



- (12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ
(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11) 
CỤC SỞ HỮU TRÍ TUỆ
1-0051289
(51)^{2006.01} **H04B 1/66**; H04N 7/46; H04N 7/32; (13) **B**
H04N 7/12; H04N 7/26

-
- (21) 1-2022-02571 (22) 17/09/2020
(86) PCT/US2020/051314 17/09/2020 (87) WO 2021/061494 01/04/2021
(30) 62/905,236 24/09/2019 US
(45) 25/09/2025 450 (43) 26/09/2022 414A
(73) HUAWEI TECHNOLOGIES CO., LTD. (CN)
Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong
518129, China
(72) WANG, Ye-Kui (US).
(74) Công ty TNHH Tầm nhìn và Liên danh (VISION & ASSOCIATES CO.LTD.)
-

- (54) PHƯƠNG PHÁP ĐƯỢC THỰC HIỆN BỞI BỘ GIẢI MÃ VÀ BỘ MÃ HÓA,
THIẾT BỊ GIẢI MÃ VÀ MÃ HÓA VÀ PHƯƠNG TIỆN LƯU TRỮ ĐƯỢC ĐƯỢC
BẰNG MÁY TÍNH

(21) 1-2022-02571

(57) Sáng chế đề cập đến cơ chế mã hóa video, mà bao gồm phương pháp giải mã được thực hiện bởi bộ giải mã, phương tiện đọc được bằng máy tính không tạm thời và thiết bị giải mã. Phương pháp này bao gồm bước nhận dòng bit bao gồm nhiều ảnh. Không kết xuất cờ các ảnh trước (NoOutputOfPriorPicsFlag) được thiết lập khi giá trị của chiều rộng ảnh tối đa trong các mẫu độ chói (PicWidthMaxInSamplesY) dùng cho đơn vị truy cập (access unit, AU) hiện thời khác với giá trị của PicWidthMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã. Bộ đệm ảnh được giải mã (decoded picture buffer, DPB) được làm trống mà không kết xuất các ảnh đã chứa dựa vào giá trị của NoOutputOfPriorPicsFlag. Ảnh hiện thời được giải mã và lưu trữ trong DPB. Ảnh hiện thời này được kết xuất từ DPB để hiển thị dưới dạng một phân chuỗi video được giải mã.

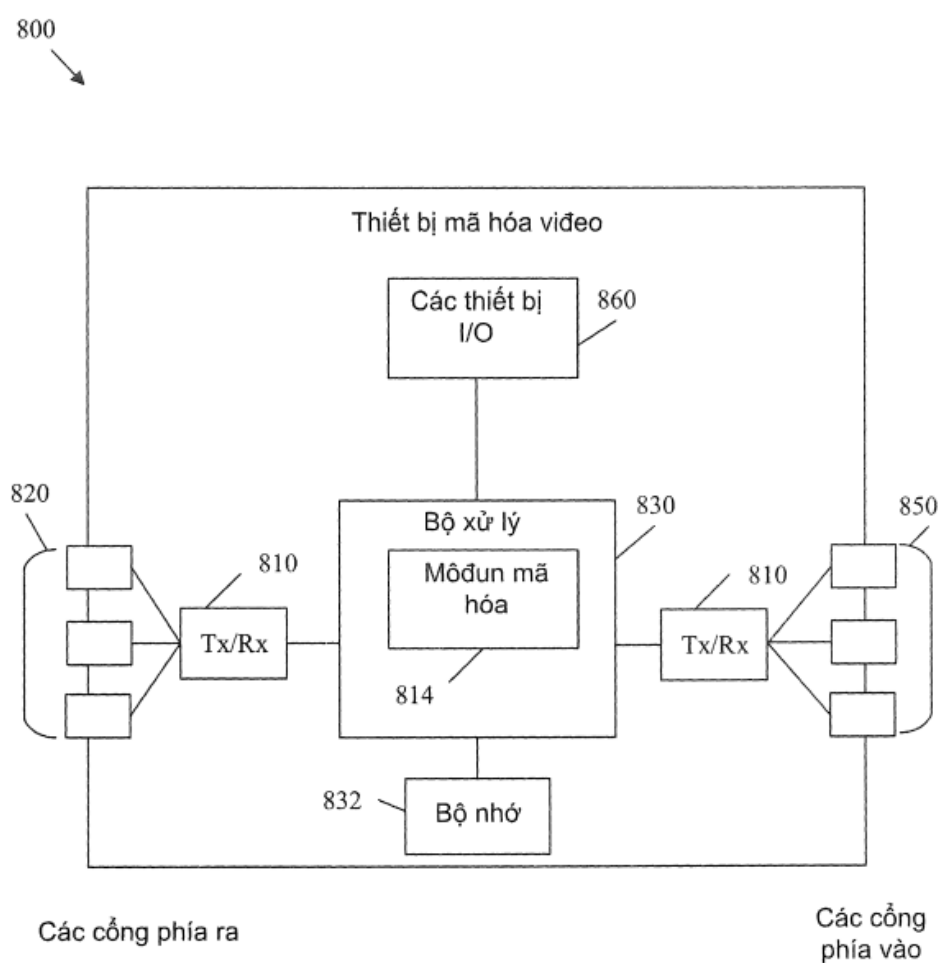


FIG. 8

Lĩnh vực kỹ thuật được đề cập

Nói chung, sáng chế đề cập đến lĩnh vực mã hóa video, và cụ thể là đề cập đến việc cải thiện các tham số báo hiệu để hỗ trợ việc mã hóa các dòng bit đa lớp.

Tình trạng kỹ thuật của sáng chế

Lượng dữ liệu video cần để mô tả ngay cả video tương đối ngắn có thể là đáng kể, điều này có thể dẫn đến các khó khăn khi dữ liệu cần được truyền theo luồng hoặc truyền thông theo cách khác qua mạng truyền thông với dung lượng băng thông giới hạn. Do đó, dữ liệu video thường được nén trước khi được truyền thông qua các mạng viễn thông hiện đại ngày nay. Kích thước của video cũng có thể là vấn đề khi video được lưu trữ trên thiết bị lưu trữ vì các tài nguyên bộ nhớ có thể bị giới hạn. Các thiết bị nén video thường sử dụng phần mềm và/hoặc phần cứng ở nguồn để mã hóa dữ liệu video trước khi truyền hoặc lưu trữ, nhờ đó giảm bớt lượng dữ liệu cần để biểu diễn các hình ảnh video số. Sau đó, dữ liệu đã nén nhận được ở đích bởi thiết bị giải nén video để giải mã dữ liệu video. Với các tài nguyên mạng giới hạn và nhu cầu ngày càng cao về chất lượng video, mong muốn các kỹ thuật nén và giải nén được cải thiện để cải thiện tỷ lệ nén mà không làm giảm chất lượng hình ảnh.

Bản chất kỹ thuật của sáng chế

Theo một phương án, sáng chế đề xuất phương pháp được thực hiện bởi bộ giải mã, phương pháp này bao gồm các bước: nhận, bởi bộ thu của bộ giải mã, dòng bit bao gồm các ảnh; thiết lập, bởi bộ xử lý, giá trị không kết xuất cờ các ảnh trước (NoOutputOfPriorPicsFlag) khi giá trị của chiều rộng ảnh tối đa trong các mẫu độ chói (PicWidthMaxInSamplesY) đối với đơn vị truy cập (access unit, AU) hiện thời khác với giá trị của PicWidthMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã; làm trống bộ đệm ảnh được giải mã (decoded picture buffer, DPB) mà không kết xuất các ảnh đã chứa dựa vào giá trị của NoOutputOfPriorPicsFlag; giải mã, bởi bộ xử lý, ảnh hiện thời; và lưu trữ, bởi DPB, ảnh hiện thời.

Các hệ thống mã hóa video có thể mã hóa chuỗi video thành nhiều lớp. Các lớp này có thể được trích xuất từ dòng bit được mã hóa và được truyền đến bộ giải mã để cho phép bộ giải mã hiển thị chuỗi video bằng cách sử dụng các đặc điểm khác nhau như mong muốn. Ví dụ, các ảnh có thể được truyền ở độ phân giải cao hơn/kích thước lớn hơn khi các điều kiện mạng là tối ưu và được truyền ở độ phân giải thấp hơn/kích thước thấp

hơn khi các điều kiện mạng xấu đi. Một vấn đề với phương án thực hiện này đó là DPB ở bộ giải mã có thể bị tràn trong quá trình thay đổi lớp, mà có thể gây ra trải nghiệm kém cho người dùng. NoOutputOfPriorPicsFlag có thể được sử dụng để giải quyết vấn đề này. NoOutputOfPriorPicsFlag có thể được thiết lập để khiến làm trống DPB để tạo ra chỗ cho các ảnh mới. Điều này có thể xảy ra ngay cả khi các ảnh đứng trước trong DPB được chỉ định để kết xuất. Trong một số hệ thống mã hóa video, NoOutputOfPriorPicsFlag được thiết lập dựa vào sự thay đổi kích thước ảnh mà được phát hiện do sự thay đổi về chiều rộng hoặc chiều cao giữa các ảnh liên tiếp. Sự thay đổi độ phân giải thích ứng (Adaptive resolution change, ARC) là cơ chế để hỗ trợ với các thay đổi độ phân giải. ARC có thể thay đổi động độ phân giải của các ảnh trong DPB để khớp với độ phân giải của các ảnh tiếp theo để cho phép các thay đổi độ phân giải xảy ra ở các ảnh khác với các ảnh IRAP. Việc sử dụng ARC có thể ngăn NoOutputOfPriorPicsFlag không hoạt động phù hợp. Cụ thể, ARC có thể thay đổi độ phân giải về không gian của ảnh đứng trước giống với ảnh hiện thời, mà ngăn việc kiểm tra đối với sự thay đổi về kích thước ảnh không hoạt động chính xác. Như vậy, ARC có thể dẫn đến tràn DPB bằng cách vô hiệu hóa một cách hiệu quả việc sử dụng NoOutputOfPriorPicsFlag.

Một khía cạnh của sáng chế đề xuất cơ chế để kích hoạt NoOutputOfPriorPicsFlag khi ARC được sử dụng. Ví dụ, bộ giải mã có thể kiểm tra đối với sự thay đổi về chiều cao ảnh tối đa và/hoặc chiều rộng ảnh tối đa thay vì kiểm tra đối với chiều cao và chiều rộng ảnh. Chiều cao ảnh tối đa và chiều rộng ảnh tối đa không bị thay đổi bởi ARC. Theo đó, NoOutputOfPriorPicsFlag có thể thiết lập thích hợp để ngăn sự tràn DPB do các thay đổi độ phân giải về không gian ngay cả khi ARC được sử dụng để làm cân đối các độ phân giải trong quá trình thay đổi lớp. Do đó, có thể tránh được các lỗi khác nhau. Quy trình tương tự cũng có thể được sử dụng bởi bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD) được vận hành ở bộ mã hóa khi kiểm tra dòng bit được mã hóa đối với tính phù hợp với các tiêu chuẩn. Kết quả là, chức năng của bộ mã hóa và bộ giải mã được tăng lên. Hơn nữa, hiệu quả mã hóa có thể được tăng lên, điều này giảm bớt mức sử dụng tài nguyên báo hiệu của bộ xử lý, bộ nhớ và/hoặc mạng ở cả bộ mã hóa và bộ giải mã.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước thiết lập, bởi bộ xử lý, giá trị của NoOutputOfPriorPicsFlag khi giá trị của chiều cao ảnh tối đa trong các mẫu độ chói (PicHeightMaxInSamplesY) dùng cho AU hiện thời khác với giá trị của PicHeightMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống trước khi giải mã ảnh hiện thời.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống sau khi phân tích cú pháp phần đầu lát của lát thứ nhất của ảnh hiện thời.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước giảm bớt biến lấp đầy DPB đối với mỗi ảnh được loại bỏ khi làm trống DPB.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước xây dựng, bởi bộ xử lý, danh sách ảnh tham chiếu và thực hiện việc đánh dấu danh sách ảnh tham chiếu đối với AU hiện thời trước khi thiết lập NoOutputOfPriorPicsFlag.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống khi ảnh đứng trước được đánh dấu là chưa được sử dụng để tham chiếu.

Theo một phương án, sáng chế đề xuất phương pháp được thực hiện bởi bộ mã hóa, phương pháp này bao gồm các bước: mã hóa, bởi bộ xử lý của bộ mã hóa, các ảnh thành dòng bit; kiểm tra, bởi HRD hoạt động trên bộ xử lý, dòng bit về tính phù hợp bằng cách: thiết lập, bởi bộ xử lý, giá trị của NoOutputOfPriorPicsFlag khi giá trị của PicWidthMaxInSamplesY dùng cho AU hiện thời khác với giá trị của PicWidthMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã; làm trống DPB mà không chọn ảnh đã chứa để kết xuất dựa vào giá trị của NoOutputOfPriorPicsFlag; giải mã, bởi bộ xử lý, ảnh hiện thời; lưu trữ, bởi DPB, ảnh hiện thời; và lưu trữ, bởi bộ nhớ được ghép nối với bộ xử lý, dòng bit để truyền thông về phía bộ giải mã.

Các hệ thống mã hóa video có thể mã hóa chuỗi video thành nhiều lớp. Các lớp này có thể được trích xuất từ dòng bit được mã hóa và được truyền đến bộ giải mã để cho phép bộ giải mã hiển thị chuỗi video bằng cách sử dụng các đặc điểm khác nhau như mong muốn. Ví dụ, các ảnh có thể được truyền ở độ phân giải cao hơn/kích thước lớn hơn khi các điều kiện mạng là tối ưu và được truyền ở độ phân giải thấp hơn/kích thước thấp hơn khi các điều kiện mạng xấu đi. Một vấn đề với phương án thực hiện này đó là DPB ở bộ giải mã có thể bị tràn trong quá trình thay đổi lớp, mà có thể gây ra trải nghiệm kém cho người dùng. NoOutputOfPriorPicsFlag có thể được sử dụng để giải quyết vấn đề này. NoOutputOfPriorPicsFlag có thể được thiết lập để khiến làm trống DPB để tạo ra chỗ cho các ảnh mới. Điều này có thể xảy ra ngay cả khi các ảnh đứng trước trong DPB được chỉ

định để kết xuất. Trong một số hệ thống mã hóa video, NoOutputOfPriorPicsFlag được thiết lập dựa vào sự thay đổi kích thước ảnh mà được phát hiện do sự thay đổi về chiều rộng hoặc chiều cao giữa các ảnh liên tiếp. ARC là cơ chế để hỗ trợ với các thay đổi độ phân giải. ARC có thể thay đổi động độ phân giải của các ảnh trong DPB để khớp với độ phân giải của các ảnh tiếp theo để cho phép các thay đổi độ phân giải xảy ra ở các ảnh khác với các ảnh IRAP. Việc sử dụng ARC có thể ngăn NoOutputOfPriorPicsFlag không hoạt động phù hợp. Cụ thể, ARC có thể thay đổi độ phân giải về không gian của ảnh đứng trước giống với ảnh hiện thời, mà ngăn việc kiểm tra đối với sự thay đổi về kích thước ảnh không hoạt động chính xác. Như vậy, ARC có thể dẫn đến tràn DPB bằng cách vô hiệu hóa một cách hiệu quả việc sử dụng NoOutputOfPriorPicsFlag.

Một khía cạnh của sáng chế đề xuất cơ chế để kích hoạt NoOutputOfPriorPicsFlag khi ARC được sử dụng. Ví dụ, bộ giải mã có thể kiểm tra đối với sự thay đổi về chiều cao ảnh tối đa và/hoặc chiều rộng ảnh tối đa thay vì kiểm tra đối với chiều cao và chiều rộng ảnh. Chiều cao ảnh tối đa và chiều rộng ảnh tối đa không bị thay đổi bởi ARC. Theo đó, NoOutputOfPriorPicsFlag có thể thiết lập thích hợp để ngăn sự tràn DPB do các thay đổi độ phân giải về không gian ngay cả khi ARC được sử dụng để làm cân đối các độ phân giải trong quá trình thay đổi lớp. Do đó, có thể tránh được các lỗi khác nhau. Quy trình tương tự có thể còn được sử dụng bởi bộ giải mã tham chiếu giả định (HRD) được vận hành ở bộ mã hóa khi kiểm tra dòng bit được mã hóa về tính phù hợp với các tiêu chuẩn. Kết quả là, chức năng của bộ mã hóa và bộ giải mã được tăng lên. Hơn nữa, hiệu quả mã hóa có thể được tăng lên, điều này giảm bớt mức sử dụng tài nguyên báo hiệu của bộ xử lý, bộ nhớ và/hoặc mạng ở cả bộ mã hóa và bộ giải mã.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước thiết lập, bởi HRD hoạt động trên bộ xử lý, giá trị của NoOutputOfPriorPicsFlag khi giá trị của PicHeightMaxInSamplesY dùng cho AU hiện thời khác với giá trị của PicHeightMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống trước khi giải mã ảnh hiện thời.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống sau khi phân tích cú pháp phần đầu lát của lát thứ nhất của ảnh hiện thời.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước giảm bớt, bởi HRD

hoạt động trên bộ xử lý, biến lấp đầy DPB đối với mỗi ảnh được loại bỏ khi làm trống DPB.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước xây dựng, bởi HRD hoạt động trên bộ xử lý, danh sách ảnh tham chiếu và thực hiện việc đánh dấu danh sách ảnh tham chiếu này.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống khi ảnh đứng trước được đánh dấu là chưa được sử dụng để tham chiếu.

Theo một phương án, sáng chế đề xuất thiết bị mã hóa video bao gồm: bộ xử lý, bộ thu được ghép nối với bộ xử lý, bộ nhớ được ghép nối với bộ xử lý, và bộ phát được ghép nối với bộ xử lý, trong đó bộ xử lý, bộ thu, bộ nhớ và bộ phát được tạo cấu hình để thực hiện phương pháp theo khía cạnh bất kỳ trong số các khía cạnh nêu trên.

Theo một phương án, sáng chế đề xuất phương tiện đọc được bằng máy tính không tạm thời bao gồm sản phẩm chương trình máy tính để được sử dụng bởi thiết bị mã hóa video, sản phẩm chương trình máy tính này bao gồm các lệnh thực thi được bằng máy tính được lưu trữ trên phương tiện đọc được bằng máy tính không tạm thời sao cho khi được thực thi bởi bộ xử lý thì khiến thiết bị mã hóa video thực hiện phương pháp theo khía cạnh bất kỳ trong số các khía cạnh nêu trên.

Theo một phương án, sáng chế đề xuất bộ giải mã bao gồm: phương tiện nhận để nhận dòng bit bao gồm các ảnh; phương tiện thiết lập để thiết lập giá trị của NoOutputOfPriorPicsFlag khi giá trị của PicWidthMaxInSamplesY dùng cho AU hiện thời khác với giá trị của PicWidthMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã; phương tiện làm trống để làm trống DPB mà không kết xuất các ảnh đã chứa dựa vào giá trị của NoOutputOfPriorPicsFlag; phương tiện giải mã để giải mã ảnh hiện thời; phương tiện lưu trữ để lưu trữ ảnh hiện thời trong DPB; và phương tiện kết xuất để kết xuất ảnh hiện thời từ DPB để hiển thị dưới dạng một phần chuỗi video được giải mã.

Các hệ thống mã hóa video có thể mã hóa chuỗi video thành nhiều lớp. Các lớp này có thể được trích xuất từ dòng bit được mã hóa và được truyền đến bộ giải mã để cho phép bộ giải mã hiển thị chuỗi video bằng cách sử dụng các đặc điểm khác nhau như mong muốn. Ví dụ, các ảnh có thể được truyền ở độ phân giải cao hơn/kích thước lớn hơn khi các điều kiện mạng là tối ưu và được truyền ở độ phân giải thấp hơn/kích thước thấp hơn khi các điều kiện mạng xấu đi. Một vấn đề với phương án thực hiện này đó là DPB ở bộ giải mã có thể bị tràn trong quá trình thay đổi lớp, mà có thể gây ra trải nghiệm kém cho người dùng. NoOutputOfPriorPicsFlag có thể được sử dụng để giải quyết vấn đề này.

NoOutputOfPriorPicsFlag có thể được thiết lập để khiến làm trống DPB để tạo ra chỗ cho các ảnh mới. Điều này có thể xảy ra ngay cả khi các ảnh đứng trước trong DPB được chỉ định để kết xuất. Trong một số hệ thống mã hóa video, NoOutputOfPriorPicsFlag được thiết lập dựa vào sự thay đổi kích thước ảnh mà được phát hiện do sự thay đổi về chiều rộng hoặc chiều cao giữa các ảnh liên tiếp. ARC là cơ chế để hỗ trợ với các thay đổi độ phân giải. ARC có thể thay đổi động độ phân giải của các ảnh trong DPB để khớp với độ phân giải của các ảnh tiếp theo để cho phép các thay đổi độ phân giải xảy ra ở các ảnh khác với các ảnh IRAP. Việc sử dụng ARC có thể ngăn NoOutputOfPriorPicsFlag không hoạt động phù hợp. Cụ thể, ARC có thể thay đổi độ phân giải về không gian của ảnh đứng trước giống với ảnh hiện thời, mà ngăn việc kiểm tra đối với sự thay đổi về kích thước ảnh không hoạt động chính xác. Như vậy, ARC có thể dẫn đến tràn DPB bằng cách vô hiệu hóa một cách hiệu quả việc sử dụng NoOutputOfPriorPicsFlag.

Một khía cạnh của sáng chế đề xuất cơ chế để kích hoạt NoOutputOfPriorPicsFlag khi ARC được sử dụng. Ví dụ, bộ giải mã có thể kiểm tra đối với sự thay đổi về chiều cao ảnh tối đa và/hoặc chiều rộng ảnh tối đa thay vì kiểm tra đối với chiều cao và chiều rộng ảnh. Chiều cao ảnh tối đa và chiều rộng ảnh tối đa không bị thay đổi bởi ARC. Theo đó, NoOutputOfPriorPicsFlag có thể thiết lập thích hợp để ngăn sự tràn DPB do các thay đổi độ phân giải về không gian ngay cả khi ARC được sử dụng để làm cân đối các độ phân giải trong quá trình thay đổi lớp. Do đó, có thể tránh được các lỗi khác nhau. Quy trình tương tự có thể còn được sử dụng bởi bộ giải mã tham chiếu giả định (HRD) được vận hành ở bộ mã hóa khi kiểm tra dòng bit được mã hóa về tính phù hợp với các tiêu chuẩn. Kết quả là, chức năng của bộ mã hóa và bộ giải mã được tăng lên. Hơn nữa, hiệu quả mã hóa có thể được tăng lên, điều này giảm bớt mức sử dụng tài nguyên báo hiệu của bộ xử lý, bộ nhớ và/hoặc mạng ở cả bộ mã hóa và bộ giải mã.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một phương án thực hiện khác về khía cạnh này đề xuất, trong đó bộ giải mã còn được tạo cấu hình để thực hiện phương pháp theo khía cạnh bất kỳ trong số các khía cạnh nêu trên.

Theo một phương án, sáng chế đề xuất bộ mã hóa bao gồm: phương tiện mã hóa để mã hóa các ảnh thành dòng bit; phương tiện HRD để kiểm tra dòng bit về tính phù hợp bằng cách: thiết lập giá trị của NoOutputOfPriorPicsFlag khi giá trị của PicWidthMaxInSamplesY dùng cho AU hiện thời khác với giá trị của PicWidthMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã; làm trống DPB mà không chọn ảnh đã chứa để kết xuất dựa vào giá trị của NoOutputOfPriorPicsFlag; giải mã ảnh hiện thời; và lưu trữ ảnh hiện thời trong DPB; và phương tiện lưu trữ để lưu trữ dòng bit để truyền thông về phía bộ giải mã.

Các hệ thống mã hóa video có thể mã hóa chuỗi video thành nhiều lớp. Các lớp này có thể được trích xuất từ dòng bit được mã hóa và được truyền đến bộ giải mã để cho phép bộ giải mã hiển thị chuỗi video bằng cách sử dụng các đặc điểm khác nhau như mong muốn. Ví dụ, các ảnh có thể được truyền ở độ phân giải cao hơn/kích thước lớn hơn khi các điều kiện mạng là tối ưu và được truyền ở độ phân giải thấp hơn/kích thước thấp hơn khi các điều kiện mạng xấu đi. Một vấn đề với phương án thực hiện này đó là DPB ở bộ giải mã có thể bị tràn trong quá trình thay đổi lớp, mà có thể gây ra trải nghiệm kém cho người dùng. NoOutputOfPriorPicsFlag có thể được sử dụng để giải quyết vấn đề này. NoOutputOfPriorPicsFlag có thể được thiết lập để khiến làm trống DPB để tạo ra chỗ cho các ảnh mới. Điều này có thể xảy ra ngay cả khi các ảnh đứng trước trong DPB được chỉ định để kết xuất. Trong một số hệ thống mã hóa video, NoOutputOfPriorPicsFlag được thiết lập dựa vào sự thay đổi kích thước ảnh mà được phát hiện do sự thay đổi về chiều rộng hoặc chiều cao giữa các ảnh liên tiếp. ARC là cơ chế để hỗ trợ với các thay đổi độ phân giải. ARC có thể thay đổi động độ phân giải của các ảnh trong DPB để khớp với độ phân giải của các ảnh tiếp theo để cho phép các thay đổi độ phân giải xảy ra ở các ảnh khác với các ảnh IRAP. Việc sử dụng ARC có thể ngăn NoOutputOfPriorPicsFlag không hoạt động phù hợp. Cụ thể, ARC có thể thay đổi độ phân giải về không gian của ảnh đứng trước giống với ảnh hiện thời, mà ngăn việc kiểm tra đối với sự thay đổi về kích thước ảnh không hoạt động chính xác. Như vậy, ARC có thể dẫn đến tràn DPB bằng cách vô hiệu hóa một cách hiệu quả việc sử dụng NoOutputOfPriorPicsFlag.

Một khía cạnh của sáng chế đề xuất cơ chế để kích hoạt NoOutputOfPriorPicsFlag khi ARC được sử dụng. Ví dụ, bộ giải mã có thể kiểm tra đối với sự thay đổi về chiều cao ảnh tối đa và/hoặc chiều rộng ảnh tối đa thay vì kiểm tra đối với chiều cao và chiều rộng ảnh. Chiều cao ảnh tối đa và chiều rộng ảnh tối đa không bị thay đổi bởi ARC. Theo đó, NoOutputOfPriorPicsFlag có thể thiết lập thích hợp để ngăn sự tràn DPB do các thay đổi độ phân giải về không gian ngay cả khi ARC được sử dụng để làm cân đối các độ phân giải trong quá trình thay đổi lớp. Do đó, có thể tránh được các lỗi khác nhau. Quy trình tương tự có thể còn được sử dụng bởi bộ giải mã tham chiếu giả định (HRD) được vận hành ở bộ mã hóa khi kiểm tra dòng bit được mã hóa về tính phù hợp với các tiêu chuẩn. Kết quả là, chức năng của bộ mã hóa và bộ giải mã được tăng lên. Hơn nữa, hiệu quả mã hóa có thể được tăng lên, điều này giảm bớt mức sử dụng tài nguyên báo hiệu của bộ xử lý, bộ nhớ và/hoặc mạng ở cả bộ mã hóa và bộ giải mã.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, một phương án thực hiện khác về khía cạnh này đề xuất, trong đó bộ mã hóa còn được tạo cấu hình để thực hiện phương pháp theo khía cạnh bất kỳ trong số các khía cạnh nêu trên.

Theo một phương án, sáng chế đề xuất phương pháp được thực hiện bởi bộ giải mã, phương pháp này bao gồm các bước: nhận, bởi bộ thu của bộ giải mã, dòng bit bao gồm các ảnh; làm trống DPB mà không kết xuất các ảnh đã chứa khi giá trị của `PicWidthMaxInSamplesY` dùng cho AU hiện thời khác với giá trị của `PicWidthMaxInSamplesY` dùng cho AU đứng trước theo thứ tự giải mã; giải mã, bởi bộ xử lý, ảnh hiện thời; và lưu trữ, bởi DPB, ảnh hiện thời.

Các hệ thống mã hóa video có thể mã hóa chuỗi video thành nhiều lớp. Các lớp này có thể được trích xuất từ dòng bit được mã hóa và được truyền đến bộ giải mã để cho phép bộ giải mã hiển thị chuỗi video bằng cách sử dụng các đặc điểm khác nhau như mong muốn. Ví dụ, các ảnh có thể được truyền ở độ phân giải cao hơn/kích thước lớn hơn khi các điều kiện mạng là tối ưu và được truyền ở độ phân giải thấp hơn/kích thước thấp hơn khi các điều kiện mạng xấu đi. Một vấn đề với phương án thực hiện này đó là DPB ở bộ giải mã có thể bị tràn trong quá trình thay đổi lớp, mà có thể gây ra trải nghiệm kém cho người dùng. `NoOutputOfPriorPicsFlag` có thể được sử dụng để giải quyết vấn đề này. `NoOutputOfPriorPicsFlag` có thể được thiết lập để khiến làm trống DPB để tạo ra chỗ cho các ảnh mới. Điều này có thể xảy ra ngay cả khi các ảnh đứng trước trong DPB được chỉ định để kết xuất. Trong một số hệ thống mã hóa video, `NoOutputOfPriorPicsFlag` được thiết lập dựa vào sự thay đổi kích thước ảnh mà được phát hiện do sự thay đổi về chiều rộng hoặc chiều cao giữa các ảnh liên tiếp. ARC là cơ chế để hỗ trợ với các thay đổi độ phân giải. ARC có thể thay đổi động độ phân giải của các ảnh trong DPB để khớp với độ phân giải của các ảnh tiếp theo để cho phép các thay đổi độ phân giải xảy ra ở các ảnh khác với các ảnh IRAP. Việc sử dụng ARC có thể ngăn `NoOutputOfPriorPicsFlag` không hoạt động phù hợp. Cụ thể, ARC có thể thay đổi độ phân giải về không gian của ảnh đứng trước giống với ảnh hiện thời, mà ngăn việc kiểm tra đối với sự thay đổi về kích thước ảnh không hoạt động chính xác. Như vậy, ARC có thể dẫn đến tràn DPB bằng cách vô hiệu hóa một cách hiệu quả việc sử dụng `NoOutputOfPriorPicsFlag`.

