra rằng dữ liệu danh sách chọn thang được sử dụng đối với ảnh hiện tại là dữ liệu danh sách chọn thang mặc định được rút ra được chỉ ra trong mục 7.4.3.21. Khi không hiện diện, giá trị của **slice_scaling_list_present_flag** được suy ra là bằng 0.

Biến số **NumEntryPoints**, mà chỉ ra số lượng các điểm mục nhập trong lát hiện tại, được rút ra như sau:

```

NumEntryPoints = 0
for(i = 1; i < NumCtusInCurrSlice; i++) {
    ctbAddrX = CtbAddrInCurrSlice[i] % PicWidthInCtbsY
    ctbAddrY = CtbAddrInCurrSlice[i]/PicWidthInCtbsY    (145)
    prevCtbAddrX = CtbAddrInCurrSlice[i - 1] %
PicWidthInCtbsY
    prevCtbAddrY = CtbAddrInCurrSlice[i - 1]/PicWidthInCtbsY
    if( CtbToTileRowBd[ctbAddrY] !=
CtbToTileRowBd[prevCtbAddrY] ||
        CtbToTileColBd[ctbAddrX] !=

```

```

CtbToTileColBd[prevCtbAddrX] ||
    (ctbAddrY != prevCtbAddrY &&
sps_wpp_entry_point_offsets_present_flag))
    NumEntryPoints++
}

```

offset_len_minus1 plus 1 chỉ ra chiều dài, tính bằng bit, của các phần tử cú pháp **entry_point_offset_minus1[i]**. Giá trị của **offset_len_minus1** phải nằm trong khoảng từ 0 tới 31, bao gồm các giá trị biên.

entry_point_offset_minus1[i] plus 1 chỉ ra dịch vị điểm mục nhập thứ *i* tính bằng byte, và được biểu diễn bằng **offset_len_minus1** plus 1 bit. Dữ liệu lát mà theo sau tiêu đề lát chứa **NumEntryPoints** + 1 tập con, với các giá trị chỉ số tập con nằm trong khoảng từ 0 tới **NumEntryPoints**, bao gồm các giá trị biên. Byte thứ nhất của dữ liệu lát được coi là byte 0. Khi hiện diện, các byte phòng ngừa mô phỏng mà xuất hiện trong phần dữ liệu lát của đơn vị NAL lát được tạo mã được tính như phần của dữ liệu lát đối với mục đích nhận dạng tập con. Tập con 0 chứa các byte từ 0 tới **entry_point_offset_minus1[0]**, bao gồm các giá trị biên, của dữ liệu lát được tạo mã, tập con *k*, với *k* nằm trong khoảng từ 1 tới **NumEntryPoints** – 1, bao gồm các giá trị biên, chứa các byte từ **firstByte[k]** tới **lastByte[k]**, bao gồm các giá trị biên, của dữ liệu lát được tạo mã với **firstByte[k]** và **lastByte[k]** được xác định như sau:

$$\text{firstByte}[k] = \sum_{n=1}^k (\text{entry_point_offset_minus1}[n - 1] + 1) \quad (146)$$

$$\text{lastByte}[k] = \text{firstByte}[k] + \text{entry_point_offset_minus1}[k] \quad (147)$$

Tập con cuối cùng (với chỉ số tập con bằng **NumEntryPoints**) chứa các byte còn lại của dữ liệu lát được tạo mã.

Khi **sps_entropy_coding_sync_enabled_flag** là bằng 0 và lát chứa một hoặc nhiều ô ảnh hoàn chỉnh, mỗi tập con phải chứa tất cả các bit được tạo mã của tất cả các CTU trong lát mà ở trong cùng một ô ảnh, và số lượng các tập con (tức là, giá trị của **NumEntryPoints** + 1) phải bằng số lượng của các ô ảnh trong lát.

Khi **sps_entropy_coding_sync_enabled_flag** là bằng 0 và lát chứa tập con của các hàng CTU từ một ô ảnh, **NumEntryPoints** phải bằng 0, và số lượng các

tập con phải bằng 1. Tập con phải chứa tất cả các bit được tạo mã của tất cả các CTU trong lát.

Khi `sps_entropy_coding_sync_enabled_flag` là bằng 1, mỗi tập con k với k nằm trong khoảng từ 0 tới `NumEntryPoints`, bao gồm các giá trị biên, sẽ chứa tất cả các bit được tạo mã của tất cả các CTU trong hàng CTU bên trong ô ảnh, và số lượng các tập con (tức là, giá trị của `NumEntryPoints + 1`) phải bằng tổng số lượng của các hàng CTU của ô ảnh cụ thể trong lát.

`slice_header_extension_length` chỉ ra chiều dài của dữ liệu mở rộng tiêu đề lát tính bằng byte, không bao gồm các bit được sử dụng để báo tín hiệu bản thân `slice_header_extension_length`. Giá trị của `slice_header_extension_length` phải nằm trong khoảng từ 0 tới 256, bao gồm các giá trị biên. Khi không hiện diện, giá trị của `slice_header_extension_length` được suy ra là bằng 0.

`slice_header_extension_data_byte[i]` có thể có giá trị bất kỳ. Các bộ giải mã tương thích với phiên bản này của Đặc tả này phải bỏ qua các giá trị của tất cả các phần tử cú pháp `slice_header_extension_data_byte[i]`. Giá trị của nó không ảnh hưởng tới sự tương thích của bộ giải mã với các tiêu sử được chỉ ra trong phiên bản này của đặc tả.

4. Các ví dụ về các vấn đề kỹ thuật được giải quyết

Các thiết kế hiện có đối với cú pháp PPS, PH, và SH có các vấn đề sau đây:

1) Theo văn bản dự thảo VVC mới nhất, đối với ảnh bất kỳ trong CLVS, nếu cấu trúc cú pháp PH được bao gồm trong SH thay vì trong đơn vị NAL PH, thì mỗi ảnh trong CLVS chỉ có một lát và cấu trúc cú pháp PH đối với mỗi ảnh trong CLVS được bao gồm trong SH thay vì trong đơn vị NAL PH. Trong trường hợp này, sự báo tín hiệu của sáu cờ PPS `rpl_info_in_ph_flag`, `dbf_info_in_ph_flag`, `sao_info_in_ph_flag`, `alf_info_in_ph_flag`, `wp_info_in_ph_flag`, và `qp_delta_info_in_ph_flag` trong các PPS được tham chiếu bởi các ảnh trong CLVS là không hữu ích.

2) Theo văn bản dự thảo VVC mới nhất, phần tử cú pháp `num_ref_idx_active_override_flag` trong cú pháp SH được báo tín hiệu khi điều kiện “A & B || C” được đáp ứng, mà trong đó “A” chỉ báo `rpl_info_in_ph_flag`

là bằng 1, hoặc kiểu đơn vị NAL không phải là IDR, hoặc `sps_idr_rpl_present_flag` là bằng 1”, “B” chỉ báo “`slice_type` là bằng P hoặc B, và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 0 là lớn hơn 1”, và “C” chỉ báo “`slice_type` là bằng B và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 1 là lớn hơn 1”. Trong trường hợp này, khi `rpl_info_in_ph_flag` là bằng 0, kiểu đơn vị NAL là IDR, và `sps_idr_rpl_present_flag` là bằng 0, nhưng trong khi đó, `slice_type` là bằng B và số lượng các mục nhập tham chiếu trong danh sách 1 là lớn hơn 1, các thông số về `num_ref_idx_active_minus1[0]` và `num_ref_idx_active_minus1[1]` vẫn có thể được ghi đè và được thiết lập bằng những số đã biết; điều này có thể dẫn đến động thái bất ngờ của bộ giải mã, bao gồm làm hỏng bộ giải mã.

3) Cấu trúc cú pháp `ref_pic_list_struct()` có thể được bao gồm trong SPS và/hoặc PH hoặc SH. Khi `ref_pic_list_struct()` được bao gồm trong PH hoặc SH và `long_term_ref_pics_flag` là bằng 1, phần tử cú pháp `ltrp_in_header_flag[][]` sẽ được báo tín hiệu trong `ref_pic_list_struct()` để chỉ báo việc liệu các giá trị đếm thứ tự ảnh đối với các mục nhập LTRP có được báo tín hiệu trong `ref_pic_list_struct()` trong PH/SH, bởi các phần tử cú pháp `rpls_poc_lsb_lt[][][]`, hoặc được báo tín hiệu một cách trực tiếp trong `ref_pic_lists()` trong PH hoặc SH, bởi các phần tử cú pháp `poc_lsb_lt[][]`. Tuy nhiên, trong trường hợp này, sự báo tín hiệu của `ltrp_in_header_flag` trong `ref_pic_list_struct()` là vô nghĩa, do dù sao thì thông tin POC LTRP cũng được báo tín hiệu trong tiêu đề ảnh thay vì SPS.

4) Cấu trúc cú pháp `ref_pic_lists()` sẽ được bao gồm trong hoặc PH hoặc SH. Khi `ref_pic_lists()` được bao gồm trong SH, thông tin ảnh tham chiếu của cả các danh sách ảnh tham chiếu 0 lẫn danh sách ảnh tham chiếu 1 là luôn luôn được báo tín hiệu, ngay cả nếu `slice_type` là bằng P. Trong trường hợp này (khi `ref_pic_lists()` được bao gồm trong SH và `slice_type = P`), sự báo tín hiệu của thông tin tham chiếu danh sách ảnh tham chiếu 1 (ví dụ, `rpl_sps_flag[1]`, `rpl_idx[1]`, `poc_lsb_lt[1][j]`, `delta_poc_msb_present_flag[1][j]`, `delta_poc_msb_cycle_lt[1][j]` mà trong đó `j` nằm trong khoảng từ 0 tới

NumLtrpEntries[1][RplsIdx[1] - 1, bao gồm các giá trị biên) trong ref_pic_lists() đối với lát P trở nên vô ích.

5. Các kỹ thuật và các phương án ví dụ

Để giải quyết các vấn đề nêu trên và một số vấn đề khác không được nhắc đến, các phương pháp như được tổng kết trong phần liệt kê dưới đây được đề xuất. Các mục này cần được xem như các ví dụ để giải thích các ý tưởng chung và không được diễn giải theo cách hẹp. Hơn nữa, các mục này có thể được áp dụng một cách riêng rẽ hoặc được kết hợp theo cách bất kỳ.

1. Để giải quyết vấn đề thứ nhất, việc liệu báo tín hiệu sự chỉ báo của thông tin đã biết có cần phải hiện diện trong PH hoặc SH (ví dụ, tức là, rpl_info_in_ph_flag, dbf_info_in_ph_flag, sao_info_in_ph_flag, alf_info_in_ph_flag, wp_info_in_ph_flag, và qp_delta_info_in_ph_flag trong các PPS) và/hoặc sự hiện diện của cấu trúc cú pháp tiêu đề ảnh trong tiêu đề lát (ví dụ, picture_header_in_slice_header_flag) có thể phụ thuộc vào sự chỉ báo về việc liệu mỗi ảnh trong CLVS chỉ có một lát.

1) Khi có sự chỉ báo về mỗi ảnh trong CLVS có chỉ một lát, thì các sự chỉ báo (ví dụ, một hoặc nhiều trong số 6 cờ PPS) về thông tin đã biết đang hiện diện trong PH hoặc SH không cần phải được báo tín hiệu đối với CLVS trong đó mỗi ảnh chỉ có một lát, vì trong trường hợp này PH là trong SH.

a. Trong một ví dụ, sự chỉ báo về việc liệu mỗi ảnh tham chiếu tới PPS chỉ có một lát có thể được báo tín hiệu trong PPS.

i. Theo cách khác, sự chỉ báo về việc liệu mỗi ảnh tham chiếu tới PPS chỉ có một lát trong có thể được báo tín hiệu trong PPS.

ii. Theo cách khác, bộ chỉ báo về việc liệu mỗi ảnh tham chiếu tới PPS chỉ có một lát ngoài có thể được báo tín hiệu trong PPS.

b. Trong một ví dụ, sự chỉ báo có thể là cờ, ví dụ, được gọi là pps_one_slice_per_picture_flag, trong PPS. Theo cách khác, hơn nữa, cờ này bằng 1 chỉ ra rằng mỗi ảnh tham chiếu tới PPS chỉ có

một lát, và cờ này bằng 0 chỉ ra rằng mỗi ảnh tham chiếu tới PPS có một hoặc nhiều lát.

i. Theo cách khác, sự chỉ báo có thể không được báo tín hiệu nhưng được suy ra từ thông tin phân vùng lát được báo tín hiệu, như phần tử cú pháp PPS `num_slices_in_pic_minus1` là bằng 0 (hoặc được báo tín hiệu hoặc được suy ra, trong trường hợp này hoặc trường hợp kia giá trị của `rect_slice_flag` sẽ bằng 1, do khi `rect_slice_flag` là bằng 0, `num_slices_in_pic_minus1` không được báo tín hiệu và giá trị này cũng không được suy ra.) Do đó, khi `rect_slice_flag` là bằng 1, `pps_one_slice_per_picture_flag` bằng 1 là tương đương với `num_slices_in_pic_minus1` bằng 0. Sự thiếu sót của cách tiếp cận này là nó không bao gồm trường hợp khi `rect_slice_flag` là bằng 0, trong khi cách tiếp cận sử dụng `pps_one_slice_per_picture_flag` là áp dụng cả hai trường hợp của `rect_slice_flag` bằng 0 và `rect_slice_flag` bằng 1.

ii. Trong một ví dụ, ngoài ra, giá trị của `pps_one_slice_per_picture_flag` là cần phải bằng 0 khi `sps_num_subpics_minus1` là lớn hơn 0.

iii. Ngoài ra, cờ này có thể phải là giống nhau đối với tất cả các PPS được tham chiếu bởi các ảnh trong CLVS.

iv. Ngoài ra, cờ này có thể phải là bằng 1 khi giá trị của `one_slice_per_pic_constraint_flag` là bằng 1.

v. Ngoài ra, các ngữ nghĩa của `one_slice_per_pic_constraint_flag` được cập nhật bằng cách bổ sung rằng `one_slice_per_pic_constraint_flag` bằng 1 còn chỉ ra rằng `pps_one_slice_per_picture_flag` sẽ bằng 1.

vi. Sự báo tín hiệu của cờ này trong PPS có thể được bỏ qua khi được biết từ PPS rằng chỉ có một ảnh con trong mỗi ảnh (ví dụ, `sps_num_subpics_minus1` bằng 0) và mỗi ảnh

con chỉ chứa một lát (trong trường hợp này sẽ biết rằng mỗi ảnh chỉ chứa một lát).

c. Khi sự chỉ báo chỉ báo rằng mỗi ảnh chỉ chứa một lát (ví dụ, `pps_one_slice_per_picture_flag` là bằng 1), sự báo tín hiệu của các sự chỉ báo (ví dụ, 6 cờ PPS) về thông tin đã biết đang hiện diện trong hoặc PH hoặc SH có thể được bỏ qua.

i. Ngoài ra, khi `pps_one_slice_per_picture_flag` là bằng 1, các giá trị của tất cả 6 cờ được suy ra là bằng 0.

Lưu ý rằng việc suy ra chúng là 0 là tốt hơn (so với suy ra chúng là bằng 1) đối với ít nhất một số thông số, ví dụ, các thông số liên quan tới RPL như ảnh được sắp xếp và các thông số dự báo có trọng số, vì các số lượng của các mục nhập hoạt động sẽ được biết để báo tín hiệu chúng, và việc biết điều này có thể làm cho việc báo tín hiệu tối ưu hơn.

1. Theo cách khác, ngoài ra, khi `pps_one_slice_per_picture_flag` là bằng 1, các giá trị của tất cả 6 cờ được suy ra là bằng 1.

2. Theo cách khác, khi `pps_one_slice_per_picture_flag` là bằng 1, ngoài ra, giá trị của mỗi trong số 6 cờ không bao gồm `wp_info_in_ph_flag` được suy ra là bằng hoặc 1 hoặc 0, không phụ thuộc vào cách thức các cờ khác được suy ra, và đối với `wp_info_in_ph_flag`, giá trị được suy ra là bằng 0 khi `rpl_info_in_ph_flag` là bằng 0, và được suy ra bằng hoặc 1 hoặc 0 khi `rpl_info_in_ph_flag` là bằng 1, không phụ thuộc vào các giá trị của 4 cờ khác trong số 6 cờ này.

d. Khi sự chỉ báo chỉ báo rằng mỗi ảnh chỉ chứa một lát (ví dụ, `pps_one_slice_per_picture_flag` là bằng 1), sự báo tín hiệu của `picture_header_in_slice_header_flag` trong cú pháp SH có thể

được bỏ qua, ví dụ, bằng cách đặt điều kiện phần tử cú pháp là "if(!pps_one_slice_per_picture_flag)" và suy ra giá trị của picture_header_in_slice_header_flag khi pps_one_slice_per_picture_flag là bằng 1.

e. Khi pps_one_slice_per_picture_flag là bằng 1, sự báo tín hiệu của slice_address trong cú pháp SH có thể được bỏ qua như sau (các phần được thêm vào được in nghiêng và gạch dưới):

if((rect_slice_flag && NumSlicesInSubpic[CurrSubpicIdx] > 1) (!rect_slice_flag && NumTilesInPic > 1 && <i>!pps_one_slice_per_picture_flag</i>)	
<i>slice_address</i>	u(v)

Và sự suy ra sau đây được thêm vào: Khi rect_slice_flag là bằng 0 và pps_one_slice_per_picture_flag là bằng 1, giá trị của slice_address được suy ra là bằng 0.

f. Khi pps_one_slice_per_picture_flag là bằng 1, sự báo tín hiệu của num_tiles_in_slice_minus1 trong cú pháp SH có thể được bỏ qua như sau:

if(!rect_slice_flag && NumTilesInPic > 1 && <i>!pps_one_slice_per_picture_flag</i>)	
<i>num_tiles_in_slice_minus1</i>	ue(v)

Và sự suy ra sau đây được thêm vào: Khi rect_slice_flag là bằng 0 và pps_one_slice_per_picture_flag là bằng 1, giá trị của num_tiles_in_slice_minus1 được suy ra là bằng NumTilesInPic - 1.

g. Thay cho tất cả các cách trên, cờ pps_one_slice_per_picture_flag không được thêm vào, nhưng các giá trị của sáu cờ PPS được yêu cầu là bằng 0 khi lát bất kỳ tham chiếu tới PPS có picture_header_in_slice_header_flag bằng 1.

i. Điều này về mặt kỹ thuật tương đương với điều kiện ràng buộc sau đây được thêm vào các ngữ nghĩa của picture_header_in_slice_header_flag (mà là tốt hơn về mặt

biên tập): Khi cờ bất kỳ trong số sáu cờ PPS là bằng 1, giá trị của `picture_header_in_slice_header_flag` phải bằng 0.

h. Thay cho tất cả các cách trên, cờ `pps_one_slice_per_picture_flag` không được thêm vào, nhưng các giá trị của sáu cờ PPS được yêu cầu là bằng 0 khi `one_slice_per_pic_constraint_flag` là bằng 1.

i. Không phụ thuộc vào bất kỳ cách nào trong số các cách nêu trên, điều kiện ràng buộc sau đây có thể được thêm vào: Khi `one_slice_per_pic_constraint_flag` là bằng 1, giá trị của `picture_header_in_slice_header_flag` phải bằng 1. Theo cách tương đương về mặt kỹ thuật, các ngữ nghĩa của `one_slice_per_pic_constraint_flag` được cập nhật bằng cách bổ sung rằng `one_slice_per_pic_constraint_flag` bằng 1 còn chỉ ra rằng `picture_header_in_slice_header_flag` phải bằng 1 (đối với tất cả các lát trong mỗi OLS mà cấu trúc cú pháp PTL chứa cờ này áp dụng tới đó).

j. Khi `pps_one_slice_per_picture_flag` là bằng 1, sự báo tín hiệu của `rect_slice_flag` trong cú pháp PPS có thể được bỏ qua như sau (các phần được thêm vào được in nghiêng và gạch dưới):

<code>if(NumTilesInPic > 1 && !<i>pps_one_slice_per_picture_flag</i>)</code>	
<code>rect_slice_flag</code>	<code>u(1)</code>

i. Và sự suy ra sau đây được thêm vào: Khi *pps_one_slice_per_picture_flag* là bằng 1, giá trị của *rect_slice_flag* được suy ra là bằng 1.

k. Khi `pps_one_slice_per_picture_flag` là bằng 1, sự báo tín hiệu của `single_slice_per_subpic_flag` trong cú pháp PPS có thể được bỏ qua như sau (các phần được thêm vào được in nghiêng và gạch dưới):

<code>if(rect_slice_flag && !<i>pps_one_slice_per_picture_flag</i>)</code>	
<code>single_slice_per_subpic_flag</code>	<code>u(1)</code>

i. Và sự suy ra sau đây được thêm vào: Khi *pps_one_slice_per_picture_flag* là bằng 1, giá trị của *single_slice_per_subpic_flag* được suy ra là bằng 1.

l. Khi `pps_one_slice_per_picture_flag` là bằng 1, sự báo tín hiệu của `loop_filter_across_slices_enabled_flag` trong cú pháp PPS có thể được bỏ qua, ví dụ, bằng cách đặt điều kiện phân tử cú pháp là "`if( !pps_one_slice_per_picture_flag )`" và suy ra giá trị của `loop_filter_across_slices_enabled_flag` khi `pps_one_slice_per_picture_flag` là bằng 0.

m. Trong một ví dụ, sự chỉ báo về việc liệu mỗi ảnh trong CLVS chỉ có một lát có thể được báo tín hiệu trong SPS liên kết với CLVS.

n. Trong một ví dụ, sự chỉ báo về việc liệu mỗi ảnh trong CLVS chỉ có một ô ảnh có thể được báo tín hiệu trong SPS liên kết với CLVS.

o. Khi được chỉ báo rằng mỗi ảnh liên kết với PPS thứ nhất chỉ có một lát, một hoặc nhiều phân tử cú pháp liên quan tới phân vùng lát (như `rect_slice_flag`, `single_slice_per_subpic_flag`, `num_slices_in_pic_minus1`, `tile_idx_delta_present_flag`, `slice_width_in_tiles_minus1`, `slice_height_in_tiles_minus1`, `num_exp_slices_in_tile`, `exp_slice_height_in_ctus_minus1`, `tile_idx_delta`, v.v. trong JVET-Q2001-vC) có thể không được báo tín hiệu trong PPS thứ nhất.

i. Trong một ví dụ, sự chỉ báo là `pps_one_slice_per_picture_flag` mà được bao gồm trong cú pháp PPS trước cú pháp báo tín hiệu lát, ví dụ, ngay trước `rect_slice_flag`.

