



(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)



1-0025579

(51)⁷ H04N 21/2343

(13) B

(21) 1-2015-01460

(22) 18/09/2013

(86) PCT/US2013/060412 18/09/2013

(87) WO/2014/047183 27/03/2014

(30) 61/705,102 24/09/2012 US; 13/918,041 14/06/2013 US

(45) 25/09/2020 390

(43) 25/09/2015 330A

(73) QUALCOMM INCORPORATED (US)

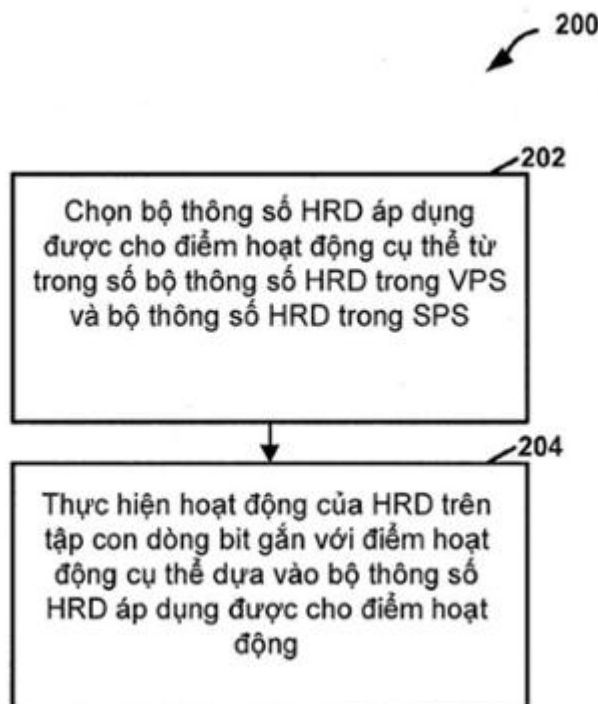
Attn: International IP Administration, 5775 Morehouse Drive, San Diego, California
92121, United States of America

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

(74) Công ty TNHH Quốc tế D & N (D&N INTERNATIONAL CO.,LTD.)

(54) PHƯƠNG PHÁP, THIẾT BỊ VÀ PHƯƠNG TIỆN LƯU TRỮ DỮ LIỆU BẤT BIẾN
ĐỌC ĐƯỢC BẰNG MÁY TÍNH ĐỂ XỬ LÝ DỮ LIỆU VIDEO

(57) Sáng chế đề cập đến phương pháp, thiết bị và phương tiện lưu trữ dữ liệu bất biến đọc được bằng máy tính để xử lý dữ liệu video. Theo sáng chế, thiết bị tính toán lựa chọn bộ thông số HRD có khả năng áp dụng cho điểm hoạt động cụ thể của dòng bit từ trong số bộ thông số giải mã chuẩn giả định (hypothetical reference decoder - HRD) trong bộ thông số video và bộ thông số HRD trong bộ thông số chuỗi. Thiết bị tính toán này thực hiện hoạt động HRD trên tập con dòng bit có liên quan đến điểm hoạt động cụ thể dựa ít nhất một phần vào bộ thông số HRD có khả năng áp dụng cho điểm hoạt động cụ thể.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến quy trình mã hóa và giải mã video (tức là, mã hóa và/hoặc giải mã dữ liệu video).

Tình trạng kỹ thuật của sáng chế

Các tính năng video số có thể được đưa vào nhiều loại thiết bị, bao gồm tivi số, hệ thống phát rộng trực tiếp số, hệ thống phát thanh không dây, thiết bị trợ giúp số cá nhân (personal digital assistant-PDA), máy tính xách tay hoặc máy tính để bàn, máy tính bảng, máy đọc sách điện tử, máy ảnh số, thiết bị ghi âm số, thiết bị nghe nhạc số, thiết bị chơi điện tử video, thiết bị giao tiếp trò chơi video, điện thoại di động hoặc điện thoại vô tuyến vệ tinh, còn gọi là “điện thoại thông minh”, thiết bị hội nghị truyền hình ảnh từ xa, thiết bị tạo chuỗi video, và tương tự. Các thiết bị video số tiến hành các kỹ thuật nén video, như các kỹ thuật được mô tả trong các tiêu chuẩn được xác định bởi MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Phần 10, mã hóa dữ liệu video cải tiến (Advanced Video Coding-AVC), tiêu chuẩn mã hóa dữ liệu video hiệu suất cao (High Efficiency Video Coding-HEVC) hiện đang được phát triển, và các phiên bản mở rộng của các tiêu chuẩn này. Các thiết bị video có thể truyền, thu, mã hóa, giải mã, và/hoặc lưu trữ thông tin video số hiệu quả hơn bằng cách tiến hành các kỹ thuật nén video này.

Các kỹ thuật nén video tiến hành thao tác dự báo không gian (nội hình ảnh) và/hoặc thao tác dự báo thời gian (liên hình ảnh) để làm giảm hoặc loại bỏ dữ liệu dư vốn có trong các chuỗi video. Để mã hóa dữ liệu video dựa vào khối, lát video (tức là, khung video hoặc một phần của khung video) có thể được phân tách thành các khối video. Các khối video trong lát hình ảnh mã hóa nội cấu trúc (I) được mã hóa bằng cách sử dụng kỹ thuật dự báo không gian dựa vào các mẫu chuẩn trong các khối lân cận trong cùng một hình ảnh. Các khối video trong lát hình ảnh mã hóa liên cấu trúc (P hoặc B) có thể sử dụng kỹ thuật dự báo không gian dựa vào các mẫu chuẩn trong các khối lân cận trong cùng một hình ảnh hoặc kỹ thuật dự báo thời gian dựa vào các mẫu chuẩn trong các hình ảnh chuẩn khác. Hình ảnh có thể được gọi là khung, và hình ảnh chuẩn có thể được gọi là khung chuẩn.

Quy trình dự báo thời gian hoặc không gian đưa ra khối dự báo cho khối cần được mã hóa. Dữ liệu dư biểu diễn độ chênh lệch điểm ảnh giữa khối gốc cần được mã hóa và khối dự báo. Khối mã hóa liên cấu trúc được mã hóa theo vector chuyển động chỉ đến khối của các mẫu chuẩn tạo thành khối dự báo, và dữ liệu dư chỉ báo độ chênh lệch giữa khối mã hóa và khối dự báo. Khối mã hóa nội cấu trúc được mã hóa theo chế độ mã hóa nội cấu trúc và dữ liệu dư. Để nén hơn nữa, dữ liệu dư có thể được chuyển đổi từ miền điểm ảnh sang miền biến đổi, đưa ra các hệ số biến đổi dư, sau đó có thể được lượng tử hóa. Các hệ số biến đổi đã lượng tử hóa, ban đầu được sắp xếp theo mảng hai chiều, có thể được quét để tạo ra vector một chiều của các hệ số, và quy trình mã hóa entropy có thể được áp dụng để nén được nhiều hơn nữa.

Dòng bit mã hóa đa cảnh nhìn có thể được tạo ra bởi các cảnh nhìn mã hóa, ví dụ từ nhiều hình ảnh phối cảnh. Một vài tiêu chuẩn video ba chiều (3D) đã được phát triển mà sử dụng các khía cạnh mã hóa đa cảnh nhìn. Ví dụ, các cảnh nhìn khác nhau có thể truyền các cảnh nhìn ở mắt trái và phải để hỗ trợ cho video 3D. Theo cách khác, một vài quy trình mã hóa dữ liệu video 3D có thể áp dụng quy trình mã hóa đa cảnh nhìn cùng với độ sâu. Để mã hóa đa cảnh nhìn cùng với độ sâu, dòng bit video 3D có thể không chỉ chứa các thành phần cảnh nhìn cấu trúc, mà còn chứa các thành phần cảnh nhìn độ sâu. Ví dụ, mỗi cảnh nhìn có thể bao gồm một thành phần cảnh nhìn cấu trúc và một thành phần cảnh nhìn độ sâu.

Bản chất kỹ thuật của sáng chế

Nói chung, sáng chế mô tả việc báo hiệu và chọn thông số của bộ giải mã chuẩn giả định (hypothetical reference decoder-HRD) để mã hóa dữ liệu video. Cụ thể hơn là, thiết bị tính toán có thể lựa chọn, từ trong số tập hợp thông số của bộ giải mã chuẩn giả định (HRD) trong bộ thông số video (video parameter set-VPS) và tập hợp thông số HRD trong bộ thông số chuỗi (sequence parameter set-SPS), tập hợp thông số HRD có thể áp dụng cho một điểm hoạt động cụ thể của dòng bit. Thiết bị tính toán thực hiện, dựa ít nhất một phần vào tập hợp thông số HRD có thể áp dụng cho một điểm hoạt động cụ thể, hoạt động HRD trên tập hợp con dòng bit liên quan đến điểm hoạt động cụ thể.

Theo một ví dụ, sáng chế mô tả phương pháp xử lý dữ liệu video. Phương pháp này bao gồm bước lựa chọn, từ trong số tập hợp thông số HRD trong VPS và tập hợp thông số HRD trong SPS, tập hợp thông số HRD có thể áp dụng cho một điểm hoạt động cụ thể của dòng bit. Phương pháp này còn bao gồm bước thực hiện, dựa ít nhất một phần

vào tập hợp thông số HRD đã chọn có thể áp dụng cho một điểm hoạt động cụ thể, hoạt động HRD trên tập hợp con dòng bit liên quan đến điểm hoạt động cụ thể.

Theo một ví dụ khác, sáng chế mô tả thiết bị chứa một hoặc nhiều bộ xử lý được tạo cấu hình để lựa chọn, từ trong số tập hợp thông số HRD trong VPS và tập hợp thông số HRD trong SPS, tập hợp thông số HRD có thể áp dụng cho một điểm hoạt động cụ thể của dòng bit. Một hoặc nhiều bộ xử lý này còn được tạo cấu hình để thực hiện, dựa ít nhất một phần vào tập hợp thông số HRD đã chọn có thể áp dụng cho một điểm hoạt động cụ thể, hoạt động HRD trên tập hợp con dòng bit liên quan đến điểm hoạt động cụ thể.

Theo một ví dụ khác, sáng chế mô tả thiết bị bao gồm phương tiện lựa chọn, từ trong số tập hợp thông số HRD trong VPS và tập hợp thông số HRD trong SPS, tập hợp thông số HRD có thể áp dụng cho một điểm hoạt động cụ thể của dòng bit. Thiết bị này còn bao gồm phương tiện thực hiện, dựa ít nhất một phần vào tập hợp thông số HRD đã chọn có thể áp dụng cho một điểm hoạt động cụ thể, hoạt động HRD trên tập hợp con dòng bit liên quan đến điểm hoạt động cụ thể.

Theo một ví dụ khác, sáng chế mô tả vật ghi dữ liệu đọc được bằng máy tính lưu trữ trên đó các lệnh mà, khi được thực hiện bởi một hoặc nhiều bộ xử lý của thiết bị, tạo cấu hình cho thiết bị để lựa chọn, từ trong số tập hợp thông số HRD trong VPS và tập hợp thông số HRD trong SPS, tập hợp thông số HRD có thể áp dụng cho một điểm hoạt động cụ thể của dòng bit. Ngoài ra, các lệnh này, khi được thực hiện, sẽ tạo cấu hình cho thiết bị để thực hiện, dựa ít nhất một phần vào tập hợp thông số HRD đã chọn có thể áp dụng cho một điểm hoạt động cụ thể, hoạt động HRD trên tập hợp con dòng bit liên quan đến điểm hoạt động cụ thể.

Mô tả chi tiết một hoặc nhiều ví dụ trong sáng chế được bộc lộ trong các hình vẽ đi kèm và phần mô tả dưới đây. Các dấu hiệu, mục đích và ưu điểm khác sẽ rõ ràng hơn từ phần mô tả, hình vẽ và yêu cầu bảo hộ.

Mô tả vắn tắt hình vẽ

FIG.1 là sơ đồ khối minh họa hệ thống mã hóa dữ liệu video làm ví dụ có thể sử dụng các kỹ thuật được mô tả trong sáng chế.

FIG.2 là sơ đồ khối minh họa bộ mã hóa dữ liệu video làm ví dụ có thể tiến hành các kỹ thuật được mô tả trong sáng chế.

FIG.3 là sơ đồ khối minh họa bộ giải mã video làm ví dụ có thể tiến hành các kỹ thuật được mô tả trong sáng chế.

FIG.4 là lưu đồ minh họa hoạt động làm ví dụ của thiết bị, theo một hoặc nhiều kỹ thuật theo sáng chế.

FIG.5 là lưu đồ minh họa hoạt động làm ví dụ của thiết bị, theo một hoặc nhiều kỹ thuật theo sáng chế.

FIG.6 là lưu đồ minh họa hoạt động bộ giải mã chuẩn giả định (HRD) làm ví dụ của thiết bị, theo một hoặc nhiều kỹ thuật theo sáng chế.

Mô tả chi tiết sáng chế

Bộ mã hóa dữ liệu video có thể tạo ra dòng bit mà bao gồm dữ liệu video mã hóa. Dòng bit có thể bao gồm chuỗi các đơn vị lớp trừu tượng mạng (network abstraction layer-NAL). Các đơn vị NAL của dòng bit có thể bao gồm các đơn vị NAL có lớp mã hóa dữ liệu video (video coding layer-VCL) và các đơn vị NAL không có VCL. Các đơn vị NAL có VCL có thể bao gồm các lát mã hóa của các hình ảnh. Đơn vị NAL không có VCL có thể bao gồm bộ thông số video (video parameter set-VPS), bộ thông số chuỗi (sequence parameter set-SPS), bộ thông số hình ảnh (picture parameter set-PPS), thông tin nâng cao phụ (supplemental enhancement information-SEI), hoặc các loại dữ liệu khác. VPS là một cấu trúc cú pháp mà có thể chứa các phần tử cú pháp áp dụng cho 0 hoặc nhiều chuỗi video mã hóa nguyên. SPS là một cấu trúc cú pháp mà có chứa các phần tử cú pháp áp dụng cho 0 hoặc nhiều chuỗi video mã hóa nguyên. VPS đơn có thể áp dụng được cho nhiều SPS. PPS là cấu trúc cú pháp có thể chứa các phần tử cú pháp áp dụng cho 0 hoặc nhiều hình ảnh mã hóa nguyên. SPS đơn có thể áp dụng cho nhiều PPS. Nói chung, VPS, SPS và PPS theo các khía cạnh khác nhau có thể được tạo thành, như được xác định bởi tiêu chuẩn HEVC.

Thiết bị, như thiết bị mạng phân phối nội dung (content delivery network-CDN), phần tử mạng nhận biết đa phương tiện (Media-Aware Network Element-MANE), hoặc bộ giải mã video, có thể tách dòng bit con từ dòng bit. Thiết bị này có thể tiến hành quy trình tách dòng bit con bằng cách loại bỏ các đơn vị NAL nhất định từ dòng bit. Dòng bit con thu được bao gồm các đơn vị NAL còn lại, không bị loại bỏ của dòng bit. Theo các ví dụ, dữ liệu video giải mã từ dòng bit con có thể có tốc độ khung thấp hơn và/hoặc có thể biểu diễn cảnh nhìn ít hơn so với dòng bit ban đầu.

Các tiêu chuẩn mã hóa dữ liệu video có thể bao gồm nhiều dấu hiệu khác nhau để hỗ trợ cho quy trình tách dòng bit con. Ví dụ, dữ liệu video của dòng bit có thể được chia thành tập hợp lớp. Đối với mỗi lớp trong số nhiều lớp, dữ liệu ở lớp dưới có thể được giải mã mà không tham chiếu đến dữ liệu ở lớp trên bất kỳ. Mỗi đơn vị NAL chỉ đóng gói giữ liệu của một lớp. Do đó, các đơn vị NAL đóng gói dữ liệu của lớp còn lại trên cùng của dòng bit có thể được loại bỏ khỏi dòng bit mà không làm ảnh hưởng đến khả năng giải mã dữ liệu trong các lớp dưới còn lại của dòng bit. Đối với mã hóa dữ liệu video có khả năng chuyển đổi cấp độ (scalable video coding-SVC), các lớp trên có thể bao gồm dữ liệu nâng cao cải thiện chất lượng hình ảnh ở lớp dưới (khả năng chuyển đổi cấp độ chất lượng), mở rộng định dạng không gian của hình ảnh ở lớp dưới (khả năng chuyển đổi cấp độ không gian), hoặc làm tăng tốc độ hình ảnh theo thời gian của hình ảnh ở lớp dưới (khả năng chuyển đổi cấp độ thời gian). Đối với mã hóa đa cảnh nhìn (multi-view coding-MVC) và mã hóa dữ liệu video ba chiều (three-dimensional video-3DV), các lớp trên có thể bao gồm các cảnh nhìn khác.

Các đơn vị NAL có thể bao gồm phần đầu và dung lượng hữu ích. Phần đầu của đơn vị NAL bao gồm phần tử cú pháp `nuh_reserved_zero_6bits`. Phần tử cú pháp `nuh_reserved_zero_6bits` của đơn vị NAL bằng 0 nếu đơn vị NAL liên quan đến lớp cơ sở đối với mã hóa đa cảnh nhìn, mã hóa 3DV, hoặc SVC. Dữ liệu trong lớp cơ sở của dòng bit có thể được giải mã mà không tham chiếu đến dữ liệu trong lớp khác bất kỳ của dòng bit. Nếu đơn vị NAL không liên quan đến lớp cơ sở đối với mã hóa đa cảnh nhìn, 3DV, hoặc SVC, thì phần tử cú pháp `nuh_reserved_zero_6bits` có thể có giá trị khác 0. Cụ thể là, nếu đơn vị NAL không có liên quan đến lớp cơ sở đối với mã hóa đa cảnh nhìn, 3DV, hoặc SVC, thì phần tử cú pháp `nuh_reserved_zero_6bits` của đơn vị NAL chỉ rõ ký hiệu nhận dạng lớp của đơn vị NAL.

Ngoài ra, một vài hình ảnh trong lớp có thể được giải mã mà không tham chiếu đến các hình ảnh khác trong lớp này. Do đó, các đơn vị NAL đóng gói dữ liệu của các hình ảnh nhất định của lớp có thể được loại bỏ khỏi dòng bit mà không làm ảnh hưởng đến khả năng giải mã các hình ảnh khác trong lớp. Ví dụ, các hình ảnh có giá trị số đếm thứ tự hình ảnh chẵn (picture order count-POC) có thể giải mã được mà không tham chiếu đến các hình ảnh có giá trị POC lẻ. Việc loại bỏ các đơn vị NAL đóng gói dữ liệu của hình ảnh này có thể làm giảm tốc độ khung của dòng bit. Một tập con các hình ảnh

trong lớp mà có thể được giải mã mà không tham chiếu đến các hình ảnh khác trong lớp này có thể được gọi trong bản mô tả này là lớp con.

Đơn vị NAL có thể bao gồm phần tử cú pháp `temporal_id`. Phần tử cú pháp `temporal_id` của đơn vị NAL chỉ rõ ký hiệu nhận dạng đơn vị NAL theo thời gian. Nếu ký hiệu nhận dạng đơn vị NAL thứ nhất theo thời gian ít hơn ký hiệu nhận dạng đơn vị NAL thứ hai theo thời gian, thì dữ liệu được đóng gói bởi đơn vị NAL thứ nhất có thể được giải mã mà không tham chiếu đến dữ liệu được đóng gói bởi đơn vị NAL thứ hai.

Mỗi điểm hoạt động của dòng bit gắn với một tập hợp ký hiệu nhận dạng lớp (tức là, tập hợp các giá trị `nuh_reserved_zero_6bits`) và ký hiệu nhận dạng theo thời gian. Tập hợp ký hiệu nhận dạng lớp có thể được ký hiệu là `OpLayerIdSet` và ký hiệu nhận dạng theo thời gian có thể được ký hiệu là `TemporalID`. Nếu ký hiệu nhận dạng lớp của đơn vị NAL nằm trong tập hợp ký hiệu nhận dạng lớp của điểm hoạt động và ký hiệu nhận dạng theo thời gian của đơn vị NAL nhỏ hơn hoặc bằng ký hiệu nhận dạng theo thời gian của điểm hoạt động, thì đơn vị NAL được gắn với điểm hoạt động. Dạng biểu diễn điểm hoạt động là tập con dòng bit gắn với điểm hoạt động. Dạng biểu diễn điểm hoạt động có thể bao gồm mỗi đơn vị NAL mà gắn với điểm hoạt động. Dạng biểu diễn điểm hoạt động không bao gồm các đơn vị NAL có VCL mà không gắn với điểm hoạt động.

Nguồn ngoài có thể chỉ rõ tập hợp các ký hiệu nhận dạng lớp đích cho điểm hoạt động. Ví dụ, thiết bị, như thiết bị CDN hoặc MANE, có thể chỉ rõ tập hợp các ký hiệu nhận dạng lớp đích. Theo ví dụ này, thiết bị có thể sử dụng tập hợp các ký hiệu nhận dạng lớp đích để nhận dạng điểm hoạt động. Sau đó, thiết bị có thể tách dạng biểu diễn điểm hoạt động của điểm hoạt động và gửi chuyển tiếp dạng biểu diễn điểm hoạt động, thay cho dòng bit ban đầu, đến thiết bị máy khách. Việc tách và gửi chuyển tiếp dạng biểu diễn điểm hoạt động đến thiết bị máy khách có thể làm giảm tốc độ bit của dòng bit.

Hơn thế nữa, các tiêu chuẩn mã hóa dữ liệu video chỉ rõ mô hình nhớ đệm video. Mô hình nhớ đệm video cũng có thể được gọi là “bộ giải mã chuẩn giả định” hoặc “HRD.” HRD mô tả cách dữ liệu được đưa vào bộ nhớ đệm để giải mã và cách dữ liệu giải mã được đưa vào bộ nhớ đệm để xuất ra. Ví dụ, HRD mô tả hoạt động của bộ nhớ đệm hình ảnh mã hóa (coded picture buffer-CPB) và bộ nhớ đệm hình ảnh giải mã (decoded picture buffer-DPB) trong bộ giải mã video. CPB là bộ nhớ đệm nhập trước xuất trước chứa các đơn vị truy nhập theo thứ tự giải mã được chỉ rõ bởi HRD. DPB là

bộ nhớ đệm giữ các hình ảnh giải mã để tham chiếu, sắp xếp lại thứ tự xuất, hoặc độ trễ xuất được chỉ rõ bởi HRD.

Bộ mã hóa dữ liệu video có thể báo hiệu bộ thông số HRD. Thông số HRD điều khiển các khía cạnh khác nhau của HRD. Thông số HRD bao gồm độ trễ lấy ra từ CPB ban đầu, dung lượng CPB, tốc độ bit, độ trễ xuất ra từ DPB ban đầu, và dung lượng DPB. Các thông số HRD này được mã hóa trong cấu trúc cú pháp `hrd_parameters()` được chỉ rõ trong VPS và/hoặc SPS. Thông số HRD cũng có thể được chỉ rõ trong thông báo thông tin nâng cao phụ chu kỳ nhớ đệm (supplemental enhancement information-SEI) hoặc thông báo SEI định thời hình ảnh.

Như được giải thích trên đây, dạng biểu diễn điểm hoạt động có thể có tốc độ khung và/hoặc tốc độ bit khác so với dòng bit ban đầu. Điều này là do dạng biểu diễn điểm hoạt động có thể không bao gồm một vài hình ảnh và/hoặc một vài dữ liệu của dòng bit ban đầu. Do đó, nếu bộ giải mã video là nhằm lấy dữ liệu từ CPB và/hoặc DPB ở tốc độ cụ thể khi xử lý dòng bit ban đầu và nếu bộ giải mã video là để lấy dữ liệu từ CPB và/hoặc DPB ở tỷ lệ này khi xử lý dạng biểu diễn điểm hoạt động, thì bộ giải mã video có thể loại bỏ quá nhiều hoặc quá ít dữ liệu khỏi CPB và/hoặc DPB. Do đó, bộ mã hóa dữ liệu video có thể báo hiệu các bộ thông số HRD khác nhau cho các điểm hoạt động khác nhau. Trong tiêu chuẩn mã hóa dữ liệu video hiệu suất cao hiện nay (HEVC), bộ mã hóa dữ liệu video có thể báo hiệu bộ thông số HRD trong VPS hoặc bộ mã hóa dữ liệu video có thể báo hiệu bộ thông số HRD trong SPS. Phiên bản dự thảo của tiêu chuẩn HEVC sắp tới, được gọi là “phiên bản dự thảo làm việc HEVC 8”, được mô tả trong tài liệu của Bross và cộng sự, “High Efficiency Video Coding (HEVC) text specification draft 8,” JCT-VC của ITU-T SG16 WP3 và ISO/IEC JTC1/SC29/WG11, cuộc họp thứ 10 tại Stockholm, Thụy Điển, tháng 7/ 2012, kể từ ngày 8/5/2013 có thể truy cập tại trang http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Sweden/wg11/JCTVC-J1003-v8.zip.

Theo một vài phiên bản của HEVC, chỉ bộ thông số HRD trong VPS được chọn cho các hoạt động HRD. Tức là, mặc dù các thông số HRD có thể được cung cấp trong SPS, nhưng bộ thông số HRD trong SPS không được chọn bởi bộ giải mã video HEVC cho các hoạt động HRD. Bộ giải mã video luôn phân tích cú pháp và giải mã VPS của dòng bit. Do đó, bộ giải mã video luôn phân tích cú pháp và giải mã bộ thông số HRD của VPS. Điều này là đúng bất kể liệu dòng bit có bao gồm các đơn vị NAL không phải

lớp cơ sở hay không. Do đó, nếu dòng bit bao gồm đơn vị NAL không phải lớp cơ sở, thì có thể lãng phí các tài nguyên tính toán để phân tích cú pháp và xử lý bộ thông số HRD trong SPS. Hơn thế nữa, nếu bộ thông số HRD có mặt trong VPS, thì bộ thông số HRD trong SPS có thể là bit lãng phí.

Theo các kỹ thuật của sáng chế, bộ mã hóa dữ liệu video có thể tạo ra dòng bit bao gồm SPS mà có thể áp dụng được cho chuỗi hình ảnh. SPS bao gồm bộ thông số HRD. Bộ thông số HRD có thể áp dụng được cho mỗi điểm hoạt động của dòng bit mà có tập hợp ký hiệu nhận dạng lớp phù hợp với tập hợp các ký hiệu nhận dạng lớp đích. Do đó, các bộ thông số HRD trong SPS không bị lãng phí, mà phần nào được sử dụng cho các hoạt động HRD. Ví dụ, thiết bị có thể chọn, từ trong số bộ thông số HRD trong VPS và bộ thông số HRD trong SPS, bộ thông số HRD áp dụng được cho điểm hoạt động cụ thể. Thiết bị có thể tiến hành, dựa ít nhất một phần vào bộ thông số HRD áp dụng được cho điểm hoạt động cụ thể, sự kiểm tra tương thích dòng bit để kiểm tra liệu tập con dòng bit gắn với điểm hoạt động cụ thể có tuân theo tiêu chuẩn mã hóa dữ liệu video hay không.

Thiết bị, như bộ mã hóa dữ liệu video, bộ giải mã video, hoặc loại thiết bị khác, như thiết bị CDN hoặc MANE, có thể tiến hành sự kiểm tra tương thích dòng bit trên dạng biểu diễn điểm hoạt động của điểm hoạt động. Sự kiểm tra tương thích dòng bit có thể xác minh rằng dạng biểu diễn điểm hoạt động tuân theo tiêu chuẩn mã hóa dữ liệu video, như HEVC. Như đã nêu trên, tập hợp các ký hiệu nhận dạng lớp đích và ký hiệu nhận dạng theo thời gian có thể được sử dụng để xác định điểm hoạt động. Tập hợp các ký hiệu nhận dạng lớp đích có thể được ký hiệu là “TargetDecLayerIdSet.” Ký hiệu nhận dạng theo thời gian có thể được ký hiệu là “TargetDecHighestTid.” Không chắc chắn là, phiên bản dự thảo làm việc HEVC 8 không chỉ rõ cách TargetDecLayerIdSet hoặc TargetDecHighestTid được đặt khi tiến hành sự kiểm tra tương thích dòng bit.

Theo một hoặc nhiều kỹ thuật theo sáng chế, thiết bị có thể tiến hành quy trình giải mã như là một phần của tiến hành sự kiểm tra tương thích dòng bit. Tiến hành quy trình giải mã bao gồm tiến hành quy trình tách dòng bit để giải mã, từ dòng bit, dạng biểu diễn điểm hoạt động của điểm hoạt động được xác định bởi tập hợp các ký hiệu nhận dạng lớp đích và ký hiệu nhận dạng theo thời gian cao nhất của đích. Tập hợp các ký hiệu nhận dạng lớp đích (tức là, TargetDecLayerIdSet) bao gồm các giá trị của các phần tử cú pháp nhận dạng lớp (ví dụ, phần tử cú pháp `nuh_reserved_zero_6bits`) có mặt trong dạng biểu diễn điểm hoạt động. Tập hợp các ký hiệu nhận dạng lớp đích là tập con các

giá trị của các phần tử cú pháp nhận dạng lớp của dòng bit. Ký hiệu nhận dạng theo thời gian cao nhất của đích (tức là, TargetDecHighestTid) bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dạng biểu diễn điểm hoạt động. Ký hiệu nhận dạng theo thời gian cao nhất của đích nhỏ hơn hoặc bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dòng bit. Tiến hành quy trình giải mã cũng có thể bao gồm giải mã đơn vị NAL của dạng biểu diễn điểm hoạt động.

Trong HEVC, SPS có thể bao gồm mảng phần tử cú pháp được ký hiệu là `sps_max_dec_pic_buffering[i]`, trong đó i nằm trong khoảng từ 0 đến số lớp thời gian tối đa trong dòng bit. `sps_max_dec_pic_buffering[i]` chỉ báo dung lượng tối đa cần thiết của DPB khi ký hiệu nhận dạng theo thời gian cao nhất (HighestTid) bằng i . `sps_max_dec_pic_buffering[i]` chỉ báo kích thước cần thiết của các đơn vị của bộ nhớ đệm lưu trữ hình ảnh.

Hơn thế nữa, trong HEVC, SPS có thể bao gồm mảng phần tử cú pháp được ký hiệu là `sps_max_num_reorder_pics[i]`, trong đó i nằm trong khoảng từ 0 đến số lớp thời gian tối đa trong dòng bit. `sps_max_num_reorder_pics[i]` chỉ báo số hình ảnh tối đa cho phép đứng trước hình ảnh bất kỳ theo thứ tự giải mã và đứng sau hình ảnh đó theo thứ tự xuất ra khi ký hiệu nhận dạng theo thời gian cao nhất (HighestTid) bằng i .

Trong HEVC, bộ thông số HRD có thể bao gồm mảng phần tử cú pháp được ký hiệu là `cpb_cnt_minus1[i]`, trong đó i nằm trong khoảng từ 0 đến số lớp thời gian tối đa trong dòng bit. `cpb_cnt_minus1[i]` chỉ rõ số lượng đặc tả CPB khác trong dòng bit của chuỗi video mã hóa khi ký hiệu nhận dạng theo thời gian cao nhất (HighestTid) bằng i , trong đó một đặc tả CPB khác được dùng để chỉ một hoạt động của CPB cụ thể với bộ thông số CPB cụ thể.

Trong phiên bản dự thảo làm việc HEVC 8, `sps_max_dec_pic_buffering[i]`, `sps_max_num_reorder_pics[i]`, và `cpb_cnt_minus1[i]` không thực sự được chọn trong các hoạt động HRD, các hoạt động tương thích dòng bit và giới hạn mức. Việc này ít nhất một phần là do phiên bản dự thảo làm việc HEVC 8 không chỉ rõ ý nghĩa ký hiệu nhận dạng theo thời gian cao nhất (HighestTid) là gì.

Theo một hoặc nhiều kỹ thuật theo sáng chế, thiết bị, như bộ mã hóa dữ liệu video, bộ giải mã video, hoặc thiết bị khác, có thể xác định ký hiệu nhận dạng theo thời gian cao nhất của tập con dòng bit gắn với điểm hoạt động được chọn của dòng bit. Hơn thế nữa, thiết bị có thể xác định, dựa vào ký hiệu nhận dạng theo thời gian cao nhất, phần

tử cú pháp cụ thể từ mảng phần tử cú pháp (ví dụ, `sps_max_dec_pic_buffering[ ]`, `sps_max_num_reorder_pics[ ]`, hoặc `cpb_cnt_minus1[ ]`). Thiết bị có thể tiến hành thao tác mà sử dụng phần tử cú pháp cụ thể để xác định sự tương thích của dòng bit với tiêu chuẩn mã hóa dữ liệu video hoặc để xác định sự tương thích của bộ giải mã video với tiêu chuẩn mã hóa dữ liệu video.

FIG.1 là sơ đồ khối minh họa hệ thống mã hóa dữ liệu video làm ví dụ 10 có thể sử dụng các kỹ thuật theo sáng chế. Như được sử dụng ở đây, thuật ngữ “bộ giải mã video” nói chung được dùng để chỉ cả bộ mã hóa dữ liệu video và bộ giải mã video. Theo sáng chế, thuật ngữ “mã hóa dữ liệu video” hoặc “mã hóa” có thể nói chung được dùng để chỉ mã hóa dữ liệu video hoặc giải mã video.

Như được thể hiện trên Fig.1, hệ thống mã hóa dữ liệu video 10 bao gồm thiết bị nguồn 12 và thiết bị đích 14. Thiết bị nguồn 12 tạo ra dữ liệu video mã hóa. Do đó, thiết bị nguồn 12 có thể được gọi là thiết bị mã hóa dữ liệu video hoặc cơ cấu mã hóa dữ liệu video. Thiết bị đích 14 có thể giải mã dữ liệu video mã hóa được tạo bởi thiết bị nguồn 12. Do đó, thiết bị đích 14 có thể được gọi là thiết bị giải mã video hoặc cơ cấu giải mã video. Thiết bị nguồn 12 và thiết bị đích 14 có thể là ví dụ của các thiết bị mã hóa dữ liệu video hoặc các cơ cấu mã hóa dữ liệu video.

Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm rất nhiều thiết bị, bao gồm máy tính để bàn, thiết bị tính toán di động, máy tính sổ tay (ví dụ, máy tính xách tay), máy tính bảng, hộp giải mã truyền hình, thiết bị điện thoại cầm tay như thiết bị gọi là máy điện thoại “thông minh”, máy thu hình, camera, thiết bị hiển thị, thiết bị phát đa phương tiện số, bàn giao tiếp trò chơi điện tử, máy tính trên xe, hoặc các thiết bị tương tự.

Thiết bị đích 14 có thể thu dữ liệu video mã hóa từ thiết bị nguồn 12 qua kênh 16. Kênh 16 có thể bao gồm một hoặc nhiều phương tiện hoặc thiết bị có khả năng chuyển dữ liệu video mã hóa từ thiết bị nguồn 12 đến thiết bị đích 14. Theo một ví dụ, kênh 16 có thể bao gồm một hoặc nhiều phương tiện truyền thông cho phép thiết bị nguồn 12 truyền dữ liệu video mã hóa trực tiếp đến thiết bị đích 14 theo thời gian thực. Theo ví dụ này, thiết bị nguồn 12 có thể điều biến dữ liệu video mã hóa theo tiêu chuẩn truyền thông, như giao thức truyền thông không dây, và có thể truyền dữ liệu video đã được điều biến đến thiết bị đích 14. Một hoặc nhiều phương tiện truyền thông có thể bao gồm phương tiện truyền thông không dây hoặc nối dây, như phổ tần số vô tuyến (RF) hoặc

một hoặc nhiều đường truyền vật lý. Một hoặc nhiều phương tiện truyền thông có thể tạo thành một phần của mạng nền truyền thông gói, như mạng cục bộ, mạng vùng rộng hoặc mạng toàn cầu (ví dụ, Internet). Một hoặc nhiều phương tiện truyền thông có thể bao gồm các bộ định tuyến, chuyển mạch, trạm cơ sở, hoặc thiết bị khác để tạo điều kiện truyền thông từ thiết bị nguồn 12 đến thiết bị đích 14.

Theo một ví dụ khác, kênh 16 có thể bao gồm phương tiện nhớ lưu trữ dữ liệu video mã hóa được tạo bởi thiết bị nguồn 12. Theo ví dụ này, thiết bị đích 14 có thể truy nhập phương tiện nhớ thông qua truy nhập đĩa hoặc truy nhập thẻ. Phương tiện nhớ có thể bao gồm nhiều loại phương tiện nhớ dữ liệu truy nhập tại chỗ khác nhau như đĩa Blu-ray, đĩa đa năng số (DVD - Digital Versatile Disc), CD-ROM, bộ nhớ tác động nhanh, hoặc các phương tiện nhớ số thích hợp khác để lưu trữ dữ liệu video mã hóa.

Theo một ví dụ khác, kênh 16 có thể bao gồm máy chủ tệp hoặc thiết bị nhớ trung gian khác để lưu trữ dữ liệu video mã hóa được tạo bởi thiết bị nguồn 12. Theo ví dụ này, thiết bị đích 14 có thể truy nhập dữ liệu video mã hóa lưu trữ ở máy chủ tệp hoặc thiết bị nhớ trung gian khác bằng cách truyền liên tục hoặc tải xuống. Máy chủ tệp có thể là kiểu máy chủ có khả năng lưu trữ dữ liệu video mã hóa và truyền dữ liệu video mã hóa đến thiết bị đích 14. Các máy chủ tệp làm ví dụ bao gồm máy chủ mạng (dùng cho website chẳng hạn), máy chủ giao thức chuyển tệp (FTP - File Transfer Protocol), thiết bị lưu trữ nối kết với mạng (NAS - Network Attached Storage), và các ổ đĩa cục bộ. Theo ví dụ nêu trong Fig.1, kênh 16 bao gồm thiết bị khác 21. Theo một vài ví dụ, thiết bị khác 21 là thiết bị CDN, MANE, hoặc loại thiết bị khác.

Thiết bị đích 14 có thể truy nhập dữ liệu video mã hóa qua kết nối dữ liệu tiêu chuẩn, như kết nối Internet. Các kiểu kết nối dữ liệu làm ví dụ có thể bao gồm kênh không dây (ví dụ, kết nối Wi-Fi), kết nối dây (ví dụ, đường thuê bao số (DSL - Digital Subscriber Line), môđem cáp, v.v.), hoặc các tổ hợp của chúng thích hợp để truy nhập dữ liệu video mã hóa lưu trữ trong máy chủ tệp. Truyền dữ liệu video mã hóa từ máy chủ tệp có thể là truyền liên tục, truyền dữ liệu tải xuống hoặc kết hợp cả hai kiểu này.

Các kỹ thuật theo sáng chế không bị giới hạn ở các ứng dụng hoặc các thiết lập không dây. Các kỹ thuật này có thể được áp dụng cho việc mã hóa dữ liệu video nhằm hỗ trợ cho nhiều ứng dụng đa phương tiện như phát sóng truyền hình qua không khí, truyền hình cáp, truyền hình vệ tinh, truyền video liên tục, qua Internet chẳng hạn, mã hóa dữ liệu video để lưu trữ trong phương tiện nhớ dữ liệu, giải mã dữ liệu video lưu trữ trong

phương tiện nhớ dữ liệu, hoặc các ứng dụng khác. Theo một số ví dụ, hệ thống mã hóa dữ liệu video 10 có thể được tạo cấu hình để hỗ trợ truyền video một chiều hoặc hai chiều nhằm trợ giúp cho các ứng dụng như truyền video liên tục, phát lại video, phát rộng video, và/hoặc điện thoại truyền hình.

FIG.1 chỉ là một ví dụ và các kỹ thuật theo sáng chế có thể áp dụng cho các thiết lập mã hóa dữ liệu video (ví dụ, mã hóa dữ liệu video hoặc giải mã video) mà không cần thiết bao gồm truyền thông dữ liệu giữa các thiết bị mã hóa và giải mã. Theo các ví dụ khác, dữ liệu được tìm kiếm từ bộ nhớ cục bộ, được truyền liên tục qua mạng, hoặc tương tự. Thiết bị mã hóa video có thể mã hóa và lưu trữ dữ liệu vào bộ nhớ, và/hoặc thiết bị giải mã video có thể tìm kiếm và giải mã dữ liệu từ bộ nhớ. Theo nhiều ví dụ, quy trình mã hóa và giải mã được tiến hành bởi các thiết bị không truyền thông với nhau, mà chỉ mã hóa dữ liệu vào bộ nhớ và/hoặc tìm kiếm và giải mã dữ liệu từ bộ nhớ.

Theo ví dụ trên Fig.1, thiết bị nguồn 12 bao gồm nguồn video 18, bộ mã hóa dữ liệu video 20, và giao diện xuất 22. Trong một số ví dụ, giao diện xuất 22 có thể bao gồm bộ điều biến/bộ giải điều biến (môđem) và/hoặc bộ truyền. Nguồn video 18 có thể bao gồm thiết bị thu nạp video, ví dụ, camera video, kho chứa video chứa dữ liệu video đã thu nạp trước đó, giao diện cung cấp video để nhận dữ liệu video từ nhà cung cấp nội dung video, và/hoặc hệ thống đồ họa máy tính để tạo lập dữ liệu video, hoặc tổ hợp của các nguồn dữ liệu video này.

Bộ mã hóa dữ liệu video 20 có thể mã hóa dữ liệu video từ nguồn video 18. Theo một vài ví dụ, thiết bị nguồn 12 trực tiếp truyền dữ liệu video mã hóa đến thiết bị đích 14 qua giao diện xuất 22. Theo một vài ví dụ khác, dữ liệu video mã hóa cũng có thể được lưu trữ trong phương tiện nhớ hoặc máy chủ tệp để thiết bị đích 14 truy nhập sau đó để giải mã và/hoặc phát lại.