Một khía cạnh của sáng chế đề xuất cơ chế để kích hoạt `NoOutputOfPriorPicsFlag` khi ARC được sử dụng. Ví dụ, bộ giải mã có thể kiểm tra đối với sự thay đổi về chiều cao ảnh tối đa và/hoặc chiều rộng ảnh tối đa thay vì kiểm tra đối với chiều cao và chiều rộng ảnh. Chiều cao ảnh tối đa và chiều rộng ảnh tối đa không bị thay đổi bởi ARC. Theo đó, `NoOutputOfPriorPicsFlag` có thể thiết lập thích hợp để ngăn sự tràn DPB do các thay đổi độ phân giải về không gian ngay cả khi ARC được sử dụng để làm cân đối các độ phân giải trong quá trình thay đổi lớp. Do đó, có thể tránh được các lỗi khác nhau. Quy trình tương tự có thể còn được sử dụng bởi bộ giải mã tham chiếu giả định (HRD) được vận hành ở bộ mã hóa khi kiểm tra dòng bit được mã hóa về tính phù hợp với các tiêu chuẩn. Kết quả là, chức năng của bộ mã hóa và bộ giải mã được tăng lên. Hơn nữa, hiệu quả mã

hóa có thể được tăng lên, điều này giảm bớt mức sử dụng tài nguyên báo hiệu của bộ xử lý, bộ nhớ và/hoặc mạng ở cả bộ mã hóa và bộ giải mã.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước làm trống DPB mà không kết xuất các ảnh đã chứa khi giá trị của `PicHeightMaxInSamplesY` dùng cho AU hiện thời khác với giá trị của `PicHeightMaxInSamplesY` dùng cho AU đứng trước theo thứ tự giải mã.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống trước khi giải mã ảnh hiện thời.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống sau khi phân tích cú pháp phần đầu lát của lát thứ nhất của ảnh hiện thời.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước giảm bớt biên lấp đầy DPB đối với mỗi ảnh được loại bỏ khi làm trống DPB.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước xây dựng, bởi bộ xử lý, danh sách ảnh tham chiếu và thực hiện việc đánh dấu danh sách ảnh tham chiếu đối với AU hiện thời trước khi làm trống DPB.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống khi ảnh đứng trước được đánh dấu là chưa được sử dụng để tham chiếu.

Theo một phương án, sáng chế đề xuất phương pháp được thực hiện bởi bộ mã hóa, phương pháp này bao gồm các bước: mã hóa, bởi bộ xử lý của bộ mã hóa, các ảnh thành dòng bit; kiểm tra, bởi HRD hoạt động trên bộ xử lý, dòng bit về tính phù hợp bằng cách làm trống DPB mà không kết xuất các ảnh đã chứa khi giá trị của `PicWidthMaxInSamplesY` dùng cho AU hiện thời khác với giá trị của `PicWidthMaxInSamplesY` dùng cho AU đứng trước theo thứ tự giải mã; giải mã, bởi bộ xử lý, ảnh hiện thời; lưu trữ, bởi DPB, ảnh hiện thời; và lưu trữ, bởi bộ nhớ được ghép nối với bộ xử lý, dòng bit để truyền thông về phía bộ giải mã.

Các hệ thống mã hóa video có thể mã hóa chuỗi video thành nhiều lớp. Các lớp này có thể được trích xuất từ dòng bit được mã hóa và được truyền đến bộ giải mã để cho phép bộ giải mã hiển thị chuỗi video bằng cách sử dụng các đặc điểm khác nhau như

mong muốn. Ví dụ, các ảnh có thể được truyền ở độ phân giải cao hơn/kích thước lớn hơn khi các điều kiện mạng là tối ưu và được truyền ở độ phân giải thấp hơn/kích thước thấp hơn khi các điều kiện mạng xấu đi. Một vấn đề với phương án thực hiện này đó là DPB ở bộ giải mã có thể bị tràn trong quá trình thay đổi lớp, mà có thể gây ra trải nghiệm kém cho người dùng. NoOutputOfPriorPicsFlag có thể được sử dụng để giải quyết vấn đề này. NoOutputOfPriorPicsFlag có thể được thiết lập để khiến làm trống DPB để tạo ra chỗ cho các ảnh mới. Điều này có thể xảy ra ngay cả khi các ảnh đứng trước trong DPB được chỉ định để kết xuất. Trong một số hệ thống mã hóa video, NoOutputOfPriorPicsFlag được thiết lập dựa vào sự thay đổi kích thước ảnh mà được phát hiện do sự thay đổi về chiều rộng hoặc chiều cao giữa các ảnh liên tiếp. ARC là cơ chế để hỗ trợ với các thay đổi độ phân giải. ARC có thể thay đổi động độ phân giải của các ảnh trong DPB để khớp với độ phân giải của các ảnh tiếp theo để cho phép các thay đổi độ phân giải xảy ra ở các ảnh khác với các ảnh IRAP. Việc sử dụng ARC có thể ngăn NoOutputOfPriorPicsFlag không hoạt động phù hợp. Cụ thể, ARC có thể thay đổi độ phân giải về không gian của ảnh đứng trước giống với ảnh hiện thời, mà ngăn việc kiểm tra đối với sự thay đổi về kích thước ảnh không hoạt động chính xác. Như vậy, ARC có thể dẫn đến tràn DPB bằng cách vô hiệu hóa một cách hiệu quả việc sử dụng NoOutputOfPriorPicsFlag.

Một khía cạnh của sáng chế đề xuất cơ chế để kích hoạt NoOutputOfPriorPicsFlag khi ARC được sử dụng. Ví dụ, bộ giải mã có thể kiểm tra đối với sự thay đổi về chiều cao ảnh tối đa và/hoặc chiều rộng ảnh tối đa thay vì kiểm tra đối với chiều cao và chiều rộng ảnh. Chiều cao ảnh tối đa và chiều rộng ảnh tối đa không bị thay đổi bởi ARC. Theo đó, NoOutputOfPriorPicsFlag có thể thiết lập thích hợp để ngăn sự tràn DPB do các thay đổi độ phân giải về không gian ngay cả khi ARC được sử dụng để làm cân đối các độ phân giải trong quá trình thay đổi lớp. Do đó, có thể tránh được các lỗi khác nhau. Quy trình tương tự có thể còn được sử dụng bởi bộ giải mã tham chiếu giả định (HRD) được vận hành ở bộ mã hóa khi kiểm tra dòng bit được mã hóa về tính phù hợp với các tiêu chuẩn. Kết quả là, chức năng của bộ mã hóa và bộ giải mã được tăng lên. Hơn nữa, hiệu quả mã hóa có thể được tăng lên, điều này giảm bớt mức sử dụng tài nguyên báo hiệu của bộ xử lý, bộ nhớ và/hoặc mạng ở cả bộ mã hóa và bộ giải mã.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước làm trống DPB mà không kết xuất các ảnh đã chứa khi giá trị của PicHeightMaxInSamplesY dùng cho AU hiện thời khác với giá trị của PicHeightMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống trước khi giải mã ảnh hiện thời.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống sau khi phân tích cú pháp phần đầu lát của lát thứ nhất của ảnh hiện thời.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước giảm bớt biến lấp đầy DPB đối với mỗi ảnh được loại bỏ khi làm trống DPB.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, còn bao gồm bước xây dựng, bởi bộ xử lý, danh sách ảnh tham chiếu và thực hiện việc đánh dấu danh sách ảnh tham chiếu đối với AU hiện thời trước khi làm trống DPB.

Tùy chọn, theo khía cạnh bất kỳ trong số các khía cạnh nêu trên, sáng chế đề xuất một phương án thực hiện khác của khía cạnh này, trong đó DPB được làm trống khi ảnh đứng trước được đánh dấu là chưa được sử dụng để tham chiếu.

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

Các dấu hiệu này và các dấu hiệu khác sẽ được hiểu rõ ràng hơn nhờ phần mô tả chi tiết sau kết hợp với các hình vẽ kèm theo và các điểm yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Để hiểu toàn diện hơn về sáng chế, hiện có thể thực hiện tham chiếu đến phần mô tả ngắn gọn sau, kết hợp với các hình vẽ kèm theo và phần mô tả chi tiết, trong đó các số chỉ dẫn giống nhau thể hiện các bộ phận giống nhau.

Fig.1 là lưu đồ của phương pháp làm ví dụ để mã hóa tín hiệu video.

Fig.2 là sơ đồ sơ lược của hệ thống mã hóa và giải mã (mã hóa-giải mã) làm ví dụ để mã hóa video.

Fig.3 là sơ đồ sơ lược minh họa bộ mã hóa video làm ví dụ.

Fig.4 là sơ đồ sơ lược minh họa bộ giải mã video làm ví dụ.

Fig.5 là sơ đồ sơ lược minh họa bộ giải mã tham chiếu giả định (HRD) làm ví dụ.

Fig.6 là sơ đồ sơ lược minh họa chuỗi video đa lớp làm ví dụ.

Fig.7 là sơ đồ sơ lược minh họa dòng bit làm ví dụ.

Fig.8 là sơ đồ sơ lược của thiết bị mã hóa video làm ví dụ.

Fig.9 là lưu đồ của phương pháp làm ví dụ để mã hóa chuỗi video thành dòng bit để hỗ trợ việc sử dụng không kết xuất cờ các ảnh đứng trước (NoOutputOfPriorPicsFlag) kết hợp với sự thay đổi độ phân giải thích ứng (ARC).

Fig.10 là lưu đồ của phương pháp làm ví dụ để giải mã chuỗi video từ dòng bit trong khi sử dụng NoOutputOfPriorPicsFlag kết hợp với ARC.

Fig.11 là sơ đồ sơ lược của hệ thống làm ví dụ để mã hóa chuỗi video thành dòng bit trong khi sử dụng NoOutputOfPriorPicsFlag kết hợp với ARC.

Mô tả chi tiết sáng chế

Ngay từ đầu, cần hiểu rằng mặc dù phương án thực hiện minh họa của một hoặc nhiều phương án được đề xuất dưới đây, nhưng các hệ thống và/hoặc phương pháp được bộc lộ có thể được thực hiện bằng cách sử dụng một số kỹ thuật bất kỳ, cho dù hiện đã biết đến hoặc đang tồn tại. Sáng chế không nên bị giới hạn ở các phương án thực hiện minh họa, các hình vẽ và các kỹ thuật được minh họa dưới đây, bao gồm các thiết kế và các phương án thực hiện làm ví dụ được minh họa và mô tả trong bản mô tả này, nhưng có thể được cải biến nằm trong phạm vi của các điểm yêu cầu bảo hộ kèm theo cùng với phạm vi đầy đủ của các phần tương ứng của điểm yêu cầu bảo hộ này.

Các thuật ngữ sau được định nghĩa như sau trừ khi được sử dụng trong ngữ cảnh trái ngược trong bản mô tả này. Cụ thể, các định nghĩa sau nhằm cung cấp thêm sự rõ ràng đối với sáng chế. Tuy nhiên, các thuật ngữ có thể được mô tả khác nhau trong các ngữ cảnh khác nhau. Theo đó, các định nghĩa sau nên được coi là bổ sung và không nên được coi là làm giới hạn các định nghĩa khác bất kỳ của các phần mô tả được cung cấp cho các thuật ngữ như vậy trong bản mô tả này.

Dòng bit là chuỗi gồm các bit bao gồm dữ liệu video mà được nén để truyền giữa bộ mã hóa và bộ giải mã. Bộ mã hóa là thiết bị mà được tạo cấu hình để sử dụng các quy trình mã hóa để nén dữ liệu video thành dòng bit. Bộ giải mã là thiết bị mà được tạo cấu hình để sử dụng các quy trình giải mã để khôi phục dữ liệu video từ dòng bit để hiển thị. Ảnh là mảng các mẫu độ chói và/hoặc mảng các mẫu sắc độ để tạo ra khung hoặc trường của ảnh này. Mẫu độ chói là giá trị độ sáng của điểm/điểm ảnh trong ảnh. Chiều rộng ảnh là khoảng cách nằm ngang giữa biên bên trái của ảnh và biên bên phải của ảnh như được đo trong các mẫu độ chói. Chiều cao ảnh là khoảng cách thẳng đứng giữa biên trên cùng của ảnh và biên dưới cùng của ảnh như được đo trong các mẫu độ chói. Lát là số nguyên của các ô hoàn chỉnh hoặc số nguyên của các hàng đơn vị cây mã hóa (Coding Tree Unit,

CTU) hoàn chỉnh (ví dụ, nằm trong ô) của ảnh được chứa riêng trong một đơn vị lớp trừu tượng hóa mạng (network abstraction layer, NAL) duy nhất. Ảnh mà đang được mã hóa hoặc giải mã có thể được gọi là ảnh hiện thời để rõ ràng cho việc thảo luận. Ảnh được mã hóa là biểu diễn được mã hóa của ảnh bao gồm các đơn vị NAL loại lớp mã hóa video (video coding layer, VCL) với giá trị cụ thể của mã nhận dạng lớp phần đầu của đơn vị NAL (nuh_layer_id) nằm trong đơn vị truy cập (AU) và chứa tất cả các đơn vị cây mã hóa (CTU) của ảnh. Phân tích cú pháp là quy trình phân tích chuỗi dữ liệu để thu các thành phần dữ liệu logic. Ảnh được giải mã là ảnh được tạo ra bằng cách áp dụng quy trình giải mã cho ảnh được mã hóa.

AU là tập hợp ảnh được mã hóa mà có trong các lớp khác nhau và được kết hợp với cùng thời gian để kết xuất từ bộ đệm ảnh được giải mã (DPB). AU hiện thời là AU mà tương ứng với thời gian kết xuất cụ thể trong quy trình giải mã và/hoặc quy trình kiểm tra tính phù hợp HRD. AU đứng trước là AU mà xuất hiện trước (ví dụ, đứng ngay trước) AU hiện thời theo thứ tự giải mã. Thứ tự giải mã là thứ tự trong đó các phần tử cú pháp được xử lý bằng quy trình giải mã. Đơn vị NAL là cấu trúc cú pháp chứa dữ liệu dưới dạng phụ tải chuỗi byte thô (Raw Byte Sequence Payload, RBSP), chỉ báo của kiểu dữ liệu, và được đặt rải rác như mong muốn với các byte chống giả lập. Đơn vị VCL NAL là đơn vị NAL được mã hóa để chứa dữ liệu video, chẳng hạn như lát được mã hóa của ảnh. Đơn vị NAL không phải VCL là đơn vị NAL mà chứa dữ liệu không phải video chẳng hạn như cú pháp và/hoặc các tham số để hỗ trợ bước giải mã dữ liệu video, hiệu suất của việc kiểm tra tính phù hợp hoặc các hoạt động khác. Lớp là tập hợp đơn vị VCL NAL để chia sẻ đặc điểm được chỉ định (ví dụ, độ phân giải, tốc độ khung, kích thước hình ảnh chung, v.v.) như được biểu thị bởi ID lớp và các đơn vị NAL không phải VCL liên kết.

DPB là bộ đệm lưu giữ các ảnh được giải mã để tham chiếu, sắp xếp lại thứ tự đầu ra hoặc độ trễ đầu ra. Biến lấp đầy DPB là phần tử dữ liệu mà biểu thị số lượng ảnh được lưu trữ trong DPB. Việc giảm bớt là giảm về giá trị. Danh sách ảnh tham chiếu là danh sách ảnh tham chiếu được sử dụng để dự đoán liên ảnh. Việc đánh dấu danh sách ảnh tham chiếu là quy trình đánh dấu ảnh được giải mã trong DPB với trạng thái tham chiếu, bao gồm chưa được sử dụng để tham chiếu, được sử dụng để tham chiếu ngắn hạn, và được sử dụng để tham chiếu dài hạn, dựa vào danh sách ảnh tham chiếu đã xây dựng đối với lát hiện thời. Chưa được sử dụng để tham chiếu là trạng thái tham chiếu biểu thị ảnh được giải mã chưa được sử dụng làm ảnh tham chiếu đối với các ảnh khác bất kỳ cần được giải mã. Không kết xuất cờ các ảnh đứng trước (NoOutputOfPriorPicsFlag) là biến thu được mà chỉ định khi DPB cần được làm trống mà không kết xuất các ảnh có trong DPB. Chiều rộng ảnh tối đa trong các mẫu độ chói (PicWidthMaxInSamplesY) là biến thu được mà biểu thị chiều rộng tối đa, trong các đơn vị của các mẫu độ chói, của mỗi

ảnh được giải mã trong chuỗi và/hoặc lớp tương ứng. Chiều cao ảnh tối đa trong các mẫu độ chói (PicHeightMaxInSamplesY) là biến thu được mà biểu thị chiều cao tối đa, trong các đơn vị của các mẫu độ chói, của mỗi ảnh được giải mã trong chuỗi và/hoặc lớp tương ứng.

Bộ giải mã tham chiếu giả định (HRD) là mô hình bộ giải mã hoạt động trên bộ mã hóa để kiểm tra độ biến đổi của các dòng bit được tạo ra bằng quy trình mã hóa để xác minh tính phù hợp với các ràng buộc đã chỉ định. Kiểm tra tính phù hợp của dòng bit là kiểm tra để xác định xem dòng bit được mã hóa có tuân thủ tiêu chuẩn hay không, chẳng hạn như mã hóa video đa năng (Versatile Video Coding, VVC). Tập hợp tham số ảnh (picture parameter set, PPS) là cấu trúc cú pháp chứa các phần tử cú pháp để áp dụng cho toàn bộ các ảnh được mã hóa như được xác định bởi phần tử cú pháp được tìm thấy trong mỗi phần đầu ảnh. Phần đầu ảnh này là cấu trúc cú pháp chứa các phần tử cú pháp để áp dụng cho tất cả các lát của ảnh được mã hóa. Phần đầu lát là một phần của lát được mã hóa chứa các phần tử dữ liệu thuộc về tất cả các ô hoặc hàng CTU nằm trong ô được biểu diễn trong lát. Chuỗi video được mã hóa là tập hợp gồm một hoặc nhiều ảnh được mã hóa. Chuỗi video được giải mã là tập hợp gồm một hoặc nhiều ảnh được mã hóa.

Các từ viết tắt sau được sử dụng ở đây, đơn vị truy cập (AU), khối cây mã hóa (Coding Tree Block, CTB), đơn vị cây mã hóa (CTU), đơn vị mã hóa (Coding Unit, CU), chuỗi video lớp được mã hóa (Coded Layer Video Sequence, CLVS), bắt đầu chuỗi video lớp được mã hóa (Coded Layer Video Sequence Start, CLVSS), chuỗi video được mã hóa (Coded Video Sequence, CVS), bắt đầu chuỗi video được mã hóa (Coded Video Sequence Start, CVSS), nhóm chuyên gia video chung (Joint Video Experts Team, JVET), bộ giải mã tham chiếu giả định (HRD), tập hợp ô ràng buộc chuyển động (Motion Constrained Tile Set, MCTS), đơn vị truyền tối đa (Maximum Transfer Unit, MTU), lớp trừu tượng hóa mạng (NAL), tập hợp lớp đầu ra (Output Layer Set, OLS), số đếm thứ tự ảnh (Picture Order Count, POC), điểm truy cập ngẫu nhiên (Random Access Point, RAP), phụ tải chuỗi byte thô (RBSP), tập hợp tham số chuỗi (Sequence Parameter Set, SPS), tập hợp tham số video (Video Parameter Set, VPS), mã hóa video đa năng (VVC).

Nhiều kỹ thuật nén video có thể được sử dụng để giảm bớt kích thước của các tệp video với sự mất mát dữ liệu tối thiểu. Ví dụ, các kỹ thuật nén video có thể bao gồm bước thực hiện dự đoán theo không gian (ví dụ, nội ảnh) và/hoặc dự đoán theo thời gian (ví dụ, liên ảnh) để giảm hoặc loại bỏ phần dư dữ liệu trong các chuỗi video. Đối với bước mã hóa video dựa trên khối, lát video (ví dụ, ảnh video hoặc một phần ảnh video) có thể được phân vùng thành các khối video, mà có thể còn được gọi là các khối cây, các khối cây mã hóa (CTB), các đơn vị cây mã hóa (CTU), các đơn vị mã hóa (CU) và/hoặc các nút mã hóa. Các khối video trong lát (I) được mã hóa nội ảnh của ảnh được mã hóa bằng cách sử

dụng phép dự đoán theo không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng ảnh. Các khối video trong lát (P) dự đoán một chiều hoặc (B) dự đoán hai chiều được mã hóa liên ảnh của ảnh có thể được mã hóa bằng cách sử dụng phép dự đoán theo không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng ảnh hoặc dự đoán theo thời gian đối với các mẫu tham chiếu trong các ảnh tham chiếu khác. Các ảnh có thể được gọi là các khung và/hoặc hình ảnh, và các ảnh tham chiếu có thể được gọi là các khung tham chiếu và/hoặc các ảnh tham chiếu. Phép dự đoán theo không gian hoặc thời gian dẫn đến khối dự đoán biểu diễn khối hình ảnh. Dữ liệu dư biểu diễn các chênh lệch điểm ảnh giữa khối hình ảnh gốc và khối dự đoán. Theo đó, khối được mã hóa liên ảnh được mã hóa theo vectơ chuyển động mà chỉ đến khối của các mẫu tham chiếu tạo thành khối dự đoán và dữ liệu dư biểu thị sự khác biệt giữa khối được mã hóa và khối dự đoán. Khối được mã hóa nội ảnh được mã hóa theo chế độ mã hóa nội ảnh và dữ liệu dư. Để nén thêm, dữ liệu dư có thể được biến đổi từ miền điểm ảnh thành miền biến đổi. Điều này dẫn đến các hệ số biến đổi dư, mà có thể được lượng tử hóa. Các hệ số biến đổi được lượng tử hóa ban đầu có thể được sắp xếp trong mảng hai chiều. Các hệ số biến đổi được lượng tử hóa có thể được quét để tạo ra vectơ một chiều của các hệ số biến đổi. Bước mã hóa entropy có thể được áp dụng để đạt được độ nén nhiều hơn. Các kỹ thuật nén video này sẽ được thảo luận một cách chi tiết hơn dưới đây.

Để đảm bảo video được mã hóa có thể được giải mã một cách chính xác, video được mã hóa và giải mã theo các tiêu chuẩn mã hóa video tương ứng. Các tiêu chuẩn mã hóa video này bao gồm lĩnh vực tiêu chuẩn viễn thông thuộc liên minh viễn thông quốc tế (International Telecommunication Union, ITU) (ITU-T) H.261, nhóm chuyên gia ảnh động (Motion Picture Experts Group, MPEG)-1 thuộc tổ chức tiêu chuẩn hóa quốc tế/ủy ban kỹ thuật điện quốc tế (International Organization for Standardization/International Electrotechnical Commission, ISO/IEC) Phần 2, ITU-T H.262 hoặc ISO/IEC MPEG-2 Phần 2, ITU-T H.263, ISO/IEC MPEG-4 Phần 2, mã hóa video nâng cao (Advanced Video Coding, AVC), còn được biết đến là ITU-T H.264 hoặc ISO/IEC MPEG-4 Phần 10 và mã hóa video hiệu quả cao (High Efficiency Video Coding, HEVC), còn được biết đến là ITU-T H.265 hoặc MPEG-H Phần 2. AVC bao gồm các phần mở rộng chẳng hạn như mã hóa video có khả năng thay đổi (Scalable Video Coding, SVC), mã hóa video đa khung nhìn (Multiview Video Coding, MVC) và mã hóa video đa khung nhìn cộng độ sâu (Multiview Video Coding plus Depth, MVC+D) và AVC ba chiều (three dimensional, 3D) (three dimensional AVC, 3D-AVC). HEVC bao gồm các phần mở rộng chẳng hạn như HEVC có khả năng thay đổi (Scalable HEVC, SHVC), HEVC đa khung nhìn (Multiview HEVC, MV-HEVC) và 3D HEVC (3D-HEVC). Nhóm chuyên gia video chung (JVET) của ITU-T và ISO/IEC đã bắt đầu phát triển tiêu chuẩn mã hóa video được

gọi là mã hóa video đa năng (VVC). VVC có trong bản thảo làm việc (Working Draft, WD), bao gồm JVET-O2001-v14.

Các hệ thống mã hóa video có thể mã hóa chuỗi video thành nhiều lớp. Các lớp này có thể được trích xuất từ dòng bit được mã hóa và được truyền đến bộ giải mã để cho phép bộ giải mã hiển thị chuỗi video bằng cách sử dụng các đặc điểm khác nhau như mong muốn. Ví dụ, các ảnh có thể được truyền ở độ phân giải cao hơn/kích thước lớn hơn khi các điều kiện mạng là tối ưu và được truyền ở độ phân giải thấp hơn/kích thước thấp hơn khi các điều kiện mạng xấu đi. Một vấn đề với phương án thực hiện này đó là bộ đệm ảnh được giải mã (DPB) ở bộ giải mã có thể bị tràn trong quá trình thay đổi lớp, mà có thể gây ra trải nghiệm kém cho người dùng. Không kết xuất cờ các ảnh trước (NoOutputOfPriorPicsFlag) có thể được sử dụng để giải quyết vấn đề này. NoOutputOfPriorPicsFlag có thể được thiết lập để khiến làm trống DPB để tạo ra chỗ cho các ảnh mới. Điều này có thể xảy ra ngay cả khi các ảnh đứng trước trong DPB được chỉ định để kết xuất. Trong một số hệ thống mã hóa video, NoOutputOfPriorPicsFlag được thiết lập dựa vào sự thay đổi kích thước ảnh mà được phát hiện do sự thay đổi về chiều rộng hoặc chiều cao của các ảnh. Sự thay đổi độ phân giải thích ứng (ARC) là cơ chế để hỗ trợ với các thay đổi độ phân giải. ARC có thể thay đổi động độ phân giải của các ảnh trong DPB để khớp với các ảnh tiếp theo để cho phép các thay đổi độ phân giải xảy ra ở các ảnh khác với các ảnh điểm truy cập ngẫu nhiên nội ảnh (IRAP). Việc sử dụng ARC có thể ngăn NoOutputOfPriorPicsFlag không hoạt động phù hợp. Cụ thể, ARC có thể thay đổi độ phân giải về không gian của ảnh đứng trước giống với ảnh hiện thời, mà ngăn việc kiểm tra đối với sự thay đổi về kích thước ảnh không hoạt động chính xác. Như vậy, ARC có thể dẫn đến tràn DPB bằng cách vô hiệu hóa một cách hiệu quả việc sử dụng NoOutputOfPriorPicsFlag.

Được bộc lộ trong bản mô tả này là cơ chế để kích hoạt NoOutputOfPriorPicsFlag khi ARC được sử dụng. Ví dụ, bộ giải mã có thể kiểm tra đối với sự thay đổi về chiều cao ảnh tối đa và chiều rộng ảnh tối đa thay vì kiểm tra đối với chiều cao và chiều rộng ảnh. Chiều cao ảnh tối đa và chiều rộng ảnh tối đa không bị thay đổi bởi ARC. Theo đó, NoOutputOfPriorPicsFlag có thể thiết lập thích hợp để ngăn sự tràn DPB do các thay đổi độ phân giải về không gian ngay cả khi ARC được sử dụng để làm cân đối các độ phân giải trong quá trình thay đổi lớp. Do đó, có thể tránh được các lỗi khác nhau. Quy trình tương tự có thể còn được sử dụng bởi bộ giải mã tham chiếu giả định (HRD) được vận hành ở bộ mã hóa khi kiểm tra dòng bit được mã hóa về tính phù hợp với các tiêu chuẩn. Kết quả là, chức năng của bộ mã hóa và bộ giải mã được tăng lên. Hơn nữa, hiệu quả mã hóa có thể được tăng lên, điều này giảm bớt mức sử dụng tài nguyên báo hiệu của bộ xử lý, bộ nhớ và/hoặc mạng ở cả bộ mã hóa và bộ giải mã.

Fig.1 là lưu đồ của phương pháp hoạt động là ví dụ 100 để mã hóa tín hiệu video. Cụ thể, tín hiệu video được mã hóa ở bộ mã hóa. Quy trình mã hóa nén tín hiệu video bằng cách sử dụng các cơ chế khác nhau để giảm kích thước tệp video. Kích thước tệp nhỏ hơn cho phép tệp video được nén được truyền về phía người dùng, trong khi giảm tổng phí băng thông liên quan. Sau đó, bộ giải mã giải mã tệp video được nén để khôi phục tín hiệu video gốc để hiển thị đến người dùng cuối. Quy trình giải mã thường phản ánh quy trình mã hóa để cho phép bộ giải mã khôi phục tín hiệu video một cách nhất quán.

Ở bước 101, tín hiệu video được nhập vào bộ mã hóa. Ví dụ, tín hiệu video có thể là tệp video không được nén được lưu trữ trong bộ nhớ. Ví dụ khác là, tệp video có thể được thu thập bởi thiết bị thu thập video, chẳng hạn như camera video, và được mã hóa để hỗ trợ việc tạo dòng trực tiếp video. Tệp video có thể bao gồm cả thành phần âm thanh và thành phần video. Thành phần video chứa chuỗi khung hình ảnh mà, khi được xem theo trình tự, tạo ra cảm giác chuyển động trực quan. Các khung chứa các điểm ảnh mà được biểu diễn bằng ánh sáng, ở đây được gọi là các thành phần độ chói (hoặc các mẫu độ chói), và màu sắc, mà được gọi là các thành phần sắc độ (hoặc các mẫu màu). Trong một số ví dụ, các khung này có thể còn chứa các giá trị độ sâu để hỗ trợ việc xem ba chiều.

Ở bước 103, video được phân vùng thành các khối. Bước phân vùng bao gồm bước chia nhỏ các điểm ảnh trong mỗi khung thành các khối hình vuông và/hoặc hình chữ nhật để nén. Ví dụ, trong mã hóa video hiệu quả cao (HEVC) (còn được biết đến là H.265 và MPEG-H Phần 2), khung trước tiên có thể được chia thành các đơn vị cây mã hóa (CTU), mà là các khối có kích thước định trước (ví dụ, 64 điểm ảnh x 64 điểm ảnh). Các CTU chứa cả mẫu độ chói và mẫu sắc độ. Các cây mã hóa có thể được sử dụng để chia các CTU thành các khối và sau đó chia nhỏ theo cách đệ quy các khối cho đến khi đạt được các cấu hình mà hỗ trợ thêm việc mã hóa. Ví dụ, các thành phần độ chói của khung có thể được chia nhỏ cho đến khi các khối riêng lẻ chứa các giá trị độ sáng tương đối đồng nhất. Ngoài ra, các thành phần sắc độ của khung có thể được chia nhỏ cho đến khi các khối riêng lẻ chứa các giá trị màu tương đối đồng nhất. Theo đó, các cơ chế phân vùng thay đổi tùy thuộc vào nội dung của các khung video.

