



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẢNG ĐỘC QUYỀN SÁNG CHẾ

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN)
CỤC SỞ HỮU TRÍ TUỆ

(11)



1-0052228

(51)^{2020.01} H04N 13/00

(13) B

(21) 1-2021-01380

(22) 16/08/2019

(86) PCT/US2019/046934 16/08/2019

(87) WO2020/037278 20/02/2020

(30) 62/719,360 17/08/2018 US

(45) 27/10/2025 451

(43) 25/05/2021 398A

(73) HUAWEI TECHNOLOGIES CO., LTD. (CN)

Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong
518129, China

(72) HENDRY, Fnu (ID); WANG, Ye-Kui (US).

(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP GIẢI MÃ DÒNG BIT VIDEO ĐƯỢC TẠO MÃ, THIẾT BỊ GIẢI
MÃ, MÁY TẠO MÃ VÀ HỆ THỐNG TẠO MÃ

(21) 1-2021-01380

(57) Sáng chế đề cập đến phương pháp giải mã dòng bit video được tạo mã bao gồm các bước: thu nhận cấu trúc danh sách hình ảnh tham chiếu cho lát hiện thời được thể hiện trong dòng bit video được tạo mã, trong đó cấu trúc danh sách hình ảnh tham chiếu chứa số lượng mục nhập; thu nhận số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu cho lát hiện thời; xây dựng danh sách hình ảnh tham chiếu cho lát hiện thời, danh sách hình ảnh tham chiếu chứa số lượng mục nhập hoạt động và không hoạt động; thiết đặt số lượng mục nhập hoạt động trong danh sách hình ảnh tham chiếu bằng số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu khi số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu lớn hơn số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu; và thu nhận, dựa vào ít nhất một mục nhập hoạt động của danh sách hình ảnh tham chiếu, ít nhất một khối được xây dựng lại của lát hiện thời. Sáng chế cũng đề cập đến phương pháp mã hóa dòng bit video được tạo mã, thiết bị mã hóa, thiết bị giải mã và phương tiện tạo mã.

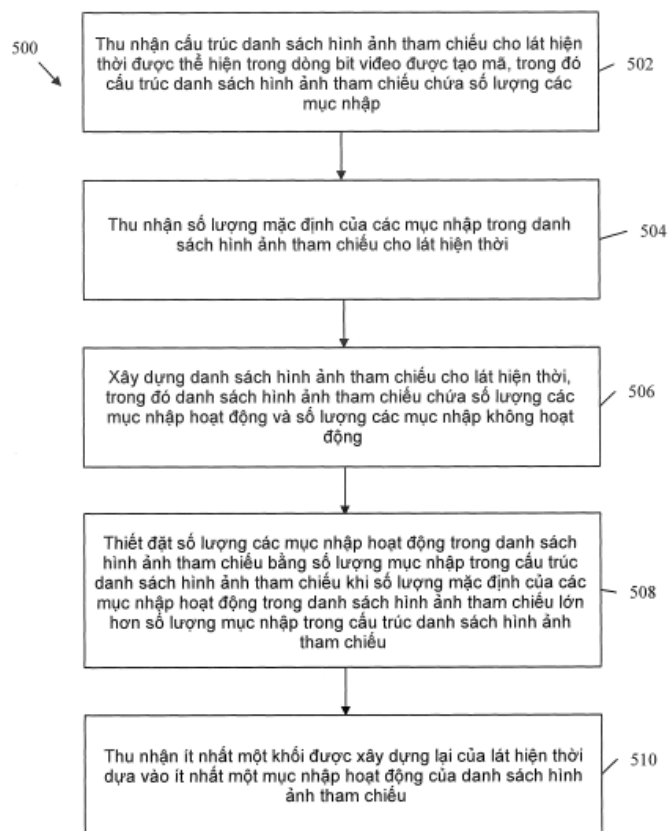


FIG. 5

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến các kỹ thuật để quản lý hình ảnh tham chiếu trong quy trình tạo mã video. Cụ thể hơn là, sáng chế mô tả các kỹ thuật để xây dựng các danh sách hình ảnh tham chiếu và đánh dấu hình ảnh tham chiếu.

Tình trạng kỹ thuật của sáng chế

Lượng dữ liệu video cần thiết để mô tả ngay cả một video tương đối ngắn cũng có thể là đáng kể, điều này có thể dẫn tới các khó khăn khi dữ liệu cần được tạo dòng hoặc được truyền thông theo cách khác qua mạng truyền thông có dung lượng băng thông giới hạn. Do đó, dữ liệu video thường được nén trước khi được truyền thông qua các mạng viễn thông ngày nay. Kích thước của video cũng có thể là vấn đề khi video được lưu trữ trên thiết bị lưu trữ bởi vì các tài nguyên bộ nhớ có thể bị giới hạn. Các thiết bị nén video thường sử dụng phần mềm và/hoặc phần cứng ở nguồn để tạo mã dữ liệu video trước khi truyền hoặc lưu trữ, nhờ đó làm giảm số lượng dữ liệu cần để thể hiện các ảnh video số. Dữ liệu được nén sau đó được thu ở đích bởi thiết bị giải nén video mà giải mã dữ liệu video. Với các tài nguyên mạng giới hạn và nhu cầu ngày càng tăng cao về chất lượng video cao hơn, có mong muốn về các kỹ thuật nén và giải nén nâng cao để nâng cao tỷ lệ nén mà ít hoặc không hy sinh chất lượng ảnh.

Bản chất kỹ thuật của sáng chế

Khía cạnh thứ nhất đề cập đến phương pháp giải mã dòng bit video được tạo mã. Phương pháp này bao gồm các bước: thu nhận cấu trúc danh sách hình ảnh tham chiếu cho lát hiện thời được thể hiện trong dòng bit video được tạo mã, trong đó cấu trúc danh sách hình ảnh tham chiếu chứa số lượng mục nhập; thu nhận số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu cho lát hiện thời; xây dựng danh sách hình ảnh tham chiếu cho lát hiện thời, trong đó danh sách hình ảnh tham chiếu chứa số lượng mục nhập hoạt động và số lượng mục nhập không hoạt động; thiết đặt số lượng mục nhập hoạt

động trong danh sách hình ảnh tham chiếu bằng số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu khi số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu lớn hơn số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu; và thu nhận, dựa vào ít nhất một mục nhập hoạt động của danh sách hình ảnh tham chiếu, ít nhất một khối được xây dựng lại của lát hiện thời.

Phương pháp này đưa ra các kỹ thuật làm đơn giản hóa và làm hiệu quả hơn việc báo hiệu các danh sách hình ảnh tham chiếu. Do đó, quy trình tạo mã tổng thể được nâng cao.

Theo dạng thực hiện thứ nhất của phương pháp theo khía cạnh thứ nhất như vậy, thứ tự của các mục nhập trong cấu trúc danh sách hình ảnh tham chiếu là giống như thứ tự của các hình ảnh tham chiếu tương ứng trong danh sách hình ảnh tham chiếu.

Theo dạng thực hiện thứ hai của phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, thứ tự là từ không đến trị số được chỉ báo.

Theo dạng thực hiện thứ ba của phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, trị số được chỉ báo là từ không đến trị số được chỉ báo bởi `sps_max_dec_pic_buffering_minus1`.

Theo dạng thực hiện thứ tư của phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, danh sách hình ảnh tham chiếu được định rõ là `RefPicList[0]`.

Theo dạng thực hiện thứ năm của phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, danh sách hình ảnh tham chiếu được định rõ là `RefPicList[1]`.

Theo dạng thực hiện thứ sáu của phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, ít nhất một khối được xây dựng lại được sử dụng để tạo ra ảnh được hiển thị trên bộ hiển thị của thiết bị điện tử.

Theo dạng thực hiện thứ bảy của phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, danh sách hình ảnh tham chiếu bao gồm danh sách của các hình ảnh tham chiếu được sử dụng cho việc dự đoán liên ảnh.

Theo dạng thực hiện thứ tám của phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, việc dự đoán liên ảnh là cho lát P hoặc cho lát B.

Theo dạng thực hiện thứ chín của phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, phần đầu lát chứa cờ tập hợp thông số chuỗi (Sequence Parameter Set, viết tắt là SPS) danh sách hình ảnh tham chiếu được định rõ là `ref_pic_list_sps_flag[ i ]`.

Theo dạng thực hiện thứ mười của phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, phần đầu lát chứa cờ ghi đè chủ động chỉ số tham chiếu số được định rõ bởi `num_ref_idx_active_override_flag`.

Theo dạng thực hiện thứ mười một của phương pháp theo khía cạnh thứ nhất hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, danh sách hình ảnh tham chiếu được định rõ là `RefPictList[0]` hoặc `RefPictList[1]`, và thứ tự của các mục nhập trong cấu trúc danh sách hình ảnh tham chiếu là giống như thứ tự của các hình ảnh tham chiếu tương ứng trong danh sách hình ảnh tham chiếu.

Khía cạnh thứ hai đề cập đến thiết bị giải mã mà bao gồm bộ thu được tạo cấu hình để thu dòng bit video được tạo mã; bộ nhớ được ghép nối với bộ thu, bộ nhớ lưu trữ các lệnh; và bộ xử lý được ghép nối với bộ nhớ, bộ xử lý được tạo cấu hình để thực hiện các lệnh được lưu trữ trong bộ nhớ để khiến bộ xử lý: thu nhận cấu trúc danh sách hình ảnh tham chiếu cho lát hiện thời được thể hiện trong dòng bit video được tạo mã, trong đó cấu trúc danh sách hình ảnh tham chiếu chứa số lượng mục nhập; thu nhận số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu cho lát hiện thời; xây dựng danh sách hình ảnh tham chiếu cho lát hiện thời, trong đó danh sách hình ảnh tham chiếu chứa số lượng mục nhập hoạt động và số lượng mục nhập không hoạt động; thiết đặt số lượng mục nhập hoạt động trong danh sách hình ảnh tham chiếu bằng số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu khi số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu lớn hơn số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu; và thu nhận, dựa vào ít nhất một mục nhập hoạt động của danh sách hình ảnh tham chiếu, ít nhất một khối được xây dựng lại của lát hiện thời.

Thiết bị giải mã đưa ra các kỹ thuật đơn giản hóa và làm hiệu quả hơn việc báo hiệu các danh sách hình ảnh tham chiếu. Do đó, quy trình tạo mã tổng thể được nâng cao.

Theo dạng thực hiện thứ nhất của thiết bị giải mã theo khía cạnh thứ hai như vậy, thiết bị giải mã còn bao gồm bộ hiển thị được tạo cấu hình để hiển thị ảnh dựa vào ít nhất một khối

được xây dựng lại.

Khía cạnh thứ ba đề cập đến máy tạo mã mà bao gồm bộ thu được tạo cấu hình để thu dòng bit để giải mã; bộ truyền được ghép nối với bộ thu, bộ truyền được tạo cấu hình để truyền ảnh được giải mã đến bộ hiển thị; bộ nhớ được ghép nối với ít nhất một trong số bộ thu hoặc bộ truyền, bộ nhớ được tạo cấu hình để lưu trữ các lệnh; và bộ xử lý được ghép nối với bộ nhớ, bộ xử lý được tạo cấu hình để thực hiện các lệnh được lưu trữ trong bộ nhớ để thực hiện phương pháp theo bất kỳ trong số các khía cạnh hoặc các cách thực hiện nêu trên.

Khía cạnh thứ tư đề cập đến hệ thống mà bao gồm bộ mã hóa và bộ giải mã truyền thông với bộ mã hóa. Bộ mã hóa hoặc bộ giải mã bao gồm thiết bị giải mã hoặc máy tạo mã theo bất kỳ trong số các khía cạnh hoặc các cách thực hiện nêu trên.

Hệ thống đưa ra các kỹ thuật đơn giản hóa và làm hiệu quả hơn việc báo hiệu các danh sách hình ảnh tham chiếu. Do đó, quy trình tạo mã tổng thể được nâng cao.

Khía cạnh thứ năm đề cập đến phương tiện mã hóa bao gồm phương tiện thu được tạo cấu hình để thu ảnh để mã hóa hoặc để thu dòng bit để giải mã, phương tiện truyền được ghép nối với phương tiện thu, phương tiện truyền được tạo cấu hình để truyền dòng bit đến bộ giải mã hoặc để truyền hình ảnh được giải mã đến phương tiện hiển thị, phương tiện lưu trữ được ghép nối với ít nhất một trong số phương tiện thu hoặc phương tiện truyền, phương tiện lưu trữ được tạo cấu hình để lưu trữ các lệnh, và phương tiện xử lý được ghép nối với phương tiện lưu trữ, phương tiện xử lý được tạo cấu hình để thực hiện các lệnh được lưu trữ trong phương tiện lưu trữ để thực hiện các phương pháp theo khía cạnh hoặc dạng thực hiện bất kỳ trong số các khía cạnh hoặc các dạng thực hiện nêu trên.

Phương tiện tạo mã đưa ra các kỹ thuật đơn giản hóa và làm hiệu quả hơn việc báo hiệu các danh sách hình ảnh tham chiếu. Do đó, quy trình tạo mã tổng thể được nâng cao.

Mô tả vắn tắt các hình vẽ

Để hiểu sáng chế đầy đủ hơn, sự tham chiếu được thực hiện cho phần mô tả vắn tắt dưới đây, liên quan đến các hình vẽ kèm theo và phần mô tả chi tiết, trong đó các số chỉ dẫn giống nhau thể hiện các bộ phận giống nhau.

Fig.1 là sơ đồ khối minh họa hệ thống tạo mã ví dụ mà có thể ứng dụng các kỹ thuật dự đoán hai chiều.

Fig.2 là sơ đồ khối minh họa ví dụ về bộ mã hóa video mà có thể thực hiện các kỹ

thuật dự đoán hai chiều.

Fig.3 là sơ đồ khối minh họa ví dụ về bộ giải mã video mà có thể thực hiện các kỹ thuật dự đoán hai chiều.

Fig.4 là sơ đồ giản lược minh họa tập hợp hình ảnh tham chiếu (Reference Picture Set, viết tắt là RPS) có hình ảnh với các mục nhập trong tất cả các tập con của RPS.

Fig.5 là phương án về phương pháp giải mã dòng bit video được tạo mã.

Fig.6 là sơ đồ giản lược của thiết bị tạo mã video.

Fig.7 là sơ đồ giản lược của phương án về phương tiện tạo mã.

Mô tả chi tiết sáng chế

Fig.1 là sơ đồ khối minh họa hệ thống tạo mã ví dụ 10 mà có thể ứng dụng các kỹ thuật tạo mã video như được mô tả ở đây. Như được thể hiện trên Fig.1, hệ thống tạo mã 10 bao gồm thiết bị nguồn 12 mà cung cấp dữ liệu video được mã hóa để được giải mã ở thời gian sau đó bởi thiết bị đích 14. Cụ thể là, thiết bị nguồn 12 có thể cung cấp dữ liệu video đến thiết bị đích 14 qua phương tiện đọc được bởi máy tính 16. Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm bất kỳ trong số nhiều thiết bị, bao gồm các máy tính bàn, các máy tính cầm tay (ví dụ, máy tính xách tay), các máy tính bảng, các thiết bị giải mã tín hiệu truyền hình, các ống nghe điện thoại chẳng hạn như các điện thoại "thông minh", các bảng "thông minh", các ti vi, các camera, các thiết bị hiển thị, các máy phát đa phương tiện số, các thiết bị chơi trò chơi video, thiết bị tạo dòng video, hoặc tương tự. Trong một số trường hợp, thiết bị nguồn 12 và thiết bị đích 14 có thể được trang bị cho truyền thông không dây.

Thiết bị đích 14 có thể thu dữ liệu video được mã hóa để được giải mã qua phương tiện đọc được bởi máy tính 16. Phương tiện đọc được bởi máy tính 16 có thể bao gồm loại bất kỳ trong số phương tiện hoặc thiết bị có khả năng di chuyển dữ liệu video được mã hóa từ thiết bị nguồn 12 đến thiết bị đích 14. Theo một ví dụ, phương tiện đọc được bởi máy tính 16 có thể bao gồm phương tiện truyền thông để cho phép thiết bị nguồn 12 truyền dữ liệu video được mã hóa trực tiếp đến thiết bị đích 14 theo thời gian thực. Dữ liệu video được mã hóa có thể được điều biến theo chuẩn truyền thông, chẳng hạn như giao thức truyền thông không dây, và được truyền đến thiết bị đích 14. Phương tiện truyền thông có thể bao gồm phương tiện truyền thông không dây hoặc nối dây bất kỳ, chẳng hạn như phổ tần số radio (Radio Frequency, viết tắt là RF) hoặc một hoặc nhiều đường truyền vật lý. Phương tiện

truyền thông có thể tạo nên một phần của mạng dựa trên gói, chẳng hạn như mạng vùng cục bộ, mạng diện rộng, hoặc mạng toàn cầu chẳng hạn như Internet. Phương tiện truyền thông có thể bao gồm các bộ định tuyến, chuyển mạch, các trạm gốc, hoặc thiết bị khác bất kỳ mà có thể hữu ích cho việc tạo thuận lợi cho truyền thông từ thiết bị nguồn 12 đến thiết bị đích 14.

Trong một vài ví dụ, dữ liệu được mã hóa có thể được đưa ra từ giao diện đầu ra 22 đến thiết bị lưu trữ. Tương tự, dữ liệu được mã hóa có thể được truy cập từ thiết bị lưu trữ bởi giao diện đầu vào. Thiết bị lưu trữ có thể bao gồm bất kỳ trong số nhiều phương tiện lưu trữ dữ liệu được phân bổ hoặc được truy cập cục bộ chẳng hạn như ổ đĩa cứng, các đĩa Blu-ray, các đĩa video số (Digital Video Disk, viết tắt là DVD), các bộ nhớ chỉ đọc đĩa compac (Compact Disc Read-Only Memories, viết tắt là CD-ROMs), bộ nhớ chớp, bộ nhớ khả biến hoặc bộ nhớ bất khả biến, hoặc phương tiện lưu trữ số thích hợp khác bất kỳ để lưu trữ dữ liệu video được mã hóa. Trong ví dụ khác, thiết bị lưu trữ có thể tương ứng với máy chủ tệp hoặc thiết bị lưu trữ trung gian khác mà có thể lưu trữ video được mã hóa được tạo ra bởi thiết bị nguồn 12. Thiết bị đích 14 có thể truy cập dữ liệu video được lưu trữ từ thiết bị lưu trữ qua việc tạo dòng hoặc tải xuống. Máy chủ tệp có thể là loại bất kỳ trong số máy chủ có khả năng lưu trữ dữ liệu video được mã hóa và truyền dữ liệu video được mã hóa đó đến thiết bị đích 14. Các máy chủ tệp ví dụ bao gồm máy chủ web (ví dụ, dùng cho trang web (website)), máy chủ giao thức truyền tải tệp (File Transfer Protocol, viết tắt là FTP), các thiết bị lưu trữ nối mạng (Network Attached Storage, viết tắt là NAS), hoặc ổ đĩa cục bộ. Thiết bị đích 14 có thể truy cập dữ liệu video được mã hóa thông qua sự kết nối dữ liệu tiêu chuẩn bất kỳ, bao gồm sự kết nối internet. Điều này có thể bao gồm kênh không dây (ví dụ, sự kết nối Wi-Fi), sự kết nối nối dây (ví dụ, đường thuê bao số (Digital Subscriber Line, viết tắt là DSL), modem cáp, v.v..), hoặc sự kết hợp của cả hai mà phù hợp cho sự truy cập dữ liệu video được mã hóa được lưu trữ trên máy chủ tệp. Việc truyền dữ liệu video được mã hóa từ thiết bị lưu trữ có thể là việc truyền tạo dòng, việc truyền tải xuống, hoặc sự kết hợp của chúng.

Các kỹ thuật của sáng chế không nhất thiết giới hạn ở các ứng dụng hoặc các cài đặt không dây. Các kỹ thuật có thể được áp dụng để tạo mã video trong việc hỗ trợ bất kỳ trong số nhiều ứng dụng đa phương tiện, chẳng hạn như phát rộng truyền hình vô tuyến, truyền hình cáp, truyền hình vệ tinh, truyền video tạo dòng trên Internet, chẳng hạn như tạo dòng

thích ứng động trên HTTP (dynamic adaptive streaming over HTTP, viết tắt là DASH), video số được mã hóa trên phương tiện lưu trữ dữ liệu, giải mã video số được lưu trữ trên phương tiện lưu trữ dữ liệu, hoặc các ứng dụng khác. Trong một vài ví dụ, hệ thống tạo mã 10 có thể được tạo cấu hình để hỗ trợ việc truyền video một chiều hoặc hai chiều để hỗ trợ các ứng dụng chẳng hạn như tạo dòng video, phát lại video, phát rộng video, và/hoặc điện thoại video.

Trong ví dụ trên Fig.1, thiết bị nguồn 12 bao gồm nguồn video 18, bộ mã hóa video 20, và giao diện đầu ra 22. Thiết bị đích 14 bao gồm giao diện đầu vào 28, bộ giải mã video 30, và thiết bị hiển thị 32. Theo sáng chế, bộ mã hóa video 20 của thiết bị nguồn 12 và/hoặc bộ giải mã video 30 của thiết bị đích 14 có thể được tạo cấu hình để áp dụng các kỹ thuật để tạo mã video. Trong các ví dụ khác, thiết bị nguồn và thiết bị đích có thể bao gồm các thành phần hoặc các sự bố trí khác. Ví dụ, thiết bị nguồn 12 có thể thu dữ liệu video từ nguồn video bên ngoài, chẳng hạn như camera bên ngoài. Tương tự, thiết bị đích 14 có thể giao tiếp với thiết bị hiển thị bên ngoài, thay vì bao gồm thiết bị hiển thị tích hợp.

Hệ thống tạo mã 10 được minh họa trên Fig.1 chỉ là một ví dụ. Các kỹ thuật để tạo mã video có thể được thực hiện bởi thiết bị mã hóa và/hoặc giải mã video số bất kỳ. Mặc dù các kỹ thuật của sáng chế thường được thực hiện bởi thiết bị tạo mã video, nhưng các kỹ thuật này có thể cũng được thực hiện bởi bộ mã hóa/bộ giải mã video, thường được đề cập đến là “codecs” (CODEC). Ngoài ra, các kỹ thuật của sáng chế có thể cũng được thực hiện bởi bộ tiền xử lý video. Bộ mã hóa và/hoặc bộ giải mã video có thể là bộ xử lý đồ họa (Graphics Processing Unit, viết tắt là GPU) hoặc thiết bị tương tự.

Thiết bị nguồn 12 và thiết bị đích 14 chỉ là các ví dụ về các thiết bị tạo mã như vậy trong đó thiết bị nguồn 12 tạo ra dữ liệu video được tạo mã để truyền đến thiết bị đích 14. Trong một vài ví dụ, thiết bị nguồn 12 và thiết bị đích 14 có thể hoạt động theo cách về cơ bản đối xứng sao cho mỗi trong số các thiết bị nguồn và đích 12, 14 bao gồm các thành phần mã hóa và giải mã video. Do đó, hệ thống tạo mã 10 có thể hỗ trợ việc truyền video một chiều hoặc hai chiều giữa các thiết bị video 12, 14, ví dụ, để tạo dòng video, phát lại video, phát rộng video, hoặc điện thoại video.

Nguồn video 18 của thiết bị nguồn 12 có thể bao gồm thiết bị quay video, chẳng hạn như máy quay video, phương tiện lưu trữ video chứa video được quay trước đó, và/hoặc giao diện cấp video để thu video từ nhà cung cấp nội dung video. Như một sự thay thế khác, nguồn

video 18 có thể tạo ra dữ liệu dựa vào đồ họa máy tính làm video nguồn, hoặc sự kết hợp của video trực tiếp, video được lưu trữ, và video được tạo ra bởi máy tính.

Trong một số trường hợp, khi nguồn video 18 là máy quay video, thiết bị nguồn 12 và thiết bị đích 14 có thể tạo nên các điện thoại camera hoặc các điện thoại video. Tuy nhiên, như được nêu trên, các kỹ thuật được mô tả trong sáng chế có thể áp dụng được cho việc tạo mã video, và có thể được áp dụng cho các ứng dụng không dây hoặc nối dây. Trong mỗi trường hợp, video được quay, được quay trước hoặc video được tạo ra bởi máy tính có thể được mã hóa bởi bộ mã hóa video 20. Sau đó, thông tin video được mã hóa có thể được đưa ra bởi giao diện đầu ra 22 trên phương tiện đọc được bởi máy tính 16.

Phương tiện đọc được bởi máy tính 16 có thể bao gồm phương tiện nhất thời, chẳng hạn như phương tiện phát rộng không dây hoặc truyền mạng nối dây, hoặc phương tiện lưu trữ (nghĩa là, phương tiện lưu trữ không nhất thời), chẳng hạn như đĩa cứng, ổ đĩa chớp, đĩa compac, đĩa video số, đĩa Blu-ray, hoặc phương tiện đọc được bởi máy tính khác. Trong một vài ví dụ, máy chủ mạng (không được thể hiện) có thể thu dữ liệu video được mã hóa từ thiết bị nguồn 12 và cung cấp dữ liệu video được mã hóa đến thiết bị đích 14, ví dụ, thông qua việc truyền mạng. Tương tự, thiết bị tính toán của nhà máy sản xuất phương tiện, chẳng hạn như nhà máy dập đĩa, có thể thu dữ liệu video được mã hóa từ thiết bị nguồn 12 và tạo ra đĩa chứa dữ liệu video được mã hóa. Do đó, phương tiện đọc được bởi máy tính 16 có thể được hiểu là bao gồm một hoặc nhiều dạng phương tiện đọc được bởi máy tính, trong các ví dụ khác nhau.

Giao diện đầu vào 28 của thiết bị đích 14 thu thông tin từ phương tiện đọc được bởi máy tính 16. Thông tin của phương tiện đọc được bởi máy tính 16 có thể bao gồm thông tin cú pháp được xác định bởi bộ mã hóa video 20, mà cũng được sử dụng bởi bộ giải mã video 30, mà bao gồm các phần tử cú pháp mô tả các đặc điểm và/hoặc việc xử lý các khối và các đơn vị được tạo mã khác, ví dụ, nhóm hình ảnh (Group Of Pictures, viết tắt là GOPs). Thiết bị hiển thị 32 hiển thị dữ liệu video được giải mã đến người dùng, và có thể bao gồm bất kỳ trong số nhiều thiết bị hiển thị chẳng hạn như ống tia catôt (Cathode Ray Tube, viết tắt là CRT), bộ hiển thị tinh thể lỏng (Liquid Crystal Display, viết tắt là LCD), bộ hiển thị plasma, bộ hiển thị điốt phát quang hữu cơ (Organic Light Emitting Diode, viết tắt là OLED), hoặc loại thiết bị hiển thị khác.

Bộ mã hóa video 20 và bộ giải mã video 30 có thể hoạt động theo chuẩn tạo mã video, chẳng hạn như chuẩn tạo mã video hiệu suất cao (High Efficiency Video Coding, viết tắt là HEVC) đang được phát triển, và có thể phù hợp với mô hình kiểm tra HEVC (HEVC Test Model, viết tắt là HM). Theo cách khác, bộ mã hóa video 20 và bộ giải mã video 30 có thể hoạt động theo các chuẩn riêng hoặc công nghiệp khác, chẳng hạn như chuẩn lĩnh vực Tiêu chuẩn viễn thông, viết tắt là thuộc Tổ chức Viễn thông quốc tế (ITU-T) H.264, theo cách khác được đề cập đến là nhóm chuyên gia ảnh động (MPEG)-4, phần 10, tạo mã video cải tiến (Advanced Video Coding, viết tắt là AVC), H.265/HEVC, hoặc sự mở rộng của các chuẩn như vậy. Tuy nhiên, các kỹ thuật của sáng chế không giới hạn ở chuẩn tạo mã cụ thể bất kỳ. Các ví dụ khác về các chuẩn tạo mã video bao gồm MPEG-2 và ITU-T H.263. Mặc dù không được thể hiện trên Fig.1, theo một vài khía cạnh, bộ mã hóa video 20 và bộ giải mã video 30 có thể đều được kết hợp với bộ mã hóa và bộ giải mã audio, và có thể bao gồm các bộ đa hợp-bộ giải đa hợp (multiplexer-demultiplexer, viết tắt là MUX-DEMUX) thích hợp, hoặc phần cứng và phần mềm khác, để xử lý mã hóa cả audio và video trong luồng dữ liệu chung hoặc các luồng dữ liệu riêng biệt. Nếu có thể, các bộ MUX-DEMUX có thể phù hợp với giao thức đa hợp ITU H.223, hoặc các giao thức khác chẳng hạn như giao thức gói dữ liệu người dùng (User Datagram Protocol, viết tắt là UDP).

Bộ mã hóa video 20 và bộ giải mã video 30 đều có thể được thực hiện dưới dạng bất kỳ trong số nhiều hệ mạch mã hóa thích hợp, chẳng hạn như một hoặc nhiều bộ vi xử lý, bộ xử lý tín hiệu số (Digital Signal Processor, viết tắt là DSP), mạch tích hợp chuyên dụng (Application Specific Integrated Circuit, viết tắt là ASIC), các mảng cổng lập trình được dạng trường (Field Programmable Gate Array, viết tắt là FPGA), thành phần logic rời rạc, phần mềm, phần cứng, phần sụn hoặc các kết hợp bất kỳ của chúng. Khi các kỹ thuật được thực hiện một phần trong phần mềm, thiết bị có thể lưu trữ các lệnh cho phần mềm thích hợp, phương tiện đọc được bởi máy tính không nhất thời và thực hiện các lệnh trong phần cứng sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật của sáng chế. Mỗi trong số bộ mã hóa video 20 và bộ giải mã video 30 có thể được bao gồm trong một hoặc nhiều bộ mã hóa hoặc bộ giải mã, hoặc có thể được kết hợp dưới dạng một phần của bộ mã hóa/bộ giải mã (codec) kết hợp trong thiết bị tương ứng. Thiết bị bao gồm bộ mã hóa video 20 và/hoặc bộ giải mã video 30 có thể bao gồm mạch tích hợp, bộ vi xử lý, và/hoặc thiết bị truyền thông

không dây, chẳng hạn như điện thoại di động.

Fig.2 là sơ đồ khối minh họa ví dụ về bộ mã hóa video 20 mà có thể thực hiện các kỹ thuật tạo mã video. Bộ mã hóa video 20 có thể thực hiện việc tạo mã trong ảnh (intra-coding) và tạo mã liên ảnh (inter-coding) các khối video nằm trong các lát video. Tạo mã trong ảnh dựa vào dự đoán theo không gian để làm giảm hoặc loại bỏ sự dư thừa theo không gian trong video nằm trong khung video hoặc ảnh được đưa ra. Tạo mã liên ảnh dựa vào dự đoán theo thời gian để làm giảm hoặc loại bỏ sự dư thừa theo thời gian trong video nằm trong các khung hoặc các ảnh liên kế của chuỗi video. Chế độ trong ảnh (Intra-mode - chế độ I) có thể đề cập đến bất kỳ trong số một vài chế độ tạo mã dựa vào không gian. Các chế độ liên ảnh (Inter-mode), chẳng hạn như dự đoán một chiều (còn được gọi là, uni prediction) (chế độ P) hoặc dự đoán hai chiều (còn được gọi là, bi prediction) (chế độ B), có thể đề cập đến bất kỳ trong số một vài chế độ tạo mã dựa vào thời gian.

Như được thể hiện trên Fig.2, bộ mã hóa video 20 thu khối video hiện thời nằm trong khung video cần được mã hóa. Trong ví dụ trên Fig.2, bộ mã hóa video 20 bao gồm bộ phận lựa chọn chế độ 40, bộ nhớ khung tham chiếu 64, bộ cộng 50, bộ phận xử lý biến đổi 52, bộ phận lượng tử hóa 54, và bộ phận tạo mã entropy 56. Bộ phận lựa chọn chế độ 40, lần lượt, bao gồm bộ phận bù chuyển động 44, bộ phận đánh giá chuyển động 42, bộ phận dự đoán trong ảnh (intra prediction) (còn được gọi là, nội dự đoán) 46, và bộ phận phân chia 48. Đối với việc xây dựng lại khối video, bộ mã hóa video 20 cũng bao gồm bộ phận lượng tử hóa ngược 58, bộ phận biến đổi ngược 60, và bộ cộng 62. Bộ lọc giải khối (không được thể hiện trên Fig.2) có thể cũng được bao gồm để lọc các biên khối để loại bỏ các ảnh giả tạo khối từ video được xây dựng lại. Nếu muốn, bộ lọc giải khối thường sẽ lọc đầu ra của bộ cộng 62. Các bộ lọc bổ sung (trong vòng lặp hoặc vòng lặp sau) có thể cũng được sử dụng ngoài bộ lọc giải khối. Các bộ lọc như vậy không được thể hiện để ngắn gọn, nhưng nếu muốn, có thể lọc đầu ra của bộ cộng 50 (dưới dạng bộ lọc trong vòng lặp).

Trong suốt quy trình mã hóa, bộ mã hóa video 20 thu khung video hoặc lát cần được tạo mã. Khung hoặc lát có thể được phân chia thành nhiều khối video. Bộ phận đánh giá chuyển động 42 và bộ phận bù chuyển động 44 thực hiện việc tạo mã dự đoán liên ảnh khối video thu được liên quan đến một hoặc nhiều khối trong một hoặc nhiều khung tham chiếu để cung cấp việc dự đoán theo thời gian. Bộ phận dự đoán trong ảnh 46 có thể theo cách khác

thực hiện việc tạo mã dự đoán trong ảnh khối video thu được liên quan đến một hoặc nhiều khối lân cận trong khung hoặc lát giống như khối cần được tạo mã để cung cấp việc dự đoán theo không gian. Bộ mã hóa video 20 có thể thực hiện nhiều bước tạo mã, ví dụ, để lựa chọn chế độ tạo mã thích hợp cho mỗi khối của dữ liệu video.

Ngoài ra, bộ phận phân chia 48 có thể phân chia các khối của dữ liệu video thành các khối con, dựa vào việc đánh giá của các sơ đồ phân chia trước đó trong những bước tạo mã trước đó. Ví dụ, bộ phận phân chia 48 có thể phân chia ban đầu khung hoặc lát thành các đơn vị tạo mã lớn nhất (Largest Coding Unit, viết tắt là LCU), và phân chia mỗi trong số các LCU thành các đơn vị tạo mã con (các CU con) dựa vào sự phân tích biến dạng tỷ lệ (ví dụ, sự tối ưu hóa biến dạng tỷ lệ). Bộ phận lựa chọn chế độ 40 có thể còn tạo ra cấu trúc dữ liệu cây tứ phân chỉ báo việc phân chia LCU thành các CU con. Các CU nút lá của cây tứ phân có thể bao gồm một hoặc nhiều đơn vị dự đoán (Prediction Unit, viết tắt là PU) và một hoặc nhiều đơn vị biến đổi (Transform Unit, viết tắt là TU).

Sáng chế sử dụng thuật ngữ "khối" để đề cập đến bất kỳ trong số CU, PU, hoặc TU, trong ngữ cảnh của HEVC, hoặc các cấu trúc dữ liệu tương tự trong ngữ cảnh của các chuẩn khác (ví dụ, các khối cỡ lớn (macroblocks) và các khối con của nó trong H.264/AVC). CU bao gồm nút tạo mã, các PU, và các TU được kết hợp với nút tạo mã. Kích thước của CU tương ứng với kích thước của nút tạo mã và có dạng hình vuông. Kích thước của CU có thể trong phạm vi từ 8×8 điểm ảnh lên đến kích thước của khối cây với lớn nhất là 64×64 điểm ảnh hoặc lớn hơn. Mỗi CU có thể chứa một hoặc nhiều PU và một hoặc nhiều TU. Dữ liệu cú pháp được kết hợp với CU có thể mô tả, ví dụ, sự phân chia của CU thành một hoặc nhiều PU. Các chế độ phân chia có thể khác nhau giữa việc xem liệu CU được bỏ qua hay chế độ trực tiếp được mã hóa, chế độ dự đoán trong ảnh được mã hóa, hoặc chế độ dự đoán liên ảnh (inter-prediction) (còn được gọi là, liên dự đoán) được mã hóa. Các PU có thể được phân chia thành dạng không vuông. Dữ liệu cú pháp được kết hợp với CU có thể cũng mô tả, ví dụ, sự phân chia của CU thành một hoặc nhiều TU theo cây tứ phân. TU có thể có dạng hình vuông hoặc không vuông (ví dụ, hình chữ nhật).

Bộ phận lựa chọn chế độ 40 có thể lựa chọn một trong số các chế độ tạo mã, trong ảnh hoặc liên ảnh, ví dụ, dựa vào các kết quả lỗi, và cung cấp khối được tạo mã trong ảnh hoặc liên ảnh thu được đến bộ cộng 50 để tạo ra dữ liệu khối dư và đến bộ cộng 62 để xây dựng

lại khối được mã hóa để sử dụng làm khung tham chiếu. Bộ phận lựa chọn chế độ 40 cũng cung cấp các phần tử cú pháp, chẳng hạn như các vectơ chuyển động, các ký hiệu chỉ báo chế độ trong ảnh, thông tin phân chia, và thông tin cú pháp như vậy khác, đến bộ phận tạo mã entropy 56.

Bộ phận đánh giá chuyển động 42 và bộ phận bù chuyển động 44 có thể được kết hợp cao, nhưng được minh họa riêng biệt cho mục đích nhận thức. Việc đánh giá chuyển động, được thực hiện bởi bộ phận đánh giá chuyển động 42, là quy trình tạo ra các vectơ chuyển động, mà đánh giá sự chuyển động cho các khối video. Vectơ chuyển động, ví dụ, có thể chỉ báo sự dịch chuyển của PU của khối video nằm trong khung video hoặc ảnh hiện thời liên quan đến khối dự đoán nằm trong khung tham chiếu (hoặc đơn vị được tạo mã khác) liên quan đến khối hiện thời được tạo mã nằm trong khung hiện thời (hoặc đơn vị được tạo mã khác). Khối dự đoán là khối được tìm kiếm để so khớp gần với khối cần được tạo mã, xét về sự khác biệt điểm ảnh, mà có thể được xác định bằng cách tính tổng chênh lệch tuyệt đối (Sum of Absolute Difference, viết tắt là SAD), tổng của chênh lệch bình phương (Sum of Square Difference, viết tắt là SSD), hoặc các số liệu chênh lệch khác. Trong một vài ví dụ, bộ mã hóa video 20 có thể tính toán các giá trị cho các vị trí điểm ảnh nguyên con của các ảnh tham chiếu được lưu trữ trong bộ nhớ khung tham chiếu 64. Ví dụ, bộ mã hóa video 20 có thể nội suy các giá trị của các vị trí điểm ảnh một phần tư, các vị trí điểm ảnh một phần tám, hoặc các vị trí điểm ảnh phân số khác của ảnh tham chiếu. Do đó, bộ phận đánh giá chuyển động 42 có thể thực hiện việc tìm kiếm chuyển động liên quan đến các vị trí điểm ảnh đầy đủ và các vị trí điểm ảnh phân số và đưa ra vectơ chuyển động với độ chính xác điểm ảnh phân số.

Bộ phận đánh giá chuyển động 42 tính toán vectơ chuyển động cho PU của khối video trong lát được tạo mã liên ảnh bằng cách so sánh vị trí của PU với vị trí của khối dự đoán của ảnh tham chiếu. Ảnh tham chiếu có thể được lựa chọn từ danh sách hình ảnh tham chiếu thứ nhất (Danh sách 0) hoặc danh sách hình ảnh tham chiếu thứ hai (Danh sách 1), mỗi trong số đó nhận dạng một hoặc nhiều ảnh tham chiếu được lưu trữ trong bộ nhớ khung tham chiếu 64. Bộ phận đánh giá chuyển động 42 gửi vectơ chuyển động được tính toán đến bộ phận mã hóa entropy 56 và bộ phận bù chuyển động 44.

