



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

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

(11)



1-0047708

(51)^{2020.01} H04N 7/12; H04N 19/70

(13) B

(21) 1-2022-02548

(22) 08/09/2020

(86) PCT/US2020/049725 08/09/2020

(87) WO2021/061389 01/04/2021

(30) 62/905,244 24/09/2019 US

(45) 25/06/2025 447

(43) 25/07/2022 412A

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

Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong
518129, P. R. China

(72) WANG, Ye-Kui (US).

(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP MÃ HOÁ VÀ GIẢI MÃ, THIẾT BỊ TẠO MÃ VIDEO, BỘ GIẢI
MÃ, BỘ MÃ HÓA, VÀ PHƯƠNG TIỆN ĐỌC ĐƯỢC BỞI MÁY TÍNH

(21) 1-2022-02548

(57) Sáng chế đề cập đến các phương pháp mã hóa và giải mã, thiết bị tạo mã video, bộ giải mã, bộ mã hóa, và phương tiện đọc được bởi máy tính, và cơ chế tạo mã video. Cơ chế gồm việc mã hóa dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã. Cấu trúc cú pháp các tham số bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD) được mã hóa vào trong dòng bit. Cấu trúc cú pháp các tham số HRD chỉ định tất cả các lớp được kết hợp với cùng số lượng lập lịch phân phối bộ đệm ảnh được tạo mã (coded picture buffer, CPB). Tập các thử nghiệm sự phù hợp của dòng bit được thực hiện trên các lớp được dựa trên CPB lập lịch phân phối. Dòng bit được lưu trữ để truyền thông với bộ giải mã.

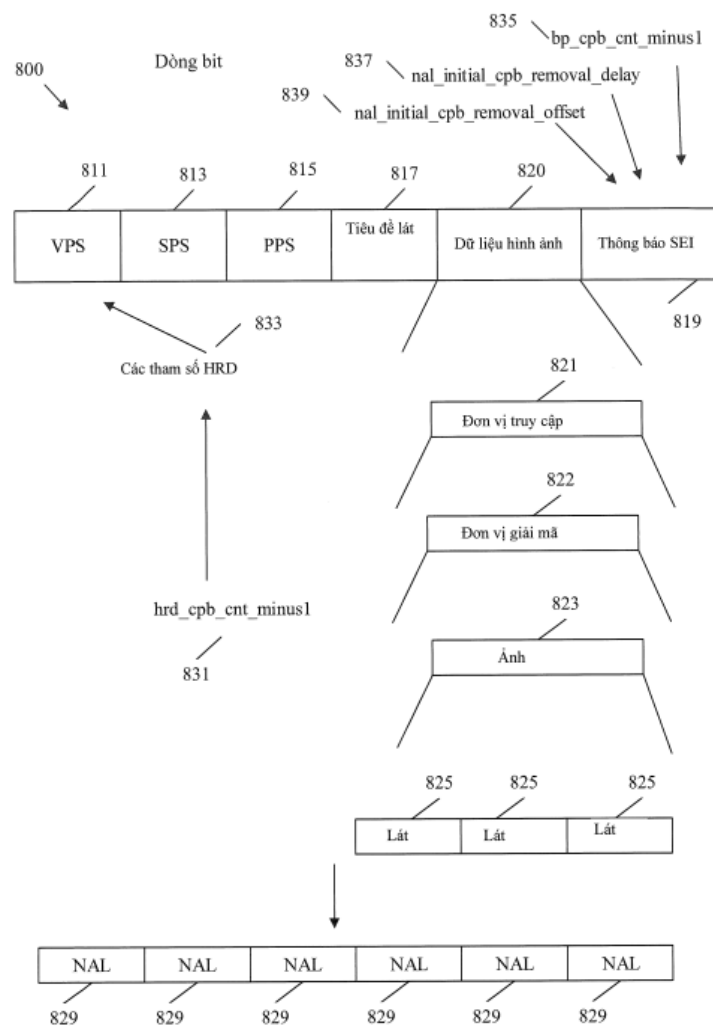


FIG. 8

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến tạo mã video, và cụ thể là đề cập đến các thay đổi tham số bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD) để hỗ trợ mã hóa hiệu quả và/hoặc thử nghiệm sự phù hợp của các dòng bit đa lớp.

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 gây ra khó khăn khi dữ liệu được truyền trực tuyến hoặc được truyền thông qua mạng truyền thông có dung lượng băng thông hạn chế. Do đó, dữ liệu video thường được nén trước khi được truyền thông qua các mạng viễn thông hiện đại. Kích thước của video cũng có thể là một vấn đề khi video được lưu trữ trên thiết bị lưu trữ vì tài nguyên bộ nhớ có thể bị hạn chế. Các thiết bị nén video thường sử dụng phần mềm và/hoặc phần cứng tại nguồn để mã hóa dữ liệu video trước khi truyền hoặc lưu trữ, do đó làm giảm số lượng dữ liệu cần thiết để biểu diễn hình ảnh video kỹ thuật số. Dữ liệu nén sau đó sẽ được thu tại đích bởi thiết bị giải nén video giải mã dữ liệu video. Với nguồn tài nguyên mạng hạn chế và nhu cầu ngày càng cao về chất lượng video, các kỹ thuật nén và giải nén được cải tiến nhằm cải thiện tỷ lệ nén mà ít hoặc không giảm chất lượng hình ảnh là mong muốn.

Bản chất kỹ thuật của sáng chế

Theo phương án, sáng chế gồm phương pháp được thực hiện bởi bộ giải mã, phương pháp bao gồm: thu, bởi bộ thu của bộ giải mã, dòng bit bao gồm cấu trúc cú pháp các tham số bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD) và một hoặc nhiều lớp của các ảnh được tạo mã sao cho tất cả các lớp được kết hợp với cùng số lượng lập lịch phân phối bộ đệm ảnh được tạo mã (coded picture buffer, CPB) như được chỉ định bởi cấu trúc cú pháp các tham số HRD; và giải mã, bởi bộ xử lý của bộ giải mã, ảnh từ một trong các lớp để thu ảnh được giải mã.

Sự cố có thể xảy ra khi các hệ thống tạo mã video thực hiện các kiểm tra sự phù hợp trên các dòng bit. Video có thể được tạo mã vào trong nhiều lớp và/hoặc các lớp phụ, mà có thể sau đó được tổ chức vào trong các OLS. Mỗi lớp và/hoặc lớp phụ của mỗi OLS được kiểm tra về sự phù hợp theo các lập lịch phân phối. Mỗi lập lịch phân phối được kết hợp với kích thước CPB khác và độ trễ CPB để giải thích cho các băng thông truyền và khả năng hệ thống khác nhau. Một số các hệ thống tạo mã video cho phép mỗi lớp phụ xác định số lượng

bất kỳ của lập lịch phân phối. Điều này có thể dẫn đến lượng lớn tín hiệu để hỗ trợ các kiểm tra sự phù hợp, điều này dẫn đến trong giảm hiệu quả tạo mã đối với dòng bit. Ví dụ minh họa gồm cơ chế tăng hiệu quả tạo mã cho video gồm đa lớp. Cụ thể, tất cả các lớp và/hoặc các lớp phụ bị hạn chế gồm cùng số lượng các lập lịch phân phối CPB. Ví dụ, bộ mã hóa có thể xác định số lượng lớn nhất của các lập lịch phân phối CPB được sử dụng để một lớp bất kỳ và tập số lượng của các lập lịch phân phối CPB cho tất cả các lớp thành số tối đa. Số lượng lập lịch phân phối có thể sau đó được báo hiệu một lần, ví dụ như một phần của các tham số HRD trong VPS. Điều này tránh sự cần thiết để báo hiệu số lượng lập lịch cho mỗi lớp/lớp phụ. Trong một số ví dụ, tất cả các lớp/lớp phụ trong OLS có thể cũng chia sẻ cùng chỉ số lập lịch phân phối. Những thay đổi này giảm lượng dữ liệu được sử dụng để báo hiệu dữ liệu liên quan tới việc kiểm tra sự phù hợp. Điều này làm giảm kích thước dòng bit, và từ đó làm giảm việc sử dụng bộ xử lý, bộ nhớ, và/hoặc tài nguyên mạng tại cả hai bộ mã hóa và bộ giải mã.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó việc lập lịch phân phối CPB được kết hợp với tập các thử nghiệm sự phù hợp của dòng bit được thực hiện bởi HRD hoạt động trên bộ mã hóa.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó số lượng các lập lịch phân phối CPB được chỉ định trong HRD trừ một (`hrd_cpb_cnt_minus1`), và trong đó `hrd_cpb_cnt_minus1` trong khoảng từ 0 tới 31.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó các lớp được tổ chức vào trong một hoặc nhiều tập lớp đầu ra (output layer set, OLS), và trong đó lập lịch phân phối tương ứng CPB cho OLS đích được tham chiếu bởi chỉ số lập lịch (`ScIdx`) trong khoảng từ 0 tới giá trị của `hrd_cpb_cnt_minus1`.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó mỗi trong các lập lịch phân phối CPB được mô tả bởi độ trễ xóa CPB ban đầu lớp trừu tượng mạng (network abstraction layer, NAL) (`nal_initial_cpb_removal_delay`) và phần bù xóa CPB ban đầu NAL (`nal_initial_cpb_removal_offset`).

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó mỗi lớp được kết hợp với số đếm CPB khoảng đệm (BP, buffering period) trừ 1 (`bp_cpb_cnt_minus1`) cộng 1 mà chỉ định số lượng cấp `nal_initial_cpb_removal_delay` và `nal_initial_cpb_removal_offset` cho mỗi lớp, và trong đó `bp_cpb_cnt_minus1` bằng giá trị của `hrd_cpb_cnt_minus1`.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó cấu trúc cú pháp các tham số HRD được gồm trong dòng bit trong tập tham số video (video parameter set, VPS) chỉ định các OLS.

Theo phương án, sáng chế gồm phương pháp được thực hiện bởi bộ mã hóa, phương pháp bao gồm: mã hóa, bởi bộ xử lý của bộ mã hóa, dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã; mã hóa vào trong dòng bit, bởi bộ xử lý cấu trúc cú pháp các tham số HRD chỉ định tất cả các lớp được kết hợp với cùng số lượng các lập lịch phân phối CPB; và thực hiện, bởi bộ xử lý, tập các thử nghiệm sự phù hợp của dòng bit trên các lớp được dựa trên lập lịch phân phối CPB.

Sự cố có thể xảy ra khi các hệ thống tạo mã video thực hiện các kiểm tra sự phù hợp trên các dòng bit. Video có thể được tạo mã vào trong nhiều lớp và/hoặc các lớp phụ, mà có thể sau đó được tổ chức vào trong các OLS. Mỗi lớp và/hoặc lớp phụ của mỗi OLS được kiểm tra về sự phù hợp theo lập lịch phân phối. Mỗi lập lịch phân phối được kết hợp với kích thước CPB khác và độ trễ CPB để giải thích cho các băng thông truyền và khả năng hệ thống khác nhau. Một số các hệ thống tạo mã video cho phép mỗi lớp phụ xác định số lượng bất kỳ của lập lịch phân phối. Điều này có thể dẫn đến lượng lớn tín hiệu để hỗ trợ các kiểm tra sự phù hợp, điều này dẫn đến trong hiệu quả tạo mã đối với dòng bit giảm. Các ví dụ bộc lộ gồm cơ chế tăng hiệu quả tạo mã cho video gồm đa lớp. Cụ thể, tất cả các lớp và/hoặc các lớp phụ bị hạn chế để gồm cùng số lượng các lập lịch phân phối CPB. Ví dụ, bộ mã hóa có thể xác định số lượng lớn nhất của các lập lịch phân phối CPB được sử dụng để một lớp bất kỳ và tập số lượng của các lập lịch phân phối CPB cho tất cả các lớp tối đa. Số lượng lập lịch phân phối có thể sau đó được báo hiệu một lần, ví dụ như một phần của các tham số HRD trong VPS. Điều này tránh sự cần thiết để báo hiệu số lượng lập lịch cho mỗi lớp/lớp phụ. Trong một số ví dụ, tất cả các lớp/lớp phụ trong OLS có thể cũng chia sẻ cùng chỉ số lập lịch phân phối. Những thay đổi này giảm lượng dữ liệu được sử dụng để báo hiệu dữ liệu liên quan tới việc kiểm tra sự phù hợp. Điều này làm giảm kích thước dòng bit, và và từ đó làm giảm việc sử dụng bộ xử lý, bộ nhớ, và/hoặc tài nguyên mạng tại cả hai bộ mã hóa và bộ giải mã.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó tập các thử nghiệm sự phù hợp của dòng bit được thực hiện bởi HRD hoạt động trên bộ xử lý.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó số lượng của các lập lịch phân phối CPB được chỉ định trong `hrd_cpb_cnt_minus1`, và trong đó `hrd_cpb_cnt_minus1` trong khoảng từ 0 tới 31, bao gồm.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó các lớp được tổ chức vào trong một hoặc nhiều OLS, và trong đó lập lịch phân phối tương ứng CPB cho OLS đích được tham chiếu bởi ScIdx trong khoảng từ 0 tới giá trị của `hrd_cpb_cnt_minus1`, bao gồm.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó mỗi trong các lập lịch phân phối CPB được mô tả bởi `nal_initial_cpb_removal_delay` và `nal_initial_cpb_removal_offset`.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó mỗi lớp được kết hợp với `bp_cpb_cnt_minus1` cộng 1 mà chỉ định số lượng cặp `nal_initial_cpb_removal_delay` và `nal_initial_cpb_removal_offset` cho mỗi lớp, và trong đó `bp_cpb_cnt_minus1` bằng giá trị của `hrd_cpb_cnt_minus1`.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó cấu trúc cú pháp các tham số HRD được mã hóa trong dòng bit trong VPS chỉ định các OLS.

Theo phương án, sáng chế gồm thiết bị tạo mã video bao gồm: bộ xử lý, bộ thu được ghép với bộ xử lý, bộ nhớ được ghép với bộ xử lý, và bộ phát được ghép với bộ xử lý, trong đó bộ xử lý, bộ thu, bộ nhớ, và bộ phát được tạo cấu hình để thực hiện phương pháp của bất kỳ khía cạnh nào trước đó.

Theo phương án, sáng chế gồm phương tiện đọc được bằng máy tính không tạm thời bao gồm sản phẩm chương trình máy tính để sử dụng bởi thiết bị tạo mã video, sản phẩm chương trình máy tính bao gồm các lệnh thực thi được bằng máy tính được lưu trữ trên phương tiện đọc được bằng máy tính không tạm thời sao cho khi được thực thi bởi bộ xử lý làm cho thiết bị tạo mã video để thực hiện phương pháp của bất kỳ khía cạnh nào trước đó.

Theo phương án, sáng chế gồm bộ giải mã bao gồm: phương tiện thu để thu dòng bit bao gồm cấu trúc cú pháp các tham số HRD và một hoặc nhiều lớp của các ảnh được tạo mã sao cho tất cả các lớp được kết hợp với cùng số lượng các lập lịch phân phối CPB như được chỉ định bởi các tham số HRD; phương tiện giải mã để giải mã ảnh từ một trong các lớp để thu ảnh được giải mã; và phương tiện chuyển tiếp để chuyển tiếp ảnh được giải mã để hiển thị như một phần của chuỗi video được giải mã.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó bộ giải mã còn được tạo cấu hình để thực hiện phương pháp của bất kỳ khía cạnh nào trước đó.

Theo phương án, sáng chế gồm bộ mã hóa bao gồm: phương tiện mã hóa để mã hóa dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã; và mã hóa vào trong dòng

bit cấu trúc cú pháp các tham số HRD chỉ định rằng tất cả các lớp được kết hợp với cùng số lượng các lập lịch phân phối CPB; phương tiện HRD để thực hiện tập các thử nghiệm sự phù hợp của dòng bit trên các lớp được dựa trên lập lịch phân phối CPB; và phương tiện lưu trữ để lưu trữ dòng bit để truyền thông với bộ giải mã.

Tùy chọn, theo khía cạnh bất kỳ trước đó, việc triển khai khía cạnh khác đề xuất, trong đó bộ mã hóa còn được tạo cấu hình để thực hiện phương pháp của bất kỳ khía cạnh nào trước đó.

Với mục đích làm rõ ràng, bất kỳ một trong các phương án nêu trên có thể được kết hợp với bất kỳ một hoặc nhiều phương án nêu trên để tạo ra một phương án mới trong phạm vi của sáng chế.

Những điều này và các tính năng khác sẽ được hiểu rõ ràng hơn từ mô tả chi tiết sau đây được thực hiện cùng với các bản vẽ và yêu cầu bảo hộ kèm theo.

Mô tả vắn tắt các hình vẽ

Để có hiểu biết đầy đủ hơn về sáng chế này, hiện tại, tham khảo phần mô tả ngắn gọn sau đây, liên quan đến các bản vẽ và mô tả chi tiết kèm theo, trong đó tham số giống nhau thể hiện các bộ phận giống nhau.

FIG. 1 là lưu đồ phương pháp ví dụ tạo mã tín hiệu video.

FIG. 2 là sơ đồ khối hệ thống tạo mã và giải mã (codec) ví dụ tạo mã video.

FIG. 3 là sơ đồ khối minh họa bộ mã hóa video ví dụ.

FIG. 4 là sơ đồ khối minh họa bộ giải mã video ví dụ.

FIG. 5 là sơ đồ khối minh họa bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD) ví dụ.

FIG. 6 là sơ đồ khối minh họa ví dụ về chuỗi video nhiều lớp được tạo cấu hình để dự đoán liên lớp.

FIG. 7 là sơ đồ khối minh họa ví dụ về chuỗi video nhiều lớp được tạo cấu hình để có khả năng mở rộng theo thời gian.

FIG. 8 là sơ đồ khối minh họa ví dụ về dòng bit.

FIG. 9 là sơ đồ khối ví dụ về thiết bị tạo mã video.

FIG. 10 là lưu đồ phương pháp ví dụ về việc mã hóa chuỗi video vào trong dòng bit trong đó tất cả các lớp trong dòng bit có cùng số lượng lập lịch phân phối bộ đệm ảnh được tạo mã (coded picture buffer, CPB).

FIG. 11 là lưu đồ phương pháp ví dụ giải mã chuỗi video từ dòng bit trong đó tất cả các lớp trong dòng bit có cùng số lượng các lập lịch phân phối CPB.

FIG. 12 là sơ đồ khối của hệ thống ví dụ để tạo mã chuỗi video sử dụng dòng bit trong đó tất cả các lớp trong dòng bit có cùng số lượng lập lịch phân phối bộ đệm ảnh được tạo mã (coded picture buffer, CPB).

Mô tả chi tiết sáng chế

Cần hiểu rằng mặc dù cách triển khai minh họa một hoặc nhiều phương án được đề xuất dưới đây, nhưng các hệ thống và/hoặc phương pháp được bộc lộ có thể được triển khai bằng cách sử dụng bất kỳ kỹ thuật nào, cho dù hiện đã biết hoặc đang tồn tại. Sáng chế không bị giới hạn ở các cách triển khai, bản vẽ và kỹ thuật minh họa được minh họa dưới đây, bao gồm cả các thiết kế và cách triển khai mẫu được minh họa và mô tả ở đây, nhưng có thể được sửa đổi trong phạm vi của các yêu cầu bảo hộ phụ thuộc cùng với phạm vi tương đương đầy đủ của chúng.

Các thuật ngữ sau đây được định nghĩa như sau trừ khi được sử dụng trong ngữ cảnh trái ngược ở đây. Cụ thể, các định nghĩa sau đây nhằm cung cấp thêm sự rõ ràng cho sáng chế. Tuy nhiên, các thuật ngữ có thể được mô tả khác nhau trong các ngữ cảnh khác nhau. Do đó, các định nghĩa sau đây nên được coi là phân bổ sung và không được coi là để hạn chế bất kỳ định nghĩa mô tả nào khác được cung cấp cho các thuật ngữ đó ở đây.

Dòng bit là chuỗi các bit gồm dữ liệu video mà được nén để truyền giữa bộ mã hóa và bộ giải mã. Bộ mã hóa là thiết bị mà được tạo cấu hình để dùng các quy trình mã hóa để nén dữ liệu video vào trong dòng bit. Bộ giải mã là thiết bị mà được tạo cấu hình để dùng các quy trình giải mã để giải nén dữ liệu video từ dòng bit để hiển thị. Ảnh là mảng các mẫu độ chói và/hoặc mảng các mẫu sắc độ mà tạo khung hoặc trường của chúng. Ảnh mà được mã hóa hoặc được giải mã có thể được coi là ảnh hiện tại để làm rõ cuộc thảo luận. Đơn vị lớp trừu tượng mạng (network abstraction layer, NAL) là cấu trúc cú pháp chứa dữ liệu dưới dạng của khối lượng chuỗi byte thô (Raw Byte Sequence Payload, RBSP), sự chỉ báo loại dữ liệu, và các byte ngăn mô phỏng, được xen kẽ theo ý muốn. Đơn vị NAL lớp tạo mã video (VCL, video coding layer) là đơn vị NAL được tạo mã để chứa dữ liệu video, như là lát được tạo mã của ảnh. Đơn vị NAL không VCL là đơn vị NAL mà chứa dữ liệu không phải video như là cú pháp và/hoặc các tham số mà hỗ trợ giải mã dữ liệu video, thực hiện kiểm tra sự phù hợp, hoặc các hoạt động khác. Đơn vị truy cập (access unit, AU) là tập đơn vị NAL mà được kết hợp với nhau theo nguyên tắc phân loại cụ thể và thuộc về một thời gian đầu ra cụ thể. Đơn vị giải mã (decoding unit, DU) là AU hoặc tập con của AU và các đơn vị NAL không VCL được liên kết. Ví dụ, AU gồm các đơn vị VCL NAL và các đơn vị NAL không VCL bất kỳ được kết hợp với các đơn vị VCL NAL trong AU. Hơn nữa, DU gồm tập các đơn vị VCL NAL từ AU hoặc tập phụ của chúng, cũng như các đơn vị NAL không VCL bất

kỳ được kết hợp với các đơn vị VCL NAL trong DU. Lớp là tập các đơn vị VCL NAL mà có chung đặc tính cụ thể (ví dụ, độ phân giải, tốc độ khung, kích thước hình ảnh, v.v.) và các đơn vị NAL không VCL được liên kết. Thứ tự giải mã là thứ tự mà trong đó các phần tử cú pháp được xử lý bởi quy trình giải mã. Tập tham số video (video parameter set, VPS) là đơn vị dữ liệu mà chứa các tham số liên quan tới toàn bộ video.

Dòng bit có thể mở rộng theo thời gian là dòng bit được tạo mã trong đa lớp cung cấp độ phân giải/tốc độ khung theo thời gian khác nhau (ví dụ, mỗi lớp được tạo mã để hỗ trợ tốc độ khung khác nhau). Lớp phụ là lớp có thể mở rộng theo thời gian của dòng bit có thể mở rộng theo thời gian gồm các đơn vị VCL NAL với giá trị bộ nhận dạng theo thời gian cụ thể và các đơn vị NAL không VCL được liên kết. Ví dụ, lớp phụ theo thời gian là lớp mà chứa dữ liệu video được kết hợp với tốc độ khung cụ thể. Sự biểu diễn lớp phụ là tập phụ của dòng bit chứa các đơn vị NAL của lớp phụ cụ thể và các lớp phụ thấp hơn. Do đó, một hoặc nhiều lớp phụ theo thời gian có thể được kết hợp để đạt được sự biểu diễn lớp phụ mà có thể được giải mã dẫn đến chuỗi video với tốc độ khung cụ thể. Tập lớp đầu ra (OLS, output layer set) là tập các lớp trong đó một hoặc nhiều lớp được chỉ định là (các) lớp đầu ra. Lớp đầu ra là lớp mà được chỉ định cho đầu ra (ví dụ, tới bộ hiển thị). Chỉ số OLS là chỉ số mà nhận dạng duy nhất OLS tương ứng. OLS thứ 0 (0-th) là OLS mà chỉ chứa lớp thấp nhất (lớp với bộ nhận dạng lớp thấp nhất) và do đó chỉ chứa lớp đầu ra. Bộ nhận dạng theo thời gian (ID) là phần tử dữ liệu mà chỉ báo dữ liệu tương ứng với vị trí theo thời gian trong chuỗi video. Quy trình trích xuất dòng bit phụ là quy trình loại bỏ các đơn vị NAL từ dòng bit mà không thuộc về tập đích như được xác định bởi chỉ số OLS đích và ID theo thời gian cao nhất đích. Quy trình trích xuất dòng bit phụ dẫn đến dòng bit phụ đầu ra chứa các đơn vị NAL từ dòng bit mà là một phần của tập đích.

HRD là mô hình giải mã hoạt động trên bộ mã hóa mà kiểm tra tính thay đổi của các dòng bit được tạo ra bởi quy trình mã hóa để xác minh sự phù hợp với các ràng buộc cụ thể. Kiểm tra sự phù hợp của dòng bit là thử nghiệm để xác định xem liệu dòng bit được mã hóa tuân theo tiêu chuẩn hay không, như là tạo mã video đa năng (VVC, Versatile Video Coding). Các tham số HRD là các phần tử cú pháp mà khởi tạo và/hoặc xác định các điều kiện hoạt động của HRD. Tham số HRD có thể được chứa trong cấu trúc cú pháp tham số HRD. Cấu trúc cú pháp là đối tượng dữ liệu được tạo cấu hình để bao gồm nhiều tham số khác nhau. Phần tử cú pháp là đối tượng dữ liệu có chứa một hoặc nhiều tham số cùng kiểu. Do đó, cấu trúc cú pháp có thể chứa nhiều phần tử cú pháp. Các tham số HRD mức độ chuỗi là các tham số HRD mà áp dụng toàn bộ chuỗi video được tạo mã. ID theo thời gian HRD lớn nhất ($hrd_max_tid[i]$) xác định ID theo thời gian của sự biểu diễn lớp phụ cao nhất mà các tham

số HRD được chứa trong tập các tham số OLS HRD thứ i. Cấu trúc cú pháp các tham số HRD chung (general_hrd_parameters) là cấu trúc cú pháp mà chứa các tham số HRD mức độ chuỗi. Điểm hoạt động (operation point, OP) là tập phụ theo thời gian của OLS mà được nhận dạng bởi chỉ số OLS và ID theo thời gian cao nhất. OP đang thử nghiệm (targetOp) là OP mà được lựa chọn để thử nghiệm sự phù hợp tại HRD. OLS đích là OLS mà được lựa chọn để trích xuất từ dòng bit. Cờ hiện diện các tham số HRD đơn vị giải mã (decoding_unit_hrd_params_present_flag) là cờ mà chỉ báo xem liệu có hay không các tham số HRD tương ứng hoạt động tại mức độ DU hoặc mức độ AU. Bộ đệm ảnh được tạo mã (coded picture buffer, CPB) là bộ đệm vào trước ra trước trong HRD mà chứa các ảnh được tạo mã theo thứ tự giải mã để sử dụng trong khi xác thực sự phù hợp dòng bit. Bộ đệm ảnh được giải mã (DPB, decoded picture buffer) là bộ đệm để giữ các ảnh được giải mã để tham chiếu, sắp xếp lại đầu ra, và/hoặc độ trễ đầu ra.

Thông báo thông tin nâng cao bổ sung (SEI, supplemental enhancement information) là cấu trúc cú pháp với các ngữ nghĩa cụ thể để truyền tải thông tin mà không cần bởi quy trình giải mã để xác định các giá trị của các mẫu trong các ảnh được giải mã. Thông báo SEI có thể lồng nhau có thể mở rộng là thông báo mà chứa nhiều thông báo SEI mà tương ứng với một hoặc nhiều OLS hoặc một hoặc nhiều lớp. Thông báo SEI lồng nhau không thể mở rộng là thông báo mà không được lồng nhau và do chứa một thông báo SEI. Thông báo SEI khoảng đệm (BP, buffering period) là thông báo SEI mà chứa các tham số HRD để khởi tạo HRD để quản lý CPB. Thông báo SEI định thời ảnh (picture timing, PT) là thông báo SEI mà chứa các tham số HRD để quản lý thông tin phân phối cho các AU tại CPB và/hoặc DPB. Thông báo SEI thông tin đơn vị giải mã (decoding unit information, DUI) là thông báo SEI mà chứa các tham số HRD để quản lý thông tin phân phối cho các DU tại CPB và/hoặc DPB.

Độ trễ xóa CPB là khoảng thời gian mà AU hiện tại tương ứng có thể vẫn còn trong CPB trước khi xóa và đưa ra DPB. Độ trễ xóa CPB ban đầu là độ trễ xóa CPB mặc định đối với mỗi ảnh, AU, và/hoặc DU trong dòng bit, OLS, và/hoặc lớp. Phần bù xóa CPB là vị trí trong CPB được sử dụng để xác định các ranh giới của AU tương ứng trong CPB. Phần bù xóa CPB ban đầu là phần bù xóa CPB mặc định được kết hợp với mỗi ảnh, AU, và/hoặc DU trong dòng bit, OLS, và/hoặc lớp. Thông tin độ trễ đầu ra bộ đệm ảnh được giải mã (DPB, decoded picture buffer) là khoảng thời gian mà AU tương ứng có thể vẫn còn trong DPB trước khi xuất ra. Thông tin độ trễ xóa CPB là thông tin liên quan tới việc xóa DU tương ứng từ CPB. Lập lịch phân phối xác định thời gian để phân phối dữ liệu video tới và/hoặc từ vị trí bộ nhớ, như là CPB và/hoặc DPB. ID lớp VPS (vps_layer_id) là phần tử cú pháp mà chỉ báo ID lớp của lớp thứ i được chỉ báo trong VPS. Số lượng tập lớp đầu ra trừ một

(num_output_layer_sets_minus1) là phần tử cú pháp mà xác định số tổng của các OLS được chỉ định bởi VPS. Số bộ đệm ảnh được tạo mã HRD (hrd_cpb_cnt_minus1) là phần tử cú pháp mà chỉ định số lượng của các lập lịch phân phối CPB thay thế. Cờ hiện diện các tham số CPB lớp phụ (sublayer_cpb_params_present_flag) là phần tử cú pháp mà chỉ định xem liệu có hay không tập các tham số OLS HRD gồm các tham số HRD đối với các sự biểu diễn lớp phụ được chỉ định. Chỉ số lập lịch (ScIdx) là chỉ số mà nhận dạng lập lịch phân phối. Số BP CPB trừ 1 (bp_cpb_cnt_minus1) là phần tử cú pháp mà chỉ định số lượng độ trễ xóa CPB ban đầu và cập phần bù, và do đó số lượng của lập lịch phân phối mà có sẵn cho lớp phụ theo thời gian. Bộ nhận dạng lớp tiêu đề đơn vị NAL (nuh_layer_id) là phần tử cú pháp mà chỉ định bộ nhận dạng của lớp mà gồm đơn vị NAL. Phần tử cú pháp cờ chung tốc độ ảnh cố định (fixed_pic_rate_general_flag) là phần tử cú pháp mà chỉ định xem liệu có hay không khoảng cách theo thời gian giữa các lần đưa ra HRD của các ảnh liên tiếp theo thứ tự đầu ra bị hạn chế. Cấu trúc cú pháp các tham số HRD lớp phụ (sublayer_hrd_parameters) là cấu trúc cú pháp mà gồm các tham số HRD cho lớp phụ tương ứng. Cờ hiện diện các tham số HRD chung VCL (general_vcl_hrd_params_present_flag) là cờ mà chỉ định xem liệu có hay không các tham số HRD VCL được hiện diện trong cấu trúc cú pháp các tham số HRD chung. Phần tử cú pháp các lớp phụ lớn nhất BP trừ một (bp_max_sublayers_minus1) là phần tử cú pháp mà chỉ định số lượng lớn nhất của các lớp phụ theo thời gian cho độ trễ xóa CPB và phần bù xóa CPB được chỉ báo trong thông báo BP SEI. Phần tử cú pháp các lớp phụ lớn nhất VPS trừ một (vps_max_sublayers_minus1) là phần tử cú pháp mà chỉ định số lượng lớn nhất của các lớp phụ theo thời gian mà có thể được hiện diện trong lớp được chỉ định bởi VPS. Cờ OLS lồng nhau có thể mở rộng là cờ mà chỉ định xem liệu có hay không thông báo SEI được lồng nhau có thể mở rộng áp dụng cho các OLS cụ thể hoặc các lớp cụ thể. Số lượng các OLS trừ một lồng nhau có thể mở rộng (num_olss_minus1) là phần tử cú pháp mà chỉ định số lượng của các OLS mà trong đó thông báo SEI được lồng nhau có thể mở rộng áp dụng. Chỉ số OLS lồng nhau (NestingOlsIdx) là phần tử cú pháp mà chỉ định chỉ số OLS của OLS mà trong đó thông báo SEI được lồng nhau có thể mở rộng áp dụng. Chỉ số OLS đích (targetOlsIdx) mà biến mà nhận dạng chỉ số OLS của OLS đích để được giải mã. Tổng số các OLS trừ một (TotalNumOlss-1) là phần tử cú pháp mà chỉ định tổng số các OLS được chỉ định trong VPS.

Các từ viết tắt sau được sử dụng ở đây, Đơn vị truy cập (access unit, AU), khối cây tạo mã (Coding Tree Block, CTB), đơn vị cây tạo mã (Coding Tree Unit, CTU), đơn vị tạo mã (Coding Unit, CU), chuỗi video lớp tạo mã (Coded Layer Video Sequence, CLVS), Bắt đầu chuỗi video lớp được tạo mã (CLVSS, Coded Layer Video Sequence Start), chuỗi video

được tạo mã (CVS, Coded Video Sequence), bắt đầu chuỗi video được tạo mã (CVSS, Coded Video Sequence Start), Nhóm chuyên gia video chung (JVET, Joint Video Experts Team), Bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD), Tập ô bị hạn chế chuyển động (MCTS, Motion Constrained Tile Set), Đơn vị trao đổi tối đa (MTU, Maximum Transfer Unit), Lớp trừu tượng mạng (network abstraction layer, NAL), Tập lớp đầu ra (OLS, Output Layer Set), Số thứ tự ảnh (POC, Picture Order Count), Điểm truy cập ngẫu nhiên (RAP, Random Access Point), Khối lượng chuỗi byte thô (Raw Byte Sequence Payload, RBSP), Tập tham số chuỗi (SPS, Sequence Parameter Set), Tập tham số video (video parameter set, VPS), Tạo mã video đa năng (VVC).