Ở bước 105, các cơ chế nén khác nhau được sử dụng để nén các khối hình ảnh được phân vùng ở bước 103. Ví dụ, phép dự đoán liên ảnh và/hoặc dự đoán nội ảnh có thể được sử dụng. Phép dự đoán liên ảnh được thiết kế để tận dụng lợi thế của thực tế là các đối tượng trong cảnh chung có xu hướng xuất hiện trong các khung tiếp theo. Theo đó, khối mô tả đối tượng trong khung tham chiếu không cần được mô tả lặp lại trong các khung liền kề. Cụ thể, đối tượng, chẳng hạn như bàn, có thể giữ nguyên ở vị trí không đổi trên nhiều khung. Do đó bảng được mô tả một lần và các khung liền kề có thể tham chiếu trở lại khung tham chiếu. Các cơ chế so khớp hình mẫu có thể được sử dụng để so khớp

các đối tượng trên nhiều khung. Ngoài ra, các đối tượng di chuyển có thể được biểu diễn trên nhiều khung, ví dụ, do chuyển động của đối tượng hoặc chuyển động của camera. Là ví dụ cụ thể, video có thể thể hiện xe ô tô mà di chuyển trên màn hình trên nhiều khung. Các vectơ chuyển động có thể được sử dụng để mô tả chuyển động này. Vectơ chuyển động là vectơ hai chiều mà tạo ra độ lệch từ các tọa độ của đối tượng trong khung đến các tọa độ của đối tượng trong khung tham chiếu. Như vậy, phép dự đoán liên ảnh có thể mã hóa khối hình ảnh trong khung hiện thời dưới dạng tập hợp vectơ chuyển động biểu thị độ lệch từ khối tương ứng trong khung tham chiếu.

Phép dự đoán nội ảnh mã hóa các khối trong khung chung. Phép dự đoán nội ảnh tận dụng lợi thế của thực tế là các thành phần độ chói và sắc độ có xu hướng tạo cụm trong khung. Ví dụ, mảng màu xanh lá cây trong một phần cây có xu hướng được định vị liền kề với các mảng màu xanh lá cây tương tự. Phép dự đoán nội ảnh sử dụng nhiều chế độ dự đoán có hướng (ví dụ, ba mươi ba trong HEVC), chế độ phẳng, và chế độ dòng một chiều (direct current, DC). Các chế độ có hướng biểu thị rằng khối hiện thời tương tự/giống với các mẫu của các khối lân cận theo hướng tương ứng. Chế độ phẳng biểu thị rằng chuỗi các khối dọc theo hàng/cột (ví dụ, mặt phẳng) có thể được nội suy dựa vào các khối lân cận ở các mép của hàng. Trên thực tế, chế độ phẳng biểu thị sự chuyển đổi mượt mà của ánh sáng/màu trên một hàng/cột bằng cách sử dụng độ dốc tương đối không đổi khi thay đổi các giá trị. Chế độ DC được sử dụng để làm mịn biên và biểu thị rằng khối tương tự/giống với giá trị trung bình được kết hợp với các mẫu của tất cả các khối lân cận được kết hợp với các hướng có góc của các chế độ dự đoán có hướng. Theo đó, các khối dự đoán nội ảnh có thể biểu diễn các khối hình ảnh dưới dạng các giá trị chế độ dự đoán quan hệ khác nhau thay vì các giá trị thực. Ngoài ra, các khối dự đoán liên ảnh có thể biểu diễn các khối hình ảnh dưới dạng các giá trị vectơ chuyển động thay vì các giá trị thực tế. Trong cả hai trường hợp, các khối dự đoán có thể không biểu diễn một cách chính xác các khối hình ảnh trong một số trường hợp. Sự khác biệt bất kỳ được lưu trữ trong các khối dư. Các phép biến đổi có thể được áp dụng cho các khối dư để nén thêm tệp.

Ở bước 107, các kỹ thuật lọc khác nhau có thể được áp dụng. Trong HEVC, các bộ lọc được áp dụng theo sơ đồ lọc trong vòng lặp. Phép dự đoán dựa trên khối được thảo luận ở trên có thể dẫn đến việc tạo các hình ảnh dạng khối ở bộ giải mã. Ngoài ra, sơ đồ dự đoán dựa trên khối có thể mã hóa khối và sau đó khôi phục khối được mã hóa để sử dụng sau đó làm khối tham chiếu. Sơ đồ lọc trong vòng lặp áp dụng lặp lại các bộ lọc triệt tiêu nhiễu, bộ lọc khử khối, bộ lọc vòng lặp thích ứng và bộ lọc độ lệch thích ứng theo mẫu (Sample Adaptive Offset, SAO) cho các khối/khung. Các bộ lọc này giảm thiểu các thành phần lạ tạo khối này sao cho tệp được mã hóa có thể được khôi phục một cách chính xác. Ngoài ra, các bộ lọc này giảm thiểu các thành phần lạ trong các khối tham

chiều được khôi phục sao cho các thành phần lạ này ít có khả năng tạo thêm các thành phần lạ trong các khối tiếp theo mà được mã hóa dựa vào các khối tham chiếu được khôi phục.

Khi tín hiệu video đã được phân vùng, nén và lọc, thì dữ liệu thu được được mã hóa trong dòng bit ở bước 109. Dòng bit này bao gồm dữ liệu được thảo luận ở trên cũng như dữ liệu báo hiệu bất kỳ được mong muốn để hỗ trợ việc khôi phục tín hiệu video thích hợp ở bộ giải mã. Ví dụ, dữ liệu này có thể bao gồm dữ liệu phân vùng, dữ liệu dự đoán, các khối dư và các cờ khác nhau tạo ra các lệnh mã hóa cho bộ giải mã. Dòng bit có thể được lưu trữ trong bộ nhớ để truyền về phía bộ giải mã ngay khi yêu cầu. Dòng bit có thể còn phát rộng và/hoặc phát đa hướng về phía các bộ giải mã. Việc tạo dòng bit là quy trình lặp lại. Theo đó, các bước 101, 103, 105, 107 và 109 có thể xảy ra liên tục và/hoặc đồng thời trên nhiều khung và khối. Thứ tự được thể hiện trên Fig.1 được trình bày cho rõ ràng và dễ dàng thảo luận, và không nhằm làm giới hạn quy trình mã hóa video theo thứ tự cụ thể.

Bộ giải mã nhận dòng bit và bắt đầu quy trình giải mã ở bước 111. Cụ thể, bộ giải mã sử dụng sơ đồ giải mã entropy để chuyển đổi dòng bit thành cú pháp tương ứng và dữ liệu video. Bộ giải mã sử dụng dữ liệu cú pháp từ dòng bit để xác định các phân vùng cho các khung ở bước 111. Bước phân vùng cần khớp với các kết quả phân vùng khối ở bước 103. Bước mã hóa/giải mã entropy như được sử dụng ở bước 111 hiện được mô tả. Bộ mã hóa thực hiện nhiều lựa chọn trong quy trình nén, chẳng hạn như chọn các sơ đồ phân vùng khối từ một số lựa chọn khả thi dựa vào vị trí theo không gian của các giá trị trong (các) hình ảnh đầu vào. Bước báo hiệu các lựa chọn chính xác có thể sử dụng số lượng bin (bin) lớn. Như được sử dụng ở đây, bin là giá trị nhị phân được coi là biến (ví dụ, giá trị bit mà có thể thay đổi tùy thuộc vào ngữ cảnh). Bước mã hóa entropy cho phép bộ mã hóa loại bỏ các tùy chọn bất kỳ mà rõ ràng không khả dụng đối với trường hợp cụ thể, để lại tập hợp tùy chọn được cho phép. Sau đó, mỗi tùy chọn được cho phép này được gán từ mã. Chiều dài của các từ mã là dựa vào số lượng tùy chọn được cho phép (ví dụ, một bin cho hai tùy chọn, hai bin cho ba hoặc bốn tùy chọn, v.v.). Sau đó, bộ mã hóa giải mã từ mã cho tùy chọn được chọn. Sơ đồ này giảm bớt kích thước của các từ mã khi các từ mã có độ lớn như mong muốn để biểu thị duy nhất lựa chọn từ tập hợp con lựa chọn được cho phép nhỏ trái ngược với việc biểu thị duy nhất lựa chọn từ tập hợp tất cả các tùy chọn khả thi có thể lớn. Sau đó, bộ giải mã giải mã lựa chọn này bằng cách xác định tập hợp tùy chọn được cho phép theo cách tương tự với bộ mã hóa. Bằng cách xác định tập hợp tùy chọn được cho phép, bộ giải mã có thể đọc từ mã và xác định lựa chọn được thực hiện bởi bộ mã hóa.

Ở bước 113, bộ giải mã thực hiện bước giải mã khối. Cụ thể, bộ giải mã sử dụng các phép biến đổi ngược để tạo ra các khối dư. Sau đó, bộ giải mã sử dụng các khối dư và khối dự đoán tương ứng để khôi phục các khối hình ảnh theo bước phân vùng. Các khối dự đoán có thể bao gồm cả các khối dự đoán nội ảnh và các khối dự đoán liên ảnh như được tạo ra ở bộ mã hóa ở bước 105. Sau đó, các khối hình ảnh được khôi phục được định vị vào các khung của tín hiệu video được khôi phục theo dữ liệu phân vùng được xác định ở bước 111. Cú pháp dùng cho bước 113 có thể còn được báo hiệu trong dòng bit qua bước mã hóa entropy như được thảo luận ở trên.

Ở bước 115, bước lọc được thực hiện đối với các khung của tín hiệu video được khôi phục theo cách tương tự với bước 107 ở bộ mã hóa. Ví dụ, các bộ lọc triệt tiêu nhiễu, bộ lọc khử khối, bộ lọc vòng lặp thích ứng và bộ lọc SAO có thể được áp dụng cho các khung để loại bỏ các thành phần lạ tạo khối. Khi các khung được lọc, thì tín hiệu video có thể được kết xuất đến màn hiển thị ở bước 117 để xem bởi người dùng cuối.

Fig.2 là sơ đồ sơ lược của hệ thống mã hóa và giải mã (mã hóa-giải mã) làm ví dụ 200 để mã hóa video. Cụ thể, hệ thống mã hóa-giải mã 200 cung cấp chức năng để hỗ trợ việc thực hiện phương pháp hoạt động 100. Hệ thống mã hóa-giải mã 200 được chuẩn hóa để mô tả các thành phần được sử dụng trong cả bộ mã hóa và bộ giải mã. Hệ thống mã hóa-giải mã 200 nhận và phân vùng tín hiệu video như được thảo luận liên quan đến các bước 101 và 103 theo phương pháp hoạt động 100, mà tạo ra tín hiệu video được phân vùng 201. Sau đó, hệ thống mã hóa-giải mã 200 nén tín hiệu video được phân vùng 201 thành dòng bit được mã hóa khi hoạt động dưới dạng bộ mã hóa như được thảo luận liên quan đến các bước 105, 107 và 109 theo phương pháp 100. Khi hoạt động dưới dạng bộ giải mã, hệ thống mã hóa-giải mã 200 tạo ra tín hiệu video đầu ra từ dòng bit như được thảo luận liên quan đến các bước 111, 113, 115 và 117 theo phương pháp hoạt động 100. Hệ thống mã hóa-giải mã 200 bao gồm thành phần điều khiển bộ mã hóa chung 211, thành phần định tỷ lệ biến đổi và lượng tử hóa 213, thành phần đánh giá nội ảnh 215, thành phần dự đoán nội ảnh 217, thành phần bù chuyển động 219, thành phần đánh giá chuyển động 221, thành phần định tỷ lệ và biến đổi ngược 229, thành phần phân tích điều khiển bộ lọc 227, thành phần bộ lọc trong vòng lặp 225, thành phần bộ đệm ảnh được giải mã 223 và thành phần định dạng phần đầu và mã hóa số học nhị phân thích ứng theo ngữ cảnh (context adaptive binary arithmetic coding, CABAC) 231. Các thành phần này được ghép nối như được thể hiện trên Fig.2, các đường màu đen biểu thị chuyển động của dữ liệu cần được mã hóa/giải mã trong khi các đường nét đứt biểu thị chuyển động của dữ liệu điều khiển để điều khiển hoạt động của các thành phần khác. Các thành phần của hệ thống mã hóa-giải mã 200 đều có thể có mặt trong bộ mã hóa. Bộ giải mã có thể bao gồm tập hợp con thành phần của hệ thống mã hóa-giải mã 200. Ví dụ, bộ giải mã có thể bao gồm

thành phần dự đoán nội ảnh 217, thành phần bù chuyển động 219, thành phần định tỷ lệ và biến đổi ngược 229, thành phần bộ lọc trong vòng lặp 225 và thành phần bộ đệm ảnh được giải mã 223. Các thành phần này hiện được mô tả.

Tín hiệu video được phân vùng 201 là chuỗi video được thu thập mà đã được phân vùng thành các khối điểm ảnh bởi cây mã hóa. Cây mã hóa sử dụng các chế độ phân chia khác nhau để chia nhỏ khối điểm ảnh thành các khối điểm ảnh nhỏ hơn. Sau đó, các khối này có thể còn được chia nhỏ thành các khối nhỏ hơn. Các khối này có thể được gọi là các nút trên cây mã hóa. Các nút cha lớn hơn được phân chia thành các nút con nhỏ hơn. Số lần một nút được chia nhỏ được gọi là độ sâu của nút/cây mã hóa. Các khối được chia có thể có trong các đơn vị mã hóa (CU) trong một số trường hợp. Ví dụ, CU có thể là phần nhỏ của CTU mà chứa khối độ chói, (các) khối sắc độ khác biệt màu đỏ (Cr) và (các) khối sắc độ khác biệt màu xanh da trời (Cb) cùng với các lệnh cú pháp tương ứng đối với CU. Các chế độ phân chia có thể bao gồm cây nhị phân (Binary Tree, BT), cây tam phân (Triple Tree, TT) và cây tứ phân (Quad Tree, QT) được sử dụng để lần lượt phân vùng nút thành hai, ba hoặc bốn nút con, có các hình dạng thay đổi tùy thuộc vào các chế độ phân chia được sử dụng. Tín hiệu video được phân vùng 201 được chuyển tiếp đến thành phần điều khiển bộ mã hóa chung 211, thành phần định tỷ lệ biến đổi và lượng tử hóa 213, thành phần đánh giá nội ảnh 215, thành phần phân tích điều khiển bộ lọc 227, và thành phần đánh giá chuyển động 221 để nén.

Thành phần điều khiển bộ mã hóa chung 211 được tạo cấu hình để đưa ra các quyết định liên quan đến việc mã hóa các hình ảnh của chuỗi video thành dòng bit theo các ràng buộc về ứng dụng. Ví dụ, thành phần điều khiển bộ mã hóa chung 211 quản lý việc tối ưu hóa kích thước tốc độ bit/dòng bit so với chất lượng khôi phục. Các quyết định này có thể được đưa ra dựa vào tính khả dụng của không gian lưu trữ/băng thông và các yêu cầu về độ phân giải của hình ảnh. Thành phần điều khiển bộ mã hóa chung 211 còn quản lý việc sử dụng bộ đệm dựa vào tốc độ truyền để giảm thiểu các vấn đề trống bộ đệm và tràn bộ đệm. Để quản lý các vấn đề này, thành phần điều khiển bộ mã hóa chung 211 quản lý việc phân vùng, dự đoán và lọc bởi các thành phần khác. Ví dụ, thành phần điều khiển bộ mã hóa chung 211 có thể tăng độ phức tạp nén theo cách động để tăng độ phân giải và tăng mức sử dụng băng thông hoặc giảm độ phức tạp nén để giảm độ phân giải và mức sử dụng băng thông. Do đó, thành phần điều khiển bộ mã hóa chung 211 điều khiển các thành phần khác của hệ thống mã hóa-giải mã 200 để cân bằng chất lượng khôi phục tín hiệu video với các mối quan tâm về tốc độ bit. Thành phần điều khiển bộ mã hóa chung 211 tạo ra dữ liệu điều khiển, mà điều khiển hoạt động của các thành phần khác. Dữ liệu điều khiển cũng được chuyển tiếp đến thành phần định dạng phần đầu và CABAC 231 để được mã hóa trong dòng bit để báo hiệu các tham số để giải mã ở bộ giải mã.

Tín hiệu video được phân vùng 201 còn được gửi đến thành phần đánh giá chuyển động 221 và thành phần bù chuyển động 219 để dự đoán liên ảnh. Khung hoặc lát của tín hiệu video được phân vùng 201 có thể được chia thành nhiều khối video. Thành phần đánh giá chuyển động 221 và thành phần bù chuyển động 219 thực hiện bước mã hóa dự đoán liên ảnh của khối video nhận được liên quan đến một hoặc nhiều khối trong một hoặc nhiều khung tham chiếu để tạo ra phép dự đoán theo thời gian. Hệ thống mã hóa-giải mã 200 có thể thực hiện bước mã hóa nhiều lần, ví dụ, để chọn chế độ mã hóa thích hợp cho mỗi khối của dữ liệu video.

Thành phần đánh giá chuyển động 221 và thành phần bù chuyển động 219 có thể được tích hợp ở mức độ cao, nhưng được minh họa riêng biệt nhằm mục đích nêu ý tưởng. Phép đánh giá chuyển động, được thực hiện bởi thành phần đánh giá chuyển động 221, là quy trình tạo ra các vectơ chuyển động, mà đánh giá chuyển động đối với các khối video. Vectơ chuyển động, ví dụ, có thể biểu thị sự dịch chuyển của đối tượng được mã hóa liên quan đến khối dự đoán. Khối dự đoán là khối được nhận thấy là gần khớp với khối cần được mã hóa, về mặt khác biệt điểm ảnh. Khối dự đoán cũng có thể được gọi là khối tham chiếu. Sự khác biệt điểm ảnh như vậy có thể được xác định bởi tổng chênh lệch tuyệt đối (Sum of Absolute Difference, SAD), tổng chênh lệch bình phương (Sum of Square Difference, SSD) hoặc các số đo khác nhau khác. HEVC sử dụng một số đối tượng được mã hóa bao gồm CTU, các khối cây mã hóa (Coding Tree Block, CTB) và các CU. Ví dụ, CTU có thể được chia thành CTB, sau đó có thể được chia thành các CB để đưa vào các CU. CU có thể được mã hóa dưới dạng đơn vị dự đoán chứa dữ liệu dự đoán và/hoặc đơn vị biến đổi (Transform Unit, TU) chứa dữ liệu dư được biến đổi cho CU. Thành phần đánh giá chuyển động 221 tạo ra các vectơ chuyển động, đơn vị dự đoán và TU bằng cách sử dụng phép phân tích tỷ lệ méo là một phần của quy trình tối ưu hóa tỷ lệ méo. Ví dụ, thành phần đánh giá chuyển động 221 có thể xác định nhiều khối tham chiếu, nhiều vectơ chuyển động, v.v. cho khối/khung hiện thời, và có thể chọn các khối tham chiếu, vectơ chuyển động, v.v. có các đặc điểm tỷ lệ méo tốt nhất. Các đặc điểm tỷ lệ méo tốt nhất cân bằng cả chất lượng khôi phục video (ví dụ, lượng dữ liệu mất mát do nén) với hiệu quả mã hóa (ví dụ, kích thước mã hóa cuối cùng).

Trong một số ví dụ, hệ thống mã hóa-giải mã 200 có thể tính toán các giá trị đối với các vị trí điểm ảnh nguyên con của các ảnh tham chiếu được lưu trong thành phần bộ đệm ảnh được giải mã 223. Ví dụ, hệ thống mã hóa-giải mã video 200 có thể nội suy các giá trị của các vị trí điểm ảnh một phần tư, các vị trí điểm ảnh một phần tám hoặc các vị trí điểm ảnh phân số khác của ảnh tham chiếu. Do đó, thành phần đánh giá chuyển động 221 có thể thực hiện việc tìm kiếm chuyển động liên quan đến các vị trí điểm ảnh đầy đủ và các vị trí điểm ảnh phân số và kết xuất vectơ chuyển động có độ chính xác điểm ảnh

phân số. Thành phần đánh giá chuyển động 221 tính toán vectơ chuyển động cho đơn vị dự đoán của video khối trong lát được mã hóa liên ảnh bằng cách so sánh vị trí của đơn vị dự đoán với vị trí của khối dự đoán của ảnh tham chiếu. Thành phần đánh giá chuyển động 221 kết xuất vectơ chuyển động được tính toán dưới dạng dữ liệu chuyển động cho thành phần định dạng phần đầu và CABAC 231 để mã hóa và chuyển động cho thành phần bù chuyển động 219.

Bước bù chuyển động, được thực hiện bởi thành phần bù chuyển động 219, có thể liên quan đến việc nạp hoặc tạo ra khối dự đoán dựa vào vectơ chuyển động được xác định bởi thành phần đánh giá chuyển động 221. Một lần nữa, thành phần đánh giá chuyển động 221 và thành phần bù chuyển động 219 có thể được tích hợp về mặt chức năng, trong một số ví dụ. Ngay khi nhận vectơ chuyển động đối với đơn vị dự đoán của khối video hiện thời, thành phần bù chuyển động 219 có thể xác định vị trí khối dự đoán mà vectơ chuyển động chỉ đến. Sau đó, khối video dự được tạo ra bằng cách trừ các giá trị điểm ảnh của khối dự đoán từ các giá trị điểm ảnh của khối video hiện thời đang được mã hóa, tạo ra các giá trị chênh lệch điểm ảnh. Nói chung, thành phần đánh giá chuyển động 221 thực hiện bước đánh giá chuyển động liên quan đến các thành phần độ chói, và thành phần bù chuyển động 219 sử dụng các vectơ chuyển động được tính dựa vào các thành phần độ chói đối với cả các thành phần sắc độ và các thành phần độ chói. Khối dự đoán và khối dự được chuyển tiếp đến thành phần định tỷ lệ biến đổi và lượng tử hóa 213.

Tín hiệu video được phân vùng 201 cũng được gửi đến thành phần đánh giá nội ảnh 215 và thành phần dự đoán nội ảnh 217. Như với thành phần đánh giá chuyển động 221 và thành phần bù chuyển động 219, thành phần đánh giá nội ảnh 215 và thành phần dự đoán nội ảnh 217 có thể được tích hợp ở mức độ cao, nhưng được minh họa riêng biệt nhằm mục đích nêu ý tưởng. Thành phần đánh giá nội ảnh 215 và thành phần dự đoán nội ảnh 217 dự đoán nội ảnh khối hiện thời so với các khối trong khung hiện thời, như sự thay thế cho phép dự đoán liên ảnh được thực hiện bởi thành phần đánh giá chuyển động 221 và thành phần bù chuyển động 219 giữa các khung, như được mô tả ở trên. Cụ thể là, thành phần đánh giá nội ảnh 215 xác định chế độ dự đoán nội ảnh sử dụng để mã hóa khối hiện thời. Trong một số ví dụ, thành phần đánh giá nội ảnh 215 chọn chế độ dự đoán nội ảnh thích hợp để mã hóa khối hiện thời từ nhiều chế độ dự đoán nội ảnh đã thử nghiệm. Sau đó, các chế độ dự đoán nội ảnh đã chọn được chuyển tiếp đến thành phần định dạng phần đầu và CABAC 231 để mã hóa.

Ví dụ, thành phần đánh giá nội ảnh 215 tính các giá trị tỷ lệ méo sử dụng phép phân tích tỷ lệ méo đối với các chế độ dự đoán nội ảnh đã thử nghiệm khác nhau, và chọn chế độ dự đoán nội ảnh có các đặc điểm tỷ lệ méo tốt nhất trong số các chế độ đã thử nghiệm. Phép phân tích tỷ lệ méo thường xác định lượng méo (hoặc sai số) giữa khối

được mã hóa và khối không được mã hóa gốc mà đã được mã hóa để tạo ra khối được mã hóa, cũng như tốc độ bit (ví dụ, số lượng bit) được sử dụng để tạo ra khối được mã hóa. Thành phần đánh giá nội ảnh 215 tính toán các tỷ lệ từ các độ méo và các tỷ lệ đối với các khối được mã hóa khác nhau để xác định chế độ dự đoán nội ảnh thể hiện giá trị tỷ lệ méo tốt nhất cho khối này. Ngoài ra, thành phần đánh giá nội ảnh 215 có thể được tạo cấu hình để mã hóa các khối độ sâu của bản đồ độ sâu bằng cách sử dụng chế độ mô hình hóa độ sâu (Depth Modeling Mode, DMM) dựa vào phép tối ưu tỷ lệ méo (Rate-Distortion Optimization, RDO).

Thành phần dự đoán nội ảnh 217 có thể tạo ra khối dư từ khối dự đoán dựa vào các chế độ dự đoán nội ảnh được chọn được xác định bởi thành phần đánh giá nội ảnh 215 khi được thực hiện trên bộ mã hóa hoặc đọc khối dư từ dòng bit khi được thực hiện trên bộ giải mã. Khối dư bao gồm sự chênh lệch về các giá trị giữa khối dự đoán và khối gốc, được biểu diễn dưới dạng ma trận. Sau đó, khối dư được chuyển tiếp đến thành phần định tỷ lệ biến đổi và lượng tử hóa 213. Thành phần đánh giá nội ảnh 215 và thành phần dự đoán nội ảnh 217 có thể hoạt động trên cả các thành phần độ chói và sắc độ.

Thành phần định tỷ lệ biến đổi và lượng tử hóa 213 được tạo cấu hình để nén thêm khối dư. Thành phần định tỷ lệ biến đổi và lượng tử hóa 213 này áp dụng phép biến đổi, chẳng hạn như biến đổi cosin rời rạc (Discrete Cosine Transform, DCT), biến đổi sin rời rạc (Discrete Sine Transform, DST) hoặc phép biến đổi tương tự về mặt khái niệm, đối với khối dư, tạo ra khối video bao gồm các giá trị hệ số biến đổi dư. Các phép biến đổi Wavelet, biến đổi nguyên, biến đổi dải con hoặc các kiểu biến đổi khác cũng có thể được sử dụng. Phép biến đổi này có thể chuyển đổi thông tin dư từ miền giá trị điểm ảnh thành miền biến đổi, chẳng hạn như miền tần số. Thành phần định tỷ lệ biến đổi và lượng tử hóa 213 cũng được tạo cấu hình để định tỷ lệ thông tin dư được biến đổi, ví dụ dựa vào tần số. Việc định tỷ lệ này bao gồm việc áp dụng hệ số định tỷ lệ cho thông tin dư sao cho thông tin tần số khác nhau được lượng tử hóa ở các độ chi tiết khác nhau, mà có thể ảnh hưởng đến chất lượng trực quan cuối cùng của video được khôi phục. Thành phần định tỷ lệ biến đổi và lượng tử hóa 213 cũng được tạo cấu hình để lượng tử hóa các hệ số biến đổi để giảm thêm tốc độ bit. Quy trình lượng tử hóa có thể làm giảm độ sâu bit được kết hợp với một số hoặc tất cả các hệ số. Mức độ lượng tử hóa có thể được biến đổi bằng cách điều chỉnh tham số lượng tử hóa. Trong một số ví dụ, thành phần định tỷ lệ biến đổi và lượng tử hóa 213 sau đó có thể thực hiện việc quét ma trận bao gồm các hệ số biến đổi được lượng tử hóa. Các hệ số biến đổi được lượng tử hóa này được chuyển tiếp đến thành phần định dạng phân đầu và CABAC 231 để được mã hóa trong dòng bit.

Thành phần định tỷ lệ và biến đổi ngược 229 áp dụng hoạt động đảo ngược của thành phần định tỷ lệ biến đổi và lượng tử hóa 213 để hỗ trợ việc đánh giá chuyển động.

Thành phần định tỷ lệ và biến đổi ngược 229 áp dụng bước định tỷ lệ, biến đổi và/hoặc lượng tử hóa ngược để khôi phục khối dư trong miền điểm ảnh, ví dụ, để sau đó sử dụng làm khối tham chiếu mà có thể trở thành khối dự đoán cho khối hiện thời khác. Thành phần đánh giá chuyển động 221 và/hoặc thành phần bù chuyển động 219 có thể tính toán khối tham chiếu bằng cách thêm khối dư trở lại khối dự đoán tương ứng để sử dụng trong đánh giá chuyển động khối/khung sau đó. Các bộ lọc được áp dụng cho các khối tham chiếu được khôi phục để giảm thiểu các thành phần lạ được tạo ra trong quá trình định tỷ lệ, lượng tử hóa và biến đổi. Các thành phần lạ này có thể gây ra việc dự đoán không chính xác (và tạo ra các thành phần lạ bổ sung) khi các khối tiếp theo được dự đoán.

Thành phần phân tích điều khiển bộ lọc 227 và thành phần bộ lọc trong vòng lặp 225 áp dụng các bộ lọc cho các khối dư và/hoặc để khôi phục các khối hình ảnh. Ví dụ, khối dư được biến đổi từ thành phần định tỷ lệ và biến đổi ngược 229 có thể được kết hợp với khối dự đoán tương ứng từ thành phần dự đoán nội ảnh 217 và/hoặc thành phần bù chuyển động 219 để khôi phục khối hình ảnh gốc. Sau đó, các bộ lọc được áp dụng cho các khối hình ảnh được khôi phục. Trong một số ví dụ, thay vào đó các bộ lọc có thể được áp dụng cho các khối dư. Như với các thành phần khác trên Fig.2, thành phần phân tích điều khiển bộ lọc 227 và thành phần bộ lọc trong vòng lặp 225 được tích hợp sâu và có thể được thực hiện cùng nhau, nhưng được mô tả riêng biệt nhằm các mục đích nêu ý tưởng. Các bộ lọc được áp dụng cho các khối tham chiếu được khôi phục được áp dụng cho các vùng không gian cụ thể và bao gồm nhiều tham số để điều chỉnh cách các bộ lọc được áp dụng. Thành phần phân tích điều khiển bộ lọc 227 phân tích các khối tham chiếu được khôi phục để xác định nơi mà các bộ lọc này cần được áp dụng và thiết lập các tham số tương ứng. Dữ liệu này được chuyển tiếp đến thành phần định dạng phần đầu và CABAC 231 dưới dạng dữ liệu điều khiển bộ lọc để mã hóa. Thành phần bộ lọc trong vòng lặp 225 áp dụng các bộ lọc này dựa vào dữ liệu điều khiển bộ lọc. Các bộ lọc có thể bao gồm bộ lọc khử khối, bộ lọc triệt tiêu nhiễu, bộ lọc SAO và bộ lọc vòng lặp thích ứng. Các bộ lọc này có thể được áp dụng trong miền không gian/điểm ảnh (ví dụ, trên khối điểm ảnh được khôi phục) hoặc trong miền tần số, tùy thuộc vào ví dụ.