Việc bù chuyển động, được thực hiện bởi bộ phận bù chuyển động 44, có thể bao gồm

việc tìm nạp hoặc tạo ra khối dự đoán dựa vào vectơ chuyển động được xác định bởi bộ phận đánh giá chuyển động 42. Ngoài ra, bộ phận đánh giá chuyển động 42 và bộ phận bù chuyển động 44 có thể được kết hợp theo chức năng, trong một vài ví dụ. Khi thu vectơ chuyển động cho PU của khối video hiện thời, bộ phận bù chuyển động 44 có thể định vị khối dự đoán mà các điểm vectơ chuyển động trong một trong số các danh sách ảnh tham chiếu. Bộ cộng 50 tạo nên khối video dự bằng cách trừ các giá trị điểm ảnh của khối dự đoán từ các giá trị điểm ảnh của khối video hiện thời được tạo mã, tạo nên các giá trị khác biệt điểm ảnh, như được mô tả dưới đây. Nhìn chung, bộ phận đánh giá chuyển động 42 thực hiện việc đánh giá chuyển động liên quan đến các thành phần độ chói, và bộ phận bù chuyển động 44 sử dụng các vectơ chuyển động được tính toán dựa vào các thành phần độ chói cho cả các thành phần sắc độ và các thành phần độ chói. Bộ phận lựa chọn chế độ 40 có thể cũng tạo ra các phần tử cú pháp được kết hợp với các khối video và lát video để sử dụng bởi bộ giải mã video 30 trong việc giải mã các khối video của lát video.

Bộ phận dự đoán trong ảnh 46 có thể dự đoán trong ảnh khối hiện thời, như một giải pháp thay cho việc dự đoán liên ảnh được thực hiện bởi bộ phận đánh giá chuyển động 42 và bộ phận bù chuyển động 44, như được nêu trên. Cụ thể là, bộ phận dự đoán trong ảnh 46 có thể xác định chế độ dự đoán trong ảnh để sử dụng để mã hóa khối hiện thời. Trong một vài ví dụ, bộ phận dự đoán trong ảnh 46 có thể mã hóa khối hiện thời sử dụng các chế độ dự đoán trong ảnh khác nhau, ví dụ, trong các bước mã hóa riêng biệt, và bộ phận dự đoán trong ảnh 46 (hoặc bộ phận lựa chọn chế độ 40, trong một vài ví dụ) có thể lựa chọn chế độ dự đoán trong ảnh thích hợp để sử dụng từ các chế độ được kiểm tra.

Ví dụ, bộ phận dự đoán trong ảnh 46 có thể tính toán các giá trị biến dạng tỷ lệ sử dụng sự phân tích biến dạng tỷ lệ cho các chế độ dự đoán trong ảnh được kiểm tra khác nhau, và lựa chọn chế độ dự đoán trong ảnh có các đặc điểm biến dạng tỷ lệ tốt nhất trong số các chế độ được kiểm tra. Sự phân tích biến dạng tỷ lệ thường xác định lượng biến dạng (hoặc lỗi) giữa khối được mã hóa và khối không được mã hóa ban đầu mà được mã hóa để tạo ra khối được mã hóa, cũng như tốc độ bit (nghĩa là, số lượng bit) được sử dụng để tạo ra khối được mã hóa. Bộ phận dự đoán trong ảnh 46 có thể tính toán các tỷ lệ từ các biến dạng và các tỷ lệ cho các khối được mã hóa khác nhau để xác định chế độ dự đoán trong ảnh nào thể hiện giá trị biến dạng tỷ lệ tốt nhất cho khối.

Ngoài ra, bộ phận dự đoán trong ảnh 46 có thể được tạo cấu hình để tạo mã các khối độ sâu của ảnh xạ độ sâu sử dụng chế độ tạo mô hình độ sâu (Depth Modeling Mode, viết tắt là DMM). Bộ phận lựa chọn chế độ 40 có thể xác định xem liệu chế độ DMM khả dụng có tạo ra các kết quả tạo mã tốt hơn so với chế độ dự đoán trong ảnh và các chế độ DMM khác hay không, ví dụ, sử dụng sự tối ưu hóa biến dạng tỷ lệ (Rate-Distortion Optimization, viết tắt là RDO). Dữ liệu cho hình ảnh kết cấu tương ứng với ảnh xạ độ sâu có thể được lưu trữ trong bộ nhớ khung tham chiếu 64. Bộ phận đánh giá chuyển động 42 và bộ phận bù chuyển động 44 có thể cũng được tạo cấu hình để dự đoán liên ảnh các khối độ sâu của ảnh xạ độ sâu.

Sau khi lựa chọn chế độ dự đoán trong ảnh cho khối (ví dụ, chế độ dự đoán trong ảnh thông thường hoặc một trong số các chế độ DMM), bộ phận dự đoán trong ảnh 46 có thể cung cấp chỉ báo thông tin của chế độ dự đoán trong ảnh được lựa chọn cho khối đến bộ phận tạo mã entropy 56. Bộ phận tạo mã entropy 56 có thể mã hóa thông tin chỉ báo chế độ dự đoán trong ảnh được lựa chọn. Bộ mã hóa video 20 có thể bao gồm trong dữ liệu cấu hình dòng bit được truyền, mà có thể bao gồm các bảng chỉ số chế độ dự đoán trong ảnh và các bảng chỉ số chế độ dự đoán trong ảnh được cải biến (cũng được đề cập đến là các bảng từ mã ảnh xạ), các định nghĩa về các ngữ cảnh mã hóa cho các khối khác nhau, và các chỉ báo của chế độ dự đoán trong ảnh có thể xảy ra nhất, bảng chỉ số chế độ dự đoán trong ảnh, và bảng chỉ số chế độ dự đoán trong ảnh được cải biến để sử dụng cho mỗi trong số các ngữ cảnh.

Bộ mã hóa video 20 tạo nên khối video dự bằng cách trừ dữ liệu dự đoán từ bộ phận lựa chọn chế độ 40 từ khối video ban đầu được tạo mã. Bộ cộng 50 thể hiện thành phần hoặc các thành phần mà thực hiện thao tác trừ này.

Bộ phận xử lý biến đổi 52 áp dụng biến đổi, chẳng hạn như biến đổi cosin rời rạc (Discrete Cosine Transform, viết tắt là DCT) hoặc biến đổi tương tự về mặt nguyên lý, cho khối dự, tạo ra khối video bao gồm các giá trị hệ số biến đổi dự. Bộ phận xử lý biến đổi 52 có thể thực hiện các biến đổi khác mà tương tự về mặt nguyên lý với DCT. Các biến đổi Wavelet, các biến đổi số nguyên, các biến đổi dải con hoặc các loại biến đổi khác có thể cũng được sử dụng.

Bộ phận xử lý biến đổi 52 áp dụng biến đổi cho khối dự, tạo ra khối của các hệ số biến đổi dự. Biến đổi có thể chuyển đổi thông tin dự từ miền giá trị điểm ảnh đến miền biến đổi,

chẳng hạn như miền tần số. Bộ phận xử lý biến đổi 52 có thể gửi các hệ số biến đổi thu được đến bộ phận lượng tử hóa 54. Bộ phận lượng tử hóa 54 lượng tử hóa các hệ số biến đổi để làm giảm thêm tốc độ bit. Quy trình lượng tử hóa có thể làm giảm độ sâu bit được kết hợp với một vài hoặc tất cả các hệ số. Mức độ lượng tử hóa có thể được cải biến bằng cách điều chỉnh thông số lượng tử hóa. Trong một vài ví dụ, bộ phận lượng tử hóa 54 sau đó có thể thực hiện việc quét ma trận bao gồm các hệ số biến đổi được lượng tử hóa. Theo cách khác, bộ phận mã hóa entropi 56 có thể thực hiện việc quét.

Tiếp theo việc lượng tử hóa, bộ phận tạo mã entropi 56 tạo mã entropi các hệ số biến đổi được lượng tử hóa. Ví dụ, bộ phận tạo mã entropi 56 có thể thực hiện tạo mã độ dài biến đổi thích ứng theo ngữ cảnh (Context Adaptive Variable Length Coding, viết tắt là CAVLC), tạo mã số học nhị phân thích ứng theo ngữ cảnh (Context Adaptive Binary Arithmetic Coding, viết tắt là CABAC), tạo mã số học nhị phân thích ứng theo ngữ cảnh dựa trên cú pháp (Syntax-Based Context-Adaptive Binary Arithmetic Coding, viết tắt là SBAC), tạo mã entropi phân chia khoảng xác suất (Probability Interval Partitioning Entropy, viết tắt là PIPE) hoặc kỹ thuật tạo mã entropi khác. Trong trường hợp tạo mã entropi dựa trên ngữ cảnh, ngữ cảnh có thể dựa vào các khối lân cận. Tiếp theo việc tạo mã entropi bởi bộ phận tạo mã entropi 56, dòng bit được mã hóa có thể được truyền đến thiết bị khác (ví dụ, bộ giải mã video 30) hoặc được lưu trữ để truyền hoặc truy xuất sau đó.

Bộ phận lượng tử hóa ngược 58 và bộ phận biến đổi ngược 60 áp dụng việc lượng tử hóa ngược và biến đổi ngược, tương ứng, để xây dựng lại khối dư trong miền điểm ảnh, ví dụ, cho việc sử dụng sau đó dưới dạng khối tham chiếu. Bộ phận bù chuyển động 44 có thể tính toán khối tham chiếu bằng cách bổ sung khối dư vào khối dự đoán của một trong số các khung của bộ nhớ khung tham chiếu 64. Bộ phận bù chuyển động 44 có thể cũng áp dụng một hoặc nhiều bộ lọc nội suy cho khối dư được xây dựng lại để tính toán các giá trị điểm ảnh nguyên con để sử dụng trong việc đánh giá chuyển động. Bộ cộng 62 cộng khối dư được xây dựng lại vào khối dự đoán được bù chuyển động được tạo ra bởi bộ phận bù chuyển động 44 để tạo ra khối video được xây dựng lại để lưu trữ trong bộ nhớ khung tham chiếu 64. Khối video được xây dựng lại có thể được sử dụng bởi bộ phận đánh giá chuyển động 42 và bộ phận bù chuyển động 44 dưới dạng khối tham chiếu để tạo mã liên ảnh khối trong khung video sau đó.

Fig.3 là sơ đồ khối minh họa ví dụ về bộ giải mã video 30 mà có thể thực hiện các kỹ thuật tạo mã video. Trong ví dụ trên Fig.3, bộ giải mã video 30 bao gồm bộ phận giải mã entropy 70, bộ phận bù chuyển động 72, bộ phận dự đoán trong ảnh 74, bộ phận lượng tử hóa ngược 76, bộ phận biến đổi ngược 78, bộ nhớ khung tham chiếu 82, và bộ cộng 80. Bộ giải mã video 30 có thể, trong một vài ví dụ, thực hiện bước giải mã thường ngược với bước mã hóa được mô tả đối với bộ mã hóa video 20 (Fig.2). Bộ phận bù chuyển động 72 có thể tạo ra dữ liệu dự đoán dựa vào các vector chuyển động thu được từ bộ phận giải mã entropy 70, trong khi bộ phận dự đoán trong ảnh 74 có thể tạo ra dữ liệu dự đoán dựa vào các ký hiệu chỉ báo chế độ dự đoán trong ảnh thu được từ bộ phận giải mã entropy 70.

Trong suốt quy trình giải mã, bộ giải mã video 30 thu dòng bit video được mã hóa thể hiện các khối video của lát video được mã hóa và các phần tử cú pháp được kết hợp từ bộ mã hóa video 20. Bộ phận giải mã entropy 70 của bộ giải mã video 30 giải mã entropy dòng bit để tạo ra các hệ số được lượng tử hóa, các vector chuyển động hoặc các ký hiệu chỉ báo chế độ dự đoán trong ảnh, và các phần tử cú pháp khác. Bộ phận giải mã entropy 70 chuyển tiếp các vector chuyển động và các phần tử cú pháp khác đến bộ phận bù chuyển động 72. Bộ giải mã video 30 có thể thu các phần tử cú pháp ở mức lát video và/hoặc mức khối video.

Khi lát video được tạo mã dưới dạng lát được mã hóa trong ảnh (I), bộ phận dự đoán trong ảnh 74 có thể tạo ra dữ liệu dự đoán cho khối video của lát video hiện thời dựa vào chế độ dự đoán trong ảnh được báo hiệu và dữ liệu từ các khối được giải mã trước đó của khung hoặc ảnh hiện thời. Khi khung video được tạo mã dưới dạng lát được tạo mã liên ảnh (ví dụ, B, P, hoặc GPB), bộ phận bù chuyển động 72 tạo ra các khối dự đoán cho khối video của lát video hiện thời dựa vào các vector chuyển động và các phần tử cú pháp khác thu được từ bộ phận giải mã entropy 70. Các khối dự đoán có thể được tạo ra từ một trong số các ảnh tham chiếu nằm trong một trong số các danh sách ảnh tham chiếu. Bộ giải mã video 30 có thể xây dựng các danh sách khung tham chiếu, Danh sách 0 và Danh sách 1, sử dụng các kỹ thuật xây dựng mặc định dựa vào các ảnh tham chiếu được lưu trữ trong bộ nhớ khung tham chiếu 82.

Bộ phận bù chuyển động 72 xác định thông tin dự đoán cho khối video của lát video hiện thời bằng cách phân tích các vector chuyển động và các phần tử cú pháp khác, và sử dụng thông tin dự đoán để tạo ra các khối dự đoán cho khối video hiện thời được giải mã. Ví dụ,

bộ phận bù chuyển động 72 sử dụng một vài trong số các phần tử cú pháp thu được để xác định chế độ dự đoán (ví dụ, dự đoán trong ảnh hoặc liên ảnh) được sử dụng để tạo mã các khối video của lát video, loại lát dự đoán liên ảnh (ví dụ, lát B, lát P, hoặc lát GPB), thông tin xây dựng cho một hoặc nhiều trong số các danh sách ảnh tham chiếu cho lát, các vector chuyển động cho mỗi khối video được mã hóa liên ảnh của lát, trạng thái dự đoán liên ảnh cho mỗi khối video được tạo mã liên ảnh của lát, và thông tin khác để giải mã các khối video trong lát video hiện thời.

Bộ phận bù chuyển động 72 có thể cũng thực hiện việc nội suy dựa vào các bộ lọc nội suy. Bộ phận bù chuyển động 72 có thể sử dụng các bộ lọc nội suy như được sử dụng bởi bộ mã hóa video 20 trong suốt quá trình mã hóa của các khối video để tính toán các giá trị được nội suy cho các điểm ảnh số nguyên con của các khối tham chiếu. Trong trường hợp này, bộ phận bù chuyển động 72 có thể xác định các bộ lọc nội suy được sử dụng bởi bộ mã hóa video 20 từ các phần tử cú pháp thu được và sử dụng các bộ lọc nội suy để tạo ra các khối dự đoán.

Dữ liệu cho hình ảnh kết cấu tương ứng với ánh xạ độ sâu có thể được lưu trữ trong bộ nhớ khung tham chiếu 82. Bộ phận bù chuyển động 72 có thể cũng được tạo cấu hình để dự đoán liên ảnh các khối độ sâu của ánh xạ độ sâu.

Nén ảnh và video đã có sự phát triển nhanh chóng, dẫn đến các chuẩn tạo mã khác nhau. Các chuẩn tạo mã video như vậy bao gồm ITU-T H.261, ISO/IEC nhóm chuyên gia ảnh động (Motion Picture Experts Group, viết tắt là MPEG)-1 phần 2, ITU-T H.262 hoặc Tổ chức quốc tế về Tiêu chuẩn hóa (International Organization for Standardization, viết tắt là ISO)/ Ủy ban Kỹ thuật điện quốc tế (International Electrotechnical Commission, viết tắt là IEC) MPEG-2 phần 2, ITU-T H.263, ISO/IEC MPEG-4 phần 2, tạo mã video cải tiến (Advanced Video Coding, viết tắt là AVC), cũng được biết đến là ITU-T H.264 hoặc ISO/IEC MPEG-4 phần 10, và tạo mã video hiệu suất cao (High Efficiency Video Coding, viết tắt là HEVC), cũng được biết đến là ITU-T H.265 hoặc MPEG-H phần 2. AVC bao gồm các mở rộng chẳng hạn như tạo mã video có thể mở rộng (Scalable Video Coding, viết tắt là SVC), tạo mã video đa diện (Multiview Video Coding, viết tắt là MVC) và tạo mã video đa diện chiều sâu mở rộng (Multiview Video Coding plus Depth, viết tắt là MVC+D), và 3D AVC (3D-AVC). HEVC bao gồm các mở rộng chẳng hạn như HEVC mở rộng được

(Scalable HEVC, viết tắt là SHVC), HEVC đa diện (Multiview HEVC, viết tắt là MV-HEVC), và 3D HEVC (3D-HEVC).

Tạo mã video đa năng (Versatile Video Coding, viết tắt là VVC) là chuẩn tạo mã video mới được phát triển bởi nhóm chuyên gia video nổi (Joint Video Experts Team, viết tắt là JVET) của ITU-T và ISO/IEC. Ở thời điểm ghi, dự thảo (Working Draft, viết tắt là WD) cuối cùng của VVC được bao gồm trong JVET-K1001-v1. Tài liệu JVET JVET-K0325-v3 bao gồm bản cập nhật cho cú pháp mức cao của VVC.

Nhìn chung, sáng chế mô tả các kỹ thuật dựa vào sự đang phát triển của chuẩn VCC. Tuy nhiên, các kỹ thuật cũng áp dụng cho các đặc tả codec video/phương tiện khác.

Các kỹ thuật nén video thực hiện việc dự đoán theo không gian (intra-picture - trong ảnh) và/hoặc việc dự đoán theo thời gian (inter-picture - liên ảnh) để làm giảm hoặc loại bỏ dư thừa vốn có trong các chuỗi video. Đối với tạo mã video dựa trên khối, lát video (ví dụ, hình ảnh video hoặc một phần của hình ảnh video) có thể được phân chia thành các khối video, mà có thể cũng được đề cập đến là các khối cây, các khối cây tạo mã (Coding Tree Block, viết tắt là CTB), các đơn vị cây tạo mã (Coding Tree Unit, viết tắt là CTU), các đơn vị tạo mã (Coding Unit, viết tắt là CU) và/hoặc các nút tạo mã. Các khối video trong lát được tạo mã trong ảnh (I) của hình ảnh được mã hóa sử dụng dự đoán theo không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng hình ảnh. Các khối video trong lát được tạo mã liên ảnh (P hoặc B) của hình ảnh có thể sử dụng dự đoán theo không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng hình ảnh hoặc dự đoán theo thời gian đối với các mẫu tham chiếu trong các hình ảnh tham chiếu khác. Các hình ảnh có thể được đề cập đến là các khung, và các hình ảnh tham chiếu có thể được đề cập đến là các khung tham chiếu.

Dự đoán theo không gian hoặc thời gian dẫn đến khối dự đoán cho khối cần được tạo mã. Dữ liệu dư thể hiện các khác biệt điểm ảnh giữa khối ban đầu cần được tạo mã và khối dự đoán. Khối được tạo mã liên ảnh được mã hóa theo vector chuyển động mà chỉ đến khối của các mẫu tham chiếu tạo nên khối dự đoán, và dữ liệu dư chỉ báo sự khác biệt giữa khối được tạo mã và khối dự đoán. Khối được tạo mã trong ảnh được mã hóa theo chế độ tạo mã trong ảnh và dữ liệu dư. Để nén thêm, dữ liệu dư có thể được biến đổi từ miền điểm ảnh sang miền biến đổi, dẫn đến các hệ số biến đổi dư, mà sau đó có thể được lượng tử hóa. Các hệ số

biến đổi được lượng tử hóa, được bố trí ban đầu trong mảng hai chiều, có thể được quét để tạo ra vector một chiều của các hệ số biến đổi, và việc tạo mã entropi có thể được áp dụng để đạt được thậm chí mức nén cao hơn.

Trong đặc tả codec video, các hình ảnh được nhận dạng cho nhiều mục đích, bao gồm việc sử dụng như hình ảnh tham chiếu trong việc dự đoán liên ảnh, để đưa ra các hình ảnh từ bộ đệm hình ảnh được giải mã (Decoded Picture Buffer, viết tắt là DPB), để định tỷ lệ các vector chuyển động, cho việc dự đoán có trọng số, v.v.. Trong AVC và HEVC, các hình ảnh có thể được nhận dạng bởi số đếm thứ tự hình ảnh (Picture Order Count, viết tắt là POC). Trong AVC và HEVC, các hình ảnh trong DPB có thể được đánh dấu là "được sử dụng cho việc tham chiếu ngắn hạn", "được sử dụng cho việc tham chiếu dài hạn", hoặc "không được sử dụng cho việc tham chiếu". Một khi hình ảnh đã được đánh dấu "không được sử dụng cho việc tham chiếu", hình ảnh có thể không còn được sử dụng để dự đoán. Khi hình ảnh không còn cần để đưa ra, hình ảnh có thể được loại bỏ khỏi DPB.

Trong AVC, có hai loại hình ảnh tham chiếu, ngắn hạn và dài hạn. Hình ảnh tham chiếu có thể được đánh dấu là "không được sử dụng cho việc tham chiếu" khi nó trở nên không còn cần cho việc tham chiếu dự đoán. Sự chuyển đổi giữa ba trạng thái này (ngắn hạn, dài hạn, và không được sử dụng cho việc tham chiếu) được điều khiển bởi quy trình đánh dấu hình ảnh tham chiếu được giải mã. Có hai cơ chế đánh dấu hình ảnh tham chiếu được giải mã thay thế, quy trình cửa sổ trượt âm và quy trình thao tác điều khiển quản lý bộ nhớ (Memory Management Control Operation, viết tắt là MMCO) hiện. Quy trình cửa sổ trượt đánh dấu hình ảnh tham chiếu ngắn hạn là "không được sử dụng cho việc tham chiếu" khi số lượng của các khung tham chiếu bằng số lớn nhất xác định (`max_num_ref_frames` trong tập hợp thông số chuỗi (Sequence Parameter Set, viết tắt là SPS)). Các hình ảnh tham chiếu ngắn hạn được lưu trữ theo cách vào trước ra trước sao cho các hình ảnh ngắn hạn được giải mã gần đây nhất được duy trì trong DPB.

Quy trình MMCO hiện có thể bao gồm nhiều lệnh MMCO. Lệnh MMCO có thể đánh dấu một hoặc nhiều hình ảnh tham chiếu ngắn hạn hoặc dài hạn là "không được sử dụng cho việc tham chiếu", có thể đánh dấu tất cả các hình ảnh là "không được sử dụng cho việc tham chiếu", hoặc có thể đánh dấu hình ảnh tham chiếu hiện thời hoặc hình ảnh tham chiếu ngắn hạn hiện thời là dài hạn và sau đó gán chỉ số hình ảnh dài hạn cho hình ảnh tham chiếu dài

hạn đó.

Trong AVC, các thao tác đánh dấu hình ảnh tham chiếu cũng như quy trình để đưa ra và loại bỏ các hình ảnh khỏi DPB được thực hiện sau khi hình ảnh đã được giải mã.

HEVC giới thiệu cách tiếp cận khác nhau cho việc quản lý hình ảnh tham chiếu, được đề cập đến là tập hợp hình ảnh tham chiếu (Reference Picture Set, viết tắt là RPS). Sự khác biệt cơ bản nhất của khái niệm RPS so với MMCO/quy trình của sổ trượt của AVC là đối với mỗi lát cụ thể, tập hợp đầy đủ của các hình ảnh tham chiếu được sử dụng bởi hình ảnh hiện thời hoặc hình ảnh sau đó bất kỳ được đưa ra. Do đó, tập hợp đầy đủ tất cả các hình ảnh phải được duy trì trong DPB để sử dụng bởi hình ảnh hiện thời hoặc tương lai được báo hiệu. Điều này khác với sơ đồ AVC ở chỗ chỉ thay đổi tương đối với DPB được báo hiệu. Với khái niệm RPS, không có thông tin từ các hình ảnh sớm hơn theo thứ tự giải mã cần để duy trì trạng thái đúng của các hình ảnh tham chiếu trong DPB.

Thứ tự của các thao tác giải mã hình ảnh và DPB trong HEVC được thay đổi so với AVC để lợi dụng ưu điểm của RPS và nâng cao khả năng phục hồi lỗi. Trong AVC, các thao tác đánh dấu hình ảnh và đệm (cả đưa ra và loại bỏ các hình ảnh được giải mã khỏi DPB) thường được áp dụng sau khi hình ảnh hiện thời đã được giải mã. Trong HEVC, RPS được giải mã đầu tiên từ phần đầu lát của hình ảnh hiện thời, sau đó các thao tác đánh dấu hình ảnh và đệm thường được áp dụng trước khi giải mã hình ảnh hiện thời.

Mỗi phần đầu lát trong HEVC phải bao gồm các thông số để báo hiệu RPS cho hình ảnh chứa các lát. Ngoại lệ duy nhất là không có RPS được báo hiệu cho các lát làm mới giải mã tức thời (Instantaneous Decoding Refresh, viết tắt là IDR). Thay vào đó, RPS được suy ra là trống. Đối với các lát I không thuộc về hình ảnh IDR, RPS có thể được đưa ra, ngay cả nếu chúng thuộc về hình ảnh I vì có thể có các hình ảnh tiếp theo hình ảnh I theo thứ tự giải mã mà sử dụng dự đoán liên ảnh từ các hình ảnh mà đứng trước hình ảnh I theo thứ tự giải mã. Số lượng của các hình ảnh trong RPS sẽ không vượt quá giới hạn kích thước DPB như được định rõ bởi phần tử cú pháp `sps_max_dec_pic_buffering` trong SPS.

Mỗi hình ảnh được kết hợp với trị số POC mà thể hiện thứ tự đầu ra. Các phần đầu lát chứa từ mã độ dài cố định, `pic_order_cnt_lsb`, thể hiện các bit có trọng số thấp nhất (Least Significant Bit, viết tắt là LSB) của trị số POC đầy đủ, cũng được biết đến là POC LSB. Độ dài của từ mã được báo hiệu trong SPS và có thể là, ví dụ, giữa 4 và 16 bit. Khái niệm RPS

sử dụng POC để nhận dạng các hình ảnh tham chiếu. Bên cạnh trị số POC của chính nó, mỗi phần đầu lát trực tiếp chứa hoặc kế thừa từ SPS sự thể hiện được tạo mã của các trị số POC (hoặc các LSB) của mỗi hình ảnh trong RPS.

RPS đối với mỗi hình ảnh bao gồm năm danh sách khác nhau của các hình ảnh tham chiếu, cũng được đề cập đến là năm tập con RPS. RefPicSetStCurrBefore bao gồm tất cả các hình ảnh tham chiếu ngắn hạn mà đứng trước hình ảnh hiện thời trong cả thứ tự giải mã và thứ tự đưa ra, và có thể được sử dụng trong việc dự đoán liên ảnh của hình ảnh hiện thời. RefPicSetStCurrAfter bao gồm tất cả các hình ảnh tham chiếu ngắn hạn mà đứng trước hình ảnh hiện thời theo thứ tự giải mã, mà nối tiếp hình ảnh hiện thời theo thứ tự đưa ra, và có thể được sử dụng trong việc dự đoán liên ảnh của hình ảnh hiện thời. RefPicSetStFoll bao gồm tất cả các hình ảnh tham chiếu ngắn hạn mà có thể được sử dụng trong việc dự đoán liên ảnh của một hoặc nhiều trong số các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã, và không được sử dụng trong việc dự đoán liên ảnh của hình ảnh hiện thời. RefPicSetLtCurr bao gồm tất cả các hình ảnh tham chiếu dài hạn mà có thể được sử dụng trong việc dự đoán liên ảnh của hình ảnh hiện thời. RefPicSetLtFoll bao gồm tất cả các hình ảnh tham chiếu dài hạn mà có thể được sử dụng trong việc dự đoán liên ảnh của một hoặc nhiều trong số các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã, và không được sử dụng trong việc dự đoán liên ảnh của hình ảnh hiện thời.

RPS được báo hiệu sử dụng lên tới ba vòng lặp lại trên các loại hình ảnh tham chiếu khác nhau; các hình ảnh tham chiếu ngắn hạn có trị số POC thấp hơn so với hình ảnh hiện thời, các hình ảnh tham chiếu ngắn hạn có trị số POC cao hơn so với hình ảnh hiện thời, và các hình ảnh tham chiếu dài hạn. Ngoài ra, cờ (`used_by_curr_pic_X_flag`) được gửi cho mỗi hình ảnh tham chiếu chỉ báo xem liệu hình ảnh tham chiếu có được sử dụng cho việc tham chiếu bởi hình ảnh hiện thời (được bao gồm trong một trong số các danh sách RefPicSetStCurrBefore, RefPicSetStCurrAfter hoặc RefPicSetLtCurr) hay không (được bao gồm trong một trong số các danh sách RefPicSetStFoll hoặc RefPicSetLtFoll).

Fig.4 minh họa RPS 400 có hình ảnh hiện thời B14 với các mục nhập (ví dụ, hình ảnh) trong tất cả các tập con 402 của RPS 400. Trong ví dụ trên Fig.4, hình ảnh hiện thời B14 chứa chính xác một hình ảnh trong mỗi trong số năm tập con 402 (còn được biết đến là, các tập con RPS). P8 là hình ảnh trong tập con 402 được đề cập đến là RefPicSetStCurrBefore

bởi vì hình ảnh có trước theo thứ tự đưa ra và được sử dụng bởi B14. P12 là hình ảnh trong tập con 402 được đề cập đến là RefPicSetStCurrAfter bởi vì hình ảnh có sau theo thứ tự đưa ra và được sử dụng bởi B14. P13 là hình ảnh trong tập con 402 được đề cập đến là RefPicSetStFoll bởi vì hình ảnh này là hình ảnh tham chiếu ngắn hạn không được sử dụng bởi B14 (nhưng phải được duy trì trong DPB vì nó được sử dụng bởi B15). P4 là hình ảnh trong tập con 402 được đề cập đến là RefPicSetLtCurr bởi vì hình ảnh này là hình ảnh tham chiếu dài hạn được sử dụng bởi B14. I0 là hình ảnh trong tập con 402 được đề cập đến là RefPicSetLtFoll vì hình ảnh là hình ảnh tham chiếu dài hạn không được sử dụng bởi hình ảnh hiện thời (nhưng phải được duy trì trong DPB vì nó được sử dụng bởi B15).

Phần ngắn hạn của RPS 400 có thể được bao gồm trực tiếp trong phần đầu lát. Theo cách khác, phần đầu lát có thể chứa chỉ phần tử cú pháp mà thể hiện chỉ số, nối tiếp danh sách được xác định trước của các RPS được gửi trong SPS hoạt động. Phần ngắn hạn của RPS 402 có thể được báo hiệu sử dụng một trong hai sơ đồ khác nhau; RPS liên ảnh, như được mô tả dưới đây, hoặc RPS trong ảnh, như được mô tả ở đây. Khi RPS trong ảnh được sử dụng, num_negative_pics và num_positive_pics được báo hiệu thể hiện độ dài của hai danh sách khác nhau của các hình ảnh tham chiếu. Các danh sách này chứa các hình ảnh tham chiếu lần lượt có chênh lệch POC âm và chênh lệch POC dương so với hình ảnh hiện thời. Mỗi phần tử trong các danh sách này được mã hóa với mã độ dài biến đổi thể hiện sự khác biệt trị số POC so với phần tử trước đó trong danh sách trừ một. Đối với hình ảnh thứ nhất trong mỗi danh sách, báo hiệu có liên quan tới trị số POC của hình ảnh hiện thời trừ một.

Khi mã hóa các RPS định kỳ trong tập hợp thông số chuỗi, có thể mã hóa các phần tử của một RPS (ví dụ, RPS 400) dựa vào RPS khác đã được mã hóa trong tập hợp thông số chuỗi. Đây được gọi là RPS liên ảnh. Không có các vấn đề về độ mạnh của lỗi liên quan tới phương pháp này vì tất cả các RPS của tập hợp thông số chuỗi nằm trong cùng đơn vị lớp trừu tượng hóa mạng (Network Abstraction Layer, viết tắt là NAL). Cú pháp RPS liên ảnh lợi dụng thực tế là RPS của hình ảnh hiện thời có thể được dự đoán từ RPS của hình ảnh được giải mã trước đó. Đó là bởi vì tất cả các hình ảnh tham chiếu của hình ảnh hiện thời phải hoặc là các hình ảnh tham chiếu của hình ảnh trước đó hoặc chính hình ảnh được giải mã trước đó. Chỉ cần chỉ báo những hình ảnh nào sẽ là các hình ảnh tham chiếu và được sử dụng cho việc dự đoán hình ảnh hiện thời. Do đó, cú pháp này bao gồm như sau: chỉ số chỉ ra RPS để sử

dụng như bộ dự đoán, `delta_POC` được bổ sung vào `delta_POC` của bộ dự đoán để thu nhận `delta_POC` của RPS hiện thời, và tập hợp các ký hiệu chỉ báo để chỉ báo các hình ảnh là các hình ảnh tham chiếu và liệu chúng chỉ được sử dụng cho việc dự đoán các hình ảnh tương lai hay không.

Các bộ mã hóa muốn lợi dụng việc sử dụng các hình ảnh tham chiếu dài hạn phải thiết đặt phần tử cú pháp `SPS long_term_ref_pics_present_flag` đến một. Các hình ảnh tham chiếu dài hạn có thể sau đó được báo hiệu trong phần đầu lát bởi các từ mã độ dài cố định, `poc_lsb_lt`, thể hiện các bit có trọng số thấp nhất của trị số POC đầy đủ của mỗi hình ảnh dài hạn. Mỗi `poc_lsb_lt` là bản sao của `pic_order_cnt_lsb` từ mã mà được báo hiệu cho hình ảnh dài hạn cụ thể. Cũng có thể báo hiệu tín hiệu tập hợp các hình ảnh dài hạn trong SPS dưới dạng danh sách của các trị số POC LSB. POC LSB cho hình ảnh dài hạn có thể sau đó được báo hiệu trong phần đầu lát dưới dạng chỉ số đến danh sách này.

Phần tử cú pháp `delta_POC_msb_cycle_lt_minus1` có thể được báo hiệu bổ sung để cho phép việc tính toán khoảng cách POC đầy đủ của hình ảnh tham chiếu dài hạn so với hình ảnh hiện thời. Yêu cầu rằng từ mã `delta_poc_msb_cycle_lt_minus1` được báo hiệu cho mỗi hình ảnh tham chiếu dài hạn mà có trị số POC LSB giống như hình ảnh tham chiếu khác bất kỳ trong RPS.

Đối với việc đánh dấu hình ảnh tham chiếu trong HEVC, thường sẽ có một số các hình ảnh có mặt trong DPB trước khi giải mã hình ảnh. Một vài hình ảnh có thể là khả dụng để dự đoán và, như vậy, được đánh dấu là "được sử dụng cho việc tham chiếu". Các hình ảnh khác có thể là không khả dụng để dự đoán nhưng đang chờ để đưa ra và, như vậy, được đánh dấu là "không được sử dụng cho việc tham chiếu". Khi phần đầu lát đã được phân tích, quy trình đánh dấu hình ảnh được thực hiện trước khi dữ liệu lát được giải mã. Các hình ảnh mà có mặt trong DPB và được đánh dấu là "được sử dụng cho việc tham chiếu" nhưng không được bao gồm trong RPS được đánh dấu "không được sử dụng cho việc tham chiếu". Các hình ảnh không có mặt trong DPB nhưng được bao gồm trong tập hợp hình ảnh tham chiếu được bỏ qua khi `used_by_curr_pic_X_flag` bằng không. Tuy nhiên, khi `used_by_curr_pic_X_flag` bằng một, hình ảnh tham chiếu này được dự định để được sử dụng cho việc dự đoán trong hình ảnh hiện thời nhưng bị thiếu. Sau đó sự mất hình ảnh vô tình được suy ra và bộ giải mã sẽ thực hiện hành động thích hợp.

Sau khi giải mã hình ảnh hiện thời, nó được đánh dấu "được sử dụng cho việc tham chiếu ngắn hạn".

Tiếp theo, việc xây dựng danh sách hình ảnh tham chiếu trong HEVC được thảo luận. Trong HEVC, thuật ngữ dự đoán liên ảnh được sử dụng để thể hiện dự đoán được dẫn xuất từ các phần tử dữ liệu (ví dụ, các trị số mẫu hoặc các vector chuyển động) của các hình ảnh tham chiếu khác với hình ảnh được giải mã hiện thời. Giống như trong AVC, hình ảnh có thể được dự đoán từ nhiều hình ảnh tham chiếu. Các hình ảnh tham chiếu được sử dụng cho việc dự đoán liên ảnh được sắp xếp trong một hoặc nhiều danh sách hình ảnh tham chiếu. Chỉ số tham chiếu nhận dạng các hình ảnh tham chiếu trong danh sách sẽ được sử dụng để tạo tín hiệu dự đoán.

Một danh sách hình ảnh tham chiếu, danh sách 0, được sử dụng cho lát P và hai danh sách hình ảnh tham chiếu, danh sách 0 và danh sách 1 được sử dụng cho các lát B. Tương tự như AVC, việc xây dựng danh sách hình ảnh tham chiếu trong HEVC bao gồm việc khởi tạo danh sách hình ảnh tham chiếu và việc cải biến danh sách hình ảnh tham chiếu.

Trong AVC, quy trình khởi tạo đối với danh sách 0 là khác nhau đối với lát P (mà thứ tự giải mã được sử dụng) và các lát B (mà thứ tự đưa ra được sử dụng). Trong HEVC, thứ tự đưa ra được sử dụng trong cả hai trường hợp.

Việc khởi tạo danh sách hình ảnh tham chiếu tạo danh sách 0 và danh sách 1 mặc định (nếu lát là lát B) dựa vào ba tập con RPS: RefPicSetStCurrBefore, RefPicSetStCurrAfter, và RefPicSetLtCurr. Các hình ảnh ngắn hạn có thứ tự đưa ra sớm hơn (muộn hơn) được đưa vào đầu tiên danh sách 0 (danh sách 1) theo thứ tự tăng dần của khoảng cách POC đến hình ảnh hiện thời, sau đó các hình ảnh ngắn hạn có thứ tự đưa ra muộn hơn (sớm hơn) được đưa vào danh sách 0 (danh sách 1) theo thứ tự tăng dần của khoảng cách POC đến hình ảnh hiện thời, và sau đó cuối cùng các hình ảnh dài hạn được đưa vào ở cuối. Xét về RPS, đối với danh sách 0, các mục nhập trong RefPicSetStCurrBefore được đưa vào trong danh sách ban đầu, tiếp theo là các mục nhập trong RefPicSetStCurrAfter. Sau đó, các mục nhập trong RefPicSetLtCurr, nếu khả dụng, được thêm vào.

Trong HEVC, quy trình nêu trên được lặp lại (các hình ảnh tham chiếu đã được bổ sung vào danh sách hình ảnh tham chiếu được bổ sung lại) khi số lượng mục nhập trong danh sách nhỏ hơn so với số lượng đích của các hình ảnh tham chiếu hoạt động (được báo hiệu

trong tập hợp thông số hình ảnh hoặc phần đầu lát). Khi số lượng mục nhập lớn hơn so với số lượng đích, danh sách này bị cắt.

Sau khi danh sách hình ảnh tham chiếu đã được khởi tạo, nó có thể được cải biến sao cho các hình ảnh tham chiếu cho hình ảnh hiện thời có thể được bố trí theo thứ tự bất kỳ, bao gồm trường hợp mà ở đó một hình ảnh tham chiếu cụ thể có thể xuất hiện trong nhiều hơn một vị trí trong danh sách, dựa vào các lệnh cải biến danh sách hình ảnh tham chiếu. Khi chỉ báo sự có mặt của các cải biến danh sách được thiết đặt đến một, số lượng cố định (bằng số lượng đích của các mục nhập trong danh sách hình ảnh tham chiếu) của các lệnh được báo hiệu, và mỗi lệnh đưa vào một mục nhập cho danh sách hình ảnh tham chiếu. Hình ảnh tham chiếu được nhận dạng trong lệnh này bởi chỉ số đến danh sách của các hình ảnh tham chiếu cho hình ảnh hiện thời được dẫn xuất từ báo hiệu RPS. Điều này khác với việc cải biến danh sách hình ảnh tham chiếu trong H.264/AVC, trong đó hình ảnh được nhận dạng hoặc là bởi số hình ảnh (được dẫn xuất từ phần tử cú pháp `frame_num`) hoặc chỉ số hình ảnh tham chiếu dài hạn, và có khả năng rằng một ít lệnh được cần đến, ví dụ, để hoán vị hai mục nhập thứ nhất của danh sách ban đầu hoặc đưa vào một mục nhập ở vị trí bắt đầu của danh sách ban đầu và dịch chuyển các phần khác.

Danh sách hình ảnh tham chiếu không được cho phép để bao gồm hình ảnh tham chiếu bất kỳ có `TemporalId` lớn hơn hình ảnh hiện thời. Dòng bit HEVC có thể bao gồm một vài lớp con tạm thời. Mỗi đơn vị NAL thuộc về lớp con cụ thể như được chỉ báo bởi `TemporalId` (bằng `temporal_id_plus1 - 1`).