Nhiều kỹ thuật nén video có thể được sử dụng để giảm kích thước tệp video với mức mất dữ liệu tối thiểu. Ví dụ, các kỹ thuật nén video gồm thực hiện dự đoán theo không gian (ví dụ, nội ảnh) và/hoặc dự đoán theo thời gian (ví dụ, liên ảnh) để giảm hoặc xóa sự thừa dữ liệu trong chuỗi video. Để tạo mã video dựa trên khối, lát video (ví dụ, ảnh video hoặc một phần của ảnh video) có thể được phân vùng thành các khối video, mà cũng có thể được tham chiếu tới như các khối cây, các khối cây tạo mã (CTBs, coding tree blocks), các đơn vị cây tạo mã (CTUs, coding tree units), các đơn vị tạo mã (Cus, coding units), và/hoặc các nút tạo mã. Các khối video trang lát tạo mã nội bộ (I) của ảnh được tạo mã sử dụng dự đoán theo không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng ảnh. Các khối video trong lát dự đoán một chiều tạo mã liên (P) hoặc dự đoán hai chiều (B) của ảnh có thể được tạo mã bằng cách dùng dự đoán theo không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng ảnh hoặc dự đoán theo thời gian đối với các mẫu tham chiếu trong các ảnh tham chiếu khác. Các ảnh có thể được gọi là các khung và/hoặc các hình ảnh, và các ảnh tham chiếu có thể được gọi là các khung tham chiếu và/hoặc các hình ảnh tham chiếu. Dự đoán theo không gian hoặc thời gian dẫn đến khối dự đoán đại diện khối hình ảnh. Dữ liệu dư biểu diễn sự khác biệt về điểm ảnh giữa khối hình ảnh gốc và khối dự đoán. Theo đó, khối tạo mã liên được mã hóa theo vectơ chuyển động mà trỏ tới khối các mẫu tham chiếu tạo khối dự đoán và dữ liệu dư chỉ báo sự khác biệt giữa khối được tạo mã và khối dự đoán. Khối tạo mã nội bộ được mã hóa theo chế độ tạo mã nội bộ và dữ liệu dư. Để nén thêm, dữ liệu dư có thể được biến đổi từ miền điểm ảnh sang miền biến đổi. Kết quả này trong hệ số biến đổi dư, có thể được lượng tử hóa. Hệ số biến đổi lượng tử hóa có thể được sắp xếp ban đầu trong mảng hai chiều. Hệ số biến đổi lượng tử hóa có thể được quét để tạo ra vectơ một chiều của hệ số biến đổi. Tạo mã entropy có thể được áp dụng để đạt được khả năng nén nhiều hơn. Các kỹ thuật nén video này được bộc lộ chi tiết hơn bên dưới.

Để đảm bảo video được mã hóa có thể được giải mã chính xác, video được mã hóa và giải mã theo các tiêu chuẩn tạo mã video tương ứng. Các tiêu chuẩn mã hóa video bao gồm Khu vực tiêu chuẩn hóa của Liên minh Viễn thông Quốc tế (ITU) (ITU-T) H.261, Tổ chức Quốc tế về Tiêu chuẩn hóa / Ủy ban Kỹ thuật Điện Quốc tế (ISO / IEC) Nhóm chuyên gia Hình ảnh chuyển động (MPEG) -1 Phần 2, ITU-T H.262 hoặc ISO / IEC MPEG-2 Phần 2, ITU-T H.263, ISO / IEC MPEG-4 Phần 2, Tạo mã video nâng cao (AVC), còn được gọi là ITU-T H.264 hoặc ISO / IEC MPEG-4 Phần 10, và Tạo mã video hiệu quả cao (HEVC), còn được gọi là ITU-T H.265 hoặc MPEG-H Phần 2. AVC bao gồm các phần mở rộng như Tạo mã video có thể mở rộng (SVC), Tạo mã video nhiều lần (MVC) và Tạo mã video đa góc nhìn cộng với Độ sâu (MVC + D) và AVC ba chiều (3D) (3D-AVC). HEVC bao gồm các phần mở rộng như HEVC có thể mở rộng (SHVC), Multiview HEVC (MV-HEVC) và 3D HEVC (3D-HEVC). Nhóm chuyên gia video chung (JVET) của ITU-T và ISO / IEC đã bắt đầu phát triển tiêu chuẩn mã hóa video được gọi là Mã hóa video đa năng (VVC). VVC được bao gồm trong Bản thảo làm việc (WD), bao gồm JVET-O2001-v14.

Các hệ thống tạo mã video sử dụng các thử nghiệm sự phù hợp khác nhau để đảm bảo dòng bit có thể được giải mã bởi bộ giải mã. Ví dụ, kiểm tra sự phù hợp có thể gồm thử nghiệm toàn bộ dòng bit về sự phù hợp, sau đó thử nghiệm mỗi lớp của dòng bit về sự phù hợp, và cuối cùng kiểm tra các đầu ra có thể giải mã tiềm năng về sự phù hợp. Để thực hiện các kiểm tra sự phù hợp, các tham số tương ứng được gồm trong dòng bit. Bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD) có thể đọc các tham số và thực hiện các thử nghiệm. Video có thể gồm nhiều lớp và nhiều tập lớp đầu ra (output layer set, OLS) khác nhau. Theo yêu cầu, bộ mã hóa truyền một hoặc nhiều lớp của OLS được lựa chọn. Ví dụ, bộ mã hóa có thể truyền (các) lớp tốt nhất từ OLS mà có thể được hỗ trợ bởi băng thông mạng hiện tại. Nhược điểm thứ nhất theo cách tiếp cận này là số lượng đáng kể các lớp được thử nghiệm, nhưng không được truyền tới bộ giải mã. Tuy nhiên, các tham số để hỗ trợ thử nghiệm này có thể vẫn được gồm trong dòng bit, mà không cần thiết tăng kích thước dòng bit.

Trong ví dụ thứ nhất, được bộc lộ ở đây là cơ chế để áp dụng các kiểm tra sự phù hợp của dòng bit chỉ cho mỗi OLS. Trong cách này, toàn bộ dòng bit, mỗi lớp, và các đầu ra có thể giải mã được thử nghiệm chung khi OLS tương ứng được thử nghiệm. Do đó, số lượng của các thử nghiệm sự phù hợp bị giảm, mà làm giảm sự sử dụng bộ xử lý và tài nguyên nhớ tại bộ mã hóa. Hơn nữa, giảm số lượng của các thử nghiệm sự phù hợp có thể giảm số lượng của các tham số được liên kết được gồm trong dòng bit. Điều này làm giảm kích thước dòng

bit, và từ đó làm giảm việc sử dụng bộ xử lý, bộ nhớ, và/hoặc tài nguyên mạng tại cả hai bộ mã hóa và bộ giải mã.

Nhược điểm thứ hai là quy trình báo hiệu tham số HRD được sử dụng cho thử nghiệm sự phù hợp HRD trong một số hệ thống tạo mã video có thể trở nên phức tạp trong ngữ cảnh nhiều lớp. Ví dụ, tập các tham số HRD có thể được báo hiệu đối với mỗi lớp trong mỗi OLS. Các tham số HRD này có thể được báo hiệu trong các vị trí khác trong dòng bit phụ thuộc vào phạm vi dự kiến của các tham số. Điều này dẫn đến sơ đồ mà trở nên phức tạp hơn khi nhiều lớp hơn và/hoặc các OLS được bổ sung. Hơn nữa, các tham số HRD cho các lớp khác và/hoặc các OLS có thể chứa thông tin dư.

Trong ví dụ thứ hai, được bộc lộ ở đây là cơ chế để báo hiệu tập các tham số HRD toàn cầu cho các OLS và các lớp tương ứng. Ví dụ, tất cả các tham số HRD mức độ chuỗi mà áp dụng cho tất cả các OLS và tất cả các lớp được chứa trong các OLS được báo hiệu trong tập tham số video (video parameter set, VPS). VPS được báo hiệu một lần trong dòng bit, và do đó các tham số HRD mức độ chuỗi được báo hiệu một lần. Hơn nữa, các tham số HRD mức độ chuỗi có thể bị hạn chế là giống nhau cho tất cả OLS. Trong cách này, tín hiệu dư bị giảm, mà tăng hiệu quả tạo mã. Ngoài ra, cách tiếp cận này làm đơn giản hóa quy trình HRD. Do vậy, việc sử dụng tài nguyên bộ xử lý, bộ nhớ, và/hoặc tín hiệu mạng bị giảm tại cả hai bộ mã hóa và bộ giải mã.

Nhược điểm thứ ba có thể xảy ra khi các hệ thống tạo mã video thực hiện các kiểm tra sự phù hợp trên các dòng bit. Video có thể được tạo mã vào trong nhiều lớp và/hoặc các lớp phụ, mà sau đó có thể được tổ chức thành các OLS. Mỗi lớp và/hoặc lớp phụ của mỗi OLS được kiểm tra về sự phù hợp theo lập lịch phân phối. Mỗi lập lịch phân phối được kết hợp với kích thước bộ đệm ảnh được tạo mã (coded picture buffer, CPB) khác và độ trễ CPB để giải thích cho các băng thông truyền và khả năng hệ thống khác nhau. Một số các hệ thống tạo mã video cho phép mỗi lớp phụ xác định số lượng bất kỳ của lập lịch phân phối. Điều này có thể dẫn đến lượng lớn tín hiệu để hỗ trợ các kiểm tra sự phù hợp, điều này dẫn đến tăng hiệu quả tạo mã giảm đối với dòng bit.

Trong ví dụ thứ ba, được bộc lộ ở đây là cơ chế để tăng hiệu quả tạo mã cho video gồm đa lớp. Cụ thể, tất cả các lớp và/hoặc lớp phụ bị hạn chế gồm cùng số lượng các lập lịch phân phối CPB. Ví dụ, bộ mã hóa có thể xác định số lượng lớn nhất của các lập lịch phân phối CPB được sử dụng cho một lớp bất kỳ và tập số lượng của các lập lịch phân phối CPB cho tất cả các lớp thành số tối đa. Số lượng của lập lịch phân phối có thể sau đó được báo hiệu một lần, ví dụ như một phần của các tham số HRD trong VPS. Điều này tránh sự cần thiết để báo hiệu số lượng lập lịch cho mỗi lớp/lớp phụ. Trong một số ví dụ, tất cả các lớp/lớp

phụ trong OLS có thể cũng chia sẻ cùng chỉ số lập lịch phân phối. Những thay đổi này giảm lượng dữ liệu được sử dụng để báo hiệu dữ liệu liên quan tới việc kiểm tra sự phù hợp. Điều này làm giảm kích thước dòng bit, và từ đó làm giảm việc sử dụng bộ xử lý, bộ nhớ, và/hoặc tài nguyên mạng tại cả hai bộ mã hóa và bộ giải mã.

Nhược điểm thứ tư có thể xảy ra khi video được tạo mã trong nhiều lớp và/hoặc các lớp phụ, sau đó được tổ chức thành các OLS. Các OLS có thể gồm OLS thứ 0 (0-th) mà chỉ gồm lớp đầu ra. Thông báo thông tin nâng cao bổ sung (SEI) có thể được gồm trong dòng bit để thông báo cho HRD của lớp/các tham số cụ thể OLS được sử dụng để thử nghiệm các lớp của dòng bit về sự phù hợp với các tiêu chuẩn. Cụ thể, các thông báo SEI lồng nhau có thể mở rộng được dùng khi các OLS được gồm trong dòng bit. Thông báo SEI lồng nhau có thể mở rộng chứa các nhóm của các thông báo SEI được lồng nhau mà áp dụng một hoặc nhiều OLS và/hoặc một hoặc nhiều lớp của OLS. Mỗi thông báo SEI được lồng nhau có thể chứa bộ chỉ báo để chỉ báo sự kết hợp với OLS và/hoặc lớp tương ứng. Thông báo SEI được lồng nhau được tạo cấu hình để sử dụng đa lớp và có thể chứa thông tin không liên quan khi được áp dụng cho OLS thứ 0 chứa một lớp.

Trong ví dụ thứ tư, được bộc lộ ở đây là cơ chế để tăng hiệu quả tạo mã đối với video gồm OLS thứ 0. Thông báo SEI lồng nhau không thể mở rộng được dùng cho OLS thứ 0. Thông báo SEI lồng nhau không thể mở rộng bị hạn chế chỉ để áp dụng cho OLS thứ 0 và do đó chỉ cho lớp đầu ra được chứa trong OLS thứ 0. Trong cách này, thông tin không liên quan, như là các mối quan hệ lồng nhau, các chỉ báo lớp, v.v., có thể được bỏ qua từ thông báo SEI. Thông báo SEI lồng nhau không thể mở rộng có thể được sử dụng như thông báo SEI khoảng đệm (BP, buffering period), thông báo SEI định thời ảnh (picture timing, PT), thông báo SEI đơn vị giải mã (decoding unit, DU), hoặc các sự kết hợp của chúng. Những thay đổi này giảm lượng dữ liệu được sử dụng để báo hiệu thông tin liên quan tới việc kiểm tra sự phù hợp cho OLS thứ 0. Điều này làm giảm kích thước dòng bit, và từ đó làm giảm việc sử dụng bộ xử lý, bộ nhớ, và/hoặc tài nguyên mạng tại cả hai bộ mã hóa và bộ giải mã.

Nhược điểm thứ năm có thể xảy ra khi video bị tách thành nhiều lớp và/hoặc các lớp phụ. Bộ mã hóa có thể mã hóa các lớp này vào trong dòng bit. Hơn nữa, bộ mã hóa có thể dùng HRD để thực hiện các thử nghiệm sự phù hợp để kiểm tra dòng bit về sự phù hợp với các tiêu chuẩn. Bộ mã hóa có thể được tạo cấu hình để gồm các tham số HRD lớp cụ thể vào trong dòng bit để hỗ trợ các thử nghiệm sự phù hợp này. Các tham số HRD lớp cụ thể có thể được mã hóa đối với mỗi lớp trong một số hệ thống tạo mã video. Trong một số trường hợp, các tham số HRD lớp cụ thể là giống nhau đối với mỗi lớp, điều này dẫn đến trong thông tin dư mà làm tăng một cách không cần thiết kích thước của việc mã hóa video.

Trong ví dụ thứ năm, được bộc lộ ở đây các cơ chế để giảm sự dư thừa tham số HRD cho các video dùng nhiều lớp. Bộ mã hóa có thể mã hóa các tham số HRD cho lớp cao nhất. Bộ mã hóa có thể cũng mã hóa cờ hiện diện các tham số CPB lớp phụ (sublayer_cpb_params_present_flag). Sublayer_cpb_params_present_flag có thể được đặt là 0 để chỉ báo rằng tất cả các lớp thấp hơn phải sử dụng cùng các tham số HRD như lớp cao nhất. Trong nội dung này, lớp cao nhất có bộ nhận dạng (identifier, ID) lớp lớn nhất và lớp thấp hơn là lớp bất kỳ mà có ID lớp mà ít hơn ID lớp của lớp cao nhất. Trong cách này, các tham số HRD cho các lớp thấp hơn có thể được bỏ qua khỏi dòng bit. Điều này làm giảm kích thước dòng bit, và từ đó làm giảm việc sử dụng bộ xử lý, bộ nhớ, và/hoặc tài nguyên mạng tại cả hai bộ mã hóa và bộ giải mã.

Nhược điểm thứ sáu liên quan tới việc dùng các tập tham số chuỗi (SPSs) để chứa các phần tử cú pháp liên quan tới mỗi chuỗi video trong video. Các hệ thống tạo mã video có thể tạo mã video trong các lớp và/hoặc các lớp phụ. Các chuỗi video có thể hoạt động khác tại các lớp và/hoặc các lớp phụ khác. Do đó, các lớp khác có thể tham chiếu tới các SPS khác. Thông báo BP SEI có thể chỉ báo các lớp/các lớp phụ được kiểm tra về sự phù hợp với các tiêu chuẩn. Một số các hệ thống tạo mã video có thể chỉ báo rằng thông báo BP SEI áp dụng cho các lớp/các lớp phụ được chỉ báo trong SPS. Điều này có thể gây ra các nhược điểm khi các lớp khác có các SPS khác được tham chiếu như là các SPS có thể gồm thông tin trái ngược, điều này dẫn đến trong các lỗi không mong muốn.

Trong ví dụ thứ sáu, được bộc lộ ở đây là các cơ chế để giải quyết các lỗi liên quan đến việc kiểm tra sự phù hợp khi đa lớp được dùng trong chuỗi video. Cụ thể, thông báo BP SEI được sửa đổi để chỉ báo rằng số lượng bất kỳ các lớp/các lớp phụ được mô tả trong VPS có thể được kiểm tra về sự phù hợp. Ví dụ, thông báo BP SEI có thể chứa phần tử cú pháp các lớp phụ lớn nhất BP trừ một (bp_max_sublayers_minus1) mà chỉ báo số lượng của các lớp/các lớp phụ mà được kết hợp với dữ liệu trong thông báo BP SEI. Trong khi đó, phần tử cú pháp các lớp phụ lớn nhất VPS trừ một (vps_max_sublayers_minus1) trong VPS chỉ báo số lượng của các lớp phụ trong toàn bộ video. Phần tử cú pháp bp_max_sublayers_minus1 có thể được đặt giá trị bất kỳ từ 0 đến giá trị của phần tử cú pháp vps_max_sublayers_minus1. Trong cách này, số lượng bất kỳ các lớp/các lớp phụ trong video có thể được kiểm tra về sự phù hợp trong khi tránh chuỗi dựa trên lớp liên quan đến sự không đồng bộ SPS. Theo đó, sáng chế bộc lộ tránh các lỗi tạo mã dựa trên lớp, và do đó tăng chức năng của bộ mã hóa và/hoặc bộ giải mã. Hơn nữa, ví dụ hiện tại hỗ trợ việc tạo mã dựa trên lớp, mà có thể tăng hiệu quả tạo mã. Như vậy, ví dụ hỗ trợ giảm việc sử dụng bộ xử lý, bộ nhớ và/hoặc tài nguyên mạng ở bộ mã hóa và/hoặc bộ giải mã.

Nhược điểm thứ bảy liên quan tới các lớp mà được gồm trong các OLS. Mỗi OLS chứa ít nhất một lớp đầu ra mà được tạo cấu hình được hiển thị tại bộ giải mã. HRD tại bộ mã hóa có thể kiểm tra mỗi OLS về sự phù hợp với các tiêu chuẩn. OLS phù hợp có thể luôn được giải mã và được hiển thị ở bộ giải mã thích hợp. Quy trình HRD có thể được quản lý một phần bởi các thông báo SEI. Ví dụ, thông báo SEI lồng nhau có thể mở rộng có thể chứa các thông báo SEI được lồng nhau có thể mở rộng. Mỗi thông báo SEI được lồng nhau có thể mở rộng có thể chứa dữ liệu mà liên quan tới lớp tương ứng. Khi thực hiện việc kiểm tra sự phù hợp, HRD có thể thực hiện quy trình trích xuất bit trên OLS đích. Dữ liệu mà không liên quan tới các lớp trong OLS thường bị xóa trước thử nghiệm sự phù hợp nên mỗi OLS có thể được kiểm tra riêng rẽ (ví dụ, trước khi truyền). Một số các hệ thống tạo mã video không xóa các thông báo SEI lồng nhau có thể mở rộng trong quy trình trích xuất dòng bit phụ vì các thông báo này liên quan tới nhiều lớp. Điều này có thể dẫn đến các thông báo SEI lồng nhau có thể mở rộng mà còn lại trong dòng bit sau khi trích xuất dòng bit phụ chỉ khi các thông báo SEI lồng nhau có thể mở rộng không liên quan tới lớp bất kỳ trong OLS đích (OLS được trích xuất). Điều này có thể tăng kích thước của dòng bit cuối cùng mà không cung cấp chức năng bổ sung bất kỳ.

Trong ví dụ thứ bảy, được bộc lộ ở đây các cơ chế để giảm kích thước của các dòng bit đa lớp. Trong khi trích xuất dòng bit phụ, các thông báo SEI lồng nhau có thể mở rộng có thể được xem xét để xóa từ dòng bit. Khi thông báo SEI lồng nhau có thể mở rộng liên quan tới một hoặc nhiều OLS, các thông báo SEI được lồng nhau có thể mở rộng trong thông báo SEI lồng nhau có thể mở rộng được kiểm tra. Khi các thông báo SEI được lồng nhau có thể mở rộng không liên quan tới lớp bất kỳ trong OLS đích, sau đó toàn bộ thông báo SEI lồng nhau có thể mở rộng có thể được xóa khỏi dòng bit. Điều này dẫn đến giảm kích thước của dòng bit được gửi tới bộ giải mã. Theo đó, ví dụ hiện tại tăng hiệu quả tạo mã và giảm việc sử dụng bộ xử lý, bộ nhớ, và/hoặc tài nguyên mạng tại cả hai bộ mã hóa và bộ giải mã.

FIG. 1 là lưu đồ phương pháp hoạt động ví dụ 100 của việc tạo mã tín hiệu video. Cụ thể, tín hiệu video được mã hóa tại bộ mã hóa. Quy trình mã hóa nén tín hiệu video bằng cách dùng các cơ chế để giảm kích thước tệp video khác nhau. Kích thước tệp ít hơn cho phép tệp video được nén được truyền tới người dùng, đồng thời giảm chi phí băng thông được liên kết. Bộ giải mã sau đó giải mã tệp video được nén để tái cấu trúc tín hiệu video gốc để hiển thị tới người dùng cuối. Quy trình giải mã phản ánh chung quy trình mã hóa để cho phép bộ giải mã để tái cấu trúc một cách phù hợp tín hiệu video.

Ở bước 101, tín hiệu video được đưa vào trong bộ mã hóa. Ví dụ, tín hiệu video có thể là tệp video không được nén được lưu trữ trong bộ nhớ. Như ví dụ khác, tệp video có thể

được chụp bởi thiết bị chụp video, như là máy ảnh video, và được mã hóa để hỗ trợ phát trực tiếp video. Tập video có thể gồm cả hai thành phần audio và thành phần video. Thành phần video chứa loạt khung hình ảnh mà, khi được xem theo chuỗi, tạo ấn tượng trực quan về chuyển động. Các khung chứa các điểm ảnh mà được đại diện được biểu thị về ánh sáng, được tham chiếu ở đây như là các thành phần độ chói (hoặc các mẫu độ chói), và màu sắc, mà được tham chiếu như là các thành phần sắc độ (hoặc mẫu màu sắc). Trong một số ví dụ, các khung có thể cũng chứa các giá trị độ sâu để hỗ trợ xem ba chiều.

Ở bước 103, video được phân vùng vào trong các khối. Việc phân vùng gồm chia nhỏ các điểm ảnh trong mỗi khung thành các khối vuông và/hoặc chữ nhật để nén. Ví dụ, trong Mã hóa video hiệu quả cao (HEVC, High Efficiency Video Coding) (còn được gọi là H.265 và MPEG-H Phần 2), trước tiên, khung có thể được chia thành các đơn vị cây tạo mã (CTU, coding tree unit), là các khối có kích thước được xác định trước (ví dụ: 64 điểm ảnh x 64 điểm ảnh). Các CTU chứa cả mẫu độ chói và sắc độ. Cây mã hóa có thể được sử dụng để chia các CTU thành các khối và sau đó chia nhỏ các khối một cách đệ quy cho đến khi đạt được các cấu hình hỗ trợ mã hóa hơn nữa. Ví dụ, các thành phần độ chói của khung có thể được chia nhỏ cho đến khi các khối riêng lẻ chứa các giá trị chiếu sáng tương đối đồng nhất. Hơn nữa, các thành phần sắc độ của khung có thể được chia nhỏ cho đến khi các khối riêng lẻ chứa các giá trị màu tương đối đồng nhất. Theo đó, các cơ chế phân vùng khác nhau tùy thuộc vào nội dung của các khung video.

Ở bước 105, các cơ chế nén khác nhau được sử dụng để nén các khối hình ảnh được phân vùng ở bước 103. Ví dụ, dự đoán liên và/hoặc dự đoán nội bộ có thể được dùng. Dự đoán liên được thiết kế để tận dụng lợi thế của thực tế là các đối tượng trong một cảnh chung có xu hướng xuất hiện trong các khung hình liên tiếp. Theo đó, khối mô tả đối tượng trong khung tham chiếu không cần được mô tả riêng biệt trong các khung liên kế. Cụ thể, đối tượng, như là bảng, có thể ở một vị trí cố định trên nhiều khung. Do đó bảng được mô tả một lần và các khung liên kế có thể tham chiếu trở lại khung tham chiếu. Cơ chế so khớp mẫu có thể được dùng để so khớp các đối tượng qua nhiều khung. Hơn nữa, các đối tượng chuyển động có thể được biểu diễn qua nhiều khung, ví dụ như do chuyển động của đối tượng hoặc chuyển động của máy ảnh. Ví dụ cụ thể, video có thể hiển thị ô tô di chuyển trên màn hình qua nhiều khung. Các vector chuyển động có thể được dùng để mô tả các chuyển động này. Vector chuyển động là vector hai chiều mà cung cấp độ lệch từ tọa độ của đối tượng trong khung tới tọa độ của đối tượng trong khung tham chiếu. Như vậy, dự đoán liên có thể mã hóa khối hình ảnh trong khung hiện tại như tập các vector chuyển động chỉ báo độ lệch từ khối tương ứng trong khung tham chiếu.

Dự đoán nội bộ mã hóa các khối trong khung chung. Dự đoán nội bộ dùng ưu điểm của thực tế là độ chói và các thành phần sắc độ xu hướng tụ lại trong khung. Ví dụ, mảng màu xanh lá trong một phần của cửa cây có xu hướng được đặt liền kề với những mảng xanh tương tự. Dự đoán nội bộ dùng nhiều chế độ dự đoán hướng (ví dụ, 33 trong HEVC), chế độ phẳng, và chế độ dòng điện một chiều (direct current, DC). Các chế độ hướng chỉ báo rằng khối hiện tại là tương tự hoặc giống như mẫu của khối lân cận trong hướng tương ứng. Chế độ phẳng chỉ báo rằng loạt các khối theo hàng/cột (ví dụ, mặt phẳng) có thể được nội suy được dựa trên các khối lân cận tại cạnh của hàng. Chế độ phẳng, trên thực tế, chỉ báo sự chuyển đổi mượt mà của độ sáng/màu sắc trên hàng/cột bằng cách dùng độ dốc không đổi tương đối trong việc thay đổi các giá trị. Chế độ DC được dùng cho làm mịn đường biên và chỉ báo rằng khối tương tự/giống giá trị trung bình được kết hợp với các mẫu của tất cả các khối lân cận được kết hợp với các hướng góc của các chế độ dự đoán hướng. Theo đó, dự đoán nội bộ các khối có thể biểu diễn các khối hình ảnh như các giá trị chế độ dự đoán liên quan khác nhau thay vì các giá trị thực tế. Hơn nữa, dự đoán liên các khối có thể biểu diễn các khối hình ảnh như các giá trị vector chuyển động thay vì các giá trị thực tế. Hơn nữa, các khối dự đoán có thể không được biểu diễn chính xác các khối hình ảnh trong một số trường hợp. Các khác biệt bất kỳ được lưu trữ trong các khối dư. Các biến đổi có thể được áp dụng cho các khối dư để còn nén tệp.

Ở bước 107, các kỹ thuật lọc khác nhau có thể được áp dụng. Trong HEVC, các bộ lọc được áp dụng theo sơ đồ lọc lặp. Sự dự đoán dựa trên khối được thảo luận nêu trên có thể dẫn đến việc tạo ra các hình ảnh kiểu khối tại bộ giải mã. Hơn nữa, sơ đồ dự đoán dựa trên khối có thể mã hóa khối và sau đó tái cấu trúc khối được mã hóa sau đó sử dụng như khối tham chiếu. Sơ đồ lọc lặp áp dụng lặp lại các bộ lọc triệt nhiễu, các bộ lọc giải khối, các bộ lọc lặp thích ứng, và các bộ lọc bù thích ứng mẫu (sample adaptive offset, SAO) tới các khối/các khung. Các bộ lọc này giảm thiểu các hiện tượng tạo khối như vậy để có thể tái cấu trúc chính xác tệp được mã hóa. Hơn nữa, các bộ lọc này giảm thiểu các thành phần lạ trong các khối tham chiếu được tái cấu trúc nên thành phần lạ ít có khả năng tạo thành phần lạ bổ sung trong các khối tiếp theo mà được tạo mã được dựa trên các khối tham chiếu được tái cấu trúc.

Khi tín hiệu video được phân vùng, được nén, và được lọc, dữ liệu kết quả được mã hóa trong dòng bit ở bước 109. Dòng bit gồm dữ liệu được thảo luận ở trên cũng như dữ liệu báo hiệu bất kỳ được mong muốn để hỗ trợ việc tái cấu trúc tín hiệu video chính xác tại bộ giải mã. Ví dụ, dữ liệu này có thể gồm dữ liệu phân vùng, dữ liệu dự đoán, các khối dư, và các cờ khác nhau cung cấp các lệnh tạo mã tới bộ giải mã. Dòng bit có thể được lưu trữ trong

bộ nhớ để truyền tới bộ giải mã theo yêu cầu. Dòng bit có thể cũng được phát rộng và/hoặc phát đa hướng tới nhiều bộ giải mã. Việc tạo dòng bit là quy trình lặp. Theo đó, các bước 101, 103, 105, 107, và 109 có thể xảy ra liên tục và/hoặc đồng thời qua các khung và các khối. Thứ tự được minh họa trong FIG. 1 được trình bày để rõ ràng và dễ thảo luận, và không nhằm giới hạn quy trình tạo mã video ở thứ tự cụ thể.

Bộ giải mã thu dòng bit và bắt đầu quy trình giải mã ở bước 111. Cụ thể, bộ giải mã dùng sơ đồ giải mã entropi để chuyển dòng bit thành cú pháp và dữ liệu video tương ứng. Bộ giải mã dùng dữ liệu cú pháp từ dòng bit để xác định các phân vùng cho các khung ở bước 111. Việc phân vùng cần khớp kết quả của việc phân vùng khối ở bước 103. Mã hóa/giải mã entropi như được dùng ở bước 111 được mô tả. Bộ mã hóa thực hiện nhiều lựa chọn trong quy trình nén, như là lựa chọn các sơ đồ phân vùng khối từ vài lựa chọn khả thi được dựa trên vị trí không gian của nhiều giá trị trong (các) hình ảnh đầu vào. Báo hiệu các lựa chọn trích xuất có thể dùng số lượng lớn các bin. Như được sử dụng ở đây, bin là giá trị nhị phân mà được coi là biến (ví dụ, giá trị bit có thể thay đổi phụ thuộc vào ngữ cảnh). Tạo mã entropi cho phép bộ mã hóa loại bỏ các tùy chọn bất kỳ mà rõ ràng là không khả thi với trường hợp cụ thể, để lại tập tùy chọn được cho phép. Mỗi tùy chọn được cho phép sau đó được gán từ mã. Độ dài của các từ mã được dựa trên số lượng của các tùy chọn được phép (ví dụ, 1 bin cho 2 tùy chọn, 2 bin cho 3 đến 4 tùy chọn, v.v.) Bộ mã hóa sau đó mã hóa từ mã cho tùy chọn được lựa chọn. Sơ đồ này giảm kích thước của các từ mã như các từ mã lớn như được mong muốn để chỉ báo duy nhất lựa chọn từ tập phụ nhỏ các tùy chọn được phép thay vì chỉ báo duy nhất việc lựa chọn từ tập lớn tiềm năng của tất cả các tùy chọn khả thi. Bộ giải mã sau đó giải mã việc lựa chọn bằng cách xác định tập các tùy chọn được phép theo cách thức tương tự cho bộ mã hóa. Bằng cách xác định tập các tùy chọn được phép, bộ giải mã có thể đọc từ mã và xác định việc lựa chọn được tạo ra bởi bộ mã hóa.

Ở bước 113, bộ giải mã thực hiện giải mã khối. Cụ thể, bộ giải mã dùng các biến đổi ngược để tạo các khối dư. Sau đó bộ giải mã dùng các khối dư và các khối dự đoán tương ứng để tái cấu trúc các khối hình ảnh theo việc phân vùng. Các khối dự đoán có thể gồm cả hai khối dự đoán nội bộ và khối dự đoán liên như được tạo tại bộ mã hóa ở bước 105. Các khối hình ảnh được tái cấu trúc sau đó được đặt vào trong các khung của tín hiệu video được tái cấu trúc theo dữ liệu phân vùng được xác định ở bước 111. Cú pháp cho bước 113 có thể cũng được báo hiệu trong dòng bit qua tạo mã entropi như được thảo luận ở trên.

Ở bước 115, việc được thực hiện trên các khung của tín hiệu video được tái cấu trúc theo cách thức tương tự như bước 107 tại bộ mã hóa. Ví dụ, các bộ lọc triệt nhiễu, các bộ lọc giải khối, các bộ lọc lặp thích ứng, và các bộ lọc SAO có thể được áp dụng cho các khung để

loại bỏ các thành phần lạ của khối. Khi các khung được lọc, tín hiệu video có thể được đưa ra bộ hiển thị ở bước 117 để người dùng cuối xem.

FIG. 2 là sơ đồ khối hệ thống tạo mã và giải mã (codec) ví dụ 200 để tạo mã video. Cụ thể, hệ thống codec 200 cung cấp chức năng để hỗ trợ việc triển khai của phương pháp điều hành 100. Hệ thống codec 200 được khái quát hóa để mô tả các thành phần được dùng trong cả hai bộ mã hóa và bộ giải mã. Hệ thống codec 200 thu và phân vùng tín hiệu video như được thảo luận đối với các bước 101 và 103 trong phương pháp điều hành 100, điều này dẫn đến trong tín hiệu được phân vùng video 201. Hệ thống codec 200 sau đó nén tín hiệu được phân vùng video 201 vào trong dòng bit được tạo mã khi hoạt động như bộ mã hóa như được thảo luận đối với các bước 105, 107, và 109 theo phương pháp 100. Khi hoạt động như bộ giải mã, hệ thống codec 200 tạo tín hiệu video đầu ra từ dòng bit như được thảo luận đối với các bước 111, 113, 115, và 117 theo phương pháp điều hành 100. Hệ thống codec 200 gồm thành phần điều khiển tạo mã chung 211, thành phần chia tỷ lệ và lượng tử hóa biến đổi 213, thành phần ước tính nội ảnh 215, thành phần dự đoán nội ảnh 217, thành phần bù chuyển động 219, thành phần ước tính chuyển động 221, thành phần biến đổi ngược và tỷ lệ 229, thành phần phân tích điều khiển bộ lọc 227, thành phần các bộ lọc lặp 225, thành phần bộ đệm ảnh được giải mã 223, và thành phần tạo mã nhị phân thích ứng với ngữ cảnh và định dạng tiêu đề (context adaptive binary arithmetic coding, CABAC) 231. Các thành phần này được ghép như minh họa. Trong FIG. 2, các đường đen chỉ báo sự chuyển động của dữ liệu được mã hóa/được giải mã trong khi các đường đứt nét chỉ báo sự chuyển động của dữ liệu điều khiển mà điều khiển hoạt động của các thành phần khác. Các thành phần của hệ thống codec 200 có thể được hiện diện trong bộ mã hóa. Bộ giải mã có thể gồm tập phụ của các thành phần của hệ thống codec 200. Ví dụ, bộ giải mã có thể gồm thành phần dự đoán nội ảnh 217, thành phần bù chuyển động 219, thành phần biến đổi ngược và tỷ lệ 229, thành phần các bộ lọc lặp 225, và thành phần bộ đệm ảnh được giải mã 223. Các thành phần này hiện đã được mô tả.