Khi hoạt động dưới dạng bộ mã hóa, khối hình ảnh được khôi phục được lọc, khối dư và/hoặc khối dự đoán được lưu trữ trong thành phần bộ đệm ảnh được giải mã 223 để sử dụng sau khi đánh giá chuyển động như được thảo luận ở trên. Khi hoạt động dưới dạng bộ giải mã, thành phần bộ đệm ảnh được giải mã 223 lưu trữ và chuyển tiếp các khối đã khôi phục và lọc về phía màn hiển thị dưới dạng một phần tín hiệu video đầu ra. Thành phần bộ đệm ảnh được giải mã 223 có thể là thiết bị bộ nhớ bất kỳ có khả năng lưu trữ các khối dự đoán, khối dư và/hoặc khối hình ảnh được khôi phục.

Thành phần định dạng phần đầu và CABAC 231 nhận dữ liệu từ các thành phần khác của hệ thống mã hóa-giải mã 200 và mã hóa dữ liệu này thành dòng bit được mã hóa để truyền về phía bộ giải mã. Cụ thể, thành phần định dạng phần đầu và CABAC 231 tạo ra các phần đầu khác nhau để mã hóa dữ liệu điều khiển, chẳng hạn như dữ liệu điều khiển chung và dữ liệu điều khiển bộ lọc. Ngoài ra, dữ liệu dự đoán, bao gồm dữ liệu dự đoán nội ảnh và chuyển động, cũng như dữ liệu dư dưới dạng dữ liệu hệ số biến đổi được lượng tử hóa đều được mã hóa trong dòng bit. Dòng bit cuối cùng bao gồm tất cả thông tin được mong muốn bởi bộ giải mã để khôi phục tín hiệu video gốc được phân vùng 201. Thông tin này có thể còn bao gồm các bảng chỉ số chế độ dự đoán nội ảnh (còn được gọi là các bảng ánh xạ từ mã), các định nghĩa về ngữ cảnh mã hóa đối với các khối khác nhau, chỉ báo của hầu hết các chế độ dự đoán nội ảnh khả dụng, chỉ báo của thông tin phân vùng, v.v. Dữ liệu này có thể được mã hóa bằng cách sử dụng bước mã hóa entropy. Ví dụ, thông tin này có thể được mã hóa bằng cách sử dụng mã hóa chiều dài biến đổi thích ứng theo ngữ cảnh (Context Adaptive Variable Length Coding, CAVLC), CABAC, mã hóa số học nhị phân thích ứng theo ngữ cảnh dựa trên cú pháp (Syntax-Based Context-Adaptive Binary Arithmetic Coding, SBAC), mã hóa entropy phân vùng theo khoảng xác suất (Probability Interval Partitioning Entropy, PIPE) hoặc kỹ thuật mã hóa entropy khác. Sau khi mã hóa entropy, dòng bit được mã hóa có thể được truyền đến thiết bị khác (ví dụ, bộ giải mã video) hoặc được lưu trữ để truyền hoặc truy xuất sau đó.

Fig.3 là sơ đồ khối minh họa bộ mã hóa video 300 làm ví dụ. Bộ mã hóa video 300 này có thể được sử dụng để thực hiện các chức năng mã hóa của hệ thống mã hóa-giải mã 200 và/hoặc thực hiện các bước 101, 103, 105, 107 và/hoặc 109 của phương pháp hoạt động 100. Bộ mã hóa 300 phân vùng tín hiệu video đầu vào, dẫn đến tín hiệu video được phân vùng 301, gần như tương tự với tín hiệu video được phân vùng 201. Tín hiệu video được phân vùng 301 này sau đó được nén và mã hóa thành dòng bit bởi các thành phần của bộ mã hóa 300.

Cụ thể, tín hiệu video được phân vùng 301 được chuyển tiếp đến thành phần dự đoán nội ảnh 317 để dự đoán nội ảnh. Thành phần dự đoán nội ảnh 317 có thể gần như tương tự với thành phần đánh giá nội ảnh 215 và thành phần dự đoán nội ảnh 217. Tín hiệu video được phân vùng 301 cũng được chuyển tiếp đến thành phần bù chuyển động 321 để dự đoán liên ảnh dựa vào các khối tham chiếu trong thành phần bộ đệm ảnh được giải mã 323. Thành phần bù chuyển động 321 có thể gần như tương tự với thành phần đánh giá chuyển động 221 và thành phần bù chuyển động 219. Các khối dự đoán và các khối dư từ thành phần dự đoán nội ảnh 317 và thành phần bù chuyển động 321 được chuyển tiếp đến thành phần biến đổi và lượng tử hóa 313 để biến đổi và lượng tử hóa các khối dư. Thành phần biến đổi và lượng tử hóa 313 có thể gần như tương tự với thành phần

định tỷ lệ biến đổi và lượng tử hóa 213. Các khối dư được biến đổi và được lượng tử hóa và các khối dự đoán tương ứng (cùng với dữ liệu điều khiển liên quan) được chuyển tiếp đến thành phần mã hóa entropy 331 để mã hóa thành dòng bit. Thành phần mã hóa entropy 331 có thể gần như tương tự với thành phần định dạng phần đầu và CABAC 231.

Các khối dư được biến đổi và lượng tử hóa và/hoặc các khối dự đoán tương ứng cũng được chuyển tiếp từ thành phần biến đổi và lượng tử hóa 313 đến thành phần biến đổi và lượng tử hóa ngược 329 để khôi phục thành các khối tham chiếu để sử dụng bởi thành phần bù chuyển động 321. Thành phần biến đổi và lượng tử hóa ngược 329 có thể gần như tương tự với thành phần định tỷ lệ và biến đổi ngược 229. Các bộ lọc trong vòng lặp trong thành phần bộ lọc trong vòng lặp 325 cũng được áp dụng cho các khối dư và/hoặc các khối tham chiếu được khôi phục, tùy thuộc vào ví dụ này. Thành phần bộ lọc trong vòng lặp 325 có thể gần như tương tự với thành phần phân tích điều khiển bộ lọc 227 và thành phần bộ lọc trong vòng lặp 225. Thành phần bộ lọc trong vòng lặp 325 có thể bao gồm nhiều bộ lọc như đã thảo luận liên quan đến thành phần bộ lọc trong vòng lặp 225. Các khối được lọc sau đó được lưu trữ trong thành phần bộ đệm ảnh được giải mã 323 để sử dụng làm các khối tham chiếu bởi thành phần bù chuyển động 321. Thành phần bộ đệm ảnh được giải mã 323 có thể gần như tương tự với thành phần bộ đệm ảnh được giải mã 223.

Fig.4 là sơ đồ khối minh họa bộ giải mã video 400 làm ví dụ. Bộ giải mã video 400 này có thể được sử dụng để thực hiện các chức năng giải mã của hệ thống mã hóa-giải mã 200 và/hoặc thực hiện các bước 111, 113, 115 và/hoặc 117 của phương pháp hoạt động 100. Bộ giải mã 400 nhận dòng bit, ví dụ từ bộ mã hóa 300, và tạo ra tín hiệu video đầu ra được khôi phục dựa thành dòng bit để hiển thị cho người dùng cuối.

Dòng bit nhận được bởi thành phần giải mã entropy 433. Thành phần giải mã entropy 433 được tạo cấu hình để thực hiện sơ đồ giải mã entropy, chẳng hạn như mã hóa CAVLC, CABAC, SBAC, PIPE, hoặc các kỹ thuật mã hóa entropy khác. Ví dụ, thành phần giải mã entropy 433 có thể sử dụng thông tin phần đầu để tạo ra ngữ cảnh để biên dịch dữ liệu bổ sung được mã hóa dưới dạng các từ mã trong dòng bit. Thông tin được giải mã bao gồm thông tin mong muốn bất kỳ để giải mã tín hiệu video, chẳng hạn như dữ liệu điều khiển chung, dữ liệu điều khiển bộ lọc, thông tin phân vùng, dữ liệu chuyển động, dữ liệu dự đoán và các hệ số biến đổi được lượng tử hóa từ các khối dư. Các hệ số biến đổi được lượng tử hóa này được chuyển tiếp đến thành phần biến đổi và lượng tử hóa ngược 429 để khôi phục thành các khối dư. Thành phần biến đổi và lượng tử hóa ngược 429 có thể tương tự với thành phần biến đổi và lượng tử hóa ngược 329.

Các khối dư được khôi phục và/hoặc các khối dự đoán được chuyển tiếp đến thành phần dự đoán nội ảnh 417 để khôi phục thành các khối hình ảnh dựa vào các hoạt động

dự đoán nội ảnh. Thành phần dự đoán nội ảnh 417 có thể tương tự với thành phần đánh giá nội ảnh 215 và thành phần dự đoán nội ảnh 217. Cụ thể, thành phần dự đoán nội ảnh 417 sử dụng các chế độ dự đoán để xác định vị trí khối tham chiếu trong khung và áp dụng khối dư cho kết quả để khôi phục các khối hình ảnh được dự đoán nội ảnh. Các khối hình ảnh được dự đoán nội ảnh được khôi phục và/hoặc các khối dư và dữ liệu dự đoán liên ảnh tương ứng được chuyển tiếp đến thành phần bộ đệm ảnh được giải mã 423 qua thành phần bộ lọc trong vòng lặp 425, lần lượt có thể gần như tương tự với thành phần bộ đệm ảnh được giải mã 223 và thành phần bộ lọc trong vòng lặp 225. Thành phần bộ lọc trong vòng lặp 425 lọc các khối hình ảnh được khôi phục, khối dư và/hoặc khối dự đoán, và thông tin này được lưu trữ trong thành phần bộ đệm ảnh được giải mã 423. Các khối hình ảnh được khôi phục từ thành phần bộ đệm ảnh được giải mã 423 được chuyển tiếp đến thành phần bù chuyển động 421 để dự đoán liên ảnh. Thành phần bù chuyển động 421 có thể gần như tương tự với thành phần đánh giá chuyển động 221 và/hoặc thành phần bù chuyển động 219. Cụ thể, thành phần bù chuyển động 421 sử dụng các vectơ chuyển động từ khối tham chiếu để tạo ra khối dự đoán và áp dụng khối dư cho kết quả để khôi phục khối hình ảnh. Các khối được khôi phục được tạo ra cũng có thể được chuyển tiếp qua thành phần bộ lọc trong vòng lặp 425 đến thành phần bộ đệm ảnh được giải mã 423. Thành phần bộ đệm ảnh được giải mã 423 tiếp tục lưu trữ các khối hình ảnh được khôi phục bổ sung, mà có thể được khôi phục thành các khung qua thông tin phân vùng. Các khung này cũng có thể được đặt theo trình tự. Trình tự này được kết xuất về phía màn hiển thị dưới dạng tín hiệu video đầu ra được khôi phục.

Fig.5 là sơ đồ sơ lược minh họa HRD 500 làm ví dụ. HRD 500 này có thể được sử dụng trong bộ mã hóa, chẳng hạn như hệ thống mã hóa-giải mã 200 và/hoặc bộ mã hóa 300. HRD 500 có thể kiểm tra dòng bit được tạo ra ở bước 109 của phương pháp 100 trước khi dòng bit được chuyển tiếp đến bộ giải mã, chẳng hạn như bộ giải mã 400. Trong một số ví dụ, dòng bit có thể được chuyển tiếp liên tục qua HRD 500 khi dòng bit được mã hóa. Trong trường hợp mà một phần của dòng bit không phù hợp với các ràng buộc liên quan, thì HRD 500 có thể biểu thị lỗi này cho bộ mã hóa để khiến bộ mã hóa mã hóa lại phần tương ứng của dòng bit với các cơ chế khác.

HRD 500 bao gồm trình lập lịch biểu dòng giả định (hypothetical stream scheduler, HSS) 541. HSS 541 là thành phần được tạo cấu hình để thực hiện cơ chế vận chuyển giả định. Cơ chế vận chuyển giả định được sử dụng để kiểm tra tính phù hợp của dòng bit hoặc bộ giải mã về thời gian và luồng dữ liệu của dòng bit 551 được nhập vào HRD 500. Ví dụ, HSS 541 có thể nhận dòng bit 551 được kết xuất từ bộ mã hóa và quản lý quy trình kiểm tra tính phù hợp đối với dòng bit 551. Trong một ví dụ cụ thể, HSS 541 có thể điều

khuyến tốc độ mà các ảnh được mã hóa di chuyển qua HRD 500 và xác minh rằng dòng bit 551 không chứa dữ liệu không phù hợp.

HSS 541 có thể chuyển tiếp dòng bit 551 đến CPB 543 ở tốc độ định trước. HRD 500 có thể quản lý dữ liệu trong các đơn vị giải mã (decoding unit, DU) 553. DU 553 là đơn vị truy cập (AU) hoặc tập hợp con gồm AU và các đơn vị lớp trừu tượng hóa mạng (NAL) không phải lớp mã hóa video (VCL) được kết hợp. Cụ thể là, AU chứa một hoặc nhiều ảnh được kết hợp với thời gian kết xuất. Ví dụ, AU có thể chứa một ảnh duy nhất trong dòng bit một lớp duy nhất, và có thể chứa ảnh đối với mỗi lớp trong dòng bit nhiều lớp. Mỗi ảnh của AU có thể được chia thành các lát mà mỗi lát này có trong đơn vị VCL NAL tương ứng. Do đó, DU 553 có thể chứa một hoặc nhiều ảnh, một hoặc nhiều lát của ảnh, hoặc các kết hợp của chúng. Ngoài ra, các tham số được sử dụng để giải mã AU, các ảnh và/hoặc các lát có thể có trong các đơn vị NAL không phải VCL. Như vậy, DU 553 chứa các đơn vị NAL không phải VCL mà chứa dữ liệu cần để hỗ trợ việc giải mã các đơn vị VCL NAL trong DU 553. CPB 543 là bộ đệm vào trước-ra trước trong HRD 500. CPB 543 chứa các DU 553 bao gồm dữ liệu video theo thứ tự giải mã. CPB 543 lưu trữ dữ liệu video để sử dụng trong quá trình xác minh tính phù hợp của dòng bit.

CPB 543 chuyển tiếp các DU 553 đến thành phần quy trình giải mã 545. Thành phần quy trình giải mã 545 là thành phần phù hợp với tiêu chuẩn VVC. Ví dụ, thành phần quy trình giải mã 545 có thể mô phỏng bộ giải mã 400 được sử dụng bởi người dùng cuối. Thành phần quy trình giải mã 545 này giải mã các DU 553 ở tốc độ mà có thể đạt được bởi bộ giải mã người dùng cuối làm ví dụ. Nếu thành phần quy trình giải mã 545 không thể giải mã các DU 553 đủ nhanh để ngăn chặn việc tràn CPB 543, thì dòng bit 551 không phù hợp với tiêu chuẩn và cần được mã hóa lại.

Thành phần quy trình giải mã 545 giải mã các DU 553, mà tạo ra các DU được giải mã 555. DU được giải mã 555 chứa ảnh được giải mã. Các DU được giải mã 555 được chuyển tiếp đến DPB 547. DPB 547 có thể gần như tương tự với thành phần bộ đệm ảnh được giải mã 223, 323 và/hoặc 423. Để hỗ trợ việc dự đoán liên ảnh, các ảnh mà được đánh dấu để sử dụng làm các ảnh tham chiếu 556 mà thu được từ các DU được giải mã 555 được trả về thành phần quy trình giải mã 545 để hỗ trợ thêm việc giải mã. DPB 547 kết xuất chuỗi video được giải mã dưới dạng chuỗi ảnh 557. Các ảnh 557 là các ảnh được khôi phục mà thường phản chiếu các ảnh được mã hóa thành dòng bit 551 bởi bộ mã hóa.

Các ảnh 557 được chuyển tiếp đến thành phần cắt xén đầu ra 549. Thành phần cắt xén đầu ra 549 được tạo cấu hình để áp dụng cửa sổ cắt xén phù hợp cho các ảnh 557. Điều này tạo ra các ảnh được cắt xén đầu ra 559. Ảnh được cắt xén đầu ra 559 là ảnh được khôi phục hoàn toàn. Theo đó, ảnh được cắt xén đầu ra 559 phản chiếu những gì người

dùng cuối sẽ nhìn thấy ngay khi giải mã dòng bit 551. Như vậy, bộ mã hóa có thể đánh giá các ảnh được cắt xén đầu ra 559 để đảm bảo việc mã hóa đạt yêu cầu.

HRD 500 được khởi tạo dựa vào các tham số HRD trong dòng bit 551. Ví dụ, HRD 500 có thể đọc các tham số HRD từ VPS, SPS và/hoặc các thông điệp SEI. Sau đó, HRD 500 có thể thực hiện các hoạt động kiểm tra tính phù hợp đối với dòng bit 551 dựa vào thông tin trong các tham số HRD này. Là một ví dụ cụ thể, HRD 500 có thể xác định một hoặc nhiều lịch biểu vận chuyển CPB từ các tham số HRD. Lịch biểu vận chuyển này chỉ định thời gian để vận chuyển dữ liệu video đến và/hoặc từ địa chỉ bộ nhớ, chẳng hạn như CPB và/hoặc DPB. Do đó, lịch biểu vận chuyển CPB chỉ định thời gian để vận chuyển các AU, các DU 553 và/hoặc các ảnh, đến/từ CPB 543. Cần lưu ý rằng HRD 500 có thể sử dụng các lịch biểu vận chuyển DPB đối với DPB 547 mà tương tự với các lịch biểu vận chuyển CPB.

Video có thể được mã hóa thành các lớp và/hoặc các OLS khác nhau để được sử dụng bởi các bộ giải mã với các mức độ tương thích phần cứng thay đổi cũng như đối với các điều kiện mạng thay đổi. Các lịch biểu vận chuyển CPB được chọn để phản ánh các vấn đề này. Theo đó, các dòng bit con lớp cao hơn được chỉ định cho các điều kiện mạng và phần cứng tối ưu và do đó các lớp cao hơn có thể nhận một hoặc nhiều lịch biểu vận chuyển CPB mà sử dụng lượng lớn bộ nhớ trong CPB 543 và các độ trễ ngắn để chuyển các DU 553 đến DPB 547. Tương tự như vậy, các dòng bit con lớp thấp hơn được chỉ định cho các khả năng phần cứng bộ giải mã giới hạn và/hoặc điều kiện mạng kém. Do đó, các lớp thấp hơn có thể nhận một hoặc nhiều lịch biểu vận chuyển CPB mà sử dụng lượng nhỏ bộ nhớ trong CPB 543 và các độ trễ dài hơn để chuyển các DU 553 đến DPB 547. Sau đó, các OLS, lớp, lớp con hoặc kết hợp của chúng có thể được kiểm tra theo lịch biểu vận chuyển tương ứng để đảm bảo rằng dòng bit con được tạo ra có thể được giải mã một cách chính xác dưới các điều kiện mà được dự tính cho dòng bit con. Theo đó, các tham số HRD trong dòng bit 551 có thể biểu thị các lịch biểu vận chuyển CPB cũng như bao gồm đủ dữ liệu để cho phép HRD 500 xác định các lịch biểu vận chuyển CPB và tạo tương quan các lịch biểu vận chuyển CPB với các OLS, lớp và/hoặc lớp con tương ứng.

Fig.6 là sơ đồ sơ lược minh họa chuỗi video đa lớp 600 làm ví dụ. Chuỗi video đa lớp 600 này có thể được mã hóa bởi bộ mã hóa, chẳng hạn như hệ thống mã hóa-giải mã 200 và/hoặc bộ mã hóa 300 và được giải mã bởi bộ giải mã, chẳng hạn như hệ thống mã hóa-giải mã 200 và/hoặc bộ giải mã 400, ví dụ theo phương pháp 100. Ngoài ra, chuỗi video đa lớp 600 có thể được kiểm tra về tính phù hợp tiêu chuẩn bởi HRD, chẳng hạn như HRD 500. Chuỗi video đa lớp 600 được bao gồm để mô tả ứng dụng làm ví dụ đối với các lớp trong chuỗi video được mã hóa. Chuỗi video đa lớp 600 này là chuỗi video bất kỳ mà sử dụng nhiều lớp, chẳng hạn như lớp N 631 và lớp N+1 632.

Trong một ví dụ, chuỗi video đa lớp 600 có thể sử dụng phép dự đoán liên lớp 621. Phép dự đoán liên lớp 621 này được áp dụng giữa các ảnh 611, 612, 613 và 614 và các ảnh 615, 616, 617 và 618 ở các lớp khác nhau. Trong một ví dụ được thể hiện, các ảnh 611, 612, 613 và 614 là một phần của lớp N+1 632 và các ảnh 615, 616, 617 và 618 là một phần của lớp N 631. Lớp, chẳng hạn như lớp N 631 và/hoặc lớp N+1 632, là nhóm gồm các ảnh mà đều được kết hợp với giá trị tương tự của đặc điểm, chẳng hạn như kích thước, chất lượng, độ phân giải, tỷ lệ tín hiệu trên nhiễu, dung lượng tương tự, v.v. Lớp có thể được xác định chính thức dưới dạng tập hợp các đơn vị VCL NAL mà chia sẻ cùng ID lớp và các đơn vị NAL không phải VCL được kết hợp. Đơn vị VCL NAL là đơn vị NAL được mã hóa để chứa dữ liệu video, chẳng hạn như lát được mã hóa của ảnh. Đơn vị NAL không phải VCL là đơn vị NAL mà chứa dữ liệu không phải video chẳng hạn như cú pháp và/hoặc các tham số để hỗ trợ việc giải mã dữ liệu video, hiệu suất của việc kiểm tra tính phù hợp hoặc các hoạt động khác.

Trong một ví dụ được thể hiện, lớp N+1 632 được kết hợp với kích thước hình ảnh lớn hơn so với lớp N 631. Theo đó, các ảnh 611, 612, 613 và 614 trong lớp N+1 632 có kích thước ảnh lớn hơn (ví dụ, chiều cao và chiều rộng lớn hơn và do đó có nhiều mẫu hơn) so với các ảnh 615, 616, 617 và 618 trong lớp N 631 trong ví dụ này. Tuy nhiên, các ảnh này có thể được tách rời giữa lớp N+1 632 và lớp N 631 bởi các đặc điểm khác. Trong khi chỉ hai lớp, lớp N+1 632 và lớp N 631, được thể hiện, tập hợp ảnh có thể được tách rời thành số lượng lớp bất kỳ dựa vào các đặc điểm đã kết hợp. Lớp N+1 632 và lớp N 631 có thể còn được biểu thị bởi ID lớp. ID lớp là mục của dữ liệu mà được kết hợp với ảnh và biểu thị ảnh này là một phần của lớp được biểu thị. Theo đó, mỗi ảnh 611-618 có thể được kết hợp với ID lớp tương ứng để biểu thị lớp N+1 632 hoặc lớp N 631 nào bao gồm ảnh tương ứng. Ví dụ, ID lớp có thể bao gồm mã nhận dạng lớp phần đầu đơn vị NAL (`nuh_layer_id`), mà là phần tử cú pháp mà chỉ định mã nhận dạng của lớp bao gồm đơn vị NAL (ví dụ, bao gồm các lát và/hoặc các tham số của các ảnh trong lớp). Lớp được kết hợp với chất lượng thấp hơn/kích thước hình ảnh nhỏ hơn/kích thước dòng bit nhỏ hơn, chẳng hạn như lớp N 631, thường được gán ID lớp thấp hơn và được gọi là lớp thấp hơn. Ngoài ra, lớp được kết hợp với chất lượng cao hơn/kích thước hình ảnh lớn hơn/dòng bit kích thước lớn hơn, chẳng hạn như lớp N+1 632, thường được gán ID lớp cao hơn và được gọi là lớp cao hơn.

Các ảnh 611-618 ở các lớp 631-632 khác nhau được tạo cấu hình để được hiển thị thay phiên nhau. Là một ví dụ cụ thể, bộ giải mã có thể giải mã và hiển thị ảnh 615 ở thời gian hiển thị hiện thời nếu mong muốn ảnh nhỏ hơn hoặc bộ giải mã có thể giải mã và hiển thị ảnh 611 ở thời gian hiển thị hiện thời nếu mong muốn ảnh lớn hơn. Như vậy, các ảnh 611-614 ở lớp cao hơn N+1 632 chứa dữ liệu hình ảnh gần giống với các ảnh 615-

618 tương ứng ở lớp thấp hơn N 631 (bất kể khác biệt về kích thước ảnh). Cụ thể là, ảnh 611 chứa dữ liệu hình ảnh gần giống với ảnh 615, ảnh 612 chứa dữ liệu hình ảnh gần giống với ảnh 616, v.v.

Các ảnh 611-618 có thể được mã hóa bằng cách tham chiếu đến các ảnh 611-618 khác ở cùng lớp N 631 hoặc N+1 632. Bước mã hóa ảnh tham chiếu đến ảnh khác ở cùng lớp dẫn đến phép dự đoán liên ảnh 623. Phép dự đoán liên ảnh 623 được mô tả bằng các mũi tên nét liền. Ví dụ, ảnh 613 có thể được mã hóa bằng cách sử dụng phép dự đoán liên ảnh 623 bằng cách sử dụng một hoặc hai trong số các ảnh 611, 612 và/hoặc 614 trong lớp N+1 632 làm tham chiếu, trong đó một ảnh được tham chiếu để dự đoán liên ảnh một hướng và/hoặc hai ảnh được tham chiếu để dự đoán liên ảnh hai hướng. Ngoài ra, ảnh 617 có thể được mã hóa bằng cách sử dụng phép dự đoán liên ảnh 623 bằng cách sử dụng một hoặc hai trong số các ảnh 615, 616 và/hoặc 618 trong lớp N 631 làm tham chiếu, trong đó một ảnh được tham chiếu để dự đoán liên ảnh một hướng và/hoặc hai hình ảnh được tham chiếu để dự đoán liên ảnh hai hướng. Khi ảnh được sử dụng làm tham chiếu cho ảnh khác ở cùng lớp khi thực hiện phép dự đoán liên ảnh 623, thì ảnh này có thể được gọi là ảnh tham chiếu. Ví dụ, ảnh 612 có thể ảnh tham chiếu được sử dụng để mã hóa ảnh 613 theo phép dự đoán liên ảnh 623. Phép dự đoán liên ảnh 623 có thể còn được gọi là phép dự đoán nội bộ lớp theo ngữ cảnh nhiều lớp. Như vậy, phép dự đoán liên ảnh 623 là cơ chế của các mẫu giải mã của ảnh hiện thời bằng cách tham chiếu đến các mẫu được biểu thị trong ảnh tham chiếu mà khác với ảnh hiện thời trong đó ảnh tham chiếu và ảnh hiện thời ở cùng lớp.

Các ảnh 611-618 có thể còn được mã hóa bằng cách tham chiếu đến các ảnh 611-618 khác ở các lớp khác nhau. Quy trình này được biết đến dưới dạng phép dự đoán liên lớp 621, và được mô tả bằng các mũi tên nét gạch. Phép dự đoán liên lớp 621 là cơ chế giải mã các mẫu của ảnh hiện thời bằng cách tham chiếu đến các mẫu được biểu thị trong ảnh tham chiếu trong đó ảnh hiện thời và ảnh tham chiếu ở các lớp khác nhau và do đó có các ID lớp khác nhau. Ví dụ, ảnh trong lớp thấp hơn N 631 có thể được sử dụng làm ảnh tham chiếu để mã hóa ảnh tương ứng ở lớp cao hơn N+1 632. Là một ví dụ cụ thể, ảnh 611 có thể được mã hóa bằng cách tham chiếu đến ảnh 615 theo phép dự đoán liên lớp 621. Trong trường hợp này, ảnh 615 được sử dụng làm ảnh tham chiếu liên lớp. Ảnh tham chiếu liên lớp là ảnh tham chiếu được sử dụng để dự đoán liên lớp 621. Trong hầu hết các trường hợp, phép dự đoán liên lớp 621 được ràng buộc sao cho ảnh hiện thời, chẳng hạn như ảnh 611, chỉ có thể sử dụng (các) ảnh tham chiếu liên lớp mà có trong cùng AU và ở lớp thấp hơn, chẳng hạn như ảnh 615. Khi nhiều lớp (ví dụ, nhiều hơn hai) là khả dụng, thì phép dự đoán liên lớp 621 có thể mã hóa/giải mã ảnh hiện thời dựa vào nhiều ảnh tham chiếu liên lớp ở các mức thấp hơn so với ảnh hiện thời.

Bộ mã hóa video có thể sử dụng chuỗi video đa lớp 600 để mã hóa các ảnh 611-618 qua nhiều kết hợp khác nhau và/hoặc các hoán vị của phép dự đoán liên ảnh 623 và dự đoán liên lớp 621. Ví dụ, ảnh 615 có thể được mã hóa theo phép dự đoán nội ảnh. Sau đó, các ảnh 616-618 có thể được mã hóa theo phép dự đoán liên ảnh 623 bằng cách sử dụng ảnh 615 làm ảnh tham chiếu. Ngoài ra, ảnh 611 có thể được mã hóa theo phép dự đoán liên lớp 621 bằng cách sử dụng ảnh 615 làm ảnh tham chiếu liên lớp. Sau đó, các ảnh 612-614 có thể được mã hóa theo phép dự đoán liên ảnh 623 bằng cách sử dụng ảnh 611 làm ảnh tham chiếu. Như vậy, ảnh tham chiếu có thể đóng vai trò làm cả ảnh tham chiếu một lớp duy nhất và ảnh tham chiếu liên lớp đối với các cơ chế mã hóa khác nhau. Bằng cách mã hóa các ảnh lớp cao hơn $N+1$ 632 dựa vào các ảnh lớp thấp hơn N 631, lớp cao hơn $N+1$ 632 có thể tránh sử dụng phép dự đoán nội ảnh, mà có hiệu quả mã hóa thấp hơn nhiều so với phép dự đoán liên ảnh 623 và dự đoán liên lớp 621. Như vậy, hiệu quả mã hóa kém của phép dự đoán nội ảnh có thể bị giới hạn ở các ảnh nhỏ nhất/có chất lượng thấp nhất, và do đó bị giới hạn để mã hóa lượng dữ liệu video nhỏ nhất. Các ảnh được sử dụng làm các ảnh tham chiếu và/hoặc các ảnh tham chiếu liên lớp có thể được biểu thị trong các mục của (các) danh sách ảnh tham chiếu có trong cấu trúc danh sách ảnh tham chiếu.