Việc quản lý hình ảnh tham chiếu trực tiếp dựa vào các danh sách hình ảnh tham chiếu. Tài liệu JCT-VC JCTVC-G643 bao gồm cách tiếp cận để trực tiếp sử dụng ba danh sách hình ảnh tham chiếu, danh sách hình ảnh tham chiếu 0, danh sách hình ảnh tham chiếu 1, và danh sách hình ảnh tham chiếu để không, cho việc quản lý các hình ảnh tham chiếu trong DPB, nhờ đó tránh cần đến các quy trình báo hiệu và giải mã bao gồm hoặc là 1) các quy trình của số trượt và MMCO cũng như các quy trình khởi tạo và cải biến danh sách hình ảnh tham chiếu trong AVC, hoặc 2) tập hợp hình ảnh tham chiếu cũng như các quy trình khởi tạo và cải biến danh sách hình ảnh tham chiếu trong HEVC.

Các cách tiếp cận cho việc quản lý hình ảnh tham chiếu có thể có một vài vấn đề. Cách tiếp cận AVC bao gồm các quy trình của số trượt, MMCO, và các quy trình khởi tạo và cải

biến danh sách hình ảnh tham chiếu phức tạp. Ngoài ra, việc mất các hình ảnh có thể dẫn đến mất trạng thái của DPB xét về các hình ảnh đã nằm trong DPB cho mục đích tham khảo dự đoán liên ảnh tiếp. Cách tiếp cận HEVC không có vấn đề mất trạng thái DPB. Tuy nhiên, cách tiếp cận HEVC bao gồm quy trình báo hiệu và dẫn xuất tập hợp hình ảnh tham chiếu phức tạp, cũng như các quy trình khởi tạo và cải biến danh sách hình ảnh tham chiếu phức tạp. Cách tiếp cận trong JCTVC-G643 để trực tiếp sử dụng ba danh sách hình ảnh tham chiếu, danh sách hình ảnh tham chiếu 0, danh sách hình ảnh tham chiếu 1, cũng như danh sách hình ảnh tham chiếu để không, cho việc quản lý các hình ảnh tham chiếu trong DPB bao gồm các khía cạnh sau đây: danh sách hình ảnh tham chiếu thứ ba, tức là, danh sách hình ảnh tham chiếu để không; tạo mã hai phần các chênh lệch POC dưới dạng phần "ngắn hạn" và phần "dài hạn" được tạo mã $ue(v)$; độ chi tiết POC dựa vào TemporalId cho việc tạo mã chênh lệch POC, việc sử dụng tạo mã hai phần các chênh lệch POC để xác định dấu giữa "được sử dụng cho việc tham chiếu ngắn hạn" hoặc "được sử dụng cho việc tham chiếu dài hạn"; phần mô tả tập con danh sách hình ảnh tham chiếu, mà cho phép khả năng định rõ danh sách hình ảnh tham chiếu bằng cách loại bỏ các hình ảnh tham chiếu từ đuôi của phần mô tả danh sách hình ảnh tham chiếu sớm hơn nhất định; chế độ sao chép danh sách hình ảnh tham chiếu được cho phép bởi phần tử cú pháp `ref_pic_list_copy_flag`; và quy trình mô tả danh sách hình ảnh tham chiếu. Mỗi trong số các khía cạnh nêu trên làm cho cách tiếp cận trở nên phức tạp không cần thiết. Ngoài ra, quy trình giải mã cho các danh sách hình ảnh tham chiếu trong JCTVC-G643 cũng phức tạp. Việc báo hiệu các hình ảnh tham chiếu dài hạn có thể cần đến báo hiệu chu kỳ POC trong các phần đầu lát. Điều này không hiệu quả.

Để giải quyết các vấn đề được liệt kê ở trên, được bộc lộ ở đây là các giải pháp sau đây, mỗi trong số này có thể được áp dụng riêng biệt, và một số trong số này có thể được áp dụng kết hợp. 1) Việc đánh dấu hình ảnh tham chiếu trực tiếp dựa vào hai danh sách hình ảnh tham chiếu, nghĩa là danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1. 1a) Thông tin để dẫn xuất hai danh sách hình ảnh tham chiếu được báo hiệu dựa vào các phần tử cú pháp và các cấu trúc cú pháp trong SPS, PPS, và/hoặc phần đầu lát. 1b) Mỗi trong số hai danh sách hình ảnh tham chiếu cho hình ảnh được báo hiệu rõ ràng trong cấu trúc danh sách hình ảnh tham chiếu. 1b.i) Một hoặc nhiều cấu trúc danh sách hình ảnh tham chiếu có thể được báo hiệu trong SPS và mỗi trong số chúng có thể được tham chiếu bởi chỉ số từ

phần đầu lát. 1b.ii) Mỗi trong số danh sách hình ảnh tham chiếu 0 và 1 có thể được báo hiệu trực tiếp trong phần đầu lát. 2) Thông tin để dẫn xuất hai danh sách hình ảnh tham chiếu được báo hiệu cho tất cả các loại lát, tức là, các lát B (dự đoán hai chiều), P (dự đoán một chiều), và I (trong ảnh). Thuật ngữ lát đề cập đến tập hợp của các đơn vị cây tạo mã chẳng hạn như lát trong HEVC hoặc VVC WD cuối cùng; nó có thể cũng đề cập đến một vài tập hợp khác của các đơn vị cây tạo mã chẳng hạn như mảnh trong HEVC. 3) Hai danh sách hình ảnh tham chiếu được tạo ra cho tất cả các loại lát, tức là, các lát B, P, và I. 4) Hai danh sách hình ảnh tham chiếu được xây dựng trực tiếp mà không sử dụng quy trình khởi tạo danh sách hình ảnh tham chiếu và quy trình cải biến danh sách hình ảnh tham chiếu. 5) Trong mỗi trong số hai danh sách hình ảnh tham chiếu, các hình ảnh tham chiếu mà có thể được sử dụng cho việc dự đoán liên ảnh hình ảnh hiện thời có thể chỉ được tham chiếu bởi số lượng mục nhập ở vị trí bắt đầu của danh sách. Các mục nhập này được đề cập đến là các mục nhập hoạt động trong danh sách, trong khi các mục nhập khác được đề cập đến là các mục nhập không hoạt động trong danh sách. Số lượng của tổng các mục nhập và số lượng mục nhập hoạt động trong danh sách đều có thể được dẫn xuất. 6) Hình ảnh được tham chiếu bởi mục nhập không hoạt động trong danh sách hình ảnh tham chiếu không được cho phép để được tham chiếu bởi mục nhập khác trong danh sách hình ảnh tham chiếu hoặc mục nhập bất kỳ trong danh sách hình ảnh tham chiếu khác. 7) Các hình ảnh tham chiếu dài hạn chỉ được nhận dạng bởi một số lượng nhất định các POC LSB, trong đó số lượng này có thể lớn hơn số lượng của các POC LSB được báo hiệu trong các phần đầu lát để dẫn xuất các trị số POC, và số lượng này được chỉ báo trong SPS. 8) Các cấu trúc danh sách hình ảnh tham chiếu được báo hiệu chỉ trong các phần đầu lát, cả các hình ảnh tham chiếu ngắn hạn và các hình ảnh tham chiếu dài hạn được nhận dạng bởi các POC LSB của chúng, mà có thể được thể hiện bởi các số lượng của các bit mà khác với số lượng của các bit được sử dụng để thể hiện các POC LSB được báo hiệu trong các phần đầu lát để dẫn xuất các trị số POC, và các số lượng của các bit được sử dụng để thể hiện các POC LSB để nhận dạng các hình ảnh tham chiếu ngắn hạn và các hình ảnh tham chiếu dài hạn có thể là khác nhau. 9) Các cấu trúc danh sách hình ảnh tham chiếu được báo hiệu chỉ trong các phần đầu lát, không có sự khác biệt được tạo ra giữa các hình ảnh tham chiếu ngắn hạn và dài hạn, tất cả các hình ảnh tham chiếu chỉ là các hình ảnh tham chiếu được đặt tên, và các hình ảnh tham chiếu được nhận dạng bởi các POC LSB của

chúng, mà có thể được thể hiện bởi số lượng của các bit mà khác với số lượng của các bit được sử dụng để thể hiện các POC LSB được báo hiệu trong các phần đầu lát để dẫn xuất các trị số POC.

Phương án thứ nhất của sáng chế được đưa ra. Phần mô tả này liên quan tới VVC WD cuối cùng. Theo phương án này, hai tập của các cấu trúc danh sách hình ảnh tham chiếu được báo hiệu trong SPS, một cho mỗi trong số danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1.

Định nghĩa một vài thuật ngữ được sử dụng trong sáng chế được đưa ra. Hình ảnh điểm truy cập ngẫu nhiên trong ảnh (Intra Random Access Point, viết tắt là IRAP): hình ảnh được tạo mã mà mỗi đơn vị NAL lớp tạo mã video (Video Coding Layer, viết tắt là VCL) có nal_unit_type bằng IRAP_NUT. Hình ảnh không phải IRAP: hình ảnh được tạo mã mà mỗi đơn vị NAL VCL có nal_unit_type bằng NON_IRAP_NUT. Danh sách hình ảnh tham chiếu: danh sách của các hình ảnh tham chiếu được sử dụng cho việc dự đoán liên ảnh của lát P hoặc B. Hai danh sách hình ảnh tham chiếu, danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1, được tạo ra cho mỗi lát của hình ảnh không phải IRAP. Tập hợp các hình ảnh duy nhất được tham chiếu bởi tất cả các mục nhập trong hai danh sách hình ảnh tham chiếu được kết hợp với hình ảnh bao gồm tất cả các hình ảnh tham chiếu mà có thể được sử dụng cho việc dự đoán liên ảnh của hình ảnh kết hợp hoặc hình ảnh bất kỳ tiếp theo hình ảnh kết hợp theo thứ tự giải mã. Để giải mã dữ liệu lát của lát P, chỉ danh sách hình ảnh tham chiếu 0 được sử dụng cho việc dự đoán liên ảnh. Để giải mã dữ liệu lát của lát B, cả hai danh sách hình ảnh tham chiếu được sử dụng cho việc dự đoán liên ảnh. Để giải mã dữ liệu lát của lát I, không có danh sách hình ảnh tham chiếu nào được sử dụng cho việc dự đoán liên ảnh. Hình ảnh tham chiếu dài hạn (LTRP): hình ảnh được đánh dấu là "được sử dụng cho việc tham chiếu dài hạn". Hình ảnh tham chiếu ngắn hạn (STRP): hình ảnh được đánh dấu là "được sử dụng cho việc tham chiếu ngắn hạn".

Các thuật ngữ "được sử dụng cho việc tham chiếu ngắn hạn", "được sử dụng cho việc tham chiếu dài hạn", hoặc "không được sử dụng cho việc tham chiếu" được xác định trong VVC trong phần 8.3.3 quy trình giải mã cho việc đánh dấu hình ảnh tham chiếu, được xác định trong HEVC trong phần 8.3.2 quy trình giải mã cho tập hợp hình ảnh tham chiếu, và được xác định trong AVC trong phần 7.4.3.3 các ngữ nghĩa đánh dấu hình ảnh tham chiếu

được giải mã. Như được sử dụng trong sáng chế, các thuật ngữ có cùng ý nghĩa.

Cú pháp và các ngữ nghĩa liên quan cho phương án thứ nhất được đưa ra dưới đây.

Cú pháp phân đầu đơn vị NAL.

nal_unit_header() {	Ký hiệu mô tả
forbidden_zero_bit	f(1)
nal_unit_type	u(5)
nuh_temporal_id_plus1	u(3)
nuh_reserved_zero_7bits	u(7)
}	

Cú pháp tải tin chuỗi byte chưa xử lý (RBSP) tập hợp thông số chuỗi.

seq_parameter_set_rbsp() {	Ký hiệu mô tả
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
 separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)

long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag)	
additional_lt_poc_lsb	ue(v)
for(i = 0; i < 2; 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, long_term_ref_pics_flag)	
}	
rbsp_trailing_bits()	
}	

Cú pháp tập hợp thông số hình ảnh RBSP.

pic_parameter_set_rbsp() {	Ký hiệu mô tả
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rbsp_trailing_bits()	
}	

Cú pháp phần đầu lát.

slice_header() {	Ký hiệu mô tả
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	

slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
}	
if(slice_type == P slice_type == B) {	
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++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Cú pháp cấu trúc danh sách hình ảnh tham chiếu.

ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) {	Ký hiệu mô tả
num_strp_entries[listIdx][rplIdx]	ue(v)
if(ltrpFlag)	
num_ltrp_entries[listIdx][rplIdx]	ue(v)
for(i = 0; i < NumEntriesInList[listIdx][rplIdx]; i++) {	
if(num_ltrp_entries[listIdx][rplIdx] > 0)	
lt_ref_pic_flag[listIdx][rplIdx][i]	
if(!lt_ref_pic_flag[listIdx][rplIdx][i])	

delta_poc_st [listIdx][rplsIdx][i] // không sử dụng “delta_poc_st_minus1” ở đây, để cho phép cùng hình ảnh tham chiếu được bao gồm trong nhiều vị trí của danh sách hình ảnh tham chiếu. Như một hiệu quả phụ, điều này cũng cho phép tham chiếu hình ảnh hiện thời (tức là, bản sao khối trong ảnh). Cũng lưu ý rằng nó đã được ký.	se(v)
else	
poc_lsb_lt [listIdx][rplsIdx][i]	u(v)
}	
}	

Các ngữ nghĩa phần đầu đơn vị NAL.

forbidden_zero_bit sẽ bằng 0. nal_unit_type định rõ loại của cấu trúc dữ liệu RBSP được chứa trong đơn vị NAL.

Bảng 7-1 – các mã loại đơn vị NAL và các lớp loại đơn vị NAL

nal_unit_type	Tên của nal_unit_type	Nội dung của đơn vị NAL và cấu trúc cú pháp RBSP	Lớp loại đơn vị NAL
0	NON_IRAP_NUT	Phân đoạn lát được tạo mã của hình ảnh không phải IRAP slice_layer_rbsp()	VCL
1	IRAP_NUT	Lát được tạo mã của hình ảnh IRAP slice_layer_rbsp()	VCL
2-15	RSV_VCL_NUT	Các đơn vị NAL VCL dành riêng	VCL
16	SPS_NUT	Tập hợp thông số chuỗi seq_parameter_set_rbsp()	không phải VCL

17	PPS_NUT	Tập hợp thông số hình ảnh pic_parameter_set_rbsp()	không phải VCL
18	EOS_NUT	Phần cuối của chuỗi end_of_seq_rbsp()	không phải VCL
19	EOB_NUT	Phần cuối của dòng bit end_of_bitstream_rbsp()	không phải VCL
20, 21	PREFIX_SEI_NUT SUFFIX_SEI_NUT	Thông tin nâng cao bổ sung sei_rbsp()	không phải VCL
22-26	RSV_NVCL	Dành riêng	không phải VCL
27-31	UNSPEC	Không được định rõ	không phải VCL

nuh_temporal_id_plus1 trừ 1 định rõ ký hiệu nhận dạng tạm thời cho đơn vị NAL. Trị số của nuh_temporal_id_plus1 sẽ không bằng 0. Biến TemporalId được định rõ như dưới đây: $\text{TemporalId} = \text{nuh_temporal_id_plus1} - 1$. Khi nal_unit_type bằng IRAP_NUT, lát được tạo mã thuộc về hình ảnh IRAP, TemporalId sẽ bằng 0. Trị số của TemporalId sẽ giống nhau đối với tất cả các đơn vị NAL VCL của đơn vị truy cập. Trị số của TemporalId của hình ảnh được tạo mã hoặc đơn vị truy cập là trị số của TemporalId của các đơn vị NAL VCL của hình ảnh được tạo mã hoặc đơn vị truy cập. Trị số của TemporalId cho các đơn vị NAL không phải VCL bị ràng buộc như dưới đây: Nếu nal_unit_type bằng SPS_NUT, thì TemporalId sẽ bằng 0 và TemporalId của đơn vị truy cập chứa đơn vị NAL sẽ bằng 0. Nếu không thì nếu nal_unit_type bằng EOS_NUT hoặc EOB_NUT, thì TemporalId sẽ bằng 0. Nếu không thì,

TemporalId sẽ lớn hơn hoặc bằng TemporalId của đơn vị truy cập chứa đơn vị NAL. Khi đơn vị NAL là đơn vị NAL không phải VCL, trị số của TemporalId bằng trị số nhỏ nhất của các trị số TemporalId của tất cả các đơn vị truy cập mà đơn vị NAL không phải VCL áp dụng. Khi nal_unit_type bằng PPS_NUT, TemporalId có thể lớn hơn hoặc bằng TemporalId của đơn vị truy cập chứa, vì tất cả các tập hợp thông số hình ảnh (Picture Parameter Set, viết tắt là PPS) có thể được bao gồm trong phần đầu của dòng bit, trong đó hình ảnh được tạo mã thứ nhất có TemporalId bằng 0. Khi nal_unit_type bằng PREFIX_SEI_NUT hoặc SUFFIX_SEI_NUT, TemporalId có thể lớn hơn hoặc bằng TemporalId của đơn vị truy cập chứa, vì đơn vị NAL SEI có thể chứa thông tin áp dụng cho tập con dòng bit bao gồm các đơn vị truy cập mà các trị số TemporalId lớn hơn TemporalId của đơn vị truy cập chứa đơn vị NAL SEI. nuh_reserved_zero_7bits sẽ bằng '0000000'. Các trị số khác của nuh_reserved_zero_7bits có thể được định rõ trong tương lai bởi ITU-T | ISO/IEC. Các bộ giải mã sẽ bỏ qua (tức là, loại bỏ khỏi dòng bit và loại trừ) các đơn vị NAL có các trị số của nuh_reserved_zero_7bits không bằng '0000000'.

Các ngữ nghĩa tập hợp thông số chuỗi RBSP.

$\log_2_max_pic_order_cnt_lsb_minus4$ định rõ trị số của biến MaxPicOrderCntLsb được sử dụng trong quy trình giải mã đối với số đếm thứ tự hình ảnh như dưới đây: $MaxPicOrderCntLsb = 2^{(\log_2_max_pic_order_cnt_lsb_minus4 + 4)}$. Trị số của $\log_2_max_pic_order_cnt_lsb_minus4$ sẽ nằm trong dải từ 0 đến 12. $sps_max_dec_pic_buffering_minus1$ cộng 1 định rõ kích thước yêu cầu lớn nhất của bộ đệm hình ảnh được giải mã cho CVS trong các đơn vị của các bộ đệm lưu trữ hình ảnh. Trị số của $sps_max_dec_pic_buffering_minus1$ sẽ nằm trong dải từ 0 đến $MaxDpbSize - 1$, trong đó $MaxDpbSize$ như được định rõ ở phần khác. $long_term_ref_pics_flag$ bằng 0 định rõ rằng không có LTRP được sử dụng cho việc dự đoán liên ảnh của hình ảnh được tạo mã bất kỳ trong CVS. $long_term_ref_pics_flag$ bằng 1 định rõ rằng các LTRP có thể được sử dụng cho việc dự đoán liên ảnh của một hoặc nhiều hình ảnh được tạo mã trong CVS. $additional_lt_poc_lsb$ định rõ trị số của biến MaxLtPicOrderCntLsb được sử dụng trong quy trình giải mã cho các danh sách hình ảnh tham chiếu như dưới đây: $MaxLtPicOrderCntLsb = 2^{(\log_2_max_pic_order_cnt_lsb_minus4 + 4 + additional_lt_poc_lsb)}$. Trị số của $additional_lt_poc_lsb$ sẽ nằm trong dải từ 0 đến

$32 - \log_2_max_pic_order_cnt_lsb_minus4 - 4$. Khi không có mặt, trị số của `additional_lt_poc_lsb` được suy ra bằng 0. `num_ref_pic_lists_in_sps[ i ]` định rõ số lượng của các cấu trúc cú pháp `ref_pic_list_struct( listIdx, rplIdx, ltrpFlag )` có `listIdx` bằng `i` được bao gồm trong SPS. Trị số của `num_ref_pic_lists_in_sps[ i ]` sẽ nằm trong dải từ 0 đến 64. Đối với mỗi trị số của `listIdx` (bằng 0 hoặc 1), bộ giải mã sẽ cấp phát bộ nhớ cho tổng số lượng của các cấu trúc cú pháp `num_ref_pic_lists_in_sps[ i ] + 1` `ref_pic_list_struct( listIdx, rplIdx, ltrpFlag )` vì có thể là một cấu trúc cú pháp `ref_pic_list_struct( listIdx, rplIdx, ltrpFlag )` được báo hiệu trực tiếp trong các phần đầu lát của hình ảnh hiện thời.

Các ngữ nghĩa tập hợp thông số hình ảnh RBSP.

`num_ref_idx_default_active_minus1[ i ]` cộng 1, khi `i` bằng 0, định rõ trị số được suy ra của biến `NumRefIdxActive[ 0 ]` đối với các lát P hoặc B có `num_ref_idx_active_override_flag` bằng 0, và, khi `i` bằng 1, định rõ trị số được suy ra của `NumRefIdxActive[ 1 ]` đối với các lát B có `num_ref_idx_active_override_flag` bằng 0. Trị số của `num_ref_idx_default_active_minus1[ i ]` sẽ nằm trong dải từ 0 đến 14.

Các ngữ nghĩa phần đầu lát.

Khi có mặt, trị số của mỗi trong số các phần tử cú pháp phần đầu lát `slice_pic_parameter_set_id` và `slice_pic_order_cnt_lsb` sẽ giống nhau trong tất cả các phần đầu lát của hình ảnh được tạo mã. ... `slice_type` định rõ loại tạo mã của lát theo bảng 7-3.

Bảng 7-3 – Tên của `slice_type`

<code>slice_type</code>	Tên của <code>slice_type</code>
0	B (lát B)
1	P (lát P)
2	I (lát I)

Khi `nal_unit_type` bằng `IRAP_NUT`, tức là, hình ảnh là hình ảnh IRAP, `slice_type` sẽ bằng 2. ... `slice_pic_order_cnt_lsb` định rõ số đếm thứ tự hình ảnh modulo `MaxPicOrderCntLsb` cho hình ảnh hiện thời. Độ dài của phần tử cú pháp `slice_pic_order_cnt_lsb` là $\log_2_max_pic_order_cnt_lsb_minus4 + 4$ bit. Trị số của `slice_pic_order_cnt_lsb` sẽ nằm trong dải từ 0 đến `MaxPicOrderCntLsb - 1`. Khi

slice_pic_order_cnt_lsb không có mặt, slice_pic_order_cnt_lsb được suy ra bằng 0. ref_pic_list_sps_flag[i] bằng 1 định rõ rằng danh sách hình ảnh tham chiếu i của hình ảnh hiện thời được dẫn xuất dựa vào một trong số các cấu trúc cú pháp ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) có listIdx bằng i trong SPS hoạt động. ref_pic_list_sps_flag[i] bằng 0 định rõ rằng danh sách hình ảnh tham chiếu i của hình ảnh hiện thời được dẫn xuất dựa vào cấu trúc cú pháp ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) có listIdx bằng i được bao gồm trực tiếp trong các phần đầu lát của hình ảnh hiện thời. Khi num_ref_pic_lists_in_sps[i] bằng 0, trị số của ref_pic_list_sps_flag[i] sẽ bằng 0. ref_pic_list_idx[i] định rõ chỉ số, vào danh sách của các cấu trúc cú pháp ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) có listIdx bằng i được bao gồm trong SPS hoạt động, của cấu trúc cú pháp ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) có listIdx bằng i được sử dụng cho việc dẫn xuất danh sách hình ảnh tham chiếu i của hình ảnh hiện thời. Phần tử cú pháp ref_pic_list_idx[i] được thể hiện bởi $\text{Ceil}(\text{Log2}(\text{num_ref_pic_lists_in_sps}[i]))$ bit. Khi không có mặt, trị số của ref_pic_list_idx[i] được suy ra bằng 0. Trị số của ref_pic_list_idx[i] sẽ nằm trong dải từ 0 đến num_ref_pic_lists_in_sps[i] - 1. num_ref_idx_active_override_flag bằng 1 định rõ rằng phần tử cú pháp num_ref_idx_active_minus1[0] có mặt đối với các lát P và B và phần tử cú pháp num_ref_idx_active_minus1[1] là có mặt đối với các lát B. num_ref_idx_active_override_flag bằng 0 định rõ rằng các phần tử cú pháp num_ref_idx_active_minus1[0] và num_ref_idx_active_minus1[1] không có mặt. num_ref_idx_active_minus1[i], khi có mặt, định rõ trị số của biến NumRefIdxActive[i] như dưới đây: $\text{NumRefIdxActive}[i] = \text{num_ref_idx_active_minus1}[i] + 1$. Trị số của num_ref_idx_active_minus1[i] sẽ nằm trong dải từ 0 đến 14.

Trị số của NumRefIdxActive[i] - 1 định rõ chỉ số tham chiếu lớn nhất đối với danh sách hình ảnh tham chiếu i mà có thể được sử dụng để giải mã lát. Khi trị số của NumRefIdxActive[i] bằng 0, không có chỉ số tham chiếu đối với danh sách hình ảnh tham chiếu i có thể được sử dụng để giải mã lát. Đối với i bằng 0 hoặc 1, khi lát hiện thời là lát B và num_ref_idx_active_override_flag bằng 0, NumRefIdxActive[i] được suy ra bằng num_ref_idx_default_active_minus1[i] + 1. Khi lát hiện thời là lát P và num_ref_idx_active_override_flag bằng 0, NumRefIdxActive[0] được suy ra bằng num_ref_idx_default_active_minus1[0] + 1. Khi lát hiện thời là lát P,

NumRefIdxActive[1] được suy ra bằng 0. Khi lát hiện thời là lát I, cả NumRefIdxActive[0] và NumRefIdxActive[1] được suy ra bằng 0.

Theo cách khác, đối với i bằng 0 hoặc 1, áp dụng phần dưới đây sau phần nêu trên:
Đặt $rplsIdx1$ bằng $ref_pic_list_sps_flag[i] ? ref_pic_list_idx[i] :$
 $num_ref_pic_lists_in_sps[i]$, và $numRpEntries[i]$ bằng
 $num_strp_entries[i][rplsIdx1] + num_ltrp_entries[i][rplsIdx1]$. Khi
NumRefIdxActive[i] lớn hơn $numRpEntries[i]$, trị số của NumRefIdxActive[i] được thiết
đặt bằng $numRpEntries[i]$

Các ngữ nghĩa cấu trúc danh sách hình ảnh tham chiếu.

Cấu trúc cú pháp $ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)$ có thể có mặt trong SPS hoặc trong phần đầu lát. Phụ thuộc vào liệu cấu trúc cú pháp có được bao gồm trong phần đầu lát hoặc SPS hay không, áp dụng phần dưới đây: Nếu có mặt trong phần đầu lát, thì cấu trúc cú pháp $ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)$ định rõ danh sách hình ảnh tham chiếu ListIdx của hình ảnh hiện thời (hình ảnh chứa lát). Nếu không thì (có mặt trong SPS), cấu trúc cú pháp $ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)$ định rõ ứng viên đối với danh sách hình ảnh tham chiếu ListIdx, và thuật ngữ “hình ảnh hiện thời” trong các ngữ nghĩa được định rõ trong phần còn lại của phần này đề cập đến mỗi hình ảnh mà 1) có một hoặc nhiều lát chứa $ref_pic_list_idx[listIdx]$ bằng chỉ số trong danh sách của các cấu trúc cú pháp $ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)$ được bao gồm trong SPS, và 2) nằm trong CVS mà có SPS là SPS hoạt động. $num_strp_entries[listIdx][rplsIdx]$ định rõ số lượng mục nhập STRP trong cấu trúc cú pháp $ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)$. $num_ltrp_entries[listIdx][rplsIdx]$ định rõ số lượng mục nhập LTRP trong cấu trúc cú pháp $ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)$. Khi không có mặt, trị số của $num_ltrp_entries[listIdx][rplsIdx]$ được suy ra bằng 0. Biến NumEntriesInList[$listIdx$][$rplsIdx$] được dẫn xuất như dưới đây:
 $NumEntriesInList[listIdx][rplsIdx] = num_strp_entries[listIdx][rplsIdx] + num_ltrp_entries[listIdx][rplsIdx]$. Trị số của NumEntriesInList[$listIdx$][$rplsIdx$] sẽ nằm trong dải từ 0 đến $sps_max_dec_pic_buffering_minus1$. $lt_ref_pic_flag[listIdx][rplsIdx][i]$ bằng 1 định rõ rằng mục nhập thứ i trong cấu trúc cú pháp $ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)$ là mục nhập LTRP.

$lt_ref_pic_flag[listIdx][rplsIdx][i]$ bằng 0 định rõ rằng mục nhập thứ i trong cấu trúc cú pháp $ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)$ là mục nhập STRP. Khi không có mặt, trị số của $lt_ref_pic_flag[listIdx][rplsIdx][i]$ được suy ra bằng 0. Yêu cầu về sự tương thích dòng bit là tổng của $lt_ref_pic_flag[listIdx][rplsIdx][i]$ cho tất cả các trị số của i nằm trong dải từ 0 đến $NumEntriesInList[listIdx][rplsIdx] - 1$, sẽ bằng $num_ltrp_entries[listIdx][rplsIdx]. delta_poc_st[listIdx][rplsIdx][i]$, khi mục nhập thứ i là mục nhập STRP thứ nhất trong cấu trúc cú pháp $ref_pic_list_struct(rplsIdx, ltrpFlag)$, định rõ sự khác biệt giữa các trị số số đếm thứ tự hình ảnh của hình ảnh hiện thời và hình ảnh được tham chiếu bởi mục nhập thứ i , hoặc, khi mục nhập thứ i là mục nhập STRP nhưng không phải mục nhập STRP thứ nhất trong cấu trúc cú pháp $ref_pic_list_struct(rplsIdx, ltrpFlag)$, định rõ sự khác biệt giữa các trị số số đếm thứ tự hình ảnh của các hình ảnh được tham chiếu bởi mục nhập thứ i và bởi mục nhập STRP trước đó trong cấu trúc cú pháp $ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)$. Trị số của $delta_poc_st[listIdx][rplsIdx][i]$ sẽ nằm trong dải từ -215 đến $215 - 1$. $poc_lsb_lt[listIdx][rplsIdx][i]$ định rõ trị số của số đếm thứ tự hình ảnh modulo $MaxLtPicOrderCntLsb$ của hình ảnh được tham chiếu bởi mục nhập thứ i trong cấu trúc cú pháp $ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)$. Độ dài của phần tử cú pháp $poc_lsb_lt[listIdx][rplsIdx][i]$ là $\log_2(MaxLtPicOrderCntLsb)$ bit.

Quy trình giải mã được thảo luận. Quy trình giải mã hoạt động như dưới đây cho hình ảnh hiện thời $CurrPic$. Việc giải mã các đơn vị NAL được định rõ dưới đây. Các quy trình dưới đây định rõ các quy trình giải mã tiếp theo sử dụng các phần tử cú pháp trong phần đầu lát lớp và ở trên. Các biến và các chức năng liên quan đến số đếm thứ tự hình ảnh được dẫn xuất. Điều này cần phải được gọi ra chỉ cho lát thứ nhất của hình ảnh. Ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP, quy trình giải mã cho việc xây dựng các danh sách hình ảnh tham chiếu được gọi ra cho việc dẫn xuất danh sách hình ảnh tham chiếu 0 ($RefPicList[0]$) và danh sách hình ảnh tham chiếu 1 ($RefPicList[1]$). Quy trình giải mã cho việc đánh dấu hình ảnh tham chiếu được gọi ra, trong đó các hình ảnh tham chiếu có thể được đánh dấu là "không được sử dụng cho việc tham chiếu" hoặc "được sử dụng cho việc tham chiếu dài hạn". Điều này cần phải được gọi ra chỉ cho lát thứ nhất của hình ảnh. Các quy trình giải mã để tạo mã các đơn vị cây, định tỷ lệ, biến đổi, lọc trong vòng

lập, v.v., được gọi ra. Sau khi tất cả các lát của hình ảnh hiện thời đã được giải mã, hình ảnh được giải mã hiện thời được đánh dấu là "được sử dụng cho việc tham chiếu ngắn hạn".

Quy trình giải mã đơn vị NAL được thảo luận. Các đầu vào quy trình này là các đơn vị NAL của hình ảnh hiện thời và các đơn vị NAL không phải VCL được kết hợp của chúng. Các đầu ra của quy trình này là các cấu trúc cú pháp RBSP được phân tích được gói gọn trong các đơn vị NAL. Quy trình giải mã cho mỗi đơn vị NAL trích xuất cấu trúc cú pháp RBSP từ đơn vị NAL và sau đó phân tích cấu trúc cú pháp RBSP.

Quy trình giải mã lát được thảo luận, bao gồm quy trình giải mã cho số đếm thứ tự hình ảnh. Đầu ra của quy trình này là PicOrderCntVal, số đếm thứ tự hình ảnh của hình ảnh hiện thời. Các số đếm thứ tự hình ảnh được sử dụng để nhận dạng các hình ảnh, để dẫn xuất các thông số chuyển động trong chế độ hợp nhất và dự đoán vector chuyển động, và để kiểm tra độ tương thích của bộ giải mã. Mỗi hình ảnh được tạo mã được kết hợp với biến số đếm thứ tự hình ảnh, được thể hiện là PicOrderCntVal. Khi hình ảnh hiện thời không phải hình ảnh IRAP, các biến prevPicOrderCntLsb và prevPicOrderCntMsb được dẫn xuất như dưới đây: Đặt prevTid0Pic là hình ảnh trước đó theo thứ tự giải mã mà có TemporalId bằng 0. Biến prevPicOrderCntLsb được thiết đặt bằng slice_pic_order_cnt_lsb của prevTid0Pic. Biến prevPicOrderCntMsb được thiết đặt bằng PicOrderCntMsb của prevTid0Pic.

Biến PicOrderCntMsb của hình ảnh hiện thời được dẫn xuất như dưới đây: Nếu hình ảnh hiện thời là hình ảnh IRAP, thì PicOrderCntMsb được thiết đặt bằng 0. Nếu không thì, PicOrderCntMsb được dẫn xuất như dưới đây:

```

if( ( slice_pic_order_cnt_lsb < prevPicOrderCntLsb ) &&
    ( ( prevPicOrderCntLsb - slice_pic_order_cnt_lsb ) >= ( MaxPicOrderCntLsb
/
    2
    )
    )
    PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb
else if( ( slice_pic_order_cnt_lsb > prevPicOrderCntLsb ) &&
    ( ( slice_pic_order_cnt_lsb - prevPicOrderCntLsb ) > ( MaxPicOrderCntLsb /
2
    )
    )
    PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb
else
    PicOrderCntMsb = prevPicOrderCntMsb

```

PicOrderCntVal được dẫn xuất như dưới đây: $\text{PicOrderCntVal} = \text{PicOrderCntMsb} + \text{slice_pic_order_cnt_lsb}$.

Tất cả các hình ảnh IRAP sẽ có PicOrderCntVal bằng 0 vì slice_pic_order_cnt_lsb được suy ra là bằng 0 đối với các hình ảnh IRAP và prevPicOrderCntLsb và prevPicOrderCntMsb đều được thiết đặt bằng 0. Trị số của PicOrderCntVal sẽ nằm trong dải từ -231 đến $231 - 1$. Trong một CVS, các trị số PicOrderCntVal đối với hai hình ảnh được tạo mã bất kỳ sẽ không giống nhau.

Ở thời điểm bất kỳ trong quy trình giải mã, các trị số của PicOrderCntVal & $(\text{MaxLtPicOrderCntLsb} - 1)$ đối với hai hình ảnh tham chiếu bất kỳ trong DPB sẽ không giống nhau. Hàm PicOrderCnt(picX) được định rõ như dưới đây: PicOrderCnt(picX) = PicOrderCntVal của hình ảnh picX. Hàm DiffPicOrderCnt(picA, picB) được định rõ như dưới đây: $\text{DiffPicOrderCnt}(\text{picA}, \text{picB}) = \text{PicOrderCnt}(\text{picA}) - \text{PicOrderCnt}(\text{picB})$. Dòng bit sẽ không chứa dữ liệu mà dẫn đến các trị số của DiffPicOrderCnt(picA, picB) được sử dụng trong quy trình giải mã không nằm trong dải từ -215 đến $215 - 1$. Đặt X là hình ảnh hiện thời và Y và Z là hai hình ảnh khác trong cùng chuỗi video được tạo mã (CVS), Y và Z được xem là nằm trong cùng hướng thứ tự đưa ra từ X khi cả DiffPicOrderCnt(X, Y) và DiffPicOrderCnt(X, Z) là dương hoặc đều là âm.