1. Trong một ví dụ, ngoài ra, khi `pps_one_slice_per_picture_flag` là bằng 1 và `rect_slice_flag` không được báo tín hiệu, giá trị của nó được suy ra là bằng 1.

2. Trong một ví dụ, ngoài ra, khi `pps_one_slice_per_picture_flag` là bằng 1 và

`single_slice_per_subpic_flag` không được báo tín hiệu, giá trị của nó được suy ra là bằng 1.

3. Trong một ví dụ, ngoài ra, khi `pps_one_slice_per_picture_flag` là bằng 1 và `num_slices_in_pic_minus1` không được báo tín hiệu, giá trị của nó được suy ra là bằng 0.

p. Khi được chỉ báo rằng mỗi ảnh liên kết với PPS thứ nhất chỉ có một ô ảnh, một hoặc nhiều phần tử cú pháp liên quan tới phân vùng lát (như `num_exp_tile_columns_minus1`, `num_exp_tile_rows_minus1`, `tile_column_width_minus1`, `tile_row_height_minus1`, v.v. trong JVET-Q2001-vC) có thể không được báo tín hiệu trong PPS thứ nhất.

i. Trong một ví dụ, sự chỉ báo là `pps_one_tile_per_picture_flag` mà được bao gồm trong cú pháp PPS trước cú pháp báo tín hiệu ô ảnh, ví dụ, ngay trước `pps_log2_ctu_size_minus5` (nhưng theo cùng điều kiện cú pháp "if(!no_pic_partition_flag)").

2. Liên quan đến sự báo tín hiệu của phần tử cú pháp `num_ref_idx_active_override_flag` trong SH:

1) Để giải quyết vấn đề thứ hai, phần tử cú pháp `num_ref_idx_active_override_flag` trong SH có thể được báo tín hiệu bởi điều kiện là "A & (B || C)", mà trong đó "A" chỉ báo "`rpl_info_in_ph_flag` = 1, hoặc kiểu đơn vị NAL không bằng IDR, hoặc `sps_idr_rpl_present_flag` là bằng 1", và "B" chỉ báo "`slice_type` là bằng P hoặc B, và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 0 là lớn hơn 1", và "C" chỉ báo "`slice_type` là bằng B và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 1 là lớn hơn 1".

a. Trong một ví dụ, cú pháp sau đây trong PH:

if((rpl_info_in_ph_flag ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag)) && (slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	

được thay đổi như sau:

if((rpl_info_in_ph_flag ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag)) && ((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1))) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	

3. Để giải quyết vấn đề thứ ba, cách tiếp cận sau đây liên quan đến sự báo tín hiệu của các phần tử cú pháp `ltrp_in_header_flag[listIdx][rplsIdx]` và `rpls_poc_lsb_lt[listIdx][rplsIdx][i]` trong cấu trúc cú pháp `ref_pic_list_struct()` được đề xuất:

1) Liệu việc báo tín hiệu các phần tử cú pháp liên quan tới giá trị đếm thứ tự ảnh đối với tham chiếu dài hạn (ví dụ, `ltrp_in_header_flag` và/hoặc `rpls_poc_lsb_lt`) trong cấu trúc cú pháp `ref_pic_list_struct()` có thể phụ thuộc vào việc liệu chỉ số của các danh sách ảnh tham chiếu (ví dụ, `rplsIdx`) là nhỏ hơn số lượng của cú pháp `ref_pic_list_struct(listIdx,`

rplsIdx) trong SPS (ví dụ, num_ref_pic_lists_in_sps[listIdx]), như trong phương án thứ hai,

a. Trong một ví dụ, nếu chỉ số của các danh sách ảnh tham chiếu là bằng số lượng cú pháp ref_pic_list_struct(listIdx, rplsIdx) trong SPS (tức là, rplsIdx bằng num_ref_pic_lists_in_sps[listIdx]), thì các phần tử cú pháp liên quan tới giá trị đếm thứ tự ảnh đối với tham chiếu dài hạn (ví dụ, ltrp_in_header_flag) có thể không được báo tín hiệu và giá trị của ltrp_in_header_flag được suy ra là bằng hoặc 1 hoặc 0.

2) Theo cách khác, liệu việc báo tín hiệu các phần tử cú pháp liên quan tới giá trị đếm thứ tự ảnh đối với tham chiếu dài hạn (ví dụ, ltrp_in_header_flag và/hoặc rpls_poc_lsb_lt) trong cấu trúc cú pháp ref_pic_list_struct() có thể phụ thuộc vào việc liệu cấu trúc cú pháp ref_pic_list_struct() có được bao gồm trong SPS hay không, như trong phương án thứ hai.

a. Trong một ví dụ, nếu cấu trúc cú pháp ref_pic_list_struct() KHÔNG được bao gồm trong SPS, thì các phần tử cú pháp liên quan tới giá trị đếm thứ tự ảnh đối với tham chiếu dài hạn (ví dụ, ltrp_in_header_flag) có thể không được báo tín hiệu và giá trị của ltrp_in_header_flag được suy ra là bằng hoặc 1 hoặc 0.

3) Theo cách khác, thay vì báo tín hiệu ltrp_in_header_flag bên trong ref_pic_list_struct(), nó được báo tín hiệu một cách trực tiếp trong SPS, một đối với mỗi ref_pic_list_struct() trong SPS, ví dụ, như sau:

for(i = 0; i < !rpl1 same as rpl0 flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++) {	
ltrp_in_header_flag[i][j]	u(1)
ref_pic_list_struct(i, j)	
}	
}	

ltrp_in_header_flag[i][j] bằng 0 chỉ ra rằng các LSB POC của các mục nhập LTRP được chỉ báo trong cấu trúc cú pháp ref_pic_list_struct(i, j) là hiện diện trong cùng cấu trúc cú pháp. ltrp_in_header_flag[i][j] bằng 1 chỉ ra rằng các LSB POC của các mục nhập LTRP được chỉ báo trong cấu trúc cú pháp

ref_pic_list_struct(i, j) là không hiện diện trong cùng cấu trúc cú pháp. Giá trị của ltrp_in_header_flag[i][num_ref_pic_lists_in_sps[i]] được suy ra là bằng 0. (Theo cách khác, giá trị của ltrp_in_header_flag[i][num_ref_pic_lists_in_sps[i]] được suy ra là bằng 1.)

Và cú pháp ref_pic_list_struct(listIdx, rplsIdx) được thay đổi như sau. Các đoạn văn bản bị xóa được đánh dấu bằng các ngoặc kép (ví dụ, [[a]] ký hiệu xóa ký tự “a”):

ref_pic_list_struct(listIdx, rplsIdx) {	Mô tả
num_ref_entries[listIdx][rplsIdx]	ue(v)
[[if (long term ref pics flag)	
ltrp_in_header_flag[listIdx][rplsIdx]	u(1)]
for(i = 0, j = 0; i < num_ref_entries[listIdx][rplsIdx]; i++) {	
if(inter_layer_ref_pics_present_flag)	
inter_layer_ref_pic_flag[listIdx][rplsIdx][i]	u(1)
if(!inter_layer_ref_pic_flag[listIdx][rplsIdx][i]) {	
if(long term ref pics flag)	
st_ref_pic_flag[listIdx][rplsIdx][i]	u(1)
if(st_ref_pic_flag[listIdx][rplsIdx][i]) {	
abs_delta_poc_st[listIdx][rplsIdx][i]	ue(v)
if(AbsDeltaPocSt[listIdx][rplsIdx][i] > 0)	
strp_entry_sign_flag[listIdx][rplsIdx][i]	u(1)
} else if(!ltrp_in_header_flag[listIdx][rplsIdx])	
rpls_poc_lsb_lt[listIdx][rplsIdx][j++]	u(v)
} else	
ilrp_idx[listIdx][rplsIdx][i]	ue(v)
}	
}	

4) Theo cách khác, thay vì báo tín hiệu ltrp_in_header_flag bên trong ref_pic_list_struct(), nó được báo tín hiệu một cách trực tiếp trong SPS, một đối với mỗi trong số danh sách 0 và danh sách 1, ví dụ, như sau:

for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
ltrp_in_header_flag[i]	u(1)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	

ltrp_in_header_flag[i] bằng 0 chỉ ra rằng các LSB POC của các mục nhập LTRP được chỉ báo trong mỗi trong số các cấu trúc cú pháp ref_pic_list_struct(i, j) đối với j nằm trong khoảng từ 0 tới num_ref_pic_lists_in_sps[i], bao gồm các giá trị biên, là hiện diện trong cùng cấu trúc cú pháp. ltrp_in_header_flag[i][j]

bảng 1 chỉ ra rằng các LSB POC của các mục nhập LTRP được chỉ báo trong mỗi trong số các cấu trúc cú pháp `ref_pic_list_struct(i, j)` này) là không hiện diện trong cùng cấu trúc cú pháp.

Và `ref_pic_lists()` syntax được thay đổi như sau:

<code>if(ltrp in header flag[i][[RplsIdx[i]]])</code>	
<code>poc_lsb_lt[i][j]</code>	<code>u(v)</code>

Và cú pháp `ref_pic_list_struct(listIdx, rplsIdx)` được thay đổi như sau:

<code>ref_pic_list_struct(listIdx, rplsIdx) {</code>	Mô tả
<code> num_ref_entries[listIdx][rplsIdx]</code>	<code>ue(v)</code>
<code> [[if(long term ref pics flag)</code>	
<code> ltrp in header flag[listIdx][rplsIdx]</code>	<code>u(1)]</code>
<code> for(i = 0, j = 0; i < num_ref_entries[listIdx][rplsIdx]; i++) {</code>	
<code> if(inter layer ref pics present flag)</code>	
<code> inter_layer_ref_pic_flag[listIdx][rplsIdx][i]</code>	<code>u(1)</code>
<code> if(!inter_layer_ref_pic_flag[listIdx][rplsIdx][i]) {</code>	
<code> if(long term ref pics flag)</code>	
<code> st_ref_pic_flag[listIdx][rplsIdx][i]</code>	<code>u(1)</code>
<code> if(st_ref_pic_flag[listIdx][rplsIdx][i]) {</code>	
<code> abs_delta_poc_st[listIdx][rplsIdx][i]</code>	<code>ue(v)</code>
<code> if(AbsDeltaPocSt[listIdx][rplsIdx][i] > 0)</code>	
<code> strp_entry_sign_flag[listIdx][rplsIdx][i]</code>	<code>u(1)</code>
<code> } else if(!ltrp in header flag[listIdx][[rplsIdx]])</code>	
<code> rpls_poc_lsb_lt[listIdx][rplsIdx][j++]</code>	<code>u(v)</code>
<code> } else</code>	
<code> ilrp_idx[listIdx][rplsIdx][i]</code>	<code>ue(v)</code>
<code> }</code>	
<code>}</code>	

5) Theo cách khác, thay vì báo tín hiệu `ltrp_in_header_flag` bên trong `ref_pic_list_struct()`, nó được báo tín hiệu một cách trực tiếp trong SPS, chỉ một, ví dụ, như sau:

<u><code>ltrp in header flag</code></u>	<u><code>u(1)</code></u>
<code>for(i = 0; i < !rpl1 same as rpl0 flag ? 2 : 1; i++) {</code>	
<code> num_ref_pic_lists_in_sps[i]</code>	<code>ue(v)</code>
<code> for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)</code>	
<code> ref_pic_list_struct(i, j)</code>	
<code>}</code>	

`ltrp in header flag` bằng 0 chỉ ra rằng các LSB POC của các mục nhập LTRP được chỉ báo trong mỗi trong số các cấu trúc cú pháp `ref_pic_list_struct(i, j)` đối với i nằm trong khoảng từ 0 tới $(!rpl1_same_as_rpl0_flag ? 2 : 1)$, bao gồm các giá trị biên, và j nằm trong khoảng từ 0 tới `num_ref_pic_lists_in_sps[i]`, bao gồm các giá trị biên, là hiện diện trong cùng cấu trúc cú pháp.

ltrp_in_header_flag[i][j] bằng 1 chỉ ra rằng các LSB POC của các mục nhập LTRP được chỉ báo trong mỗi trong số các cấu trúc cú pháp ref_pic_list_struct(i, j) này là không hiện diện trong cùng cấu trúc cú pháp.

Và cú pháp ref_pic_lists() được thay đổi như sau:

if(ltrp_in_header_flag [[i][RplsIdx[i]]])	
poc_lsb_lt[i][j]	u(v)

Và cú pháp ref_pic_list_struct(listIdx, rplsIdx) được thay đổi như sau:

ref_pic_list_struct(listIdx, rplsIdx) {	Mô tả
num_ref_entries [listIdx][rplsIdx]	ue(v)
[[if(long term ref pics flag)	
ltrp_in_header_flag [listIdx][rplsIdx]	u(1)]]
for(i = 0, j = 0; i < num_ref_entries[listIdx][rplsIdx]; i++) {	
if(inter layer ref pics present flag)	
inter_layer_ref_pic_flag [listIdx][rplsIdx][i]	u(1)
if(!inter_layer_ref_pic_flag[listIdx][rplsIdx][i]) {	
if(long term ref pics flag)	
st_ref_pic_flag [listIdx][rplsIdx][i]	u(1)
if(st_ref_pic_flag[listIdx][rplsIdx][i]) {	
abs_delta_poc_st [listIdx][rplsIdx][i]	ue(v)
if(AbsDeltaPocSt[listIdx][rplsIdx][i] > 0)	
strp_entry_sign_flag [listIdx][rplsIdx][i]	u(1)
} else if(!ltrp_in_header_flag [[listIdx][rplsIdx]])	
rpls_poc_lsb_lt [listIdx][rplsIdx][j++]	u(v)
} else	
ilrp_idx [listIdx][rplsIdx][i]	ue(v)
}	
}	

6) Theo cách khác, thay vì báo tín hiệu ltrp_in_header_flag và rpls_poc_lsb_lt bên trong ref_pic_list_struct(), chúng được báo tín hiệu một cách trực tiếp trong SPS, một tập đối với mỗi ref_pic_list_struct() trong SPS, ví dụ, như trong phương án thứ hai.

7) Theo cách khác, thay vì báo tín hiệu ltrp_in_header_flag và rpls_poc_lsb_lt bên trong ref_pic_list_struct(), chúng được báo tín hiệu một cách trực tiếp trong SPS, một tập đối với mỗi danh sách trong số danh sách 0 và danh sách 1, ví dụ, như trong phương án thứ hai.

8) Theo cách khác, thay vì báo tín hiệu ltrp_in_header_flag và rpls_poc_lsb_lt bên trong ref_pic_list_struct(), chúng được báo tín hiệu một cách trực tiếp trong SPS, chỉ một tập, ví dụ, như trong phương án thứ hai.

4. Liên quan đến sự báo tín hiệu của các phần tử cú pháp liên quan tới các ảnh tham chiếu trong danh sách ảnh tham chiếu 1 trong cấu trúc cú pháp `ref_pic_list()`:

1) Để giải quyết vấn đề thứ tư, liệu việc báo tín hiệu các phần tử cú pháp liên quan tới thông tin tham chiếu của danh sách ảnh tham chiếu 1 (ví dụ, `rpl_sps_flag[1]`, `rpl_idx[1]`, `poc_lsb_lt[1][j]`, `delta_poc_msb_present_flag[1][j]`, `delta_poc_msb_cycle_lt[1][j]` mà trong đó `j` nằm trong khoảng từ 0 tới `NumLtrpEntries[1][RplsIdx[1]] - 1`, bao gồm các giá trị biên) trong cấu trúc cú pháp `ref_pic_list()` có thể phụ thuộc vào biến số `X` (ví dụ, `X = 1` hoặc `2` hoặc phụ thuộc vào kiểu lát), như trong phương án thứ ba.

6. Các phương án

Dưới đây là một số phương án ví dụ đối với một số khía cạnh của sáng chế được tổng kết trên đây trong Phần 5, mà có thể được áp dụng cho đặc tả VVC. Các đoạn văn bản được thay đổi là dựa trên văn bản VVC cuối cùng trong JVET-Q2001-vC. Những phần thích hợp nhất mà đã được bổ sung hoặc được cải biến được làm nổi bật bằng cách in nghiêng và gạch dưới, và một số phần bị xóa được đánh dấu bằng các ngoặc kép (ví dụ, `[[a]]` ký hiệu xóa ký tự “a”).

Phương án thứ nhất

Đây là phương án của các mục 1, 1.1, 1.1.a, 1.1.b, 1.1.b.iii, 1.1.b.iv, 1.1.c, 1.1.c.i, 1.1.e, 1.1.f, và 1.1.i được tổng kết trên đây trong Phần 5.

7.3.2.4 Cú pháp RBSP tập thông số ảnh

pic_parameter set rbsp() {	Mô tả
...	
<u><i>pps one slice per picture flag</i></u>	<u><i>u(1)</i></u>
<u><i>if (!pps one slice per picture flag) {</i></u>	
<u><i>rpl info in ph flag</i></u>	<u><i>u(1)</i></u>
<u><i>if(deblocking filter override enabled flag)</i></u>	
<u><i>dbf info in ph flag</i></u>	<u><i>u(1)</i></u>
<u><i>sao info in ph flag</i></u>	<u><i>u(1)</i></u>
<u><i>alf info in ph flag</i></u>	<u><i>u(1)</i></u>
<u><i>if((pps_weighted_pred_flag pps_weighted_bipred_flag) && rpl info in ph flag)</i></u>	
<u><i>wp info in ph flag</i></u>	<u><i>u(1)</i></u>
<u><i>qp delta info in ph flag</i></u>	<u><i>u(1)</i></u>

}	
...	
}	

7.4.3.4 Các ngữ nghĩa RBSP tập thông số ảnh

...

pps_one_slice_per_picture_flag bằng 1 chỉ ra rằng mỗi ảnh tham chiếu tới PPS chỉ có một lát. pps_one_slice_per_picture_flag bằng 0 chỉ ra rằng mỗi ảnh tham chiếu tới PPS có một hoặc nhiều lát.

Giá trị của pps_one_slice_per_picture_flag phải giống nhau trong tất cả các PPS mà được tham chiếu tới bởi các ảnh được tạo mã trong CLVS.