Các ảnh 611-618 có thể còn có trong các AU. AU là tập hợp ảnh được mã hóa mà có trong các lớp khác nhau và được kết hợp với cùng thời gian kết xuất trong quá trình giải mã. Để thảo luận rõ về các khái niệm được mô tả trong bản mô tả này, các AU bao gồm AU hiện thời 628 và AU đứng trước 627. AU hiện thời 628 là AU mà tương ứng với thời gian kết xuất cụ thể trong quy trình giải mã và/hoặc quy trình kiểm tra tính phù hợp HRD. AU đứng trước 627 là AU mà xuất hiện trước AU hiện thời 628 theo thứ tự giải mã. Ví dụ, AU đứng trước 627 có thể là AU ngay trước AU hiện thời 628 theo thứ tự giải mã.

Các ảnh được mã hóa trong cùng AU được lập lịch để kết xuất từ DPB ở bộ giải mã ở cùng thời điểm. Ví dụ, các ảnh 613 và 617 trong cùng AU hiện thời 628. Các ảnh 612 và 616 trong AU đứng trước 627 mà khác với AU hiện thời 628 bao gồm các ảnh 613 và 617. Các ảnh 612 và/hoặc 616 có thể được giải mã trước các ảnh 613 và/hoặc 617 vì AU đứng trước 627 đứng trước AU hiện thời 628 theo thứ tự giải mã. Ngoài ra, các ảnh 613 và 617 trong AU hiện thời 627 có thể được hiển thị khi thay thế. Ví dụ, ảnh 617 có thể được hiển thị khi kích thước ảnh nhỏ được mong muốn và ảnh 613 có thể được hiển thị khi kích thước ảnh lớn được mong muốn. Khi kích thước ảnh lớn được mong muốn, thì ảnh 613 được kết xuất và ảnh 617 chỉ được sử dụng để dự đoán liên lớp 621. Trong trường hợp này, ảnh 617 bị loại bỏ mà không được kết xuất một khi phép dự đoán liên lớp 621 được hoàn thành. Ngoài ra, trong sự kiện để chuyển đổi từ lớp N 631 sang lớp

N+1 632, ảnh 616 của AU đứng trước 627 có thể được kết xuất trong khi ảnh 613 từ AU hiện thời 628 có thể được kết xuất. Kiểu chuyển đổi này có thể xảy ra, ví dụ, khi người dùng tăng kích thước của đầu ra video (ví dụ, tăng kích thước của cửa sổ video hoặc chuyển đổi sang màn hình lớn hơn) và/hoặc khi các điều kiện mạng được cải thiện. Tương tự như vậy, trong sự kiện để chuyển đổi từ lớp N+1 632 sang lớp N 631, ảnh 612 của AU đứng trước 627 có thể được kết xuất trong khi ảnh 617 từ AU hiện thời 628 có thể được kết xuất. Kiểu chuyển đổi này có thể xảy ra, ví dụ, khi người dùng giảm kích thước của đầu ra video (ví dụ, giảm kích thước của cửa sổ video hoặc chuyển đổi sang màn hình nhỏ hơn) và/hoặc khi các điều kiện mạng được cải thiện.

AU có thể còn được chia thành một hoặc nhiều đơn vị ảnh (PU) 625. PU 625 là tập con của AU mà chứa một ảnh được mã hóa duy nhất. PU 625 có thể được xác định chính thức dưới dạng tập hợp đơn vị NAL mà được kết hợp với nhau theo quy tắc phân loại được chỉ định, là liên tiếp theo thứ tự giải mã, và chứa chính xác một ảnh được mã hóa. Cần lưu ý rằng PU 625 có thể được gọi là đơn vị giải mã (DU) khi được thảo luận về HRD và/hoặc các thử nghiệm tính phù hợp được kết hợp.

Fig.7 là sơ đồ sơ lược minh họa dòng bit 700 làm ví dụ. Ví dụ, dòng bit 700 có thể được tạo ra bởi hệ thống mã hóa-giải mã 200 và/hoặc bộ mã hóa 300 để giải mã bởi hệ thống mã hóa-giải mã 200 và/hoặc bộ giải mã 400 theo phương pháp 100. Ngoài ra, dòng bit 700 có thể bao gồm chuỗi video đa lớp 600. Ngoài ra, dòng bit 700 có thể bao gồm các tham số khác nhau để điều khiển hoạt động của HRD, chẳng hạn như HRD 500. Dựa vào các tham số này, HRD có thể kiểm tra dòng bit 700 để phù hợp với các tiêu chuẩn trước khi truyền về phía bộ giải mã để giải mã.

Dòng bit 700 bao gồm VPS 711, một hoặc nhiều SPS 713, các tập hợp tham số ảnh (picture parameter set, PPS) 715, các tập hợp tham số thích ứng (adaptation parameter set, APS) 716, các phần đầu ảnh 718, các phần đầu lát 717, dữ liệu ảnh 720 và các thông điệp SEI 719. VPS 711 chứa dữ liệu liên quan đến toàn bộ dòng bit 700. Ví dụ, VPS 711 có thể chứa OLS, các lớp, và/hoặc các lớp con liên quan đến dữ liệu được sử dụng trong dòng bit 700. SPS 713 chứa dữ liệu chuỗi chung cho tất cả các hình ảnh trong chuỗi video được mã hóa có trong dòng bit 700. Ví dụ, mỗi lớp có thể chứa một hoặc nhiều chuỗi video được mã hóa, và mỗi chuỗi video được mã hóa có thể tham chiếu SPS 713 cho các tham số tương ứng. Các tham số trong SPS 713 có thể bao gồm kích thước ảnh, độ sâu bit, các tham số công cụ mã hóa, các ràng buộc tốc độ bit, v.v. Cần lưu ý rằng, trong khi mỗi chuỗi tham chiếu đến SPS 713, một SPS 713 duy nhất có thể chứa dữ liệu cho nhiều chuỗi trong một số ví dụ. PPS 715 chứa các tham số áp dụng cho toàn bộ ảnh. Do đó, mỗi hình ảnh trong chuỗi video có thể tham chiếu đến PPS 715. Cần lưu ý rằng, trong khi mỗi ảnh tham chiếu đến PPS 715, một PPS 715 duy nhất có thể chứa dữ liệu cho nhiều ảnh

trong một số ví dụ. Ví dụ, nhiều ảnh tương tự có thể được mã hóa theo các tham số tương tự. Trong trường hợp này, một PPS 715 có thể chứa dữ liệu cho các hình ảnh tương tự này. PPS 715 có thể biểu thị các công cụ mã hóa khả dụng cho các lát trong các hình ảnh tương ứng, các tham số lượng tử hóa, các độ lệch, v.v.

APS 716 là cấu trúc cú pháp chứa các phần tử cú pháp/các tham số mà áp dụng cho một hoặc nhiều lát 727 trong một hoặc nhiều ảnh 725. Các tương quan này có thể được xác định dựa vào các phần tử cú pháp được tìm thấy trong các phần đầu lát 717 được kết hợp với các lát 727. Ví dụ, APS 716 có thể áp dụng cho ít nhất một, nhưng nhỏ hơn tất cả, các lát 727 trong ảnh thứ nhất 721, cho ít nhất một, nhưng nhỏ hơn tất cả, các lát 727 trong ảnh thứ hai 725, v.v. APS 716 có thể được tách rời thành nhiều kiểu dựa vào các tham số có trong APS 716. Các kiểu này có thể bao gồm APS bộ lọc vòng lặp thích ứng (ALF), ánh xạ độ chói với APS định tỷ lệ sắc độ (luma mapping with chroma scaling, LMCS) và APS danh sách định tỷ lệ (Scaling). ALF là bộ lọc dựa vào khối thích ứng bao gồm chức năng truyền được điều khiển bởi các tham số biến và sử dụng phản hồi từ vòng phản hồi để tinh chỉnh chức năng truyền. Ngoài ra, ALF được sử dụng để hiệu chỉnh các thành phần luma mã hóa (ví dụ, các sai số) mà xảy ra là kết quả của mã hóa dựa trên khối, chẳng hạn như các thành phần luma nhòe và răng cưa. Do đó, các tham số ALF có trong APS ALF có thể bao gồm các tham số được chọn bởi bộ mã hóa để khiến ALF loại bỏ các thành phần luma mã hóa dựa trên khối trong quá trình giải mã ở bộ giải mã. LMCS là quy trình mà được áp dụng dưới dạng một phần của quy trình giải mã mà ánh xạ các mẫu độ chói đến các giá trị cụ thể và trong một số trường hợp còn áp dụng hoạt động định tỷ lệ cho các giá trị của các mẫu sắc độ. Công cụ LMCS có thể định hình lại các thành phần độ chói dựa vào việc ánh xạ đến các thành phần sắc độ tương ứng để giảm bớt tỷ lệ méo. Do đó, LMCS APS bao gồm các tham số được chọn bởi bộ mã hóa khiến công cụ LMCS định hình lại các thành phần độ chói. APS danh sách định tỷ lệ chứa các tham số công cụ mã hóa được kết hợp với các ma trận lượng tử hóa được sử dụng bởi các bộ lọc quy định. Do đó, APS 716 có thể chứa các tham số được sử dụng để áp dụng các bộ lọc khác nhau cho các lát được mã hóa 727 đang thử nghiệm tính phù hợp ở HRD và/hoặc sau khi giải mã ở bộ giải mã.

Phần đầu ảnh 718 là cấu trúc cú pháp chứa các phần tử cú pháp để áp dụng cho tất cả các lát 727 của ảnh được mã hóa 725. Ví dụ, phần đầu ảnh 718 có thể chứa thông tin đếm thứ tự hình ảnh, dữ liệu ảnh tham chiếu, dữ liệu liên quan trong các ảnh điểm truy cập ngẫu nhiên nội ảnh (intra-random access point, IRAP), dữ liệu liên quan đến việc áp dụng bộ lọc đối với ảnh 725, v.v. PU có thể chứa chính xác một phần đầu ảnh 718 và chính xác một ảnh 725. Do đó, dòng bit 700 có thể bao gồm chính xác một phần đầu ảnh 718 trên mỗi ảnh 725. Phần đầu lát 717 chứa các tham số quy định cho mỗi lát 727 trong

ảnh 725. Do đó, có thể có một phần đầu lát 717 mỗi lát 727 trong chuỗi video. Phần đầu lát 717 có thể chứa thông tin kiểu lát, thông tin lọc, trọng số dự đoán, các điểm mục ô, các tham số khử nhiễu, v.v. Trong một số ví dụ, các phần tử cú pháp có thể là giống nhau đối với tất cả các lát 727 trong ảnh 725. Để giảm bớt phần dư, phần đầu ảnh 718 và phần đầu lát 717 có thể có cùng kiểu thông tin nhất định. Ví dụ, các tham số nhất định (ví dụ, các tham số lọc) có thể có trong phần đầu ảnh 718 khi chúng áp dụng cho toàn bộ ảnh 725 hoặc có trong phần đầu lát 717 khi chúng áp dụng cho nhóm gồm các lát 727 mà là tập hợp con gồm toàn bộ ảnh 725.

Dữ liệu ảnh 720 chứa dữ liệu video được mã hóa theo phép dự đoán liên ảnh và/hoặc dự đoán nội ảnh cũng như dữ liệu dư được biến đổi và được lượng tử hóa tương ứng. Ví dụ, dữ liệu ảnh 720 có thể bao gồm các lớp 723, ảnh 725 và/hoặc lát 727. Lớp 723 là tập hợp đơn vị VCL NAL mà có cùng đặc điểm được chỉ định (ví dụ, độ phân giải, tốc độ khung, kích thước hình ảnh chung, v.v.) như được biểu thị bởi ID lớp, chẳng hạn như `nuh_layer_id`, và các đơn vị NAL không phải VCL được kết hợp. Ví dụ, lớp 723 có thể bao gồm tập hợp ảnh 725 mà có cùng `nuh_layer_id`. Lớp 723 có thể gần như tương tự với các lớp 631, và/hoặc 632. `Nuh_layer_id` là phần tử cú pháp mà chỉ định mã nhận dạng của lớp 723 bao gồm ít nhất một đơn vị NAL. Ví dụ, lớp chất lượng thấp nhất 723, được biết đến dưới dạng lớp cơ sở, có thể bao gồm giá trị thấp nhất của `nuh_layer_id` có `nuh_layer_id` có giá trị tăng lên đối với các lớp 723 có chất lượng cao hơn. Do đó, lớp thấp hơn là lớp 723 có `nuh_layer_id` có giá trị nhỏ hơn và lớp cao hơn là lớp 723 có `nuh_layer_id` có giá trị lớn hơn.

Ảnh 725 là mảng gồm các mẫu độ chói và/hoặc mảng gồm các mẫu sắc độ mà tạo ra khung hoặc trường của nó. Ví dụ, ảnh 725 là ảnh được mã hóa mà có thể được kết xuất để hiển thị hoặc được sử dụng để hỗ trợ mã hóa (các) ảnh khác 725 để kết xuất. Ảnh 725 chứa một hoặc nhiều lát 727. Lát 727 có thể được xác định dưới dạng số nguyên các ô hoàn chỉnh hoặc số nguyên các hàng đơn vị cây mã hóa (CTU) hoàn chỉnh (ví dụ, trong ô) của ảnh 725 mà được chứa riêng trong một đơn vị NAL. Các lát 727 còn được chia thành các CTU và/hoặc các khối cây mã hóa (CTB). CTU là nhóm gồm các mẫu có kích thước định trước mà có thể được phân vùng bởi cây mã hóa. CTB là tập hợp con gồm CTU và chứa các thành phần độ chói hoặc các thành phần sắc độ của CTU. Các CTU/CTB còn được chia thành các khối mã hóa dựa vào các cây mã hóa. Sau đó, các khối mã hóa có thể được mã hóa/được giải mã theo các cơ chế dự đoán.

Thông điệp SEI 719 là cấu trúc cú pháp với các ngữ nghĩa được chỉ định mà truyền thông tin không cần thiết bằng quy trình giải mã để xác định các giá trị của các mẫu trong các ảnh được mã hóa. Ví dụ, các thông điệp SEI 719 có thể chứa dữ liệu để hỗ trợ các quy

trình HRD hoặc dữ liệu hỗ trợ khác không liên quan trực tiếp để giải mã dòng bit 700 ở bộ giải mã.

Các tham số khác nhau được báo hiệu trong dòng bit 700 để hỗ trợ giải mã các ảnh 725, các lát 727 và/hoặc các phân vùng khác của tham số này để khôi phục chuỗi video được giải mã. Ví dụ, SPS 713 có thể chứa tối đa chiều rộng ảnh SPS trong các mẫu độ chói (`sps_pic_width_max_in_luma_samples`) 731 và tối đa chiều cao ảnh SPS trong các mẫu độ chói (`sps_pic_height_max_in_luma_samples`) 732. `sps_pic_width_max_in_luma_samples` 731 là phần tử dữ liệu được báo hiệu để chỉ định chiều rộng tối đa, trong các đơn vị của các mẫu độ chói, của mỗi ảnh 725 được giải mã dựa vào SPS 713. `sps_pic_width_max_in_luma_samples` 732 là phần tử dữ liệu được báo hiệu để chỉ định chiều cao tối đa, trong các đơn vị của các mẫu độ chói, của mỗi ảnh 725 được giải mã dựa vào SPS 713. Theo đó, `sps_pic_width_max_in_luma_samples` 731 và `sps_pic_width_max_in_luma_samples` 732 lần lượt chỉ định chiều rộng và chiều cao tối đa, của mỗi ảnh 725 trong chuỗi video được mã hóa liên quan đến SPS 713.

Như được lưu ý ở trên, bộ mã hóa có thể truyền các lớp 723 khác nhau của các ảnh 725 đến bộ giải mã tùy thuộc vào các điều kiện ở bộ giải mã. Hơn nữa, các lớp 723 đã truyền có thể thay đổi khi các điều kiện ở bộ giải mã thay đổi. Ví dụ, các ảnh 725 có thể được truyền ở độ phân giải cao hơn/kích thước lớn hơn khi các điều kiện mạng là tối ưu và được truyền ở độ phân giải thấp hơn/kích thước thấp hơn khi các điều kiện mạng xấu đi. Một vấn đề với phương án thực hiện này đó là DPB ở bộ giải mã có thể tràn trong quá trình lớp 723 thay đổi, mà có thể gây ra trải nghiệm kém cho người dùng. Ví dụ, DPB có thể chứa đủ các ảnh 725 ở độ phân giải thấp hơn trước khi lớp 723 thay đổi thành độ phân giải cao hơn. Như vậy, DPB có thể không có đủ không gian để lưu trữ các ảnh 725 có độ phân giải cao hơn nhận được từ lớp cao hơn 723 sau khi chuyển đổi. Điều này có thể khiến DPB tràn, mà dẫn đến sai số và ngăn việc giải mã và hiển thị thích hợp các ảnh 725 có độ phân giải cao hơn.

Không kết xuất cờ các ảnh trước (`NoOutputOfPriorPicsFlag`) có thể được sử dụng để giải quyết vấn đề này. `NoOutputOfPriorPicsFlag` là biến mà được suy ra ở bộ giải mã và/hoặc được suy ra ở HRD trong quá trình kiểm tra tính phù hợp. `NoOutputOfPriorPicsFlag` chỉ định khi DPB nên được làm trống mà không kết xuất các ảnh 725 có trong DPB. Theo đó, `NoOutputOfPriorPicsFlag` có thể được thiết lập để khiến làm trống DPB để tạo ra chỗ cho các ảnh mới 725. Điều này có thể xảy ra ngay cả khi ảnh 725 đứng trước trong DPB được chỉ định để kết xuất. Trong một số hệ thống mã hóa video, `NoOutputOfPriorPicsFlag` được thiết lập dựa vào sự thay đổi kích thước của ảnh 725 mà được phát hiện do sự thay đổi về chiều rộng ảnh và/hoặc chiều cao ảnh giữa ảnh 725 hiện thời và ảnh 725 đứng trước.

ARC là cơ chế để hỗ trợ với các thay đổi độ phân giải. Nói chung, bộ giải mã hóa đổi giữa các chuỗi và/hoặc các lớp 723 ở các ảnh IRAP. Điều này là vì ảnh IRAP có thể được giải mã mà không tham chiếu đến các ảnh 725 khác, và do đó có thể luôn được giải mã. Các ảnh không phải IRAP được mã hóa dựa vào các ảnh 725 khác, và do đó bộ giải mã có thể không có đủ thông tin để giải mã ảnh không phải IRAP dưới dạng ảnh thứ nhất 725 trong chuỗi và/hoặc lớp 723. ARC có thể hỗ trợ các quá trình chuyển đổi như vậy. ARC có thể thay đổi động độ phân giải của các ảnh 725 trong DPB để khớp với độ phân giải của các ảnh 725 tiếp theo. Theo đó, ảnh 725 tham chiếu trong DPB có thể được thay đổi từ độ phân giải thứ nhất sang độ phân giải thứ hai sao cho ảnh không phải IRAP ở độ phân giải thứ hai có thể được giải mã dựa vào ảnh 725 tham chiếu mà không yêu cầu ảnh IRAP xen kẽ. Vấn đề với ARC đó là ARC có thể ngăn NoOutputOfPriorPicsFlag không hoạt động phù hợp. Cụ thể, ARC có thể thay đổi độ phân giải về không gian của ảnh 725 đứng trước giống với ảnh 725 hiện thời, mà ngăn việc kiểm tra đối với sự thay đổi về kích thước ảnh không hoạt động chính xác. Như vậy, ARC có thể dẫn đến tràn DPB bằng cách vô hiệu hóa một cách hiệu quả việc sử dụng NoOutputOfPriorPicsFlag.

Dòng bit 700 được tạo cấu hình để cho phép NoOutputOfPriorPicsFlag được kích hoạt một cách chính xác khi ARC được sử dụng. Ví dụ, bộ giải mã có thể kiểm tra đối với sự thay đổi về chiều cao ảnh tối đa và chiều rộng ảnh tối đa thay vì kiểm tra đối với chiều cao và chiều rộng ảnh. Chiều cao ảnh tối đa và chiều rộng ảnh tối đa không bị thay đổi bởi ARC. Theo đó, NoOutputOfPriorPicsFlag có thể thiết lập thích hợp để ngăn sự tràn DPB do các thay đổi độ phân giải về không gian ngay cả khi ARC được sử dụng trong quá trình thay đổi lớp 723.

Trong một ví dụ cụ thể, bộ giải mã có thể sử dụng biến PicWidthMaxInSamplesY và biến PicHeightMaxInSamplesY khi thiết lập NoOutputOfPriorPicsFlag. PicWidthMaxInSamplesY là biến thu được mà biểu thị chiều rộng tối đa, trong các đơn vị của các mẫu độ chói, của mỗi ảnh 725 được giải mã trong chuỗi và/hoặc lớp tương ứng 723. PicHeightMaxInSamplesY là biến thu được mà biểu thị chiều cao tối đa, trong các đơn vị của các mẫu độ chói, của mỗi ảnh 725 được giải mã trong chuỗi và/hoặc lớp 723 tương ứng. PicWidthMaxInSamplesY có thể được suy ra dựa vào sps_pic_width_max_in_luma_samples 731 trong SPS 713. Hơn nữa, PicHeightMaxInSamplesY có thể được suy ra dựa vào sps_pic_height_max_in_luma_samples 732 trong SPS 713.

Sau đó, NoOutputOfPriorPicsFlag có thể thiết lập dựa vào PicHeightMaxInSamplesY và/hoặc PicWidthMaxInSamplesY. Ví dụ, NoOutputOfPriorPicsFlag có thể thiết lập khi giá trị của PicWidthMaxInSamplesY dùng cho AU (bao gồm ảnh 725 hiện thời) khác với giá trị của PicWidthMaxInSamplesY

dùng cho AU đứng trước (bao gồm ảnh 725 đứng trước) theo thứ tự giải mã. Trong một ví dụ khác, NoOutputOfPriorPicsFlag có thể thiết lập khi giá trị của PicHeightMaxInSamplesY dùng cho AU (bao gồm ảnh 725 hiện thời) khác với giá trị của PicHeightMaxInSamplesY dùng cho AU đứng trước (bao gồm ảnh 725 đứng trước) theo thứ tự giải mã.

Một khi NoOutputOfPriorPicsFlag được thiết lập, DPB có thể được làm trống khi cần để ngăn DPB tràn. Ví dụ, DPB có thể được làm trống sau khi phân tích cú pháp phần đầu lát 717 của lát thứ nhất 727 của ảnh 725 hiện thời và trước khi giải mã ảnh 725 hiện thời. Điều này làm trống ảnh 725 đứng trước từ AU đứng trước từ DPB trước khi ảnh 725 hiện thời được giải mã, và do đó trước khi cần không gian lưu trữ trong DPB đối với ảnh 725 hiện thời.

Trong một ví dụ khác, danh sách ảnh tham chiếu có thể được xây dựng cho lát 727 và quá trình đánh dấu danh sách ảnh tham chiếu có thể được thực hiện trước khi thiết lập NoOutputOfPriorPicsFlag. Quá trình xây dựng danh sách ảnh tham chiếu có thể xác định các ảnh 725 nào là các ảnh tham chiếu cho ảnh 725 hiện thời. Sau đó, quá trình đánh dấu danh sách ảnh tham chiếu có thể biểu thị xem các ảnh 725 trong DPB chưa được sử dụng để tham chiếu, được sử dụng để tham chiếu ngắn hạn và/hoặc được sử dụng để tham chiếu dài hạn hay không dựa vào quá trình xây dựng danh sách ảnh tham chiếu. Trong một ví dụ cụ thể, NoOutputOfPriorPicsFlag được thiết lập và DPB được làm trống khi mỗi trong số các ảnh 725 đứng trước trong DPB được đánh dấu là chưa được sử dụng để tham chiếu. Theo cách này, DPB không được làm trống nếu DPB chứa các ảnh tham chiếu. Hơn nữa, DPB có thể sử dụng biến lấp đầy DPB, mà là phần tử dữ liệu mà biểu thị số lượng ảnh được lưu trữ trong DPB. Biến lấp đầy DPB có thể được giảm bớt đi một đối với mỗi ảnh được loại bỏ khi làm trống DPB.

Bằng cách sử dụng các ví dụ được mô tả trong bản mô tả này, NoOutputOfPriorPicsFlag có thể thiết lập để sử dụng DPB khi cần để ngăn DPB tràn. Do đó, các phương án có trong bản mô tả này có thể được sử dụng để tránh các sai số mã hóa khác nhau liên quan đến việc chuyển đổi giữa các lớp 723. Quy trình tương tự có thể còn được sử dụng bởi HRD được vận hành ở bộ mã hóa khi kiểm tra dòng bit được mã hóa về tính phù hợp với các tiêu chuẩn. Quy trình này có thể ngăn HRD không bị tràn DPB khi kiểm tra tính phù hợp của các lớp 723, và do đó giảm sự xuất hiện của các lỗi kiểm tra tính phù hợp trong quá trình mã hóa. Kết quả là, chức năng của bộ mã hóa và bộ giải mã được tăng lên. Hơn nữa, hiệu quả mã hóa có thể được tăng lên, điều này giảm bớt mức sử dụng tài nguyên báo hiệu của bộ xử lý, bộ nhớ và/hoặc mạng ở cả bộ mã hóa và bộ giải mã.

Thông tin nêu trên hiện được mô tả một cách chi tiết hơn dưới đây. Mã hóa video phân lớp còn được gọi là mã hóa video định tỷ lệ được hoặc mã hóa video có khả năng mở rộng. Khả năng mở rộng trong mã hóa video có thể được hỗ trợ bằng cách sử dụng các kỹ thuật mã hóa đa lớp. Dòng bit đa lớp bao gồm lớp cơ sở (base layer, BL) và một hoặc nhiều các lớp tăng cường (enhancement layer, EL). Ví dụ về khả năng mở rộng bao gồm khả năng mở rộng theo không gian, khả năng mở rộng theo chất lượng/định tỷ lệ tín hiệu trên nhiễu (SNR), khả năng mở rộng đa khung nhìn, khả năng mở rộng tốc độ khung, v.v. Khi kỹ thuật mã hóa đa lớp được sử dụng, thì ảnh hoặc một phần của ảnh có thể được mã hóa mà không sử dụng ảnh tham chiếu (dự đoán nội ảnh), có thể được mã hóa bằng cách tham chiếu đến các ảnh tham chiếu mà ở cùng lớp (dự đoán liên ảnh), và/hoặc có thể được mã hóa bằng cách tham chiếu đến các ảnh tham chiếu mà ở (các) lớp khác (dự đoán liên lớp). Ảnh tham chiếu được sử dụng cho phép dự đoán liên lớp của ảnh hiện thời được gọi là ảnh tham chiếu liên lớp (ILRP). Fig.6 minh họa ví dụ về bước mã hóa đa lớp cho khả năng mở rộng theo không gian trong đó các hình ảnh ở các lớp khác nhau có các độ phân giải khác nhau.

Một số hệ mã hóa video cung cấp hỗ trợ cho khả năng mở rộng trong (các) hồ sơ được tách rời từ (các) hồ sơ để mã hóa một-lớp duy nhất. Mã hóa video định tỷ lệ được (SVC) là phần mở rộng định tỷ lệ được của mã hóa video nâng cao (AVC) cung cấp hỗ trợ cho khả năng mở rộng không gian, thời gian và chất lượng. Đối với SVC, cờ được báo hiệu trong mỗi khối macrô (macroblock, MB) trong các hình ảnh EL để biểu thị xem EL MB có được dự đoán bằng cách sử dụng khối được sắp đặt từ lớp thấp hơn hay không. Phép dự đoán từ khối được sắp đặt có thể bao gồm kết cấu, các vectơ chuyển động, và/hoặc các chế độ giải mã. Các phương án thực hiện của SVC có thể không sử dụng lại trực tiếp cách thực hiện AVC chưa được cải biến về thiết kế của chúng. Cú pháp và quy trình giải mã của EL SVC khác với cú pháp và quy trình giải mã của AVC.

HEVC định tỷ lệ được (SHVC) là phần mở rộng của HEVC mà tạo ra hỗ trợ cho khả năng mở rộng theo không gian và chất lượng. HEVC đa khung nhìn (MV-HEVC) là phần mở rộng của HEVC cung cấp hỗ trợ cho khả năng mở rộng đa khung nhìn. HEVC 3D (3D-HEVC) là phần mở rộng của HEVC mà tạo ra hỗ trợ cho bước mã hóa video 3D mà cao cấp hơn và hiệu quả hơn so với MV-HEVC. Khả năng mở rộng theo thời gian có thể được bao gồm dưới dạng một phần tích hợp của bộ mã hóa/giải mã HEVC đơn lớp. Trong phần mở rộng đa lớp của HEVC, các ảnh được mã hóa được sử dụng để dự đoán liên lớp chỉ đến từ cùng AU và được coi là các ảnh tham chiếu dài hạn (long-term reference picture, LTRP). Các hình ảnh này được gán các chỉ số tham chiếu trong (các) danh sách ảnh tham chiếu cùng với các ảnh tham chiếu thời gian khác trong lớp hiện thời. Phép dự đoán liên lớp (ILP) đạt được ở mức đơn vị dự đoán bằng cách thiết lập giá trị của chỉ số

tham chiếu để tham chiếu đến (các) ảnh tham chiếu liên lớp trong (các) danh sách ảnh tham chiếu. Khả năng mở rộng không gian lấy mẫu lại ảnh tham chiếu hoặc một phần của nó khi ILRP có độ phân giải về không gian khác với ảnh hiện thời đang được mã hóa hoặc giải mã. Việc lấy lại mẫu ảnh tham chiếu có thể được thực hiện ở mức độ hình ảnh hoặc mức độ khối mã hóa.

VVC có thể còn hỗ trợ mã hóa video phân lớp. Dòng bit VVC có thể bao gồm nhiều lớp. Các lớp đều có thể độc lập với nhau. Ví dụ, mỗi lớp có thể được mã hóa mà không sử dụng phép dự đoán liên lớp. Trong trường hợp này, các lớp còn được gọi là các lớp phát đồng thời. Trong một số trường hợp, một số lớp được mã hóa bằng cách sử dụng ILP. Còn trong VPS có thể biểu thị xem các lớp có phải là các lớp phát đồng thời hay không hoặc xem một số lớp có sử dụng ILP hay không. Khi một số lớp sử dụng ILP, thì mối quan hệ phụ thuộc lớp giữa các lớp còn được báo hiệu trong VPS. Không giống như SHVC và MV-HEVC, VVC có thể không chỉ định các OLS. OLS bao gồm tập hợp gồm các lớp đã chỉ định, trong đó một hoặc nhiều lớp trong tập hợp các lớp được chỉ định cho các lớp đầu ra. Lớp đầu ra là lớp của OLS mà được kết xuất. Theo một số phương án thực hiện của VVC, chỉ một lớp có thể được chọn để giải mã và kết xuất khi các lớp là các lớp phát đồng thời. Theo một số phương án thực hiện của VVC, toàn bộ dòng bit bao gồm tất cả các lớp được chỉ định để được giải mã khi lớp bất kỳ sử dụng ILP. Ngoài ra, một số lớp nhất định trong số các lớp được chỉ định cho các lớp đầu ra. Các lớp đầu ra có thể được biểu thị là chỉ lớp cao nhất, tất cả các lớp, hoặc lớp cao nhất cộng tập hợp gồm các lớp thấp hơn được biểu thị.