Tín hiệu được phân vùng video 201 là chuỗi video được quay mà được phân vùng vào trong các khối của các điểm ảnh bởi cây tạo mã. Cây tạo mã dùng các chế độ chia khác nhau để chia nhỏ khối của các điểm ảnh vào trong các khối ít hơn của các điểm ảnh. Các khối này có thể sau đó còn được chia nhỏ vào trong các khối ít hơn. Các khối có thể được gọi là các nút trên cây tạo mã. Các nút cha lớn hơn được chia được chia vào trong các nút con ít hơn. Số lần nút được chia nhỏ được gọi là độ sâu của nút/cây tạo mã. Các khối được chia có thể được gồm trong các đơn vị tạo mã (CU) trong một số trường hợp. Ví dụ, CU có thể là phần phụ của CTU mà chứa khối độ chói, (các) khối sắc độ (Cr) khác nhau về màu đỏ, các)

khối sắc độ (Cr) khác nhau về màu xanh cùng với các lệnh cú pháp tương ứng cho CU. Các chế độ chia có thể gồm cây nhị phân (BT, binary tree), cây tam phân (TT, triple tree), và cây tứ phân (QT, quad tree) được dùng để chia nút vào trong hai, ba hoặc bốn các nút con, tương ứng, theo các hình khác nhau phụ thuộc vào các chế độ chia được dùng. Tín hiệu được phân vùng video 201 được chuyển tiếp đến thành phần điều khiển tạo mã chung 211, thành phần chia tỷ lệ và lượng tử hóa biến đổi 213, thành phần ước tính nội ảnh 215, thành phần phân tích điều khiển bộ lọc 227, và thành phần ước tính chuyển động 221 để nén.

Thành phần điều khiển tạo mã chung 211 được tạo cấu hình để đưa ra các quyết định liên quan tới tạo mã của các hình ảnh của chuỗi video vào trong dòng bit theo các ràng buộc áp dụng. Ví dụ, thành phần điều khiển tạo mã chung 211 quản lý việc tối ưu hóa của tốc độ bit/kích thước dòng bit so với chất lượng tái cấu trúc. Các quyết định này có thể được tạo ra dựa trên các yêu cầu không gian lưu trữ/hiệu lực băng thông và độ phân giải hình ảnh. Thành phần điều khiển tạo mã chung 211 cũng quản lý việc sử dụng bộ đệm bởi vì tốc độ truyền để giảm thiểu vấn đề chạy dưới mức và chạy quá ngưỡng của bộ đệm. Để quản lý những vấn đề này, thành phần điều khiển tạo mã chung 211 quản lý phân vùng, dự đoán, và lọc bởi các thành phần khác. Ví dụ, thành phần điều khiển tạo mã chung 211 có thể tăng độ phức tạp của việc nén để tăng độ phân giải và tăng băng thông sử dụng hoặc giảm độ phức tạp của việc nén để giảm độ phân giải và băng thông sử dụng. Do đó, thành phần điều khiển tạo mã chung 211 điều khiển các thành phần khác của hệ thống codec 200 để cân bằng chất lượng tái cấu trúc tín hiệu video với mối quan tâm về tốc độ bit. Thành phần điều khiển tạo mã chung 211 tạo dữ liệu điều khiển, mà điều khiển việc hoạt động của các thành phần khác. Dữ liệu điều khiển cũng được chuyển tiếp tới thành phần định dạng tiêu đề và CABAC 231 được mã hóa trong dòng bit thành các tham số tín hiệu để giải mã tại bộ giải mã.

Tín hiệu được phân vùng video 201 cũng được gửi tới thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219 để dự đoán liên. Khung hoặc lát của tín hiệu được phân vùng video 201 có thể được chia vào trong các khối video. Thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219 thực hiện tạo mã dự đoán liên của khối video thu được liên quan tới một hoặc nhiều khối trong một hoặc nhiều khung tham chiếu để cung cấp dự đoán theo thời gian. Hệ thống codec 200 có thể thực hiện tạo mã nhiều lần, ví dụ, để lựa chọn chế độ tạo mã thích hợp cho mỗi khối dữ liệu video.

Thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219 có thể được tích hợp cao, nhưng được minh họa riêng biệt cho các mục đích định khái niệm. Việc ước tính chuyển động, được thực hiện bởi thành phần ước tính chuyển động 221, là quy trình tạo các vectơ chuyển động, mà ước tính chuyển động cho các khối video. Vectơ chuyển động,

ví dụ, có thể chỉ báo sự dịch chuyển của đối tượng được tạo mã liên quan tới khối dự đoán. Khối dự đoán là khối mà được tìm thấy khớp chặt chẽ với khối được tạo mã, về sự khác biệt điểm ảnh. Khối dự đoán cũng có thể được tham chiếu tới như khối tham chiếu. Sự khác biệt điểm ảnh này có thể được xác định bởi tổng chênh lệch tuyệt đối (SAD, sum of absolute difference), tổng chênh lệch bình phương (SSD, sum of square difference), hoặc các chỉ số chênh lệch khác. HEVC dùng một vài đối tượng được tạo mã gồm CTU, các khối cây tạo mã (CTBs), và các CU. Ví dụ, CTU có thể được chia vào trong các CTB, mà sau đó có thể được chia vào trong các CB để gồm trong các CU. CU có thể được mã hóa như đơn vị dự đoán (PU) chứa dữ liệu dự đoán và/hoặc đơn vị biến đổi (TU) chứa dữ liệu dư được biến đổi cho CU. Thành phần ước tính chuyển động 221 tạo các vector chuyển động, các PU, và các TU bằng cách sử dụng phân tích biến dạng tỷ lệ như một phần của quy trình tối ưu hóa biến dạng tỷ lệ. Ví dụ, thành phần ước tính chuyển động 221 có thể xác định các khối tham chiếu, các vector chuyển động, v.v. cho khối/khung hiện tại, và có thể lựa chọn các khối tham chiếu, các vector chuyển động, v.v. có đặc tính biến dạng tỷ lệ tốt nhất. Đặc tính biến dạng tỷ lệ tốt nhất cân bằng cả hai chất lượng việc tái cấu trúc video (ví dụ, lượng dữ liệu mất do việc nén) với hiệu quả tạo mã (ví dụ, kích thước của việc mã hóa cuối cùng).

Trong một số ví dụ, hệ thống codec 200 có thể tính toán các giá trị cho các vị trí điểm ảnh số nguyên phụ của các ảnh tham chiếu được lưu trữ trong thành phần bộ đệm ảnh được giải mã 223. Ví dụ, hệ thống codec video 200 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, thành phần ước tính chuyển động 221 có thể thực hiện tìm kiếm chuyển động liên quan tới các vị trí điểm ảnh đầy đủ và các vị trí điểm ảnh phân số và đưa ra vector chuyển động với độ chính xác điểm ảnh phân số. Thành phần ước tính chuyển động 221 tính toán vector chuyển động đối với PU của khối video trong lát được tạo mã liên bằng cách so sánh vị trí của PU với vị trí của khối dự đoán của ảnh tham chiếu. Thành phần ước tính chuyển động 221 đưa ra vector chuyển động được tính toán như dữ liệu chuyển động tới thành phần định dạng tiêu đề và CABAC 231 để mã hóa và chuyển động thành phần bù chuyển động 219.

Việc bù chuyển động, được thực hiện bởi thành phần bù chuyển động 219, có thể liên quan đến việc tìm nạp hoặc tạo khối dự đoán được dựa trên vector chuyển động được xác định bởi thành phần ước tính chuyển động 221. Lần nữa, thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219 có thể được tích hợp về mặt chức năng, trong một số ví dụ. Khi thu được vector chuyển động cho PU của khối video hiện tại, thành phần bù chuyển động 219 có thể định vị khối dự đoán tới các điểm vector chuyển động. Khối video dư sau đó

được tạo bằng cách trừ các giá trị điểm ảnh của khối dự đoán từ các giá trị điểm ảnh của khối video hiện tại đang được tạo mã, tạo thành các giá trị chênh lệch điểm ảnh. Nói chung, thành phần ước tính chuyển động 221 thực hiện việc ước tính chuyển động liên quan tới các thành phần độ chói, và thành phần bù chuyển động 219 sử dụng các vector chuyển động được tính toán dựa trên các thành phần độ chói cho cả hai các thành phần sắc độ và các thành phần độ chói. Khối dự đoán và khối dư được chuyển tiếp tới thành phần chia tỷ lệ và lượng tử hóa biến đổi 213.

Tín hiệu được phân vùng video 201 cũng được gửi tới thành phần ước tính nội ảnh 215 và thành phần dự đoán nội ảnh 217. Như với thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219, thành phần ước tính nội ảnh 215 và thành phần dự đoán nội ảnh 217 có thể được tích hợp cao, nhưng được minh họa riêng biệt cho các mục đích định khái niệm. Thành phần ước tính nội ảnh 215 và thành phần dự đoán nội ảnh 217 dự đoán nội bộ khối hiện tại liên quan tới các khối trong khung hiện tại, như sự thay thế cho dự đoán liên được thực hiện bởi thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219 giữa các khung, như được mô tả phía trên. Đặc biệt, thành phần ước tính nội ảnh 215 xác định chế độ dự đoán nội bộ để sử dụng mã hóa khối hiện tại. Trong một số ví dụ, thành phần ước tính nội ảnh 215 lựa chọn chế độ dự đoán nội bộ phù hợp để mã hóa khối hiện tại từ nhiều chế độ dự đoán nội bộ được thử nghiệm. Các chế độ dự đoán nội bộ được lựa chọn sau đó được chuyển tiếp tới thành phần định dạng tiêu đề và CABAC 231 để mã hóa.

Ví dụ, thành phần ước tính nội ảnh 215 tính toán các giá trị biến dạng tỷ lệ sử dụng phân tích biến dạng tỷ lệ cho các chế độ dự đoán nội bộ thử nghiệm khác nhau, và lựa chọn chế độ dự đoán nội bộ có đặc tính biến dạng tỷ lệ tốt nhất giữa các chế độ thử nghiệm. Phân tích biến dạng tỷ lệ xác định lượng biến dạng (hoặc lỗi) giữa khối được mã hóa và khối không được mã hóa gốc được mã hóa để cung cấp khối được mã hóa, cũng như tốc độ bit (ví dụ, số lượng bit) được sử dụng để cung cấp khối được mã hóa. Thành phần ước tính nội ảnh 215 tính toán các tỷ số từ các biến dạng và các tốc độ cho các khối được mã hóa khác nhau để xác định chế độ dự đoán nội bộ thể hiện giá trị biến dạng tỷ lệ tốt nhất cho khối. Hơn nữa, thành phần ước tính nội ảnh 215 có thể được tạo cấu hình để tạo mã các khối độ sâu của sơ đồ độ sâu sử dụng chế độ mô hình độ sâu (DMM, depth modeling mode) được dựa trên tối ưu hóa biến dạng tỷ lệ (RDO, rate-distortion optimization).

Thành phần dự đoán nội ảnh 217 có thể tạo khối dư từ khối dự đoán được dựa trên các chế độ dự đoán nội bộ được lựa chọn được xác định bởi thành phần ước tính nội ảnh 215 khi được triển khai trên bộ mã hóa hoặc đọc khối dư từ dòng bit khi được triển khai trên bộ giải mã. Khối dư gồm sự khác biệt về các giá trị giữa khối dự đoán và khối gốc, được biểu

diễn như ma trận. Khối dư sau đó được chuyển tiếp tới thành phần chia tỷ lệ và lượng tử hóa biến đổi 213. Thành phần ước tính nội ảnh 215 và thành phần dự đoán nội ảnh 217 có thể hoạt động trên cả hai độ chói và các thành phần sắc độ.

Thành phần chia tỷ lệ và lượng tử hóa biến đổi 213 được tạo cấu hình còn để nén khối dư. Thành phần chia tỷ lệ và lượng tử hóa biến đổi 213 áp dụng biến đổi, như là biến đổi cosin rời rạc (DCT, discrete cosine transform), biến đổi sin rời rạc (DST, discrete sine transform), hoặc biến đổi tương tự về mặt khái niệm, tới khối dư, cung cấp khối video bao gồm các giá trị hệ số biến đổi dư. Các biến đổi sóng cầu thứ cấp, các biến đổi số nguyên, các biến đổi băng phụ hoặc các loại khác của các biến đổi có thể cũng được sử dụng. Việc biến đổi có thể chuyển đổi thông tin dư từ miền giá trị điểm ảnh tới miền biến đổi, như là miền tần số. Thành phần chia tỷ lệ và lượng tử hóa biến đổi 213 cũng được tạo cấu hình để chia tỷ lệ thông tin dư biến đổi, ví dụ được dựa trên tần số. Việc chia tỷ lệ này liên quan đến việc áp dụng hệ số tỷ lệ cho thông tin dư để thông tin tần số khác được lượng tử hóa ở mức độ chi tiết khác, điều này có thể ảnh hưởng đến chất lượng hình ảnh cuối cùng của video được tái cấu trúc. Thành phần chia tỷ lệ và lượng tử hóa biến đổi 213 cũng được tạo cấu hình để lượng tử hóa hệ số biến đổi còn để giảm tốc độ bit. Quy trình lượng tử hóa có thể giảm độ sâu bit được kết hợp với một số hoặc tất cả các hệ số. Mức độ lượng tử hóa có thể được sửa đổi bằng cách điều chỉnh tham số lượng tử hóa. Trong một số ví dụ, thành phần chia tỷ lệ và lượng tử hóa biến đổi 213 có thể sau đó thực hiện quét của ma trận gồm hệ số biến đổi lượng tử hóa. Hệ số biến đổi lượng tử hóa được chuyển tiếp tới thành phần định dạng tiêu đề và CABAC 231 được mã hóa trong dòng bit.

Thành phần biến đổi ngược và tỷ lệ 229 áp dụng hoạt động ngược của thành phần chia tỷ lệ và lượng tử hóa biến đổi 213 để hỗ trợ việc ước tính chuyển động. Thành phần biến đổi ngược và tỷ lệ 229 áp dụng chia tỷ lệ, biến đổi, và/hoặc lượng tử hóa ngược để tái cấu trúc khối dư trong miền điểm ảnh, ví dụ, sau đó sử dụng như khối tham chiếu mà có thể trở thành một khối dự đoán cho khối hiện tại khác. Thành phần ước tính chuyển động 221 và/hoặc thành phần bù chuyển động 219 có thể tính toán khối tham chiếu bằng cách bổ sung khối dư quay lại khối dự đoán tương ứng để sử dụng trong việc ước tính chuyển động của khối/khung sau đó. Các bộ lọc được áp dụng cho các khối tham chiếu được tái cấu trúc để giảm thiểu các thành phần lạ được tạo trong khi chia tỷ lệ, lượng tử, và biến đổi. Các thành phần lạ này mặt khác có thể gây ra dự đoán không chính xác (và tạo các thành phần lạ bổ sung) khi các khối tiếp theo được dự đoán.

Thành phần phân tích điều khiển bộ lọc 227 và thành phần các bộ lọc lặp 225 áp dụng các bộ lọc cho các khối dư và/hoặc các khối hình ảnh được tái cấu trúc. Ví dụ, khối dư được

biến đổi từ thành phần biến đổi ngược và tỷ lệ 229 có thể được kết hợp với khối dự đoán tương ứng từ thành phần dự đoán nội ảnh 217 và/hoặc thành phần bù chuyển động 219 để tái cấu trúc khối hình ảnh gốc. Các bộ lọc có thể sau đó được áp dụng cho khối hình ảnh được tái cấu trúc. Trong một số ví dụ, các bộ lọc có thể thay thế được áp dụng cho các khối dự. Như với các thành phần khác trong FIG. 2, thành phần phân tích điều khiển bộ lọc 227 và thành phần các bộ lọc lặp 225 được tích hợp cao và có thể được triển khai cùng nhau, nhưng được mô tả riêng biệt cho các mục đích định khái niệm. Các bộ lọc được áp dụng cho các khối tham chiếu được tái cấu trúc được áp dụng cho các vùng không gian cụ thể và gồm nhiều tham số để điều chỉnh theo cách các bộ lọc này được áp dụng. Thành phần phân tích điều khiển bộ lọc 227 phân tích các khối tham chiếu được tái cấu trúc để xác định trong đó các bộ lọc này được áp dụng và các tập tham số tương ứng. dữ liệu này được chuyển tiếp đến thành phần định dạng tiêu đề và CABAC 231 như dữ liệu điều khiển bộ lọc để mã hóa. Thành phần các bộ lọc lặp 225 áp dụng các bộ lọc này được dựa trên dữ liệu điều khiển bộ lọc. Các bộ lọc có thể gồm bộ lọc giải khối, bộ lọc triệt nhiễu, bộ lọc SAO, và bộ lọc lặp thích ứng. Các bộ lọc này có thể được áp dụng trong miền không gian/điểm ảnh (ví dụ, trên khối điểm ảnh được tái cấu trúc) hoặc trong miền tần số, phụ thuộc vào ví dụ.

Khi hoạt động như bộ mã hóa, khối hình ảnh được tái cấu trúc được lọc, khối dự, và/hoặc khối dự đoán được lưu trữ trong thành phần bộ đệm ảnh được giải mã 223 để sử dụng sau này trong việc ước tính chuyển động như được thảo luận ở trên. Khi hoạt động như bộ giải mã, thành phần bộ đệm ảnh được giải mã 223 lưu trữ và chuyển tiếp các khối được tái cấu trúc và được lọc tới bộ hiển thị như một phần của tín hiệu video đầu ra. Thành phần bộ đệm ảnh được giải mã 223 có thể là thiết bị nhớ bất kỳ có khả năng lưu trữ các khối dự đoán, các khối dự, và/hoặc các khối hình ảnh được tái cấu trúc.

Thành phần định dạng tiêu đề và CABAC 231 thu dữ liệu từ các thành phần khác nhau của hệ thống codec 200 và mã hóa dữ liệu này vào trong dòng bit được tạo mã để truyền tới bộ giải mã. Cụ thể, thành phần định dạng tiêu đề và CABAC 231 tạo các tiêu đề khác nhau để mã hóa dữ liệu điều khiển, như là dữ liệu điều khiển và dữ liệu điều khiển bộ lọc nói chung. Hơn nữa, dữ liệu dự đoán, gồm dự đoán nội bộ và dữ liệu chuyển động, cũng như dữ liệu dư trong dạng của dữ liệu hệ số biến đổi được lượng tử hóa tất cả được mã hóa trong dòng bit. Dòng bit cuối cùng gồm tất cả thông tin được mong muốn bởi bộ giải mã để tái cấu trúc tín hiệu được phân vùng video gốc 201. Thông tin này có thể cũng gồm các bảng chỉ số chế độ dự đoán nội bộ (còn được gọi là các bảng ánh xạ từ mã), các định nghĩa về các ngữ cảnh mã hóa cho các khối khác nhau, các chỉ báo của các chế độ dự đoán nội bộ để xảy ra nhất, sự chỉ báo của thông tin phân vùng, v.v. dữ liệu này có thể được mã hóa bằng cách dùng

tạo mã entropy. Ví dụ, thông tin có thể được mã hóa bằng cách dùng việc tạo mã độ dài biến đổi thích ứng với ngữ cảnh (CAVLC, context adaptive variable length coding), CABAC, tạo mã số học nhị phân thích ứng với ngữ cảnh dựa trên cú pháp (SBAC, syntax-based context-adaptive binary arithmetic coding), tạo mã entropy phân vùng theo khoảng xác suất (PIPE, probability interval partitioning entropy), hoặc các kỹ thuật tạo mã entropy khác. Sau đây tạo mã entropy, dòng bit được tạo mã có thể được truyền tới thiết bị khác (ví dụ, bộ giải mã video) hoặc được lưu trữ để truyền hoặc trích xuất sau này.

FIG. 3 là sơ đồ khối minh họa bộ mã hóa video ví dụ 300. Bộ mã hóa video 300 có thể được dùng để triển khai chức năng mã hóa của hệ thống codec 200 và/hoặc triển khai các bước 101, 103, 105, 107, và/hoặc 109 của phương pháp điều hành 100. Bộ mã hóa 300 phân vùng tín hiệu video đầu vào, dẫn đến tín hiệu được phân vùng video 301, về cơ bản tương tự như tín hiệu được phân vùng video 201. Tín hiệu được phân vùng video 301 sau đó được nén và được mã hóa vào trong dòng bit bởi các thành phần của bộ mã hóa 300.

Cụ thể, tín hiệu được phân vùng video 301 được chuyển tiếp đến thành phần dự đoán nội ảnh 317 để dự đoán nội bộ. Thành phần dự đoán nội ảnh 317 về cơ bản có thể tương tự như thành phần ước tính nội ảnh 215 và thành phần dự đoán nội ảnh 217. Tín hiệu được phân vùng video 301 cũng được chuyển tiếp tới thành phần bù chuyển động 321 để dự đoán liên được dựa trên các khối tham chiếu trong thành phần bộ đệm ảnh được giải mã 323. Thành phần bù chuyển động 321 về cơ bản có thể tương tự như thành phần ước tính chuyển động 221 và thành phần bù chuyển động 219. Các khối dự đoán và các khối dư từ thành phần dự đoán nội ảnh 317 và thành phần bù chuyển động 321 được chuyển tiếp tới thành phần biến đổi và lượng tử hóa 313 để biến đổi và lượng tử hóa các khối dư. Thành phần biến đổi và lượng tử hóa 313 về cơ bản có thể tương tự như thành phần chia tỷ lệ và lượng tử hóa biến đổi 213. Các khối dư được biến đổi và được lượng tử hóa và các khối dự đoán tương ứng (along with dữ liệu điều khiển được liên kết) được chuyển tiếp tới thành phần tạo mã entropy 331 để tạo mã vào trong dòng bit. Thành phần tạo mã entropy 331 về cơ bản có thể tương tự như thành phần định dạng tiêu đề và CABAC 231.

Các khối dư được biến đổi và được lượng tử hóa và/hoặc các khối dự đoán tương ứng cũng được chuyển tiếp từ thành phần biến đổi và lượng tử hóa 313 tới thành phần biến đổi và lượng tử hóa ngược 329 cho việc tái cấu trúc vào trong các khối tham chiếu để sử dụng bởi thành phần bù chuyển động 321. Thành phần biến đổi và lượng tử hóa ngược 329 về cơ bản có thể tương tự như thành phần biến đổi ngược và tỷ lệ 229. Các bộ lọc lặp trong thành phần các bộ lọc lặp 325 cũng được áp dụng cho các khối dư và/hoặc các khối tham chiếu được tái cấu trúc, phụ thuộc vào ví dụ. Thành phần các bộ lọc lặp 325 về cơ bản có thể tương

tự như thành phần phân tích điều khiển bộ lọc 227 và thành phần các bộ lọc lặp 225. Thành phần các bộ lọc lặp 325 có thể gồm nhiều bộ lọc như được thảo luận đối với thành phần các bộ lọc lặp 225. Các khối lọc sau đó được lưu trữ trong thành phần bộ đệm ảnh được giải mã 323 để sử dụng như các khối tham chiếu bởi thành phần bù chuyển động 321. Thành phần bộ đệm ảnh được giải mã 323 về cơ bản có thể tương tự như thành phần bộ đệm ảnh được giải mã 223.

FIG. 4 là sơ đồ khối minh họa bộ giải mã video ví dụ 400. Bộ giải mã video 400 có thể được dùng để triển khai các chức năng giải mã của hệ thống codec 200 và/hoặc triển khai các bước 111, 113, 115, và/hoặc 117 của phương pháp điều hành 100. Bộ giải mã 400 thu dòng bit, ví dụ từ bộ mã hóa 300, và tạo tín hiệu video đầu ra được tái cấu trúc được dựa trên dòng bit để hiển thị tới người dùng cuối.

Dòng bit thu được bởi thành phần giải mã entropi 433. Thành phần giải mã entropi 433 được tạo cấu hình để triển khai sơ đồ giải mã entropi, như là CAVLC, CABAC, SBAC, tạo mã PIPE, hoặc các kỹ thuật tạo mã entropi khác. Ví dụ, thành phần giải mã entropi 433 có thể dùng thông tin phần tiêu đề để cung cấp ngữ cảnh để diễn giải dữ liệu bổ sung được mã hóa dưới dạng các từ mã trong dòng bit. Thông tin giải mã gồm thông tin được mong muốn bất kỳ để giải mã tín hiệu video, như là dữ liệu điều khiển chung, dữ liệu điều khiển bộ lọc, thông tin phân vùng, dữ liệu chuyển động, dữ liệu dự đoán, và được lượng tử hóa hệ số biến đổi từ các khối dư. Hệ số biến đổi lượng tử hóa được chuyển tiếp tới thành phần biến đổi và lượng tử hóa ngược 429 cho việc tái cấu trúc vào trong các khối dư. Thành phần biến đổi và lượng tử hóa ngược 429 có thể giống với thành phần biến đổi và lượng tử hóa ngược 329.

Các khối dư và/hoặc các khối dự đoán tái cấu trúc được chuyển tiếp tới thành phần dự đoán nội ảnh 417 cho việc tái cấu trúc vào trong các khối hình ảnh được dựa trên các hoạt động dự đoán nội bộ. Thành phần dự đoán nội ảnh 417 có thể tương tự với thành phần ước tính nội ảnh 215 và thành phần dự đoán nội ảnh 217. Cụ thể, thành phần dự đoán nội ảnh 417 dùng các chế độ dự đoán để định vị khối tham chiếu trong khung và áp dụng khối dư cho kết quả để tái cấu trúc các khối hình ảnh dự đoán nội bộ. Các khối hình ảnh dự đoán nội bộ được tái cấu trúc và/hoặc các khối dư và dữ liệu dự đoán liên tương ứng được chuyển tiếp tới thành phần bộ đệm ảnh được giải mã 423 qua thành phần các bộ lọc lặp 425, mà về cơ bản có thể tương tự như thành phần bộ đệm ảnh được giải mã 223 và thành phần các bộ lọc lặp 225, tương ứng. Thành phần các bộ lọc lặp 425 lọc các khối hình ảnh được tái cấu trúc, các khối dư và/hoặc các khối dự đoán, và thông tin này được lưu trữ trong thành phần bộ đệm ảnh được giải mã 423. Các khối hình ảnh được tái cấu trúc từ thành phần bộ đệm ảnh được giải

mã 423 được chuyển tiếp tới thành phần bù chuyển động 421 cho dự đoán liên. Thành phần bù chuyển động 421 về cơ bản có thể tương tự như thành phần ước tính chuyển động 221 và/hoặc thành phần bù chuyển động 219. Cụ thể, thành phần bù chuyển động 421 dùng các vector chuyển động từ khối tham chiếu để tạo khối dự đoán và áp dụng khối dư cho kết quả để tái cấu trúc khối hình ảnh. Các khối tái cấu trúc kết quả có thể cũng được chuyển tiếp qua thành phần các bộ lọc lặp 425 tới thành phần bộ đệm ảnh được giải mã 423. Thành phần bộ đệm ảnh được giải mã 423 tiếp tục lưu trữ các khối hình ảnh được tái cấu trúc bổ sung, mà có thể được tái cấu trúc vào trong các khung qua thông tin phân vùng. Các khung này có thể cũng được đặt trong chuỗi. chuỗi đưa ra tới bộ hiển thị như tín hiệu video đầu ra được tái cấu trúc.

FIG. 5 là sơ đồ khối minh họa HRD 500 ví dụ. HRD 500 có thể được dùng trong bộ mã hóa, như là hệ thống codec 200 và/hoặc bộ mã hóa 300. HRD 500 có thể kiểm tra dòng bit được tạo ở bước 109 của phương pháp 100 trước dòng bit được chuyển tiếp đến bộ giải mã, như là bộ giải mã 400. Trong một số ví dụ, dòng bit có thể được chuyển tiếp liên tục thông qua HRD 500 như dòng bit được mã hóa. Trong trường hợp đó một phần của dòng bit không tuân thủ các ràng buộc được liên kết, HRD 500 có thể chỉ báo lỗi này cho bộ mã hóa để bộ mã hóa để mã hóa lại phần tương ứng của dòng bit với các cơ chế khác.

HRD 500 gồm bộ lập lịch dòng giả định (HSS, hypothetical stream scheduler) 541. HSS 541 là thành phần được tạo cấu hình để thực hiện cơ chế phân phối giả định. Cơ chế phân phối giả định được sử dụng để kiểm tra sự phù hợp của dòng bit hoặc bộ giải mã liên quan tới luồng thời gian và dữ liệu của dòng bit 551 đưa vào vào trong HRD 500. Ví dụ, HSS 541 có thể thu đầu ra dòng bit 551 từ bộ mã hóa và quản lý quy trình thử nghiệm sự phù hợp trên dòng bit 551. Trong ví dụ cụ thể, HSS 541 có thể điều khiển tốc độ mà các ảnh được tạo mã di chuyển thông qua HRD 500 và xác minh rằng dòng bit 551 không chứa dữ liệu không phù hợp.

HSS 541 có thể chuyển tiếp dòng bit 551 tới CPB 543 ở tốc độ định trước. HRD 500 có thể quản lý dữ liệu trong các đơn vị giải mã (DU, decoding unit) 553. DU 553 là AU hoặc tập con của AU và các đơn vị lớp trừu tượng mạng (network abstraction layer, NAL) không phải lớp tạo mã video (VCL, video coding layer) liên kết. Cụ thể, AU chứa một hoặc nhiều ảnh được kết hợp với thời gian đầu ra. Ví dụ, AU có thể chứa một ảnh trong dòng bit một lớp, và có thể chứa ảnh đối với mỗi lớp trong dòng bit đa lớp. Mỗi ảnh của AU có thể được chia vào trong các lát mà mỗi lát được gồm trong đơn vị VCL NAL tương ứng. Do đó, DU 553 có thể chứa một hoặc nhiều ảnh, một hoặc nhiều lát ảnh, hoặc các sự kết hợp của chúng. Ngoài ra, các tham số được sử dụng để giải mã AU, các ảnh, và/hoặc các lát có thể được gồm

trong các đơn vị NAL không VCL. Như vậy, DU 553 chứa các đơn vị NAL không VCL mà chứa dữ liệu cần thiết để hỗ trợ giải mã các đơn vị VCL NAL trong DU 553. CPB 543 là bộ đệm vào trước ra trước trong HRD 500. CPB 543 chứa các DU 553 gồm dữ liệu video theo thứ tự giải mã. CPB 543 lưu trữ dữ liệu video để sử dụng trong khi xác thực sự phù hợp dòng bit.

CPB 543 chuyển tiếp các DU 553 tới thành phần quy trình giải mã 545. Thành phần quy trình giải mã 545 là thành phần mà phù hợp với tiêu chuẩn VVC. Ví dụ, thành phần quy trình giải mã 545 có thể vượt qua bộ giải mã 400 được dùng bởi người dùng cuối. Thành phần quy trình giải mã 545 giải mã các DU 553 với tốc độ mà có thể đặt được bởi bộ giải mã người dùng cuối ví dụ. Nếu thành phần quy trình giải mã 545 không thể giải mã các DU 553 đủ nhanh để ngăn chặn tràn CPB 543, sau đó dòng bit 551 không phù hợp với tiêu chuẩn và cần được mã hóa lại.

Thành phần quy trình giải mã 545 giải mã các DU 553, mà tạo các DU được giải mã 555. DU được giải mã 555 chứa ảnh được giải mã. Các DU được giải mã 555 được chuyển tiếp tới DPB 547. DPB 547 về cơ bản có thể tương tự như thành phần bộ đệm ảnh được giải mã 223, 323, và/hoặc 423. Để hỗ trợ dự đoán liên, các ảnh được đánh dấu để sử dụng như các ảnh tham chiếu 556 mà thu được từ các DU được giải mã 555 được trả lại cho thành phần quy trình giải mã 545 để hỗ trợ giải mã. DPB 547 đưa ra chuỗi video được giải mã như loạt ảnh 557. Các ảnh 557 là các ảnh được tái cấu trúc thường là những bức ảnh phản chiếu được mã hóa vào trong dòng bit 551 bởi bộ mã hóa.

Các ảnh 557 được chuyển tiếp tới thành phần cắt đầu ra 549. Thành phần cắt đầu ra 549 được tạo cấu hình để áp dụng cửa sổ cắt phù hợp cho các ảnh 557. Điều này dẫn đến các ảnh được cắt đầu ra 559. Ảnh được cắt đầu ra 559 là ảnh được tái cấu trúc hoàn chỉnh. Theo đó, ảnh được cắt đầu ra 559 bất chước những gì người dùng cuối sẽ thấy khi giải mã dòng bit 551. Như vậy, bộ mã hóa có thể xem các ảnh được cắt đầu ra 559 để đảm bảo việc mã hóa đạt yêu cầu.

HRD 500 được khởi tạo được dựa trên các tham số HRD trong dòng bit 551. Ví dụ, HRD 500 có thể đọc các tham số HRD từ VPS, SPS, và/hoặc các thông báo SEI. HRD 500 có thể sau đó thực hiện các hoạt động thử nghiệm sự phù hợp trên dòng bit 551 được dựa trên thông tin trong các tham số HRD này. Như ví dụ cụ thể, HRD 500 có thể xác định một hoặc nhiều lập lịch phân phối CPB 561 từ các tham số HRD. Lập lịch phân phối xác định thời gian để phân phối dữ liệu video tới và/hoặc từ vị trí bộ nhớ, như là CPB và/hoặc DPB. Do đó, lập lịch phân phối CPB 561 xác định thời gian để phân phối của các AU, DU 553, và/hoặc các ảnh, tới/từ CPB 543. Ví dụ, lập lịch phân phối CPB 561 có thể mô tả các tốc độ bit và các

kích thước bộ đệm cho CPB 543, trong đó các tốc độ bit và các kích thước bộ đệm này tương ứng với phân lớp cụ thể của bộ giải mã và/hoặc điều kiện mạng. Do đó, lập lịch phân phối CPB 561 có thể chỉ báo dữ liệu dài bao nhiêu có thể vẫn còn trong CPB 543 trước khi loại bỏ. Việc không duy trì lập lịch phân phối CPB 561 tại HRD 500 trong khi thử nghiệm sự phù hợp là sự chỉ báo rằng bộ giải mã mà tương ứng với lập lịch phân phối CPB 561 sẽ không thể giải mã dòng bit tương ứng. Cần lưu ý rằng HRD 500 có thể dùng DPB lập lịch phân phối cho DPB 547 mà tương tự với các lập lịch phân phối CPB 561.

Video có thể được tạo mã vào trong các lớp khác và/hoặc các OLS để sử dụng bởi các bộ giải mã với các mức độ khác nhau của dung lượng phần cứng cũng như các điều kiện mạng khác nhau. Các lập lịch phân phối CPB 561 được lựa chọn để phản ánh những vấn đề này. Theo đó, các dòng bit phụ lớp cao hơn được chỉ định cho phần cứng và các điều kiện mạng tối ưu và do đó các lớp cao hơn có thể thu một hoặc nhiều lập lịch phân phối CPB 561 mà dùng lượng lớn bộ nhớ trong CPB 543 và các trễ ngắn để chuyển các DU 553 tới DPB 547. Tương tự như vậy, các dòng bit phụ lớp thấp hơn được chỉ định cho dung lượng phần cứng bộ giải mã giới hạn và/hoặc điều kiện mạng kém. Do đó, các lớp thấp hơn có thể thu một hoặc nhiều lập lịch phân phối CPB 561 mà dùng lượng nhỏ bộ nhớ trong CPB 543 và các độ trễ dài hơn để chuyển các DU 553 tới DPB 547. Các OLS, các lớp, các lớp phụ, hoặc các sự kết hợp của chúng sau đó có thể được thử nghiệm theo lập lịch phân phối tương ứng 561 để đảm bảo mà dòng bit phụ kết quả có thể được giải mã chính xác dưới các điều kiện mà được mong đợi cho cho dòng bit phụ. Các lập lịch phân phối CPB 561 mà mỗi trong số chúng được kết hợp với chỉ số lập lịch (ScIdx) 563. ScIdx 563 là chỉ số mà nhận dạng lập lịch phân phối. Theo đó, các tham số HRD trong dòng bit 551 có thể chỉ báo các lập lịch phân phối CPB 561 bởi ScIdx 563 cũng như gồm dữ liệu đầy đủ để cho phép HRD 500 để xác định các lập lịch phân phối CPB 561 và các lập lịch phân phối CPB 561 thương quan với các OLS tương ứng, các lớp, và/hoặc các lớp phụ.