Theo ví dụ trên Fig.1, thiết bị đích 14 bao gồm giao diện nhập 28, bộ giải mã video 30, và thiết bị hiển thị 32. Trong một số ví dụ, giao diện nhập 28 bao gồm bộ thu và/hoặc môđem. Giao diện nhập 28 có thể thu dữ liệu video mã hóa qua kênh 16. Thiết bị hiển thị 32 có thể được tích hợp hoặc có thể gắn ngoài với thiết bị đích 14. Nói chung, thiết bị hiển thị 32 hiển thị dữ liệu video giải mã. Thiết bị hiển thị 32 có thể bao gồm nhiều loại thiết bị hiển thị khác nhau như màn hình tinh thể lỏng (LCD - Liquid Crystal Display), màn hình plasma, màn hình điốt phát quang hữu cơ (OLED - Organic Light Emitting Diode), hoặc kiểu thiết bị hiển thị khác.

Mỗi bộ mã hóa dữ liệu video 20 và bộ giải mã video 30 có thể được tiến hành dưới dạng mạch bất kỳ trong nhiều loại mạch thích hợp như một hoặc nhiều bộ vi xử lý, bộ xử lý tín hiệu số (DSP - Digital Signal Processor), mạch tích hợp chuyên dụng (ASIC - Application Specific Integrated Circuit), mảng cổng lập trình được bằng trường (FPGA - Field Programmable Gate Array), mạch logic rời rạc, phần cứng, hoặc các tổ hợp bất kỳ của chúng. Nếu các kỹ thuật này được tiến hành một phần bằng phần mềm, thì thiết bị có thể lưu trữ các lệnh cho phần mềm trong phương tiện nhớ đọc được bằng máy tính bền vững thích hợp và có thể thực thi các lệnh này trong phần cứng bằng cách sử dụng một hoặc nhiều bộ xử lý để tiến hành các kỹ thuật theo sáng chế. Bất kỳ một trong số các bộ phận nêu trên (bao gồm phần cứng, phần mềm, tổ hợp của phần cứng và phần mềm, v.v.) có thể được coi là một hoặc nhiều bộ xử lý. Mỗi bộ mã hóa dữ liệu video 20 và bộ giải mã video 30 có thể được gộp trong một hoặc nhiều bộ mã hóa hoặc bộ giải mã, có thể được tích hợp thành một phần của bộ mã hóa/giải mã kết hợp (CODEC) trong thiết bị tương ứng.

Nói chung, sáng chế có thể đề cập đến bộ mã hóa dữ liệu video 20 “báo hiệu” thông tin nhất định đến thiết bị khác, như bộ giải mã video 30 hoặc thiết bị khác 21. Nói chung, thuật ngữ “báo hiệu” có thể dùng để chỉ việc truyền thông phần tử cú pháp và/hoặc dữ liệu khác được sử dụng để giải mã dữ liệu video nén. Việc truyền thông này có thể xảy ra theo thời gian thực hoặc gần thực. Theo cách khác, việc truyền thông này có thể xảy ra trong một khoảng thời gian, như có thể xảy ra khi lưu trữ các phần tử cú pháp vào phương tiện nhớ đọc được bằng máy tính trong dòng bit mã hóa tại thời điểm mã hóa, sau đó có thể được tìm kiếm bởi thiết bị giải mã tại thời điểm bất kỳ sau khi được lưu trữ vào phương tiện này.

Theo một vài ví dụ, bộ mã hóa dữ liệu video 20 và bộ giải mã video 30 hoạt động theo tiêu chuẩn nén video, như ISO/IEC MPEG-4 Visual và ITU-T H.264 (cũng được biết đến là ISO/IEC MPEG-4 AVC), kể cả phiên bản mở rộng của mã hóa dữ liệu video có khả năng chuyển đổi cấp độ (Scalable Video Coding-SVC), phiên bản mở rộng của mã hóa dữ liệu video đa cảnh nhìn (Multiview Video Coding-MVC), và/hoặc phiên bản mở rộng 3DV dựa vào MVC. Trong một vài trường hợp, dòng bit bất kỳ tương thích với 3DV dựa vào MVC luôn chứa dòng bit con mà phù hợp với profin MVC, ví dụ profin cao cấp stereo. Hơn thế nữa, nỗ lực tạo ra phiên bản mở rộng mã hóa dữ liệu video ba chiều (3DV) cho H.264/AVC, được gọi là 3DV dựa vào AVC vẫn đang diễn ra. Theo các

ví dụ khác, bộ mã hóa dữ liệu video 20 và bộ giải mã video 30 có thể hoạt động theo ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 hoặc ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual, và ITU-T H.264, ISO/IEC Visual.

Theo các ví dụ khác, bộ mã hóa dữ liệu video 20 và bộ giải mã video 30 có thể hoạt động theo tiêu chuẩn mã hóa dữ liệu video hiệu suất cao (High Efficiency Video Coding-HEVC) hiện nay được phát triển bởi nhóm hợp tác chung về mã hóa dữ liệu video (Joint Collaboration Team on Video Coding-JCT-VC) trực thuộc nhóm chuyên gia mã hóa dữ liệu video (Video Coding Experts Group-VCEG) của ITU-T và nhóm chuyên gia về hình ảnh chuyển động (Motion Picture Experts Group-MPEG) của ISO/IEC. Phiên bản dự thảo của chuẩn HEVC sắp tới, được gọi là “phiên bản dự thảo làm việc HEVC 9” được mô tả trong tài liệu của Bross và cộng sự, “High Efficiency Video Coding (HEVC) text specification draft 9,” JCT-VC của ITU-T SG16 WP3 và ISO/IEC JTC1/SC29/WG11, cuộc họp thứ 11 tại Thượng Hải, Trung Quốc, tháng 10/2012, kể từ ngày 8/5/2013 có thể truy cập tại trang http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v13.zip. Hơn thế nữa, các nỗ lực tạo ra SVC, mã hóa đa cảnh nhìn, và các phiên bản mở rộng 3DV của HEVC vẫn đang diễn ra. Phiên bản mở rộng 3DV của HEVC có thể được dùng để chỉ là 3DV dựa vào HEVC hoặc 3D-HEVC.

Trong HEVC và các tiêu chuẩn mã hóa dữ liệu video khác, chuỗi video thường bao gồm chuỗi hình ảnh. Hình ảnh cũng có thể được gọi là “khung.” Hình ảnh có thể bao gồm ba mảng mẫu, được gọi là S_L , S_{Cb} và S_{Cr} . S_L là mảng hai chiều (tức là, khối) của các mẫu độ chói. S_{Cb} là mảng hai chiều của các mẫu màu Cb. S_{Cr} là mảng hai chiều của các mẫu màu Cr. Các mẫu màu cũng có thể được gọi ở đây là các mẫu “màu”. Trong các trường hợp khác, hình ảnh có thể là đơn màu và có thể chỉ bao gồm mảng các mẫu độ chói.

Để tạo ra dạng biểu diễn mã hóa của hình ảnh, bộ mã hóa dữ liệu video 20 có thể tạo ra một tập hợp đơn vị cây mã hóa (coding tree unit-CTU). Mỗi CTU có thể là khối cây mã hóa của các mẫu độ chói, hai khối cây mã hóa tương ứng của mẫu màu, và các cấu trúc cú pháp dùng để mã hóa các mẫu của khối cây mã hóa. Khối cây mã hóa có thể là khối $N \times N$ các mẫu. CTU cũng có thể được gọi là “khối cây” hoặc “đơn vị mã hóa lớn nhất” (largest coding unit-LCU). CTU của HEVC có thể gần như tương tự với khối macrô của các tiêu chuẩn khác, như H.264/AVC. Tuy nhiên, CTU không nhất thiết bị

giới hạn ở kích thước cụ thể và có thể bao gồm một hoặc nhiều đơn vị mã hóa (coding unit-CU). Lát có thể bao gồm một số nguyên CTU được sắp xếp theo thứ tự liên tiếp khi quét từng đường.

Để tạo ra CTU mã hóa, bộ mã hóa dữ liệu video 20 có thể tiến hành đệ quy việc phân tách cây tứ phân trên khối cây mã hóa của CTU để chia khối cây mã hóa thành các khối mã hóa, từ đó gọi là “đơn vị cây mã hóa.” Khối mã hóa là khối $N \times N$ các mẫu. Một CU có thể là khối mã hóa của các mẫu độ chói và hai khối mã hóa tương ứng của mẫu màu của hình ảnh có mảng mẫu độ chói, mảng mẫu Cb và mảng mẫu Cr, và các cấu trúc cú pháp được dùng để mã hóa các mẫu khối mã hóa. Bộ mã hóa dữ liệu video 20 có thể phân tách khối mã hóa của CU thành một hoặc nhiều khối dự báo. Khối dự báo có thể là khối mẫu hình chữ nhật (tức là, hình vuông hoặc không phải hình vuông) mà được áp dụng cùng một loại dự báo. Đơn vị dự báo (prediction unit-PU) của CU có thể là khối dự báo của các mẫu độ chói, hai khối dự báo tương ứng của các mẫu màu của hình ảnh, và các cấu trúc cú pháp được dùng để dự báo các mẫu khối dự báo. Bộ mã hóa dữ liệu video 20 có thể tạo ra các khối độ chói, Cb và Cr dự báo cho các khối dự báo độ chói, Cb và Cr của mỗi PU của CU.

Bộ mã hóa dữ liệu video 20 có thể sử dụng thao tác dự báo nội cấu trúc hoặc dự báo liên cấu trúc để tạo ra các khối dự báo cho PU. Nếu bộ mã hóa dữ liệu video 20 sử dụng thao tác dự báo nội cấu trúc để tạo ra các khối dự báo cho PU, thì bộ mã hóa dữ liệu video 20 có thể tạo ra các khối dự báo cho PU dựa vào các mẫu giải mã của hình ảnh gắn với PU.

Nếu bộ mã hóa dữ liệu video 20 sử dụng thao tác dự báo liên cấu trúc để tạo ra các khối dự báo cho PU, thì bộ mã hóa dữ liệu video 20 có thể tạo ra các khối dự báo cho PU dựa vào các mẫu giải mã của một hoặc nhiều hình ảnh ngoài hình ảnh gắn với PU. Bộ mã hóa dữ liệu video 20 có thể sử dụng thao tác dự báo một hướng hoặc thao tác dự báo hai hướng để tạo ra các khối dự báo cho PU. Nếu bộ mã hóa dữ liệu video 20 sử dụng thao tác dự báo một hướng để tạo ra các khối dự báo cho PU, thì PU có thể có một vector chuyển động. Nếu bộ mã hóa dữ liệu video 20 sử dụng thao tác dự báo hai hướng để tạo ra các khối dự báo cho PU, thì PU có thể có hai vector chuyển động.

Sau khi bộ mã hóa dữ liệu video 20 tạo ra các khối dự báo độ chói, Cb và Cr cho một hoặc nhiều PU của CU, thì bộ mã hóa dữ liệu video 20 có thể tạo ra khối dư độ chói cho CU. Mỗi mẫu trong khối dư độ chói của CU chỉ báo độ chênh lệch giữa mẫu độ chói

trong một trong số các khối độ chói dự báo của CU và mẫu tương ứng trong khối mã hóa độ chói ban đầu của CU. Ngoài ra, bộ mã hóa dữ liệu video 20 có thể tạo ra khối dư Cb cho CU. Mỗi mẫu trong khối dư Cb của CU có thể chỉ báo độ chênh lệch giữa mẫu Cb trong một trong số các khối Cb dự báo của CU và mẫu tương ứng trong khối mã hóa Cb ban đầu của CU. Bộ mã hóa dữ liệu video 20 cũng có thể tạo ra khối dư Cr cho CU. Mỗi mẫu trong khối dư Cr của CU có thể chỉ báo độ chênh lệch giữa mẫu Cr trong một trong số các khối Cr dự báo của CU và mẫu tương ứng trong khối mã hóa Cr ban đầu của CU.

Hơn thế nữa, bộ mã hóa dữ liệu video 20 có thể sử dụng kỹ thuật phân tách cây từ phân để phân tích các khối dư độ chói, Cb và Cr của CU thành một hoặc nhiều khối biến đổi độ chói, Cb và Cr. Khối biến đổi của thể là khối mẫu hình chữ nhật mà được áp dụng cùng một kỹ thuật biến đổi. Đơn vị biến đổi (transform unit-TU) của CU có thể là khối mẫu độ chói biến đổi, hai khối biến đổi tương ứng của mẫu màu, và các cấu trúc cú pháp được dùng để biến đổi các mẫu khối biến đổi. Do đó, mỗi TU của CU có thể được gắn với khối biến đổi độ chói, khối biến đổi Cb, và khối biến đổi Cr. Khối biến đổi độ chói gắn với TU có thể là khối con của khối dư độ chói của CU. Khối biến đổi Cb có thể là khối con của khối dư Cb của CU. Khối biến đổi Cr có thể là khối con của khối dư Cr của CU.

Bộ mã hóa dữ liệu video 20 có thể áp dụng một hoặc nhiều kỹ thuật biến đổi cho khối biến đổi độ chói của TU để tạo ra khối hệ số độ chói cho TU. Khối hệ số có thể là mảng hai chiều của các hệ số biến đổi. Hệ số biến đổi có thể là lượng vô hướng. Bộ mã hóa dữ liệu video 20 có thể áp dụng một hoặc nhiều kỹ thuật biến đổi cho khối biến đổi Cb của TU để tạo ra khối hệ số Cb cho TU. Bộ mã hóa dữ liệu video 20 có thể áp dụng một hoặc nhiều kỹ thuật biến đổi cho khối biến đổi Cr của TU để tạo ra khối hệ số Cr cho TU.

Sau khi tạo ra khối hệ số (ví dụ, khối hệ số độ chói, khối hệ số Cb hoặc khối hệ số Cr), bộ mã hóa dữ liệu video 20 có thể lượng tử hóa khối hệ số. Lượng tử hóa thường được dùng để chỉ quy trình trong đó các hệ số biến đổi được lượng tử hóa để có thể làm giảm lượng dữ liệu dùng để biểu diễn hệ số biến đổi, cho phép nén hơn nữa. Sau khi bộ mã hóa dữ liệu video 20 lượng tử hóa khối hệ số, bộ mã hóa dữ liệu video 20 có thể mã hóa entropy các phần tử cú pháp chỉ báo hệ số biến đổi đã lượng tử hóa. Ví dụ, bộ mã hóa dữ liệu video 20 có thể tiến hành kỹ thuật mã hóa số học nhị phân thích ứng ngữ cảnh (Context-Adaptive Binary Arithmetic Coding-CABAC) trên các phần tử cú pháp

chỉ báo hệ số biến đổi đã lượng tử hóa. Bộ mã hóa dữ liệu video 20 có thể xuất các phần tử cú pháp đã mã hóa entropy trong dòng bit.

Bộ mã hóa dữ liệu video 20 có thể xuất dòng bit chứa chuỗi bit mà tạo ra dạng biểu diễn của hình ảnh mã hóa và dữ liệu gắn kèm. Dòng bit có thể bao gồm chuỗi các đơn vị lớp trừu tượng mạng (network abstraction layer-NAL). Đơn vị NAL có thể là cấu trúc cú pháp chứa chỉ báo về loại dữ liệu đi kèm và số byte chứa dữ liệu đó ở dạng phần tải tin chuỗi byte thô (raw byte sequence payload-RBSP) được đệm thêm khi cần các byte chống mô phỏng. Tức là, mỗi đơn vị NAL có thể bao gồm phần đầu đơn vị NAL và đóng gói RBSP. Phần đầu đơn vị NAL có thể bao gồm phần tử cú pháp chỉ báo mã loại đơn vị NAL. Mã loại đơn vị NAL được chỉ rõ bởi phần đầu đơn vị NAL của đơn vị NAL chỉ báo loại đơn vị NAL. RBSP có thể là cấu trúc cú pháp chứa một số nguyên byte được đóng gói trong đơn vị NAL. Trong một vài trường hợp, RBSP bao gồm 0 bit.

Các loại đơn vị NAL khác nhau có thể đóng gói các loại RBSP khác nhau. Ví dụ, loại đơn vị NAL thứ nhất có thể đóng gói RBSP cho bộ thông số hình ảnh (picture parameter set-PPS), loại đơn vị NAL thứ hai có thể đóng gói RBSP cho lát mã hóa, loại đơn vị NAL thứ ba có thể đóng gói RBSP cho SEI, và v.v.. Các đơn vị NAL đóng gói RBSP cho dữ liệu mã hóa dữ liệu video (trái ngược với RBSP cho bộ thông số và thông báo SEI) có thể được gọi là đơn vị NAL có lớp mã hóa dữ liệu video (video coding layer-VCL).

Bộ giải mã video 30 có thể thu dòng bit được tạo ra bởi bộ mã hóa dữ liệu video 20. Ngoài ra, bộ giải mã video 30 có thể phân tích cú pháp dòng bit để giải mã các phần tử cú pháp từ dòng bit. Bộ giải mã video 30 có thể phục hồi lại các hình ảnh của dữ liệu video dựa ít nhất một phần vào các phần tử cú pháp được giải mã từ dòng bit. Quy trình phục hồi lại dữ liệu video có thể thường ngược với quy trình được tiến hành bởi bộ mã hóa dữ liệu video 20. Ví dụ, bộ giải mã video 30 có thể sử dụng các vector chuyển động của PU để xác định các khối dự báo cho PU của CU hiện thời. Ngoài ra, bộ giải mã video 30 có thể lượng tử hóa ngược các khối hệ số biến đổi gắn với TU của CU hiện thời. Bộ giải mã video 30 có thể tiến hành các kỹ thuật biến đổi ngược trên các khối hệ số biến đổi để phục hồi lại các khối biến đổi gắn với các TU của CU hiện thời. Bộ giải mã video 30 có thể phục hồi lại các khối mã hóa của CU hiện thời bằng cách bổ sung các mẫu của các khối dự báo cho PU của CU hiện thời vào các mẫu tương ứng của các khối biến đổi

của TU của CU hiện thời. Bằng cách phục hồi lại các khối mã hóa cho mỗi CU của hình ảnh, bộ giải mã video 30 có thể phục hồi lại hình ảnh.

Trong mã hóa đa cảnh nhìn, có nhiều khung hình trong cùng một cảnh nhìn từ các điểm nhìn khác nhau. Thuật ngữ “đơn vị truy nhập” được dùng để chỉ tập hợp hình ảnh tương ứng với cùng một thời điểm. Do đó, dữ liệu video có thể được khái niệm hóa là chuỗi các đơn vị truy nhập xảy ra theo thời gian. “Thành phần cảnh nhìn” có thể là dạng biểu diễn mã hóa của cảnh nhìn trong một đơn vị truy nhập. Theo sáng chế, “cảnh nhìn” có thể được dùng để chỉ chuỗi các thành phần cảnh nhìn gắn với cùng một ký hiệu nhận dạng cảnh nhìn.

Mã hóa đa cảnh nhìn hỗ trợ cho dự báo liên cảnh nhìn. Dự báo liên cảnh nhìn là tương tự với dự báo liên cấu trúc được sử dụng trong H.264/AVC và HEVC và có thể sử dụng các phần tử cú pháp giống nhau. Tuy nhiên, khi bộ mã hóa dữ liệu video tiến hành thao tác dự báo liên cảnh nhìn trên đơn vị video hiện thời (như PU), thì bộ mã hóa dữ liệu video 20 có thể sử dụng hình ảnh mà nằm trong cùng một đơn vị truy nhập với đơn vị video hiện thời làm hình ảnh chuẩn, nhưng trong cảnh nhìn khác. Ngược lại, thao tác dự báo liên cấu trúc thông thường chỉ sử dụng các hình ảnh trong các đơn vị truy nhập khác nhau làm hình ảnh chuẩn.

Trong mã hóa đa cảnh nhìn, cảnh nhìn có thể được dùng để chỉ là “cảnh nhìn cơ sở” nếu bộ giải mã video (ví dụ, bộ giải mã video 30) có thể giải mã các hình ảnh trong cảnh nhìn mà không tham chiếu các hình ảnh trong cảnh nhìn khác bất kỳ. Khi mã hóa hình ảnh trong một trong số các cảnh nhìn không phải cơ sở, bộ giải mã video (như bộ mã hóa dữ liệu video 20 hoặc bộ giải mã video 30) có thể bổ sung hình ảnh vào danh mục hình ảnh chuẩn nếu hình ảnh này nằm trong cảnh nhìn khác nhưng ở cùng một thời điểm (tức là, đơn vị truy nhập) như hình ảnh mà bộ mã hóa dữ liệu video hiện đang mã hóa. Giống như các hình ảnh chuẩn dự báo liên cấu trúc khác, bộ mã hóa dữ liệu video có thể gắn hình ảnh chuẩn dự báo liên cảnh nhìn vào vị trí bất kỳ trong danh mục hình ảnh chuẩn.

Tiêu chuẩn mã hóa dữ liệu video chỉ rõ các mô hình nhớ đệm dữ liệu video. Trong H.264/AVC và HEVC, mô hình nhớ đệm được gọi là “bộ giải mã chuẩn giả định” hoặc “HRD.” Trong phiên bản dự thảo làm việc HEVC 8, HRD được mô tả trong phụ lục C.

HRD mô tả cách dữ liệu được đưa vào bộ nhớ đệm để giải mã và cách dữ liệu đã giải mã được đưa vào bộ nhớ đệm để xuất ra. Ví dụ, HRD mô tả hoạt động của bộ nhớ

đệm hình ảnh mã hóa (coded picture buffer-CPB), bộ nhớ đệm hình ảnh giải mã (decoded picture buffer-DPB), và bộ giải mã video. CPB là bộ nhớ đệm nhập trước xuất trước chứa các đơn vị truy nhập theo trật tự giải mã được chỉ rõ bởi HRD. DPB là bộ nhớ đệm chứa các hình ảnh giải mã để tham chiếu, sắp xếp lại trật tự xuất, hoặc độ trễ xuất được chỉ rõ bởi HRD. Các biến diễn của CPB và DPB có thể được chỉ rõ bằng toán học. HRD có thể trực tiếp áp các ràng buộc về thời gian, dung lượng bộ nhớ đệm, và tốc độ bit. Hơn thế nữa, HRD có thể gián tiếp áp các ràng buộc về các đặc điểm và thống kê dòng bit khác nhau.

Trong H.264/AVC và HEVC, sự tương thích dòng bit và sự tương thích bộ giải mã được chỉ rõ là các phần của đặc tả HRD. Nói cách khác, mô hình HRD chỉ rõ các phép kiểm tra để xác định liệu dòng bit có tuân theo tiêu chuẩn hay không và các phép kiểm tra để xác định liệu bộ giải mã có tuân theo tiêu chuẩn hay không. Mặc dù HRD được gọi là một loại bộ giải mã nào đó, nhưng bộ mã hóa dữ liệu video thường sử dụng HRD để đảm bảo sự tương thích dòng bit, trong khi đó bộ giải mã video thường không cần đến HRD.

Cả H.264/AVC và HEVC đều chỉ rõ hai loại tương thích dòng bit hoặc HRD, được gọi là loại I và loại II. Dòng bit loại I là dòng đơn vị NAL chỉ chứa các đơn vị NAL có VCL và đơn vị NAL chứa dữ liệu đệm cho tất cả các đơn vị truy nhập trong dòng bit. Dòng bit loại II là dòng đơn vị NAL chứa ít nhất một trong số các đơn vị sau, ngoài các đơn vị NAL có VCL và đơn vị NAL chứa dữ liệu đệm cho tất cả các đơn vị truy nhập trong dòng bit: đơn vị NAL khác không có VCL ngoài đơn vị NAL chứa dữ liệu đệm; và toàn bộ các phần tử cú pháp `leading_zero_8bits`, `zero_byte`, `start_coded_prefix_one_3bytes`, và `trailing_zero_8bits` tạo nên dòng byte từ dòng đơn vị NAL.

Khi thiết bị tiến hành sự kiểm tra tương thích dòng bit để xác định liệu dòng bit có tuân theo tiêu chuẩn mã hóa dữ liệu video hay không, thì thiết bị này có thể chọn điểm hoạt động trong dòng bit. Sau đó, thiết bị này có thể xác định bộ thông số HRD áp dụng được cho điểm hoạt động đã chọn. Thiết bị này có thể sử dụng bộ thông số HRD áp dụng được cho điểm hoạt động đã chọn để tạo cấu hình biến diễn của HRD. Cụ thể hơn là, thiết bị này có thể sử dụng bộ thông số HRD áp dụng được để tạo cấu hình các biến diễn của các thành phần cụ thể của HRD, như bộ lập lịch dòng giả thiết (hypothetical stream

scheduler-HSS), CPB, quy trình giải mã, DPB, và v.v.. Sau đó, HSS có thể đưa dữ liệu video mã hóa của dòng bit vào CPB của HRD theo lịch biểu cụ thể.

Hơn thế nữa, thiết bị này có thể gọi ra quy trình giải mã để giải mã dữ liệu video mã hóa trong CPB. Quy trình giải mã này có thể xuất các hình ảnh giải mã đến DPB. Vì thiết bị này chuyển dữ liệu thông qua HRD, nên thiết bị này có thể xác định được liệu còn tập hợp ràng buộc cụ thể nào đáp ứng hay không. Ví dụ, thiết bị có thể xác định được liệu điều kiện tràn hoặc tràn dưới có xảy ra trong CPB hoặc DPB hay không trong khi HRD giải mã dạng biểu diễn điểm hoạt động của điểm hoạt động được chọn. Thiết bị này có thể chọn và xử lý mỗi điểm hoạt động của dòng bit theo cách này. Nếu không có điểm hoạt động nào của dòng bit vi phạm các ràng buộc, thì thiết bị này có thể xác định rằng dòng bit tuân theo tiêu chuẩn mã hóa dữ liệu video.

Cả H.264/AVC và HEVC chỉ rõ hai loại tương thích bộ giải mã, được gọi là tương thích bộ giải mã theo thời gian xuất và tương thích bộ giải mã theo thứ tự xuất ra. Bộ giải mã yêu cầu tương thích với profin cụ thể, bậc và mức có thể giải mã thành công tất cả các dòng bit mà tuân theo các yêu cầu sự tương thích dòng bit của tiêu chuẩn mã hóa dữ liệu video, như HEVC. Theo sáng chế, “profin” có thể được dùng để chỉ tập con cú pháp dòng bit. “Bậc” và “mức” có thể được chỉ rõ trong mỗi profin. Mức của bậc có thể là một tập hợp các ràng buộc cụ thể được áp trên các giá trị của các phần tử cú pháp trong dòng bit. Các ràng buộc này có thể là các giới hạn đơn giản về các giá trị. Theo cách khác, các ràng buộc này có thể là dạng các ràng buộc về tổ hợp số học của các giá trị (ví dụ, độ rộng hình ảnh nhân với chiều cao hình ảnh nhân với số hình ảnh được giải mã mỗi giây). Mức được chỉ rõ cho bậc dưới bị ràng buộc nhiều hơn mức được chỉ rõ cho bậc trên.

Khi thiết bị tiến hành sự kiểm tra tương thích bộ giải mã để xác định liệu bộ giải mã kiểm tra (decoder under test-DUT) có tuân theo tiêu chuẩn mã hóa dữ liệu video hay không, thiết bị có thể cung cấp dòng bit mà tuân theo tiêu chuẩn mã hóa dữ liệu video cho cả HRD và DUT. HRD có thể xử lý dòng bit theo cách được mô tả trên đây theo sự kiểm tra tương thích dòng bit. Thiết bị có thể xác định rằng DUT tuân theo tiêu chuẩn mã hóa dữ liệu video nếu thứ tự các hình ảnh giải mã xuất ra bởi DUT so khớp với thứ tự các hình ảnh giải mã được xuất ra bởi HRD. Hơn thế nữa, thiết bị có thể xác định rằng DUT tuân theo tiêu chuẩn mã hóa dữ liệu video nếu thời gian mà DUT xuất ra các hình ảnh giải mã so khớp với thời gian mà HRD xuất ra các hình ảnh giải mã.

Ngoài các kiểm tra tương thích dòng bit và kiểm tra tương thích bộ giải mã, các thiết bị có thể sử dụng các thông số HRD cho các mục đích khác. Ví dụ, độ trễ lấy ra từ CPB ban đầu có thể được sử dụng để chỉ dẫn hệ thống thiết lập độ trễ truyền dẫn từ đầu này đến đầu kia ban đầu thích hợp và các thời điểm xuất từ DPB có thể được sử dụng để tìm ra dấu thời gian giao thức thời gian thực (real-time protocol-RTP) khi dòng bit chứa dữ liệu video được vận chuyển qua RTP.

Trong các mô hình H.264/AVC và HEVC HRD, quy trình giải mã hoặc lấy ra từ CPB có thể được dựa trên đơn vị truy nhập. Tức là, HRD được giả sử là giải mã các đơn vị truy nhập hoàn toàn ở một thời điểm và lấy các đơn vị truy nhập hoàn toàn từ CPB. Hơn thế nữa, trong các mô hình H.264/AVC và HEVC HRD, giả sử là quy trình giải mã hình ảnh là tức thì. Bộ mã hóa dữ liệu video 20 có thể báo hiệu các thời điểm giải mã để bắt đầu giải mã các đơn vị truy nhập trong các thông báo SEI định thời hình ảnh. Trong các ứng dụng thực tế, nếu bộ giải mã video tương thích tuân thủ chặt chẽ các thời điểm giải mã được báo hiệu để bắt đầu giải mã các đơn vị truy nhập, thời điểm sớm nhất có thể để xuất hình ảnh đã giải mã cụ thể là bằng thời điểm giải mã hình ảnh cụ thể đó cộng với thời gian cần để giải mã hình ảnh cụ thể đó. Tuy nhiên, thực tế, thời điểm cần để giải mã hình ảnh không thể bằng 0.

Các thông số HRD có thể kiểm soát các khía cạnh khác nhau của HRD. Nói cách khác, HRD có thể dựa vào các thông số HRD. Các thông số HRD có thể bao gồm độ trễ lấy ra từ CPB ban đầu, dung lượng CPB, tốc độ bit, độ trễ xuất ra từ DPB ban đầu, và dung lượng DPB. Bộ mã hóa dữ liệu video 20 có thể báo hiệu các thông số HRD này trong cấu trúc cú pháp `hrd_parameters()` được chỉ rõ trong bộ thông số video (VPS) và/hoặc bộ thông số chuỗi (SPS). Mỗi VPS và/hoặc SPS có thể bao gồm nhiều cấu trúc cú pháp `hrd_parameters()` cho các bộ thông số HRD khác nhau. Theo một vài ví dụ, bộ mã hóa dữ liệu video 20 có thể báo hiệu các thông số HRD trong các thông báo SEI về chu kỳ nhớ đệm hoặc thông báo SEI định thời hình ảnh.

Như được giải thích trên đây, điểm hoạt động của dòng bit gắn với tập hợp ký hiệu nhận dạng lớp (tức là, tập hợp các giá trị `nuh_reserved_zero_6bits`) và ký hiệu nhận dạng theo thời gian. Dạng biểu diễn điểm hoạt động có thể bao gồm mỗi đơn vị NAL gắn với điểm hoạt động. Dạng biểu diễn điểm hoạt động có thể có tốc độ khung và/hoặc tốc độ bit khác so với dòng bit ban đầu. Điều này là do dạng biểu diễn điểm hoạt động có thể không bao gồm một vài hình ảnh và/hoặc một vài dữ liệu của dòng bit ban đầu. Do đó,

nếu bộ giải mã video 30 được dùng để lấy dữ liệu từ CPB và/hoặc DPB ở tốc độ cụ thể khi xử lý dòng bit ban đầu và nếu bộ giải mã video 30 được dùng để lấy dữ liệu từ CPB và/hoặc DPB ở cùng tốc độ khi xử lý dạng biểu diễn điểm hoạt động, thì bộ giải mã video 30 có thể lấy quá nhiều hoặc quá ít dữ liệu từ CPB và/hoặc DPB. Do đó, bộ mã hóa dữ liệu video 20 có thể báo hiệu các bộ thông số HRD khác nhau cho các điểm hoạt động khác nhau. Ví dụ, bộ mã hóa dữ liệu video 20 có thể bao gồm nhiều cấu trúc cú pháp `hrd_parameters()` chứa các thông số HRD cho các điểm hoạt động khác nhau trong VPS.

Trong phiên bản dự thảo làm việc HEVC 8, bộ thông số HRD tùy ý bao gồm tập hợp thông tin mà chung cho tất cả các lớp con. Nói cách khác, bộ thông số HRD có thể tùy ý bao gồm tập hợp các phần tử cú pháp chung áp dụng được cho các điểm hoạt động mà bao gồm lớp con thời gian bất kỳ. Lớp con thời gian là lớp có khả năng chuyển đổi cấp độ theo thời gian của dòng bit có khả năng chuyển đổi cấp độ theo thời gian bao gồm các đơn vị NAL có VCL với giá trị cụ thể của `TemporalId` và đơn vị NAL không có VCL gắn kèm. Ngoài tập hợp thông tin chung, các bộ thông số HRD có thể bao gồm tập hợp các phần tử cú pháp riêng cho từng lớp con thời gian. Ví dụ, cấu trúc cú pháp `hrd_parameters()` có thể tùy ý bao gồm tập hợp thông tin mà chung cho tất cả các lớp con và luôn bao gồm thông tin riêng cho lớp con. Do tập hợp thông tin chung là chung cho nhiều bộ thông số HRD, nên có thể không cần báo hiệu tập hợp thông tin chung trong nhiều bộ thông số HRD. Đúng hơn là, trong phiên bản dự thảo làm việc HEVC 8, thông tin chung có thể có mặt trong bộ thông số HRD khi bộ thông số HRD là bộ thông số HRD thứ nhất trong VPS hoặc thông tin chung có thể có mặt trong bộ thông số HRD khi bộ thông số HRD gắn với chỉ số điểm hoạt động thứ nhất. Ví dụ, phiên bản dự thảo làm việc HEVC 8 hỗ trợ cho sự có mặt của thông tin chung khi cấu trúc cú pháp `hrd_parameters()` là cấu trúc cú pháp `hrd_parameters()` thứ nhất trong VPS hoặc khi cấu trúc cú pháp `hrd_parameters()` gắn với điểm hoạt động thứ nhất.

Bảng 1, dưới đây, là cấu trúc cú pháp làm ví dụ cho cấu trúc cú pháp `hrd_parameters()` trong HEVC.

Bảng 1 – Các thông số HRD

hrd_parameters(commonInfPresentFlag, MaxNumSubLayersMinus1) {	Bộ tả	mô
if(commonInfPresentFlag) {		
timing_info_present_flag	u(1)	
if(timing_info_present_flag) {		
num_units_in_tick	u(32)	
time_scale	u(32)	
}		
nal_hrd_parameters_present_flag	u(1)	
vcl_hrd_parameters_present_flag	u(1)	
if(nal_hrd_parameters_present_flag vcl_hrd_parameters_present_flag){		
sub_pic_cpb_params_present_flag	u(1)	
if(sub_pic_cpb_params_present_flag) {		
tick_divisor_minus2	u(8)	
du_cpb_removal_delay_length_minus1	u(5)	
}		
bit_rate_scale	u(4)	
cpb_size_scale	u(4)	
initial_cpb_removal_delay_length_minus1	u(5)	
cpb_removal_delay_length_minus1	u(5)	
dpb_output_delay_length_minus1	u(5)	
}		
}		
for(i = 0; i <= MaxNumSubLayersMinus1; i++) {		
fixed_pic_rate_flag[i]	u(1)	
if(fixed_pic_rate_flag[i])		
pic_duration_in_tc_minus1[i]	ue(v)	
low_delay_hrd_flag[i]	u(1)	
cpb_cnt_minus1[i]	ue(v)	
if(nal_hrd_parameters_present_flag)		
sub_layer_hrd_parameters(i)		
if(vcl_hrd_parameters_present_flag)		
sub_layer_hrd_parameters(i)		
}		
}		

Theo ví dụ trong bảng 1, trên đây, và các bảng cú pháp khác trong sáng chế, các phần tử cú pháp có loại bộ mô tả $ue(v)$ có thể là các số nguyên không dấu độ dài thay đổi được mã hóa bằng quy trình mã hóa Golomb (Exp-Golomb) số mũ thứ tự thứ 0 với bit bên trái trước. Theo ví dụ trong bảng 1 và các bảng dưới đây, phần tử cú pháp có bộ mô tả dạng $u(n)$, trong đó n là một số nguyên không âm, là các giá trị không dấu của độ dài n .

Trong cú pháp làm ví dụ trong bảng 1, các phần tử cú pháp trong khối “if (commonInfPresentFlag) { ... }” là thông tin chung của các bộ thông số HRD. Nói cách khác, thông tin chung của bộ thông số HRD có thể bao gồm các phần tử cú pháp `timing_info_present_flag`, `num_units_in_tick`, `time_scale`, `nal_hrd_parameters_present_flag`, `vcl_hrd_parameters_present_flag`, `sub_pic_cpb_params_present_flag`, `tick_divisor_minus2`, `du_cpb_removal_delay_length_minus1`, `bit_rate_scale`, `cpb_size_scale`, `initial_cpb_removal_delay_length_minus1`, `cpb_removal_delay_length_minus1`, và `dpb_output_delay_length_minus1`.

Hơn thế nữa, theo ví dụ trong bảng 1, các phần tử cú pháp `fixed_pic_rate_flag[i]`, `pic_duration_in_tc_minus1[i]`, `low_delay_hrd_flag[i]`, và `cpb_cnt_minus1[i]` có thể là tập hợp các thông số HRD riêng cho lớp con. Nói cách khác, các phần tử cú pháp này của cấu trúc cú pháp `hrd_parameters()` chỉ có thể áp dụng được cho các điểm hoạt động mà bao gồm lớp con riêng. Do đó, các thông số HRD của cấu trúc cú pháp `hrd_parameters()` có thể bao gồm, ngoài thông tin chung tùy ý được bao gồm, bộ thông số HRD riêng cho lớp con mà riêng cho lớp con cụ thể của dòng bit.

Phần tử cú pháp `fixed_pic_rate_flag[i]` có thể chỉ báo rằng, khi `HighestTid` bằng i , thì khoảng cách thời gian giữa các thời điểm xuất ra từ HRD của hai hình ảnh liên tiếp bất kỳ theo thứ tự xuất bị ràng buộc theo cách cụ thể. `HighestTid` có thể là biến mà xác định lớp con thời gian cao nhất (ví dụ, của điểm hoạt động). Phần tử cú pháp `pic_duration_in_tc_minus1[i]` có thể chỉ rõ, khi `HighestTid` bằng i , khoảng cách thời gian, tính theo nhịp đồng hồ, giữa các thời điểm xuất ra từ HRD của các hình ảnh liên tiếp bất kỳ theo thứ tự xuất trong chuỗi video mã hóa. Phần tử cú pháp `low_delay_hrd_flag[i]` có thể chỉ rõ kiểu hoạt động HRD, khi `HighestTid` bằng i , như được chỉ rõ trong phụ lục C của phiên bản dự thảo làm việc HEVC 8. Phần tử cú pháp

`cpb_cnt_minus1[i]` có thể chỉ rõ số lượng đặc tả CPB khác trong dòng bit của chuỗi video mã hóa khi `HighestTid` bằng `i`.

Bộ mã hóa dữ liệu video 20 có thể sử dụng các thông báo SEI để bao gồm, trong dòng bit, siêu dữ liệu không cần giải mã chính xác các giá trị mẫu của hình ảnh. Tuy nhiên, bộ giải mã video 30 hoặc các thiết bị khác có thể sử dụng siêu dữ liệu được bao gồm trong các thông báo SEI cho nhiều mục đích khác nhau. Ví dụ, bộ giải mã video 30 có thể sử dụng siêu dữ liệu trong các thông báo SEI để định thời xuất hình ảnh, hiển thị hình ảnh, phát hiện dữ liệu bị mất, và che giấu lỗi.

Bộ mã hóa dữ liệu video 20 có thể bao gồm một hoặc nhiều đơn vị NAL SEI trong đơn vị truy nhập. Nói cách khác, số đơn vị NAL SEI bất kỳ có thể được gắn với đơn vị truy nhập. Hơn thế nữa, mỗi đơn vị NAL SEI có thể chứa một hoặc nhiều thông báo SEI. Tiêu chuẩn HEVC mô tả cú pháp và ngữ nghĩa cho nhiều loại thông báo SEI khác nhau. Tuy nhiên, tiêu chuẩn HEVC không mô tả việc xử lý các thông báo SEI bởi vì các thông báo SEI không làm ảnh hưởng đến quy trình giải mã thông thường. Một lý do để có các thông báo SEI trong tiêu chuẩn HEVC là để cho phép dữ liệu bổ sung được diễn dịch giống nhau trong các hệ thống sử dụng HEVC khác nhau. Các đặc tả và các hệ thống sử dụng HEVC có thể cần các bộ mã hóa dữ liệu video để tạo lập các thông báo SEI nhất định hoặc có thể xác định được cách xử lý riêng các loại thông báo SEI thu được cụ thể. Bảng 2, dưới đây, liệt kê các thông báo SEI được chỉ rõ trong HEVC và mô tả vắn tắt mục đích của chúng.