Các khía cạnh nêu trên chứa các vấn đề nhất định. Ví dụ, các giá trị của `nuh_layer_id` cho các đơn vị SPS NAL, PPS và APS có thể không được ràng buộc phù hợp. Ngoài ra, giá trị của `TemporalId` đối với các đơn vị SEI NAL có thể không được ràng buộc phù hợp. Ngoài ra, việc thiết lập `NoOutputOfPriorPicsFlag` có thể không được chỉ định phù hợp khi việc lấy mẫu lại ảnh tham chiếu được cho phép và các ảnh nằm trong CLVS có các độ phân giải không gian khác nhau. Ngoài ra, trong một số hệ thống mã hóa video, các thông điệp SEI hậu tố không thể được chứa trong thông điệp SEI lồng có khả năng mở rộng. Là một ví dụ khác, các thông điệp SEI chu kỳ đệm, thời gian hình ảnh, và thông tin đơn vị giải mã có thể bao gồm các phụ thuộc phân tích cú pháp trên VPS và/hoặc SPS.

Nói chung, sáng chế mô tả các cách tiếp cận cải thiện việc mã hóa video. Các phần mô tả về các kỹ thuật dựa vào VVC. Tuy nhiên, các kỹ thuật cũng áp dụng cho mã hóa video phân lớp dựa vào các đặc tả mã hóa-giải mã video khác.

Một hoặc nhiều trong số các vấn đề nêu trên có thể được giải quyết như sau. Các giá trị của `nuh_layer_id` đối với các đơn vị SPS NAL, PPS và APS được ràng buộc phù

hợp ở đây. Giá trị của TemporalId đối với các đơn vị SEI NAL được ràng buộc phù hợp ở đây. Việc thiết lập NoOutputOfPriorPicsFlag được chỉ định phù hợp khi việc lấy mẫu lại ảnh tham chiếu được cho phép và các ảnh trong CLVS có các độ phân giải không gian khác nhau. Các thông điệp SEI hậu tố được phép có trong thông điệp SEI lồng có khả năng mở rộng. Các phụ thuộc phân tích cú pháp của các thông điệp BP, PT và DUI SEI trên VPS hoặc SPS có thể được loại bỏ bằng cách lặp lại phần tử cú pháp `decoding_unit_hrd_params_present_flag` trong cú pháp thông điệp BP SEI, các phần tử cú pháp `decoding_unit_hrd_params_present_flag` và `decoding_unit_cpb_params_in_pic_timing_sei_flag` trong cú pháp thông điệp PT SEI, và phần tử cú pháp `decoding_unit_cpb_params_in_pic_timing_sei_flag` trong thông điệp DUI SEI.

Một phương án thực hiện làm ví dụ về các cơ chế nêu trên là như sau. Ngữ nghĩa đơn vị NAL chung làm ví dụ là như sau.

`nuh_temporal_id_plus1` trừ 1 chỉ định mã nhận dạng theo thời gian dùng cho đơn vị NAL. Giá trị của `nuh_temporal_id_plus1` không nên bằng không. Biến TemporalId có thể được suy ra như sau:

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

Khi `nal_unit_type` nằm trong khoảng từ `IDR_W_RADL` đến `RSV_IRAP_13`, bao gồm, TemporalId nên bằng không. Khi `nal_unit_type` bằng `STSA_NUT`, TemporalId không nên bằng không.

Giá trị của TemporalId cần giống nhau đối với tất cả các đơn vị VCL NAL của đơn vị truy cập. Giá trị của TemporalId của ảnh được mã hóa, đơn vị truy cập lớp, hoặc đơn vị truy cập có thể là giá trị của TemporalId của các đơn vị VCL NAL của ảnh được mã hóa, đơn vị truy cập lớp hoặc đơn vị truy cập. Giá trị của TemporalId của biểu diễn lớp con có thể là giá trị lớn nhất của TemporalId của tất cả các đơn vị VCL NAL trong biểu diễn lớp con.

Giá trị của TemporalId đối với các đơn vị NAL không phải VCL được ràng buộc như sau. Nếu `nal_unit_type` bằng `DPS_NUT`, `VPS_NUT` hoặc `SPS_NUT`, thì TemporalId bằng không và TemporalId của đơn vị truy cập chứa đơn vị NAL nên bằng không. Nếu không thì, nếu `nal_unit_type` bằng `EOS_NUT` hoặc `EOB_NUT`, thì TemporalId nên bằng không. Nếu không thì, nếu `nal_unit_type` bằng `AUD_NUT`, `FD_NUT`, `PREFIX_SEI_NUT` hoặc `SUFFIX_SEI_NUT`, thì TemporalId cần bằng TemporalId của đơn vị truy cập chứa đơn vị NAL. Nếu không thì, khi `nal_unit_type` bằng `PPS_NUT` hoặc `APS_NUT`, thì TemporalId cần lớn hơn hoặc bằng TemporalId của đơn vị truy cập chứa đơn vị NAL. Khi đơn vị NAL là đơn vị NAL không phải VCL, thì giá trị của TemporalId

cần bằng giá trị tối thiểu của các giá trị của TemporalId của tất cả các đơn vị truy cập mà áp dụng đơn vị NAL không phải VCL. Khi nal_unit_type bằng PPS_NUT hoặc APS_NUT, TemporalId có thể lớn hơn hoặc bằng TemporalId của đơn vị truy cập chứa. Điều này là vì tất cả các PPS và các APS có thể có trong phần khởi đầu của dòng bit. Ngoài ra, ảnh được mã hóa thứ nhất có TemporalId bằng không.

Ngữ nghĩa RBSP tập hợp tham số chuỗi làm ví dụ là như sau. SPS RBSP cần khả dụng đối với quy trình giải mã trước khi được tham chiếu. SPS có thể có trong ít nhất một đơn vị truy cập với TemporalId bằng không hoặc được cung cấp thông qua cơ chế bên ngoài. Đơn vị SPS NAL chứa SPS có thể được ràng buộc để có nuh_layer_id bằng giá trị của nuh_layer_id thấp nhất của các đơn vị PPS NAL mà tham chiếu đến SPS.

Ngữ nghĩa RBSP tập hợp tham số ảnh làm ví dụ là như sau. PPS RBSP cần khả dụng đối với quy trình giải mã trước khi được tham chiếu. PPS cần có trong ít nhất một đơn vị truy cập với TemporalId nhỏ hơn hoặc bằng TemporalId của đơn vị PPS NAL hoặc được cung cấp thông qua cơ chế bên ngoài. Đơn vị PPS NAL chứa PPS RBSP cần có nuh_layer_id bằng giá trị của nuh_layer_id thấp nhất của các đơn vị NAL lát được mã hóa mà tham chiếu đến PPS.

Ngữ nghĩa tập hợp tham số thích ứng làm ví dụ là như sau. Mỗi APS RBSP cần khả dụng đối với quy trình giải mã trước khi được tham chiếu. APS cũng cần có trong ít nhất một đơn vị truy cập với TemporalId nhỏ hơn hoặc bằng TemporalId của đơn vị NAL lát được mã hóa mà tham chiếu đến APS hoặc được cung cấp thông qua cơ chế bên ngoài. Đơn vị APS NAL được phép để được chia sẻ bởi các ảnh/các lát của nhiều lớp. nuh_layer_id của đơn vị APS NAL cần bằng giá trị của nuh_layer_id thấp nhất của các đơn vị NAL lát được mã hóa mà tham chiếu đến đơn vị APS NAL. Theo cách khác, đơn vị APS NAL có thể không được chia sẻ bởi các ảnh/các lát của nhiều lớp. nuh_layer_id của đơn vị APS NAL cần bằng nuh_layer_id của các lát tham chiếu đến APS.

Trong một ví dụ, việc loại bỏ các ảnh khỏi DPB trước khi giải mã ảnh hiện thời được thảo luận như sau. Việc loại bỏ các ảnh khỏi DPB trước khi giải mã ảnh hiện thời (nhưng sau khi phân tích cú pháp phần đầu lát của lát thứ nhất của ảnh hiện thời) có thể xảy ra ở thời gian loại bỏ CPB của đơn vị giải mã thứ nhất của đơn vị truy cập n (chứa ảnh hiện thời). Điều này diễn ra như sau. Quy trình giải mã để xây dựng danh sách ảnh tham chiếu được gọi ra và quy trình giải mã để đánh dấu ảnh tham chiếu được gọi ra.

Khi ảnh hiện thời là ảnh bắt đầu chuỗi video lớp được mã hóa (coded layer video sequence start, CLVSS) mà không phải ảnh không, thì các bước theo thứ tự sau đây được áp dụng. Biến NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang thử nghiệm như sau. Nếu giá trị của pic_width_max_in_luma_samples,

pic_height_max_in_luma_samples, chroma_format_idc, separate_colour_plane_flag, bit_depth_luma_minus8, bit_depth_chroma_minus8 hoặc sps_max_dec_pic_buffering_minus1[Htid] lần lượt được suy ra từ SPS khác với giá trị của pic_width_in_luma_samples, pic_height_in_luma_samples, chroma_format_idc, separate_colour_plane_flag, bit_depth_luma_minus8, bit_depth_chroma_minus8 hoặc sps_max_dec_pic_buffering_minus1[Htid], được suy ra từ SPS được tham chiếu đến bởi ảnh đứng trước, NoOutputOfPriorPicsFlag có thể được thiết lập thành một bởi bộ giải mã đang thử nghiệm, bất kể giá trị của no_output_of_prior_pics_flag. Cần lưu ý rằng, mặc dù việc thiết lập NoOutputOfPriorPicsFlag bằng no_output_of_prior_pics_flag có thể được ưu tiên dưới các điều kiện này, nhưng bộ giải mã đang thử nghiệm được phép thiết lập NoOutputOfPriorPicsFlag thành một trong trường hợp này. Nếu không thì, NoOutputOfPriorPicsFlag có thể được thiết lập bằng no_output_of_prior_pics_flag.

Giá trị của NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang thử nghiệm được áp dụng cho HRD, sao cho khi giá trị của NoOutputOfPriorPicsFlag bằng 1, tất cả các bộ đệm lưu trữ ảnh trong DPB được làm trống mà không kết xuất các ảnh mà chúng chứa, và độ đầy DPB được thiết lập bằng không. Khi cả hai điều kiện sau là đúng đối với các ảnh k bất kỳ trong DPB, thì tất cả các hình ảnh k này trong DPB này được loại bỏ khỏi DPB. Ảnh k được đánh dấu là chưa được sử dụng để tham chiếu, và ảnh k có PictureOutputFlag bằng không hoặc thời gian kết xuất DPB tương ứng nhỏ hơn hoặc bằng thời gian loại bỏ CPB của đơn vị giải mã thứ nhất (được biểu thị dưới dạng đơn vị giải mã m) của ảnh hiện thời n. Điều này có thể xảy ra khi DpbOutputTime[k] nhỏ hơn hoặc bằng DuCpbRemovalTime[m]. Đối với mỗi ảnh mà được loại bỏ khỏi DPB, độ đầy DPB giảm đi một.

Trong một ví dụ, việc kết xuất và việc loại bỏ các ảnh khỏi DPB được thảo luận như sau. Việc kết xuất và loại bỏ các ảnh khỏi DPB trước khi giải mã ảnh hiện thời (nhưng sau khi phân tích cú pháp phân đầu lát của lát thứ nhất của ảnh hiện thời) có thể xảy ra khi đơn vị giải mã thứ nhất của đơn vị truy cập chứa ảnh hiện thời được loại bỏ khỏi CPB và diễn ra như sau. Quy trình giải mã để xây dựng danh sách ảnh tham chiếu và quy trình giải mã để đánh dấu ảnh tham chiếu được gọi ra.

Nếu ảnh hiện thời là ảnh CLVSS mà không phải ảnh không, các bước theo thứ tự sau đây được áp dụng. Biến NoOutputOfPriorPicsFlag có thể được suy ra cho bộ giải mã đang thử nghiệm như sau. Nếu giá trị của pic_width_max_in_luma_samples, pic_height_max_in_luma_samples, chroma_format_idc, separate_colour_plane_flag, bit_depth_luma_minus8, bit_depth_chroma_minus8 hoặc sps_max_dec_pic_buffering_minus1[Htid] lần lượt được suy ra từ SPS khác với giá trị của pic_width_in_luma_samples, pic_height_in_luma_samples, chroma_format_idc,

`separate_colour_plane_flag`, `bit_depth_luma_minus8`, `bit_depth_chroma_minus8` hoặc `sps_max_dec_pic_buffering_minus1[ Htid ]`, được suy ra từ SPS được tham chiếu đến bởi ảnh đứng trước, `NoOutputOfPriorPicsFlag` có thể được thiết lập thành một bởi bộ giải mã đang thử nghiệm, bất kể giá trị của `no_output_of_prior_pics_flag`. Cần lưu ý rằng mặc dù việc thiết lập `NoOutputOfPriorPicsFlag` bằng `no_output_of_prior_pics_flag` được ưu tiên dưới các điều kiện này, bộ giải mã đang thử nghiệm có thể thiết lập `NoOutputOfPriorPicsFlag` là một trong trường hợp này. Nếu không thì, `NoOutputOfPriorPicsFlag` có thể được thiết lập bằng `no_output_of_prior_pics_flag`.

Giá trị của `NoOutputOfPriorPicsFlag` được suy ra cho bộ giải mã đang thử nghiệm có thể được áp dụng cho HRD như sau. Nếu `NoOutputOfPriorPicsFlag` bằng một, thì tất cả các bộ đệm lưu trữ ảnh trong DPB được làm trống mà không kết xuất các ảnh mà chúng chứa và độ đầy DPB được thiết lập bằng không. Nếu không thì (`NoOutputOfPriorPicsFlag` bằng không), tất cả các bộ đệm lưu trữ ảnh chứa hình ảnh mà được đánh dấu là không cần thiết để kết xuất và chưa được sử dụng để tham chiếu được làm trống (không có đầu ra) và tất cả các bộ đệm lưu trữ ảnh không trống trong DPB được làm trống bằng cách gọi ra lặp lại quy trình va chạm và độ đầy DPB được thiết lập thành không.

Nếu không thì (ảnh hiện thời không là ảnh CLVSS), tất cả các bộ đệm lưu trữ ảnh chứa ảnh mà được đánh dấu là không cần thiết để kết xuất và chưa được sử dụng để tham chiếu được làm trống (không kết xuất). Đối với mỗi bộ đệm lưu trữ ảnh mà được làm trống, độ đầy DPB giảm đi một. Khi một hoặc nhiều trong số các điều kiện sau là đúng, thì quy trình va chạm được gọi ra lặp lại trong khi giảm bớt thêm độ đầy DPB đi một đối với mỗi bộ đệm lưu trữ ảnh bổ sung mà được làm trống cho đến khi không còn điều kiện nào sau đây là đúng. Điều kiện là số lượng của các ảnh trong DPB mà được đánh dấu là cần thiết để kết xuất lớn hơn `sps_max_num_reorder_pics[ Htid ]`. Điều kiện khác là `sps_max_latency_increase_plus1[ Htid ]` không bằng không và có ít nhất một ảnh trong DPB mà được đánh dấu là cần thiết để kết xuất để biến được kết hợp `PicLatencyCount` lớn hơn hoặc bằng `SpsMaxLatencyPictures[ Htid ]`. Điều kiện khác là số lượng của các ảnh trong DPB lớn hơn hoặc bằng `SubDpbSize[ Htid ]`.

Cú pháp thông điệp SEI chung làm ví dụ là như sau.

<code>sei_payload(payloadType, payloadSize) {</code>	Ký hiệu mô tả
<code>if(nal_unit_type == PREFIX_SEI_NUT)</code>	
<code>if(payloadType == 0)</code>	
<code>buffering_period(payloadSize)</code>	
<code>else if(payloadType == 1)</code>	

pic_timing(payloadSize)	
else if(payloadType == 3)	
filler_payload(payloadSize)	
else if(payloadType == 130)	
decoding_unit_info(payloadSize)	
else if(payloadType == 133)	
scalable_nesting(payloadSize)	
else if(payloadType == 145)	
dependent_rap_indication(payloadSize) // Được chỉ định trong ITU-T H.SEI ISO/IEC 23002-7.	
else if(payloadType == 168)	
frame_field_info(payloadSize)	
else	
reserved_sei_message(payloadSize)	
else /* nal_unit_type == SUFFIX_SEI_NUT */	
if(payloadType == 3)	
filler_payload(payloadSize)	
if(payloadType == 132)	
decoded_picture_hash(payloadSize) // Được chỉ định trong ITU-T H.SEI ISO/IEC 23002-7.	
else if(payloadType == 133)	
scalable_nesting(payloadSize)	
else	
reserved_sei_message(payloadSize)	
if(more_data_in_payload()) {	
if(payload_extension_present())	
reserved_payload_extension_data	u(v)
payload_bit_equal_to_one /* equal to 1 */	f(1)
while(!byte_aligned())	
payload_bit_equal_to_zero /* equal to 0 */	f(1)
}	
}	

Cú pháp thông điệp SEI lồng có khả năng mở rộng làm ví dụ là như sau.

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

Ngữ nghĩa thông điệp SEI lồng có khả năng mở rộng làm ví dụ là như sau. Thông điệp SEI lồng có khả năng mở rộng cung cấp cơ chế để kết hợp các thông điệp SEI với các lớp cụ thể nằm trong ngữ cảnh của các OLS cụ thể hoặc với các lớp cụ thể không ở trong ngữ cảnh của OLS. Thông điệp SEI lồng có khả năng mở rộng chứa một hoặc nhiều thông điệp SEI. Các thông điệp SEI có trong thông điệp SEI lồng có khả năng mở rộng còn được gọi là các thông điệp SEI được lồng có khả năng mở rộng. Tính phù hợp dòng

bit có thể yêu cầu rằng các ràng buộc sau áp dụng khi các thông điệp SEI có trong thông điệp SEI lồng có khả năng mở rộng.

Thông điệp SEI mà có payloadType bằng một trăm ba mươi hai (hàm băm ảnh được giải mã) hoặc một trăm ba mươi ba (lồng có khả năng mở rộng) không nên có trong thông điệp SEI lồng có khả năng mở rộng. Khi thông điệp SEI lồng có khả năng mở rộng chứa thông điệp SEI chu kỳ đệm, thời gian ảnh hoặc thông tin đơn vị giải mã, thì thông điệp SEI lồng có khả năng mở rộng không được chứa thông điệp SEI bất kỳ khác có payloadType không bằng không (chu kỳ đệm), một (thời gian ảnh), hoặc 130 (thông tin đơn vị giải mã).

Tính phù hợp dòng bit có thể còn yêu cầu rằng các ràng buộc sau được áp dụng trên giá trị của nal_unit_type của đơn vị SEI NAL chứa thông điệp SEI lồng có khả năng mở rộng. Khi thông điệp SEI lồng có khả năng mở rộng chứa thông điệp SEI mà có payloadType bằng không (chu kỳ đệm), một (thời gian ảnh), 130 (thông tin đơn vị giải mã), 145 (chỉ báo RAP phụ thuộc), hoặc 168 (thông tin khung-trường), đơn vị SEI NAL chứa thông điệp SEI lồng có khả năng mở rộng cần có nal_unit_type được thiết lập bằng PREFIX_SEI_NUT. Khi thông điệp SEI lồng có khả năng mở rộng chứa thông điệp SEI mà có payloadType bằng một trăm ba mươi hai (hàm băm ảnh được giải mã), thì đơn vị SEI NAL chứa thông điệp SEI lồng có khả năng mở rộng cần có nal_unit_type được thiết lập bằng SUFFIX_SEI_NUT.

nesting_ols_flag có thể được thiết lập bằng một để chỉ định rằng các thông điệp SEI được lồng có khả năng mở rộng áp dụng cho các lớp cụ thể trong ngữ cảnh các OLS cụ thể. nesting_ols_flag có thể được thiết lập bằng không để chỉ định rằng các thông điệp SEI được lồng có khả năng mở rộng thường áp dụng (ví dụ, không trong ngữ cảnh OLS) cho các lớp cụ thể.

Tính phù hợp dòng bit có thể yêu cầu rằng các ràng buộc sau được áp dụng cho giá trị của nesting_ols_flag. Khi thông điệp SEI lồng có khả năng mở rộng chứa thông điệp SEI mà có payloadType bằng không (chu kỳ đệm), một (thời gian ảnh), hoặc một trăm ba mươi (thông tin đơn vị giải mã), thì giá trị của nesting_ols_flag cần bằng một. Khi thông điệp SEI lồng có khả năng mở rộng chứa thông điệp SEI mà có payloadType bằng giá trị trong VclAssociatedSeiList, thì giá trị của nesting_ols_flag nên bằng không.

nesting_num_olss_minus1 cộng một chỉ định số lượng của các OLS mà áp dụng các thông điệp SEI được lồng có khả năng mở rộng. Giá trị của nesting_num_olss_minus1 cần nằm trong khoảng từ, bao gồm, không đến TotalNumOlss - 1. nesting_ols_idx_delta_minus1[i] được sử dụng để suy ra biến NestingOlsIdx[i] mà chỉ định chỉ số OLS của OLS thứ i mà áp dụng các thông điệp SEI được lồng có khả năng

mở rộng khi `nesting_ols_flag` bằng một. Giá trị `nesting_ols_idx_delta_minus1[ i ]` cần nằm trong khoảng từ, bao gồm, không đến `TotalNumOlss - 2`. Biến `NestingOlsIdx[ i ]` có thể được suy ra như sau:

`if( i == 0 )`

`NestingOlsIdx[ i ] = nesting_ols_idx_delta_minus1[ i ]`

`else`

`NestingOlsIdx[ i ] = NestingOlsIdx[ i - 1 ] + nesting_ols_idx_delta_minus1[ i ] + 1`

`nesting_num_ols_layers_minus1[ i ]` cộng một chỉ định số lượng của các lớp mà áp dụng các thông điệp SEI được lồng có khả năng mở rộng trong ngữ cảnh của OLS thứ `NestingOlsIdx[ i ]`. Giá trị của `nesting_num_ols_layers_minus1[ i ]` cần nằm trong khoảng, bao gồm, từ không đến `NumLayersInOls[ NestingOlsIdx[ i ] ] - 1`.

`nesting_ols_layer_idx_delta_minus1[ i ][ j ]` được sử dụng để suy ra biến `NestingOlsLayerIdx[ i ][ j ]` mà chỉ định chỉ số lớp OLS của lớp thứ `j` mà áp dụng các thông điệp SEI được lồng có khả năng mở rộng trong ngữ cảnh OLS thứ `NestingOlsIdx[ i ]` khi `nesting_ols_flag` bằng một. Giá trị `nesting_ols_layer_idx_delta_minus1[ i ]` cần nằm trong khoảng, bao gồm, từ không đến `NumLayersInOls[ nestingOlsIdx[ i ] ] - 2`.

Biến `NestingOlsLayerIdx[ i ][ j ]` có thể được suy ra như sau:

`if( j == 0 )`

`NestingOlsLayerIdx[ i ][ j ] = nesting_ols_layer_idx_delta_minus1[ i ][ j ]`

`else`

`NestingOlsLayerIdx[ i ][ j ] = NestingOlsLayerIdx[ i ][ j - 1 ] +`

`nesting_ols_layer_idx_delta_minus1[ i ][ j ] + 1`

Giá trị thấp nhất trong số tất cả các giá trị của `LayerIdInOls[ NestingOlsIdx[ i ] ][ NestingOlsLayerIdx[ i ][ 0 ] ]` với `i` nằm trong khoảng, bao gồm, từ không đến `nesting_num_olss_minus1`, cần bằng `nuh_layer_id` của đơn vị SEI NAL hiện thời (ví dụ, đơn vị SEI NAL chứa thông điệp SEI lồng có khả năng mở rộng). `nesting_all_layers_flag` có thể được thiết lập thành một để chỉ định rằng các thông điệp SEI được lồng có khả năng mở rộng thường áp dụng cho tất cả các lớp mà có `nuh_layer_id` lớn hơn hoặc bằng `nuh_layer_id` của đơn vị SEI NAL hiện thời. `nesting_all_layers_flag` có thể được thiết lập bằng không để chỉ định rằng các thông điệp SEI được lồng có khả năng mở rộng có thể hoặc có thể thường không áp dụng cho tất cả

các lớp mà có `nuh_layer_id` lớn hơn hoặc bằng `nuh_layer_id` của đơn vị SEI NAL hiện thời.

`nesting_num_layers_minus1` cộng một chỉ định số lượng của các lớp mà áp dụng chung các thông điệp SEI được lồng có khả năng mở rộng. Giá trị của `nesting_num_layers_minus1` cần nằm trong khoảng, bao gồm, từ không đến `vps_max_layers_minus1 - GeneralLayerIdx[ nuh_layer_id ]`, trong đó `nuh_layer_id` là `nuh_layer_id` của đơn vị SEI NAL hiện thời. `nesting_layer_id[ i ]` chỉ định giá trị của `nuh_layer_id` của lớp thứ `i` mà áp dụng chung các thông điệp SEI được lồng có khả năng mở rộng khi `nesting_all_layers_flag` bằng không. Giá trị của `nesting_layer_id[ i ]` cần lớn hơn `nuh_layer_id`, trong đó `nuh_layer_id` là `nuh_layer_id` của đơn vị SEI NAL hiện thời.

Khi `nesting_ols_flag` bằng một, thì biến `NestingNumLayers`, chỉ định số lượng lớp mà áp dụng chung các thông điệp SEI được lồng có khả năng mở rộng, và danh sách `NestingLayerId[ i ]` với `i` nằm trong khoảng, bao gồm, từ không đến `NestingNumLayers - 1`, chỉ định danh sách của giá trị của `nuh_layer_id` của các lớp mà áp dụng chung các thông điệp SEI được lồng có khả năng mở rộng, được suy ra như sau, trong đó `nuh_layer_id` là `nuh_layer_id` của đơn vị SEI NAL hiện thời:

```

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

```

`Nesting_num_seis_minus1` cộng một chỉ định số lượng của các thông điệp SEI được lồng có khả năng mở rộng. Giá trị của `nesting_num_seis_minus1` cần nằm trong khoảng, bao gồm, từ không đến sáu mươi ba. `nesting_zero_bit` cần được thiết lập bằng không.

Fig.8 là sơ đồ sơ lược của thiết bị mã hóa video 800 làm ví dụ. Thiết bị giải mã video 800 này thích hợp để thực hiện các ví dụ/phương án được bộc lộ như được mô tả

trong bản mô tả này. Thiết bị mã hóa video 800 bao gồm các cổng phía ra 820, các cổng phía vào 850 và/hoặc các đơn vị thu-phát (Tx/Rx) 810, bao gồm các bộ phát và/hoặc các bộ thu để truyền thông dữ liệu phía vào và/hoặc phía ra qua mạng. Thiết bị giải mã video 800 còn bao gồm bộ xử lý 830 bao gồm khối logic và/hoặc đơn vị xử lý trung tâm (central processing unit, CPU) để xử lý dữ liệu và bộ nhớ 832 để lưu trữ dữ liệu. Thiết bị mã hóa video 800 cũng có thể bao gồm các thành phần điện, quang thành điện (Optical-to-Electrical, OE), thành phần điện thành quang (Electrical-to-Optical, EO), và/hoặc thành phần truyền thông không dây được ghép nối với các cổng phía vào 850 và/hoặc các cổng phía ra 820 để truyền thông dữ liệu qua các mạng truyền thông điện, quang, hoặc không dây. Thiết bị mã hóa video 800 có thể cũng bao gồm các thiết bị nhập và/hoặc xuất (input and/or output, I/O) 860 để truyền thông dữ liệu đến và từ người dùng. Các thiết bị I/O 860 có thể bao gồm các thiết bị kết xuất chẳng hạn như màn hiển thị để hiển thị dữ liệu video, loa để kết xuất dữ liệu âm thanh, v.v. Các thiết bị I/O 860 có thể cũng bao gồm các thiết bị nhập, chẳng hạn như bàn phím, chuột, bi lăn, v.v., và/hoặc các giao diện tương ứng để tương tác với các thiết bị đầu ra này.

Bộ xử lý 830 được thực hiện bởi phần cứng và phần mềm. Bộ xử lý 830 này có thể được thực hiện dưới dạng một hoặc nhiều chip CPU, lõi (ví dụ, dưới dạng bộ xử lý đa lõi), các mảng cổng lập trình được dạng trường (field-programmable gate array, FPGA), các mạch tích hợp cho ứng dụng cụ thể (application specific integrated circuit, ASIC), và các bộ xử lý tín hiệu số (digital signal processor, DSP). Bộ xử lý 830 truyền thông với các cổng phía ra 820, Tx/Rx 810, các cổng phía vào 850, và bộ nhớ 832. Bộ xử lý 830 bao gồm môđun mã hóa 814. Môđun mã hóa 814 này thực hiện các phương án được bộc lộ được mô tả trong bản mô tả này, chẳng hạn như các phương pháp 100, 900 và 1000, mà có thể sử dụng chuỗi video đa lớp 600 và/hoặc dòng bit 700. Môđun mã hóa 814 cũng có thể thực hiện phương pháp/cơ chế bất kỳ khác được mô tả trong bản mô tả này. Ngoài ra, môđun mã hóa 814 có thể thực hiện hệ thống mã hóa-giải mã 200, bộ mã hóa 300, bộ giải mã 400 và/hoặc HRD 500. Ví dụ, môđun mã hóa 814 có thể sử dụng tín hiệu và/hoặc đọc các tham số khác nhau như được mô tả trong bản mô tả này. Ngoài ra, môđun mã hóa có thể được sử dụng để mã hóa và/hoặc giải mã chuỗi video dựa vào các tham số này. Do đó, các thay đổi về báo hiệu được mô tả trong bản mô tả này có thể làm tăng hiệu quả và/hoặc tránh được các lỗi trong môđun mã hóa 814. Theo đó, môđun mã hóa 814 có thể được tạo cấu hình để thực hiện các cơ chế để giải quyết một hoặc nhiều vấn đề được thảo luận ở trên. Do đó, môđun mã hóa 814 làm cho thiết bị mã hóa video 800 cung cấp chức năng bổ sung và/hoặc hiệu quả mã hóa khi mã hóa dữ liệu video. Như vậy, môđun mã hóa 814 cải thiện chức năng của thiết bị mã hóa video 800 cũng như giải quyết các vấn đề cụ thể đối với tình trạng kỹ thuật của việc mã hóa video. Ngoài ra, môđun mã hóa 814

có tác dụng biến đổi thiết bị mã hóa video 800 sang trạng thái khác. Theo cách khác, môđun mã hóa 814 được thực hiện dưới dạng các lệnh được lưu trữ trong bộ nhớ 832 và được thực thi bởi bộ xử lý 830 (ví dụ, dưới dạng sản phẩm chương trình máy tính được lưu trữ trên phương tiện không tạm thời).