FIG. 6 là sơ đồ khối minh họa chuỗi video nhiều lớp ví dụ 600 được tạo cấu hình để dự đoán liên lớp 621. Chuỗi video nhiều lớp 600 có thể được mã hóa bởi bộ mã hóa, như là hệ thống codec 200 và/hoặc bộ mã hóa 300 và được giải mã bởi bộ giải mã, như là hệ thống codec 200 và/hoặc bộ giải mã 400, ví dụ theo phương pháp 100. Hơn nữa, chuỗi video nhiều lớp 600 có thể được kiểm tra cho sự phù hợp tiêu chuẩn bởi HRD, như là HRD 500. Chuỗi video nhiều lớp 600 được bao gồm để mô tả ứng dụng ví dụ cho các lớp trong chuỗi video được tạo mã. Chuỗi video nhiều lớp 600 là chuỗi video bất kỳ mà dùng nhiều lớp, như là lớp N 631 và lớp N+1 632.

Trong ví dụ, chuỗi video nhiều lớp 600 có thể dùng dự đoán liên lớp 621. Dự đoán liên lớp 621 được áp dụng giữa các ảnh 611, 612, 613, và 614 và các ảnh 615, 616, 617, và 618 trong các lớp khác. Trong ví dụ minh họa, các ảnh 611, 612, 613, và 614 là một phần của lớp N+1 632 và các ảnh 615, 616, 617, và 618 là một phần của lớp N 631. Lớp, như là lớp N 631 và/hoặc lớp N+1 632, là nhóm các ảnh mà tất cả được kết hợp với giá trị đặc tính tương tự, như là tương tự về kích thước, chất lượng, độ phân giải, tỷ lệ tín hiệu trên nhiễu, dung lượng, v.v.. Lớp có thể được định nghĩa về hình thức như tập các đơn vị VCL NAL và các đơn vị NAL không VCL được liên kết. Đơn vị VCL NAL là đơn vị NAL được tạo mã để chứa dữ liệu video, như là lát được tạo mã của ảnh. Đơn vị NAL không VCL là đơn vị NAL mà chứa dữ liệu không phải video như là cú pháp và/hoặc các tham số mà hỗ trợ giải mã dữ liệu video, việc thực hiện kiểm tra sự phù hợp, hoặc các hoạt động khác.

Trong ví dụ được minh họa, lớp N+1 632 được kết hợp với kích thước hình ảnh lớn hơn lớp N 631. Theo đó, các ảnh 611, 612, 613, và 614 trong lớp N+1 632 có kích thước ảnh lớn hơn (ví dụ, chiều cao và chiều rộng lớn hơn và do đó nhiều mẫu hơn) các ảnh 615, 616, 617, và 618 trong lớp N 631 trong ví dụ này. Tuy nhiên, các ảnh này có thể được chia giữa lớp N+1 632 và lớp N 631 bởi các đặc tính khác. Trong khi chỉ hai lớp, lớp N+1 632 và lớp N 631, được minh họa, tập các ảnh có thể được chia vào trong số lượng bất kỳ các lớp được dựa trên các đặc tính được liên kết. Lớp N+1 632 và lớp N 631 có thể cũng được biểu thị bởi ID lớp. ID lớp là mục dữ liệu mà được kết hợp với ảnh và biểu thị ảnh là một phần của lớp được chỉ báo. Theo đó, mỗi ảnh 611-618 có thể được kết hợp với ID lớp tương ứng để chỉ báo lớp N+1 632 hoặc lớp N 631 gồm ảnh tương ứng. Ví dụ, ID lớp có thể gồm bộ nhận dạng lớp tiêu đề đơn vị NAL (nuh_layer_id), là phần tử cú pháp mà chỉ định bộ nhận dạng của lớp mà gồm đơn vị NAL (ví dụ, gồm các lát và/hoặc các tham số của các ảnh trong lớp). Lớp được kết hợp với dòng bit chất lượng/kích thước thấp hơn, như là lớp N 631, được gán chung ID lớp thấp hơn và được gọi là lớp thấp hơn. Hơn nữa, lớp được kết hợp với dòng bit chất lượng/kích thước cao hơn, như là lớp N+1 632, được gán chung ID lớp cao hơn và được gọi là lớp cao hơn.

Các ảnh 611-618 trong các lớp khác 631-632 được tạo cấu hình để được hiển thị thay thế. Như vậy, các ảnh trong các lớp khác 631-632 có thể chia sẻ ID theo thời gian 622 miễn là các ảnh được gồm trong cùng AU. ID theo thời gian 622 là phần tử dữ liệu mà chỉ báo dữ liệu tương ứng với vị trí theo thời gian trong chuỗi video. AU là tập đơn vị NAL mà được kết hợp với với nhau theo nguyên tắc phân loại cụ thể và thuộc về một thời gian đầu ra cụ thể. Ví dụ, AU có thể gồm một hoặc nhiều ảnh trong các lớp khác, như là ảnh 611 và ảnh 615 khi các ảnh này được kết hợp với cùng ID theo thời gian 622. Như ví dụ cụ thể, bộ giải mã

có thể giải mã và hiển thị ảnh 615 tại thời gian hiển thị hiện tại nếu ảnh ít hơn được mong muốn hoặc bộ giải mã có thể giải mã và hiển thị ảnh 611 tại thời gian hiển thị hiện tại nếu ảnh lớn hơn được mong đợi. Như vậy, các ảnh 611-614 ở lớp cao hơn N+1 632 về cơ bản chứa cùng dữ liệu hình ảnh như các ảnh tương ứng 615-618 ở lớp thấp hơn N 631 (mặc dù có sự khác biệt về kích thước ảnh). Cụ thể, ảnh 611 chứa về cơ bản cùng dữ liệu hình ảnh như ảnh 615, ảnh 612 chứa về cơ bản cùng dữ liệu hình ảnh như ảnh 616, v.v..

Các ảnh 611-618 có thể được tạo mã bằng cách tham chiếu tới các ảnh khác 611-618 trong cùng lớp N 631 hoặc N+1 632. Ảnh tạo mã tham chiếu tới ảnh khác trong cùng lớp dẫn đến dự đoán liên 623. Dự đoán liên 623 được mô tả bằng các mũi tên liên nét. Ví dụ, ảnh 613 có thể được tạo mã bằng cách dùng dự đoán liên 623 sử dụng một hoặc hai ảnh 611, 612, và/hoặc 614 trong lớp N+1 632 như tham chiếu, trong đó một ảnh được tham chiếu cho dự đoán liên một chiều và/hoặc hai ảnh được tham chiếu cho dự đoán liên hai chiều. Hơn nữa, ảnh 617 có thể được tạo mã bằng cách dùng dự đoán liên 623 sử dụng một hoặc hai ảnh 615, 616, và/hoặc 618 trong lớp N 631 như tham chiếu, trong đó một ảnh được tham chiếu cho dự đoán liên một chiều và/hoặc hai ảnh được tham chiếu cho dự đoán liên hai chiều. Khi ảnh được sử dụng như tham chiếu cho ảnh khác trong cùng lớp khi thực hiện dự đoán liên 623, ảnh có thể được gọi là ảnh tham chiếu. Ví dụ, ảnh 612 có thể là ảnh tham chiếu được sử dụng để tạo mã ảnh 613 theo dự đoán liên 623. Dự đoán liên 623 có thể cũng được tham chiếu như dự đoán lớp nội bộ trong ngữ cảnh nhiều lớp. Như vậy, dự đoán liên 623 là cơ chế của các mẫu tạo mã của ảnh hiện tại bằng cách tham chiếu tới các mẫu được chỉ báo trong ảnh tham chiếu mà khác ảnh hiện tại trong đó ảnh tham chiếu và ảnh hiện tại trong cùng lớp.

Các ảnh 611-618 có thể cũng được tạo mã bằng cách tham chiếu tới các ảnh khác 611-618 trong các lớp khác. Quy trình này được biết đến như là dự đoán liên lớp 621, và được mô tả bởi các mũi tên nét đứt. Dự đoán liên lớp 621 là cơ chế của các mẫu tạo mã của ảnh hiện tại bằng cách tham chiếu tới các mẫu được chỉ báo trong ảnh tham chiếu trong đó ảnh hiện tại và ảnh tham chiếu trong các lớp khác và do đó có các lớp ID khác. Ví dụ, ảnh trong lớp thấp hơn N 631 có thể được sử dụng như ảnh tham chiếu để tạo mã ảnh tương ứng tại lớp cao hơn N+1 632. Như ví dụ cụ thể, ảnh 611 có thể được tạo mã bằng cách tham chiếu tới ảnh 615 theo dự đoán liên lớp 621. Trong trường hợp này, ảnh 615 được sử dụng như ảnh tham chiếu lớp liên. Ảnh tham chiếu lớp liên là ảnh tham chiếu được sử dụng để dự đoán liên lớp 621. Trong hầu hết các trường hợp, dự đoán liên lớp 621 bị hạn chế sao cho ảnh hiện tại, như là ảnh 611, có thể chỉ dùng (các) ảnh tham chiếu lớp liên mà được gồm trong cùng AU và mà tại lớp thấp hơn, như là ảnh 615. Khi đa lớp (ví dụ, nhiều hơn 2) có sẵn, dự đoán liên

lớp 621 có thể mã hóa/giải mã ảnh hiện tại được dựa trên nhiều ảnh tham chiếu lớp liên ở mức độ thấp hơn so với ảnh hiện tại.

Bộ mã hóa video có thể dùng chuỗi video nhiều lớp 600 để mã hóa các ảnh 611-618 qua nhiều sự kết hợp khác nhau và/hoặc các hoán vị của dự đoán liên 623 và dự đoán liên lớp 621. Ví dụ, ảnh 615 có thể được tạo mã theo dự đoán nội bộ. Các ảnh 616-618 có thể sau đó được tạo mã theo dự đoán liên 623 bằng cách sử dụng ảnh 615 như ảnh tham chiếu. Hơn nữa, ảnh 611 có thể được tạo mã theo dự đoán liên lớp 621 bằng cách sử dụng ảnh 615 như ảnh tham chiếu lớp liên. Các ảnh 612-614 có thể sau đó được tạo mã theo dự đoán liên 623 bằng cách sử dụng ảnh 611 như ảnh tham chiếu. Như vậy, ảnh tham chiếu có thể phục vụ cả hai ảnh tham chiếu một lớp và ảnh tham chiếu lớp liên cho các cơ chế tạo mã khác. Bằng cách tạo mã các ảnh lớp cao hơn $N+1$ 632 được dựa trên các ảnh lớp thấp hơn N 631, lớp cao hơn $N+1$ 632 có thể tránh dùng dự đoán nội bộ, mà có hiệu quả tạo mã thấp hơn nhiều so với dự đoán liên 623 và dự đoán liên lớp 621. Như vậy, hiệu quả tạo mã thấp của dự đoán nội bộ có thể bị giới hạn thành các ảnh chất lượng thấp nhất/nhỏ nhất, và do đó được giới hạn để tạo mã lượng dữ liệu video nhỏ nhất. Các ảnh được sử dụng như các ảnh tham chiếu và/hoặc các ảnh tham chiếu lớp liên có thể được chỉ báo trong các mục của (các) danh sách ảnh tham chiếu được chứa trong cấu trúc danh sách ảnh tham chiếu.

Thứ tự để thực hiện các hoạt động này, các lớp như là lớp N 631 và lớp $N+1$ 632 có thể được gồm trong một hoặc nhiều OLS 625 và 626. Cụ thể, các ảnh 611-618 được mã hóa dưới dạng các lớp 631-632 trong dòng bit 600, và sau đó mỗi lớp 631-632 của các ảnh được gán cho một hoặc nhiều OLS 625 và 626. OLS 625 và/hoặc 626 có thể sau đó được lựa chọn và các lớp tương ứng 631 và/hoặc 632 có thể được truyền tới bộ giải mã, phụ thuộc vào các dung lượng tại bộ giải mã và/hoặc các điều kiện mạng. OLS 625 là tập các lớp trong đó một hoặc nhiều lớp được chỉ định là các lớp đầu ra. Lớp đầu ra là lớp mà được chỉ định cho đầu ra (ví dụ, tới bộ hiển thị). Ví dụ, lớp N 631 có thể chỉ được bao gồm để hỗ trợ dự đoán liên lớp 621 và có thể không bao giờ được đưa ra. Trong trường hợp này, lớp $N+1$ 632 được giải mã được dựa trên lớp N 631 và được đưa ra. Trong trường hợp này, OLS 625 gồm lớp $N+1$ 632 như lớp đầu ra. Khi OLS chứa chỉ lớp đầu ra, OLS được gọi là OLS thứ 0 626. LS thứ 0 626 là OLS mà chỉ chứa lớp thấp nhất (lớp với bộ nhận dạng lớp thấp nhất) và do đó chỉ chứa lớp đầu ra. Trong trường hợp khác, OLS 625 có thể chứa nhiều lớp trong các kết hợp khác nhau. Ví dụ, lớp đầu ra trong OLS 625 có thể được tạo mã theo dự đoán liên lớp 621 được dựa trên một, hai, hoặc nhiều lớp thấp hơn. Hơn nữa, OLS 625 có thể chứa nhiều hơn một lớp đầu ra. Do đó, OLS 625 có thể chứa một hoặc nhiều lớp đầu ra và các lớp hỗ trợ bất kỳ cần để tái cấu trúc các lớp đầu ra. Trong khi chỉ hai OLS 625 và 626 được minh họa, chuỗi

video nhiều lớp 600 có thể được tạo mã bằng cách dùng nhiều OLS khác 625 và/hoặc 626 mỗi loại sử dụng các kết hợp khác nhau của các lớp. Các OLS 625 và 626 là mỗi loại được kết hợp với chỉ số OLS 629, là chỉ số mà nhận dạng duy nhất OLS tương ứng 625 và 626.

Việc kiểm tra chuỗi video nhiều lớp 600 đối với sự phù hợp các tiêu chuẩn ở HRD 500 có thể trở nên phức tạp phụ thuộc vào số lượng của các lớp 631-632 và các OLS 625 và 626. HRD 500 có thể tách chuỗi video nhiều lớp 600 vào trong chuỗi các điểm hoạt động 627 để thử nghiệm. OLS 625 và/hoặc 626 được nhận dạng bởi chỉ số OLS 629. Điểm hoạt động 627 là tập phụ theo thời gian của OLS 625/626. Điểm hoạt động 627 có thể được nhận dạng bởi cả hai chỉ số OLS 629 của OLS tương ứng 625/626 cũng như ID theo thời gian cao nhất 622. Như ví dụ cụ thể, điểm hoạt động thứ nhất 627 có thể gồm tất cả ảnh trong OLS thứ nhất 625 từ ID theo thời gian 0 tới ID theo thời gian 200, điểm hoạt động thứ hai 627 có thể gồm tất cả các ảnh trong OLS thứ nhất 625 từ ID theo thời gian 200 và 1 tới ID theo thời gian 400, v.v.. Trong trường hợp này, điểm hoạt động thứ nhất 627 được mô tả bởi chỉ số OLS 629 của OLS thứ nhất 625 và ID theo thời gian 200. Hơn nữa, điểm hoạt động thứ hai 627 được mô tả bởi chỉ số OLS 629 của OLS thứ nhất 625 và ID theo thời gian 400. Điểm hoạt động 627 lựa chọn để thử nghiệm tại thời điểm xác định được gọi là OP đang thử nghiệm (targetOp). Do đó, targetOp là điểm hoạt động 627 mà được lựa chọn để thử nghiệm sự phù hợp tại HRD 500.

FIG. 7 là sơ đồ khối minh họa chuỗi video nhiều lớp ví dụ 700 được tạo cấu hình để mở rộng về thời gian. Chuỗi video nhiều lớp 700 có thể được mã hóa bởi bộ mã hóa, như là hệ thống codec 200 và/hoặc bộ mã hóa 300 và được giải mã bởi bộ giải mã, như là hệ thống codec 200 và/hoặc bộ giải mã 400, ví dụ theo phương pháp 100. Hơn nữa, chuỗi video nhiều lớp 700 có thể được kiểm tra cho sự phù hợp tiêu chuẩn bởi HRD, như là HRD 500. Chuỗi video nhiều lớp 700 được gồm để mô tả ứng dụng ví dụ khác cho các lớp trong chuỗi video được tạo mã. Ví dụ, chuỗi video nhiều lớp 700 có thể được dùng như phương án riêng hoặc có thể được kết hợp với các kỹ thuật được mô tả đối với chuỗi video nhiều lớp 600.

Chuỗi video nhiều lớp 700 gồm các lớp phụ 710, 720, và 730. Lớp phụ là lớp có thể mở rộng theo thời gian của dòng bit có thể mở rộng theo thời gian mà gồm các đơn vị VCL NAL (ví dụ, các ảnh) với giá trị bộ nhận dạng theo thời gian cụ thể cũng như các đơn vị NAL không VCL được liên kết (ví dụ, các tham số hỗ trợ). Ví dụ, lớp, như là lớp N 631 và/hoặc lớp N+1 632, có thể còn được chia vào trong các lớp phụ 710, 720, và 730 để hỗ trợ mở rộng về thời gian. Lớp phụ 710 có thể được gọi là lớp gốc và các lớp phụ 720 và 730 có thể được gọi là các lớp nâng cao. Như được minh họa, lớp phụ 710 gồm các ảnh 711 ở tốc độ khung thứ nhất, như là 30 khung trên giây. Lớp phụ 710 là lớp gốc do lớp phụ 710 gồm tốc độ khung

gốc/thấp nhất. Lớp phụ 720 chứa các ảnh 721 mà phần bù về thời gian từ các ảnh 711 của lớp phụ 710. Kết quả là lớp phụ 710 và lớp phụ 720 có thể được kết hợp, điều này dẫn đến trong tốc độ khung chung cao hơn tốc độ khung của riêng lớp phụ 710. Ví dụ, lớp phụ 710 và 720 có thể có tốc độ khung được kết hợp 60 khung trên giây. Theo đó, lớp phụ 720 nâng cao tốc độ khung của lớp phụ 710. Hơn nữa, lớp phụ 730 chứa các ảnh 731 mà phần bù về thời gian từ các ảnh 721 và 711 của các lớp phụ 720 và 710. Như vậy, lớp phụ 730 có thể được kết hợp với các lớp phụ 720 và 710 còn để nâng cao lớp phụ 710. Ví dụ, các lớp phụ 710, 720, và 730 có thể có tốc độ khung được kết hợp 90 khung trên giây.

Sự biểu diễn lớp phụ 740 có thể được tạo động bằng cách kết hợp các lớp phụ 710, 720, và/hoặc 730. Sự biểu diễn lớp phụ 740 là tập phụ của dòng bit chứa các đơn vị NAL của lớp phụ cụ thể và các lớp phụ thấp hơn. Trong ví dụ minh họa, sự biểu diễn lớp phụ 740 chứa các ảnh 741, mà các ảnh được kết hợp 711, 721, và 731 của các lớp phụ 710, 720, và 730. Theo đó, chuỗi video nhiều lớp 700 có thể được chia tỷ lệ theo thời gian thành tốc độ khung được mong muốn bằng cách lựa chọn sự biểu diễn lớp phụ 740 mà gồm tập các lớp phụ được mong muốn 710, 720, và/hoặc 730. Sự biểu diễn lớp phụ 740 có thể được tạo bằng cách dùng OLS mà gồm lớp phụ 710, 720, và/hoặc 730 như các lớp. Trong trường hợp này, sự biểu diễn lớp phụ 740 được lựa chọn như là lớp đầu ra. Như vậy, mở rộng về thời gian là một trong một số cơ chế mà có thể được thực hiện bằng cách sử dụng cơ chế nhiều lớp.

FIG. 8 là sơ đồ khối minh họa dòng bit ví dụ 800. Ví dụ, dòng bit 800 có thể được tạo bởi hệ thống codec 200 và/hoặc bộ mã hóa 300 để giải mã bởi hệ thống codec 200 và/hoặc bộ giải mã 400 theo phương pháp 100. Hơn nữa, dòng bit 800 có thể gồm chuỗi video nhiều lớp 600 và/hoặc 700. Hơn nữa, dòng bit 800 có thể gồm các tham số khác nhau để điều khiển hoạt động của HRD, như là HRD 500. được dựa trên các tham số này, HRD có thể kiểm tra dòng bit 800 về sự phù hợp với các tiêu chuẩn trước khi truyền tới bộ giải mã để giải mã.

Dòng bit 800 gồm VPS 811, một hoặc nhiều SPS 813, nhiều tập tham số ảnh (PPS, picture parameter set) 815, nhiều tiêu đề lát 817, dữ liệu hình ảnh 820, và các thông báo SEI 819. VPS 811 chứa dữ liệu liên quan tới toàn bộ dòng bit 800. Ví dụ, VPS 811 có thể chứa các OLS liên quan dữ liệu, các lớp, và/hoặc các lớp phụ được sử dụng trong dòng bit 800. SPS 813 chứa dữ liệu chuỗi chung cho tất cả các ảnh trong chuỗi video được tạo mã được chứa trong dòng bit 800. Ví dụ, mỗi lớp có thể chứa một hoặc nhiều chuỗi video được tạo mã, và mỗi chuỗi video được tạo mã có thể tham chiếu SPS 813 cho các tham số tương ứng. Các tham số trong SPS 813 có thể gồm kích thước ảnh, độ sâu bit, các tham số công cụ tạo mã, hạn chế tốc độ bit, v.v. Cần lưu ý rằng, trong khi mỗi chuỗi đề cập đến SPS 813, một SPS 813 có thể chứa dữ liệu cho nhiều chuỗi trong một số ví dụ. PPS 815 chứa các tham số

mà áp dụng toàn bộ ảnh. Do đó, mỗi ảnh trong chuỗi video có thể tham chiếu tới PPS 815. Cần lưu ý rằng, trong khi mỗi ảnh đề cập đến PPS 815, một PPS 815 có thể chứa dữ liệu cho nhiều ảnh trong một số ví dụ. Ví dụ, tương tự nhiều ảnh có thể được tạo mã theo các tham số tương tự. Trong trường hợp này, một PPS 815 có thể chứa dữ liệu cho các ảnh tương tự này. PPS 815 có thể chỉ báo các công cụ tạo mã sẵn có cho các lát trong các ảnh tương ứng, các tham số lượng tử hóa, các phân bù, v.v..

Tiêu đề lát 817 chứa các tham số mà cụ thể cho từng lát trong ảnh. Do đó, có thể có 1 tiêu đề lát 817 trên lát trong chuỗi video. Tiêu đề lát 817 có thể chứa thông tin loại lát, POCs, các danh sách ảnh tham chiếu, các trọng số dự đoán, các điểm nhập ô, các tham số giải khối, v.v. Cần lưu ý rằng trong một số ví dụ, dòng bit 800 có thể cũng gồm tiêu đề ảnh, mà là cấu trúc cú pháp mà chứa các tham số mà áp dụng tất các lát trong một ảnh. Vì lý do này, tiêu đề ảnh và tiêu đề lát 817 có thể được sử dụng thay thế cho nhau trong một số ngữ cảnh. Ví dụ, các tham số nhất định có thể được di chuyển giữa tiêu đề lát 817 và tiêu đề ảnh phụ thuộc vào xem liệu có các tham số này chung cho tất cả các lát trong ảnh hay không.

Dữ liệu hình ảnh 820 chứa dữ liệu video được mã hóa theo dự đoán liên và/hoặc dự đoán nội bộ cũng như dữ liệu dư được biến đổi và được lượng tử hóa tương ứng. Ví dụ, dữ liệu hình ảnh 820 có thể gồm các AU 821, các DU 822, và/hoặc các ảnh 823. AU 821 là tập đơn vị NAL mà được kết hợp với với nhau theo nguyên tắc phân loại cụ thể và thuộc về một thời gian đầu ra cụ thể. DU 822 là AU hoặc tập con của AU và các đơn vị NAL không VCL được liên kết. Ảnh 823 là mảng các mẫu độ chói và/hoặc mảng các mẫu sắc độ mà tạo khung hoặc trường của chúng. Bằng ngôn ngữ đơn giản, AU 821 chứa dữ liệu video khác nhau mà có thể được hiển thị tại thời điểm xác định trong chuỗi video cũng như hỗ trợ dữ liệu cú pháp. Do đó, AU 821 có thể chứa một ảnh 823 trong dòng bit một lớp hoặc nhiều ảnh từ đa lớp mà tất cả được kết hợp với tức thì trong dòng bit đa lớp. Trong khi đó, ảnh 823 là hình ảnh được tạo mã mà có thể đưa ra để hiển thị hoặc được sử dụng để hỗ trợ tạo mã (các) ảnh khác 823 để đưa ra. DU 822 có thể chứa một hoặc nhiều ảnh 823 và hỗ trợ dữ liệu cú pháp cần để giải mã. Ví dụ, DU 822 và AU 821 có thể được sử dụng thay thế cho nhau trong các dòng bit đơn giản (ví dụ, khi AU chứa một ảnh). Tuy nhiên, trong các dòng bit đa lớp phức tạp hơn, DU 822 có thể chỉ chứa một phần của dữ liệu video từ AU 821. Ví dụ, AU 821 có thể chứa các ảnh 823 tại một vài các lớp và/hoặc các lớp phụ trong đó một số các ảnh 823 được kết hợp với các OLS khác. Trong trường hợp này, DU 822 có thể chỉ chứa (các) ảnh 823 từ OLS được chỉ định và/hoặc lớp/lớp phụ được chỉ định.

Ảnh 823 chứa một hoặc nhiều lát 825. Lát 825 có thể được định nghĩa là số nguyên các ô hoàn chỉnh hoặc số nguyên của các hàng đơn vị cây tạo mã hoàn chỉnh liên tiếp (coding

tree unit, CTU) (ví dụ, trong ô) của ảnh 823, trong đó các ô hoặc các hàng CTU được chứa riêng trong một đơn vị NAL 829. Do đó, lát 825 cũng được chứa trong một đơn vị NAL 829. Các lát 825 còn được chia vào trong các CTU và/hoặc các khối cây tạo mã (CTB). CTU là nhóm của các mẫu kích thước được định trước mà có thể được phân vùng bởi cây tạo mã. CTB là tập phụ của CTU và chứa các thành phần độ chói hoặc các thành phần sắc độ của CTU. Các CTU/các CTB còn được chia vào trong các khối tạo mã được dựa trên các cây tạo mã. Các khối tạo mã có thể sau đó được mã hóa/được giải mã theo các cơ chế dự đoán.

Dòng bit 800 là chuỗi các đơn vị NAL 829. Đơn vị NAL 829 là vùng chứa dữ liệu video và/hoặc cú pháp hỗ trợ. Đơn vị NAL 829 có thể là đơn vị VCL NAL hoặc đơn vị NAL không VCL. Đơn vị VCL NAL là đơn vị NAL 829 được tạo mã để chứa dữ liệu video, như là lát được tạo mã 825 và tiêu đề lát được kết hợp 817. Đơn vị NAL không VCL là đơn vị NAL 829 mà chứa dữ liệu không phải video như là cú pháp và/hoặc các tham số mà hỗ trợ giải mã dữ liệu video, thực hiện kiểm tra sự phù hợp, hoặc các hoạt động khác. Ví dụ, đơn vị NAL không VCL có thể chứa VPS 811, SPS 813, PPS 815, thông báo SEI 819, hoặc cú pháp hỗ trợ khác.

Thông báo SEI 819 là cấu trúc cú pháp với các ngữ nghĩa cụ thể mà truyền tải thông tin mà không cần bởi quy trình giải mã để xác định các giá trị của các mẫu trong các ảnh được giải mã. Ví dụ, các thông báo SEI có thể chứa dữ liệu để hỗ trợ các quy trình HRD hoặc dữ liệu hỗ trợ khác mà không liên quan trực tiếp để giải mã dòng bit 800 tại bộ giải mã. Thông báo SEI 819 có thể gồm các thông báo SEI có thể lồng nhau và/hoặc các thông báo SEI lồng nhau không thể mở rộng. Thông báo SEI có thể lồng nhau có thể mở rộng là thông báo mà chứa nhiều thông báo SEI mà tương ứng với một hoặc nhiều OLS hoặc một hoặc nhiều lớp. Thông báo SEI lồng nhau không thể mở rộng là thông báo mà không được lồng nhau và do chứa một thông báo SEI. Các thông báo SEI 819 có thể gồm thông báo BP SEI mà chứa các tham số HRD để khởi tạo HRD để quản lý CPB. Các thông báo SEI 819 có thể cũng gồm thông báo PT SEI mà chứa các tham số HRD để quản lý thông tin phân phối cho các AU 821 tại CPB và/hoặc DPB. Các thông báo SEI 819 có thể cũng gồm thông báo DUI SEI mà chứa các tham số HRD để quản lý thông tin phân phối cho các DU 822 tại CPB và/hoặc DPB.

Dòng bit 800 gồm số nguyên (i) các tập tham số HRD 833 mức độ chuỗi, mà các phần tử cú pháp khởi tạo và/hoặc xác định các điều kiện hoạt động của HRD, như là HRD 500. Trong một số ví dụ, cấu trúc cú pháp các tham số HRD chung (`general_hrd_parameters`) có thể chứa các tham số HRD 833 mức độ chuỗi mà áp dụng cho tất cả các OLS được chỉ định bởi VPS 811. Các tham số HRD mức độ chuỗi là các tham số HRD mà áp dụng toàn bộ

chuỗi video được tạo mã. Chuỗi video được tạo mã là một hoặc nhiều ảnh 823. Trong ví dụ, bộ mã hóa có thể mã hóa chuỗi video vào trong các lớp. Các lớp có thể gồm các chuỗi video được tạo mã. Bộ mã hóa có thể sau đó mã hóa các tham số HRD 833 mức độ chuỗi vào trong dòng bit để cấu hình chính xác HRD để thực hiện các kiểm tra sự phù hợp trên chuỗi trong các OLS. Các tham số HRD 833 có thể cũng chỉ báo cho bộ giải mã rằng bộ giải mã có khả năng giải mã dòng bit theo lập lịch phân phối. Các tham số HRD 833 mức độ chuỗi có thể được gồm trong VPS 811 và/hoặc SPS 813. Các tham số bổ sung được sử dụng để cấu hình HRD có thể cũng được gồm trong các thông báo SEI 819.

Như được lưu ý ở trên, luồng video có thể gồm nhiều OLS và nhiều lớp, như là OLS 625, lớp N 631, lớp N+1 632, lớp phụ 710, lớp phụ 720, và/hoặc lớp phụ 730. Hơn nữa, một số lớp có thể được gồm trong nhiều OLS. Mỗi lớp và/hoặc lớp phụ của mỗi OLS được kiểm tra về sự phù hợp bởi HRD theo lập lịch phân phối. Mỗi lập lịch phân phối được kết hợp với kích thước CPB khác và độ trễ CPB để giải thích cho các băng thông truyền và khả năng hệ thống khác nhau. Một số các hệ thống tạo mã video cho phép mỗi lớp phụ xác định số lượng bất kỳ của lập lịch phân phối. Điều này có thể dẫn đến lượng lớn tín hiệu để hỗ trợ các kiểm tra sự phù hợp, điều này dẫn đến trong hiệu quả tạo mã giảm đối với dòng bit. Hơn nữa, sự linh hoạt như vậy tạo ra sự phức tạp đáng kể, điều này dẫn đến sự gia tăng đáng kể của bộ xử lý và bộ nhớ việc sử dụng tài nguyên tại bộ mã hóa.

Sáng chế bộc lộ gồm các cơ chế cho báo hiệu HRD đơn giản và tăng hiệu quả tạo mã cho video gồm đa lớp. Cụ thể, tất cả các lớp và/hoặc lớp phụ bị hạn chế để gồm cùng số lượng các lập lịch phân phối CPB. Ví dụ, bộ mã hóa có thể xác định số lượng lớn nhất của các lập lịch phân phối CPB được sử dụng cho một lớp bất kỳ và tập số lượng của các lập lịch phân phối CPB cho tất cả các lớp thành số tối đa. Số lượng lập lịch phân phối có thể sau đó được báo hiệu một lần, ví dụ như một phần của các tham số HRD trong VPS. Điều này tránh sự cần thiết để báo hiệu số lượng lập lịch cho mỗi lớp/lớp phụ. Ngoài ra, tất cả các lớp/lớp phụ trong OLS có thể cũng chia sẻ cùng chỉ số lập lịch phân phối. Những thay đổi này giảm lượng dữ liệu được sử dụng để báo hiệu dữ liệu liên quan tới việc kiểm tra sự phù hợp. Điều này làm giảm kích thước dòng bit, và từ đó làm giảm việc sử dụng bộ xử lý, bộ nhớ, và/hoặc tài nguyên mạng tại cả hai bộ mã hóa và bộ giải mã.

Cụ thể, các tham số HRD 833 trong VPS 811 có thể gồm số đếm HRD CPB, mà có thể được ký hiệu là `hrd_cpb_cnt_minus1` 831. `Hrd_cpb_cnt_minus1` 831 là phần tử cú pháp mà chỉ định số lượng của các lập lịch phân phối CPB được sử dụng trong dòng bit 800. `Minus1` chỉ báo rằng giá trị của `hrd_cpb_cnt_minus1` 831 ít hơn 1 so với giá trị được chỉ báo

và mà cộng 1 cần bổ sung để thu giá trị thật. `Hrd_cpb_cnt_minus1` 831 có thể trong khoảng từ 0 tới 31. Điều này chỉ báo rằng 1 tới 32 lập lịch phân phối CPB có thể được chỉ định đối với dòng bit 800.

Các thông báo SEI 819 có thể chứa dữ liệu mô tả các lập lịch phân phối CPB được chỉ báo trong các tham số HRD 833. Ví dụ, thông báo BP SEI có thể chứa số BP CPB trừ 1 (`bp_cpb_cnt_minus1`) 835, độ trễ xóa CPB ban đầu (`nal_initial_cpb_removal_delay`) 837, và phần bù xóa CPB ban đầu (`nal_initial_cpb_removal_offset`) 839 mà có thể được sử dụng để mô tả các lập lịch phân phối CPB cho các lớp tương ứng. Các lập lịch phân phối CPB mà mỗi cặp trong số chúng `nal_initial_cpb_removal_delay` 837 và `nal_initial_cpb_removal_offset` 839. `Nal_initial_cpb_removal_delay` 837 là độ trễ có thể xóa mặc định được sử dụng khi loại bỏ đơn vị dữ liệu từ CPB của HRD khi kiểm tra ảnh, AU, và/hoặc DU trong dòng bit, OLS, và/hoặc lớp. Do đó, `nal_initial_cpb_removal_delay` 837 mô tả việc định thời của lập lịch phân phối CPB. `Nal_initial_cpb_removal_offset` 839 là phần bù xóa CPB mặc định được kết hợp với mỗi ảnh, AU, và/hoặc DU trong dòng bit, OLS, và/hoặc lớp. Ví dụ, phần bù có thể chỉ báo kích thước của đơn vị dữ liệu được loại bỏ từ CPB của HRD. Do đó, `nal_initial_cpb_removal_offset` 839 mô tả kích thước bộ nhớ cache được sử dụng trong lập lịch phân phối CPB. Như vậy, HRD có thể xác định lập lịch phân phối CPB được dựa trên `nal_initial_cpb_removal_delay` 837 và `nal_initial_cpb_removal_offset` 839. Theo đó, cặp giá trị này được sử dụng để xác định mỗi lập lịch phân phối CPB, như là lập lịch phân phối CPB 561.