Quy trình giải mã cho việc xây dựng các danh sách hình ảnh tham chiếu được thảo luận. Quy trình này được gọi ra ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP. Các hình ảnh tham chiếu được gửi qua các chỉ số tham chiếu. Chỉ số tham chiếu là chỉ số trong danh sách hình ảnh tham chiếu. Khi giải mã lát I, không có danh sách hình ảnh tham chiếu nào được sử dụng trong việc giải mã dữ liệu lát. Khi giải mã lát P, chỉ danh sách hình ảnh tham chiếu 0 (tức là, RefPicList[0]), được sử dụng trong việc giải mã dữ liệu lát. Khi giải mã lát B, cả danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1 (tức là, RefPicList[1]) được sử dụng trong việc giải mã dữ liệu lát. Ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP, các danh sách hình ảnh tham chiếu RefPicList[0] và RefPicList[1] được dẫn xuất. Các danh sách hình ảnh tham chiếu được sử dụng trong việc đánh dấu các hình ảnh tham chiếu hoặc trong việc giải mã dữ liệu lát. Đối với lát I của hình ảnh không phải IRAP mà không phải là lát thứ nhất của hình

ảnh, RefPicList[0] và RefPicList[1] có thể được dẫn xuất cho mục đích kiểm tra độ tương thích dòng bit, nhưng việc suy ra của chúng không cần thiết cho việc giải mã hình ảnh hiện thời hoặc các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Đối với lát P mà không phải là lát thứ nhất của hình ảnh, RefPicList[1] có thể được dẫn xuất cho mục đích kiểm tra độ tương thích dòng bit, nhưng sự dẫn xuất của nó không cần thiết cho việc giải mã hình ảnh hiện thời hoặc các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Các danh sách hình ảnh tham chiếu RefPicList[0] và RefPicList[1] được xây dựng như dưới đây:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplsIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplsIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplsIdx[ i ] ]; j++ )
    {
        if( !lt_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplsIdx[ i ] ][ j ]
            if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
            RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = “no reference picture”
            pocBase = RefPicPocList[ i ][ j ]
        } else {
            if( there is a reference picture picA in the DPB with
            PicOrderCntVal & ( MaxLtPicOrderCntLsb - 1 )
                equal to poc_lsb_lt[ i ][ RplsIdx[ i ] ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = “no reference picture”
        }
    }
}

```

```

    }
  }
}

```

Đối với mỗi i bằng 0 hoặc 1, áp dụng phần dưới đây: Các mục nhập $\text{NumRefIdxActive}[i]$ thứ nhất trong $\text{RefPicList}[i]$ được đề cập đến là các mục nhập hoạt động trong $\text{RefPicList}[i]$, và các mục nhập khác trong $\text{RefPicList}[i]$ được đề cập đến là các mục nhập không hoạt động trong $\text{RefPicList}[i]$. Mỗi mục nhập trong $\text{RefPicList}[i][j]$ đối với j nằm trong dải từ 0 đến $\text{NumEntriesInList}[i][\text{RplIdx}[i]] - 1$, được đề cập đến là mục nhập STRP nếu $\text{lt_ref_pic_flag}[i][\text{RplIdx}[i]][j]$ bằng 0, và nếu không thì là mục nhập LTRP. Có khả năng rằng hình ảnh cụ thể được tham chiếu bởi cả mục nhập trong $\text{RefPicList}[0]$ và mục nhập trong $\text{RefPicList}[1]$. Cũng có thể rằng hình ảnh cụ thể được tham chiếu bởi nhiều hơn một mục nhập trong $\text{RefPicList}[0]$ hoặc bởi nhiều hơn một mục nhập trong $\text{RefPicList}[1]$. Các mục nhập hoạt động trong $\text{RefPicList}[0]$ và các mục nhập hoạt động trong $\text{RefPicList}[1]$ đề cập chung tới tất cả các hình ảnh tham chiếu mà có thể được sử dụng cho việc dự đoán liên ảnh của hình ảnh hiện thời và một hoặc nhiều hình ảnh mà tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Các mục nhập không hoạt động trong $\text{RefPicList}[0]$ và các mục nhập không hoạt động trong $\text{RefPicList}[1]$ đề cập chung tới tất cả các hình ảnh tham chiếu không được sử dụng cho việc dự đoán liên ảnh của hình ảnh hiện thời nhưng có thể được sử dụng trong việc dự đoán liên ảnh cho một hoặc nhiều hình ảnh mà tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Có thể là một hoặc nhiều mục nhập trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ mà bằng "không có hình ảnh tham chiếu" bởi vì các hình ảnh tương ứng không có mặt trong DPB. Mỗi mục nhập không hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ bằng "không có hình ảnh tham chiếu" sẽ được bỏ qua. Sự mất hình ảnh vô tình sẽ được suy ra cho mỗi mục nhập hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ mà bằng "không có hình ảnh tham chiếu".

Yêu cầu về sự tương thích dòng bit áp dụng các ràng buộc sau đây: Đối với mỗi i bằng 0 hoặc 1, $\text{NumEntriesInList}[i][\text{RplIdx}[i]]$ sẽ không nhỏ hơn $\text{NumRefIdxActive}[i]$. Hình ảnh được tham chiếu bởi mỗi mục nhập hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ sẽ có mặt trong DPB và sẽ có TemporalId nhỏ hơn hoặc bằng TemporalId của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: chỉ số mục nhập của mục

nhập không hoạt động bất kỳ trong RefPicList[0] hoặc RefPicList[1] sẽ không được sử dụng làm chỉ số tham chiếu để giải mã của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: mục nhập không hoạt động trong RefPicList[0] hoặc RefPicList[1] sẽ không tham chiếu đến hình ảnh giống như mục nhập khác bất kỳ trong RefPicList[0] hoặc RefPicList[1]. Mục nhập STRP trong RefPicList[0] hoặc RefPicList[1] của lát của hình ảnh và mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1] của cùng lát hoặc lát khác nhau của cùng hình ảnh sẽ không tham chiếu đến cùng hình ảnh. Chính hình ảnh hiện thời sẽ không được tham chiếu bởi mục nhập bất kỳ trong RefPicList[0] hoặc RefPicList[1]. Sẽ không có mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1] mà sự khác biệt giữa PicOrderCntVal của hình ảnh hiện thời và PicOrderCntVal của hình ảnh được tham chiếu bởi mục nhập lớn hơn hoặc bằng 224. Đặt setOfRefPics là tập hợp các hình ảnh duy nhất được tham chiếu bởi tất cả các mục nhập trong RefPicList[0] và tất cả các mục nhập trong RefPicList[1]. Số lượng của các hình ảnh trong setOfRefPics sẽ nhỏ hơn hoặc bằng sps_max_dec_pic_buffering_minus1 và setOfRefPics sẽ giống nhau đối với tất cả các lát của hình ảnh.

Quy trình giải mã cho việc đánh dấu hình ảnh tham chiếu.

Quy trình này được gọi ra một lần cho mỗi hình ảnh, sau khi giải mã phần đầu lát và quy trình giải mã cho việc xây dựng danh sách hình ảnh tham chiếu cho lát, nhưng trước việc giải mã dữ liệu lát. Quy trình này có thể dẫn đến một hoặc nhiều hình ảnh tham chiếu trong DPB được đánh dấu là "không được sử dụng cho việc tham chiếu" hoặc "được sử dụng cho việc tham chiếu dài hạn". Hình ảnh được giải mã trong DPB có thể được đánh dấu là "không được sử dụng cho việc tham chiếu", "được sử dụng cho việc tham chiếu ngắn hạn", hoặc "được sử dụng cho việc tham chiếu dài hạn", nhưng chỉ một trong số ba lựa chọn này ở thời điểm xác định bất kỳ trong suốt hoạt động của quy trình giải mã. Việc gán một trong số các dấu này cho hình ảnh loại bỏ hoàn toàn các sự đánh dấu còn lại khi thích hợp. Khi hình ảnh được đề cập đến là được đánh dấu là "được sử dụng cho việc tham chiếu", điều này đề cập chung đến hình ảnh được đánh dấu là "được sử dụng cho việc tham chiếu ngắn hạn" hoặc "được sử dụng cho việc tham chiếu dài hạn" (nhưng không phải là cả hai). Khi hình ảnh hiện thời là hình ảnh IRAP, tất cả các hình ảnh tham chiếu hiện trong DPB (nếu có) được đánh dấu là "không được sử dụng cho việc tham chiếu". Các STRP được nhận dạng bởi các trị số

PicOrderCntVal của chúng. Các LTRP được nhận dạng bởi $\text{Log2}(\text{MaxLtPicOrderCntLsb})$ LSB của các trị số PicOrderCntVal của chúng. Áp dụng phần dưới đây: đối với mỗi mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1], khi hình ảnh được tham chiếu là STRP, hình ảnh được đánh dấu là "được sử dụng cho việc tham chiếu dài hạn". Mỗi hình ảnh tham chiếu trong DPB mà không được tham chiếu bởi mục nhập bất kỳ trong RefPicList[0] hoặc RefPicList[1] được đánh dấu là "không được sử dụng cho việc tham chiếu".

Phần mô tả chi tiết của phương án thứ hai của sáng chế được đưa ra. Phần này mô tả phương án thứ hai của sáng chế như được nêu trên. Phần mô tả này liên quan tới VVC WD cuối cùng. Theo phương án này, một tập hợp các cấu trúc danh sách hình ảnh tham chiếu được báo hiệu trong SPS, được chia sẻ bởi danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1.

Cú pháp tập hợp thông số chuỗi RBSP.

seq_parameter_set_rbsp() {	Ký hiệu mô tả
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)

log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
num_ref_pic_lists_in_sps	ue(v)
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag)	
additional_lt_poc_lsb	ue(v)
for(i = 0; i < num_ref_pic_lists_in_sps; i++)	
ref_pic_list_struct(i, long_term_ref_pics_flag)	
rbsp_trailing_bits()	
}	

Cú pháp tập hợp thông số hình ảnh RBSP.

pic_parameter_set_rbsp() {	Ký hiệu mô tả
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rbsp_trailing_bits()	
}	

Cú pháp phân đầu lát.

slice_header() {	Ký hiệu mô tả
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)

if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps > 1)	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(num_ref_pic_lists_in_sps + i, long_term_ref_pics_flag)	
}	
if(slice_type == P slice_type == B) {	
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++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Cú pháp cấu trúc danh sách hình ảnh tham chiếu.

ref_pic_list_struct(rplsIdx, ltrpFlag) {	Ký hiệu mô tả
num_strp_entries[rplsIdx]	ue(v)

if(ltrpFlag)	
num_ltrp_entries [rplsIdx]	ue(v)
for(i = 0; i < NumEntriesInList[rplsIdx]; i++) {	
if(num_ltrp_entries[rplsIdx] > 0)	
lt_ref_pic_flag [rplsIdx][i]	
if(!lt_ref_pic_flag[rplsIdx][i])	
delta_poc_st [rplsIdx][i] // không có _minus1 ở đây, để cho phép cùng hình ảnh tham chiếu được bao gồm trong nhiều vị trí của danh sách hình ảnh tham chiếu. Như một hiệu quả phụ, nó cũng cho phép tham chiếu hình ảnh hiện thời (tức là, bản sao khối trong ảnh). Cũng lưu ý rằng nó đã được ký.	se(v)
else	
poc_lsb_lt [rplsIdx][i]	u(v)
}	
}	

Các ngữ nghĩa phần đầu đơn vị NAL được thảo luận.

Các ngữ nghĩa tập hợp thông số chuỗi RBSP.

$\log_2_max_pic_order_cnt_lsb_minus4$ định rõ trị số của biến $MaxPicOrderCntLsb$ được sử dụng trong quy trình giải mã đối với số đếm thứ tự hình ảnh như dưới đây: $MaxPicOrderCntLsb = 2(\log_2_max_pic_order_cnt_lsb_minus4 + 4)$. Trị số của $\log_2_max_pic_order_cnt_lsb_minus4$ sẽ nằm trong dải từ 0 đến 12. $sps_max_dec_pic_buffering_minus1$ cộng 1 định rõ kích thước yêu cầu lớn nhất của bộ đệm hình ảnh được giải mã cho CVS trong các đơn vị của các bộ đệm lưu trữ hình ảnh. Trị số của $sps_max_dec_pic_buffering_minus1$ sẽ nằm trong dải từ 0 đến $MaxDpbSize - 1$, trong đó $MaxDpbSize$ như được định rõ ở phần khác. $num_ref_pic_lists_in_sps$ định rõ số lượng của các cấu trúc cú pháp $ref_pic_list_struct(rplsIdx, ltrpFlag)$ được bao gồm trong SPS. Trị số của $num_ref_pic_lists_in_sps$ sẽ nằm trong dải từ 0 đến 128. Bộ giải mã sẽ cấp phát bộ nhớ cho tổng số lượng của các cấu trúc cú pháp $num_short_term_ref_pic_sets + 2$

`ref_pic_list_struct( rplsIdx, ltrpFlag )` vì có thể là hai cấu trúc cú pháp `ref_pic_list_struct( rplsIdx, ltrpFlag )` được báo hiệu trực tiếp trong các phần đầu lát của hình ảnh hiện thời. `long_term_ref_pics_flag` bằng 0 định rõ rằng không có LTRP được sử dụng cho việc dự đoán liên ảnh của hình ảnh được tạo mã bất kỳ trong CVS. `long_term_ref_pics_flag` bằng 1 định rõ rằng các LTRP có thể được sử dụng cho việc dự đoán liên ảnh của một hoặc nhiều hình ảnh được tạo mã trong CVS. `additional_lt_poc_lsb` định rõ trị số của biến `MaxLtPicOrderCntLsb` được sử dụng trong quy trình giải mã cho các danh sách hình ảnh tham chiếu như dưới đây:

$$\text{MaxLtPicOrderCntLsb} = 2(\log_2_max_pic_order_cnt_lsb_minus4 + 4 + \text{additional_lt_poc_lsb})$$

trị số của `additional_lt_poc_lsb` sẽ nằm trong dải từ 0 đến $32 - \log_2_max_pic_order_cnt_lsb_minus4 - 4$. Khi không có mặt, trị số của `additional_lt_poc_lsb` được suy ra bằng 0.

Các ngữ nghĩa tập hợp thông số hình ảnh RBSP được thảo luận.

Các ngữ nghĩa phần đầu lát.

Khi có mặt, trị số của mỗi trong số các phần tử cú pháp phần đầu lát `slice_pic_parameter_set_id` và `slice_pic_order_cnt_lsb` sẽ giống nhau trong tất cả các phần đầu lát của hình ảnh được tạo mã. `slice_type` định rõ loại tạo mã của lát theo bảng 7-3.

Bảng 7-3 – Tên của `slice_type`

<code>slice_type</code>	Tên của <code>slice_type</code>
0	B (lát B)
1	P (lát P)
2	I (lát I)

Khi `nal_unit_type` bằng `IRAP_NUT`, tức là, hình ảnh là hình ảnh IRAP, `slice_type` sẽ bằng 2. ... `slice_pic_order_cnt_lsb` định rõ số đếm thứ tự hình ảnh modulo `MaxPicOrderCntLsb` cho hình ảnh hiện thời. Độ dài của phần tử cú pháp `slice_pic_order_cnt_lsb` là $\log_2_max_pic_order_cnt_lsb_minus4 + 4$ bit. Trị số của `slice_pic_order_cnt_lsb` sẽ nằm trong dải từ 0 đến `MaxPicOrderCntLsb - 1`. Khi `slice_pic_order_cnt_lsb` không có mặt, `slice_pic_order_cnt_lsb` được suy ra bằng 0.

$\text{ref_pic_list_sps_flag}[i]$ bằng 1 định rõ rằng danh sách hình ảnh tham chiếu i của hình ảnh hiện thời được dẫn xuất dựa vào một trong số cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$ trong SPS hoạt động. $\text{ref_pic_list_sps_flag}[i]$ bằng 0 định rõ rằng danh sách hình ảnh tham chiếu i của hình ảnh hiện thời được dẫn xuất dựa vào cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$ được bao gồm trực tiếp trong các phần đầu lát của hình ảnh hiện thời. Khi $\text{num_ref_pic_lists_in_sps}$ bằng 0, trị số của $\text{ref_pic_list_sps_flag}[i]$ sẽ bằng 0. $\text{ref_pic_list_idx}[i]$ định rõ chỉ số, vào danh sách của các cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$ được bao gồm trong SPS hoạt động, của cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$ được sử dụng cho việc dẫn xuất danh sách hình ảnh tham chiếu i của hình ảnh hiện thời. Phần tử cú pháp $\text{ref_pic_list_idx}[i]$ được thể hiện bởi $\text{Ceil}(\text{Log2}(\text{num_ref_pic_lists_in_sps}))$ bit. Khi không có mặt, trị số của $\text{ref_pic_list_idx}[i]$ được suy ra bằng 0. Trị số của $\text{ref_pic_list_idx}[i]$ sẽ nằm trong dải từ 0 đến $\text{num_ref_pic_lists_in_sps} - 1$. $\text{num_ref_idx_active_override_flag}$ bằng 1 định rõ rằng phần tử cú pháp $\text{num_ref_idx_active_minus1}[0]$ là có mặt đối với các lát P và B và phần tử cú pháp $\text{num_ref_idx_active_minus1}[1]$ là có mặt đối với các lát B. $\text{num_ref_idx_active_override_flag}$ bằng 0 định rõ rằng các phần tử cú pháp $\text{num_ref_idx_active_minus1}[0]$ và $\text{num_ref_idx_active_minus1}[1]$ không có mặt.

$\text{num_ref_idx_active_minus1}[i]$, khi có mặt, định rõ trị số của biến $\text{NumRefIdxActive}[i]$ như dưới đây: $\text{NumRefIdxActive}[i] = \text{num_ref_idx_active_minus1}[i] + 1$. Trị số của $\text{num_ref_idx_active_minus1}[i]$ sẽ nằm trong dải từ 0 đến 14. Trị số của $\text{NumRefIdxActive}[i] - 1$ định rõ chỉ số tham chiếu lớn nhất đối với danh sách hình ảnh tham chiếu i mà có thể được sử dụng để giải mã lát. Khi trị số của $\text{NumRefIdxActive}[i]$ bằng 0, không có chỉ số tham chiếu đối với danh sách hình ảnh tham chiếu i có thể được sử dụng để giải mã lát. Đối với i bằng 0 hoặc 1, khi lát hiện thời là lát B và $\text{num_ref_idx_active_override_flag}$ bằng 0, $\text{NumRefIdxActive}[i]$ được suy ra bằng $\text{num_ref_idx_default_active_minus1}[i] + 1$. Khi lát hiện thời là lát P và $\text{num_ref_idx_active_override_flag}$ bằng 0, $\text{NumRefIdxActive}[0]$ được suy ra bằng $\text{num_ref_idx_default_active_minus1}[0] + 1$. Khi lát hiện thời là lát P, $\text{NumRefIdxActive}[1]$ được suy ra bằng 0. Khi lát hiện thời là lát I, cả $\text{NumRefIdxActive}[0]$ và $\text{NumRefIdxActive}[1]$ được suy ra bằng 0.

Theo cách khác, đối với i bằng 0 hoặc 1, áp dụng phần dưới đây sau phần nêu trên:
 Đặt rplsIdx1 bằng $\text{ref_pic_list_sps_flag}[i] ? \text{ref_pic_list_idx}[i] :$
 $\text{num_ref_pic_lists_in_sps}[i]$, và $\text{numRpEntries}[i]$ bằng
 $\text{num_strp_entries}[i][\text{rplsIdx1}] + \text{num_ltrp_entries}[i][\text{rplsIdx1}]$. Khi
 $\text{NumRefIdxActive}[i]$ lớn hơn $\text{numRpEntries}[i]$, trị số của $\text{NumRefIdxActive}[i]$ được thiết
 đặt bằng $\text{numRpEntries}[i]$.

Các ngữ nghĩa cấu trúc danh sách hình ảnh tham chiếu.

Cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$ có thể có mặt trong SPS hoặc trong phần đầu lát. Phụ thuộc vào liệu cấu trúc cú pháp có được bao gồm trong phần đầu lát hoặc SPS hay không, áp dụng phần dưới đây: Nếu có mặt trong phần đầu lát, thì cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$ định rõ danh sách hình ảnh tham chiếu của hình ảnh hiện thời (hình ảnh chứa lát). Nếu không thì (có mặt trong SPS), cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$ định rõ ứng viên danh sách hình ảnh tham chiếu, và thuật ngữ “hình ảnh hiện thời” trong các ngữ nghĩa được định rõ trong phần còn lại của phần này đề cập đến mỗi hình ảnh mà 1) có một hoặc nhiều lát chứa $\text{ref_pic_list_idx}[i]$ bằng chỉ số trong danh sách của các cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$ được bao gồm trong SPS, và 2) nằm trong CVS mà có SPS là SPS hoạt động. $\text{num_strp_entries}[\text{rplsIdx}]$ định rõ số lượng mục nhập STRP trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$. $\text{num_ltrp_entries}[\text{rplsIdx}]$ định rõ số lượng mục nhập LTRP trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$. Khi không có mặt, trị số của $\text{num_ltrp_entries}[\text{rplsIdx}]$ được suy ra bằng 0.

Biến $\text{NumEntriesInList}[\text{rplsIdx}]$ được dẫn xuất như dưới đây:
 $\text{NumEntriesInList}[\text{rplsIdx}] = \text{num_strp_entries}[\text{rplsIdx}] + \text{num_ltrp_entries}[\text{rplsIdx}]$. Trị số của $\text{NumEntriesInList}[\text{rplsIdx}]$ sẽ nằm trong dải từ 0 đến $\text{sps_max_dec_pic_buffering_minus1}$. $\text{lt_ref_pic_flag}[\text{rplsIdx}][i]$ bằng 1 định rõ rằng mục nhập thứ i trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$ là mục nhập LTRP. $\text{lt_ref_pic_flag}[\text{rplsIdx}][i]$ bằng 0 định rõ rằng mục nhập thứ i trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$ là mục nhập STRP. Khi không có mặt, trị số của $\text{lt_ref_pic_flag}[\text{rplsIdx}][i]$ được suy ra bằng 0. Yêu cầu về sự tương thích dòng bit là tổng của $\text{lt_ref_pic_flag}[\text{rplsIdx}][i]$ cho tất cả các trị số của i nằm trong dải từ 0 đến

$\text{NumEntriesInList}[\text{rplsIdx}] - 1$, sẽ bằng $\text{num_ltrp_entries}[\text{rplsIdx}]$. $\text{delta_poc_st}[\text{rplsIdx}][i]$, khi mục nhập thứ i là mục nhập STRP thứ nhất trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$, định rõ sự khác biệt giữa các trị số số đếm thứ tự hình ảnh của hình ảnh hiện thời và hình ảnh được tham chiếu bởi mục nhập thứ i , hoặc, khi mục nhập thứ i là mục nhập STRP nhưng không phải mục nhập STRP thứ nhất trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$, định rõ sự khác biệt giữa các trị số số đếm thứ tự hình ảnh của các hình ảnh được tham chiếu bởi mục nhập thứ i và bởi mục nhập STRP trước đó trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$. Trị số của $\text{delta_poc_st}[\text{rplsIdx}][i]$ sẽ nằm trong dải từ 0 đến $215 - 1$. $\text{poc_lsb_lt}[\text{rplsIdx}][i]$ định rõ trị số của số đếm thứ tự hình ảnh modulo $\text{MaxLtPicOrderCntLsb}$ của hình ảnh được tham chiếu bởi mục nhập thứ i trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{rplsIdx}, \text{ltrpFlag})$. Độ dài của phần tử cú pháp $\text{poc_lsb_lt}[\text{rplsIdx}][i]$ là $\text{Log2}(\text{MaxLtPicOrderCntLsb})$ bit.

Quy trình giải mã chung được định rõ là áp dụng một phần của phần mô tả chi tiết của phương án thứ nhất của sáng chế. Quy trình giải mã đơn vị NAL được mô tả. Quy trình giải mã đơn vị NAL được định rõ là áp dụng một phần của phần mô tả chi tiết của phương án thứ nhất của sáng chế.

Quy trình giải mã lát được đưa ra.

Quy trình giải mã cho số đếm thứ tự hình ảnh.

Quy trình giải mã cho số đếm thứ tự hình ảnh được định rõ là áp dụng một phần của phần mô tả chi tiết của phương án thứ nhất của sáng chế.

Quy trình giải mã cho việc xây dựng các danh sách hình ảnh tham chiếu.

Quy trình này được gọi ra ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP. Các hình ảnh tham chiếu được gửi qua các chỉ số tham chiếu. Chỉ số tham chiếu là chỉ số trong danh sách hình ảnh tham chiếu. Khi giải mã lát I, không có danh sách hình ảnh tham chiếu nào được sử dụng trong việc giải mã dữ liệu lát. Khi giải mã lát P, chỉ danh sách hình ảnh tham chiếu 0 (tức là, $\text{RefPicList}[0]$) được sử dụng trong việc giải mã dữ liệu lát. Khi giải mã lát B, cả danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1 (tức là, $\text{RefPicList}[1]$) được sử dụng trong việc giải mã dữ liệu lát. Ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP, các danh sách hình ảnh tham chiếu $\text{RefPicList}[0]$ và $\text{RefPicList}[1]$ được dẫn xuất. Các danh sách hình ảnh tham

chiều được sử dụng trong việc đánh dấu các hình ảnh tham chiếu hoặc trong việc giải mã dữ liệu lát. Đối với lát I của hình ảnh không phải IRAP mà không phải là lát thứ nhất của hình ảnh, RefPicList[0] và RefPicList[1] có thể được dẫn xuất cho mục đích kiểm tra độ tương thích dòng bit, nhưng việc suy ra của chúng không cần thiết cho việc giải mã hình ảnh hiện thời hoặc các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Đối với lát P mà không phải là lát thứ nhất của hình ảnh, RefPicList[1] có thể được dẫn xuất cho mục đích kiểm tra độ tương thích dòng bit, nhưng sự dẫn xuất của nó không cần thiết cho việc giải mã hình ảnh hiện thời hoặc các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã.

Các danh sách hình ảnh tham chiếu RefPicList[0] và RefPicList[1] được xây dựng như dưới đây:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplsIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplsIdx[ i ] = num_ref_pic_lists_in_sps
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ RplsIdx[ i ] ]; j++ ) {
        if( !lt_ref_pic_flag[ RplsIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ RplsIdx[ i ] ][ j ]
            if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
            RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "no reference picture"
            pocBase = RefPicPocList[ i ][ j ]
        } else {
            if( there is a reference picA in the DPB with
            PicOrderCntVal & ( MaxLtPicOrderCntLsb - 1 )
            equal to poc_lsb_lt[ RplsIdx[ i ] ][ j ] )
                RefPicList[ i ][ j ] = picA
            else

```

```

RefPicList[ i ][ j ] = “no reference picture”
}
}
}

```

Đối với mỗi i bằng 0 hoặc 1, áp dụng phần dưới đây: các mục nhập $\text{NumRefIdxActive}[i]$ thứ nhất trong $\text{RefPicList}[i]$ được đề cập đến là các mục nhập hoạt động trong $\text{RefPicList}[i]$, và các mục nhập khác trong $\text{RefPicList}[i]$ được đề cập đến là các mục nhập không hoạt động trong $\text{RefPicList}[i]$. Mỗi mục nhập $\text{RefPicList}[i][j]$ đối với j nằm trong dải từ 0 đến $\text{NumEntriesInList}[\text{RplsIdx}[i]] - 1$, được đề cập đến là mục nhập STRP nếu $\text{lt_ref_pic_flag}[\text{RplsIdx}[i]][j]$ bằng 0, và nếu không thì là mục nhập LTRP. Có khả năng rằng hình ảnh cụ thể được tham chiếu bởi cả mục nhập trong $\text{RefPicList}[0]$ và mục nhập trong $\text{RefPicList}[1]$. Cũng có thể rằng hình ảnh cụ thể được tham chiếu bởi nhiều hơn một mục nhập trong $\text{RefPicList}[0]$ hoặc bởi nhiều hơn một mục nhập trong $\text{RefPicList}[1]$. Các mục nhập hoạt động trong $\text{RefPicList}[0]$ và các mục nhập hoạt động trong $\text{RefPicList}[1]$ đề cập chung tới tất cả các hình ảnh tham chiếu mà có thể được sử dụng cho việc dự đoán liên ảnh của hình ảnh hiện thời và một hoặc nhiều hình ảnh mà tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Các mục nhập không hoạt động trong $\text{RefPicList}[0]$ và các mục nhập không hoạt động trong $\text{RefPicList}[1]$ đề cập chung tới tất cả các hình ảnh tham chiếu không được sử dụng cho việc dự đoán liên ảnh của hình ảnh hiện thời nhưng có thể được sử dụng trong việc dự đoán liên ảnh cho một hoặc nhiều hình ảnh mà tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Có thể là một hoặc nhiều mục nhập trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ mà bằng “không có hình ảnh tham chiếu” bởi vì các hình ảnh tương ứng không có mặt trong DPB. Mỗi mục nhập không hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ bằng “không có hình ảnh tham chiếu” sẽ được bỏ qua. Sự mất hình ảnh vô tình sẽ được suy ra cho mỗi mục nhập hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ mà bằng “không có hình ảnh tham chiếu”.

Yêu cầu về sự tương thích dòng bit áp dụng các ràng buộc sau đây: Đối với mỗi i bằng 0 hoặc 1, $\text{NumEntriesInList}[\text{RplsIdx}[i]]$ sẽ không nhỏ hơn $\text{NumRefIdxActive}[i]$. Hình ảnh được tham chiếu bởi mỗi mục nhập hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ sẽ có mặt trong DPB và sẽ có TemporalId nhỏ hơn hoặc bằng TemporalId của hình ảnh hiện

thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: chỉ số mục nhập của mục nhập không hoạt động bất kỳ trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` sẽ không được sử dụng làm chỉ số tham chiếu để giải mã của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: mục nhập không hoạt động trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` sẽ không tham chiếu đến hình ảnh giống như mục nhập khác bất kỳ trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]`. Mục nhập STRP trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` của lát của hình ảnh và mục nhập LTRP trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` của cùng lát hoặc lát khác nhau của cùng hình ảnh sẽ không tham chiếu đến cùng hình ảnh. Chính hình ảnh hiện thời sẽ không được tham chiếu bởi mục nhập bất kỳ trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]`. Sẽ không có mục nhập LTRP trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` mà sự khác biệt giữa `PicOrderCntVal` của hình ảnh hiện thời và `PicOrderCntVal` của hình ảnh được tham chiếu bởi mục nhập lớn hơn hoặc bằng 224. Đặt `setOfRefPics` là tập hợp các hình ảnh duy nhất được tham chiếu bởi tất cả các mục nhập trong `RefPicList[ 0 ]` và tất cả các mục nhập trong `RefPicList[ 1 ]`. Số lượng của các hình ảnh trong `setOfRefPics` sẽ nhỏ hơn hoặc bằng `sps_max_dec_pic_buffering_minus1` và `setOfRefPics` sẽ giống nhau đối với tất cả các lát của hình ảnh.

Quy trình giải mã cho việc đánh dấu hình ảnh tham chiếu được thảo luận.

Quy trình này được gọi ra một lần cho mỗi hình ảnh, sau khi giải mã phần đầu lát và quy trình giải mã cho việc xây dựng danh sách hình ảnh tham chiếu cho lát, nhưng trước việc giải mã dữ liệu lát. Quy trình này có thể dẫn đến một hoặc nhiều hình ảnh tham chiếu trong DPB được đánh dấu là "không được sử dụng cho việc tham chiếu" hoặc "được sử dụng cho việc tham chiếu dài hạn". Hình ảnh được giải mã trong DPB có thể được đánh dấu là "không được sử dụng cho việc tham chiếu", "được sử dụng cho việc tham chiếu ngắn hạn" hoặc "được sử dụng cho việc tham chiếu dài hạn", nhưng chỉ một trong số ba lựa chọn này ở thời điểm xác định bất kỳ trong suốt hoạt động của quy trình giải mã. Việc gán một trong số các dấu này cho hình ảnh loại bỏ hoàn toàn các sự đánh dấu còn lại khi thích hợp. Khi hình ảnh được đề cập đến là được đánh dấu là "được sử dụng cho việc tham chiếu", điều này đề cập chung đến hình ảnh được đánh dấu là "được sử dụng cho việc tham chiếu ngắn hạn" hoặc "được sử dụng cho việc tham chiếu dài hạn" (nhưng không phải là cả hai). Khi hình ảnh hiện thời là hình ảnh IRAP, tất cả các hình ảnh tham chiếu hiện trong DPB (nếu có) được đánh

dấu là "không được sử dụng cho việc tham chiếu". Các STRP được nhận dạng bởi các trị số PicOrderCntVal của chúng. Các LTRP được nhận dạng bởi $\text{Log}_2(\text{MaxLtpPicOrderCntLsb})$ LSB của các trị số PicOrderCntVal của chúng.

Áp dụng phần dưới đây: đối với mỗi mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1], khi hình ảnh được tham chiếu là STRP, hình ảnh được đánh dấu là "được sử dụng cho việc tham chiếu dài hạn". Mỗi hình ảnh tham chiếu trong DPB mà không được tham chiếu bởi mục nhập bất kỳ trong RefPicList[0] hoặc RefPicList[1] được đánh dấu là "không được sử dụng cho việc tham chiếu".

Fig.5 là phương án về phương pháp 500 để giải mã dòng bit video được tạo mã được thực hiện bởi bộ giải mã video (ví dụ, bộ giải mã video 30). Phương pháp 500 có thể được thực hiện sau khi dòng bit được giải mã đã được thu trực tiếp hoặc gián tiếp từ bộ mã hóa video (ví dụ, bộ mã hóa video 20). Phương pháp 500 có thể được thực hiện để nâng cao quy trình giải mã (ví dụ, làm cho quy trình giải mã hiệu quả hơn, nhanh hơn, v.v., so với các quy trình giải mã thông thường) bởi vì số lượng mục nhập hoạt động trong danh sách hình ảnh tham chiếu được thiết đặt bằng số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu khi số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu lớn hơn số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu. Điều này trái ngược với cách mà trong đó danh sách hình ảnh tham chiếu được xử lý trong HEVC và AVC. Do đó, là một vấn đề thực tế, hiệu suất của codec có thể được nâng cao, mà mang đến trải nghiệm của người dùng tốt hơn.

Ở khối 502, cấu trúc danh sách hình ảnh tham chiếu được thu nhận cho lát hiện thời được thể hiện trong dòng bit video được tạo mã. Theo một phương án, cấu trúc danh sách hình ảnh tham chiếu chứa số lượng mục nhập. Theo một phương án, thứ tự của các mục nhập trong cấu trúc danh sách hình ảnh tham chiếu là giống như thứ tự của các hình ảnh tham chiếu tương ứng trong danh sách hình ảnh tham chiếu. Theo một phương án, thứ tự là từ không đến trị số được chỉ báo. Theo một phương án, trị số được chỉ báo là từ không đến trị số được chỉ báo bởi `sps_max_dec_pic_buffering_minus1`.

Ở khối 504, số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu được thu nhận cho lát hiện thời. Theo một phương án, danh sách hình ảnh tham chiếu được định rõ là RefPictList[0] hoặc RefPictList[1].

Ở khối 506, danh sách hình ảnh tham chiếu được xây dựng cho lát hiện thời. Theo một phương án, danh sách hình ảnh tham chiếu chứa số lượng mục nhập hoạt động và số lượng mục nhập không hoạt động.

Ở khối 508, số lượng mục nhập hoạt động trong danh sách hình ảnh tham chiếu được thiết đặt bằng số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu khi số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu lớn hơn số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu.

Ở khối 510, ít nhất một khối được xây dựng lại của lát hiện thời được thu nhận dựa vào ít nhất một mục nhập hoạt động của danh sách hình ảnh tham chiếu. Theo một phương án, ít nhất một khối được xây dựng lại được sử dụng để tạo ra ảnh được hiển thị trên bộ hiển thị của thiết bị điện tử.

Theo một phương án, danh sách hình ảnh tham chiếu bao gồm danh sách của các hình ảnh tham chiếu được sử dụng cho việc dự đoán liên ảnh. Theo một phương án, việc dự đoán liên ảnh là cho lát P hoặc cho lát B.

Theo một phương án, phần đầu lát chứa cờ tập hợp thông số chuỗi (Sequence Parameter Set, viết tắt là SPS) danh sách hình ảnh tham chiếu được định rõ là `ref_pic_list_sps_flag[ i ]`. Khi cờ này bằng 1, danh sách hình ảnh tham chiếu thứ *i*, tức là, `RefPicList[ i ]`, không được báo hiệu trực tiếp trong phần đầu lát nhưng được tham chiếu từ SPS. Khi cờ này bằng 0, danh sách hình ảnh tham chiếu thứ *i*, tức là, `RefPicList[ i ]`, được báo hiệu trực tiếp trong phần đầu lát, không được tham chiếu từ SPS. Theo một phương án, phần đầu lát chứa cờ ghi đè chủ động chỉ số tham chiếu số được định rõ bởi `num_ref_idx_active_override_flag`. Khi cờ này bằng 1, số lượng mục nhập hoạt động trong mỗi danh sách hình ảnh tham chiếu là trị số mặc định được báo hiệu trong PPS. Khi cờ này bằng 0, số lượng mục nhập hoạt động trong mỗi danh sách hình ảnh tham chiếu được báo hiệu rõ ràng trong phần đầu lát.

Theo một phương án, danh sách hình ảnh tham chiếu được định rõ là `RefPicList[0]` hoặc `RefPicList[1]`, và thứ tự của các mục nhập trong cấu trúc danh sách hình ảnh tham chiếu là giống như thứ tự của các hình ảnh tham chiếu tương ứng trong danh sách hình ảnh tham chiếu.

Tóm lược về các phương án khác dựa vào các phương án thứ nhất và thứ hai được

đưa ra.

Phần này đưa ra các tóm lược ngắn gọn về các phương án khác của sáng chế. Các phần tóm lược này liên quan tới phần mô tả của phương án thứ nhất. Tuy nhiên, khái niệm cơ bản của sáng chế đối với các phương án khác cũng thích hợp cho cách thực hiện sáng chế đối với phương án thứ hai. Cách thực hiện như vậy theo cùng tinh thần về cách các khía cạnh được thực hiện trên phương án thứ nhất.

Các ngữ nghĩa của delta POC của các mục nhập hình ảnh tham chiếu ngắn hạn.

Theo một phương án khác của sáng chế, ngữ nghĩa của phần tử cú pháp định rõ delta POC của mục nhập thứ i trong cấu trúc danh sách hình ảnh tham chiếu `ref_pic_list_struct()` được xác định là chênh lệch POC giữa hình ảnh hiện thời và hình ảnh tham chiếu được kết hợp với mục nhập thứ i. Một số phần mô tả được sử dụng trong sáng chế có liên quan tới dự thảo chuẩn hiện thời (ví dụ, dự thảo VVC) trong đó chỉ delta được thể hiện hoặc được mô tả. Văn bản đã xóa được chỉ báo bởi đường gạch ngang và văn bản được bổ sung bất kỳ được đánh dấu nổi bật.

Ngữ nghĩa của `delta_poc_st[listIdx][rplsIdx][i]` được xác định như dưới đây: `delta_poc_st[listIdx][rplsIdx][i]` định rõ sự khác biệt giữa các trị số số đếm thứ tự hình ảnh của hình ảnh hiện thời và hình ảnh được tham chiếu bởi mục nhập thứ i. Trị số của `delta_poc_st[listIdx][rplsIdx][i]` sẽ nằm trong dải từ -215 đến 215 - 1.

Phương trình trong quy trình xây dựng danh sách hình ảnh tham chiếu cần phải được cập nhật. Các danh sách hình ảnh tham chiếu `RefPicList[0]` và `RefPicList[1]` được xây dựng như dưới đây:

```
for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplsIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplsIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplsIdx[ i ] ]; j++ )
    {
        if( !lt_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] =
```

```

pocBasePicOrderCntVal – delta_poc_st[ i ][ RplIdx[ i ] ][ j ]
    if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
RefPicPocList[ i ][ j ] )
        RefPicList[ i ][ j ] = picA
    else
        RefPicList[ i ][ j ] = “no reference picture”
    pocBase = RefPicPocList[ i ][ j ]
} else {
    if( there is a reference picA in the DPB with
PicOrderCntVal & ( MaxLtPicOrderCntLsb – 1 )
        equal to poc_lsb_lt[ i ][ RplIdx[ i ] ][ j ] )
        RefPicList[ i ][ j ] = picA
    else
        RefPicList[ i ][ j ] = “no reference picture”
}
}
}

```

Việc báo hiệu các mục nhập hình ảnh tham chiếu dài hạn.

Theo một phương án khác của sáng chế, các mục nhập hình ảnh tham chiếu dài hạn không được báo hiệu trong cùng cấu trúc danh sách hình ảnh tham chiếu chứa các mục nhập hình ảnh tham chiếu ngắn hạn. Các mục nhập hình ảnh tham chiếu dài hạn được báo hiệu trong cấu trúc riêng biệt và đối với mỗi mục nhập trong cấu trúc có phần tử cú pháp mô tả vị trí dự định của mục nhập hình ảnh tham chiếu dài hạn để dẫn xuất chỉ số mục nhập tương ứng trong danh sách hình ảnh tham chiếu cuối cùng.

Cú pháp tập hợp thông số chuỗi RBSP.

seq_parameter_set_rbsp() {	Ký hiệu mô tả
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)

if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag)	
additional_lt_poc_lsb	ue(v)
for(i = 0; i < 2; 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)	
}	
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag) {	
additional_lt_poc_lsb	ue(v)
num_ref_pic_lists_lt_in_sps	ue(v)

}	
for(i = 0; i < num_ref_pic_lists_lt_in_sps; i++)	
ref_pic_list_lt_struct(i)	
rbsp_trailing_bits()	
}	

Cú pháp phân đầu lát.

slice_header() {	Ký hiệu mô tả
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
if(long_term_ref_pics_flag)	
ref_pic_list_lt_idx[i]	u(v)
} else {	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
if(long_term_ref_pics_flag)	

ref_pic_list_lt_struct(num_ref_pic_lists_lt_in_sps + 1)	
}	
}	
if(slice_type == P slice_type == B) {	
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++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Cú pháp cấu trúc danh sách hình ảnh tham chiếu.

ref_pic_list_struct(listIdx, rplsIdx) {	Ký hiệu mô tả
num_strp_entries [listIdx][rplsIdx]	ue(v)
for(i = 0; i < NumEntriesInList[listIdx][rplsIdx]; i++) {	
delta_poc_st [listIdx][rplsIdx][i]	se(v)
}	
}	

Cú pháp cấu trúc danh sách hình ảnh tham chiếu dài hạn.

ref_pic_list_lt_struct(ltRplsIdx) {	Ký hiệu mô tả
num_ltrp_entries [ltRplsIdx]	ue(v)
for(i = 0; i < num_ltrp_entries[ltRplsIdx]; i++) {	

poc_lsb_lt [ltRplsIdx][i]	u(v)
lt_pos_idx [ltRplsIdx][i] // định rõ chỉ số của LT ref pic này trong RPL này	u(v)
}	
}	

Các ngữ nghĩa tập hợp thông số chuỗi RBSP.

num_ref_pic_lists_lt_in_sps định rõ số lượng của các cấu trúc cú pháp **ref_pic_list_lt_struct(ltRplsIdx)** được bao gồm trong SPS. Trị số của **num_ref_pic_lists_lt_in_sps** sẽ nằm trong dải từ 0 đến 64. Khi không có mặt, trị số của **num_ref_pic_lists_lt_in_sps** được suy ra bằng 0.

Các ngữ nghĩa phần đầu lát.

ref_pic_list_lt_idx[i] định rõ chỉ số trong danh sách của các cấu trúc cú pháp **ref_pic_list_lt_struct(ltRplsIdx)** được bao gồm trong SPS hoạt động mà được sử dụng cho việc dẫn xuất danh sách hình ảnh tham chiếu *i* của hình ảnh hiện thời. Phần tử cú pháp **ref_pic_list_lt_idx[i]** được thể hiện bởi $\text{Ceil}(\text{Log2}(\text{num_ref_pic_lists_lt_in_sps}))$ bit. Trị số của **ref_pic_list_lt_idx** sẽ nằm trong dải từ 0 đến **num_ref_pic_lists_lt_in_sps** – 1.

Các ngữ nghĩa cấu trúc danh sách hình ảnh tham chiếu.