Bộ nhớ 832 bao gồm một hoặc nhiều kiểu bộ nhớ chẳng hạn như các đĩa, ổ đĩa băng, ổ đĩa thể rắn, bộ nhớ chỉ đọc (Read Only Memory, ROM), bộ nhớ truy nhập ngẫu nhiên (Random Access Memory, RAM), bộ nhớ chớp, bộ nhớ định địa chỉ được nội dung tam phân (Ternary Content-Addressable Memory, TCAM), bộ nhớ truy nhập ngẫu nhiên tĩnh (Static Random-Access Memory, SRAM), v.v. Bộ nhớ 832 có thể được sử dụng làm thiết bị lưu trữ dữ liệu trên luồng, để lưu trữ các chương trình khi các chương trình này được chọn để thực thi, và để lưu trữ các lệnh và dữ liệu mà đọc trong quá trình thực thi chương trình.

Fig.9 là lưu đồ của phương pháp 900 làm ví dụ để mã hóa chuỗi video thành dòng bit, chẳng hạn như dòng bit 700, để hỗ trợ việc sử dụng NoOutputOfPriorPicsFlag kết hợp với ARC. Phương pháp 900 này có thể được sử dụng bởi bộ mã hóa, chẳng hạn như hệ thống mã hóa-giải mã 200, bộ mã hóa 300 và/hoặc thiết bị mã hóa video 800 khi thực hiện phương pháp 100. Ngoài ra, phương pháp 900 có thể hoạt động trên HRD 500 và do đó có thể thực hiện các thử nghiệm tính phù hợp đối với chuỗi video đa lớp 600.

Phương pháp 900 có thể bắt đầu khi bộ mã hóa nhận chuỗi video và để mã hóa chuỗi video thành dòng bit đa lớp, ví dụ dựa vào đầu vào người dùng. Ở bước 901, bộ mã hóa mã hóa các ảnh thành dòng bit. Ví dụ, các ảnh có thể có trong các AU. Hơn nữa, các ảnh có thể có trong các lớp. Bộ mã hóa có thể mã hóa một hoặc nhiều lớp bao gồm các ảnh thành dòng bit đa lớp. Lớp có thể bao gồm tập hợp gồm các đơn vị VCL NAL mà chứa dữ liệu video của các ảnh được mã hóa cũng như các tập hợp tham số bất kỳ được sử dụng để mã hóa các ảnh này. Một hoặc nhiều trong số các lớp có thể là các lớp đầu ra. Các lớp mà không phải là lớp đầu ra được mã hóa để hỗ trợ việc khôi phục (các) lớp đầu ra, nhưng các lớp hỗ trợ này không được nhằm làm đầu ra ở bộ giải mã. Theo cách này, bộ mã hóa có thể mã hóa các kết hợp khác nhau của các lớp để truyền đến bộ giải mã ngay khi yêu cầu. Các lớp có thể được truyền như mong muốn để cho phép bộ giải mã thu các biểu diễn khác nhau của chuỗi video tùy thuộc vào các điều kiện mạng, khả năng phần cứng, và/hoặc các thiết lập người dùng. Khi dòng bit được mã hóa, thì bộ mã hóa có thể sử dụng HRD để kiểm tra dòng bit về tính phù hợp.

Ở bước 903, HRD có thể giải mã ảnh đứng trước trong AU đứng trước và lưu trữ ảnh đứng trước trong DPB. Ví dụ, ảnh đứng trước có thể là một phần của lớp thứ nhất. Sau đó, HRD có thể kiểm tra đối với các vấn đề mà xảy ra trong quá trình chuyển đổi sang lớp thứ hai.

Ở bước 905, HRD có thể chuẩn bị để giải mã ảnh hiện thời trong lớp thứ hai của AU hiện thời. HRD có thể phân tích cú pháp phần đầu lát của lát thứ nhất của ảnh hiện thời. HRD có thể còn xây dựng danh sách ảnh tham chiếu đối với lát thứ nhất dựa vào danh sách ảnh tham chiếu. HRD có thể còn thực hiện quá trình đánh dấu danh sách ảnh tham chiếu đối với các ảnh trong DPB (ví dụ, bao gồm ảnh đứng trước) để đánh dấu mỗi ảnh là chưa được sử dụng để tham chiếu, được sử dụng để tham chiếu ngắn hạn, và/hoặc được sử dụng để tham chiếu dài hạn.

Ở bước 907, HRD có thể thiết lập NoOutputOfPriorPicsFlag thành giá trị tương ứng (ví dụ, thành một) khi giá trị của PicWidthMaxInSamplesY đối với ảnh hiện thời trong AU hiện thời khác với giá trị của PicWidthMaxInSamplesY đối với ảnh đứng trước trong AU đứng trước theo thứ tự giải mã. PicWidthMaxInSamplesY có thể được suy ra dựa vào `sps_pic_width_max_in_luma_samples` thu được từ SPS trong dòng bit. Hơn nữa, HRD có thể thiết lập NoOutputOfPriorPicsFlag khi giá trị của PicHeightMaxInSamplesY đối với ảnh hiện thời trong AU hiện thời khác với giá trị của PicHeightMaxInSamplesY đối với ảnh đứng trước trong AU đứng trước theo thứ tự giải mã. PicHeightMaxInSamplesY có thể được suy ra dựa vào `sps_pic_height_max_in_luma_samples` thu được từ SPS trong dòng bit. NoOutputOfPriorPicsFlag có thể thiết lập thành một để chỉ định rằng tất cả các bộ đệm lưu trữ ảnh trong DPB cần được làm trống mà không kết xuất các ảnh mà chúng chứa, và biến lấp đầy DPB này cần được thiết lập bằng không.

Ở bước 909, HRD có thể làm trống DPB mà không chọn ảnh đã chứa để kết xuất dựa vào giá trị của NoOutputOfPriorPicsFlag (ví dụ, khi NoOutputOfPriorPicsFlag được thiết lập bằng một). HRD có thể còn giảm bớt biến lấp đầy DPB (ví dụ, giảm xuống một) đối với mỗi ảnh được loại bỏ khi làm trống DPB. Cần lưu ý rằng NoOutputOfPriorPicsFlag có thể chỉ được thiết lập ở bước 907, và do đó DPB có thể chỉ được làm trống ở bước 909, khi tất cả các ảnh trong DPB bao gồm ảnh đứng trước được đánh dấu là chưa được sử dụng để tham chiếu theo quy trình đánh dấu danh sách ảnh tham chiếu của bước 905.

Sau đó, HRD có thể giải mã ảnh hiện thời trong AU hiện thời và lưu trữ ảnh hiện thời trong DPB ở bước 911. Theo đó, DPB được làm trống ở bước 909 trước khi ảnh hiện thời trong AU hiện thời được giải mã ở bước 911. Hơn nữa, DPB được làm trống ở bước 909 sau khi phân tích cú pháp phần đầu lát của lát thứ nhất của ảnh hiện thời ở bước 905.

Trong một số ví dụ, NoOutputOfPriorPicsFlag có thể ngăn DPB tràn khi chuyển đổi giữa lớp thứ nhất chứa ảnh đứng trước và lớp thứ hai chứa ảnh hiện thời. Theo đó, việc kiểm tra tính phù hợp HRD có thể thành công khi kiểm tra ảnh hiện thời về tính phù hợp. Giả sử không gặp phải các sai số nào khác, việc kiểm tra tính phù hợp HRD được

hoàn thành. Giả sử việc kiểm tra tính phù hợp thành công, dòng bit được lưu trữ trong bộ nhớ dài hạn để truyền thông về phía bộ giải mã ở bước 913, ví dụ qua máy chủ nội dung. Sau đó, các lớp của dòng bit có thể được truyền về phía bộ giải mã ngay khi yêu cầu.

Fig.10 là lưu đồ của phương pháp 1000 làm ví dụ để giải mã chuỗi video từ dòng bit, chẳng hạn như dòng bit 700, trong khi sử dụng NoOutputOfPriorPicsFlag kết hợp với ARC. Phương pháp 1000 có thể được sử dụng bởi bộ giải mã, chẳng hạn như hệ thống mã hóa-giải mã 200, bộ giải mã 400 và/hoặc thiết bị mã hóa video 800 khi thực hiện phương pháp 100. Ngoài ra, phương pháp 1000 có thể được sử dụng trên chuỗi video đa lớp 600 mà đã được kiểm tra tính phù hợp bởi HRD, chẳng hạn như HRD 500.

Phương pháp 1000 có thể bắt đầu khi bộ giải mã bắt đầu nhận dòng bit của dữ liệu được mã hóa biểu diễn chuỗi video đa lớp, ví dụ là kết quả của phương pháp 900 và/hoặc đáp lại yêu cầu bởi bộ giải mã. Ở bước 1001, bộ giải mã nhận dòng bit bao gồm dòng bit bao gồm các ảnh. Ví dụ, các ảnh có thể có trong các AU. Hơn nữa, các ảnh có thể có trong các lớp. Lớp có thể bao gồm tập hợp các đơn vị VCL NAL với cùng ID lớp và các đơn vị NAL không phải VCL. Ví dụ, tập hợp các đơn vị VCL NAL là một phần của lớp khi tập hợp các đơn vị VCL NAL đều có cùng ID lớp. Lớp có thể bao gồm tập hợp các đơn vị VCL NAL mà chứa dữ liệu video của các ảnh cũng như các tập hợp tham số bất kỳ được sử dụng để mã hóa các ảnh trong các đơn vị NAL không phải VCL. Một hoặc nhiều trong số các lớp có thể là các lớp đầu ra. Các lớp mà không phải là lớp đầu ra được mã hóa để hỗ trợ việc khôi phục (các) lớp đầu ra, nhưng các lớp hỗ trợ này không được nhằm để kết xuất. Theo cách này, bộ giải mã có thể thu các biểu diễn khác nhau của chuỗi video tùy thuộc vào các điều kiện mạng, các khả năng phần cứng, và/hoặc các thiết lập người dùng.

Ở bước 1003, bộ giải mã có thể giải mã ảnh đứng trước trong AU đứng trước và lưu trữ ảnh đứng trước trong DPB. Ví dụ, ảnh đứng trước có thể là một phần của lớp thứ nhất. Bộ giải mã có thể xác định để chuyển đổi sang lớp thứ hai, ví dụ do đầu vào người dùng và/hoặc do sự thay đổi về các điều kiện hoạt động. Ví dụ, bộ giải mã có thể yêu cầu để chuyển đổi sang lớp thứ hai.

Ở bước 1005, bộ giải mã có thể chuẩn bị để giải mã ảnh hiện thời trong lớp thứ hai của AU hiện thời. Bộ giải mã có thể phân tích cú pháp phần đầu lát của lát thứ nhất của ảnh hiện thời. Bộ giải mã có thể còn xây dựng danh sách ảnh tham chiếu đối với lát thứ nhất dựa vào danh sách ảnh tham chiếu. Bộ giải mã có thể còn thực hiện quá trình đánh dấu danh sách ảnh tham chiếu đối với các ảnh trong DPB (ví dụ, bao gồm ảnh đứng trước) để đánh dấu mỗi ảnh là chưa được sử dụng để tham chiếu, được sử dụng để tham chiếu ngắn hạn, và/hoặc được sử dụng để tham chiếu dài hạn.

Ở bước 1007, bộ giải mã có thể thiết lập NoOutputOfPriorPicsFlag thành giá trị tương ứng (ví dụ, thành một) khi giá trị của PicWidthMaxInSamplesY đối với ảnh hiện thời trong AU hiện thời khác với giá trị của PicWidthMaxInSamplesY đối với ảnh đứng trước trong AU đứng trước theo thứ tự giải mã. PicWidthMaxInSamplesY có thể được suy ra dựa vào sps_pic_width_max_in_luma_samples thu được từ SPS trong dòng bit. Hơn nữa, bộ giải mã có thể thiết lập NoOutputOfPriorPicsFlag khi giá trị của PicHeightMaxInSamplesY đối với ảnh hiện thời trong AU hiện thời khác với giá trị của PicHeightMaxInSamplesY đối với ảnh đứng trước trong AU đứng trước theo thứ tự giải mã. PicHeightMaxInSamplesY có thể được suy ra dựa vào sps_pic_height_max_in_luma_samples thu được từ SPS trong dòng bit. NoOutputOfPriorPicsFlag có thể thiết lập thành một để chỉ định rằng tất cả các bộ đệm lưu trữ ảnh trong DPB cần được làm trống mà không kết xuất các ảnh mà chúng chứa, và biến lấp đầy DPB này cần được thiết lập bằng không.

Ở bước 1009, bộ giải mã có thể làm trống DPB mà không chọn ảnh đã chứa để kết xuất dựa vào giá trị của NoOutputOfPriorPicsFlag (ví dụ, khi NoOutputOfPriorPicsFlag được thiết lập bằng một). Bộ giải mã có thể còn giảm bớt biến lấp đầy DPB (ví dụ, giảm xuống một) đối với mỗi ảnh được loại bỏ khi làm trống DPB. Cần lưu ý rằng NoOutputOfPriorPicsFlag có thể chỉ được thiết lập ở bước 1007, và do đó DPB có thể chỉ được làm trống ở bước 1009, khi tất cả các ảnh trong DPB bao gồm ảnh đứng trước được đánh dấu là chưa được sử dụng để tham chiếu theo quy trình đánh dấu danh sách ảnh tham chiếu của bước 1005.

Sau đó, bộ giải mã có thể giải mã ảnh hiện thời trong AU hiện thời và lưu trữ ảnh hiện thời trong DPB bước 1011. Theo đó, DPB được làm trống ở bước 1009 trước khi ảnh hiện thời trong AU hiện thời được giải mã ở bước 1011. Hơn nữa, DPB được làm trống ở bước 1009 sau khi phân tích cú pháp phần đầu lát của lát thứ nhất của ảnh hiện thời ở bước 1005. Trong một số ví dụ, NoOutputOfPriorPicsFlag có thể ngăn DPB tràn khi chuyển đổi giữa lớp thứ nhất chứa ảnh đứng trước và lớp thứ hai chứa ảnh hiện thời. Theo đó, việc sử dụng NoOutputOfPriorPicsFlag có thể làm tăng chức năng của bộ giải mã.

Ở bước 1013, bộ giải mã có thể kết xuất ảnh hiện thời từ DPB để hiển thị dưới dạng một phần chuỗi video được giải mã.

Fig.11 là sơ đồ sơ lược của hệ thống 1100 làm ví dụ để mã hóa chuỗi video thành dòng bit trong khi sử dụng NoOutputOfPriorPicsFlag kết hợp với ARC. Hệ thống 1100 này có thể được thực hiện bởi bộ mã hóa và bộ giải mã chẳng hạn như hệ thống mã hóa-giải mã 200, bộ mã hóa 300, bộ giải mã 400 và/hoặc thiết bị mã hóa video 800. Ngoài ra, hệ thống 1100 có thể sử dụng HRD 500 để thực hiện các thử nghiệm tính phù hợp trên

chuỗi video đa lớp 600 và/hoặc dòng bit 700. Ngoài ra, hệ thống 1100 có thể được sử dụng khi thực hiện phương pháp 100, 900, và/hoặc 1000.

Hệ thống 1100 bao gồm bộ mã hóa video 1102. Bộ mã hóa video 1102 bao gồm môđun mã hóa 1103 để mã hóa các ảnh thành dòng bit. Bộ mã hóa video 1102 còn bao gồm môđun HRD 1105 để kiểm tra dòng bit về tính phù hợp bằng cách: thiết lập NoOutputOfPriorPicsFlag khi giá trị của PicWidthMaxInSamplesY dùng cho AU hiện thời khác với giá trị của PicWidthMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã; làm trống DPB mà không chọn ảnh đã chứa để kết xuất dựa vào giá trị của NoOutputOfPriorPicsFlag; giải mã ảnh hiện thời; và lưu trữ ảnh hiện thời trong DPB. Bộ mã hóa video 1102 còn bao gồm môđun lưu trữ 1106 để lưu trữ dòng bit để truyền thông về phía bộ giải mã. Bộ mã hóa video 1102 còn bao gồm môđun truyền 1107 để truyền dòng bit đến bộ giải mã video 1110. Bộ mã hóa video 1102 có thể còn được tạo cấu hình để thực hiện bước bất kỳ trong số các bước của phương pháp 900.

Hệ thống 1100 cũng bao gồm bộ giải mã video 1110. Bộ giải mã video 1110 bao gồm môđun thu 1111 để nhận dòng bit bao gồm các ảnh. Bộ giải mã video 1110 còn bao gồm môđun thiết lập 1113 để thiết lập NoOutputOfPriorPicsFlag khi giá trị của PicWidthMaxInSamplesY dùng cho AU hiện thời khác với giá trị của PicWidthMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã. Bộ giải mã video 1110 còn bao gồm môđun làm trống 1115 để làm trống DPB mà không kết xuất các ảnh đã chứa dựa vào giá trị của NoOutputOfPriorPicsFlag. Bộ giải mã video 1110 còn bao gồm môđun giải mã 1117 để giải mã ảnh hiện thời. Bộ giải mã video 1110 còn bao gồm môđun lưu trữ 1119 để lưu trữ ảnh hiện thời trong DPB. Bộ giải mã video 1110 còn bao gồm môđun kết xuất 1121 để kết xuất ảnh hiện thời từ DPB để hiển thị là một phần của chuỗi video được giải mã. Bộ giải mã video 1110 có thể còn được tạo cấu hình để thực hiện bước bất kỳ trong số các bước của phương pháp 1000.

Thành phần thứ nhất được ghép nối trực tiếp với thành phần thứ hai khi không có các thành phần xen giữa, ngoại trừ đường, vết hoặc phương tiện khác giữa thành phần thứ nhất và thành phần thứ hai. Thành phần thứ nhất được ghép nối gián tiếp với thành phần thứ hai khi có các thành phần xen giữa khác với đường, vết, hoặc phương tiện khác giữa thành phần thứ nhất và thành phần thứ hai. Thuật ngữ “được ghép nối” và các biến thể của nó bao gồm cả được ghép nối trực tiếp và được ghép nối gián tiếp. Việc sử dụng thuật ngữ “khoảng” có nghĩa là phạm vi bao gồm $\pm 10\%$ số tiếp sau trừ khi có quy định khác.

Cũng cần hiểu rằng các bước của các phương pháp làm ví dụ được nêu ở đây không nhất thiết cần được thực hiện theo thứ tự được mô tả, và thứ tự của các bước của các phương pháp này cần được hiểu chỉ để làm ví dụ. Tương tự, các bước bổ sung có thể có

trong các phương pháp này, và các bước nhất định có thể được lược bỏ hoặc được kết hợp, trong các phương pháp phù hợp với các phương án khác nhau của sáng chế.

Trong khi một số phương án đã được đề xuất trong bản mô tả này, nhưng cần hiểu rằng các hệ thống và các phương pháp được bộc lộ có thể được thực hiện theo nhiều dạng cụ thể khác mà không lệch khỏi khỏi mục đích hoặc phạm vi của sáng chế. Các ví dụ của sáng chế được xem là minh họa và không giới hạn, và chủ đích là không bị giới hạn ở các chi tiết được đưa ra ở đây. Ví dụ, các phần tử hoặc các thành phần khác nhau có thể được kết hợp hoặc tích hợp trong hệ thống khác hoặc các dấu hiệu có thể được lược bỏ, hoặc không được thực hiện.

Ngoài ra, các kỹ thuật, hệ thống, hệ thống phụ và các phương pháp được mô tả và minh họa theo các phương án khác nhau theo cách riêng biệt hoặc riêng rẽ có thể được kết hợp hoặc được tích hợp với các hệ thống, các thành phần, các kỹ thuật hoặc các phương pháp khác mà không lệch khỏi phạm vi của sáng chế. Các ví dụ khác về các thay đổi, thay thế và biến đổi xác định được bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật và có thể được thực hiện mà không lệch khỏi mục đích và phạm vi được bộc lộ trong bản mô tả này.

YÊU CẦU BẢO HỘ

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

nhận, bởi bộ thu của bộ giải mã, dòng bit bao gồm các ảnh;

thiết lập, bởi bộ xử lý của bộ giải mã, giá trị không kết xuất cờ các ảnh trước (NoOutputOfPriorPicsFlag) thành một khi giá trị của chiều rộng ảnh tối đa trong các mẫu độ chói (PicWidthMaxInSamplesY) dùng cho đơn vị truy cập (access unit, AU) hiện thời khác với giá trị của PicWidthMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã;

làm trống bộ đệm ảnh được giải mã (decoded picture buffer, DPB) mà không kết xuất các ảnh đã chứa khi giá trị của NoOutputOfPriorPicsFlag bằng một;

giải mã, bởi bộ xử lý, ảnh hiện thời; và

lưu trữ, bởi DPB, ảnh hiện thời.

2. Phương pháp theo điểm 1, còn bao gồm bước thiết lập, bởi bộ xử lý, giá trị của NoOutputOfPriorPicsFlag thành một khi giá trị của chiều cao ảnh tối đa trong các mẫu độ chói (PicHeightMaxInSamplesY) dùng cho AU hiện thời khác với giá trị của PicHeightMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã.

3. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 2, trong đó DPB được làm trống trước khi giải mã ảnh hiện thời.

4. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó DPB được làm trống sau khi phân tích cú pháp phần đầu lát của lát thứ nhất của ảnh hiện thời.

5. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 4, còn bao gồm bước giảm bớt biến lấp đầy DPB đối với mỗi ảnh được loại bỏ khi làm trống DPB.

6. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 5, còn bao gồm bước xây dựng, bởi bộ xử lý, danh sách ảnh tham chiếu và thực hiện việc đánh dấu danh sách ảnh tham chiếu dùng cho AU hiện thời trước khi thiết lập NoOutputOfPriorPicsFlag.

7. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 6, trong đó DPB được làm trống khi ảnh đứng trước được đánh dấu là chưa được sử dụng để tham chiếu.

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

mã hóa các ảnh thành dòng bit;

kiểm tra, bởi bộ giải mã tham chiếu giả định (hypothetical reference decoder, HRD), dòng bit về tính phù hợp bằng cách:

thiết lập giá trị không kết xuất cờ các ảnh trước (NoOutputOfPriorPicsFlag) thành một khi giá trị của chiều rộng ảnh tối đa trong các mẫu độ chói (PicWidthMaxInSamplesY) dùng cho đơn vị truy cập hiện thời (AU) khác với giá trị của PicWidthMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã;

làm trống bộ đệm ảnh được giải mã (DPB) mà không chọn các ảnh đã chứa để kết xuất khi giá trị của NoOutputOfPriorPicsFlag bằng một.

9. Phương pháp theo điểm 8, còn bao gồm bước thiết lập, bởi HRD, giá trị của NoOutputOfPriorPicsFlag thành một khi giá trị của chiều cao ảnh tối đa trong các mẫu độ chói (PicHeightMaxInSamplesY) dùng cho AU hiện thời khác với giá trị của PicHeightMaxInSamplesY dùng cho AU đứng trước theo thứ tự giải mã.

10. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 9, trong đó DPB được làm trống trước khi giải mã ảnh hiện thời.

11. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 10, trong đó DPB được làm trống sau khi phân tích cú pháp phần đầu lát của lát thứ nhất của ảnh hiện thời.

12. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 11, còn bao gồm bước giảm bớt, bởi HRD, biến lấp đầy DPB đối với mỗi ảnh được loại bỏ khi làm trống DPB.

13. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 12, còn bao gồm bước xây dựng, bởi HRD, danh sách ảnh tham chiếu và thực hiện việc đánh dấu danh sách ảnh tham chiếu dùng cho AU hiện thời trước khi thiết lập NoOutputOfPriorPicsFlag.

14. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 13, trong đó DPB được làm trống khi ảnh đứng trước được đánh dấu là chưa được sử dụng để tham chiếu.

15. Phương tiện lưu trữ đọc được bằng máy tính có chương trình được ghi trên đó; trong đó chương trình làm cho máy tính thực thi phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 7.

16. Phương tiện lưu trữ đọc được bằng máy tính có chương trình được ghi trên đó; trong đó chương trình làm cho máy tính thực thi phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 14.

17. Thiết bị giải mã, bao gồm:

bộ thu được tạo cấu hình để nhận dòng bit bao gồm các ảnh;

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

bộ xử lý được ghép nối với bộ nhớ, bộ xử lý được tạo cấu hình để thực thi các lệnh để khiến thiết bị giải mã: thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 7.

18. Thiết bị mã hóa, bao gồm:

bộ xử lý;

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

bộ xử lý được tạo cấu hình để thực thi các lệnh để khiến thiết bị mã hóa: thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 14.