Bảng 2- Khái quát các thông báo SEI

Thông báo SEI	Mục đích
Chu kỳ đưa vào bộ nhớ đệm	Các độ trễ ban đầu cho hoạt động của bộ giải mã chuẩn giải định (hypothetical reference decoder-HRD)
Định thời hình ảnh	Thời gian xuất ra hình ảnh và thời gian lấy ra hình ảnh/hình ảnh con cho hoạt động HRD
Hình chữ nhật quét dọc	Hiển thị ở khuôn dạng hình ảnh (picture aspect ratio-PAR) khác nhau so với PAR của các hình ảnh xuất ra
Phần tải tin điền đầy	Điều chỉnh tốc độ bit để đáp ứng các ràng buộc cụ thể
Dữ liệu người dùng đăng ký Dữ liệu người dùng không đăng ký	Các thông báo SEI được xác định bởi các thực thể bên ngoài
Điểm phục hồi	Thông tin bổ sung cho sự truy nhập ngẫu nhiên sạch. Làm sạch quy trình giải mã từ từ.
Thông tin cảnh nhìn	Thông tin về các thay đổi và các chỗ chuyển tiếp cảnh nhìn
Chụp nhanh khung hoàn chỉnh	Chỉ báo để đánh dấu hình ảnh giải mã liên quan là chụp nhanh ảnh tĩnh của nội dung video này
Phân đoạn lọc tiến triển	Chỉ báo rằng các nh liên tiếp nhất định biểu diễn việc lọc tiến triển chất lượng của ảnh hơn là cảnh chuyển động
Các đặc điểm độ hạt phim	Cho phép các bộ giải mã tổng hợp độ hạt phim
Ưu tiên hiển thị lọc mở khối	Khuyến cáo liệu các ảnh hiển thị có nên trải qua quá trình lọc mở khối trong vòng lặp
Dấu vết sau bộ lọc	Cung cấp các hệ số sau bộ lọc đề nghị hoặc thông tin tương quan cho thiết kế sau bộ lọc
Thông tin ánh xạ tín hiệu số	Ánh xạ lại đến không gian sắc màu khác so với sắc màu được sử dụng hoặc được giả thiết trong mã hóa
Sắp xếp gói khung	Gói video lập thể vào thành chuỗi bit HEVC
Hướng hiển thị	Chỉ định lật và/hoặc quay mà nên được áp dụng cho các ảnh xuất khi chúng được hiển thị
Chỉ báo trường	Cung cấp thông tin liên quan đến nội dung video xen kẽ và/hoặc mã hóa trường, ví dụ chỉ báo liệu ảnh này là khung tăng lên, trường hoặc khung chứa hai trường xen kẽ
Băm ảnh được giải mã	Tổng kiểm tra ảnh giải mã, mà có thể được sử dụng để phát hiện lỗi
Định thời ảnh con	Thời gian lấy ra ảnh con cho hoạt động g của HDR
Các bộ thông số hoạt động	Cung cấp các thông tin về VPS, SPS hoạt động, v.v...
Cấu trúc mô tả các ảnh	Mô tả cấu trúc dự báo liên cấu trúc và thời gian của chuỗi bit này

Có một vài vấn đề hoặc thiếu sót ở các kỹ thuật hiện có để báo hiệu thông số HRD và chọn các thông số HRD và các thông số khác hiện nay. Ví dụ, trong phiên bản dự thảo làm việc HEVC 8, chỉ các bộ thông số HRD trong VPS được chọn cho các hoạt động HRD. Tức là, mặc dù các thông số HRD có thể được cung cấp trong SPS, nhưng các bộ thông số HRD trong SPS không được chọn bởi các bộ giải mã video HEVC cho các hoạt động HRD. Bộ giải mã video luôn phân tích cú pháp và giải mã VPS của dòng bit. Do đó, bộ giải mã video luôn phân tích cú pháp và giải mã các bộ thông số HRD của VPS.

Điều này là đúng bất kể liệu dòng bit có bao gồm các đơn vị NAL không phải lớp cơ sở hay không. Ví dụ, chỉ cấu trúc cú pháp `hrd_parameters()` được mã hóa trong VPS có thể được chọn cho các hoạt động HRD, và cấu trúc cú pháp `hrd_parameters()` có thể có trong SPS có thể không bao giờ được chọn. Điều này có thể cần phân tích cú pháp và xử lý VPS, ngay cả khi giải mã dòng bit không chứa `nuh_reserved_zero_6bits` lớn hơn 0 (tức là, dòng bit chỉ chứa lớp cơ sở trong phiên bản mở rộng đa cảnh nhìn, 3DV, hoặc SVC của HEVC).

Do đó, nếu dòng bit bao gồm đơn vị NAL không phải lớp cơ sở, có thể lãng phí các tài nguyên tính toán để phân tích cú pháp và xử lý các bộ thông số HRD trong SPS. Hơn thế nữa, nếu các bộ thông số HRD có mặt trong VPS, thì các bộ thông số HRD trong SPS có thể là các bit bị lãng phí. Ví dụ, nếu cấu trúc cú pháp `hrd_parameters()` có mặt trong SPS, thì các bit mã hóa của cấu trúc cú pháp này có thể chỉ là sự lãng phí bit.

Theo một hoặc nhiều kỹ thuật theo sáng chế, bộ mã hóa dữ liệu video 20 có thể tạo lập dòng bit chứa SPS áp dụng được cho chuỗi hình ảnh. SPS bao gồm bộ thông số HRD. Bộ thông số HRD có thể áp dụng được cho mỗi điểm hoạt động của dòng bit mà có tập hợp ký hiệu nhận dạng lớp so khớp với tập hợp các ký hiệu nhận dạng lớp đích. Do đó, các bộ thông số HRD trong SPS không bị lãng phí, nhưng phần nào đó có thể được sử dụng cho các hoạt động HRD. Ví dụ, các điểm hoạt động mà cấu trúc cú pháp `hrd_parameters()` được mã hóa trong SPS có thể được xác định rõ ràng, ví dụ là các điểm hoạt động mà chỉ một giá trị của `nuh_reserved_zero_6bits` (tức là, lớp ID trong phiên bản mở rộng mã hóa dữ liệu video đa cảnh nhìn, 3DV hoặc có khả năng chuyển đổi cấp độ) có mặt trong dòng bit.

Ví dụ, thiết bị, như bộ mã hóa dữ liệu video 20 hoặc bộ giải mã video 30, có thể chọn, từ trong số bộ thông số HRD trong bộ thông số video và bộ thông số HRD trong SPS, bộ thông số HRD áp dụng được cho điểm hoạt động cụ thể. Theo ví dụ này, thiết bị có thể tiến hành, dựa ít nhất một phần vào bộ thông số HRD áp dụng được cho điểm hoạt động cụ thể, sự kiểm tra tương thích dòng bit để kiểm tra liệu tập con dòng bit gắn với điểm hoạt động cụ thể có tuân theo tiêu chuẩn mã hóa dữ liệu video hay không. Sự kiểm tra tương thích dòng bit có thể xác minh rằng dạng biểu diễn điểm hoạt động tuân theo tiêu chuẩn mã hóa dữ liệu video, như HEVC.

Theo sáng chế, điểm hoạt động có thể được xác định bằng tập hợp các giá trị `nuh_reserved_zero_6bits values`, được ký hiệu là `OpLayerIdSet`, và giá trị `TemporalId`, được ký hiệu là `OpTid`. Tập con dòng bit gắn kèm thu được ở dạng đầu ra của quy trình tách dòng bit con như được chỉ rõ trong điểm 10.1 của phiên bản dự thảo làm việc HEVC 8 với `OpTid` và `OpLayerIdSet` dưới dạng các đầu vào là có thể giải mã độc lập. Điểm 10.1 của phiên bản dự thảo làm việc HEVC 8 mô tả thao tác tách dòng bit con (tức là, dạng biểu diễn điểm hoạt động) từ dòng bit. Cụ thể là, điểm 10.1 của phiên bản dự thảo làm việc HEVC 8 cho biết rằng dòng bit con thu được bằng cách lấy tất cả các đơn vị NAL có ký hiệu nhận dạng theo thời gian (ví dụ, `TemporalID`) lớn hơn `tIdTarget` hoặc các ký hiệu nhận dạng lớp (ví dụ, `nuh_reserved_zero_6bits`) không phải trong số các giá trị trong `targetDecLayerIdSet` từ dòng bit. `tIdTarget` và `targetDecLayerIdSet` là các thông số của quy trình tách dòng bit.

Đối với vấn đề hoặc thiếu sót làm ví dụ khác ở các kỹ thuật báo hiệu thông số HRD hiện nay, thiết bị, như bộ mã hóa dữ liệu video, bộ giải mã video, hoặc loại thiết bị khác, có thể tiến hành sự kiểm tra tương thích dòng bit trên dạng biểu diễn điểm hoạt động của điểm hoạt động. Như đã nêu trên, tập hợp các ký hiệu nhận dạng lớp đích và ký hiệu nhận dạng theo thời gian có thể được sử dụng để xác định điểm hoạt động. Tập hợp các ký hiệu nhận dạng lớp đích có thể được ký hiệu là “`TargetDecLayerIdSet`.” Ký hiệu nhận dạng theo thời gian có thể được ký hiệu là “`TargetDecHighestTid`.” Không chắc chắn là, phiên bản dự thảo làm việc HEVC 8 không chỉ rõ cách `TargetDecLayerIdSet` hoặc `TargetDecHighestTid` được đặt khi tiến hành sự kiểm tra tương thích dòng bit. Ví dụ, khi quy trình giải mã được gọi ra để kiểm tra sự tương thích dòng bit, các ngữ nghĩa của phần tử cú pháp không được xác định rõ ràng khi các giá trị của `TargetDecLayerIdSet` và `TargetDecHighestTid` không được đặt thích hợp.

Một hoặc nhiều kỹ thuật theo sáng chế cho biết cách đặt TargetDecLayerIdSet và TargetDecHighestTid khi tiến hành sự kiểm tra tương thích dòng bit. Ví dụ, quy trình giải mã chung cho dòng bit (hoặc dạng biểu diễn điểm hoạt động) được cải biến sao cho nếu dòng bit (hoặc dạng biểu diễn điểm hoạt động) được giải mã khi kiểm tra sự tương thích dòng bit, thì TargetDecLayerIdSet được đặt như được chỉ rõ trong điểm C.1 của tiêu chuẩn HEVC. Tương tự, quy trình giải mã chung cho dòng bit (hoặc dạng biểu diễn điểm hoạt động) có thể được cải biến sao cho nếu dòng bit (hoặc dạng biểu diễn điểm hoạt động) được giải mã khi kiểm tra sự tương thích dòng bit, thì TargetDecHighestTid được đặt như được chỉ rõ trong điểm C.1 của phiên bản dự thảo làm việc HEVC 8. Nói cách khác, thiết bị có thể xác định được tập hợp ký hiệu nhận dạng lớp đích của điểm hoạt động cụ thể mà chứa mỗi ký hiệu nhận dạng lớp có mặt trong tập con dòng bit và tập hợp ký hiệu nhận dạng lớp của điểm hoạt động cụ thể là tập con của ký hiệu nhận dạng lớp có mặt trong dòng bit. Ngoài ra, thiết bị có thể xác định ký hiệu nhận dạng theo thời gian đích của điểm hoạt động cụ thể bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong tập con dòng bit và ký hiệu nhận dạng theo thời gian đích của điểm hoạt động cụ thể là nhỏ hơn hoặc bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dòng bit.

Trong điểm C.1 của phiên bản dự thảo làm việc HEVC 8, TargetDecLayerIdSet được đặt bằng targetOpLayerIdSet. targetOpLayerIdSet chứa tập hợp các giá trị cho nuh_reserved_zero_6bits có mặt trong dạng biểu diễn điểm hoạt động của điểm hoạt động kiểm tra. targetOpLayerIdSet là tập con các giá trị của nuh_reserved_zero_6bits có mặt trong dòng bit kiểm tra.

Hơn thế nữa, biến TargetDecHighestTid xác định lớp con thời gian cao nhất cần được giải mã. Lớp con thời gian là lớp có khả năng chuyển đổi cấp độ theo thời gian của dòng bit có khả năng chuyển đổi cấp độ theo thời gian chứa các đơn vị NAL có VCL với giá trị cụ thể của TemporalId và các đơn vị NAL không có VCL đi kèm. Trong điểm C.1 của tiêu chuẩn HEVC, TargetDecHighestTid được đặt bằng targetOpTid. targetOpTid là bằng temporal_id lớn nhất có mặt trong dạng biểu diễn điểm hoạt động của điểm hoạt động kiểm tra và nhỏ hơn hoặc bằng temporal_id lớn nhất có mặt trong dòng bit kiểm tra. Do đó, khi quy trình giải mã được gọi ra để kiểm tra tương thích dòng bit, các giá trị của TargetDecLayerIdSet và TargetDecHighestTid được đặt bằng tập hợp các giá trị

nuh_reserved_zero_6bits và giá trị TemporalId lớn nhất có mặt trong dòng bit con tương ứng với điểm hoạt động kiểm tra để kiểm tra sự tương thích dòng bit cụ thể.

Theo cách này, thiết bị (như bộ mã hóa dữ liệu video 20, bộ giải mã video 30, thiết bị khác 21, hoặc thiết bị khác) có thể, theo một hoặc nhiều kỹ thuật theo sáng chế, tiến hành quy trình giải mã như một phần của việc tiến hành kiểm tra tương thích dòng bit. Tiến hành quy trình giải mã có thể bao gồm tiến hành quy trình tách dòng bit để tách dạng biểu diễn điểm hoạt động của điểm hoạt động được xác định bởi tập hợp các ký hiệu nhận dạng lớp đích và ký hiệu nhận dạng theo thời gian cao nhất của đích từ dòng bit. Tập hợp các ký hiệu nhận dạng lớp đích (tức là, TargetDecLayerIdSet) bao gồm các giá trị của các phần tử cú pháp nhận dạng lớp (ví dụ, các phần tử cú pháp nuh_reserved_zero_6bits) có mặt trong dạng biểu diễn điểm hoạt động. Tập hợp các ký hiệu nhận dạng lớp đích là tập con các giá trị của các phần tử cú pháp nhận dạng lớp của dòng bit. Ký hiệu nhận dạng theo thời gian cao nhất của đích (tức là, TargetDecHighestTid) bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dạng biểu diễn điểm hoạt động. Ký hiệu nhận dạng theo thời gian cao nhất của đích là nhỏ hơn hoặc bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dòng bit. Tiến hành quy trình giải mã cũng bao gồm giải mã đơn vị NAL của dạng biểu diễn điểm hoạt động.

Quy trình giải mã không phải lúc nào cũng được tiến hành như một phần của sự tiến hành kiểm tra tương thích dòng bit. Đúng hơn là, quy trình giải mã có thể là quy trình chung để giải mã dòng bit. Khi quy trình giải mã không được tiến hành như là một phần của sự kiểm tra tương thích dòng bit, thì nguồn ngoài có thể chỉ rõ TargetDecLayerIdSet và TargetDecHighestTid cho điểm hoạt động. Nguồn ngoài có thể là nguồn thông tin bất kỳ bên ngoài dòng bit. Ví dụ, thiết bị CDN có thể xác định bằng lập trình và chỉ rõ TargetDecLayerIdSet và TargetDecHighestTid dựa vào cấu hình của bộ giải mã video cụ thể. Thiết bị tiến hành quy trình giải mã có thể sử dụng TargetDecLayerIdSet và TargetDecHighestTid được chỉ rõ bên ngoài để tách dạng biểu diễn điểm hoạt động từ dòng bit. Sau đó, thiết bị tiến hành quy trình giải mã có thể giải mã các đơn vị NAL của dạng biểu diễn điểm hoạt động được tách.

Do đó, khi quy trình giải mã không được tiến hành như một phần của sự kiểm tra tương thích dòng bit, thiết bị tiến hành quy trình giải mã có thể thu tập hợp các ký hiệu nhận dạng lớp đích và ký hiệu nhận dạng theo thời gian cao nhất của đích từ nguồn ngoài. Tập hợp các ký hiệu nhận dạng lớp đích bao gồm các giá trị của các phần tử cú

pháp nhận dạng lớp có mặt trong dạng biểu diễn điểm hoạt động. Ký hiệu nhận dạng theo thời gian cao nhất của đích bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dạng biểu diễn điểm hoạt động thứ hai. Hơn thế nữa, thiết bị tiến hành quy trình giải mã có thể tiến hành quy trình tách dòng bit để tách dạng biểu diễn điểm hoạt động từ dòng bit. Sau đó, thiết bị tiến hành quy trình giải mã có thể giải mã đơn vị NAL của dạng biểu diễn điểm hoạt động.

Trong các trường hợp khác, nguồn ngoài không chỉ rõ TargetDecLayerIdSet hoặc TargetDecHighestTid. Trong các trường hợp này, quy trình giải mã có thể được tiến hành trên toàn bộ dòng bit. Ví dụ, thiết bị có thể tiến hành quy trình tách dòng bit để tách dạng biểu diễn điểm hoạt động từ dòng bit. Theo ví dụ này, 0 là giá trị duy nhất của các phần tử cú pháp nhận dạng lớp (tức là, `nuh_reserved_zero_6bits`) có mặt trong dạng biểu diễn điểm hoạt động. Hơn thế nữa, theo ví dụ này, ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dòng bit bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dạng biểu diễn điểm hoạt động. Theo ví dụ này, thiết bị tiến hành quy trình giải mã có thể giải mã các đơn vị NAL của dạng biểu diễn điểm hoạt động.

Như đã nêu trên, SPS có thể bao gồm mảng phần tử cú pháp được ký hiệu là `sps_max_dec_pic_buffering[i]`, trong đó i nằm trong khoảng từ 0 đến số lớp thời gian tối đa trong dòng bit. `sps_max_dec_pic_buffering[i]` chỉ báo dung lượng tối đa cần thiết của DPB khi ký hiệu nhận dạng theo thời gian cao nhất (HighestTid) bằng i . `sps_max_dec_pic_buffering[i]` chỉ báo kích thước cần thiết của các đơn vị của bộ nhớ đệm lưu trữ hình ảnh. Hơn thế nữa, SPS có thể bao gồm mảng phần tử cú pháp được ký hiệu là `sps_max_num_reorder_pics[i]`, trong đó i nằm trong khoảng từ 0 đến số lớp thời gian tối đa trong dòng bit. `sps_max_num_reorder_pics[i]` chỉ báo số hình ảnh tối đa được cho phép đứng trước hình ảnh bất kỳ theo thứ tự giải mã và đứng sau hình ảnh đó theo thứ tự xuất ra khi ký hiệu nhận dạng theo thời gian cao nhất (HighestTid) bằng i . Ngoài ra, bộ thông số HRD có thể bao gồm mảng phần tử cú pháp được ký hiệu là `cpb_cnt_minus1[i]`, trong đó i nằm trong khoảng từ 0 đến số lớp thời gian tối đa trong dòng bit. `cpb_cnt_minus1[i]` chỉ rõ số đặc tả CPB khác trong dòng bit của chuỗi video mã hóa khi ký hiệu nhận dạng theo thời gian cao nhất (HighestTid) bằng i .

Bởi vì phiên bản dự thảo làm việc HEVC 8 không chỉ rõ nghĩa ký hiệu nhận dạng theo thời gian cao nhất (HighestTid) là gì, nên phiên bản dự thảo làm việc HEVC 8, `sps_max_dec_pic_buffering[i]`, `sps_max_num_reorder_pics[i]`, và `cpb_cnt_minus1[i]`

không được chọn thích hợp trong các hoạt động HRD, các thao tác tương thích dòng bit, và các giới hạn mức. Nói cách khác, các thông số `sps_max_num_reorder_pics[i]`, `sps_max_dec_pic_buffering[i]`, và `cpb_cnt_minus1[i]` trong các hoạt động HRD, các yêu cầu về sự tương thích dòng bit và giới hạn mức không được chọn thích hợp.

Theo một hoặc nhiều kỹ thuật theo sáng chế, `sps_max_dec_pic_buffering[i]` được xác định sao cho `sps_max_dec_pic_buffering[i]` chỉ báo dung lượng tối đa cần thiết của DPB khi `TargetDecHighestTid` bằng `i`. `TargetDecHighestTid` được xác định theo cách được mô tả trên đây. Điều này có thể trái ngược với phiên bản dự thảo làm việc HEVC 8, trong đó `HighestTid` không được xác định. Giá trị của `sps_max_dec_pic_buffering[i]` sẽ nằm trong khoảng bao hàm từ 0 đến `MaxDpbSize` (như được chỉ rõ trong điểm A.4 của phiên bản dự thảo làm việc HEVC 8). Khi `i` lớn hơn 0, `sps_max_dec_pic_buffering[i]` sẽ bằng hoặc lớn hơn `sps_max_dec_pic_buffering[i - 1]`. Giá trị của `sps_max_dec_pic_buffering[i]` nhỏ hơn hoặc bằng `vps_max_dec_pic_buffering[i]` đối với mỗi giá trị `i`.

Tương tự, theo một hoặc nhiều kỹ thuật theo sáng chế, `sps_max_num_reorder_pics[i]` được xác định sao cho `sps_max_num_reorder_pics[i]` chỉ báo số hình ảnh tối đa được cho phép đứng trước hình ảnh bất kỳ theo thứ tự giải mã và đứng sau hình ảnh đó theo thứ tự xuất ra khi `TargetDecHighestTid` bằng `i`. `TargetDecHighestTid` được xác định theo cách được mô tả trên đây. Giá trị của `sps_max_num_reorder_pics[i]` nằm trong khoảng bao hàm từ 0 đến `sps_max_dec_pic_buffering[i]`. Khi `i` lớn hơn 0, `sps_max_num_reorder_pics[i]` bằng hoặc lớn hơn `sps_max_num_reorder_pics[i - 1]`. Giá trị của `sps_max_num_reorder_pics[i]` nhỏ hơn hoặc bằng `vps_max_num_reorder_pics[i]` đối với mỗi giá trị của `i`.

Hơn thế nữa, theo một hoặc nhiều kỹ thuật theo sáng chế, `cpb_cnt_minus1[i]` có thể chỉ rõ số đặc tả CPB khác trong dòng bit của chuỗi video mã hóa khi `TargetDecHighestTid` bằng `i`, trong đó `i` nằm trong khoảng từ 0 đến số lớp thời gian tối đa trong dòng bit. `TargetDecHighestTid` được xác định theo cách được mô tả trên đây. Giá trị của `cpb_cnt_minus1[i]` nằm trong khoảng bao hàm từ 0 đến 31. Khi `low_delay_hrd_flag[i]` bằng 1, thì `cpb_cnt_minus1[i]` bằng 0. Khi `cpb_cnt_minus1[i]` không có mặt, thì `cpb_cnt_minus1[i]` được suy ra là bằng 0.

Do đó, theo một hoặc nhiều kỹ thuật theo sáng chế, thiết bị có thể xác định, dựa vào ký hiệu nhận dạng theo thời gian cao nhất, phần tử cú pháp cụ thể từ trong số mảng phần tử cú pháp. Ký hiệu nhận dạng theo thời gian cao nhất được xác định sao cho ký hiệu nhận dạng theo thời gian cao nhất luôn nhận dạng được lớp thời gian cao nhất cần được giải mã. Do đó, `sps_max_num_reorder_pics[i]`, `sps_max_dec_pic_buffering[i]`, và `cpb_cnt_minus1[i]` trong các hoạt động HRD, các yêu cầu sự tương thích dòng bit và các giới hạn mức được chọn thích hợp với i bằng giá trị được xác định rõ ràng của `TargetDecHighestTid`.

Theo cách này, thiết bị (như bộ mã hóa dữ liệu video 20, bộ giải mã video 30, thiết bị khác 21, hoặc thiết bị khác) có thể tiến hành hoạt động HRD để xác định sự tương thích dòng bit với tiêu chuẩn mã hóa dữ liệu video hoặc xác định sự tương thích của bộ giải mã video với tiêu chuẩn mã hóa dữ liệu video. Như là một phần của việc tiến hành hoạt động HRD, thiết bị này có thể xác định được ký hiệu nhận dạng theo thời gian cao nhất của tập con dòng bit gắn với điểm hoạt động được chọn của dòng bit. Ngoài ra, thiết bị có thể xác định, dựa vào ký hiệu nhận dạng theo thời gian cao nhất, phần tử cú pháp cụ thể từ trong số mảng phần tử cú pháp (ví dụ, `sps_max_num_reorder_pics[i]`, `sps_max_dec_pic_buffering[i]`, hoặc `cpb_cnt_minus1[i]`). Hơn thế nữa, thiết bị có thể sử dụng phần tử cú pháp cụ thể trong hoạt động HRD.

Hơn thế nữa, trong phiên bản dự thảo làm việc HEVC 8, mỗi cấu trúc cú pháp `hrd_parameters()` trong VPS có thể gắn với cấu trúc cú pháp `operation_point_layer_ids()` dựa vào cấu trúc cú pháp `hrd_parameters()` nào được chọn để sử dụng trong hoạt động HRD. Tương ứng với mỗi cấu trúc cú pháp `hrd_parameters()` được chọn, tập hợp các thông báo SEI về chu kỳ nhớ đệm và các thông báo SEI định thời hình ảnh cũng có thể cần cho các hoạt động HRD. Tuy nhiên, không có cách nào để gắn thông báo SEI về chu kỳ nhớ đệm hoặc thông báo SEI định thời hình ảnh với cấu trúc cú pháp `hrd_parameters()` mà cấu trúc cú pháp `operation_point_layer_ids()` đi kèm bao gồm nhiều giá trị của `nuh_reserved_zero_6bits` (tức là, nhiều ID lớp trong phiên bản mở rộng mã hóa dữ liệu video đa cảnh nhìn, 3DV hoặc có khả năng chuyển đổi cấp độ của HEVC).

Giải pháp cho vấn đề này có thể là áp dụng thông báo SEI lồng nhau có khả năng chuyển đổi cấp độ mã hóa đa cảnh nhìn như được chỉ rõ trong phụ lục H của H.264/AVC hoặc thông báo SEI tương tự. Tuy nhiên, thông báo SEI lồng nhau có khả năng chuyển

đổi cấp độ mã hóa đa cảnh nhìn hoặc thông báo SEI tương tự có thể có các nhược điểm sau. Thứ nhất là, vì đơn vị NAL SEI trong H.264/AVC chỉ có phần đầu đơn vị NAL một byte, nên không có cách nào để sử dụng thông tin được tải trong `nuh_reserved_zero_6bits` và `temporal_id_plus1` trong phần đầu đơn vị NAL của đơn vị NAL SEI HEVC để gắn thông báo SEI định thời hình ảnh hoặc về chu kỳ nhớ đệm với điểm hoạt động. Thứ hai là, mỗi thông báo SEI được lồng chỉ có thể gắn với một điểm hoạt động.

Một hoặc nhiều kỹ thuật theo sáng chế có thể cung cấp cơ chế để xác định rõ các điểm hoạt động mà áp dụng thông báo SEI về chu kỳ nhớ đệm, thông báo SEI định thời hình ảnh hoặc thông báo SEI định thời hình ảnh con, thông qua cấu trúc cú pháp `applicable_operation_points()` có thể được tải trong thông báo SEI về chu kỳ nhớ đệm, thông báo SEI định thời hình ảnh, hoặc thông báo SEI định thời hình ảnh con. Cơ chế này có thể cho phép sử dụng thông tin được tải trong các phần tử cú pháp `nuh_reserved_zero_6bits` và `temporal_id_plus1` ở phần đầu đơn vị NAL của đơn vị NAL SEI, và có thể cho phép nhiều điểm hoạt động dùng chung thông tin được truyền tải ở cùng chu kỳ nhớ đệm, thông báo SEI định thời hình ảnh con hoặc định thời hình ảnh.

FIG.2 là sơ đồ khối minh họa bộ mã hóa dữ liệu video làm ví dụ 20 mà có thể tiến hành các kỹ thuật theo sáng chế. FIG.2 được đề xuất nhằm giải thích và không nên được coi là làm giới hạn các kỹ thuật như được minh họa ví dụ và được mô tả rộng rãi trong sáng chế. Để giải thích, sáng chế mô tả bộ mã hóa dữ liệu video 20 trong ngữ cảnh mã hóa HEVC. Tuy nhiên, các kỹ thuật theo sáng chế có thể áp dụng được cho các tiêu chuẩn hoặc phương pháp mã hóa khác.

Theo ví dụ nêu trong FIG.2, bộ mã hóa dữ liệu video 20 bao gồm bộ xử lý dự báo 100, bộ tạo dữ liệu dư 102, bộ xử lý biến đổi 104, bộ lượng tử hóa 106, bộ lượng tử hóa ngược 108, bộ xử lý biến đổi ngược 110, bộ phục hồi lại 112, bộ lọc 114, bộ nhớ đệm hình ảnh giải mã 116, và bộ mã hóa entropy 118. Bộ xử lý dự báo 100 bao gồm bộ xử lý dự báo liên cấu trúc 120 và bộ xử lý dự báo nội cấu trúc 126. Bộ xử lý dự báo liên cấu trúc 120 bao gồm bộ đánh giá chuyển động 122 và bộ bù chuyển động 124. Theo các ví dụ khác, bộ mã hóa dữ liệu video 20 có thể bao gồm các nhiều bộ phận chức năng hơn, ít bộ phận chức năng hơn hoặc các bộ phận chức năng khác nhau.

Bộ mã hóa dữ liệu video 20 có thể thu dữ liệu video. Bộ mã hóa dữ liệu video 20 có thể mã hóa mỗi CTU trong lát hình ảnh của dữ liệu video. Mỗi CTU có thể gắn với các

khối cây mã hóa độ chói có kích thước bằng nhau (coding tree block-CTB) và các CTB tương ứng của hình ảnh. Như là một phần của việc mã hóa CTU, bộ xử lý dự báo 100 có thể tiến hành phân tách cây tứ phân để phân chia các CTB của CTU thành các khối nhỏ dần. Khối nhỏ hơn có thể là khối mã hóa của CU. Ví dụ, bộ xử lý dự báo 100 có thể phân tách CTB gắn với CTU thành bốn khối con có kích thước bằng nhau, phân tách một hoặc nhiều trong số các khối con thành bốn khối con nữa có kích thước bằng nhau, và v.v.

Bộ mã hóa dữ liệu video 20 có thể mã hóa các CU của CTU để tạo ra các dạng biểu diễn mã hóa của CU (tức là, CU mã hóa). Như là một phần của quy trình mã hóa CU, bộ xử lý dự báo 100 có thể phân tách các khối mã hóa gắn với CU trong số một hoặc nhiều PU của CU. Do đó, mỗi PU có thể gắn với khối dự báo độ chói và khối dự báo màu tương ứng. Bộ mã hóa dữ liệu video 20 và bộ giải mã video 30 có thể hỗ trợ các PU có kích thước khác nhau. Kích thước của CU có thể được dùng để chỉ kích thước của khối mã hóa độ chói của CU và kích thước của PU có thể được dùng để chỉ là kích thước của khối dự báo độ chói của PU. Giả sử là kích thước của CU cụ thể là bằng $2N \times 2N$, bộ mã hóa dữ liệu video 20 và bộ giải mã video 30 có thể hỗ trợ các kích thước PU bằng $2N \times 2N$ hoặc $N \times N$ để dự báo nội cấu trúc, và các kích thước PU đối xứng bằng $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, hoặc tương tự để dự báo liên cấu trúc. Bộ mã hóa dữ liệu video 20 và bộ giải mã video 30 cũng có thể hỗ trợ phân tách không đối xứng đối với các kích thước PU bằng $2N \times nU$, $2N \times nD$, $nL \times 2N$, và $nR \times 2N$ để dự báo liên cấu trúc.

Bộ xử lý dự báo liên cấu trúc 120 có thể tạo lập dữ liệu dự báo cho PU bằng cách tiến hành thao tác dự báo liên cấu trúc trên mỗi PU của CU. Dữ liệu dự báo cho PU có thể bao gồm các khối dự báo của PU và thông tin chuyển động của PU. Bộ xử lý dự báo liên cấu trúc 120 có thể tiến hành các thao tác khác nhau đối với PU của CU tùy thuộc vào liệu PU có nằm trong lát I, lát P, hoặc lát B hay không. Trong lát I, tất cả các PU được dự báo nội lát. Do đó, nếu PU nằm trong lát I, thì bộ xử lý dự báo liên cấu trúc 120 không tiến hành dự báo liên cấu trúc trên PU. Do đó, đối với các khối được mã hóa trong lát I, thì khối dự báo được tạo thành bằng cách sử dụng thao tác dự báo không gian từ các khối lân cận được mã hóa trước đó trong cùng một khung.

Nếu PU nằm trong lát P, thì bộ đánh giá chuyển động 122 có thể tìm kiếm các hình ảnh chuẩn trong danh mục hình ảnh chuẩn (ví dụ, "RefPicList0") cho vùng chuẩn của PU. Vùng chuẩn của PU có thể là vùng, trong hình ảnh chuẩn, mà chứa các khối mẫu gần như tương ứng với khối dự báo của PU. Bộ đánh giá chuyển động 122 có thể tạo lập

chỉ số chuẩn chỉ báo vị trí của hình ảnh chuẩn chứa vùng chuẩn của PU trong RefPicList0. Ngoài ra, bộ đánh giá chuyển động 122 có thể tạo ra vector chuyển động chỉ báo sự dịch chuyển trong không gian giữa khối dự báo của PU và vị trí chuẩn gắn với vùng chuẩn. Ví dụ, vector chuyển động có thể là vector hai chiều cho biết độ lệch từ các tọa độ trong hình ảnh hiện thời đến các tọa độ trong hình ảnh chuẩn. Bộ đánh giá chuyển động 122 có thể xuất chỉ số chuẩn và vector chuyển động dưới dạng thông tin chuyển động của PU. Bộ bù chuyển động 124 có thể tạo ra các khối dự báo của PU dựa vào các mẫu thực hoặc nội suy ở vị trí chuẩn được chỉ báo bởi vector chuyển động của PU.

Nếu PU nằm trong lát B, bộ đánh giá chuyển động 122 có thể tiến hành thao tác dự báo một hướng hoặc thao tác dự báo hai hướng cho PU. Để tiến hành thao tác dự báo một hướng cho PU, bộ đánh giá chuyển động 122 có thể tìm kiếm các hình ảnh chuẩn của RefPicList0 hoặc danh mục hình ảnh chuẩn thứ hai (“RefPicList1”) cho vùng chuẩn của PU. Bộ đánh giá chuyển động 122 có thể xuất, làm thông tin chuyển động của PU, chỉ số chuẩn mà chỉ báo vị trí của hình ảnh chuẩn mà chứa vùng chuẩn trong RefPicList0 hoặc RefPicList1, vector chuyển động mà chỉ báo sự dịch chuyển trong không gian giữa khối mẫu của PU và vị trí chuẩn gắn với vùng chuẩn, và một hoặc nhiều bộ chỉ báo hướng dự báo để chỉ báo liệu hình ảnh chuẩn có nằm trong RefPicList0 hoặc RefPicList1 hay không. Bộ bù chuyển động 124 có thể tạo ra các khối dự báo của PU dựa ít nhất một phần vào các mẫu thực hoặc mẫu nội suy ở vùng chuẩn được chỉ báo bởi vector chuyển động của PU.

Để tiến hành thao tác dự báo liên cấu trúc hai chiều cho PU, bộ đánh giá chuyển động 122 có thể tìm kiếm các hình ảnh chuẩn trong RefPicList0 cho vùng chuẩn của PU và cũng có thể tìm kiếm các hình ảnh chuẩn trong RefPicList1 cho vùng chuẩn khác của PU. Bộ đánh giá chuyển động 122 có thể tạo ra các chỉ số chuẩn mà chỉ báo các vị trí của các hình ảnh chuẩn chứa các vùng chuẩn trong RefPicList0 và RefPicList1. Ngoài ra, bộ đánh giá chuyển động 122 có thể tạo ra vector chuyển động chỉ báo sự dịch chuyển trong không gian giữa vị trí chuẩn gắn với các vùng chuẩn và khối mẫu của PU. Thông tin chuyển động của PU có thể bao gồm các chỉ số chuẩn và vector chuyển động của PU. Bộ bù chuyển động 124 có thể tạo ra các khối dự báo của PU dựa ít nhất một phần vào các mẫu thực hoặc nội suy ở vùng chuẩn được chỉ báo bởi vector chuyển động của PU.

Bộ xử lý dự báo nội cấu trúc 126 có thể tạo ra dữ liệu dự báo cho PU bằng cách tiến hành thao tác dự báo nội cấu trúc trên PU. Dữ liệu dự báo cho PU có thể bao gồm

các khối dự báo cho PU và nhiều phần tử cú pháp khác. Bộ xử lý dự báo nội cấu trúc 126 có thể tiến hành thao tác dự báo nội cấu trúc trên PU trong lát I, lát P, và lát B.

Để tiến hành thao tác dự báo nội cấu trúc trên PU, bộ xử lý dự báo nội cấu trúc 126 có thể sử dụng nhiều kiểu dự báo nội cấu trúc để tạo ra nhiều bộ dữ liệu dự báo cho PU. Để sử dụng kiểu dự báo nội cấu trúc để tạo ra bộ dữ liệu dự báo cho PU, bộ xử lý dự báo nội cấu trúc 126 có thể mở rộng các mẫu từ các khối mẫu của các PU lân cận dọc các khối mẫu của PU theo hướng liên quan đến kiểu dự báo nội cấu trúc. Các PU lân cận có thể ở phía trên, phía trên bên phải, phía trên bên trái, hoặc bên trái của PU, giả sử thứ tự mã hóa là trái sang phải, trên xuống dưới đối với PU, CU, và CTU. Bộ xử lý dự báo nội cấu trúc 126 có thể sử dụng nhiều kiểu dự báo nội cấu trúc, ví dụ, 33 kiểu dự báo nội cấu trúc định hướng. Theo một vài ví dụ, số lượng kiểu dự báo nội cấu trúc có thể tùy thuộc vào kích thước vùng gắn với PU.

Bộ xử lý dự báo 100 có thể chọn dữ liệu dự báo cho PU của CU từ trong số dữ liệu dự báo được tạo ra bởi bộ xử lý dự báo liên cấu trúc 120 cho PU hoặc dữ liệu dự báo được tạo ra bởi bộ xử lý dự báo nội cấu trúc 126 cho PU. Theo một vài ví dụ, bộ xử lý dự báo 100 chọn dữ liệu dự báo cho PU của CU dựa vào số đo tỷ lệ/biến dạng của bộ dữ liệu dự báo. Các khối dự báo của dữ liệu dự báo được chọn có thể được gọi ở đây là các khối dự báo được chọn.

Bộ tạo dữ liệu dư 102 có thể tạo ra, dựa vào khối mã hóa độ chói, khối mã hóa Cb và Cr của CU và các khối độ chói, Cb và Cr dự báo được chọn của PU của CU, các khối độ chói, Cb và Cr dư của CU. Ví dụ, bộ tạo dữ liệu dư 102 có thể tạo ra các khối dư của CU sao cho mỗi mẫu trong các khối dư có giá trị bằng độ chênh lệch giữa mẫu trong khối mã hóa của CU và mẫu tương ứng trong khối dự báo được chọn tương ứng của PU của CU.

Bộ xử lý biến đổi 104 có thể tiến hành phân tách cây tứ phân để phân tách các khối dư gắn với CU thành các khối biến đổi gắn với TU của CU. Do đó, TU có thể gắn với khối biến đổi độ chói và hai khối biến đổi màu. Kích thước và vị trí của các khối biến đổi độ chói và màu của TU của CU có thể được hoặc có thể không được dựa trên kích thước và vị trí của các khối dự báo của PU của CU. Cấu trúc cây tứ phân được biết đến là “cây tứ phân dư” (residual quad-tree-RQT) có thể bao gồm các nút gắn với mỗi vùng. Các TU của CU có thể tương ứng với các nút lá của RQT.

Bộ xử lý biến đổi 104 có thể tạo các khối hệ số biến đổi cho mỗi TU của CU bằng cách áp dụng một hoặc nhiều kỹ thuật biến đổi cho các khối biến đổi của TU. Bộ xử lý biến đổi 104 có thể áp dụng nhiều kỹ thuật biến đổi khác nhau cho các khối biến đổi gắn với TU. Ví dụ, bộ xử lý biến đổi 104 có thể áp dụng kỹ thuật biến đổi cosin rời rạc (discrete cosine transform-DCT), kỹ thuật biến đổi định hướng, hoặc kỹ thuật biến đổi tương tự khái niệm cho khối biến đổi. Theo một vài ví dụ, bộ xử lý biến đổi 104 không áp dụng các kỹ thuật biến đổi cho khối biến đổi. Theo ví dụ này, khối biến đổi có thể được xử lý như là khối hệ số biến đổi.

Bộ lượng tử hóa 106 có thể lượng tử hóa các hệ số biến đổi trong khối hệ số. Quy trình lượng tử hóa có thể làm giảm độ sâu bit gắn với một vài hoặc toàn bộ các hệ số biến đổi. Ví dụ, hệ số biến đổi n -bit có thể được làm tròn xuống hệ số biến đổi m -bit khi lượng tử hóa, trong đó n lớn hơn m . Bộ lượng tử hóa 106 có thể lượng tử hóa khối hệ số gắn với TU của CU dựa vào giá trị thông số lượng tử hóa (quantization parameter-QP) gắn với CU. Bộ mã hóa dữ liệu video 20 có thể điều chỉnh mức lượng tử hóa áp dụng cho các khối hệ số gắn với CU bằng cách điều chỉnh giá trị QP gắn với CU. Lượng tử hóa có thể gây mất thông tin, do đó các hệ số biến đổi được lượng tử hóa có thể có độ chính xác thấp hơn các hệ số biến đổi ban đầu.

Bộ lượng tử hóa ngược 108 và bộ xử lý biến đổi ngược 110 có thể áp dụng kỹ thuật lượng tử hóa ngược và các kỹ thuật biến đổi ngược cho khối hệ số, tương ứng, để phục hồi lại khối dư từ khối hệ số. Bộ phục hồi lại 112 có thể bổ sung khối dư đã phục hồi lại cho các mẫu tương ứng từ một hoặc nhiều khối dự báo được tạo ra bởi bộ xử lý dự báo 100 để tạo ra khối biến đổi phục hồi lại gắn với TU. Bằng cách phục hồi lại các khối biến đổi cho mỗi TU của CU theo cách này, bộ mã hóa dữ liệu video 20 có thể phục hồi lại các khối mã hóa của CU.