Cấu trúc cú pháp **ref_pic_list_struct(listIdx, rplsIdx)** có thể có mặt trong SPS hoặc trong phần đầu lát. Phụ thuộc vào việc liệu cấu trúc cú pháp có được bao gồm trong phần đầu lát hoặc SPS hay không, áp dụng phần dưới đây: nếu có mặt trong phần đầu lát, thì cấu trúc cú pháp **ref_pic_list_struct(listIdx, rplsIdx)** định rõ danh sách hình ảnh tham chiếu ngắn hạn **listIdx** của hình ảnh hiện thời (hình ảnh chứa lát). Nếu không thì (có mặt trong SPS), cấu trúc cú pháp **ref_pic_list_struct(listIdx, rplsIdx)** định rõ ứng viên cho danh sách hình ảnh tham chiếu ngắn hạn **listIdx**, và thuật ngữ “hình ảnh hiện thời” trong các ngữ nghĩa được định rõ trong phần còn lại của phần này đề cập đến mỗi hình ảnh mà 1) có một hoặc nhiều lát chứa **ref_pic_list_idx[listIdx]** bằng chỉ số trong danh sách của các cấu trúc cú pháp **ref_pic_list_struct(listIdx, rplsIdx)** được bao gồm trong SPS, và 2) nằm trong CVS mà có SPS là SPS hoạt động. **num_strp_entries[listIdx][rplsIdx]** định rõ số lượng mục nhập STRP trong cấu trúc cú pháp **ref_pic_list_struct(listIdx, rplsIdx)**.

$\text{num_ltrp_entries}[\text{listIdx}][\text{rplsIdx}]$ định rõ số lượng mục nhập LTRP trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx}, \text{ltrpFlag})$. Khi không có mặt, trị số của $\text{num_ltrp_entries}[\text{listIdx}][\text{rplsIdx}]$ được suy ra bằng 0.

Biến $\text{NumEntriesInList}[\text{listIdx}][\text{rplsIdx}]$ được dẫn xuất như dưới đây:

$$\text{NumRefPicEntriesInRpl}[\text{listIdx}][\text{rplsIdx}] = \text{num_strp_entries}[\text{listIdx}][\text{rplsIdx}] + \text{num_ltrp_entries}[\text{listIdx}][\text{rplsIdx}] \quad (7-34)$$

Trị số của $\text{NumRefPicEntries}[\text{listIdx}][\text{rplsIdx}]$ sẽ nằm trong dải từ 0 đến $\text{sps_max_dec_pic_buffering_minus1}$.

$\text{lt_ref_pic_flag}[\text{listIdx}][\text{rplsIdx}][i]$ bằng 1 định rõ rằng mục nhập thứ i trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx}, \text{ltrpFlag})$ là mục nhập LTRP. $\text{lt_ref_pic_flag}[\text{listIdx}][\text{rplsIdx}][i]$ bằng 0 định rõ rằng mục nhập thứ i trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx}, \text{ltrpFlag})$ là mục nhập STRP. Khi không có mặt, trị số của $\text{lt_ref_pic_flag}[\text{listIdx}][\text{rplsIdx}][i]$ được suy ra bằng 0.

Yêu cầu về sự tương thích dòng bit là tổng của $\text{lt_ref_pic_flag}[\text{listIdx}][\text{rplsIdx}][i]$ đối với tất cả các trị số của i nằm trong dải từ 0 đến $\text{NumRefPicEntries}[\text{listIdx}][\text{rplsIdx}] - 1$, sẽ bằng $\text{num_ltrp_entries}[\text{listIdx}][\text{rplsIdx}]$.

$\text{delta_poc_st}[\text{listIdx}][\text{rplsIdx}][i]$, khi mục nhập thứ i là mục nhập STRP thứ nhất trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx})$, định rõ sự khác biệt giữa các trị số đếm thứ tự hình ảnh của hình ảnh hiện thời và hình ảnh được tham chiếu bởi mục nhập thứ i , hoặc, khi mục nhập thứ i là mục nhập STRP nhưng không phải mục nhập STRP thứ nhất trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx})$, định rõ sự khác biệt giữa các trị số đếm thứ tự hình ảnh của các hình ảnh được tham chiếu bởi mục nhập thứ i và bởi mục nhập STRP trước đó trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx})$. Trị số của $\text{delta_poc_st}[\text{listIdx}][\text{rplsIdx}][i]$ sẽ nằm trong dải từ -215 đến $215 - 1$.

$\text{poc_lsb_lt}[\text{listIdx}][\text{rplsIdx}][i]$ định rõ trị số của số đếm thứ tự hình ảnh modulo $\text{MaxLtPicOrderCntLsb}$ của hình ảnh được tham chiếu bởi mục nhập thứ i trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx}, \text{ltrpFlag})$. Độ dài của phần tử cú pháp $\text{poc_lsb_lt}[\text{listIdx}][\text{rplsIdx}][i]$ là $\text{Log2}(\text{MaxLtPicOrderCntLsb})$ bit.

Các ngữ nghĩa cấu trúc danh sách hình ảnh tham chiếu dài hạn.

Cấu trúc cú pháp $\text{ref_pic_list_lt_struct}(\text{ltRplsIdx})$ có thể có mặt trong SPS hoặc trong

phần đầu lát. Phụ thuộc vào việc liệu cấu trúc cú pháp có được bao gồm trong phần đầu lát hoặc SPS hay không, áp dụng phần dưới đây: nếu có mặt trong phần đầu lát, thì cấu trúc cú pháp `ref_pic_list_lt_struct( ltRplsIdx )` định rõ danh sách hình ảnh tham chiếu dài hạn của hình ảnh hiện thời (hình ảnh chứa lát). Nếu không thì (có mặt trong SPS), cấu trúc cú pháp `ref_pic_list_struct( listIdx, rplsIdx )` định rõ ứng viên cho danh sách hình ảnh tham chiếu dài hạn, và thuật ngữ “hình ảnh hiện thời” trong các ngữ nghĩa được định rõ trong phần còn lại của phần này đề cập đến mỗi hình ảnh mà 1) có một hoặc nhiều lát chứa `ref_pic_list_lt_idx[i]` bằng chỉ số trong danh sách của các cấu trúc cú pháp `ref_pic_list_lt_struct( ltRplsIdx )` được bao gồm trong SPS, và 2) nằm trong CVS mà có SPS là SPS hoạt động. `num_ltrp_entries[ ltRplsIdx ]` định rõ số lượng mục nhập LTRP trong cấu trúc cú pháp `ref_pic_list_lt_struct( ltRplsIdx )`. `poc_lsb_lt[ rplsIdx ][ i ]` định rõ trị số của số đếm thứ tự hình ảnh modulo `MaxLtPicOrderCntLsb` của hình ảnh được tham chiếu bởi mục nhập thứ `i` trong cấu trúc cú pháp `ref_pic_list_lt_struct( rplsIdx )`. Độ dài của phần tử cú pháp `poc_lsb_lt[ rplsIdx ][ i ]` là $\text{Log2}(\text{MaxLtPicOrderCntLsb})$ bit. `lt_pos_idx[ rplsIdx ][ i ]` định rõ chỉ số của mục nhập thứ `i` trong cấu trúc cú pháp `ref_pic_list_lt_struct( rplsIdx )` trong danh sách hình ảnh tham chiếu sau khi xây dựng danh sách hình ảnh tham chiếu. Độ dài của `lt_pos_idx[ rplsIdx ][ i ]` phần tử cú pháp là $\text{Log2}(\text{sps_max_dec_pic_buffering_minus1} + 1)$ bit. Khi `num_ltrp_entries[ ltRplsIdx ]` lớn hơn 1, `poc_lsb_lt[ rplsIdx ][ i ]` và `lt_pos_idx[ rplsIdx ][ i ]` sẽ theo thứ tự giảm dần của các trị số `lt_pos_idx[ rplsIdx ][ i ]`.

Quy trình giải mã được mô tả.

Quy trình giải mã cho việc xây dựng các danh sách hình ảnh tham chiếu.

Quy trình này được gọi ra ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP. Các hình ảnh tham chiếu được gửi qua các chỉ số tham chiếu. Chỉ số tham chiếu là chỉ số trong danh sách hình ảnh tham chiếu. Khi giải mã lát I, không có danh sách hình ảnh tham chiếu nào được sử dụng trong việc giải mã dữ liệu lát. Khi giải mã lát P, chỉ danh sách hình ảnh tham chiếu 0 (tức là, `RefPicList[ 0 ]`), được sử dụng trong việc giải mã dữ liệu lát. Khi giải mã lát B, cả danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1 (tức là, `RefPicList[ 1 ]`) được sử dụng trong việc giải mã dữ liệu lát. Ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP, các danh sách hình ảnh tham chiếu `RefPicList[ 0 ]` và `RefPicList[ 1 ]` được dẫn xuất. Các danh sách hình ảnh tham

chiều được sử dụng trong việc đánh dấu các hình ảnh tham chiếu hoặc trong việc giải mã dữ liệu lát. Đối với lát I của hình ảnh không phải IRAP mà không phải là lát thứ nhất của hình ảnh, RefPicList[0] và RefPicList[1] có thể được dẫn xuất cho mục đích kiểm tra độ tương thích dòng bit, nhưng việc suy ra của chúng không cần thiết cho việc giải mã hình ảnh hiện thời hoặc các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Đối với lát P mà không phải là lát thứ nhất của hình ảnh, RefPicList[1] có thể được dẫn xuất cho mục đích kiểm tra độ tương thích dòng bit, nhưng sự dẫn xuất của nó không cần thiết cho việc giải mã hình ảnh hiện thời hoặc các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Các danh sách hình ảnh tham chiếu RefPicList[0] và RefPicList[1] được xây dựng như dưới đây:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplIdx[ i ] ]; j++)
    {
        if( !lt_ref_pic_flag[ i ][ RplIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplIdx[ i ] ][ j ]
            if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
            RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "no reference picture"
            pocBase = RefPicPocList[ i ][ j ]
        } else {
            if( there is a reference picA in the DPB with
            PicOrderCntVal & ( MaxLtPicOrderCntLsb - 1 )
            equal to poc_lsb_lt[ i ][ RplIdx[ i ] ][ j ] )
                RefPicList[ i ][ j ] = picA
        }
    }
}

```

```

else
    RefPicList[ i ][ j ] = “no reference picture”
}
if( ref_pic_list_lt_sps_flag[ i ] )
    LtRplsIdx = ref_pic_list_lt_idx[ i ]
else
    LtRplsIdx = num_ref_pic_lists_lt_in_sps[ i ]
for( j = 0; j < num_ltrp_entries[ LtRplsIdx[ i ] ]; j++ ) {
    if( there is a reference picA in the DPB with
        PicOrderCntVal & ( MaxLtPicOrderCntLsb - 1 ) {
            equal to poc_lsb_lt[ LtRplsIdx[ i ] ][ j ] )
        for( k = sps_max_dec_pic_buffering_minus1; k > lt_pos_idx[ LtRplsIdx[ i ] ]
            [ j ]; k-- )
            RefPicList[ i ][ k ] = RefPicList[ i ][ k - 1 ]
        RefPicList[ i ][ lt_pos_idx[ LtRplsIdx[ i ] ][ j ] ] = picA
    } else {
        for( k = sps_max_dec_pic_buffering_minus1; k > lt_pos_idx[ LtRplsIdx[ i ] ]
            [ j ]; k-- )
            RefPicList[ i ][ k ] = RefPicList[ i ][ k - 1 ]
        RefPicList[ i ][ lt_pos_idx[ RplsIdx[ i ] ][ j ] ] = “no reference picture”
    }
}
}
}

```

Đối với mỗi i bằng 0 hoặc 1, áp dụng phần dưới đây: các mục nhập $\text{NumRefIdxActive}[i]$ thứ nhất trong $\text{RefPicList}[i]$ được đề cập đến là các mục nhập hoạt động trong $\text{RefPicList}[i]$, và các mục nhập khác trong $\text{RefPicList}[i]$ được đề cập đến là các mục nhập không hoạt động trong $\text{RefPicList}[i]$. Mỗi mục nhập trong $\text{RefPicList}[i][j]$ đối với j nằm trong dải từ 0 đến $\text{NumEntriesInList}[i][\text{RplsIdx}[i]] - 1$, được đề cập đến là mục nhập STRP nếu $\text{lt_ref_pic_flag}[i][\text{RplsIdx}[i]][j]$ bằng 0, và nếu không thì là mục nhập LTRP. Có khả năng rằng hình ảnh cụ thể được tham chiếu bởi cả mục nhập trong

RefPicList[0] và mục nhập trong RefPicList[1]. Cũng có thể rằng hình ảnh cụ thể được tham chiếu bởi nhiều hơn một mục nhập trong RefPicList[0] hoặc bởi nhiều hơn một mục nhập trong RefPicList[1]. Các mục nhập hoạt động trong RefPicList[0] và các mục nhập hoạt động trong RefPicList[1] đề cập chung tới tất cả các hình ảnh tham chiếu mà có thể được sử dụng cho việc dự đoán liên ảnh của hình ảnh hiện thời và một hoặc nhiều hình ảnh mà tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Các mục nhập không hoạt động trong RefPicList[0] và các mục nhập không hoạt động trong RefPicList[1] đề cập chung tới tất cả các hình ảnh tham chiếu không được sử dụng cho việc dự đoán liên ảnh của hình ảnh hiện thời nhưng có thể được sử dụng trong việc dự đoán liên ảnh cho một hoặc nhiều hình ảnh mà tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Có thể là một hoặc nhiều mục nhập trong RefPicList[0] hoặc RefPicList[1] mà bằng "không có hình ảnh tham chiếu" bởi vì các hình ảnh tương ứng không có mặt trong DPB. Mỗi mục nhập không hoạt động trong RefPicList[0] hoặc RefPicList[0] bằng "không có hình ảnh tham chiếu" sẽ được bỏ qua. Sự mất hình ảnh vô tình sẽ được suy ra cho mỗi mục nhập hoạt động trong RefPicList[0] hoặc RefPicList[1] mà bằng "không có hình ảnh tham chiếu".

Yêu cầu về sự tương thích dòng bit áp dụng các ràng buộc sau đây: đối với mỗi i bằng 0 hoặc 1, số lượng mục nhập trong RefPicList[i] sẽ không nhỏ hơn NumRefIdxActive[i]. Hình ảnh được tham chiếu bởi mỗi mục nhập hoạt động trong RefPicList[0] hoặc RefPicList[1] sẽ có mặt trong DPB và sẽ có TemporalId nhỏ hơn hoặc bằng TemporalId của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: chỉ số mục nhập của mục nhập không hoạt động bất kỳ trong RefPicList[0] hoặc RefPicList[1] sẽ không được sử dụng làm chỉ số tham chiếu để giải mã của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: mục nhập không hoạt động trong RefPicList[0] hoặc RefPicList[1] sẽ không tham chiếu đến hình ảnh giống như mục nhập khác bất kỳ trong RefPicList[0] hoặc RefPicList[1]. Mục nhập STRP trong RefPicList[0] hoặc RefPicList[1] của lát của hình ảnh và mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1] của cùng lát hoặc lát khác nhau của cùng hình ảnh sẽ không tham chiếu đến cùng hình ảnh. Chính hình ảnh hiện thời sẽ không được tham chiếu bởi mục nhập bất kỳ trong RefPicList[0] hoặc RefPicList[1]. Sẽ không có mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1] mà sự khác biệt giữa PicOrderCntVal của hình ảnh hiện thời và

PicOrderCntVal của hình ảnh được tham chiếu bởi mục nhập lớn hơn hoặc bằng 224. Đặt setOfRefPics là tập hợp các hình ảnh duy nhất được tham chiếu bởi tất cả các mục nhập trong RefPicList[0] và tất cả các mục nhập trong RefPicList[1]. Số lượng của các hình ảnh trong setOfRefPics sẽ nhỏ hơn hoặc bằng sps_max_dec_pic_buffering_minus1 và setOfRefPics sẽ giống nhau đối với tất cả các lát của hình ảnh.

Việc báo hiệu số lượng mục nhập hình ảnh tham chiếu ngắn hạn được thảo luận.

Theo một phương án khác của sáng chế, phần tử cú pháp mà định rõ số lượng mục nhập được kết hợp với các hình ảnh tham chiếu ngắn hạn trong cấu trúc danh sách hình ảnh tham chiếu ref_pic_list_struct() được xác định là num_strp_entries_minus1[listIdx][rplsIdx], thay vì num_strp_entries[listIdx][rplsIdx]. Sự thay đổi này có hai tác dụng đối với việc báo hiệu danh sách hình ảnh tham chiếu: Có thể tiết kiệm các bit để báo hiệu số lượng mục nhập được kết hợp với hình ảnh tham chiếu ngắn hạn trong cấu trúc danh sách hình ảnh tham chiếu khi phần tử được tạo mã sử dụng ue(v). Điều này nhằm đặt một ràng buộc sao cho mỗi danh sách hình ảnh tham chiếu sẽ chứa ít nhất một hình ảnh tham chiếu ngắn hạn. Để phù hợp với ý tưởng này, cần một vài thay đổi so với phương án thứ nhất.

Đối với báo hiệu danh sách hình ảnh tham chiếu trong các phần đầu lát, chỉ danh sách hình ảnh tham chiếu cần thiết được báo hiệu theo loại lát, tức là, một danh sách hình ảnh tham chiếu (tức là, danh sách hình ảnh tham chiếu 0) đối với các lát I hoặc P và hai danh sách hình ảnh tham chiếu (tức là, cả danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1) đối với các lát B. Cú pháp phần đầu lát được thay đổi như dưới đây:

slice_header() {	Ký hiệu mô tả
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	

slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < (slice_type == B ? 2: 1); i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
}	
if(slice_type == P slice_type == B) {	
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++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Bằng cách áp dụng sự thay đổi nêu trên trong phần đầu lát (tức là, danh sách hình ảnh tham chiếu 0 đối với các lát I hoặc P; hình ảnh tham chiếu 0 và hình ảnh tham chiếu 1 đối với các lát B), sẽ tránh sơ đồ khởi vấn đề trong đó đối với lát P chỉ có một hình ảnh tham chiếu ngắn hạn. Tuy nhiên, hình ảnh tham chiếu ngắn hạn trùng lặp không thể được báo hiệu trong danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1, trong đó mục nhập trong danh sách hình ảnh tham chiếu 1 là mục nhập không hoạt động khi số lượng mục nhập hoạt động trong danh sách hình ảnh tham chiếu 1 có bằng 0. Ngữ nghĩa của `num_strp_entries_minus1[ listIdx ][ rplIdx ]` được thay đổi như dưới đây:

num_strp_entries_minus1[listIdx][rplsIdx] cộng 1 định rõ số lượng mục nhập STRP trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag). Biến NumEntriesInList[listIdx][rplsIdx] được dẫn xuất như dưới đây: NumRefPicEntriesInRpl[listIdx][rplsIdx] = num_strp_entries_minus1[listIdx][rplsIdx] + 1 + num_ltrp_entries[listIdx][rplsIdx]. Trị số của NumRefPicEntries[listIdx][rplsIdx] sẽ nằm trong dải từ 1 đến sps_max_dec_pic_buffering_minus1.

Cho phép việc bao gồm hình ảnh hiện thời trong các danh sách hình ảnh tham chiếu.

Theo một phương án khác của sáng chế, hình ảnh hiện thời được cho phép để được bao gồm trong các danh sách hình ảnh tham chiếu của nó. Để hỗ trợ đặc điểm này, không có sự thay đổi cú pháp và các ngữ nghĩa được yêu cầu liên quan tới các phần mô tả đó trong các phương án thứ nhất và thứ hai. Tuy nhiên, các ràng buộc độ tương thích dòng bit được mô tả trong quy trình giải mã cho việc xây dựng danh sách hình ảnh tham chiếu cần phải được cải biến như dưới đây: Yêu cầu về sự tương thích dòng bit áp dụng các ràng buộc sau đây: Đối với mỗi i bằng 0 hoặc 1, NumEntriesInList[i][RplsIdx[i]] sẽ không nhỏ hơn NumRefIdxActive[i]. Hình ảnh được tham chiếu bởi mỗi mục nhập hoạt động trong RefPicList[0] hoặc RefPicList[1] sẽ có mặt trong DPB và sẽ có TemporalId nhỏ hơn hoặc bằng TemporalId của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: Chỉ số mục nhập của mục nhập không hoạt động bất kỳ trong RefPicList[0] hoặc RefPicList[1] sẽ không được sử dụng làm chỉ số tham chiếu để giải mã của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: Mục nhập không hoạt động trong RefPicList[0] hoặc RefPicList[1] sẽ không tham chiếu đến hình ảnh giống như mục nhập khác bất kỳ trong RefPicList[0] hoặc RefPicList[1]. Mục nhập STRP trong RefPicList[0] hoặc RefPicList[1] của lát của hình ảnh và mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1] của cùng lát hoặc lát khác nhau của cùng hình ảnh sẽ không tham chiếu đến cùng hình ảnh. Chính hình ảnh hiện thời sẽ không được tham chiếu bởi mục nhập bất kỳ trong RefPicList[0] hoặc RefPicList[1]. Khi hình ảnh hiện thời được tham chiếu bởi mục nhập trong RefPicList[i], đối với i bằng 0 hoặc 1, chỉ số mục nhập sẽ nhỏ hơn NumRefIdxActive[i]. Sẽ không có mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1] mà sự khác biệt giữa PicOrderCntVal của hình ảnh hiện thời và

PicOrderCntVal của hình ảnh được tham chiếu bởi mục nhập lớn hơn hoặc bằng 224. Đặt setOfRefPics là tập hợp các hình ảnh duy nhất được tham chiếu bởi tất cả các mục nhập trong RefPicList[0] và tất cả các mục nhập trong RefPicList[1]. Nếu hình ảnh hiện thời không được bao gồm trong setOfRefPics, thì số lượng của các hình ảnh trong setOfRefPics sẽ nhỏ hơn hoặc bằng sps_max_dec_pic_buffering_minus1, nếu không thì, số lượng của các hình ảnh trong setOfRefPics sẽ nhỏ hơn hoặc bằng sps_max_dec_pic_buffering_minus1 + 1. setOfRefPics sẽ giống nhau đối với tất cả các lát của hình ảnh.

Sử dụng các bit POC LSB khác nhau đối với các mục nhập LTRP trong các danh sách hình ảnh tham chiếu.

Theo một phương án khác của sáng chế, số lượng của các bit được sử dụng để nhận dạng các hình ảnh tham chiếu dài hạn trong cấu trúc danh sách hình ảnh tham chiếu được cho phép để khác nhau giữa danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1. Để hỗ trợ đặc điểm này, cần đến các thay đổi sau đây:

seq_parameter_set_rbsp() {	Ký hiệu mô tả
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)

log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag)	
— additional_lt_poc_lsb	ue(v)
for(i = 0; i < 2; i++) {	
if(long_term_ref_pics_flag)	
additional_lt_poc_lsb[i]	ue(v)
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, long_term_ref_pics_flag)	
}	
rbsp_trailing_bits()	
}	

additional_lt_poc_lsb[i] định rõ trị số của biến **MaxLtPicOrderCntLsb[i]** được sử dụng trong quy trình giải mã đối với danh sách hình ảnh tham chiếu **ListIdx** bằng **i** như dưới đây:

$\text{MaxLtPicOrderCntLsb}[i] = 2(\text{log2_max_pic_order_cnt_lsb_minus4} + 4 + \text{additional_lt_poc_lsb}[i])$. Trị số của **additional_lt_poc_lsb[i]** sẽ nằm trong dải từ 0 đến $32 - \text{log2_max_pic_order_cnt_lsb_minus4} - 4$. Khi không có mặt, trị số của **additional_lt_poc_lsb[i]** được suy ra bằng 0.

poc_lsb_lt[listIdx][rplIdx][i] định rõ trị số của số đếm thứ tự hình ảnh modulo **MaxLtPicOrderCntLsb[listIdx]** của hình ảnh được tham chiếu bởi mục nhập thứ **i** trong cấu trúc cú pháp **ref_pic_list_struct(listIdx, rplIdx, ltrpFlag)**. Độ dài của phần tử cú pháp **poc_lsb_lt[listIdx][rplIdx][i]** là $\text{Log2}(\text{MaxLtPicOrderCntLsb}[\text{listIdx}])$ bit.

Các danh sách hình ảnh tham chiếu **RefPicList[0]** và **RefPicList[1]** được xây dựng như dưới đây:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplsIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplsIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplsIdx[ i ] ]; j++ )
    {
        if( !lt_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplsIdx[ i ] ][ j ]
            if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
            RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = “no reference picture”
            pocBase = RefPicPocList[ i ][ j ]
        } else {
            if( there is a reference picA in the DPB with
            PicOrderCntVal & ( MaxLtPicOrderCntLsb[ i ] - 1 )
            equal to poc_lsb_lt[ i ][ RplsIdx[ i ] ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = “no reference picture”
        }
    }
}

```

Sử dụng cùng `ref_pic_list_sps_flag` cho các danh sách hình ảnh tham chiếu 0 và 1.

Theo một phương án khác của sáng chế, thay vì sử dụng hai cờ để chỉ báo xem liệu danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1 có được dẫn xuất dựa vào các cấu trúc cú pháp `ref_pic_list_struct()` trong SPS hoạt động hay không, một cờ được sử dụng cho cả hai danh sách hình ảnh tham chiếu. Các ràng buộc khác sao cho hoặc là cả

hai danh sách hình ảnh tham chiếu được dẫn xuất dựa vào `ref_pic_list_struct()` trong SPS hoạt động hoặc chúng được dẫn xuất dựa vào các cấu trúc cú pháp `ref_pic_list_struct()` được bao gồm trực tiếp trong các phần đầu lát của hình ảnh hiện thời. Để hỗ trợ đặc điểm này, cần đến các thay đổi sau đây:

<code>slice_header()</code> {	Ký hiệu mô tả
<code>slice_pic_parameter_set_id</code>	<code>ue(v)</code>
<code>slice_address</code>	<code>u(v)</code>
<code>slice_type</code>	<code>ue(v)</code>
<code>if (slice_type != I)</code>	
<code>log2_diff_ctu_max_bt_size</code>	<code>ue(v)</code>
<code>if(nal_unit_type != IRAP_NUT) {</code>	
<code>slice_pic_order_cnt_lsb</code>	<code>u(v)</code>
<code>ref_pic_list_sps_flag</code>	<code>u(1)</code>
<code>for(i = 0; i < 2; i++) {</code>	
<code>ref_pic_list_sps_flag[i]</code>	<code>u(1)</code>
<code>if(ref_pic_list_sps_flag[i]) {</code>	
<code>if(num_ref_pic_lists_in_sps[i] > 1)</code>	
<code>ref_pic_list_idx[i]</code>	<code>u(v)</code>
<code>} else</code>	
<code>ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i],</code> <code>long_term_ref_pics_flag)</code>	
<code>}</code>	
<code>if(slice_type == P slice_type == B) {</code>	
<code>num_ref_idx_active_override_flag</code>	<code>u(1)</code>
<code>if(num_ref_idx_active_override_flag)</code>	

for(i = 0; i < (slice_type == B ? 2: 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

ref_pic_list_sps_flag[i] bằng 1 định rõ rằng các danh sách hình ảnh tham chiếu i của hình ảnh hiện thời được dẫn xuất dựa vào một trong số các cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag) có listIdx bằng i trong SPS hoạt động. ref_pic_list_sps_flag[i] bằng 0 định rõ rằng các danh sách hình ảnh tham chiếu i của hình ảnh hiện thời được dẫn xuất dựa vào các cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag) được bao gồm trực tiếp trong các phần đầu lát của hình ảnh hiện thời. Khi hoặc là num_ref_pic_lists_in_sps[0] hoặc num_ref_pic_lists_in_sps[1] bằng 0, trị số của ref_pic_list_sps_flag[i] sẽ bằng 0. pic_lists_in_sps[1] bằng 0, trị số của ref_pic_list_sps_flag sẽ bằng 0.

Các danh sách hình ảnh tham chiếu RefPicList[0] và RefPicList[1] được xây dựng như dưới đây:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplsIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplsIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplsIdx[ i ] ]; j++ )
    {
        if( !lt_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplsIdx[ i ] ][ j ]
            if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
            RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
        }
    }
}

```

```

else
    RefPicList[ i ][ j ] = “no reference picture”
    pocBase = RefPicPocList[ i ][ j ]
} else {
    if( there is a reference picA in the DPB with
    PicOrderCntVal & ( MaxLtPicOrderCntLsb – 1 )
        equal to poc_lsb_lt[ i ][ RplsIdx[ i ][ j ] ] )
        RefPicList[ i ][ j ] = picA
    else
        RefPicList[ i ][ j ] = “no reference picture”
    }
}
}

```

Báo hiệu delta POC bit có trọng số cao nhất (Most Significant Bit, viết tắt là MSB) cho các mục nhập hình ảnh tham chiếu dài hạn.

Theo một phương án khác của sáng chế, thay vì sử dụng các bit bổ sung để thể hiện POC LSB của các mục nhập hình ảnh tham chiếu dài hạn trong `ref_pic_list_struct()`, chu kỳ POC MSB được báo hiệu để phân biệt các hình ảnh tham chiếu dài hạn. Khi được báo hiệu, thông tin chu kỳ POC MSB được báo hiệu cho mỗi mục nhập trong `ref_pic_list_struct()` mà tham chiếu đến hình ảnh tham chiếu dài hạn. Cấu trúc cú pháp `ref_pic_list_struct()` không được báo hiệu trong SPS nhưng chỉ trong các phần đầu lát. Để hỗ trợ đặc điểm này, cần đến các thay đổi sau đây:

<code>seq_parameter_set_rbsp()</code> {	Ký hiệu mô tả
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
<code>if(chroma_format_idc == 3)</code>	
separate_colour_plane_flag	u(1)

pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag)	
 additional_lt_poc_lsb	ue(v)
for(i = 0; i < 2; 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, long_term_ref_pics_flag)	
 }	
rbsp_trailing_bits()	
}	

slice_header() {	Ký hiệu mô tả
slice_pic_parameter_set_id	ue(v)

slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
—— ref_pic_list_sps_flag[i]	u(1)
—— if(ref_pic_list_sps_flag[i]) {	
—— if(num_ref_pic_lists_in_sps[i] > 1)	
—— ref_pic_list_idx[i]	u(v)
—— } else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
}	
if(slice_type == P slice_type == B) {	
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++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

ref_pic_list_struct(listIdx, rplsIdx , ltrpFlag) {	Ký hiệu mô tả
num_strp_entries [listIdx][rplsIdx]	ue(v)
if(ltrpFlag)	
num_ltrp_entries [listIdx][rplsIdx]	ue(v)
for(i = 0; i < NumEntriesInList[listIdx][rplsIdx]; i++) {	
if(num_ltrp_entries[listIdx][rplsIdx] > 0)	
lt_ref_pic_flag [listIdx][rplsIdx][i]	
if(!lt_ref_pic_flag[listIdx][rplsIdx][i])	
delta_poc_st [listIdx][rplsIdx][i]	se(v)
else {	
poc_lsb_lt [listIdx][rplsIdx][i]	u(v)
delta_poc_msb_present_flag [listIdx][i]	u(1)
if(delta_poc_msb_present_flag[listIdx][i])	
delta_poc_msb_cycle_lt [listIdx][i]	ue(v)
}	
}	
}	
}	

Cấu trúc cú pháp ref_pic_list_struct(listIdx, ltrpFlag) có thể có mặt trong phần đầu lát. Khi nó có mặt trong phần đầu lát, cấu trúc cú pháp ref_pic_list_struct(listIdx, ltrpFlag) định rõ danh sách hình ảnh tham chiếu ListIdx của hình ảnh hiện thời (hình ảnh chứa lát). num_strp_entries[listIdx][rplsIdx] định rõ số lượng mục nhập STRP trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag). num_ltrp_entries[listIdx][rplsIdx] định rõ số lượng mục nhập LTRP trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag). Khi không có mặt, trị số của num_ltrp_entries[listIdx][rplsIdx] được suy ra bằng 0.

Biến NumEntriesInList[listIdx] [rplsIdx] được dẫn xuất như dưới đây:

NumRefPicEntriesInRpl[listIdx] [rplsIdx] = num_strp_entries[listIdx] [rplsIdx]

+

$$\text{num_ltp_entries}[\text{listIdx}][\text{rplsIdx}]$$

Trị số của $\text{NumRefPicEntries}[\text{listIdx}][\text{rplsIdx}]$ sẽ nằm trong dải từ 0 đến $\text{sps_max_dec_pic_buffering_minus1}$. $\text{lt_ref_pic_flag}[\text{listIdx}][\text{rplsIdx}][i]$ bằng 1 định rõ rằng mục nhập thứ i trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx}, \text{ltpFlag})$ là mục nhập LTRP. $\text{lt_ref_pic_flag}[\text{listIdx}][\text{rplsIdx}][i]$ bằng 0 định rõ rằng mục nhập thứ i trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx}, \text{ltpFlag})$ là mục nhập STRP. Khi không có mặt, trị số của $\text{lt_ref_pic_flag}[\text{listIdx}][\text{rplsIdx}][i]$ được suy ra bằng 0. Yêu cầu về sự tương thích dòng bit là tổng của $\text{lt_ref_pic_flag}[\text{listIdx}][\text{rplsIdx}][i]$ cho tất cả các trị số của i nằm trong dải từ 0 đến $\text{NumRefPicEntries}[\text{listIdx}][\text{rplsIdx}] - 1$, sẽ bằng $\text{num_ltp_entries}[\text{listIdx}][\text{rplsIdx}]$. $\text{delta_poc_st}[\text{listIdx}][\text{rplsIdx}][i]$, khi mục nhập thứ i là mục nhập STRP thứ nhất trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx}, \text{ltpFlag})$, định rõ sự khác biệt giữa các trị số số đếm thứ tự hình ảnh của hình ảnh hiện thời và hình ảnh được tham chiếu bởi mục nhập thứ i , hoặc, khi mục nhập thứ i là mục nhập STRP nhưng không phải mục nhập STRP thứ nhất trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx}, \text{ltpFlag})$, định rõ sự khác biệt giữa các trị số số đếm thứ tự hình ảnh của các hình ảnh được tham chiếu bởi mục nhập thứ i và bởi mục nhập STRP trước đó trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx}, \text{ltpFlag})$. Trị số của $\text{delta_poc_st}[\text{listIdx}][\text{rplsIdx}][i]$ sẽ nằm trong dải từ -215 đến $215 - 1$. $\text{poc_lsb_lt}[\text{listIdx}][\text{rplsIdx}][i]$ định rõ trị số của số đếm thứ tự hình ảnh modulo $\text{MaxLtPicOrderCntLsb}$ của hình ảnh được tham chiếu bởi mục nhập thứ i trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{rplsIdx}, \text{ltpFlag})$. Độ dài của phần tử cú pháp $\text{poc_lsb_lt}[\text{listIdx}][\text{rplsIdx}][i]$ là $\text{Log2}(\text{MaxLtPicOrderCntLsb})$ bit. $\text{delta_poc_msb_present_flag}[\text{listIdx}][i]$ bằng 1 định rõ rằng $\text{delta_poc_msb_cycle_lt}[\text{listIdx}][i]$ là có mặt. $\text{delta_poc_msb_present_flag}[\text{listIdx}][i]$ bằng 0 định rõ rằng $\text{delta_poc_msb_cycle_lt}[\text{listIdx}][i]$ không có mặt. Khi $\text{num_ltp_entries}[\text{listIdx}]$ lớn hơn 0 và có nhiều hơn một hình ảnh tham chiếu trong DPB ở thời điểm khi phần đầu lát này được giải mã mà PicOrderCntVal modulo MaxPicOrderCntLsb bằng $\text{poc_lsb_lt}[\text{listIdx}][i]$, $\text{delta_poc_msb_present_flag}[\text{listIdx}][i]$ sẽ bằng 1. Khi không có mặt, trị số của

$\text{delta_poc_msb_cycle_lt}[\text{listIdx}][i]$ được suy ra bằng 0. $\text{delta_poc_msb_cycle_lt}[\text{listIdx}][i]$ được sử dụng để xác định trị số của các bit có trọng số cao nhất của trị số số đếm thứ tự hình ảnh của mục nhập thứ i trong cấu trúc cú pháp $\text{ref_pic_list_struct}(\text{listIdx}, \text{ltrpFlag})$. Khi $\text{delta_poc_msb_cycle_lt}[\text{listIdx}][i]$ không có mặt, nó được suy ra bằng 0. Các thay đổi đối với quy trình giải mã cho số đếm thứ tự hình ảnh: Ở thời điểm bất kỳ trong quy trình giải mã, các trị số của $\text{PicOrderCntVal} \& (\text{MaxLtPicOrderCntLsb} - 1)$ đối với hai hình ảnh tham chiếu bất kỳ trong DPB sẽ không giống nhau.

Các danh sách hình ảnh tham chiếu $\text{RefPicList}[0]$ và $\text{RefPicList}[1]$ được xây dựng như dưới đây:

```
for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplIdx[ i ] ]; j++ )
    {
        if( !lt_ref_pic_flag[ i ][ RplIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplIdx[ i ] ][ j ]
            if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
            RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "no reference picture"
            pocBase = RefPicPocList[ i ][ j ]
        } else {
            if( delta_poc_msb_cycle_lt[ i ][ j ] is equal to 0 &&
                there is a reference picture picA in the DPB with PicOrderCntVal &
                ( MaxLtPicOrderCntLsb - 1 ) equal to poc_lsb_lt[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
```

else if(delta_poc_msb_cycle_lt[i][j] is equal to 1 &&
 there is a reference picture picA in the DPB with PicOrderCntVal
 equal to

```
( MaxPicOrderCntLsb * delta_poc_msb_cycle_lt[ i ][ j ] ) + poc_lsb_lt[ i ][ j ] ) )
    RefPicList[ i ][ j ] = picA
else
    RefPicList[ i ][ j ] = "no reference picture"
}
}
}
```

Theo cách khác, các ngữ nghĩa của delta_poc_msb_cycle_lt[listIdx][i] có thể được thể hiện là delta của delta sao cho việc xây dựng danh sách hình ảnh tham chiếu có thể được cập nhật như dưới đây: các danh sách hình ảnh tham chiếu RefPicList[0] và RefPicList[1] được xây dựng như dưới đây:

```
for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    prevMsbCycle = 0
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplIdx[ i ] ]; j++ )
    { (8-5)
        if( !lt_ref_pic_flag[ i ][ RplIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplIdx[ i ] ][ j ]
            if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
            RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
```