`Bp_cpb_cnt_minus1` 835 là phần tử cú pháp mà chỉ định số lượng cặp `nal_initial_cpb_removal_delay` 837 và `nal_initial_cpb_removal_offset` 839 được sử dụng để tương ứng với lớp phụ theo thời gian, như là lớp phụ 710, 720, và/hoặc 730. Do đó, `bp_cpb_cnt_minus1` 835 chỉ định số lượng lập lịch phân phối mà có sẵn cho mỗi lớp/lớp phụ. Giá trị của `bp_cpb_cnt_minus1` 835 có thể đặt bằng giá trị của `hrd_cpb_cnt_minus1` 831. Điều này đảm bảo rằng số lượng lập lịch phân phối đối với mỗi lớp là cùng tổng số lượng các lập lịch phân phối CPB. Theo đó, tất cả các lớp có cùng số lượng các lập lịch phân phối CPB. Đảm bảo rằng tất cả các lớp có cùng số lượng các lập lịch phân phối CPB làm giảm sự phức tạp và/hoặc tín hiệu dư trong dòng bit 800, mà tăng hiệu quả tạo mã. Ngoài ra, cách tiếp cận này làm đơn giản hóa quy trình HRD. Do vậy, việc sử dụng tài nguyên bộ xử lý, bộ nhớ, và/hoặc tín hiệu mạng bị giảm tại cả hai bộ mã hóa và bộ giải mã. Cũng cần lưu ý rằng mỗi lập lịch phân phối CPB cho lớp đích/OLS có thể được tham chiếu bởi chỉ số lập lịch duy nhất (`ScIdx`). `ScIdx` có thể gồm giá trị trong khoảng từ 0 tới giá trị của `hrd_cpb_cnt_minus1` 831. Điều này chỉ báo rằng `ScIdx` có thể thay đổi từ giá trị 0 đối với lập lịch phân phối CPB thứ

nhất thành giá trị của `hrd_cpb_cnt_minus1` đối với lập lịch phân phối CPB cuối cùng. Do mỗi lớp/lớp phụ gồm cùng số lượng các lập lịch phân phối CPB và do `ScIdx` có thể nhận dạng duy nhất lập lịch phân phối CPB bất kỳ, các lớp/các lớp phụ trong OLS có thể cũng chia sẻ cùng tập giá trị `ScIdx`. Điều này làm giảm thêm sự phức tạp của quy trình HRD. Như vậy, cơ chế mô tả đối với dòng bit 800 có thể tăng chức năng của bộ mã hóa và/hoặc bộ giải mã. Hơn nữa, cơ chế mô tả đối với dòng bit 800 có thể hỗ trợ tăng hiệu quả tạo mã và/hoặc hỗ trợ việc giảm tải nguyên truyền thông bộ xử lý, bộ nhớ, và/hoặc mạng tại bộ mã hóa và/hoặc bộ giải mã.

Thông tin trước đó được mô tả trong chi tiết hơn dưới đây. Tạo mã video phân lớp còn được gọi là tạo mã video có thể mở rộng hoặc tạo mã video với khả năng mở rộng. Khả năng mở rộng trong tạo mã video có thể được hỗ trợ bằng cách sử dụng các kỹ thuật tạo mã nhiều lớp. Dòng bit đa lớp gồm lớp gốc (BL) và một hoặc nhiều lớp nâng cao (ELs). Ví dụ về các khả năng mở rộng gồm khả năng mở rộng không gian, khả năng mở rộng chất lượng/tỷ lệ tín hiệu trên nhiễu (SNR), khả năng mở rộng nhiều góc nhìn, tốc độ khung khả năng mở rộng, v.v.. Khi kỹ thuật tạo mã nhiều lớp được sử dụng, ảnh hoặc một phần của chúng có thể được tạo mã mà không sử dụng ảnh tham chiếu (dự đoán nội bộ), có thể được tạo mã bằng cách tham chiếu các ảnh tham chiếu mà trong cùng lớp (dự đoán liên), và/hoặc có thể được tạo mã bằng cách tham chiếu các ảnh tham chiếu mà trong (các) lớp khác (dự đoán liên lớp). Ảnh tham chiếu được sử dụng để dự đoán liên lớp của ảnh hiện tại được gọi là ảnh tham chiếu lớp liên (ILRP). FIG. 6 minh họa ví dụ việc tạo mã nhiều lớp khả năng mở rộng không gian trong đó các ảnh trong các lớp khác có các độ phân giải khác.

Một số họ tạo mã video cung cấp hỗ trợ cho khả năng mở rộng trong (các) đặc trưng riêng biệt từ (các) đặc trưng cho việc tạo mã một lớp. Tạo mã video có thể mở rộng (SVC) là phần có thể mở rộng của việc tạo mã video nâng cao (AVC) mà cung cấp hỗ trợ cho các khả năng mở rộng không gian, thời gian, và chất lượng. Đối với SVC, cờ được báo hiệu trong mỗi khối macro (MB, macroblock) trong các ảnh EL để chỉ báo xem liệu có hay không EL MB được mô tả sử dụng khối được sắp xếp từ lớp thấp hơn. Việc dự đoán từ khối được sắp xếp có thể gồm văn bản, các vectơ chuyển động, và/hoặc các chế độ tạo mã. Các triển khai của SVC có thể không được trực tiếp sử dụng lại các triển khai AVC không được sửa đổi trong thiết kế của chúng. Cú pháp khối macro SVC EL và quy trình giải mã khác với cú pháp và quy trình giải mã AVC.

HEVC có thể mở rộng (SHVC, Scalable HEVC) là phần mở rộng của HEVC mà cung cấp hỗ trợ các khả năng mở rộng không gian và chất lượng. HEVC nhiều góc nhìn (MV-HEVC, Multiview HEVC) là phần mở rộng của HEVC mà cung cấp hỗ trợ khả năng mở

rộng nhiều góc nhìn. 3D HEVC (3D-HEVC) là phần mở rộng của HEVC mà cung cấp hỗ trợ tạo mã video 3D mà nâng cao hơn và hiệu quả hơn MV-HEVC. Mở rộng về thời gian có thể được gồm như một phần không thể thiếu của codec HEVC một lớp. Trong sự mở rộng nhiều lớp của HEVC, các ảnh được giải mã được sử dụng để dự đoán liên lớp chỉ đến từ cùng một AU và được coi là các ảnh tham chiếu dài hạn (LTRPs, long-term reference pictures). Các ảnh này được gán các chỉ số tham chiếu trong (các) danh sách ảnh tham chiếu cùng với các ảnh tham chiếu thời gian khác trong lớp hiện tại. Dự đoán liên lớp (ILP) đạt được tại mức độ đơn vị dự đoán (PU) bằng cách thiết lập giá trị của chỉ số tham chiếu để tham chiếu đến (các) ảnh tham chiếu lớp liên trong (các) danh sách ảnh tham chiếu. Khả năng mở rộng không gian lấy mẫu lại ảnh tham chiếu hoặc một phần của chúng khi ILRP có độ phân giải không gian khác với ảnh hiện tại được tạo mã hoặc giải mã. Ảnh tham chiếu lấy mẫu lại có thể được thực hiện tại mức độ ảnh hoặc mức độ khối tạo mã.

VVC có thể cũng hỗ trợ tạo mã video phân lớp. Dòng bit VVC có thể gồm đa lớp. Các lớp có thể độc lập với nhau. Ví dụ, mỗi lớp có thể được tạo mã mà không sử dụng dự đoán liên lớp. Trong trường hợp này, các lớp còn được gọi là các lớp truyền đồng thời. Trong một số trường hợp, một số các lớp được tạo mã sử dụng ILP. Còn trong VPS có thể chỉ báo xem liệu có hay không các lớp là các lớp truyền đồng thời hoặc xem liệu có một số lớp sử dụng ILP hay không. Khi một số lớp sử dụng ILP, mối quan hệ phụ thuộc lớp giữa các lớp cũng được báo hiệu trong VPS. Không giống SHVC và MV-HEVC, VVC có thể không chỉ định các OLS. OLS gồm tập các lớp được chỉ định, trong đó một hoặc nhiều lớp trong tập các lớp được chỉ định là các lớp đầu ra. Lớp đầu ra là lớp của OLS mà được đưa ra. Trong một số triển khai của VVC, chỉ một lớp có thể được lựa chọn để giải mã và đưa ra khi các lớp là các lớp truyền đồng thời. Trong một số triển khai của VVC, toàn bộ dòng bit gồm tất cả các lớp được chỉ định để được giải mã khi lớp bất kỳ sử dụng ILP. Hơn nữa, các lớp nhất định giữa các lớp được chỉ định là các lớp đầu ra. Các lớp đầu ra có thể được chỉ báo là chỉ lớp cao nhất, tất cả các lớp, hoặc lớp cao nhất cộng tập các lớp thấp hơn được chỉ báo.

Các tiêu chuẩn tạo mã video có thể chỉ định HRD để xác minh sự phù hợp của các dòng bit thông qua các thử nghiệm sự phù hợp HRD được chỉ định. Trong SHVC và MV-HEVC, ba tập các thử nghiệm sự phù hợp của dòng bit được dùng cho kiểm tra sự phù hợp của dòng bit. Dòng bit được gọi là toàn bộ dòng bit và được biểu thị là *entireBitstream*. Tập các thử nghiệm sự phù hợp của dòng bit thứ nhất để thử nghiệm sự phù hợp của toàn bộ dòng bit và các tập phụ theo thời gian tương ứng. Các thử nghiệm này được dùng bất chấp việc có hay không có tập lớp được chỉ định bởi VPS đang hoạt động mà chứa tất cả các giá trị của các đơn vị VCL NAL *nuh_layer_id* được hiện diện trong toàn bộ dòng bit. Theo đó, toàn bộ

dòng bit luôn được kiểm tra về sự phù hợp ngay cả khi một hoặc nhiều lớp không được gồm trong tập đầu ra. Tập các thử nghiệm sự phù hợp của dòng bit thứ hai được dùng để thử nghiệm sự phù hợp của tập các lớp được chỉ định bởi VPS đang hoạt động và tập phụ theo thời gian được liên kết. Đối với tất cả thử nghiệm này, chỉ lớp gốc các ảnh (ví dụ, các ảnh với `nuh_layer_id` bằng 0) được giải mã và đưa ra. Các ảnh khác bị bỏ qua bởi bộ giải mã khi quy trình giải mã được gọi. Tập các thử nghiệm sự phù hợp của dòng bit thứ ba được dùng để thử nghiệm sự phù hợp của các OLS được xác định bởi phần mở rộng VPS của VPS đang hoạt động và tập phụ theo thời gian được liên kết được dựa trên sự phân vùng của OLS và dòng bit. Sự phân vùng dòng bit gồm một hoặc nhiều lớp của OLS của dòng bit đa lớp.

Các khía cạnh trước chứa một số vấn đề nhất định. Ví dụ, hai tập các thử nghiệm sự phù hợp thứ nhất có thể được áp dụng cho các lớp mà không được giải mã và không đưa ra. Ví dụ, các lớp khác với lớp thấp nhất có thể không được giải mã và có thể không được đưa ra. Trong các ứng dụng thực tế, bộ giải mã có thể thu chỉ dữ liệu được giải mã. Như vậy, việc dùng hai tập các thử nghiệm sự phù hợp thứ nhất đều làm phức tạp thiết kế codec và có thể lãng phí các bit để mang cả hai các tham số mức độ chuỗi và mức độ ảnh được sử dụng để hỗ trợ các thử nghiệm sự phù hợp. Tập các thử nghiệm sự phù hợp thứ ba liên quan đến sự phân vùng dòng bit. Các phân vùng này có thể liên quan tới một hoặc nhiều lớp của OLS của dòng bit đa lớp. HRD có thể được đơn giản hóa rất nhiều nếu các thử nghiệm sự phù hợp luôn hoạt động riêng đối với mỗi lớp.

Việc báo hiệu của các tham số HRD mức độ chuỗi có thể bị phức tạp hóa. Ví dụ, các tham số HRD mức độ chuỗi có thể được báo hiệu trong nhiều nơi như là cả trong SPS và VPS. Hơn nữa, các tham số HRD mức độ chuỗi báo hiệu có thể gồm phần dư. Ví dụ, thông tin mà có thể giống nhau cho toàn bộ dòng bit có thể được lặp lại tại mỗi lớp của mỗi OLS. Hơn nữa, ví dụ sơ đồ HRD cho phép lập lịch phân phối khác được lựa chọn cho mỗi lớp. Các lập lịch phân phối này có thể được lựa chọn từ danh sách các lập lịch được báo hiệu đối với mỗi lớp đối với mỗi điểm hoạt động trong đó điểm hoạt động là OLS hoặc tập phụ theo thời gian của OLS. Hệ thống này phức tạp. Hơn nữa, ví dụ sơ đồ HRD cho phép các AU hoàn chỉnh được kết hợp với khoảng đệm các thông báo SEI. AU hoàn chỉnh là AU mà không có các ảnh cho tất cả các lớp được hiện diện trong CVS. Tuy nhiên, khởi tạo HRD tại một AU như vậy có thể có vấn đề. Ví dụ, HRD có thể không được khởi tạo đúng cách cho các lớp với các đơn vị truy cập lớp mà không được hiện diện trong AU không hoàn chỉnh. Hơn nữa, quy trình phân kênh để lấy dòng bit của lớp có thể không loại bỏ đủ và hiệu quả các thông báo SEI được lồng nhau mà không áp dụng cho lớp đích. Lớp dòng bit xảy ra khi sự phân vùng dòng bit chứa chỉ một lớp. Hơn nữa, OLS có thể áp dụng của khoảng đệm lồng nhau không

thể mở rộng, định thời ảnh, và các thông báo SEI thông tin đơn vị giải mã có thể được chỉ định cho toàn bộ dòng bit. Tuy nhiên, thay vào đó, khoảng đệm lồng nhau không thể mở rộng cần được áp dụng cho OLS thứ 0.

Hơn nữa, một số triển khai VVC có thể không suy ra các tham số HDR khi `sub_layer_cpb_params_present_flag` bằng 0. Tham chiếu như vậy có thể cho phép các hoạt động HRD thích hợp. Hơn nữa, các giá trị của `bp_max_sub_layers_minus1` và `pt_max_sub_layers_minus1` có thể được yêu cầu bằng giá trị của `sps_max_sub_layers_minus1`. Tuy nhiên, khoảng đệm và định thời ảnh các thông báo SEI có thể được lồng nhau và có thể áp dụng cho nhiều OLS và đa lớp của mỗi trong nhiều OLS. Trong các ngữ cảnh này, các lớp liên quan có thể tham chiếu tới nhiều SPS. Do đó, hệ thống có thể có khó khăn trong việc theo dõi SPS nào là SPS mà tương ứng với mỗi lớp. Do đó, thay vào đó, các giá trị của hai phần tử cú pháp này bị hạn chế được dựa trên giá trị của `vps_max_sub_layers_minus1`. Hơn nữa, do các lớp khác có thể có số lượng lớp phụ khác, các giá trị của hai phần tử cú pháp có thể không luôn bằng giá trị cụ thể trong tất cả khoảng đệm và định thời ảnh các thông báo SEI.

Ngoài ra, nhược điểm sau được kết hợp với thiết kế HDR trong cả hai SHVC/MV-HEVC và VVC. Quy trình trích xuất dòng bit phụ có thể không loại bỏ các đơn vị SEI NAL chứa các thông báo SEI được lồng nhau mà không cần cho OLS đích.

Nói chung, sáng chế này bộc lộ các cách tiếp cận lồng nhau có thể mở rộng của các thông báo SEI cho tập các lớp đầu ra trong các dòng bit video nhiều lớp. Phần mô tả các kỹ thuật được dựa trên VVC. Tuy nhiên, các kỹ thuật cũng áp dụng cho tạo mã video phân lớp được dựa trên các mô tả codec video khác.

Một hoặc nhiều trong số các vấn đề nêu trên có thể được giải quyết như sau. Cụ thể, sáng chế đề xuất các phương pháp cho thiết kế HDR và các khía cạnh liên quan mà cho phép báo hiệu hiệu quả các tham số HRD với các hoạt động HDR đơn giản hơn so với SHVC và MV-HEVC. Mỗi giải pháp được mô tả dưới đây tương ứng với các vấn đề được mô tả ở trên. Ví dụ, thay vì yêu cầu ba tập các thử nghiệm sự phù hợp, sáng chế bộc lộ chỉ có thể sử dụng một tập các thử nghiệm sự phù hợp để thử nghiệm sự phù hợp của các OLS được chỉ định bởi VPS. Hơn nữa, thay vì thiết kế mà được dựa trên các sự phân vùng dòng bit, cơ chế HRD được bộc lộ có thể luôn hoạt động riêng cho mỗi lớp của OLS. Hơn nữa, các tham số HRD mức độ chuỗi là chung cho tất cả các lớp và các lớp phụ của tất cả các OLS có thể được báo hiệu chỉ một lần, ví dụ trong VPS. Hơn nữa, một số lượng lập lịch phân phối có thể được báo hiệu cho tất cả các lớp và các lớp phụ của tất cả các OLS. Cùng chỉ số lập lịch phân phối có thể cũng được áp dụng cho tất cả các lớp trong OLS. Hơn nữa, các AU hoàn chỉnh có thể

không được kết hợp với thông báo SEI khoảng đệm. AU hoàn chỉnh là AU không gồm các ảnh cho tất cả các lớp được hiện diện trong CVS. Điều này đảm bảo rằng HRD có thể luôn được khởi tạo đúng cách cho tất cả các lớp trong OLS. Ngoài ra, cơ chế được tiết lộ để loại bỏ hiệu quả các thông báo SEI được lồng nhau mà không áp dụng cho lớp đích trong OLS. Điều này hỗ trợ quy trình phân kênh để lấy dòng bit của lớp. Hơn nữa, OLS có thể áp dụng khoảng đệm lồng nhau không thể mở rộng, định thời ảnh, và các thông báo SEI thông tin đơn vị giải mã có thể được chỉ định tới OLS thứ 0. Hơn nữa, các tham số HDR có thể được suy ra khi `sub_layer_cpb_params_present_flag` bằng 0, có thể cho phép các hoạt động HRD thích hợp. Các giá trị của `bp_max_sub_layers_minus1` và `pt_max_sub_layers_minus1` có thể được yêu cầu nằm trong khoảng từ 0 tới `vps_max_sub_layers_minus1`. Trong cách này, các tham số này không được yêu cầu là giá trị cụ thể đối với tất cả khoảng đệm và định thời ảnh các thông báo SEI. Ngoài ra, quy trình trích xuất dòng bit phụ có thể loại bỏ Các đơn vị SEI NAL chứa các thông báo SEI được lồng nhau mà không áp dụng cho OLS đích.

Ví dụ việc triển khai các cơ chế trước như sau. Lớp đầu ra là lớp tập lớp đầu ra mà được đưa ra. OLS là tập các lớp gồm tập các lớp được chỉ định, trong đó một hoặc nhiều lớp trong tập các lớp được chỉ định là các lớp đầu ra. Chỉ số lớp OLS là chỉ số, của lớp trong OLS, tới danh sách của các lớp trong OLS. Quy trình trích xuất dòng bit phụ là quy trình được mô tả bởi các đơn vị NAL trong dòng bit mà không phụ thuộc vào tập đích, được xác định bởi chỉ số OLS đích và TemporalId cao nhất đích, được loại bỏ từ dòng bit, với dòng bit phụ đầu ra gồm các đơn vị NAL trong dòng bit mà thuộc về tập đích.

Ví dụ cú pháp tập tham số video như sau.

<code>video_parameter_set_rbsp() {</code>	Bộ mô tả
<code>...</code>	
<code>general_hrd_params_present_flag</code>	u(1)
<code>if(general_hrd_params_present_flag) {</code>	
<code>num_units_in_tick</code>	u(32)
<code>time_scale</code>	u(32)
<code>general_hrd_parameters()</code>	
<code>}</code>	
<code>vps_extension_flag</code>	u(1)

if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbbsp_trailing_bits()	
}	

Ví dụ cú pháp RBSP tập tham số chuỗi như sau.

seq_parameter_set_rbsp() {	Bộ mô tả
sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_reserved_zero_4bits	u(4)
same_nonoutput_level_and_dpb_size_flag	u(1)
profile_tier_level(1, sps_max_sub_layers_minus1)	
if(!same_nonoutput_level_and_dpb_size_flag)	
profile_tier_level(0, sps_max_sub_layers_minus1)	
...	
if(sps_max_sub_layers_minus1 > 0)	
sps_sub_layer_ordering_info_present_flag	u(1)
dpb_parameters(1)	
if(!same_nonoutput_level_and_dpb_size_flag)	
dpb_parameters(0)	
long_term_ref_pics_flag	u(1)
...	
sps_scaling_list_enabled_flag	u(1)
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	

vui_parameters()	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Ví dụ cú pháp các tham số DPB như sau.

dpb_parameters(reorderMaxLatencyPresentFlag) {	Bộ mô tả
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1); i <= sps_max_sub_layers_minus1; i++) {	
sps_max_dec_pic_buffering_minus1[i]	ue(v)
if(reorderMaxLatencyPresentFlag) {	
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase_plus1[i]	ue(v)
}	
}	
}	

Ví dụ cú pháp các tham số HRD chung như sau.

general_hrd_parameters() {	Bộ mô tả
general_nal_hrd_params_present_flag	u(1)
general_vcl_hrd_params_present_flag	u(1)
if(general_nal_hrd_params_present_flag general_vcl_hrd_params_present_flag) {	
decoding_unit_hrd_params_present_flag	u(1)

if(decoding_unit_hrd_params_present_flag) {	
tick_divisor_minus2	u(8)
decoding_unit_cpb_params_in_pic_timing_sei_flag	u(1)
}	
bit_rate_scale	u(4)
cpb_size_scale	u(4)
if(decoding_unit_hrd_params_present_flag)	
cpb_size_du_scale	u(4)
}	
if(vps_max_sub_layers_minus1 > 0)	
sub_layer_cpb_params_present_flag	u(1)
if(TotalNumOls > 1)	
num_layer_hrd_params_minus1	ue(v)
hrd_cpb_cnt_minus1	ue(v)
for(i = 0; i <= num_layer_hrd_params_minus1; i++) {	
if(vps_max_sub_layers_minus1 > 0)	
hrd_max_temporal_id[i]	u(3)
layer_level_hrd_parameters(hrd_max_temporal_id[i])	
}	
if(num_layer_hrd_params_minus1 > 0)	
for(i = 1; i < TotalNumOls; i++)	
for(j = 0; j < NumLayersInOls[i]; j++)	
layer_level_hrd_idx[i][j]	ue(v)
}	

Ví dụ ngữ nghĩa RBSP tập tham số video như sau. Each_layer_is_an_ols_flag được đặt bằng 1 để chỉ định rằng mỗi tập lớp đầu ra chứa chỉ một lớp và mỗi lớp trong dòng bit là tập lớp đầu ra với lớp gồm duy nhất chỉ một lớp đầu ra. Each_layer_is_an_ols_flag đặc đặt bằng 0 để chỉ định rằng tập lớp đầu ra có thể chứa nhiều hơn một lớp. Nếu

vps_max_layers_minus1 bằng 0, giá trị của each_layer_is_an_ols_flag được suy ra bằng 1. Cách khác, khi vps_all_independent_layers_flag bằng 0, giá trị của each_layer_is_an_ols_flag được suy ra bằng 0.

Ols_mode_idc được đặt bằng 0 để chỉ định rằng tổng số các OLS được chỉ định bởi VPS bằng vps_max_layers_minus1 + 1, OLS thứ i gồm các lớp với các chỉ số lớp từ 0 đến i, bao gồm, và đối với mỗi OLS chỉ lớp cao nhất trong OLS được đưa ra. Ols_mode_idc được đặt bằng 1 để chỉ định rằng tổng số các OLS được chỉ định bởi VPS bằng vps_max_layers_minus1 + 1, OLS thứ i gồm các lớp với các chỉ số lớp từ 0 đến i, bao gồm, và đối với mỗi OLS tất cả các lớp trong OLS được đưa ra. Ols_mode_idc được đặt bằng 2 để chỉ định rằng tổng số các OLS được chỉ định bởi VPS được báo hiệu rõ ràng và đối với mỗi OLS lớp cao nhất và tập các lớp thấp hơn được báo hiệu rõ ràng trong OLS được đưa ra. Giá trị của ols_mode_idc sẽ nằm trong khoảng từ 0 đến 2. Giá trị ba của ols_mode_idc được bảo lưu. Khi vps_all_independent_layers_flag bằng 1 và each_layer_is_an_ols_flag bằng 0, giá trị của ols_mode_idc được suy ra bằng 2. Num_output_layer_sets_minus1 cộng 1 chỉ định tổng số các OLS được chỉ định bởi VPS khi ols_mode_idc bằng 2.

Biến TotalNumOlss, chỉ định tổng số các OLS được chỉ định bởi VPS, được suy ra như sau.

```

if( vps_max_layers_minus1 == 0 )
    TotalNumOlss = 1
else if( each_layer_is_an_ols_flag || ols_mode_idc == 0 || ols_mode_idc ==
1 )
    TotalNumOlss = vps_max_layers_minus1 + 1
else if( ols_mode_idc == 2 )
    TotalNumOlss = num_output_layer_sets_minus1 + 1

```

Layer_included_flag[i][j] chỉ định xem liệu có hay không lớp thứ j (lớp với nuh_layer_id bằng vps_layer_id[j]) được gồm trong OLS thứ i khi ols_mode_idc bằng 2. Layer_included_flag[i][j] được đặt bằng 1 để chỉ định rằng lớp thứ j được gồm trong OLS thứ i. Layer_included_flag[i][j] được đặt bằng 0 để chỉ định lớp thứ j không được gồm trong OLS thứ i.

Biến NumLayersInOls[i], chỉ định số lượng của các lớp trong OLS thứ i, và biến LayerIdInOls[i][j], chỉ định nuh_layer_id giá trị của lớp thứ j trong OLS thứ i, được suy ra như sau.

```

    NumLayersInOls[ 0 ] = 1
    LayerIdInOls[ 0 ][ 0 ] = vps_layer_id[ 0 ]
    for( i = 1, i < TotalNumOlss; i++ ) {
        if( each_layer_is_an_ols_flag ) {
            NumLayersInOls[ i ] = 1
            LayerIdInOls[ i ][ 0 ] = vps_layer_id[ i ]
        } else if( ols_mode_idc == 0 || ols_mode_idc == 1 ) {
            NumLayersInOls[ i ] = i + 1
            for( j = 0; j < NumLayersInOls[ i ]; j++ )
                LayerIdInOls[ i ][ j ] = vps_layer_id[ j ]
        } else if( ols_mode_idc == 2 ) {
            for( k = 0, j = 0; k <= vps_max_layers_minus1; k++ )
                if( layer_included_flag[ i ][ k ] )
                    LayerIdInOls[ i ][ j++ ] = vps_layer_id[ k ]
            NumLayersInOls[ i ] = j
        }
    }
}

```

Biến `OlsLayerIdx[ i ][ j ]`, chỉ định chỉ số lớp OLS của lớp với `nuh_layer_id` bằng `LayerIdInOls[ i ][ j ]`, được suy ra như sau.

```

    for( i = 0, i < TotalNumOlss; i++ )
        for j = 0; j < NumLayersInOls[ i ]; j++ )
            OlsLayerIdx[ i ][ LayerIdInOls[ i ][ j ] ] = j

```

Lớp thấp nhất trong mỗi OLS sẽ là lớp độc lập. Cách khác, đối với mỗi i trong khoảng từ 0 tới $TotalNumOlss - 1$, bao gồm, giá trị của `vps_independent_layer_flag[ GeneralLayerIdx[ LayerIdInOls[ i ][ 0 ] ] ]` sẽ bằng 1. Mỗi lớp sẽ gồm ít nhất một OLS được chỉ định bởi VPS. Cách khác, đối với mỗi lớp với giá trị cụ thể của `nuh_layer_id` `nuhLayerId`, bằng 1 của `vps_layer_id[ k ]` đối với k trong khoảng từ 0 tới `vps_max_layers_minus1`, bao gồm, sẽ có ít nhất một cặp các giá trị của i và j , trong đó i nằm trong khoảng từ 0 tới $TotalNumOlss - 1$, bao gồm, và j nằm trong khoảng $NumLayersInOls[i] - 1$, bao gồm, sao cho giá trị của `LayerIdInOls[ i ][ j ]` bằng `nuhLayerId`. Lớp bất kỳ trong OLS sẽ là lớp đầu ra của OLS hoặc lớp tham chiếu (trực tiếp hoặc không trực tiếp) của lớp đầu ra của OLS.

Vps_output_layer_flag[i][j] chỉ định xem liệu có hay không lớp thứ j trong OLS thứ i được đưa ra khi ols_mode_idc bằng 2. Vps_output_layer_flag[i] bằng 1 chỉ định rằng lớp thứ j trong OLS thứ i được đưa ra. Vps_output_layer_flag[i] được đặt bằng 0 để chỉ định rằng lớp thứ j trong OLS thứ i không đưa ra. Khi vps_all_independent_layers_flag bằng 1 và each_layer_is_an_ols_flag bằng 0, giá trị của vps_output_layer_flag[i] được suy ra bằng 1. Biến OutputLayerFlag[i][j], đối với giá trị 1 chỉ định rằng lớp thứ j trong OLS thứ i được đưa ra và giá trị 0 chỉ định rằng lớp thứ j trong OLS thứ i không đưa ra, được suy ra như sau.

```
for( i = 0, i < TotalNumOlss; i++ ) {
    OutputLayerFlag[ i ][ NumLayersInOls[ i ] - 1 ] = 1
    for( j = 0; j < NumLayersInOls[ i ] - 1; j++ )
        if( ols_mode_idc[ i ] == 0 )
            OutputLayerFlag[ i ][ j ] = 0
        else if( ols_mode_idc[ i ] == 1 )
            OutputLayerFlag[ i ][ j ] = 1
        else if( ols_mode_idc[ i ] == 2 )
            OutputLayerFlag[ i ][ j ] = vps_output_layer_flag[ i ][ j ]
    }
```

OLS thứ 0 chỉ chứa lớp thấp nhất (lớp với nuh_layer_id bắt đầu từ vps_layer_id[0]) và cho OLS thứ 0 lớp duy nhất được bao gồm được đưa ra.

Vps_extension_flag được đặt bằng 0 để chỉ định rằng không có các phần tử cú pháp vps_extension_data_flag được hiện diện trong cấu trúc cú pháp VPS RBSP. Vps_extension_flag được đặt bằng 1 để chỉ định rằng có các phần tử cú pháp vps_extension_data_flag được hiện diện trong cấu trúc cú pháp VPS RBSP. Vps_extension_data_flag có thể có giá trị bất kỳ. Sự hiện diện và giá trị của vps_extension_data_flag không ảnh hưởng đến sự phù hợp của bộ giải mã đối với các đặc trưng được chỉ định. Các bộ giải mã sẽ bỏ qua tất cả các phần tử cú pháp vps_extension_data_flag.

Ví dụ về ngữ nghĩa của các tham số DPB như sau. Cấu trúc cú pháp dpb_parameters() cung cấp thông tin kích thước DBP, và, tùy chọn, số lượng sắp xếp lại ảnh tối đa và thông tin về độ trễ tối đa (MRML). Mỗi SPS gồm 1 hoặc các cấu trúc cú pháp dpb_parameters(). Cấu trúc cú pháp dpb_parameters() thứ nhất trong SPS chứa cả hai thông tin kích thước DBP và thông tin MRML. Khi được biểu diễn, cấu trúc cú pháp dpb_parameters() thứ hai trong SPS chỉ chứa thông tin kích thước DBP. Thông tin MRML trong cấu trúc cú pháp

`dpb_parameters()` thứ nhất trong SPS áp dụng cho lớp tham chiếu tới SPS bất chấp việc có hay không lớp là lớp đầu ra trong OLS. Thông tin kích thước DBP trong cấu trúc cú pháp `dpb_parameters()` thứ nhất trong SPS áp dụng cho lớp tham chiếu tới SPS khi lớp là lớp đầu ra của OLS. Thông tin kích thước DBP được gồm trong cấu trúc cú pháp `dpb_parameters()` thứ hai, khi được biểu diễn, trong SPS áp dụng cho lớp tham chiếu tới SPS khi lớp không phải là lớp đầu ra của OLS. Khi SPS chỉ gồm một cấu trúc cú pháp `dpb_parameters()`, thông tin kích thước DBP cho lớp không phải là lớp đầu ra được suy ra giống như đối với lớp như lớp đầu ra.

Ví dụ các ngữ nghĩa các tham số HRD chung như sau. Cấu trúc cú pháp `general_hrd_parameters()` cung cấp các tham số HRD được sử dụng trong các hoạt động HRD. `Sub_layer_cpb_params_present_flag` được đặt bằng 1 để chỉ định rằng cấu trúc cú pháp `layer_level_hrd_parameters()` thứ *i* chứa các tham số HRD cho các biểu diễn lớp phụ với `TemporalId` trong khoảng từ 0 tới `hrd_max_temporal_id[i]`. `Sub_layer_cpb_params_present_flag` được đặt bằng 0 để chỉ định rằng cấu trúc cú pháp `layer_level_hrd_parameters()` thứ *i* chứa các tham số HRD cho biểu diễn lớp phụ với `TemporalId` chỉ bằng `hrd_max_temporal_id[i]`. Khi `vps_max_sub_layers_minus1` bằng 0, giá trị của `sub_layer_cpb_params_present_flag` được suy ra bằng 0. Khi `sub_layer_cpb_params_present_flag` bằng 0, các tham số HRD cho các biểu diễn lớp phụ với `TemporalId` trong khoảng từ 0 tới `hrd_max_temporal_id[i] - 1`, bao gồm, được suy ra giống như vậy cho biểu diễn lớp phụ với `TemporalId` bằng `hrd_max_temporal_id[i]`. Chúng gồm các tham số HRD bắt đầu từ phần tử cú pháp `fixed_pic_rate_general_flag[i]` đến cấu trúc cú pháp `sub_layer_hrd_parameters(i)` ngay lập tức với điều kiện `if(general_vcl_hrd_params_present_flag)` trong cấu trúc cú pháp `layer_level_hrd_parameters`. `Num_layer_hrd_params_minus1` cộng 1 chỉ định số lượng của các cấu trúc cú pháp `layer_level_hrd_parameters()` được hiện diện trong cấu trúc cú pháp `general_hrd_parameters()`. Giá trị của `num_layer_hrd_params_minus1` sẽ nằm trong khoảng từ 0 tới 63. `Hrd_cpb_cnt_minus1` cộng 1 chỉ định số lượng của các mô tả CPB thay thế trong dòng bit của CVS. Giá trị của `hrd_cpb_cnt_minus1` sẽ nằm trong khoảng từ 0 tới 31. `Hrd_max_temporal_id[i]` chỉ định `TemporalId` của sự biểu diễn lớp phụ cao nhất mà các tham số HRD được chứa trong cấu trúc cú pháp `layer_level_hrd_parameters()` thứ *i*. Giá trị của `hrd_max_temporal_id[i]` sẽ nằm trong khoảng từ 0 tới `vps_max_sub_layers_minus1`. Khi `vps_max_sub_layers_minus1` bằng 0, giá trị của `hrd_max_temporal_id[i]` được suy ra bằng 0. `Layer_level_hrd_idx[i][j]` chỉ định chỉ số của cấu trúc cú pháp `layer_level_hrd_parameters()` mà áp dụng cho lớp thứ *j* trong OLS thứ *i*. Giá trị của

`layer_level_hrd_idx[[ i ][ j ]` sẽ nằm trong khoảng từ 0 tới `num_layer_hrd_params_minus1`. Khi không được biểu diễn, giá trị của `layer_level_hrd_idx[[ 0 ][ 0 ]` được suy ra bằng 0.