1/11

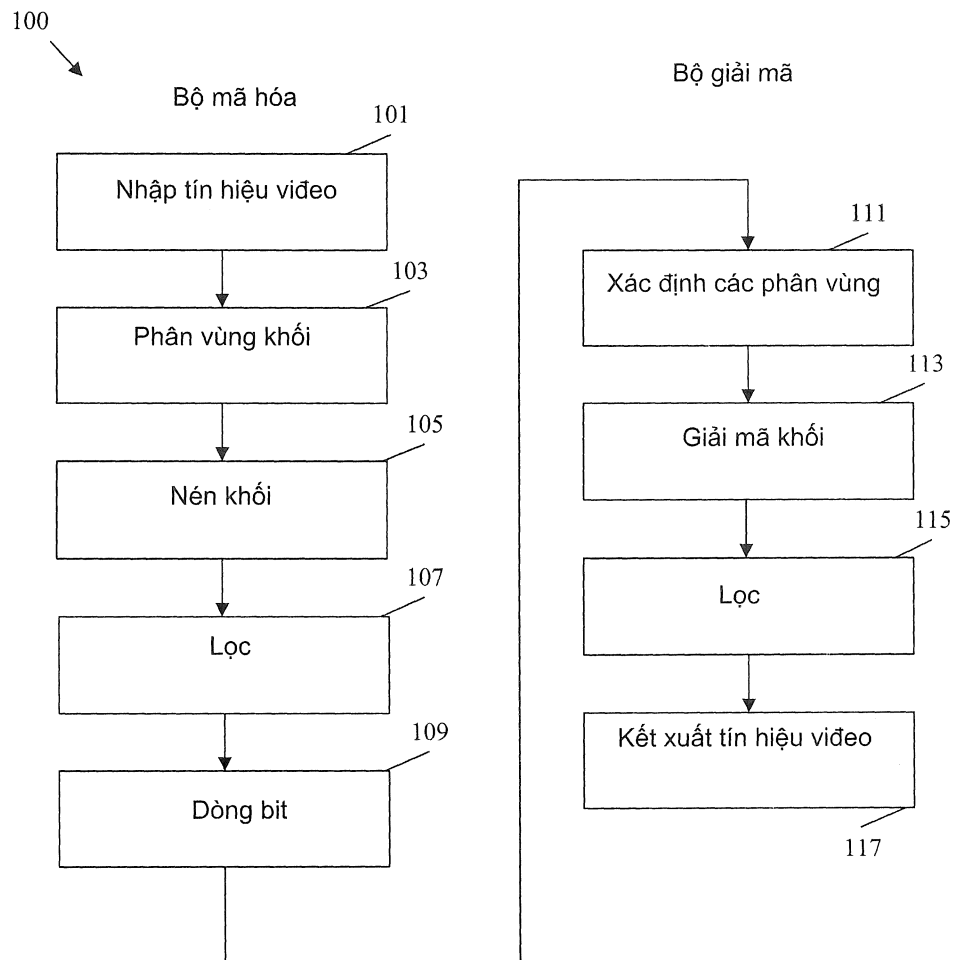


FIG. 1

2/11

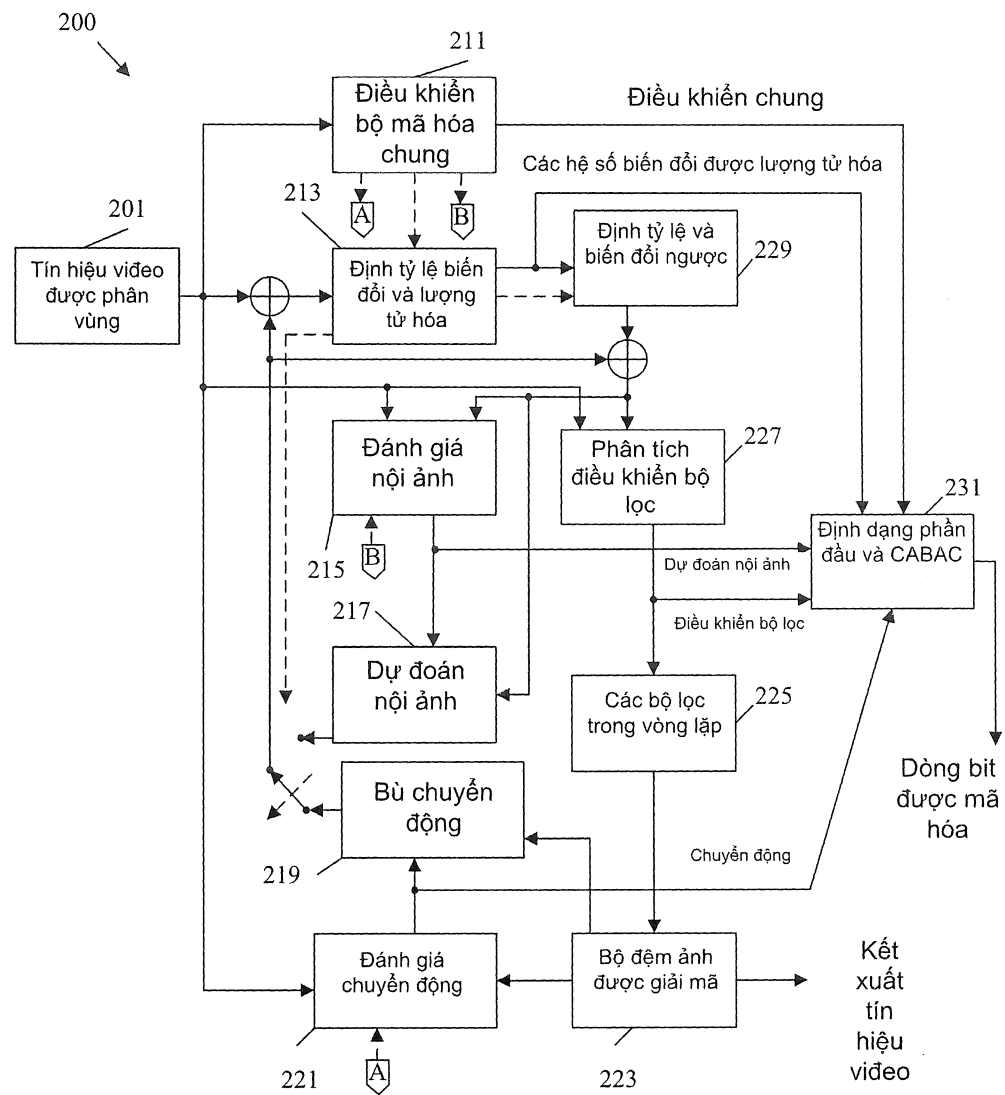


FIG. 2

3/11

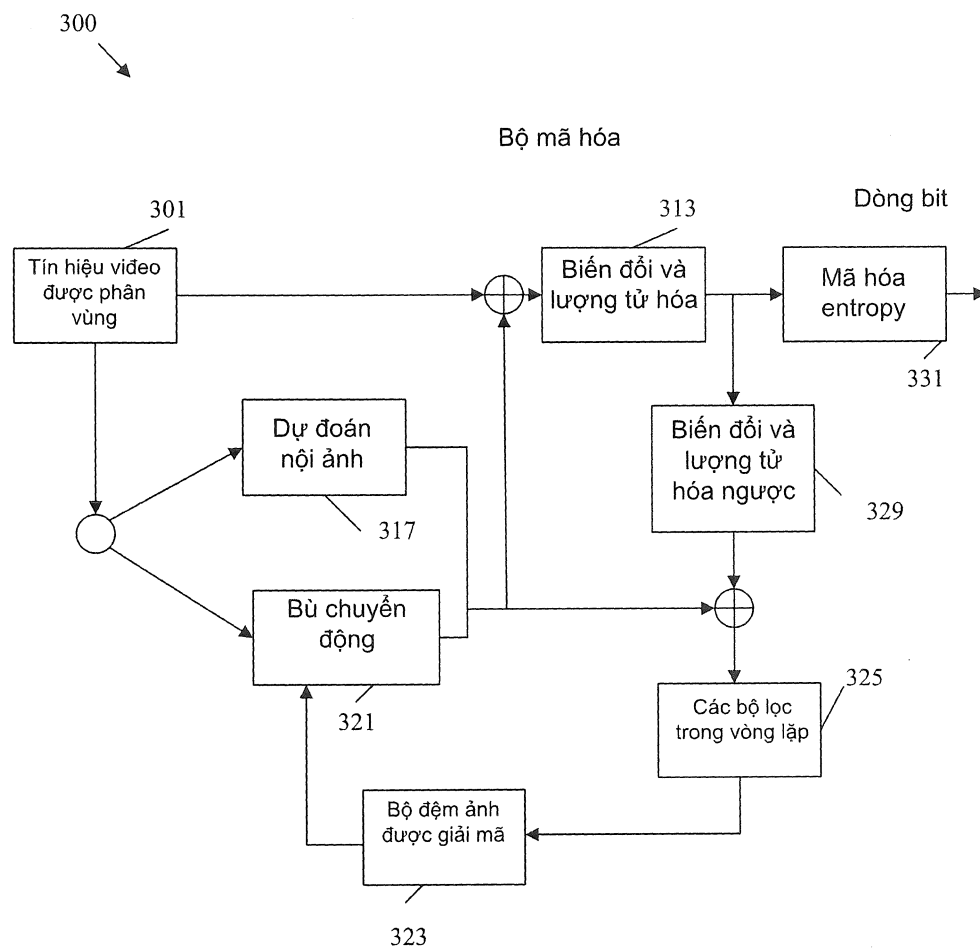


FIG. 3

4/11

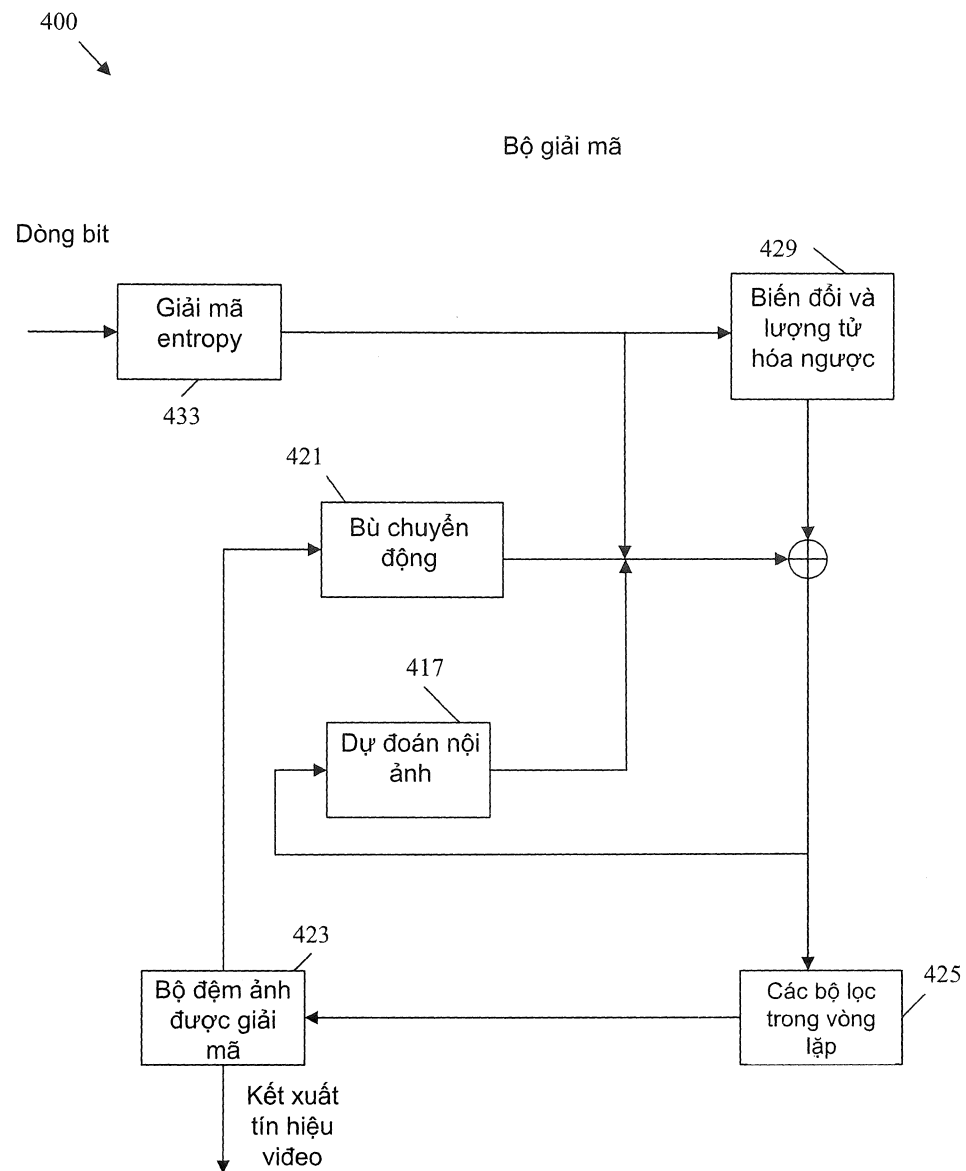


FIG. 4

5/11

Bộ giải mã tham chiếu giả định

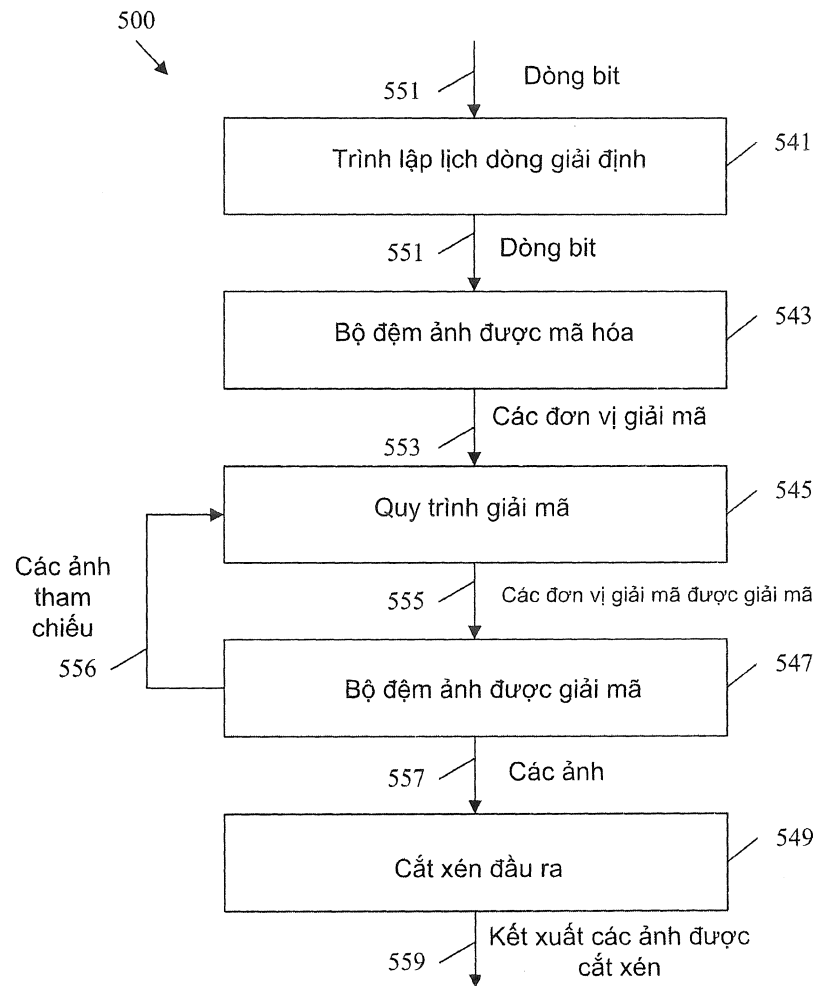


FIG. 5

6/11

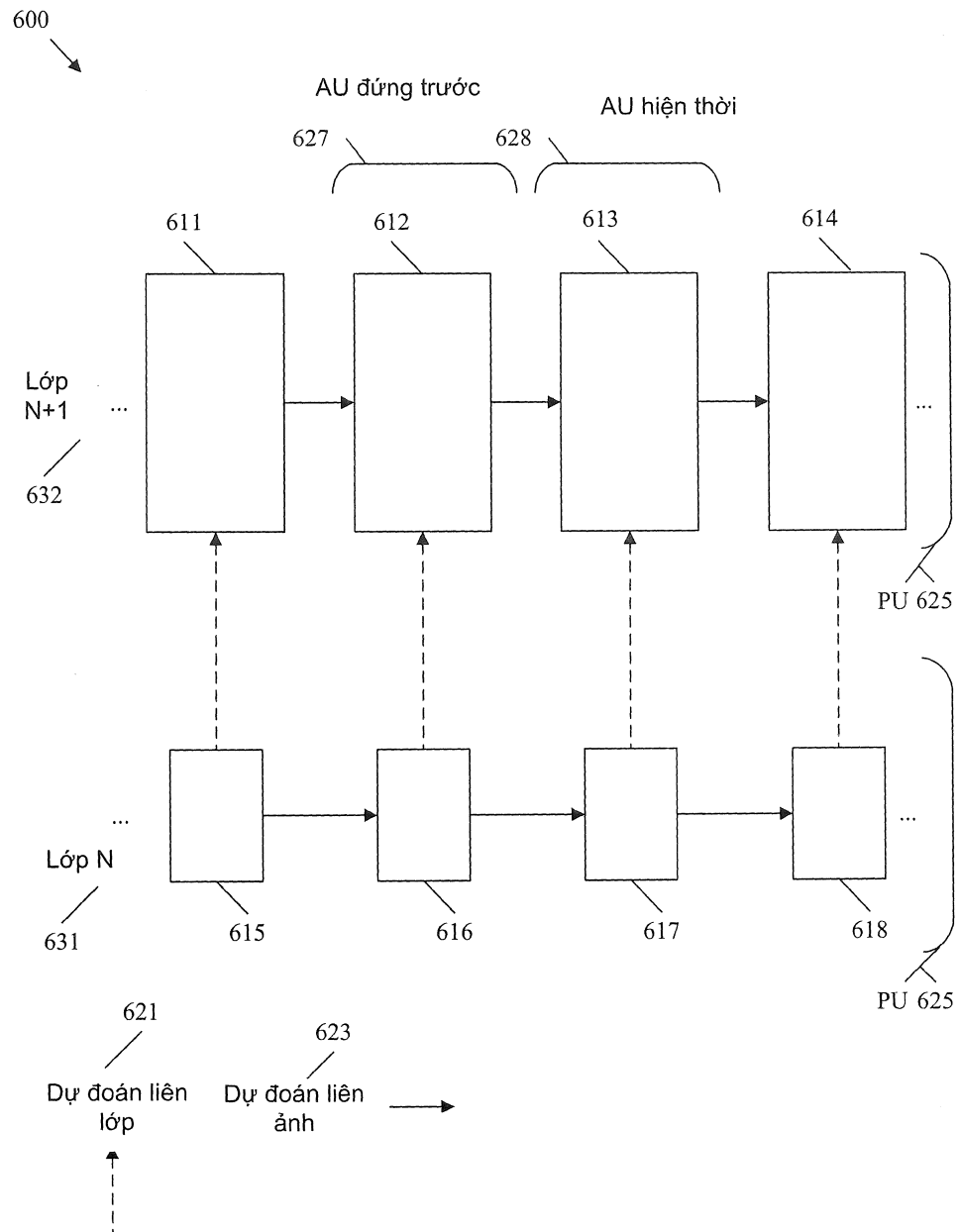


FIG. 6

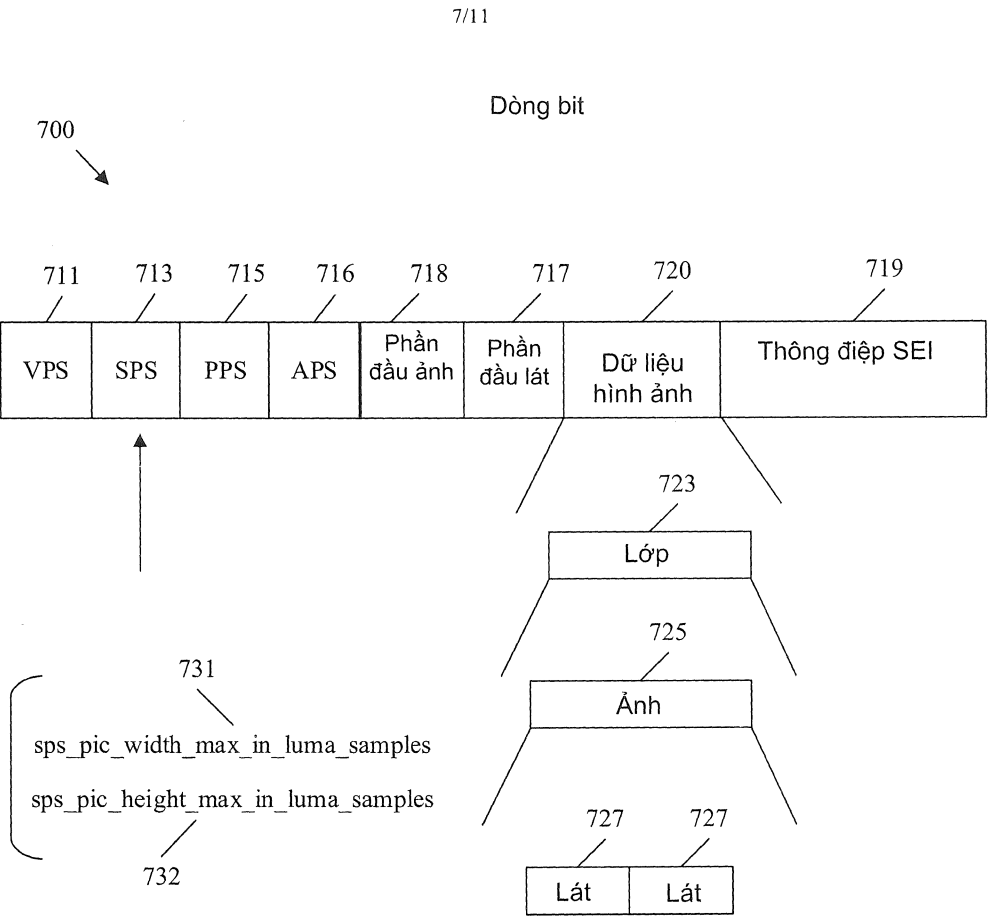


FIG. 7

8/11

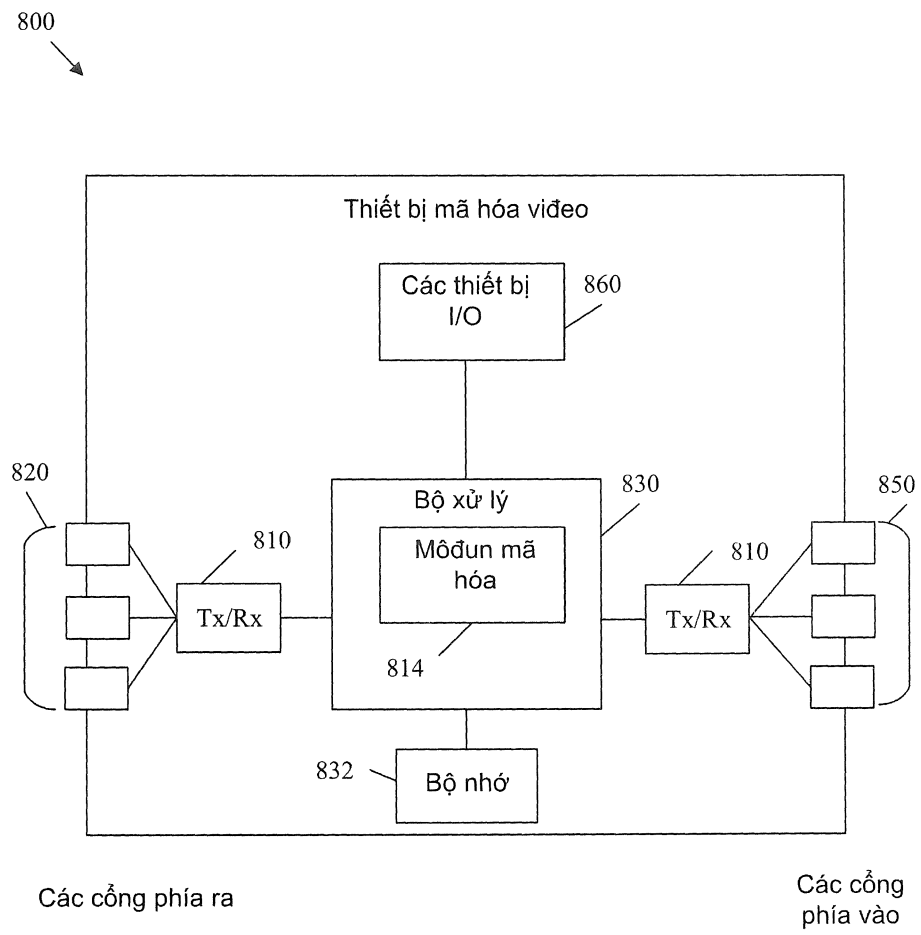


FIG. 8

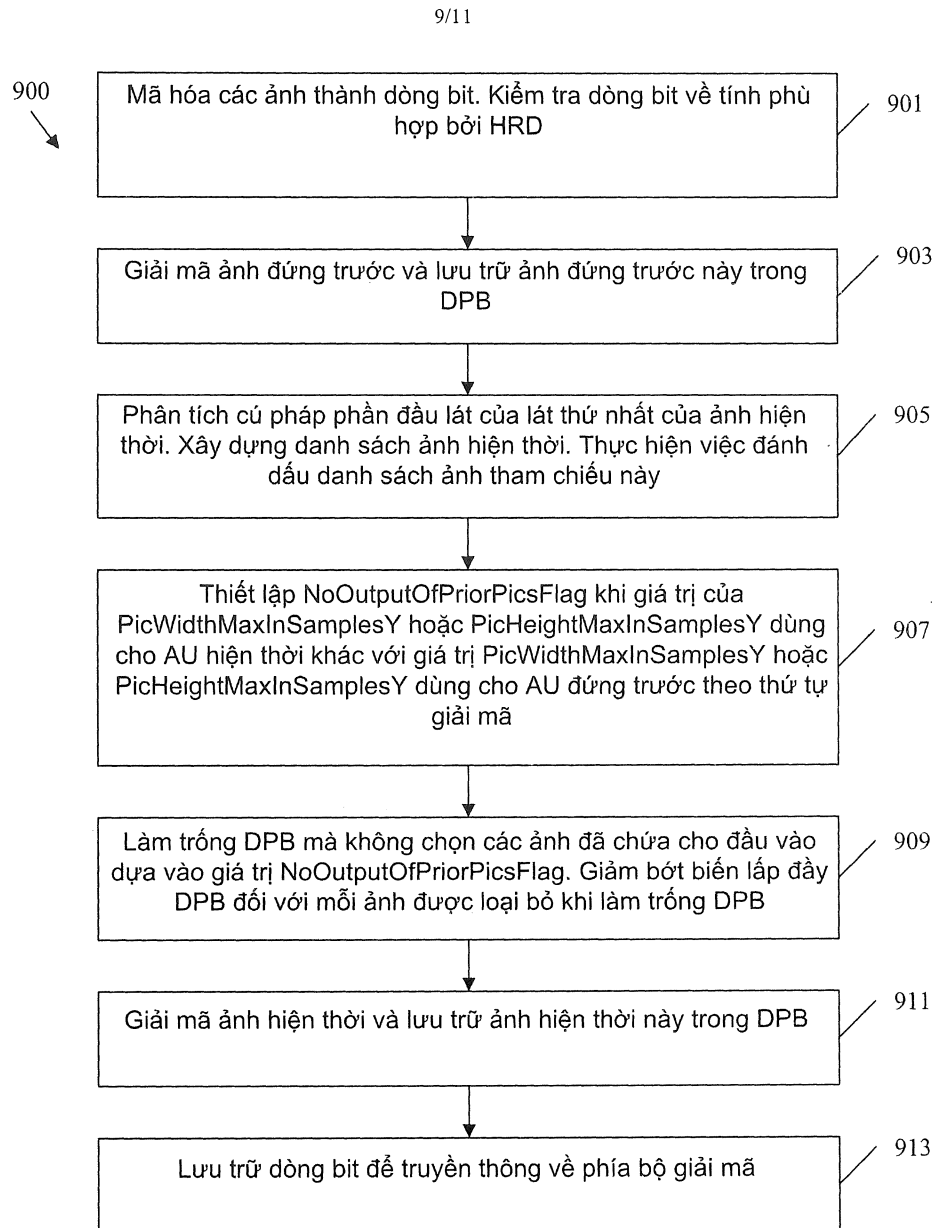


FIG. 9

10/11

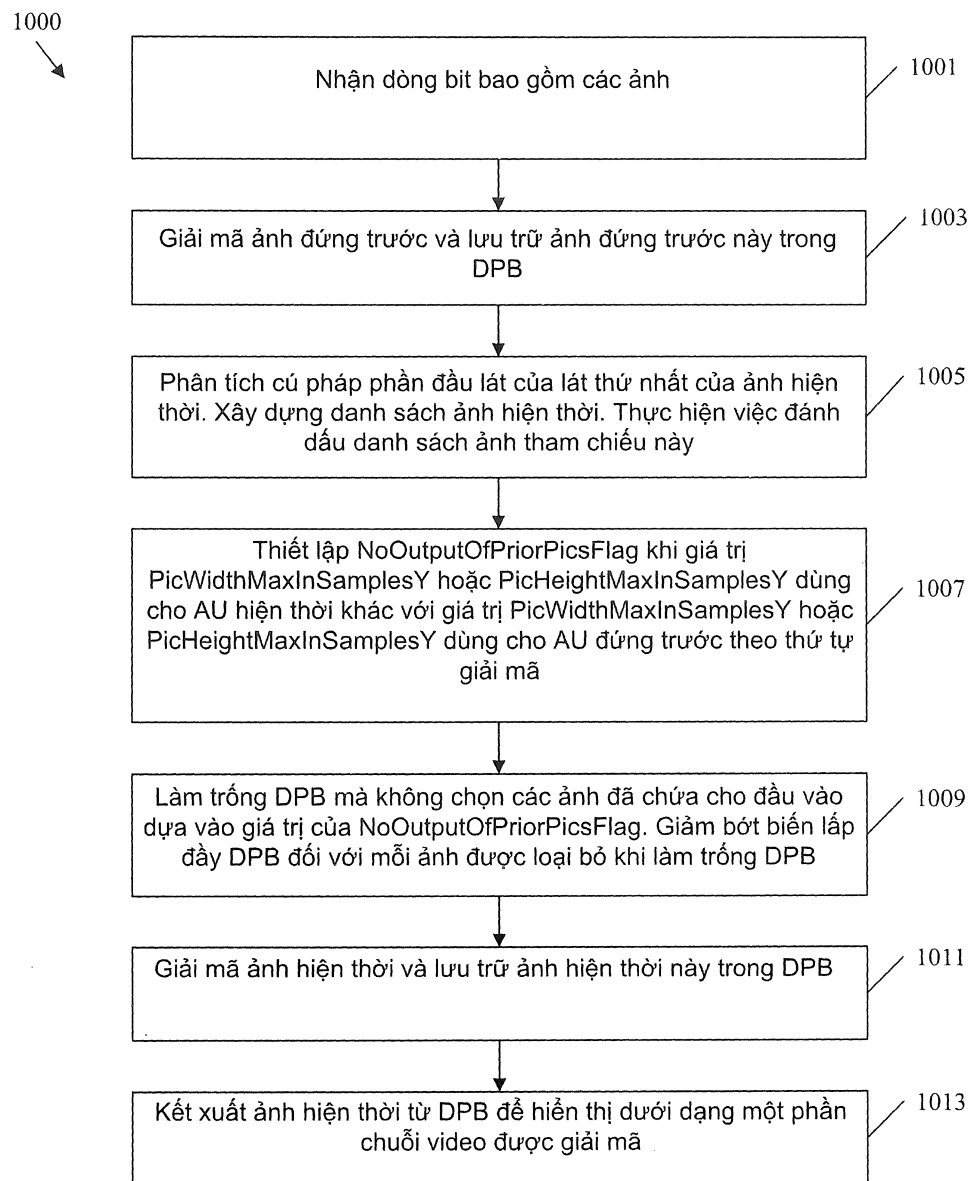


FIG. 10

11/11

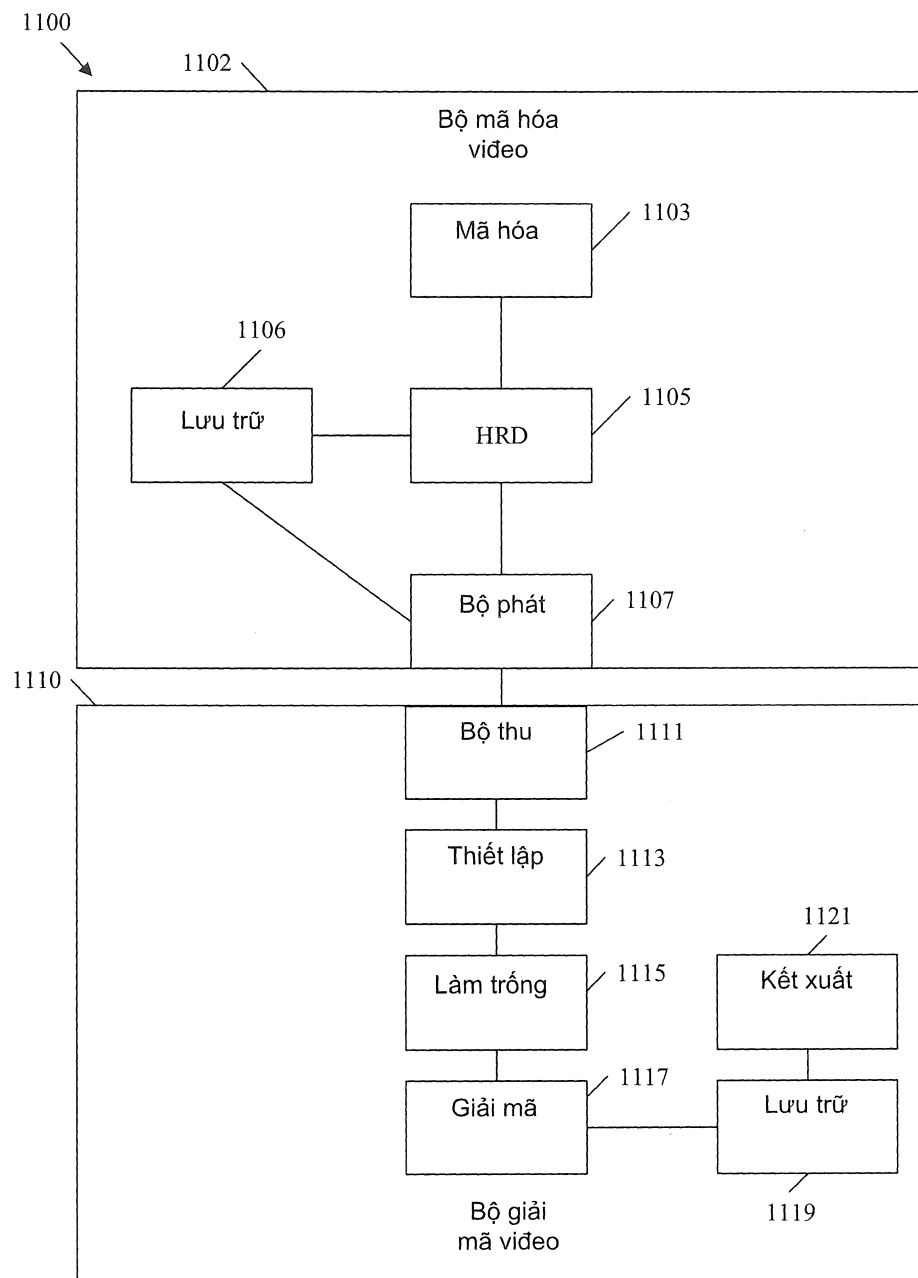


FIG. 11