```

        RefPicList[ i ][ j ] = “no reference picture”
        pocBase = RefPicPocList[ i ][ j ]
    } else {
        if( delta_poc_msb_cycle_lt[ i ][ j ] is equal to 0 &&
            there is a reference picture picA in the DPB with PicOrderCntVal &
            ( MaxLtpicOrderCntLsb - 1 ) equal to poc_lsb_lt[ i ][ j ] )
            RefPicList[ i ][ j ] = picA
        else if( delta_poc_msb_cycle_lt[ i ][ j ] is equal to 1 &&
            there is a reference picA in the DPB with PicOrderCntVal equal to
            ( MaxPicOrderCntLsb * ( delta_poc_msb_cycle_lt[ i ][ j ] + prevMsb
Cycle ) )
            + poc_lsb_lt[ i ][ j ] ) ) {
            RefPicList[ i ][ j ] = picA
            prevMsbCycle += delta_poc_msb_cycle_lt[ i ][ j ]
        } else
            RefPicList[ i ][ j ] = “no reference picture”
    }
}
}

```

Yêu cầu về sự tương thích dòng bit áp dụng các ràng buộc sau đây: đối với mỗi i bằng 0 hoặc 1, $\text{NumEntriesInList}[i][\text{RplIdx}[i]]$ sẽ không nhỏ hơn $\text{NumRefIdxActive}[i]$. Hình ảnh được tham chiếu bởi mỗi mục nhập hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ sẽ có mặt trong DPB và sẽ có TemporalId nhỏ hơn hoặc bằng TemporalId của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: chỉ số mục nhập của mục nhập không hoạt động bất kỳ trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ sẽ không được sử dụng làm chỉ số tham chiếu để giải mã của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: Mục nhập không hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ sẽ không tham chiếu đến hình ảnh giống như mục nhập khác bất kỳ trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$. Mục nhập STRP trong $\text{RefPicList}[0]$ hoặc

RefPicList[1] của lát của hình ảnh và mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1] của cùng lát hoặc lát khác nhau của cùng hình ảnh sẽ không tham chiếu đến cùng hình ảnh. Chính hình ảnh hiện thời sẽ không được tham chiếu bởi mục nhập bất kỳ trong RefPicList[0] hoặc RefPicList[1]. Sẽ không có mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1] mà sự khác biệt giữa PicOrderCntVal của hình ảnh hiện thời và PicOrderCntVal của hình ảnh được tham chiếu bởi mục nhập lớn hơn hoặc bằng 224. Đặt setOfRefPics là tập hợp các hình ảnh duy nhất được tham chiếu bởi tất cả các mục nhập trong RefPicList[0] và tất cả các mục nhập trong RefPicList[1]. Số lượng của các hình ảnh trong setOfRefPics sẽ nhỏ hơn hoặc bằng sps_max_dec_pic_buffering_minus1 và setOfRefPics sẽ giống nhau đối với tất cả các lát của hình ảnh.

Mỗi STRP được nhận dạng bởi trị số PicOrderCntVal của nó. Đối với mỗi LTRP, nếu nó được tham chiếu bởi mục nhập trong RefPicList[0] hoặc RefPicList[1] với delta_poc_msb_present_flag[listIdx][i] bằng 1, thì nó được nhận dạng bởi trị số PicOrderCntVal của nó, nếu không thì, nó được nhận dạng bởi $\text{Log2}(\text{MaxLtPicOrderCntLsb})$ LSB của trị số PicOrderCntVal của nó.

Phương án thay thế 1 để báo hiệu delta POC MSB cho các mục nhập hình ảnh tham chiếu dài hạn.

Phương án này đề xuất phương án thay thế cho phương án được mô tả trong phần mô tả nêu trên. Tương tự như ý tưởng trong phần mô tả nêu trên, thay vì sử dụng các bit bổ sung để thể hiện POC LSB của hình ảnh tham chiếu dài hạn trong ref_pic_list_struct(), chu kỳ POC MSB được báo hiệu để phân biệt các hình ảnh tham chiếu dài hạn. Tuy nhiên, trong phương án thay thế này, khi được báo hiệu, thông tin chu kỳ POC MSB không được báo hiệu trong ref_pic_list_struct(), thay vào đó, khi cần đến thông tin chu kỳ POC MSB, nó được báo hiệu trong phần đầu lát. Cấu trúc cú pháp ref_pic_list_struct() có thể được báo hiệu trong SPS và trong các phần đầu lát.

seq_parameter_set_rbsp() {	Ký hiệu mô tả
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)

if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
long_term_ref_pics_flag	u(1)
—if(long_term_ref_pics_flag)	
— additional_lt_poc_lsb	ue(v)
for(i = 0; i < 2; 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, long_term_ref_pics_flag)	
}	
rbsp_trailing_bits()	
}	

slice_header() {	Ký hiệu mô tả
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
if(long_term_ref_pics_flag) {	
NumLtrpEntries[i] = ref_pic_list_sps_flag[i] ? num_ltrp_entries[i][ref_pic_list_idx[i]] : num_ltrp_entries[i][num_ref_pic_lists_in_sps[i]]	
for(j = 0; j < NumLtrpEntries[i]; j++) {	
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)

}	
}	
}	
if(slice_type == P slice_type == B) {	
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++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

$\text{delta_POC_msb_present_flag}[i][j]$ bằng 1 định rõ rằng $\text{delta_poc_msb_cycle_lt}[i][j]$ có mặt. $\text{delta_poc_msb_present_flag}[i][j]$ bằng 0 định rõ rằng $\text{delta_poc_msb_cycle_lt}[i][j]$ không có mặt. Khi $\text{NumLtRpEntries}[i]$ lớn hơn 0 và đối với mục nhập LTRP thứ j trong cấu trúc cú pháp $\text{ref_pic_list_struct}(i, \text{rplsIdx}, 1)$ có nhiều hơn một hình ảnh tham chiếu trong DPB ở thời điểm khi phần đầu lát này được giải mã mà PicOrderCntVal modulo MaxPicOrderCntLsb bằng $\text{poc_lsb_lt}[i][\text{rplsIdx}][jj]$, trong đó jj là chỉ số mục nhập của mục nhập trong cấu trúc cú pháp $\text{ref_pic_list_struct}(i, \text{rplsIdx}, 1)$ mà là mục nhập LTRP thứ j trong cấu trúc cú pháp $\text{ref_pic_list_struct}(i, \text{rplsIdx}, 1)$, $\text{delta_poc_msb_present_flag}[i][j]$ sẽ bằng 1. Khi không có mặt, trị số của $\text{delta_poc_msb_cycle_lt}[i][j]$ được suy ra bằng 0. $\text{delta_poc_msb_cycle_lt}[i][j]$ được sử dụng để xác định trị số của các bit có trọng số cao nhất của trị số số đếm thứ tự hình ảnh của mục nhập LTRP thứ j trong cấu trúc cú pháp $\text{ref_pic_list_struct}(i, \text{rplsIdx}, 1)$. Khi $\text{delta_poc_msb_cycle_lt}[i][j]$ không có mặt, nó được suy ra bằng 0.

$\text{slice_header}()$ {	Ký hiệu mô tả
slice_pic_parameter_set_id	ue(v)

slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
if(long_term_ref_pics_flag) {	
rplsIdx = ref_pic_list_sps_flag[i] ? ref_pic_list_idx[i] : num_ref_pic_lists_in_sps[i]	
numRpEntries[i] = num_strp_entries[i][rplsIdx] + num_ltrp_entries[i][rplsIdx]	
NumLtrpEntries[i] = num_ltrp_entries[i][rplsIdx]	
for(j = 0; j < numRpEntries[i]; j++) {	
if(lt_ref_pic_flag[i][rplsIdx][j])	
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)
}	

}	
}	
if(slice_type == P slice_type == B) {	
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++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

delta_POC_msb_present_flag[i][j] bằng 1 định rõ rằng delta_poc_msb_cycle_lt[i][j] có mặt. delta_poc_msb_present_flag[i][j] bằng 0 định rõ rằng delta_poc_msb_cycle_lt[i][j] không có mặt. Khi NumLtRpEntries[i] lớn hơn 0 và có nhiều hơn một hình ảnh tham chiếu trong DPB ở thời điểm khi phần đầu lát này được giải mã mà PicOrderCntVal modulo MaxPicOrderCntLsb bằng poc_lsb_lt[i][rplsIdx][j], delta_poc_msb_present_flag[i][j] sẽ bằng 1. Khi không có mặt, trị số của delta_poc_msb_cycle_lt[i][j] được suy ra bằng 0. delta_poc_msb_cycle_lt[i][j] được sử dụng để xác định trị số của các bit có trọng số cao nhất của trị số số đếm thứ tự hình ảnh của mục nhập thứ j trong cấu trúc cú pháp ref_pic_list_struct(i, rplsIdx, 1). Khi delta_poc_msb_cycle_lt[i][j] không có mặt, nó được suy ra bằng 0. poc_lsb_lt[listIdx][rplsIdx][i] định rõ trị số của số đếm thứ tự hình ảnh modulo MaxLtPicOrderCntLsb MaxPicOrderCntLsb của hình ảnh được tham chiếu bởi mục nhập thứ i trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag). Độ dài của phần tử cú pháp poc_lsb_lt[listIdx][rplsIdx][i] là $\text{Log2}(\text{MaxLtPicOrderCntLsb} \text{MaxPicOrderCntLsb})$ bit.

Các thay đổi đối với quy trình giải mã cho số đếm thứ tự hình ảnh: Ở thời điểm bất kỳ trong quy trình giải mã, các trị số của PicOrderCntVal & (MaxLtPicOrderCntLsb - 1) đối

với hai hình ảnh tham chiếu bất kỳ trong DPB sẽ không giống nhau.

Đối với thiết kế phân đầu lát 1, các danh sách hình ảnh tham chiếu RefPicList[0] và RefPicList[1] được xây dựng như dưới đây:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplsIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplsIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    msbCycleIdx = 0
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplsIdx[ i ] ]; j++ )
    {
        if( !lt_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplsIdx[ i ] ][ j ]
            if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
            RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "no reference picture"
            pocBase = RefPicPocList[ i ][ j ]
        } else {
            if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] is equal to 0 &&
                there is a reference picture picA in the DPB with PicOrderCntVal &
                ( MaxLtPicOrderCntLsb - 1 ) equal to
                poc_lsb_lt[ i ][ RplsIdx[ i ] ][ j ] )
                RefPicList[ i ][ j ] = picA
            else if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] is equal to 1 &&
                there is a reference picture picA in the DPB with PicOrderCntVal
                equal to
                ( MaxPicOrderCntLsb *

```

```

delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] ) + poc_lsb_lt[ i ][ RplsIdx[ i ][ j ] ] ) {
    RefPicList[ i ][ j ] = picA
    msbCycleIdx++
} else
    RefPicList[ i ][ j ] = “no reference picture”
}
}
}

```

Theo cách khác, đối với thiết kế phần đầu lát 1, các ngữ nghĩa của `delta_poc_msb_cycle_lt[ listIdx ][ i ]` có thể được thể hiện là delta của delta sao cho việc xây dựng danh sách hình ảnh tham chiếu có thể được cập nhật như dưới đây: Các danh sách hình ảnh tham chiếu `RefPicList[ 0 ]` và `RefPicList[ 1 ]` được xây dựng như dưới đây:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplsIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplsIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    pevMsbCycle = 0
    msbCycleIdx = 0
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplsIdx[ i ] ]; j++ )
    {
        if( !lt_ref_pic_flag[ i ][ RplsIdx[ i ][ j ] ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplsIdx[ i ][ j ] ]
            if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
            RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = “no reference picture”
            pocBase = RefPicPocList[ i ][ j ]
        } else {

```

```

if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] is equal to 0 &&
    there is a reference picture picA in the DPB with PicOrderCntVal &
    ( MaxPicOrderCntLsb - 1 ) equal to
poc_lsb_lt[ i ][ RplIdx[ i ][ j ] )
    RefPicList[ i ][ j ] = picA
else if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] is equal to 1 &&
    there is a reference picture picA in the DPB with PicOrderCntVal
equal to
    ( MaxPicOrderCntLsb *
( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] +
    prevMsbCycle ) + poc_lsb_lt[ i ][ RplIdx[ i ][ j ] ) ) {
    RefPicList[ i ][ j ] = picA
    prevMsbCycle += delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ]
    msbCycleIdx++
} else
    RefPicList[ i ][ j ] = "no reference picture"
}
}
}

```

Đối với thiết kế phần đầu lát 2, các danh sách hình ảnh tham chiếu RefPicList[0] và RefPicList[1] được xây dựng như dưới đây:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplIdx[ i ] ]; j++ )
    {
        if( !lt_ref_pic_flag[ i ][ RplIdx[ i ][ j ] ] ) {
            RefPicList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplIdx[ i ][ j ] ]

```

```

        if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
RefPicPocList[ i ][ j ] )
            RefPicList[ i ][ j ] = picA
        else
            RefPicList[ i ][ j ] = “no reference picture”
        pocBase = RefPicPocList[ i ][ j ]
    } else {
        if( delta_poc_msb_cycle_lt[ i ][ j ] is equal to 0 &&
            there is a reference picture picA in the DPB with PicOrderCntVal &
            ( MaxPicOrderCntLsb - 1 ) equal to
poc_lsb_lt[ i ][ RplIdx[ i ][ j ] ] )
            RefPicList[ i ][ j ] = picA
        else if( delta_poc_msb_cycle_lt[ i ][ j ] is equal to 1 &&
            there is a reference picture picA in the DPB with PicOrderCntVal
equal to (
            MaxPicOrderCntLsb *
            delta_poc_msb_cycle_lt[ i ][ j ] ) + poc_lsb_lt[ i ][ RplIdx[ i ][ j ] ] )
        )
            RefPicList[ i ][ j ] = picA
        else
            RefPicList[ i ][ j ] = “no reference picture”
    }
}
}
}

```

Theo cách khác, đối với thiết kế phần đầu lát 2, các ngữ nghĩa của $\text{delta_poc_msb_cycle_lt}[\text{listIdx}][i]$ có thể được thể hiện là delta của delta sao cho việc xây dựng danh sách hình ảnh tham chiếu có thể được cập nhật như dưới đây: Các danh sách hình ảnh tham chiếu $\text{RefPicList}[0]$ và $\text{RefPicList}[1]$ được xây dựng như dưới đây:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )

```

```

    RplsIdx[ i ] = ref_pic_list_idx[ i ]
else
    RplsIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
prevMsbCycle = 0
for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplsIdx[ i ] ]; j++)
{
    if( !lt_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ] ) {
        RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplsIdx[ i ] ][ j ]
        if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
        RefPicPocList[ i ][ j ] )
            RefPicList[ i ][ j ] = picA
        else
            RefPicList[ i ][ j ] = "no reference picture"
        pocBase = RefPicPocList[ i ][ j ]
    } else {
        if( delta_poc_msb_cycle_lt[ i ][ j ] is equal to 0 &&
            there is a reference picture picA in the DPB with PicOrderCntVal &
            ( MaxLtPicOrderCntLsb - 1 ) equal to
            poc_lsb_lt[ i ][ RplsIdx[ i ] ][ j ] )
            RefPicList[ i ][ j ] = picA
        else if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] is equal to 1 &&
            there is a reference picture picA in the DPB with PicOrderCntVal
            equal to
            ( MaxPicOrderCntLsb * ( delta_poc_msb_cycle_lt[ i ][ j ] +
            prevMsbCycle ) + poc_lsb_lt[ i ][ RplsIdx[ i ] ][ j ] ) ) {
            RefPicList[ i ][ j ] = picA
            prevMsbCycle += delta_poc_msb_cycle_lt[ i ][ j ]
        } else
            RefPicList[ i ][ j ] = "no reference picture"
    }
}

```

```

    }
}

```

Yêu cầu về sự tương thích dòng bit áp dụng các ràng buộc sau đây: Đối với mỗi i bằng 0 hoặc 1, $\text{NumEntriesInList}[i][\text{RplIdx}[i]]$ sẽ không nhỏ hơn $\text{NumRefIdxActive}[i]$. Hình ảnh được tham chiếu bởi mỗi mục nhập hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ sẽ có mặt trong DPB và sẽ có TemporalId nhỏ hơn hoặc bằng TemporalId của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: Chỉ số mục nhập của mục nhập không hoạt động bất kỳ trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ sẽ không được sử dụng làm chỉ số tham chiếu để giải mã của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: Mục nhập không hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ sẽ không tham chiếu đến hình ảnh giống như mục nhập khác bất kỳ trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$. Mục nhập STRP trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ của lát của hình ảnh và mục nhập LTRP trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ của cùng lát hoặc lát khác nhau của cùng hình ảnh sẽ không tham chiếu đến cùng hình ảnh. Chính hình ảnh hiện thời sẽ không được tham chiếu bởi mục nhập bất kỳ trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$. Sẽ không có mục nhập LTRP trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ mà sự khác biệt giữa PicOrderCntVal của hình ảnh hiện thời và PicOrderCntVal của hình ảnh được tham chiếu bởi mục nhập lớn hơn hoặc bằng 224. Đặt setOfRefPics là tập hợp các hình ảnh duy nhất được tham chiếu bởi tất cả các mục nhập trong $\text{RefPicList}[0]$ và tất cả các mục nhập trong $\text{RefPicList}[1]$. Số lượng của các hình ảnh trong setOfRefPics sẽ nhỏ hơn hoặc bằng $\text{sps_max_dec_pic_buffering_minus1}$ và setOfRefPics sẽ giống nhau đối với tất cả các lát của hình ảnh.

Mỗi STRP được nhận dạng bởi trị số PicOrderCntVal của nó. Đối với mỗi LTRP, nếu nó được tham chiếu bởi mục nhập trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ với $\text{delta_poc_msb_present_flag}[i][j]$ bằng 1, thì nó được nhận dạng bởi trị số PicOrderCntVal của nó, nếu không thì, nó được nhận dạng bởi $\text{Log2}(\text{MaxLtPicOrderCntLsb})$ LSB của trị số PicOrderCntVal của nó.

Phương án thay thế 2 để báo hiệu delta POC MSB cho các mục nhập hình ảnh tham chiếu dài hạn.

Theo một phương án khác của sáng chế, sáng chế được mô tả trong phương án thứ nhất

hoặc phương án thứ hai có thể được kết hợp với các phương án nêu trên và lần lượt được đặt tên là “báo hiệu delta POC MSB cho các mục nhập hình ảnh tham chiếu dài hạn” và “Phương án thay thế 1 để báo hiệu delta POC MSB cho các mục nhập hình ảnh tham chiếu dài hạn”. Các khía cạnh của sáng chế được kết hợp là báo hiệu `additional_lt_poc_lsb` (tức là, từ phương án thứ nhất hoặc phương án thứ hai) và thông tin chu kỳ POC MSB (tức là, từ phương án nêu trên và được đặt tên là “báo hiệu delta POC MSB cho các mục nhập hình ảnh tham chiếu dài hạn” hoặc “Phương án thay thế 1 để báo hiệu delta POC MSB cho các mục nhập hình ảnh tham chiếu dài hạn”). Một ví dụ về cách kết hợp, kết hợp phương án thứ nhất và phương án nêu trên và được đặt tên là “Phương án thay thế 1 để báo hiệu delta POC MSB cho các mục nhập hình ảnh tham chiếu dài hạn”, có thể được thực hiện được mô tả như dưới đây:

<code>slice_header() {</code>	Ký hiệu mô tả
<code> slice_pic_parameter_set_id</code>	<code>ue(v)</code>
<code> slice_address</code>	<code>u(v)</code>
<code> slice_type</code>	<code>ue(v)</code>
<code> if (slice_type != I)</code>	
<code> log2_diff_ctu_max_bt_size</code>	<code>ue(v)</code>
<code> if(nal_unit_type != IRAP_NUT) {</code>	
<code> slice_pic_order_cnt_lsb</code>	<code>u(v)</code>
<code> for(i = 0; i < 2; i++) {</code>	
<code> ref_pic_list_sps_flag[i]</code>	<code>u(1)</code>
<code> if(ref_pic_list_sps_flag[i]) {</code>	
<code> if(num_ref_pic_lists_in_sps[i] > 1)</code>	
<code> ref_pic_list_idx[i]</code>	<code>u(v)</code>
<code> } else</code>	
<code> ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i],</code> <code> long_term_ref_pics_flag)</code>	

if(long_term_ref_pics_flag) {	
NumLtrpEntries[i] = ref_pic_list_sps_flag[i] ? num_ltrp_entries[i][ref_pic_list_idx[i]] :	
num_ltrp_entries[i][num_ref_pic_lists_in_sps[i]]	
for(j = 0; j < NnumLtrpEntries[i]; j++) {	
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)
}	
}	
}	
if(slice_type == P slice_type == B) {	
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++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

delta_POC_msb_present_flag[i][j] bằng 1 định rõ rằng delta_poc_msb_cycle_lt[i][j] có mặt. delta_poc_msb_present_flag[i][i] bằng 0 định rõ rằng delta_poc_msb_cycle_lt[i][j] không có mặt. Khi NumLtrpEntries[i] lớn hơn 0 và đối với mục nhập LTRP thứ j trong cấu trúc cú pháp ref_pic_list_struct(i, rplIdx, 1) có nhiều hơn một hình ảnh tham chiếu trong DPB ở thời điểm khi phần đầu lát này được giải mã mà PicOrderCntVal modulo MaxPicOrderLtCntLsb bằng poc_lsb_lt[i][rplIdx][jj], trong đó

j là chỉ số mục nhập của mục nhập trong cấu trúc cú pháp `ref_pic_list_struct( i, rplsIdx, 1 )` mà là mục nhập LTRP thứ j trong cấu trúc cú pháp `ref_pic_list_struct( i, rplsIdx, 1 )`, `delta_poc_msb_present_flag[ i ][ j ]` sẽ bằng 1. Khi không có mặt, trị số của `delta_poc_msb_cycle_lt[ i ][ j ]` được suy ra bằng 0. `delta_poc_msb_cycle_lt[ i ][ j ]` được sử dụng để xác định trị số của các bit có trọng số cao nhất của trị số số đếm thứ tự hình ảnh của mục nhập LTRP thứ j trong cấu trúc cú pháp `ref_pic_list_struct( i, rplsIdx, 1 )`. Khi `delta_poc_msb_cycle_lt[ i ][ j ]` không có mặt, nó được suy ra bằng 0.

Các thay đổi đối với quy trình giải mã cho số đếm thứ tự hình ảnh: Ở thời điểm bất kỳ trong quy trình giải mã, các trị số của `PicOrderCntVal` & `( MaxLtPicOrderCntLsb - 1 )` đối với hai hình ảnh tham chiếu bất kỳ trong DPB sẽ không giống nhau.

Các danh sách hình ảnh tham chiếu `RefPicList[ 0 ]` và `RefPicList[ 1 ]` được xây dựng như dưới đây:

```
for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplsIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplsIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    msbCycleIdx = 0
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplsIdx[ i ] ]; j++ )
    {
        if( !lt_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplsIdx[ i ] ][ j ]
            if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
            RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "no reference picture"
            pocBase = RefPicPocList[ i ][ j ]
        } else {
            if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] is equal to 0 &&
```

```

        there is a reference picture picA in the DPB with PicOrderCntVal &
        ( MaxLtPicOrderCntLsb - 1 ) equal to
poc_lsb_lt[ i ][ RplsIdx[ i ] ][ j ] )
    RefPicList[ i ][ j ] = picA
    else if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] is equal to 1 &&
        there is a reference picture picA in the DPB with    PicOrderCntVal
equal to (
    MaxLtPicOrderCntLsb *
    delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] ) + poc_lsb_lt[ i ][ RplsIdx[ i ] ][ j ] ) ) {
        RefPicList[ i ][ j ] = picA
        msbCycleIdx++
    } else
        RefPicList[ i ][ j ] = “no reference picture”
    }
}
}
}

```

Theo cách khác, các ngữ nghĩa của `delta_poc_msb_cycle_lt[ listIdx ][ i ]` có thể được thể hiện là delta của delta sao cho việc xây dựng danh sách hình ảnh tham chiếu có thể được cập nhật như dưới đây: Các danh sách hình ảnh tham chiếu `RefPicList[ 0 ]` và `RefPicList[ 1 ]` được xây dựng như dưới đây:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplsIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplsIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    prevMsbCycle = 0
    msbCycleIdx = 0
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplsIdx[ i ] ]; j++ )
    {

```

```

if( !lt_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ] ) {
    RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplsIdx[ i ] ][ j ]
    if( there is a reference picture picA in the DPB with PicOrderCntVal equal to
    RefPicPocList[ i ][ j ] )
        RefPicList[ i ][ j ] = picA
    else
        RefPicList[ i ][ j ] = “no reference picture”
    pocBase = RefPicPocList[ i ][ j ]
} else {
    if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] is equal to 0 &&
        there is a reference picture picA in the DPB with PicOrderCntVal &
        ( MaxLtPicOrderCntLsb - 1 ) equal to
    poc_lsb_lt[ i ][ RplsIdx[ i ] ][ j ] )
        RefPicList[ i ][ j ] = picA
    else if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] is equal to 1 &&
        there is a reference picture picA in the DPB with PicOrderCntVal
    equal to
        ( MaxLtPicOrderCntLsb *
    ( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] +
        prevMsbCycle ) + poc_lsb_lt[ i ][ RplsIdx[ i ] ][ j ] ) ) {
        RefPicList[ i ][ j ] = picA
        prevMsbCycle += delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ]
        msbCycleIdx++
    } else
        RefPicList[ i ][ j ] = “no reference picture”
    }
}
}
}

```

Yêu cầu về sự tương thích dòng bit áp dụng các ràng buộc sau đây: Đối với mỗi i bằng 0 hoặc 1, $\text{NumEntriesInList}[i][\text{RplsIdx}[i]]$ sẽ không nhỏ hơn $\text{NumRefIdxActive}[i]$. Hình

ảnh được tham chiếu bởi mỗi mục nhập hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ sẽ có mặt trong DPB và sẽ có TemporalId nhỏ hơn hoặc bằng TemporalId của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: Chỉ số mục nhập của mục nhập không hoạt động bất kỳ trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ sẽ không được sử dụng làm chỉ số tham chiếu để giải mã của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: Mục nhập không hoạt động trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ sẽ không tham chiếu đến hình ảnh giống như mục nhập khác bất kỳ trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$. Mục nhập STRP trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ của lát của hình ảnh và mục nhập LTRP trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ của cùng lát hoặc lát khác nhau của cùng hình ảnh sẽ không tham chiếu đến cùng hình ảnh. Chính hình ảnh hiện thời sẽ không được tham chiếu bởi mục nhập bất kỳ trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$. Sẽ không có mục nhập LTRP trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ mà sự khác biệt giữa PicOrderCntVal của hình ảnh hiện thời và PicOrderCntVal của hình ảnh được tham chiếu bởi mục nhập lớn hơn hoặc bằng 224. Đặt setOfRefPics là tập hợp các hình ảnh duy nhất được tham chiếu bởi tất cả các mục nhập trong $\text{RefPicList}[0]$ và tất cả các mục nhập trong $\text{RefPicList}[1]$. Số lượng của các hình ảnh trong setOfRefPics sẽ nhỏ hơn hoặc bằng $\text{sps_max_dec_pic_buffering_minus1}$ và setOfRefPics sẽ giống nhau đối với tất cả các lát của hình ảnh.

Mỗi STRP được nhận dạng bởi trị số PicOrderCntVal của nó. Đối với mỗi LTRP, nếu nó được tham chiếu bởi mục nhập trong $\text{RefPicList}[0]$ hoặc $\text{RefPicList}[1]$ với $\text{delta_poc_msb_present_flag}[i][j]$ bằng 1, thì nó được nhận dạng bởi trị số PicOrderCntVal của nó, nếu không thì, nó được nhận dạng bởi $\text{Log2}(\text{MaxLtPicOrderCntLsb})$ LSB của trị số PicOrderCntVal của nó.

Luôn báo hiệu các danh sách hình ảnh tham chiếu trong các phần đầu lát có sự khác biệt giữa các hình ảnh tham chiếu ngắn hạn và dài hạn.

Phần này mô tả phương án khác của sáng chế. Phần mô tả này liên quan tới VVC WD cuối cùng (tức là, chỉ delta liên quan tới VVC WD cuối cùng trong JVET-K1001-v1 được mô tả, trong khi các văn bản trong VVC WD cuối cùng mà không được đề cập dưới đây áp dụng như hiện tại). Phương án khác này được tóm tắt như dưới đây: Các cấu trúc danh sách hình ảnh tham chiếu được báo hiệu chỉ trong các phần đầu lát. Cả các hình ảnh tham chiếu

ngắn hạn và các hình ảnh tham chiếu dài hạn được nhận dạng bởi các POC LSB của chúng, mà có thể được thể hiện bởi các số lượng của các bit mà khác với số lượng của các bit được sử dụng để thể hiện các POC LSB được báo hiệu trong các phần đầu lát để dẫn xuất các trị số POC. Ngoài ra, các số lượng của các bit được sử dụng để thể hiện các POC LSB để nhận dạng các hình ảnh tham chiếu ngắn hạn và các hình ảnh tham chiếu dài hạn có thể là khác nhau.

Cú pháp phần đầu đơn vị NAL.

nal_unit_header() {	Ký hiệu mô tả
forbidden_zero_bit	f(1)
nal_unit_type	u(5)
nuh_temporal_id_plus1	u(3)
nuh_reserved_zero_7bits	u(7)
}	

Cú pháp tập hợp thông số chuỗi RBSP.

seq_parameter_set_rbsp() {	Ký hiệu mô tả
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)

qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
additional_st_poc_lsb	ue(v)
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag)	
additional_lt_poc_lsb	ue(v)
rbsp_trailing_bits()	
}	

Cú pháp tập hợp thông số hình ảnh RBSP.

pic_parameter_set_rbsp() {	Ký hiệu mô tả
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rbsp_trailing_bits()	
}	

Cú pháp phân đầu lát.

slice_header() {	Ký hiệu mô tả
slice_pic_parameter_set_id	ue(v)

slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++)	
ref_pic_list_struct(i, long_term_ref_pics_flag)	
if(slice_type == P slice_type == B) {	
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++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Cú pháp cấu trúc danh sách hình ảnh tham chiếu.

ref_pic_list_struct(listIdx, ltrpFlag) {	Ký hiệu mô tả
num_strp_entries[listIdx]	ue(v)
if(ltrpFlag)	
num_ltrp_entries[listIdx]	ue(v)
for(i = 0; i < NumEntriesInList[listIdx]; i++) {	
if(num_ltrp_entries[listIdx] > 0)	

lt_ref_pic_flag[listIdx][i]	
if(!lt_ref_pic_flag[listIdx][i])	
poc_lsb_st[listIdx][i]	u(v)
else	
poc_lsb_lt[listIdx][i]	u(v)
}	
}	

Các ngữ nghĩa phần đầu đơn vị NAL.

forbidden_zero_bit sẽ bằng 0. nal_unit_type định rõ loại của cấu trúc dữ liệu RBSP được chứa trong đơn vị NAL.

Bảng 7-1 – các mã loại đơn vị NAL và các lớp loại đơn vị NAL

nal_unit_type	Tên của nal_unit_type	Nội dung của đơn vị NAL và cấu trúc cú pháp RBSP	Lớp loại đơn vị NAL
0	NON_IRAP_NUT	Phân đoạn lát được tạo mã của hình ảnh không phải IRAP slice_layer_rbsp()	VCL
1	IRAP_NUT	Lát được tạo mã của hình ảnh IRAP slice_layer_rbsp()	VCL
2-15	RSV_VCL_NUT	Các đơn vị NAL VCL dành riêng	VCL
16	SPS_NUT	Tập hợp thông số chuỗi seq_parameter_set_rbsp()	không phải VCL
17	PPS_NUT	Tập hợp thông số hình ảnh pic_parameter_set_rbsp()	không phải

			VCL
18	EOS_NUT	Phần cuối của chuỗi end_of_seq_rbsp()	không phải VCL
19	EOB_NUT	Phần cuối của dòng bit end_of_bitstream_rbsp()	không phải VCL
20, 21	PREFIX_SEI_NUT SUFFIX_SEI_NUT	Thông tin nâng cao bổ sung sei_rbsp()	không phải VCL
22-26	RSV_NVCL	Dành riêng	không phải VCL
27-31	UNSPEC	Không được định rõ	không phải VCL

nuh_temporal_id_plus1 trừ 1 định rõ ký hiệu nhận dạng tạm thời cho đơn vị NAL. Trị số của nuh_temporal_id_plus1 sẽ không bằng 0. Biến TemporalId được định rõ như dưới đây: $\text{TemporalId} = \text{nuh_temporal_id_plus1} - 1$.

Khi nal_unit_type bằng IRAP_NUT, lát được tạo mã thuộc về hình ảnh IRAP, và TemporalId sẽ bằng 0. Trị số của TemporalId sẽ giống nhau đối với tất cả các đơn vị NAL VCL của đơn vị truy cập. Trị số của TemporalId của hình ảnh được tạo mã hoặc đơn vị truy cập là trị số của TemporalId của các đơn vị NAL VCL của hình ảnh được tạo mã hoặc đơn vị truy cập. Trị số của TemporalId cho các đơn vị NAL không phải VCL bị ràng buộc như dưới đây: Nếu nal_unit_type bằng SPS_NUT, thì TemporalId sẽ bằng 0 và TemporalId của đơn vị truy cập chứa đơn vị NAL sẽ bằng 0. Nếu không thì, nếu nal_unit_type bằng EOS_NUT hoặc EOB_NUT, thì TemporalId sẽ bằng 0. Nếu không thì, TemporalId sẽ lớn hơn hoặc bằng TemporalId của đơn vị truy cập chứa đơn vị NAL. Khi đơn vị NAL là đơn vị

NAL không phải VCL, trị số của TemporalId bằng trị số nhỏ nhất của các trị số TemporalId của tất cả các đơn vị truy cập mà đơn vị NAL không phải VCL áp dụng. Khi nal_unit_type bằng PPS_NUT, TemporalId có thể lớn hơn hoặc bằng TemporalId của đơn vị truy cập chứa, vì tất cả các tập hợp thông số hình ảnh (Picture Parameter Set, viết tắt là PPS) có thể được bao gồm trong phần đầu của dòng bit, trong đó hình ảnh được tạo mã thứ nhất có TemporalId bằng 0. Khi nal_unit_type bằng PREFIX_SEI_NUT hoặc SUFFIX_SEI_NUT, TemporalId có thể lớn hơn hoặc bằng TemporalId của đơn vị truy cập chứa, vì đơn vị NAL thông tin nâng cao bổ sung (SEI) có thể chứa thông tin mà áp dụng cho tập con dòng bit bao gồm các đơn vị truy cập mà các trị số TemporalId lớn hơn TemporalId của đơn vị truy cập chứa đơn vị NAL SEI. nuh_reserved_zero_7bits sẽ bằng '0000000'. Các trị số khác của nuh_reserved_zero_7bits có thể được định rõ trong tương lai bởi ITU-T | ISO/IEC. Các bộ giải mã sẽ bỏ qua (tức là, loại bỏ khỏi dòng bit và loại trừ) các đơn vị NAL có các trị số của nuh_reserved_zero_7bits không bằng '0000000'.

Các ngữ nghĩa tập hợp thông số chuỗi RBSP.

log2_max_pic_order_cnt_lsb_minus4 định rõ trị số của biến MaxPicOrderCntLsb được sử dụng trong quy trình giải mã đối với số đếm thứ tự hình ảnh như dưới đây:

$$\text{MaxPicOrderCntLsb} = 2(\text{log2_max_pic_order_cnt_lsb_minus4} + 4)$$

Trị số của log2_max_pic_order_cnt_lsb_minus4 sẽ nằm trong dải từ 0 đến 12. sps_max_dec_pic_buffering_minus1 cộng 1 định rõ kích thước yêu cầu lớn nhất của bộ đệm hình ảnh được giải mã cho CVS trong các đơn vị của các bộ đệm lưu trữ hình ảnh. Trị số của sps_max_dec_pic_buffering_minus1 sẽ nằm trong dải từ 0 đến MaxDpbSize - 1, trong đó MaxDpbSize như được định rõ ở phần khác. additional_st_poc_lsb định rõ trị số của biến MaxStPicOrderCntLsb được sử dụng trong quy trình giải mã cho các danh sách hình ảnh tham chiếu như dưới đây:

$$\text{MaxStPicOrderCntLsb} = 2(\text{log2_max_pic_order_cnt_lsb_minus4} + 4 + \text{additional_st_poc_lsb})$$

Trị số của additional_st_poc_lsb sẽ nằm trong dải từ 0 đến $32 - \text{log2_max_pic_order_cnt_lsb_minus4} - 4$. long_term_ref_pics_flag bằng 0 định rõ

rằng không có LTRP được sử dụng cho việc dự đoán liên ảnh của hình ảnh được tạo mã bất kỳ trong CVS. `long_term_ref_pics_flag` bằng 1 định rõ rằng các LTRP có thể được sử dụng cho việc dự đoán liên ảnh của một hoặc nhiều hình ảnh được tạo mã trong CVS. `additional_lt_poc_lsb` định rõ trị số của biến `MaxLtPicOrderCntLsb` được sử dụng trong quy trình giải mã cho các danh sách hình ảnh tham chiếu như dưới đây:

$$\text{MaxLtPicOrderCntLsb} = 2(\log_2_{\text{max_pic_order_cnt_lsb_minus4}} + 4 + \text{additional_st_poc_lsb} + \text{additional_lt_poc_lsb})$$

Trị số của `additional_lt_poc_lsb` sẽ nằm trong dải từ 0 đến $32 - \log_2_{\text{max_pic_order_cnt_lsb_minus4}} - 4 - \text{additional_st_poc_lsb}$. Khi không có mặt, trị số của `additional_lt_poc_lsb` được suy ra bằng 0.

Các ngữ nghĩa tập hợp thông số hình ảnh RBSP.

`num_ref_idx_default_active_minus1[i]` cộng 1, khi *i* bằng 0, định rõ trị số được suy ra của biến `NumRefIdxActive[0]` đối với các lát P hoặc B với `num_ref_idx_active_override_flag` bằng 0, và, khi *i* bằng 1, định rõ trị số được suy ra của `NumRefIdxActive[1]` đối với các lát B với `num_ref_idx_active_override_flag` bằng 0. Trị số của `num_ref_idx_default_active_minus1[i]` sẽ nằm trong dải từ 0 đến 14.

Các ngữ nghĩa phần đầu lát.

Khi có mặt, trị số của mỗi trong số các phần tử cú pháp phần đầu lát `slice_pic_parameter_set_id` và `slice_pic_order_cnt_lsb` sẽ giống nhau trong tất cả các phần đầu lát của hình ảnh được tạo mã. `slice_type` định rõ loại tạo mã của lát theo bảng 7-3.

Bảng 7-3 – Tên của `slice_type`

<code>slice_type</code>	Tên của <code>slice_type</code>
0	B (lát B)
1	P (lát P)
2	I (lát I)

Khi `nal_unit_type` bằng `IRAP_NUT`, tức là, hình ảnh là hình ảnh IRAP, `slice_type` sẽ bằng 2.

`slice_pic_order_cnt_lsb` định rõ số đếm thứ tự hình ảnh modulo `MaxPicOrderCntLsb` cho hình ảnh hiện thời. Độ dài của phần tử cú pháp `slice_pic_order_cnt_lsb` là $\log_2_max_pic_order_cnt_lsb_minus4 + 4$ bit. Trị số của `slice_pic_order_cnt_lsb` sẽ nằm trong dải từ 0 đến `MaxPicOrderCntLsb - 1`. Khi `slice_pic_order_cnt_lsb` không có mặt, `slice_pic_order_cnt_lsb` được suy ra bằng 0. `num_ref_idx_active_override_flag` bằng 1 định rõ rằng phần tử cú pháp `num_ref_idx_active_minus1[ 0 ]` là có mặt đối với các lát P và B và phần tử cú pháp `num_ref_idx_active_minus1[ 1 ]` là có mặt đối với các lát B. `num_ref_idx_active_override_flag` bằng 0 định rõ rằng các phần tử cú pháp `num_ref_idx_active_minus1[ 0 ]` và `num_ref_idx_active_minus1[ 1 ]` không có mặt. `num_ref_idx_active_minus1[ i ]`, khi có mặt, định rõ trị số của biến `NumRefIdxActive[ i ]` như dưới đây:

$$\text{NumRefIdxActive}[i] = \text{num_ref_idx_active_minus1}[i] + 1$$

Trị số của `num_ref_idx_active_minus1[ i ]` sẽ nằm trong dải từ 0 đến 14. Trị số của `NumRefIdxActive[ i ] - 1` định rõ chỉ số tham chiếu lớn nhất đối với danh sách hình ảnh tham chiếu `i` mà có thể được sử dụng để giải mã lát. Khi trị số của `NumRefIdxActive[ i ]` bằng 0, không có chỉ số tham chiếu đối với danh sách hình ảnh tham chiếu `i` có thể được sử dụng để giải mã lát. Đối với `i` bằng 0 hoặc 1, khi lát hiện thời là lát B và `num_ref_idx_active_override_flag` bằng 0, `NumRefIdxActive[ i ]` được suy ra bằng `num_ref_idx_default_active_minus1[ i ] + 1`. Khi lát hiện thời là lát P và `num_ref_idx_active_override_flag` bằng 0, `NumRefIdxActive[ 0 ]` được suy ra bằng `num_ref_idx_default_active_minus1[ 0 ] + 1`. Khi lát hiện thời là lát P, `NumRefIdxActive[ 1 ]` được suy ra bằng 0. Khi lát hiện thời là lát I, cả `NumRefIdxActive[ 0 ]` và `NumRefIdxActive[ 1 ]` được suy ra bằng 0. Theo cách khác, đối với `i` bằng 0 hoặc 1, áp dụng phần dưới đây sau phần nêu trên: Đặt `rplsIdx1` bằng `ref_pic_list_sps_flag[ i ] ? ref_pic_list_idx[ i ] : num_ref_pic_lists_in_sps[ i ]`, và `numRpEntries[ i ]` bằng `num_strp_entries[ i ][ rplsIdx1 ] + num_ltrp_entries[ i ][ rplsIdx1 ]`. Khi `NumRefIdxActive[ i ]` lớn hơn `numRpEntries[ i ]`, trị số của `NumRefIdxActive[ i ]` được thiết đặt bằng `numRpEntries[ i ]`.

Các ngữ nghĩa cấu trúc danh sách hình ảnh tham chiếu.