Bộ lọc 114 có thể tiến hành một hoặc nhiều thao tác tách khối để làm giảm các thành phần lạ tạo khối trong các khối mã hóa gắn với CU. Bộ nhớ đệm hình ảnh giải mã 116 có thể lưu trữ các khối mã hóa được phục hồi lại sau khi bộ lọc 114 tiến hành một hoặc nhiều thao tác tách khối trên các khối mã hóa được phục hồi lại. Bộ dự báo liên cấu trúc 120 có thể sử dụng hình ảnh chuẩn chứa các khối mã hóa phục hồi lại để tiến hành dự báo liên cấu trúc trên các PU của các hình ảnh khác. Ngoài ra, bộ xử lý dự báo nội cấu trúc 126 có thể sử dụng các khối mã hóa phục hồi lại trong bộ nhớ đệm hình ảnh giải mã

116 để tiến hành dự báo nội cấu trúc trên các PU khác trong cùng hình ảnh giống như CU.

Bộ mã hóa entropy 118 có thể thu dữ liệu từ các thành phần chức năng khác của bộ mã hóa dữ liệu video 20. Ví dụ, bộ mã hóa entropy 118 có thể thu các khối hệ số từ bộ lượng tử hóa 106 và có thể thu phần tử cú pháp từ bộ xử lý dự báo 100. Bộ mã hóa entropy 118 có thể tiến hành một hoặc nhiều thao tác mã hóa entropy trên dữ liệu để tạo ra dữ liệu mã hóa entropy. Ví dụ, bộ mã hóa entropy 118 có thể tiến hành thao tác mã hóa độ dài thay đổi thích ứng ngữ cảnh (context-adaptive variable length coding-CAVLC), thao tác mã hóa số học nhị phân thích ứng ngữ cảnh (context adaptive binary arithmetic coding-CABAC), thao tác mã hóa độ dài theo từng biến (variable-to-variable-V2V), thao tác mã hóa số học nhị phân thích ứng ngữ cảnh dựa trên cú pháp (syntax-based context-adaptive binary arithmetic coding-SBAC), thao tác mã hóa entropy phân chia khoảng xác suất (Probability Interval Partitioning Entropy-PIPE), thao tác mã hóa Golomb theo số mũ, hoặc loại thao tác mã hóa entropy khác trên dữ liệu. Bộ mã hóa dữ liệu video 20 có thể xuất dòng bit chứa dữ liệu mã hóa entropy được tạo ra bởi bộ mã hóa entropy 118. Ví dụ, dòng bit có thể chứa dữ liệu biểu diễn RQT cho CU.

Như được chỉ ra trong sáng chế, bộ mã hóa dữ liệu video 20 có thể báo hiệu VPS trong dòng bit. Theo phiên bản dự thảo làm việc HEVC 8, các phần tử cú pháp cụ thể của VPS (tức là, `vps_max_dec_pic_buffering[i]`, `vps_max_num_reorder_pics[i]`, và `vps_max_latency_increase[i]`) được xác định dựa vào giá trị `HighestTid`, mà không được xác định. Theo một hoặc nhiều kỹ thuật theo sáng chế, các phần tử cú pháp của VPS này có thể được xác định dựa vào giá trị `TargetDecHighestTid`, mà được xác định sao cho `TargetDecHighestTid` như được mô tả trong sáng chế. Bảng 3, dưới đây, minh họa cú pháp của VPS theo một hoặc nhiều kỹ thuật theo sáng chế.

Bảng 3 - VPS

video_parameter_set_rbsp() {	Bộ mô tả
video_parameter_set_id	u(4)
vps_temporal_id_nesting_flag	u(1)
vps_reserved_zero_2bits	u(2)
vps_reserved_zero_6bits	u(6)
vps_max_sub_layers_minus1	u(3)
profile_bậc_level(1, vps_max_sub_layers_minus1)	
vps_reserved_zero_12bits	u(12)
for(i = 0; i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering[i]	ue(v)
vps_max_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
vps_num_hrd_parameters	ue(v)
for(i = 0; i < vps_num_hrd_parameters; i++) {	
operation_point_layer_ids(i)	
hrd_parameters(i == 0, vps_max_sub_layers_minus1)	
}	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
}	
rbp_trailing_bits()	
}	

Phần in nghiêng trong bảng 3 và các bảng cú pháp khác hoặc các phần mô tả ngữ nghĩa trong bản mô tả này có thể thể hiện các khác biệt so với phiên bản dự thảo làm việc HEVC 8. Theo một hoặc nhiều kỹ thuật theo sáng chế, các ngữ nghĩa của các phần tử cú pháp dưới đây của VPS có thể được thay đổi như sau. Các ngữ nghĩa của các phần tử cú pháp khác của VPS có thể vẫn giống như trong phiên bản dự thảo làm việc HEVC 8.

vps_max_dec_pic_buffering[i] chỉ rõ dung lượng cần thiết của bộ nhớ đệm hình ảnh giải mã trong các đơn vị của bộ nhớ đệm lưu trữ hình ảnh khi *TargetDecHighestTid* là bằng i. Giá trị của **vps_max_dec_pic_buffering[i]** nằm trong khoảng bao hàm từ 0 đến *MaxDpbSize* (như được chỉ rõ trong điểm A.4).

Khi i lớn hơn 0, $vps_max_dec_pic_buffering[i]$ bằng hoặc lớn hơn $vps_max_dec_pic_buffering[i - 1]$.

$vps_max_num_reorder_pics[i]$ cho biết số hình ảnh tối đa được cho phép đứng trước hình ảnh bất kỳ theo thứ tự giải mã và đứng sau hình ảnh đó theo thứ tự xuất khi $TargetDecHighestTid$ bằng i . Giá trị của $vps_max_num_reorder_pics[i]$ nằm trong khoảng bao hàm từ 0 đến $vps_max_dec_pic_buffering[i]$. Khi i lớn hơn 0, $vps_max_num_reorder_pics[i]$ bằng hoặc lớn hơn $vps_max_num_reorder_pics[i - 1]$.

$vps_max_latency_increase[i]$ không bằng 0 được dùng để tính toán giá trị của $MaxLatencyPictures[i]$ như được chỉ rõ bằng cách đặt $MaxLatencyPictures[i]$ bằng $vps_max_num_reorder_pics[i] + vps_max_latency_increase[i]$. Khi $vps_max_latency_increase[i]$ không bằng 0, giá trị của $MaxLatencyPictures[i]$ chỉ rõ số hình ảnh tối đa có thể đứng trước hình ảnh bất kỳ trong chuỗi video mã hóa theo thứ tự xuất ra và đứng sau hình ảnh đó theo thứ tự giải mã khi $TargetDecHighestTid$ bằng i . Khi $vps_max_latency_increase[i]$ bằng 0, không có giới hạn tương ứng nào được thể hiện. Giá trị của $vps_max_latency_increase[i]$ nằm trong khoảng bao hàm từ 0 đến $2^{32} - 2$, bao gồm cả giá trị đầu mút.

Như được mô tả trên đây, các ngữ nghĩa của $vps_max_dec_buffering[i]$, $vps_max_num_reorder_pics[i]$, và $vps_max_latency_increase[i]$ có thể được xác định dựa vào $TargetDecHighestTid$. Ngược lại, phiên bản dự thảo làm việc HEVC 8 xác định $vps_max_dec_pic_buffering[i]$, $vps_max_num_reorder_pics[i]$, và $vps_max_latency_increase[i]$ dựa vào $HighestTid$, trong đó $HighestTid$ không được xác định.

Như được thể hiện trong cú pháp làm ví dụ trong bảng 3, VPS bao gồm các cặp cấu trúc cú pháp `operation_point_layer_ids()` và cấu trúc cú pháp `hrd_parameters()`. Các cấu trúc cú pháp `hrd_parameters()` bao gồm các phần tử cú pháp mà chỉ rõ bộ thông số HRD. Cấu trúc cú pháp `operation_point_layer_ids()` bao gồm các phần tử cú pháp mà xác định tập hợp các điểm hoạt động. Bộ thông số HRD được chỉ rõ trong cấu trúc cú pháp `hrd_parameters()` có thể áp dụng được cho các điểm hoạt động được xác định bởi

các phần tử cú pháp trong cấu trúc cú pháp `operation_point_layer_ids( )` tương ứng. Bảng 4, dưới đây, cung cấp cú pháp làm ví dụ cho cấu trúc cú pháp `operation_point_layer_ids( )`.

Bảng 4 – Ký hiệu nhận dạng lớp của điểm hoạt động

<code>operation_point_layer_ids(opIdx) {</code>	Bộ mô tả
<code> op_num_layer_id_values_minus1[opIdx]</code>	ue(v)
<code> for(i = 0; i <= op_num_layer_id_values_minus1[opIdx];</code> <code> i++)</code>	
<code> op_layer_id[opIdx][i]</code>	u(6)
<code>}</code>	

Mục 7.4.4 của phiên bản dự thảo làm việc HEVC 8 mô tả các ngữ nghĩa của cấu trúc cú pháp `op_point`. Theo một hoặc nhiều kỹ thuật theo sáng chế, mục 7.4.4 của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi như sau để cung cấp các ngữ nghĩa của cấu trúc cú pháp `operation_point_layer_ids( )` nêu trong bảng 4.

Cấu trúc cú pháp `operation_point_layer_ids( opIdx )` chỉ rõ tập hợp các giá trị `nuh_reserved_zero_6bits` được bao gồm trong `OpLayerIdSet` của các điểm hoạt động mà áp dụng cấu trúc cú pháp `opIdx-th hrd_parameters( )` trong bộ thông số video.

`op_num_layer_id_values_minus1[ opIdx ]` cộng 1 chỉ rõ số giá trị `nuh_reserved_zero_6bits` được bao gồm trong `OpLayerIdSet` của điểm hoạt động mà áp dụng cấu trúc cú pháp `opIdx-th hrd_parameters( )` trong bộ thông số video. `op_num_layer_id_values_minus1[ opIdx ]` nhỏ hơn hoặc bằng 63. Đối với các dòng bit tuân theo đặc tả này, thì `op_num_layer_id_values_minus1[ opIdx ]` bằng 0. Mặc dù giá trị của `op_num_layer_id_values_minus1[ opIdx ]` được yêu cầu bằng 0 trong phiên bản này của đặc tả này, nhưng các bộ giải mã sẽ cho phép các giá trị khác xuất hiện trong cú pháp `op_num_layer_id_values_minus1[ opIdx ]`.

`op_layer_id[ opIdx ][ i ]` chỉ rõ giá trị thứ `i` của `nuh_reserved_zero_6bits` được bao gồm trong `OpLayerIdSet` của điểm hoạt động mà áp dụng cấu trúc cú pháp `opIdx-th hrd_parameters( )` trong bộ thông số video. Không giá trị nào của `op_layer_id[ opIdx ][ i ]` bằng với `op_layer_id[ opIdx ][ j ]` khi `i` không bằng `j` và

cả i và j đều nằm trong khoảng bao hàm từ 0 đến $op_num_layer_id_values_minus1$, bao gồm cả giá trị đầu mút. $op_layer_id[0][0]$ được suy ra là bằng 0.

Như được chỉ ra trên đây, phần tử cú pháp $op_num_layer_id_values_minus1[opIdx]$ cộng 1, chỉ rõ số giá trị $nuh_reserved_zero_6bits$ được bao gồm trong $OpLayerIdSet$ của các điểm hoạt động mà áp dụng cấu trúc cú pháp $opIdx$ -th $hrd_parameters()$ trong bộ thông số video. Ngược lại, phiên bản dự thảo làm việc HEVC 8 mô tả rằng phần tử cú pháp $op_num_layer_id_values_minus1[opIdx]$ cộng 1, chỉ rõ số giá trị $nuh_reserved_zero_6bits$ được bao gồm trong điểm hoạt động được xác định bởi $opIdx$. Tương tự, trong ví dụ nêu trong bảng 4, phần tử cú pháp $op_layer_id[opIdx][i]$ chỉ rõ giá trị thứ i của $nuh_reserved_zero_6bits$ được bao gồm trong $OpLayerIdSet$ của điểm hoạt động mà áp dụng cấu trúc cú pháp $opIdx$ -th $hrd_parameters()$ trong bộ thông số video. Ngược lại, phiên bản dự thảo làm việc HEVC 8 mô tả rằng phần tử cú pháp $op_layer_id[opIdx][i]$ chỉ rõ giá trị thứ i của $nuh_reserved_zero_6bits$ được bao gồm trong điểm hoạt động được xác định bởi $opIdx$.

Mục 7.4.2.2 của phiên bản dự thảo làm việc HEVC 8 mô tả các ngữ nghĩa của SPS. Theo một hoặc nhiều kỹ thuật theo sáng chế, các thay đổi dưới đây có thể được tiến hành với mục 7.4.2.2 của phiên bản dự thảo làm việc HEVC 8. Các ngữ nghĩa của các phần tử cú pháp khác của SPS có thể giống như trong phiên bản dự thảo làm việc HEVC 8:

$sps_max_dec_pic_buffering[i]$ chỉ rõ dung lượng tối đa cần thiết của bộ nhớ đệm hình ảnh giải mã trong các đơn vị của bộ nhớ đệm lưu trữ hình ảnh khi $TargetDecHighestTid$ bằng i . Giá trị của $sps_max_dec_pic_buffering[i]$ sẽ nằm trong khoảng từ 0 đến $MaxDpbSize$ (như được chỉ rõ trong điểm A.4), bao gồm cả giá trị đầu mút. Khi i lớn hơn 0, $sps_max_dec_pic_buffering[i]$ sẽ bằng hoặc lớn hơn $sps_max_dec_pic_buffering[i - 1]$. Giá trị của $sps_max_dec_pic_buffering[i]$ nhỏ hơn hoặc bằng $vps_max_dec_pic_buffering[i]$ đối với mỗi giá trị của i .

$sps_max_num_reorder_pics[i]$ chỉ báo số hình ảnh tối đa được cho phép đứng trước hình ảnh bất kỳ theo thứ tự giải mã và đứng sau hình ảnh đó theo thứ tự xuất ra khi $TargetDecHighestTid$ bằng i . Giá trị của $sps_max_num_reorder_pics[i]$

nằm trong khoảng từ 0 đến `sps_max_dec_pic_buffering[i]`, bao gồm cả giá trị đầu mút. Khi i lớn hơn 0, `sps_max_num_reorder_pics[i]` bằng hoặc lớn hơn `sps_max_num_reorder_pics[i - 1]`. Giá trị của `sps_max_num_reorder_pics[i]` sẽ nhỏ hơn hoặc bằng `vps_max_num_reorder_pics[i]` đối với mỗi giá trị của i .

`sps_max_latency_increase[i]` không bằng 0 được sử dụng để tính toán giá trị của `MaxLatencyPictures[i]` như được chỉ rõ bằng cách đặt `MaxLatencyPictures[i]` bằng `sps_max_num_reorder_pics[i] + sps_max_latency_increase[i]`. Khi `sps_max_latency_increase[i]` không bằng 0, giá trị của `MaxLatencyPictures[i]` chỉ rõ số hình ảnh tối đa có thể đứng trước hình ảnh bất kỳ trong chuỗi video mã hóa theo thứ tự xuất ra và đứng sau hình ảnh đó theo thứ tự giải mã khi *TargetDecHighestTid* bằng i . Khi `sps_max_latency_increase[i]` bằng 0, không có giới hạn tương ứng nào được thể hiện. Giá trị của `sps_max_latency_increase[i]` sẽ nằm trong khoảng từ 0 đến $2^{32} - 2$, bao gồm cả giá trị đầu mút. Giá trị của `sps_max_latency_increase[i]` sẽ nhỏ hơn hoặc bằng `vps_max_latency_increase[i]` đối với mỗi giá trị của i .

Như được thể hiện trên đây, các ngữ nghĩa của `sps_max_dec_pic_buffering[i]`, `sps_max_num_reorder_pics[i]`, và `sps_max_latency_increase[i]` được xác định dựa vào *TargetDecHighestTid*. *TargetDecHighestTid* được xác định như được mô tả trong bản mô tả. Ngược lại, phiên bản dự thảo làm việc HEVC 8 xác định các ngữ nghĩa của `sps_max_dec_pic_buffering[i]`, `sps_max_num_reorder_pics[i]`, và `sps_max_latency_increase[i]` dựa vào *HighestTid*, mà không được xác định.

Mục 7.4.5.1 của phiên bản dự thảo làm việc HEVC 8 mô tả các ngữ nghĩa của phần đầu lát chung. Theo một hoặc nhiều kỹ thuật theo sáng chế, các thay đổi sau có thể được tiến hành đối với phần 7.4.5.1 của phiên bản dự thảo làm việc HEVC 8. Các phần khác của mục 7.4.5.1 của phiên bản dự thảo làm việc HEVC 8 có thể vẫn giữ nguyên.

`no_output_of_prior_pics_flag` chỉ rõ cách xử lý các hình ảnh giải mã trước trong bộ nhớ đệm hình ảnh giải mã sau khi giải mã hình ảnh làm mới quy trình giải mã tức thời (Instantaneous Decoding Refresh-IDR) hoặc hình ảnh truy nhập liên kết hỏng (Broken Link Access-BLA). Xem phụ lục C. Khi hình ảnh hiện thời là hình ảnh truy nhập ngẫu nhiên sạch (Clean Random Access-CRA), hoặc hình ảnh hiện

thời là hình ảnh IDR hoặc BLA là hình ảnh đầu tiên trong dòng bit, thì giá trị của `no_output_of_prior_pics_flag` không ảnh hưởng đến quy trình giải mã. Khi hình ảnh hiện thời là hình ảnh IDR hoặc BLA không phải là hình ảnh trong dòng bit, và giá trị của `pic_width_in_luma_samples` hoặc `pic_height_in_luma_samples` hoặc `sps_max_dec_pic_buffering[TargetDecHighestTid]` được tìm ra từ bộ thông số chuỗi hoạt động là khác so với giá trị của `pic_width_in_luma_samples` hoặc `pic_height_in_luma_samples` hoặc `sps_max_dec_pic_buffering[TargetDecHighestTid]` được tìm ra từ bộ thông số chuỗi hoạt động của hình ảnh trước, `no_output_of_prior_pics_flag` bằng 1 có thể (nhưng tốt hơn là không) được suy ra bằng bộ giải mã, bất kể giá trị thực của `no_output_of_prior_pics_flag`.

Như được thể hiện trên đây, các ngữ nghĩa của `no_output_of_prior_pics_flag` được xác định dựa vào `sps_max_dec_pic_buffering[TargetDecHighestTid]`. `TargetDecHighestTid` được xác định như được mô tả trong bản mô tả. Ngược lại, phiên bản dự thảo làm việc HEVC 8 xác định các ngữ nghĩa của `no_output_of_prior_pics_flags` dựa vào `sps_max_dec_pic_buffering[HighestTid]`, trong đó `HighestTid` không được xác định.

Mục 8.1 của phiên bản dự thảo làm việc HEVC 8 mô tả quy trình giải mã chung. Theo một hoặc nhiều kỹ thuật theo sáng chế, quy trình giải mã chung của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi như sau.

Đầu vào của quy trình này là dòng bit và đầu ra là danh mục hình ảnh giải mã.

Tập hợp `TargetDecLayerIdSet`, chỉ rõ tập hợp các giá trị của `nuh_reserved_zero_6bits` của các đơn vị NAL có VCL cần được giải mã, được chỉ rõ như sau:

- Nếu một vài phương tiện ngoài không được chỉ rõ trong đặc tả này có mặt để thiết lập `TargetDecLayerIdSet`, thì `TargetDecLayerIdSet` được thiết lập bởi phương tiện ngoài.*
- Mặt khác nếu quy trình giải mã được gọi ra để kiểm tra tương thích dòng bit như được chỉ rõ trong điểm C.1, `TargetDecLayerIdSet` được thiết lập như được chỉ rõ trong điểm C.1.*
- Mặt khác, `TargetDecLayerIdSet` chỉ chứa một giá trị của `nuh_reserved_zero_6bits`, mà bằng 0.*

Biến *TargetDecHighestTid*, xác định lớp con thời gian cao nhất cần được giải mã, được chỉ rõ như sau:

- Nếu một vài phương tiện ngoài không được chỉ rõ trong đặc tả này có mặt để thiết lập *TargetDecHighestTid*, *TargetDecHighestTid* được thiết lập bằng phương tiện ngoài.
- Mặt khác nếu quy trình giải mã được gọi ra để kiểm tra tương thích dòng bit như được chỉ rõ trong điểm C.1, *TargetDecHighestTid* được thiết lập như được chỉ rõ trong điểm C.1.
- Mặt khác, *TargetDecHighestTid* được đặt bằng *sps_max_sub_layers_minus1*.

Quy trình tách dòng bit con như được chỉ rõ trong điểm 10.1 được áp dụng với *TargetDecHighestTid* và *TargetDecLayerIdSet* là đầu vào và đầu ra được phân định cho dòng bit được gọi là *BitstreamToDecode*.

Áp dụng như sau cho mỗi hình ảnh mã hóa (được gọi là hình ảnh hiện thời, mà được ký hiệu là biến *CurrPic*) trong *BitstreamToDecode*.

Tùy thuộc vào giá trị của *chroma_format_idc*, số mảng mẫu của hình ảnh hiện thời là như sau.

- Nếu *chroma_format_idc* bằng 0, thì hình ảnh hiện thời bao gồm 1 mảng mẫu S_L .
- Mặt khác (*chroma_format_idc* bằng 0), thì hình ảnh hiện thời bao gồm 3 mảng mẫu S_L , S_{Cb} , S_{Cr} .

Quy trình giải mã hình ảnh hiện thời lấy các phần tử cú pháp và các biến in hoa từ mệnh đề 7 làm đầu vào. Khi dịch các ngữ nghĩa của mỗi phần tử cú pháp trong mỗi đơn vị NAL và "dòng bit" hoặc một phần của chúng (ví dụ, chuỗi video mã hóa) được gọi ra, thì dòng bit hoặc một phần của nó có nghĩa là *BitstreamToDecode* hoặc một phần của nó.

Quy trình giải mã được chỉ rõ là tất cả các bộ giải mã sẽ thu được các kết quả giống nhau về số lượng. Quy trình giải mã bất kỳ mà thu được các kết quả giống với quy trình được mô tả ở đây tuân theo các yêu cầu về quy trình giải mã của đặc tả này.

Khi hình ảnh hiện thời là hình ảnh CRA, áp dụng như sau:

- Nếu một vài phương tiện ngoài không được chỉ rõ trong đặc tả này là có mặt để đặt biến `HandleCraAsBlaFlag` bằng một giá trị, `HandleCraAsBlaFlag` được đặt bằng giá trị được cung cấp bởi phương tiện ngoài.
- Mặt khác, giá trị của `HandleCraAsBlaFlag` được đặt bằng 0.

Khi hình ảnh hiện thời là hình ảnh CRA và `HandleCraAsBlaFlag` bằng 1, áp dụng như sau khi quy trình giải mã và phân tích cú pháp cho mỗi đơn vị NAL của lát mã hóa:

- Giá trị của `nal_unit_type` được đặt bằng `BLA_W_LP`.
- Giá trị của `no_output_of_prior_pics_flag` được đặt bằng 1.

Lưu ý 1 – bộ giải mã theo các phương án thực hiện này có thể chọn để đặt giá trị của `no_output_of_prior_pics_flag` bằng 0 khi việc đặt này không làm ảnh hưởng đến quy trình giải mã hình ảnh hiện thời và các hình ảnh đứng sau theo thứ tự giải mã, ví dụ khi luôn có bộ nhớ đệm lưu trữ hình ảnh khi cần.

Mỗi hình ảnh được đề cập trong mệnh đề này là hình ảnh được mã hóa hoàn toàn. Tùy thuộc vào giá trị của `separate_colour_plane_flag`, quy trình giải mã được thiết lập như sau.

- Nếu `separate_colour_plane_flag` là bằng 0, thì quy trình giải mã được gọi ra một lần với hình ảnh hiện thời được xuất ra.
- Mặt khác (`separate_colour_plane_flag` là bằng 1), quy trình giải mã được gọi ra ba lần. Đầu vào của quy trình giải mã là tất cả các đơn vị NAL chứa hình ảnh mã hóa với giá trị `colour_plane_id` giống nhau. Quy trình giải mã các đơn vị NAL với giá trị cụ thể của `colour_plane_id` được chỉ rõ cứ như thể chỉ chuỗi video mã hóa với định dạng màu đơn màu với giá trị cụ thể đó của `colour_plane_id` sẽ có mặt trong dòng bit. Đầu ra của mỗi trong ba quy trình giải mã được phân định là 3 mảng mẫu của hình ảnh hiện thời với đơn vị NAL có `colour_plane_id` bằng 0 được phân định cho S_L , các đơn vị NAL với `colour_plane_id` bằng 1 được phân định cho S_{Cb} , và các đơn vị NAL với `colour_plane_id` bằng 2 được phân định cho S_{Cr} .

Lưu ý 1 – Biến `ChromaArrayType` được coi là 0 khi `separate_colour_plane_flag` bằng 1 và `chroma_format_idc` bằng 3. Trong

quy trình giải mã, giá trị của biến này được đánh giá dẫn đến các thao tác giống với thao tác của hình ảnh đơn màu với `chroma_format_idc` bằng 0.

Quy trình giải mã hoạt động như sau đối với *hình ảnh hiện thời CurrPic*:

1. Quy trình giải mã các đơn vị NAL được chỉ rõ trong điểm 8.2.
2. Quy trình theo điểm 8.3 chỉ rõ các quy trình giải mã sử dụng các phần tử cú pháp trong lớp lát và trên đây:
 - Các biến và chức năng có liên quan đến số đếm thứ tự hình ảnh được nêu trong điểm 8.3.1 (mà chỉ các yêu cầu đối với lát đầu tiên của hình ảnh được gọi ra).
 - Quy trình giải mã bộ hình ảnh chuẩn trong điểm 8.3.2 được gọi ra, trong đó các hình ảnh chuẩn có thể được đánh dấu là "không dùng làm chuẩn" hoặc "dùng làm chuẩn dài hạn" (mà chỉ các yêu cầu đối với lát đầu của hình ảnh được gọi ra).
 - Khi hình ảnh hiện thời là hình ảnh BLA hoặc là hình ảnh CRA là hình ảnh đầu tiên trong dòng bit, thì quy trình giải mã tạo ra các hình ảnh chuẩn không có mặt được chỉ rõ trong điểm 8.3.3 được gọi ra (mà chỉ các yêu cầu đối với lát đầu của hình ảnh được gọi ra).
 - `PicOutputFlag` được thiết lập như sau:
 - Nếu hình ảnh hiện thời là hình ảnh TFD và hình ảnh RAP trước theo thứ tự giải mã là hình ảnh BLA hoặc là hình ảnh CRA là hình ảnh giải mã đầu tiên trong dòng bit, `PicOutputFlag` được đặt bằng 0.
 - Mặt khác, `PicOutputFlag` được đặt bằng `pic_output_flag`.
 - Khi bắt đầu quy trình giải mã cho mỗi lát P hoặc B, quy trình giải mã cho danh mục hình ảnh chuẩn được chỉ rõ trong điểm 8.3.4 được gọi ra để tìm ra danh mục hình ảnh chuẩn 0 (`RefPicList0`), và khi giải mã lát B được gọi ra để tìm ra danh mục hình ảnh chuẩn 1 (`RefPicList1`).
 - Sau khi tất cả các lát của hình ảnh hiện thời được giải mã, hình ảnh giải mã được đánh dấu là "dùng làm chuẩn ngắn hạn".

3. Quy trình theo các điểm 8.4, 8.5, 8.6, và 8.7 chỉ rõ quy trình giải mã bằng cách sử dụng các phần tử cú pháp trong lớp đơn vị của cây mã hóa và trên đây.

Như được chỉ rõ trong bản mô tả, trong phiên bản dự thảo làm việc HEVC 8, khi quy trình giải mã được gọi ra để kiểm tra tương thích dòng bit, các ngữ nghĩa của các phần tử cú pháp không được xác định rõ ràng khi các giá trị của `TargetDecLayerIdSet` và `TargetDecHighestTid` không được thiết lập thích hợp. Các thay đổi nêu trên đối với quy trình giải mã chung có thể khắc phục được vấn đề này. Như được thể hiện trên đây, khi quy trình giải mã chung được gọi ra để kiểm tra tương thích dòng bit, các giá trị của `TargetDecLayerIdSet` và `TargetDecHighestTid` được đặt như được chỉ rõ trong điểm C.1. Như được mô tả dưới đây, phiên bản cải biến của điểm C.1 có thể đặt `TargetDecLayerIdSet` bằng tập hợp các giá trị `nuh_reserved_zero_6bits` có mặt trong dòng bit con tương ứng với điểm hoạt động kiểm tra. Phiên bản cải biến của điểm C.1 có thể đặt `TargetDecHighestTid` bằng giá trị `TemporalId` lớn nhất có mặt trong dòng bit con tương ứng với điểm hoạt động kiểm tra.

Theo cách này, thiết bị, như bộ giải mã video 30, có thể tiến hành quy trình giải mã như là một phần của việc tiến hành sự kiểm tra tương thích dòng bit. Tiến hành quy trình giải mã có thể bao gồm tiến hành quy trình tách dòng bit để tách, từ dòng bit, dạng biểu diễn điểm hoạt động của điểm hoạt động được xác định bằng tập hợp các ký hiệu nhận dạng lớp đích và ký hiệu nhận dạng theo thời gian cao nhất của đích. Tập hợp các ký hiệu nhận dạng lớp đích có thể chứa các giá trị của các phần tử cú pháp nhận dạng lớp có mặt trong dạng biểu diễn điểm hoạt động, tập hợp các ký hiệu nhận dạng lớp đích là tập con các giá trị của các phần tử cú pháp nhận dạng lớp của dòng bit. Ký hiệu nhận dạng theo thời gian cao nhất của đích có thể bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dạng biểu diễn điểm hoạt động, ký hiệu nhận dạng theo thời gian cao nhất của đích là nhỏ hơn hoặc bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dòng bit. Hơn thế nữa, thiết bị có thể giải mã các đơn vị NAL của dạng biểu diễn điểm hoạt động.

Như được chỉ rõ trong các phần thay đổi trong mục 8.1 trên đây, quy trình giải mã không cần phải được tiến hành như là một phần của sự kiểm tra tương thích dòng bit. Trong một vài trường hợp mà quy trình giải mã không được tiến hành như là một phần của sự kiểm tra tương thích dòng bit, thiết bị có thể tiến hành quy trình tách dòng bit để

tách, từ dòng bit, dạng biểu diễn điểm hoạt động của điểm hoạt động. Trong trường hợp này, 0 có thể là giá trị duy nhất của các phần tử cú pháp nhận dạng lớp (ví dụ, `nuh_reserved_zero_6bits`) có mặt trong dạng biểu diễn điểm hoạt động, và ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dòng bit là bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dạng biểu diễn điểm hoạt động của điểm hoạt động. Thiết bị này có thể giải mã các đơn vị NAL của dạng biểu diễn điểm hoạt động của điểm hoạt động thứ hai.

Theo cách khác, thiết bị có thể thu, từ nguồn ngoài, tập hợp các ký hiệu nhận dạng lớp đích và ký hiệu nhận dạng theo thời gian cao nhất của đích. Tập hợp các ký hiệu nhận dạng lớp đích có thể chứa các giá trị của các phần tử cú pháp nhận dạng lớp có mặt trong dạng biểu diễn điểm hoạt động của điểm hoạt động mà được xác định bởi tập hợp các ký hiệu nhận dạng lớp đích và ký hiệu nhận dạng theo thời gian cao nhất của đích. Ký hiệu nhận dạng theo thời gian cao nhất của đích có thể bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dạng biểu diễn điểm hoạt động của điểm hoạt động. Hơn thế nữa, thiết bị có thể tiến hành quy trình tách dòng bit để tách, từ dòng bit, dạng biểu diễn điểm hoạt động của điểm hoạt động. Ngoài ra, thiết bị có thể giải mã các đơn vị NAL của dạng biểu diễn điểm hoạt động của điểm hoạt động.

Hơn thế nữa, theo một hoặc nhiều kỹ thuật theo sáng chế, quy trình tách dòng bit con được mô tả trong điểm 10.1 của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi như sau.

Yêu cầu của sự tương thích dòng bit là dòng bit con bất kỳ mà được bao gồm trong đầu ra của quy trình được chỉ rõ trong điểm này với `tIdTarget` bằng giá trị bất kỳ nằm trong khoảng từ 0 đến 6, bao gồm cả giá trị đầu mút, và với `layerIdSetTarget` chỉ chứa duy nhất giá trị 0 tuân theo đặc tả này.

Lưu ý – Dòng bit tương thích chứa một hoặc nhiều đơn vị NAL của lát mã hóa với `nuh_reserved_zero_6bits` bằng 0 và `TemporalId` bằng 0.

Đầu vào của quy trình này là biến `tIdTarget` và tập hợp `layerIdSetTarget`.

Đầu ra của quy trình này là dòng bit con.

Dòng bit con thu được bằng cách lấy tất cả các đơn vị NAL có `TemporalId` lớn hơn `tIdTarget` hoặc `nuh_reserved_zero_6bits` không phải trong số các giá trị trong `layerIdSetTarget` từ dòng bit.

Theo điểm 10.1 của phiên bản dự thảo làm việc HEVC 8, tên biến `targetDecLayerIdSet` được sử dụng ở vị trí mà `layerIdSetTarget` được sử dụng trên đây. Các thay đổi được nêu trên đây đối với điểm 10.1 của phiên bản dự thảo làm việc HEVC để sử dụng `layerIdSetTarget` có thể nhằm chỉ rõ rằng có thể có sự khác biệt giữa tập hợp ký hiệu nhận dạng lớp được sử dụng trong quy trình tách dòng bit con và `targetDecLayerIdSet`, như được mô tả trong bản mô tả, có định nghĩa cụ thể.

Hơn thế nữa, theo một hoặc nhiều kỹ thuật theo sáng chế, các đặc tả của mức và bậc chung của mục A.4.1 của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi như sau. Trong bản mô tả này, “*profin*” có thể được dùng để chỉ tập con của cú pháp dòng bit. “*Bậc*” và “*mức*” có thể được chỉ rõ trong mỗi *profin*. Mức của bậc có thể là tập hợp các ràng buộc xác định được đặt vào các giá trị của các phần tử cú pháp trong dòng bit. Các ràng buộc này có thể là các giới hạn đơn giản về giá trị.

Theo cách khác, các ràng buộc có thể là dạng các ràng buộc về các tổ hợp số học của các giá trị (ví dụ, chiều rộng hình ảnh nhân với chiều cao hình ảnh nhân với số hình ảnh được giải mã mỗi giây). Mức được chỉ rõ cho bậc thấp hơn bị ràng buộc nhiều hơn mức được chỉ rõ cho bậc cao hơn. Theo ví dụ của sáng chế, mục “các đặc tả của mức chung” (tức là mục A.4.1) của phiên bản dự thảo làm việc HEVC 8 được đặt tên lại là “các đặc tả của mức và bậc chung,” và phần nội dung được thay đổi như sau. Bảng A-1 có thể vẫn giống như trong phiên bản dự thảo làm việc HEVC 8.

Để so sánh các tính năng của bậc, bậc có `general_bậc_flag` bằng 0 sẽ được coi là bậc thấp hơn so với bậc có `general_bậc_flag` bằng 1.

Để so sánh các tính năng của mức, đối với bậc cụ thể, mức thấp hơn có giá trị `general_level_idc` thấp hơn.

Dưới đây mô tả các ràng buộc trong phụ lục này.

- Gọi đơn vị truy nhập *n* là đơn vị truy nhập thứ *n* theo thứ tự giải mã, với đơn vị truy nhập thứ nhất là đơn vị truy nhập 0 (tức là, đơn vị truy nhập thứ 0).
- Gọi hình ảnh *n* là hình ảnh mã hóa hoặc hình ảnh giải mã tương ứng của đơn vị truy nhập *n*.
- Đặt biến `fR` bằng $1 \div 300$.

Dòng bit tương thích với *profin* ở mức cụ thể sẽ tuân theo các ràng buộc sau đối với mỗi lần kiểm tra sự tương thích dòng bit như được chỉ rõ trong phụ lục C:

- a) Thời gian lấy ra danh định của đơn vị truy nhập n (với $n > 0$) từ CPB như được chỉ rõ trong điểm C.2.2 đáp ứng ràng buộc là $t_{r,n}(n) - t_r(n-1)$ bằng hoặc lớn hơn $\text{Max}(\text{PicSizeInSamplesY} \div \text{MaxLumaSR}, fR)$ đối với giá trị của PicSizeInSamplesY của hình ảnh $n-1$, trong đó MaxLumaSR là giá trị được xác định trong bảng A-1 áp dụng cho hình ảnh $n-1$.
- b) Sự khác biệt giữa thời gian xuất ra liên tiếp của các hình ảnh từ DPB như được chỉ rõ trong điểm C.3.2 đáp ứng ràng buộc là $\Delta t_{o,dpb}(n) \geq \text{Max}(\text{PicSizeInSamplesY} \div \text{MaxLumaSR}, fR)$ đối với giá trị của PicSizeInSamplesY của hình ảnh n , trong đó MaxLumaSR là giá trị được xác định trong bảng A-1 đối với hình ảnh n , miễn là hình ảnh n là hình ảnh được xuất ra và không phải là hình ảnh cuối cùng của dòng bit được xuất ra.
- c) $\text{PicSizeInSamplesY} \leq \text{MaxLumaPS}$, trong đó MaxLumaPS được chỉ rõ trong bảng A-1.
- d) $\text{pic_width_in_luma_samples} \leq \text{Sqrt}(\text{MaxLumaPS} * 8)$
- e) $\text{pic_height_in_luma_samples} \leq \text{Sqrt}(\text{MaxLumaPS} * 8)$
- f) $\text{sps_max_dec_pic_buffering}[\text{TargetDecHighestTid}] \leq \text{MaxDpbSize}$, trong đó MaxDpbSize được tìm ra như được chỉ rõ dưới đây:

```

if ( PicSizeInSamplesY <= ( MaxLumaPS >> 2 ) )
    MaxDpbSize = Min( 4 * MaxDpbPicBuf, 16 )
else if ( PicSizeInSamplesY <= ( MaxLumaPS >> 1 ) )
    MaxDpbSize = Min( 2 * MaxDpbPicBuf, 16 )
else if ( PicSizeInSamplesY <= ( MaxLumaPS << 1 ) / 3 )
    MaxDpbSize = Min( (3 * MaxDpbPicBuf) >> 1, 16 )
else if ( PicSizeInSamplesY <= ( ( 3 * MaxLumaPS ) >> 2 ) )
    MaxDpbSize = Min( (4 * MaxDpbPicBuf) / 3, 16 )
else
    MaxDpbSize = MaxDpbPicBuf

```

trong đó MaxLumaPS được chỉ rõ trong bảng A-1 và MaxDpbPicBuf là bảng 6.

Bảng A-1 chỉ rõ các giới hạn của mỗi mức của mỗi bậc. Sử dụng cột thông số MinCR của bảng A-1 được chỉ rõ trong điểm A.4.2.

Bậc và mức mà dòng bit tương thích sẽ được chỉ báo bởi các phần tử cú pháp `general_tier_flag` và `general_level_idc` như sau.

- `general_tier_flag` bằng 0 chỉ báo sự tương thích với bậc chính, và `general_tier_flag` bằng 1 chỉ báo sự tương thích với bậc cao, theo các đặc tả về ràng buộc bậc trong bảng A-1. `general_tier_flag` bằng 0 đối với các mức dưới mức 4 (tương ứng với các mục nhập trong bảng A-1 được đánh dấu "-"). Các giới hạn mức ngoài MaxBR và MaxCPB trong bảng A-1 là chung cho cả bậc chính và bậc cao.
- `general_level_idc` sẽ được đặt bằng giá trị bằng 30 lần mức được chỉ rõ trong bảng A-1.

Như được chỉ rõ trong mục (f) trên đây, dòng bit tương thích với profin ở mức cụ thể tuân theo ràng buộc là `sps_max_dec_pic_buffering[TargetDecHighestTid] <= MaxDpbSize`.

`TargetDecHighestTid` có thể được xác định theo cách được mô tả trong bản mô tả sáng chế. Ngược lại, phiên bản dự thảo làm việc HEVC 8 chỉ rõ ở mục (f) rằng dòng bit tương thích với profin ở mức xác định tuân theo ràng buộc là `sps_max_dec_pic_buffering[sps_max_temporal_layers_minus1] <= MaxDpbSize`. Như được chỉ rõ trong bản mô tả sáng chế, các thông số `sps_max_dec_pic_buffering[i]` có thể không được chọn thích hợp trong các giới hạn mức. Thay thế `sps_max_temporal_layers_minus1` bằng `TargetDecHighestTid` là chỉ số `i` của `sps_max_dec_pic_buffering[i]` có thể, theo một hoặc nhiều kỹ thuật theo sáng chế, đảm bảo rằng các giới hạn mức được chọn phù hợp với `i` bằng giá trị được xác định rõ của `TargetDecHighestTid`.

Theo cách này, quy trình giải mã của HRD có thể giải mã, từ SPS, mảng phần tử cú pháp (ví dụ, `sps_max_dec_pic_buffering[]`) trong đó mỗi phần tử cú pháp trong mảng này chỉ báo dung lượng tối đa cần thiết của DPB của HRD. Hơn thế nữa, khi thiết bị tiến hành hoạt động HRD, thiết bị này có thể xác định, dựa vào ký hiệu nhận dạng theo thời gian cao nhất của đích (ví dụ, `TargetDecHighestTid`), phần tử cú pháp cụ thể trong mảng

(ví dụ, `sps_max_dec_pic_buffering[TargetDecHighestTid]`). Hơn thế nữa, thiết bị có thể xác định rằng dòng bit không tuân theo tiêu chuẩn mã hóa dữ liệu video khi giá trị của phần tử cú pháp cụ thể lớn hơn dung lượng DPB tối đa (ví dụ, `MaxDpbSize`).