Khi one_slice_per_pic_constraint_flag là bằng 1, giá trị của pps_one_slice_per_picture_flag phải bằng 1.

rpl_info_in_ph_flag bằng 1 chỉ ra rằng thông tin danh sách ảnh tham chiếu là hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. rpl_info_in_ph_flag bằng 0 chỉ ra rằng thông tin danh sách ảnh tham chiếu không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. Khi không hiện diện, giá trị của rpl_info_in_ph_flag được suy ra là bằng 0.

dbf_info_in_ph_flag bằng 1 chỉ ra rằng thông tin lọc giải khối là hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. dbf_info_in_ph_flag bằng 0 chỉ ra rằng thông tin lọc giải khối không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. Khi không hiện diện, giá trị của dbf_info_in_ph_flag được suy ra là bằng 0.

sao_info_in_ph_flag bằng 1 chỉ ra rằng thông tin lọc SAO là hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. sao_info_in_ph_flag bằng 0 chỉ ra rằng thông tin lọc SAO không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện

trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. Khi không hiện diện, giá trị của sao_info_in_ph_flag được suy ra là bằng 0.

alf_info_in_ph_flag bằng 1 chỉ ra rằng thông tin ALF là hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. **alf_info_in_ph_flag** bằng 0 chỉ ra rằng thông tin ALF không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. Khi không hiện diện, giá trị của alf_info_in_ph_flag được suy ra là bằng 0.

wp_info_in_ph_flag bằng 1 chỉ ra rằng thông tin dự báo có trọng số có thể hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. **wp_info_in_ph_flag** bằng 0 chỉ ra rằng thông tin dự báo có trọng số không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. Khi không hiện diện, giá trị của wp_info_in_ph_flag được suy ra là bằng 0.

qp_delta_info_in_ph_flag bằng 1 chỉ ra rằng thông tin delta QP là hiện diện trong cấu trúc cú pháp PH và không hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. **qp_delta_info_in_ph_flag** bằng 0 chỉ ra rằng thông tin delta QP không hiện diện trong cấu trúc cú pháp PH và có thể hiện diện trong các tiêu đề lát tham chiếu tới PPS mà không chứa cấu trúc cú pháp PH. Khi không hiện diện, giá trị của qp_delta_info_in_ph_flag được suy ra là bằng 0.

...

7.3.7.1 Cú pháp tiêu đề lát chung

slice_header() {	Mô tả
picture_header_in_slice_header_flag	u(1)
if (picture_header_in_slice_header_flag)	
picture_header_structure()	
if (subpic_info_present_flag)	
slice_subpic_id	u(v)
if ((rect_slice_flag && NumSlicesInSubpic[CurrSubpicIdx] > 1) (!rect_slice_flag && NumTilesInPic > 1 && !pps_one_slice_per_picture_flag))	
slice_address	u(v)
for (i = 0; i < NumExtraShCac bit; i++)	
sh_extra_bit[i]	u(1)

if(!rect_slice_flag && NumTilesInPic > 1 && !pps_one_slice_per_picture_flag)	
num_tiles_in_slice_minus1	ue(v)
...	
}	

7.4.8.1 Các ngữ nghĩa tiêu đề lát chung

...

picture_header_in_slice_header_flag bằng 1 chỉ ra rằng cấu trúc cú pháp PH hiện diện trong tiêu đề lát. **picture_header_in_slice_header_flag** bằng 0 chỉ ra rằng cấu trúc cú pháp PH không hiện diện trong tiêu đề lát.

Sự tương thích dòng bit yêu cầu rằng giá trị của **picture_header_in_slice_header_flag** phải giống nhau trong tất cả các lát được tạo mã trong CLVS.

Khi **picture_header_in_slice_header_flag** là bằng 1 đối với lát được tạo mã, sự tương thích dòng bit yêu cầu rằng không có đơn vị NAL VCL với **nal_unit_type** bằng PH_NUT sẽ hiện diện trong CLVS.

Khi **picture_header_in_slice_header_flag** là bằng 0, tất cả các lát được tạo mã trong ảnh hiện tại phải có **picture_header_in_slice_header_flag** bằng 0, và PU hiện tại phải có đơn vị NAL PH.

Khi **one_slice_per_pic_constraint_flag** là bằng 1, giá trị của **picture_header_in_slice_header_flag** phải bằng 1.

slice_subpic_id chỉ ra ID ảnh con của ảnh con mà chứa lát. Nếu **slice_subpic_id** hiện diện, giá trị của biến số **CurrSubpicIdx** được rút ra sao cho **SubpicIdVal[CurrSubpicIdx]** là bằng **slice_subpic_id**. Ngược lại (**slice_subpic_id** không hiện diện), **CurrSubpicIdx** được rút ra là bằng 0. Chiều dài của **slice_subpic_id** bằng **sps_subpic_id_len_minus1 + 1** bit.

slice_address chỉ ra địa chỉ lát của lát. Khi không hiện diện, giá trị của **slice_address** được suy ra là bằng 0. [[Khi **rect_slice_flag** là bằng 1 và **NumSlicesInSubpic[CurrSubpicIdx]** là bằng 1, giá trị của **slice_address** được suy ra là bằng 0.]]

Nếu **rect_slice_flag** là bằng 0, điều sau đây được áp dụng:

- Địa chỉ lát là chỉ số ô ảnh quét màn hình.

- Chiều dài của `slice_address` bằng $\text{Ceil}(\text{Log2}(\text{NumTilesInPic}))$ bit.
- Giá trị của `slice_address` phải nằm trong khoảng từ 0 tới $\text{NumTilesInPic} - 1$, bao gồm các giá trị biên.

Ngược lại (`rect_slice_flag` là bằng 1), điều sau đây được áp dụng:

- Địa chỉ lát là chỉ số lát bậc ảnh con của lát.
- Chiều dài của `slice_address` bằng $\text{Ceil}(\text{Log2}(\text{NumSlicesInSubpic}[\text{CurrSubpicIdx}]))$ bit.
- Giá trị của `slice_address` phải nằm trong khoảng từ 0 tới $\text{NumSlicesInSubpic}[\text{CurrSubpicIdx}] - 1$, bao gồm các giá trị biên.

Sự tương thích dòng bit yêu cầu rằng các điều kiện ràng buộc sau đây được áp dụng:

- Nếu `rect_slice_flag` là bằng 0 hoặc `subpic_info_present_flag` là bằng 0, giá trị của `slice_address` phải không bằng giá trị của `slice_address` của đơn vị NAL lát được tạo mã bất kỳ khác của cùng ảnh được tạo mã.
- Ngược lại, cặp giá trị `slice_subpic_id` và `slice_address` phải không bằng cặp giá trị `slice_subpic_id` và `slice_address` của đơn vị NAL lát được tạo mã bất kỳ khác của cùng ảnh được tạo mã.
- Các hình dạng của các lát của ảnh phải sao cho mỗi CTU, khi được giải mã, phải có toàn bộ đường biên trái và toàn bộ đường biên đỉnh của nó chứa đường biên ảnh hoặc chứa các biên của (các) CTU được giải mã trước đó.

`sh_extra_bit[i]` có thể bằng 1 hoặc 0. Các bộ giải mã tương thích với phiên bản này của Đặc tả này phải bỏ qua giá trị của `sh_extra_bit[i]`. Giá trị của nó không ảnh hưởng tới sự tương thích của bộ giải mã với các tiêu sử được chỉ ra trong phiên bản này của đặc tả.

`num_tiles_in_slice_minus1` plus 1[[, khi hiện diện,]] chỉ ra số lượng của các ô ảnh trong lát khi `rect_slice_flag` là bằng 0. Giá trị của `num_tiles_in_slice_minus1` phải nằm trong khoảng từ 0 tới $\text{NumTilesInPic} - 1$, bao gồm các giá trị biên. Khi `rect_slice_flag` là bằng 0 và

num_tiles_in_slice_minus1 không hiện diện, giá trị của num_tiles_in_slice_minus1 được suy ra là bằng NumTilesInPic - 1.

...

6.1. Phương án thứ hai

Đây là phương án của các mục 3.1, 3.1a, 3.2, 3.2a, 3.6, 3.7, và 3.8 được tổng kết trên đây trong Phần 5.

6.1.1. Phương án của các mục 3.1 và 3.1a

Trong một ví dụ, ref_pic_list_struct(listIdx, rplsIdx) được thay đổi như sau:

<u>ref_pic_list_struct(listIdx, rplsIdx)</u> {	Mô tả
num_ref_entries [listIdx][rplsIdx]	ue(v)
if(long term ref pics flag && <u>rplsIdx < num_ref_pic_lists_in_sps[listIdx]</u>)	
ltrp_in_header_flag [listIdx][rplsIdx]	u(1)
for(i = 0, j = 0; i < num_ref_entries[listIdx][rplsIdx]; i++) {	
if(inter layer ref pics present flag)	
inter_layer_ref_pic_flag [listIdx][rplsIdx][i]	u(1)
if(!inter_layer_ref_pic_flag[listIdx][rplsIdx][i]) {	
if(long term ref pics flag)	
st_ref_pic_flag [listIdx][rplsIdx][i]	u(1)
if(st_ref_pic_flag[listIdx][rplsIdx][i]) {	
abs_delta_poc_st [listIdx][rplsIdx][i]	ue(v)
if(AbsDeltaPocSt[listIdx][rplsIdx][i] > 0)	
strp_entry_sign_flag [listIdx][rplsIdx][i]	u(1)
} else if(!ltrp_in_header_flag[listIdx][rplsIdx])	
rpls_poc_lsb_lt [listIdx][rplsIdx][j++]	u(v)
} else	
ilrp_idx [listIdx][rplsIdx][i]	ue(v)
}	
}	

6.1.2. Phương án của các mục 3.2 và 3.2a

Theo cách khác, ref_pic_list_struct(listIdx, rplsIdx) được thay đổi như sau:

<u>ref_pic_list_struct(listIdx, rplsIdx)</u> {	Mô tả
num_ref_entries [listIdx][rplsIdx]	ue(v)
if(long term ref pics flag && <u>nal_unit_type == SPS_NUT</u>)	
ltrp_in_header_flag [listIdx][rplsIdx]	u(1)
for(i = 0, j = 0; i < num_ref_entries[listIdx][rplsIdx]; i++) {	
if(inter layer ref pics present flag)	
inter_layer_ref_pic_flag [listIdx][rplsIdx][i]	u(1)
if(!inter_layer_ref_pic_flag[listIdx][rplsIdx][i]) {	
if(long term ref pics flag)	
st_ref_pic_flag [listIdx][rplsIdx][i]	u(1)
if(st_ref_pic_flag[listIdx][rplsIdx][i]) {	
abs_delta_poc_st [listIdx][rplsIdx][i]	ue(v)
if(AbsDeltaPocSt[listIdx][rplsIdx][i] > 0)	
strp_entry_sign_flag [listIdx][rplsIdx][i]	u(1)

<code>} else if(!ltp_in_header_flag[listIdx][rplsIdx])</code>	
<code> rpls_poc_lsb_lt[listIdx][rplsIdx][j++]</code>	$u(v)$
<code>} else</code>	
<code> ilrp_idx[listIdx][rplsIdx][i]</code>	$ue(v)$
<code>}</code>	
<code>}</code>	

`ltp_in_header_flag[listIdx][rplsIdx]` bằng 0 chỉ ra rằng các LSB POC của các mục nhập LTRP trong cấu trúc cú pháp `ref_pic_list_struct(listIdx, rplsIdx)` là hiện diện trong cấu trúc cú pháp `ref_pic_list_struct(listIdx, rplsIdx)`. `ltp_in_header_flag[listIdx][rplsIdx]` bằng 1 chỉ ra rằng các LSB POC của các mục nhập LTRP trong cấu trúc cú pháp `ref_pic_list_struct(listIdx, rplsIdx)` là không hiện diện trong cấu trúc cú pháp `ref_pic_list_struct(listIdx, rplsIdx)`. Khi không hiện diện và long_term_ref_pics_flag là bằng 1, giá trị của `ltp_in_header_flag[listIdx][rplsIdx]` được suy ra là bằng 0. (Theo cách khác, khi không hiện diện và long_term_ref_pics_flag là bằng 1, giá trị của `ltp_in_header_flag[listIdx][rplsIdx]` được suy ra là bằng 1.)

6.1.3. Phương án của mục 3.6

Theo cách khác, `seq_parameter_set_rbsp()` và `ref_pic_list_struct()` được thay đổi như sau:

<code>seq_parameter_set_rbsp() {</code>	Mô tả
<code> ...</code>	
<code> for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {</code>	
<code> num_ref_pic_lists_in_sps[i]</code>	$ue(v)$
<code> for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++) {</code>	
<code> <u><code>ltp_in_header_flag[i][j]</code></u></code>	$u(1)$
<code> ref_pic_list_struct(i, j)</code>	
<code> for(k = 0; k < NumLtpEntries[i][j]; k++) {</code>	
<code> <u><code>if(!ltp_in_header_flag[i][j])</code></u></code>	
<code> <u><code>rpls_poc_lsb_lt[i][j][k]</code></u></code>	$u(v)$
<code> }</code>	
<code> }</code>	
<code> }</code>	
<code> ...</code>	
<code>}</code>	

`ltp_in_header_flag[i][j]` bằng 0 chỉ ra rằng các LSB POC của các mục nhập LTRP trong danh sách ảnh tham chiếu thứ j trong danh sách ảnh tham chiếu i là hiện diện trong cấu trúc cú pháp SPS. `ltp_in_header_flag[i][j]` bằng 1 chỉ ra rằng các LSB POC của các mục nhập LTRP trong danh sách ảnh tham

chiều thứ j trong danh sách ảnh tham chiếu i là hiện diện trong PH hoặc SH. Giá trị của $ltrp_in_header_flag[i][num_ref_pic_lists_in_sps[i]]$ được suy ra là bằng 0. (Theo cách khác, giá trị của $ltrp_in_header_flag[i][num_ref_pic_lists_in_sps[i]]$ được suy ra là bằng 1.)

$rpls_poc_lsb_lt[i][j][k]$ chỉ ra giá trị của môđun đếm thứ tự ảnh $MaxPicOrderCntLsb$ của mục nhập LTRP thứ k trong danh sách ảnh tham chiếu thứ j trong danh sách ảnh tham chiếu i. Chiều dài của phần tử cú pháp $rpls_poc_lsb_lt[i][j][k]$ bằng $\log_2 max_pic_order_cnt_lsb_minus4 + 4$ bit.

Và cú pháp `ref_pic_list_struct(listIdx, rplsIdx)` được thay đổi như sau:

<code>ref_pic_list_struct(listIdx, rplsIdx) {</code>	Mô tả
<code> num_ref_entries[listIdx][rplsIdx]</code>	ue(v)
<code> [[if(long term ref pics flag)</code>	
<code> ltrp_in_header_flag[listIdx][rplsIdx]</code>	u(1)]
<code> for(i = 0, j = 0; i < num_ref_entries[listIdx][rplsIdx]; i++) {</code>	
<code> if(inter layer ref pics present flag)</code>	
<code> inter_layer_ref_pic_flag[listIdx][rplsIdx][i]</code>	u(1)
<code> if(!inter_layer_ref_pic_flag[listIdx][rplsIdx][i]) {</code>	
<code> if(long term ref pics flag)</code>	
<code> st_ref_pic_flag[listIdx][rplsIdx][i]</code>	u(1)
<code> if(st_ref_pic_flag[listIdx][rplsIdx][i]) {</code>	
<code> abs_delta_poc_st[listIdx][rplsIdx][i]</code>	ue(v)
<code> if(AbsDeltaPocSt[listIdx][rplsIdx][i] > 0)</code>	
<code> strp_entry_sign_flag[listIdx][rplsIdx][i]</code>	u(1)
<code> } [[else if(!ltrp_in_header_flag[listIdx][rplsIdx])</code>	
<code> rpls_poc_lsb_lt[listIdx][rplsIdx][j++]</code>	u(v)]
<code> } else</code>	
<code> ilrp_idx[listIdx][rplsIdx][i]</code>	ue(v)
<code> }</code>	
<code>}</code>	

$[[ltrp_in_header_flag[listIdx][rplsIdx]$ bằng 0 chỉ ra rằng các LSB POC của các mục nhập LTRP trong cấu trúc cú pháp `ref_pic_list_struct(listIdx, rplsIdx)` là hiện diện trong cấu trúc cú pháp `ref_pic_list_struct(listIdx, rplsIdx)`. $ltrp_in_header_flag[listIdx][rplsIdx]$ bằng 1 chỉ ra rằng các LSB POC của các mục nhập LTRP trong cấu trúc cú pháp `ref_pic_list_struct(listIdx, rplsIdx)` là không hiện diện trong cấu trúc cú pháp `ref_pic_list_struct(listIdx, rplsIdx)`.

$rpls_poc_lsb_lt[listIdx][rplsIdx][i]$ chỉ ra giá trị của môđun đếm thứ tự ảnh $MaxPicOrderCntLsb$ của ảnh được tham chiếu tới bởi mục nhập thứ i trong cấu trúc cú pháp `ref_pic_list_struct(listIdx, rplsIdx)`. Chiều dài của phần tử cú pháp

`rpls_poc_lsb_lt[listIdx][rplsIdx][i]`

bằng

`log2_max_pic_order_cnt_lsb_minus4 + 4 bit.]]`

6.1.4. Phương án của mục 3.7

Theo cách khác, `seq_parameter_set_rbsp()` và `ref_pic_list_struct()` được thay đổi như sau:

<code>seq_parameter_set_rbsp()</code> {	Mô tả
...	
for(i = 0; i < rpl1 same as rpl0 flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
ltrp_in_header_flag[i]	u(1)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++) {	
ref_pic_list_struct(i, j)	
for(k = 0; k < NumLtrpEntries[i][j]; k++) {	
if(!ltrp_in_header_flag[i])	
rpls_poc_lsb_lt[i][j][k]	u(v)
}	
}	
}	
...	
}	

ltrp_in_header_flag[i] bằng 0 chỉ ra rằng các LSB POC của các mục nhập LTRP trong danh sách ảnh tham chiếu i là hiện diện trong cấu trúc cú pháp SPS.

ltrp_in_header_flag[i] bằng 1 chỉ ra rằng các LSB POC của các mục nhập LTRP trong danh sách ảnh tham chiếu i là hiện diện trong PH hoặc SH.

rpls_poc_lsb_lt[i][j][k] chỉ ra giá trị của môđun đếm thứ tư ảnh MaxPicOrderCntLsb của mục nhập LTRP thứ k trong danh sách ảnh tham chiếu thứ j trong danh sách ảnh tham chiếu i. Chiều dài của phần tử cú pháp *rpls_poc_lsb_lt[i][j][k]* bằng *log2_max_pic_order_cnt_lsb_minus4 + 4 bit*.