Cấu trúc cú pháp `ref_pic_list_struct( listIdx, ltrpFlag )` có thể có mặt trong phần đầu lát. Khi nó có mặt trong phần đầu lát, cấu trúc cú pháp `ref_pic_list_struct( listIdx, ltrpFlag )` định rõ danh sách hình ảnh tham chiếu `ListIdx` của hình ảnh hiện thời (hình ảnh chứa lát). `num_strp_entries[ listIdx ]` định rõ số lượng mục nhập STRP trong cấu trúc cú pháp `ref_pic_list_struct( listIdx, ltrpFlag )`. `num_ltrp_entries[ listIdx ]` định rõ số lượng mục nhập LTRP trong cấu trúc cú pháp `ref_pic_list_struct( listIdx, ltrpFlag )`. Khi không có mặt, trị số của `num_ltrp_entries[ listIdx ]` được suy ra bằng 0. Biến `NumEntriesInList[ listIdx ]` được dẫn xuất như dưới đây:

$$\text{NumEntriesInList[listIdx]} = \text{num_strp_entries[listIdx]} + \text{num_ltrp_entries[listIdx]}$$

Trị số của `NumEntriesInList[ listIdx ]` sẽ nằm trong dải từ 0 đến `sps_max_dec_pic_buffering_minus1`. `lt_ref_pic_flag[ listIdx ][ i ]` bằng 1 định rõ rằng mục nhập thứ `i` trong cấu trúc cú pháp `ref_pic_list_struct( listIdx, ltrpFlag )` là mục nhập LTRP. `lt_ref_pic_flag[ listIdx ][ i ]` bằng 0 định rõ rằng mục nhập thứ `i` trong cấu trúc cú pháp `ref_pic_list_struct( listIdx, ltrpFlag )` là mục nhập STRP. Khi không có mặt, trị số của `lt_ref_pic_flag[ listIdx ][ i ]` được suy ra bằng 0. Yêu cầu về sự tương thích dòng bit là tổng của `lt_ref_pic_flag[ listIdx ][ i ]` cho tất cả các trị số của `i` nằm trong dải từ 0 đến `NumEntriesInList[ listIdx ] - 1`, sẽ bằng `num_ltrp_entries[ listIdx ]`. `poc_lsb_st[ listIdx ][ i ]`, khi `lt_ref_pic_flag[ listIdx ][ i ]` bằng 0, định rõ trị số của số đếm thứ tự hình ảnh modulo `MaxStPicOrderCntLsb` của hình ảnh được tham chiếu bởi mục nhập thứ `i` trong cấu trúc cú pháp `ref_pic_list_struct( listIdx, ltrpFlag )`. Độ dài của `poc_lsb_st[ listIdx ][ i ]` phần tử cú pháp là $\text{Log2}(\text{MaxStPicOrderCntLsb})$ bit. `poc_lsb_lt[ listIdx ][ i ]`, khi `lt_ref_pic_flag[ listIdx ][ i ]` bằng 1, định rõ trị số của số đếm thứ tự hình ảnh modulo `MaxLtPicOrderCntLsb` của hình ảnh được tham chiếu bởi mục nhập thứ `i` trong cấu trúc cú pháp `ref_pic_list_struct( listIdx, ltrpFlag )`. Độ dài của `poc_lsb_lt[ listIdx ][ i ]` phần tử cú pháp là $\text{Log2}(\text{MaxLtPicOrderCntLsb})$ bit.

Quy trình giải mã được thảo luận.

Quy trình giải mã chung.

Quy trình giải mã hoạt động như dưới đây cho hình ảnh hiện thời `CurrPic`: Việc giải

mã các đơn vị NAL được định rõ dưới đây. Các quy trình dưới đây định rõ các quy trình giải mã tiếp theo sử dụng các phần tử cú pháp trong phần đầu lát lớp và ở trên: Các biến và các chức năng liên quan đến số đếm thứ tự hình ảnh được dẫn xuất. Điều này cần phải được gọi ra chỉ cho lát thứ nhất của hình ảnh. Ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP, quy trình giải mã cho việc xây dựng các danh sách hình ảnh tham chiếu được gọi ra cho việc dẫn xuất danh sách hình ảnh tham chiếu 0 (`RefPicList[ 0 ]`) và danh sách hình ảnh tham chiếu 1 (`RefPicList[ 1 ]`). Quy trình giải mã cho việc đánh dấu hình ảnh tham chiếu được gọi ra, trong đó các hình ảnh tham chiếu có thể được đánh dấu là "không được sử dụng cho việc tham chiếu" hoặc "được sử dụng cho việc tham chiếu dài hạn". Điều này cần phải được gọi ra chỉ cho lát thứ nhất của hình ảnh. Các quy trình giải mã để tạo mã các đơn vị cây, định tỷ lệ, biến đổi, lọc trong vòng lặp, v.v., được gọi ra. Sau khi tất cả các lát của hình ảnh hiện thời đã được giải mã, hình ảnh được giải mã hiện thời được đánh dấu là "được sử dụng cho việc tham chiếu ngắn hạn".

Quy trình giải mã đơn vị NAL.

Các đầu vào quy trình này là các đơn vị NAL của hình ảnh hiện thời và các đơn vị NAL không phải VCL được kết hợp của chúng. Các đầu ra của quy trình này là các cấu trúc cú pháp RBSP được phân tích được gói gọn trong các đơn vị NAL. Quy trình giải mã cho mỗi đơn vị NAL trích xuất cấu trúc cú pháp RBSP từ đơn vị NAL và sau đó phân tích cấu trúc cú pháp RBSP.

Quy trình giải mã lát.

Quy trình giải mã cho số đếm thứ tự hình ảnh.

Đầu ra của quy trình này là `PicOrderCntVal`, số đếm thứ tự hình ảnh của hình ảnh hiện thời. Các số đếm thứ tự hình ảnh được sử dụng để nhận dạng các hình ảnh, để dẫn xuất các thông số chuyển động trong chế độ hợp nhất và dự đoán vectơ chuyển động, và để kiểm tra độ tương thích của bộ giải mã. Mỗi hình ảnh được tạo mã được kết hợp với biến số đếm thứ tự hình ảnh, được thể hiện là `PicOrderCntVal`. Khi hình ảnh hiện thời không phải hình ảnh IRAP, các biến `prevPicOrderCntLsb` và `prevPicOrderCntMsb` được dẫn xuất như dưới đây: Đặt `prevTid0Pic` là hình ảnh trước đó theo thứ tự giải mã mà có `TemporalId` bằng 0. Biến `prevPicOrderCntLsb` được thiết đặt bằng `slice_pic_order_cnt_lsb` của `prevTid0Pic`. Biến

prevPicOrderCntMsb được thiết đặt bằng PicOrderCntMsb của prevTid0Pic. Biến PicOrderCntMsb của hình ảnh hiện thời được dẫn xuất như dưới đây: Nếu hình ảnh hiện thời là hình ảnh IRAP, thì PicOrderCntMsb được thiết đặt bằng 0. Nếu không thì, PicOrderCntMsb được dẫn xuất như dưới đây:

```

if( ( slice_pic_order_cnt_lsb < prevPicOrderCntLsb ) &&
    ( ( prevPicOrderCntLsb - slice_pic_order_cnt_lsb ) >= ( MaxPicOrderCntLsb /
2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb
else if( ( slice_pic_order_cnt_lsb > prevPicOrderCntLsb ) &&
    ( ( slice_pic_order_cnt_lsb - prevPicOrderCntLsb ) > ( MaxPicOrderCntLsb / 2
) ) )
    PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb
else
    PicOrderCntMsb = prevPicOrderCntMsb

```

PicOrderCntVal được dẫn xuất như dưới đây:

$$\text{PicOrderCntVal} = \text{PicOrderCntMsb} + \text{slice_pic_order_cnt_lsb}$$

Tất cả các hình ảnh IRAP sẽ có PicOrderCntVal bằng 0 vì slice_pic_order_cnt_lsb được suy ra là bằng 0 đối với các hình ảnh IRAP và prevPicOrderCntLsb và prevPicOrderCntMsb đều được thiết đặt bằng 0. Trị số của PicOrderCntVal sẽ nằm trong dải từ -231 đến 231 - 1. Trong một CVS, các trị số PicOrderCntVal đối với hai hình ảnh được tạo mã bất kỳ sẽ không giống nhau. Ở thời điểm bất kỳ trong quy trình giải mã, các trị số của PicOrderCntVal & (MaxStPicOrderCntLsb - 1) đối với hai hình ảnh tham chiếu bất kỳ ngắn hạn trong DPB sẽ không giống nhau. Ở thời điểm bất kỳ trong quy trình giải mã, các trị số của PicOrderCntVal & (MaxLtPicOrderCntLsb - 1) đối với hai hình ảnh tham chiếu bất kỳ trong DPB sẽ không giống nhau.

Hàm PicOrderCnt(picX) được định rõ như dưới đây:

$$\text{PicOrderCnt}(\text{picX}) = \text{PicOrderCntVal của hình ảnh picX}$$

Hàm `DiffPicOrderCnt( picA, picB )` được định rõ như dưới đây:

$$\text{DiffPicOrderCnt}(\text{picA}, \text{picB}) = \text{PicOrderCnt}(\text{picA}) - \text{PicOrderCnt}(\text{picB})$$

Dòng bit sẽ không chứa dữ liệu mà dẫn đến các trị số của `DiffPicOrderCnt( picA, picB )` được sử dụng trong quy trình giải mã mà không nằm trong dải từ -215 đến $215 - 1$. Đặt X là hình ảnh hiện thời và Y và Z là hai hình ảnh khác trong cùng CVS, Y và Z được xem là nằm trong cùng hướng thứ tự đưa ra từ X khi cả `DiffPicOrderCnt( X, Y )` và `DiffPicOrderCnt( X, Z )` là dương hoặc đều là âm.

Quy trình giải mã cho việc xây dựng các danh sách hình ảnh tham chiếu.

Quy trình này được gọi ra ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP. Các hình ảnh tham chiếu được gửi qua các chỉ số tham chiếu. Chỉ số tham chiếu là chỉ số trong danh sách hình ảnh tham chiếu. Khi giải mã lát I , không có danh sách hình ảnh tham chiếu nào được sử dụng trong việc giải mã dữ liệu lát. Khi giải mã lát P , chỉ danh sách hình ảnh tham chiếu 0 (tức là, `RefPicList[ 0 ]`), được sử dụng trong việc giải mã dữ liệu lát. Khi giải mã lát B , cả danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1 (tức là, `RefPicList[ 1 ]`) được sử dụng trong việc giải mã dữ liệu lát. Ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP, các danh sách hình ảnh tham chiếu `RefPicList[ 0 ]` và `RefPicList[ 1 ]` được dẫn xuất. Các danh sách hình ảnh tham chiếu được sử dụng trong việc đánh dấu các hình ảnh tham chiếu hoặc trong việc giải mã dữ liệu lát. Đối với lát I của hình ảnh không phải IRAP mà không phải là lát thứ nhất của hình ảnh, `RefPicList[ 0 ]` và `RefPicList[ 1 ]` có thể được dẫn xuất cho mục đích kiểm tra độ tương thích dòng bit, nhưng việc suy ra của chúng không cần thiết cho việc giải mã hình ảnh hiện thời hoặc các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Đối với lát P mà không phải là lát thứ nhất của hình ảnh, `RefPicList[ 1 ]` có thể được dẫn xuất cho mục đích kiểm tra độ tương thích dòng bit, nhưng sự dẫn xuất của nó không cần thiết cho việc giải mã hình ảnh hiện thời hoặc các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã.

Các danh sách hình ảnh tham chiếu `RefPicList[ 0 ]` và `RefPicList[ 1 ]` được xây dựng như dưới đây:

```
for( i = 0; i < 2; i++ ) {
    for( j = 0; j < NumEntriesInList[ i ]; j++ ) {
```

```

    if( lt_ref_pic_flag[ i ][ j ] ) {
        if( there is a reference picA in the DPB with
        PicOrderCntVal & ( MaxLtPicOrderCntLsb - 1 )
            equal to poc_lsb_lt[ i ][ j ] )
            RefPicList[ i ][ j ] = picA
        else
            RefPicList[ i ][ j ] = “no reference picture”
    }
}
}
for( i = 0; i < 2; i++ ) {
    for( j = 0; j < NumEntriesInList[ i ]; j++ ) {
        if( !lt_ref_pic_flag[ i ][ j ] ) {
            if( there is a short-term reference picture picA in the DPB
                with PicOrderCntVal & ( MaxStPicOrderCntLsb - 1 ) equal to
                poc_lsb_st[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = “no reference picture”
        }
    }
}
}

```

Đối với mỗi i bằng 0 hoặc 1, áp dụng phần dưới đây:

Các mục nhập NumRefIdxActive[i] thứ nhất trong RefPicList[i] được đề cập đến là các mục nhập hoạt động trong RefPicList[i], và các mục nhập khác trong RefPicList[i] được đề cập đến là các mục nhập không hoạt động trong RefPicList[i]. Mỗi mục nhập trong RefPicList[i][j] đối với j nằm trong dải từ 0 đến NumEntriesInList[i] - 1, được đề cập đến là mục nhập STRP nếu lt_ref_pic_flag[i][j] bằng 0, và nếu không thì là mục nhập LTRP. Có khả năng rằng hình ảnh cụ thể được tham chiếu bởi cả mục nhập trong

RefPicList[0] và mục nhập trong RefPicList[1]. Cũng có thể rằng hình ảnh cụ thể được tham chiếu bởi nhiều hơn một mục nhập trong RefPicList[0] hoặc bởi nhiều hơn một mục nhập trong RefPicList[1]. Các mục nhập hoạt động trong RefPicList[0] và các mục nhập hoạt động trong RefPicList[1] đề cập chung tới tất cả các hình ảnh tham chiếu mà có thể được sử dụng cho việc dự đoán liên ảnh của hình ảnh hiện thời và một hoặc nhiều hình ảnh mà tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Các mục nhập không hoạt động trong RefPicList[0] và các mục nhập không hoạt động trong RefPicList[1] đề cập chung tới tất cả các hình ảnh tham chiếu không được sử dụng cho việc dự đoán liên ảnh của hình ảnh hiện thời nhưng có thể được sử dụng trong việc dự đoán liên ảnh cho một hoặc nhiều hình ảnh mà tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Có thể là một hoặc nhiều mục nhập trong RefPicList[0] hoặc RefPicList[1] mà bằng "không có hình ảnh tham chiếu" bởi vì các hình ảnh tương ứng không có mặt trong DPB. Mỗi mục nhập không hoạt động trong RefPicList[0] hoặc RefPicList[0] bằng "không có hình ảnh tham chiếu" sẽ được bỏ qua. Sự mất hình ảnh vô tình sẽ được suy ra cho mỗi mục nhập hoạt động trong RefPicList[0] hoặc RefPicList[1] mà bằng "không có hình ảnh tham chiếu".

Yêu cầu về sự tương thích dòng bit áp dụng các ràng buộc sau đây: Đối với mỗi i bằng 0 hoặc 1, NumEntriesInList[i] sẽ không nhỏ hơn NumRefIdxActive[i]. Hình ảnh được tham chiếu bởi mỗi mục nhập hoạt động trong RefPicList[0] hoặc RefPicList[1] sẽ có mặt trong DPB và sẽ có TemporalId nhỏ hơn hoặc bằng TemporalId của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: chỉ số mục nhập của mục nhập không hoạt động bất kỳ trong RefPicList[0] hoặc RefPicList[1] sẽ không được sử dụng làm chỉ số tham chiếu để giải mã của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: mục nhập không hoạt động trong RefPicList[0] hoặc RefPicList[1] sẽ không tham chiếu đến hình ảnh giống như mục nhập khác bất kỳ trong RefPicList[0] hoặc RefPicList[1]. Mục nhập STRP trong RefPicList[0] hoặc RefPicList[1] của lát của hình ảnh và mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1] của cùng lát hoặc lát khác nhau của cùng hình ảnh sẽ không tham chiếu đến cùng hình ảnh. Chính hình ảnh hiện thời sẽ không được tham chiếu bởi mục nhập bất kỳ trong RefPicList[0] hoặc RefPicList[1]. Sẽ không có mục nhập LTRP trong RefPicList[0] hoặc RefPicList[1] mà sự khác biệt giữa PicOrderCntVal của hình ảnh hiện thời và PicOrderCntVal của hình ảnh được tham chiếu

bởi mục nhập lớn hơn hoặc bằng 224. Đặt `setOfRefPics` là tập hợp các hình ảnh duy nhất được tham chiếu bởi tất cả các mục nhập trong `RefPicList[ 0 ]` và tất cả các mục nhập trong `RefPicList[ 1 ]`. Số lượng của các hình ảnh trong `setOfRefPics` sẽ nhỏ hơn hoặc bằng `sps_max_dec_pic_buffering_minus1` và `setOfRefPics` sẽ giống nhau đối với tất cả các lát của hình ảnh.

Quy trình giải mã cho việc đánh dấu hình ảnh tham chiếu.

Quy trình này được gọi ra một lần cho mỗi hình ảnh, sau khi giải mã phần đầu lát và quy trình giải mã cho việc xây dựng danh sách hình ảnh tham chiếu cho lát, nhưng trước việc giải mã dữ liệu lát. Quy trình này có thể dẫn đến một hoặc nhiều hình ảnh tham chiếu trong DPB được đánh dấu là "không được sử dụng cho việc tham chiếu" hoặc "được sử dụng cho việc tham chiếu dài hạn". Hình ảnh được giải mã trong DPB có thể được đánh dấu là "không được sử dụng cho việc tham chiếu", "được sử dụng cho việc tham chiếu ngắn hạn" hoặc "được sử dụng cho việc tham chiếu dài hạn", nhưng chỉ một trong số ba lựa chọn này ở thời điểm xác định bất kỳ trong suốt hoạt động của quy trình giải mã. Việc gán một trong số các dấu này cho hình ảnh loại bỏ hoàn toàn các sự đánh dấu còn lại khi thích hợp. Khi hình ảnh được đề cập đến là được đánh dấu là "được sử dụng cho việc tham chiếu", điều này đề cập chung đến hình ảnh được đánh dấu là "được sử dụng cho việc tham chiếu ngắn hạn" hoặc "được sử dụng cho việc tham chiếu dài hạn" (nhưng không phải là cả hai). Khi hình ảnh hiện thời là hình ảnh IRAP, tất cả các hình ảnh tham chiếu hiện trong DPB (nếu có) được đánh dấu là "không được sử dụng cho việc tham chiếu". Các STRP được nhận dạng bởi $\text{Log}_2(\text{MaxStPicOrderCntLsb})$ LSB của các trị số `PicOrderCntVal` của chúng. Các LTRP được nhận dạng bởi $\text{Log}_2(\text{MaxLtPicOrderCntLsb})$ LSB của các trị số `PicOrderCntVal` của chúng.

Áp dụng phần dưới đây: Đối với mỗi mục nhập LTRP trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]`, khi hình ảnh được tham chiếu là STRP, hình ảnh được đánh dấu là "được sử dụng cho việc tham chiếu dài hạn". Mỗi hình ảnh tham chiếu trong DPB mà không được tham chiếu bởi mục nhập bất kỳ trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` được đánh dấu là "không được sử dụng cho việc tham chiếu".

Luôn báo hiệu các danh sách hình ảnh tham chiếu trong các phần đầu lát mà không

có sự khác biệt giữa các hình ảnh tham chiếu ngắn hạn và dài hạn.

Phần này mô tả phương án khác của sáng chế. Phần mô tả này liên quan tới VVC WD cuối cùng (tức là, chỉ delta liên quan tới VVC WD cuối cùng trong JVET-K1001-v1 được mô tả, trong khi các văn bản trong VVC WD cuối cùng mà không được đề cập dưới đây áp dụng như hiện tại). Phương án khác này được tóm tắt như dưới đây: Các cấu trúc danh sách hình ảnh tham chiếu được báo hiệu chỉ trong các phần đầu lát. Không có sự khác biệt được tạo ra giữa các hình ảnh tham chiếu ngắn hạn và dài hạn. Tất cả các hình ảnh tham chiếu chỉ là các hình ảnh tham chiếu được đặt tên. Các hình ảnh tham chiếu được nhận dạng bởi các POC LSB của chúng, mà có thể được thể hiện bởi số lượng của các bit mà khác với số lượng của các bit được sử dụng để thể hiện các POC LSB được báo hiệu trong các phần đầu lát để dẫn xuất các trị số POC.

Các từ viết tắt. Áp dụng văn bản trong khoản 4 của VVC WD.

Cú pháp phần đầu đơn vị NAL.

nal_unit_header() {	Ký hiệu mô tả
forbidden_zero_bit	f(1)
nal_unit_type	u(5)
nuh_temporal_id_plus1	u(3)
nuh_reserved_zero_7bits	u(7)
}	

Cú pháp tập hợp thông số chuỗi RBSP.

seq_parameter_set_rbsp() {	Ký hiệu mô tả
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)

pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
additional_ref_poc_lsb	ue(v)
rbsp_trailing_bits()	
}	

Cú pháp tập hợp thông số hình ảnh RBSP.

pic_parameter_set_rbsp() {	Ký hiệu mô tả
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rbsp_trailing_bits()	
}	

Cú pháp phần đầu lát.

slice_header() {	Ký hiệu mô tả
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++)	
ref_pic_list_struct(i)	
if(slice_type == P slice_type == B) {	
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++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Cú pháp cấu trúc danh sách hình ảnh tham chiếu.

ref_pic_list_struct(listIdx) {	Ký hiệu mô tả
num_ref_entries[listIdx]	ue(v)
for(i = 0; i < NumEntriesInList[listIdx]; i++)	
poc_ref_lsb[listIdx][i]	u(v)

}	
---	--

Các ngữ nghĩa phần đầu đơn vị NAL.

forbidden_zero_bit sẽ bằng 0. nal_unit_type định rõ loại của cấu trúc dữ liệu RBSP được chứa trong đơn vị NAL.

Bảng 7-1 – các mã loại đơn vị NAL và các lớp loại đơn vị NAL

nal_unit_type	Tên của nal_unit_type	Nội dung của đơn vị NAL và cấu trúc cú pháp RBSP	Lớp loại đơn vị NAL
0	NON_IRAP_NUT	Phân đoạn lát được tạo mã của hình ảnh không phải IRAP slice_layer_rbsp()	VCL
1	IRAP_NUT	Lát được tạo mã của hình ảnh IRAP slice_layer_rbsp()	VCL
2-15	RSV_VCL_NUT	Các đơn vị NAL VCL dành riêng	VCL
16	SPS_NUT	Tập hợp thông số chuỗi seq_parameter_set_rbsp()	không phải VCL
17	PPS_NUT	Tập hợp thông số hình ảnh pic_parameter_set_rbsp()	không phải VCL
18	EOS_NUT	Phần cuối của chuỗi end_of_seq_rbsp()	không phải VCL
19	EOB_NUT	Phần cuối của dòng bit end_of_bitstream_rbsp()	không phải VCL
20, 21	PREFIX_SEI_NUT SUFFIX_SEI_NUT	Thông tin nâng cao bổ sung sei_rbsp()	không phải VCL
22-26	RSV_NVCL	Dành riêng	không phải VCL

27-31	UNSPEC	Không được định rõ	không phải VCL
-------	--------	--------------------	-------------------

nuh_temporal_id_plus1 trừ 1 định rõ ký hiệu nhận dạng tạm thời cho đơn vị NAL. Trị số của nuh_temporal_id_plus1 sẽ không bằng 0. Biến TemporalId được định rõ như dưới đây:

$$\text{TemporalId} = \text{nuh_temporal_id_plus1} - 1$$

Khi nal_unit_type bằng IRAP_NUT, lát được tạo mã thuộc về hình ảnh IRAP, TemporalId sẽ bằng 0. Trị số của TemporalId sẽ giống nhau đối với tất cả các đơn vị NAL VCL của đơn vị truy cập. Trị số của TemporalId của hình ảnh được tạo mã hoặc đơn vị truy cập là trị số của TemporalId của các đơn vị NAL VCL của hình ảnh được tạo mã hoặc đơn vị truy cập. Trị số của TemporalId cho các đơn vị NAL không phải VCL bị ràng buộc như dưới đây:

Nếu nal_unit_type bằng SPS_NUT, thì TemporalId sẽ bằng 0 và TemporalId của đơn vị truy cập chứa đơn vị NAL sẽ bằng 0. Nếu không thì, nếu nal_unit_type bằng EOS_NUT hoặc EOB_NUT, thì TemporalId sẽ bằng 0. Nếu không thì, TemporalId sẽ lớn hơn hoặc bằng TemporalId của đơn vị truy cập chứa đơn vị NAL. Khi đơn vị NAL là đơn vị NAL không phải VCL, trị số của TemporalId bằng trị số nhỏ nhất của các trị số TemporalId của tất cả các đơn vị truy cập mà đơn vị NAL không phải VCL áp dụng. Khi nal_unit_type bằng PPS_NUT, TemporalId có thể lớn hơn hoặc bằng TemporalId của đơn vị truy cập chứa, vì tất cả các tập hợp thông số hình ảnh (Picture Parameter Set, viết tắt là PPS) có thể được bao gồm trong phần đầu của dòng bit, trong đó hình ảnh được tạo mã thứ nhất có TemporalId bằng 0. Khi nal_unit_type bằng PREFIX_SEI_NUT hoặc SUFFIX_SEI_NUT, TemporalId có thể lớn hơn hoặc bằng TemporalId của đơn vị truy cập chứa, vì đơn vị NAL SEI có thể chứa thông tin áp dụng cho tập con dòng bit bao gồm các đơn vị truy cập mà các trị số TemporalId lớn hơn TemporalId của đơn vị truy cập chứa đơn vị NAL SEI. nuh_reserved_zero_7bits sẽ bằng '0000000'. Các trị số khác của nuh_reserved_zero_7bits có thể được định rõ trong tương lai bởi ITU-T | ISO/IEC. Các bộ giải mã sẽ bỏ qua (tức là, loại bỏ khỏi dòng bit và loại trừ) các đơn vị NAL có các trị số của nuh_reserved_zero_7bits không bằng '0000000'.

Các ngữ nghĩa tập hợp thông số chuỗi RBSP.

$\log2_max_pic_order_cnt_lsb_minus4$ định rõ trị số của biến $MaxPicOrderCntLsb$ được sử dụng trong quy trình giải mã cho số đếm thứ tự hình ảnh như dưới đây:

$$MaxPicOrderCntLsb = 2(\log2_max_pic_order_cnt_lsb_minus4 + 4)$$

Trị số của $\log2_max_pic_order_cnt_lsb_minus4$ sẽ nằm trong dải từ 0 đến 12. $sps_max_dec_pic_buffering_minus1$ cộng 1 định rõ kích thước yêu cầu lớn nhất của bộ đệm hình ảnh được giải mã cho CVS trong các đơn vị của các bộ đệm lưu trữ hình ảnh. Trị số của $sps_max_dec_pic_buffering_minus1$ sẽ nằm trong dải từ 0 đến $MaxDpbSize - 1$, trong đó $MaxDpbSize$ như được định rõ ở phần khác. $additional_ref_poc_lsb$ định rõ trị số của biến $MaxRefPicOrderCntLsb$ được sử dụng trong quy trình giải mã cho các danh sách hình ảnh tham chiếu như dưới đây:

$$MaxRefPicOrderCntLsb = 2(\log2_max_pic_order_cnt_lsb_minus4 + 4 + additional_ref_poc_lsb)$$

Trị số của $additional_ref_poc_lsb$ sẽ nằm trong dải từ 0 đến $32 - \log2_max_pic_order_cnt_lsb_minus4 - 4$.

Các ngữ nghĩa tập hợp thông số hình ảnh RBSP.

$num_ref_idx_default_active_minus1[i]$ cộng 1, khi i bằng 0, định rõ trị số được suy ra của biến $NumRefIdxActive[0]$ đối với các lát P hoặc B với $num_ref_idx_active_override_flag$ bằng 0, và, khi i bằng 1, định rõ trị số được suy ra của $NumRefIdxActive[1]$ đối với các lát B với $num_ref_idx_active_override_flag$ bằng 0. Trị số của $num_ref_idx_default_active_minus1[i]$ sẽ nằm trong dải từ 0 đến 14.

Các ngữ nghĩa phần đầu lát.

Khi có mặt, trị số của mỗi trong số các phần tử cú pháp phần đầu lát $slice_pic_parameter_set_id$ và $slice_pic_order_cnt_lsb$ sẽ giống nhau trong tất cả các phần đầu lát của hình ảnh được tạo mã. ... $slice_type$ định rõ loại tạo mã của lát theo bảng 7-3.

Bảng 7-3 – Tên của $slice_type$

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

Khi `nal_unit_type` bằng `IRAP_NUT`, tức là, hình ảnh là hình ảnh IRAP, `slice_type` sẽ bằng 2. ... `slice_pic_order_cnt_lsb` định rõ số đếm thứ tự hình ảnh modulo `MaxPicOrderCntLsb` cho hình ảnh hiện thời. Độ dài của phần tử cú pháp `slice_pic_order_cnt_lsb` là `log2_max_pic_order_cnt_lsb_minus4 + 4` bit. Trị số của `slice_pic_order_cnt_lsb` sẽ nằm trong dải từ 0 đến `MaxPicOrderCntLsb - 1`. Khi `slice_pic_order_cnt_lsb` không có mặt, `slice_pic_order_cnt_lsb` được suy ra bằng 0. `num_ref_idx_active_override_flag` bằng 1 định rõ rằng phần tử cú pháp `num_ref_idx_active_minus1[0]` là có mặt đối với các lát P và B và phần tử cú pháp `num_ref_idx_active_minus1[1]` là có mặt đối với các lát B. `num_ref_idx_active_override_flag` bằng 0 định rõ rằng các phần tử cú pháp `num_ref_idx_active_minus1[0]` và `num_ref_idx_active_minus1[1]` không có mặt. `num_ref_idx_active_minus1[i]`, khi có mặt, định rõ trị số của biến `NumRefIdxActive[i]` như dưới đây:

$$\text{NumRefIdxActive}[i] = \text{num_ref_idx_active_minus1}[i] + 1$$

Trị số của `num_ref_idx_active_minus1[i]` sẽ nằm trong dải từ 0 đến 14. Trị số của `NumRefIdxActive[i] - 1` định rõ chỉ số tham chiếu lớn nhất đối với danh sách hình ảnh tham chiếu `i` mà có thể được sử dụng để giải mã lát. Khi trị số của `NumRefIdxActive[i]` bằng 0, không có chỉ số tham chiếu đối với danh sách hình ảnh tham chiếu `i` có thể được sử dụng để giải mã lát. Đối với `i` bằng 0 hoặc 1, khi lát hiện thời là lát B và `num_ref_idx_active_override_flag` bằng 0, `NumRefIdxActive[i]` được suy ra bằng `num_ref_idx_default_active_minus1[i] + 1`. Khi lát hiện thời là lát P và `num_ref_idx_active_override_flag` bằng 0, `NumRefIdxActive[0]` được suy ra bằng `num_ref_idx_default_active_minus1[0] + 1`. Khi lát hiện thời là lát P, `NumRefIdxActive[1]` được suy ra bằng 0. Khi lát hiện thời là lát I, cả `NumRefIdxActive[0]` và `NumRefIdxActive[1]` được suy ra bằng 0. Theo cách khác, đối với `i` bằng 0 hoặc 1, áp

dụng phần dưới đây sau phần nêu trên: Đặt $rplsIdx1$ bằng $ref_pic_list_sps_flag[i]$?
 $ref_pic_list_idx[i]$: $num_ref_pic_lists_in_sps[i]$, và $numRpEntries[i]$ bằng
 $num_strp_entries[i][rplsIdx1] + num_ltrp_entries[i][rplsIdx1]$. Khi
 $NumRefIdxActive[i]$ lớn hơn $numRpEntries[i]$, trị số của $NumRefIdxActive[i]$ được thiết
đặt bằng $numRpEntries[i]$.

Các ngữ nghĩa cấu trúc danh sách hình ảnh tham chiếu.

Cấu trúc cú pháp $ref_pic_list_struct(listIdx)$ có thể có mặt trong phần đầu lát. Khi nó
có mặt trong phần đầu lát, cấu trúc cú pháp $ref_pic_list_struct(listIdx)$ định rõ danh sách
hình ảnh tham chiếu $ListIdx$ của hình ảnh hiện thời (hình ảnh chứa lát).
 $num_ref_entries[listIdx]$ định rõ số lượng mục nhập trong cấu trúc cú pháp
 $ref_pic_list_struct(listIdx)$. Biến $NumEntriesInList[listIdx]$ được dẫn xuất như dưới đây:

$$NumRefPicEntriesInRpl[listIdx] = num_ref_entries[listIdx]$$

Trị số của $NumRefPicEntries[listIdx]$ sẽ nằm trong dải từ 0 đến
 $sps_max_dec_pic_buffering_minus1.poc_ref_lsb[listIdx][i]$ định rõ trị số của số đếm thứ
tự hình ảnh modulo $MaxRefPicOrderCntLsb$ của hình ảnh được tham chiếu bởi mục nhập
thứ i trong cấu trúc cú pháp $ref_pic_list_struct(listIdx)$. Độ dài của $poc_ref_lsb[listIdx][i]$
phần tử cú pháp là $\log_2(MaxRefPicOrderCntLsb)$ bit.

Quy trình giải mã được thảo luận.

Quy trình giải mã chung.

Quy trình giải mã hoạt động như dưới đây cho hình ảnh hiện thời $CurrPic$: việc giải
mã các đơn vị NAL được định rõ dưới đây. Các quy trình dưới đây định rõ các quy trình giải
mã tiếp theo sử dụng các phần tử cú pháp trong phần đầu lát lớp và ở trên: các biến và các
chức năng liên quan đến số đếm thứ tự hình ảnh được dẫn xuất. Điều này cần phải được gọi
ra chỉ cho lát thứ nhất của hình ảnh. Ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của
hình ảnh không phải IRAP, quy trình giải mã cho việc xây dựng các danh sách hình ảnh tham
chiếu được gọi ra cho việc dẫn xuất danh sách hình ảnh tham chiếu 0 ($RefPicList[0]$) và
danh sách hình ảnh tham chiếu 1 ($RefPicList[1]$). Quy trình giải mã cho việc đánh dấu hình
ảnh tham chiếu được gọi ra, trong đó các hình ảnh tham chiếu có thể được đánh dấu là "không

được sử dụng cho việc tham chiếu". Điều này cần phải được gọi ra chỉ cho lát thứ nhất của hình ảnh. Các quy trình giải mã để tạo mã các đơn vị cây, định tỷ lệ, biến đổi, lọc trong vòng lặp, v.v., được gọi ra. Sau khi tất cả các lát của hình ảnh hiện thời đã được giải mã, hình ảnh được giải mã hiện thời được đánh dấu là "được sử dụng cho việc tham chiếu".

Quy trình giải mã đơn vị NAL.

Các đầu vào quy trình này là các đơn vị NAL của hình ảnh hiện thời và các đơn vị NAL không phải VCL được kết hợp của chúng. Các đầu ra của quy trình này là các cấu trúc cú pháp RBSP được phân tích được gói gọn trong các đơn vị NAL. Quy trình giải mã cho mỗi đơn vị NAL trích xuất cấu trúc cú pháp RBSP từ đơn vị NAL và sau đó phân tích cấu trúc cú pháp RBSP.

Quy trình giải mã lát.

Quy trình giải mã cho số đếm thứ tự hình ảnh.

Đầu ra của quy trình này là PicOrderCntVal, số đếm thứ tự hình ảnh của hình ảnh hiện thời. Các số đếm thứ tự hình ảnh được sử dụng để nhận dạng các hình ảnh, để dẫn xuất các thông số chuyển động trong chế độ hợp nhất và dự đoán vector chuyển động, và để kiểm tra độ tương thích của bộ giải mã. Mỗi hình ảnh được tạo mã được kết hợp với biến số đếm thứ tự hình ảnh, được thể hiện là PicOrderCntVal. Khi hình ảnh hiện thời không phải hình ảnh IRAP, các biến prevPicOrderCntLsb và prevPicOrderCntMsb được dẫn xuất như dưới đây: đặt prevTid0Pic là hình ảnh trước đó theo thứ tự giải mã mà có TemporalId bằng 0. Biến prevPicOrderCntLsb được thiết đặt bằng slice_pic_order_cnt_lsb của prevTid0Pic. Biến prevPicOrderCntMsb được thiết đặt bằng PicOrderCntMsb của prevTid0Pic. Biến PicOrderCntMsb của hình ảnh hiện thời được dẫn xuất như dưới đây: nếu hình ảnh hiện thời là hình ảnh IRAP, thì PicOrderCntMsb được thiết đặt bằng 0. Nếu không thì, PicOrderCntMsb được dẫn xuất như dưới đây:

```
if( ( slice_pic_order_cnt_lsb < prevPicOrderCntLsb ) &&
    ( ( prevPicOrderCntLsb - slice_pic_order_cnt_lsb ) >= ( MaxPicOrderCntLsb /
2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb
```

```

else if( (slice_pic_order_cnt_lsb > prevPicOrderCntLsb) &&
        ((slice_pic_order_cnt_lsb - prevPicOrderCntLsb) > (MaxPicOrderCntLsb / 2
        )))
    PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb
else
    PicOrderCntMsb = prevPicOrderCntMsb

```

PicOrderCntVal được dẫn xuất như dưới đây:

$$\text{PicOrderCntVal} = \text{PicOrderCntMsb} + \text{slice_pic_order_cnt_lsb}$$

Tất cả các hình ảnh IRAP sẽ có PicOrderCntVal bằng 0 vì slice_pic_order_cnt_lsb được suy ra là bằng 0 đối với các hình ảnh IRAP và prevPicOrderCntLsb và prevPicOrderCntMsb đều được thiết đặt bằng 0. Trị số của PicOrderCntVal sẽ nằm trong dải từ -231 đến $231 - 1$. Trong một CVS, các trị số PicOrderCntVal đối với hai hình ảnh được tạo mã bất kỳ sẽ không giống nhau. Ở thời điểm bất kỳ trong quy trình giải mã, các trị số của PicOrderCntVal & $(\text{MaxRefPicOrderCntLsb} - 1)$ đối với hai hình ảnh tham chiếu bất kỳ trong DPB sẽ không giống nhau.

Hàm PicOrderCnt(picX) được định rõ như dưới đây:

$$\text{PicOrderCnt}(\text{picX}) = \text{PicOrderCntVal của hình ảnh picX}$$

Hàm DiffPicOrderCnt(picA, picB) được định rõ như dưới đây:

$$\text{DiffPicOrderCnt}(\text{picA}, \text{picB}) = \text{PicOrderCnt}(\text{picA}) - \text{PicOrderCnt}(\text{picB})$$

Dòng bit sẽ không chứa dữ liệu mà dẫn đến các trị số của DiffPicOrderCnt(picA, picB) được sử dụng trong quy trình giải mã không nằm trong dải từ -215 đến $215 - 1$. Đặt X là hình ảnh hiện thời và Y và Z là hai hình ảnh khác trong cùng CVS, Y và Z được xem là nằm trong cùng hướng thứ tự đưa ra từ X khi cả DiffPicOrderCnt(X, Y) và DiffPicOrderCnt(X, Z) là dương hoặc đều là âm.

Quy trình giải mã cho việc xây dựng các danh sách hình ảnh tham chiếu.