Hơn thế nữa, theo một hoặc nhiều kỹ thuật làm ví dụ theo sáng chế, mục A.4.2 của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi như sau. Mục A.4.2 của phiên bản dự thảo làm việc HEVC 8 mô tả giới hạn mức riêng của profin cho profin chính. Bảng A-2 có thể vẫn giống như trong phiên bản dự thảo làm việc HEVC 8.

Dòng bit tương thích với profin chính ở bậc và mức cụ thể sẽ tuân theo các ràng buộc sau của sự kiểm tra tương thích dòng bit như được chỉ rõ trong phức lục C:

- a) Số lát (với *dependent_slice_flag* bằng 0 hoặc 1) trong hình ảnh nhỏ hơn hoặc bằng `MaxSlicesPerPicture`, trong đó `MaxSlicesPerPicture` được chỉ rõ trong bảng A-1.
- b) Đối với các thông số HRD VCL, $\text{BitRate}[\text{SchedSelIdx}] \leq \text{cpbBrVclFactor} * \text{MaxBR}$ và $\text{CpbSize}[\text{SchedSelIdx}] \leq \text{cpbBrVclFactor} * \text{MaxCPB}$ đối với ít nhất một giá trị của `SchedSelIdx`, trong khi đó `cpbBrVclFactor` được chỉ rõ trong bảng A-2 và $\text{BitRate}[\text{SchedSelIdx}]$ và $\text{CpbSize}[\text{SchedSelIdx}]$ được xác định như sau.
 - Nếu `vcl_hrd_parameters_present_flag` bằng 1, $\text{BitRate}[\text{SchedSelIdx}]$ và $\text{CpbSize}[\text{SchedSelIdx}]$ được xác định bằng phương trình E-45 và E-46, tương ứng, sử dụng các phần tử cú pháp được chọn như được chỉ rõ trong điểm C.1.
 - Mặt khác (`vcl_hrd_parameters_present_flag` bằng 0), $\text{BitRate}[\text{SchedSelIdx}]$ và $\text{CpbSize}[\text{SchedSelIdx}]$ được suy ra như được chỉ rõ trong điểm E.2.3 đối với các thông số HRD VCL.

`MaxBR` và `MaxCPB` được chỉ rõ trong bảng A-1 trong các đơn vị của `cpbBrVclFactor` tính theo bit/giây và `cpbBrVclFactor` tính theo bit, tương ứng. Dòng bit sẽ đáp ứng các điều kiện này vì ít nhất một giá trị của `SchedSelIdx` nằm trong khoảng từ 0 đến `cpb_cnt_minus1[TargetDecHighestTid]`, bao gồm cả giá trị đầu mút.

c) Đối với các thông số HRD NAL, $\text{BitRate}[\text{SchedSelIdx}] \leq \text{cpbBrNalFactor} * \text{MaxBR}$ và $\text{CpbSize}[\text{SchedSelIdx}] \leq \text{cpbBrNalFactor} * \text{MaxCPB}$ đối với ít nhất một giá trị của SchedSelIdx , trong đó cpbBrNalFactor được chỉ rõ trong bảng A-2 và $\text{BitRate}[\text{SchedSelIdx}]$ và $\text{CpbSize}[\text{SchedSelIdx}]$ được xác định như sau.

- Nếu $\text{nal_hrd_parameters_present_flag}$ bằng 1, $\text{BitRate}[\text{SchedSelIdx}]$ và $\text{CpbSize}[\text{SchedSelIdx}]$ được xác định bằng phương trình E-45 và E-46, tương ứng, sử dụng các phần tử cú pháp được chọn như được chỉ rõ trong điểm C.1.
- Mặt khác ($\text{nal_hrd_parameters_present_flag}$ bằng 0), $\text{BitRate}[\text{SchedSelIdx}]$ và $\text{CpbSize}[\text{SchedSelIdx}]$ được suy ra như được chỉ rõ trong điểm E.2.3 đối với các thông số HRD NAL.

MaxBR và MaxCPB được chỉ rõ trong bảng A-1 trong các đơn vị của cpbBrNalFactor tính theo bit/giây và cpbBrNalFactor tính theo bit, tương ứng. Dòng bit sẽ đáp ứng các điều kiện này vì ít nhất một giá trị của SchedSelIdx nằm trong khoảng từ 0 đến $\text{cpb_cnt_minus1}[\text{TargetDecHighestTid}]$, bao gồm cả giá trị đầu mút.

d) Tổng của các biến NumBytesInNALunit đối với đơn vị truy nhập 0 là nhỏ hơn hoặc bằng $1.5 * (\text{Max}(\text{PicSizeInSamplesY}, \text{fR} * \text{MaxLumaSR}) + \text{MaxLumaSR} * (\text{t}_r(0) - \text{t}_{r,n}(0))) \div \text{MinCR}$ đối với giá trị của PicSizeInSamplesY của hình ảnh 0, trong đó MaxLumaPR và MinCR là các giá trị được chỉ rõ trong bảng A-1 áp dụng cho hình ảnh 0.

e) Tổng của các biến NumBytesInNALunit đối với đơn vị truy nhập n với $n > 0$ là nhỏ hơn hoặc bằng $1.5 * \text{MaxLumaSR} * (\text{t}_r(n) - \text{t}_r(n-1)) \div \text{MinCR}$, trong đó MaxLumaSR và MinCR là các giá trị được chỉ rõ trong bảng A-1 áp dụng cho hình ảnh n .

f) Đối với mức 5 và mức cao hơn, các biến CtbSizeY sẽ bằng 32 hoặc 64.

g) Giá trị của NumPocTotalCurr sẽ nhỏ hơn hoặc bằng 8.

h) Giá trị của num_tile_columns_minus1 sẽ nhỏ hơn MaxTileCols và num_tile_rows_minus1 sẽ nhỏ hơn MaxTileRows, trong đó MaxTileCols và MaxTileRows là như được chỉ rõ trong bảng A-1.

Như được chỉ rõ trong bản mô tả sáng chế, các thông số cpb_cnt_minus1[i] có thể không được chọn thích hợp trong các giới hạn mức. Phiên bản dự thảo làm việc HEVC 8 chỉ rõ rằng “dòng bit sẽ đáp ứng các điều kiện này vì ít nhất một giá trị của SchedSelIdx nằm trong khoảng từ 0 đến cpb_cnt_minus1, bao gồm cả giá trị đầu mút...” Việc chỉ rõ TargetDecHighestTid là chỉ số i của cpb_cnt_minus1[i] có thể, theo một hoặc nhiều kỹ thuật theo sáng chế, đảm bảo rằng các giới hạn mức được chọn thích hợp với i bằng giá trị được xác định rõ của TargetDecHighestTid.

Hơn thế nữa, theo một hoặc nhiều kỹ thuật theo sáng chế, điểm C.1 chung trong phụ lục C của phiên bản dự thảo làm việc HEVC 8 có thể được sửa đổi. Các hình vẽ C-1 và C-2 của điểm C.1 của Phiên bản dự thảo làm việc HEVC 8 có thể vẫn giống như trong HEVC WD8. Nội dung của điểm C.1 của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi như sau.

Phụ lục này chỉ rõ bộ giải mã chuẩn giả định (HRD) và sử dụng bộ giải mã này để kiểm tra sự tương thích dòng bit và bộ giải mã.

Hai loại dòng bit được kiểm tra sự tương thích HRD đối với đặc tả này. Loại dòng bit thứ nhất, được gọi là dòng bit loại I, là dòng đơn vị NAL chỉ chứa các đơn vị NAL có VCL và các đơn vị NAL với nal_unit_type bằng FD_NUT (đơn vị NAL chứa dữ liệu đệm) đối với tất cả các đơn vị truy nhập trong dòng bit. Loại dòng bit thứ hai, được gọi là dòng bit loại II, chứa, ngoài đơn vị NAL có VCL và đơn vị NAL chứa dữ liệu đệm đối với tất cả các đơn vị truy nhập trong dòng bit, ít nhất một trong số các đơn vị dưới đây:

- Các đơn vị NAL khác không có VCL ngoài đơn vị NAL chứa dữ liệu đệm,
- tất cả các phần tử cú pháp leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes, và trailing_zero_8bits tạo nên dòng byte từ dòng đơn vị NAL (như được chỉ rõ trong phụ lục B).

Hình vẽ C-1 thể hiện các loại điểm tương thích dòng bit được kiểm tra bởi HRD.

Các phần tử cú pháp của đơn vị NAL không có VCL (hoặc các giá trị ngầm định của một vài phần tử cú pháp) cần cho HRD được chỉ rõ trong các điểm ngữ nghĩa của mệnh đề 7, phụ lục D và E.

Hai loại thông số HRD (thông số HRD NAL và thông số HRD VCL) được sử dụng. Thông số HRD được báo hiệu thông qua cấu trúc cú pháp của bộ thông số video hoặc thông qua thông tin sử dụng dữ liệu video như được chỉ rõ trong các điểm E.1 và E.2, là một phần của cấu trúc cú pháp của bộ thông số chuỗi.

Có thể cần tiến hành kiểm tra nhiều lần để kiểm tra sự tương thích dòng bit. Đối với mỗi lần kiểm tra, áp dụng các bước sau theo thứ tự được liệt kê:

1. *Điểm hoạt động kiểm tra, được ký hiệu là TargetOp, được chọn. TargetOp được xác định bởi OpLayerIdSet bằng targetOpLayerIdSet và OpTid bằng targetOpTid. targetOpLayerIdSet chứa tập hợp các giá trị của nuh_reserved_zero_6bits có mặt trong tập con dòng bit gắn với TargetOp và sẽ là tập con các giá trị của nuh_reserved_zero_6bits có mặt trong dòng bit kiểm tra. targetOpTid bằng TemporalId lớn nhất có mặt trong tập con dòng bit gắn với TargetOp và nhỏ hơn hoặc bằng TemporalId lớn nhất có mặt trong dòng bit kiểm tra.*
2. *TargetDecLayerIdSet được đặt bằng targetOpLayerIdSet, TargetDecHighestTid được đặt bằng targetOpTid, và BitstreamToDecode được đặt làm đầu ra của quy trình tách dòng bit con như được chỉ rõ trong điểm 10.1 với TargetDecHighestTid và TargetDecLayerIdSet là đầu vào.*
3. *Cấu trúc cú pháp hrd_parameters() và cấu trúc cú pháp sub_layer_hrd_parameters() áp dụng được cho TargetOp được chọn. Nếu TargetDecLayerIdSet chứa duy nhất giá trị 0, thì cấu trúc cú pháp hrd_parameters() trong bộ thông số chuỗi hoạt động được chọn. Mặt khác, cấu trúc cú pháp hrd_parameters() nằm trong bộ thông số chuỗi hoạt động (hoặc được cung cấp thông qua phương tiện ngoài) và cấu trúc cú pháp mà có tập hợp các giá trị được xác định bởi op_layer_id[opIdx][i] với i nằm trong khoảng bao hàm từ 0 đến op_num_layer_id_values_minus1[opIdx], giống với TargetDecLayerIdSet được chọn. Trong cấu trúc cú pháp hrd_parameters() được chọn, nếu*

BitstreamToDecode là dòng bit loại I, thì cấu trúc cú pháp *sub_layer_hrd_parameters(TargetDecHighestTid)* tức thì tuân theo điều kiện "*if(vcl_hrd_parameters_present_flag)*" được chọn (trong trường hợp biến *NalHrdModeFlag* được đặt bằng 0), mặt khác (*BitstreamToDecode* là dòng bit loại II), thì cấu trúc cú pháp *sub_layer_hrd_parameters(TargetDecHighestTid)* tức thì tuân theo điều kiện "*if(vcl_hrd_parameters_present_flag)*" (trong trường hợp biến *NalHrdModeFlag* được đặt bằng 0) hoặc điều kiện "*if(nal_hrd_parameters_present_flag)*" (trong trường hợp biến *NalHrdModeFlag* được đặt bằng 1) được chọn, và tất cả các đơn vị NAL không có VCL ngoại trừ đơn vị NAL chứa dữ liệu đệm được lấy ra từ *BitstreamToDecode* ở trường hợp trước và kết quả được phân định cho *BitstreamToDecode*.

4. Đơn vị truy nhập gắn với thông báo SEI về chu kỳ nhớ đệm áp dụng được cho *TargetOp* được chọn làm điểm khởi đầu HRD và được dùng để gọi là đơn vị truy nhập 0.
5. Các thông báo SEI chứa thông tin định thời được chọn. Thông báo SEI về chu kỳ nhớ đệm được mã hóa trong đơn vị truy nhập 0 và áp dụng được cho *TargetOp*, như được chỉ báo bởi cấu trúc cú pháp *applicable_operation_points()*, được chọn. Đối với mỗi đơn vị truy nhập trong *BitstreamToDecode* bắt đầu từ đơn vị truy nhập 0, thông báo SEI định thời hình ảnh gắn với điểm truy nhập và áp dụng được cho *TargetOp*, như được chỉ báo bởi cấu trúc cú pháp *applicable_operation_points()*, được chọn, và khi *SubPicCpbFlag* bằng 1 và *sub_pic_cpb_params_in_pic_timing_sei_flag* bằng 0, các thông báo SEI định thời hình ảnh con gắn với các đơn vị giải mã trong đơn vị truy nhập và áp dụng cho *TargetOp*, như được chỉ báo bằng cấu trúc cú pháp *applicable_operation_points()*, được chọn.
6. Giá trị của *SchedSelIdx* được chọn. *SchedSelIdx* được chọn sẽ nằm trong khoảng từ 0 đến *cpb_cnt_minus1[TargetDecHighestTid]*, bao gồm cả giá trị đầu mút trong đó *cpb_cnt_minus1[TargetDecHighestTid]* được tìm

thấy trong cấu trúc cú pháp *sub_layer_hrd_parameters(TargetDecHighestTid)* như được chọn trên đây.

7. Độ trễ lấy ra từ CPB ban đầu và độ lệch độ trễ được chọn và các đơn vị truy nhập TFD gắn với đơn vị truy nhập 0 có thể được lấy ra từ *BitstreamToDecode*. Nếu hình ảnh mã hóa trong đơn vị truy nhập 0 có *nal_unit_type* bằng *CRA_NUT* hoặc *BLA_W_LP*, và *rap_cpb_params_present_flag* trong thông báo SEI về chu kỳ nhớ đệm được chọn là bằng 1, mỗi độ trễ lấy ra từ CPB ban đầu ngầm định và độ lệch độ trễ được biểu diễn bởi *initial_cpb_removal_delay[SchedSelIdx]* và *initial_cpb_removal_delay_offset[SchedSelIdx]* tương ứng với *NalHrdModeFlag* (trong trường hợp này biến *DefaultInitCpbParamsFlag* được đặt bằng 1) hoặc độ trễ lấy ra từ CPB ban đầu khác và độ lệch độ trễ được biểu diễn bởi *initial_alt_cpb_removal_delay[SchedSelIdx]* và *initial_alt_cpb_removal_delay_offset[SchedSelIdx]* tương ứng với *NalHrdModeFlag* (trong trường hợp này biến *DefaultInitCpbParamsFlag* được đặt bằng 0) được chọn, và các đơn vị truy nhập TFD gắn với đơn vị truy nhập 0 được lấy ra từ *BitstreamToDecode* trong trường hợp sau và kết quả được phân định cho *BitstreamToDecode*. Mặt khác, độ trễ lấy ra từ CPB ban đầu ngầm định và độ lệch độ trễ được chọn (trong trường hợp này biến *DefaultInitCpbParamsFlag* được đặt bằng 1).

Số lần kiểm tra sự tương thích dòng bit được tiến hành là bằng $N1 * N2 * N3 * (N4 * 2 + N5)$, trong đó các giá trị của *N1*, *N2*, *N3*, *N4* và *N5* được chỉ rõ như sau.

- *N1* là số điểm hoạt động nằm trong dòng bit kiểm tra.
- Nếu *BitstreamToDecode* là dòng bit loại I, *N2* là bằng 1, mặt khác (*BitstreamToDecode* là dòng bit loại II) *N2* bằng 2.
- *N3* bằng *cpb_cnt_minus1[TargetDecHighestTid] + 1*.
- *N4* là số đơn vị truy nhập gắn với thông báo SEI về chu kỳ nhớ đệm áp dụng được cho *TargetOp* trong *BitstreamToDecode*, trong đó hình ảnh mã hóa ở mỗi đơn vị truy nhập này có *nal_unit_type* bằng *CRA_NUT* hoặc *BLA_W_LP*,

và thông báo SEI về chu kỳ nhớ đệm đi kèm áp dụng được cho TargetOp có rap_cpb_params_present_flag bằng 1.

- N5 là số đơn vị truy nhập gắn với thông báo SEI về chu kỳ nhớ đệm áp dụng được cho TargetOp trong BitstreamToDecode, trong đó hình ảnh mã hóa ở mỗi đơn vị truy nhập có nal_unit_type không bằng một trong số CRA_NUT và BLA_W_LP, hoặc thông báo SEI về chu kỳ nhớ đệm đi kèm áp dụng được cho TargetOp có rap_cpb_params_present_flag bằng 0.

Khi BitstreamToDecode là dòng bit loại II, nếu cấu trúc cú pháp sub_layer_hrd_parameters(TargetDecHighestTid) đáp ứng ngay điều kiện "if(vcl_hrd_parameters_present_flag)" được chọn, thì tiến hành kiểm tra ở các điểm tương thích loại I được thể hiện trong hình vẽ C-1, và chỉ VCL và đơn vị NAL chứa dữ liệu đệm được tính toán tốc độ bit đầu vào và bộ nhớ CPB; mặt khác (cấu trúc cú pháp sub_layer_hrd_parameters(TargetDecHighestTid) đáp ứng ngay điều kiện "if(nal_hrd_parameters_present_flag)" được chọn, tiến hành kiểm tra ở điểm tương thích loại II được thể hiện trong hình vẽ C-1, và tất cả các đơn vị NAL (của dòng đơn vị NAL loại II) hoặc tất cả các byte (của dòng byte) được tính toán cho tốc độ bit đầu vào và bộ nhớ CPB.

Lưu ý 3 – Các thông số HRD NAL được thiết lập bởi giá trị của SchedSelIdx cho điểm tương thích loại II được thể hiện trong hình vẽ C-1 cũng đủ để thiết lập sự tương thích HRD VCL cho điểm tương thích loại I được thể hiện trong hình vẽ C-1 đối với cùng giá trị của InitCpbRemovalDelay[SchedSelIdx], BitRate[SchedSelIdx], và CpbSize[SchedSelIdx] trong trường hợp VBR (cbr_flag[SchedSelIdx] bằng 0). Điều này là do dữ liệu chảy vào điểm tương thích loại I là tập con của dữ liệu chảy vào điểm tương thích loại II và do, đối với trường hợp VBR, CPB được cho phép rỗng và để rỗng cho đến khi thời gian hình ảnh tiếp theo được lập lịch biểu để bắt đầu đi đến. Ví dụ, khi giải mã chuỗi video mã hóa tương thích với một hoặc nhiều profin được chỉ rõ trong phụ lục A bằng cách sử dụng quy trình giải mã được chỉ rõ trong các mệnh đề 2-9, khi các thông số HRD NAL được cung cấp cho điểm tương thích loại II mà không chỉ nằm trong các giới hạn được đặt ra cho các thông số HRD NAL đối với sự tương thích profin ở mục c) của điểm A.4.2 mà còn nằm trong các

giới hạn được đặt ra cho các thông số HRD VCL đối với sự tương thích profin ở mục b) của điểm A.4.2, sự tương thích của HRD VCL đối với điểm tương thích loại I cũng được đảm bảo nằm trong các giới hạn của mục b) của điểm A.4.2.

Tất cả các *bộ thông số video*, bộ thông số chuỗi và bộ thông số hình ảnh được đề cập trong các đơn vị NAL có VCL và thông báo SEI định thời hình ảnh và thông báo SEI về chu kỳ nhớ đệm tương ứng sẽ được truyền đến HRD, đúng lúc, trong dòng bit (bởi các đơn vị NAL không có VCL), hoặc bằng phương tiện khác không được chỉ rõ trong đặc tả này.

Trong các phụ lục C, D, và E, đặc tả về “sự có mặt” của đơn vị NAL không có VCL cũng đáp ứng khi các đơn vị NAL này (hoặc chỉ một vài trong số chúng) được truyền đến bộ giải mã (hoặc đến HRD) bằng các phương tiện khác không được chỉ rõ trong đặc tả này. Để tính toán số bit, chỉ tính các bit phù hợp thực sự có mặt trong dòng bit.

Lưu ý 1 – Theo một ví dụ, việc đồng bộ hóa đơn vị NAL không có VCL, được truyền bằng phương tiện không có mặt trong dòng bit, với đơn vị NAL có mặt trong dòng bit, có thể được thực hiện bằng cách chỉ báo hai điểm trong dòng bit, giữa hai điểm trong dòng bit này đơn vị NAL không có VCL sẽ có mặt trong dòng bit khiến cho bộ mã hóa quyết định truyền nó đi trong dòng bit.

Khi nội dung của đơn vị NAL không có VCL được truyền để ứng dụng cho một vài phương tiện không có mặt trong dòng bit, thì dạng biểu diễn nội dung của đơn vị NAL không có VCL không cần sử dụng cú pháp giống như được chỉ rõ trong đặc tả này.

Lưu ý 2 – Khi thông tin HRD được chứa trong dòng bit, có thể kiểm tra sự tương thích của dòng bit với các yêu cầu của điểm này chỉ dựa vào thông tin chứa trong dòng bit. Khi thông tin HRD không có mặt trong dòng bit, như là trường hợp đối với tất cả các dòng bit loại I “độc lập”, sự tương thích có thể chỉ được kiểm tra khi dữ liệu HRD được cung cấp bởi một vài phương tiện khác không được chỉ rõ trong đặc tả này.

HRD chứa bộ nhớ đệm hình ảnh mã hóa (coded picture buffer-CPB), quy trình giải mã tức thời, bộ nhớ đệm hình ảnh giải mã (decoded picture buffer-DPB), và cắt bỏ đầu ra như được thể hiện trong Fig.C-2.

Đối với mỗi lần kiểm tra sự tương thích dòng bit, dung lượng CPB (số bit) là $CpbSize[SchedSelIdx]$ như được chỉ rõ bằng phương trình E-46, trong đó $SchedSelIdx$ và thông số HRD được chọn như được chỉ rõ trên đây trong điểm này. Dung lượng DPB (số bộ nhớ đệm lưu trữ hình ảnh) là $sps_max_dec_pic_buffering[TargetDecHighestTid]$.

Biến $SubPicCpbPreferredFlag$ được chỉ rõ bằng phương tiện ngoài, hoặc nếu không được chỉ rõ bằng phương tiện ngoài, được đặt bằng 0.

Biến $SubPicCpbFlag$ được tìm ra như sau:

$$SubPicCpbFlag = SubPicCpbPreferredFlag \&\& \\ sub_pic_cpb_params_present_flag(C-1)$$

Nếu $SubPicCpbFlag$ bằng 0, CPB hoạt động ở mức đơn vị truy nhập và mỗi đơn vị giải mã là đơn vị truy nhập. Mặt khác, CPB hoạt động ở mức hình ảnh con và mỗi đơn vị giải mã là tập con của đơn vị truy nhập.

HRD hoạt động như sau. Dữ liệu gắn với đơn vị giải mã mà chảy vào CPB theo lịch biểu đến cụ thể được phân phối bởi HSS. Dữ liệu gắn với mỗi đơn vị giải mã được lấy ra và được giải mã tức thời bằng quy trình giải mã tức thời ở thời điểm lấy ra đơn vị giải mã từ CPB. Mỗi hình ảnh giải mã được đặt trong DPB. Hình ảnh giải mã được lấy ra từ DPB như được chỉ rõ trong điểm C.3.1 hoặc điểm C.5.2.

Hoạt động của CPB đối với mỗi lần kiểm tra sự tương thích dòng bit được chỉ rõ trong điểm C.2. Hoạt động giải mã tức thì được chỉ rõ trong mệnh đề 2-9. Hoạt động của DPB đối với mỗi lần kiểm tra sự tương thích dòng bit được chỉ rõ trong điểm C.3. Việc cắt bỏ đầu ra đối với mỗi lần kiểm tra sự tương thích dòng bit được chỉ rõ trong điểm C.3.2 và điểm C.5.2.

Thông tin HSS và HRD liên quan đến số lịch biểu phân phối được liệt kê và tốc độ bit đi kèm và dung lượng bộ nhớ đệm được chỉ rõ trong các điểm E.1.1, E.1.2, E.2.1, và E.2.2. HRD được khởi tạo như được chỉ rõ bởi thông báo SEI về chu kỳ nhớ đệm như được chỉ rõ trong điểm D.1.1 và D.2.1. Việc định thời gian lấy ra các đơn vị giải mã từ CPB và việc định thời gian xuất ra hình ảnh giải mã từ DPB được chỉ rõ trong thông báo SEI định thời hình ảnh như được chỉ rõ trong điểm

D.1.2 và D.2.1. Toàn bộ thông tin định thời liên quan đến đơn vị giải mã cụ thể sẽ đến trước thời gian lấy ra đơn vị giải mã từ CPB.

Các yêu cầu về sự tương thích dòng bit được chỉ rõ trong điểm C.4, và HRD được sử dụng để kiểm tra sự tương thích của bộ giải mã như được chỉ rõ trong điểm C.5.

Lưu ý 3 – Khi sự tương thích được bảo đảm bằng giả định là cả tốc độ và nhịp hình ảnh được sử dụng để tạo ra dòng bit so khớp chính xác với các giá trị được báo hiệu trong dòng bit, trong hệ thống thực mỗi tốc độ và nhịp hình ảnh trong số các tốc độ và nhịp hình ảnh này có thể khác so với giá trị được báo hiệu hoặc được chỉ rõ.

Toàn bộ thuật toán trong phụ lục này được thực hiện với các giá trị thực, sao cho không làm lan truyền các sai số làm tròn. Ví dụ, số bit trong CPB trước hoặc sau khi lấy ra đơn vị giải mã không nhất thiết phải là số nguyên.

Biến t_c được tính toán như sau và được gọi là nhịp đồng hồ:

$$t_c = \text{num_units_in_tick} \div \text{time_scale}(C-1)$$

Biến t_{c_sub} được tính toán như sau và được gọi là nhịp đồng hồ hình ảnh con:

$$t_{c_sub} = t_c \div (\text{tick_divisor_minus2} + 2)(C-2)$$

Dưới đây thể hiện các ràng buộc trong phụ lục này:

- Gọi đơn vị truy nhập n là đơn vị truy nhập thứ n theo thứ tự giải mã với đơn vị truy nhập thứ nhất là đơn vị truy nhập 0 (tức là, đơn vị truy nhập thứ 0).
- Gọi hình ảnh n là hình ảnh mã hóa hoặc hình ảnh giải mã của đơn vị truy nhập n .
- Gọi đơn vị giải mã m là đơn vị giải mã thứ m theo thứ tự giải mã với đơn vị giải mã thứ nhất là đơn vị giải mã 0.

Các thay đổi ở mục C.1 của phiên bản dự thảo làm việc HEVC 8 trên đây có thể làm rõ sự kiểm tra tương thích dòng bit. Như được nêu trên đây, khi quy trình giải mã được gọi ra để kiểm tra tương thích dòng bit trong phiên bản dự thảo làm việc HEVC 8, ngữ nghĩa của các phần tử cú pháp không được xác định rõ vì các giá trị của `TargetDecLayerIdSet` và `TargetDecHighestTid` không được thiết lập thích hợp. Các thay

đổi ở mục C.1 làm rõ các định nghĩa của `TargetDecLayerIdSet` và `TargetDecHighestTid`.

Như được thể hiện trong các phần thay đổi ở mục C.1 của phiên bản dự thảo làm việc HEVC 8 trên đây, thiết bị có thể tiến hành hoạt động HRD (như sự kiểm tra tương thích dòng bit) mà chọn điểm hoạt động, xác định tập hợp các ký hiệu nhận dạng lớp đích (`TargetDecLayerIdSet`) của điểm hoạt động và ký hiệu nhận dạng theo thời gian cao nhất (`TargetDecHighestTid`). Hơn thế nữa, trong hoạt động HRD, thiết bị có thể chọn bộ thông số HRD áp dụng được cho điểm hoạt động và sử dụng bộ thông số HRD đã chọn để tạo cấu hình HRD để tiến hành quy trình giải mã. Bộ thông số HRD áp dụng được cho điểm hoạt động cụ thể có thể bao gồm các thông số chỉ rõ độ trễ lấy ra từ CPB ban đầu, dung lượng CPB, tốc độ bit, độ trễ xuất ra từ DPB ban đầu, dung lượng DPB, và v.v.. Hoạt động HRD có thể bao gồm tiến hành quy trình giải mã.

Theo một vài ví dụ, thiết bị có thể chọn, từ trong số một hoặc nhiều bộ thông số HRD (ví dụ, các cấu trúc cú pháp `hrd_parameters()` trong VPS và bộ thông số HRD trong SPS, bộ thông số HRD áp dụng được cho điểm hoạt động. Theo một vài ví dụ, thiết bị có thể xác định bộ thông số HRD trong SPS là áp dụng được cho điểm hoạt động cụ thể khi tập hợp ký hiệu nhận dạng lớp của điểm hoạt động chứa tập hợp gồm tất cả các ký hiệu nhận dạng lớp có mặt trong chuỗi video mã hóa gắn với SPS. Hơn thế nữa, theo một vài ví dụ, thiết bị có thể chọn bộ thông số HRD trong SPS đáp lại việc xác định rằng tập hợp ký hiệu nhận dạng đích (ví dụ, `TargetDecLayerIdSet`) của điểm hoạt động chứa duy nhất giá trị 0. Theo một vài ví dụ, thiết bị có thể chọn bộ thông số HRD trong VPS đáp lại việc xác định rằng tập hợp ký hiệu nhận dạng lớp (ví dụ, `op_layer_id[ ][ ]`) là giống với tập hợp các ký hiệu nhận dạng lớp đích (ví dụ, `TargetDecLayerIdSet`) của điểm hoạt động.

Hơn thế nữa, như được thể hiện trong các phần thay đổi ở mục C.1 của phiên bản dự thảo làm việc HEVC 8 trên đây và các phần khác của bản mô tả, thiết bị có thể giải mã, từ SPS, mảng phần tử cú pháp (`sps_max_dec_pic_buffering[ ]`) mà mỗi phần tử cú pháp chỉ báo dung lượng tối đa cần thiết của DPB của HRD. Thiết bị có thể xác định, dựa vào ký hiệu nhận dạng theo thời gian cao nhất của đích, phần tử cú pháp cụ thể trong mảng (tức là, `sps_max_dec_pic_buffering[TargetDecHighestTid]`). Như được thể hiện trên đây, số bộ nhớ đệm lưu trữ hình ảnh trong DPB được chỉ báo bởi phần tử cú pháp cụ

thể (tức là, dung lượng DPB (số bộ nhớ đệm lưu trữ hình ảnh) là `sps_max_dec_pic_buffering[ TargetDecHighestTid ]`).

Ngoài ra, quy trình giải mã có thể giải mã cấu trúc cú pháp chứa thông số HRD (`hrd_parameters( )`) mà bao gồm bộ thông số HRD đã chọn. Bộ thông số HRD đã chọn bao gồm mảng các phần tử cú pháp (`cbp_cnt_minus1[ ]`) trong đó mỗi phần tử cú pháp chỉ báo số đặc tả CPB khác trong dòng bit. Các thay đổi ở mục C.1 của phiên bản dự thảo làm việc HEVC 8 chỉ rõ rằng khi thiết bị tiến hành hoạt động HRD, thiết bị có thể chọn, dựa vào ký hiệu nhận dạng theo thời gian cao nhất của đích (`TargetDecHighestTid`), phần tử cú pháp cụ thể trong mảng (`cbp_cnt_minus1[TargetDecHighestTid]`) và có thể chọn chỉ số chọn bộ lập lịch biểu (`SchedSelIdx`) nằm trong khoảng từ 0 đến giá trị của phần tử cú pháp cụ thể. Thiết bị này có thể xác định, dựa ít nhất một phần vào chỉ số chọn bộ lập lịch biểu, độ trễ lấy ra từ CPB ban đầu của CPB của HRD.

Mục C.2.1 của phiên bản dự thảo làm việc HEVC 8 đề cập đến việc lấy ra các hình ảnh từ DPB để kiểm tra sự tương thích dòng bit. Theo một hoặc nhiều kỹ thuật làm ví dụ theo sáng chế, mục C.2.1 của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi như sau:

Các đặc tả trong điểm này áp dụng độc lập cho mỗi bộ thông số DPB đã chọn như được chỉ rõ trong điểm C.1.

Việc lấy ra các hình ảnh từ DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân tích cú pháp phần đầu lát của lát thứ nhất của hình ảnh hiện thời) xảy ra tức thời ở thời điểm lấy ra đơn vị giải mã thứ nhất của đơn vị truy nhập n (chứa hình ảnh hiện thời) từ CPB và tiến hành như sau.

Quy trình giải mã bộ hình ảnh chuẩn như được chỉ rõ trong điểm 8.3.2 được gọi ra.

Nếu hình ảnh hiện thời là hình ảnh IDR hoặc hình ảnh BLA, thì có thể áp dụng như sau:

1. Khi hình ảnh IDR hoặc BLA không phải là hình ảnh đầu tiên được giải mã và giá trị của `pic_width_in_luma_samples` hoặc `pic_height_in_luma_samples` hoặc `sps_max_dec_pic_buffering[ TargetDecHighestTid ]` được tìm ra từ bộ thông số chuỗi hoạt động là khác giá trị của `pic_width_in_luma_samples`

hoặc $\text{pic_height_in_luma_samples}$ hoặc $\text{sps_max_dec_pic_buffering}[\text{TargetDecHighestTid}]$ được tìm ra từ bộ thông số chuỗi hoạt động đối với hình ảnh trước đó, tương ứng, $\text{no_output_of_prior_pics_flag}$ được suy ra là bằng 1 bởi HRD, bất kể giá trị thực của $\text{no_output_of_prior_pics_flag}$.

Lưu ý – Bộ giải mã theo các phương án thực hiện này cố gắng xử lý sự thay đổi của hình ảnh hoặc dung lượng DPB tốt hơn so với HRD liên quan đến sự thay đổi của giá trị $\text{pic_width_in_luma_samples}$, $\text{pic_height_in_luma_samples}$, hoặc $\text{sps_max_dec_pic_buffering}[\text{TargetDecHighestTid}]$.

2. Khi $\text{no_output_of_prior_pics_flag}$ bằng 1 hoặc được suy ra là bằng 1, thì tất cả các bộ nhớ đệm lưu trữ hình ảnh trong DPB được làm rỗng mà không xuất ra các hình ảnh trong đó, và độ đầy của DPB được đặt bằng 0.

Tất cả các hình ảnh k trong DPB, mà cả hai điều kiện dưới đây là đúng, được lấy ra từ DPB:

- hình ảnh k được đánh dấu là "không dùng làm chuẩn",
- hình ảnh k có PicOutputFlag bằng 0 hoặc thời gian xuất ra từ DPB nhỏ hơn hoặc bằng thời gian lấy ra đơn vị giải mã đầu tiên từ CPB (được ký hiệu là đơn vị giải mã m) của hình ảnh hiện thời n ; tức là $t_{o,dpb}(k) \leq t_r(m)$

Khi một hình ảnh được lấy ra từ DPB, thì độ đầy của DPB được giảm đi một.

Như được chỉ rõ trong bản mô tả sáng chế, các thông số $\text{sps_max_dec_pic_buffering}[i]$ có thể không được chọn thích hợp trong các hoạt động HRD. Phiên bản dự thảo làm việc HEVC 8 chỉ chỉ báo $\text{sps_max_dec_pic_buffering}[i]$ thay vì $\text{sps_max_dec_pic_buffering}[\text{TargetDecHighestTid}]$, như được nêu trên. Phiên bản dự thảo làm việc HEVC 8 không chỉ báo các ngữ nghĩa của chỉ số i trong mục C.2.1. Việc chỉ rõ $\text{TargetDecHighestTid}$ là chỉ số i của $\text{sps_max_dec_pic_buffering}[i]$ có thể, theo một hoặc nhiều kỹ thuật theo sáng chế, đảm bảo rằng i bằng giá trị được xác định rõ của $\text{TargetDecHighestTid}$ được sử dụng trong $\text{sps_max_dec_pic_buffering}[i]$ khi tiến hành hoạt động HRD lấy hình ảnh từ DPB.

Như được thể hiện trong các thay đổi ở mục C.2.1 của phiên bản dự thảo làm việc HEVC 8 trên đây, thiết bị có thể giải mã, từ SPS có hiệu lực đối với hình ảnh hiện thời, mảng các phần tử cú pháp thứ nhất (`sps_max_dec_pic_buffering[]`) trong đó mỗi phần tử cú pháp chỉ báo dung lượng tối đa cần thiết của DPB của HRD. Ngoài ra, thiết bị có thể giải mã, từ SPS có hiệu lực đối với hình ảnh trước đó, mảng phần tử cú pháp thứ hai (`sps_max_dec_pic_buffering[]`) trong đó mỗi phần tử cú pháp chỉ báo dung lượng tối đa cần thiết của DPB của HRD. Thiết bị có thể xác định, dựa vào ký hiệu nhận dạng theo thời gian cao nhất của đích (`TargetDecHighestTid`), phần tử cú pháp thứ nhất trong mảng thứ nhất (`sps_max_dec_pic_buffering[TargetDecHighestTid]`). Ngoài ra, thiết bị có thể xác định, dựa vào ký hiệu nhận dạng theo thời gian cao nhất của đích, phần tử cú pháp thứ hai trong mảng thứ hai (`sps_max_dec_pic_buffering[TargetDecHighestTid]`). Khi hình ảnh hiện thời là hình ảnh làm mới quy trình giải mã tức thời (Instantaneous decoding refresh-IDR) hoặc hình ảnh truy nhập liên kết hỏng (Broken Link Access-BLA) và giá trị của phần tử cú pháp thứ nhất khác giá trị của phần tử cú pháp thứ hai, thì thiết bị có thể suy ra giá trị của phần tử cú pháp thứ ba (`no_output_of_prior_pics_flag`) bất kể giá trị được chỉ báo bởi phần tử cú pháp thứ ba. Phần tử cú pháp thứ ba có thể chỉ rõ cách xử lý hình ảnh giải mã trước trong DPB sau khi giải mã hình ảnh IDR hoặc hình ảnh BLA.

Hình ảnh IDR có thể là hình ảnh điểm truy cập ngẫu nhiên (random access point-RAP) mà mỗi đoạn lát có `nal_unit_type` bằng `IDR_W_LP` hoặc `IDR_N_LP`. Hình ảnh IDR chỉ chứa lát I, và có thể là hình ảnh đầu tiên trong dòng bit theo thứ tự giải mã, hoặc có thể xuất hiện sau trong dòng bit. Hình ảnh IDR có `nal_unit_type` bằng `IDR_N_LP` không có các hình ảnh dẫn gắn kèm có mặt trong dòng bit. Hình ảnh dẫn là hình ảnh đứng trước hình ảnh RAP đi kèm theo thứ tự xuất ra. Hình ảnh IDR có `nal_unit_type` bằng `IDR_W_LP` không có các hình ảnh được đánh dấu loại bỏ (tagged-for-discard-TFD) gắn kèm có mặt trong dòng bit, nhưng có thể có các hình ảnh DLP gắn kèm trong dòng bit.

Hình ảnh BLA là hình ảnh RAP mà mỗi đoạn lát có `nal_unit_type` bằng `BLA_W_TFD`, `BLA_W_DLP` hoặc `BLA_N_LP`. Hình ảnh BLA có `nal_unit_type` bằng `BLA_W_TFD` có thể có các hình ảnh TFD gắn kèm có mặt trong dòng bit. Hình ảnh BLA có `nal_unit_type` bằng `BLA_N_LP` không có hình ảnh dẫn gắn kèm có mặt trong dòng bit. Hình ảnh BLA có `nal_unit_type` bằng `BLA_W_DLP` không có hình ảnh TFD

gắn kèm có mặt trong dòng bit, nhưng có thể có các hình ảnh DLP gắn kèm trong dòng bit.

Phần C.3 của phiên bản dự thảo làm việc HEVC 8 mô tả các hoạt động tương thích dòng bit. Theo một hoặc nhiều kỹ thuật làm ví dụ theo sáng chế, phần C.3 của phiên bản dự thảo làm việc HEVC 8 có thể được cải biến như sau:

Dòng bit chứa dữ liệu mã hóa tương thích với đặc tả này sẽ đáp ứng tất cả các yêu cầu được chỉ rõ trong điểm này.

Dòng bit được thiết lập theo cú pháp, các ngữ nghĩa, và ràng buộc được chỉ rõ trong đặc tả này bên ngoài phụ lục này.

Hình ảnh mã hóa thứ nhất trong dòng bit là hình ảnh RAP, tức là hình ảnh IDR, hình ảnh CRA, hoặc hình ảnh BLA.

Đối với mỗi hình ảnh hiện thời được giải mã, đặt các biến maxPicOrderCnt và minPicOrderCnt bằng giá trị tối đa và tối thiểu, tương ứng, của giá trị PicOrderCntVal của các hình ảnh sau:

- Hình ảnh hiện thời.
- hình ảnh đứng trước theo thứ tự giải mã có TemporalId bằng 0.
- Hình ảnh chuẩn ngắn hạn trong bộ hình ảnh chuẩn của hình ảnh hiện thời.
- Tất cả các hình ảnh n có PicOutputFlag bằng 1 và $t_r(n) < t_r(\text{currPic})$ và $t_{o,dpb}(n) \geq t_r(\text{currPic})$, trong đó currPic là hình ảnh hiện thời.