Và `ref_pic_lists()` syntax được thay đổi như sau:

<code>if(ltrp_in_header_flag[i][[RplsIdx[i]]])</code>	
<code> poc_lsb_lt[i][j]</code>	u(v)

Và cú pháp `ref_pic_list_struct(listIdx, rplsIdx)` được thay đổi như sau:

<code>ref_pic_list_struct(listIdx, rplsIdx)</code> {	Mô tả
num_ref_entries[listIdx][rplsIdx]	ue(v)
[[if(long_term_ref_pics_flag)	
ltrp_in_header_flag[listIdx][rplsIdx]	u(1)]]
for(i = 0, j = 0; i < num_ref_entries[listIdx][rplsIdx]; i++) {	
if(inter_layer_ref_pics_present_flag)	
inter_layer_ref_pic_flag[listIdx][rplsIdx][i]	u(1)
if(!inter_layer_ref_pic_flag[listIdx][rplsIdx][i]) {	

if(long term ref pics flag)	
st_ref_pic_flag [listIdx][rplsIdx][i]	u(1)
if(st_ref_pic_flag[listIdx][rplsIdx][i]) {	
abs_delta_poc_st [listIdx][rplsIdx][i]	ue(v)
if(AbsDeltaPocSt[listIdx][rplsIdx][i] > 0)	
strp_entry_sign_flag [listIdx][rplsIdx][i]	u(1)
} [[else if(!ltrp_in_header_flag[listIdx][rplsIdx])	
rpls_poc_lsb_lt [listIdx][rplsIdx][j++]	u(v)]
} else	
ilrp_idx [listIdx][rplsIdx][i]	ue(v)
}	
}	

[[**ltrp_in_header_flag**[listIdx][rplsIdx] bằng 0 chỉ ra rằng các LSB POC của các mục nhập LTRP trong cấu trúc cú pháp **ref_pic_list_struct**(listIdx, rplsIdx) là hiện diện trong cấu trúc cú pháp **ref_pic_list_struct**(listIdx, rplsIdx). **ltrp_in_header_flag**[listIdx][rplsIdx] bằng 1 chỉ ra rằng các LSB POC của các mục nhập LTRP trong cấu trúc cú pháp **ref_pic_list_struct**(listIdx, rplsIdx) là không hiện diện trong cấu trúc cú pháp **ref_pic_list_struct**(listIdx, rplsIdx).

rpls_poc_lsb_lt[listIdx][rplsIdx][i] chỉ ra giá trị của mô đun đếm thứ tự ảnh **MaxPicOrderCntLsb** của ảnh được tham chiếu tới bởi mục nhập thứ i trong cấu trúc cú pháp **ref_pic_list_struct**(listIdx, rplsIdx). Chiều dài của phần tử cú pháp **rpls_poc_lsb_lt**[listIdx][rplsIdx][i] bằng $\log_2_max_pic_order_cnt_lsb_minus4 + 4 \text{ bit.}]$

6.1.5. Phương án của mục 3.8

Theo cách khác, **seq_parameter_set_rbsp()** và **ref_pic_list_struct()** được thay đổi như sau:

seq_parameter_set_rbsp() {	Mô tả
...	
ltrp_in_header_flag	u(1)
for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps [i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++) {	
ref_pic_list_struct (i, j)	
for (k = 0; k < NumLtrpEntries [i][j]; k++) {	
if (! ltrp_in_header_flag [i])	
rpls_poc_lsb_lt [i][j][k]	u(v)
}	
}	
}	
...	
}	

ltrp_in_header_flag bằng 0 chỉ ra rằng các LSB POC của các mục nhập LTRP được chỉ báo trong mỗi trong số các cấu trúc cú pháp ref_pic_list_struct(i, j) đối với i nằm trong khoảng từ 0 tới (!rpl1_same_as_rpl0_flag ? 2 : 1), bao gồm các giá trị biên, và j nằm trong khoảng từ 0 tới num_ref_pic_lists_in_sps[i], bao gồm các giá trị biên, là hiện diện trong cấu trúc cú pháp SPS. ltrp_in_header_flag[i][j] bằng 1 chỉ ra rằng các LSB POC của các mục nhập LTRP được chỉ báo trong mỗi trong số các cấu trúc cú pháp ref_pic_list_struct(i, j) này là không hiện diện trong PH hoặc SH.

rpls_poc_lsb_lt[i][j][k] chỉ ra giá trị của môđun đếm thứ tư ảnh MaxPicOrderCntLsb của mục nhập LTRP thứ k trong danh sách ảnh tham chiếu thứ j trong danh sách ảnh tham chiếu i . Chiều dài của phần tử cú pháp rpls_poc_lsb_lt[i][j][k] bằng $\log_2 \text{max_pic_order_cnt_lsb_minus4} + 4$ bit.

Và cú pháp ref_pic_lists() được thay đổi như sau:

if(ltrp_in_header_flag [[i][RplsIdx[i]]])	
<u>poc_lsb_lt[i][j]</u>	u(v)

Và cú pháp ref_pic_list_struct(listIdx, rplsIdx) được thay đổi như sau:

<u>ref_pic_list_struct(listIdx, rplsIdx)</u> {	Mô tả
<u>num_ref_entries</u> [listIdx][rplsIdx]	ue(v)
[[if(long_term_ref_pics_flag)	
<u>ltrp_in_header_flag</u> [listIdx][rplsIdx]	u(1)]
for(i = 0, j = 0; i < <u>num_ref_entries</u> [listIdx][rplsIdx]; i++) {	
if(inter_layer_ref_pics_present_flag)	
<u>inter_layer_ref_pic_flag</u> [listIdx][rplsIdx][i]	u(1)
if(!inter_layer_ref_pic_flag[listIdx][rplsIdx][i]) {	
if(long_term_ref_pics_flag)	
<u>st_ref_pic_flag</u> [listIdx][rplsIdx][i]	u(1)
if(st_ref_pic_flag[listIdx][rplsIdx][i]) {	
<u>abs_delta_poc_st</u> [listIdx][rplsIdx][i]	ue(v)
if(AbsDeltaPocSt[listIdx][rplsIdx][i] > 0)	
<u>strp_entry_sign_flag</u> [listIdx][rplsIdx][i]	u(1)
} [[else if(!ltrp_in_header_flag[listIdx][rplsIdx])	
<u>rpls_poc_lsb_lt</u> [listIdx][rplsIdx][j++]	u(v)]
} else	
<u>ilrp_idx</u> [listIdx][rplsIdx][i]	ue(v)
}	
}	

[[ltrp_in_header_flag[listIdx][rplsIdx] bằng 0 chỉ ra rằng các LSB POC của các mục nhập LTRP trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) là hiện diện trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx).

ltp_in_header_flag[listIdx][rplsIdx] bằng 1 chỉ ra rằng các LSB POC của các mục nhập LTRP trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) là không hiện diện trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx).

rpls_poc_lsb_lt[listIdx][rplsIdx][i] chỉ ra giá trị của môđun đếm thứ tự ảnh MaxPicOrderCntLsb của ảnh được tham chiếu tới bởi mục nhập thứ i trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx). Chiều dài của phần tử cú pháp rpls_poc_lsb_lt[listIdx][rplsIdx][i] bằng $\log_2_max_pic_order_cnt_lsb_minus4 + 4 \text{ bit.}]$

6.2. Phương án thứ ba

Đây là phương án của mục 4.1 được tổng kết trên đây trong Phần 5.

Trong một ví dụ, các cấu trúc cú pháp picture_header_structure(), slice_header() và ref_pic_lists () được thay đổi như sau:

picture_header_structure() {	Mô tả
gdr or irap pic flag	u(1)
...	
if(rpl_info_in_ph_flag)	
ref_pic_lists(2)	
...	

slice_header() {	Mô tả
picture_header_in_slice_header_flag	u(1)
...	
if(!intra_only_constraint_flag && !rpl_info_in_ph_flag && ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag))	
ref_pic_lists(slice_type == B ? 2: 1)	
...	

ref_pic_lists(maxRefList) {	Mô tả
for(i = 0; i < maxRefList; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
rpl_sps_flag[i]	u(1)
if(rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtpEntries[i][RplsIdx[i]]; j++) {	
if(ltp_in_header_flag[i][RplsIdx[i]])	
poc_lsb_lt[i][j]	u(v)
delta_poc_msb_present_flag[i][j]	u(1)

if(delta poc msb present flag[i][j])	
delta poc msb cycle lt[i][j]	ue(v)
}	
}	
}	

FIG. 1 là sơ đồ khối thể hiện hệ thống xử lý video ví dụ 1900 mà trong đó các kỹ thuật khác nhau được mô tả ở đây có thể được thực hiện. Các phương án thực hiện khác nhau có thể bao gồm một số hoặc tất cả các thành phần của hệ thống 1900. Hệ thống 1900 có thể bao gồm đầu vào 1902 để nhận nội dung video. Nội dung video có thể được nhận trong định dạng thô hoặc không nén, ví dụ, các giá trị điểm ảnh đa thành phần 8 hoặc 10 bit, hoặc có thể trong định dạng được nén hoặc được mã hóa. Đầu vào 1902 có thể biểu diễn giao diện mạng, giao diện buýt ngoại vi, hoặc giao diện bộ lưu trữ. Các ví dụ về giao diện mạng bao gồm các giao diện dây dẫn như Ethernet, mạng quang học thụ động (PON – passive optical network), v.v. và các giao diện không dây như Wi-Fi hoặc các giao diện dạng ô.

Hệ thống 1900 có thể bao gồm thành phần tạo mã 1904 mà có thể thực hiện các phương pháp tạo mã hoặc mã hóa khác nhau mà được mô tả trong bản mô tả này. Thành phần tạo mã 1904 có thể giảm tốc độ bit trung bình của video từ đầu vào 1902 tới đầu ra của thành phần tạo mã 1904 để tạo ra sự biểu diễn được tạo mã của video. Do đó, các kỹ thuật tạo mã này đôi khi được gọi là nén video hoặc các kỹ thuật chuyển mã video. Đầu ra của thành phần tạo mã 1904 có thể hoặc được lưu, hoặc được truyền qua đường truyền thông được kết nối, như được biểu diễn bởi thành phần 1906. Sự biểu diễn dòng bit được lưu hoặc được truyền thông (hoặc được tạo mã) của video được nhận ở đầu vào 1902 có thể được sử dụng bởi thành phần 1908 để tạo ra các giá trị điểm ảnh hoặc video có thể hiển thị mà được gửi tới giao diện hiển thị 1910. Quá trình tạo ra video người dùng xem được từ sự biểu diễn dòng bit đôi khi được gọi là giải nén video. Hơn nữa, trong khi những hoạt động xử lý video đã biết được gọi là các hoạt động hoặc các công cụ “tạo mã”, thì cần hiểu rằng các công cụ hoặc các hoạt động tạo mã được sử dụng ở bộ mã hóa và các công cụ hoặc các hoạt động giải mã tương ứng mà đảo ngược các kết quả của việc tạo mã sẽ được thực hiện bởi bộ giải mã.

Các ví dụ về giao diện buýt ngoại vi hoặc giao diện hiển thị có thể bao gồm buýt nối tiếp vạn năng (USB - universal serial bus) hoặc giao diện đa phương tiện độ phân giải cao (HDMI - high definition multimedia interface) hoặc Displayport, và v.v. Các ví dụ về các giao diện bộ lưu trữ bao gồm giao diện SATA (serial advanced technology attachment), PCI, IDE, và tương tự. Các kỹ thuật được mô tả trong bản mô tả có thể được thực hiện trong các thiết bị điện tử khác nhau như các điện thoại di động, các máy tính xách tay, các điện thoại thông minh hoặc các thiết bị khác mà có khả năng thực hiện xử lý dữ liệu kỹ thuật số và/hoặc hiển thị video.

FIG. 2 là sơ đồ khối của thiết bị xử lý video 3600. Thiết bị 3600 có thể được sử dụng để thực hiện một hoặc nhiều trong số các phương pháp được mô tả ở đây. Thiết bị 3600 có thể được thực hiện trong điện thoại thông minh, máy tính bảng, máy tính, bộ thu internet vạn vật (IoT - Internet of Things), và v.v.. Thiết bị 3600 có thể bao gồm một hoặc nhiều bộ xử lý 3602, một hoặc nhiều bộ nhớ 3604 và phần cứng xử lý video 3606. (Các) bộ xử lý 3602 có thể được tạo cấu hình để thực hiện một hoặc nhiều phương pháp được mô tả trong bản mô tả. Bộ nhớ (các bộ nhớ) 3604 có thể được sử dụng để lưu trữ dữ liệu và mã được sử dụng để thực hiện các phương pháp và các kỹ thuật được mô tả ở đây. Phần cứng xử lý video 3606 có thể được sử dụng để thực hiện, trong hệ mạch phần cứng, một số kỹ thuật được mô tả trong bản mô tả. Theo một số phương án, phần cứng 3606 có thể một phần hoặc toàn bộ trong bộ xử lý 3602, ví dụ, bộ xử lý đồ họa.

FIG. 4 là sơ đồ khối mà minh họa hệ thống tạo mã video 100 ví dụ mà có thể sử dụng các kỹ thuật theo sáng chế.

Như được thể hiện trên FIG. 4, hệ thống tạo mã video 100 có thể bao gồm thiết bị nguồn 110 và thiết bị đích 120. Thiết bị nguồn 110 tạo ra dữ liệu video được mã hóa mà có thể được gọi là thiết bị mã hóa video. Thiết bị đích 120 có thể giải mã dữ liệu video được mã hóa được tạo ra bởi thiết bị nguồn 110 mà có thể được gọi là thiết bị giải mã video.

Thiết bị nguồn 110 có thể bao gồm nguồn video 112, bộ mã hóa video 114, và giao diện đầu vào/đầu ra (I/O) 116.

Nguồn video 112 có thể bao gồm nguồn như thiết bị quay video, giao diện để nhận dữ liệu video từ nhà cung cấp nội dung video, và/hoặc hệ thống đồ họa máy tính để tạo ra dữ liệu video, hoặc sự kết hợp của các nguồn này. Dữ liệu video có thể bao gồm một hoặc nhiều ảnh. Bộ mã hóa video 114 mã hóa dữ liệu video từ nguồn video 112 để tạo ra dòng bit. Dòng bit có thể bao gồm dãy của các bit mà tạo thành sự biểu diễn được tạo mã của dữ liệu video. Dòng bit có thể bao gồm các ảnh được tạo mã và dữ liệu liên kết. Ảnh được tạo mã là sự biểu diễn được tạo mã của ảnh. Dữ liệu liên kết có thể bao gồm các tập thông số dãy, các tập thông số ảnh, và các cấu trúc cú pháp khác. Giao diện I/O 116 có thể bao gồm bộ điều biến/giải điều biến (môdem) và/hoặc bộ truyền. Dữ liệu video được mã hóa có thể được truyền trực tiếp tới thiết bị đích 120 qua giao diện I/O 116 thông qua mạng 130a. Dữ liệu video được mã hóa cũng có thể được lưu trong vật lưu trữ/máy chủ 130b để truy nhập bởi thiết bị đích 120.

Thiết bị đích 120 có thể bao gồm giao diện I/O 126, bộ giải mã video 124, và thiết bị hiển thị 122.

Giao diện I/O 126 có thể bao gồm bộ thu và/hoặc môdem. Giao diện I/O 126 có thể thu thập dữ liệu video được mã hóa từ thiết bị nguồn 110 hoặc vật lưu trữ/máy chủ 130b. Bộ giải mã video 124 có thể giải mã dữ liệu video được mã hóa. Thiết bị hiển thị 122 có thể hiển thị dữ liệu video được giải mã tới người dùng. Thiết bị hiển thị 122 có thể được tích hợp với thiết bị đích 120, hoặc có thể là ngoại vi đối với thiết bị đích 120 mà được tạo cấu hình để giao diện với thiết bị hiển thị ngoại vi.

Bộ mã hóa video 114 và bộ giải mã video 124 có thể hoạt động theo chuẩn nén video, như chuẩn tạo mã video hiệu quả cao (HEVC - High Efficiency Video Coding), chuẩn tạo mã video đa năng (VVC – Versatile Video Coding) và các chuẩn hiện tại và/hoặc tiếp theo khác.

FIG. 5 là sơ đồ khối minh họa ví dụ về bộ mã hóa video 200, mà có thể là bộ mã hóa video 114 trong hệ thống 100 được minh họa trên FIG. 4.

Bộ mã hóa video 200 có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật theo sáng chế. Trong ví dụ trên FIG. 5, bộ mã hóa video 200 bao gồm nhiều thành phần chức năng. Các kỹ thuật được mô tả trong bản

mô tả này có thể được chia sẻ giữa các thành phần khác nhau của bộ mã hóa video 200. Trong một số ví dụ, bộ xử lý có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật được mô tả trong bản mô tả này.

Các thành phần chức năng của bộ mã hóa video 200 có thể bao gồm khối phân vùng 201, khối xác nhận 202 mà có thể bao gồm khối chọn chế độ 203, khối ước tính chuyển động 204, khối bù chuyển động 205 và khối dự báo trong 206, khối tạo dư 207, khối chuyển đổi 208, khối lượng tử hóa 209, khối lượng tử hóa ngược 210, khối chuyển đổi ngược 211, khối tái tạo 212, bộ đệm 213, và khối mã hóa entropi 214.

Trong các ví dụ khác, bộ mã hóa video 200 có thể bao gồm nhiều hơn, ít hơn, hoặc các thành phần chức năng khác nhau. Trong một ví dụ, khối xác nhận 202 có thể bao gồm khối sao chép khối trong (IBC). Khối IBC có thể thực hiện việc xác nhận trong chế độ IBC mà trong đó ít nhất một ảnh tham chiếu là ảnh mà khối video hiện tại nằm trong đó.

Hơn nữa, một số thành phần, như khối ước tính chuyển động 204 và khối bù chuyển động 205 có thể được tích hợp ở mức độ cao, nhưng được biểu diễn trong ví dụ trên FIG. 5 một cách riêng biệt cho mục đích giải thích.

Khối phân vùng 201 có thể phân vùng ảnh thành một hoặc nhiều khối video. Bộ mã hóa video 200 và bộ giải mã video 300 có thể hỗ trợ các kích thước khối video khác nhau.

Khối chọn chế độ 203 có thể chọn một trong số các chế độ tạo mã, trong hoặc ngoài, ví dụ, dựa trên các kết quả lỗi, và cấp khối được tạo mã trong hoặc ngoài thu được tới khối tạo dư 207 để tạo ra khối dữ liệu dư và tới khối tái tạo 212 để tái tạo khối được mã hóa để sử dụng làm ảnh tham chiếu. Trong một số ví dụ, khối chọn chế độ 203 có thể chọn chế độ sự kết hợp của xác nhận trong và ngoài (CIIP – combination of intra and inter predication) mà trong đó sự xác nhận là dựa trên tín hiệu xác nhận ngoài và tín hiệu xác nhận trong. Khối chọn chế độ 203 cũng có thể chọn độ phân giải đối với vector chuyển động (ví dụ, độ chính xác điểm ảnh con hoặc điểm ảnh nguyên) đối với khối này trong trường hợp xác nhận ngoài.

Để thực hiện dự báo ngoài trên khối video hiện tại, khối ước tính chuyển động 204 có thể tạo ra thông tin chuyển động đối với khối video hiện tại bằng cách so sánh một hoặc nhiều khung tham chiếu từ bộ đệm 213 với khối video hiện tại. Khối bù chuyển động 205 có thể xác định khối video được dự báo đối với khối video hiện tại dựa trên thông tin chuyển động và các mẫu được giải mã của các ảnh từ bộ đệm 213 khác với ảnh được liên kết với khối video hiện tại.

Khối ước tính chuyển động 204 và khối bù chuyển động 205 có thể thực hiện các hoạt động khác nhau đối với khối video hiện tại, ví dụ, phụ thuộc vào việc liệu khối video hiện tại đang ở trong lát I, lát P, hoặc lát B.

Trong một số ví dụ, khối ước tính chuyển động 204 có thể thực hiện dự báo đơn hướng đối với khối video hiện tại, và khối ước tính chuyển động 204 có thể tìm kiếm các ảnh tham chiếu của danh sách 0 hoặc danh sách 1 đối với khối video tham chiếu đối với khối video hiện tại. Sau đó khối ước tính chuyển động 204 có thể tạo ra chỉ số tham chiếu mà chỉ báo ảnh tham chiếu trong danh sách 0 hoặc danh sách 1 mà chứa khối video tham chiếu và vector chuyển động mà chỉ báo sự dịch chuyển không gian giữa khối video hiện tại và khối video tham chiếu. Khối ước tính chuyển động 204 có thể xuất ra chỉ số tham chiếu, bộ chỉ báo hướng dự báo, và vector chuyển động làm thông tin chuyển động của khối video hiện tại. Khối bù chuyển động 205 có thể tạo ra khối video được dự báo của khối hiện tại dựa trên khối video tham chiếu được chỉ báo bởi thông tin chuyển động của khối video hiện tại.

Trong các ví dụ khác, khối ước tính chuyển động 204 có thể thực hiện dự báo hai hướng đối với khối video hiện tại, khối ước tính chuyển động 204 có thể tìm kiếm các ảnh tham chiếu trong danh sách 0 đối với khối video tham chiếu đối với khối video hiện tại và cũng có thể tìm kiếm các ảnh tham chiếu trong danh sách 1 đối với khối video tham chiếu khác đối với khối video hiện tại. Sau đó khối ước tính chuyển động 204 có thể tạo ra các chỉ số tham chiếu mà chỉ báo các ảnh tham chiếu trong danh sách 0 và danh sách 1 chứa các khối video tham chiếu và các vector chuyển động mà chỉ báo các dịch chuyển không gian giữa các khối video tham chiếu và khối video hiện tại. Khối ước tính chuyển động 204 có thể xuất ra các chỉ số tham chiếu và các vector chuyển động của khối video hiện

tại làm thông tin chuyển động của khối video hiện tại. Khối bù chuyển động 205 có thể tạo ra khối video được dự báo của khối video hiện tại dựa trên các khối video tham chiếu được chỉ báo bởi thông tin chuyển động của khối video hiện tại.

Trong một số ví dụ, khối ước tính chuyển động 204 có thể xuất ra tập đầy đủ của thông tin chuyển động cho quá trình xử lý giải mã của bộ giải mã.