Ví dụ quy trình trích xuất dòng bit phụ như sau. Đầu vào quy trình này là dòng bit `inBitstream`, chỉ số OLS đích `targetOlsIdx`, và giá trị `TemporalId` cao nhất đích `tIdTarget`. Đầu ra của quy trình này là dòng bit phụ `OutBitstream`. Yêu cầu về sự phù hợp của dòng bit đối với dòng bit đầu vào là dòng bit phụ đầu ra bất kỳ that là đầu ra của quy trình được chỉ định trong điều khoản này với dòng bit, `targetOlsIdx` bằng chỉ số cho danh sách các OLS được chỉ định bởi VPS, và `tIdTarget` bằng giá trị bất kỳ trong khoảng từ 0 tới 6, bao gồm, các đầu vào, và thỏa mãn các điều kiện sau sẽ là dòng bit phù hợp. Dòng bit phụ đầu ra có thể chứa ít nhất một đơn vị VCL NAL với `nuh_layer_id` bằng mỗi trong các giá trị `nuh_layer_id` trong `LayerIdInOls[ targetOlsIdx ]`. Dòng bit phụ đầu ra có thể chứa ít nhất một đơn vị VCL NAL với `TemporalId` bằng `tIdTarget`. Dòng bit phù hợp chứa một hoặc nhiều lát được tạo mã các đơn vị NAL với `TemporalId` bằng 0, nhưng không có các đơn vị NAL được tạo mã để chứa lát với `nuh_layer_id` bằng 0.

Dòng bit phụ đầu ra `OutBitstream` được suy ra như sau. Dòng bit `OutBitstream` được đặt giống với dòng bit `inBitstream`. Việc loại bỏ khỏi `OutBitstream` tất cả các đơn vị NAL với `TemporalId` lớn hơn `tIdTarget`. Loại bỏ từ `OutBitstream` tất cả các đơn vị NAL với `nuh_layer_id` không được gồm trong danh sách `LayerIdInOls[ targetOlsIdx ]`. Việc loại bỏ khỏi `OutBitstream` tất cả các đơn vị SEI NAL mà chứa thông báo SEI lồng nhau có thể mở rộng mà có `nesting_ols_flag` bằng 1 và không có giá trị của `i` trong khoảng từ 0 tới `nesting_num_olss_minus1`, bao gồm, sao cho `NestingOlsIdx[ i ]` bằng `targetOlsIdx`. Khi `targetOlsIdx` lớn hơn 0, loại bỏ từ `OutBitstream` tất cả các đơn vị SEI NAL mà chứa thông báo SEI lồng nhau không thể mở rộng với `payloadType` bằng 0 (khoảng đậm), bằng 1 (định thời ảnh), hoặc bằng 130 (thông tin đơn vị giải mã).

Ví dụ các khía cạnh chung của HRD như sau. Phiên này chỉ định HRD và việc sử dụng của nó để kiểm tra dòng bit và sự phù hợp của bộ giải mã. Tập các thử nghiệm sự phù hợp của dòng bit được dùng cho kiểm tra sự phù hợp của dòng bit, mà được tham chiếu như là toàn bộ dòng bit, được biểu thị là `entireBitstream`. Tập các thử nghiệm sự phù hợp của dòng bit là để thử nghiệm sự phù hợp của mỗi OLS được chỉ định bởi VPS và các tập phụ theo thời gian của mỗi OLS. Đối với mỗi thử nghiệm, các bước thứ tự sau áp dụng cho thứ tự được liệt kê.

Điểm hoạt động đang thử nghiệm, được ký hiệu là `targetOp`, được lựa chọn bằng cách lựa chọn OLS đích với `opOlsIdx` chỉ số OLS và `opTid` giá trị `TemporalId` cao nhất. Giá trị của `opOlsIdx` nằm trong khoảng từ 0 tới `TotalNumOlss - 1`. Giá trị của `opTid` nằm trong

khoảng từ 0 tới `vps_max_sub_layers_minus1`. Các giá trị của `opOlsIdx` và `opTid` sao cho dòng bit phụ `BitstreamToDecode` là đầu ra bằng cách gọi quy trình trích xuất dòng bit phụ với `entireBitstream`, `opOlsIdx`, và `opTid` làm đầu vào thỏa mãn các điều kiện sau. Có ít nhất một đơn vị VCL NAL với `nuh_layer_id` bằng mỗi trong các giá trị `nuh_layer_id` trong `LayerIdInOls[ opOlsIdx ]` trong `BitstreamToDecode`. Có ít nhất một đơn vị VCL NAL với `TemporalId` bằng `opTid` trong `BitstreamToDecode`.

Các giá trị của `TargetOlsIdx` và `Htid` được đặt bằng `opOlsIdx` và `opTid`, tương ứng, của `targetOp`. Giá trị của `ScIdx` được lựa chọn. `ScIdx` được lựa chọn sẽ nằm trong khoảng từ 0 tới `hrd_cpb_cnt_minus1`. Đơn vị truy cập trong `BitstreamToDecode` được kết hợp với khoảng đệm các thông báo SEI (được hiện diện trong `TargetLayerBitstream` hoặc có sẵn thông qua cơ chế bên ngoài không được chỉ định trong bản mô tả này) có thể áp dụng cho `TargetOlsIdx` được lựa chọn như là điểm khởi tạo HRD và được gọi là đơn vị truy cập số 0 đối với mỗi lớp trong OLS đích.

Các bước tiếp theo áp dụng cho mỗi lớp với chỉ số lớp OLS `TargetOlsLayerIdx` trong OLS đích. Nếu có chỉ một lớp trong OLS đích, dòng bit của lớp đang thử nghiệm `TargetLayerBitstream` được đặt giống với `BitstreamToDecode`. Cách khác, `TargetLayerBitstream` được tạo ra bằng cách gọi quy trình phân kênh để lấy dòng bit của lớp với `BitstreamToDecode`, `TargetOlsIdx`, và `TargetOlsLayerIdx` như các đầu vào và đầu ra được gán cho `TargetLayerBitstream`.

Cấu trúc cú pháp `layer_level_hrd_parameters( )` và cấu trúc cú pháp `sub_layer_hrd_parameters( )` có thể áp dụng cho `TargetLayerBitstream` được lựa chọn như sau. Cấu trúc cú pháp `layer_level_hrd_idx[ TargetOlsIdx ][ TargetOlsLayerIdx ]`-th `layer_level_hrd_parameters( )` trong VPS (hoặc được cung cấp thông qua cơ chế bên ngoài như là đầu vào người dùng) được lựa chọn. Trong cấu trúc cú pháp `layer_level_hrd_parameters( )` được lựa chọn, nếu `BitstreamToDecode` là dòng bit loại I, cấu trúc cú pháp `sub_layer_hrd_parameters( Htid )` ngay sau điều kiện nếu (`general_vcl_hrd_params_present_flag`) được lựa chọn và biến `NalHrdModeFlag` được đặt bằng 0. Mặt khác (`BitstreamToDecode` là dòng bit loại II), cấu trúc cú pháp `sub_layer_hrd_parameters( Htid )` mà ngay sau điều kiện nếu (`general_vcl_hrd_params_present_flag`) (trong trường hợp này biến `NalHrdModeFlag` được đặt bằng 0) hoặc điều kiện nếu (`general_nal_hrd_params_present_flag`) (trong trường hợp này biến `NalHrdModeFlag` được đặt bằng 1) được lựa chọn. Khi `BitstreamToDecode` là dòng bit loại II và `NalHrdModeFlag` bằng 0, tất cả các đơn vị NAL không VCL ngoại trừ các đơn vị NAL dữ liệu bộ lọc, và tất cả các phần tử cú pháp `leading_zero_8bits`, `zero_byte`,

start_code_prefix_one_3bytes và trailing_zero_8bits mà tạo luồng byte từ luồng đơn vị NAL, khi được biểu diễn, bị loại bỏ khỏi TargetLayerBitstream và dòng bit còn lại được gán cho TargetLayerBitstream.

Khi decoding_unit_hrd_params_present_flag bằng 1, CPB được lập lịch để hoạt động ở mức độ đơn vị truy cập (trong đó trường hợp biến DecodingUnitHrdFlag được đặt bằng 0) hoặc ở mức độ đơn vị giải mã (trong đó trường hợp biến DecodingUnitHrdFlag được đặt bằng 1). Cách khác, DecodingUnitHrdFlag được đặt bằng 0 và CPB được lập lịch để hoạt động ở mức độ đơn vị truy cập. Đối với mỗi đơn vị truy cập trong TargetLayerBitstream bắt đầu từ đơn vị truy cập số 0, thông báo SEI khoảng đệm (được hiện diện trong TargetLayerBitstream hoặc có sẵn thông qua cơ chế bên ngoài) mà được kết hợp với đơn vị truy cập và áp dụng cho TargetOlsIdx và TargetOlsLayerIdx được lựa chọn, thông báo SEI định thời ảnh (được hiện diện trong TargetLayerBitstream hoặc có sẵn thông qua cơ chế bên ngoài) mà được kết hợp với đơn vị truy cập và áp dụng cho TargetOlsIdx và TargetOlsLayerIdx được lựa chọn, và khi DecodingUnitHrdFlag bằng 1 và decoding_unit_cpb_params_in_pic_timing_sei_flag bằng 0, các thông báo SEI thông tin đơn vị giải mã (được hiện diện trong TargetLayerBitstream hoặc có sẵn thông qua cơ chế bên ngoài) mà được kết hợp với các đơn vị giải mã trong đơn vị truy cập và áp dụng cho TargetOlsIdx và TargetOlsLayerIdx được lựa chọn.

Mỗi lần thử nghiệm sự phù hợp gồm sự kết hợp của một tùy chọn trong mỗi bước trên. Khi có nhiều hơn một tùy chọn đối với bước, đối với thử nghiệm sự phù hợp cụ thể bất kỳ chỉ một tùy chọn được lựa chọn. Tất cả các kết hợp khả thi của tất cả các bước tạo thành toàn bộ tập các thử nghiệm sự phù hợp. Đối với mỗi điểm hoạt động đang thử nghiệm, số lượng của các kiểm tra sự phù hợp của dòng bit được thực hiện bằng $n0 * n1 * n2 * n3$, trong đó các giá trị của $n0$, $n1$, $n2$, và $n3$ được chỉ định là như sau. $n1$ bằng $hrd_cpb_cnt_minus1 + 1$. $n1$ là số lượng các đơn vị truy cập trong BitstreamToDecode mà được kết hợp với khoảng đệm các thông báo SEI. $n2$ được suy ra như sau. Nếu BitstreamToDecode là dòng bit loại I, $n0$ bằng 1. Mặt khác (BitstreamToDecode là dòng bit loại II), $n0$ bằng 2. $n3$ được suy ra như sau. Nếu decoding_unit_hrd_params_present_flag bằng 0, $n3$ bằng 1. Cách khác, $n3$ bằng 2.

HRD chứa bộ tách kênh dòng bit (được biểu diễn tùy chọn), bộ đệm ảnh được tạo mã (coded picture buffer, CPB) cho mỗi lớp, quy trình giải mã tức thời cho mỗi lớp, bộ đệm ảnh được giải mã (DPB) mà chứa DPB phụ cho mỗi lớp, và cắt đầu ra.

Trong ví dụ, HRD hoạt động như sau. HRD được khởi tạo tại đơn vị giải mã số 0, with mỗi CPB và mỗi DPB phụ của DPB được đặt là trống. Sự đầy đủ của DPB phụ cho mỗi

DPB phụ được đặt bằng 0. Sau khi khởi tạo, HRD không được khởi tạo lại sau khoảng đệm các thông báo SEI. Dữ liệu được kết hợp với các đơn vị giải mã mà chảy vào trong mỗi CPB theo lập lịch đến cụ thể được giao bởi HSS. Dữ liệu được kết hợp với mỗi đơn vị giải mã được loại bỏ và được giải mã ngay lập tức bởi quy trình giải mã tức thời tại thời điểm loại bỏ CPB của đơn vị giải mã. Mỗi ảnh được giải mã được đặt trong DPB. Ảnh được giải mã được loại bỏ khỏi DPB khi nó trở nên không cần thiết để tham chiếu dự đoán liên và không cần thiết để đưa ra.

Trong ví dụ, quy trình phân kênh để lấy dòng bit của lớp như sau. Đầu vào quy trình này là dòng bit inBitstream, chỉ số OLS đích targetOlsIdx, và chỉ số lớp OLS đích targetOlsLayerIdx. Đầu ra của quy trình này là dòng bit của lớp OutBitstream. Lớp đầu ra dòng bit OutBitstream được suy ra như sau. Dòng bit OutBitstream được đặt giống với dòng bit inBitstream. Việc loại bỏ khỏi OutBitstream tất cả các đơn vị NAL với nuh_layer_id không bằng LayerIdInOls[targetOlsIdx][targetOlsLayerIdx]. Việc loại bỏ khỏi OutBitstream tất cả các đơn vị SEI NAL mà chứa thông báo SEI lồng nhau có thể mở rộng mà có nesting_ols_flag bằng 1 và không có các giá trị của i và j trong khoảng từ 0 tới nesting_num_olss_minus1, bao gồm, và 0 tới nesting_num_ols_layers_minus1[i], bao gồm, tương ứng, sao cho NestingOlsLayerIdx[i][j] bằng targetOlsLayerIdx. Việc loại bỏ khỏi OutBitstream tất cả các đơn vị SEI NAL mà chứa thông báo SEI lồng nhau có thể mở rộng mà có nesting_ols_flag bằng 1 và có các giá trị của i và j trong khoảng từ 0 tới nesting_num_olss_minus1, bao gồm, và 0 tới nesting_num_ols_layers_minus1[i], bao gồm, tương ứng, sao cho NestingOlsLayerIdx[i][j] ít hơn targetOlsLayerIdx. Việc loại bỏ khỏi OutBitstream tất cả các đơn vị SEI NAL mà chứa thông báo SEI lồng nhau có thể mở rộng mà có nesting_ols_flag bằng 0 và không có giá trị của i trong khoảng từ 0 tới NestingNumLayers – 1, bao gồm, sao cho NestingLayerId[i] bằng LayerIdInOls[targetOlsIdx][targetOlsLayerIdx]. Việc loại bỏ khỏi OutBitstream tất cả các đơn vị SEI NAL mà chứa thông báo SEI lồng nhau có thể mở rộng mà có nesting_ols_flag bằng 0 và có ít nhất một giá trị của i trong khoảng từ 0 tới NestingNumLayers – 1, bao gồm, sao cho NestingLayerId[i] ít hơn LayerIdInOls[targetOlsIdx][targetOlsLayerIdx].

Ví dụ cú pháp thông báo SEI khoảng đệm như sau.

buffering_period(payloadSize) {	Bộ mô tả
...	
bp_max_sub_layers_minus1	u(3)

bp_cpb_cnt_minus1	ue(v)
...	
}	

Ví dụ cú pháp thông báo SEI lồng nhau có thể mở rộng như sau.

scalable_nesting(payloadSize) {	Bộ mô tả
nesting_ols_flag	u(1)
if(nesting_ols_flag) {	
nesting_num_olss_minus1	ue(v)
for(i = 0; i <= nesting_num_olss_minus1; i++) {	
nesting_ols_idx_delta_minus1[i]	ue(v)
if(NumLayersInOls[NestingOlsIdx[i]] > 1) {	
nesting_num_ols_layers_minus1[i]	ue(v)
for(j = 0; j <= nesting_num_ols_layers_minus1[i]; j++)	
nesting_ols_layer_idx_delta_minus1[i][j]	ue(v)
}	
}	
} else {	
nesting_all_layers_flag	u(1)
if(!nesting_all_layers_flag) {	
nesting_num_layers_minus1	ue(v)
for(i = 1; i <= nesting_num_layers_minus1; i++)	
nesting_layer_id[i]	u(6)
}	
}	
nesting_num_seis_minus1	ue(v)
while(!byte_aligned())	
nesting_zero_bit /* equal to 0 */	u(1)

<code>for(i = 0; i <= nesting_num_seis_minus1; i++)</code>	
<code>sei_message()</code>	
<code>}</code>	

Ví dụ ngữ nghĩa tải SEI nói chung như sau. Dưới đây áp dụng trên các lớp có thể áp dụng (trong ngữ cảnh của OLS hoặc ngữ cảnh nói chung) của các thông báo SEI lồng nhau không thể mở rộng. Đối với thông báo SEI lồng nhau không thể mở rộng, khi payloadType bằng 0 (khoảng đệm), bằng 1 (định thời ảnh), hoặc bằng 130 (thông tin đơn vị giải mã), thông báo SEI lồng nhau không thể mở rộng chỉ áp dụng cho lớp thấp nhất trong ngữ cảnh của OLS thứ 0. Đối với thông báo SEI lồng nhau không thể mở rộng, khi payloadType bằng giá trị bất kỳ trong VclAssociatedSeiList, thông báo SEI lồng nhau không thể mở rộng chỉ áp dụng cho lớp mà các đơn vị VCL NAL có nuh_layer_id bằng nuh_layer_id của đơn vị SEI NAL chứa thông báo SEI.

Ví dụ ngữ nghĩa thông báo SEI khoảng đệm như sau. Thông báo SEI khoảng đệm cung cấp độ trễ xóa CPB ban đầu và thông tin bù độ trễ loại bỏ CPB ban đầu để việc khởi tạo của HRD tại vị trí của đơn vị truy cập được liên kết theo thứ tự giải mã. Khi thông báo SEI khoảng đệm được biểu diễn, ảnh nêu trên là ảnh notDiscardablePic khi ảnh có TemporalId bằng 0 và không phải RASL hoặc ảnh hàng đầu có thể giải mã truy cập ngẫu nhiên (random access decodable leading, RADL). Khi ảnh hiện tại không là ảnh thứ nhất trong dòng bit theo thứ tự giải mã, để prevNonDiscardablePic là ảnh có trước theo thứ tự giải mã với TemporalId bằng 0 mà không phải ảnh RASL hoặc RADL.

Sự biểu diễn của khoảng đệm các thông báo SEI cụ thể như sau. Nếu NalHrdBpPresentFlag bằng 1 hoặc VclHrdBpPresentFlag bằng 1, sau đây áp dụng đối với mỗi đơn vị truy cập trong CVS. Nếu đơn vị truy cập là IRAP hoặc đơn vị truy cập làm mới bộ giải mã dần dần (GDR, Gradual Decoder Refresh), thông báo SEI khoảng đệm có thể áp dụng cho điểm hoạt động sẽ được kết hợp với đơn vị truy cập. Cách khác, nếu đơn vị truy cập chứa notDiscardablePic, thông báo SEI khoảng đệm có thể áp dụng cho điểm hoạt động có thể có hoặc có thể không được kết hợp với đơn vị truy cập. Cách khác, đơn vị truy cập sẽ không được kết hợp với thông báo SEI khoảng đệm có thể áp dụng cho điểm hoạt động. Mặt khác (NalHrdBpPresentFlag và VclHrdBpPresentFlag cả hai bằng 0), không có đơn vị truy cập trong CVS sẽ được kết hợp với thông báo SEI khoảng đệm. Đối với một số ứng dụng, sự hiện diện thường xuyên của khoảng đệm các thông báo SEI có thể là mong muốn (ví dụ, để truy cập ngẫu nhiên tại ảnh IRAP hoặc ảnh không phải IRAP hoặc nối dòng bit). Khi ảnh

trong đơn vị truy cập được liên kết với thông báo SEI khoảng đệm, đơn vị truy cập sẽ có ảnh trong mỗi lớp được hiện diện trong CVS, và mỗi ảnh trong đơn vị truy cập sẽ là thông báo SEI khoảng đệm.

Bp_max_sub_layers_minus1 cộng 1 chỉ định số lượng lớn nhất của các lớp phụ theo thời gian cho độ trễ xóa CPB và phần bù có thể xóa CBP được chỉ báo trong thông báo SEI khoảng đệm. Giá trị của bp_max_sub_layers_minus1 sẽ nằm trong khoảng từ 0 tới vps_max_sub_layers_minus1. Bp_cpb_cnt_minus1 cộng 1 chỉ định số lượng của các cặp phần tử cú pháp nal_initial_cpb_removal_delay[i][j] và nal_initial_cpb_removal_offset[i][j] của lớp phụ theo thời gian thứ i khi bp_nal_hrd_params_present_flag bằng 1, và số lượng của các cặp phần tử cú pháp vcl_initial_cpb_removal_delay[i][j] và vcl_initial_cpb_removal_offset[i][j] của lớp phụ theo thời gian thứ i khi bp_vcl_hrd_params_present_flag bằng 1. Giá trị của bp_cpb_cnt_minus1 sẽ nằm trong khoảng từ 0 tới 31. Giá trị của bp_cpb_cnt_minus1 sẽ bằng giá trị của hrd_cpb_cnt_minus1.

Ví dụ ngữ nghĩa thông báo SEI định thời ảnh như sau. Thông báo SEI định thời ảnh cung cấp thông tin độ trễ xóa CPB và độ trễ đầu ra DPB cho đơn vị truy cập được kết hợp với thông báo SEI. Nếu bp_nal_hrd_params_present_flag hoặc bp_vcl_hrd_params_present_flag của thông báo SEI khoảng đệm có thể áp dụng được cho đơn vị truy cập hiện thời bằng 1, biến CpbDpbDelaysPresentFlag được đặt bằng 1. Cách khác, CpbDpbDelaysPresentFlag được đặt bằng 0. Việc biểu diễn định thời ảnh các thông báo SEI được mô tả như sau. Nếu CpbDpbDelaysPresentFlag bằng 1, thông báo SEI định thời ảnh sẽ được kết hợp với đơn vị truy cập hiện tại. Mặt khác (CpbDpbDelaysPresentFlag bằng 0), ở đó sẽ không thông báo SEI định thời ảnh được kết hợp với đơn vị truy cập hiện tại. Cú pháp trong thông báo SEI định thời ảnh TemporalId là TemporalId của đơn vị SEI NAL chứa thông báo SEI định thời ảnh. Pt_max_sub_layers_minus1 cộng 1 chỉ định TemporalId của sự biểu diễn lớp phụ cao nhất để thông tin độ trễ xóa CPB được chứa trong thông báo SEI định thời ảnh. Giá trị của pt_max_sub_layers_minus1 sẽ nằm trong khoảng từ 0 tới vps_max_sub_layers_minus1.

Ví dụ ngữ nghĩa thông báo SEI lồng nhau có thể mở rộng như sau. Thông báo SEI lồng nhau có thể mở rộng cung cấp các cơ chế để liên kết các thông báo SEI với các lớp cụ thể trong ngữ cảnh của các OLS cụ thể hoặc với các lớp cụ thể không trong ngữ cảnh của OLS. Thông báo SEI lồng nhau có thể mở rộng chứa một hoặc nhiều thông báo SEI. Các thông báo SEI được chứa trong thông báo SEI lồng nhau có thể mở rộng còn được gọi là thông báo SEI được lồng nhau có thể mở rộng. Nó là yêu cầu về sự phù hợp của dòng bit

rằng các hạn chế sau áp dụng đối với việc chứa các thông báo SEI trong thông báo SEI lồng nhau có thể mở rộng. Thông báo SEI mà có payloadType bằng 132 (hàm băm ảnh được giải mã) hoặc 133 (lồng nhau có thể mở rộng) sẽ không được chứa trong thông báo SEI lồng nhau có thể mở rộng. Khi thông báo SEI lồng nhau có thể mở rộng chứa khoảng đệm, định thời ảnh, hoặc thông báo SEI thông tin đơn vị giải mã, thông báo SEI lồng nhau có thể mở rộng sẽ không chứa thông báo SEI khác bất kỳ với payloadType không bằng 0 (khoảng đệm), 1 (định thời ảnh), hoặc 130 (thông tin đơn vị giải mã).

Nó là yêu cầu về sự phù hợp của dòng bit mà các hạn chế sau áp dụng trên giá trị của `nal_unit_type` của đơn vị SEI NAL chứa thông báo SEI lồng nhau có thể mở rộng. Khi thông báo SEI lồng nhau có thể mở rộng chứa thông báo SEI mà có payloadType bằng 0 (khoảng đệm), 1 (định thời ảnh), 130 (thông tin đơn vị giải mã), 145 (chỉ báo RAP phụ thuộc), hoặc 168 (thông tin trường khung), đơn vị SEI NAL chứa thông báo SEI lồng nhau có thể mở rộng sẽ có `nal_unit_type` bằng `PREFIX_SEI_NUT`. Khi thông báo SEI lồng nhau có thể mở rộng chứa thông báo SEI mà có payloadType bằng 132 (hàm băm ảnh được giải mã), đơn vị SEI NAL chứa thông báo SEI lồng nhau có thể mở rộng sẽ có `nal_unit_type` bằng `SUFFIX_SEI_NUT`.

`Nesting_ols_flag` được đặt là 1 để chỉ định rằng thông báo SEI được lồng nhau có thể mở rộng áp dụng cho các lớp cụ thể trong ngữ cảnh của các OLS cụ thể. `Nesting_ols_flag` được đặt là 0 để chỉ định rằng thông báo SEI được lồng nhau có thể mở rộng áp dụng chung (không trong ngữ cảnh của OLS) cho các lớp cụ thể. Nó là yêu cầu về sự phù hợp của dòng bit mà các hạn chế sau áp dụng trên giá trị của `nesting_ols_flag`. Khi thông báo SEI lồng nhau có thể mở rộng chứa thông báo SEI mà có payloadType bằng 0 (khoảng đệm), 1 (định thời ảnh), hoặc 130 (thông tin đơn vị giải mã), giá trị của `nesting_ols_flag` sẽ bằng 1. Khi thông báo SEI lồng nhau có thể mở rộng chứa thông báo SEI mà có payloadType bằng giá trị trong `VclAssociatedSeiList`, giá trị của `nesting_ols_flag` sẽ bằng 0. `Nesting_num_olss_minus1` cộng 1 chỉ định số lượng của các OLS mà trong đó thông báo SEI được lồng nhau có thể mở rộng áp dụng. Giá trị của `nesting_num_olss_minus1` sẽ nằm trong khoảng từ 0 tới `TotalNumOlss - 1`. `Nesting_ols_idx_delta_minus1[i]` được sử dụng để suy ra biến `NestingOlsIdx[i]` mà chỉ định chỉ số OLS của OLS thứ *i* mà thông báo SEI được lồng nhau có thể mở rộng áp dụng khi `nesting_ols_flag` bằng 1. Giá trị của `nesting_ols_idx_delta_minus1[i]` sẽ nằm trong khoảng từ 0 tới `TotalNumOlss` trừ 2. Biến `NestingOlsIdx[i]` được suy ra như sau:

$$\begin{aligned}
 & \text{if}(i == 0) \\
 & \text{NestingOlsIdx}[i] = \text{nesting_ols_idx_delta_minus1}[i] \\
 & \text{else} \\
 & \text{NestingOlsIdx}[i] = \text{NestingOlsIdx}[i - 1] + \text{nesting_ols_idx_delta_minus1}[i] \\
 & + 1
 \end{aligned} \tag{D-2}$$

$\text{Nesting_num_ols_layers_minus1}[i]$ cộng 1 chỉ định số lượng của các lớp mà thông báo SEI được lồng nhau có thể mở rộng áp dụng trong ngữ cảnh của $\text{NestingOlsIdx}[i]$ -th OLS. Giá trị của $\text{nesting_num_ols_layers_minus1}[i]$ sẽ nằm trong khoảng từ 0 tới $\text{NumLayersInOls}[\text{NestingOlsIdx}[i]] - 1$. $\text{Nesting_ols_layer_idx_delta_minus1}[i][j]$ được sử dụng để suy ra biến $\text{NestingOlsLayerIdx}[i][j]$ mà chỉ định chỉ số lớp OLS của lớp thứ j tới thông báo SEI được lồng nhau có thể mở rộng áp dụng trong ngữ cảnh của $\text{NestingOlsIdx}[i]$ -th OLS khi nesting_ols_flag bằng 1. Giá trị của $\text{nesting_ols_layer_idx_delta_minus1}[i]$ sẽ nằm trong khoảng từ 0 tới $\text{NumLayersInOls}[\text{nestingOlsIdx}[i]]$ trừ 2. Biến $\text{NestingOlsLayerIdx}[i][j]$ được suy ra như sau:

$$\begin{aligned}
 & \text{if}(j == 0) \\
 & \text{NestingOlsLayerIdx}[i][j] = \text{nesting_ols_layer_idx_delta_minus1}[i][j] \\
 & \text{else} \\
 & \text{NestingOlsLayerIdx}[i][j] = \text{NestingOlsLayerIdx}[i][j - 1] + \\
 & \quad \text{nesting_ols_layer_idx_delta_minus1}[i][j] + 1
 \end{aligned} \tag{D-2}$$

Giá trị thấp nhất giữa tất cả các giá trị của $\text{LayerIdxInOls}[\text{NestingOlsIdx}[i]]$ [$\text{NestingOlsLayerIdx}[i][0]$] cho i trong khoảng từ 0 tới $\text{nesting_num_olss_minus1}$, bao gồm, sẽ bằng nuh_layer_id của đơn vị SEI NAL hiện tại (đơn vị SEI NAL chứa thông báo SEI lồng nhau có thể mở rộng). $\text{Nesting_all_layers_flag}$ được đặt là 1 để chỉ định rằng thông báo SEI được lồng nhau có thể mở rộng áp dụng chung cho tất cả các lớp mà có nuh_layer_id lớn hơn hoặc bằng nuh_layer_id của đơn vị SEI NAL hiện tại. $\text{Nesting_all_layers_flag}$ được đặt là 0 để chỉ định rằng thông báo SEI được lồng nhau có thể mở rộng hoặc có thể không áp dụng chung cho tất cả các lớp mà có nuh_layer_id lớn hơn hoặc bằng nuh_layer_id của đơn vị SEI NAL hiện tại. $\text{Nesting_num_layers_minus1}$ cộng 1 chỉ định số lượng của các lớp mà thông báo SEI được lồng nhau có thể mở rộng áp dụng chung. Giá trị của $\text{nesting_num_layers_minus1}$ sẽ nằm trong khoảng từ 0 tới $\text{vps_max_layers_minus1} - \text{GeneralLayerIdx}[\text{nuh_layer_id}]$, bao gồm, trong đó

nuh_layer_id là nuh_layer_id của đơn vị SEI NAL hiện tại. Nesting_layer_id[i] chỉ định giá trị nuh_layer_id của lớp thứ i mà thông báo SEI được lồng nhau có thể mở rộng áp dụng chung khi nesting_all_layers_flag bằng 0. Giá trị của nesting_layer_id[i] sẽ lớn hơn nuh_layer_id, trong đó nuh_layer_id là nuh_layer_id của đơn vị SEI NAL hiện tại. Khi nesting_ols_flag bằng 0, biến NestingNumLayers, chỉ định số lượng của (các) lớp mà thông báo SEI được lồng nhau có thể mở rộng áp dụng chung, và danh sách NestingLayerId[i] cho i trong khoảng từ 0 tới NestingNumLayers – 1, bao gồm, chỉ định danh sách của giá trị nuh_layer_id của các lớp mà thông báo SEI được lồng nhau có thể mở rộng áp dụng chung, được suy ra như sau, trong đó nuh_layer_id là nuh_layer_id của đơn vị SEI NAL hiện tại.

```

        if( nesting_all_layers_flag ) {
            NestingNumLayers =
vps_max_layers_minus1 + 1 – GeneralLayerIdx[ nuh_layer_id ]
            for( i = 0; i < NestingNumLayers; i ++ )
                NestingLayerId[ i ] = vps_layer_id[ GeneralLayerIdx[ nuh_layer_id ] + i ]
        } else {
(D-2)
            NestingNumLayers = nesting_num_layers_minus1 + 1
            for( i = 0; i < NestingNumLayers; i ++ )
                NestingLayerId[ i ] = ( i == 0 ) ? nuh_layer_id : nesting_layer_id[ i ]
        }

```

Nesting_num_seis_minus1 cộng 1 chỉ định số lượng của thông báo SEI được lồng nhau có thể mở rộng. Giá trị của nesting_num_seis_minus1 sẽ nằm trong khoảng từ 0 tới 63. Nesting_zero_bit sẽ bằng 0.

FIG. 9 là sơ đồ khối của ví dụ thiết bị tạo mã video 900. Thiết bị tạo mã video 900 phù hợp để triển khai các ví dụ/phương án được bộc lộ như được mô tả ở đây. Thiết bị tạo mã video 900 bao gồm các cổng luồng xuống 920, các cổng luồng lên 950, và/hoặc các đơn vị thu phát (Tx/Rx) 910, gồm các bộ phát và/hoặc các bộ thu để truyền thông luồng lên và/hoặc luồng xuống dữ liệu qua mạng. Thiết bị tạo mã video 900 cũng gồm bộ xử lý 930 gồm đơn vị logic và/hoặc đơn vị xử lý trung tâm (CPU, central processing unit) để xử lý dữ liệu và bộ nhớ 932 để lưu trữ dữ liệu. Thiết bị tạo mã video 900 có thể cũng bao gồm các thành phần điện, quang-điện (optical-to-electrical, OE), các thành phần điện-quang (electrical-to-optical, EO), và/hoặc các thành phần truyền thông không dây được ghép với các cổng luồng lên 950 và/hoặc các cổng luồng xuống 920 để truyền thông dữ liệu qua các

mạng truyền thông điện, quang, hoặc không dây. Thiết bị tạo mã video 900 có thể cũng gồm các thiết bị đầu vào và/hoặc đầu ra (I/O) 960 để truyền thông dữ liệu tới và từ người dùng. Các thiết bị I/O 960 có thể gồm các thiết bị đầu ra như là bộ hiển thị để hiển thị dữ liệu video, loa để đưa ra dữ liệu âm thanh, v.v.. Các thiết bị I/O 960 có thể cũng gồm các thiết bị đầu vào, như là bàn phím, chuột, bi xoay, v.v., và/hoặc các giao diện tương ứng để tương tác với các thiết bị đầu ra này.