Tất cả các điều kiện dưới đây được đáp ứng đối với mỗi lần kiểm tra sự tương thích dòng bit:

1. Đối với mỗi đơn vị truy nhập n , với $n > 0$, gắn với thông báo SEI về chu kỳ nhớ đệm, với $\Delta t_{g,90}(n)$ được chỉ rõ bằng

$$\Delta t_{g,90}(n) = 90000 * (t_{r,n}(n) - t_{af}(n-1)) \quad (\text{C-18})$$

giá trị của InitCpbRemovalDelay[SchedSelIdx] được ràng buộc như sau.

- Nếu cbr_flag[SchedSelIdx] bằng 0,

$$\text{InitCpbRemovalDelay[SchedSelIdx]} \leq \text{Ceil}(\Delta t_{g,90}(n)) \quad (\text{C-19})$$

- Mặt khác (cbr_flag[SchedSelIdx] bằng 1),

$$\text{Floor}(\Delta t_{g,90}(n)) \leq \text{InitCpbRemovalDelay[SchedSelIdx]} \leq \text{Ceil}(\Delta t_{g,90}(n)) \quad (\text{C-20})$$

Lưu ý 4 – Số bit chính xác trong CPB ở thời điểm lấy ra mỗi hình ảnh có thể tùy thuộc vào thông báo SEI về chu kỳ nhớ đệm nào được chọn để khởi tạo HRD. Bộ mã hóa phải tính đến vấn đề này để đảm bảo rằng tất cả các ràng buộc cụ thể phải được tuân thủ bất kể thông báo SEI về chu kỳ nhớ đệm nào được chọn để khởi tạo HRD, vì HRD có thể được khởi tạo ở thông báo bất kỳ trong số các thông báo SEI về chu kỳ nhớ đệm.

2. Tràn CPB được chỉ rõ là tình trạng trong đó tổng số bit trong CPB lớn hơn dung lượng CPB. CPB chưa bao giờ tràn.
3. Tràn dưới CPB được chỉ rõ là tình trạng trong đó thời gian danh định lấy ra đơn vị giải mã m từ CPB $t_{r,n}(m)$ là nhỏ hơn thời gian cuối cùng đơn vị giải mã m đến CPB $t_{af}(m)$ đối với ít nhất một giá trị của m . Khi `low_delay_hrd_flag` bằng 0, CPB chưa bao giờ tràn dưới.
4. Khi `low_delay_hrd_flag` bằng 1, tràn dưới CPB có thể xảy ra ở đơn vị giải mã m . Trong trường hợp này, thời gian đến CPB cuối cùng của đơn vị truy nhập n chứa đơn vị giải mã m $t_{af}(n)$ lớn hơn thời gian danh định lấy ra đơn vị truy nhập n chứa đơn vị giải mã m $t_{r,n}(n)$ từ CPB.
5. Thời gian lấy ra danh định của các hình ảnh từ CPB (bắt đầu từ hình thứ hai theo thứ tự giải mã), sẽ đáp ứng các ràng buộc về $t_{r,n}(n)$ và $t_r(n)$ được thể hiện trong điểm A.4.1 đến A.4.2.
6. Đối với mỗi hình ảnh hiện thời được giải mã, sau khi gọi ra quy trình lấy ra các hình ảnh từ DPB như được chỉ rõ trong điểm C.3.1, số hình ảnh giải mã trong DPB, bao gồm tất cả các hình ảnh n được đánh dấu là "dùng làm chuẩn" hoặc có `PicOutputFlag` bằng 1 và $t_{o,dpb}(n) \geq t_r(currPic)$, trong đó `currPic` là hình ảnh hiện thời, nhỏ hơn hoặc bằng $Max(0, sps_max_dec_pic_buffering[TargetDecHighestTid] - 1)$.
7. Tất cả các hình ảnh chuẩn sẽ có mặt trong DPB khi cần dự báo. Mỗi hình ảnh có `OutputFlag` bằng 1 sẽ có mặt trong DPB ở thời gian xuất ra từ DPB trừ khi nó được lấy ra từ DPB trước thời gian xuất ra bằng một trong số các quy trình được chỉ rõ trong điểm C.3.

8. Đối với mỗi hình ảnh hiện thời được giải mã, giá trị của $\text{maxPicOrderCnt} - \text{minPicOrderCnt}$ sẽ nhỏ hơn $\text{MaxPicOrderCntLsb} / 2$.
9. Giá trị của $\Delta_{\text{to,dpb}}(n)$ như được tính toán bằng phương trình C-17, giá trị này là độ chênh lệch giữa thời gian xuất ra của một hình ảnh và thời gian xuất ra của hình ảnh đầu tiên đứng sau hình ảnh này theo thứ tự xuất ra và có PicOutputFlag bằng 1, sẽ đáp ứng ràng buộc được thể hiện trong điểm A.4.1 về profin, bậc và mức được chỉ rõ trong dòng bit bằng cách sử dụng quy trình giải mã được chỉ rõ trong các mệnh đề 2 – 9.

Như được chỉ rõ trong bản mô tả, các thông số $\text{sps_max_dec_pic_buffering}[i]$ có thể không được chọn thích hợp trong các hoạt động tương thích dòng bit. Ở điều 6 của mục C.3, phiên bản dự thảo làm việc HEVC 8 chỉ báo rằng “số hình ảnh giải mã trong DPB ... sẽ nhỏ hơn hoặc bằng $\text{Min}(0, \text{sps_max_dec_pic_buffering}[\text{TemporalId}] - 1)$,” trong đó TemporalId không được xác định. Việc chỉ rõ $\text{TargetDecHighestTid}$ là chỉ số i của $\text{sps_max_dec_pic_buffering}[i]$ có thể, theo một hoặc nhiều kỹ thuật theo sáng chế, đảm bảo rằng i bằng giá trị được xác định rõ của $\text{TargetDecHighestTid}$ được sử dụng trong $\text{sps_max_dec_pic_buffering}[i]$ khi tiến hành các hoạt động tương thích dòng bit.

Khi thiết bị tiến hành quy trình giải mã như là một phần của hoạt động HRD, thiết bị có thể giải mã, từ SPS, mảng phần tử cú pháp ($\text{sps_max_dec_pic_buffering}[]$), trong đó mỗi phần tử cú pháp chỉ báo dung lượng tối đa cần thiết của DPB của HRD. Hơn thế nữa, như là một phần của việc tiến hành hoạt động HRD, thiết bị có thể xác định, dựa vào ký hiệu nhận dạng theo thời gian cao nhất của đích ($\text{TargetDecHighestTid}$), phần tử cú pháp cụ thể trong mảng. Hơn thế nữa, như được thể hiện trong các phần thay đổi trên đây ở mục C.3 của phiên bản dự thảo làm việc HEVC 8, thiết bị có thể xác định, dựa ít nhất một phần vào liệu số hình ảnh giải mã trong DPB có ít hơn hoặc bằng giá trị tối đa của 0 và giá trị của phần tử cú pháp cụ thể trừ 1 hay không, liệu dòng bit có tuân theo tiêu chuẩn mã hóa dữ liệu video hay không.

Mục C.4 của phiên bản dự thảo làm việc HEVC 8 mô tả sự tương thích bộ giải mã. Theo một hoặc nhiều kỹ thuật làm ví dụ theo sáng chế, mục C.4 của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi như sau:

Bộ giải mã tương thích với đặc tả này sẽ đáp ứng tất cả các yêu cầu được chỉ rõ trong điểm này.

Bộ giải mã yêu cầu sự tương thích với profin, bậc và mức cụ thể có thể giải mã thành công tất cả các dòng bit mà tuân theo các yêu cầu tương thích dòng bit được chỉ rõ trong điểm C.4, theo cách được chỉ rõ trong phụ lục A, miễn là tất cả các *bộ thông số video*, bộ thông số chuỗi, và bộ thông số hình ảnh được đề cập trong đơn vị NAL có VCL, và thông báo SEI định thời hình ảnh và thông báo SEI về chu kỳ nhớ đệm thích hợp được truyền đến bộ giải mã, kịp thời, trong dòng bit (bằng đơn vị NAL không có VCL), hoặc bằng phương tiện ngoài không được chỉ rõ trong đặc tả này.

Khi dòng bit chứa các phần tử cú pháp có các giá trị được chỉ rõ là được dự trữ và được chỉ rõ là các bộ giải mã sẽ bỏ qua các giá trị của các phần tử cú pháp hoặc đơn vị NAL chứa các phần tử cú pháp có giá trị dự trữ, và mặt khác dòng bit tương thích với đặc tả này, thì bộ giải mã tương thích sẽ giải mã dòng bit theo cách giống như bộ giải mã này giải mã dòng bit tương thích và bỏ qua các giá trị của các phần tử cú pháp hoặc đơn vị NAL chứa các phần tử cú pháp có giá trị dự trữ như được chỉ rõ.

Có hai loại tương thích mà có thể được yêu cầu bởi bộ giải mã: sự tương thích theo thứ tự xuất ra và sự tương thích định thời xuất ra.

Để kiểm tra sự tương thích của bộ giải mã, dòng bit kiểm tra tương thích với profin, bậc và mức yêu cầu, như được chỉ rõ bởi điểm C.4 đều được phân phối bởi bộ lập lịch biểu dòng giả định (hypothetical stream Scheduler-HSS) đến cả HRD và đến bộ giải mã kiểm tra (decoder under test-DUT). Tất cả các hình ảnh xuất ra bởi HRD cũng được xuất ra bởi DUT và, đối với mỗi hình ảnh xuất ra bởi HRD, các giá trị của tất cả các mẫu mà được xuất ra bởi DUT của hình ảnh tương ứng sẽ bằng các giá trị của các mẫu được xuất ra bởi HRD.

Đối với sự tương thích bộ giải mã định thời xuất ra, HSS hoạt động như được mô tả trên đây, với lịch biểu phân phối chỉ được chọn từ tập con các giá trị của SchedSelIdx mà tốc độ bit và dung lượng CPB được giới hạn như được chỉ rõ trong phụ lục A đối với profin, bậc và mức cụ thể, hoặc với lịch biểu phân phối "nội suy" như được chỉ rõ dưới đây mà tốc độ bit và dung lượng CPB được giới hạn như được chỉ rõ trong phụ lục A. Lịch biểu phân phối tương tự được sử dụng cho cả HRD và DUT.

Khi bộ thông số HRD và thông báo SEI về chu kỳ nhớ đệm có mặt với $\text{cpb_cnt_minus1}[\text{TargetDecHighestTid}]$ lớn hơn 0, thì bộ giải mã có khả năng giải mã dòng bit khi được phân phối từ HSS đang hoạt động bằng cách sử dụng lịch biểu phân phối "nội suy" được chỉ rõ là có tốc độ bit đỉnh r , dung lượng CPB $c(r)$, và độ trễ lấy ra từ CPB ban đầu $(f(r) \div \square r)$ như sau:

$$\alpha = (r - \text{BitRate}[\text{SchedSelIdx} - 1]) \div (\text{BitRate}[\text{SchedSelIdx}] - \text{BitRate}[\text{SchedSelIdx} - 1]), \quad (\text{C-22})$$

$$c(r) = \alpha * \text{CpbSize}[\text{SchedSelIdx}] + (1 - \alpha * \text{CpbSize}[\text{SchedSelIdx} - 1]), \quad (\text{C-23})$$

$$f(r) = \alpha * \text{InitCpbRemovalDelay}[\text{SchedSelIdx}] * \text{BitRate}[\text{SchedSelIdx}] + (1 - \alpha) * \text{InitCpbRemovalDelay}[\text{SchedSelIdx} - 1] * \text{BitRate}[\text{SchedSelIdx} - 1] \quad (\text{C-24})$$

đối với $\text{SchedSelIdx} > 0$ và r sao cho $\text{BitRate}[\text{SchedSelIdx} - 1] \leq r \leq \text{BitRate}[\text{SchedSelIdx}]$ sao cho r và $c(r)$ nằm trong giới hạn như được chỉ rõ trong phụ lục A về tốc độ bit và dung lượng bộ nhớ đệm tối đa của profin, bậc và mức cụ thể.

Lưu ý 1 – $\text{InitCpbRemovalDelay}[\text{SchedSelIdx}]$ có thể khác nhau từ chu kỳ nhớ đệm này đến chu kỳ nhớ đệm khác và phải được tính toán lại.

Đối với sự tương thích bộ giải mã định thời xuất ra, HRD như được mô tả trên đây được sử dụng và thời gian (so với thời gian phân phối bit thứ nhất) xuất ra hình ảnh là như nhau đối với cả HRD và DUT lên đến độ trễ cố định.

Đối với sự tương thích bộ giải mã theo thứ tự xuất ra, áp dụng như sau.

- HSS phân phối dòng bit *BitstreamToDecode* đến DUT "theo nhu cầu" của DUT, có nghĩa là HSS phân phối bit (theo thứ tự giải mã) chỉ khi DUT cần nhiều bit hơn để tiếp tục xử lý.

Lưu ý 2 – Điều này có nghĩa là đối với thử nghiệm này, bộ nhớ đệm hình ảnh mã hóa của DUT có thể có dung lượng nhỏ bằng dung lượng của đơn vị giải mã lớn nhất.

- HRD đã cải biến như được mô tả dưới đây được sử dụng, và HSS phân phối dòng bit đến HRD bằng một trong số các lịch biểu được chỉ rõ trong dòng bit *BitstreamToDecode* sao cho tốc độ bit và dung lượng CPB được giới hạn như được chỉ rõ trong phức lục A. Thứ tự các hình ảnh xuất ra là giống nhau ở cả HRD và DUT.
- Đối với sự tương thích bộ giải mã theo thứ tự xuất ra, *dung lượng CPB* là $CpbSize[SchedSelIdx]$ như được xác định bằng phương trình E-46, trong đó các thông số HRD $SchedSelIdx$ và được chọn như được chỉ rõ trên đây trong điểm C.1. *Dung lượng DPB* là $sps_max_dec_pic_buffering[TargetDecHighestTid]$. Thời gian lấy ra từ CPB đối với HRD là bằng thời gian đến bit cuối và việc giải mã là tức thì. Hoạt động của DPB của HRD này là như được mô tả trong điểm C.5.1 đến C.5.3.

Như được chỉ rõ trong bản mô tả, các thông số $cpb_cnt_minus1[i]$ và $sps_max_dec_pic_buffering[i]$ có thể không được chọn thích hợp trong các yêu cầu tương thích bộ giải mã. Ví dụ, mục C.4 của phiên bản dự thảo làm việc HEVC 8 không chỉ rõ chỉ số của cpb_cnt_minus1 . Việc chỉ rõ $TargetDecHighestTid$ là chỉ số i của $cpb_cnt_minus1[i]$ và $sps_max_dec_pic_buffering[i]$ có thể, theo một hoặc nhiều kỹ thuật theo sáng chế, đảm bảo rằng các hoạt động tương thích bộ giải mã được tiến hành thích hợp với i bằng giá trị được xác định rõ của $TargetDecHighestTid$.

Hơn thế nữa, mục C.4.2 của phiên bản dự thảo làm việc HEVC 8 mô tả việc lấy ra các hình ảnh từ DPB để kiểm tra sự tương thích bộ giải mã. Theo một hoặc nhiều kỹ thuật làm ví dụ theo sáng chế, tên mục C.4.2 có thể được thay đổi từ “lấy ra hình ảnh từ DPB” thành “xuất và lấy ra hình ảnh từ DPB.” Nội dung của mục C.4.2 của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi như sau:

Việc *xuất và lấy ra* các hình ảnh từ DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân tích cú pháp phần đầu lát của lát thứ nhất của hình ảnh hiện thời) xảy ra tức thời khi đơn vị giải mã đầu tiên của đơn vị truy nhập chứa hình ảnh hiện thời được lấy ra từ CPB và được tiến hành như sau.

Quy trình giải mã bộ hình ảnh chuẩn như được chỉ rõ trong điểm 8.3.2 được gọi ra.

- Nếu hình ảnh hiện thời là hình ảnh IDR hoặc BLA, thì áp dụng như sau.

1. Khi hình ảnh IDR hoặc BLA không phải là hình ảnh đầu tiên được giải mã và giá trị của `pic_width_in_luma_samples` hoặc `pic_height_in_luma_samples` hoặc `sps_max_dec_pic_buffering[ TargetDecHighestTid ]` được tìm ra từ bộ thông số chuỗi hoạt động khác với giá trị của `pic_width_in_luma_samples` hoặc `pic_height_in_luma_samples` hoặc `sps_max_dec_pic_buffering[ TargetDecHighestTid ]` được tìm ra từ bộ thông số chuỗi mà có hiệu lực đối với hình ảnh trước đó, tương ứng, `no_output_of_prior_pics_flag` được suy ra là bằng 1 bởi HRD, bất kể giá trị thực của `no_output_of_prior_pics_flag`.
 Lưu ý – Bộ giải mã theo các phương án thực hiện này cố gắng xử lý các thay đổi về hình ảnh hoặc dung lượng DPB tốt hơn so với HRD có liên quan đến các thay đổi về giá trị `pic_width_in_luma_samples`, `pic_height_in_luma_samples` hoặc `sps_max_dec_pic_buffering[ TargetDecHighestTid ]`.
 2. Khi `no_output_of_prior_pics_flag` bằng 1 hoặc được suy ra là bằng 1, tất cả các bộ nhớ đệm lưu trữ hình ảnh trong DPB được làm rỗng mà không xuất ra các hình ảnh chứa trong các bộ nhớ đệm này.
 3. *Khi `no_output_of_prior_pics_flag` không bằng 1 và không được suy ra là bằng 1, các bộ nhớ đệm lưu trữ hình ảnh chứa hình ảnh mà được đánh dấu là "không cần xuất ra" và "không dùng làm chuẩn" được làm rỗng (mà không xuất ra), và tất cả các bộ nhớ đệm lưu trữ hình ảnh không được làm rỗng trong DPB được làm rỗng bằng cách gọi ra nhiều lần quy trình "xuất ra" được chỉ rõ trong điểm C.5.2.1.*
- mặt khác (hình ảnh hiện thời không phải là hình ảnh IDR hoặc BLA), các bộ nhớ đệm lưu trữ hình ảnh chứa hình ảnh mà được đánh dấu là "không cần xuất ra" và "không dùng làm chuẩn" được làm rỗng (mà không xuất ra). Khi một hoặc nhiều điều kiện dưới đây là đúng, quy trình "xuất ra" được chỉ rõ trong điểm C.5.2.1 được gọi ra nhiều lần cho đến khi thu được bộ nhớ đệm lưu trữ hình ảnh rỗng để lưu trữ hình ảnh giải mã hiện thời.

1. Số hình ảnh trong DPB được đánh dấu là "cần xuất ra" là lớn hơn $\text{sps_max_num_reorder_pics}[\text{TargetDecHighestTid}]$.
2. Số hình ảnh trong DPB là bằng $\text{sps_max_dec_pic_buffering}[\text{TargetDecHighestTid}]$.

Quy trình “xuất ra”

Quy trình “xuất ra” được gọi ra trong các trường hợp sau.

- Hình ảnh hiện thời là hình ảnh IDR hoặc hình ảnh BLA và $\text{no_output_of_prior_pics_flag}$ không phải bằng 1 và không được suy ra là bằng 1, như được chỉ rõ trong điểm C.5.2.
- *Hình ảnh hiện thời không phải là hình ảnh IDR hoặc hình ảnh BLA, và số lượng hình ảnh trong DPB được đánh dấu là "cần xuất ra" lớn hơn so với $\text{sps_max_num_reorder_pics}[\text{TargetDecHighestTid}]$, như được chỉ rõ trong điểm C.5.2.*
- *Hình ảnh hiện thời không phải là hình ảnh IDR hoặc hình ảnh BLA và số lượng hình ảnh trong DPB bằng $\text{sps_max_dec_pic_buffering}[\text{TargetDecHighestTid}]$, như được chỉ rõ trong điểm C.5.2.*

Quy trình “xuất ra” bao gồm các bước theo thứ tự như sau:

1. Hình ảnh là hình ảnh đầu tiên để xuất ra có thể được chọn làm hình ảnh có giá trị PicOrderCntVal nhỏ nhất trong số tất cả các hình ảnh trong bộ nhớ DPB được đánh dấu là “cần xuất ra”.
2. Hình ảnh được cắt xén, sử dụng hình chữ nhật cắt xén được xác định trong bộ thông số chuỗi có hiệu lực đối với hình ảnh này, hình ảnh cắt xén được xuất ra, và hình ảnh này được đánh dấu là “không cần xuất ra”.
3. Nếu bộ nhớ đệm lưu trữ hình ảnh chứa hình ảnh đã được cắt xén và được xuất ra lưu trữ hình ảnh được đánh dấu là “không dùng làm chuẩn”, thì bộ nhớ đệm lưu trữ hình ảnh đó được làm rỗng.

Như được nêu trong phần mô tả này, các thông số $\text{sps_max_dec_pic_buffering}[i]$ và $\text{sps_max_num_reorder_pics}[i]$ có thể không được chọn hoàn toàn trong các hoạt động HRD, như lấy ra các hình ảnh từ DPB. Việc xác định $\text{TargetDecHighestTid}$ là chỉ số i của $\text{sps_max_dec_pic_buffering}[i]$ và $\text{sps_max_num_reorder_pics}[i]$ có thể, theo một

hoặc nhiều kỹ thuật theo sáng chế, đảm bảo rằng i bằng giá trị được xác định rõ của `TargetDecHighestTid` được sử dụng trong `sps_max_dec_pic_buffering[i]` và `sps_max_num_reorder_pics[i]` khi tiến hành hoạt động HRD để lấy ra các hình ảnh từ DPB.

Khi thiết bị tiến hành quy trình giải mã trong hoạt động HRD, thiết bị này có thể giải mã, từ SPS, mảng phần tử cú pháp (`sps_max_dec_pic_buffering[]`) trong đó mỗi phần tử cú pháp chỉ báo kích thước tối đa cần thiết của DPB của HRD. Hơn thế nữa, khi thiết bị thực hiện hoạt động HRD, thiết bị này có thể xác định, dựa vào ký hiệu nhận dạng theo thời gian cao nhất của đích, phần tử cú pháp cụ thể trong mảng (`sps_max_dec_pic_buffering[TargetDecHighestTid]`). Hơn thế nữa, thiết bị này có thể tiến hành quy trình xuất ra để làm rỗng một hoặc nhiều bộ nhớ đệm lưu trữ hình ảnh của DPB khi hình ảnh hiện thời không phải là hình ảnh IDR hoặc hình ảnh BLA và số lượng hình ảnh trong DPB được đánh dấu là cần xuất ra lớn hơn giá trị của phần tử cú pháp cụ thể.

Tương tự, khi thiết bị thực hiện quy trình giải mã trong hoạt động HRD, thiết bị này có thể giải mã, từ SPS, mảng phần tử cú pháp (`sps_max_dec_pic_buffering[]`) trong đó mỗi phần tử cú pháp chỉ báo kích thước được yêu cầu tối đa của DPB của HRD. Hơn thế nữa, khi thiết bị tiến hành hoạt động HRD, thiết bị này có thể xác định, dựa vào ký hiệu nhận dạng theo thời gian cao nhất của đích, phần tử cú pháp cụ thể trong mảng (`sps_max_dec_pic_buffering[TargetDecHighestTid]`). Hơn thế nữa, thiết bị này có thể tiến hành quy trình xuất ra để làm rỗng một hoặc nhiều bộ nhớ đệm lưu trữ hình ảnh của DPB khi hình ảnh hiện thời không phải là hình ảnh IDR hoặc hình ảnh BLA và số lượng hình ảnh trong DPB là bằng nhau được chỉ báo bởi phần tử cú pháp cụ thể.

Hơn thế nữa, theo một hoặc nhiều kỹ thuật theo sáng chế, cấu trúc cú pháp `applicable_operation_points()` và các ngữ nghĩa đi kèm có thể được bổ sung vào phiên bản dự thảo làm việc HEVC 8. Bảng 5, dưới đây, thể hiện cú pháp làm ví dụ của cấu trúc cú pháp, `applicable_operation_points()`.

Bảng 5 –Điểm hoạt động có thể áp dụng

<code>applicable_operation_points() {</code>	
<code> num_applicable_ops_minus1</code>	<code>ue(v)</code>
<code> if(num_applicable_ops_minus1 > 0)</code>	
<code> default_op_applicable_flag</code>	<code>u(1)</code>
<code> mumOpsSignalled = default_op_applicable_flag ?</code> <code> num_applicable_ops_minus1 :</code> <code> num_applicable_ops_minus1 + 1</code>	
<code> for(i = 0; i < mumOpsSignalled; i++) {</code>	
<code> operation_point_layer_ids(i)</code>	
<code> op_temporal_id[i]</code>	<code>u(3)</code>
<code> }</code>	
<code> }</code>	

Cấu trúc cú pháp `applicable_operation_point( )` được thể hiện ở bảng 5 chỉ rõ điểm hoạt động mà áp dụng thông báo SEI liên quan đến cấu trúc cú pháp này. Thông báo SEI gắn với cấu trúc cú pháp `applicable_operation_point( )` (còn được gọi là thông báo SEI đi kèm) là thông báo SEI chứa cấu trúc cú pháp `applicable_operation_point( )`. Thông báo SEI gắn với cấu trúc cú pháp `applicable_operation_point( )` có thể là thông báo SEI về chu kỳ nhớ đệm, thông báo SEI định thời hình ảnh hoặc thông báo SEI định thời hình ảnh con.

Điểm hoạt động ngầm định có thể được định nghĩa là điểm hoạt động được xác định dựa vào `OpLayerIdSet` chứa các giá trị nằm trong khoảng từ 0 đến `num_reserved_zero_6bits`, bao gồm cả các giá trị đầu mút, trong đó `num_reserved_zero_6bits` được mã hóa trong phần đầu đơn vị NAL của đơn vị NAL SEI chứa thông báo SEI đi kèm, và `OpTid` là bằng với giá trị `TemporalId` của đơn vị NAL SEI chứa thông báo SEI đi kèm. Theo cách khác, điểm hoạt động ngầm định có thể được định nghĩa là điểm hoạt động được xác định dựa vào `OpLayerIdSet` chỉ chứa `num_reserved_zero_6bits` trong phần đầu đơn vị NAL của đơn vị NAL SEI chứa thông báo SEI đi kèm, và `OpTid` là bằng với giá trị `TemporalId` của đơn vị NAL SEI chứa thông báo SEI đi kèm. Theo cách khác, điểm hoạt động ngầm định có thể được định nghĩa là điểm hoạt động được xác định dựa vào `OpLayerIdSet` chứa duy nhất giá trị 0, và `OpTid` là bằng với giá trị `TemporalId` của đơn vị NAL SEI chứa thông báo SEI đi kèm.

Nếu `default_op_applicable_flag` bằng 1, thì các điểm hoạt động mà áp dụng thông báo SEI đi kèm là điểm hoạt động ngầm định và các điểm hoạt động

`num_applicable_ops_minus1` được xác định dựa vào `OpLayerIdSet` như được xác định bằng `operation_point_layer_ids(i)` và `OpTid` bằng với `op_temporal_id[i]`, với i nằm trong khoảng từ 0 đến `num_applicable_ops_minus1`, bao gồm cả các giá trị đầu mút. Mặt khác (`default_op_applicable_flag` bằng 0), thì các điểm hoạt động mà áp dụng thông báo SEI đi kèm có thể là các điểm hoạt động `num_applicable_ops_minus1 + 1` được xác định dựa vào `OpLayerIdSet` như được xác định bởi `operation_point_layer_ids(i)` và `OpTid` bằng `op_temporal_id[i]`, với i nằm trong khoảng bao hàm từ 0 đến `num_applicable_ops_minus1 + 1`.

Hơn thế nữa, trong cú pháp làm ví dụ nêu trong bảng 5, phần tử cú pháp `num_applicable_ops_minus1`, cộng 1, chỉ rõ số điểm hoạt động mà áp dụng thông báo SEI đi kèm. Giá trị của `num_applicable_ops_minus1` có thể nằm trong khoảng từ 0 đến 63, bao gồm cả giá trị đầu mút. Trong ví dụ nêu trong bảng 5, phần tử cú pháp `default_op_applicable_flag` bằng 1 chỉ rõ rằng thông báo SEI đi kèm áp dụng cho điểm hoạt động ngầm định. Phần tử cú pháp `default_op_applicable_flag` bằng 0 chỉ rõ rằng thông báo SEI đi kèm không áp dụng cho điểm hoạt động ngầm định. Phần tử cú pháp `op_temporal_id[i]` chỉ rõ giá trị `OpTid` thứ i được báo hiệu chính xác trong cấu trúc cú pháp `applicable_operation_point()`. Giá trị của `op_temporal_id[i]` có thể nằm trong khoảng từ 0 đến 6, bao gồm cả giá trị đầu mút.

Như nêu trên, phiên bản dự thảo làm việc HEVC 8 không cung cấp cách để gắn thông báo SEI về chu kỳ nhớ đệm hoặc thông báo SEI định thời hình ảnh với cấu trúc cú pháp `hrd_parameters()` mà cấu trúc cú pháp `operation_point_layer_ids()` đi kèm bao gồm nhiều giá trị của `nuh_reserved_zero_6bits` (tức là, ID nhiều lớp trong phiên bản mở rộng mã hoá dữ liệu video đa cảnh nhìn, 3DV hoặc có khả năng chuyển đổi cấp độ của HEVC). Việc bao gồm cấu trúc cú pháp `applicable_operation_point()` có thể giải quyết ít nhất một phần vấn đề này. Cấu trúc cú pháp `applicable_operation_point()` có thể xác định rõ ràng các điểm hoạt động mà áp dụng thông báo SEI về chu kỳ nhớ đệm, thông báo SEI định thời hình ảnh hoặc thông báo SEI định thời hình ảnh con. Nhờ đó có thể cho phép sử dụng thông tin được mang trong các phần tử cú pháp `nuh_reserved_zero_6bits` và `temporal_id_plus1` ở phần đầu đơn vị NAL của các đơn vị NAL SEI, và có thể cho phép chia sẻ thông tin được truyền trong cùng một thông báo SEI về chu kỳ nhớ đệm, định thời hình ảnh hoặc định thời hình ảnh con để xử lý dữ liệu video gắn với nhiều điểm hoạt động.

Mục D.1.1 của Phiên bản dự thảo làm việc HEVC 8 mô tả cú pháp của thông báo SEI về chu kỳ nhớ đệm. Theo một hoặc nhiều kỹ thuật làm ví dụ nêu trong sáng chế, cú pháp của thông báo SEI về chu kỳ nhớ đệm có thể được thay đổi như được nêu ở bảng 6 dưới đây. Các thay đổi đối với cú pháp của thông báo SEI về chu kỳ nhớ đệm có thể cho phép các thông báo SEI về chu kỳ nhớ đệm bao gồm cấu trúc cú pháp `applicable_operation_points()`.

Bảng 6 – Chu kỳ nhớ đệm

Buffering_period(payloadSize) {	Bộ mô tả
seq_parameter_set_id	ue(v)
<i>applicable_operation_points()</i>	
if(!sub_pic_cpb_params_present_flag)	
rap_cpb_params_present_flag	u(1)
if(NalHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <CpbCnt; SchedSelIdx++) {	
initial_cpb_removal_delay[SchedSelIdx]	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	u(v)
if(sub_pic_cpb_params_present_flag rap_cpb_params_present_flag) {	
initial_alt_cpb_removal_delay[SchedSelIdx]	u(v)
initial_alt_cpb_removal_delay_offset[SchedSelIdx]	u(v)
}	
}	
}	
if(VclHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <CpbCnt; SchedSelIdx++) {	
initial_cpb_removal_delay[SchedSelIdx]	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	u(v)
if(sub_pic_cpb_params_present_flag rap_cpb_params_present_flag) {	
initial_alt_cpb_removal_delay[SchedSelIdx]	u(v)
initial_alt_cpb_removal_delay_offset[SchedSelIdx]	u(v)
}	
}	

}	
}	

Mục D.2.1 của Phiên bản dự thảo làm việc HEVC 8 mô tả các ngữ nghĩa của các phần tử cú pháp của thông báo SEI về chu kỳ nhớ đệm. Theo một hoặc nhiều kỹ thuật theo sáng chế, các ngữ nghĩa của cấu trúc cú pháp buffering_period(payloadSize) có thể được thay đổi như sau. Các ngữ nghĩa của các phần tử cú pháp này không được đề cập là giống như trong Phiên bản dự thảo làm việc HEVC 8.

Thông báo SEI về chu kỳ nhớ đệm cung cấp thông tin về độ trễ lấy ra từ CPB ban đầu và độ lệch độ trễ lấy ra từ CPB ban đầu.

Áp dụng như sau cho các ngữ nghĩa và cú pháp của thông báo SEI về chu kỳ nhớ đệm:

- Các phần tử cú pháp *initial_cpb_removal_delay_length_minus1* và *sub_pic_cpb_params_present_flag*, và các biến *NalHrdBpPresentFlag*, *VclHrdBpPresentFlag*, *CpbSize[SchedSelIdx]*, *BitRate[SchedSelIdx]*, và *CpbCnt* được thấy ở hoặc tìm được từ các phần tử cú pháp được thấy trong cấu trúc cú pháp *hrd_parameters()* và cấu trúc cú pháp *sub_layer_hrd_parameters()* có thể áp dụng cho điểm hoạt động bất kỳ trong số các điểm hoạt động mà áp dụng thông báo SEI về chu kỳ nhớ đệm.
- Hai điểm hoạt động bất kỳ mà áp dụng thông báo SEI về chu kỳ nhớ đệm để có các giá trị *OpTid* khác nhau *tIdA* và *tIdB* chỉ báo rằng các giá trị *cpb_cnt_minus1[tIdA]* và *cpb_cnt_minus1[tIdB]* được mã hóa trong (các) cấu trúc cú pháp *hrd_parameters()* áp dụng được cho hai điểm hoạt động là giống nhau.
- Hai điểm hoạt động bất kỳ mà áp dụng thông báo SEI về chu kỳ nhớ đệm để có các giá trị *OpLayerIdSet* khác nhau *layerIdSetA* và *layerIdSetB* chỉ báo rằng các giá trị của *nal_hrd_parameters_present_flag* và *vcl_hrd_parameters_present_flag*, tương ứng, cho hai cấu trúc cú pháp *hrd_parameters()* có thể áp dụng cho hai điểm hoạt động là giống nhau.

- Dòng bit (hoặc một phần của nó) chỉ tập con dòng bit (hoặc một phần của nó) gắn với điểm hoạt động bất kỳ trong số các điểm hoạt động mà áp dụng thông báo SEI về chu kỳ nhớ đệm.

Nếu *NalHrdBpPresentFlag* hoặc *VclHrdBpPresentFlag* bằng 1, thông báo SEI về chu kỳ nhớ đệm có thể áp dụng cho điểm hoạt động cụ thể có thể có mặt trong đơn vị truy nhập bất kỳ trong chuỗi dữ liệu video được mã hóa, và thông báo SEI về chu kỳ nhớ đệm có thể áp dụng cho điểm hoạt động cụ thể sẽ có mặt trong mỗi đơn vị truy nhập RAP, và trong mỗi đơn vị truy nhập gắn với thông báo SEI về điểm khôi phục. Mặt khác (cả *NalHrdBpPresentFlag* và *VclHrdBpPresentFlag* đều bằng 0), không có đơn vị truy nhập nào trong chuỗi dữ liệu video mã hóa có thông báo SEI về chu kỳ nhớ đệm có thể áp dụng cho điểm hoạt động cụ thể.

Lưu ý – Đối với một số ứng dụng, có thể mong muốn sự có mặt thường xuyên của thông báo SEI về chu kỳ nhớ đệm.

Khi xuất hiện đơn vị NAL SEI chứa thông báo SEI về chu kỳ nhớ đệm và có *nuh_reserved_zero_6bits* bằng 0, thì đơn vị NAL SEI sẽ đứng trước đơn vị NAL có VCL thứ nhất trong đơn vị truy nhập tính theo thứ tự giải mã.

Chu kỳ nhớ đệm được xác định là tập hợp đơn vị truy nhập nằm giữa hai thông báo SEI về chu kỳ nhớ đệm *liên tiếp* theo thứ tự giải mã.

Biến *CpbCnt* tìm được phải bằng *cpb_cnt_minus1[tId] + 1*, trong đó *cpb_cnt_minus1[tId]* được mã hóa trong cấu trúc cú pháp *hrd_parameters()* mà có thể áp dụng cho điểm hoạt động bất kỳ trong số các điểm hoạt động mà áp dụng thông báo SEI về chu kỳ nhớ đệm và có *OpTid* bằng *tId*.

seq_parameter_set_id chỉ bộ thông số chuỗi hoạt động. Giá trị của *seq_parameter_set_id* sẽ bằng giá trị của *seq_parameter_set_id* trong bộ thông số hình ảnh dựa vào hình ảnh được mã hóa gắn với thông báo SEI về chu kỳ nhớ đệm. Giá trị của *seq_parameter_set_id* sẽ nằm trong khoảng từ 0 đến 31, bao gồm cả giá trị đầu mút.

rap_cpb_params_present_flag bằng 1 chỉ rõ sự xuất hiện của các phần tử cú pháp *initial_alt_cpb_removal_delay[SchedSelIdx]* và *initial_alt_cpb_removal_delay_offset[SchedSelIdx]*. Khi không xuất hiện, giá trị của *alt_cpb_params_present_flag* được suy ra là bằng 0. Khi hình ảnh gắn kèm

không phải là hình ảnh CRA hay hình ảnh BLA, thì giá trị của `alt_cpb_params_present_flag` sẽ bằng 0.

`initial_cpb_removal_delay[ SchedSelIdx ]` và

`initial_alt_cpb_removal_delay[ SchedSelIdx ]` lần lượt xác định độ trễ lấy ra từ CPB ban đầu *ngầm định* và độ trễ lấy ra từ CPB ban đầu *khác*, của `SchedSelIdx`-th CPB. Các phần tử cú pháp này có độ dài tính theo bit dựa vào công thức `initial_cpb_removal_delay_length_minus1 + 1`, và có thể được tính theo đơn vị nhịp đồng hồ bằng 90 kHz. Giá trị của các phần tử cú pháp này sẽ không bằng 0 và nhỏ hơn hoặc bằng $90000 * (\text{CpbSize}[\text{SchedSelIdx}] \div \text{BitRate}[\text{SchedSelIdx}])$, giá trị tương đương với thời gian của dung lượng CPB tính theo đơn vị nhịp đồng hồ bằng 90 kHz.

`initial_cpb_removal_delay_offset[ SchedSelIdx ]` và

`initial_alt_cpb_removal_delay_offset[ SchedSelIdx ]` lần lượt xác định độ lệch độ trễ lấy ra từ CPB ban đầu *ngầm định* và độ lệch độ trễ lấy ra từ CPB ban đầu *khác*, cho `SchedSelIdx`-th CPB. Các phần tử cú pháp này có độ dài tính theo bit dựa vào công thức `initial_cpb_removal_delay_length_minus1 + 1` và được tính theo đơn vị nhịp đồng hồ bằng 90 kHz. Các phần tử cú pháp này không được dùng bởi các bộ giải mã và có thể chỉ cần dùng cho bộ lập lịch biểu phân phối (HSS) được xác định trong phụ lục C.

Thông báo SEI về chu kỳ nhớ đệm có thể bao gồm các thông số HRD (ví dụ, `initial_cpb_removal_delay[ SchedSelIdx ]`, `initial_cpb_removal_delay_offset[ SchedSelIdx ]`, `initial_alt_cpb_removal_delay[ SchedSelIdx ]` và `initial_alt_cpb_removal_delay_offset[ SchedSelIdx ]`). Như nêu trên, phiên bản dự thảo làm việc HEVC 8 không cung cấp cách để gắn thông báo SEI về chu kỳ nhớ đệm với cấu trúc cú pháp `hrd_parameters( )` trong VPS mà cấu trúc cú pháp `operation_point_layer_ids( )` gắn kèm bao gồm nhiều giá trị của `nuh_reserved_zero_6bits` (tức là ID nhiều lớp trong phiên bản mở rộng mã hóa dữ liệu video đa cảnh nhìn, 3DV hoặc có khả năng chuyển đổi cấp độ của HEVC). Do đó, theo một hoặc nhiều kỹ thuật theo sáng chế, phần tử cú pháp `applicable_operation_points( )` trong thông báo SEI về chu kỳ nhớ đệm chỉ rõ điểm hoạt động mà áp dụng thông báo SEI về chu kỳ nhớ đệm.

Mục D.1.2 của phiên bản dự thảo làm việc HEVC 8 chỉ ra cú pháp của thông báo SEI định thời hình ảnh. Theo một hoặc nhiều kỹ thuật theo sáng chế, cú pháp của thông báo SEI định thời hình ảnh có thể được thay đổi như được nêu trong bảng 7 dưới đây. Các thay đổi đối với cú pháp của thông báo SEI định thời hình ảnh có thể cho phép các thông báo SEI định thời hình ảnh bao gồm cấu trúc cú pháp `applicable_operation_points( )`.