Trong một số ví dụ, khối ước tính chuyển động 204 có thể không xuất ra tập đầy đủ của thông tin chuyển động đối với video hiện tại. Thay vào đó, khối ước tính chuyển động 204 có thể báo tín hiệu thông tin chuyển động của khối video hiện tại nhờ tham chiếu tới thông tin chuyển động của khối video khác. Ví dụ, khối ước tính chuyển động 204 có thể xác định rằng thông tin chuyển động của khối video hiện tại là đủ giống với thông tin chuyển động của khối video lân cận.

Trong một ví dụ, khối ước tính chuyển động 204 có thể chỉ báo, trong cấu trúc cú pháp liên kết với khối video hiện tại, giá trị mà chỉ báo tới bộ giải mã video 300 rằng khối video hiện tại có thông tin chuyển động giống như khối video khác.

Trong một ví dụ khác, khối ước tính chuyển động 204 có thể nhận dạng, trong cấu trúc cú pháp liên kết với khối video hiện tại, khối video khác và hiệu số vector chuyển động (MVD). Hiệu số vector chuyển động chỉ báo hiệu số giữa vector chuyển động của khối video hiện tại và vector chuyển động của khối video được chỉ báo. Bộ giải mã video 300 có thể sử dụng vector chuyển động của khối video được chỉ báo và hiệu số vector chuyển động để xác định vector chuyển động của khối video hiện tại.

Như đã đề cập trên đây, bộ mã hóa video 200 có thể báo tín hiệu theo cách dự báo vector chuyển động. Hai ví dụ về các kỹ thuật báo tín hiệu theo cách dự báo mà có thể được thực hiện bởi bộ mã hóa video 200 bao gồm xác nhận vector chuyển động cao cấp (AMVP – advanced motion vector predication) và báo tín hiệu chế độ kết hợp.

Khối dự báo trong 206 có thể thực hiện dự báo trong trên khối video hiện tại. Khi khối dự báo trong 206 thực hiện dự báo trong trên khối video hiện tại,

khối dự báo trong 206 có thể tạo ra dữ liệu dự báo đối với khối video hiện tại dựa trên các mẫu được giải mã của các khối video khác trong cùng một ảnh. Dữ liệu dự báo đối với khối video hiện tại có thể bao gồm khối video được dự báo và các phần tử cú pháp khác nhau.

Khối tạo dư 207 có thể tạo ra dữ liệu dư đối với khối video hiện tại bằng cách trừ (ví dụ, được chỉ báo bởi dấu trừ) (các) khối video được dự báo của khối video hiện tại từ khối video hiện tại. Dữ liệu dư của khối video hiện tại có thể bao gồm các khối video dư mà tương ứng với các thành phần mẫu khác nhau của các mẫu trong khối video hiện tại.

Trong các ví dụ khác, có thể không có dữ liệu dư đối với khối video hiện tại đối với khối video hiện tại, ví dụ trong chế độ bỏ qua, và khối tạo dư 207 có thể không thực hiện hoạt động trừ.

Khối xử lý chuyển đổi 208 có thể tạo ra một hoặc nhiều khối video hệ số chuyển đổi đối với khối video hiện tại bằng cách áp dụng một hoặc nhiều sự chuyển đổi vào khối video dư liên kết với khối video hiện tại.

Sau khi khối xử lý chuyển đổi 208 tạo ra khối video hệ số chuyển đổi liên kết với khối video hiện tại, khối lượng tử hóa 209 có thể lượng tử hóa khối video hệ số chuyển đổi liên kết với khối video hiện tại dựa trên một hoặc nhiều giá trị thông số lượng tử hóa (QP) liên kết với khối video hiện tại.

Khối lượng tử hóa ngược 210 và khối chuyển đổi ngược 211 có thể lần lượt áp dụng lượng tử hóa ngược và các chuyển đổi ngược vào khối video hệ số chuyển đổi, để tái tạo khối video dư từ khối video hệ số chuyển đổi. Khối tái tạo 212 có thể bổ sung khối video dư được tái lập vào các mẫu tương ứng từ một hoặc nhiều khối video được dự báo được tạo ra bởi khối xác nhận 202 để tạo ra khối video được tái lập liên kết với khối hiện tại để lưu trữ trong bộ đệm 213.

Sau khi khối tái tạo 212 tái tạo khối video, hoạt động lọc vòng có thể được thực hiện để giảm các thành phần lạ tạo khối video trong khối video.

Khối mã hóa entropi 214 có thể nhận dữ liệu từ các thành phần chức năng khác của bộ mã hóa video 200. Khi khối mã hóa entropi 214 nhận dữ liệu, khối mã hóa entropi 214 có thể thực hiện một hoặc nhiều hoạt động mã hóa entropi

để tạo ra dữ liệu được mã hóa entropi và xuất ra dòng bit mà bao gồm dữ liệu được mã hóa entropi.

FIG. 6 là sơ đồ khối minh họa ví dụ của bộ giải mã video 300 mà có thể là bộ giải mã video 114 trong hệ thống 100 được minh họa trên FIG. 4.

Bộ giải mã video 300 có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật theo sáng chế. Trong ví dụ trên FIG. 6, bộ giải mã video 300 bao gồm nhiều thành phần chức năng. Các kỹ thuật được mô tả trong bản mô tả này có thể được chia sẻ giữa các thành phần khác nhau của bộ giải mã video 300. Trong một số ví dụ, bộ xử lý có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật được mô tả trong bản mô tả này.

Trong ví dụ trên FIG. 6, bộ giải mã video 300 bao gồm khối giải mã entropi 301, khối bù chuyển động 302, khối dự báo trong 303, khối lượng tử hóa ngược 304, khối chuyển đổi ngược 305, và khối tái tạo 306 và bộ đệm 307. Bộ giải mã video 300 có thể, trong một số ví dụ, thực hiện chuyển tác giải mã mà thường là thuận nghịch với chuyển tác mã hóa được mô tả đối với bộ mã hóa video 200 (FIG. 5).

Khối giải mã entropi 301 có thể truy xuất dòng bit được mã hóa. Dòng bit được mã hóa có thể bao gồm dữ liệu video được tạo mã entropi (ví dụ, các khối được mã hóa của dữ liệu video). Khối giải mã entropi 301 có thể giải mã dữ liệu video được tạo mã entropi, và từ dữ liệu video được giải mã entropi, khối bù chuyển động 302 có thể xác định thông tin chuyển động bao gồm các vector chuyển động, độ chính xác vector chuyển động, các chỉ số danh sách ảnh tham chiếu, và thông tin chuyển động khác. Khối bù chuyển động 302 có thể, ví dụ, xác định thông tin này bằng cách thực hiện AMVP và chế độ kết hợp.

Khối bù chuyển động 302 có thể tạo ra các khối bù chuyển động, có thể bằng cách thực hiện nội suy dựa trên các bộ lọc nội suy. Các bộ nhận dạng đối với các bộ lọc nội suy cần được sử dụng với độ chính xác điểm ảnh con có thể được bao gồm trong các phần tử cú pháp.

Khối bù chuyển động 302 có thể sử dụng các bộ lọc nội suy như được sử dụng bởi bộ mã hóa video 200 trong khi mã hóa của khối video để tính toán các giá trị nội suy đối với các điểm ảnh con số nguyên của khối tham chiếu. Khối bù

chuyển động 302 có thể xác định các bộ lọc nội suy được sử dụng bởi bộ mã hóa video 200 theo thông tin cú pháp nhận được và sử dụng các bộ lọc nội suy để tạo ra các khối dự báo.

Khối bù chuyển động 302 có thể sử dụng một số thông tin cú pháp để xác định các kích thước của các khối được sử dụng để mã hóa (các) khung và/hoặc (các) lát của dãy video được mã hóa, thông tin phân vùng mà mô tả cách thức mỗi khối macro của ảnh của dãy video được mã hóa được phân vùng, các chế độ mà chỉ báo cách thức mỗi phân vùng được mã hóa, một hoặc nhiều khung tham chiếu (và các danh sách khung tham chiếu) đối với mỗi khối được mã hóa ngoài, và thông tin khác để giải mã dãy video được mã hóa.

Khối dự báo trong 303 có thể sử dụng các chế độ dự báo trong ví dụ được nhận trong dòng bit để tạo thành khối dự báo từ các khối liền kề trong không gian. Khối lượng tử hóa ngược 303 lượng tử hóa ngược, tức là, giải lượng tử hóa, các hệ số khối video được lượng tử hóa được tạo ra trong dòng bit và được giải mã bởi khối giải mã entropi 301. Khối chuyển đổi ngược 303 áp dụng chuyển đổi ngược.

Khối tái tạo 306 có thể sum các khối dư với các khối dự báo tương ứng được tạo ra bởi khối bù chuyển động 202 hoặc khối dự báo trong 303 để tạo thành các khối được giải mã. Nếu mong muốn, bộ lọc giải khối cũng có thể được áp dụng để lọc các khối được giải mã nhằm loại bỏ các thành phần lạ tạo khối. Sau đó các khối video được giải mã được lưu trong bộ đệm 307, mà cấp các khối tham chiếu để bù chuyển động/xác nhận trong tiếp sau và còn tạo ra video được giải mã để trình diễn trên thiết bị hiển thị.

Việc liệt kê các ví dụ ưu tiên bởi một số phương án được thực hiện tiếp theo đây.

Nhóm thứ nhất của các điều thể hiện các phương án ví dụ của các kỹ thuật được đề cập trong phần trước (ví dụ, mục 1).

1. Phương pháp xử lý video (ví dụ, phương pháp 300 được mô tả trên FIG. 3), bao gồm các bước: thực hiện (302) sự biến đổi giữa video có một hoặc nhiều ảnh và sự biểu diễn được tạo mã của video, trong đó mỗi ảnh trong số một hoặc nhiều ảnh bao gồm chính xác một lát; trong đó sự biểu diễn được tạo mã tương

thích với quy tắc định dạng; trong đó quy tắc định dạng chỉ ra rằng các cờ tập thông số ảnh được loại trừ khỏi sự biểu diễn được tạo mã đúng với ảnh có chính xác một lát.

2. Phương pháp theo điều 1, trong đó quy tắc định dạng chỉ ra rằng sự chỉ báo về việc liệu mỗi ảnh mà tham chiếu tới tập thông số ảnh có được báo tín hiệu trong tập thông số ảnh.

3. Phương pháp theo điều 1, trong đó, quy tắc định dạng còn chỉ ra việc bỏ qua sự chỉ báo về sự không hiện diện của các cờ tập thông số ảnh trong sự biểu diễn được tạo mã.

4. Phương pháp theo điều 1, trong đó quy tắc định dạng chỉ ra rằng, đối với ảnh có chính xác một lần lát, sự biểu diễn được tạo mã bỏ qua sự báo tín hiệu của cờ mà chỉ báo sự hiện diện của tiêu đề ảnh trong tiêu đề lát.

Các điều sau đây thể hiện các phương án ví dụ của các kỹ thuật được đề cập trong phần trước (ví dụ, mục 2).

5. Phương pháp xử lý video, bao gồm các bước: thực hiện sự biến đổi giữa lát video của video và sự biểu diễn được tạo mã của video, trong đó sự biểu diễn được tạo mã tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra việc bao gồm trường thứ nhất chỉ báo về sự hiện diện của trường thứ hai mà nhận dạng chỉ số lớn nhất đối với ảnh tham chiếu được sử dụng để tạo mã lát video dựa trên điều kiện; trong đó điều kiện có thể được biểu diễn dưới dạng biểu thức logic “A & (B || C)”, mà trong đó A chỉ báo “rpl_info_in_ph_flag = 1, hoặc kiểu đơn vị NAL không bằng IDR, hoặc sps_idr_rpl_present_flag là bằng 1”, và “B” chỉ báo “slice_type là bằng P hoặc B, và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 0 là lớn hơn 1”, và “C” chỉ báo “slice_type là bằng B và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 1 là lớn hơn 1” và “&” là phép toán “và” logic và “||” là phép toán “hoặc” logic.

Các điều sau đây thể hiện các phương án ví dụ của các kỹ thuật được đề cập trong phần trước (ví dụ, mục 3).

6. Phương pháp xử lý video, bao gồm các bước: thực hiện sự biến đổi giữa vùng video của video và sự biểu diễn được tạo mã của video, trong đó sự biểu diễn được tạo mã tương thích với quy tắc định dạng, trong đó quy tắc định dạng

chỉ ra điều kiện mà điều khiển việc sử dụng phần tử cú pháp chỉ báo về số đếm thứ tự ảnh đối với ảnh tham chiếu dài hạn được sử dụng để biến đổi của vùng video.

7. Phương pháp theo điều 6, trong đó quy tắc định dạng chỉ ra rằng phần tử cú pháp được bao gồm trong sự biểu diễn được tạo mã nếu và chỉ nếu chỉ số của danh sách ảnh tham chiếu là nhỏ hơn số lượng của cấu trúc cú pháp danh sách ảnh tham chiếu mà được bao gồm trong tập thông số dãy.

8. Phương pháp theo điều 6, trong đó quy tắc định dạng chỉ ra rằng phần tử cú pháp được bao gồm trong sự biểu diễn được tạo mã nếu và chỉ nếu cấu trúc cú pháp danh sách ảnh tham chiếu được bao gồm trong tập thông số dãy của sự biểu diễn được tạo mã.

Các điều sau đây thể hiện các phương án ví dụ của các kỹ thuật được đề cập trong phần trước (ví dụ, mục 4).

9. Phương pháp xử lý video, bao gồm các bước: thực hiện sự biến đổi giữa vùng video của video và sự biểu diễn được tạo mã của video, trong đó sự biểu diễn được tạo mã tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra rằng các cấu trúc cú pháp liên quan đến ảnh tham chiếu được bao gồm nếu và chỉ nếu biến số X bằng giá trị cụ thể, trong đó X phụ thuộc vào kiểu lát của vùng video.

10. Phương pháp theo điều 9, trong đó giá trị cụ thể là bằng 1 hoặc 2.

11. Phương pháp theo điều bất kỳ trong số các điều trên đây, trong đó vùng video bao gồm đơn vị tạo mã video.

12. Phương pháp theo điều bất kỳ trong số các điều trên đây, trong đó vùng video bao gồm ảnh video.

13. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 12, trong đó sự biến đổi bao gồm việc mã hóa video thành sự biểu diễn được tạo mã.

14. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 12, trong đó sự biến đổi bao gồm việc giải mã sự biểu diễn được tạo mã để tạo ra các giá trị điểm ảnh của video.

15. Thiết bị giải mã video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều điều trong số các điều từ 1 tới 14.

16. Thiết bị mã hóa video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều điều trong số các điều từ 1 tới 14.

17. Sản phẩm chương trình máy tính có mã máy tính được lưu trong đó, mã này, khi được thực thi bởi bộ xử lý, khiến bộ xử lý thực hiện phương pháp được nêu trong điều bất kỳ trong số các điều từ 1 tới 14.

18. Phương pháp, thiết bị hoặc hệ thống được mô tả trong bản mô tả.

Nhóm thứ hai của các điều mô tả các dấu hiệu đã biết và các khía cạnh của các kỹ thuật được mô tả trong phần trước (ví dụ, các mục từ 1.1.a tới 1.1.f và từ 1.1.h tới 1.1.p trong phần 'Các kỹ thuật và các phương án ví dụ').

1. Phương pháp xử lý video (ví dụ, phương pháp 710 như được thể hiện trên FIG. 7A), bao gồm các bước: bước thực hiện 712 sự biến đổi giữa video bao gồm một hoặc nhiều ảnh và dòng bit của video; trong đó dòng bit tương thích với quy tắc định dạng; trong đó quy tắc định dạng chỉ ra rằng một hoặc nhiều phần tử cú pháp chỉ báo về thông tin tạo mã được bao gồm theo cách có lựa chọn trong tập thông số ảnh của dòng bit đáp lại việc liệu có được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát.

2. Phương pháp theo điều 1, trong đó một hoặc nhiều phần tử cú pháp bao gồm sáu cờ mà chỉ báo về việc liệu các phần tử cú pháp tạo mã đã biết có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

3. Phương pháp theo điều 2, trong đó sáu cờ bao gồm cờ thứ nhất chỉ báo về việc liệu thông tin danh sách ảnh tham chiếu có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

4. Phương pháp theo điều 3, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của cờ thứ nhất được suy ra là bằng 0 trong trường hợp mà cờ thứ nhất không hiện diện trong tập thông số ảnh.

5. Phương pháp theo các điều từ 2 tới 4, trong đó sáu cờ bao gồm cờ thứ hai chỉ báo về việc liệu thông tin lọc giải khối có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

6. Phương pháp theo điều bất kỳ trong số điều 5, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của cờ thứ hai được suy ra là bằng 0 trong trường hợp mà cờ thứ hai không hiện diện trong tập thông số ảnh.

7. Phương pháp theo điều bất kỳ trong số các điều từ 2 tới 6, trong đó sáu cờ bao gồm cờ thứ ba chỉ báo về việc liệu thông tin lọc dịch vị thích ứng mẫu có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

8. Phương pháp theo điều bất kỳ trong số điều 7, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của cờ thứ ba được suy ra là bằng 0 trong trường hợp mà cờ thứ ba không hiện diện trong tập thông số ảnh.

9. Phương pháp theo điều bất kỳ trong số các điều từ 2 tới 8, trong đó sáu cờ bao gồm cờ thứ tư chỉ báo về việc liệu thông tin lọc vòng thích ứng có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

10. Phương pháp theo điều bất kỳ trong số điều 9, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của cờ thứ tư được suy ra là bằng 0 trong trường hợp mà cờ thứ tư không hiện diện trong tập thông số ảnh.

11. Phương pháp theo điều bất kỳ trong số các điều từ 2 tới 10, trong đó sáu cờ bao gồm cờ thứ năm chỉ báo về việc liệu thông tin dự báo có trọng số có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

12. Phương pháp theo điều 11, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của cờ thứ năm được suy ra là bằng 0 trong trường hợp mà cờ thứ năm không hiện diện trong tập thông số ảnh.

13. Phương pháp theo điều bất kỳ trong số các điều từ 2 tới 12, trong đó sáu cờ bao gồm cờ thứ sáu chỉ báo về việc liệu thông tin delta thông số lượng tử hóa có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

14. Phương pháp theo điều bất kỳ trong số điều 13, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của cờ thứ sáu được suy ra là bằng 0 trong trường hợp mà cờ thứ sáu không hiện diện trong tập thông số ảnh.

15. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 14, trong đó ít nhất một cờ bao gồm cờ thứ bảy mà chỉ báo việc liệu các hoạt động lọc trong vòng ngang qua các biên lát có được phép đối với các ảnh tham chiếu tới tập thông số ảnh.

16. Phương pháp theo điều bất kỳ trong số điều 15, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của cờ thứ bảy được suy ra là bằng 0 trong trường hợp mà cờ thứ bảy không hiện diện trong tập thông số ảnh.

17. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 16, trong đó quy tắc định dạng còn chỉ ra việc bỏ qua một hoặc nhiều phần tử cú pháp trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát.

18. Phương pháp theo điều 17, trong đó quy tắc định dạng còn chỉ ra việc bao gồm sự chỉ báo mà chỉ báo việc liệu mỗi ảnh mà tham chiếu tới tập thông số ảnh chỉ có một lát trong tập thông số ảnh.

19. Phương pháp theo điều 17, trong đó quy tắc định dạng còn chỉ ra việc bao gồm sự chỉ báo mà chỉ báo việc liệu mỗi ảnh mà tham chiếu tới tập thông số ảnh chỉ có một lát trong tập thông số ảnh.

20. Phương pháp theo điều 17, trong đó quy tắc định dạng còn chỉ ra việc bao gồm sự chỉ báo mà chỉ báo việc liệu mỗi ảnh mà tham chiếu tới tập thông số ảnh chỉ có một lát ngoài trong tập thông số ảnh.

21. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra rằng sự chỉ báo mà chỉ báo việc liệu mỗi ảnh mà tham chiếu tới tập thông số ảnh chỉ có một lát tương ứng với cờ được bao gồm trong tập thông số ảnh.

22. Phương pháp theo điều 21, trong đó cờ có giá trị bằng 1 chỉ ra rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh chỉ có một lát và cờ có giá trị bằng 0 chỉ ra rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh có nhiều hơn một lát.

23. Phương pháp theo điều 21, trong đó quy tắc định dạng còn chỉ ra việc bỏ qua sự chỉ báo và suy ra sự chỉ báo từ thông tin phân vùng lát được bao gồm trong tập thông số ảnh.