Bộ xử lý 930 được thực hiện bởi phần cứng và phần mềm. Bộ xử lý 930 có thể được triển khai như một hoặc nhiều chip CPU, lõi (ví dụ, như bộ xử lý đa lõi), mảng cổng lập trình trường (FPGAs, field-programmable gate arrays), các mạch tích hợp ứng dụng cụ thể (ASICs, application specific integrated circuits), và các bộ xử lý tín hiệu số (DSPs, digital signal processors). Bộ xử lý 930 truyền thông với các cổng luồng xuống 920, Tx/Rx 910, các cổng luồng lên 950, và bộ nhớ 932. Bộ xử lý 930 bao gồm môđun tạo mã 914. Môđun tạo mã 914 thực hiện các phương án đã bộc lộ được mô tả ở đây, như là các phương pháp 100, 1000, và 1100, mà có thể dùng chuỗi video nhiều lớp 600, chuỗi video nhiều lớp 700, và/hoặc dòng bit 800. Môđun tạo mã 914 có thể cũng triển khai phương pháp/cơ chế khác bất kỳ được mô tả ở đây. Hơn nữa, môđun tạo mã 914 có thể triển khai hệ thống codec 200, bộ mã hóa 300, bộ giải mã 400, và/hoặc HRD 500. Ví dụ, môđun tạo mã 914 có thể được dùng để triển khai HRD. Hơn nữa, môđun tạo mã 914 có thể được dùng để mã hóa các tham số vào trong dòng bit để hỗ trợ các quy trình kiểm tra sự phù hợp HRD. Theo đó, môđun tạo mã 914 có thể được tạo cấu hình để thực hiện các cơ chế để đánh địa chỉ một hoặc nhiều vấn đề được thảo luận ở trên. Do đó, môđun tạo mã 914 làm cho thiết bị tạo mã video 900 để cung cấp chức năng bổ sung và/hoặc hiệu quả tạo mã khi tạo mã dữ liệu video. Như vậy, môđun tạo mã 914 cải thiện chức năng của thiết bị tạo mã video 900 cũng như giải quyết các vấn đề cụ thể đối với lĩnh vực tạo mã video. Hơn nữa, môđun tạo mã 914 có tác dụng biến đổi thiết bị tạo mã video 900 sang trạng thái khác. Ngoài ra, môđun tạo mã 914 có thể được thực hiện như các lệnh được lưu trữ trong bộ nhớ 932 và được thực thi bởi bộ xử lý 930 (ví dụ, như sản phẩm chương trình máy tính được lưu trữ trên phương tiện không chuyên tiếp).

Bộ nhớ 932 bao gồm một hoặc nhiều loại bộ nhớ chẳng hạn như đĩa, ổ băng, ổ đĩa thể rắn, bộ nhớ chỉ đọc (ROM), bộ nhớ truy cập ngẫu nhiên (RAM), bộ nhớ flash, bộ nhớ định địa chỉ theo nội dung bậc ba (TCAM), bộ nhớ truy cập ngẫu nhiên tĩnh (SRAM), v.v.. Bộ nhớ 932 có thể được sử dụng như thiết bị lưu trữ dữ liệu vượt luồng, để lưu trữ các chương trình khi các chương trình đó được chọn để thực thi và để lưu trữ các lệnh và dữ liệu được đọc trong quá trình thực thi chương trình.

FIG. 10 là lưu đồ phương pháp ví dụ 1000 giải mã chuỗi video vào trong dòng bit, như là dòng bit800, trong đó tất cả các lớp trong dòng bit có cùng cùng số lượng các lập lịch phân phối CPB, như là lập lịch phân phối CPB 561. Phương pháp 1000 có thể được dùng bởi bộ mã hóa, như là hệ thống codec 200, bộ mã hóa 300, và/hoặc thiết bị tạo mã video 900 khi thực hiện phương pháp 100. Hơn nữa, phương pháp 1000 có thể hoạt động trên HRD 500 và do đó có thể thực hiện các thử nghiệm sự phù hợp trên chuỗi video nhiều lớp 600 và/hoặc 700.

Phương pháp 1000 có thể bắt đầu khi bộ mã hóa thu chuỗi video và xác định để mã hóa mà chuỗi video vào trong dòng bit đa lớp, ví dụ được dựa trên đầu vào người dùng. Ở bước 1001, bộ mã hóa mã hóa dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã, như là lớp N 631, lớp N+1 632, lớp phụ 710, lớp phụ 720, và/hoặc lớp phụ 730. Trong một số ví dụ, các lớp được tổ chức vào trong một hoặc nhiều OLS như là OLS 625.

Ở bước 1003, bộ mã hóa mã hóa cấu trúc cú pháp các tham số HRD vào trong dòng bit. Cấu trúc cú pháp các tham số HRD là đối tượng dữ liệu mà chứa một hoặc nhiều tham số HRD, mà có thể được chứa trong các phần tử cú pháp trong cấu trúc cú pháp. Các tham số HRD trong cấu trúc cú pháp HRD được chỉ định rằng tất cả các lớp được kết hợp với cùng số lượng các lập lịch phân phối CPB. Trong ví dụ, số lượng của các lập lịch phân phối CPB được mô tả trong phần tử cú pháp số đếm HRD CPB trừ một (`hrd_cpb_cnt_minus1`). Minus1 chỉ báo rằng giá trị của `hrd_cpb_cnt_minus1` là 1 ít hơn giá trị được chỉ báo và mà cộng 1 cần bổ sung để thu giá trị thật. `Hrd_cpb_cnt_minus1` có thể trong khoảng từ 0 tới 31, bao gồm. Điều này chỉ báo rằng 1 tới 32 lập lịch phân phối CPB có thể được chỉ định và tất cả các lớp được kết hợp với cùng số lượng các lập lịch phân phối CPB. Cấu trúc cú pháp các tham số HRD có thể được mã hóa trong dòng bit trong VPS mà chỉ định các OLS.

Trong một số ví dụ, mỗi lập lịch phân phối tương ứng CPB cho OLS đích có thể được tham chiếu bởi chỉ số lập lịch duy nhất (`ScIdx`). `ScIdx` có thể gồm giá trị trong khoảng từ 0 tới giá trị của `hrd_cpb_cnt_minus1`, bao gồm. Điều này chỉ báo rằng `ScIdx` có thể thay đổi từ giá trị 0 để lập lịch phân phối CPB thứ nhất to giá trị của `hrd_cpb_cnt_minus1` để lập lịch phân phối CPB cuối cùng. Trong một số ví dụ, các lập lịch phân phối CPB được mô tả bởi độ trễ xóa CPB ban đầu và cập phần bù xóa CPB ban đầu. Độ trễ xóa CPB ban đầu (`nal_initial_cpb_removal_delay`) là độ trễ có thể xóa mặc định được sử dụng khi loại bỏ đơn vị dữ liệu từ CPB của HRD khi kiểm tra ảnh, AU, và/hoặc DU trong dòng bit, OLS, và/hoặc lớp. Phần bù xóa CPB ban đầu (`nal_initial_cpb_removal_offset`) là phần bù xóa CPB mặc định được kết hợp với mỗi ảnh, AU, và/hoặc DU trong dòng bit, OLS, và/hoặc lớp. Lớp có thể cũng được kết hợp với số BP CPB minus1 (`bp_cpb_cnt_minus1`) cộng 1 phần tử cú pháp.

Số BP CPB minus1 là phân tử cú pháp mà chỉ định số lượng độ trễ xóa CPB ban đầu và cặp phần bù xóa CPB ban đầu, và do đó số lượng lập lịch phân phối mà có sẵn cho mỗi lớp/lớp phụ. Giá trị của bp_cpb_cnt_minus1 có thể được đặt bằng giá trị của hrd_cpb_cnt_minus1. Điều này đảm bảo rằng số lượng lập lịch phân phối đối với mỗi lớp là giống với tổng số lập lịch phân phối CPB. Theo đó, tất cả các lớp có cùng số lượng các lập lịch phân phối CPB. Các phân tử cú pháp mà mô tả các lập lịch phân phối CPB có thể được gồm trong các thông báo SEI. Ví dụ, nal_initial_cpb_removal_delay, nal_initial_cpb_removal_offset, và bp_cpb_cnt_minus1 có thể được gồm trong thông báo BP SEI.

Ở bước 1005, HRD thu các tham số HRD từ cấu trúc cú pháp các tham số HRD và thực hiện tập các thử nghiệm sự phù hợp của dòng bit trên các lớp được dựa trên các lập lịch phân phối CPB. Ví dụ, HRD có thể thu hrd_cpb_cnt_minus1 để xác định số lượng của các lập lịch phân phối CPB. HRD có thể cũng thu bp_cpb_cnt_minus1 để xác định số lượng của các lập lịch phân phối CPB cho mỗi lớp. HRD có thể cũng thu nal_initial_cpb_removal_delay và nal_initial_cpb_removal_offset để xác định mỗi trong các lập lịch phân phối CPB. HRD có thể sau đó thực hiện các thử nghiệm sự phù hợp trên các lớp được dựa trên các lập lịch phân phối CPB. HRD có thể hoạt động trên cùng bộ xử lý như bộ mã hóa. Trong ví dụ khác, HRD có thể hoạt động trên bộ xử lý trong cùng khung với bộ mã hóa. Trong ví dụ khác, HRD có thể hoạt động trên bộ xử lý trong cùng mạng với bộ mã hóa.

Ở bước 1007, bộ mã hóa lưu trữ dòng bit để truyền thông với bộ giải mã.

FIG. 11 là lưu đồ phương pháp ví dụ 1100 giải mã chuỗi video từ dòng bit, như là dòng bit 800, trong đó tất cả các lớp trong dòng bit có cùng số lượng các lập lịch phân phối CPB, ví dụ lập lịch phân phối 561 CPB như khi được kiểm tra đối với sự phù hợp bởi HRD như là HRD 500. Phương pháp 1100 có thể được dùng bởi bộ giải mã, như là hệ thống codec 200, bộ giải mã 400, và/hoặc thiết bị tạo mã video 900 khi thực hiện phương pháp 100. Hơn nữa, phương pháp 1100 có thể hoạt động trên dòng bit mà gồm chuỗi video nhiều lớp 600 và/hoặc 700.

Phương pháp 1100 có thể bắt đầu khi bộ giải mã bắt đầu thu dòng bit của dữ liệu được tạo mã biểu diễn chuỗi video nhiều lớp, ví dụ như kết quả của phương pháp 1000. Ở bước 1101, bộ giải mã có thể thu dòng bit. Dòng bit bao gồm cấu trúc cú pháp các tham số HRD. Cấu trúc cú pháp các tham số HRD là đối tượng dữ liệu mà chứa một hoặc nhiều tham số HRD, mà có thể được chứa trong các phân tử cú pháp trong cấu trúc cú pháp. Dòng bit có thể cũng bao gồm một hoặc nhiều lớp của các ảnh được tạo mã, như là lớp N 631, lớp N+1 632, lớp phụ 710, lớp phụ 720, và/hoặc lớp phụ 730. Trong một số ví dụ, các lớp được tổ chức vào trong một hoặc nhiều OLS như là OLS 625. Các tham số HRD chỉ định rằng tất cả

các lớp được kết hợp với cùng số lượng các lập lịch phân phối CPB. Các lập lịch phân phối CPB có thể được kết hợp với tập các thử nghiệm sự phù hợp của dòng bit được thực hiện bởi HRD hoạt động trên bộ mã hóa. Các tham số HRD có thể được gồm trong VPS mà chỉ định các OLS. Trong ví dụ, số lượng của các lập lịch phân phối CPB được mô tả trong phần tử cú pháp số đếm HRD CPB trừ một (`hrd_cpb_cnt_minus1`). Minus1 chỉ báo rằng giá trị của `hrd_cpb_cnt_minus1` là 1 ít hơn giá trị được chỉ báo và việc cộng 1 cần bổ sung để thu giá trị thật. `Hrd_cpb_cnt_minus1` có thể trong khoảng từ 0 tới 31, bao gồm. Điều này chỉ báo rằng 1 tới 32 lập lịch phân phối CPB có thể được chỉ định và tất cả các lớp được kết hợp với cùng số lượng các lập lịch phân phối CPB.

Trong một số ví dụ, mỗi lập lịch phân phối tương ứng CPB cho OLS đích có thể được tham chiếu bởi chỉ số lập lịch duy nhất (`ScIdx`). `ScIdx` có thể gồm giá trị trong khoảng từ 0 tới giá trị của `hrd_cpb_cnt_minus1`, bao gồm. Điều này chỉ báo rằng `ScIdx` có thể thay đổi từ giá trị 0 để lập lịch phân phối CPB thứ nhất tới giá trị của `hrd_cpb_cnt_minus1` để lập lịch phân phối CPB cuối cùng. Trong một số ví dụ, các lập lịch phân phối CPB được mô tả bởi độ trễ xóa CPB ban đầu và cặp phần bù xóa CPB ban đầu. Độ trễ xóa CPB ban đầu (`nal_initial_cpb_removal_delay`) là độ trễ có thể xóa mặc định được sử dụng khi loại bỏ đơn vị dữ liệu từ CPB của HRD khi kiểm tra ảnh, AU, và/hoặc DU trong dòng bit, OLS, và/hoặc lớp. Phần bù xóa CPB ban đầu (`nal_initial_cpb_removal_offset`) là phần bù xóa CPB mặc định được kết hợp với mỗi ảnh, AU, và/hoặc DU trong dòng bit, OLS, và/hoặc lớp. Mỗi lớp có thể cũng được kết hợp với số BP CPB minus1 (`bp_cpb_cnt_minus1`) cộng 1 phần tử cú pháp. Số BP CPB minus1 là phần tử cú pháp mà chỉ định số lượng độ trễ xóa CPB ban đầu và cặp phần bù xóa CPB ban đầu, và do đó số lượng lập lịch phân phối mà có sẵn cho mỗi lớp/lớp phụ. Giá trị của `bp_cpb_cnt_minus1` có thể được đặt bằng giá trị của `hrd_cpb_cnt_minus1`. Điều này đảm bảo rằng số lượng lập lịch phân phối đối với mỗi lớp là giống với tổng số lập lịch phân phối CPB. Theo đó, tất cả các lớp có cùng số lượng các lập lịch phân phối CPB. Các phần tử cú pháp mà mô tả các lập lịch phân phối CPB có thể được gồm trong các thông báo SEI. Ví dụ, `nal_initial_cpb_removal_delay`, `nal_initial_cpb_removal_offset`, và `bp_cpb_cnt_minus1` có thể được gồm trong thông báo BP SEI.

Ở bước 1103, bộ giải mã có thể giải mã ảnh/ảnh được tạo mã từ một trong các lớp và/hoặc các OLS thu ảnh được giải mã. Bộ giải mã có thể cũng chuyển tiếp ảnh được giải mã để hiển thị như một phần của chuỗi video được giải mã ở bước 1105.

FIG. 12 là sơ đồ khối của hệ thống ví dụ 1200 để tạo mã chuỗi video sử dụng dòng bit trong đó tất cả các lớp trong dòng bit có cùng số lượng các lập lịch phân phối CPB, như

là lập lịch phân phối CPB 561. Hệ thống 1200 có thể được triển khai bởi bộ mã hóa và bộ giải mã như là hệ thống codec 200, bộ mã hóa 300, bộ giải mã 400, và/hoặc thiết bị tạo mã video 900. Hơn nữa, hệ thống 1200 có thể dùng HRD 500 để thực hiện các thử nghiệm sự phù hợp trên chuỗi video nhiều lớp 600, chuỗi video nhiều lớp 700, và/hoặc dòng bit 800. Hơn nữa, hệ thống 1200 có thể được dùng khi thực hiện phương pháp 100, 1000, và/hoặc 1100.

Hệ thống 1200 gồm bộ mã hóa video 1202. Bộ mã hóa video 1202 bao gồm môđun mã hóa 1203 để mã hóa dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã. Môđun mã hóa 1203 còn để mã hóa vào trong dòng bit cấu trúc cú pháp các tham số HRD chỉ định rằng tất cả các lớp được kết hợp với cùng số lượng lập lịch phân phối bộ đệm ảnh được tạo mã (coded picture buffer, CPB). Bộ mã hóa video 1202 còn bao gồm môđun HRD 1205 để thực hiện tập các thử nghiệm sự phù hợp của dòng bit trên các lớp được dựa trên lập lịch phân phối CPB. Bộ mã hóa video 1202 còn bao gồm môđun lưu trữ 1206 để lưu trữ dòng bit để truyền thông với bộ giải mã. Bộ mã hóa video 1202 còn bao gồm môđun truyền 1207 để truyền dòng bit chuyển tiếp bộ giải mã video 1210. Bộ mã hóa video 1202 có thể còn được tạo cấu hình để thực hiện bất kỳ trong các bước của phương pháp 1000.

Hệ thống 1200 cũng gồm bộ giải mã video 1210. Bộ giải mã video 1210 bao gồm môđun thu 1211 để thu dòng bit bao gồm cấu trúc cú pháp các tham số HRD và một hoặc nhiều lớp của các ảnh được tạo mã như tất cả các lớp được kết hợp với cùng số lượng các lập lịch phân phối CPB như được chỉ định bởi cấu trúc cú pháp các tham số HRD. Bộ giải mã video 1210 còn bao gồm môđun giải mã 1213 để giải mã ảnh từ một trong các lớp để thu ảnh được giải mã. Bộ giải mã video 1210 còn bao gồm môđun chuyển tiếp 1215 để chuyển tiếp ảnh được giải mã để hiển thị như một phần của chuỗi video được giải mã. Bộ giải mã video 1210 có thể còn được tạo cấu hình để thực hiện bất kỳ trong các bước của phương pháp 1100.

Thành phần thứ nhất được ghép trực tiếp với thành phần thứ hai khi không có thành phần nào xen vào, ngoại trừ một đường, một vết hoặc phương tiện khác giữa thành phần thứ nhất và thành phần thứ hai. Thành phần đầu tiên được ghép gián tiếp với thành phần thứ hai khi có các thành phần xen vào không phải là một đường, một vết hoặc một phương tiện khác giữa thành phần thứ nhất và thành phần thứ hai. Thuật ngữ "được kết hợp" và các biến thể của nó bao gồm cả được kết hợp trực tiếp và kết hợp gián tiếp. Việc sử dụng thuật ngữ "khoảng" có nghĩa là một phạm vi bao gồm $\pm 10\%$ của số tiếp theo trừ khi có quy định khác.

Cũng cần hiểu rằng các bước của các phương pháp làm ví dụ nêu ở đây không nhất thiết phải được thực hiện theo thứ tự được mô tả và thứ tự của các bước của các phương pháp

đó phải được hiểu đơn thuần chỉ là làm mẫu. Tương tự như vậy, các bước bổ sung có thể được bao gồm trong các phương pháp này và các bước nhất định có thể được bỏ qua hoặc kết hợp, trong các phương pháp phù hợp với các phương án khác nhau của sáng chế.

Mặc dù một số phương án đã được đưa ra trong sáng chế, nhưng có thể hiểu rằng các hệ thống và phương pháp được bộc lộ có thể được thể hiện dưới nhiều dạng cụ thể khác mà không khác với bản chất hoặc phạm vi của sáng chế. Các ví dụ hiện tại chỉ được coi là minh họa và không bị hạn chế, và mục đích không bị giới hạn ở các chi tiết được đưa ra ở đây. Ví dụ, các yếu tố hoặc thành phần khác nhau có thể được kết hợp hoặc tích hợp trong một hệ thống khác hoặc một số tính năng nhất định có thể bị bỏ qua hoặc không được triển khai.

Hơn nữa, các kỹ thuật, hệ thống, hệ thống phụ và phương pháp được mô tả và minh họa trong các phương án khác nhau dưới dạng rời rạc hoặc riêng biệt có thể được kết hợp hoặc tích hợp với các hệ thống, thành phần, kỹ thuật hoặc phương pháp khác mà không nằm ngoài phạm vi của sáng chế. Các ví dụ khác về thay đổi, thay thế và thay đổi có thể xác định được bởi một người có hiểu biết trung bình trong lĩnh vực này và có thể được thực hiện mà không cần vượt quá bản chất và phạm vi của sáng chế mà được bộc lộ ở đây.

YÊU CẦU BẢO HỘ

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

thu, bởi bộ thu của bộ giải mã, dòng bit bao gồm cấu trúc cú pháp các tham số bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD) và một hoặc nhiều lớp của các ảnh được tạo mã sao cho tất cả các lớp được kết hợp với cùng số lượng các lập lịch phân phối bộ đệm ảnh được tạo mã (coded picture buffer, CPB) như được chỉ định bởi cấu trúc cú pháp các tham số HRD; và

giải mã, bởi bộ xử lý của bộ giải mã, ảnh từ một trong các lớp để thu ảnh được giải mã.

2. Phương pháp theo điểm 1, trong đó các lập lịch phân phối CPB được kết hợp với tập các thử nghiệm sự phù hợp của dòng bit được thực hiện bởi HRD hoạt động trên bộ mã hóa.

3. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 2, trong đó số lượng các lập lịch phân phối CPB được chỉ định trong số đếm HRD CPB trừ một (`hrd_cpb_cnt_minus1`), và trong đó `hrd_cpb_cnt_minus1` nằm trong khoảng từ 0 tới 31, bao gồm.

4. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 2, trong đó các lớp được tổ chức vào trong một hoặc nhiều tập lớp đầu ra (output layer set, OLS), và trong đó lập lịch phân phối CPB tương ứng cho OLS đích được tham chiếu bởi chỉ số lập lịch (`ScIdx`) trong khoảng từ 0 tới giá trị của `hrd_cpb_cnt_minus1`, bao gồm.

5. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 2, trong đó mỗi trong các lập lịch phân phối CPB được mô tả bởi độ trễ xóa CPB ban đầu lớp trừu tượng mạng (network abstraction layer, NAL) (`nal_initial_cpb_removal_delay`) và phần bù xóa CPB ban đầu NAL (`nal_initial_cpb_removal_offset`).

6. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 2, trong đó mỗi lớp được kết hợp với số đếm CPB khoảng đệm (BP, buffering period) trừ 1 (`bp_cpb_cnt_minus1`) cộng 1 mà chỉ định số lượng cặp `nal_initial_cpb_removal_delay` và `nal_initial_cpb_removal_offset` cho mỗi lớp, và trong đó `bp_cpb_cnt_minus1` bằng giá trị của `hrd_cpb_cnt_minus1`.

7. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 4, trong đó các tham số HRD các tham số HRD được gồm trong dòng bit trong tập tham số video (video parameter set, VPS) chỉ định các OLS.
8. Phương pháp mã hóa được thực hiện bởi bộ mã hóa, phương pháp này bao gồm các bước:
- mã hóa, bởi bộ xử lý của bộ mã hóa, dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã;
 - mã hóa vào trong dòng bit, bởi bộ xử lý, cấu trúc cú pháp các tham số bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD) chỉ định rằng tất cả các lớp được kết hợp với cùng số lượng các lập lịch phân phối bộ đệm ảnh được tạo mã (coded picture buffer, CPB); và
 - thực hiện, bởi bộ xử lý, tập các thử nghiệm sự phù hợp của dòng bit trên các lớp được dựa trên các lập lịch phân phối CPB.
9. Phương pháp theo điểm 8, trong đó tập các thử nghiệm sự phù hợp của dòng bit được thực hiện bởi HRD hoạt động trên bộ xử lý.
10. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 9, trong đó số lượng các lập lịch phân phối CPB được chỉ định trong số đếm HRD CPB trừ một (`hrd_cpb_cnt_minus1`), và trong đó `hrd_cpb_cnt_minus1` nằm trong khoảng từ 0 tới 31, bao gồm.
11. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 9, trong đó các lớp được tổ chức vào trong một hoặc nhiều tập lớp đầu ra (output layer set, OLS), và trong đó lập lịch phân phối CPB tương ứng cho OLS đích được tham chiếu bởi chỉ số lập lịch (`ScIdx`) trong khoảng từ 0 tới giá trị của `hrd_cpb_cnt_minus1`, bao gồm.
12. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 9, trong đó mỗi trong các lập lịch phân phối CPB được mô tả bởi độ trễ xóa CPB ban đầu lớp trừu tượng mạng (network abstraction layer, NAL) (`nal_initial_cpb_removal_delay`) và phần bù xóa CPB ban đầu NAL (`nal_initial_cpb_removal_offset`).

13. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 9, trong đó mỗi lớp được kết hợp với số đếm CPB khoảng đệm (BP, buffering period) trừ 1 ($bp_cpb_cnt_minus1$) cộng 1 mà chỉ định số lượng cặp $nal_initial_cpb_removal_delay$ và $nal_initial_cpb_removal_offset$ cho mỗi lớp, và trong đó $bp_cpb_cnt_minus1$ bằng giá trị của $hrd_cpb_cnt_minus1$.

14. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 9, trong đó cấu trúc cú pháp các tham số HRD được mã hóa trong dòng bit trong tập tham số video (video parameter set, VPS) chỉ định các OLS.

15. Thiết bị tạo mã video bao gồm:

bộ xử lý, bộ thu được ghép với bộ xử lý, bộ nhớ được ghép với bộ xử lý, và bộ phát được ghép với bộ xử lý, trong đó bộ xử lý, bộ thu, bộ nhớ, và bộ phát được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 7.

16. Thiết bị tạo mã video bao gồm:

bộ xử lý, bộ thu được ghép với bộ xử lý, bộ nhớ được ghép với bộ xử lý, và bộ phát được ghép với bộ xử lý, trong đó bộ xử lý, bộ thu, bộ nhớ, và bộ phát được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 14.

17. Phương tiện đọc được bằng máy tính không tạm thời bao gồm sản phẩm chương trình máy tính để sử dụng bởi thiết bị tạo mã video, sản phẩm chương trình máy tính bao gồm các lệnh thực thi được bằng máy tính được lưu trữ trên phương tiện đọc được bằng máy tính không tạm thời sao cho khi được thực thi bởi bộ xử lý làm cho thiết bị tạo mã video để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 7.

18. Phương tiện đọc được bằng máy tính không tạm thời bao gồm sản phẩm chương trình máy tính để sử dụng bởi thiết bị tạo mã video, sản phẩm chương trình máy tính bao gồm các lệnh thực thi được bằng máy tính được lưu trữ trên phương tiện đọc được bằng máy tính không tạm thời sao cho khi được thực thi bởi bộ xử lý làm cho thiết bị tạo mã video để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 14.

19. Bộ giải mã bao gồm:

phương tiện thu để thu dòng bit bao gồm cấu trúc cú pháp các tham số bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD) và một hoặc nhiều lớp của các ảnh được

tạo mã sao cho tất cả các lớp được kết hợp với cùng số lượng các lập lịch phân phối bộ đệm ảnh được tạo mã (coded picture buffer, CPB) như được chỉ định bởi cấu trúc cú pháp các tham số HRD;

phương tiện giải mã để giải mã ảnh từ một trong các lớp để thu ảnh được giải mã; và
phương tiện chuyển tiếp để chuyển tiếp ảnh được giải mã để hiển thị như một phần của chuỗi video được giải mã.

20. Bộ giải mã theo điểm 19, trong đó bộ giải mã còn được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 7.

21. Bộ mã hóa bao gồm:

phương tiện mã hóa để:

mã hóa dòng bit bao gồm một hoặc nhiều lớp của các ảnh được tạo mã; và

mã hóa vào trong dòng bit cấu trúc cú pháp các tham số bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD) chỉ định rằng tất cả các lớp được kết hợp với cùng số lượng các lập lịch phân phối bộ đệm ảnh được tạo mã (coded picture buffer, CPB);

phương tiện HRD để thực hiện tập các thử nghiệm sự phù hợp của dòng bit trên các lớp được dựa trên các lập lịch phân phối CPB; và

phương tiện lưu trữ để lưu trữ dòng bit để truyền thông với bộ giải mã.