Bảng 7 – Thông báo SEI định thời hình ảnh

<code>pic_timing(payloadSize) {</code>	Bộ mô tả
<code> <i>applicable_operation_points()</i></code>	
<code> <i>au_cpb_removal_delay_minus1</i></code>	$u(v)$
<code> <i>pic_dpb_output_delay</i></code>	$u(v)$
<code> if(sub_pic_cpb_params_present_flag) {</code>	
<code> <i>num_decoding_units_minus1</i></code>	$ue(v)$
<code> <i>du_common_cpb_removal_delay_flag</i></code>	$u(1)$
<code> if(<i>du_common_cpb_removal_delay_flag</i>)</code>	
<code> <i>du_common_cpb_removal_delay_minus1</i></code>	$u(v)$
<code> for(i = 0; i <= num_decoding_units_minus1; i++) {</code>	
<code> <i>num_nalus_in_du_minus1[i]</i></code>	$ue(v)$
<code> if(!<i>du_common_cpb_removal_delay_flag</i>)</code>	
<code> <i>du_cpb_removal_delay_minus1[i]</i></code>	$u(v)$
<code> }</code>	
<code> }</code>	
<code>}</code>	

Ngoài ra, các ngữ nghĩa của thông báo SEI định thời hình ảnh có thể được thay đổi như sau. Các ngữ nghĩa của các phần tử cú pháp của cấu trúc cú pháp `pic_timing(payloadSize)` không được đề cập dưới đây có thể là giống như các ngữ nghĩa trong phiên bản dự thảo làm việc HEVC 8.

Thông báo SEI định thời hình ảnh cung cấp thông tin về độ trễ lấy ra từ CPB và độ trễ xuất ra từ DBP cho đơn vị truy nhập gắn với thông báo SEI.

Áp dụng như sau cho cú pháp và các ngữ nghĩa của thông báo SEI định thời hình ảnh:

- Các phần tử cú pháp `sub_pic_cpb_params_present_flag`, `cpb_removal_delay_length_minus1`, `dpb_output_delay_length_minus1`, và `du_cpb_removal_delay_length_minus1`, và biến `CpbDpbDelaysPresentFlag` được thấy ở hoặc tìm được từ phần tử cú pháp được thấy trong cấu trúc cú pháp `hrd_parameters()` và cấu trúc cú pháp `sub_layer_hrd_parameters()` có thể áp dụng cho điểm hoạt động bất kỳ trong số các điểm hoạt động mà áp dụng thông báo SEI định thời hình ảnh.
- Dòng bit (hoặc một phần của nó) chỉ tập con trong dòng bit (hoặc một phần của nó) liên quan đến điểm hoạt động bất kỳ trong số các điểm hoạt động mà áp dụng thông báo SEI định thời hình ảnh.

Lưu ý 1 – Cú pháp của thông báo SEI định thời hình ảnh phụ thuộc vào nội dung của cấu trúc cú pháp `hrd_parameters()` có thể áp dụng cho điểm hoạt động mà có hiệu lực đối với thông báo SEI định thời hình ảnh. Các cấu trúc cú pháp `hrd_parameters()` này nằm trong bộ thông số video và/hoặc bộ thông số chuỗi có hiệu lực đối với hình ảnh mã hóa gắn với thông báo SEI định thời hình ảnh. Khi thông báo SEI định thời hình ảnh gắn với đơn vị truy nhập CRA là đơn vị truy nhập thứ nhất trong dòng bit, đơn vị truy nhập IDR, hoặc đơn vị truy nhập BLA, trừ khi nó đứng sau thông báo SEI về chu kỳ nhớ đệm nằm trong cùng một đơn vị truy nhập, việc kích hoạt bộ thông số video và bộ thông số chuỗi (và, đối với các hình ảnh IDR hoặc BLA không phải là hình ảnh đầu tiên trong dòng bit, xác định rằng hình ảnh mã hóa là hình ảnh IDR hay hình ảnh BLA) không xảy ra cho đến khi giải mã đơn vị NAL lát mã hóa thứ nhất của hình ảnh mã hóa. Vì đơn vị NAL lát mã hóa của hình ảnh mã hóa đứng sau thông báo SEI định thời hình ảnh theo thứ tự đơn vị NAL, nên có thể có các trường hợp đòi hỏi bộ giải mã phải lưu trữ phần tải tin chứa chuỗi byte thô (RBSP: Raw Byte Sequence Payload) chứa thông báo SEI định thời hình ảnh cho tới khi xác định được bộ thông số video hoạt động và/hoặc bộ thông số chuỗi hoạt động, và sau đó tiến hành phân tích cú pháp của thông báo SEI định thời hình ảnh.

Sự có mặt của thông báo SEI định thời hình ảnh trong dòng bit được xác định như sau.

- Nếu `CpbDpbDelaysPresentFlag` bằng 1, thì một thông báo SEI định thời hình ảnh có thể áp dụng cho điểm hoạt động cụ thể sẽ có mặt trong mọi đơn vị truy nhập của chuỗi dữ liệu video mã hoá.
- Ngược lại (`CpbDpbDelaysPresentFlag` bằng 0), thì không có thông báo SEI định thời hình ảnh nào có thể áp dụng cho điểm hoạt động cụ thể có mặt trong bất cứ đơn vị truy nhập nào của chuỗi dữ liệu video mã hoá.

Khi đơn vị NAL SEI chứa thông báo SEI định thời hình ảnh và có `nuh_reserved_zero_6bits` bằng 0 có mặt, thì đơn vị NAL SEI này sẽ đứng trước, theo thứ tự giải mã, đơn vị NAL có VCL thứ nhất trong đơn vị truy nhập.

`au_cpb_removal_delay_minus1` cộng 1 chỉ rõ, khi HRD hoạt động ở mức đơn vị truy nhập, số lượng nhịp đồng hồ phải chờ sau khi lấy ra từ CPB đơn vị truy nhập gắn với thông báo SEI về chu kỳ nhớ đệm mới nhất trong đơn vị truy nhập trước đó trước khi lấy ra từ bộ nhớ đệm dữ liệu đơn vị truy nhập gắn với thông báo SEI định thời hình ảnh. Giá trị này cũng được HSS sử dụng để tính thời gian đến sớm nhất có thể của dữ liệu đơn vị truy nhập đến bộ nhớ CPB. Phần tử cú pháp này là mã độ dài cố định có độ dài tính theo bit được xác định theo công thức `cpb_removal_delay_length_minus1 + 1`.

Lưu ý 2 – Giá trị của `cpb_removal_delay_length_minus1` mà xác định độ dài (tính theo bit) của phần tử cú pháp `au_cpb_removal_delay_minus1` là giá trị của `cpb_removal_delay_length_minus1` được mã hoá trong bộ thông số video hoặc bộ thông số chuỗi có hiệu lực đối với hình ảnh mã hoá gắn với thông báo SEI định thời hình ảnh, mặc dù `au_cpb_removal_delay_minus1` cộng 1 xác định số lượng nhịp đồng hồ so với thời gian lấy ra đơn vị truy nhập trước đó chứa thông báo SEI về chu kỳ nhớ đệm, có thể là một đơn vị truy nhập của chuỗi dữ liệu video mã hoá khác.

`pic_dpb_output_delay` có thể được sử dụng để tính thời gian xuất ra từ DPB của hình ảnh. Phần tử cú pháp này xác định số lượng nhịp đồng hồ phải chờ sau khi lấy ra đơn vị giải mã cuối cùng trong đơn vị truy nhập từ bộ nhớ CPB trước khi hình ảnh giải mã được xuất ra từ bộ nhớ DPB.

Lưu ý 3 – Hình ảnh không được lấy ra từ bộ nhớ DPB ở thời điểm xuất ra khi nó vẫn còn được đánh dấu là “dùng làm chuẩn ngắn hạn” hoặc “dùng làm chuẩn dài hạn”.

Lưu ý 4 – Duy nhất một giá trị `pic_dpb_output_delay` được xác định cho hình ảnh giải mã.

Độ dài của phần tử cú pháp `pic_dpb_output_delay` có thể được tính theo bit dựa vào công thức `dpb_output_delay_length_minus1 + 1`. Khi `sps_max_dec_pic_buffering[ minTid ]` bằng 1, trong đó `minTid` là giá trị nhỏ nhất trong các giá trị `OpTid` của tất cả các điểm hoạt động mà áp dụng thông báo SEI định thời hình ảnh, thì `pic_dpb_output_delay` bằng 0.

Thời gian xuất ra thu được dựa vào `pic_dpb_output_delay` của hình ảnh bất kỳ được xuất ra từ bộ giải mã tuân theo sự định thời xuất ra có thể trước thời gian xuất ra thu được dựa vào `pic_dpb_output_delay` của mọi hình ảnh trong chuỗi dữ liệu video mã hoá kế tiếp bất kỳ theo thứ tự giải mã.

Thứ tự xuất ra hình ảnh được thiết lập theo các giá trị của phần tử cú pháp này có thể giống như thứ tự được thiết lập theo các giá trị của `PicOrderCntVal`.

Đối với các hình ảnh không được xuất ra bằng quy trình “xuất ra” vì chúng đứng trước, theo thứ tự giải mã, hình ảnh IDR hoặc BLA có `no_output_of_prior_pics_flag` bằng 1 hoặc được suy ra là bằng 1, thời gian xuất ra thu được từ `pic_dpb_output_delay` có thể gia tăng theo giá trị gia tăng của `PicOrderCntVal` so với tất cả các hình ảnh trong cùng chuỗi dữ liệu video mã hoá đó.

`du_common_cpb_removal_delay_flag` bằng 1 xác định rằng phần tử cú pháp `du_common_cpb_removal_delay_minus1` có mặt. `du_common_cpb_removal_delay_flag` bằng 0 xác định rằng phần tử cú pháp `du_common_cpb_removal_delay_minus1` không có mặt.

`du_common_cpb_removal_delay_minus1` cộng 1 xác định số lượng nhịp đồng hồ của hình ảnh con (xem điểm E.2.1) phải chờ trước khi lấy ra từ CPB của mỗi đơn vị giải mã trong đơn vị truy nhập gắn với thông báo SEI định thời hình ảnh, sau khi lấy ra từ CPB đơn vị giải mã trước đó theo thứ tự giải mã. Giá trị này cũng được HSS sử dụng để tính thời gian đến sớm nhất có thể của dữ liệu đơn vị giải mã đến bộ nhớ CPB, như được xác định trong Phụ lục C. Phần tử cú pháp là mã độ dài cố định có độ dài tính theo bit được xác định theo công thức `du_cpb_removal_delay_length_minus1 + 1`.

Như nêu trên, phiên bản dự thảo làm việc HEVC 8 không cung cấp cách để gắn kết thông báo SEI định thời hình ảnh với cấu trúc cú pháp `hrd_parameters()` trong VPS mà cấu trúc cú pháp đi kèm `operation_point_layer_ids()` bao gồm nhiều giá trị của `nuh_reserved_zero_6bits` (tức là ID nhiều lớp trong phiên bản mở rộng mã hóa dữ liệu video đa cảnh nhìn, 3DV hoặc có khả năng chuyển đổi cấp độ của HEVC). Do đó, theo một hoặc nhiều kỹ thuật theo sáng chế, phần tử cú pháp `applicable_operation_points()` trong thông báo SEI định thời hình ảnh chỉ rõ điểm hoạt động mà áp dụng thông báo SEI về chu kỳ nhớ đệm.

Hơn thế nữa, theo một hoặc nhiều kỹ thuật theo sáng chế, cú pháp của thông báo SEI định thời hình ảnh con có thể được thay đổi như được nêu trong bảng 8 dưới đây. Các thay đổi đối với cú pháp của thông báo SEI định thời hình ảnh con có thể cho phép thông báo SEI định thời hình ảnh con bao gồm cấu trúc cú pháp `applicable_operation_points()`. Trong phiên bản dự thảo làm việc HEVC 8, thông báo SEI định thời hình ảnh con không bao gồm cấu trúc cú pháp `applicable_operation_points()`.

Bảng 8 – Thông báo SEI định thời hình ảnh con

<code>sub_pic_timing(payloadSize) {</code>	Bộ mô tả
<code> <code>applicable_operation_points()</code></code>	
<code> <code>du_spt_cpb_removal_delay_minus1</code></code>	u(v)
<code>}</code>	

Mục D.2.2.2 của Phiên bản dự thảo làm việc HEVC 8 mô tả các ngữ nghĩa của thông báo SEI định thời hình ảnh con. Theo một hoặc nhiều kỹ thuật theo sáng chế, mục D.2.2.2 của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi như sau:

Thông báo SEI định thời hình ảnh con cung cấp thông tin về độ trễ lấy ra từ CPB cho đơn vị giải mã gắn với thông báo SEI.

Áp dụng như sau cho cú pháp và các ngữ nghĩa của thông báo SEI định thời hình ảnh con:

- Phần tử cú pháp `sub_pic_cpb_params_present_flag` và `cpb_removal_delay_length_minus1`, và biến `CpbDpbDelaysPresentFlag` được thấy ở hoặc tìm được từ phần tử cú pháp được thấy trong cấu trúc cú pháp `hrd_parameters()` và cấu trúc cú pháp `sub_layer_hrd_parameters()` có thể áp

dùng cho điểm hoạt động bất kỳ trong số các điểm hoạt động mà áp dụng thông báo SEI định thời hình ảnh con.

- *Dòng bit (hoặc một phần của nó) chỉ tập con dòng bit (hoặc một phần của nó) gắn với điểm hoạt động bất kỳ trong số các điểm hoạt động mà áp dụng thông báo SEI định thời hình ảnh con.*

Sự có mặt của thông báo SEI định thời hình ảnh con trong dòng bit được xác định như sau.

- Nếu CpbDpbDelaysPresentFlag bằng 1 và sub_pic_cpb_params_present_flag bằng 1, thì một thông báo SEI định thời hình ảnh con có thể áp dụng cho điểm hoạt động cụ thể có thể có mặt trong mỗi đơn vị giải mã trong chuỗi dữ liệu video mã hóa.
- Ngược lại (CpbDpbDelaysPresentFlag bằng 0 hoặc sub_pic_cpb_params_present_flag bằng 0), không có thông báo SEI định thời hình ảnh con nào có thể áp dụng cho điểm hoạt động cụ thể có mặt trong chuỗi dữ liệu video mã hóa.

Đơn vị giải mã gắn với thông báo SEI định thời hình ảnh con có thể bao gồm, theo thứ tự giải mã, đơn vị NAL SEI chứa thông báo SEI định thời hình ảnh con, tiếp theo là một hoặc nhiều đơn vị NAL không chứa thông báo SEI định thời hình ảnh con, tính cả các đơn vị NAL tiếp theo trong đơn vị truy nhập trở đi nhưng không tính đến đơn vị NAL SEI tiếp theo bất kỳ chứa thông báo SEI định thời hình ảnh con. Có ít nhất một đơn vị NAL có VCL trong mỗi đơn vị giải mã. Tất cả các đơn vị NAL không có VCL gắn với đơn vị NAL có VCL sẽ có trong cùng một đơn vị giải mã.

du_spt_cpb_removal_delay_minus1 cộng 1 xác định số lượng nhịp đồng hồ hình ảnh con phải chờ sau khi lấy ra từ CPB đơn vị giải mã cuối cùng trong đơn vị truy nhập gắn với thông báo SEI về chu kỳ nhớ đệm mới nhất trong đơn vị truy nhập trước đó trước khi lấy ra từ CPB đơn vị giải mã gắn với thông báo SEI định thời hình ảnh con. Giá trị này cũng được HSS sử dụng để tính thời gian đến sớm nhất có thể của dữ liệu đơn vị giải mã đến bộ nhớ CPB, như được xác định trong phụ lục C. Phần tử cú pháp được biểu diễn bởi mã độ dài cố định có độ dài tính theo bit được xác định theo công thức $\text{cpb_removal_delay_length_minus1} + 1$.

Lưu ý – Giá trị của `cpb_removal_delay_length_minus1` xác định độ dài (tính theo bit) của phần tử cú pháp `du_spt_cpb_removal_delay_minus1` là giá trị của `cpb_removal_delay_length_minus1` được mã hoá trong bộ thông số video hoặc bộ thông số chuỗi có hiệu lực đối với đơn vị truy nhập chứa đơn vị giải mã gần với thông báo SEI định thời hình ảnh con, mặc dù `du_spt_cpb_removal_delay_minus1` cộng 1 xác định số lượng nhịp đồng hồ hình ảnh con so với thời gian lấy ra đơn vị giải mã cuối cùng trong đơn vị truy nhập trước đó chứa thông báo SEI về chu kỳ nhớ đệm, đó có thể là một đơn vị truy nhập của chuỗi dữ liệu video mã hoá khác.

Mục E.2.2 của phiên bản dự thảo làm việc HEVC 8 mô tả các ngữ nghĩa của thông số HRD. Theo một hoặc nhiều kỹ thuật theo sáng chế, mục E.2.2 của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi như sau. Các ngữ nghĩa của các phần tử cú pháp này của các thông số HRD không được đề cập dưới đây có thể giống như trong phiên bản dự thảo làm việc HEVC 8.

Cấu trúc cú pháp `hrd_parameters()` mô tả các thông số HRD được sử dụng trong các hoạt động HRD. Khi cấu trúc cú pháp `hrd_parameters()` được bao gồm trong bộ thông số video, thì tập hợp số giá trị `nuh_reserved_zero_6bits` được bao gồm trong `OpLayerIdSet` của điểm hoạt động mà áp dụng cấu trúc cú pháp được xác định bằng cấu trúc cú pháp `operation_point_layer_ids()` tương ứng trong bộ thông số video hoặc được ngầm suy ra, như được xác định trong điểm 7.4.4. Khi cấu trúc cú pháp `hrd_parameters()` được bao gồm trong bộ thông số chuỗi, điểm hoạt động có thể áp dụng là tất cả các điểm hoạt động với `OpLayerIdSet` chứa duy nhất giá trị 0. Theo cách khác, khi cấu trúc cú pháp `hrd_parameters()` được bao gồm trong bộ thông số chuỗi, thì điểm hoạt động có thể áp dụng là tất cả các điểm hoạt động với `OpLayerIdSet` giống với `TargetDecLayerIdSet`.

Điều kiện của sự tương thích dòng bit cho tất cả các cấu trúc cú pháp `hrd_parameters()` trong chuỗi dữ liệu video được mã hóa (trong bộ thông số video hoặc bộ thông số chuỗi) là phải có không quá một trong số các cấu trúc cú pháp này áp dụng được cho cùng một điểm hoạt động. Theo cách khác, cần phải có không quá một cấu trúc cú pháp `hrd_parameters()` trong bộ thông số video áp dụng được cho cùng một điểm hoạt động. Theo cách khác, điều kiện là bộ thông

số video phải không bao gồm cấu trúc cú pháp *hrd_parameters()* áp dụng cho các điểm hoạt động với *OpLayerIdSet* chứa duy nhất giá trị 0.

du_cpb_removal_delay_length_minus1 cộng 1 xác định độ dài, tính theo bit, của phần tử cú pháp *du_cpb_removal_delay_minus1[i]* và *du_common_cpb_removal_delay_minus1* trong thông báo SEI định thời hình ảnh. **cpb_removal_delay_length_minus1** cộng 1 xác định độ dài, tính theo bit, của phần tử cú pháp *au_cpb_removal_delay_minus1* trong thông báo SEI định thời hình ảnh và *phần tử cú pháp du_spt_cpb_removal_delay_minus1* trong thông báo SEI định thời hình ảnh con. Khi không có phần tử cú pháp *cpb_removal_delay_length_minus1*, thì nó được suy ra là bằng 23.

dpb_output_delay_length_minus1 cộng 1 xác định độ dài, tính theo bit, của phần tử cú pháp *pic_dpb_output_delay* trong thông báo SEI định thời hình ảnh. Khi không có phần tử cú pháp *dpb_output_delay_length_minus1*, thì nó được suy ra là bằng 23.

fixed_pic_rate_flag[i] bằng 1 xác định rằng khi *TargetDecHighestTid*is bằng *i*, thì khoảng cách thời gian giữa các thời điểm xuất ra từ HRD của hai hình ảnh liên tiếp bất kỳ theo thứ tự xuất ra được ràng buộc như sau. **fixed_pic_rate_flag[i]** bằng 0 xác định rằng không quy định ràng buộc nào như vậy áp dụng cho khoảng cách thời gian giữa các thời điểm xuất ra từ HRD của hai hình ảnh liên tiếp bất kỳ theo thứ tự xuất ra.

Khi **fixed_pic_rate_flag[i]** không có mặt, thì nó được suy ra là bằng 0.

Khi *TargetDecHighestTid*is bằng *i* và **fixed_pic_rate_flag[i]** bằng 1 đối với chuỗi dữ liệu video được mã hóa chứa hình ảnh *n*, thì giá trị được tính toán cho $\Delta t_{o,dpb}(n)$ như được xác định trong phương trình C-17 sẽ bằng $t_c * (\text{pic_duration_in_tcs_minus1}[i] + 1)$, trong đó t_c là như được xác định trong phương trình C-1 (bằng cách sử dụng giá trị t_c cho chuỗi dữ liệu video được mã hóa chứa hình ảnh *n*) khi một hoặc nhiều trong số các điều kiện sau là đúng đối với hình ảnh *nn* dưới đây được xác định để sử dụng trong phương trình C-17:

- hình ảnh *nn* là ở trong cùng một chuỗi dữ liệu video mã hóa như hình ảnh *n*.
- hình ảnh *nn* là ở trong chuỗi dữ liệu video mã hóa khác và **fixed_pic_rate_flag[i]** bằng 1 trong chuỗi dữ liệu video được mã hóa chứa hình ảnh *nn*, giá trị $\text{num_units_in_tick} \div \text{time_scale}$ là giống nhau đối với cả

hai chuỗi dữ liệu video mã hóa, và giá trị của `pic_duration_in_tc_minus1[ i ]` là giống nhau đối với cả hai chuỗi video mã hóa.

`pic_duration_in_tc_minus1[ i ]` cộng 1 xác định, khi *TargetDecHighestTid* bằng *i*, thì khoảng cách thời gian, tính theo nhịp đồng hồ, giữa các thời điểm xuất ra từ HRD của hai hình ảnh liên tiếp bất kỳ theo thứ tự xuất ra trong chuỗi dữ liệu video mã hóa. Giá trị của `pic_duration_in_tc_minus1[ i ]` sẽ nằm trong khoảng từ 0 đến 2047, bao gồm cả giá trị đầu mút.

`low_delay_hrd_flag[ i ]` xác định chế độ hoạt động HRD, khi *TargetDecHighestTid* bằng *i*, như được xác định trong phụ lục C. Khi `fixed_pic_rate_flag[ i ]` bằng 1, `low_delay_hrd_flag[ i ]` sẽ bằng 0.

Lưu ý 3 – Khi `low_delay_hrd_flag[ i ]` bằng 1, "các hình ảnh lớn" mà làm gián đoạn thời gian lấy ra từ CPB danh định do số lượng các bit được sử dụng bởi một đơn vị truy nhập được cho phép. Mong muốn, nhưng không bắt buộc rằng "các hình ảnh lớn" như vậy chỉ thỉnh thoảng xuất hiện.

`cpb_cnt_minus1[ i ]` cộng 1 xác định số lượng đặc tả CPB khác trong dòng bit của chuỗi dữ liệu video được mã hóa khi *TargetDecHighestTid* bằng *i*. Giá trị của `cpb_cnt_minus1[ i ]` có thể nằm trong khoảng từ 0 đến 31, bao gồm cả giá trị đầu mút. Khi `low_delay_hrd_flag[ i ]` bằng 1, `cpb_cnt_minus1[ i ]` sẽ bằng 0. Khi `cpb_cnt_minus1[ i ]` không có mặt, thì nó được suy ra là bằng 0.

Như được mô tả trong sáng chế này, trong phiên bản dự thảo làm việc HEVC 8, chỉ cấu trúc cú pháp `hrd_parameters( )` trong VPS có thể được chọn cho hoạt động HRD trong khi cấu trúc cú pháp `hrd_parameters( )` trong SPS không bao giờ được chọn. Các thay đổi nêu trên đối với các ngữ nghĩa của cấu trúc cú pháp `hrd_parameters( )` chỉ rõ rằng khi cấu trúc cú pháp `hrd_parameters( )` được bao gồm trong SPS, thì điểm hoạt động mà có thể áp dụng cấu trúc cú pháp `hrd_parameters( )` có thể là tất cả các điểm hoạt động với `OpLayerIdSet` giống với `TargetDecLayerIdSet`. Như nêu trên trong quy trình giải mã chung đã được cải biến, nếu phương tiện bên ngoài có sẵn để thiết lập `TargetDecLayerIdSet`, thì `TargetDecLayerIdSet` có thể được xác định bằng phương tiện bên ngoài. Ngược lại, nếu quy trình giải mã được gọi ra để kiểm tra tương thích dòng bit, thì `TargetDecLayerIdSet` có thể là tập hợp ký hiệu nhận dạng lớp của điểm hoạt động đang kiểm tra. Ngược lại, `TargetDecLayerIdSet` có thể chứa duy nhất một ký hiệu nhận dạng lớp (tức là, chỉ một giá trị của `nuh_reserved_zero_6bits`), giá trị đó bằng 0. Theo

một ví dụ, phương tiện bên ngoài có thể là API, đó là một phần của việc cài đặt cuối cùng và cung cấp chức năng để thiết lập giá trị của TargetDecLayerIdSet. Theo ví dụ này, việc cài đặt cuối cùng có thể bao gồm cài đặt bộ giải mã và các chức năng nhất định không phải là các phần của việc cài đặt bộ giải mã.

Theo cách này, một thiết bị (như bộ mã hóa dữ liệu video 20, bộ giải mã video 30, thiết bị khác 21, hoặc thiết bị khác) có thể chọn, từ trong số bộ thông số HRD trong bộ thông số video và bộ thông số HRD trong SPS, bộ thông số HRD có thể áp dụng cho một điểm hoạt động cụ thể. Ngoài ra, thiết bị này có thể thực hiện, dựa ít nhất một phần vào bộ thông số HRD có thể áp dụng cho một điểm hoạt động cụ thể, sự kiểm tra tương thích dòng bit để kiểm tra xem tập con dòng bit gắn với điểm hoạt động cụ thể có tuân theo tiêu chuẩn mã hóa dữ liệu video không.

Như nêu trên, mục E.2.2 của phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi để chỉ ra rằng khi cấu trúc cú pháp hrd_parameters() có trong bộ thông số chuỗi, thì điểm hoạt động có thể áp dụng là tất cả các điểm hoạt động có OpLayerIdSet giống với TargetDecLayerIdSet. Hơn thế nữa, như nêu trên, TargetDecLayerIdSet được đặt bằng targetOpLayerIdSet, chứa tập hợp các giá trị của nuh_reserved_zero_6bits có mặt trong tập con dòng bit gắn với TargetOp. TargetOp là điểm hoạt động đang kiểm tra trong hoạt động HRD. Hơn thế nữa, hoạt động HRD (ví dụ, kiểm tra tương thích dòng bit và kiểm tra tương thích bộ giải mã) có thể gọi ra quy trình giải mã chung.

Như nêu trên, mục 8.1 của Phiên bản dự thảo làm việc HEVC 8 có thể được thay đổi để đề xuất rằng quy trình tách dòng bit con như được xác định trong điểm 10.1 được áp dụng với TargetDecHighestTid và TargetDecLayerIdSet là đầu vào và đầu ra được gán cho dòng bit được gọi là BitstreamToDecode. Do đó, chỉ các giá trị nuh_reserved_zero_6bits có mặt trong BitstreamToDecode là các giá trị của nuh_reserved_zero_6bits trong TestDecLayerIdSet (tức là, tập hợp các giá trị nuh_reserved_zero_6bits có mặt trong tập con dòng bit gắn với TargetOp). Mục 8.1 còn giải thích rằng khi dịch các ngữ nghĩa của mỗi phần tử cú pháp trong mỗi đơn vị NAL và “dòng bit” hoặc một phần của chúng (ví dụ, chuỗi dữ liệu video được mã hóa) được bao gồm, thì dòng bit hoặc một phần của nó có nghĩa là BitstreamToDecode hoặc một phần của nó.

Do đó, khi dịch phần mô tả các ngữ nghĩa của thông số HRD (ví dụ, mục E.2.2 của phiên bản dự thảo làm việc HEVC 8), thuật ngữ “chuỗi dữ liệu video mã hóa” nghĩa

là một phần của BitstreamToDecode. TargetDecLayerIdSet tương đương với tập hợp gồm tất cả các giá trị của nuh_reserved_zero_6bits có mặt trong BitstreamToDecode. Do đó mà cách diễn đạt trong phần mô tả các ngữ nghĩa của thông số HRD “khi cấu trúc cú pháp hrd_parameters() được bao gồm trong bộ thông số chuỗi, thì điểm hoạt động có thể áp dụng là tất cả các điểm hoạt động có OpLayerIdSet giống với TargetDecLayerIdSet” là tương đương với “khi cấu trúc cú pháp hrd_parameters() được bao gồm trong bộ thông số chuỗi, thì điểm hoạt động có thể áp dụng là tất cả các điểm hoạt động có OpLayerIdSet giống với tập hợp các giá trị của nuh_reserved_zero_6bits có mặt trong BitstreamToDecode.”

Vì “chuỗi dữ liệu video mã hóa” là một phần của BitstreamToDecode, nên tập hợp nuh_reserved_zero_6bits có mặt trong chuỗi dữ liệu video mã hóa là tập con của tập hợp nuh_reserved_zero_6bits có mặt trong BitstreamToDecode. Do đó, cách diễn đạt “khi cấu trúc cú pháp hrd_parameters() được bao gồm trong bộ thông số chuỗi, thì các điểm hoạt động có thể áp dụng là tất cả các điểm hoạt động có OpLayerIdSet giống với tập hợp các giá trị nuh_reserved_zero_6bits có mặt trong BitstreamToDecode” tất yếu là dẫn đến “khi cấu trúc cú pháp hrd_parameters() được bao gồm trong bộ thông số chuỗi, thì điểm hoạt động có thể áp dụng là tất cả các điểm hoạt động có OpLayerIdSet chứa tất cả các giá trị của nuh_reserved_zero_6bits có mặt trong chuỗi dữ liệu video được mã hóa.” Nói cách khác, nếu tập hợp nuh_reserved_zero_6bits của một điểm hoạt động giống với tập hợp nuh_reserved_zero_6bits có mặt trong BitstreamToDecode, thì tập hợp nuh_reserved_zero_6bits của điểm hoạt động đó nhất thiết là phải chứa tất cả các giá trị nuh_reserved_zero_6bits có mặt trong chuỗi dữ liệu video mã hóa của BitstreamToDecode. Theo cách diễn đạt này, “chuỗi dữ liệu video mã hóa” có thể chỉ chuỗi dữ liệu video mã hóa gắn với SPS cụ thể.

Khi tiến hành hoạt động HRD, thiết bị có thể xác định, từ trong số các cấu trúc cú pháp hrd_parameters() được chỉ báo trong VPS và cấu trúc cú pháp hrd_parameters() được chỉ báo trong SPS, cấu trúc cú pháp hrd_parameters() có thể áp dụng cho TargetOp. Cấu trúc cú pháp hrd_parameters() cụ thể trong VPS có thể áp dụng cho TargetOp nếu tập hợp ký hiệu nhận dạng lớp TargetOp so khớp với tập hợp ký hiệu nhận dạng lớp được xác định trong VPS cho cấu trúc cú pháp hrd_parameters() cụ thể. Cấu trúc cú pháp hrd_parameters() trong SPS có thể áp dụng cho TargetOp nếu tập hợp ký hiệu nhận dạng lớp TargetOp (tức là, TargetDecHighestTid) (tức là, tập hợp nuh_reserved_zero_6bits có

mặt trong BitstreamToDecode) chứa tất cả các `nuh_reserved_zero_6bits` có mặt trong chuỗi dữ liệu video được mã hóa của SPS (đó là tập con của tập hợp `nuh_reserved_zero_6bits` trong BitstreamToDecode). Vì tập hợp `nuh_reserved_zero_6bits` của TargetOp có thể nhất thiết phải chứa tất cả các giá trị `nuh_reserved_zero_6bits` có mặt trong chuỗi dữ liệu video được mã hóa gắn với SPS, nên cấu trúc cú pháp `hrd_parameters()` trong SPS luôn luôn có thể áp dụng cho TargetOp. Tuy nhiên, không phải tất cả các SPS đều có cấu trúc cú pháp `hrd_parameters()`. Nếu SPS có cấu trúc cú pháp `hrd_parameters()` và tập hợp `nuh_reserved_zero_6bits` có mặt trong BitstreamToDecode chứa tất cả `nuh_reserved_zero_6bits` có mặt trong chuỗi dữ liệu video mã hóa của SPS, thì cấu trúc cú pháp `hrd_parameters()` của SPS phải được sử dụng. Vì không phải tất cả SPS đều có cấu trúc cú pháp `hrd_parameters()`, nên VPS vẫn có thể được chọn.

Hơn thế nữa, như nêu trên về những thay đổi đối với mục E.2.2 của phiên bản dự thảo làm việc HEVC 8, khi thiết bị tiến hành sự kiểm tra tương thích dòng bit, thì bộ giải mã video có thể xác định rằng dòng bit không tuân theo tiêu chuẩn mã hóa dữ liệu video khi nhiều hơn một bộ thông số HRD áp dụng cho cùng một điểm hoạt động, đối với tất cả các bộ thông số HRD trong chuỗi dữ liệu video được mã hóa. Ngoài ra, khi thiết bị thực hiện sự kiểm tra tương thích dòng bit, thì bộ giải mã video có thể xác định rằng dòng bit không tuân theo tiêu chuẩn mã hóa dữ liệu video khi nhiều hơn một bộ thông số HRD trong VPS áp dụng cho cùng một điểm hoạt động. Hơn thế nữa, khi thiết bị tiến hành sự kiểm tra giải mã dòng bit, thì thiết bị này có thể xác định rằng dòng bit không tuân theo tiêu chuẩn mã hóa dữ liệu video khi VPS bao gồm bộ thông số HRD mà áp dụng cho điểm hoạt động có tập hợp ký hiệu nhận dạng lớp chứa duy nhất giá trị 0.

FIG.3 là sơ đồ khối minh họa bộ giải mã video làm ví dụ 30 được tạo cấu hình để thực hiện các kỹ thuật của sáng chế. FIG.3 được đưa ra nhằm mục đích giải thích và không giới hạn đối với các kỹ thuật như được lấy làm ví dụ và được mô tả rộng rãi trong sáng chế này. Để giải thích, sáng chế mô tả bộ giải mã video 30 trong trường hợp mã hóa HEVC. Tuy nhiên, các kỹ thuật của sáng chế có thể áp dụng cho các tiêu chuẩn hoặc phương pháp mã hóa khác.

Trong ví dụ của FIG.3, bộ giải mã video 30 bao gồm bộ giải mã entropy 150, bộ xử lý dự báo 152, bộ lượng tử hóa ngược 154, bộ xử lý biến đổi ngược 156, bộ phục hồi lại 158, bộ lọc 160, và bộ nhớ đệm hình ảnh giải mã 162. Bộ xử lý dự báo 152 bao gồm

bộ bù chuyển động 164 và bộ xử lý dự báo nội cấu trúc 166. Theo các ví dụ khác, bộ giải mã video 30 có thể bao gồm nhiều bộ phận chức năng hơn, ít bộ phận chức năng hơn hoặc các bộ phận chức năng khác.

Bộ nhớ đệm hình ảnh mã hóa (coded picture buffer: CPB) 151 có thể thu và lưu trữ các dữ liệu video mã hóa (ví dụ, các đơn vị NAL) của dòng bit. Bộ giải mã entropy 150 có thể thu các đơn vị NAL từ CPB 151 và phân tích cú pháp các đơn vị NAL này để giải mã các phần tử cú pháp. Bộ giải mã entropy 150 có thể giải mã entropy các phần tử cú pháp đã được mã hóa entropy trong các đơn vị NAL. Bộ xử lý dự báo 152, bộ lượng tử hóa ngược 154, bộ xử lý biến đổi ngược 156, bộ phục hồi lại 158, và bộ lọc 160 có thể tạo ra dữ liệu video giải mã dựa vào các phần tử cú pháp được tách ra từ dòng bit.

Các đơn vị NAL của dòng bit có thể bao gồm các đơn vị NAL lát mã hóa. Là một phần của quy trình giải mã dòng bit, bộ giải mã entropy 150 có thể tách và giải mã entropy các phần tử cú pháp từ các đơn vị NAL lát mã hóa. Mỗi lát mã hóa có thể bao gồm phần đầu lát và dữ liệu lát. Phần đầu lát có thể chứa phần tử cú pháp gắn liền với lát. Phần tử cú pháp trong phần đầu lát có thể bao gồm phần tử cú pháp nhận dạng PPS gắn với hình ảnh chứa lát.

Ngoài việc giải mã các phần tử cú pháp từ dòng bit, bộ giải mã video 30 có thể tiến hành hoạt động phục hồi lại đối với CU chưa phân tách. Để thực hiện hoạt động phục hồi lại đối với CU chưa phân tách, bộ giải mã video 30 có thể thực hiện hoạt động phục hồi lại đối với mỗi TU của CU. Bằng cách tiến hành hoạt động phục hồi lại đối với mỗi TU của CU, bộ giải mã video 30 có thể phục hồi lại khối dữ liệu dư của CU.

Là một phần của việc thực hiện hoạt động phục hồi lại đối với TU của CU, bộ lượng tử hóa ngược 154 có thể lượng tử hóa ngược, tức là khử lượng tử hóa, các khối hệ số gắn với TU. Bộ lượng tử hóa ngược 154 có thể sử dụng giá trị QP gắn với CU của TU để xác định mức độ lượng tử hóa và, tương tự, mức lượng tử hóa ngược của bộ lượng tử hóa ngược 154 để áp dụng. Cụ thể là, tỷ số nén, tức là, tỷ số của lượng bit được sử dụng để biểu diễn chuỗi ban đầu và chuỗi được nén, có thể được kiểm soát bằng cách điều chỉnh giá trị QP được sử dụng khi lượng tử hóa hệ số biến đổi. Tỷ số nén còn có thể phụ thuộc vào phương pháp mã hóa entropy được sử dụng.

Sau khi bộ lượng tử hóa ngược 154 lượng tử hóa ngược khối hệ số, bộ xử lý biến đổi ngược 156 có thể áp dụng một hoặc nhiều kỹ thuật biến đổi ngược cho khối hệ số để tạo ra khối dữ liệu dư gắn với TU. Ví dụ, bộ xử lý biến đổi ngược 156 có thể áp dụng kỹ

thuật DCT ngược, kỹ thuật biến đổi số nguyên ngược, kỹ thuật biến đổi Karhunen-Loeve ngược (Karhunen-Loeve transform-KLT), kỹ thuật biến đổi quay vòng ngược, kỹ thuật biến đổi định hướng ngược, hoặc kỹ thuật biến đổi ngược khác cho khối hệ số.

Nếu PU được mã hóa bằng cách sử dụng kỹ thuật dự báo nội cấu trúc, bộ xử lý dự báo nội cấu trúc 166 có thể tiến hành dự báo nội cấu trúc để tạo ra các khối dự báo cho PU. Bộ xử lý dự báo nội cấu trúc 166 có thể sử dụng chế độ dự báo nội cấu trúc để tạo ra khối độ chói, khối Cb và khối Cr dự báo cho PU dựa vào các khối dự báo của PU liền kề về mặt không gian. Bộ xử lý dự báo nội cấu trúc 166 có thể xác định chế độ dự báo nội cấu trúc cho PU dựa vào một hoặc nhiều phần tử cú pháp được giả mã từ dòng bit.

Bộ xử lý dự báo 152 có thể thiết lập danh mục hình ảnh chuẩn thứ nhất (RefPicList0) và danh mục hình ảnh chuẩn thứ hai (RefPicList1) dựa vào phần tử cú pháp được tách ra từ dòng bit. Hơn thế nữa, nếu PU được mã hóa bằng cách sử dụng kỹ thuật dự báo liên cấu trúc, bộ giải mã entropy 150 có thể tách thông tin chuyển động cho PU. Bộ bù chuyển động 164 có thể xác định, dựa vào thông tin chuyển động của PU, một hoặc nhiều miền chuẩn cho PU. Bộ bù chuyển động 164 có thể tạo ra, dựa vào các khối mẫu với một hoặc nhiều khối chuẩn cho PU, khối độ chói, khối Cb và khối Cr dự báo cho PU.

Bộ phục hồi lại 158 có thể sử dụng khối biến đổi độ chói, khối biến đổi Cb và Cr gắn với TU của CU và khối độ chói, khối Cb và khối Cr dự báo của PU của CU, tức là, dữ liệu dự báo nội cấu trúc hoặc dữ liệu dự báo liên cấu trúc, nếu có thể áp dụng, để phục hồi lại khối mã hóa độ chói, Cb và Cr của CU. Ví dụ, bộ phục hồi lại 158 có thể bổ sung các mẫu của các khối biến đổi độ chói, Cb và Cr vào các mẫu tương ứng của các khối độ chói, Cb và Cr dự báo để phục hồi lại khối mã hóa độ chói, Cb và Cr của CU.

Bộ lọc 160 có thể thực hiện hoạt động tách khối để loại bỏ các thành phần lạ tạo khối gắn với khối mã hóa độ chói, Cb và Cr của CU. Bộ giải mã video 30 có thể lưu trữ khối mã hóa độ chói, Cb và Cr của CU trong bộ nhớ đệm hình ảnh giải mã 162. Bộ nhớ đệm hình ảnh giải mã 162 có thể cung cấp các hình ảnh chuẩn cho quy trình bù chuyển động tiếp theo, dự báo nội cấu trúc, và trình diễn trên thiết bị hiển thị, như thiết bị hiển thị 32 của FIG.1. Ví dụ, bộ giải mã video 30 có thể tiến hành, dựa vào khối độ chói, Cb và Cr trong bộ nhớ đệm hình ảnh giải mã 162, hoạt động dự báo nội cấu trúc hoặc dự báo liên cấu trúc ở PU của CU khác. Theo cách này, bộ giải mã video 30 có thể giải mã, từ dòng bit, mức hệ số biến đổi của khối hệ số độ chói đáng kể, lượng tử hóa ngược mức hệ

số biến đổi này, áp dụng kỹ thuật biến đổi cho mức hệ số biến đổi để tạo ra khối biến đổi, tạo ra, dựa ít nhất một phần vào khối biến đổi, khối mã hóa, và xuất ra khối mã hóa để hiển thị.