24. Phương pháp theo điều 21, trong đó quy tắc định dạng còn chỉ ra rằng sự chỉ báo là bằng 0 trong trường hợp sự chỉ báo khác mà chỉ báo việc liệu chỉ có một ảnh con trong mỗi ảnh là lớn hơn 0.

25. Phương pháp theo điều 21, trong đó quy tắc định dạng còn chỉ ra rằng cờ là giống nhau đối với tất cả các tập thông số ảnh được tham chiếu bởi một hoặc nhiều ảnh trong dãy video lớp được tạo mã.

26. Phương pháp theo điều 21, trong đó quy tắc định dạng còn chỉ ra rằng cờ là bằng 1 trong trường hợp mà `one_slice_per_pic_constraint_flag` là bằng 1.

27. Phương pháp theo điều 21, trong đó quy tắc định dạng còn chỉ ra việc cập nhật các ngữ nghĩa của `one_slice_per_pic_constraint_flag` bằng cách bổ sung rằng `one_slice_per_pic_constraint_flag` bằng 1 chỉ ra rằng cờ này là bằng 1.

28. Phương pháp theo điều 21, trong đó quy tắc định dạng còn chỉ ra việc bỏ qua cờ trong trường hợp mà được biết từ tập thông số ảnh rằng chỉ có một ảnh con trong mỗi ảnh và rằng mỗi ảnh con chỉ chứa một lát.

29. Phương pháp theo điều 17, trong đó quy tắc định dạng còn chỉ ra rằng các giá trị của một hoặc nhiều phần tử cú pháp được suy ra là bằng 0 hoặc 1 trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát.

30. Phương pháp theo điều 17, trong đó một hoặc nhiều phần tử cú pháp bao gồm cờ thứ tám (`wp_info_in_ph_flag`) chỉ báo về việc liệu thông tin dự báo có trọng số có được bao gồm trong cấu trúc cú pháp tiêu đề ảnh và trong đó quy tắc định dạng còn chỉ ra, trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát, i) giá trị của mỗi phần tử trong số các phần tử cú pháp khác với `wp_info_in_ph_flag` được suy ra là bằng hoặc 1 hoặc 0 và ii) giá trị của `wp_info_in_ph_flag` được suy ra là bằng 0 trong trường hợp mà giá trị của cờ thứ chín (`rpl_info_in_ph_flag`) chỉ báo về việc liệu thông tin danh sách ảnh tham chiếu có được bao gồm trong cấu trúc cú pháp tiêu đề ảnh là bằng 0 và được suy ra bằng hoặc 1 hoặc 0 trong trường hợp mà giá trị của `rpl_info_in_ph_flag` là bằng 1.

31. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra, trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát, việc bỏ qua tiêu đề ảnh trong cờ tiêu đề lát trong cấu trúc cú pháp tiêu đề lát trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát và suy ra rằng giá trị của tiêu đề ảnh là bằng 1.

32. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra, trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh

bao gồm một lát, việc bỏ qua thông số địa chỉ lát trong cấu trúc cú pháp tiêu đề lát.

33. Phương pháp theo điều 32, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của thông số địa chỉ lát được suy ra là bằng 0 trong trường hợp i) mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát và ii) rằng `rect_slice_flag` là bằng 0.

34. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra việc bỏ qua thông số chỉ báo về số lượng của các ô ảnh trong lát trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát.

35. Phương pháp theo điều 34, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của thông số được suy ra là bằng '`NumTilesInPic - 1`' trong trường hợp i) mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát và ii) rằng `rect_slice_flag` là bằng 0, `NumTilesInPic` chỉ báo số lượng của các ô ảnh trong ảnh.

36. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của một hoặc nhiều phần tử cú pháp là bằng 0 trong trường hợp mà `one_slice_per_pic_constraint_flag` là bằng 1.

37. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của tiêu đề ảnh trong cờ tiêu đề lát là bằng 1 trong trường hợp mà `one_slice_per_pic_constraint_flag` là bằng 1.

38. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra việc bỏ qua `rect_slice_flag` trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát.

39. Phương pháp theo điều 38, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của `rect_slice_flag` được suy ra là bằng 1 trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát.

40. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra việc bỏ qua `single_slice_per_subpic_flag` trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát.

41. Phương pháp theo điều 40, trong đó quy tắc định dạng còn chỉ ra rằng giá trị của `single_slice_per_subpic_flag` được suy ra là bằng 1 trong trường hợp

mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát.

42. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra việc bỏ qua `loop_filter_across_slices_enabled_flag` trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát.

43. Phương pháp theo điều 42, trong đó quy tắc định dạng còn chỉ ra việc bỏ qua cờ thứ mười trong dòng bit trong trường hợp mà được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát, trong đó cờ thứ mười chỉ ra việc liệu các hoạt động lọc trong vòng ngang qua các biên lát có được phép đối với các ảnh tham chiếu tới tập thông số ảnh.

44. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra việc bao gồm liệu có được chỉ báo rằng mỗi ảnh mà tham chiếu tới tập thông số ảnh bao gồm một lát trong tập thông số dãy liên kết với dãy video lớp được tạo mã.

45. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra việc bao gồm sự chỉ báo mà chỉ báo việc liệu mỗi ảnh trong dãy video lớp được tạo mã chỉ có một ô ảnh trong tập thông số dãy liên kết với dãy video lớp được tạo mã.

46. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra, trong trường hợp mà mỗi ảnh liên kết với tập thông số ảnh thứ nhất chỉ có một lát, việc bỏ qua một hoặc nhiều phần tử cú pháp liên quan phân vùng lát từ tập thông số ảnh thứ nhất.

47. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra, trong trường hợp mà mỗi ảnh liên kết với tập thông số ảnh thứ nhất chỉ có một ô ảnh, việc bỏ qua một hoặc nhiều phần tử cú pháp liên quan phân vùng lát từ tập thông số ảnh thứ nhất.

48. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 47, trong đó sự biến đổi bao gồm mã hóa video thành dòng bit.

49. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 47, trong đó sự biến đổi bao gồm giải mã video từ dòng bit.

50. Phương pháp theo điều 1 tới 47, trong đó sự biến đổi bao gồm tạo ra dòng bit từ video; và phương pháp còn bao gồm: lưu trữ dòng bit trong vật ghi máy tính đọc được bất biến.

51. Thiết bị xử lý video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều điều bất kỳ trong số các điều từ 1 tới 50.

52. Phương pháp lưu trữ dòng bit của video, bao gồm, phương pháp được nêu trong điều bất kỳ trong số các điều từ 1 tới 50, và còn bao gồm bước lưu trữ dòng bit vào vật ghi máy tính đọc được bất biến.

53. Môi trường máy tính đọc được lưu trữ mã chương trình mà, khi được thực thi, khiến bộ xử lý thực hiện phương pháp được nêu trong một hoặc nhiều điều bất kỳ trong số các điều từ 1 tới 50.

54. Môi trường máy tính đọc được mà lưu trữ sự biểu diễn được tạo mã hoặc sự biểu diễn dòng bit được tạo ra theo phương pháp bất kỳ trong số các phương pháp được mô tả trên đây.

55. Thiết bị xử lý video để lưu trữ sự biểu diễn dòng bit, trong đó thiết bị xử lý video được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều điều bất kỳ trong số các điều từ 1 tới 50.

Nhóm thứ ba của các điều mô tả các dấu hiệu và các khía cạnh đã biết của các kỹ thuật được mô tả trong phần trước (ví dụ, Mục 1.1.g trong phần 'Các kỹ thuật và các phương án ví dụ').

1. Phương pháp xử lý video (ví dụ, phương pháp 720 như được thể hiện trên FIG. 7B), bao gồm các bước: thực hiện 722 sự biến đổi giữa video bao gồm nhiều ảnh mỗi ảnh bao gồm ít nhất một lát và dòng bit của video, trong đó dòng bit tương thích với quy tắc định dạng; trong đó quy tắc định dạng chỉ ra rằng cờ thứ nhất trong dòng bit chỉ báo việc liệu cấu trúc cú pháp tiêu đề ảnh của ảnh có được bao gồm trong tiêu đề lát của lát của ảnh; và trong đó quy tắc định dạng chỉ ra rằng, đáp lại cờ bất kỳ trong số sáu cờ trong tập thông số ảnh liên kết với ảnh là bằng 1, cờ thứ nhất là bằng 0.

2. Phương pháp theo điều 1, trong đó sáu cờ chỉ báo về việc liệu các phần tử cú pháp tạo mã đã biết có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh

hoặc trực tiếp trong tiêu đề lát.

3. Phương pháp theo điều bất kỳ trong số các điều 1 và 2, trong đó sáu cờ bao gồm cờ thứ hai chỉ báo về việc liệu thông tin danh sách ảnh tham chiếu có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

4. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 3, trong đó sáu cờ bao gồm cờ thứ ba chỉ báo về việc liệu thông tin lọc giải khối có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

5. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 4, trong đó sáu cờ bao gồm cờ thứ tư chỉ báo về việc liệu thông tin lọc dịch vị thích ứng mẫu có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

6. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 5, trong đó sáu cờ bao gồm cờ thứ năm chỉ báo về việc liệu thông tin lọc vòng thích ứng có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

7. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 6, trong đó sáu cờ bao gồm cờ thứ sáu chỉ báo về việc liệu thông tin dự báo có trọng số có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

8. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 7, trong đó sáu cờ bao gồm cờ thứ bảy chỉ báo về việc liệu thông tin delta thông số lượng tử hóa có được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

9. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 8, trong đó cờ thứ nhất bằng 0 chỉ ra rằng cấu trúc cú pháp tiêu đề ảnh không hiện diện trong tiêu đề lát.

10. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 9, trong đó một trong số sáu cờ bằng 1 chỉ ra rằng thông tin được liên kết được bao gồm trong cấu trúc cú pháp tiêu đề ảnh.

11. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 10, trong đó sự biến đổi bao gồm mã hóa video thành dòng bit.

12. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 10, trong đó sự biến đổi bao gồm giải mã video từ dòng bit.

13. Phương pháp theo điều từ 1 tới 10, trong đó sự biến đổi bao gồm tạo ra dòng bit từ video, và phương pháp còn bao gồm bước: lưu trữ dòng bit trong vật ghi máy tính đọc được bất biến.

14. Thiết bị xử lý video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều điều bất kỳ trong số các điều từ 1 tới 13.

15. Phương pháp lưu trữ dòng bit của video, bao gồm, phương pháp được nêu trong điều bất kỳ trong số các điều từ 1 tới 13, và còn bao gồm bước lưu trữ dòng bit vào vật ghi máy tính đọc được bất biến.

16. Môi trường máy tính đọc được lưu trữ mã chương trình mà, khi được thực thi, khiến bộ xử lý thực hiện phương pháp được nêu trong một hoặc nhiều điều bất kỳ trong số các điều từ 1 tới 13.

17. Môi trường máy tính đọc được mà lưu trữ dòng bit được tạo ra theo phương pháp bất kỳ trong số các phương pháp được mô tả trên đây.

18. Thiết bị xử lý video để lưu trữ dòng bit, trong đó thiết bị xử lý video được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều điều bất kỳ trong số các điều từ 1 tới 13.

Nhóm thứ tư của các điều mô tả các dấu hiệu và các khía cạnh đã biết của các kỹ thuật được mô tả trong phần trước (ví dụ, Mục 2 trong phần 'Các kỹ thuật và các phương án ví dụ').

1. Phương pháp xử lý video (ví dụ, phương pháp 730 như được thể hiện trên FIG. 7C), bao gồm các bước: thực hiện 732 sự biến đổi giữa video bao gồm một hoặc nhiều ảnh, mỗi ảnh bao gồm ít nhất một lát, và dòng bit của video, trong đó dòng bit tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra việc bao gồm trường thứ nhất chỉ báo về sự hiện diện của trường thứ hai mà nhận dạng chỉ số lớn nhất đối với ảnh tham chiếu được sử dụng để tạo mã lát dựa trên điều kiện; trong đó điều kiện có thể được biểu diễn dưới dạng biểu thức logic “A & (B || C)”, mà trong đó:

A chỉ báo rằng cờ thứ nhất mà chỉ báo việc liệu thông tin danh sách ảnh tham chiếu có được bao gồm trong cấu trúc cú pháp tiêu đề ảnh là bằng 1, hoặc kiểu đơn vị lớp trừu tượng mạng (NAL) không bằng kiểu làm mới giải mã

tức thời (IDR), hoặc cờ thứ hai mà chỉ báo việc liệu thông tin danh sách ảnh tham chiếu có thể hiện diện trong các tiêu đề lát của các lát với các kiểu đơn vị NAL IDR là bằng 1,

B chỉ báo rằng kiểu lát là bằng P hoặc B, và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 0 là lớn hơn 1,

C chỉ báo kiểu lát là bằng B và số lượng các mục nhập tham chiếu trong danh sách ảnh tham chiếu 1 là lớn hơn 1, và

trong đó “&” là phép toán “và” logic và “||” là phép toán “hoặc” logic.

2. Phương pháp theo điều 1, trong đó cờ thứ nhất tương ứng với `rpl_info_in_ph_flag`.

3. Phương pháp theo điều 1, trong đó cờ thứ nhất bằng 1 chỉ ra rằng thông tin danh sách ảnh tham chiếu hiện diện trong cấu trúc cú pháp tiêu đề ảnh và không hiện diện trong các tiêu đề lát và cờ thứ nhất bằng 0 chỉ ra rằng thông tin danh sách ảnh tham chiếu không hiện diện trong cấu trúc cú pháp tiêu đề ảnh và rằng có xác suất để thông tin danh sách ảnh tham chiếu hiện diện trực tiếp trong các tiêu đề lát.

4. Phương pháp theo điều 1, trong đó cờ thứ hai tương ứng với `sps_idr_rpl_present_flag`.

5. Phương pháp theo điều 1, trong đó trường thứ nhất tương ứng với `num_ref_idx_active_override_flag`.

6. Phương pháp theo điều 1, trong đó trường thứ hai tương ứng với `num_ref_idx_active_minus1`.

7. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 6, trong đó sự biến đổi bao gồm mã hóa video thành dòng bit.

8. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 6, trong đó sự biến đổi bao gồm giải mã video từ dòng bit.

9. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 6, trong đó sự biến đổi bao gồm tạo ra dòng bit từ video; và phương pháp còn bao gồm bước: lưu trữ dòng bit trong vật ghi máy tính đọc được bất biến.

10. Thiết bị xử lý video bao gồm bộ xử lý được tạo cấu hình để thực hiện

phương pháp được nêu trong một hoặc nhiều điều bất kỳ trong số các điều từ 1 tới 9.

11. Phương pháp lưu trữ dòng bit của video, bao gồm, phương pháp được nêu trong điều bất kỳ trong số các điều từ 1 tới 9, và còn bao gồm bước lưu trữ dòng bit vào vật ghi máy tính đọc được bất biến.

12. Môi trường máy tính đọc được lưu trữ mã chương trình mà, khi được thực thi, khiến bộ xử lý thực hiện phương pháp được nêu trong một hoặc nhiều điều bất kỳ trong số các điều từ 1 tới 9.

13. Môi trường máy tính đọc được mà lưu trữ dòng bit được tạo ra theo phương pháp bất kỳ trong số các phương pháp được mô tả trên đây.

Nhóm thứ năm set của các điều mô tả các dấu hiệu và các khía cạnh đã biết của các kỹ thuật được mô tả trong phần trước (ví dụ, các Mục 2 và 3 trong phần 'Các kỹ thuật và các phương án ví dụ').

1. Phương pháp xử lý video (ví dụ, phương pháp 740 như được thể hiện trên FIG. 7D), bao gồm các bước: thực hiện sự biến đổi giữa video và dòng bit của video, trong đó dòng bit tương thích với quy tắc định dạng, và trong đó quy tắc định dạng chỉ ra điều kiện mà điều khiển việc liệu hoặc cách thức một hoặc nhiều phần tử cú pháp liên quan tới các số đếm thứ tự ảnh đối với ảnh tham chiếu dài hạn có được bao gồm trong cấu trúc cú pháp danh sách ảnh tham chiếu trong dòng bit.

2. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra rằng một hoặc nhiều phần tử cú pháp được bao gồm trong cấu trúc cú pháp danh sách ảnh tham chiếu đáp lại chỉ số của danh sách ảnh tham chiếu là nhỏ hơn số lượng của các cấu trúc cú pháp danh sách ảnh tham chiếu mà được bao gồm trong tập thông số dãy.

3. Phương pháp theo điều 2, trong đó quy tắc định dạng còn chỉ ra việc liệu có mà chỉ báo việc liệu một hoặc nhiều phần tử cú pháp có được bao gồm trong cấu trúc cú pháp danh sách ảnh tham chiếu được bao gồm trong cấu trúc cú pháp danh sách ảnh tham chiếu.

4. Phương pháp theo điều 3, trong đó còn tương ứng với `ltrp_in_header_flag`.

5. Phương pháp theo điều 4, trong đó quy tắc định dạng còn chỉ ra việc bỏ

ltp_in_header_flag ra khỏi việc được bao gồm trong cấu trúc cú pháp danh sách ảnh tham chiếu khi chỉ số của danh sách ảnh tham chiếu là bằng số lượng của các cấu trúc cú pháp danh sách ảnh tham chiếu trong tập thông số dãy.

6. Phương pháp theo điều 5, trong đó quy tắc định dạng còn chỉ ra giá trị của ltp_in_header_flag được suy ra là bằng 1 khi chỉ số của danh sách ảnh tham chiếu là bằng số lượng của các cấu trúc cú pháp danh sách ảnh tham chiếu trong tập thông số dãy.

7. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra rằng một hoặc nhiều phần tử cú pháp được bao gồm trong cấu trúc cú pháp danh sách ảnh tham chiếu đáp lại việc liệu cấu trúc cú pháp danh sách ảnh tham chiếu có được bao gồm trong tập thông số dãy của dòng bit.

8. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra rằng một hoặc nhiều phần tử cú pháp được bỏ qua trong cấu trúc cú pháp danh sách ảnh tham chiếu và rằng một hoặc nhiều phần tử cú pháp được bao gồm đối với mỗi cấu trúc cú pháp trong số cấu trúc cú pháp danh sách ảnh tham chiếu này và cấu trúc cú pháp danh sách ảnh tham chiếu khác.

9. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra rằng một hoặc nhiều phần tử cú pháp được bỏ qua trong cấu trúc cú pháp danh sách ảnh tham chiếu và rằng một hoặc nhiều phần tử cú pháp được bao gồm đối với mỗi danh sách trong số các danh sách ảnh tham chiếu.

10. Phương pháp theo điều 1, trong đó quy tắc định dạng còn chỉ ra rằng một hoặc nhiều phần tử cú pháp được bỏ qua trong cấu trúc cú pháp danh sách ảnh tham chiếu và một hoặc nhiều phần tử cú pháp được bao gồm đối với cấu trúc cú pháp danh sách ảnh tham chiếu này và cấu trúc cú pháp danh sách ảnh tham chiếu khác.

11. Phương pháp xử lý video (ví dụ, phương pháp 740 như được thể hiện trên FIG. 7D), bao gồm các bước: thực hiện 742 biến đổi giữa video và dòng bit của video, trong đó dòng bit tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra việc liệu phần tử cú pháp liên quan tới các ảnh tham chiếu trong danh sách ảnh tham chiếu có được bao gồm trong cấu trúc cú pháp danh sách ảnh tham chiếu.

12. Phương pháp theo điều 11, trong đó danh sách ảnh tham chiếu là danh sách ảnh tham chiếu 1.

13. Phương pháp theo điều 11, trong đó quy tắc định dạng phụ thuộc vào việc liệu biến số X có bằng giá trị cụ thể, trong đó X phụ thuộc vào kiểu lát của vùng video.

14. Phương pháp theo điều 11, trong đó giá trị cụ thể là bằng 1 hoặc 2.

15. Phương pháp theo điều bất kỳ trong số các điều trên đây, trong đó phần tử cú pháp bao gồm cờ chỉ báo về sự hiện diện của danh sách ảnh tham chiếu trong tập thông số dãy.

16. Phương pháp theo điều bất kỳ trong số các điều trên đây, trong đó phần tử cú pháp chỉ báo chỉ số của danh sách ảnh tham chiếu.

17. Phương pháp theo điều bất kỳ trong số các điều trên đây, trong đó phần tử cú pháp chỉ báo giá trị của các bit có giá trị thấp nhất của các số đếm thứ tự ảnh của ảnh tham chiếu dài hạn.