22. Bộ mã hóa theo điểm 21, trong đó bộ mã hóa còn được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 14.

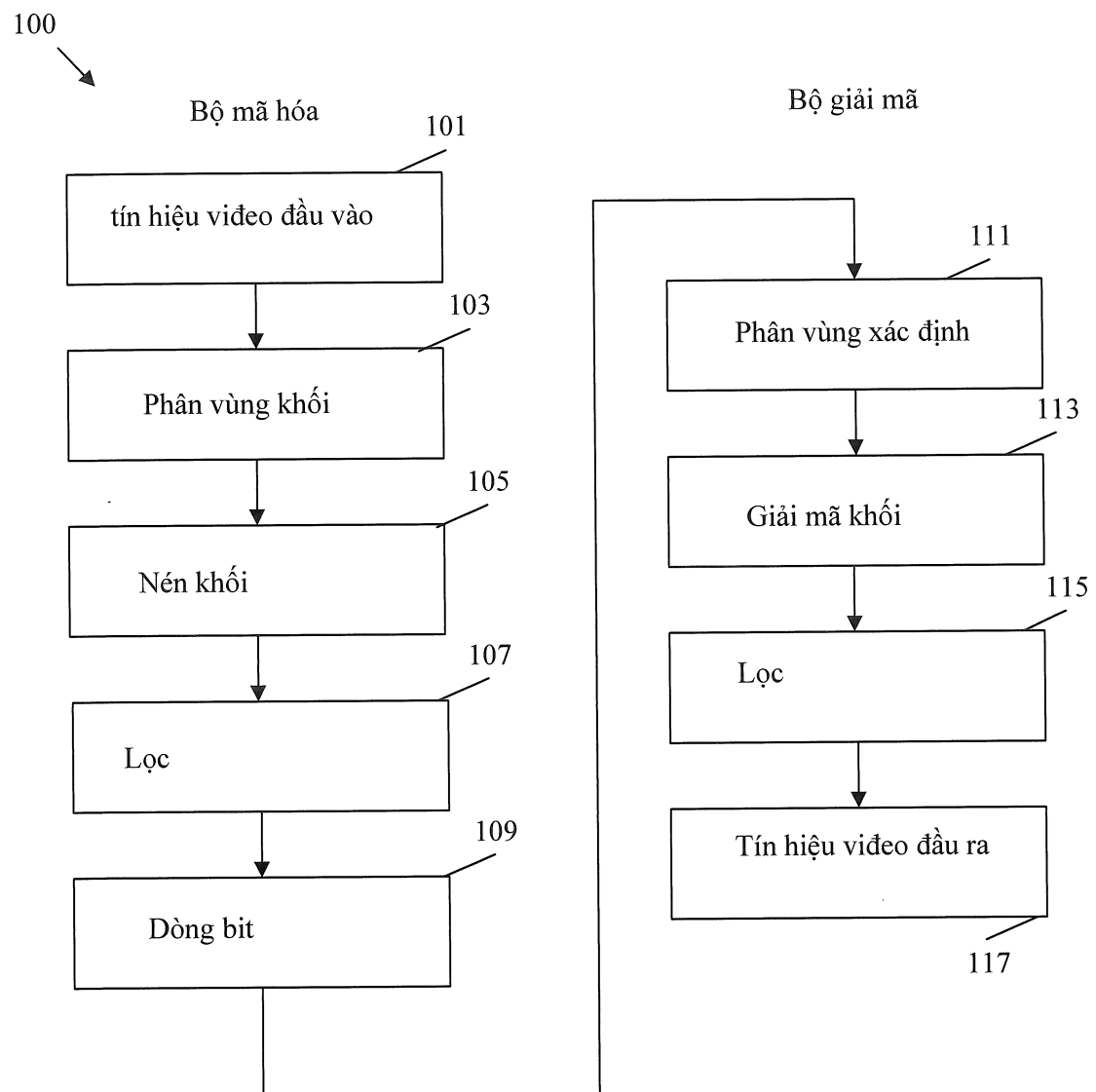


FIG. 1

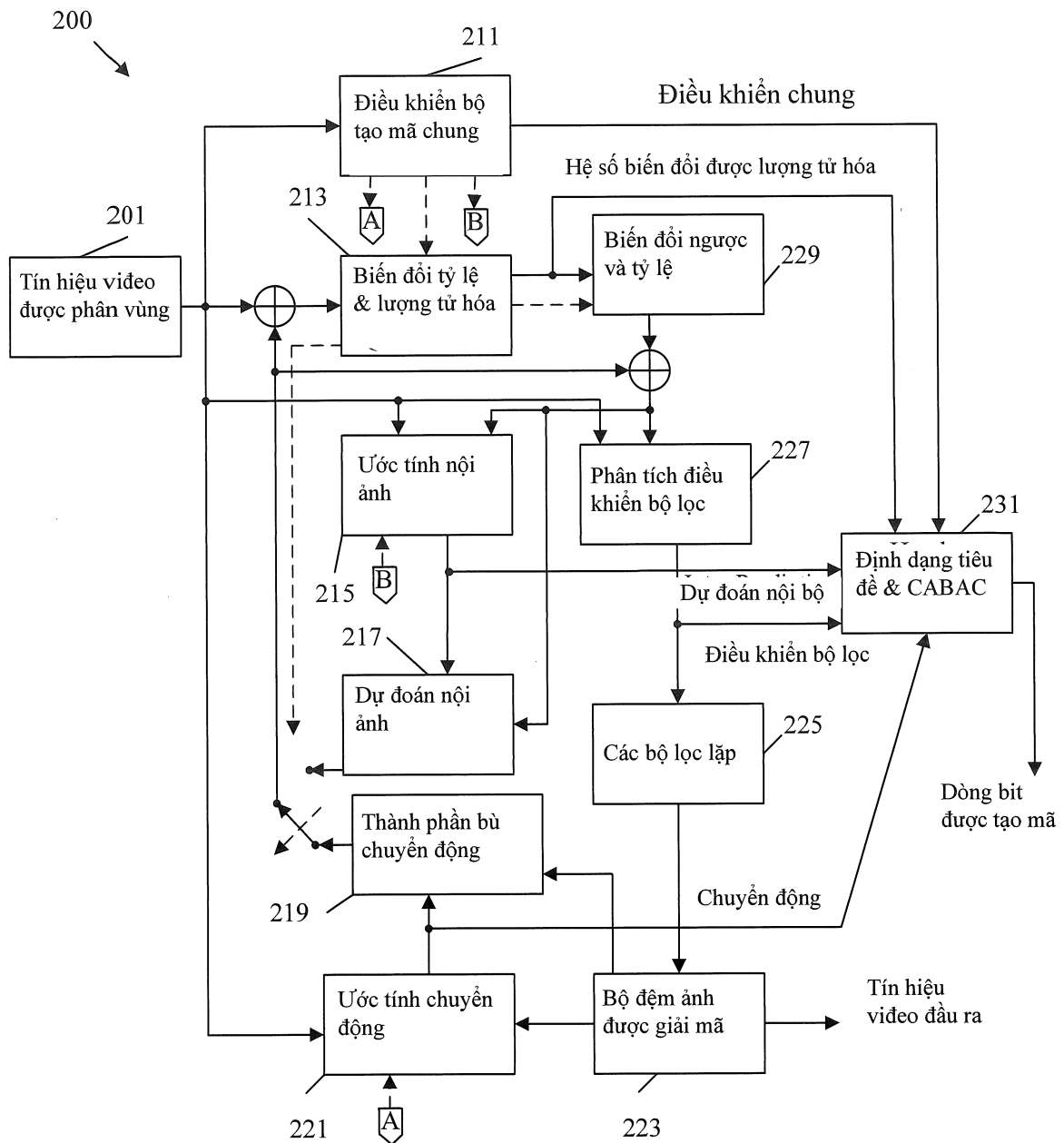


FIG. 2

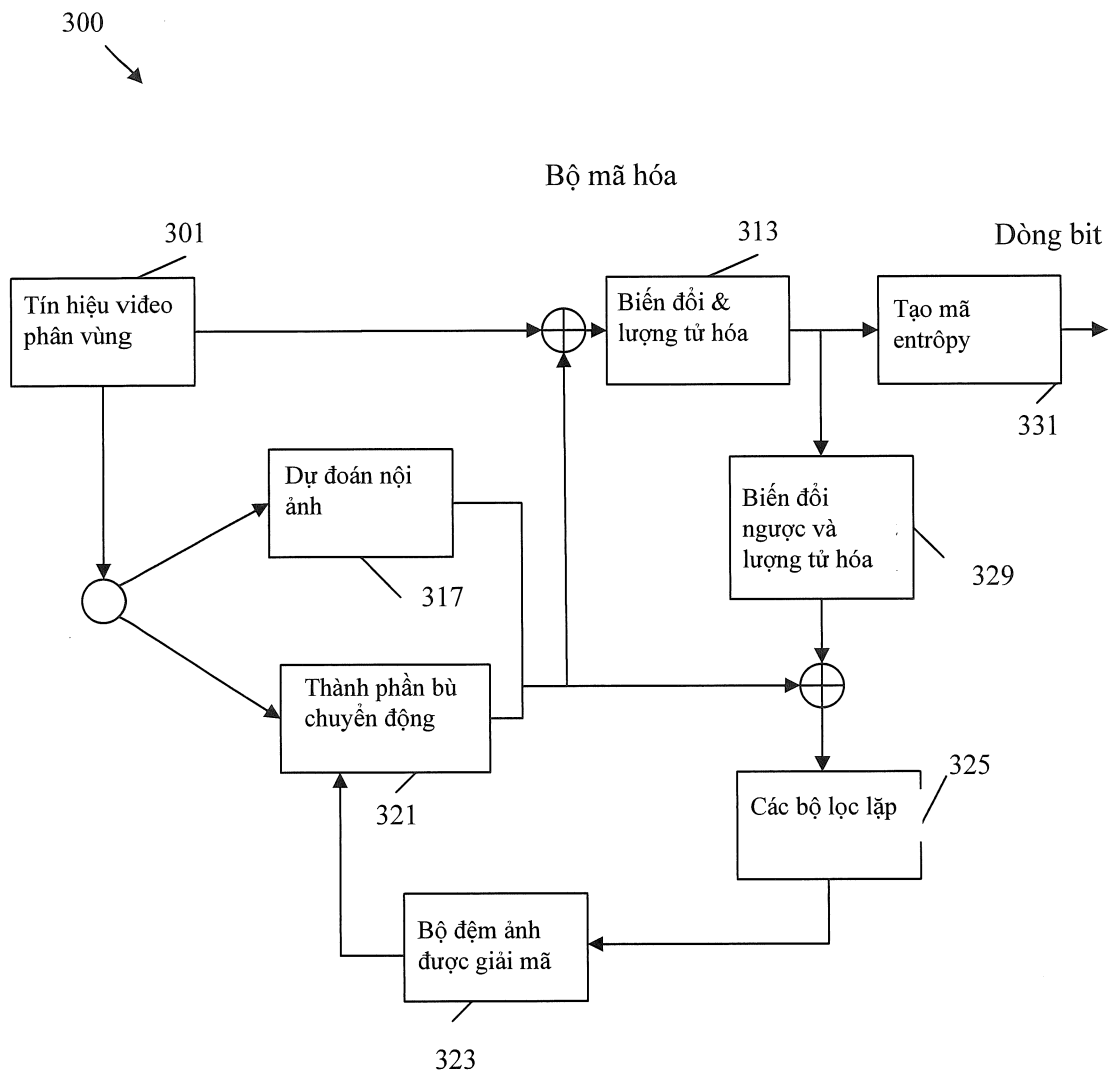


FIG. 3

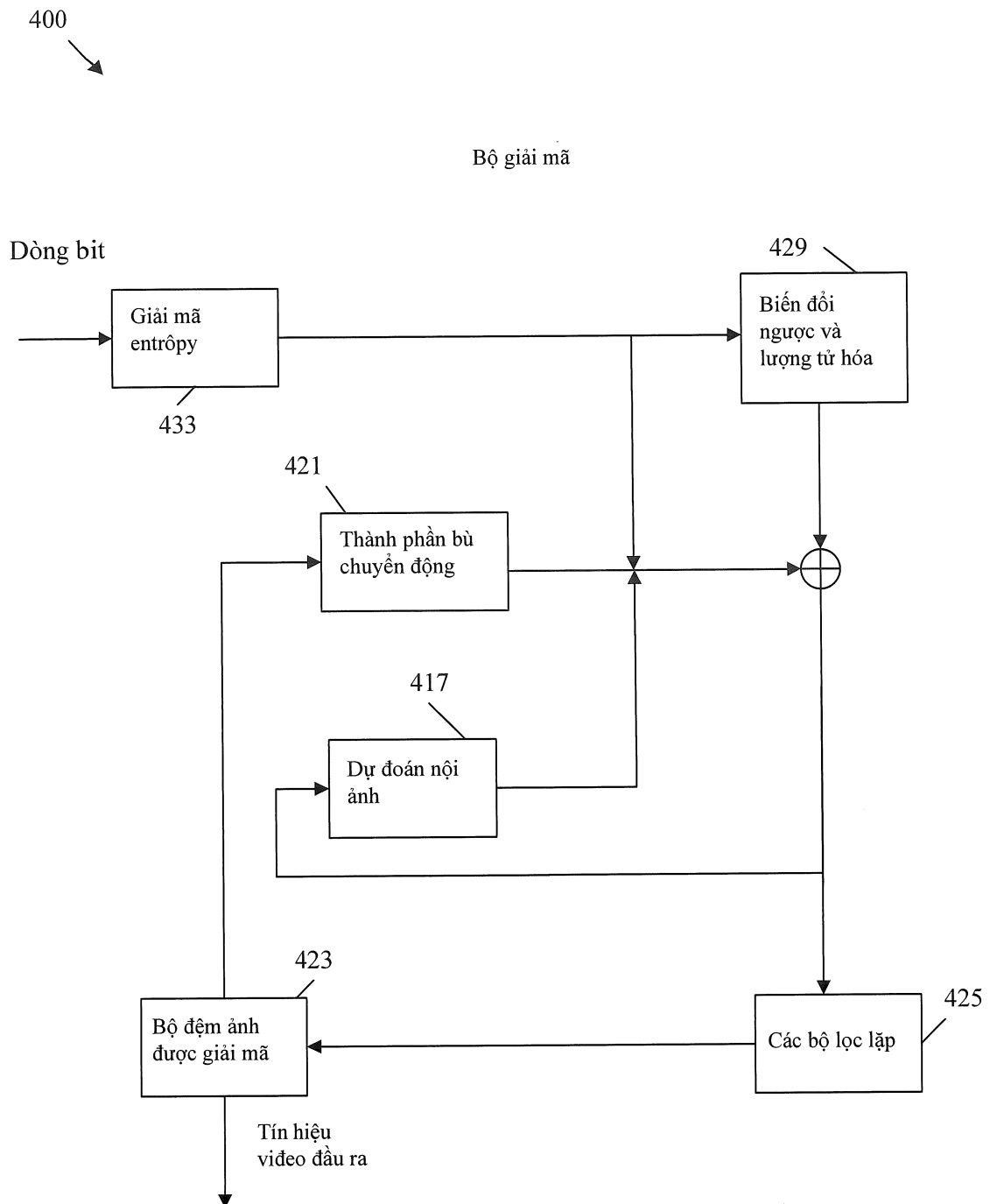


FIG. 4

5 / 12

Bộ giải mã tham chiếu giả định

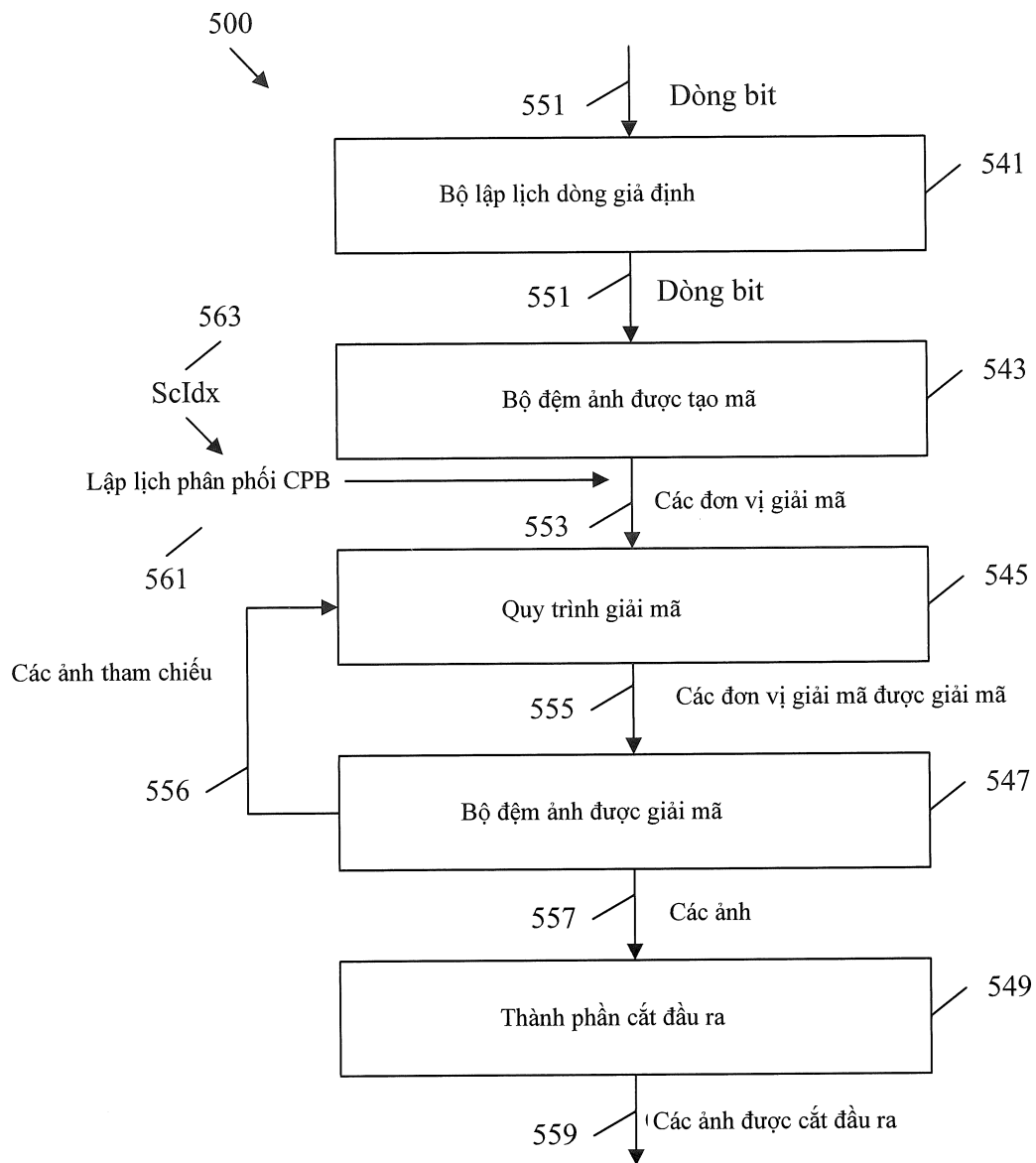


FIG. 5

6 / 12

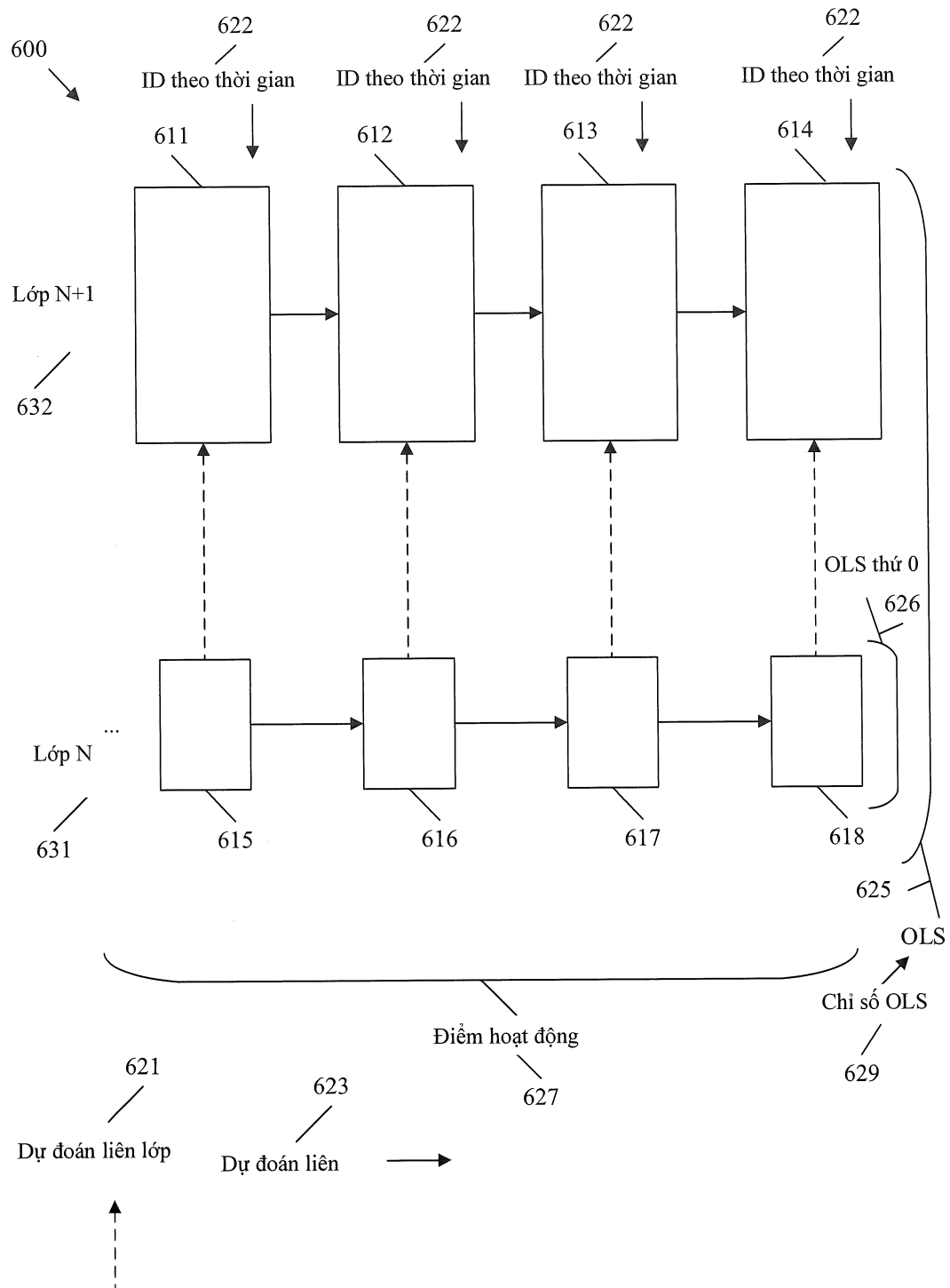


FIG. 6

7 / 12

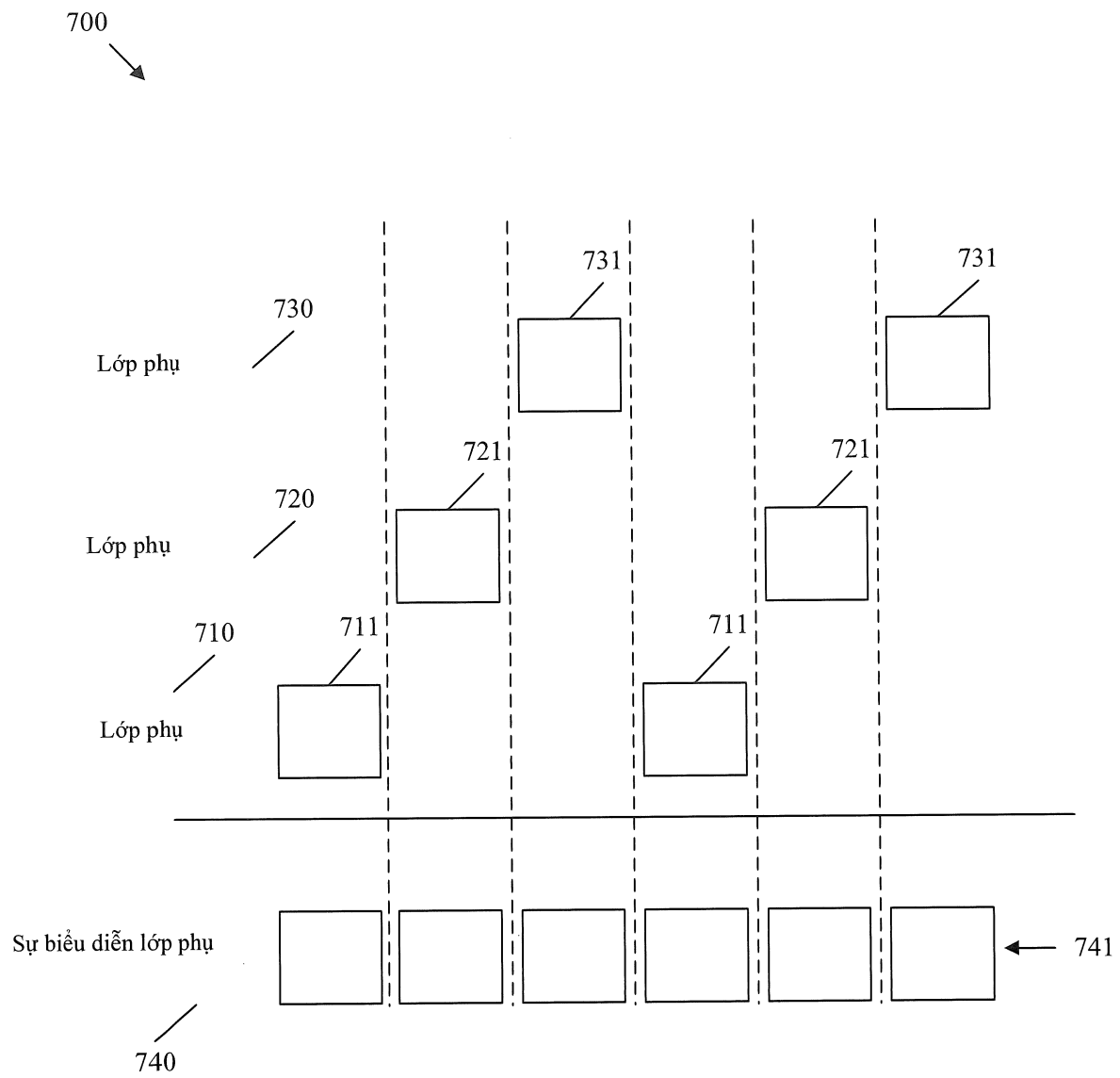


FIG. 7

8 / 12

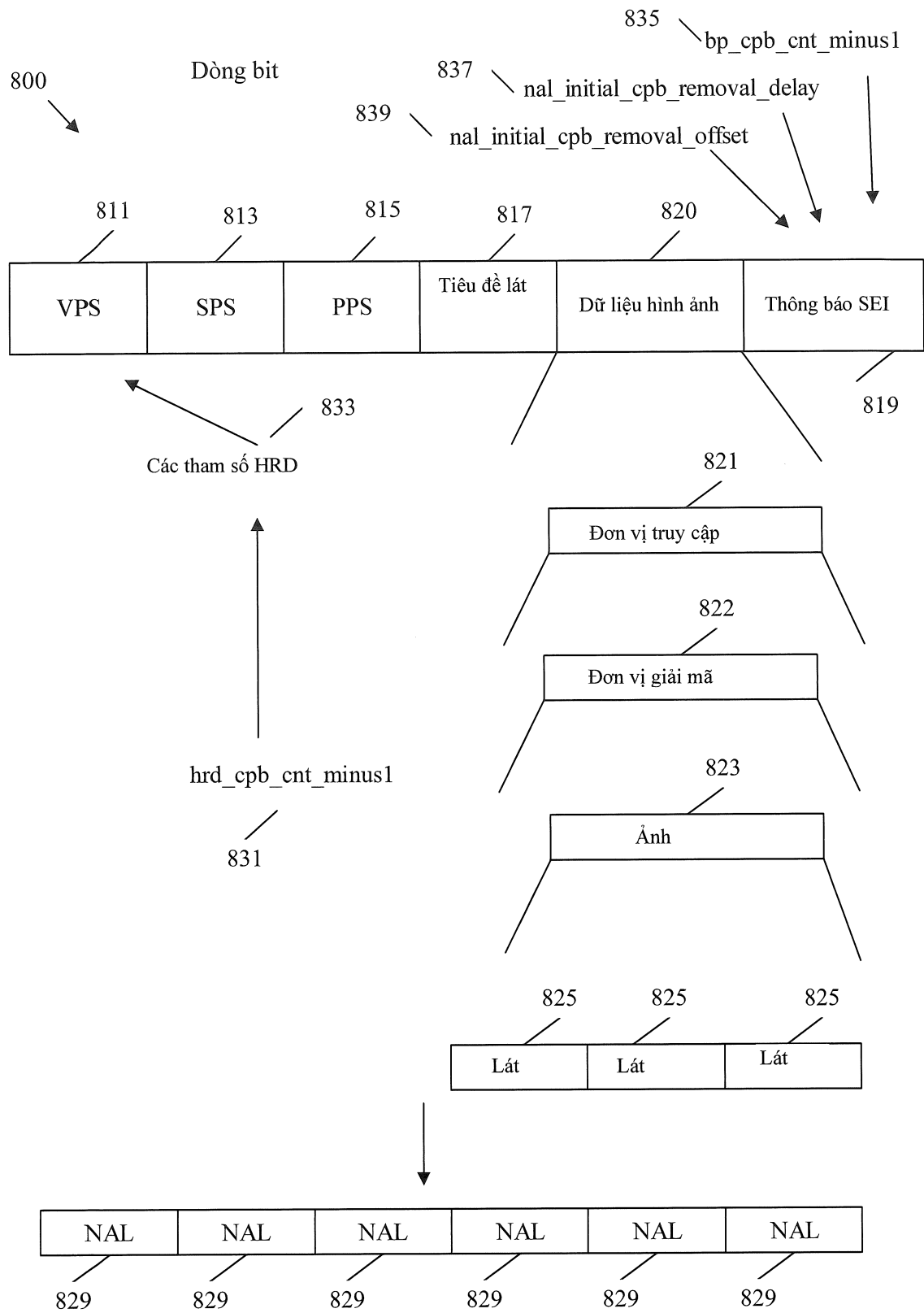


FIG. 8

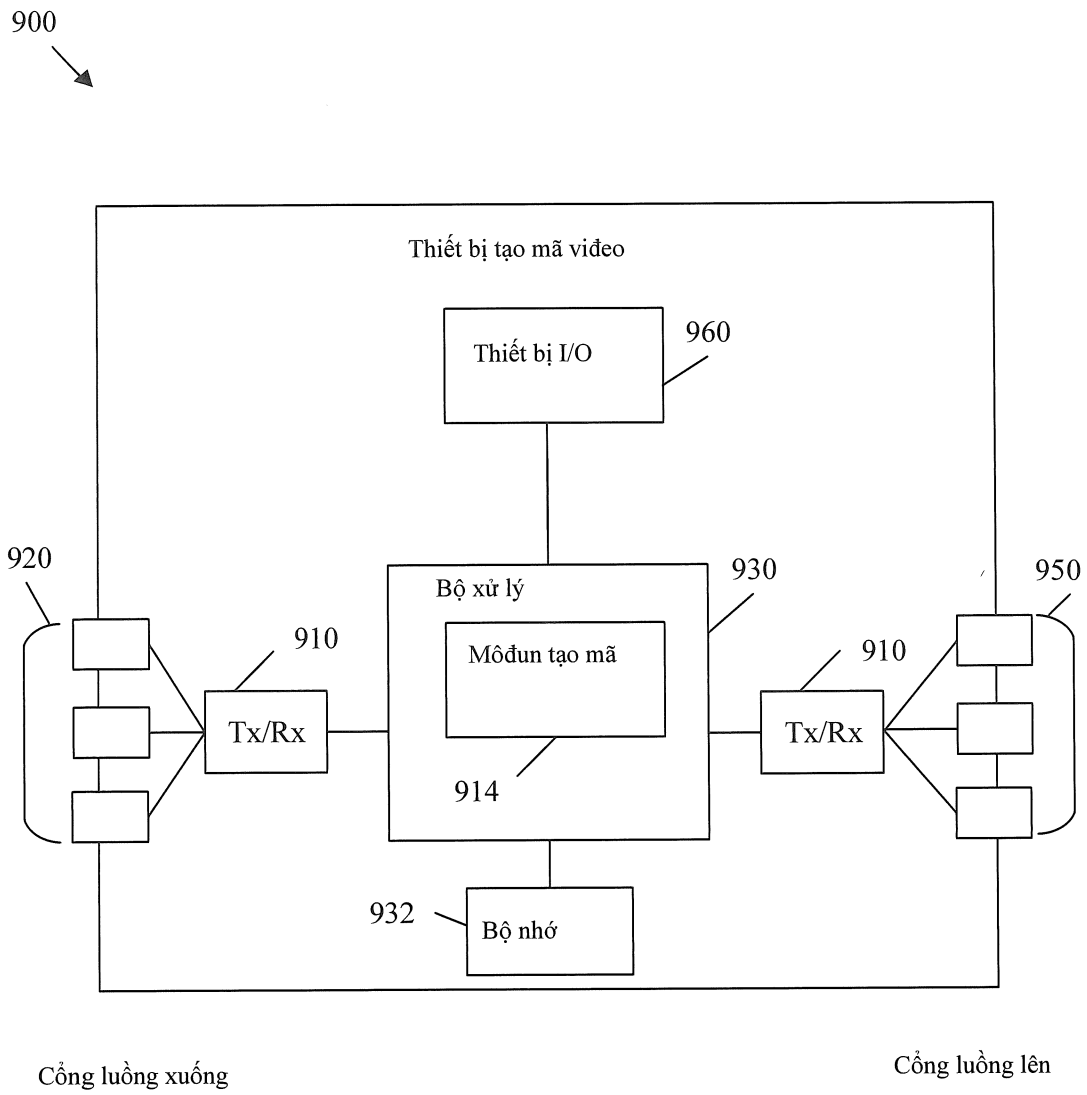


FIG. 9

10 / 12

1000

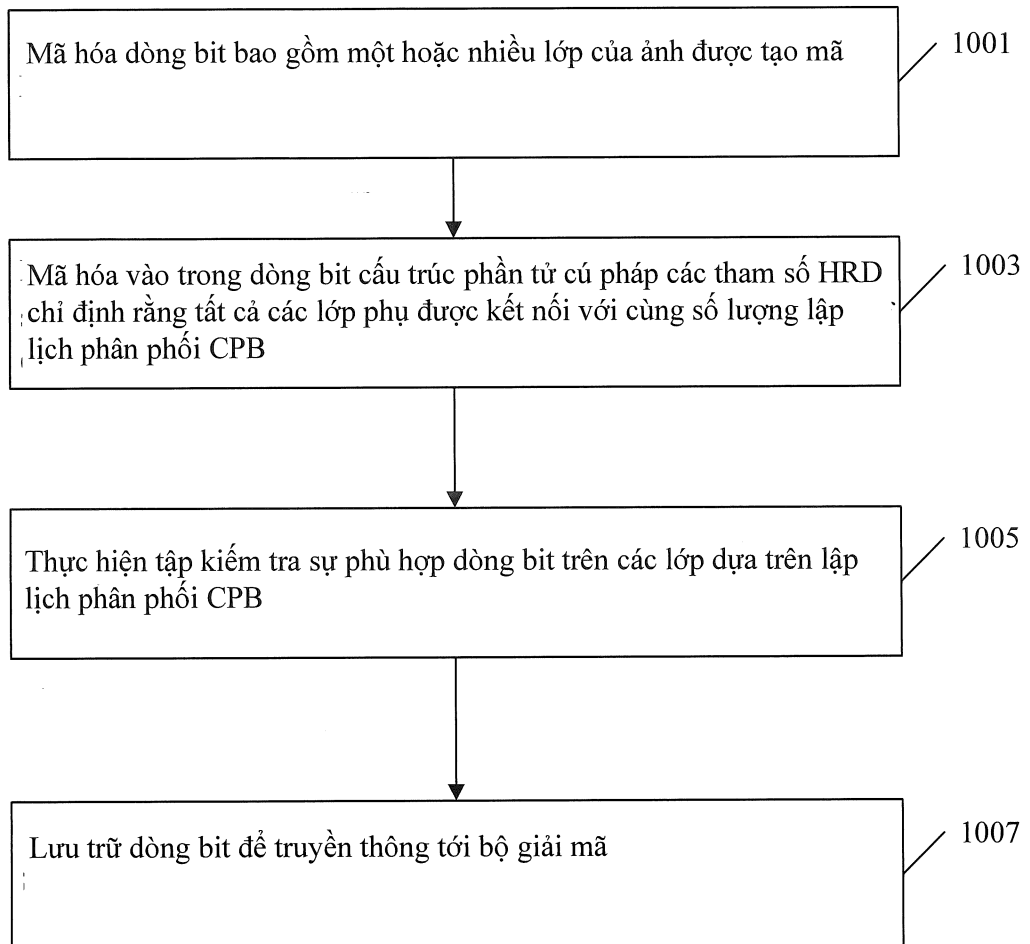


FIG. 10

11 / 12

1100

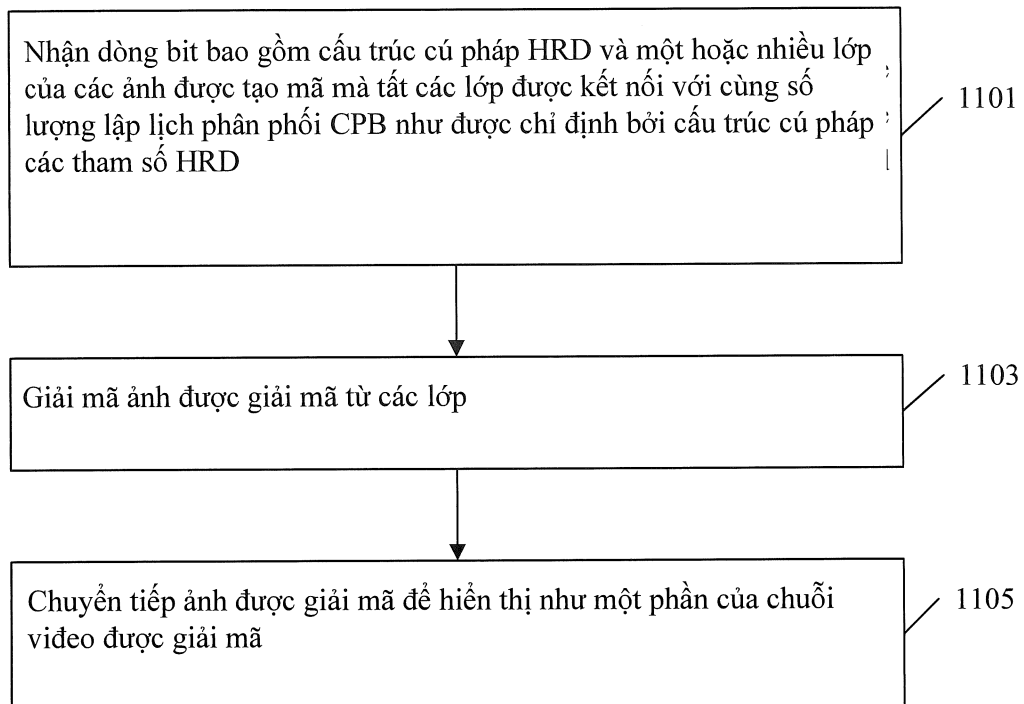


FIG. 11

12 / 12

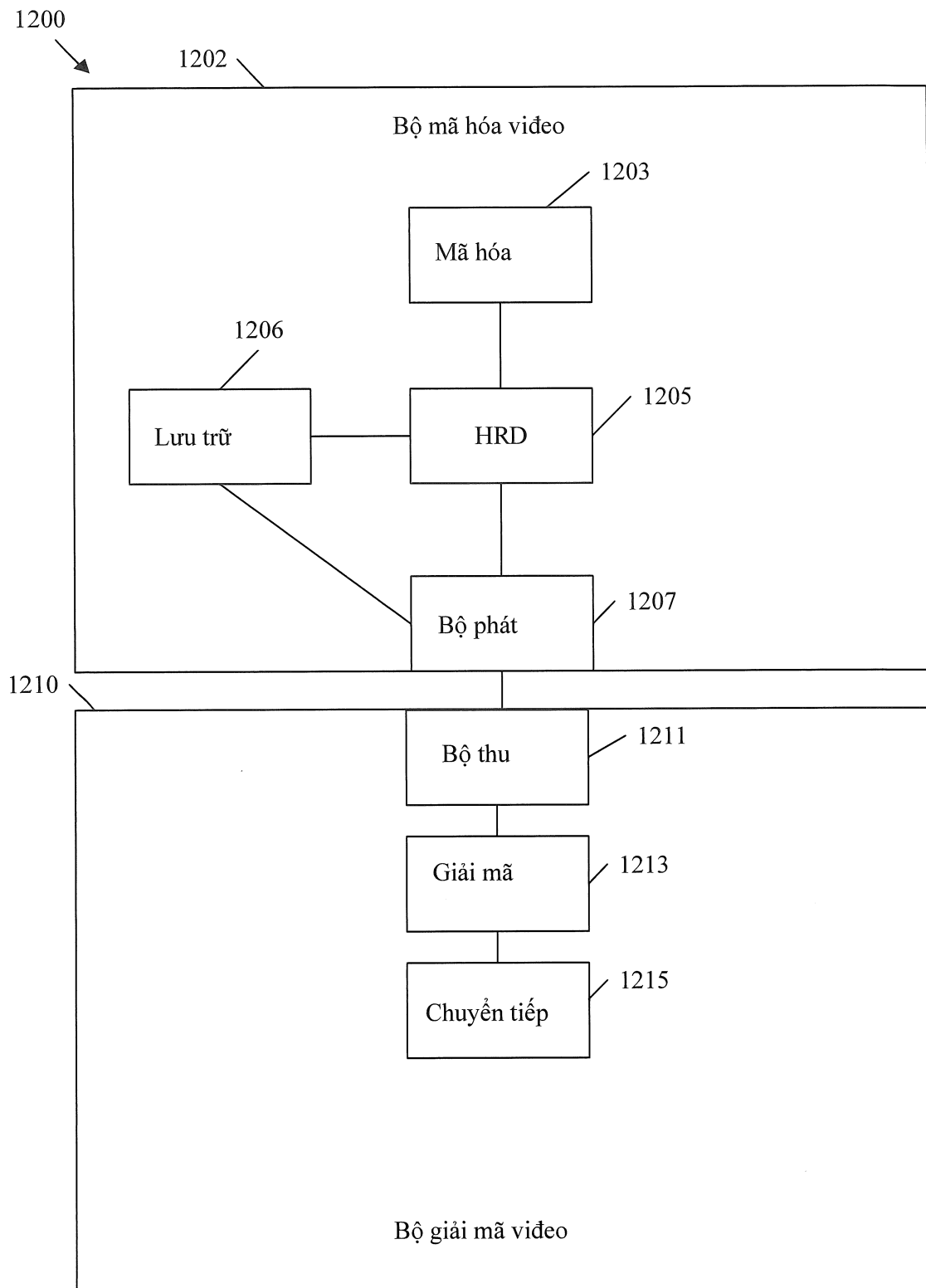


FIG. 12