Quy trình này được gọi ra ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP. Các hình ảnh tham chiếu được gửi qua các chỉ số tham chiếu. Chỉ số

tham chiếu là chỉ số trong danh sách hình ảnh tham chiếu. Khi giải mã lát I, không có danh sách hình ảnh tham chiếu nào được sử dụng trong việc giải mã dữ liệu lát. Khi giải mã lát P, chỉ danh sách hình ảnh tham chiếu 0 (tức là, `RefPicList[ 0 ]`), được sử dụng trong việc giải mã dữ liệu lát. Khi giải mã lát B, cả danh sách hình ảnh tham chiếu 0 và danh sách hình ảnh tham chiếu 1 (tức là, `RefPicList[ 1 ]`) được sử dụng trong việc giải mã dữ liệu lát. Ở vị trí bắt đầu của quy trình giải mã cho mỗi lát của hình ảnh không phải IRAP, các danh sách hình ảnh tham chiếu `RefPicList[ 0 ]` và `RefPicList[ 1 ]` được dẫn xuất. Các danh sách hình ảnh tham chiếu được sử dụng trong việc đánh dấu các hình ảnh tham chiếu hoặc trong việc giải mã dữ liệu lát. Đối với lát I của hình ảnh không phải IRAP mà không phải là lát thứ nhất của hình ảnh, `RefPicList[ 0 ]` và `RefPicList[ 1 ]` có thể được dẫn xuất cho mục đích kiểm tra độ tương thích dòng bit, nhưng việc suy ra của chúng không cần thiết cho việc giải mã hình ảnh hiện thời hoặc các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Đối với lát P mà không phải là lát thứ nhất của hình ảnh, `RefPicList[ 1 ]` có thể được dẫn xuất cho mục đích kiểm tra độ tương thích dòng bit, nhưng sự dẫn xuất của nó không cần thiết cho việc giải mã hình ảnh hiện thời hoặc các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Các danh sách hình ảnh tham chiếu `RefPicList[ 0 ]` và `RefPicList[ 1 ]` được xây dựng như dưới đây:

```
for( i = 0; i < 2; i++ ) {
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ]; j++ ) {
        if( there is a reference picA in the DPB with
            PicOrderCntVal & ( MaxRefPicOrderCntLsb - 1 )
                equal to poc_ref_lsb[ i ][ j ] )
            RefPicList[ i ][ j ] = picA
        else
            RefPicList[ i ][ j ] = "no reference picture"
    }
}
```

Đối với mỗi i bằng 0 hoặc 1, các mục nhập `NumRefIdxActive[ i ]` thứ nhất trong `RefPicList[ i ]` được đề cập đến là các mục nhập hoạt động trong `RefPicList[ i ]`, và các mục

nhập khác trong `RefPicList[ i ]` được đề cập đến là các mục nhập không hoạt động trong `RefPicList[ i ]`. Có khả năng rằng hình ảnh cụ thể được tham chiếu bởi cả mục nhập trong `RefPicList[ 0 ]` và mục nhập trong `RefPicList[ 1 ]`. Cũng có thể rằng hình ảnh cụ thể được tham chiếu bởi nhiều hơn một mục nhập trong `RefPicList[ 0 ]` hoặc bởi nhiều hơn một mục nhập trong `RefPicList[ 1 ]`. Các mục nhập hoạt động trong `RefPicList[ 0 ]` và các mục nhập hoạt động trong `RefPicList[ 1 ]` đề cập chung tới tất cả các hình ảnh tham chiếu mà có thể được sử dụng cho việc dự đoán liên ảnh của hình ảnh hiện thời và một hoặc nhiều hình ảnh mà tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Các mục nhập không hoạt động trong `RefPicList[ 0 ]` và các mục nhập không hoạt động trong `RefPicList[ 1 ]` đề cập chung tới tất cả các hình ảnh tham chiếu không được sử dụng cho việc dự đoán liên ảnh của hình ảnh hiện thời nhưng có thể được sử dụng trong việc dự đoán liên ảnh cho một hoặc nhiều hình ảnh mà tiếp theo hình ảnh hiện thời theo thứ tự giải mã. Có thể là một hoặc nhiều mục nhập trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` mà bằng "không có hình ảnh tham chiếu" bởi vì các hình ảnh tương ứng không có mặt trong DPB. Mỗi mục nhập không hoạt động trong `RefPicList[ 0 ]` hoặc `RefPicList[ 0 ]` bằng "không có hình ảnh tham chiếu" sẽ được bỏ qua. Sự mất hình ảnh vô tình sẽ được suy ra cho mỗi mục nhập hoạt động trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` mà bằng "không có hình ảnh tham chiếu".

Yêu cầu về sự tương thích dòng bit áp dụng các ràng buộc sau đây: Đối với mỗi i bằng 0 hoặc 1, `NumEntriesInList[ i ]` sẽ không nhỏ hơn `NumRefIdxActive[ i ]`. Hình ảnh được tham chiếu bởi mỗi mục nhập hoạt động trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` sẽ có mặt trong DPB và sẽ có `TemporalId` nhỏ hơn hoặc bằng `TemporalId` của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: Chỉ số mục nhập của mục nhập không hoạt động bất kỳ trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` sẽ không được sử dụng làm chỉ số tham chiếu để giải mã của hình ảnh hiện thời. Một cách tùy ý, ràng buộc sau đây có thể được định rõ thêm: Mục nhập không hoạt động trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` sẽ không tham chiếu đến hình ảnh giống như mục nhập khác bất kỳ trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]`. Chính hình ảnh hiện thời sẽ không được tham chiếu bởi mục nhập bất kỳ trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]`. Sẽ không có mục nhập trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` mà sự khác biệt giữa `PicOrderCntVal` của hình ảnh hiện thời và `PicOrderCntVal` của hình ảnh được tham chiếu bởi mục nhập lớn hơn hoặc bằng 224.

Đặt `setOfRefPics` là tập hợp các hình ảnh duy nhất được tham chiếu bởi tất cả các mục nhập trong `RefPicList[ 0 ]` và tất cả các mục nhập trong `RefPicList[ 1 ]`. Số lượng của các hình ảnh trong `setOfRefPics` sẽ nhỏ hơn hoặc bằng `sps_max_dec_pic_buffering_minus1` và `setOfRefPics` sẽ giống nhau đối với tất cả các lát của hình ảnh.

Quy trình giải mã cho việc đánh dấu hình ảnh tham chiếu.

Quy trình này được gọi ra một lần cho mỗi hình ảnh, sau khi giải mã phần đầu lát và quy trình giải mã cho việc xây dựng danh sách hình ảnh tham chiếu cho lát, nhưng trước việc giải mã dữ liệu lát. Quy trình này có thể dẫn đến một hoặc nhiều hình ảnh tham chiếu trong DPB được đánh dấu là "không được sử dụng cho việc tham chiếu". Hình ảnh được giải mã trong DPB có thể được đánh dấu là "không được sử dụng cho việc tham chiếu" hoặc "được sử dụng cho việc tham chiếu", nhưng chỉ một trong số hai lựa chọn này ở thời điểm xác định bất kỳ trong suốt hoạt động của quy trình giải mã. Việc gán một trong số các dấu này cho hình ảnh loại bỏ hoàn toàn các sự đánh dấu còn lại khi thích hợp. Khi hình ảnh hiện thời là hình ảnh IRAP, tất cả các hình ảnh tham chiếu hiện trong DPB (nếu có) được đánh dấu là "không được sử dụng cho việc tham chiếu". Các hình ảnh tham chiếu trong DPB được nhận dạng bởi $\text{Log2}(\text{MaxRefPicOrderCntLsb})$ LSB của các trị số `PicOrderCntVal` của chúng. Mỗi hình ảnh tham chiếu trong DPB mà không được tham chiếu bởi mục nhập bất kỳ trong `RefPicList[ 0 ]` hoặc `RefPicList[ 1 ]` được đánh dấu là "không được sử dụng cho việc tham chiếu".

Phương án khác nữa.

Phần này mô tả phương án khác cho cách tiếp cận được định rõ ở trên mà được đặt tên là "luôn báo hiệu các danh sách hình ảnh tham chiếu trong các phần đầu lát có sự khác biệt giữa các hình ảnh tham chiếu ngắn hạn và dài hạn". Theo phương án khác này, trong phần đầu lát, chu kỳ POC MSB có thể được báo hiệu cho mỗi mục nhập LTRP, tương tự như trong HEVC hoặc như trong các cách tiếp cận nêu trên, và ràng buộc sau đây được loại bỏ: Ở thời điểm bất kỳ trong quy trình giải mã, các trị số của `PicOrderCntVal` & $(\text{MaxLtPicOrderCntLsb} - 1)$ đối với hai hình ảnh tham chiếu bất kỳ trong DPB sẽ không giống nhau.

Fig.6 là sơ đồ giản lược của thiết bị tạo mã video 600 (ví dụ, bộ mã hóa video 20 hoặc

bộ giải mã video 30) theo một phương án của sáng chế. Thiết bị tạo mã video 600 là thích hợp để thực hiện các phương án được bộc lộ như được mô tả ở đây. Thiết bị tạo mã video 600 bao gồm các cổng vào 610 và các bộ phận thu (Rx) 620 để thu dữ liệu; bộ xử lý, bộ phận logic, hoặc bộ xử lý trung tâm (Central Processing Unit, viết tắt là CPU) 630 để xử lý dữ liệu; các bộ phận truyền (Tx) 640 và các cổng đầu ra 650 để truyền dữ liệu; và bộ nhớ 660 để lưu trữ dữ liệu. Thiết bị tạo mã video 600 có thể cũng bao gồm các thành phần quang-sang-điện (Optical-to-Electrical, viết tắt là OE) và các thành phần điện sang quang (Electrical-to-Optical, viết tắt là EO) được ghép nối với các cổng vào 610, các bộ phận thu 620, các bộ phận truyền 640, và các cổng đầu ra 650 để thu vào hoặc đưa ra các tín hiệu quang hoặc điện.

Bộ xử lý 630 được thực hiện bởi phần cứng và phần mềm. Bộ xử lý 630 có thể được thực hiện dưới dạng một hoặc nhiều chip CPU, các lõi (ví dụ, như là bộ xử lý đa lõi), các mảng cổng lập trình được định cấu hình (Field-Programmable Gate Array, viết tắt là FPGA), các mạch tích hợp chuyên dụng (Application Specific Integrated Circuit, viết tắt là ASIC), và các bộ xử lý tín hiệu số (Digital Signal Processor, viết tắt là DSP). Bộ xử lý 630 truyền thông với các cổng vào 610, các bộ phận thu 620, các bộ phận truyền 640, các cổng đầu ra 650, và bộ nhớ 660. Bộ xử lý 630 bao gồm mô-đun tạo mã 670. Mô-đun tạo mã 670 thực hiện các phương án nêu trên. Chẳng hạn, mô-đun tạo mã 670 thực hiện, xử lý, chuẩn bị hoặc cung cấp các chức năng nối mạng khác nhau. Do đó, việc bao gồm mô-đun tạo mã 670 cung cấp sự cải tiến cơ bản đối với chức năng của thiết bị tạo mã video 600 và tạo hiệu quả biến đổi thiết bị tạo mã video 600 sang trạng thái khác nhau. Theo cách khác, mô-đun tạo mã 670 được thực hiện dưới dạng các lệnh được lưu trữ trong bộ nhớ 660 và được thực hiện bởi bộ xử lý 630.

Thiết bị tạo mã video 600 có thể cũng bao gồm các thiết bị đầu vào và/hoặc đầu ra (input and/or output, viết tắt là I/O) 680 để truyền thông dữ liệu đến và từ người dùng. Các thiết bị I/O 680 có thể bao gồm các thiết bị đầu ra chẳng hạn như bộ hiển thị để hiển thị dữ liệu video, các loa để đưa ra dữ liệu audio, v.v.. Các thiết bị I/O 680 có thể cũng bao gồm các thiết bị đầu vào, chẳng hạn như bàn phím, chuột, bi xoay, v.v., và/hoặc các giao diện tương ứng để tương tác với các thiết bị đầu ra như vậy.

Bộ nhớ 660 bao gồm một hoặc nhiều ổ đĩa, ổ băng, và ổ đĩa trạng thái rắn và có thể

được sử dụng dưới dạng thiết bị lưu trữ dữ liệu vượt luồng, để lưu trữ các chương trình khi các chương trình như vậy được lựa chọn để thực hiện, và để lưu trữ các lệnh và dữ liệu được đọc trong suốt quá trình thực chương trình. Bộ nhớ 660 có thể là bộ nhớ khả biến và bất khả biến và có thể là bộ nhớ chỉ đọc (Read-Only Memory, viết tắt là ROM), bộ nhớ truy cập ngẫu nhiên (Random Access Memory, viết tắt là RAM), bộ nhớ khả lập địa chỉ nội dung bậc ba (Ternary Content-Addressable Memory, viết tắt là TCAM), và/hoặc bộ nhớ truy cập ngẫu nhiên tĩnh (Static Random-Access Memory, viết tắt là SRAM).

Fig.7 là sơ đồ giản lược của phương án về phương tiện tạo mã 700. Theo phương án này, phương tiện tạo mã 700 được thực hiện trong thiết bị tạo mã video 702 (ví dụ, bộ mã hóa video 20 hoặc bộ giải mã video 30). Thiết bị tạo mã video 702 bao gồm phương tiện thu 701. Phương tiện thu 701 được tạo cấu hình để thu ảnh để mã hóa hoặc để thu dòng bit để giải mã. Thiết bị tạo mã video 702 bao gồm phương tiện truyền 707 được ghép nối với phương tiện thu 701. Phương tiện truyền 707 được tạo cấu hình để truyền dòng bit đến bộ giải mã hoặc để truyền hình ảnh được giải mã đến phương tiện hiển thị (ví dụ, một trong số các thiết bị I/O 680).

Thiết bị tạo mã video 702 bao gồm phương tiện lưu trữ 703. Phương tiện lưu trữ 703 được ghép nối với ít nhất một trong số phương tiện thu 701 hoặc phương tiện truyền 707. Phương tiện lưu trữ 703 được tạo cấu hình để lưu trữ các lệnh. Thiết bị tạo mã video 702 cũng bao gồm phương tiện xử lý 705. Phương tiện xử lý 705 được ghép nối với phương tiện lưu trữ 703. Phương tiện xử lý 705 được tạo cấu hình để thực hiện các lệnh được lưu trữ trong phương tiện lưu trữ 703 để thực hiện các phương pháp được bộc lộ ở đây.

Trong khi một vài phương án đã được đưa ra trong sáng chế, có thể được hiểu rằng các hệ thống và các phương pháp được bộc lộ có thể được bao gồm theo nhiều dạng cụ thể khác mà không trệch khỏi tinh thần hoặc phạm vi của sáng chế. Các ví dụ của sáng chế được xem là minh họa và không giới hạn, và không giới hạn ở các chi tiết được đưa ra ở đây. Ví dụ, các phần tử hoặc các thành phần khác nhau có thể được kết hợp hoặc được kết hợp trong hệ thống khác hoặc các dấu hiệu nhất định có thể được bỏ qua, hoặc không được thực hiện.

Ngoài ra, các kỹ thuật, các hệ thống, các hệ thống con, và các phương pháp được mô tả và được minh họa trong các phương án khác nhau dưới dạng rời rạc hoặc riêng biệt có thể

được kết hợp hoặc được kết hợp với các hệ thống, các môđun, các kỹ thuật, hoặc các phương pháp khác mà không trệch khỏi phạm vi của sáng chế. Các phần tử khác được thể hiện hoặc được thảo luận như được ghép nối hoặc trực tiếp được ghép nối hoặc truyền thông với nhau có thể được ghép nối gián tiếp hoặc truyền thông qua một vài giao diện, thiết bị, hoặc thành phần trung gian cho dù bằng điện, cơ khí, hoặc theo cách khác. Các ví dụ khác về các thay đổi, các sự thay thế, và các biến đổi là xác định được bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương ứng và có thể được thực hiện mà không trệch khỏi tinh thần và phạm vi được bộc lộ trong sáng chế.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã dòng bit video được tạo mã được thực hiện bởi bộ giải mã video, phương pháp này bao gồm các bước:

thu nhận cấu trúc cú pháp danh sách hình ảnh tham chiếu cho lát hiện thời từ dòng bit video được tạo mã, trong đó cấu trúc cú pháp danh sách hình ảnh tham chiếu chứa số lượng mục nhập, trong đó cấu trúc cú pháp danh sách hình ảnh tham chiếu bao gồm số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu;

thu nhận số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu cho hình ảnh hiện thời;

xây dựng danh sách hình ảnh tham chiếu cho lát hiện thời dựa trên cấu trúc cú pháp danh sách hình ảnh tham chiếu, trong đó danh sách hình ảnh tham chiếu chứa số lượng mục nhập hoạt động và số lượng mục nhập không hoạt động;

thiết đặt số lượng mục nhập hoạt động trong danh sách hình ảnh tham chiếu bằng số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu khi số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu lớn hơn số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu; và

thu nhận, dựa trên ít nhất một mục nhập hoạt động của danh sách hình ảnh tham chiếu được xác định bởi số lượng mục nhập hoạt động, ít nhất một khối được xây dựng lại của lát hiện thời.

2. Phương pháp theo điểm 1, trong đó thứ tự của các mục nhập trong cấu trúc cú pháp danh sách hình ảnh tham chiếu giống như thứ tự của các hình ảnh tham chiếu tương ứng trong danh sách hình ảnh tham chiếu.

3. Phương pháp theo điểm 2, trong đó thứ tự này là từ không (zero) đến trị số được chỉ báo.

4. Phương pháp theo điểm 3, trong đó trị số được chỉ báo là từ không (zero) đến trị số được chỉ báo bởi `sps_max_dec_pic_buffering_minus1`.

5. Phương pháp theo điểm 1, trong đó danh sách hình ảnh tham chiếu được định rõ là `RefPictList[0]`.

6. Phương pháp theo điểm 1, trong đó danh sách hình ảnh tham chiếu được định rõ là RefPictList[1].

7. Phương pháp theo điểm 1, trong đó ít nhất một khối được xây dựng lại được sử dụng để tạo ra ảnh được hiển thị trên bộ hiển thị của thiết bị điện tử.

8. Phương pháp theo điểm 1, trong đó danh sách hình ảnh tham chiếu bao gồm danh sách của các hình ảnh tham chiếu được sử dụng cho dự đoán liên ảnh.

9. Phương pháp theo điểm 8, trong đó dự đoán liên ảnh là cho lát P hoặc cho lát B.

10. Phương pháp theo điểm 1, trong đó phần đầu lát chứa cờ tập hợp thông số chuỗi (sequence parameter set, SPS) danh sách hình ảnh tham chiếu được định rõ là ref_pic_list_sps_flag[i].

11. Phương pháp theo điểm 10, trong đó phần đầu lát chứa cờ ghi đè chủ động chỉ số tham chiếu số được định rõ bởi num_ref_idx_active_override_flag.

12. Phương pháp theo điểm 7, trong đó danh sách hình ảnh tham chiếu được định rõ là RefPictList[0] hoặc RefPictList[1], và thứ tự của các mục nhập trong cấu trúc danh sách hình ảnh tham chiếu giống như thứ tự của các hình ảnh tham chiếu tương ứng trong danh sách hình ảnh tham chiếu.

13. Thiết bị giải mã, thiết bị này bao gồm:

bộ thu được tạo cấu hình để thu dòng bit video được tạo mã;

bộ nhớ được ghép nối với bộ thu, bộ nhớ lưu trữ các lệnh; và

bộ xử lý được ghép nối với bộ nhớ, bộ xử lý được tạo cấu hình để thực hiện các lệnh được lưu trữ trong bộ nhớ để khiến cho bộ xử lý:

thu nhận cấu trúc cú pháp danh sách hình ảnh tham chiếu cho lát hiện thời từ dòng bit video được tạo mã, trong đó cấu trúc cú pháp danh sách hình ảnh tham chiếu chứa số lượng mục nhập, trong đó cấu trúc cú pháp danh sách hình ảnh tham chiếu bao gồm số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu;

thu nhận số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu cho hình ảnh hiện thời;

xây dựng danh sách hình ảnh tham chiếu cho lát hiện thời dựa trên cấu trúc cú pháp danh sách hình ảnh tham chiếu, trong đó danh sách hình ảnh tham chiếu chứa số lượng mục nhập hoạt động và số lượng mục nhập không hoạt động;

thiết đặt số lượng mục nhập hoạt động trong danh sách hình ảnh tham chiếu bằng số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu khi số lượng mặc định của các mục nhập hoạt động trong danh sách hình ảnh tham chiếu lớn hơn số lượng mục nhập trong cấu trúc danh sách hình ảnh tham chiếu; và

thu nhận, dựa trên ít nhất một mục nhập hoạt động của danh sách hình ảnh tham chiếu được xác định bởi số lượng mục nhập hoạt động, ít nhất một khối được xây dựng lại của lát hiện thời.

14. Thiết bị giải mã theo điểm 13, trong đó thiết bị giải mã còn bao gồm bộ hiển thị được tạo cấu hình để hiển thị ảnh dựa trên ít nhất một khối được xây dựng lại.

15. Máy tạo mã, máy này bao gồm:

bộ thu được tạo cấu hình để thu dòng bit để giải mã;

bộ truyền được ghép nối với bộ thu, bộ truyền được tạo cấu hình để truyền ảnh được giải mã đến bộ hiển thị;

bộ nhớ được ghép nối với ít nhất một trong số bộ thu hoặc bộ truyền, bộ nhớ được tạo cấu hình để lưu trữ các lệnh; và

bộ xử lý được ghép nối với bộ nhớ, bộ xử lý được tạo cấu hình để thực hiện các lệnh được lưu trữ trong bộ nhớ để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 12.

16. Hệ thống tạo mã, hệ thống này bao gồm:

bộ mã hóa; và

bộ giải mã truyền thông với bộ mã hóa, trong đó bộ mã hóa hoặc bộ giải mã bao gồm thiết bị giải mã theo điểm bất kỳ trong số các điểm từ 13 đến 14.

17. Hệ thống tạo mã, hệ thống này bao gồm:

bộ mã hóa; và

bộ giải mã truyền thông với bộ mã hóa, trong đó bộ mã hóa hoặc bộ giải mã bao gồm máy tạo mã theo điểm 15.

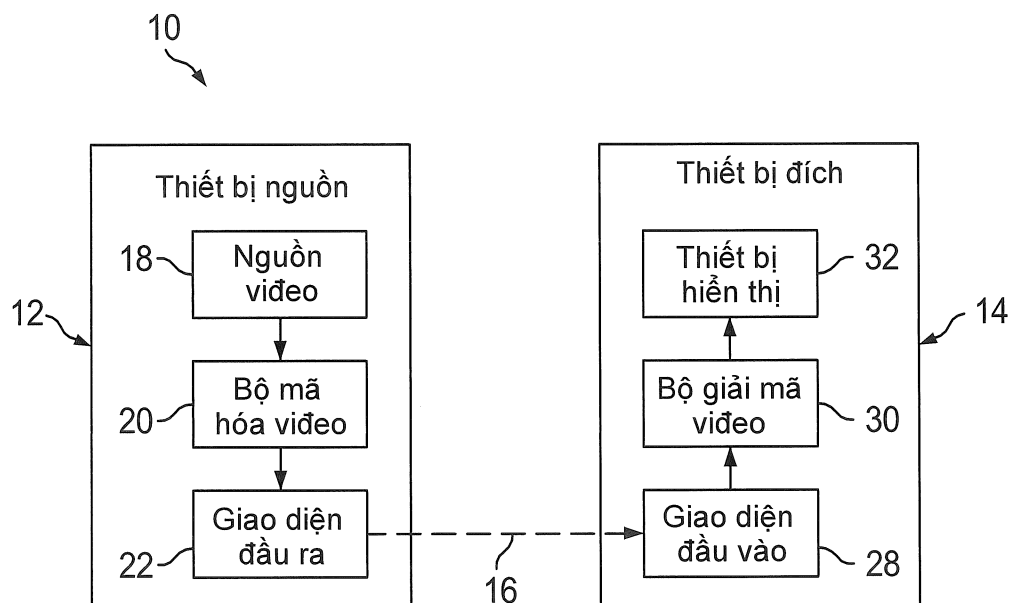


FIG. 1

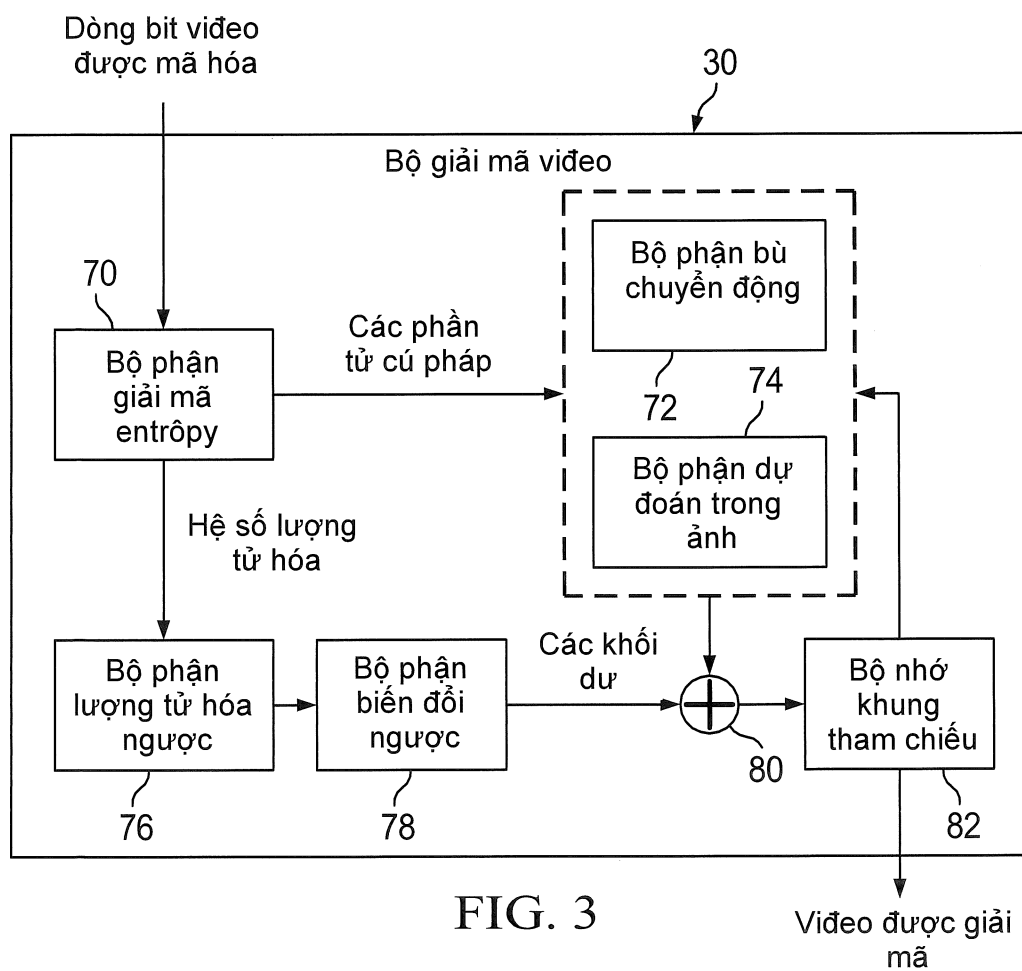
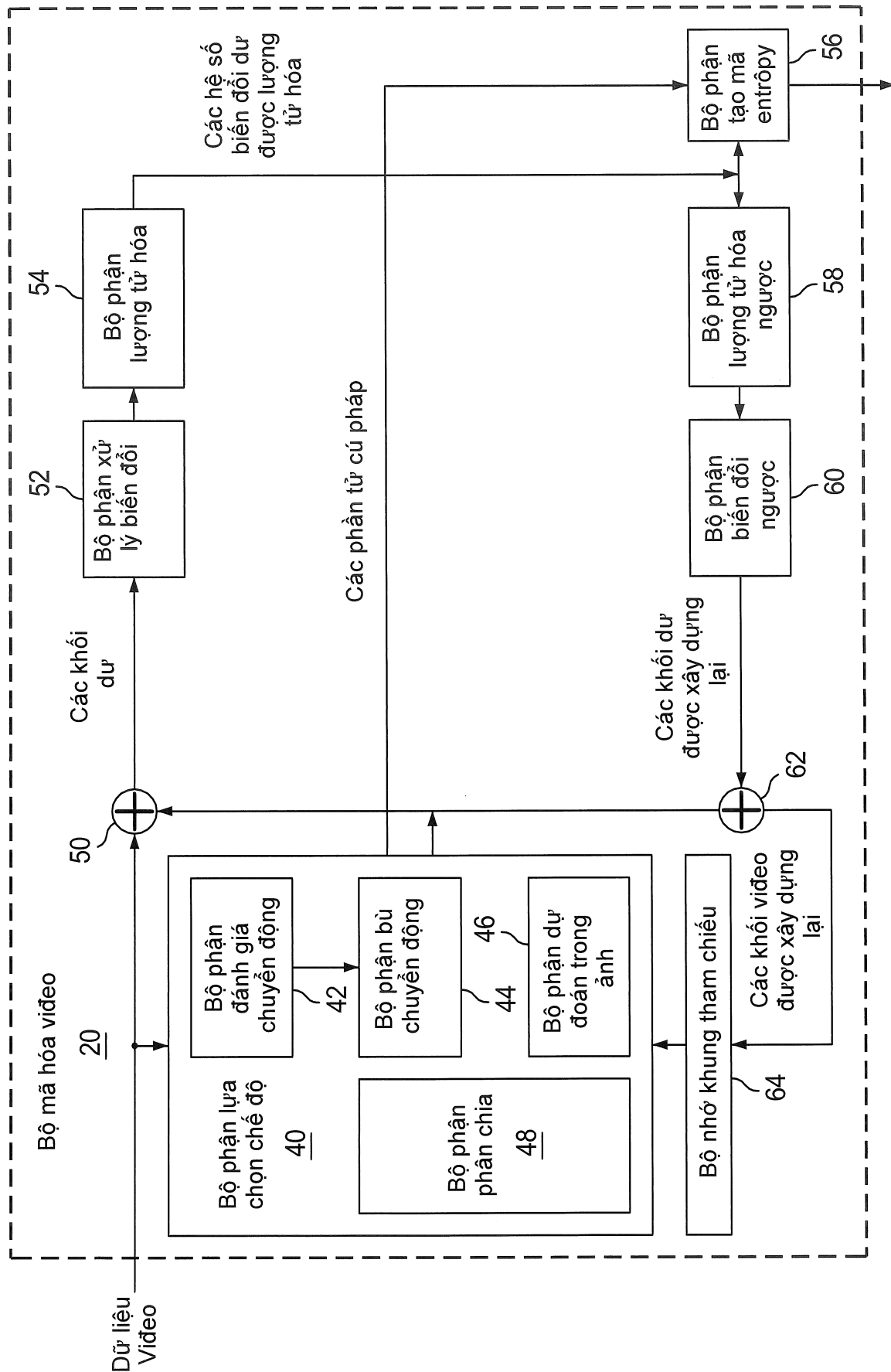


FIG. 3



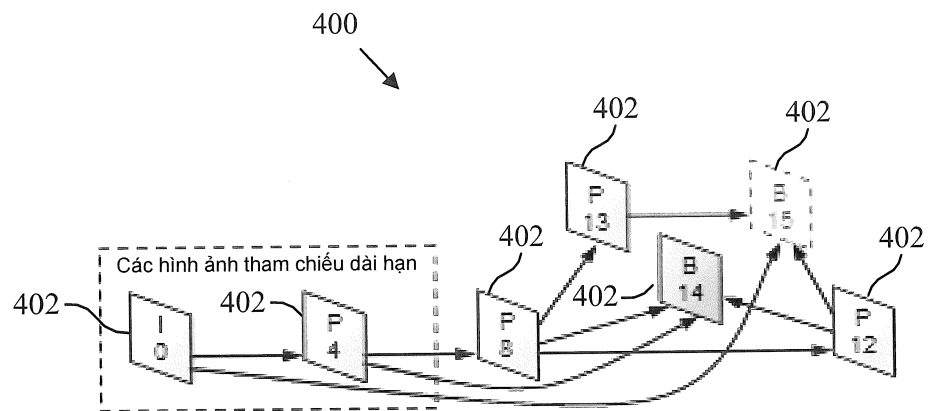


FIG. 4

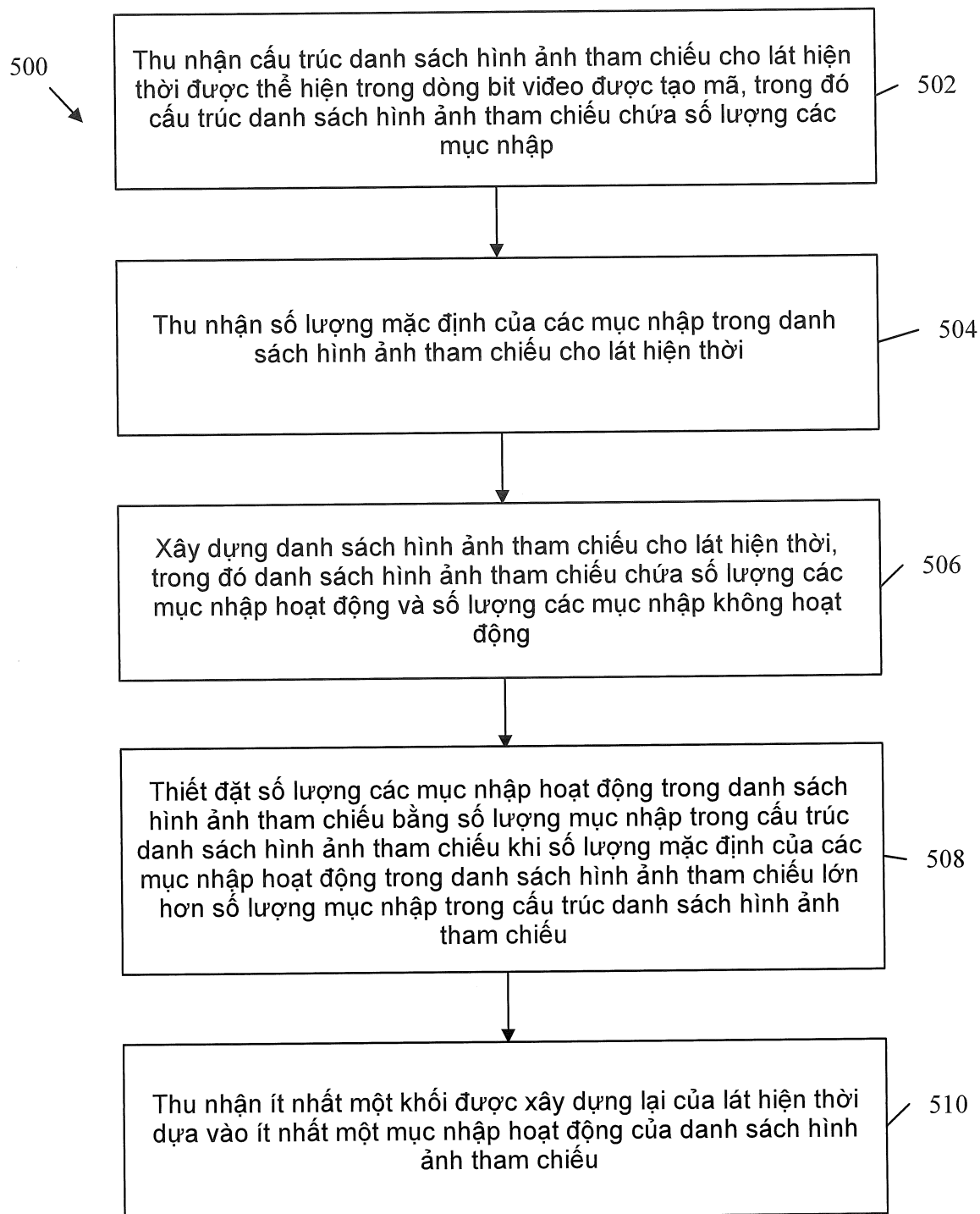


FIG. 5

600

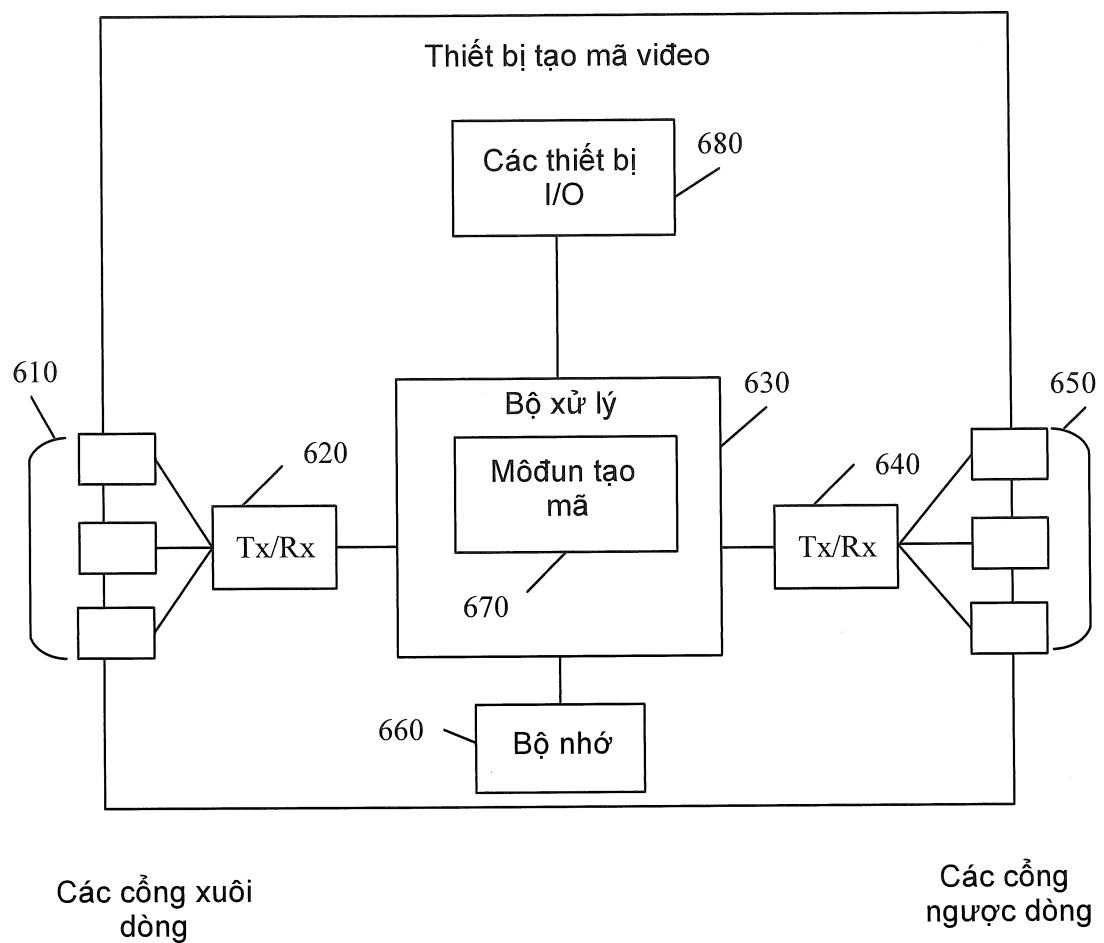


FIG. 6

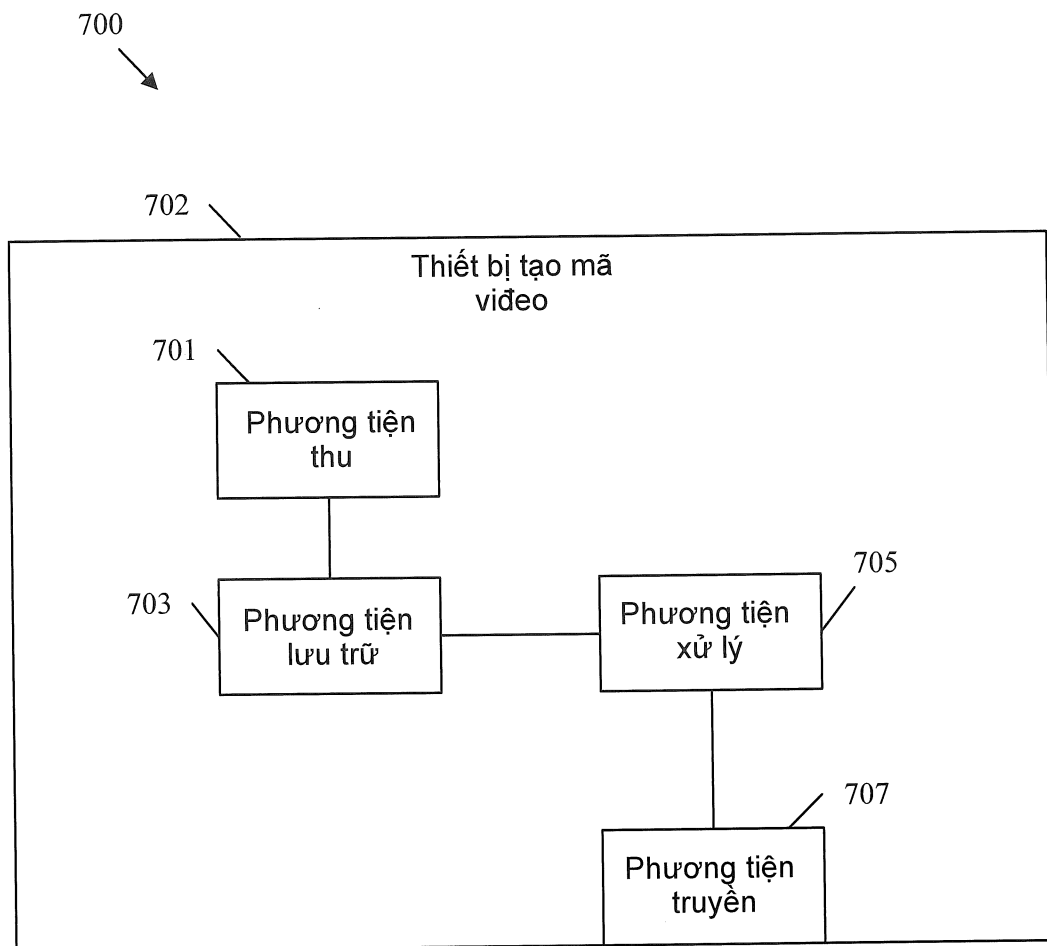


FIG. 7