18. Phương pháp theo điều bất kỳ trong số các điều trên đây, trong đó phần tử cú pháp chỉ báo việc liệu hiệu số giữa các bit có giá trị cao nhất của các số đếm thứ tự ảnh của ảnh hiện tại và ảnh tham chiếu dài hạn có được bao gồm trong dòng bit.

19. Phương pháp theo điều bất kỳ trong số các điều trên đây, trong đó phần tử cú pháp chỉ báo việc liệu hiệu số giữa các bit có giá trị cao nhất của các số đếm thứ tự ảnh của ảnh hiện tại và ảnh tham chiếu dài hạn có được bao gồm trong dòng bit.

20. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 19, trong đó sự biến đổi bao gồm mã hóa video thành dòng bit.

21. Phương pháp theo điều bất kỳ trong số các điều từ 1 tới 19, trong đó sự biến đổi bao gồm giải mã video từ dòng bit.

22. Phương pháp theo điều 1 tới 19, trong đó sự biến đổi bao gồm tạo ra dòng bit từ video; và phương pháp còn bao gồm bước: lưu trữ dòng bit trong vật ghi máy tính đọc được bất biến.

23. Thiết bị xử lý video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều điều bất kỳ trong số các điều từ 1

tới 22.

24. Phương pháp lưu trữ dòng bit của video, bao gồm, phương pháp được nêu trong điều bất kỳ trong số các điều từ 1 tới 22, và còn bao gồm bước lưu trữ dòng bit vào vật ghi máy tính đọc được bất biến.

25. Môi trường máy tính đọc được lưu trữ mã chương trình mà, khi được thực thi, khiến bộ xử lý thực hiện phương pháp được nêu trong một hoặc nhiều điều bất kỳ trong số các điều từ 1 tới 22.

26. Môi trường máy tính đọc được mà lưu trữ dòng bit được tạo ra theo phương pháp bất kỳ trong số các phương pháp được mô tả trên đây.

27. Thiết bị xử lý video để lưu trữ dòng bit, trong đó thiết bị xử lý video được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều điều bất kỳ trong số các điều từ 1 tới 22.

Trong bản mô tả, thuật ngữ “xử lý video” có thể được dùng để chỉ mã hóa video, giải mã video, nén video hoặc giải nén video. Ví dụ, các thuật toán nén video có thể được áp dụng trong khi biến đổi từ biểu diễn điểm ảnh của video sang sự biểu diễn dòng bit tương ứng hoặc ngược lại. Sự biểu diễn dòng bit của khối video hiện tại có thể, ví dụ, tương ứng với các bit mà hoặc cùng nằm ở hoặc trải dài trong các vị trí khác nhau bên trong dòng bit, như được xác định bởi cú pháp. Ví dụ, khối macro có thể được mã hóa theo các giá trị dư lỗi được chuyển đổi và được tạo mã và còn bằng cách sử dụng các bit trong các tiêu đề và các trường khác trong dòng bit. Hơn nữa, trong khi biến đổi, bộ giải mã có thể phân tích dòng bit khi biết rằng một số trường có thể hiện diện, hoặc không hiện diện, dựa trên sự xác định, như được mô tả trong các giải pháp trên đây. Tương tự, bộ mã hóa có thể xác định rằng các trường cú pháp đã biết là có hoặc không được bao gồm và tạo ra sự biểu diễn được tạo mã một cách tương ứng bằng cách bao gồm hoặc loại trừ các trường cú pháp từ sự biểu diễn được tạo mã.

Các giải pháp được mô tả và các giải pháp khác, các ví dụ, các phương án, các mô đun và các hoạt động chức năng được mô tả trong bản mô tả này có thể được thực hiện trong hệ mạch điện tử kỹ thuật số, hoặc trong phần mềm, phần sụn, hoặc phần cứng máy tính, bao gồm các cấu trúc được mô tả trong bản mô tả này và các phương án tương đương về mặt cấu trúc của chúng, hoặc trong các

sự kết hợp của một hoặc nhiều trong số chúng. Các phương án được mô tả và khác nữa có thể được thực hiện dưới dạng một hoặc nhiều sản phẩm chương trình máy tính, tức là, một hoặc nhiều môđun của các lệnh chương trình máy tính được mã hóa trong môi trường máy tính đọc được để thực thi bởi, hoặc để điều khiển hoạt động của, thiết bị xử lý dữ liệu. Môi trường máy tính đọc được có thể là thiết bị lưu trữ máy đọc được, nền lưu trữ máy đọc được, thiết bị nhớ, sản phẩm thực hiện tín hiệu được lan truyền máy đọc được, hoặc sự kết hợp của một hoặc nhiều đối tượng này. Thuật ngữ “thiết bị xử lý dữ liệu” bao hàm tất cả thiết bị, các bộ phận, và các máy để xử lý dữ liệu, bao gồm ví dụ bộ xử lý lập trình được, máy tính, hoặc nhiều bộ xử lý hoặc máy tính. Thiết bị có thể bao gồm, ngoài phần cứng, mã mà tạo ra môi trường thực thi đối với chương trình máy tính đang được xem xét, ví dụ, mã mà cấu thành phần sụn bộ xử lý, ngăn xếp giao thức, hệ thống quản lý cơ sở dữ liệu, hệ điều hành, hoặc sự kết hợp của một hoặc nhiều trong số chúng. Tín hiệu được lan truyền là tín hiệu được tạo ra một cách nhân tạo, ví dụ, tín hiệu điện, quang học, hoặc điện từ được tạo ra bằng máy, mà được tạo ra để mã hóa thông tin để truyền tới thiết bị thu thích hợp.

Chương trình máy tính (còn được biết như chương trình, phần mềm, ứng dụng phần mềm, tập lệnh, hoặc mã) có thể được viết dưới dạng bất kỳ của ngôn ngữ lập trình, bao gồm các ngôn ngữ biên dịch hoặc diễn dịch, và có thể được triển khai dưới dạng bất kỳ, bao gồm dưới dạng chương trình độc lập hoặc dưới dạng môđun, thành phần, thủ tục con, hoặc đơn vị khác thích hợp để sử dụng trong môi trường máy tính. Chương trình máy tính không nhất thiết phải tương ứng với tập tin trong hệ thống tập tin. Chương trình có thể được lưu trong phần của tập tin mà giữ các chương trình khác hoặc dữ liệu (ví dụ, một hoặc nhiều tập lệnh được lưu trong tài liệu ngôn ngữ đánh dấu), trong một tập tin được dành riêng cho chương trình đang được xem xét, hoặc trong nhiều tập tin được kết hợp (ví dụ, các tập tin mà lưu trữ một hoặc nhiều môđun, các chương trình con, hoặc các phần của mã). Chương trình máy tính có thể được triển khai để được thực thi trên một máy tính hoặc trên nhiều máy tính mà nằm ở một địa điểm hoặc được phân bố trên nhiều địa điểm và được liên kết bởi mạng truyền thông.

Các quá trình và các luồng logic được mô tả trong bản mô tả này có thể được thực hiện bởi một hoặc nhiều bộ xử lý lập trình được thực thi một hoặc nhiều chương trình máy tính để thực hiện các chức năng bằng cách điều hành trên dữ liệu đầu vào và tạo ra đầu ra. các quá trình và các luồng logic này cũng có thể được thực hiện bởi, và thiết bị cũng có thể được thực hiện dưới dạng, hệ mạch logic chuyên dụng, ví dụ, mảng cổng trường lập trình được FPGA (field programmable gate array) hoặc mạch tích hợp chuyên dụng ASIC (application specific integrated circuit).

Các bộ xử lý thích hợp để thực thi chương trình máy tính bao gồm, ví dụ, cả bộ vi xử lý vạn năng lẫn bộ vi xử lý chuyên dụng, và một hoặc nhiều bộ xử lý bất kỳ của loại máy tính kỹ thuật số bất kỳ. Thông thường, bộ xử lý sẽ nhận các lệnh và dữ liệu từ bộ nhớ chỉ đọc hoặc bộ nhớ truy cập ngẫu nhiên hoặc từ cả hai. Các phần tử cơ bản của máy tính là bộ xử lý để thực hiện các lệnh và một hoặc nhiều thiết bị nhớ để lưu trữ các lệnh và dữ liệu. Thông thường, máy tính sẽ còn bao gồm, hoặc được kết nối theo cách hoạt động được để nhận dữ liệu từ hoặc chuyển dữ liệu tới, hoặc cả hai, một hoặc nhiều thiết bị lưu trữ dung lượng lớn để lưu trữ dữ liệu, ví dụ, các đĩa từ, quang từ, hoặc các đĩa quang học. Tuy nhiên, máy tính không cần phải có các thiết bị này. Môi trường máy tính được thích hợp để lưu trữ các lệnh chương trình máy tính và dữ liệu bao gồm tất cả các dạng của bộ nhớ bất biến, phương tiện và các thiết bị nhớ, bao gồm ví dụ các thiết bị nhớ bán dẫn, ví dụ, EPROM, EEPROM, và các thiết bị nhớ nhanh; các đĩa từ, ví dụ, các ổ cứng trong hoặc ổ cứng lưu động; các đĩa quang từ; và các đĩa CD ROM và DVD-ROM. Bộ xử lý và bộ nhớ có thể được bổ sung bởi, hoặc được tích hợp trong, hệ mạch logic chuyên dụng.

Trong khi bản mô tả này chứa nhiều dấu hiệu cụ thể, chúng không được coi là là các dấu hiệu giới hạn phạm vi của đối tượng bất kỳ hoặc của những gì có thể được yêu cầu bảo hộ, mà là các phần mô tả của các dấu hiệu mà có thể là cụ thể đối với các phương án cụ thể của các kỹ thuật cụ thể. Các dấu hiệu đã biết mà được mô tả trong bản mô tả này trong ngữ cảnh của các phương án riêng biệt cũng có thể được thực hiện trong sự kết hợp trong một phương án. Ngược lại, các dấu hiệu khác nhau mà được mô tả trong ngữ cảnh của một phương án cũng

có thể được thực hiện trong nhiều phương án một cách riêng biệt hoặc trong sự kết hợp phụ thích hợp bất kỳ. Hơn nữa, mặc dù các dấu hiệu có thể được mô tả trên đây là hoạt động trong các sự kết hợp đã biết và thậm chí ban đầu được yêu cầu bảo hộ như vậy, một hoặc nhiều dấu hiệu từ sự kết hợp được yêu cầu bảo hộ có thể trong một số trường hợp được loại bỏ khỏi sự kết hợp, và sự kết hợp được yêu cầu bảo hộ có thể hướng đến sự kết hợp nhỏ hoặc biến thể của sự kết hợp nhỏ.

Tương tự, trong khi các hoạt động được mô tả trên các hình vẽ theo thứ tự cụ thể, điều này không được hiểu là yêu cầu các hoạt động này được thực hiện theo thứ tự cụ thể được thể hiện hoặc theo thứ tự liên tiếp, hoặc rằng tất cả các hoạt động được minh họa được thực hiện, để đạt được các kết quả mong muốn. Hơn nữa, sự tách biệt của các thành phần hệ thống khác nhau trong các phương án được mô tả trong bản mô tả này không được hiểu là yêu cầu rằng sự tách biệt này trong tất cả các phương án.

Chỉ một vài phương án thực hiện và các ví dụ các phương án được mô tả và các phương án thực hiện, nâng cao và các biến thể khác nữa có thể được thực hiện dựa trên những gì được mô tả và được minh họa trong bản mô tả này.

YÊU CẦU BẢO HỘ

1. Phương pháp xử lý video bao gồm các bước:

thực hiện biến đổi giữa video bao gồm một hoặc nhiều ảnh và dòng bit của video;

trong đó dòng bit tuân theo quy tắc định dạng;

trong đó quy tắc định dạng xác định rằng một hoặc nhiều phần tử cú pháp chỉ báo thông tin tạo mã được bao gồm có chọn lọc trong tập hợp tham số ảnh của dòng bit đáp lại việc liệu nó có chỉ báo rằng mỗi ảnh viện dẫn đến tập hợp tham số ảnh bao gồm một lát.

2. Phương pháp theo điểm 1, trong đó một hoặc nhiều phần tử cú pháp bao gồm nhiều cờ mà chỉ báo liệu các phần tử cú pháp tạo mã cụ thể được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát, và

trong đó quy tắc định dạng còn xác định loại bỏ nhiều cờ trong trường hợp nó chỉ báo rằng mỗi ảnh viện dẫn đến tập hợp tham số ảnh bao gồm một lát, và các giá trị của nhiều cờ được suy ra bằng 0 vốn chỉ báo rằng các phần tử cú pháp tạo mã cụ thể được bao gồm trực tiếp trong tiêu đề lát.

3. Phương pháp theo điểm 2, trong đó nhiều cờ bao gồm cờ thứ nhất chỉ báo liệu thông tin danh sách ảnh tham chiếu được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

4. Phương pháp theo điểm 2 hoặc 3, trong đó nhiều cờ bao gồm cờ thứ hai chỉ báo liệu thông tin lọc tách khỏi được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

5. Phương pháp theo điểm bất kỳ trong số các điểm từ 2 đến 4, trong đó nhiều cờ bao gồm cờ thứ ba chỉ báo liệu thông tin lọc độ lệch thích ứng mẫu được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

6. Phương pháp theo điểm bất kỳ trong số các điểm từ 2 đến 5, trong đó nhiều cờ bao gồm cờ thứ tư chỉ báo liệu thông tin lọc vòng thích ứng được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

7. Phương pháp theo điểm bất kỳ trong số các điểm từ 2 đến 6, trong đó nhiều cờ bao gồm cờ thứ năm chỉ báo liệu thông tin dự báo có trọng số được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

8. Phương pháp theo điểm bất kỳ trong số các điểm từ 2 đến 7, trong đó nhiều cờ bao gồm cờ thứ sáu chỉ báo liệu thông tin delta tham số lượng tử hóa được bao gồm hoặc trong cấu trúc cú pháp tiêu đề ảnh hoặc trực tiếp trong tiêu đề lát.

9. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 8, trong đó một hoặc nhiều phần tử cú pháp bao gồm cờ thứ bảy mà chỉ báo liệu các hoạt động lọc trong vòng giữa các đường biên lát được kích hoạt cho các ảnh viện dẫn đến tập hợp tham số ảnh.

10. Phương pháp theo điểm 9, trong đó quy tắc định dạng còn xác định loại bỏ cờ thứ bảy trong trường hợp nó chỉ báo rằng mỗi ảnh viện dẫn đến tập hợp tham số ảnh bao gồm một lát, và giá trị của cờ thứ bảy được suy ra bằng 0 vốn chỉ báo rằng các hoạt động lọc trong vòng giữa các đường biên lát bị vô hiệu hóa cho tập hợp tham số ảnh.

11. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 10, trong đó quy tắc định dạng còn xác định để bao gồm chỉ báo thực hiện chỉ báo liệu mỗi ảnh viện dẫn đến tập hợp tham số ảnh có một lát trong tập hợp tham số ảnh.

12. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 11, trong đó phép biến đổi bao gồm bước mã hóa video thành dòng bit.

13. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 11, trong đó phép biến đổi bao gồm bước giải mã video từ dòng bit.

14. Thiết bị xử lý dữ liệu video bao gồm bộ xử lý và bộ nhớ bất biến có các lệnh trên đó, trong đó các lệnh khi thực thi bởi bộ xử lý, khiến bộ xử lý:

thực hiện biến đổi giữa video bao gồm một hoặc nhiều ảnh và dòng bit của video;

trong đó dòng bit tuân theo quy tắc định dạng;

trong đó quy tắc định dạng xác định rằng một hoặc nhiều phần tử cú pháp chỉ báo thông tin tạo mã được bao gồm có chọn lọc trong tập hợp tham số ảnh của dòng bit đáp lại việc liệu nó có chỉ báo rằng mỗi ảnh viện dẫn đến tập hợp tham số ảnh bao gồm một lát.

15. Vật ghi máy tính đọc được bất biến lưu trữ các lệnh khiến bộ xử lý:

thực hiện biến đổi giữa video bao gồm một hoặc nhiều ảnh và dòng bit của video;

trong đó dòng bit tuân theo quy tắc định dạng;

trong đó quy tắc định dạng xác định rằng một hoặc nhiều phần tử cú pháp chỉ báo thông tin tạo mã được bao gồm có chọn lọc trong tập hợp tham số ảnh của dòng bit đáp lại việc liệu nó có chỉ báo rằng mỗi ảnh viện dẫn đến tập hợp tham số ảnh bao gồm một lát.

16. Phương pháp lưu trữ dòng bit của video bao gồm các bước:

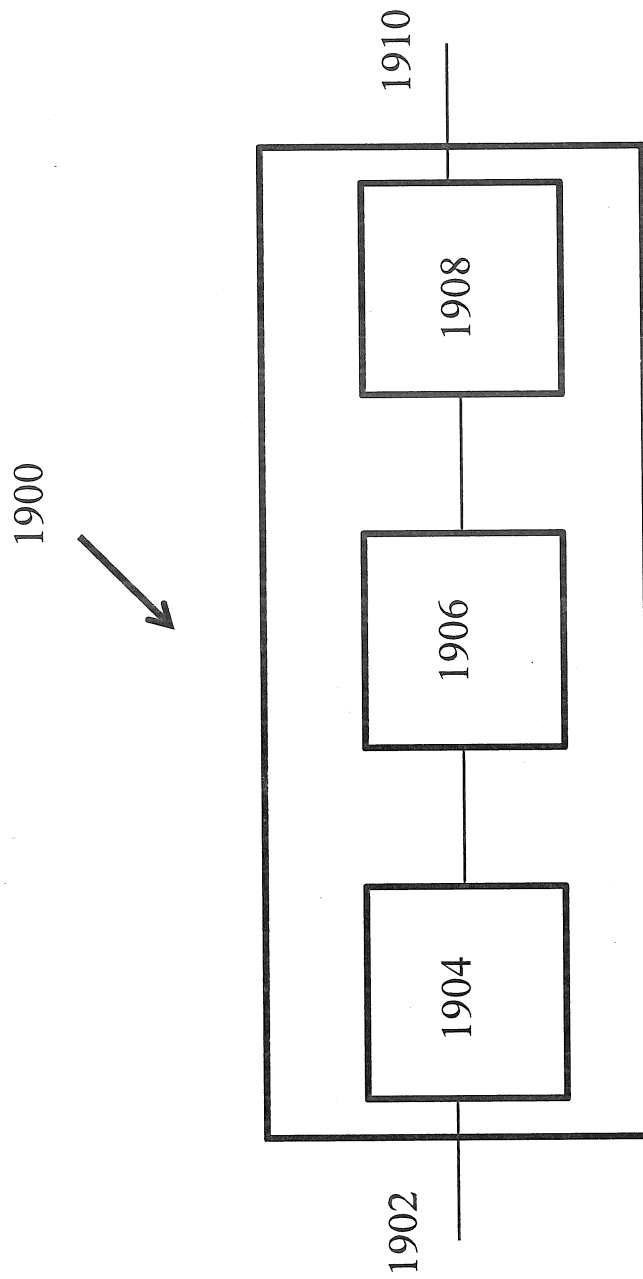
tạo dòng bit của video bao gồm một hoặc nhiều ảnh; và

lưu trữ dòng bit trong vật ghi máy tính đọc được bất biến,

trong đó dòng bit tuân theo quy tắc định dạng;

trong đó quy tắc định dạng xác định rằng một hoặc nhiều phần tử cú pháp chỉ báo thông tin tạo mã được bao gồm có chọn lọc trong tập hợp tham số ảnh của dòng bit đáp lại việc liệu nó có chỉ báo rằng mỗi ảnh viện dẫn đến tập hợp tham số ảnh bao gồm một lát.