FIG.4 là lưu đồ minh họa hoạt động làm ví dụ 200 của thiết bị, theo một hoặc nhiều kỹ thuật theo sáng chế. Hoạt động 200 có thể được thực hiện bằng bộ mã hóa dữ liệu video 20, bộ giải mã video 30, thiết bị khác 21, hoặc thiết bị khác. Như được minh họa trong ví dụ của FIG.4, thiết bị có thể chọn, từ trong số bộ thông số HRD giả định (ví dụ, cấu trúc cú pháp `hrd_parameters`) trong VPS và bộ thông số HRD trong SPS, bộ thông số HRD có thể áp dụng cho một điểm hoạt động cụ thể của dòng bit (202). Ngoài ra, thiết bị này có thể tiến hành, dựa ít nhất một phần vào bộ thông số HRD có thể áp dụng cho một điểm hoạt động cụ thể, hoạt động HRD dựa vào tập con trong dòng bit gắn với điểm hoạt động cụ thể (204). Ví dụ, thiết bị này có thể tiến hành sự kiểm tra tương thích dòng bit hoặc sự kiểm tra tương thích bộ giải mã.

FIG.5 là lưu đồ minh họa hoạt động làm ví dụ 250 của thiết bị, theo một hoặc nhiều kỹ thuật theo sáng chế. Hoạt động 200 có thể được tiến hành bằng bộ mã hóa dữ liệu video 20, bộ giải mã video 30, thiết bị khác 21, hoặc thiết bị khác. Như được minh họa trong ví dụ của FIG.5, thiết bị này có thể tiến hành sự kiểm tra tương thích dòng bit để xác định liệu dòng bit có tuân theo tiêu chuẩn mã hóa dữ liệu video hay không (252). Thiết bị này có thể tiến hành quy trình giải mã là một phần của việc thực hiện sự kiểm tra tương thích dòng bit (254).

Như được minh họa trong ví dụ của FIG.5, khi tiến hành quy trình giải mã, thiết bị này có thể tiến hành quy trình tách dòng bit để tách, từ dòng bit, dạng biểu diễn điểm hoạt động của điểm hoạt động được xác định bởi tập hợp các ký hiệu nhận dạng lớp đích và ký hiệu nhận dạng theo thời gian cao nhất của đích (256). Tập hợp các ký hiệu nhận dạng lớp đích có thể chứa các giá trị của các phần tử cú pháp nhận dạng lớp có mặt trong dạng biểu diễn điểm hoạt động. Tập hợp các ký hiệu nhận dạng lớp đích có thể là tập con các giá trị của các phần tử cú pháp nhận dạng lớp của dòng bit. Ký hiệu nhận dạng theo thời gian cao nhất của đích có thể bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dạng biểu diễn điểm hoạt động, ký hiệu nhận dạng theo thời gian cao nhất của đích nhỏ hơn hoặc bằng ký hiệu nhận dạng theo thời gian lớn nhất có mặt trong dòng bit. Hơn thế nữa, thiết bị này có thể giải mã đơn vị NAL của dạng biểu diễn điểm hoạt động (258).

FIG.6 là lưu đồ minh họa hoạt động HRD làm ví dụ 300 của thiết bị, theo một hoặc nhiều kỹ thuật theo sáng chế. Hoạt động HRD 300 có thể được tiến hành bằng bộ mã hóa dữ liệu video 20, bộ giải mã video 30, thiết bị khác 21, hoặc thiết bị khác. Các thiết bị khác có thể bao gồm bộ kiểm tra dòng bit tương thích lấy một dòng bit làm đầu vào, và xuất ra dấu hiệu chỉ báo dòng bit nhập vào có phải là dòng bit tương thích hay không. Theo một số ví dụ, hoạt động HRD 300 có thể xác định sự tương thích của dòng bit với tiêu chuẩn mã hóa dữ liệu video. Theo các ví dụ khác, hoạt động HRD 300 có thể xác định sự tương thích của bộ giải mã với tiêu chuẩn mã hóa dữ liệu video. Là một phần của việc tiến hành hoạt động HRD 300, thiết bị này có thể xác định ký hiệu nhận dạng theo thời gian cao nhất của tập con dòng bit gắn với điểm hoạt động được chọn của dòng bit (302). Ngoài ra, thiết bị này có thể xác định, dựa vào ký hiệu nhận dạng theo thời gian cao nhất, phần tử cú pháp cụ thể từ trong số mảng phần tử cú pháp (ví dụ, `sps_max_num_reorder_pics[i]`, `sps_max_dec_pic_buffering[i]`, và `cpb_cnt_minus1[i]`) (304). Thiết bị có thể sử dụng phần tử cú pháp cụ thể trong hoạt động HRD (306).

Theo một hoặc nhiều phương án làm ví dụ, các chức năng đã mô tả có thể được thực hiện bằng phần cứng, phần mềm, phần sụn, hoặc kết hợp các loại này. Nếu được thực hiện bằng phần mềm, thì các chức năng này có thể được lưu trữ hoặc được truyền dưới dạng một hoặc nhiều lệnh hoặc mã, vật ghi đọc được bằng máy tính và được thi hành bằng bộ phận xử lý dựa trên phần cứng. Vật ghi đọc được bằng máy tính có thể là phương tiện lưu trữ đọc được bằng máy tính, tương ứng với phương tiện hữu hình như phương tiện lưu trữ dữ liệu, hoặc phương tiện truyền thông bao gồm mọi phương tiện tạo điều kiện thuận lợi cho việc truyền chương trình máy tính từ nơi này đến nơi khác, ví dụ, theo giao thức truyền thông. Theo cách này, vật ghi đọc được bằng máy tính thường có thể tương ứng với (1) phương tiện lưu trữ hữu hình đọc được bằng máy tính thuộc loại không khả biến hoặc (2) phương tiện truyền thông như tín hiệu hoặc sóng mang. Phương tiện lưu trữ dữ liệu có thể là mọi loại phương tiện có sẵn có thể truy nhập được bằng một hoặc nhiều máy tính hoặc một hay nhiều bộ xử lý để tìm kiếm các lệnh, mã và/hoặc cấu trúc dữ liệu để tiến hành các kỹ thuật nêu trong sáng chế. Sản phẩm chứa chương trình máy tính có thể là vật ghi đọc được bằng máy tính.

Ví dụ, phương tiện lưu trữ đọc được bằng máy tính như vậy có thể là bộ nhớ truy nhập ngẫu nhiên (*Random Access Memory-RAM*), bộ nhớ chỉ đọc (*Read Only Memory-ROM*), bộ nhớ chỉ đọc lập trình được xóa được bằng điện (*Electrically Erasable*

Programmable Read Only Memory-EEPROM), đĩa compac-bộ nhớ chỉ đọc (*Compact Disc-Read Only Memory-CD-ROM*) hoặc đĩa quang khác, đĩa từ hoặc thiết bị nhớ từ tính khác, thiết bị nhớ truy nhập nhanh, hoặc mọi phương tiện khác có thể dùng để lưu trữ mã chương trình mong muốn dưới dạng lệnh hoặc cấu trúc dữ liệu và có thể truy nhập được bằng máy tính. Ngoài ra, mọi loại kết nối được gọi theo cách thích hợp là phương tiện đọc được bằng máy tính. Ví dụ, nếu các lệnh được truyền từ website, máy chủ hoặc nguồn từ xa khác sử dụng cáp đồng trục, cáp sợi quang, cáp xoắn đôi, đường thuê bao số (*Digital Subscriber Line- DSL*), hoặc sử dụng công nghệ không dây như hồng ngoại, vô tuyến và vi ba, thì cáp đồng trục, cáp sợi quang, cáp xoắn đôi, DSL, hoặc công nghệ không dây như hồng ngoại, vô tuyến và vi ba đó cũng nằm trong định nghĩa phương tiện. Tuy nhiên, cần phải hiểu rằng, phương tiện lưu trữ đọc được bằng máy tính và phương tiện lưu trữ dữ liệu không bao gồm các loại kết nối, sóng mang, tín hiệu hoặc phương tiện khả biến khác, mà chỉ bao gồm phương tiện lưu trữ hữu hình, không khả biến. Đĩa từ và đĩa quang, như được sử dụng ở đây, bao gồm đĩa compac (*Compact Disc-CD*), đĩa laze, đĩa quang, đĩa số đa năng (*Digital Versatile Disc-DVD*), đĩa mềm và đĩa blu-ray, trong đó đĩa từ thường tái tạo dữ liệu bằng phương pháp từ tính, còn đĩa quang thì tái tạo dữ liệu bằng phương pháp quang học sử dụng laze. Dạng kết hợp của các loại phương tiện nêu trên cũng được coi là nằm trong phạm vi vật ghi đọc được bằng máy tính.

Các lệnh có thể được thi hành bằng một hoặc nhiều bộ xử lý, như một hoặc nhiều bộ xử lý tín hiệu số (*digital signal processor-DSP*), bộ vi xử lý đa năng, mạch tích hợp chuyên dụng (*application specific integrated circuit-ASIC*), mảng logic lập trình được bằng trường (*field programmable logic array-FPGA*), hoặc mạch logic rời rạc hoặc mạch tích hợp tương đương khác. Do đó, thuật ngữ “bộ xử lý”, như được sử dụng trong sáng chế, có thể dùng để chỉ mọi cấu trúc nêu trên hoặc mọi cấu trúc khác phù hợp để thực hiện các kỹ thuật nêu trong sáng chế. Ngoài ra, theo một số khía cạnh, chức năng nêu trong sáng chế có thể được thực hiện bằng môđun phần cứng và/hoặc phần mềm chuyên dụng được tạo cấu hình để mã hoá và giải mã, hoặc được kết hợp thành một bộ mã hoá-giải mã kết hợp. Đồng thời, các kỹ thuật có thể được thực hiện hoàn toàn trong một hoặc nhiều mạch hoặc phần tử logic.

Kỹ thuật theo sáng chế có thể được áp dụng trong rất nhiều thiết bị hoặc cơ cấu, trong đó có thiết bị cầm tay không dây, mạch tích hợp (*Integrated Circuit-IC*) hoặc bộ IC (ví dụ, bộ chip). Các thành phần, môđun, hoặc bộ phận được mô tả trong sáng chế để

nhấn mạnh các khía cạnh chức năng của các thiết bị được tạo cấu hình để tiến hành các kỹ thuật đã bộc lộ, nhưng không nhất thiết phải được thực hiện bằng các bộ phận phần cứng khác nhau. Thực ra, như đã nêu trên, các bộ phận khác nhau có thể được kết hợp lại thành một bộ phận phần cứng mã hoá-giải mã hoặc được thực hiện bởi một tập hợp các bộ phận phần cứng tương tác với nhau, có một hoặc nhiều bộ xử lý như đã nêu trên, kết hợp với phần mềm và/hoặc phần sụn phù hợp.

Nhiều phương án làm ví dụ đã được mô tả. Các phương án làm ví dụ này và các phương án khác làm ví dụ khác đều nằm trong phạm vi của sáng chế như được xác định theo các điểm yêu cầu bảo hộ dưới đây.

YÊU CẦU BẢO HỘ

1. Phương pháp xử lý dữ liệu video, phương pháp này bao gồm các bước:

thu bộ thông số video (video parameter set - VPS) và bộ thông số chuỗi (sequence parameter set - SPS), VPS bao gồm nhiều cấu trúc cú pháp chứa các thông số bộ giải mã chuẩn giả định (Hypothetical Reference Decoder - HRD) bao gồm các bộ thông số HRD tương ứng, VPS còn bao gồm, đối với mỗi cấu trúc cú pháp chứa các thông số HRD tương ứng trong VPS, các phần tử cú pháp biểu thị bộ ký hiệu nhận dạng lớp tương ứng với cấu trúc cú pháp chứa các thông số HRD tương ứng, và SPS bao gồm cấu trúc cú pháp chứa các thông số HRD bao gồm bộ thông số HRD;

lựa chọn bộ thông số HRD có khả năng áp dụng cho điểm hoạt động cụ thể, trong đó bộ thông số HRD trong cấu trúc cú pháp chứa các thông số HRD trong SPS có khả năng áp dụng cho điểm hoạt động cụ thể dựa vào bộ ký hiệu nhận dạng lớp của điểm hoạt động cụ thể giống với bộ gồm tất cả các ký hiệu nhận dạng lớp có mặt trong dòng bit mà đang trải qua kiểm tra về độ tương thích của dòng bit, và bộ thông số HRD trong cấu trúc cú pháp chứa thông số HRD cụ thể trong số nhiều cấu trúc cú pháp chứa thông số HRD trong VPS có khả năng áp dụng cho điểm hoạt động cụ thể dựa vào bộ ký hiệu nhận dạng lớp của điểm hoạt động cụ thể được chỉ rõ bởi các phần tử cú pháp biểu thị các ký hiệu nhận dạng lớp cho cấu trúc cú pháp chứa các thông số HRD cụ thể trong VPS; và

thực hiện hoạt động HRD bằng cách sử dụng bộ thông số HRD được chọn.

2. Phương pháp theo điểm 1, trong đó bộ thông số HRD có khả năng áp dụng được cho điểm hoạt động cụ thể bao gồm ít nhất các thông số biểu thị độ trễ lấy ra từ bộ nhớ đệm hình ảnh mã hóa (coding picture buffer - CPB) ban đầu, dung lượng CPB, và tốc độ bit.

3. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm bước:

xác định ký hiệu nhận dạng theo thời gian đích của điểm hoạt động cụ thể mà bằng với ký hiệu nhận dạng theo thời gian lớn nhất có trong dòng bit.

4. Phương pháp theo điểm 1, trong đó bước thực hiện hoạt động HRD bao gồm thực hiện

kiểm tra về độ tương thích của dòng bit, quá trình kiểm tra về độ tương thích của dòng bit này xác định liệu dòng bit có phù hợp với chuẩn mã hóa video hay không.

5. Phương pháp theo điểm 1, trong đó hoạt động HRD xác định độ tương thích của dòng bit với chuẩn mã hóa video hoặc xác định độ tương thích của bộ giải mã video với chuẩn mã hóa video.

6. Phương pháp theo điểm 4, trong đó bước thực hiện kiểm tra về độ tương thích của dòng bit bao gồm xác định rằng dòng bit không phù hợp với chuẩn mã hóa video khi có nhiều hơn một trong số các bộ thông số HRD trong chuỗi video mã hóa áp dụng cho cùng một điểm hoạt động cụ thể, đối với tất cả các bộ thông số HRD trong chuỗi video mã hóa.

7. Phương pháp theo điểm 4, trong đó bước thực hiện kiểm tra về độ tương thích của dòng bit bao gồm xác định rằng dòng bit không phù hợp với chuẩn mã hóa video khi có nhiều hơn một bộ thông số HRD trong VPS áp dụng cho cùng một điểm hoạt động cụ thể.

8. Phương pháp theo điểm 4, trong đó bước thực hiện kiểm tra về độ tương thích của dòng bit bao gồm xác định rằng dòng bit không phù hợp với chuẩn mã hóa video khi VPS chứa bộ thông số HRD mà áp dụng cho các điểm hoạt động có các bộ ký hiệu nhận dạng lớp chứa duy nhất trị số 0.

9. Thiết bị xử lý video bao gồm:

phương tiện lưu trữ để lưu trữ dữ liệu video được mã hóa; và
một hoặc nhiều bộ xử lý được tạo cấu hình để:

thu bộ thông số video (VPS) và bộ thông số chuỗi (SPS), VPS bao gồm nhiều cấu trúc cú pháp chứa các thông số bộ giải mã chuẩn giả định (HRD) bao gồm các bộ thông số HRD tương ứng, VPS còn bao gồm, đối với mỗi cấu trúc cú pháp chứa các thông số HRD tương ứng trong VPS, các phần tử cú pháp biểu thị bộ ký hiệu nhận dạng lớp tương ứng với cấu trúc cú pháp chứa các thông số HRD

tương ứng, và SPS bao gồm cấu trúc cú pháp chứa các thông số HRD chứa bộ thông số HRD;

lựa chọn bộ thông số HRD có khả năng áp dụng cho điểm hoạt động cụ thể, trong đó một hoặc nhiều bộ xử lý được tạo cấu hình để xác định rằng bộ thông số HRD trong cấu trúc cú pháp chứa các thông số HRD trong SPS có khả năng áp dụng được cho điểm hoạt động cụ thể dựa vào bộ ký hiệu nhận dạng lớp của điểm hoạt động cụ thể giống với bộ gồm tất cả các ký hiệu nhận dạng lớp có trong dòng bit mà đang trải qua kiểm tra về độ tương thích của dòng bit, và được tạo cấu hình để xác định rằng bộ thông số HRD trong cấu trúc cú pháp chứa thông số HRD cụ thể trong số nhiều cấu trúc cú pháp chứa thông số HRD trong VPS có khả năng áp dụng cho điểm hoạt động cụ thể dựa vào bộ ký hiệu nhận dạng lớp của điểm hoạt động cụ thể được chỉ rõ bởi các phần tử cú pháp biểu thị các ký hiệu nhận dạng lớp cho cấu trúc cú pháp chứa các thông số HRD cụ thể trong VPS; và

thực hiện hoạt động HRD bằng cách sử dụng bộ thông số HRD được chọn.

10. Thiết bị xử lý video theo điểm 9, trong đó bộ thông số HRD có khả năng áp dụng cho điểm hoạt động cụ thể bao gồm ít nhất các thông số biểu thị độ trễ lấy ra từ bộ nhớ đệm hình ảnh mã hóa (CPB) ban đầu, dung lượng CPB, và tốc độ bit.

11. Thiết bị xử lý video theo điểm 9, trong đó một hoặc nhiều bộ xử lý được tạo cấu hình để:

xác định ký hiệu nhận dạng theo thời gian đích của điểm hoạt động cụ thể mà bằng với ký hiệu nhận dạng theo thời gian lớn nhất có trong dòng bit.

12. Thiết bị xử lý video theo điểm 9, trong đó một hoặc nhiều bộ xử lý được tạo cấu hình để thực hiện kiểm tra về độ tương thích của dòng bit, quá trình kiểm tra về độ tương thích của dòng bit xác định liệu dòng bit có phù hợp với chuẩn mã hóa video hay không.

13. Thiết bị xử lý video theo điểm 12, trong đó một hoặc nhiều bộ xử lý được tạo cấu hình để xác định rằng dòng bit không phù hợp với chuẩn mã hóa video khi có nhiều hơn một trong số các bộ thông số HRD trong chuỗi video mã hóa áp dụng cho cùng một điểm hoạt động cụ thể, đối với tất cả các bộ thông số HRD trong chuỗi video mã hóa.

14. Thiết bị xử lý video theo điểm 12, trong đó một hoặc nhiều bộ xử lý được tạo cấu hình để xác định rằng dòng bit không phù hợp với chuẩn mã hóa video khi có nhiều hơn một bộ thông số HRD trong VPS áp dụng cho cùng một điểm hoạt động.

15. Thiết bị xử lý video theo điểm 12, trong đó một hoặc nhiều bộ xử lý được tạo cấu hình để xác định rằng dòng bit không phù hợp với chuẩn mã hóa video khi VPS chứa bộ thông số HRD mà áp dụng cho các điểm hoạt động có các bộ ký hiệu nhận dạng lớp chứa duy nhất trị số 0.

16. Thiết bị xử lý video bao gồm:

phương tiện thu bộ thông số video (VPS) và bộ thông số chuỗi (SPS), VPS bao gồm nhiều cấu trúc cú pháp chứa các thông số bộ giải mã chuẩn giả định (HRD) bao gồm các bộ thông số HRD tương ứng, VPS còn bao gồm, đối với mỗi cấu trúc cú pháp chứa các thông số HRD tương ứng trong VPS, các phần tử cú pháp biểu thị bộ ký hiệu nhận dạng lớp tương ứng với cấu trúc cú pháp chứa các thông số HRD tương ứng, và SPS bao gồm cấu trúc cú pháp chứa các thông số HRD chứa bộ thông số HRD;

phương tiện lựa chọn bộ thông số HRD có khả năng áp dụng cho điểm hoạt động cụ thể, trong đó bộ thông số HRD trong cấu trúc cú pháp chứa các thông số HRD trong SPS có khả năng áp dụng cho điểm hoạt động cụ thể dựa vào bộ ký hiệu nhận dạng lớp của điểm hoạt động cụ thể giống với bộ gồm tất cả các ký hiệu nhận dạng lớp có trong dòng bit mà đang trải qua kiểm tra về độ tương thích của dòng bit, và bộ thông số HRD trong cấu trúc cú pháp chứa thông số HRD cụ thể trong số nhiều cấu trúc cú pháp chứa thông số HRD trong VPS có khả năng áp dụng cho điểm hoạt động cụ thể dựa vào bộ ký hiệu nhận dạng lớp của điểm hoạt động cụ thể được chỉ rõ bởi các phần tử cú pháp biểu thị các ký hiệu nhận dạng lớp cho cấu trúc cú pháp chứa các thông số HRD cụ thể trong VPS; và

phương tiện thực hiện hoạt động HRD bằng cách sử dụng bộ thông số HRD được chọn.

17. Thiết bị xử lý video theo điểm 16, trong đó thiết bị này còn bao gồm:

phương tiện xác định ký hiệu nhận dạng theo thời gian đích của điểm hoạt động cụ thể mà bằng với ký hiệu nhận dạng theo thời gian lớn nhất có trong dòng bit.

18. Phương tiện lưu trữ dữ liệu bất biến đọc được bằng máy tính lưu trữ trên đó các lệnh mà, khi được thực thi bởi một hoặc nhiều bộ xử lý của thiết bị xử lý video, tạo cấu hình cho thiết bị xử lý video để:

thu bộ thông số video (VPS) và bộ thông số chuỗi (SPS), VPS bao gồm nhiều cấu trúc cú pháp chứa các thông số bộ giải mã chuẩn giả định (HRD) bao gồm các bộ thông số HRD tương ứng, VPS còn bao gồm, đối với mỗi cấu trúc cú pháp chứa các thông số HRD tương ứng trong VPS, các phần tử cú pháp biểu thị bộ ký hiệu nhận dạng lớp tương ứng với cấu trúc cú pháp chứa các thông số HRD tương ứng, và SPS bao gồm cấu trúc cú pháp chứa các thông số HRD bao gồm bộ thông số HRD;

lựa chọn bộ thông số HRD có khả năng áp dụng cho điểm hoạt động cụ thể, trong đó bộ thông số HRD trong cấu trúc cú pháp chứa các thông số HRD trong SPS có khả năng áp dụng cho điểm hoạt động cụ thể dựa vào bộ ký hiệu nhận dạng lớp của điểm hoạt động cụ thể giống với bộ gồm tất cả các ký hiệu nhận dạng lớp có trong dòng bit mà đang trải qua kiểm tra về độ tương thích của dòng bit, và bộ thông số HRD trong cấu trúc cú pháp chứa thông số HRD cụ thể trong số nhiều cấu trúc cú pháp chứa thông số HRD trong VPS có khả năng áp dụng cho điểm hoạt động cụ thể dựa vào bộ ký hiệu nhận dạng lớp của điểm hoạt động cụ thể được chỉ rõ bởi các phần tử cú pháp biểu thị các ký hiệu nhận dạng lớp cho cấu trúc cú pháp chứa các thông số HRD cụ thể trong VPS; và

thực hiện hoạt động HRD bằng cách sử dụng bộ thông số HRD được chọn.

19. Phương tiện lưu trữ dữ liệu bất biến đọc được bằng máy tính theo điểm 18, trong đó các lệnh này tạo cấu hình cho thiết bị xử lý video để:

xác định ký hiệu nhận dạng theo thời gian đích của điểm hoạt động cụ thể mà bằng với ký hiệu nhận dạng theo thời gian lớn nhất có trong dòng bit.

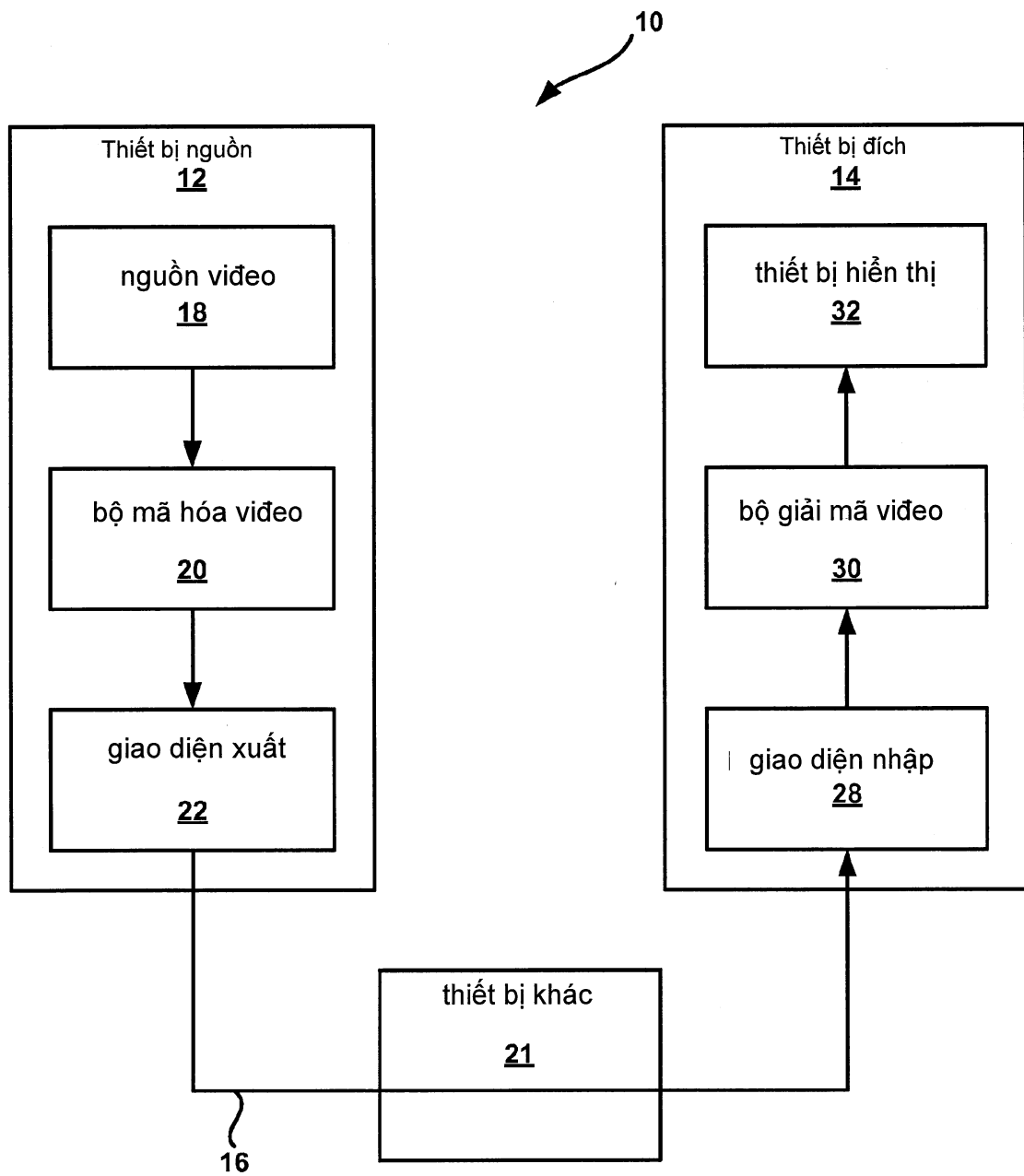


FIG. 1

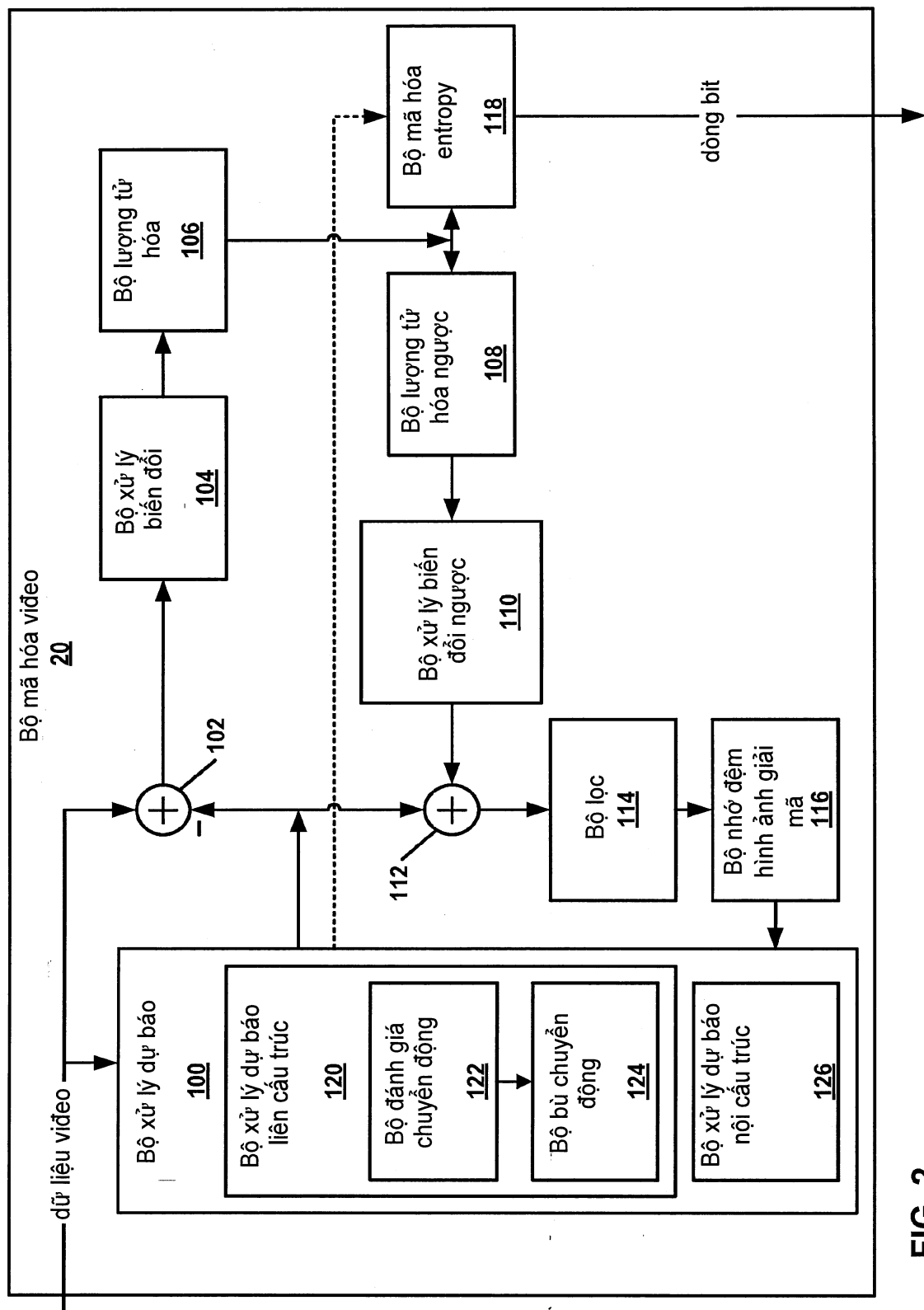


FIG. 2

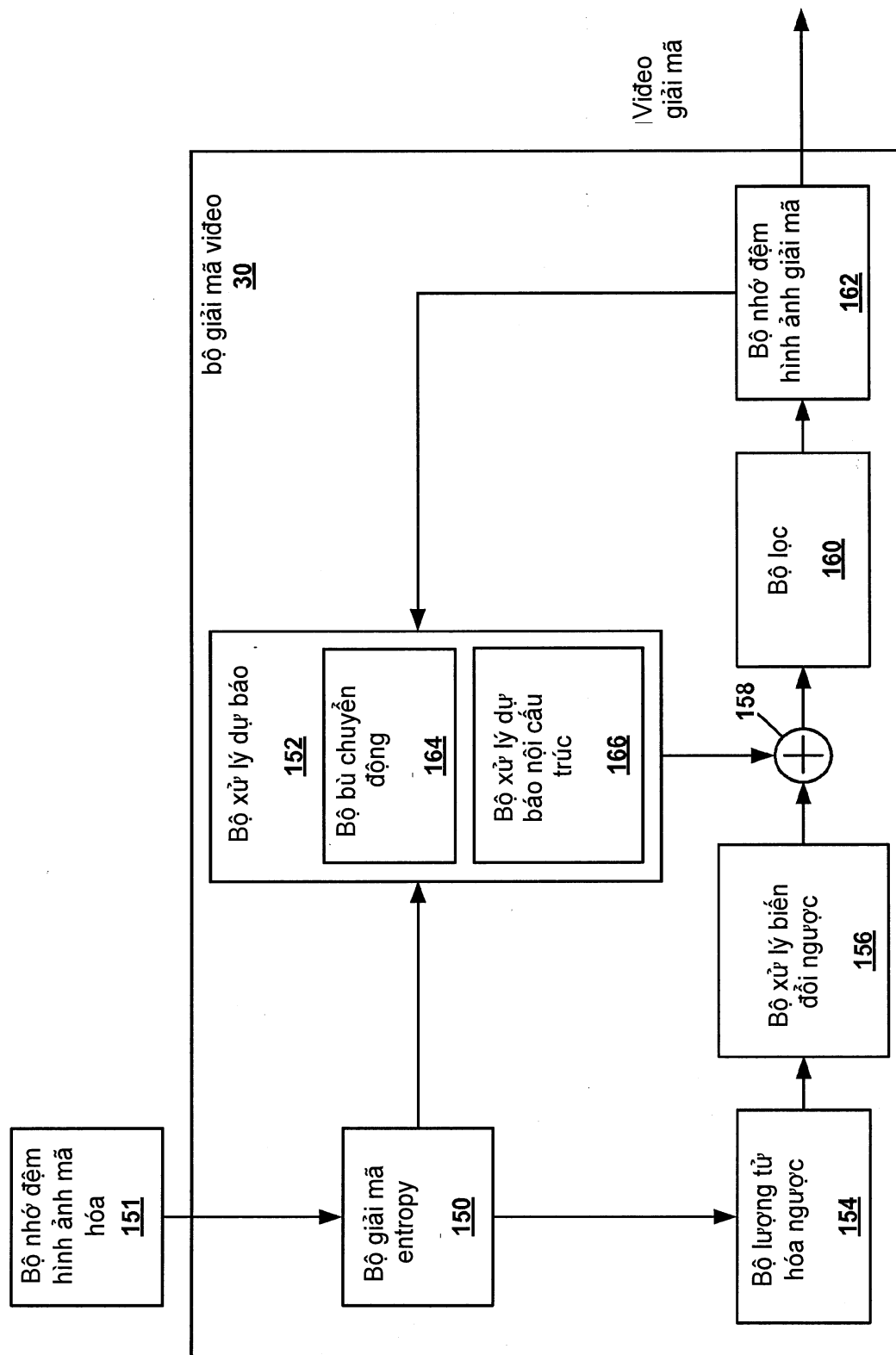


FIG. 3

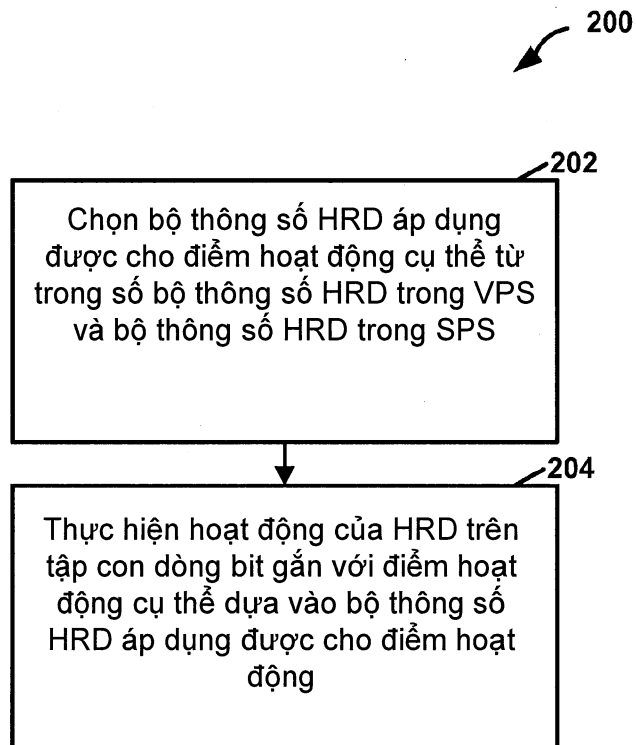


FIG. 4

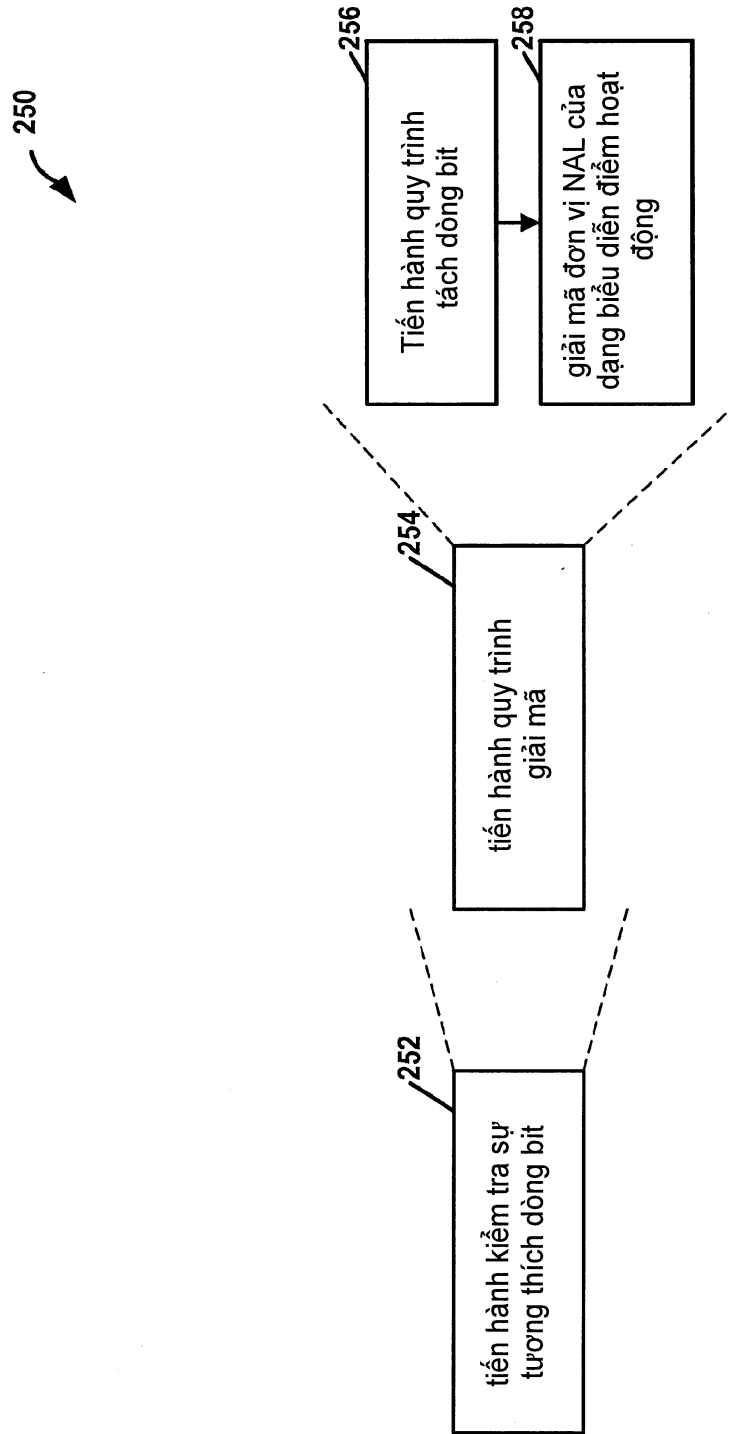


FIG. 5

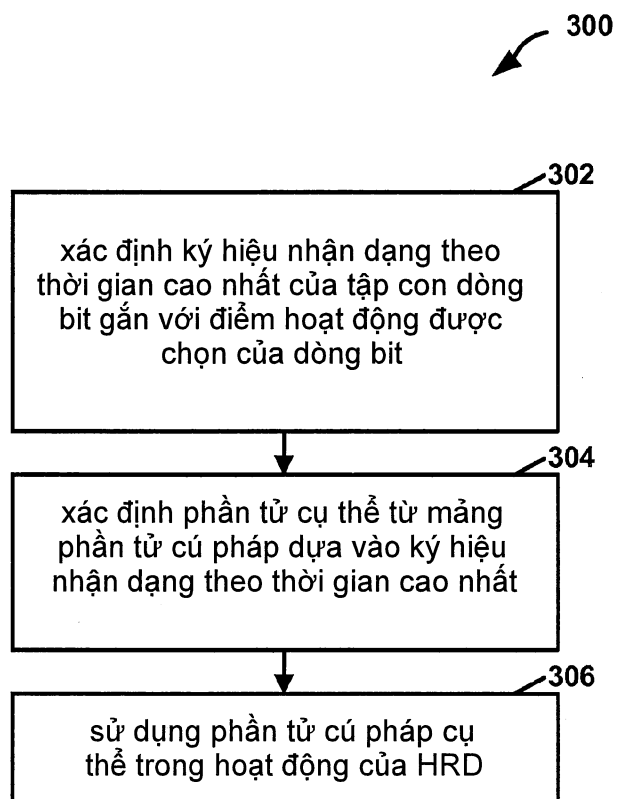


FIG. 6