1/10

**FIG. 1**

2/10

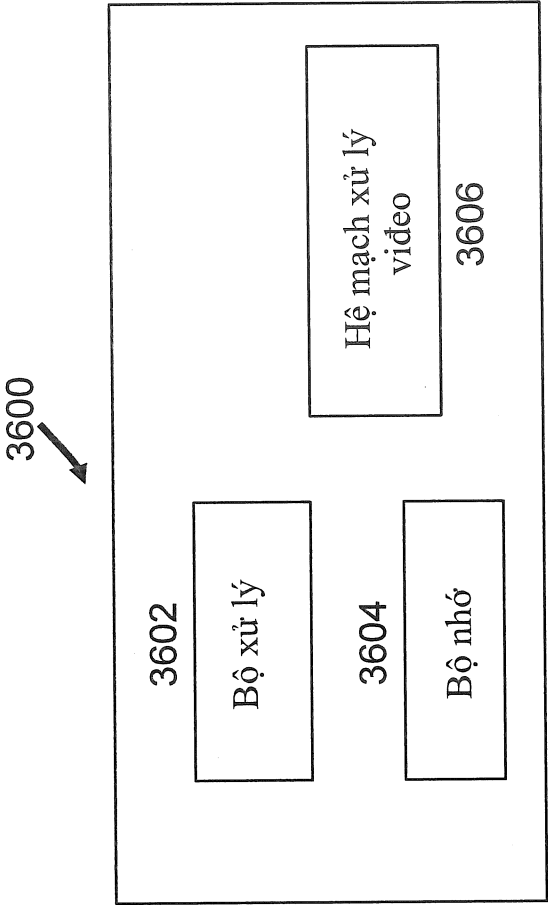
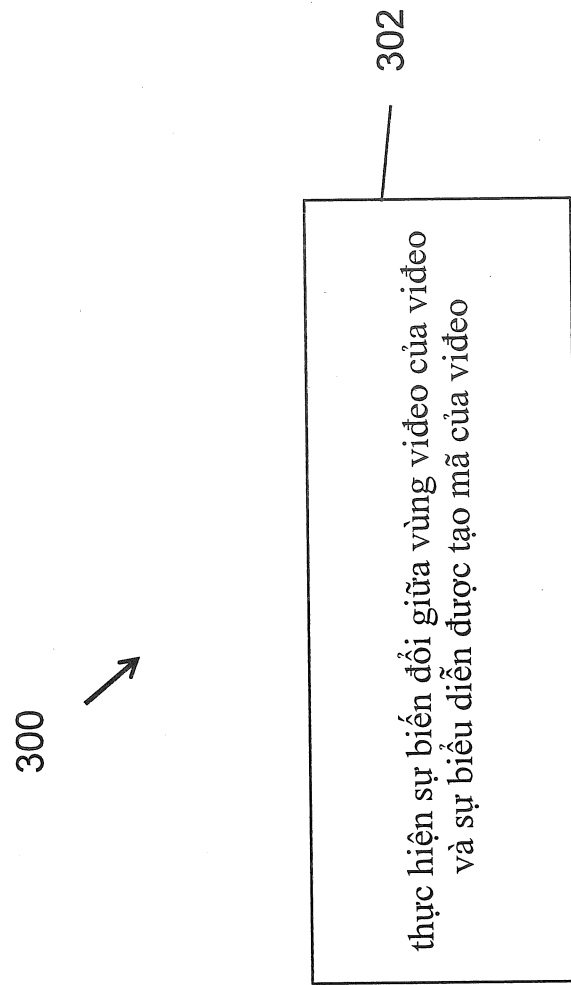


FIG. 2

3/10

**FIG. 3**

4/10

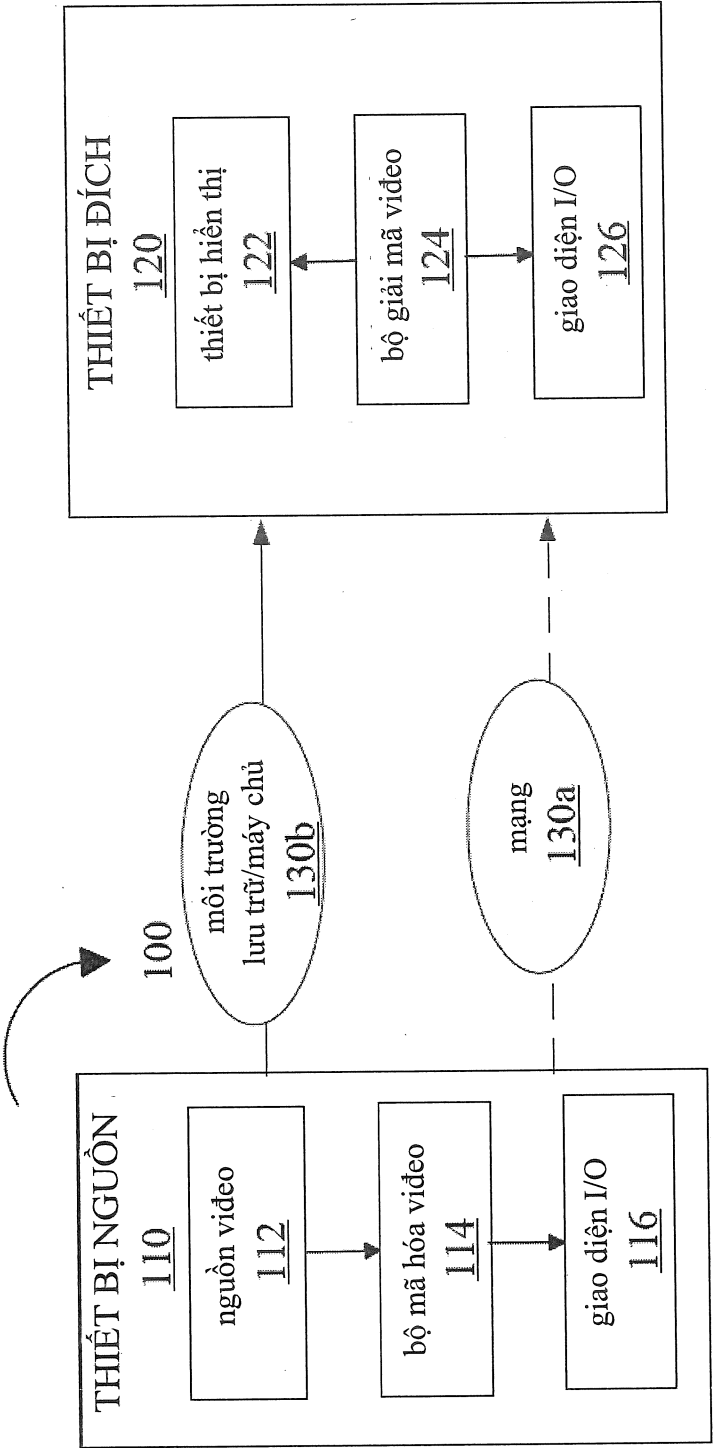


FIG. 4

5/10

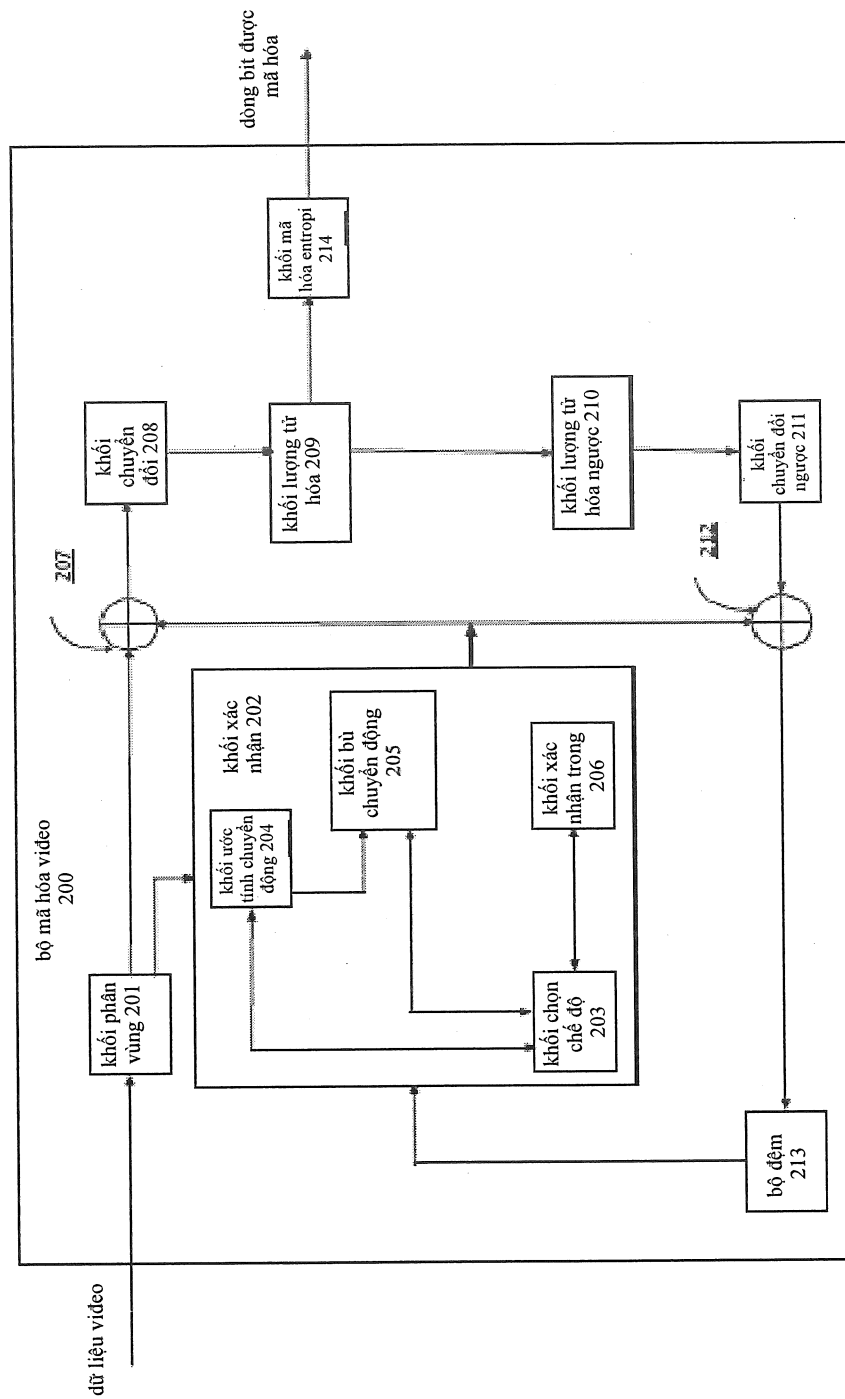


FIG. 5

6/10

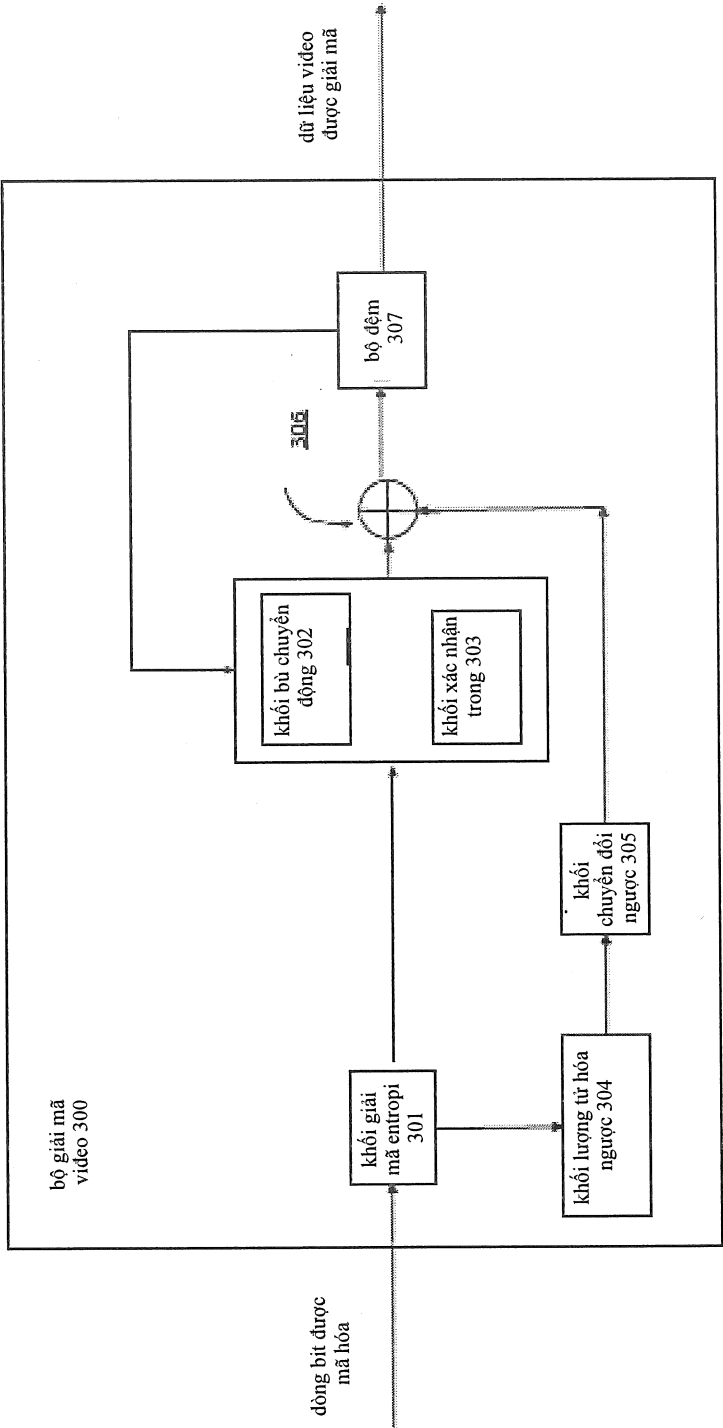
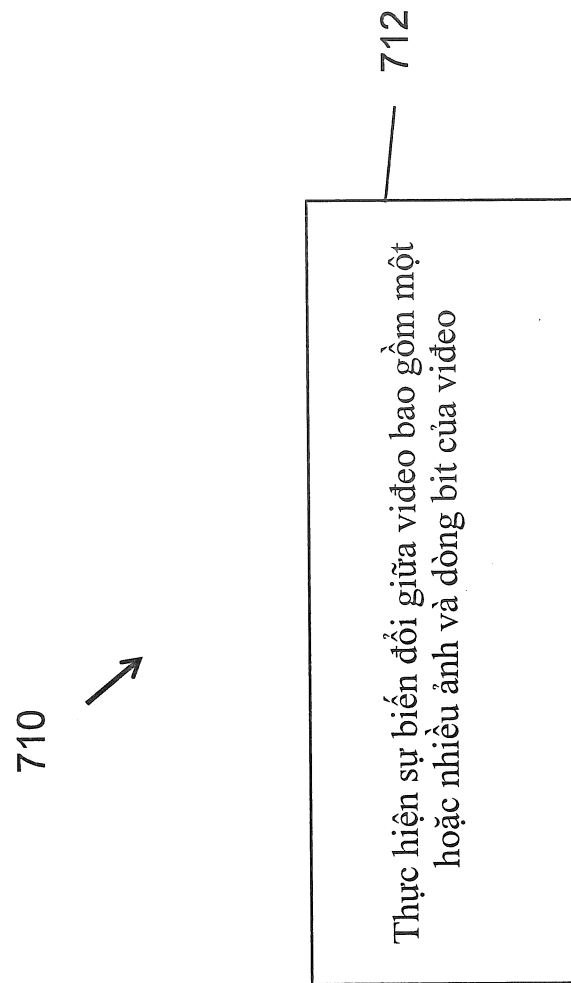
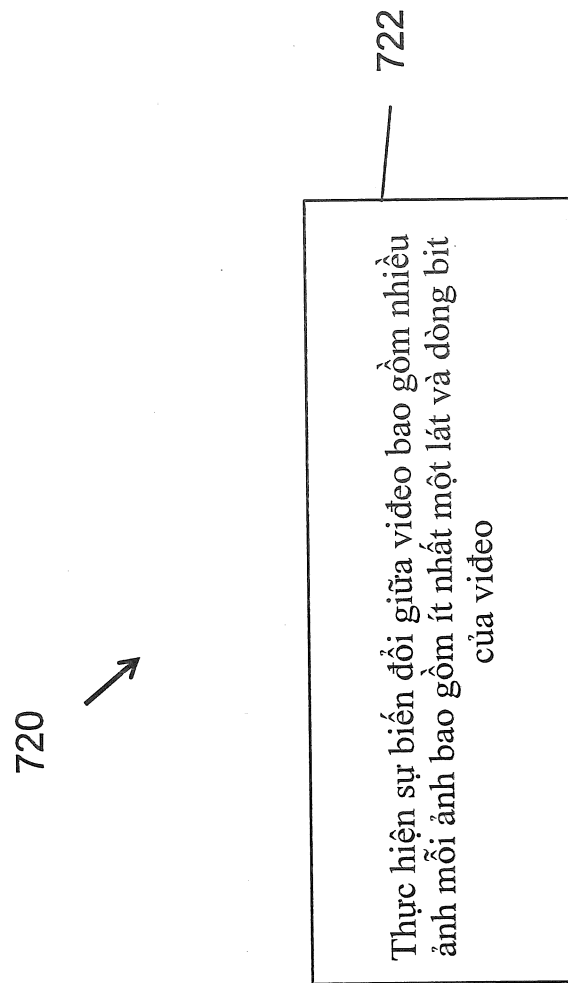


FIG. 6

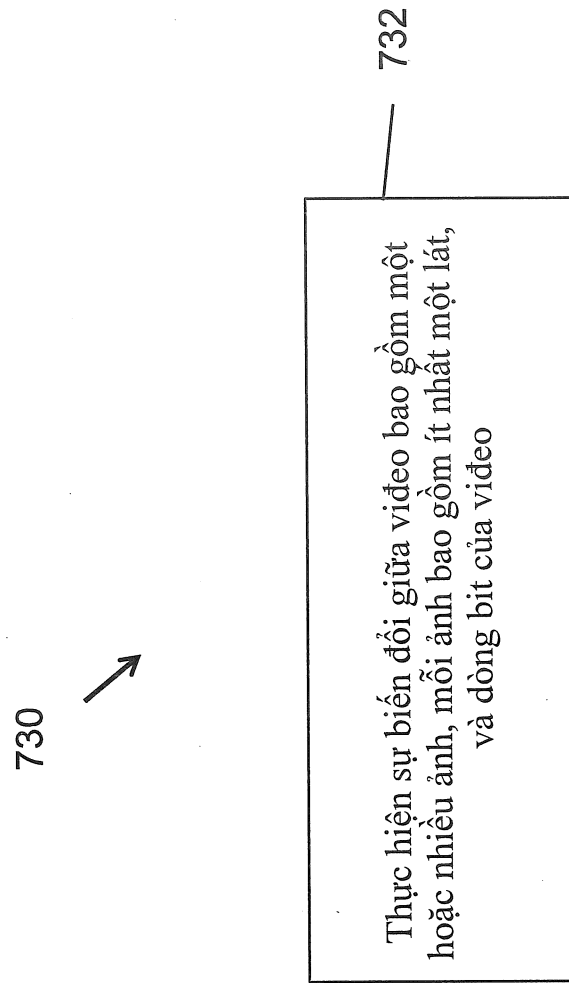
7/10

**FIG. 7A**

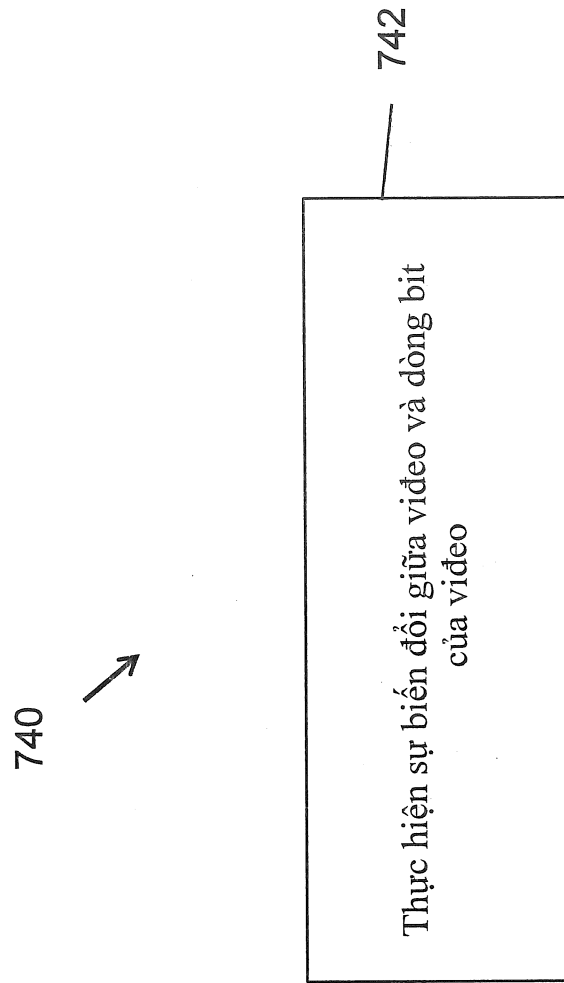
8/10

**FIG. 7B**

9/10

**FIG. 7C**

10/10

**FIG. 7D**